

SCSI Toolbox, LLC
PO Box 620520
Littleton, CO 80162
<http://www.scsitoolbox.com/>
303.972.2072



SCSI *toolboxSuite*™

SATA · ATA · SAS · FC · SCSI · iSCSI

version 8.1

32&64BIT SUPPORT

Copyright © SCSI Toolbox, LLC All Rights Reserved
support: 303.972.2072
sales: 720.249.2641
www.scsitoolbox.com

COPYRIGHT NOTICE

Copyright 1997-2009 by SCSI Toolbox LLC (STB) All rights reserved. This item and the information contained herein are the property of STB. No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the express written permission of SCSI Toolbox LLC P.O. Box 620520, Littleton, Colorado 80162.

DISCLAIMER

STB makes no representation or warranties with respect to the contents of this document and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Further, STB reserves the right to revise this publication and to make changes in it from time to time without obligation of PTI to notify any person or organization of such revision or change.

SCSI toolbox32™ is a trademark of SCSI Toolbox LLC.

Contents

INSTALLING OR UPGRADING TO STB SUITE V8.1	13
APPENDIX A – PREPARING FOR V81	14
UPDATING YOUR USB KEY (DONGLE)	14
STEP 1 – UNZIP ALL INSTALLATION FILES.....	14
STEP 2 - INSTALLING THE LATEST ALADDIN SRM RUNTIME DRIVER PACKAGE	14
STEP 3 - CHECKING YOUR SRM INSTALLATION.....	14
STEP 4 – INSTALLING STB PACKAGE INFORMATION TO YOUR KEY – PART 1.....	15
STEP 5 - INSTALLING STB PACKAGE INFORMATION TO YOUR KEY – PART 2.....	18
STEP 6 - APPLYING PERFORMA INFORMATION TO YOUR KEY – PART 1	19
STEP 7 - INSTALLING STB PACKAGE INFORMATION TO YOUR KEY – PART 2.....	21
BUG FIXES AND ENHANCEMENTS.....	22
1. NEW ALADDIN SRM	22
2. ALADDIN ADMIN CONTROL CENTER (ACC)	22
3. CALLS TO ALADDIN HASP UNLINKED IN DTB	22
4. NEW DEVICE DISCOVERY PROCESS.....	22
5. DMM SEEK COMMANDS.....	22
STB (“ORIGINAL MODE”) NEW FEATURES	23
1. NEW SATA COMMANDS.....	23
2. SATA SMART SELF-TEST LOGS	24
3. SATA FEATURES DISPLAY AND CHANGE.....	25
YOU CAN CHANGE ANY FEATURE SETTING BY CLICKING ON THE SETTING YOU DESIRE. WHEN YOU DO THIS THE FEATURE COLOR WILL CHANGE TO RED INDICATING THAT YOU HAVE MODIFIED THE SETTING BUT HAVE NOT YET SAVED THE CHANGE.	26
4. NEW SATA TESTS.....	27
5. EXECUTE SMART SELF-TEST.....	27
6. SATA DRIVE CONFIDENCE TEST #1 (QUICK QC TEST).....	31
7. SATA DRIVE CONFIDENCE TEST #2	34
8. SATA COMMAND SEQUENCER	35
9. SATA FIRMWARE DOWNLOAD	44
10 – LOCK BOOT DRIVE OPTION	45
BAM NEW FEATURES.....	46
SAVE BAM CAPTURE DATA TO A SPREADSHEET FILE.....	46
DMM (DISK MANUFACTURING MODULE) NEW FEATURES & TESTS.....	47
1. SATA-SPECIFIC TEST STEPS	48
2. SATA SMART TEST TYPE.....	49
3. SATA INFO TEST TYPE.....	51
INTRODUCTION	52
INSTALLATION	52
WINDOWS 98 USERS.....	52
PLEASE NOTE!!!!	52

USING SAT TO ACCESS SATA DRIVES.....	53
<i>What is SAT?.....</i>	53
<i>How do I know that I need to use SAT?.....</i>	53
<i>Does my controller card support SAT?.....</i>	55
<i>A Simple test command.....</i>	56
<i>Issuing the ATA SMART command.....</i>	58
<i>Details of defining a SMART command.....</i>	61
<i>Conclusion.....</i>	62
STB SUITE VERSION 8.0 NEW FEATURES.....	63
PLEASE NOTE:.....	63
STB.....	63
New Features:.....	63
STB New STBTrace driver.....	63
SATA Firmware Download.....	64
SATA/ATA Command Compliance Testing.....	66
SES Device Configuration display.....	67
Bug Fixes/Enhancements:.....	69
DMM.....	70
New Features:.....	70
“STOP ON ERROR” OPTIONS IN DMM.....	70
ASYNCHRONOUS NOTIFICATIONS OPTIONS IN DMM.....	78
SATA FIRMWARE DOWNLOAD VIA DMM.....	79
Note: SAT requirement:.....	79
SCREENING ON SATA SMART THRESHOLDS FOR DMM.....	80
1. Screening on any SMART parameter approaching its threshold value.....	81
2. Screening on specific SMART parameter/data values.....	82
3. Screening using both methods.....	83
4. Log Results.....	84
DMM BUGS FIXED:.....	88
DEVELOPER TOOLBOX API (DTB).....	88
• ADDITIONAL FEATURES:.....	88
DTB BUGS FIXED:.....	88
BAM.....	88
NEW FEATURES:.....	88
STB SUITE VERSION 7.2 NEW FEATURES.....	89
STB.....	89
NEW FEATURES:.....	89
DISK WRITE/READ ADDED TO BUFFER FUNCTIONS.....	89
ATA/SATA USER DEFINED COMMANDS –.....	91
STB BUGS FIXED:.....	91
1. ATA USER DEFINED COMMAND INFINITE HANG FIXED.....	91
2. SHORT DST TEST FIXED.....	91
3. MODE SENSE W/O BLOCK DESCRIPTOR FIXED.....	91
DMM.....	92
NEW FEATURES:.....	92
TEST BY PERCENTAGE (%) OF DISK-.....	92
REMOVE ALL TESTS.....	92
MODIFY TEST STEP.....	92

NEW TEST STEPS	94
SAVE LOG PAGE PATH	97
EXTERNAL PROGRAM – PASS ARGUEMENTS.....	98
RUN AN EXTERNAL PROGRAM UPON ERROR.....	99
RESCAN BUSSES.....	100
DMM BUGS FIXED:	100
TMM.....	101
NEW FEATURES:	101
REMOVE ALL TESTS	101
MODIFY TEST STEP	102
RESCAN BUSSES.....	103
BUGS FIXED:.....	103
BAM	104
NEW FEATURES:	104
MORE ATA COMMAND DEFINITIONS ADDED TO COMMAND PHASE DISPLAY	104
SAVE CDB AND DRIVER FILTERS.....	104
DEVICE/BUS RESET CONTROL ADDED	104
BUGS FIXED:.....	104
MAIN MENU	106
OPTIONS.....	109
LOCK OUT (PROTECTING A DRIVE)	109
LOGGING OPTIONS	111
SETTINGS	111
<i>Safe Jukebox Mode</i>	111
<i>Print Reports</i>	112
<i>Specifiy Custom Fields</i>	113
QUICK COMMANDS	113
BUS RESET CONTROL	114
MANUAL RESET.....	115
RESET BY TIME	115
RESET BY I/OS.....	115
ATA TASK REGISTER COMMANDS	117
THE ATA IDENTIFY COMMAND.....	118
THE ATA SMART COMMAND.....	119
THE ATA USER-DEFINED COMMAND MENU	120
BUS ANALYZER MODULE.....	126
DESCRIPTION OF BAM	126
PRE-CAPTURE SETUP	127
<i>Identifying devices to capture</i>	127
<i>Identifying Phases to capture</i>	128
<i>Selecting Triggers</i>	129
<i>Setting Capture Buffer size and Phase Data Size</i>	130
<i>Identifying columns to display</i>	131
<i>Selecting the mode</i>	132

STARTING AND STOPPING CAPTURE	133
MONITORING WHILE TRACING	135
POST CAPTURE ANALYSIS	136
<i>Trace Performance Analysis</i>	136
<i>Data Analysis</i>	137
<i>Individual IO information</i>	139
<i>Raw Data View</i>	139
<i>IO Statistics</i>	140
<i>Check Conditions</i>	141
DISK TESTS.....	142
DISK TEST OPTIONS	142
TEST ALARMS	144
SEQUENTIAL READ	147
SEQUENTIAL WRITE.....	148
SEQUENTIAL WRITE/READ	149
MAP DRIVE DEFECTS	150
SEQUENTIAL WRITE/READ	151
RANDOM READ.....	152
RANDOM WRITE	153
RANDOM WRITE/READ	154
RANDOM WRITE/VERIFY	155
QUICK QC	156
QUICK DRIVE PROFILE.....	157
<i>Drive Profile File Output</i>	158
AV/DATA PERFORMANCE COMPARISON	159
<i>Drive Optimization Test Dialog</i>	160
DRIVE CONFIDENCE TEST 1	161
DRIVE CONFIDENCE TEST 2	161
SEEK TESTS	163
PERFORMANCE TESTS.....	164
HP MO (MAGNETO OPTICAL) TESTS.....	165
BLOCKING FACTOR TESTS	167
BUS LOAD TEST	169
DRIVE SELF TEST.....	172
DISK ZONE TEST	174
CD PERFORMANCE TEST.....	177
DRIVE SCREEN/VERIFY TEST	178
HIGH SPEED DATA TEST.....	179
DISK COMMANDS.....	180
INQUIRY WITH EVPD.....	181
START DRIVE (MOTION COMMANDS)	181
STOP DRIVE (MOTION COMMANDS).....	182
START ALL (MOTION COMMANDS).....	182
STOP ALL (MOTION COMMANDS).....	182
READ CAPACITY (CAPACITY/BLOCK FUNCTIONS)	183
CHANGE CAPACITY (CAPACITY/BLOCK FUNCTIONS).....	183
CHANGE BLOCKSIZE (CAPACITY/BLOCK FUNCTIONS)	184
CHANGE DEFINITION.....	184
VIEW/EDIT MODE PAGES (MODE/LOG PAGE FUNCTIONS)	184
VIEW ALL MODE PAGES (MODE/LOG PAGE FUNCTIONS)	187
RESTORE DEFAULT MODE PAGES.....	187
VIEW LOG PAGES (MODE/LOG PAGE FUNCTIONS)	188
OPTIMIZE DRIVE FOR A/V USE	190
OPTIMIZE DRIVE FOR DATA USE.....	190

DOWNLOAD FIRMWARE	191
FORMAT	192
CORRUPT DATA	194
TRANSLATE ADDRESS	194
USER DEFINED CDB	195
<i>Defining or Modifying a SCSI CDB</i>	195
CDB SEQUENCER	197
BUILD RUN SCRIPT	200
WORKSTATION PREP.....	208
THE SUN WORKSTATION MENU.....	210
<i>Read Existing Label</i>	210
<i>Write New Label</i>	210
<i>Calculate Format.dat Entry</i>	211
<i>Edit Sun Partition Table</i>	212
THE SGI WORKSTATION MENU	212
<i>Read Current Label</i>	212
<i>Write Label</i>	213
THE HP WORKSTATION MENU	214
THE DEC WORKSTATION MENU.....	215
MEDIA DUPLICATION	217
TAPE TESTS.....	220
TAPEWATCH.....	221
QUICK QC TEST.....	224
SEQUENTIAL READ TEST	225
SEQUENTIAL WRITE TEST	226
DATA INTEGRITY TEST	227
MAP A TAPE'S CONTENTS.....	228
CONFIRM DEVICE (INQUIRY).....	229
QUANTUM DLT SELF TEST	230
DLT RECEIVE DIAGNOSTIC RESULTS	230
EXABYTE WARM UP TEST.....	231
DDS (4MM) MEDIA ERROR RATE TEST.....	232
DDS (4MM) DRIVE ERROR RATE TEST	233
HIGH SPEED DATA TEST.....	234
HIGH PERFORMANCE MEASUREMENT TEST	235
TAPE COMMANDS.....	236
INFORMATION FUNCTIONS – TEST UNIT READY	236
INQUIRY	237
INQUIRY WITH EVPD	237
VIEW MOST RECENT SENSE DATA	238
MODE PAGE EDIT	239
DLT LOG PAGES	239
DDS LOG PAGES	239
VIEW LOG PAGES.....	240
MOTION/DATA FUNCTIONS.....	240
<i>rewind</i>	241
<i>unload</i>	241
<i>space</i>	241
<i>read position</i>	241
<i>Locate</i>	241
<i>write filemark</i>	241
<i>write setmark</i>	242

<i>flush buffer</i>	242
<i>read block(s)</i>	242
<i>write block(s)</i>	242
USER DEFINED CDB	243
<i>Definining or Modifying a SCSI CDB</i>	243
CDB SEQUENCER	245
BUILD/RUN SCRIPT.....	248
EDIT BUFFER 1/ EDIT BUFFER 2	248
CONFIGURE TAPE DRIVE	250
DOWNLOAD FIRMWARE.....	250
<i>Segmented Download</i>	252
<i>Non-Segmented Download</i>	253
<i>DLT Firmware Download</i>	254
<i>SuperDLT Download</i>	254
<i>STK 9840 Download</i>	255
<i>HP LTO Download</i>	255
<i>Seagate LTO Download</i>	256
<i>IBM LTO Download</i>	256
<i>Sony AIT & DDS Download</i>	257
REPAIR DLT TAPE DIRECTORY	257
CREATE DLT FUP (FIRMWARE UPDATE) TAPE	258
LTO CARTRIDGE MEMORY DISPLAY	259
JUKEBOX TESTS	260
RANDOM LIBRARY TEST.....	261
CONFIRM DEVICE	262
HP JUKE TESTS	262
TOTAL JUKEBOX TEST	263
JUKEBOX FULL RANGE TEST	264
JUKEBOX COMMANDS	265
INITIALIZE ELEMENT STATUS	266
READ ELEMENT STATUS	267
POSITION TO ELEMENT	268
MOVE MEDIA.....	269
INQUIRY	270
INQUIRY WITH EVPD	270
TEST UNIT READY	270
VIEW REQUEST SENSE	271
MODE PAGE EDIT.....	272
VIEW LOG PAGES.....	273
DOWNLOAD FIRMWARE.....	274
<i>HP Firmware Download</i>	274
<i>Segmented Download</i>	275
<i>Non-Segmented Download</i>	276
HP JUKEBOX MODE PAGES	277
USER DEFINED CDB	278
<i>Definining or Modifying a SCSI CDB</i>	278
CDB SEQUENCER	280
BUILD/RUN SCRIPT.....	283
EDIT BUFFERS	284
SCSI TUTORIAL	288
INTRODUCTION TO SCSI PHILOSOPHY.....	288
<i>SCSI Hardware</i>	288
<i>SCSI Software</i>	289

Command Descriptor Blocks.....	291
SCSI Bus Operation–An Example	292
Miscellaneous SCSI Issues.....	294
Simple Peripheral Installation Guidelines.....	296
TROUBLESHOOTING SCSI TOOLBOX32 PROBLEMS	298
REQUEST SENSE & SENSE KEY INTERPRETATION	300
ADDITIONAL SENSE CODE/ QUALIFIER INTERPRETATION	301
SCSI COMMAND DEFINITIONS	304
THE DISK MANUFACTURING & SCREENING MODULE INTRODUCTION	307
TEST METHODOLOGY	307
PRE-TEST ACTIONS.....	307
TESTING	308
POST-TEST ACTIONS.....	309
GETTING STARTED	309
PRE-TEST ACTION CONFIGURATION	310
Vendor	311
Product	311
Version	311
Blocksize	312
Capacity.....	312
P-Defects.....	312
G-Defects	312
MODE PAGES	313
Spin Up All Drives	313
Clear Log Pages.....	313
Save MODE PAGES to database	313
TEST SETUP.....	314
Random or Sequential Access	314
Random	314
Sequential.....	315
Type of Test.....	315
Verify	315
Write	315
Write with Verify	315
Read.....	315
Write/Read.....	316
FW Download.....	316
Format	316
Stop Test After	316
Time.....	316
Blocks.....	316
Data Patterns	316
13 different data patterns are available for testing – and each pattern can optionally have the data block overlaid, making a total of 26 data patterns ! All Zeros	316
All Ones	316
Alternating (0/1).....	317
Alternating (1/0).....	317
Incrementing.....	317
Decrementing	317
Block Number.....	317
Random.....	317
User Defined.....	317
Walking Ones	317

<i>Walking Zeros</i>	317
<i>Alt (0/1) then Alt (1/0)</i>	317
<i>Alt (1/0) then (0/1)</i>	317
COMPARE ON READS	318
<i>Overlay Block Number</i>	318
ADVANCED OPTIONS	318
POST-TEST ACTIONS	319
SELECTING DISKS TO TEST	319
SAVING AND LOADING TEST CONFIGURATIONS	321
<i>Saving a Test Configuration</i>	321
<i>Loading a Test Configuration</i>	323
STARTING THE TEST PROCESS	323
DURING TESTING – INDIVIDUAL DRIVE	323
<i>Pausing a Test</i>	324
<i>Resuming a Test</i>	324
<i>Stopping a Test</i>	324
VIEWING TEST STATUS	324
DURING TESTING – ALL DRIVES	325
THE TEST STATUS PAGE	326
<i>The Drives Pulldown</i>	326
<i>The Tests Pulldown</i>	328
STOPPING TESTS	328
TEST RESULTS AND OUTPUT DATA	329
<i>Database data</i>	329
<i>Text File Output Data</i>	331
<i>Other output data</i>	332
THE TAPE MANUFACTURING & SCREENING MODULE INTRODUCTION	337
TEST METHODOLOGY	337
<i>Pre-Test Actions</i>	337
<i>Testing</i>	339
<i>Post-Test Actions</i>	340
GETTING STARTED	341
PRE-TEST ACTION CONFIGURATION	341
<i>Vendor</i>	343
<i>Product</i>	343
<i>Version</i>	343
<i>Compression</i>	344
<i>Rewind All Drives</i>	344
<i>Clear Log Pages</i>	345
<i>Save LOG PAGES to database</i>	345
TEST SETUP	346
<i>Type of Test</i>	347
<i>Write</i>	347
<i>Read</i>	347
<i>Write Filemark</i>	347
<i>Read Filemark</i>	347
<i>Space Forward Block</i>	347
<i>Space Forward FileMark</i>	347
<i>Space Forward to EOD</i>	347
<i>Space Reverse Block</i>	347
<i>Space Reverse FileMark</i>	347
<i>Rewind</i>	348
<i>FW Download</i>	348
<i>Synchronize</i>	348
<i>External Program</i>	348

STOP TEST AFTER	348
<i>Time</i>	348
<i>MB – Megabytes transferred</i>	348
DATA PATTERNS	349
<i>13 different data patterns are available for testing – and each pattern can optionally have the data block overlaid, making a total of 26 data patterns ! All Zeros</i>	349
<i>All Ones</i>	349
<i>Alternating (0/1)</i>	349
<i>Alternating (1/0)</i>	349
<i>Incrementing</i>	349
<i>Decrementing</i>	349
<i>Block Number</i>	349
<i>Random</i>	349
<i>User Defined</i>	349
<i>Walking Ones</i>	349
<i>Walking Zeros</i>	349
<i>Alt (0/1) then Alt (1/0)</i>	350
<i>Alt (1/0) then (0/1)</i>	350
COMPARE ON READS.....	350
POST-TEST ACTIONS.....	350
SELECTING TAPES TO TEST	351
SAVING AND LOADING TEST CONFIGURATIONS	351
<i>Saving a Test Configuration</i>	351
<i>Loading a Test Configuration</i>	353
STARTING THE TEST PROCESS	353
DURING TESTING – INDIVIDUAL DRIVE	353
<i>Pausing a Test</i>	353
<i>Resuming a Test</i>	354
<i>Stopping a Test</i>	354
<i>Viewing Test Status</i>	354
DURING TESTING – ALL DRIVES	355
<i>The Test Status Page</i>	355
<i>The Drives Pulldown</i>	357
<i>The Tests Pulldown</i>	358
STOPPING TESTS	358
TEST RESULTS AND OUTPUT DATA	359
<i>Database data</i>	359
<i>Text File Output Data</i>	361
<i>Other output data</i>	363
ATA COMMAND COMPLIANCE TESTING	368
INTRODUCTION	369
REQUIREMENTS.....	369
<i>Hardware Requirements</i>	369
<i>User Requirements</i>	370
TEST METHODOLOGY	371
NORMAL AND ERROR STATUS.....	371
<i>Note:</i>	371
EXCEPTIONS	371
<i>Idle and Standby</i>	371
<i>Sleep</i>	371
<i>Security Freeze</i>	371
DETERMINING THE DRIVE'S CAPABILITIES AND STATE	372

THE STB ATA/SATA COMMAND COMPLIANCE TEST	373
TEST EXECUTION	374
CUSTOMIZING THE TEST SOURCE CODE	377
<i>Opening the project</i>	377
<i>Adding your own commands</i>	377
<i>Defining ATA/SATA commands</i>	377
<i>Special command sequence issues</i>	377
<i>Commands issued during Non-Destructive test</i>	378
USING BAM TO CONFIRM TESTING	379
THE NEXT STEP – AUTOMATING ATA/SATA COMMAND COMPLIANCE TESTING IN THE STB SUITE DISK MANUFACTURING MODULE (DMM).....	381
SUMMARY	383
FURTHER INFORMATION.....	383

Installing or Upgrading to STB Suite v8.1

Follow these easy steps to install the STB Suite version 8.1

1. Uninstall any older or existing version of the STB Suite
 - a. During the uninstall, right after the STBTrace driver is uninstalled, you will get a message saying “System Settings Change – Reboot Now?” – You *must* answer “**NO**” at this point to allow the uninstall to completely finish.
2. Prepare your USB key dongle for the upgrade
 - a. Plug your USB dongle in and be sure its red light is on
 - b. You may see pop-up messages from the operating system saying there is a new device detected and asking to install drivers. Cancel this – the next steps will install the correct drivers for the dongle.
 - c. Follow the directions in Appendix A - ***PreparingForV81***
3. Leave your USB dongle key plugged in and be sure its red light is on
4. Unzip the STBv81 Install package
5. Run the program CkInstall.exe by double-clicking on it. This program will confirm that your installation package is not corrupt, and will confirm that your system DEP settings are correct. **Please Note:** DEP settings cannot be confirmed by the program for the operating systems Vista or Server 2008. If you are using one of these operating systems please confirm proper DEP settings manually. It is very important that DEP be set to “Turn on for Windows programs and services only”
 - a. Confirm that the md5 checksum is valid. If it is not valid contact support@scsitolbox.com
 - b. Confirm that your DEP settings are correct.
6. Install the STBTrace driver package now
 - a. Double-click on the program ***InstallStbtrace.exe***, follow all on-screen instructions
 - b. Reboot when the installation is finished
7. Install the STB Suite
 - a. Double-click on the program ***HWK_GetHasp_STBv8_1_0Install***
 - b. Follow all on-screen instructions
 - c. If you are running on a Vista or Server 2008 you may receive a message stating that your DEP settings are incorrect or cannot be determined. If you see this message you must confirm that your DEP settings are correct, then you may tell the install to proceed.
 - d. **Caution** – if your DEP settings are not correct there will be problems running the STB Suite.
 - e. Reboot when the installation is finished

Appendix A – Preparing For v81

Updating your USB Key (Dongle)

Several steps are required to update your STB USB license key to prepare for installation of the new STB Suite version 8.1. These steps will be outlined below – **it is very important that these steps be executed in the exact order described.** Doing so will insure uninterrupted use and open up access to all the new features of version 8.1

Step 1 – unzip all installation files.

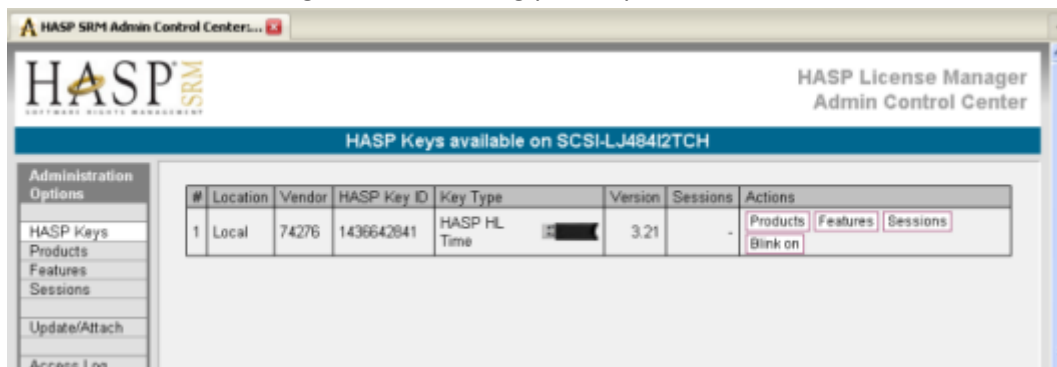
- Unzip the files “STBSuitev81.exe” and “STBCustomerKeyUpdateVer81.exe”

Step 2 - Installing the latest Aladdin SRM Runtime driver package

- Go to ftp://ftp.aladdin.com/pub/hasp/srm/Runtime_%28Drivers%29/HASP_SRM_Runtime_setup.zip
- Install this package following all on-screen instructions

Step 3 - Checking your SRM installation

- Point your web browser to <http://localhost:1947>
- You should see the Aladdin Admin Control Center (ACC) displayed. If you do not see the ACC go back to step 1 and insure that you have properly installed the SRM runtime package
- Be sure that your STB USB key (dongle) is plugged in to a USB port and that its red LED is lit.
- In the ACC click on the HASP Keys choice under Administration Options at the left of the screen.
- You should see something like this, showing your key information:

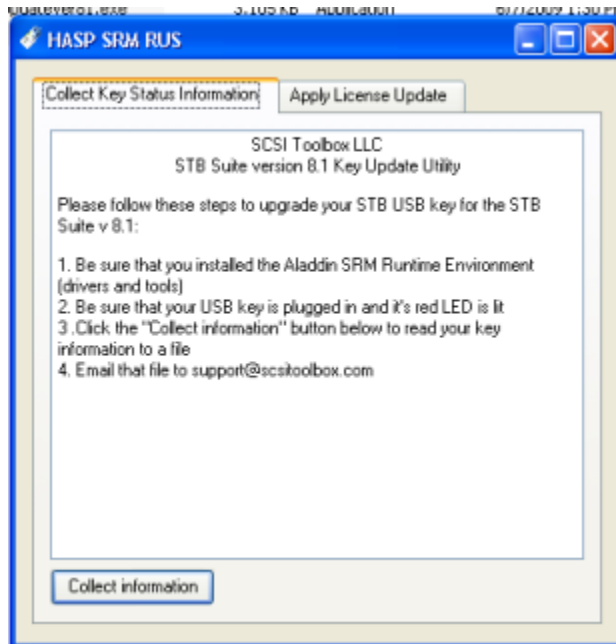


- Write down the HASP Key ID number

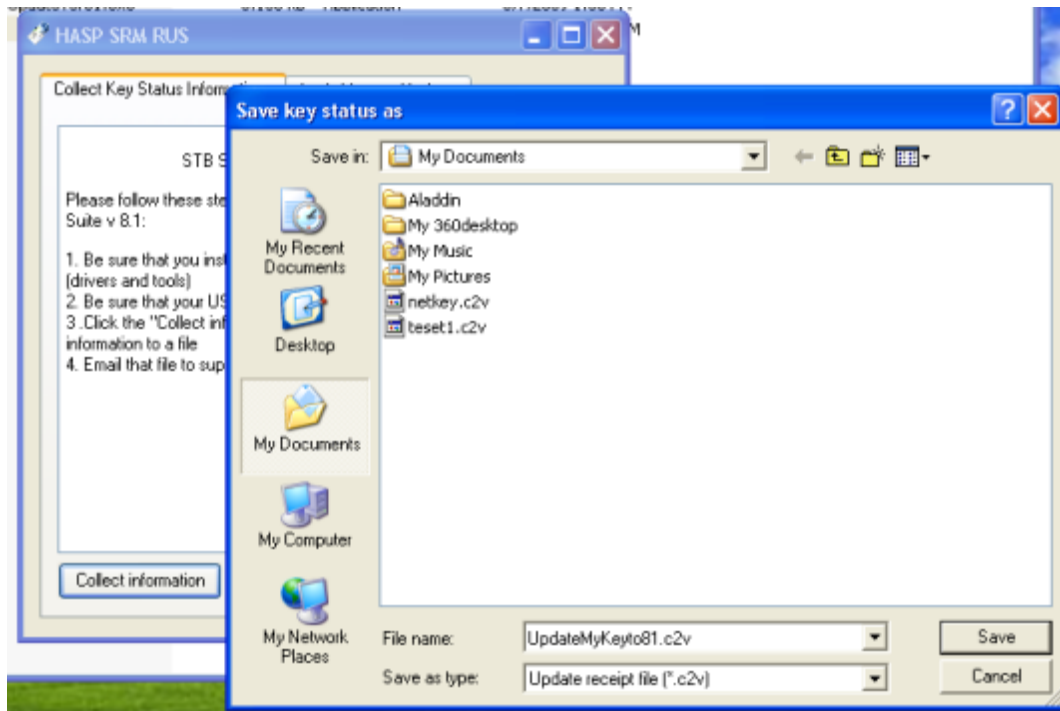
- Note that the Version is 3.20 or above. If it is not please contact support@scsitolbox.com
- Note that the Key Type is a HASP HL Time or HASP HL Net. If it is not please contact support@scsitolbox.com

Step 4 – Installing STB Package information to your key – Part 1

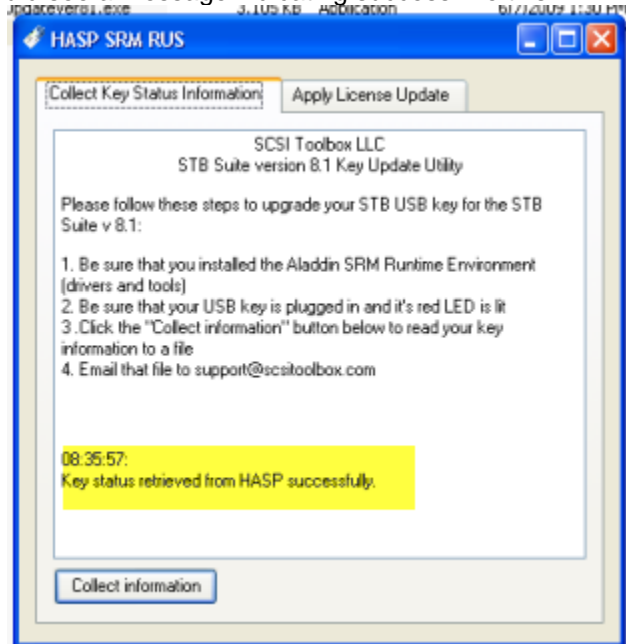
- Run the program “STBSuitev81.exe”
- You will see this screen:



- Click the “Collect Information” button
- Tell the program where to create your update file



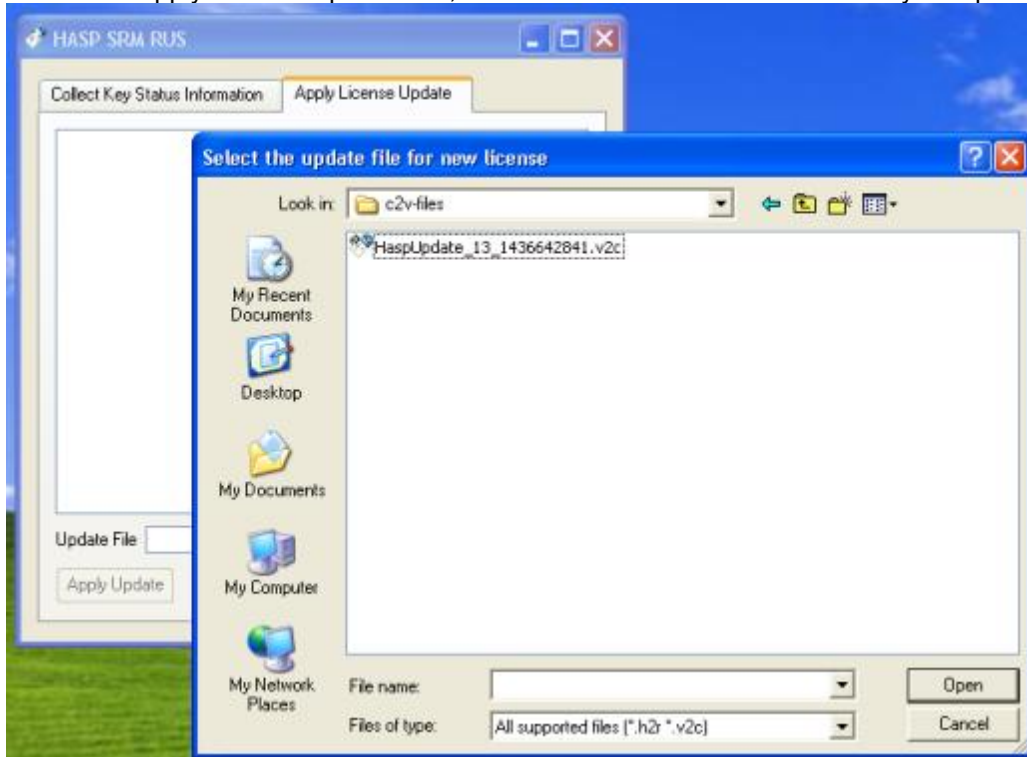
You should see a message indicating success like this:



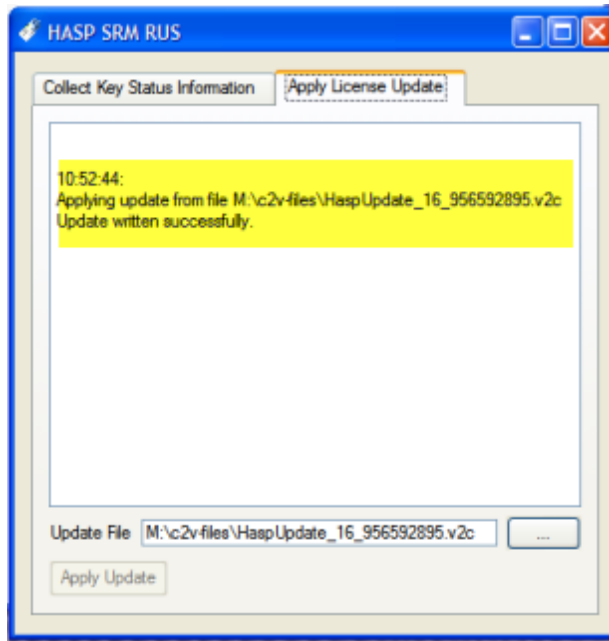
-
- Email the file you just created to support@scsitoolbox
- In the body of the email where you send this file please record the HASP Key ID number which you wrote in step 1

Step 5 - Installing STB package information to your key – Part 2

- when you receive the return email from support save the attached update file into the folder where the program “STBSuiteV81.exe” resides
- run “STBSuiteV81.exe” again
- Click the “Apply License Update” tab, then use the browse button to locate your update file

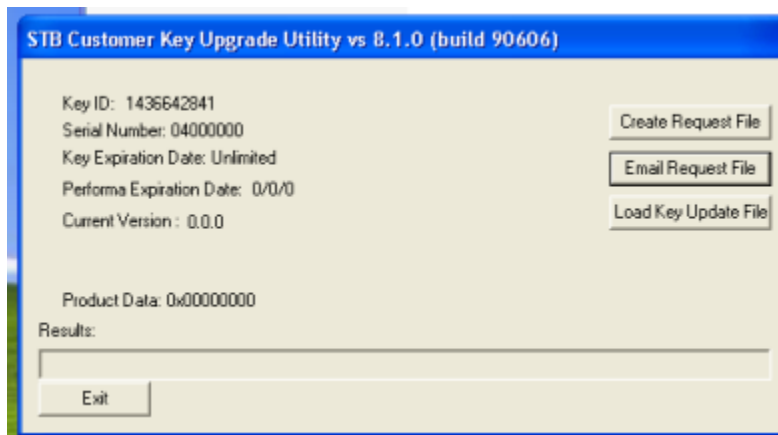


- Click “Apply Update” and you should see an indication of success like this:

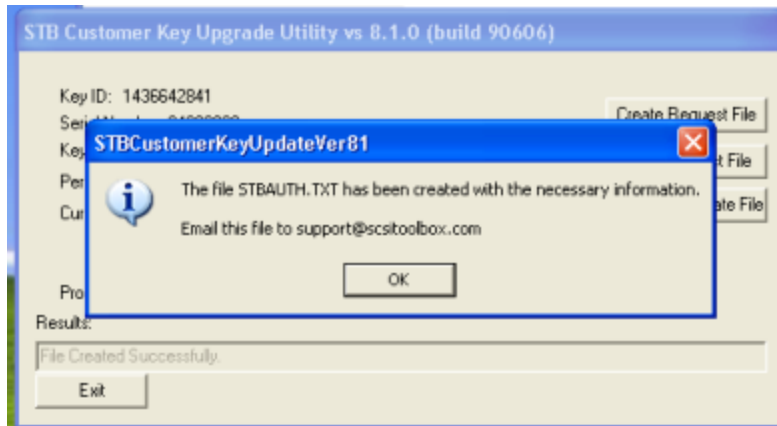


Step 6 - Applying Performa information to your key – Part 1

- Now start the STB Performa key update program by double-clicking on “STBCustomerKeyUpdateVer81.exe”
- You will see this screen:



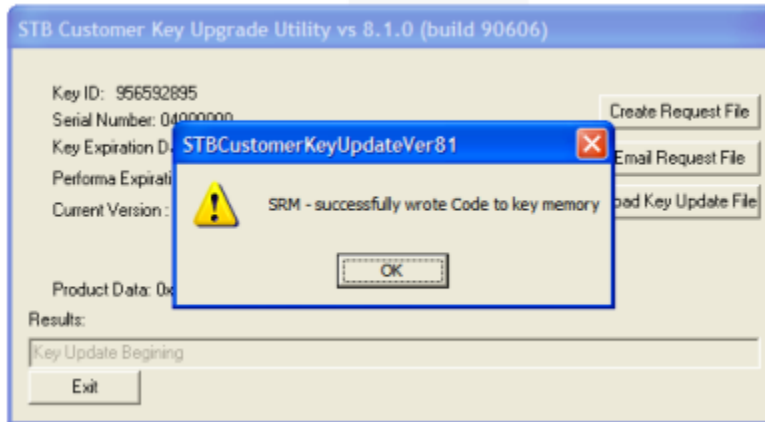
- Click on “Create Request File” – and the screen will change to



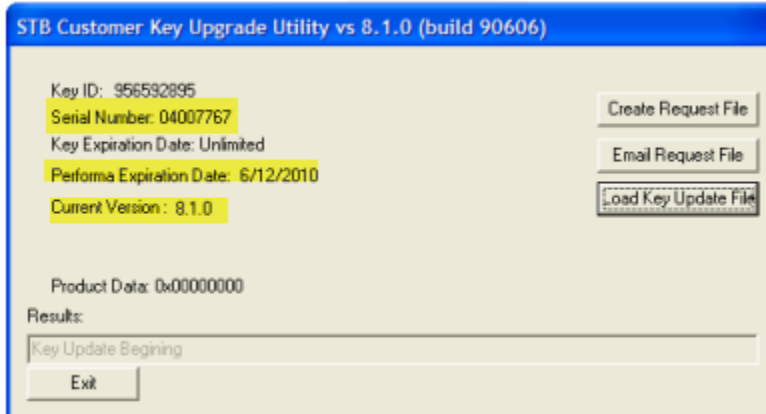
-
- Email the file "STBAuth.txt" to support@scsitolbox.com.
- Exit the STBSuiteV81.exe program

Step 7 - Installing STB Package information to your key – part 2

- Save the update file returned to you from STB Support
- Restart the program “STBSuiteV81.exe”
- Click the button “Load Key Update File”, and use the browse feature to point to your update file which was just sent to you from support.
- You should see this screen:



- Click “OK” and you should now see your valid Serial Number, Performa date and version information like this:



- If you do not see a screen indicating success please contact support@scsitolbox.com

If all of the above steps have been successfully executed then you are now ready to install the STB Suite version 8.1

Bug Fixes and Enhancements

1. New Aladdin SRM package greatly simplifies the install process. Now only one Aladdin driver is needed for all operating systems and environments (networked, remoted, etc)

2. Aladdin Admin Control Center (ACC) gives simple view of key status and gives one point to configure network settings, free stuck licenses, etc.

Simply use your internet browser and point to <http://localhost:1947> to see all STB Keys on your system(s), including networked keys. Here is an example showing a network key on a license server:

The screenshot displays the HASP License Manager Admin Control Center interface. The main heading is "HASP Keys available on HAM". On the left, there is a navigation menu with options like "Administration Options", "HASP Keys", "Products", "Features", "Sessions", "Update/Attach", "Access Log", "Configuration", "Diagnostics", "Help", and "About". The main content area shows a table of keys with columns for #, Location, Vendor, HASP Key ID, Key Type, Version, Sessions, and Actions. Below this, a sub-section titled "Net Features of HASP 1646500067 (Vendor: 74276) at HP-64" contains another table with columns for #, Feature ID, Location, Access, Counting, Logins, Limit, Detached, Restrictions, Sessions, and Actions. A "Help Top" link is visible at the bottom right of the interface.

#	Location	Vendor	HASP Key ID	Key Type	Version	Sessions	Actions
1	Local	74276	[REDACTED]	HASP SL	1.40	-	<input type="button" value="Features"/> <input type="button" value="Sessions"/>
2	HP-64	74276	[REDACTED]	HASP HL NetTime 10	3.21	-	<input type="button" value="Browse"/> <input type="button" value="Net Features"/>

#	Feature ID	Location	Access	Counting	Logins	Limit	Detached	Restrictions	Sessions	Actions
1	0	HP-64	Loc Net Display	Station	-	10	-	Perpetual	-	<input type="button" value="Browse"/>
2	1	HP-64	Loc Net Display	Station	-	4	-	Perpetual	-	<input type="button" value="Browse"/>
3	2	HP-64	Loc Net Display	Station	-	4	-	Perpetual	-	<input type="button" value="Browse"/>
4	3	HP-64	Loc Net Display	Station	-	4	-	Perpetual	-	<input type="button" value="Browse"/>
5	4	HP-64	Loc Net Display	Station	-	4	-	Perpetual	-	<input type="button" value="Browse"/>

You can see the number of allowed sessions and attached sessions, etc.

3. Calls to Aladdin HASP unlinked in DTB

This allows your DTB applications to run without a HASP key connected as long as no DTB functions are called.

4. New Device Discovery process

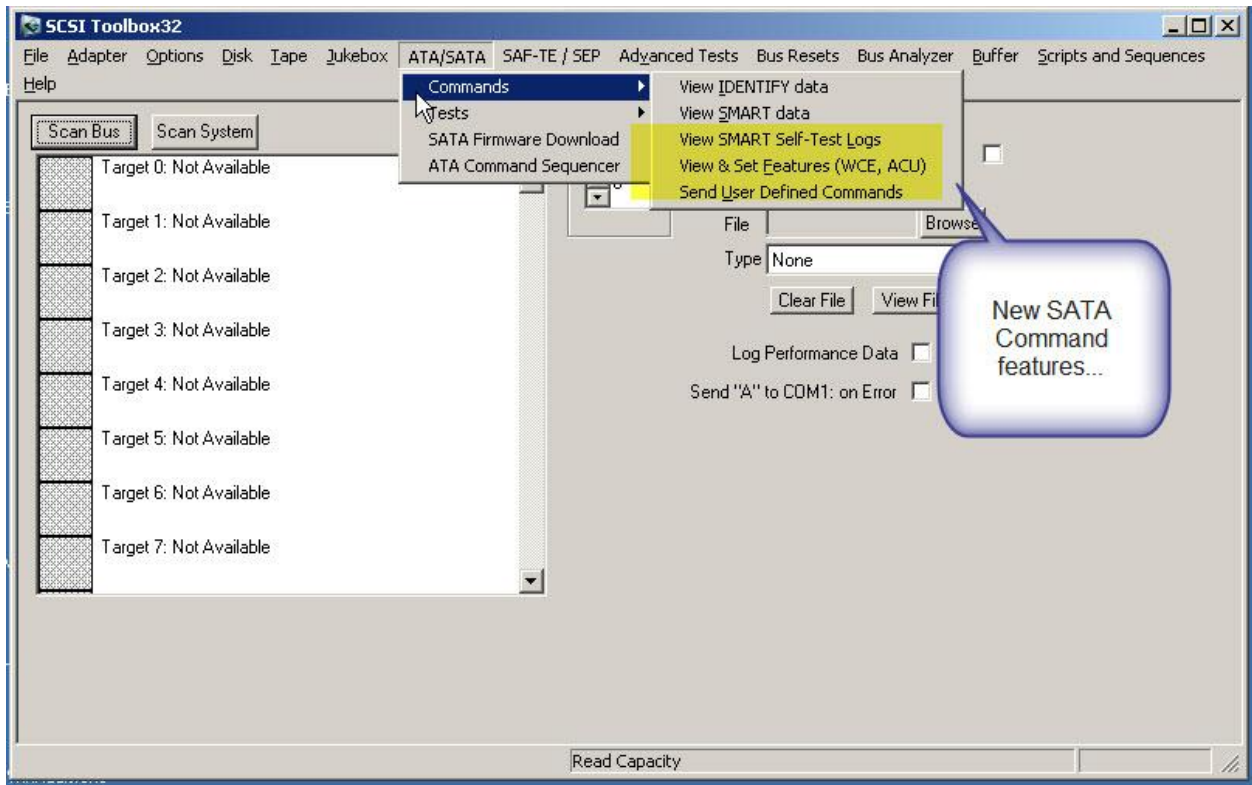
Achieves higher accuracy and correlation between Windows Device Manager, STB, DMM, BAM, and DTB device names and views.

5. DMM Seek commands

The SEEK commands have been replaced with Single-Block READ commands to allow SEEK tests to be run on SATA drives.

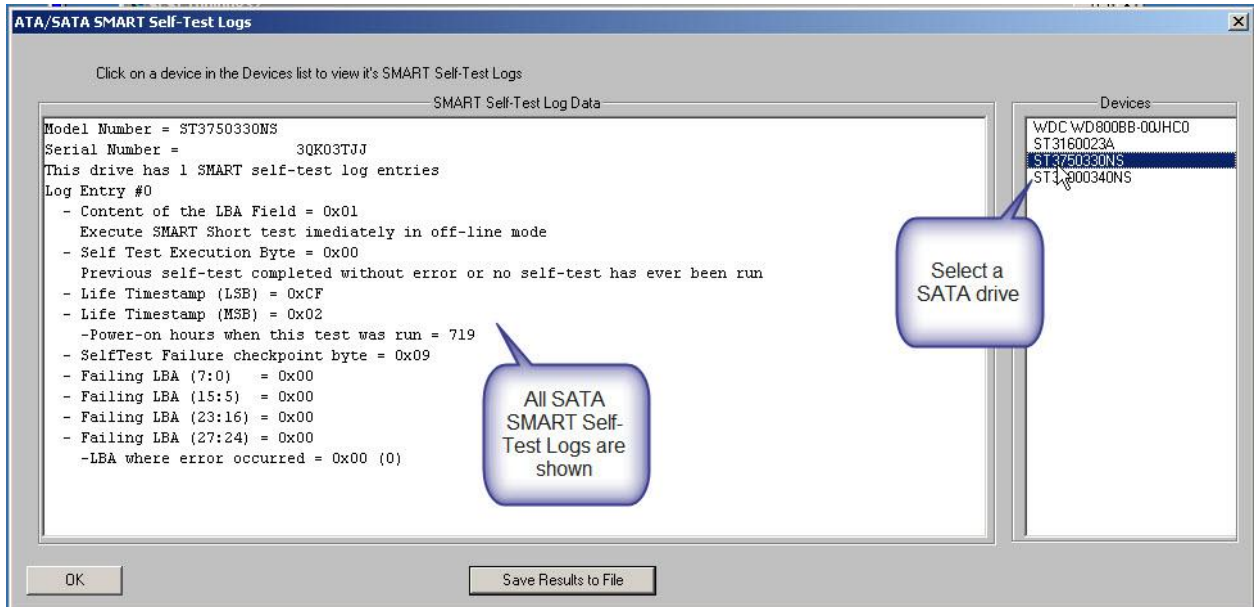
STB (“Original Mode”) new features

1. New SATA Commands



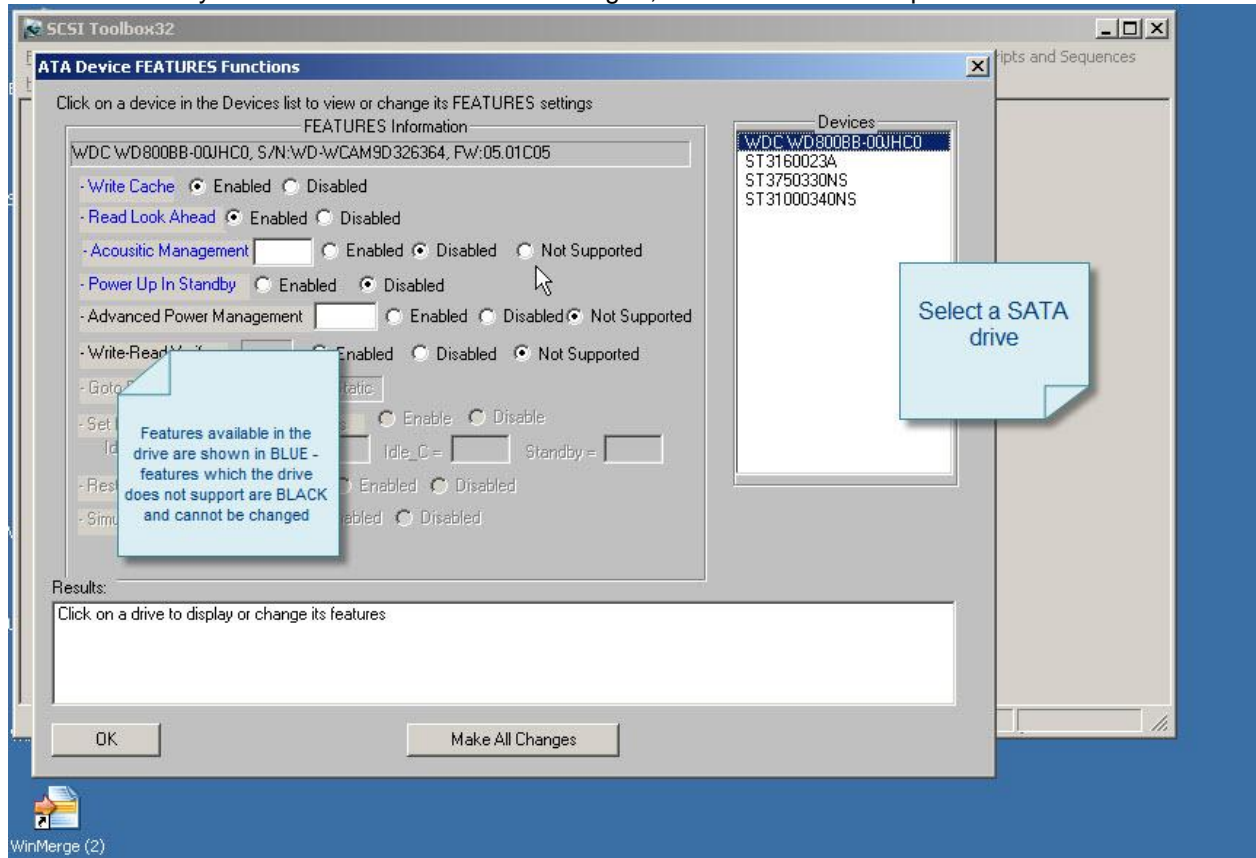
2. SATA SMART Self-Test Logs

This command lets you view and save the results of all SMART Self-Tests on a per-drive basis.



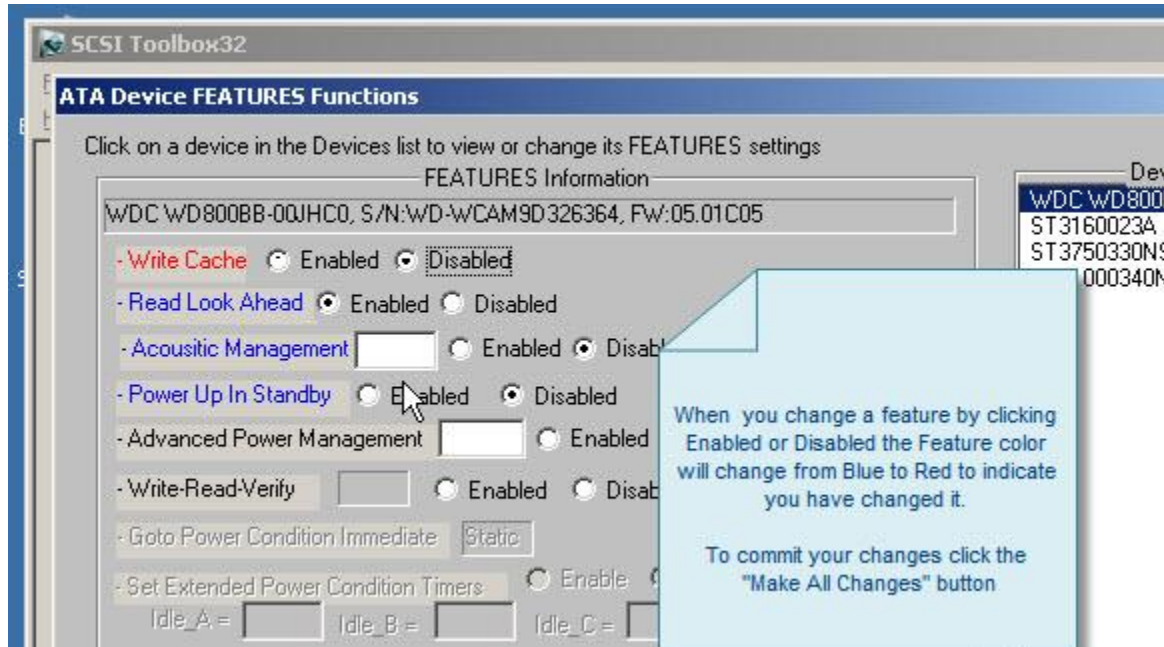
3. SATA Features Display and Change

This command will display all of the SATA FEATURES which are available on a selected disk, and will allow any available FEATURES to be changed, set or reset via a simple menu:



If a FEATURE from the SATA specification is enabled in the selected drive it will be highlighted in **BLUE** and the current setting will be displayed showing if the FEATURE is Enabled or Disabled. If there is a value associated with a given FEATURE that value will also be displayed.

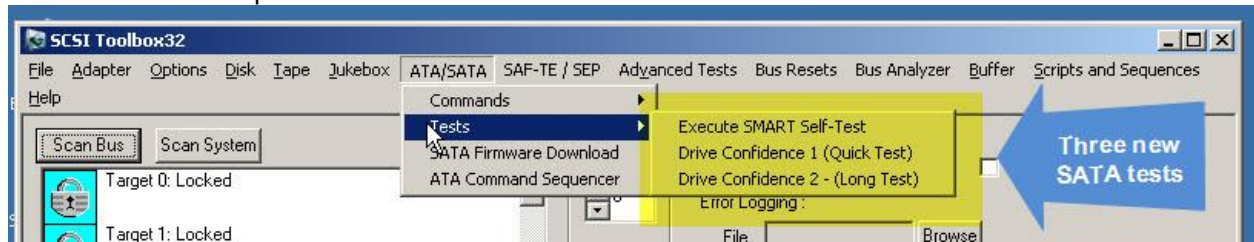
You can change any FEATURE setting by clicking on the setting you desire. When you do this the FEATURE color will change to **RED** indicating that you have modified the setting but have not yet saved the change.



To save the change you must click the "Make All Changes" button. Once the changes have been written to the drive they will be re-read and displayed. If your selected drive does not support any FEATURES the non-supported FEATURES will be displayed in a BLACK font. Which FEATURES are supported or not is up to the drive Vendor and is not changeable.

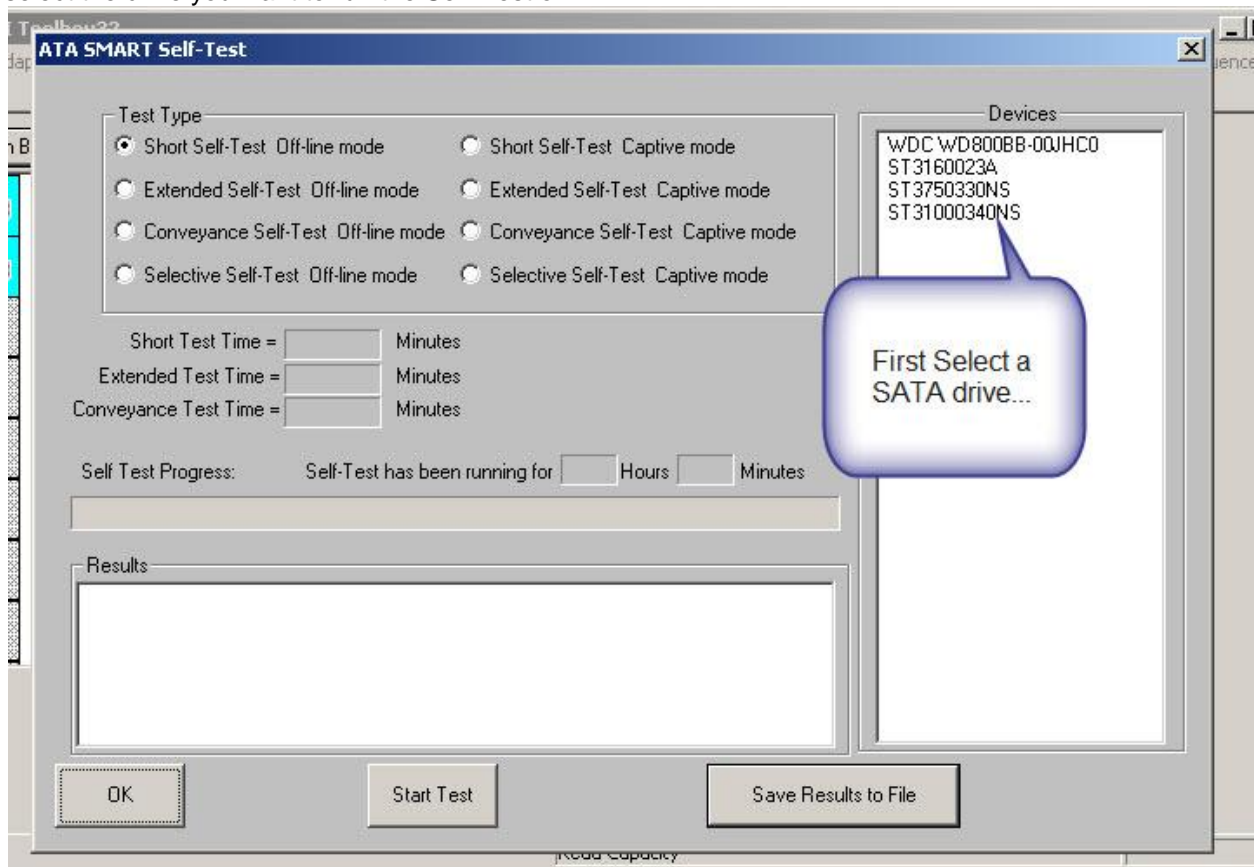
4. New SATA Tests

Several new SATA-Specific tests have been added:

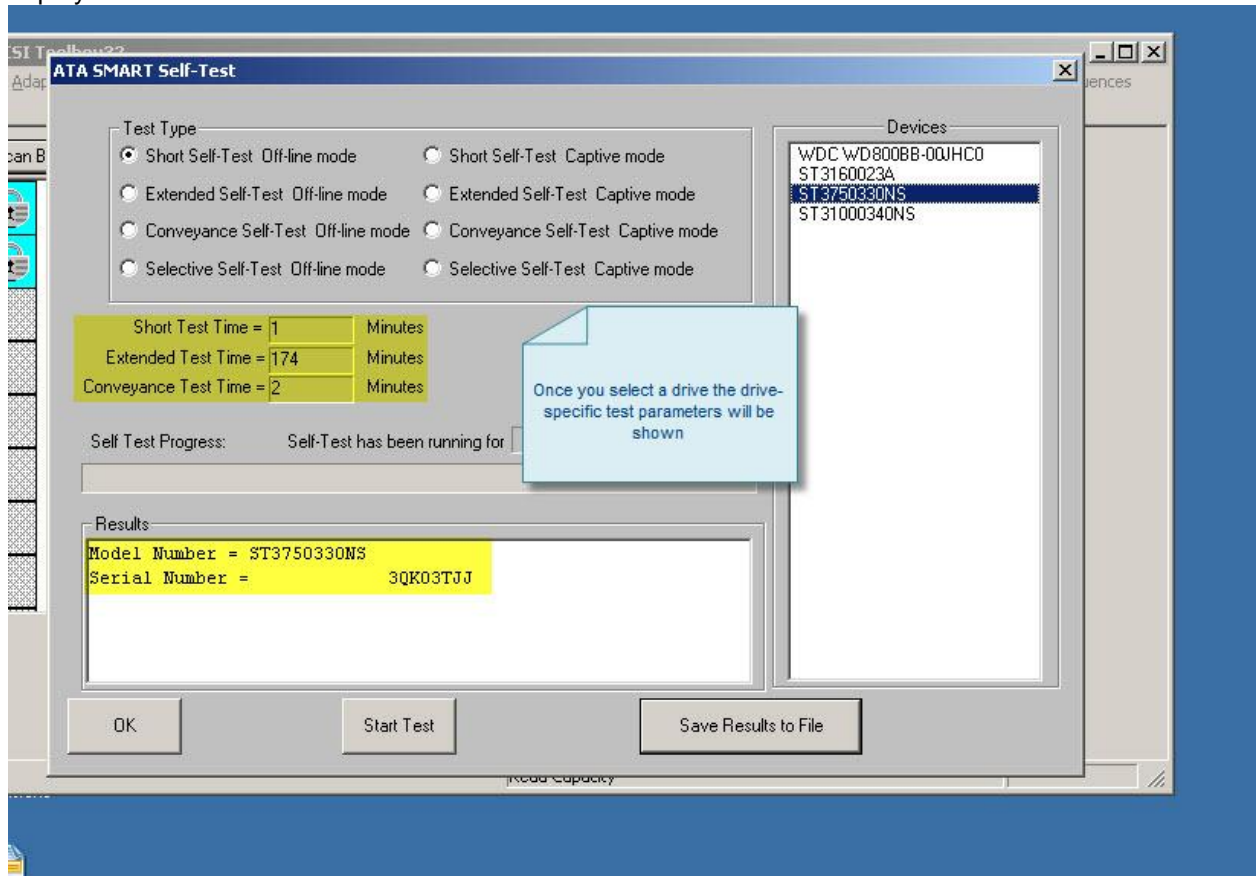


5. Execute SMART Self-Test

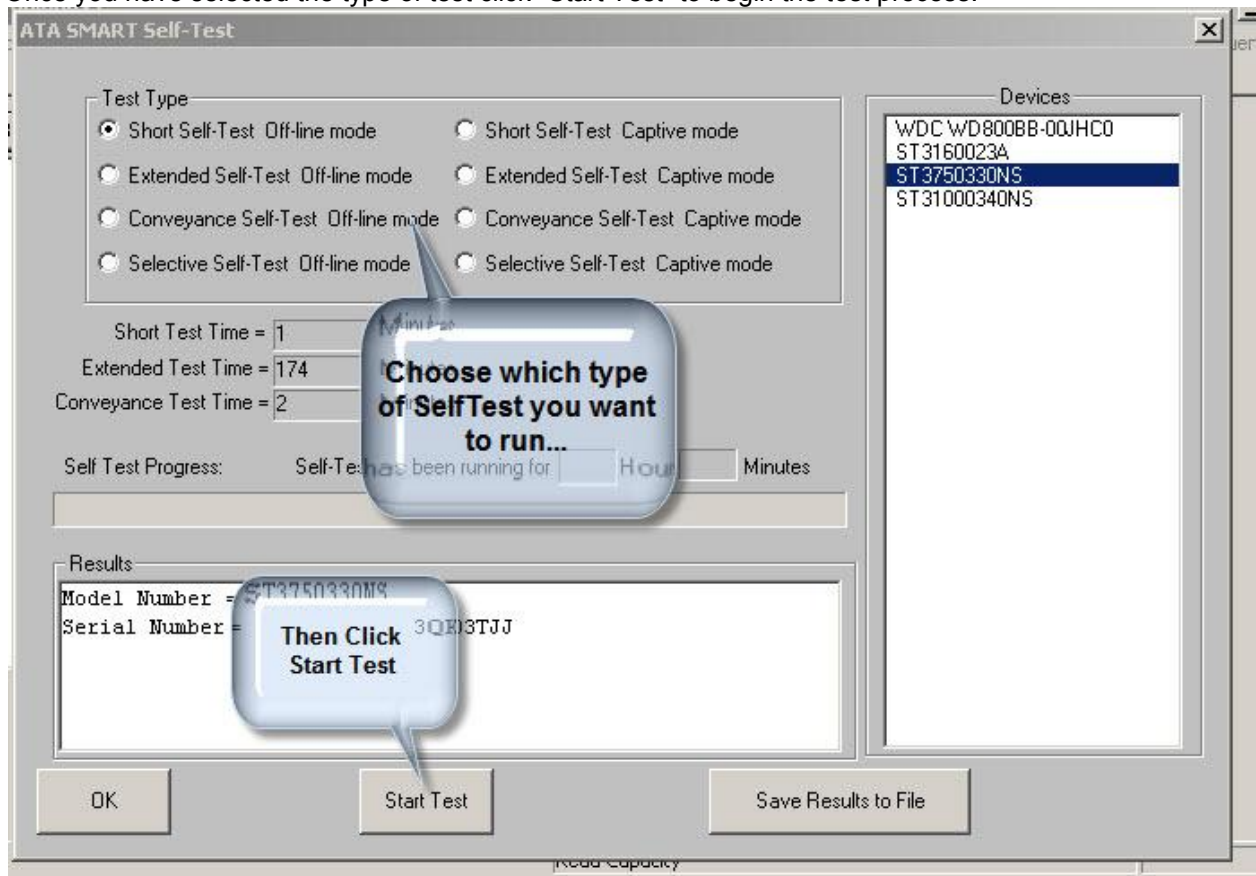
This test will let you run any of the various types of SATA SMART Self-Tests. As shown – first select the drive you want to run the Self-Test on



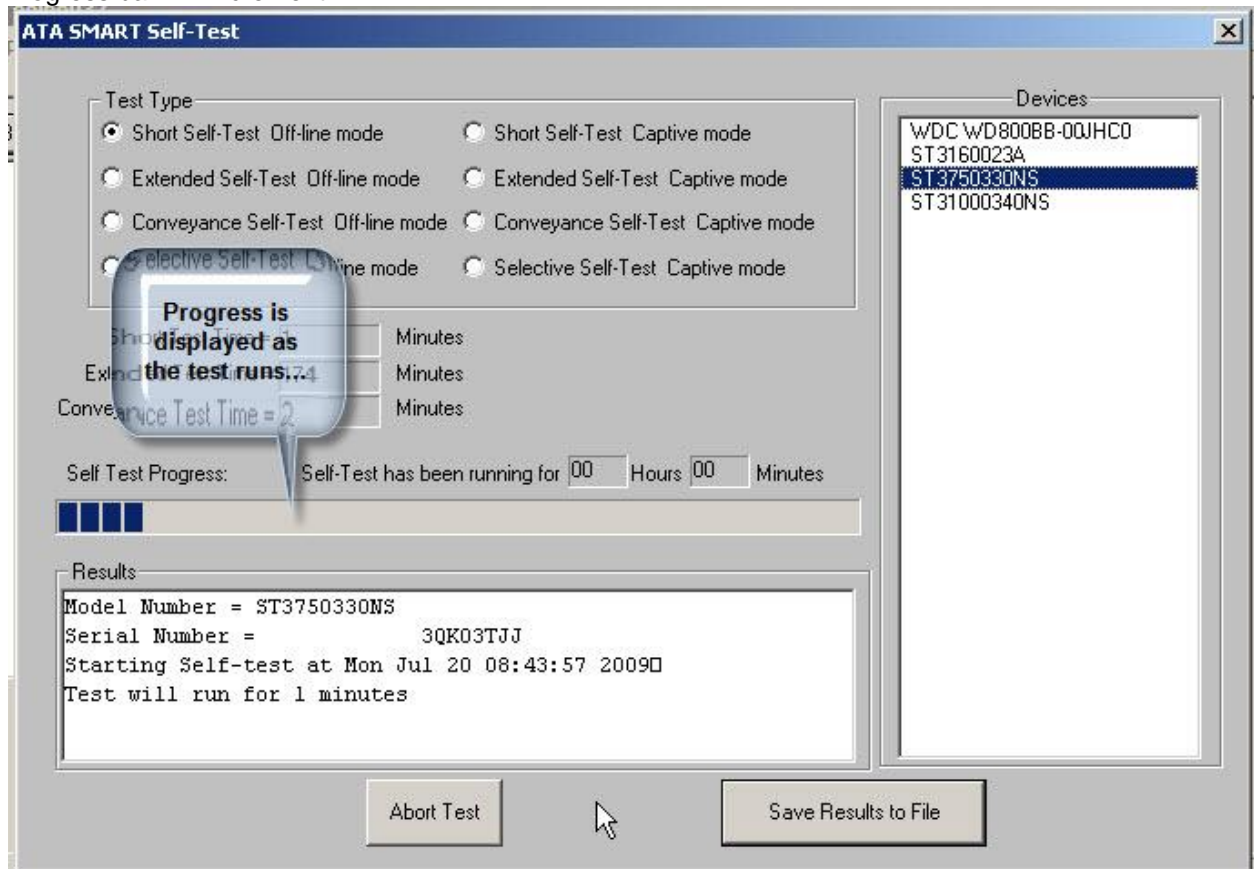
The approximate times that each of the different types of self tests will take to complete will be displayed.



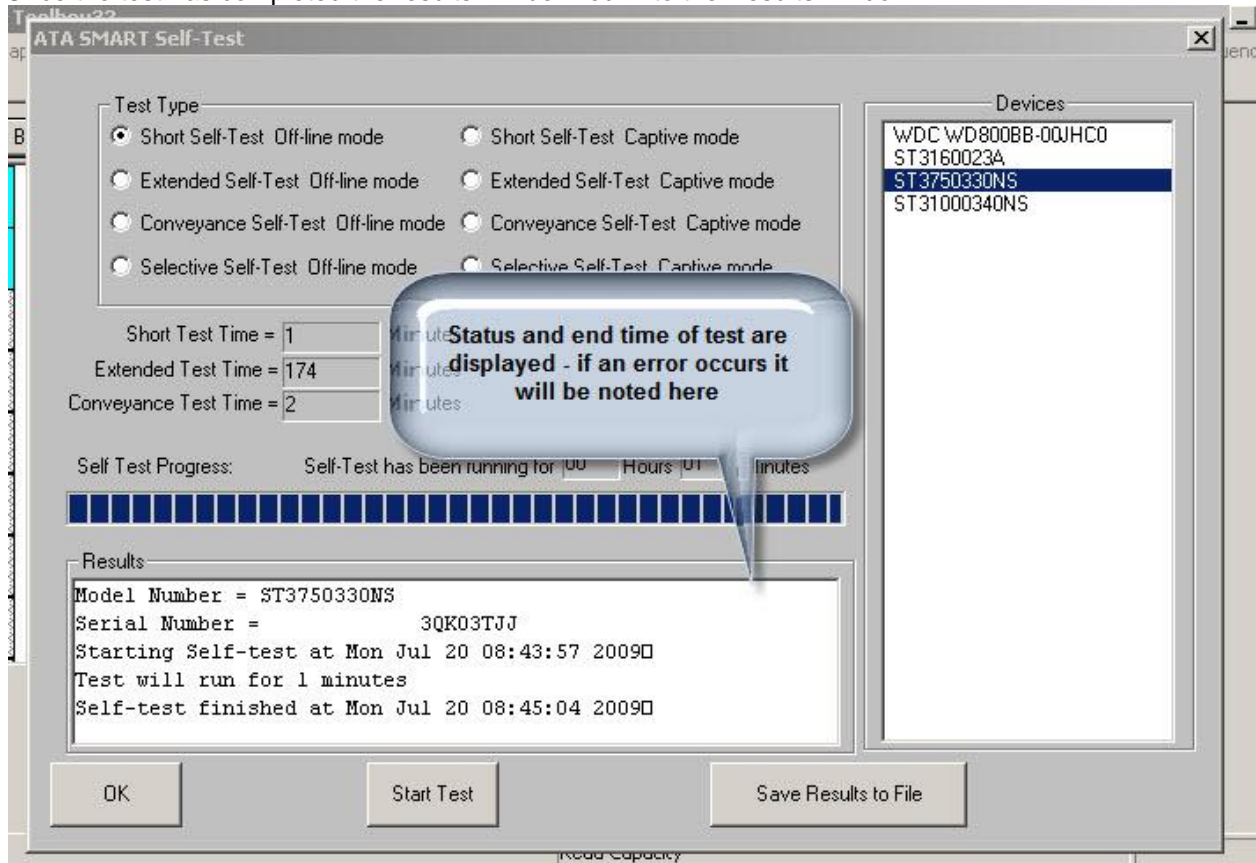
Now select which of the available types of Self-Tests you want to run in the "Test Type" area. Once you have selected the type of test click "Start Test" to begin the test process.



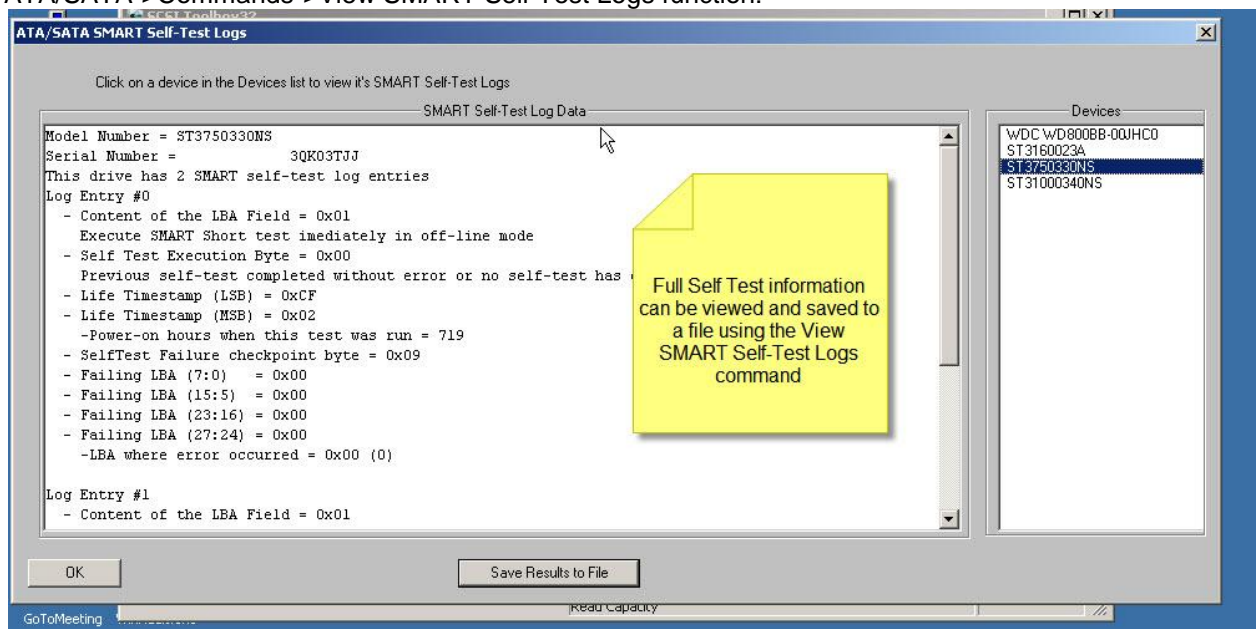
As the Self-Test runs the start time will be displayed in the Results window and the Self Test Progress bar will increment.



Once the test has completed the results will be filled in to the Results window:



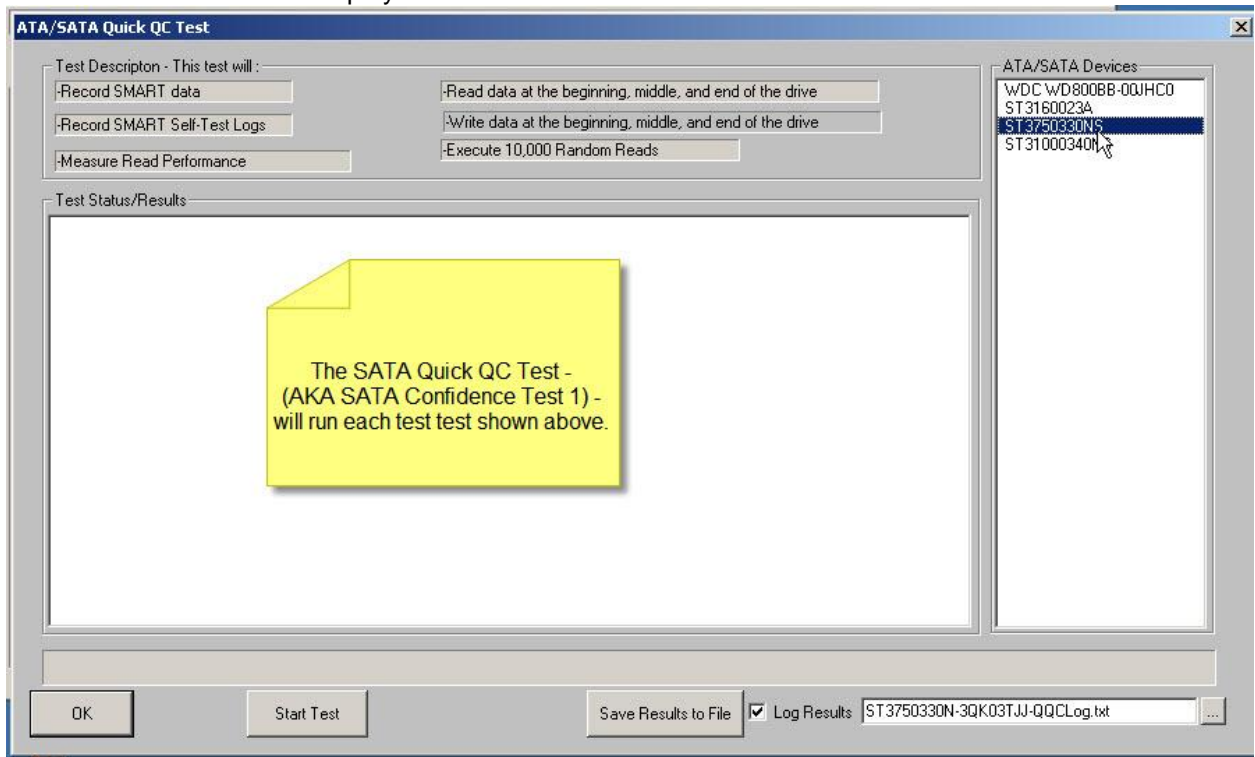
You may view the results of all self-tests which have run on a given drive by using the ATA/SATA->Commands->View SMART Self Test Logs function:



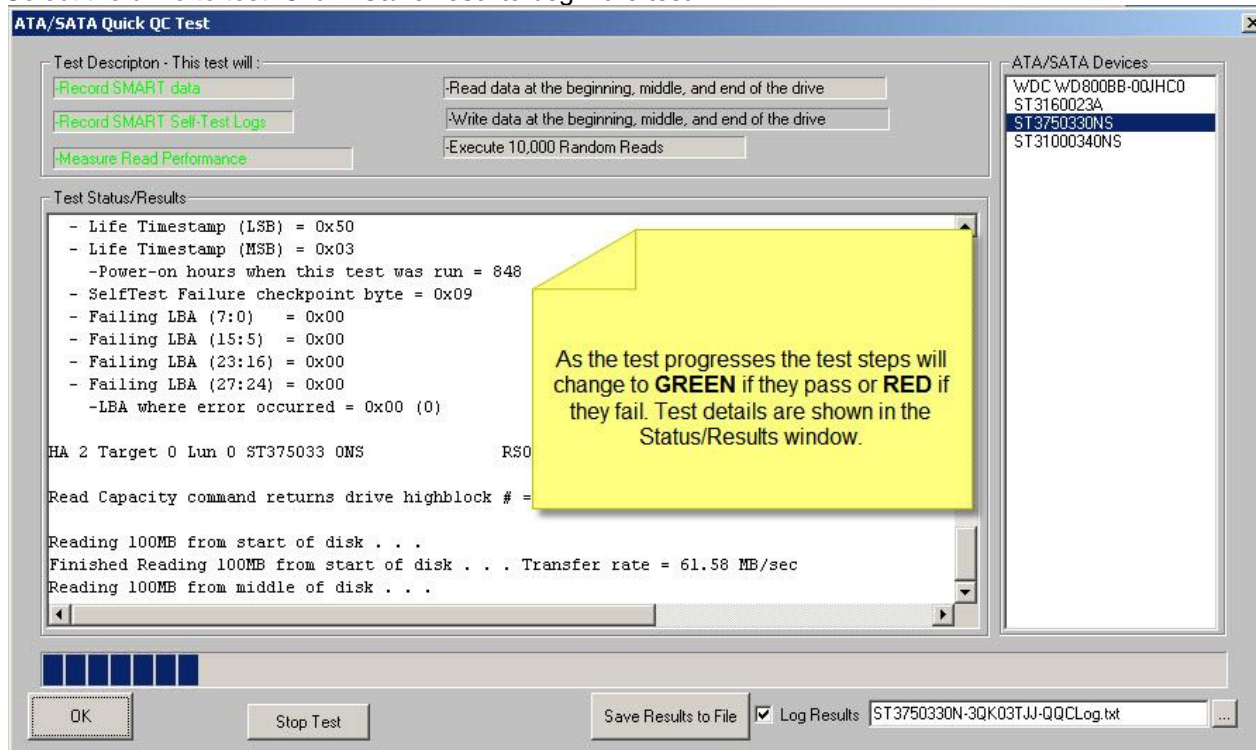
6. SATA Drive Confidence Test #1 (Quick QC Test)

This new test will execute a series of test steps to give a quick idea of drive health and functionality. Each test step which will be run is displayed in the Test Description area. As the test

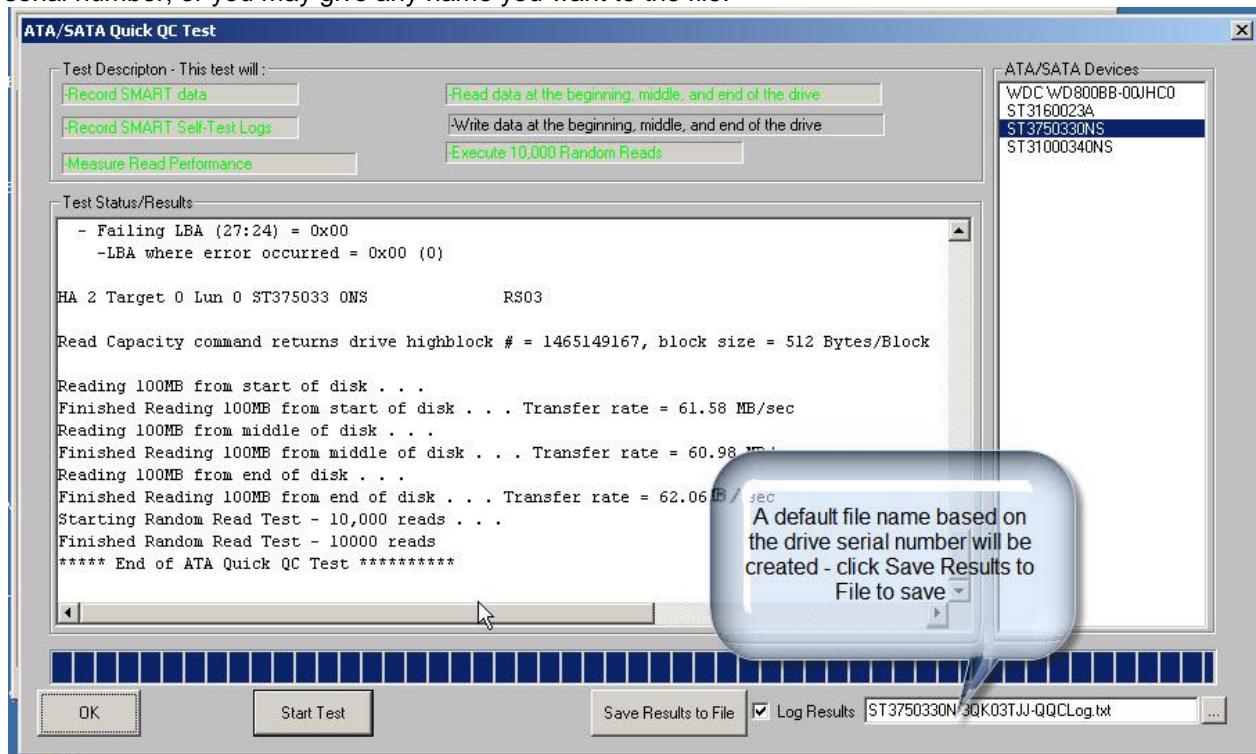
runs each test step will indicate whether it passed or failed, and detailed results will be displayed in the Test Status/Results display area:



Select the drive to test. Click **“Start Test”** to begin the test.

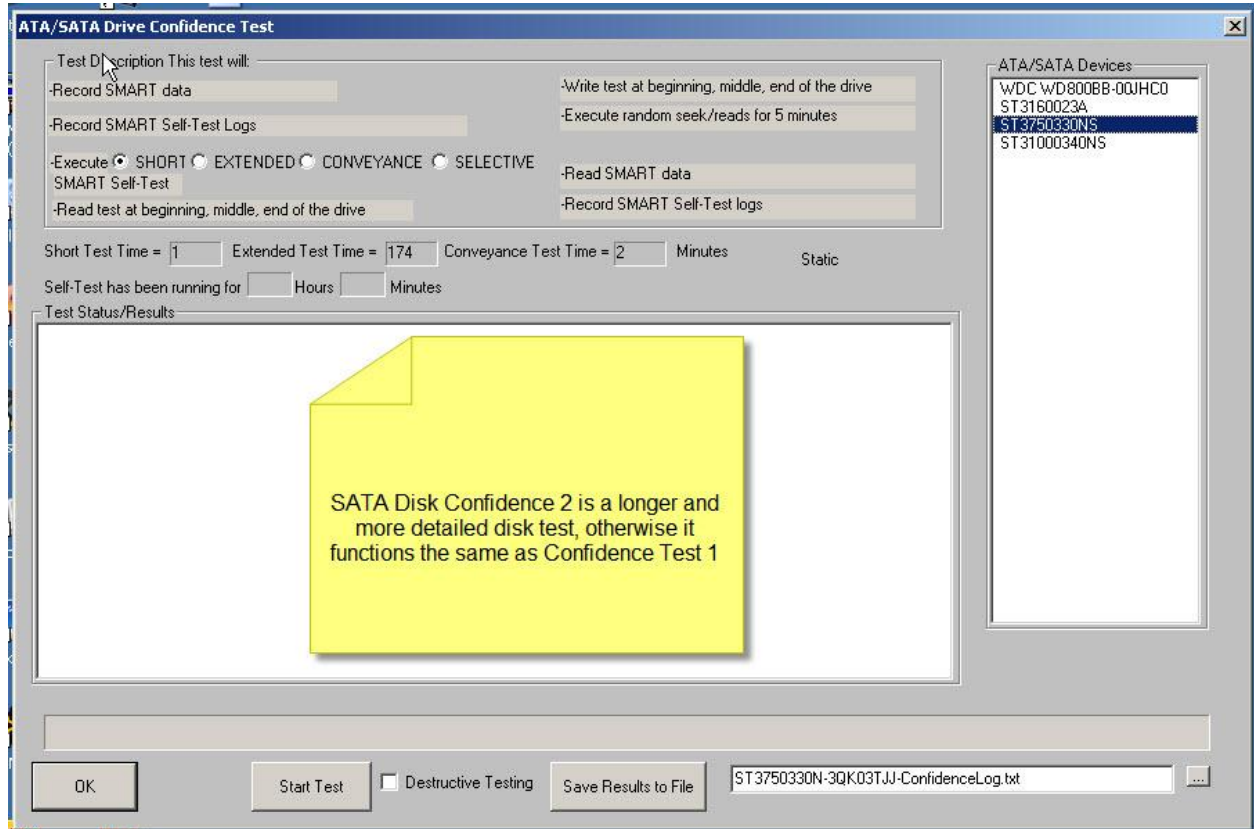


You may save the test results to a text file. A default file name is created from the drive type and serial number, or you may give any name you want to the file.



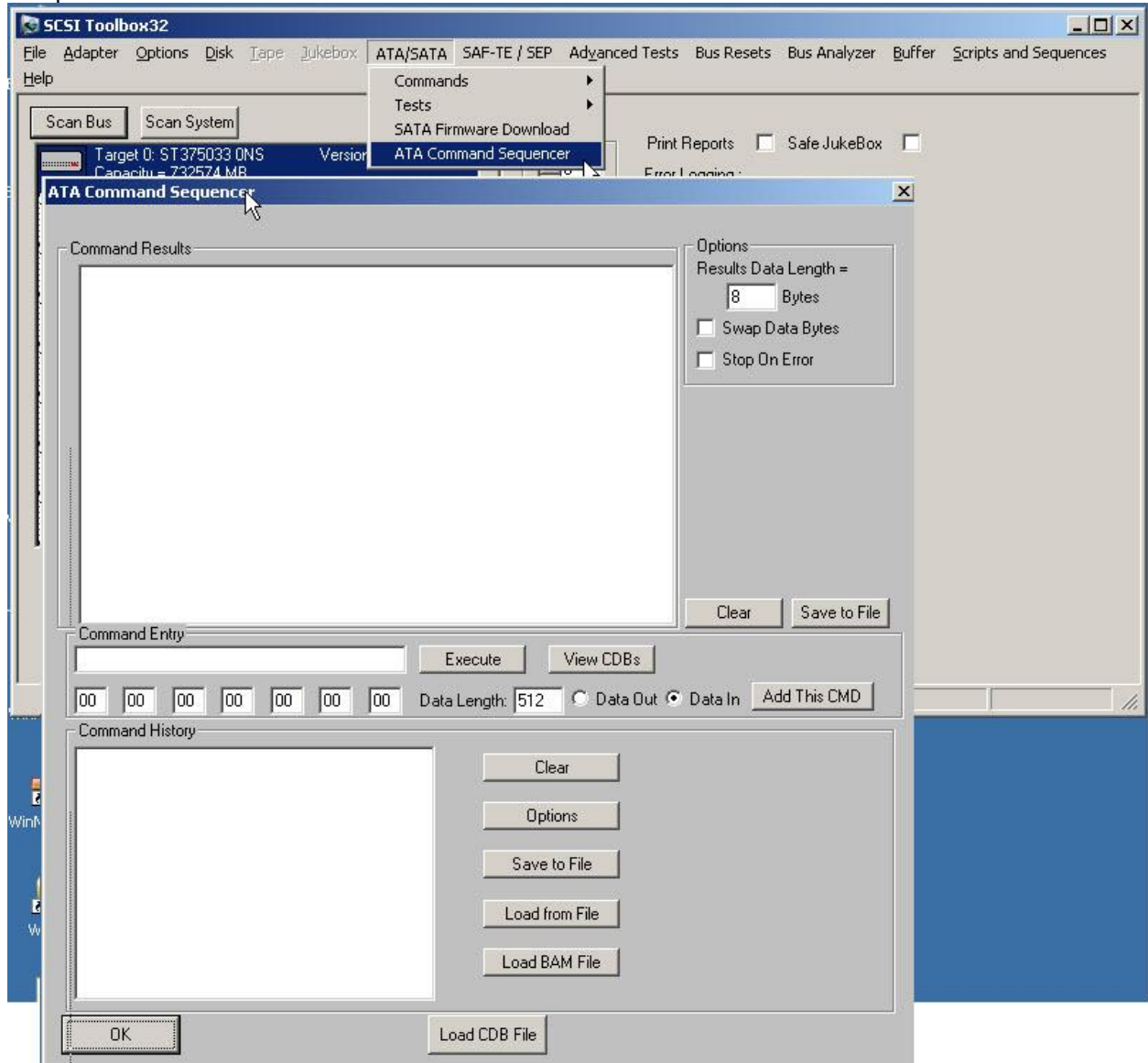
7. SATA Drive Confidence Test #2

This test also runs a sequence of test steps like Confidence Test #1 – just longer and more detailed.



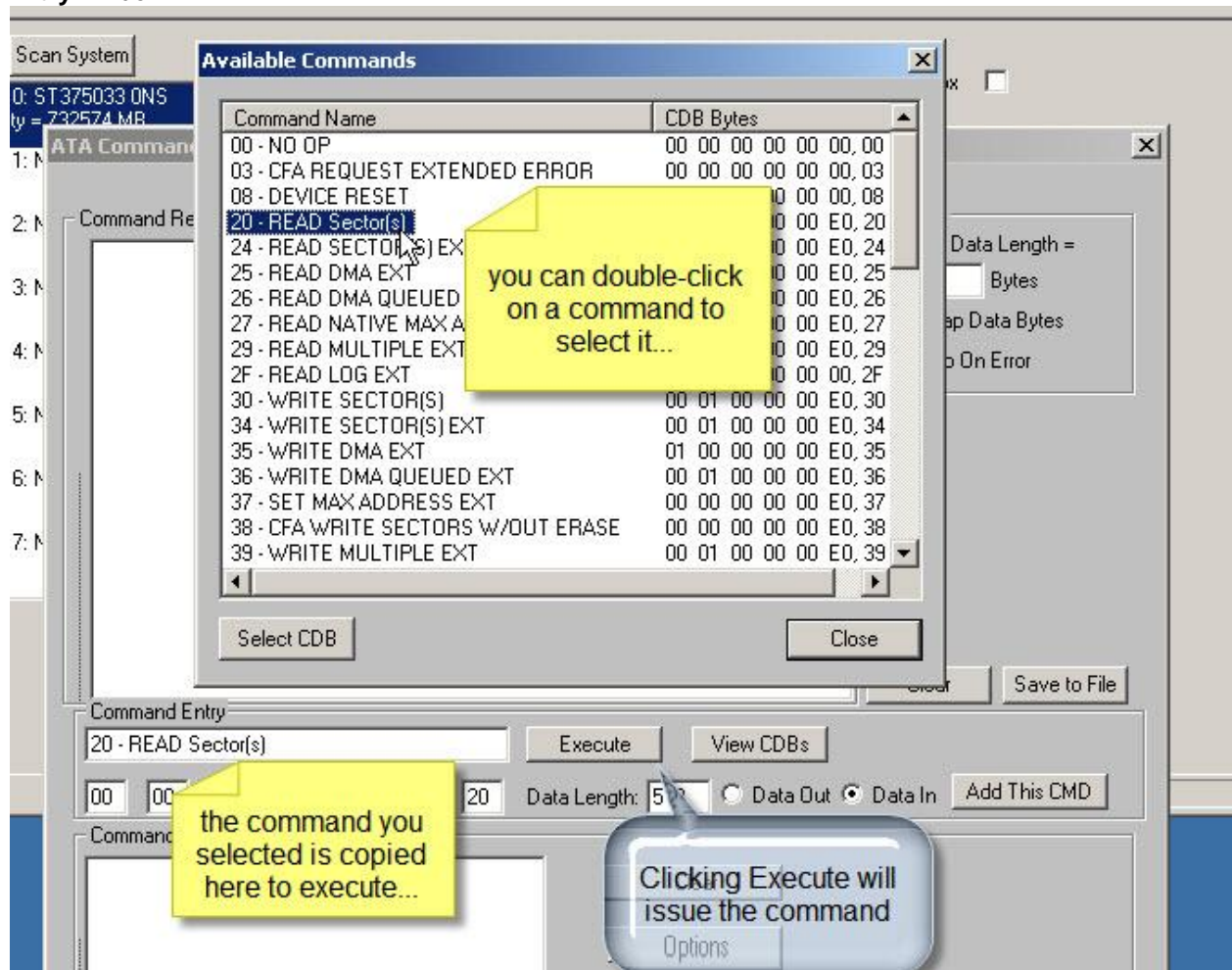
8. SATA Command Sequencer

The SATA Command Sequencer is a quick way to issue any ATA command to a drive, capture and view any data returned, and to build a list or sequence of commands which may be issued/re-issued. You can also use the Command Sequencer to quickly create a Disk Command Compliance test.



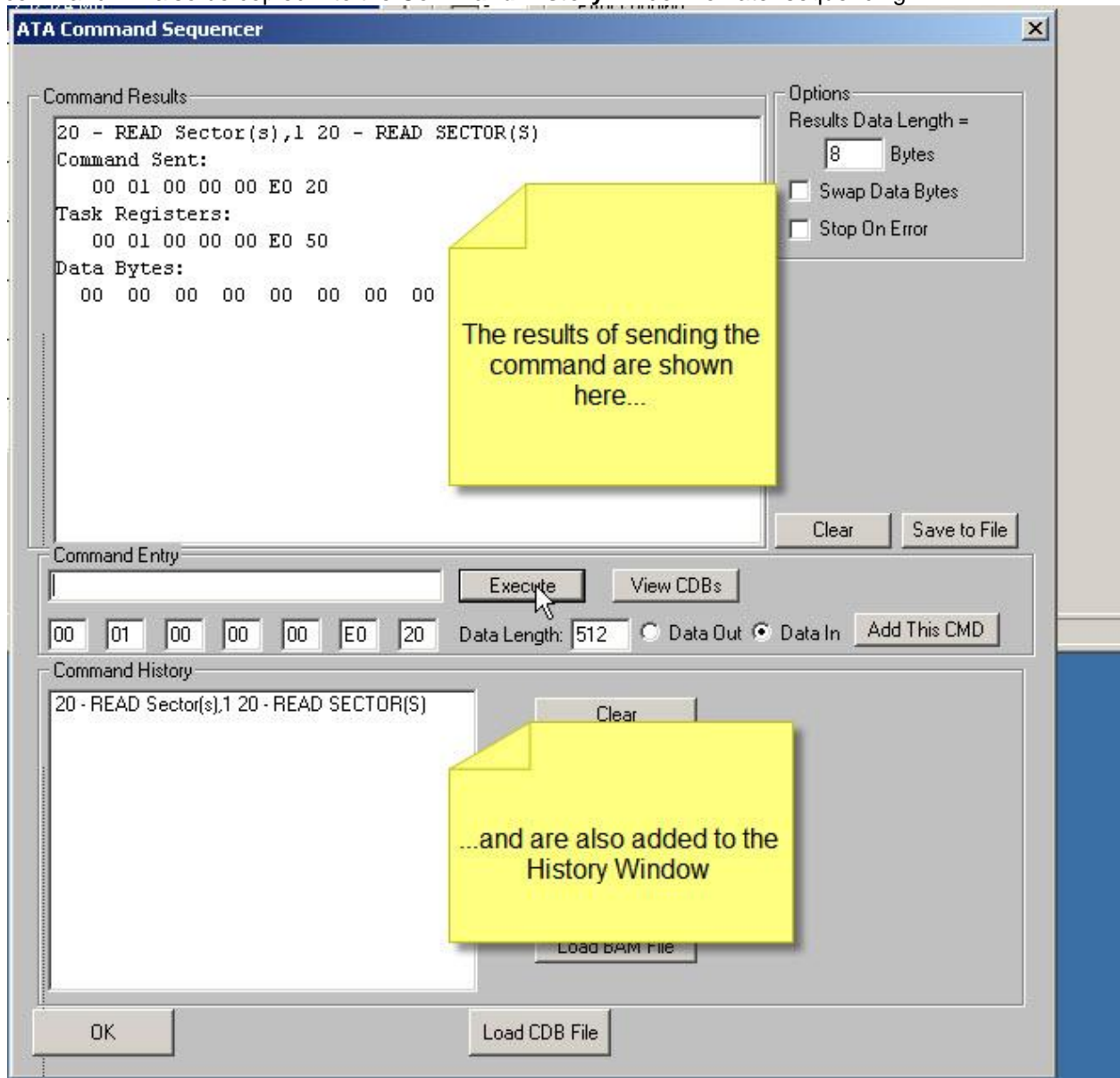
There are two ways to send a command. One is to use the **“View CDB’s”** button to display a list of all available commands. These commands are taken from the same text file which the ATA User Defined Command feature uses. You can create your own custom command files if you wish – see the documentation on ATA User Defined commands for instructions how.

Using the **View CDB's** button will display this window where you may select a command by double-clicking on it. When you double-click a command it will be copied into the **Command Entry** window.



Once a command is in the **Command Entry** window clicking the **Execute** button will issue the command.

The results of the command (Task Register values) and the number of data bytes specified in the **Options->Results Data Length** window will be displayed in the Command Results window. The command will also be copied into the **Command History** window for later sequencing.



In addition to using the **View CDB's** button to specify a command you can also simply enter the first few characters of the command name (from your command list) in the **Command Entry** window. In the example below we enter "E5" and the sequence will look for the first occurrence of "E5" in the command file. When it finds a command that matches it will copy the full command to the **Command Entry** window.

ATA Command Sequencer

Command Results

```
00 01 00 00 00 E0 50
Data Bytes:
00 00 00 00 00 00 00 00
EC - IDENTIFY DEVICE,1 EC - IDENTIFY DEVICE
Command Sent:
00 00 00 00 00 A0 EC
Task Registers:
00 00 00 00 00 A0 50
Data Bytes:
5A 0C FF 3F 37 C8 10 00
E5 - Check Power Mode,1 E5 - CHECK POWER MODE
Command Sent:
00 00 00 00 00 A0 E5
Task Registers:
00 FF 00 00 00 E0 50
```

Options

Results Data Length = 8 Bytes

Swap Data Bytes

Stop On Error

Clear Save to File

Command Entry

E5 Execute View CDBs

00 00 00 00 00 A0 E5 Data Length: 0 Data Out Data In Add This CMD

Command List

20 READ SECTOR(S) 120 READ SECTOR(S) Clear

EC - IDENTIFY DEVICE,1 EC - IDENTIFY DEVICE Option

EE - CHECK POWER MODE,1 EE - CHECK POWER MODE Save to File

You may also type in the first few characters of a command name from your list, when you press "Enter" that command will be looked up and issued.

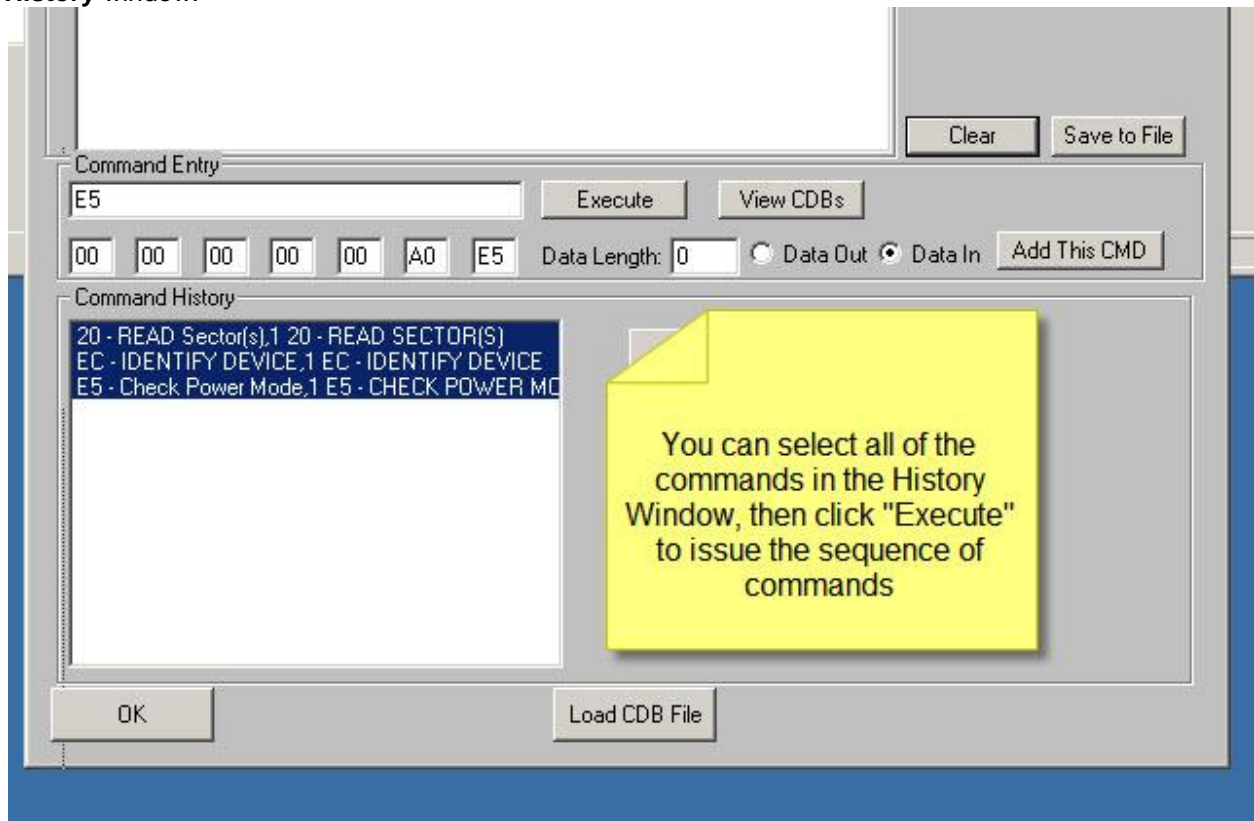
If the command is not found in your list an error will be displayed.

Load from File

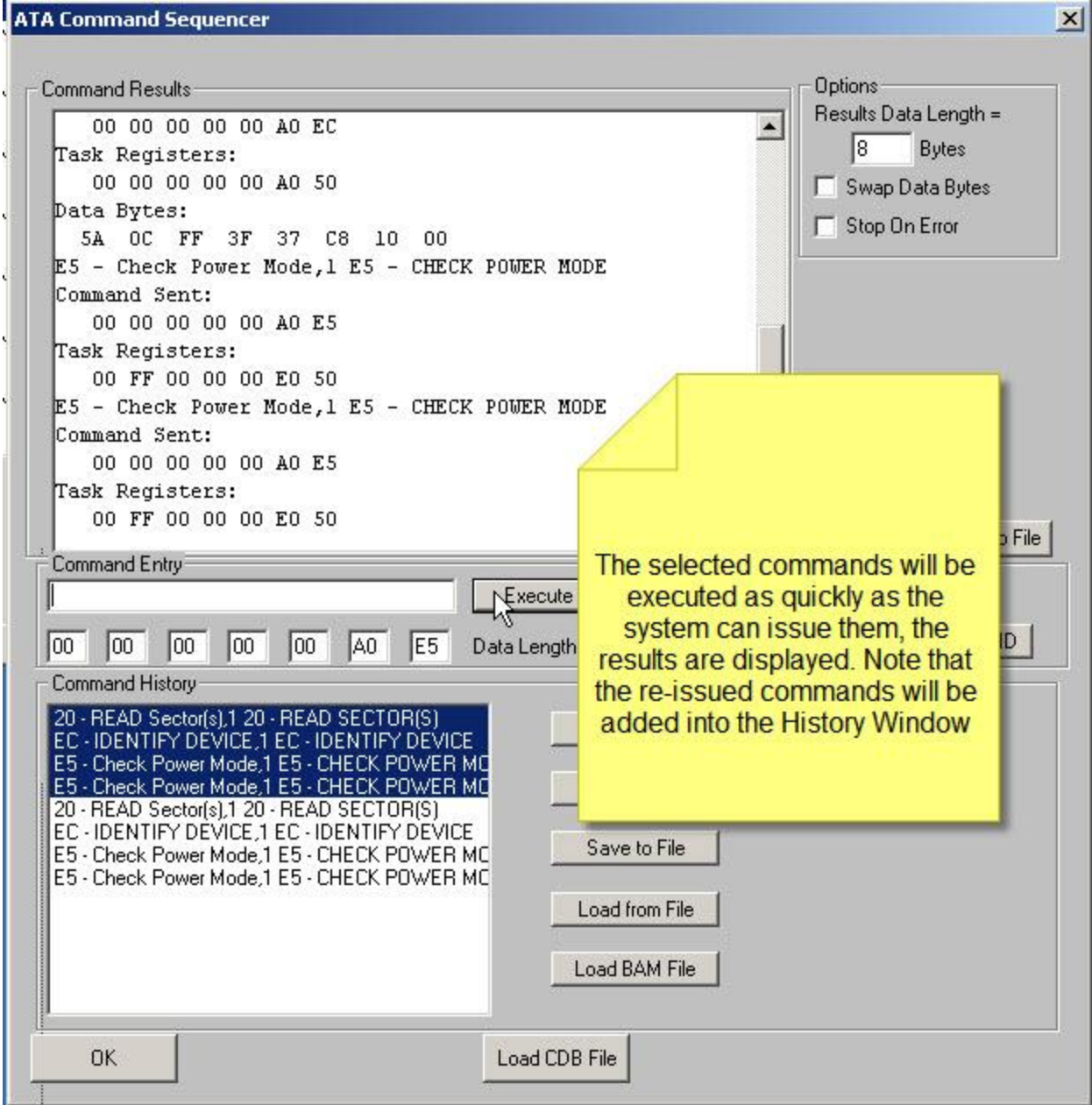
Load from File

OK Load CDB File

Now – to send sequences of commands you simple select all commands in the **Command History** window:



Once the commands are selected click the **Execute** button and the sequence of commands will be issued at full system speed.



The selected commands will be executed as quickly as the system can issue them, the results are displayed. Note that the re-issued commands will be added into the History Window

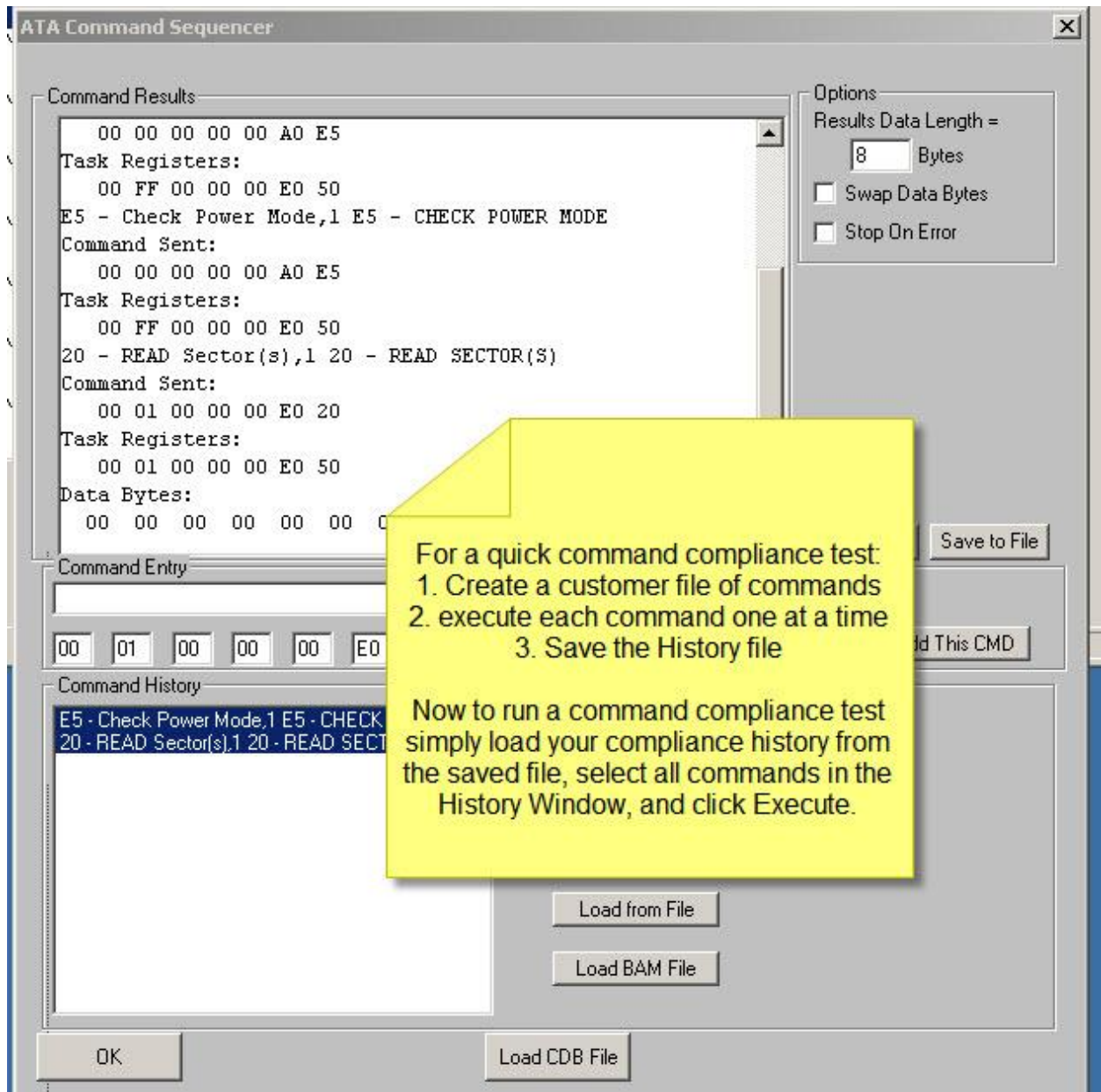
You may **save** the list of commands from the **Command History** window to a file. You can then **load** the **Command History** window from that file at any time to re-issue your command sequence.

The screenshot displays a software interface with several sections:

- Command Execution Output:** Shows hexadecimal data (00 FF 00 00 00 E0 50), a command (20 - READ Sector(s),1 20 - READ SECTOR(S)), the command sent (00 01 00 00 00 E0 20), task registers (00 01 00 00 00 E0 50), and data bytes (00 00 00 00 00 00 00 00).
- Command Entry:** Includes a text input field, an "Execute" button, and a "View CDBs" button.
- Command Parameters:** A row of buttons for hex values (00, 01, 00, 00, 00, E0, 20), a "Data Length" field set to 512, radio buttons for "Data Out" and "Data In" (with "Data In" selected), and an "Add This CMD" button.
- Command History:** A list box containing two entries: "E5 - Check Power Mode,1 E5 - CHECK POWER MO" and "20 - READ Sector(s),1 20 - READ SECTOR(S)".
- Buttons:** "Clear", "Options", "Save to File", "Load from File", "OK", and "Load CDB File".

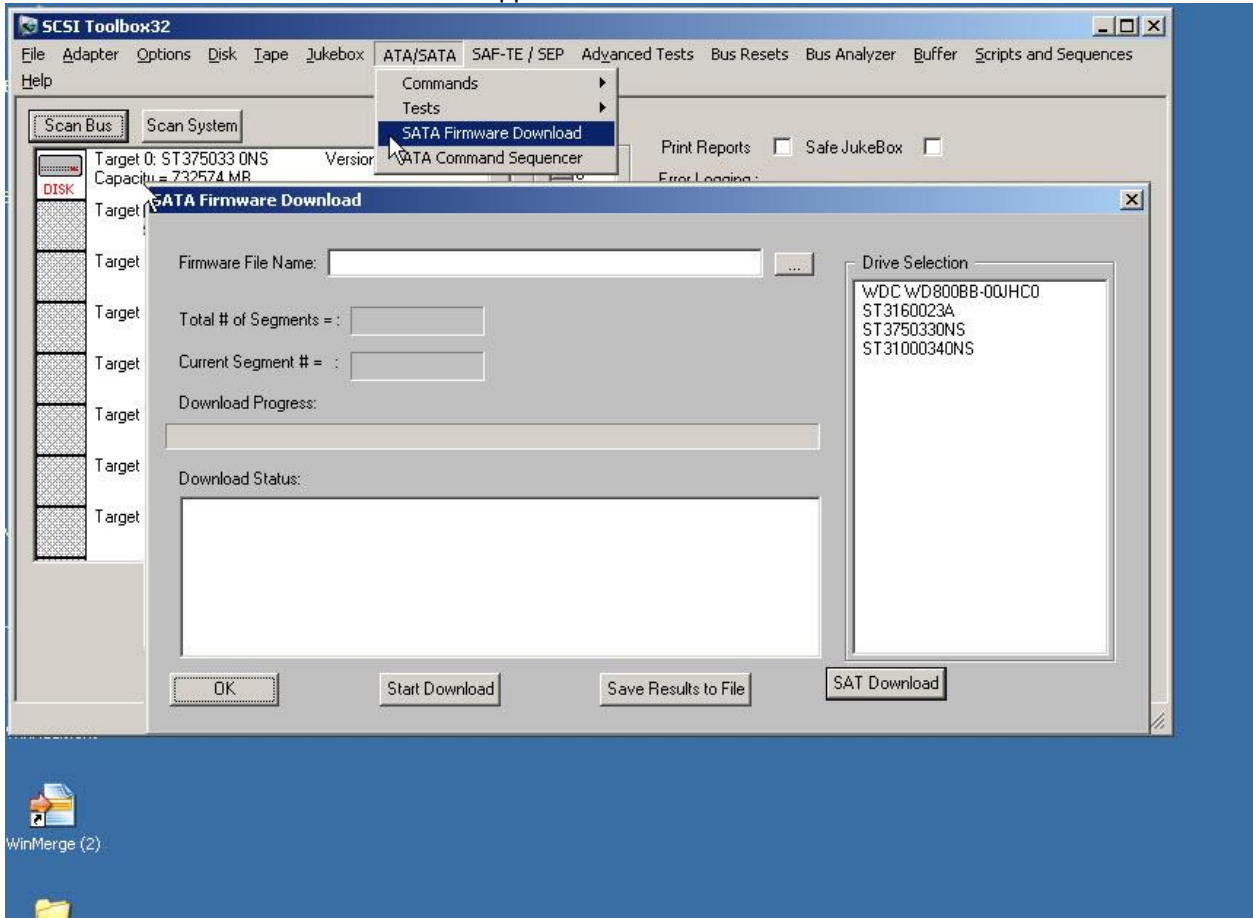
A yellow callout box on the right side of the interface contains the following text:

You can save a sequence of commands from your History Window to a file - then later you can load that command sequence from the file to re-run it.



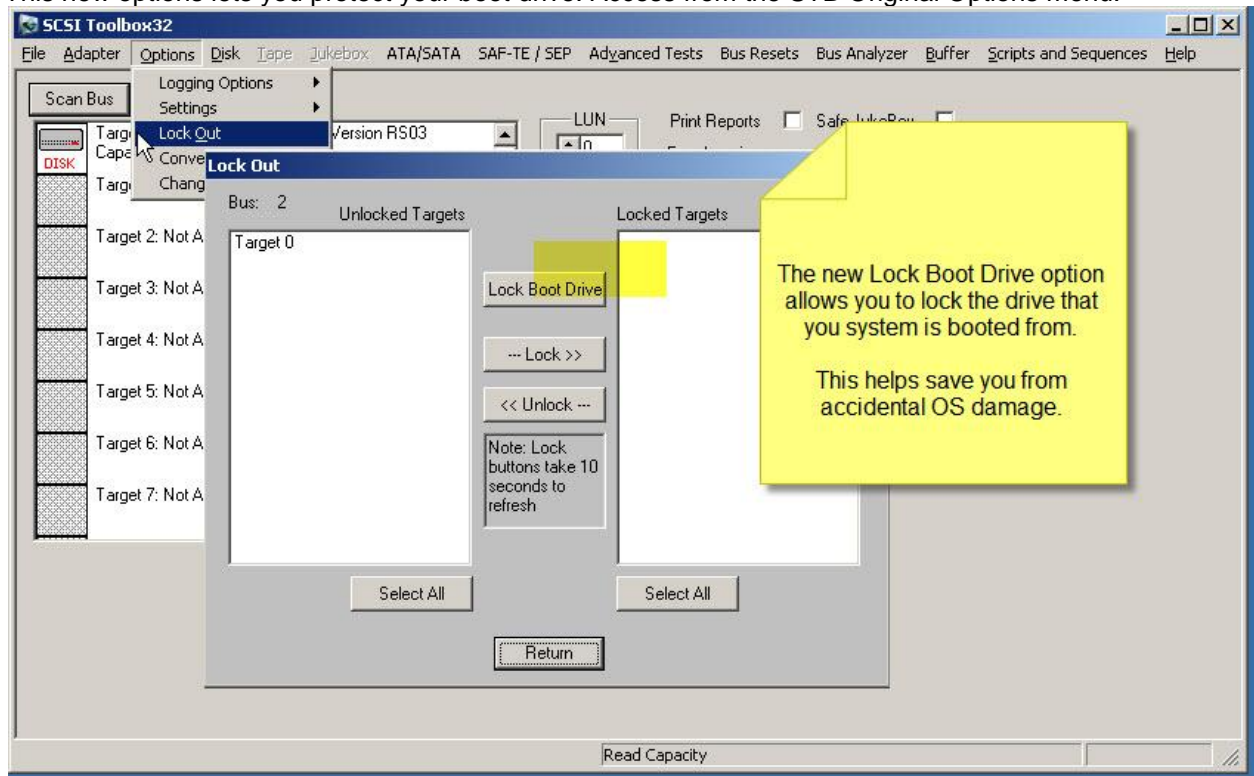
9. SATA Firmware Download

Use this command to download vendor-supplied firmware into a disk drive.



10 – Lock Boot Drive Option

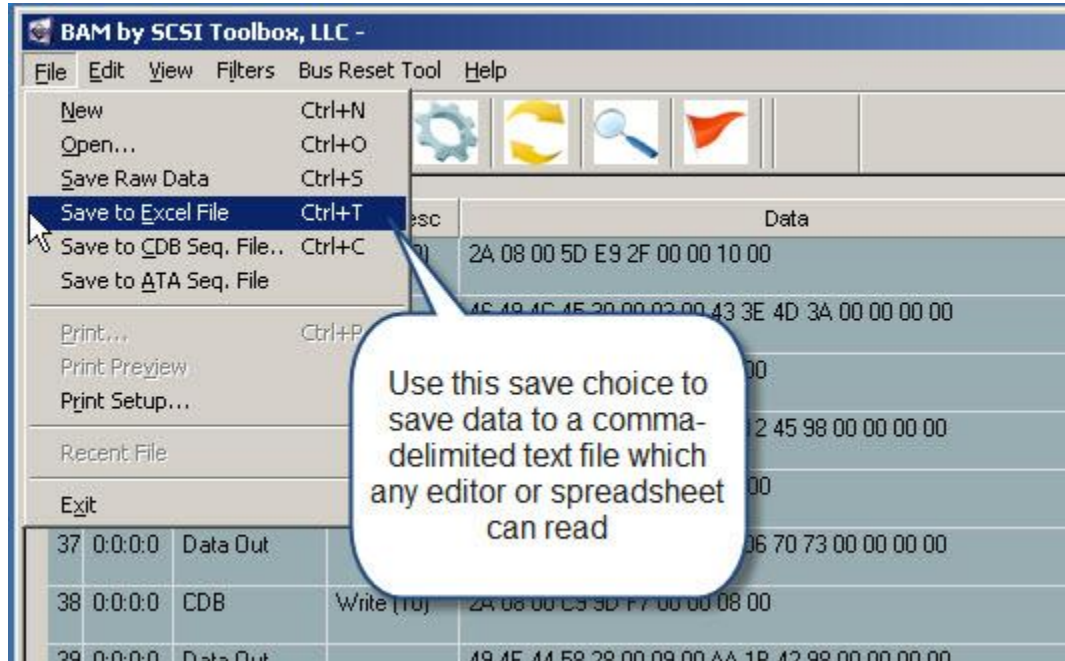
This new option lets you protect your boot drive. Access from the STB Original Options menu:



BAM new Features

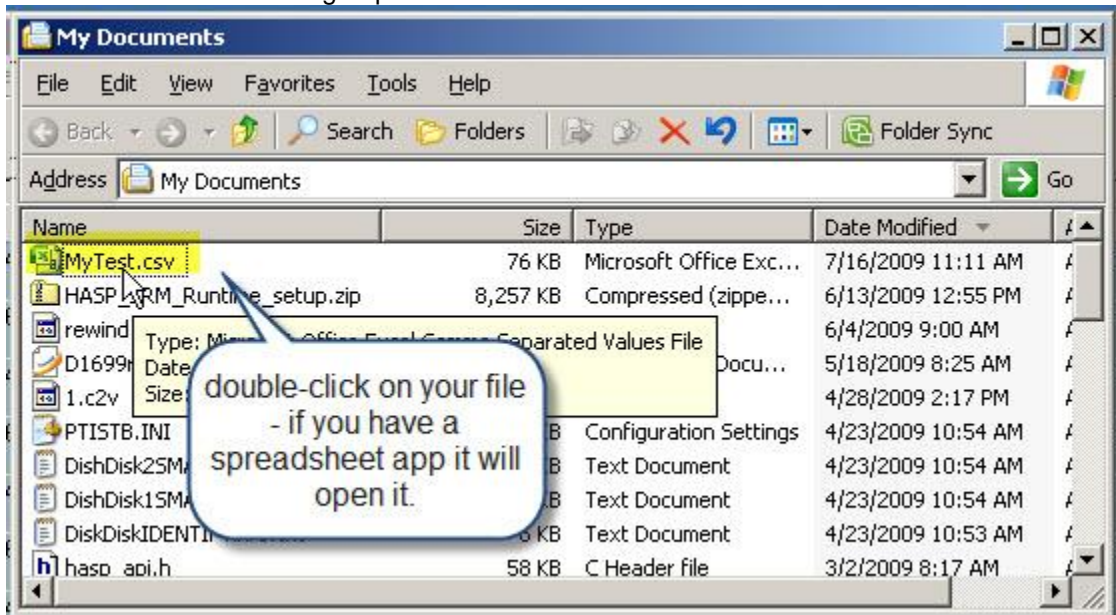
Save BAM capture data to a Spreadsheet file

This new feature lets you send data to colleagues or coworkers who do not have their own copy of BAM.



This choice will write all of the BAM capture data to a comma-delimited (.csv) text file. This type of file can be opened by any spreadsheet program or you may also open these files with any text editor.

Reading the file is simple – to open it with a spreadsheet program such as Microsoft Excel simply double-click on the file using Explorer:



Here is an example of how Excel will display our BAM data:

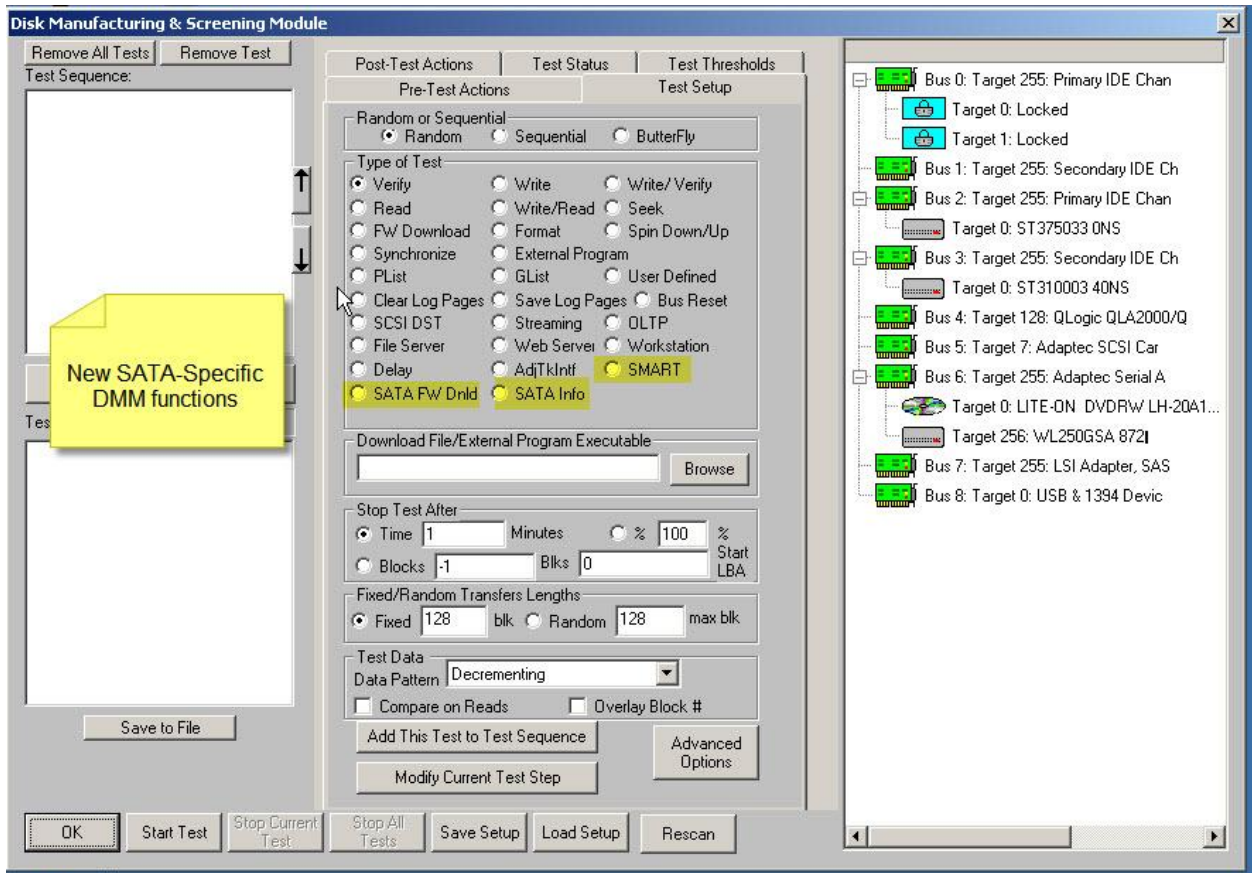
The screenshot shows the Microsoft Excel interface with a table of BAM data. The table has the following columns: Ctr, Device, Phase Type, CDB Desc, Data, Data Length, Delta, Date, and Driver. The data is organized into rows, with the first row (row 3) highlighted in orange. The table contains 23 rows of data.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
Ctrl	Device	Phase Type	CDB Desc	Data	Data Length	Delta	Date	Driver															
0	0:0:0:0	CDB	Write (10)	2A 00 00 03 0D 17 00 00 01 00	10 Bytes	722.8 ms	7/20/2009 9:46	classnpn															
1	0:0:0:0	Data Out		72 65 67 66 12 02 00 00 11 02 00 0C	512 Bytes	143 us	7/20/2009 9:46																
2	0:0:0:0	CDB	Write (10)	2A 08 01 90 FE CF 00 00 08 00	10 Bytes	160 us	7/20/2009 9:46	classnpn															
3	0:0:0:0	Data Out		52 43 52 44 28 00 09 00 F0 71 48 9E	4096 Bytes	190 us	7/20/2009 9:46																
4	0:0:0:0	CDB	Write (10)	2A 08 01 8F DD 1F 00 00 08 00	10 Bytes	46 us	7/20/2009 9:46	classnpn															
5	0:0:0:0	Data Out		52 43 52 44 28 00 09 00 00 90 43 02	4096 Bytes	165 us	7/20/2009 9:46																
6	0:0:0:0	CDB	Write (10)	2A 08 00 5D B2 87 00 00 08 00	10 Bytes	36 us	7/20/2009 9:46	classnpn															
7	0:0:0:0	Data Out		46 49 4C 45 30 00 03 00 2F 8D 31 0	4096 Bytes	173 us	7/20/2009 9:46																
8	0:0:0:0	CDB	Write (10)	2A 08 00 5D B2 77 00 00 08 00	10 Bytes	39 us	7/20/2009 9:46	classnpn															
9	0:0:0:0	Data Out		46 49 4C 45 30 00 03 00 7F 79 17 9E	4096 Bytes	177 us	7/20/2009 9:46																
10	0:0:0:0	CDB	Write (10)	2A 08 00 13 02 4F 00 00 08 00	10 Bytes	33 us	7/20/2009 9:46	classnpn															
11	0:0:0:0	Data Out		49 4E 44 58 28 00 09 00 E8 9B 5A 8	4096 Bytes	176 us	7/20/2009 9:46																
12	0:0:0:0	CDB	Sync Cache	35 00 00 00 00 00 00 00 00	10 Bytes	70 us	7/20/2009 9:46	classnpn															
13	0:0:0:0	OK			0 Bytes	68.2 ms	7/20/2009 9:46																
14	0:0:0:0	CDB	Write (10)	2A 00 00 03 0D 18 00 00 03 00	10 Bytes	80 us	7/20/2009 9:46	classnpn															
15	0:0:0:0	Data Out		44 49 52 54 FF 00 00 00 00 00 00 0C	1536 Bytes	153 us	7/20/2009 9:46																
16	0:0:0:0	CDB	Write (10)	2A 00 00 03 0D 1B 00 00 04 00	10 Bytes	146 us	7/20/2009 9:46	classnpn															
17	0:0:0:0	CDB	Write (10)	2A 00 01 33 DA 07 00 00 04 00	10 Bytes	38 us	7/20/2009 9:46	classnpn															
18	0:0:0:0	Data Out		68 62 69 6E 00 00 00 00 10 00 0C	2048 Bytes	162 us	7/20/2009 9:46																
19	0:0:0:0	Data Out		6F 6E 4F 72 64 65 72 00 E0 FF FF	2048 Bytes	138 us	7/20/2009 9:46																
20	0:0:0:0	CDB	Write (10)	2A 00 01 33 DA 0B 00 00 10 00	10 Bytes	69 us	7/20/2009 9:46	classnpn															

DMM (Disk Manufacturing Module) New Features & Tests

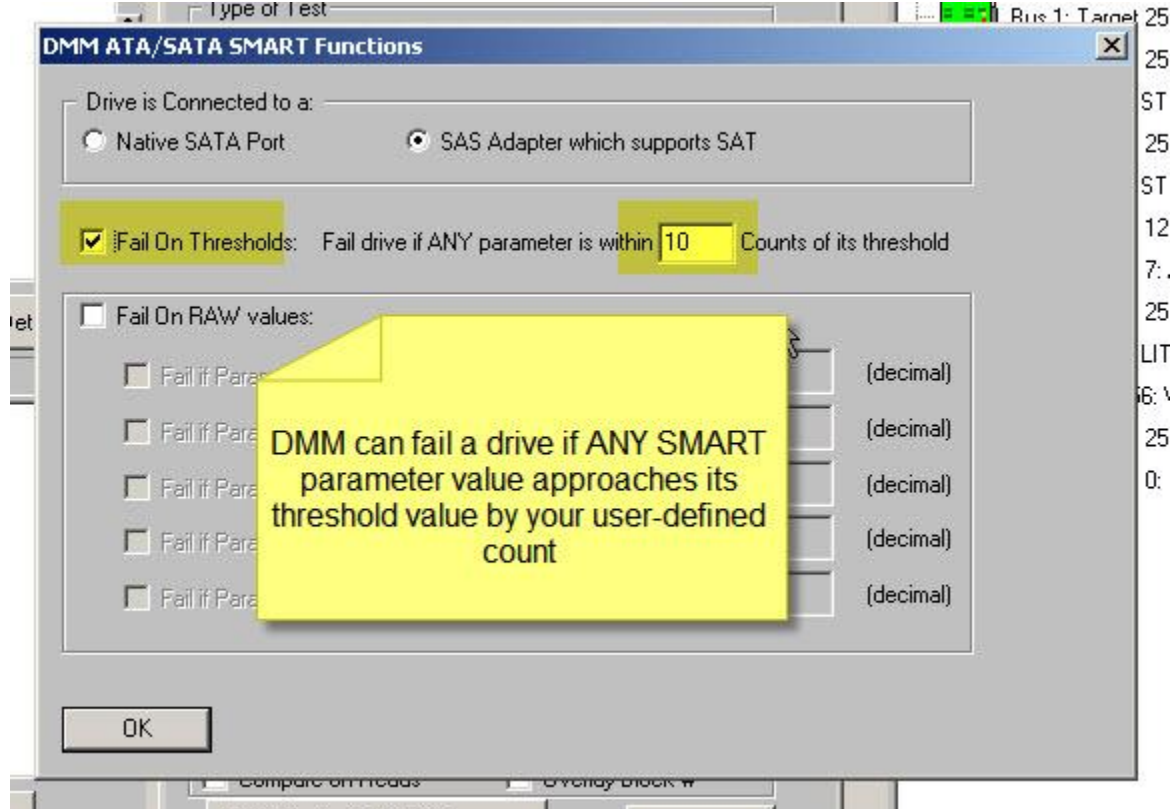
Several new test types have been added to DMM in version 8.1

1. SATA-specific test steps

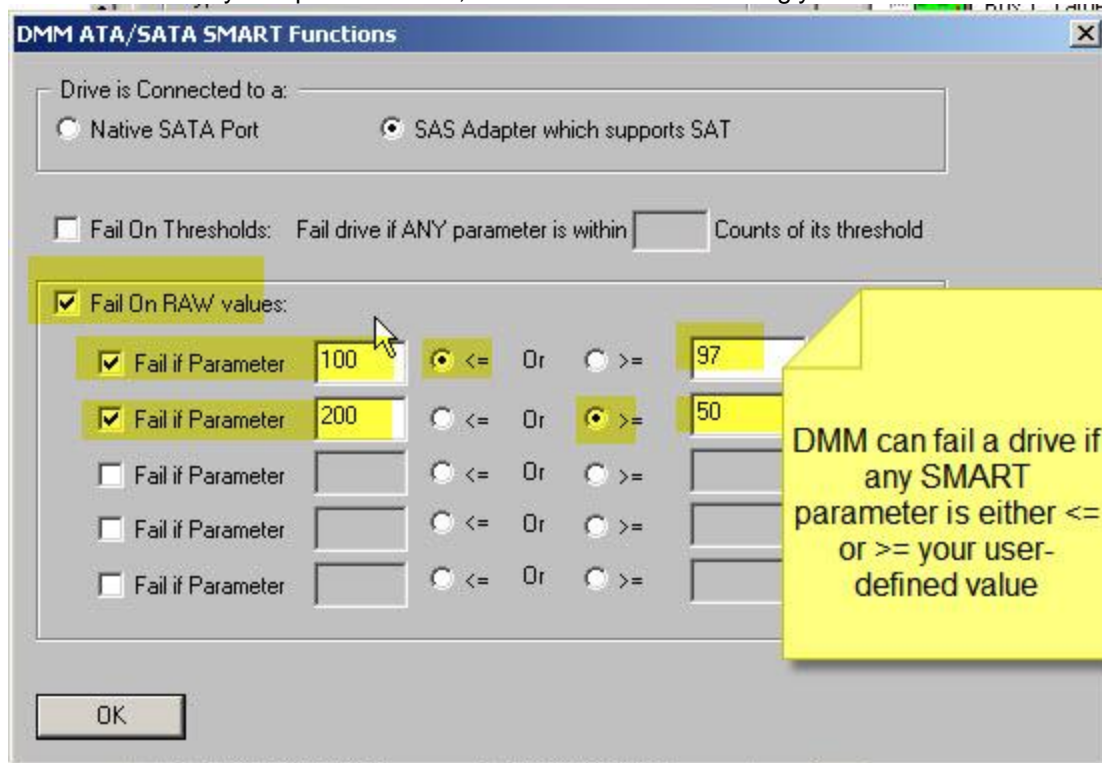


2. SATA SMART test type

This test step will retrieve the SMART data from each drive and has two methods to screen disks. The first method will check ALL SMART parameters and if ANY are within your specified count of their Thresholds the drive will be failed:

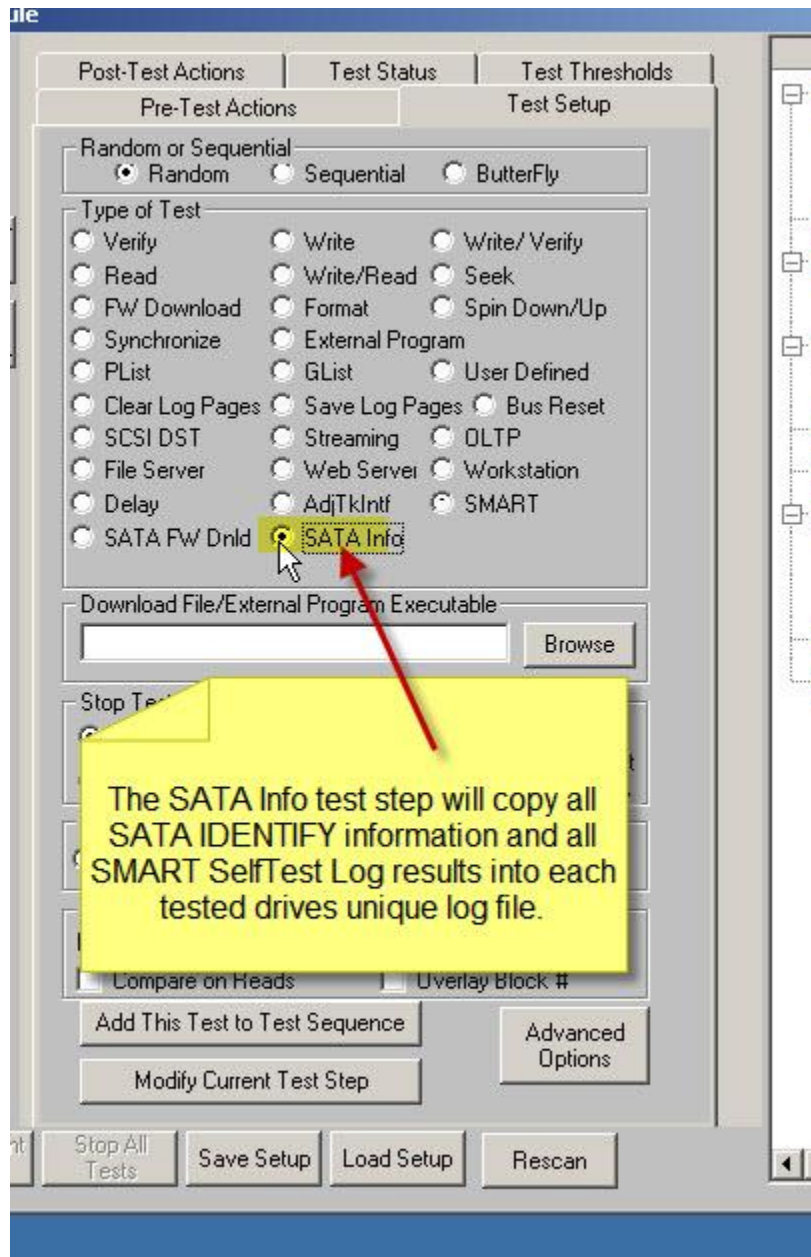


The second method lets you specify up to five SMART PARAMETER/VALUE pairs and compare either \leq or \geq to your specified value, and fail the drive accordingly:



3. SATA Info Test type

This type of test will retrieve all ATA IDENTIFY, SMART, and SMART Self-Test Log data from each drive and record this information into each drives log file.



Introduction

The SCSItoolbox32 is an engineering level diagnostic program used for testing, diagnosing, and qualifying disk, tape, jukebox, and CD devices. The industry-standard SCSI toolbox is based on solid, proved technology that has years of dependable service behind it from it's installed base of thousands.

The SCSItoolbox tests devices on virtually all storage bus types – SCSI, Fibre Channel, ATA, SATA, SAS, and iSCSI.

The SCSItoolbox32 runs under Windows NT, Windows 2000, Windows 2003, Server 2000, Server 2003 (32 and 64 bit), Windows XP (32 and 64 bit) and will operate using any SCSI, Fibre Channel, iSCSI, ATA, SATA, or SAS host bus adapter (HBA) that is supported by these operating systems.

The SCSItoolbox Suite includes a complete Manufacturing environment, the BAM Bus Analyzer Module, as well as development tools to develop test routines using Microsoft Visual Basic or Visual C++, and Linux OS environments.

Installation

Insert the SCSItoolbox CD into your computer's CDROM drive. If the drive is auto-run enabled, the installation process will start automatically. If not, use Explorer to browse to the CDROM root director and double-click on the setup.exe icon.

The installation process will install the program executable and dll files, and all necessary device driver files. In addition, it will install software to operate the Hasp copy protection "dongle".

Upon completion of the installation process you will be prompted to reboot your computer. Install the hardware key (dongle) onto the printer port of your computer and reboot.

Windows 98 Users

PLEASE NOTE!!!!

As of version 5.3.0 SCSItoolbox no longer supports running on Windows 95/98/ME

Also note that BAM. The Reset Control, and the ATA Task Register functions are only enabled when running under Windows 2000 and Windows XP.

Using SAT to access SATA drives

What is SAT?

SAT (SCSI->ATA Translation) is a mechanism whereby ATA task register commands may be sent to a device which is seen by the operating system as a SCSI device. This is most often the case when SATA drives are connected to an add-in PCI bus type of SATA controller card. Even though the card is a SATA controller in most cases Windows will see the controller as if it were a SCSI HBA, and so will not allow you to issue ATA task register level commands to the connected devices.

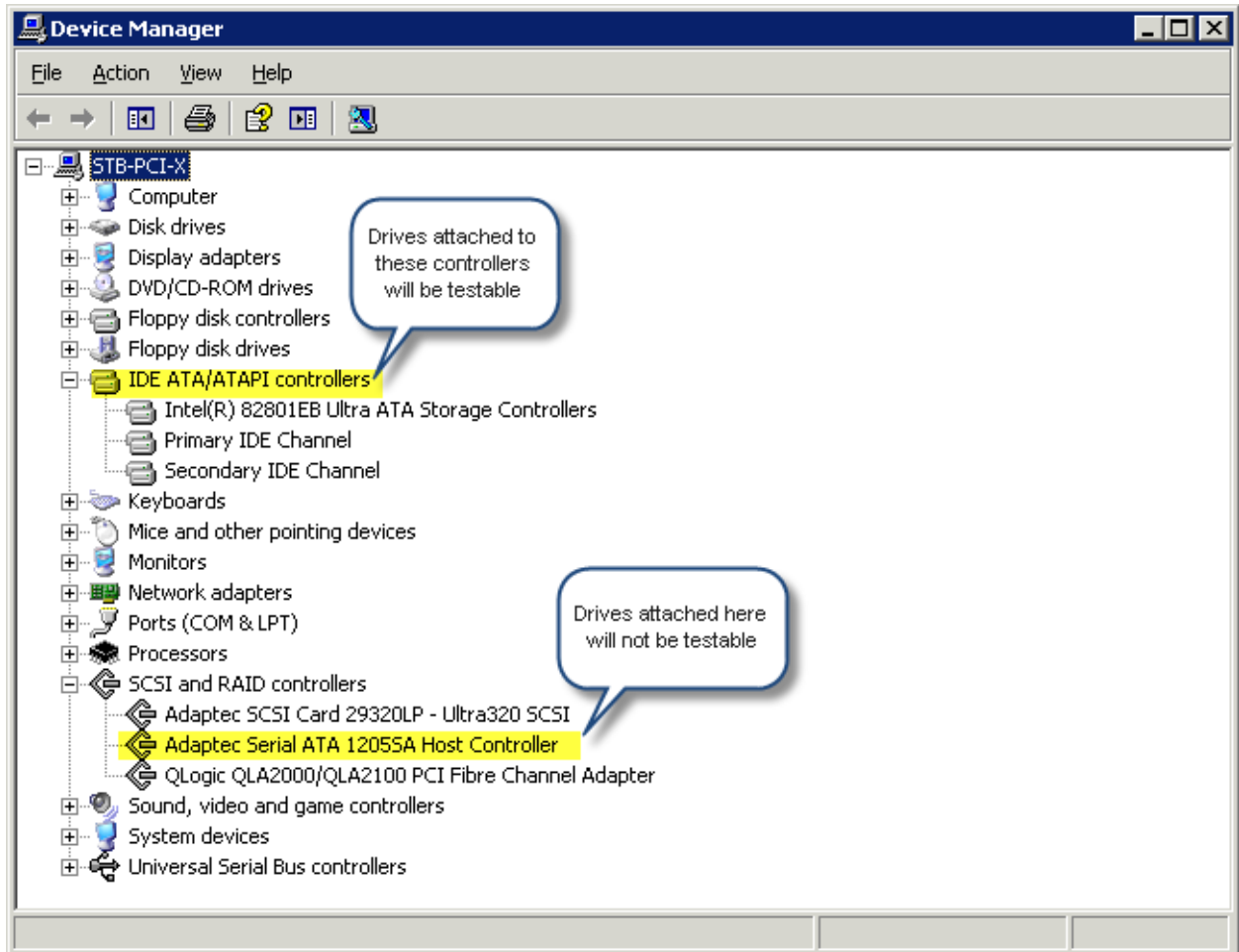
Documentation on SAT can be found at the T10.org site <http://www.t10.org/drafts.htm#sat3>
In short, SAT uses a 12 or 16 byte SCSI cdb which contains an embedded ATA task register command.

How do I know that I need to use SAT?

If your controller is seen by Windows as an ATA/IDE type of controller than you do not need to use SAT, you can simply issue normal ATA task register commands. How do you know what type of controller you have?

You can confirm how your operating system views your controller scheme by using Device Manager as shown below – note that the only drives that will be able to process actual ATA task register commands **must** be attached to a controller that Windows sees as an *IDE ATA/ATAPI controller*

In the example below, the second controller (Adaptec Serial ATA 1205A Host Controller) *might* be able to use SAT to send and ATA command to an attached drive.



Does my controller card support SAT?

The easiest way to determine if your controller supports SAT is to use the STB Suite SCSI User Defined CDB to try issuing a SAT command.

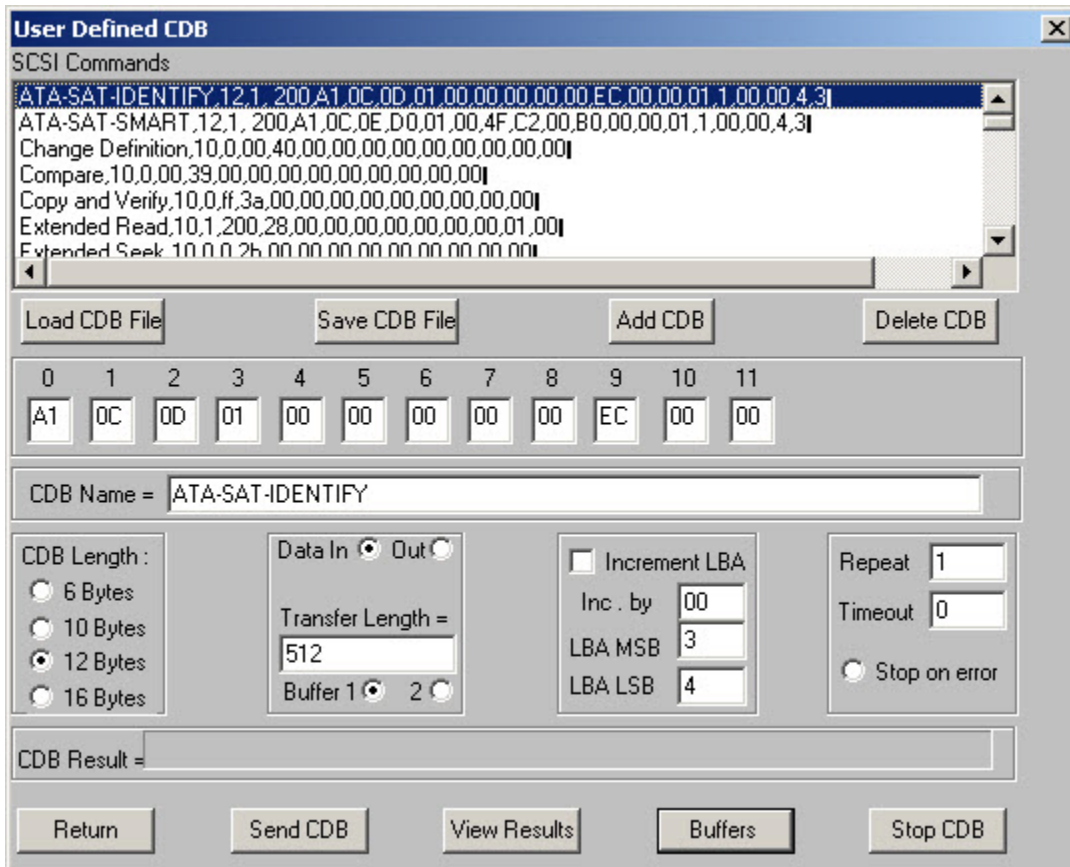
The 12-byte ATA Passthrough CDB we will use is defined as:

Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A1h)							
1	MULTIPLE_COUNT			PROTOCOL				Reserved
2	OFF_LINE	CK_COND	Reserved	T_DIR	BYTE_BLOCK	T_LENGTH		
3	FEATURES (7:0)							
4	SECTOR_COUNT (7:0)							
5	LBA_LOW (7:0)							
6	LBA_MID (7:0)							
7	LBA_HIGH (7:0)							
8	DEVICE							
9	COMMAND							
10	Reserved							
11								

There are some obscure aspects of using this command – rather than going into detail about them right now we will instead simply describe how to issue a command which will tell us immediately if the controller supports SAT or not. We will discuss the details of each parameter of this command later on in this article.

A Simple test command

Here is a picture of the STB Suite User Defined CDB with an ATA Passthrough command defined which will issue an ATA IDENTIFY command to the attached drive



Select your target drive on the SATA controller that you hope will support SAT. Then right-click on the drive and choose **User Defined CDBs**, then enter the command exactly as show above. Click the **Send CDB** button to issue the command – the CDB Result field will tell if the command was successfully issued (congratulations, your controller implements SAT!) or if it failed (sorry, you won't be able to use SAT with this controller)

If the command completed successfully you can click the **Buffer** button so view the ATA IDENTIFY data returned from the drive.

Size 1048576 (1024Kb)

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
000000	7A	42	FF	3F	37	C8	10	00	00	00	00	00	3F	00	00	00	zB.??.....?...
000010	00	00	00	00	20	20	20	20	57	20	2D	44	4D	57	4D	41 W -DMWMA
000020	39	39	30	33	34	35	35	32	00	00	00	40	41	00	30	31	99034552...@A.01
000030	30	2E	45	31	31	30	44	57	20	43	44	57	30	38	4A	30	0.E110DW CDW08J0
000040	2D	44	35	37	53	4D	31	41	20	20	20	20	20	20	20	20	-D57SM1A
000050	20	20	20	20	20	20	20	20	20	20	20	20	20	20	10	80	..
000060	00	00	00	2F	01	40	00	00	00	00	07	00	FF	3F	10	00	.../.@.....?..
000070	3F	00	10	FC	FB	00	00	01	90	2F	50	09	00	00	07	00	?...../P.....

File Operations Fill Buffer All Zeros (0)

Buffer 1 Buffer 2

Return Cancel

Note: ATA commands data byte-swapped.

Issuing the ATA SMART command

To retrieve SMART data from this drive define your cdb like this:

The screenshot shows the 'User Defined CDB' application window. The title bar reads 'User Defined CDB'. The main area is titled 'SCSI Commands' and contains a list of commands. The command 'ATA-SAT-SMART,12,1,200,A1,0C,0E,D0,01,00,4F,C2,00,B0,00,01,1,00,00,4,3' is selected and highlighted in blue. Below the list are four buttons: 'Load CDB File', 'Save CDB File', 'Add CDB', and 'Delete CDB'. A buffer configuration section shows a grid of 12 buffers (0-11) with their respective hexadecimal values: 0: A1, 1: 0C, 2: 0E, 3: D0, 4: 01, 5: 00, 6: 4F, 7: C2, 8: 00, 9: B0, 10: 00, 11: 00. Below this is a text field for 'CDB Name = ATA-SAT-SMART'. The 'CDB Length' section has radio buttons for 6 Bytes, 10 Bytes, 12 Bytes (selected), and 16 Bytes. The 'Data In/Out' section has radio buttons for 'In' (selected) and 'Out'. The 'Transfer Length' is set to 512. The 'Buffer' section has radio buttons for 'Buffer 1' (selected) and '2'. The 'Increment LBA' section has a checkbox (unchecked) and input fields for 'Inc. by' (00), 'LBA MSB' (3), and 'LBA LSB' (4). The 'Repeat' section has a spin box set to 1 and a 'Timeout' spin box set to 0. There is a 'Stop on error' radio button (unchecked). The 'CDB Result' field shows 'Status Good - Command completed without error'. At the bottom are buttons for 'Return', 'Send CDB', 'View Results', 'Buffers', and 'Stop CDB'.

SCSI Commands

- ATA-SAT-IDENTIFY,12,1,200,A1,0C,0D,01,00,00,00,00,00,EC,00,00,01,1,00,00,4,3
- ATA-SAT-SMART,12,1,200,A1,0C,0E,D0,01,00,4F,C2,00,B0,00,01,1,00,00,4,3**
- Change Definition,10,0,00,40,00,00,00,00,00,00,00,00,00,00
- Compare,10,0,00,39,00,00,00,00,00,00,00,00,00,00
- Copy and Verify,10,0,ff,3a,00,00,00,00,00,00,00,00,00
- Extended Read,10,1,200,28,00,00,00,00,00,00,00,01,00
- Extended Seek,10,0,0,2h,00,00,00,00,00,00,00,00,00

Load CDB File Save CDB File Add CDB Delete CDB

0	1	2	3	4	5	6	7	8	9	10	11
A1	0C	0E	D0	01	00	4F	C2	00	B0	00	00

CDB Name = ATA-SAT-SMART

CDB Length :
 6 Bytes
 10 Bytes
 12 Bytes
 16 Bytes

Data In Out

Transfer Length = 512

Buffer 1 2

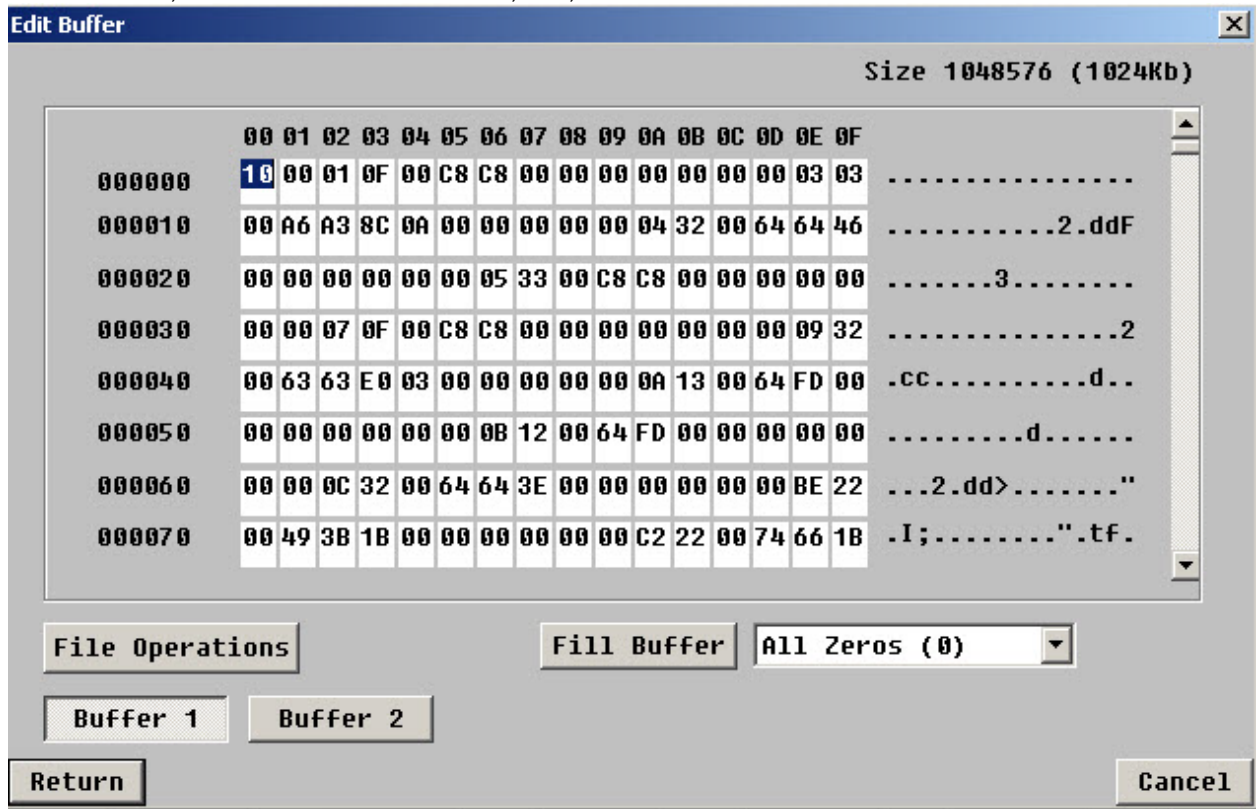
Increment LBA
Inc. by 00
LBA MSB 3
LBA LSB 4

Repeat 1
Timeout 0
 Stop on error

CDB Result = Status Good - Command completed without error

Return Send CDB View Results Buffers Stop CDB

And as before, the data is available to view, edit, or save to a file



Details of defining an IDENTIFY command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (A1h)							
1	MULTIPLE_COUNT			PROTOCOL				Reserved
2	OFF_LINE	CK_COND	Reserved	T_DIR	BYTE_BLOCK	T_LENGTH		
3	FEATURES (7:0)							
4	SECTOR_COUNT (7:0)							
5	LBA_LOW (7:0)							
6	LBA_MID (7:0)							
7	LBA_HIGH (7:0)							
8	DEVICE							
9	COMMAND							
10	Reserved							
11								

The bytes of the command we constructed to issue an IDENTIFY command were:

0xA1, 0x0C, 0x0D, 00,00,00,00,00,0xEC,00,00

Byte 0:

Looking at the command description we see that the first byte is the SCSI op code.

Byte 1:

the next byte is used to specify the protocol, as defined in this table:

Code	Description
0	ATA hardware reset
1	SRST
2	Reserved
3	Non-data
4	PIO Data-In
5	PIO Data-Out
6	DMA
7	DMA Queued
8	Device Diagnostic
9	DEVICE RESET
10	UDMA Data In
11	UDMA Data Out
12	FPDMA ^a
13, 14	Reserved
15	Return Response Information

^a See SATA-2.6.

In our command the 0x0C says we are requesting a DMA transfer. Note that you need to take care when defining this byte because of the offset caused by bit 0 being reserved.

Byte 2:

This byte is used to define how much data we are expecting, how the amount is specified, and which further bytes of the command will be used to specify. In the case of our IDENTIFY command we use 0x0D, which specifies that T_DIR = 1, BYTE_BLOCK=1, and T_LENGTH = 1. Referring to the SAT specification we see:

If the T_DIR bit is set to zero, then the SATL shall transfer data from the application client to the ATA device. If the T_DIR bit is set to one, then the SATL shall transfer data from the ATA device to the application client. The SATL shall ignore the T_DIR bit if the T_LENGTH field is set to zero.

T_DIR = 1 says that the data direction will be receiving data from the drive.

The BYTE_BLOCK (Byte/Block) bit specifies whether the transfer length in the location specified by the T_LENGTH field specifies the number of bytes to transfer or the number of blocks to transfer. If the value in the BYTE_BLOCK bit is set to zero, then the SATL shall transfer the number of bytes specified in the location specified by the T_LENGTH field. If the value in the BYTE_BLOCK bit is set to one the SATL shall transfer the number of blocks specified in the location specified by the T_LENGTH field. The SATL shall ignore the BYTE_BLOCK bit when the T_LENGTH field is set to zero.

BYTE_BLOCK=1 says that we are expecting to transfer one block (512 bytes) of data.

The Transfer Length (T_LENGTH) field specifies where in the CDB the SATL shall locate the transfer length for the command (see table 98).

Table 98 — T_LENGTH field

Code	Description
00b	No data is transferred
01b	The transfer length is an unsigned integer specified in the FEATURES (7:0) field.
10b	The transfer length is an unsigned integer specified in the SECTOR_COUNT (7:0) field.
11b	The transfer length is an unsigned integer specified in the TPSIU (see 3.1.98).

Finally, T_LENGTH = 01 tells us that our transfer length is going to be specified by the integer placed in the ATA FEATURES byte field – which in this case is byte 3 of the CDB.

Byte 3:

-as we just said – because of our T_LENGTH setting we have specified that this byte will contain the number of blocks (because of the BYTE_BLOCK setting) of data that will be transferred, in the direction specified by T_DIR.

Byte 4:

– contains the ATA task register SECTOR COUNT data, in this case we are transferring 1 block so this must be set to 1.

Byte 5:

- contains the ATA task register LBA LOW byte – the IDENTIFY command needs this to be 0.

Byte 6:

- contains the ATA task register LBA MID byte – the IDENTIFY command needs this to be 0.

Byte 7:

- contains the ATA task register LBA HIGH byte – the IDENTIFY command needs this to be 0.

Byte 8:

- contains the ATA task register DEVICE byte – the IDENTIFY command needs this to be 0.

Byte 9:

- contains the ATA task register COMMAND byte – the IDENTIFY command is 0xEC.

Bytes 10 and 11 :

-are reserved and so set to 0.

Details of defining a SMART command

Refer to the T10 SAT specification documentation.

Note that we specify T_LENGTH = 10, which uses the SECTOR COUNT field (Byte 4) to specify our data length. Why did we need to do this, rather than use the FEATURES field (Byte 3) like we did for the IDENTIFY command?

We had to do this because the ATA SMART command needs to use the FEATURES field to define which type of SMART command we are sending – 0xD0 in this case.

Conclusion

SAT is a versatile if complicated method of issuing ATA commands to drives which are connected to controllers which Windows thinks are SCSI type. It is preferable rather than being forced to implementing controller vendor-unique pass through methods.

SAT is not universally supported or implemented. All current SAS controllers manufactured by LSI, including the 8888, 3800, 3801, and PERC cards do implement SAT and so allow access to “raw” ATA commands which would otherwise not be usable.

STB Suite Version 8.0 New Features

Please Note:

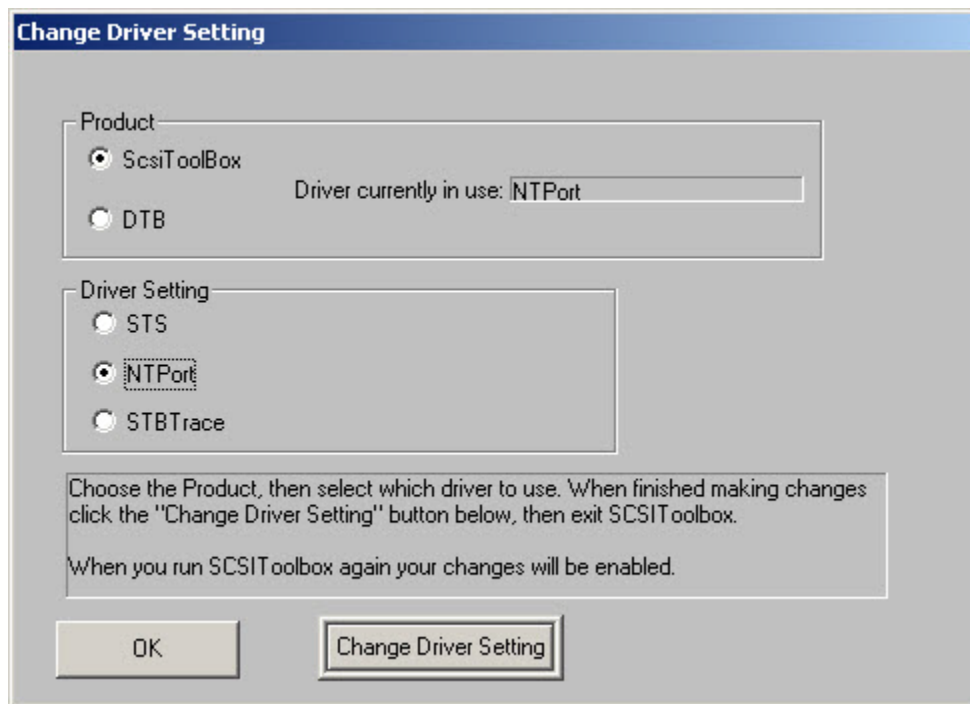
The new STBTrace driver used by BAM, the Reset Tool, and as an option in STB is only available under Windows XP and Server 2003 and above operating systems.

STB

New Features:

STB New STBTrace driver

This new driver allows identical operation under both 32-bit and 64-bit operating systems. This driver supports Command Tag Queuing and large data transfers. You may select this driver, the standard Windows ntpor driver, or the previous STB STS driver from the Original mode STB main Options menu:

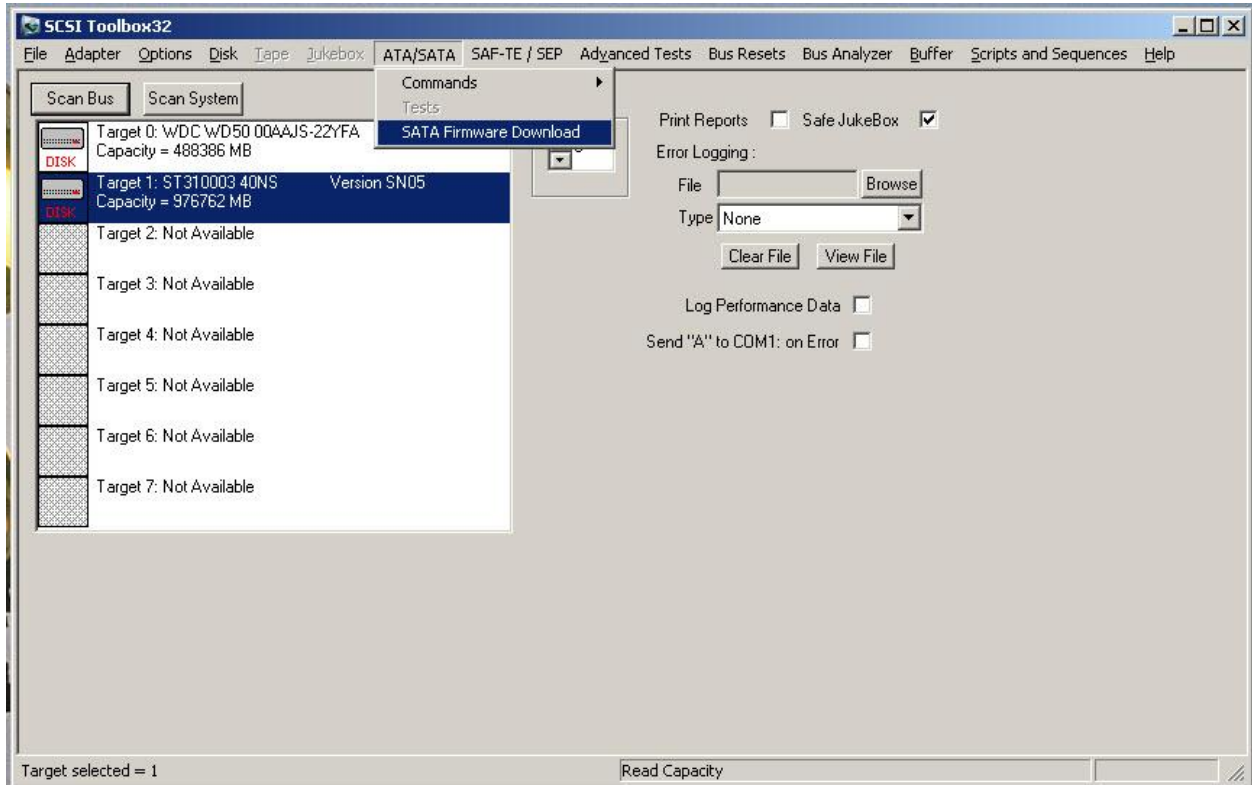


SATA Firmware Download

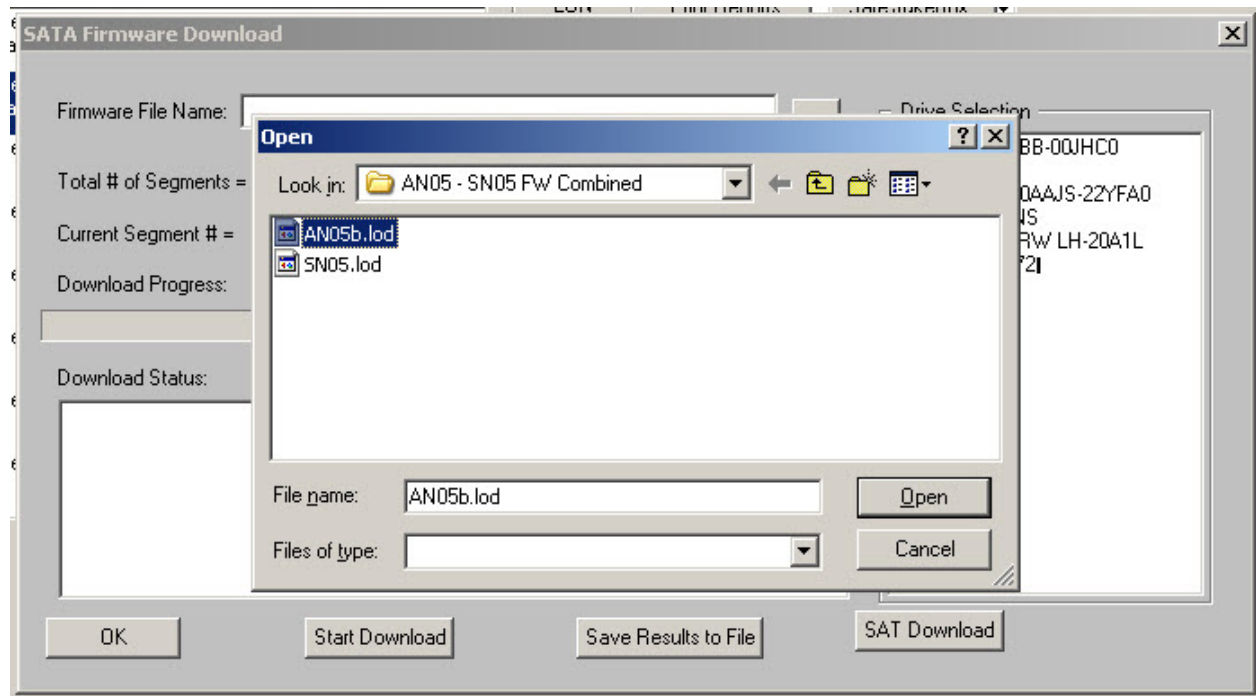
Downloading firmware via a Native SATA port

Start the STB Suite in Original Mode

Select the top menu SATA/ATA -> SATA Firmware Download choice



1. Select your firmware file and drive to download to

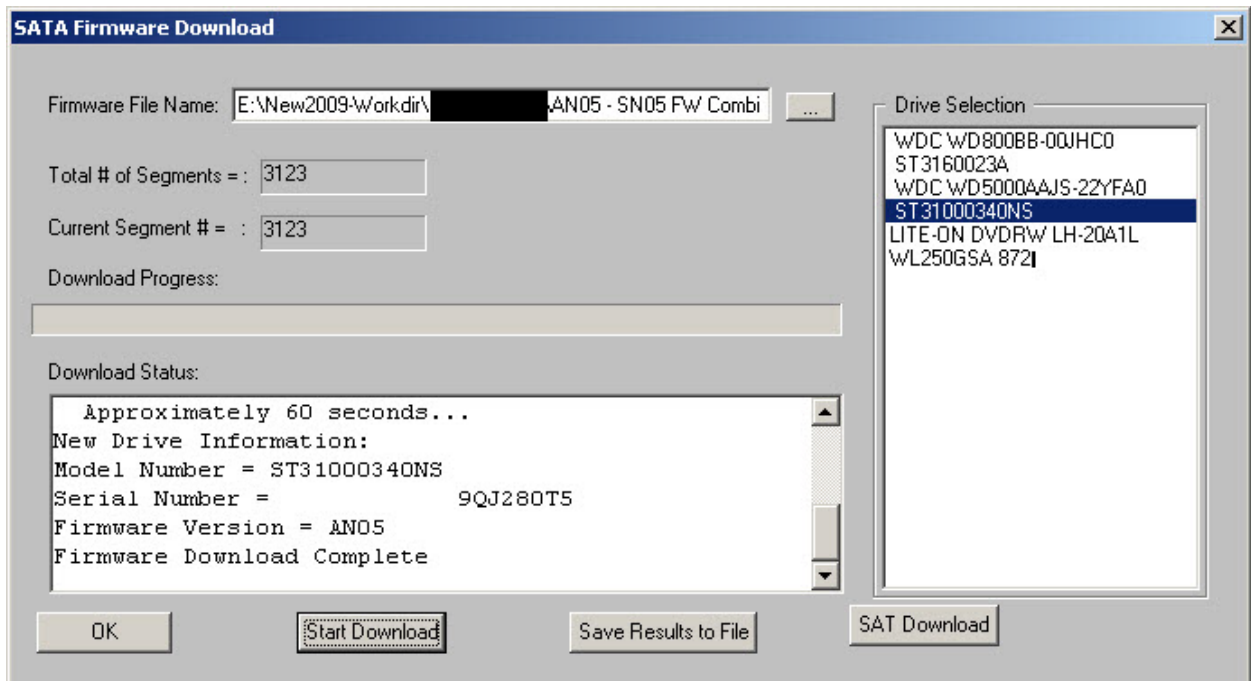


2. Click “Start Download” to begin the download process...

3. STB will confirm that drive can accept a firmware download, and will ask you to confirm that you want to proceed with the download...

**4. As the download progresses the Current Segment # field will increment...
Until the download is complete.**

5. The new drive IDENTIFY information will be displayed in the Download Status window:



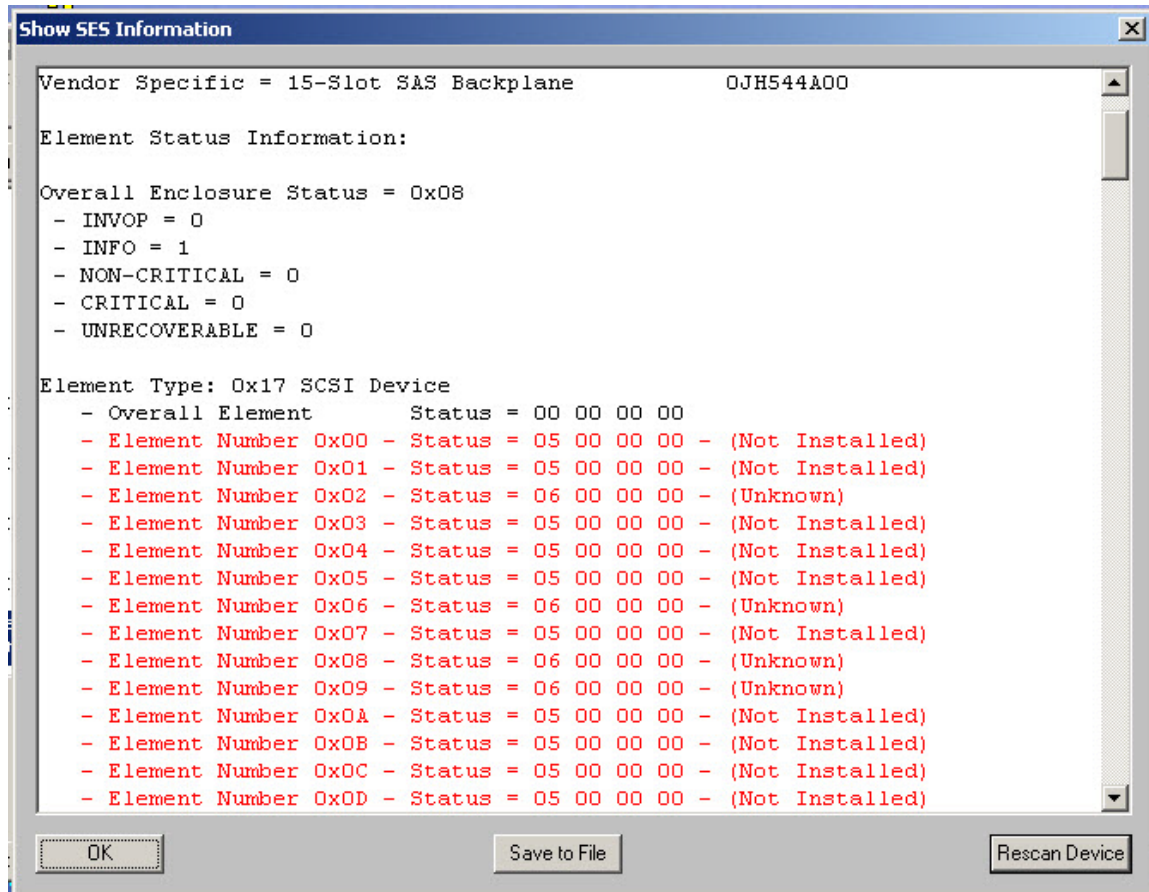
Note: Most Windows systems will need to be rebooted at this point for the drive to be recognized and available for testing. This is a Windows requirement.

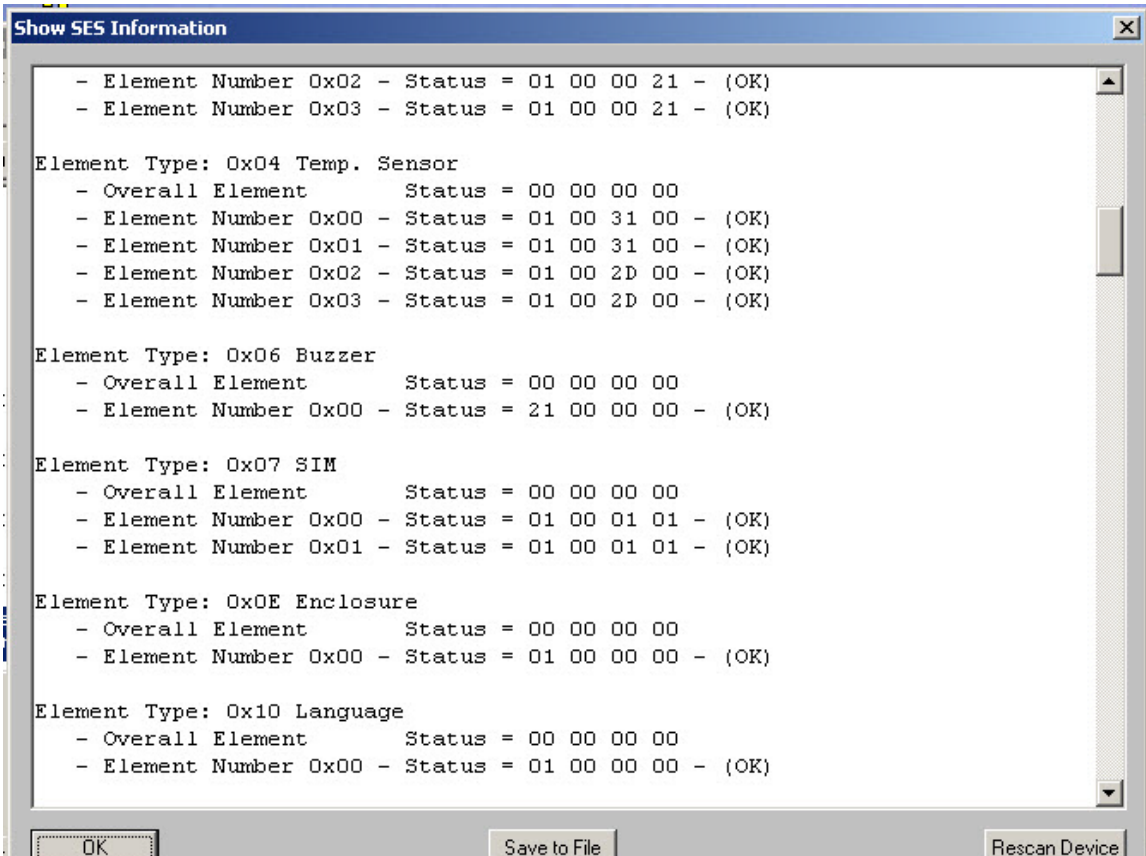
SATA/ATA Command Compliance Testing

See the reprint of the article SATA Command Compliance Testing at the end of this document.

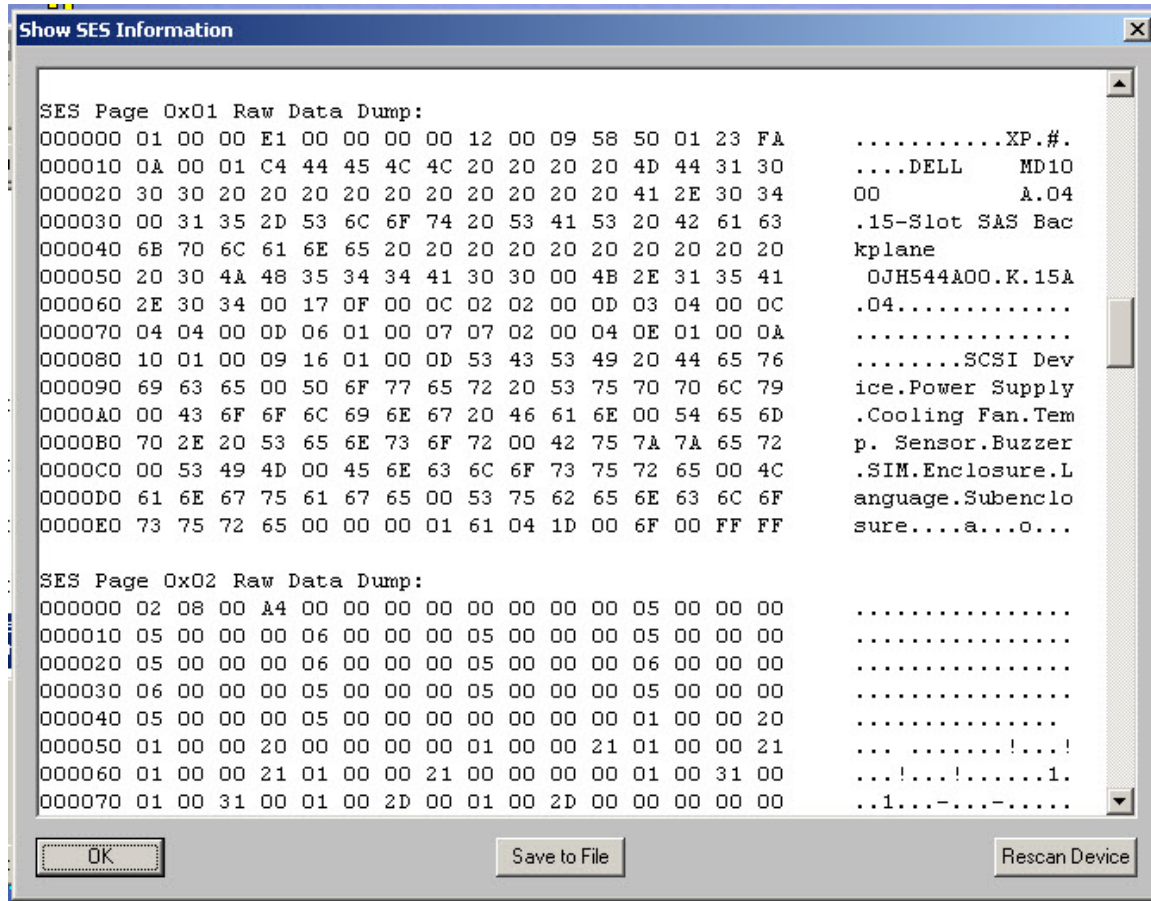
SES Device Configuration display

Enclosure processor information on all elements is displayed – right-click on the SES device in the display window and choose SES Data:





Raw data for all elements is also displayed:



Bug Fixes/Enhancements:

- Fixed problem with USB commands that have no data phase

DMM

New Features:

“STOP ON ERROR” OPTIONS in DMM

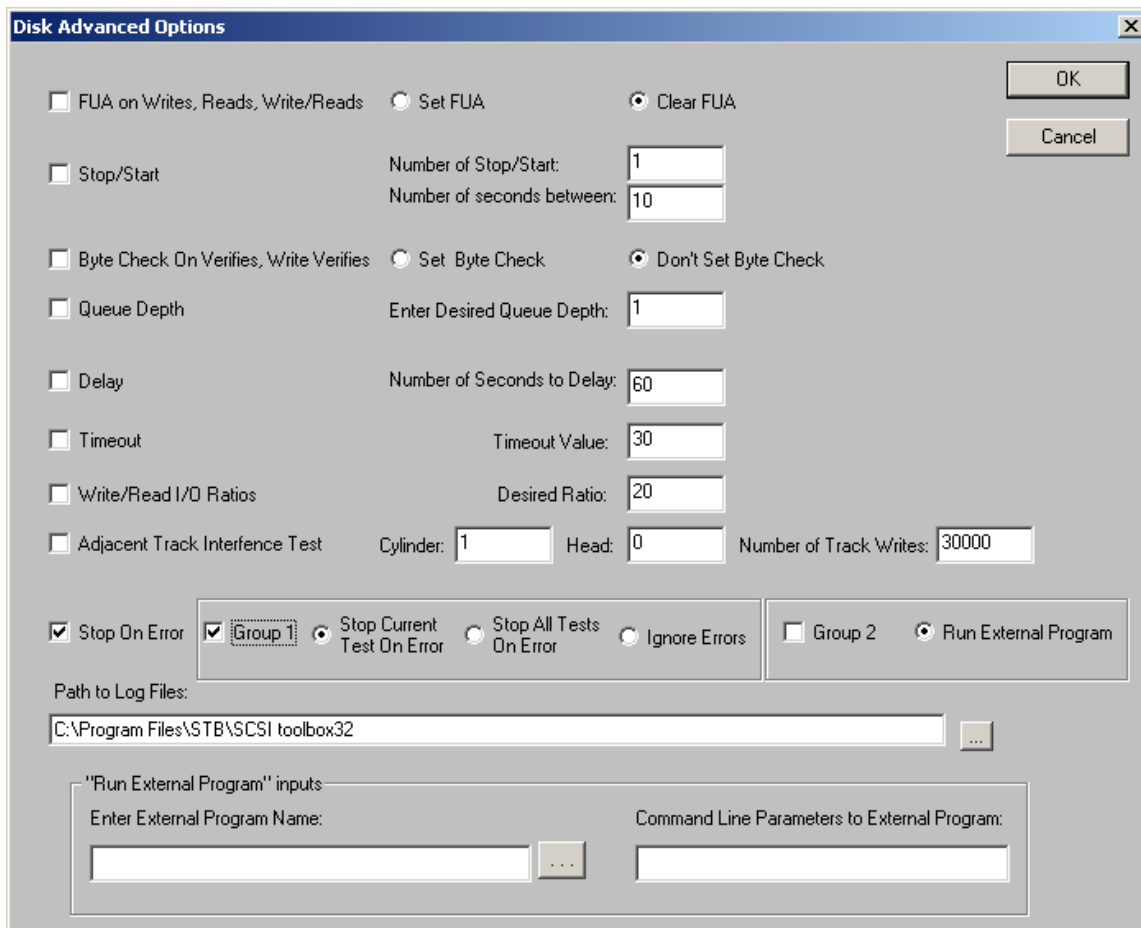
The “Stop On Error” options, which can be set on the “Advanced Options” dialog inside of DMM, were designed to allow you to accomplish the following six tasks:

- A) If an error occurs to stop the current test in the test sequence
- B) If an error occurs to stop all tests in the test sequence
- C) If an error occurs to simply ignore the error
- D) If an error occurs to run an “External Program”
- E) If an error occurs to run an “External Program” and if the “External Program” fails to stop the current test
- F) If an error occurs to run an “External Program” and if the “External Program” fails to stop all tests
- G) If an error occurs to run an “External Program” and continue executing the current test

Let’s take a closer look at each of the above options.

Option A) If an error occurs to stop the current test in the test sequence

On the “Advanced Options” dialog in DMM, you set this option as seen in the following picture:

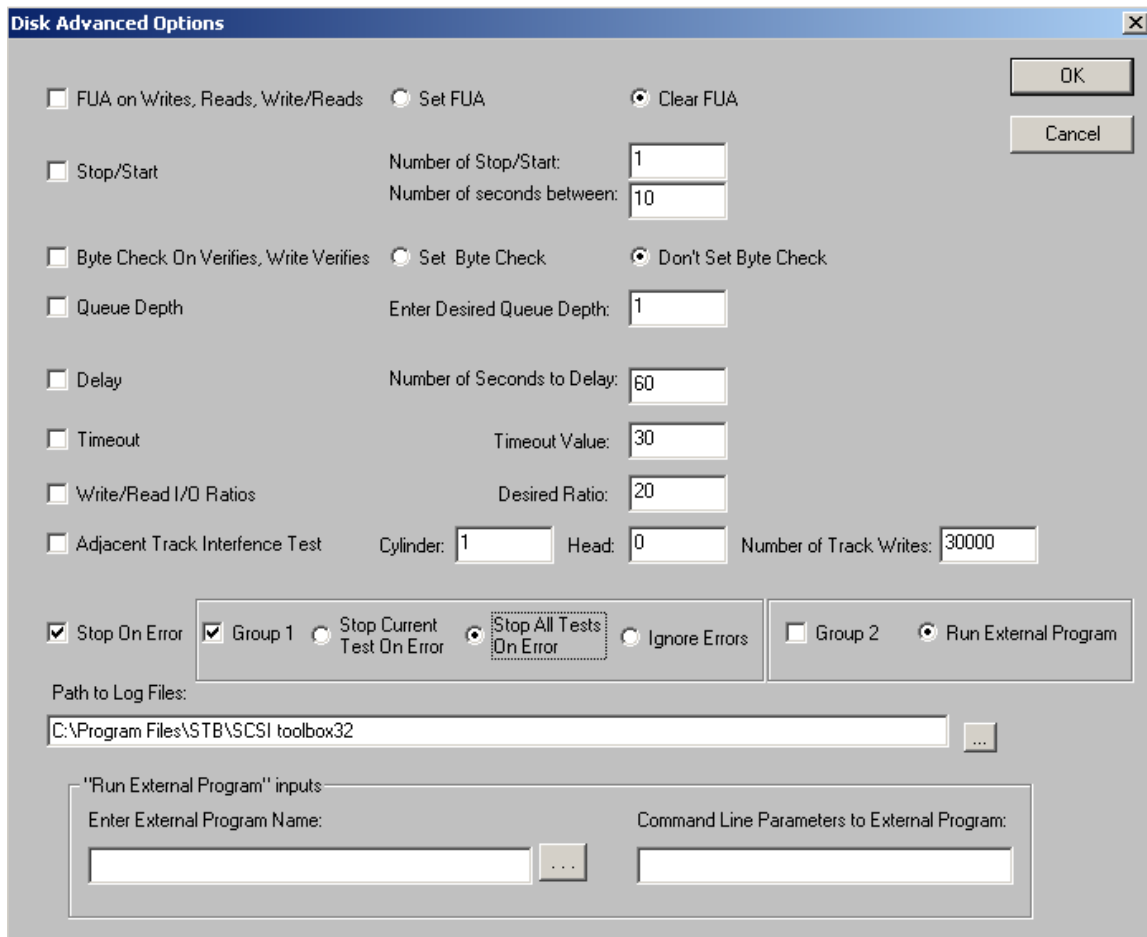


Notice that the checkbox “Stop On Error” is checked, “Group 1” is checked, and “Group 2” is unchecked

As an example, let’s say you have a 7-step test sequence, and step 3 is a sequential write test for 10 minutes, and that step 3 is currently executing when an error occurs at say the 5 minute mark of the test. What this option will do is stop the current test (i.e. step 3 of 7 steps, will NOT continue the remaining 5 minutes of the sequential write test, and proceed to step 4 of your 7 step test sequence).

Option B) If an error occurs to stop all tests in the test sequence

On the “Advanced Options” dialog in DMM, you set this option as seen in the following picture:

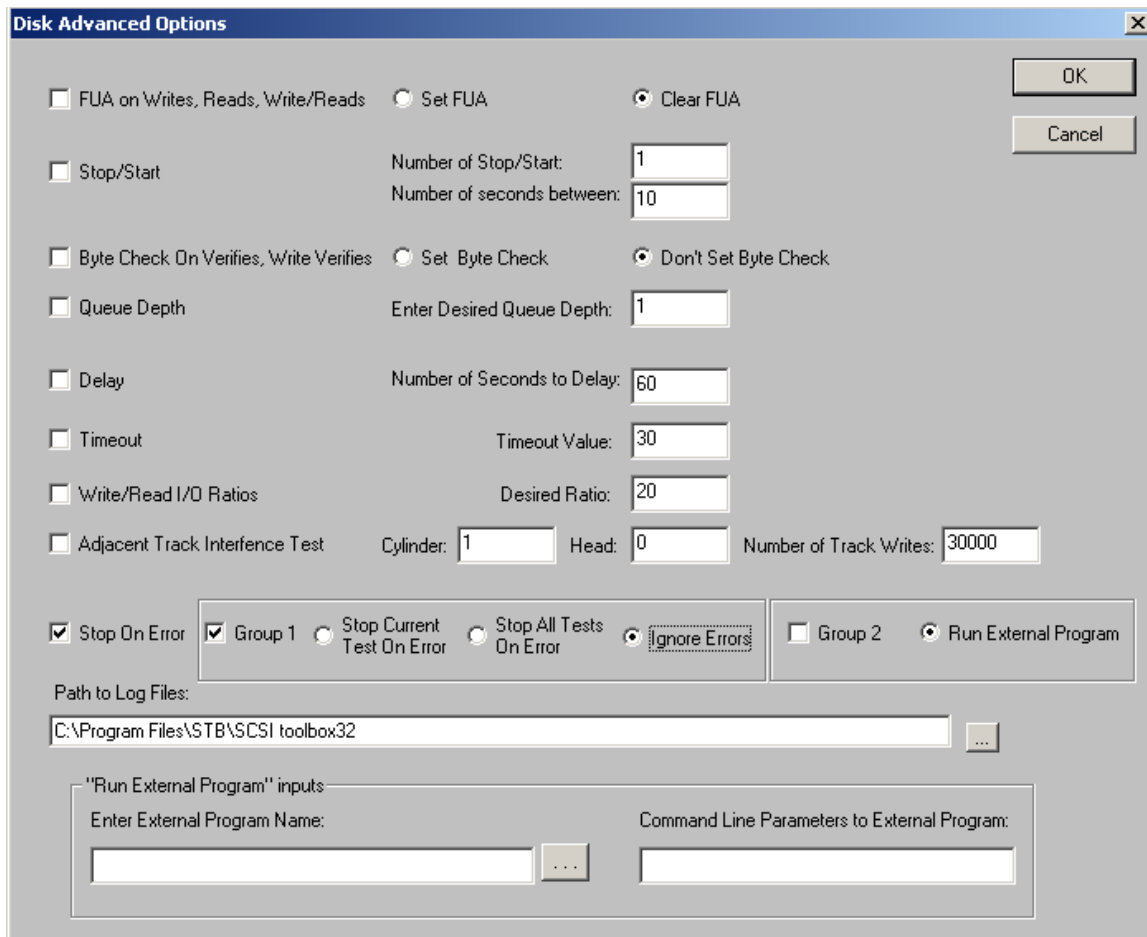


Notice that the checkbox “Stop On Error” is checked, “Group 1” is checked, and “Group 2” is unchecked

As an example, let’s say you have a 7-step test sequence, and step 3 is a sequential write test for 10 minutes, and that step 3 is currently executing when an error occurs at say the 5 minute mark of the test. What this option will do is stop all tests (i.e. it will stop step 3 and will NOT continue the remaining 5 minutes of the sequential write test, and will NOT execute steps 4-thru-7).

Option C) If an error occurs to simply ignore the error

On the “Advanced Options” dialog in DMM, you set this option as seen in the following picture:

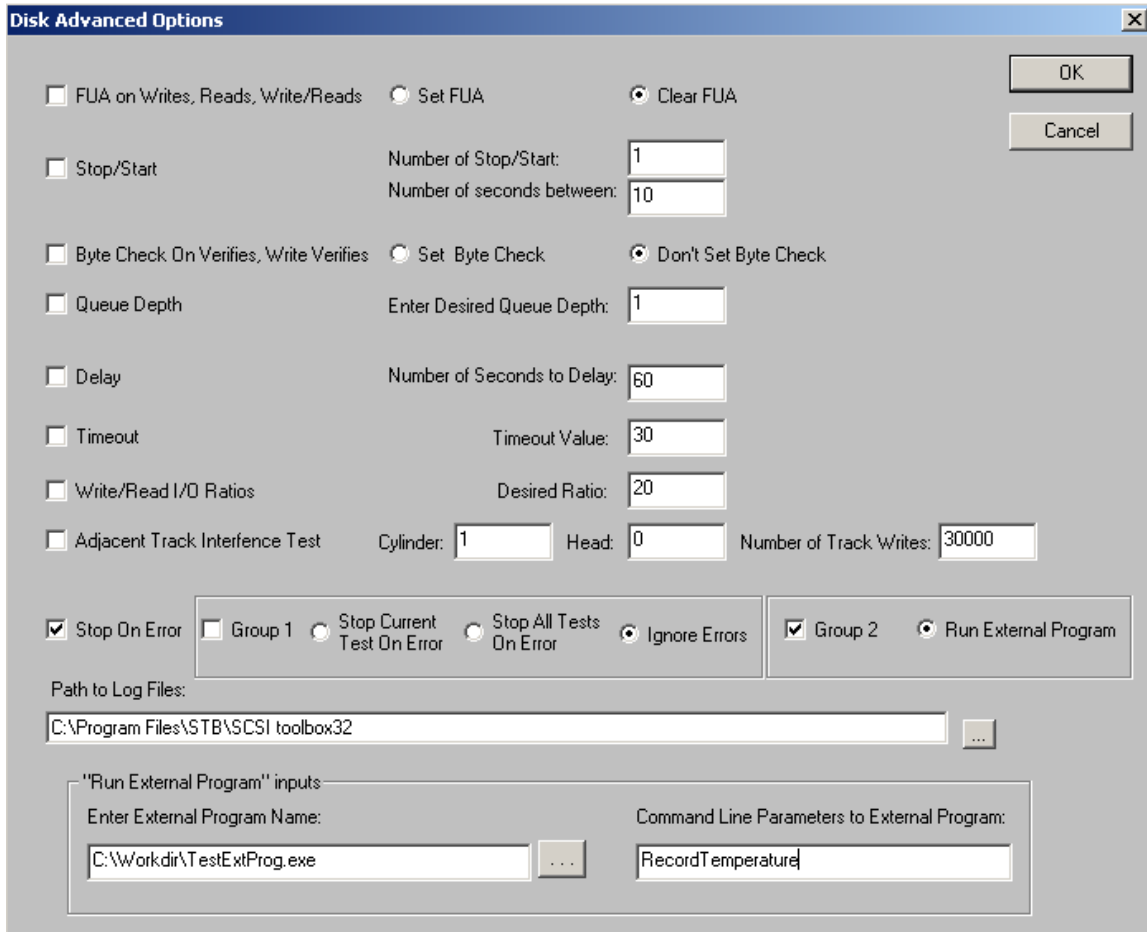


Notice that the checkbox “Stop On Error” is checked, “Group 1” is checked, and “Group 2” is unchecked

As an example, let’s say you have a 7-step test sequence, and step 3 is a sequential write test for 10 minutes, and that step 3 is currently executing when an error occurs at say the 5 minute mark of the test. What this option will do is simply ignore the error (i.e. it will continue step 3 and will continue with the remaining 5 minutes of the sequential write test, and after this step 3 has completed proceed to step 4 of your 7 step test sequence).

Option D) If an error occurs to run an “External Program”

On the “Advanced Options” dialog in DMM, you set this option as seen in the following picture:



Notice that the checkbox "Stop On Error" is checked, "Group 1" is unchecked, and "Group 2" is checked

As an example, let's say you have a 7-step test sequence, and step 3 is a sequential write test for 10 minutes, and that step 3 is currently executing when an error occurs at say the 5 minute mark of the test. What this option will do is run the "External Program" that you inputted on the "Advanced Options" dialog, and pass, as a command line argument, the following information:

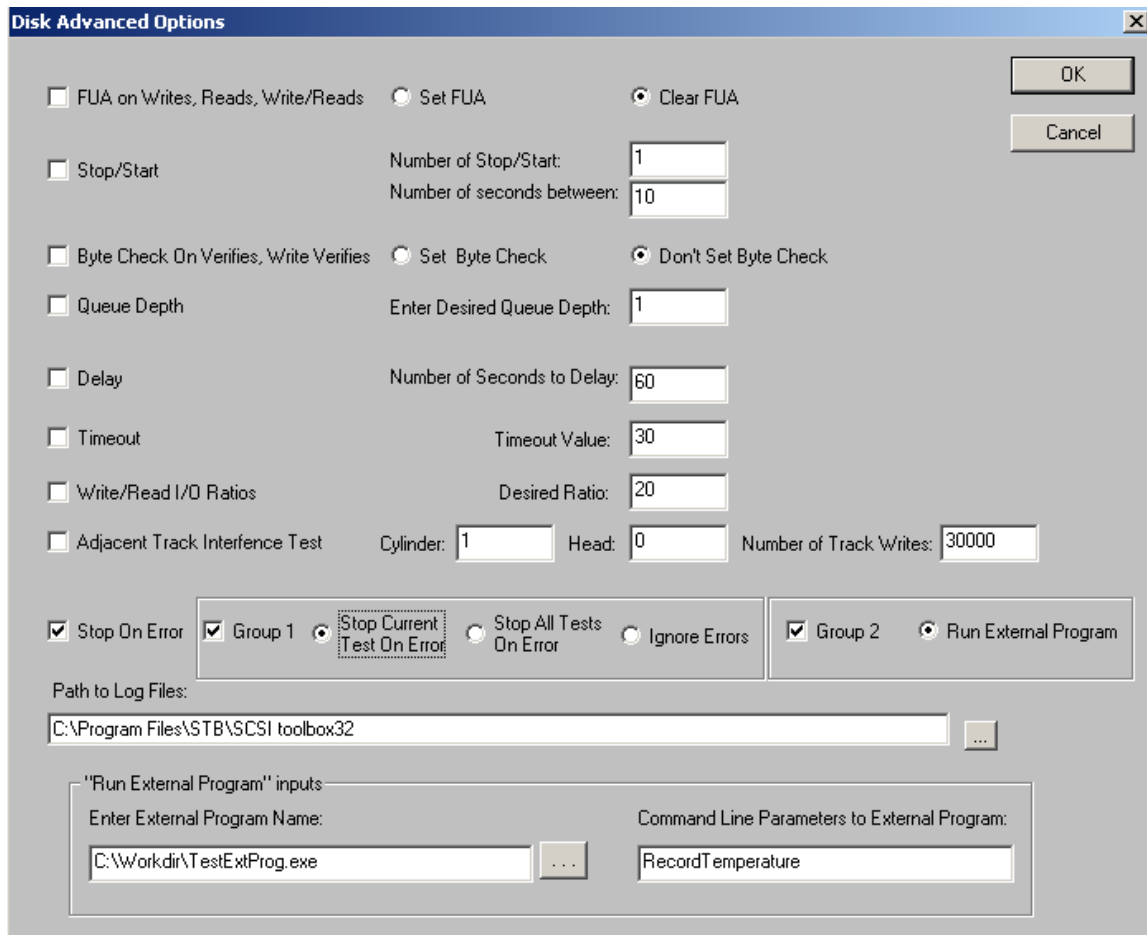
HBA=n,TID=m,LUN=x,SLOT=255,RecordTemperature.

IMPORTANT: Notice that preceding the command line parameters you inputted on the "Advanced Options" dialog are the device address of the device that had the error. After the device address information, we append the string "RecordTemperature" at the end of the command line argument to the "External Program".

After the "External Program" completes, the return code from the External Program logged to the devices logfile and step 3 is stopped (so the remaining 5 minutes of the sequential write test is not performed). Then we proceed to step 4 of your 7 step test sequence.

Option E) If an error occurs to run an "External Program" and if the "External Program" fails to stop the current test

On the "Advanced Options" dialog in DMM, you set this option as seen in the following picture:



Notice that the checkbox “Stop On Error” is checked, “Group 1” is checked, and “Group 2” is checked. Also note that the “Enter External Program Name” edit box is filled in, and so is the “Command Line Parameters to External Program” edit box is filled in.

As an example, let’s say you have a 7-step test sequence, and step 3 is a sequential write test for 10 minutes, and that step 3 is currently executing when an error occurs at say the 5 minute mark of the test. What this option will do is run the “External Program” that you inputted on the “Advanced Options” dialog, and pass, as a command line argument, the following information:

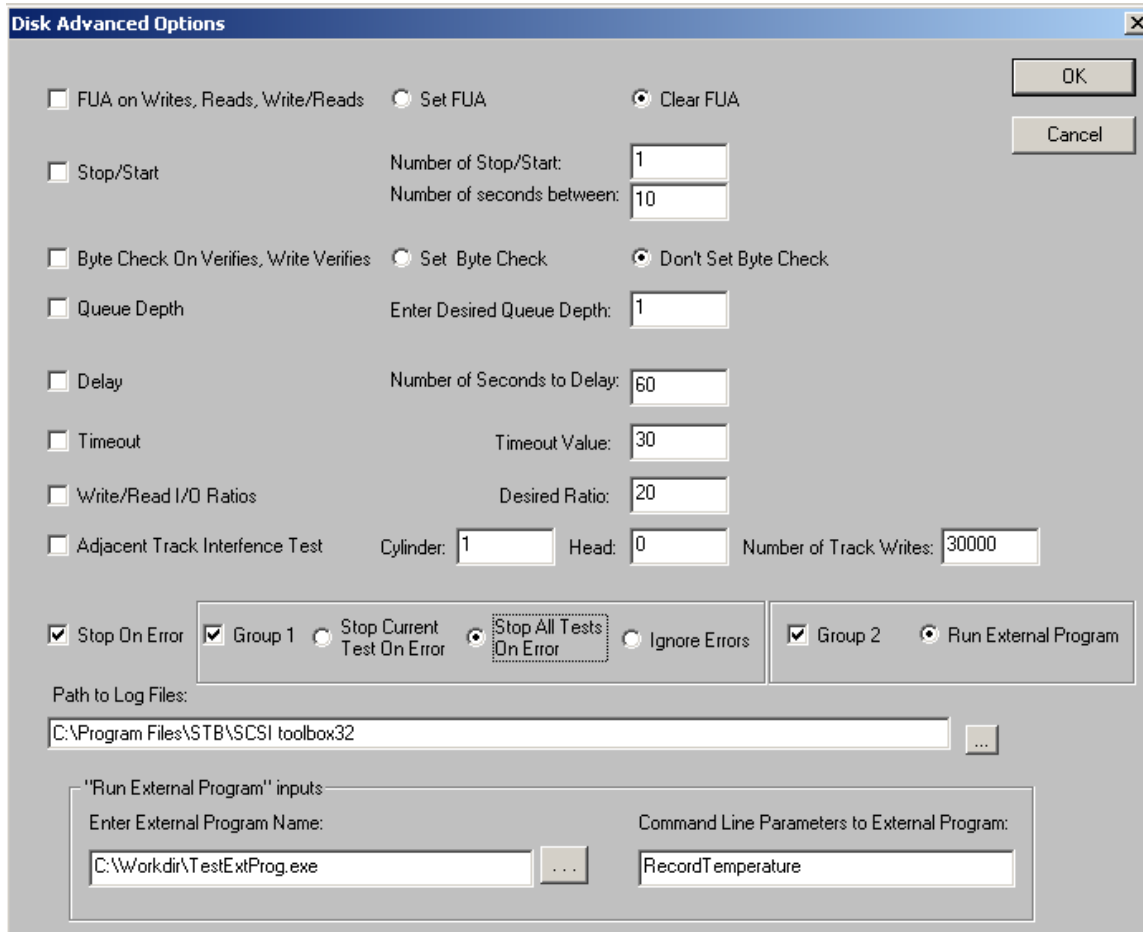
```
HBA=n,TID=m,LUN=x,SLOT=255,RecordTemperature.
```

IMPORTANT: Notice that preceeding the command line parameters you inputted on the “Advanced Options” dialog are the device address of the device that had the error. After the device address information, we append the string “RecordTemperature” at the end of the command line argument to the “External Program”.

After the “External Program” completes, the return code from the External Program logged to the devices logfile. **THEN** the return code from the “External Program” is analyzed for SUCCESS (return code of 0) or FAILURE (return code of anything other than 0). **IF** the return code from the “External Program” is 0 (SUCCESS) then step 3 is continued. **IF** the return code from the “External Program” is non-zero (FAILURE), step 3 is stopped (so the remaining 5 minutes of the sequential write test is not performed). Then we proceed to step 4 of your 7 step test sequence.

Option F) If an error occurs to run an “External Program” and if the “External Program” fails to stop all tests

On the “Advanced Options” dialog in DMM, you set this option as seen in the following picture:



Notice that the checkbox “Stop On Error” is checked, “Group 1” is checked, and “Group 2” is checked. Also note that the “Enter External Program Name” edit box is filled in, and so is the “Command Line Parameters to External Program” edit box is filled in.

As an example, let’s say you have a 7-step test sequence, and step 3 is a sequential write test for 10 minutes, and that step 3 is currently executing when an error occurs at say the 5 minute mark of the test. What this option will do is run the “External Program” that you inputted on the “Advanced Options” dialog, and pass, as a command line argument, the following information:

HBA=n,TID=m,LUN=x,SLOT=255,RecordTemperature.

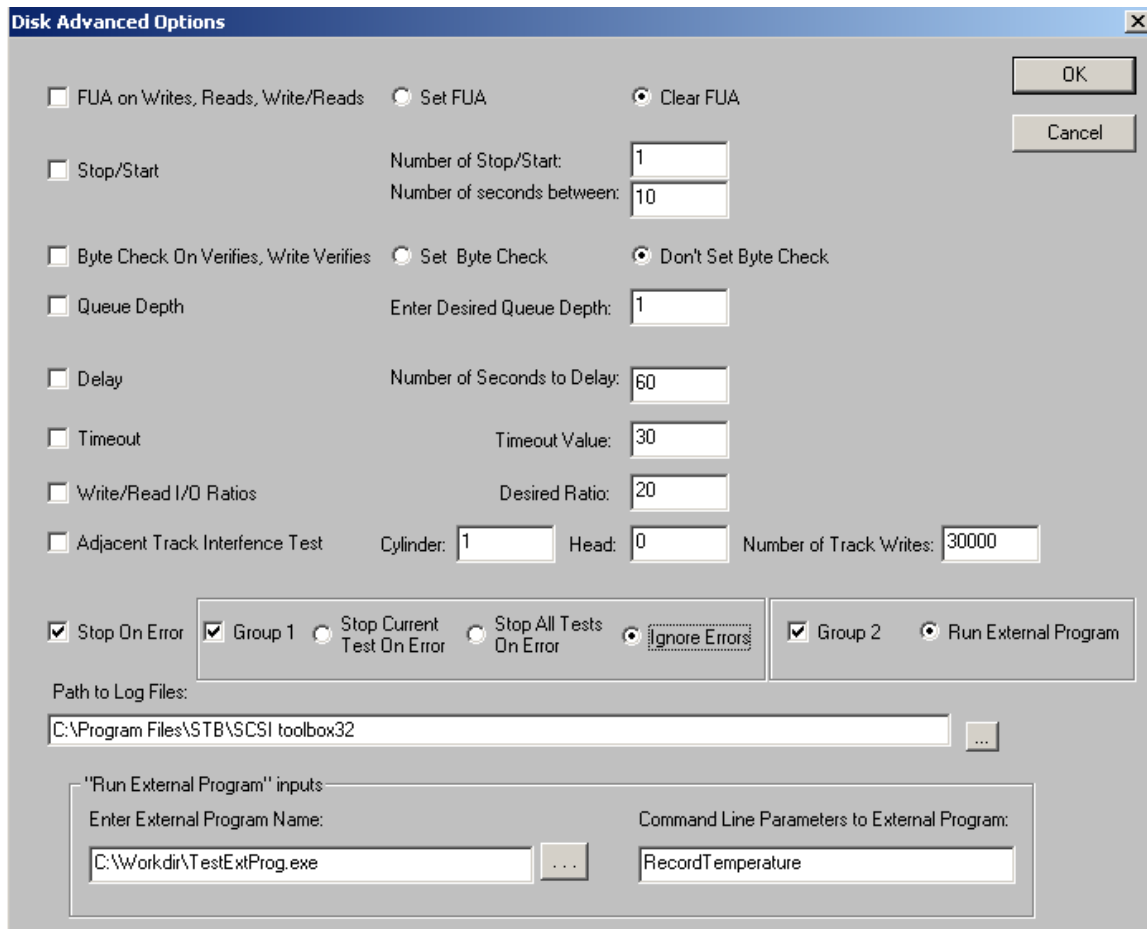
IMPORTANT: Notice that preceding the command line parameters you inputted on the “Advanced Options” dialog are the device address of the device that had the error. After the device address information, we append the string “RecordTemperature” at the end of the command line argument to the “External Program”.

After the “External Program” completes, the return code from the External Program is logged to the device’s logfile. **THEN** the return code from the “External Program” is analyzed for SUCCESS (return

code of 0) or FAILURE (return code of anything other than 0). **IF** the return code from the “External Program” is 0 (SUCCESS) then step 3 is continued. **IF** the return code from the “External Program” is non-zero (FAILURE), step 3 is stopped (so the remaining 5 minutes of the sequential write test is not performed). No other tests are executed on this device.

Option G) If an error occurs to run an “External Program” and continue executing the current test

On the “Advanced Options” dialog in DMM, you set this option as seen in the following picture:



Notice that the checkbox “Stop On Error” is checked, “Group 1” is checked, and “Group 2” is checked. Also note that the “Enter External Program Name” edit box is filled in, and so is the “Command Line Parameters to External Program” edit box is filled in.

As an example, let’s say you have a 7-step test sequence, and step 3 is a sequential write test for 10 minutes, and that step 3 is currently executing when an error occurs at say the 5 minute mark of the test. What this option will do is run the “External Program” that you inputted on the “Advanced Options” dialog, and pass, as a command line argument, the following information:

HBA=n,TID=m,LUN=x,SLOT=255,RecordTemperature.

IMPORTANT: Notice that preceding the command line parameters you inputted on the “Advanced Options” dialog are the device address of the device that had the error. After the device address information, we append the string “RecordTemperature” at the end of the command line argument to the “External Program”.

After the “External Program” completes, the return code from the External Program is logged to the device’s logfile. Step 3 is continued.

ASYNCHRONOUS NOTIFICATIONS OPTIONS IN DMM

The user can asynchronously notify DMM of the following actions:

- 1) Run an “External Program”
- 2) Log information to a device’s DMM logfile
- 3) Stop testing on a drive

We’ll take a closer look at these three options:

Option 1) Run an “External Program”

Here the goal is to run an “external program” on a device asynchronously. This is different from having an “external program” as a test step (which is predefined in the test sequence). For example, suppose you have a sequential write test currently running on a drive and you want to run an external program. You can “on-the-spur-of-a-moment” notify DMM to execute an external program. The following code snippet shows you how to accomplish this:

```
char * pBuf = new char[256];
sprintf(pBuf, "CODE=2, ExtProg=%s, CmdLine=%s, HBA=%02d, TID=%03d, LUN=%02d",
        m_strExternalProgram, m_strCommandLine, m_nHBA, m_nTid, m_nLun);

COPYDATASTRUCT * pCDS = new COPYDATASTRUCT;
pCDS->dwData = 4;
pCDS->cbData = 256;
pCDS->lpData = pBuf;

CWnd * pWnd = CWnd::FindWindow(NULL, "Disk Manufacturing & Screening Module");
if (pWnd != NULL && ::IsWindow(pWnd->m_hWnd))
{
    pWnd->SendMessage(WM_COPYDATA, (WPARAM) NULL, (LPARAM) pCDS);
}
```

When DMM receives this asynchronous notification, it will parse the device address and then execute the external program name passed in, passing to the external program as a command line argument the device’s address plus the command line you pass DMM.

Option 2) Log information to a device’s DMM logfile

Here the goal is to asynchronously log information to a device’s DMM logfile. The following code snippet shows you how to accomplish this:

```
char * pBuf = new char[256];
sprintf(pBuf, "CODE=2, LogToFile=%s, HBA=%02d, TID=%03d, LUN=%02d",
        m_strDMMText, m_nHBA, m_nTid, m_nLun);

COPYDATASTRUCT * pCDS = new COPYDATASTRUCT;
pCDS->dwData = 5;
```

```

pCDS->cbData = strlen(pBuf)+1;
pCDS->lpData = pBuf;

CWnd * pWnd = CWnd::FindWindow(NULL,"Disk Manufacturing & Screening Module");
if (pWnd != NULL && ::IsWindow(pWnd->m_hWnd))
{
    pWnd->SendMessage(WM_COPYDATA, (WPARAM) NULL, (LPARAM) pCDS);
}

```

Option 3) Stop testing on a drive

Here the goal is to asynchronously stop all testing on a drive. The following code snippet shows you how to accomplish this:

```

char * pBuf = new char[256];
sprintf(pBuf, "CODE=2, HBA=%02d, TID=%03d, LUN=%02d",
        m_nHBA, m_nTid, m_nLun);

COPYDATASTRUCT * pCDS = new COPYDATASTRUCT;
pCDS->dwData = 6;
pCDS->cbData = strlen(pBuf)+1;
pCDS->lpData = pBuf;

CWnd * pWnd = CWnd::FindWindow(NULL,"Disk Manufacturing & Screening Module");
if (pWnd != NULL && ::IsWindow(pWnd->m_hWnd))
{
    pWnd->SendMessage(WM_COPYDATA, (WPARAM) NULL, (LPARAM) pCDS);
}

```

SATA Firmware download via DMM

Note: SAT requirement:

To download SATA drive firmware via DMM the SATA drive **must** be attached to a SAS controller which implements the SCSI-ATA-Translation (SAT) protocol. All LSI SAS cards implement SAT and may be used for SATA firmware download via DMM.

1. Select the SATA FW Download test type

Random or Sequential
 Random Sequential Butterfly

Type of Test
 Verify Write Write/Verify
 Read Write/Read Seek
 FW Download Up
 Synchronize Down
 PList Set
 Clear Log Pages Read
 DST Write/Verify workstation
 File Server Web server workstation
 Delay Ad' kIntf SMART
 SATA FW Download

Download File/External Program Executable
Browse

Stop Test After
 Time 100 % Start LBA
 Blocks

Fixed/Random
 Fixed 128 max blk
 Random

Test Data
Data Pattern Decrementing
 Compare on Reads Overlay Block #

Add This Test to Test Sequence
Advanced Options
Modify Current Test Step

Once you have specified the firmware file location and selected the SATA FW Download test, click the ***"Add This Test to Test Sequence"*** button...

2. Select your drive(s)
3. Start the DMM test sequence to download firmware

Screening on SATA SMART Thresholds for DMM

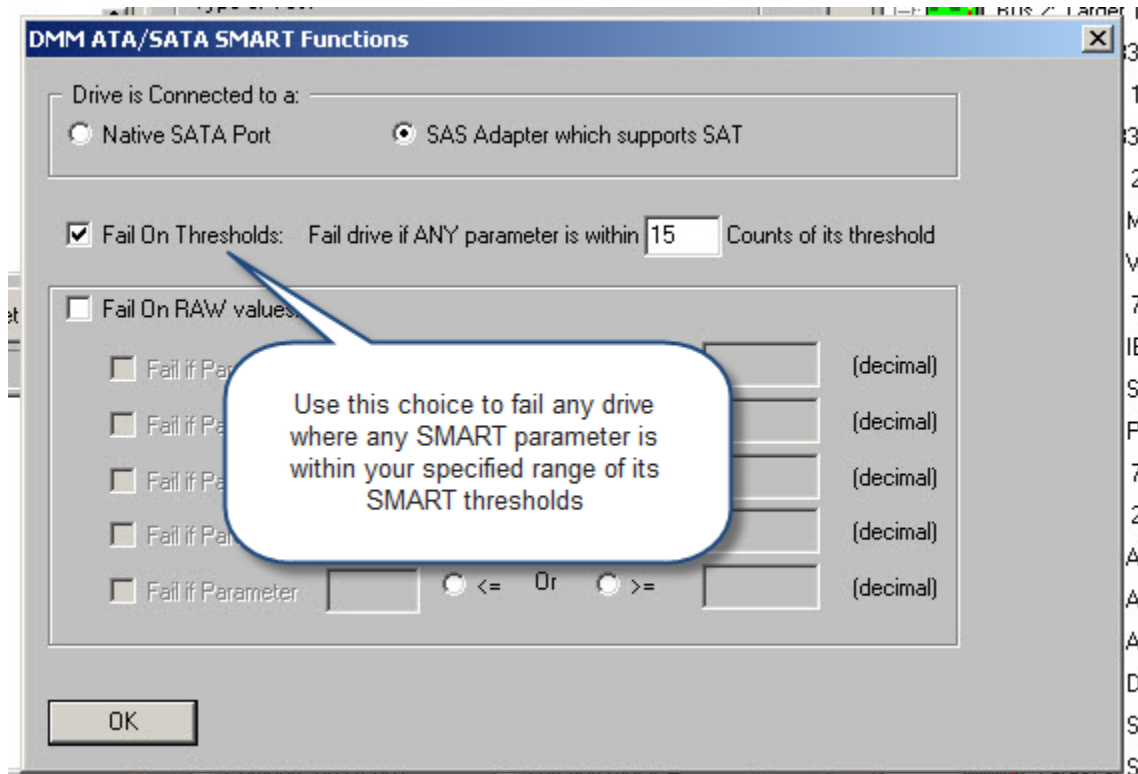
Note: SAT requirement:

To download SATA drive firmware via DMM the SATA drive **must** be attached to a SAS controller which implements the SCSI-ATA-Translation (SAT) protocol. All LSI SAS cards implement SAT and may be used for SATA firmware download via DMM.

1. Select the new SMART DMM test type, the following screen will be displayed where you may specify how you want to screen SATA drives using their SMART data.

Note: You may also use this test type to simply log the drive SMART data and not to fail a drive on any thresholds.

1. Screening on any SMART parameter approaching its threshold value



2. Screening on specific SMART parameter/data values

DMM ATA/SATA SMART Functions

Drive is Connected to a: Native SATA Port SAS Adapter which supports SAT

Fail On Thresholds: Fail drive if ANY parameter is within Counts of its threshold

Fail On RAW values:

<input checked="" type="checkbox"/> Fail if Parameter	<input type="text" value="2"/>	<input checked="" type="radio"/> <=	Or	<input type="radio"/> >=	<input type="text" value="150"/>	(decimal)
<input type="checkbox"/> Fail if Parameter	<input type="text"/>	<input type="radio"/> <=	Or	<input type="radio"/> >=	<input type="text"/>	(decimal)
<input type="checkbox"/> Fail if Parameter	<input type="text"/>	<input type="radio"/> <=	Or	<input type="radio"/> >=	<input type="text"/>	(decimal)
<input type="checkbox"/> Fail if Parameter	<input type="text"/>	<input type="radio"/> <=	Or	<input type="radio"/> >=	<input type="text"/>	(decimal)
<input type="checkbox"/> Fail if Parameter	<input type="text"/>	<input type="radio"/> <=	Or	<input type="radio"/> >=	<input type="text"/>	(decimal)

You may specify up to 5 Parameter/Data pairs, that can be greater than or less than your specified value...

OK

DMM ATA/SATA SMART Functions

Drive is Connected to a: Native SATA Port SAS Adapter which supports SAT

Fail On Thresholds: Fail drive if ANY parameter is within Counts of its threshold

Fail On RAW values:

<input checked="" type="checkbox"/> Fail if Parameter	<input type="text" value="2"/>	<input checked="" type="radio"/> <=	Or	<input type="radio"/> >=	<input type="text" value="150"/>	(decimal)
<input checked="" type="checkbox"/> Fail if Parameter	<input type="text" value="100"/>	<input type="radio"/> <=	Or	<input checked="" type="radio"/> >=	<input type="text" value="100"/>	(decimal)
<input type="checkbox"/> Fail if Parameter	<input type="text"/>	<input type="radio"/> <=	Or	<input type="radio"/> >=	<input type="text"/>	(decimal)
<input type="checkbox"/> Fail if Parameter	<input type="text"/>	<input type="radio"/> <=	Or	<input type="radio"/> >=	<input type="text"/>	(decimal)
<input type="checkbox"/> Fail if Parameter	<input type="text"/>	<input type="radio"/> <=	Or	<input type="radio"/> >=	<input type="text"/>	(decimal)

OK

3. Screening using both methods

Select both screening methods for further versatility

DMM ATA/SATA SMART Functions

Drive is Connected to a:

Native SATA Port SAS Adapter which supports SAT

Fail On Thresholds: Fail drive if ANY parameter is within Counts of its threshold

Fail On RAW values:

<input checked="" type="checkbox"/> Fail if Parameter	<input type="text" value="2"/>	<input checked="" type="radio"/> <=	Or	<input type="radio"/> >=	<input type="text" value="150"/>	(decimal)
<input checked="" type="checkbox"/> Fail if Parameter	<input type="text" value="100"/>	<input type="radio"/> <=	Or	<input checked="" type="radio"/> >=	<input type="text" value="100"/>	(decimal)
<input type="checkbox"/> Fail if Parameter	<input type="text"/>	<input type="radio"/> <=	Or	<input type="radio"/> >=	<input type="text"/>	(decimal)
<input type="checkbox"/> Fail if Parameter	<input type="text"/>	<input type="radio"/> <=	Or	<input type="radio"/> >=	<input type="text"/>	(decimal)
<input type="checkbox"/> Fail if Parameter	<input type="text"/>	<input type="radio"/> <=	Or	<input type="radio"/> >=	<input type="text"/>	(decimal)

OK

- a Drive with SMART Errors:

```
=====
>> SCSI Toolbox32, Version 8. 0. 1 (build) 9012, running on \\STB-DH  <<
=====
```

Test Date: 02/02/2009 11:19:50

Test Pass: 1

Device: 7:12:0 ATA ST31000340NS Serial: 9QJ280T5

Results:

02/02/2009 11:19:50 TEST 1 of 1:
SMART Data Test

SMART data follows:

```
0A 00 01 0F 00 4E 44 4E CD 9F 03 00 00 00 03 03
00 62 5B 00 00 00 00 00 00 04 32 00 64 64 31
01 00 00 00 00 00 05 33 00 64 64 00 00 00 00
00 00 07 0F 00 48 3C 77 A5 E1 00 00 00 09 32
00 64 64 B1 02 00 00 00 00 0A 13 00 64 64 02
00 00 00 00 00 00 0C 32 00 64 25 C8 00 00 00
00 00 B8 32 00 64 64 00 00 00 00 00 00 BB 32
00 64 64 00 00 00 00 00 00 BC 32 00 64 4B 70
00 68 00 68 00 00 BD 3A 00 64 64 00 00 00 00
00 00 BE 22 00 49 38 1B 00 19 1B 00 00 00 C2 22
00 1B 2C 1B 00 00 00 14 00 00 C3 1A 00 27 24 4E
CD 9F 03 00 00 00 C5 12 00 64 64 00 00 00 00
00 00 C6 10 00 64 64 00 00 00 00 00 00 C7 3E
00 C8 C8 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 82 00 82 02 00 7B
03 00 01 00 01 E6 02 00 00 00 00 00 00 00 00
00 00 05 00 05 00 00 00 01 01 09 01 01 01 01
01 00 00 00 00 00 00 00 01 00 00 00 00 00
00 00 00 00 00 00 00 1A 04 88 35 42 02 00 00
00 00 00 00 01 00 36 05 58 3D B4 9B 08 00 00 00
74 38 13 96 02 00 00 00 00 00 00 13 F0 F6 01
00 00 00 00 00 00 00 00 40 00 00 00 00 00 00
0E 00 00 00 00 00 00 00 00 00 00 00 00 00 00
02 00 00 00 00 00 00 00 00 00 00 00 00 00 AB
```


DMM Bugs Fixed:

- Replace SEEK commands with 1-block READ for ATA drives
- Fixed problem with I/O counter overflow after 4.2 billion I/O's

Developer Toolbox API (DTB)

- **Additional Features:**
- Function added to change default driver in registry
- New API in VCPSSL to define where to store the log files

DTB Bugs Fixed:

- Threaded test will use all the characters in the serial number when creating log files for devices under test

BAM

New Features:

- Fixed device discovery so that BAM addressing coincides with STB and system addressing

STB Suite Version 7.2 New Features

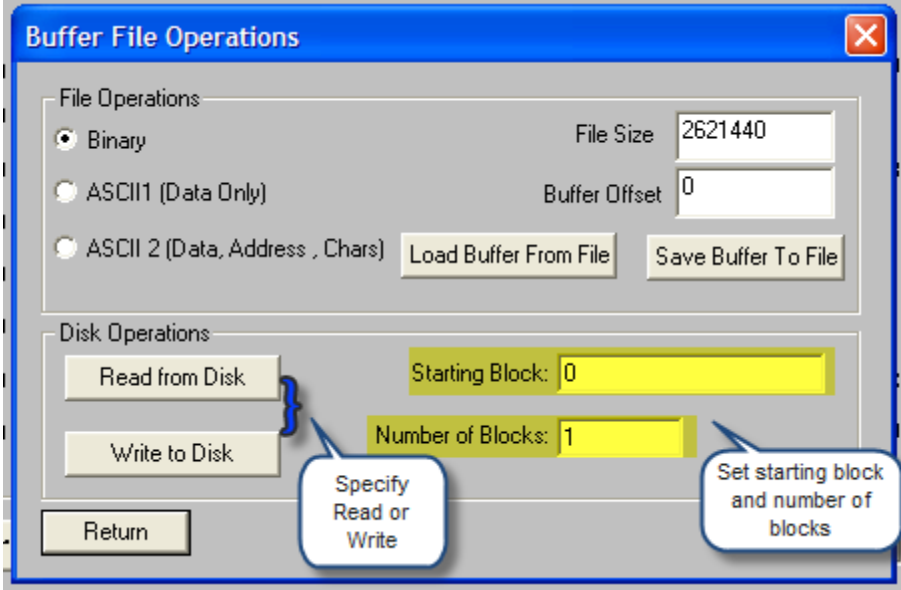
STB

New Features:

Disk Write/Read added to Buffer Functions

You can now read from the currently selected disk drive into the buffer(s), or write from the buffer(s) to the currently selected disk drive.

In the Buffer function click the **File Functions** button, then specify the starting block and number of blocks to either read or write, then click either **Read** or **Write**

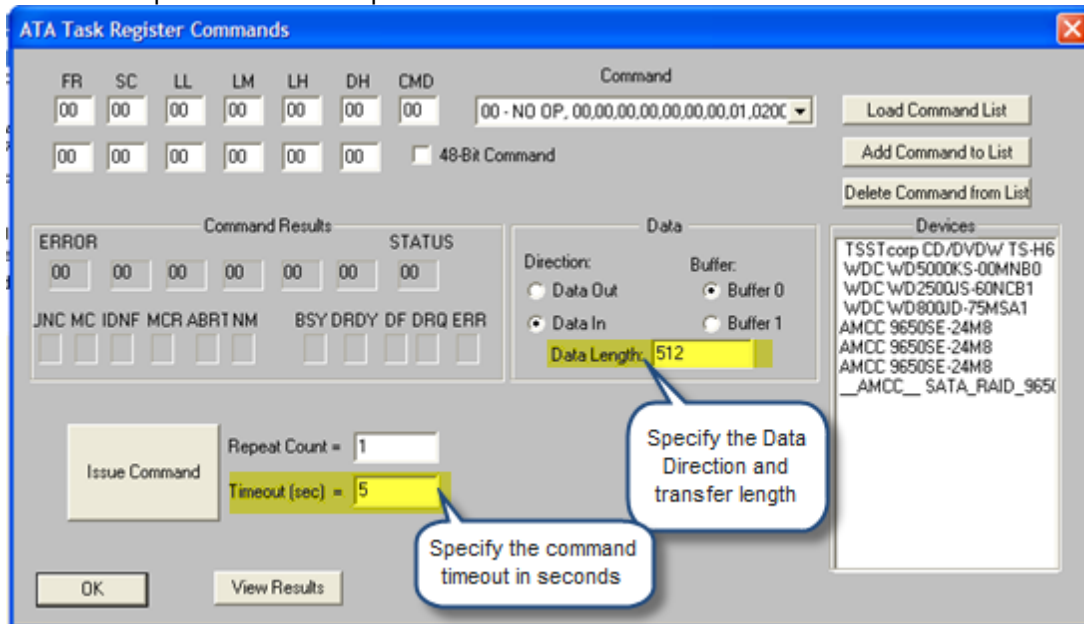


ATA/SATA User Defined Commands –

Added **Command Timeout** and **Data Transfer Length**

Specify the Data Direction, the Data Transfer Length according to what ATA command you are sending. **It is extremely important to set these values correctly**, otherwise the Windows ATA driver may hang and require a power cycle. If a command does not have a data phase you *must* set the transfer length to zero (0).

Specify the command timeout in seconds – be sure to give the command enough time to complete before timing out. For example, a SLEEP command may take 15 seconds or more for the drive to spin down and complete the command.



STB Bugs Fixed:

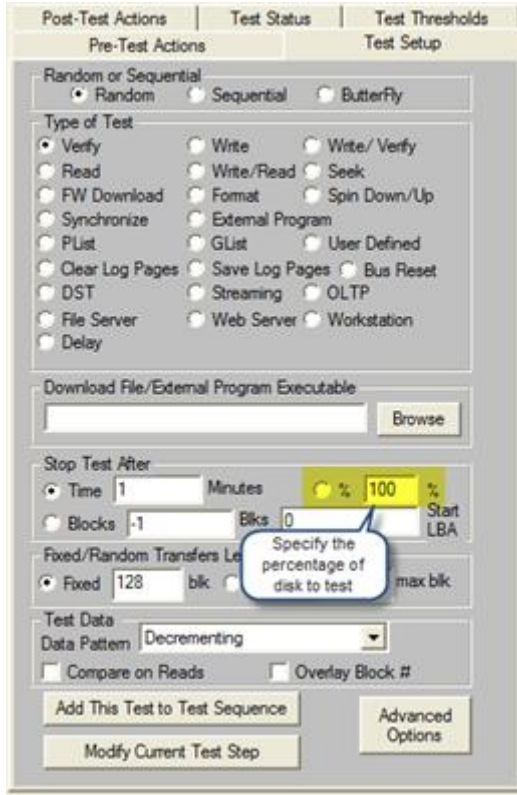
1. ATA User Defined Command infinite hang fixed
2. Short DST Test fixed
3. Mode Sense w/o BLOCK DESCRIPTOR fixed

DMM

New Features:

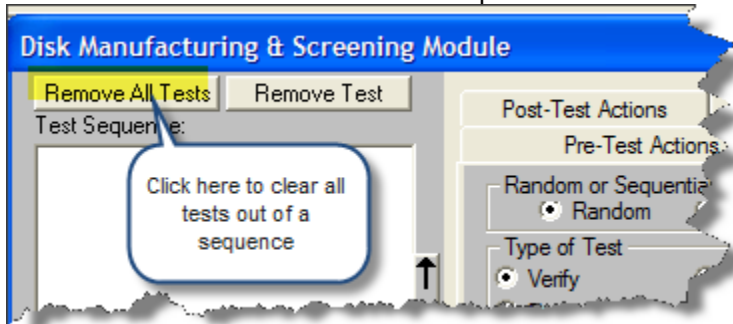
Test by Percentage (%) of disk-

Use this feature to specify a test duration to cover a percentage of the disk – for example if you specify 15%, for each drive tested DMM will calculate 15% of the drive's capacity and will test that amount.



Remove All Tests

Click this to clear all tests from a test sequence –

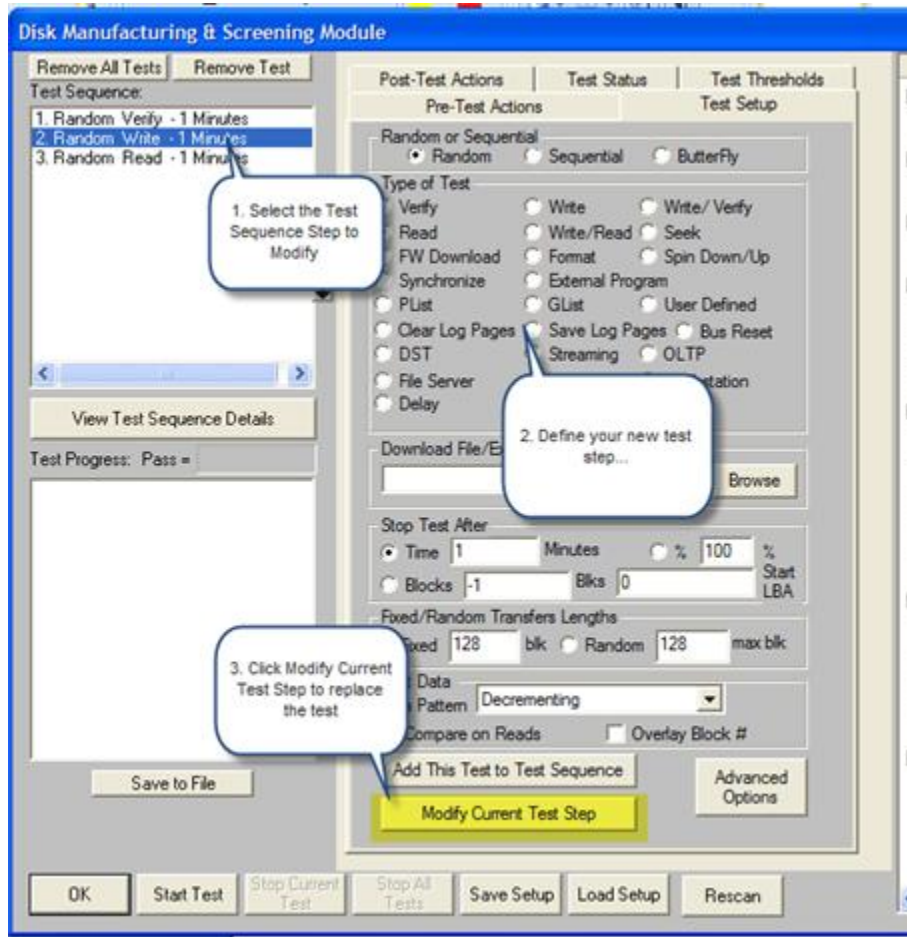


Modify Test Step

Follow these three steps to replace an existing test step with another:

1. In the Test Sequence box select the test step to replace

2. In the Test Setup Tab define your new test step
3. Click the **Modify Current Test Step** button to replace the old test with the new one



New Test Steps

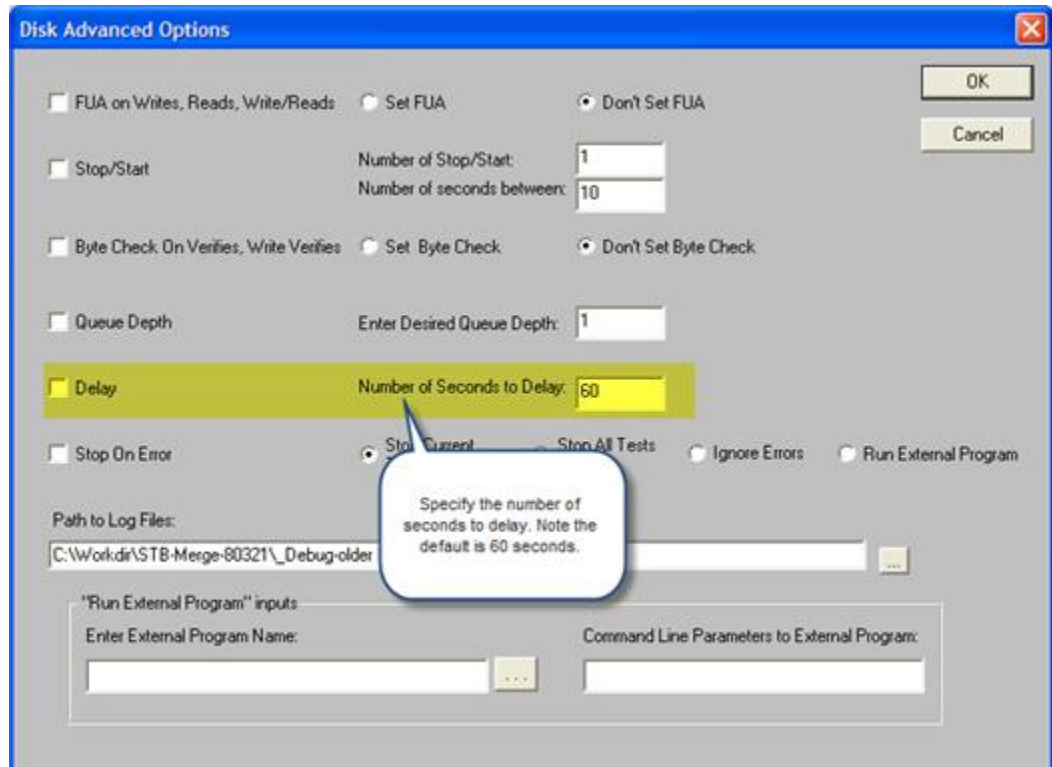
There are nine (9) new test steps available! Details of each new test are below:

The screenshot shows a software interface for configuring test steps. It is divided into several sections: 'Post-Test Actions', 'Test Status', and 'Test Thresholds' at the top. Below these are 'Pre-Test Actions' and 'Test Setup'. The 'Test Setup' section includes a 'Random or Sequential' radio button group with 'Random' selected. Under 'Type of Test', there are multiple radio buttons for different test types, with 'Clear Log Pages', 'Save Log Pages', 'DST', 'Streaming', 'OLTP', 'File Server', 'Web Server', and 'Workstation' highlighted in yellow. Other options include 'Verify', 'Write', 'Write/Verify', 'Read', 'Write/Read', 'Seek', 'FW Download', 'Format', 'Spin Down/Up', 'Synchronize', 'External Program', 'PList', 'GList', 'User Defined', and 'Bus Reset'. Below this is a 'Download File/External Program Executable' field with a 'Browse' button. The 'Stop Test After' section has radio buttons for 'Time' (set to 1 minute) and 'Blocks' (set to -1 blocks). The 'Fixed/Random Transfers Lengths' section has radio buttons for 'Fixed' (set to 128 blk) and 'Random' (set to 128 max blk). The 'Test Data' section has a 'Data Pattern' dropdown set to 'Decrementing' and checkboxes for 'Compare on Reads' and 'Overlay Block #'. At the bottom, there are buttons for 'Add This Test to Test Sequence', 'Advanced Options', and 'Modify Current Test Step'.

1. Clear Log Pages
 - a. This test step will clear all drive LOG PAGES by issuing a LOG SELECT command. Note that this test is only valid on SCSI/SAS/FC drives – it will not work with ATA/SATA/USB disks.
2. Save Log Pages
 - a. This test step will write all LOG PAGEs to each drives log file. Note that this test is only valid on SCSI/SAS/FC drives – it will not work with ATA/SATA/USB disks.

3. Delay

- a. Use this test step to insert a delay of a specified number of seconds in a test sequence. The number of seconds to delay is specified on the Advanced Options page as show below. If you do not specify a delay length it will default to 60 seconds:



4. DST Test

- a. This test step will issue a long Drive Self Test (DST) using the SCSI SEND DIAGNOSTIC command. Note that this test is only valid on SCSI/SAS/FC drives – it will not work with ATA/SATA/USB disks.

5. Streaming Test

- a. This test simulates the functions of a video or audio streaming computer application. It consists of :
 - i. transfer size = large random between 64 and 128 blocks
 - ii. access method = sequential
 - iii. each transfer step = write or read

6. OLTP Test

- a. This test simulates the functions of an online transaction processing computer application. It consists of:
 - i. transfer size = random between 4 and 32 blocks
 - ii. access method = random
 - iii. each transfer step = (read, write)

7. File Server Test

- a. This test simulates the functions of a typical file server computer application. It consists of:
 - i. transfer size = random between 8 blocks and 128 blocks
 - ii. access method = random
 - iii. each transfer
 - 1 out of 5 = write
 - 4 out of 5 = read

8. Web Server Test

- a. This test simulates the functions of a typical web server computer application. It consists of:
 - i. transfer size = random between 1 blk and 1024 blks
 - ii. access method = random
 - iii. each transfer = read

9. Workstation Test

- a. This test simulates the functions of a typical workstation computer application. It consists of:
 - i. transfer size = random between 1 and 32 blocks
 - ii. access method =
 - 1. - 8 out of 10 = random
 - 2 out of 10 = sequential from previous
 - iii. each transfer =
 - 1. 4 out of 5 = read
 - 2. 1 out of 5 = write

Save Log Page path

Default Log File Path is now saved. To change the path to the DMM log files click on the **Advanced Options** button

The screenshot shows the "Disk Advanced Options" dialog box. The "Path to Log Files" field is highlighted in yellow and contains the text "C:\Work\dr\STB-Merge-80321_Debug-older". A callout bubble points to this field with the text: "Specify the path where you want DMM to create its log files. Or use the browse button to find your path".

Path to Log Files:
C:\Work\dr\STB-Merge-80321_Debug-older

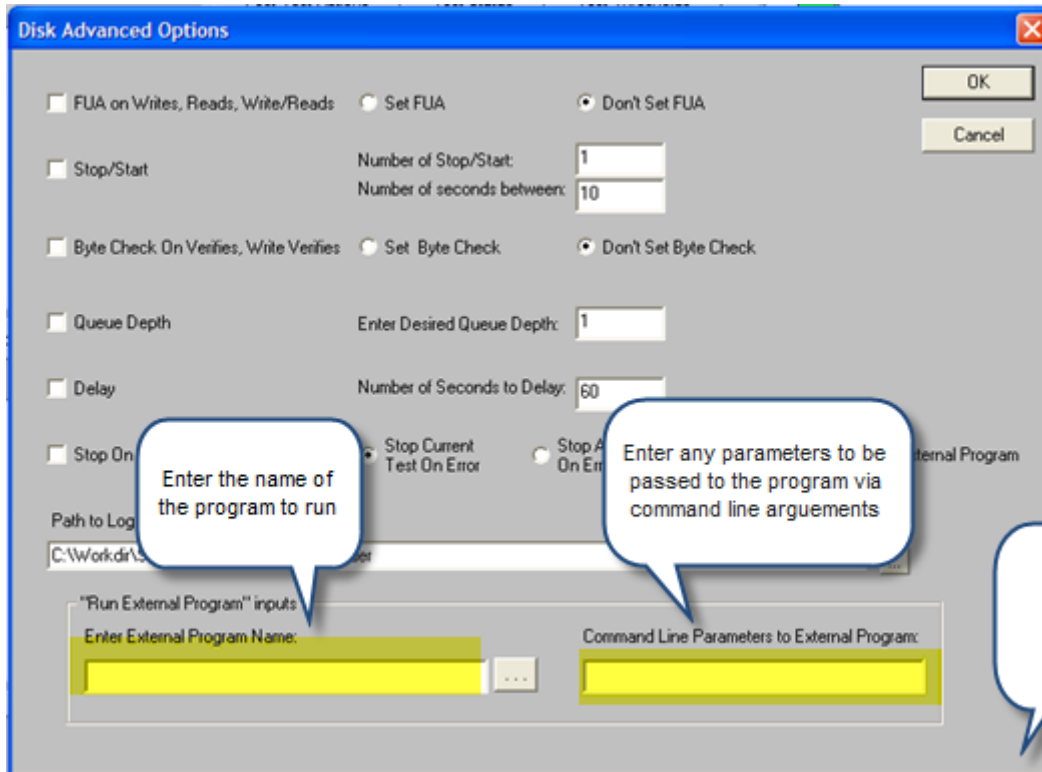
"Run External Program" inputs
Enter External Program Name: Command Line Parameters to External Program:

External program – pass arguments

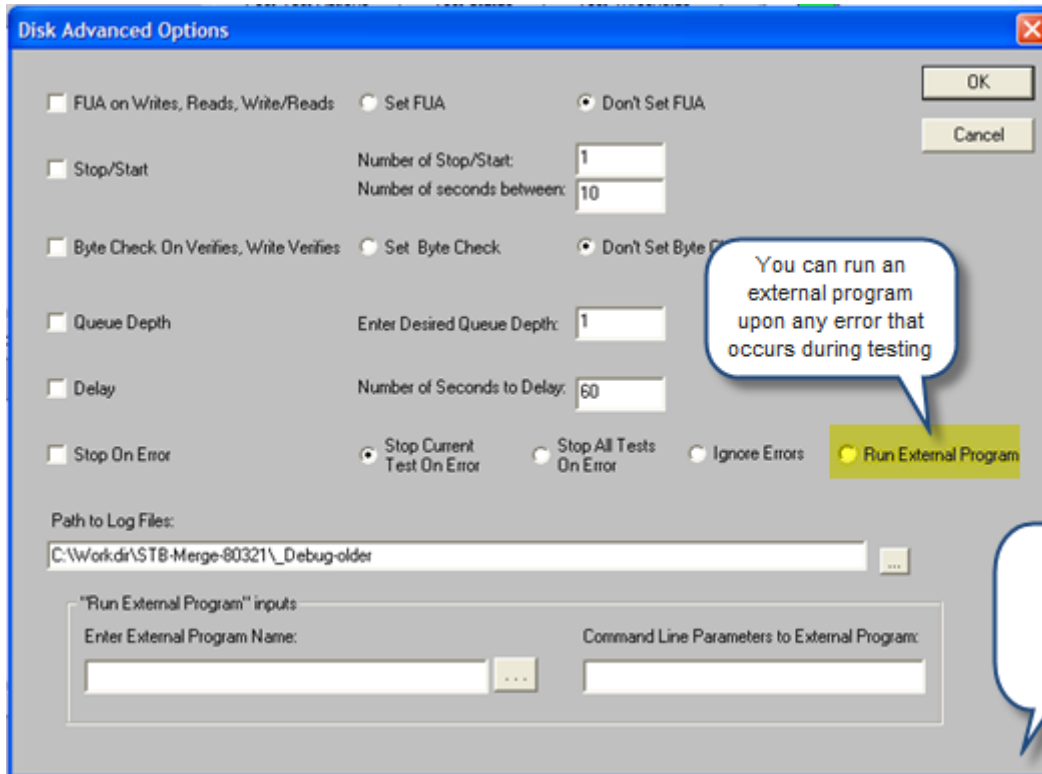
You can pass command-line arguments to any program that you run using either the “External Program” test step or the new “Run External Program” on error – see next item describing this.

By default when an external program is called from a DMM test step it will be passed the following arguments on the command line:

HA=n,Target=n,LUN=n,SLOT=n,optional command line arguments

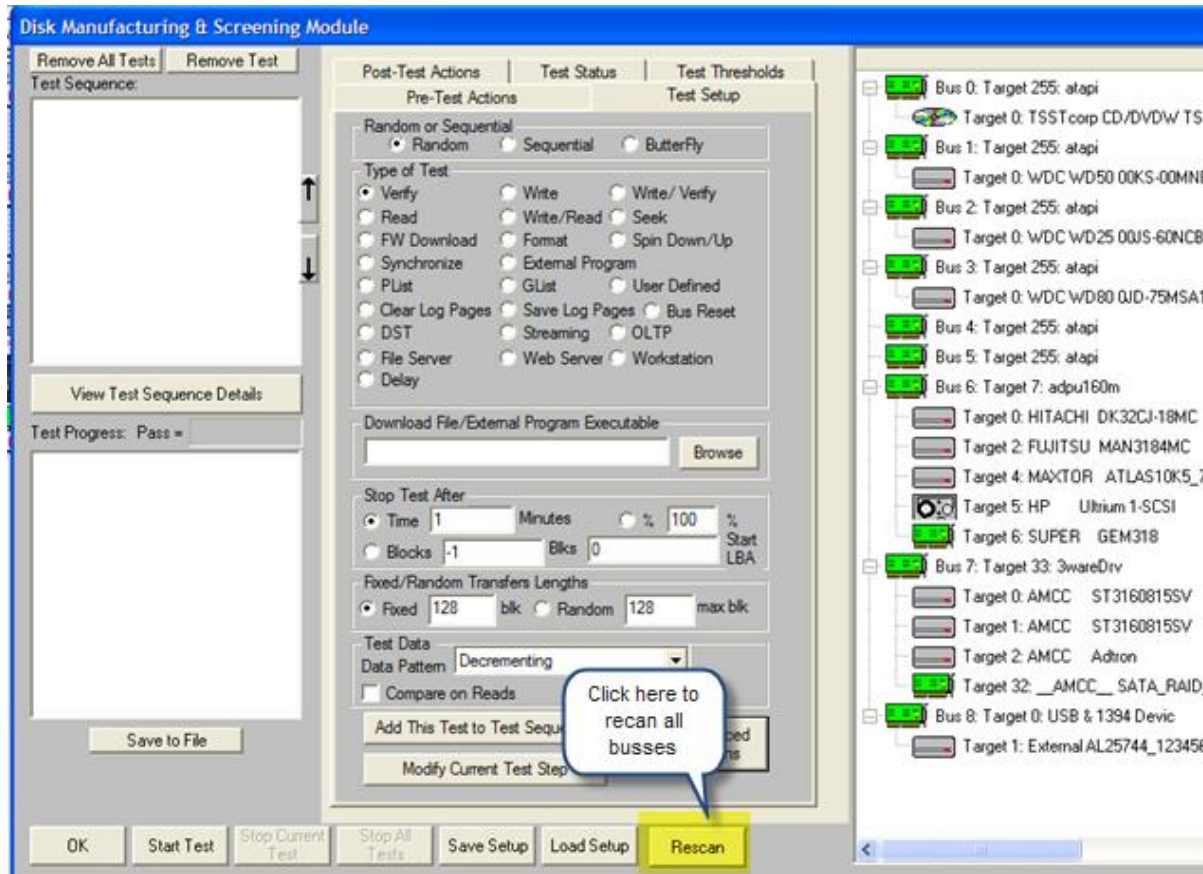


Run an external program upon error



Rescan Busses

You can rescan all busses, adapters on your system to discover new devices that have been attached (hot plugged).



DMM Bugs Fixed:

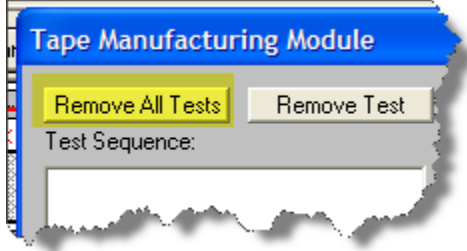
- none

TMM

New Features:

Remove All Tests

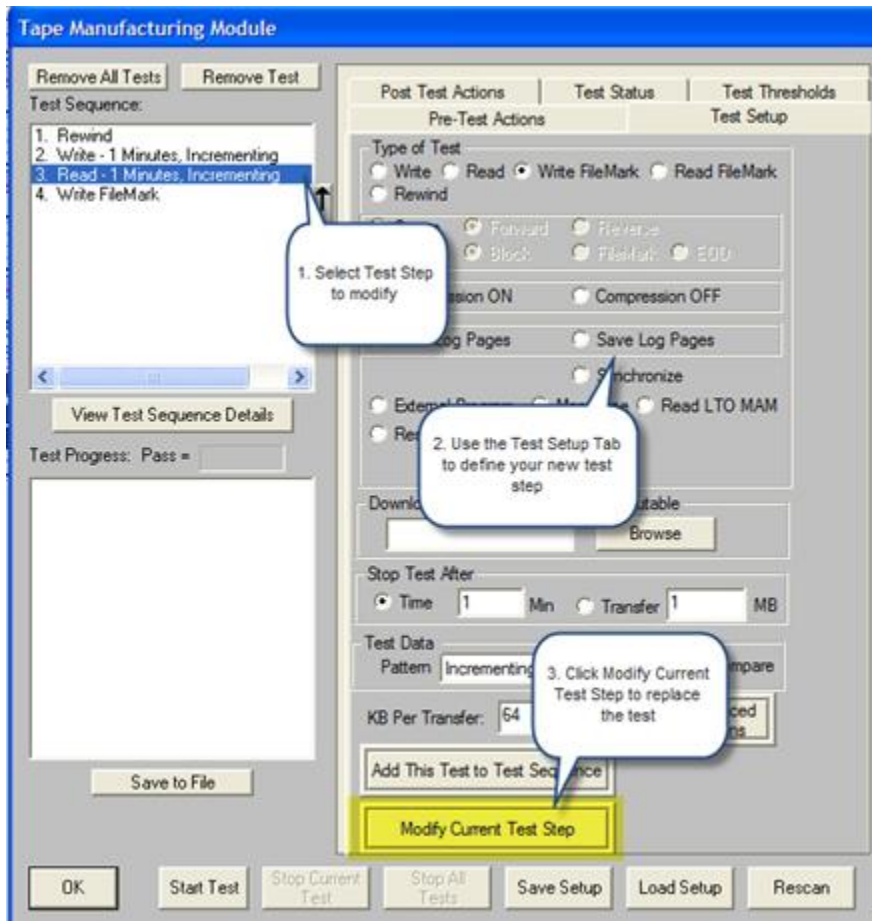
Click this to clear all tests from a test sequence –



Modify Test Step

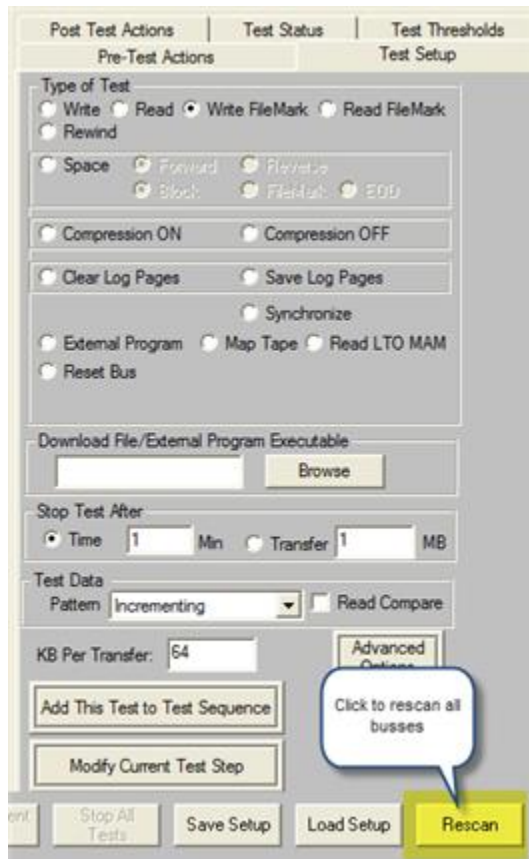
Follow these three steps to replace an existing test step with another:

1. In the Test Sequence box select the test step to replace
2. In the Test Setup Tab define your new test step
3. Click the **Modify Current Test Step** button to replace the old test with the new one



Rescan Busses

You can rescan all busses, adapters on your system to discover new devices that have been attached (hot plugged).



Bugs Fixed:

- Device selection fixed to not require opening all LUNs in order to select LUN 0 if the drive only has LUN 0
- Device display corrected to display device in red if any test fails

BAM

New Features:

More ATA command definitions added to Command Phase display

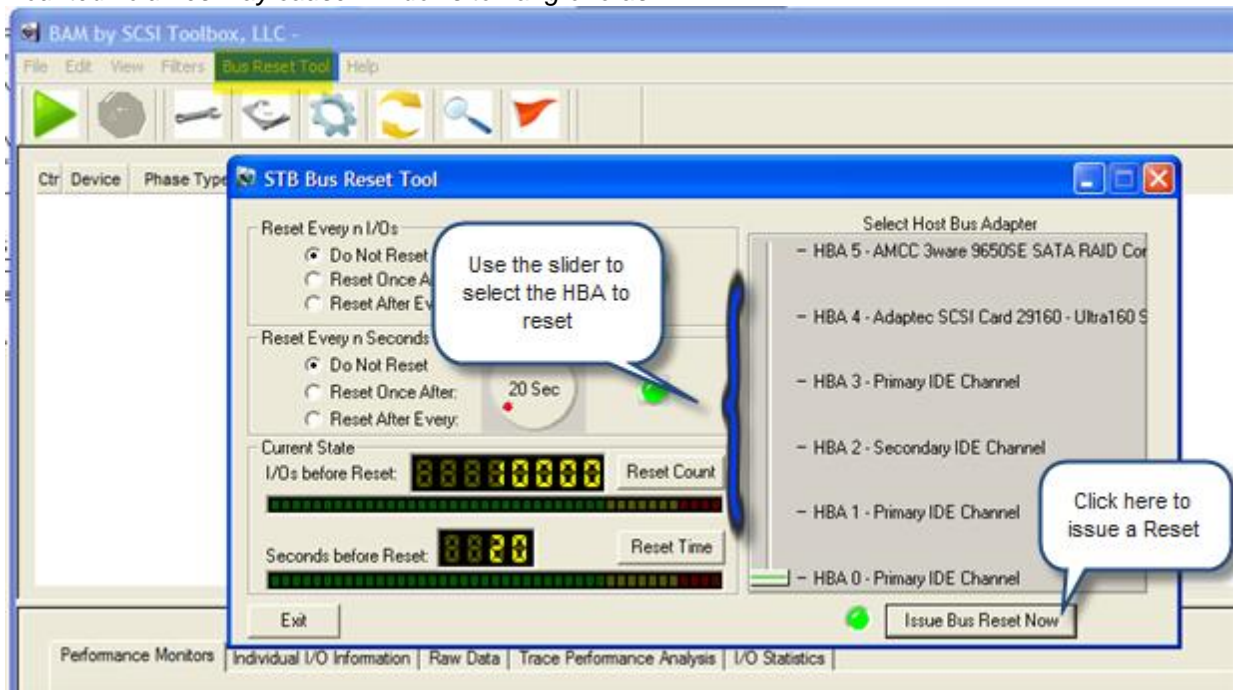
Save CDB and Driver Filters

All CDB and driver filters are now automatically saved and reloaded when you restart BAM.

Device/Bus Reset Control Added

You can now start an instance of the STB Bus Reset Tool from a choice on the BAM top menu. The Bus Reset Tool lets you issue a BUS RESET to any Host Bus Adapter on your system. Use the slider control to select which HBA to reset.

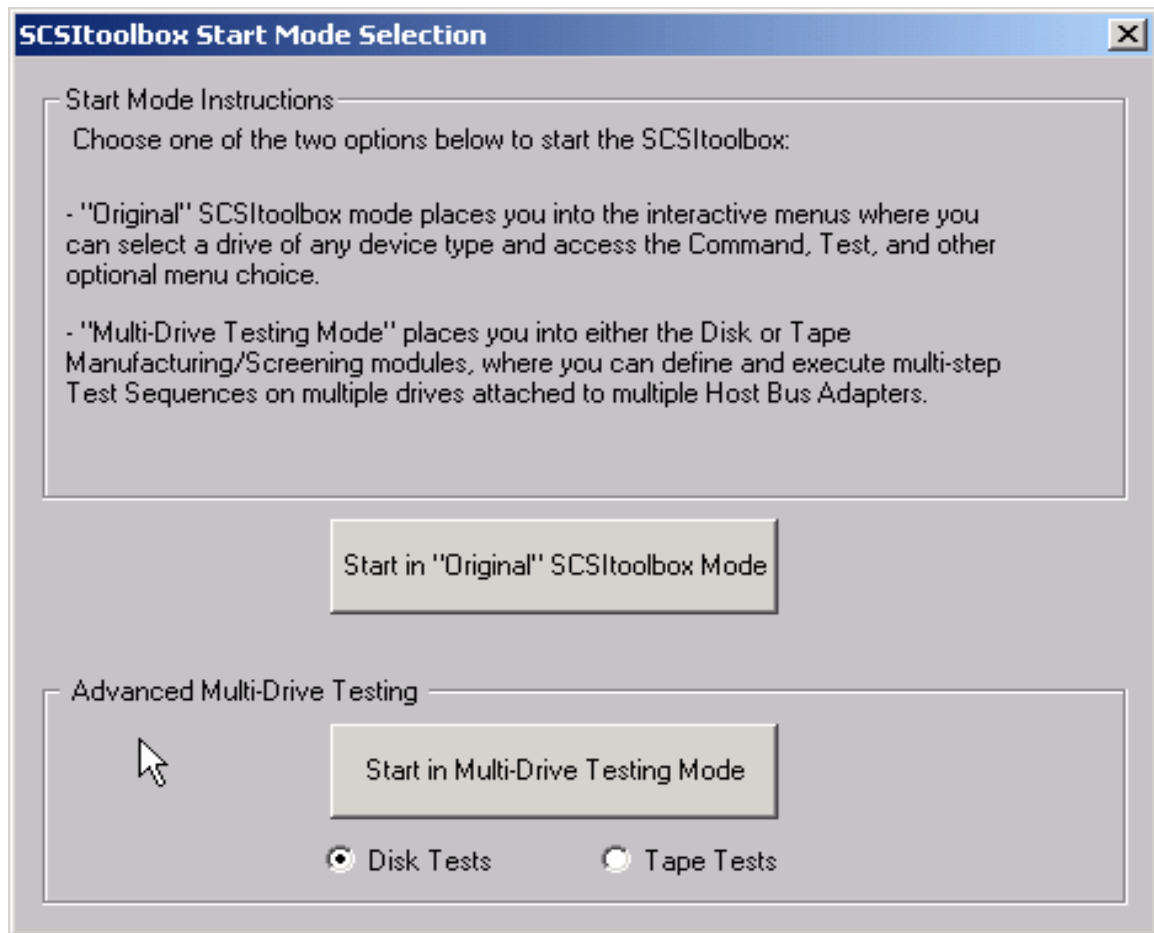
Caution: Issuing a BUS RESET to any HBA or controller that is running Windows system or mounted volumes may cause Windows to hang or crash.



Bugs Fixed:

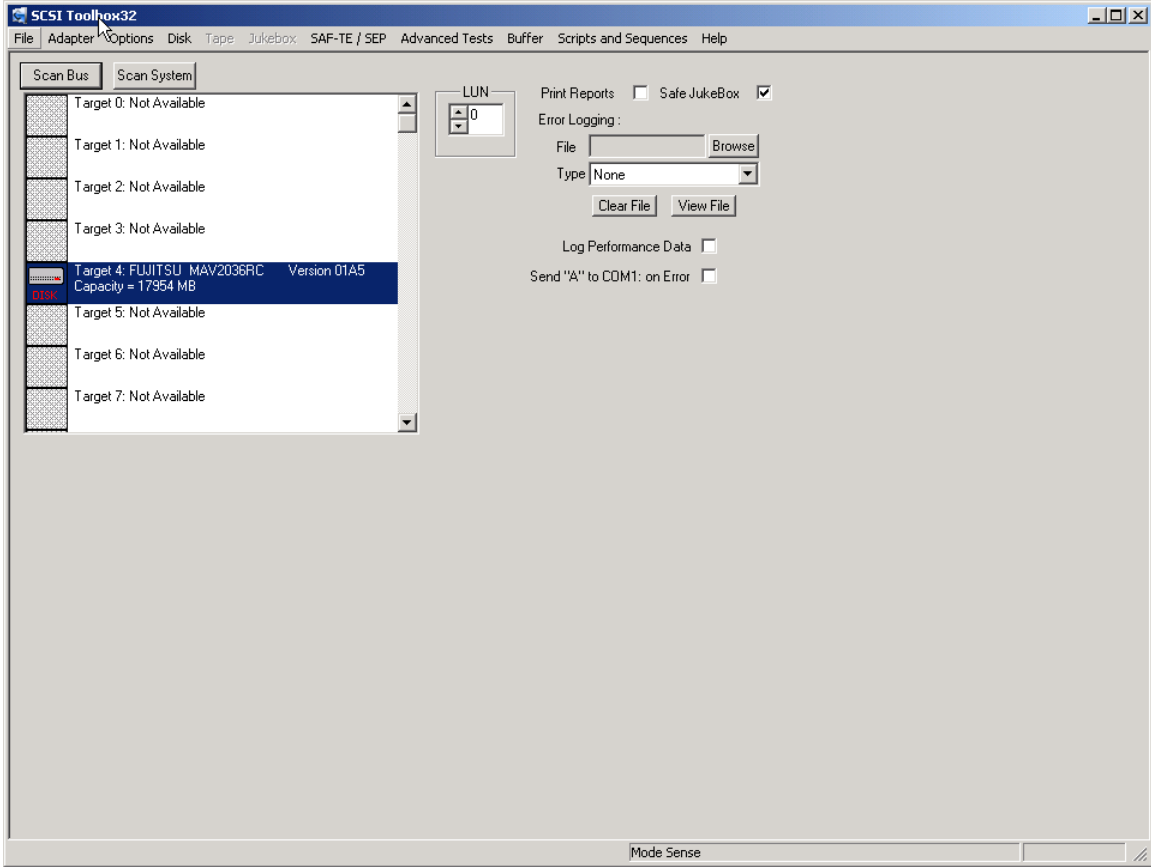
- New device discovery method improves finding all connected devices
- **Save** dialog box type changed from "Save" to "Open" – functionality remains the same
- Random crashing during very long captures fixed

Main Menu

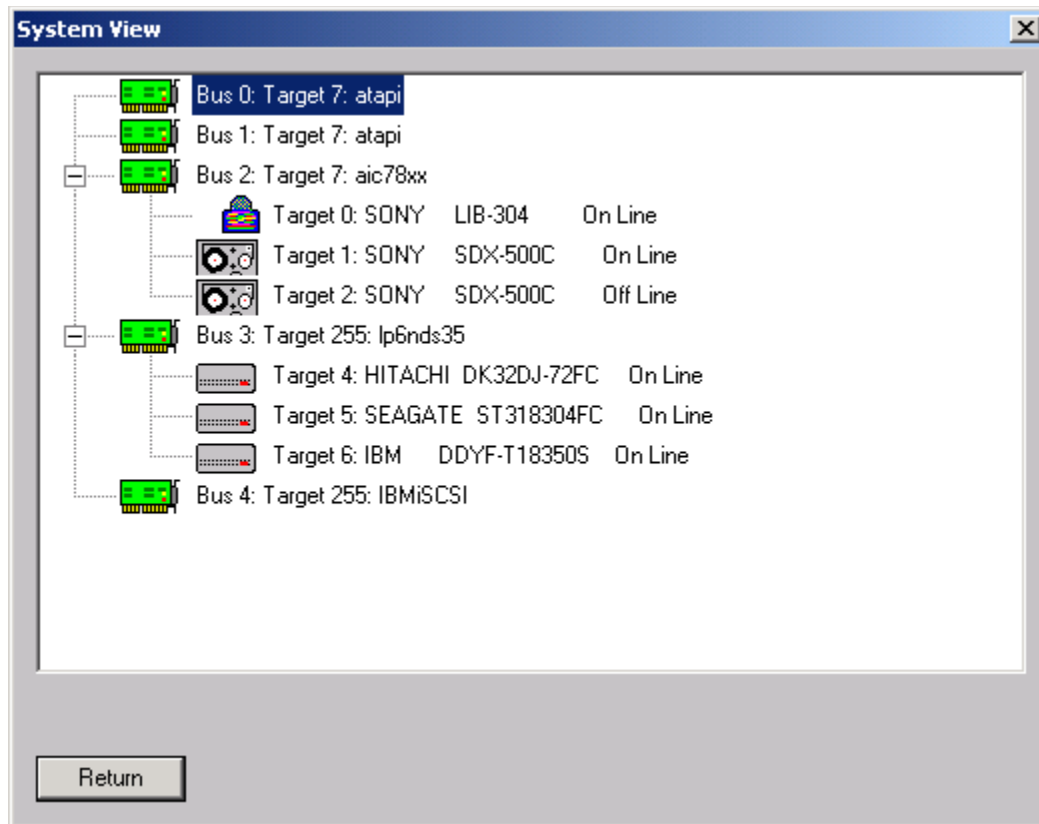


The option "Start in Original SCSIToolbox Mode" will bring up the SCSIToolbox's Main Menu, as seen below.

The option "Start in Multi-Drive Testing Mode" will bring you straight the Disk/Tape Manufacturing Module. To read more about the Manufacturing Modules, see p.181.

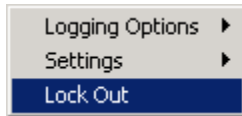


The System View shows all HBA's, all targets, and all LUN's attached to the system:



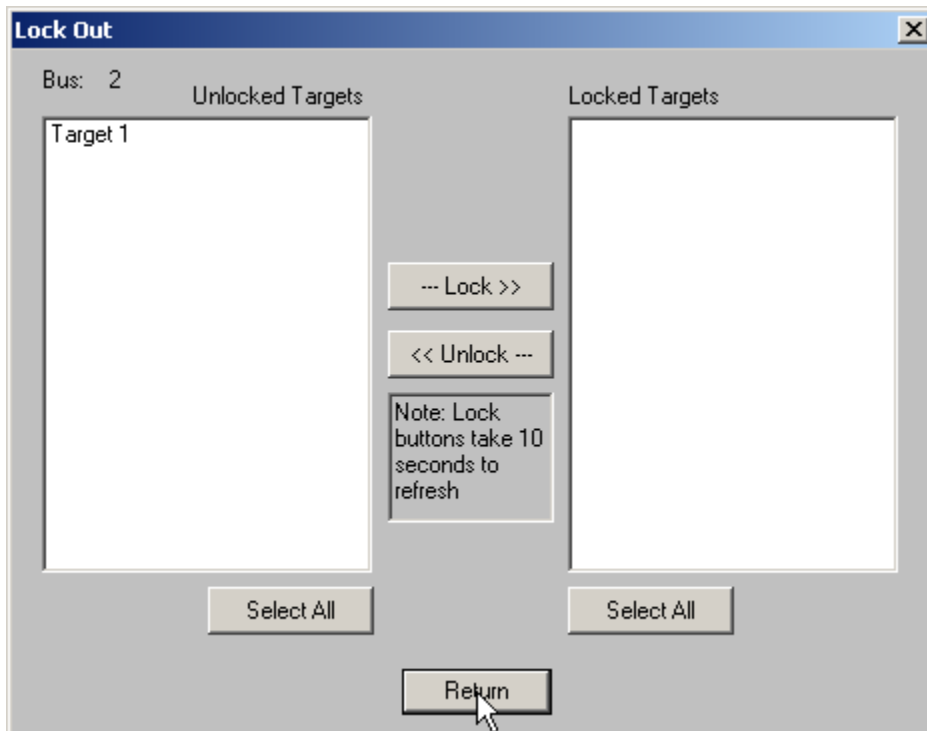
Options

The following options are available from the Options menu choice:











Lock Out (protecting a drive)

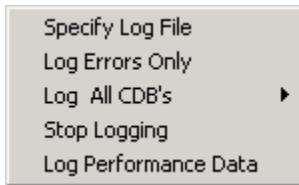
The Drive Locking option allows you to specify which drives can be accessed for testing or commands. Select the drive to lock by clicking on its target number in the left (Unlocked) box. Then click the Lock button to move this drive to the right (Locked) box.



When a drive is locked it will appear like this:

	Target 0: SONY LIB-304 Version 2.4 On Line
	Target 1: Locked
	Target 2: SONY SDX-500C Version 0108 Off Line
	Target 3: Not Available
	Target 4: Not Available
	Target 5: Not Available
	Target 6: Not Available
	Target 7: aic78xx

Logging Options



The logging options allow you to specify what type (if any) error logging is to be performed and to specify a file name for the log file.

The logging options are also available on the right side of the main program display.

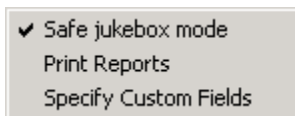
Logging options are:

- *None* – when an error occurs the error information is displayed on the screen and not logged. If the error occurs during a test the test will stop until the error is acknowledged.
- *Errors only* – any time a SCSI command or test elicits a CHECK CONDITION the error information will be logged to the log file. No information will be displayed on the screen and if a test is running it will not stop.
- *All CDB's* – every CDB issued will be logged. CAUTION – this logging option can create very large log files!

The View File button will display the current log file.

The Delete File button will clear the contents of the log file.

Settings



Safe Jukebox Mode

This option will control the operation of the JukeBox Module (if installed.) With **Safe Mode** enabled, all media moves (either by command or in a test process), will return the media to its original source. For example, a move from slot 2 to slot 3 will be followed automatically by a move from slot 3 to slot 2. This prevents the user from moving media which is being monitored by some type of system software. Also, moves to a drive must originate in a mailbox, thereby preventing the user from writing to active media in the jukebox.

If **Safe Mode** is turned off, all moves are one way, and no controls are placed on moves or writing to the media in the drives.

Print Reports

When this option is selected a report will be sent to the windows default printer after any test has been run. The reports are generated from one of two template files, report1.dat or report2.dat. The file "report1.dat" will be processed if the test was completed with no errors, the file "report2.dat" will be processed if any errors occurred during testing.

These files are template files that you create with an editor. You can embed the following tokens into these files and the report generator will fill the values in at print time:

%d	prints time and date
%v	prints SCSI VENDOR data
%p	prints SCSI PRODUCT data
%r	prints SCSI VERSION data
%s	prints SCSI serial number
%a	prints SCSI target address
%c	prints capacity
%n	prints name of last test run
%f	prints a form feed
%e	prints test error count
%k	prints SCSI Sense information
%u1	prints user defined field #1
%u2	prints user defined field #2

The sample report1.dat file is as follows (this report will print when a test passes):

STB Quality Control Report

Today's Date :%d
Manufacturer :%v
Product :%p
Firmware :%r
SCSI Target :%a
Test results for %v %p

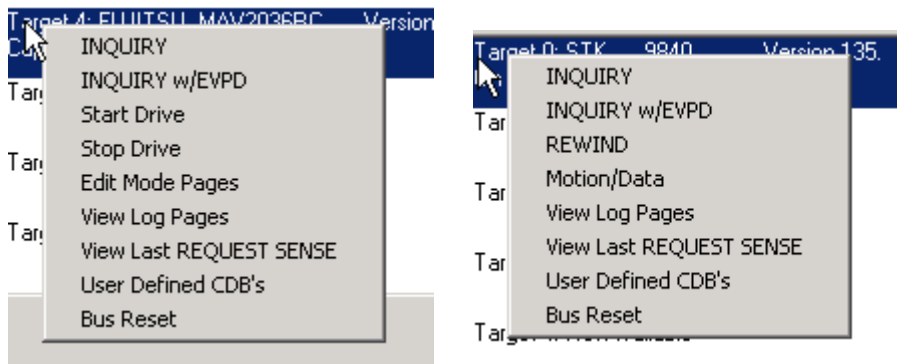
Drive PASSED %n
Tested by (Tech) %d
%f

Specify Custom Fields

This option allows the operator to specify two additional user defined fields for the report generator. A dialog box is displayed in which the user can type any character string to be assigned to user defined field 1 or 2.

Quick Commands

The Quick Commands feature lets you access frequently used commands by right-clicking on the currently selected device. This will bring up the Quick Commands menu appropriate for the selected device type. Here are screenshots of the Quick Command menus for disk and tape device types:

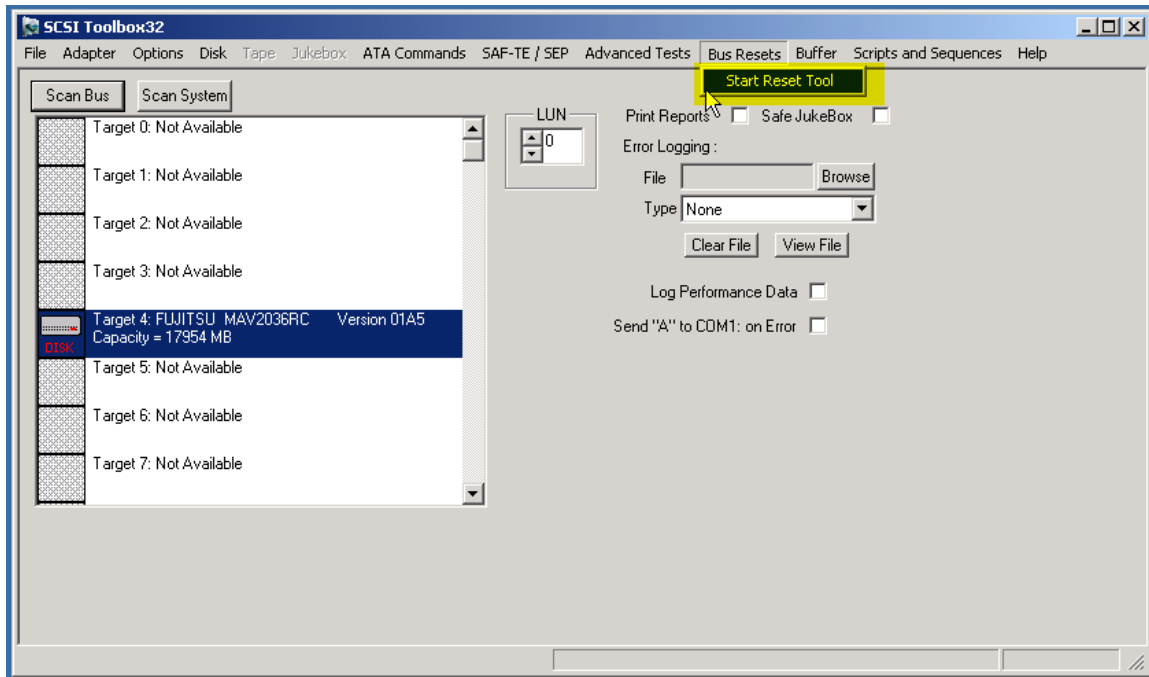


Each of the commands in the Quick Commands menus are described in the documentation Commands section.

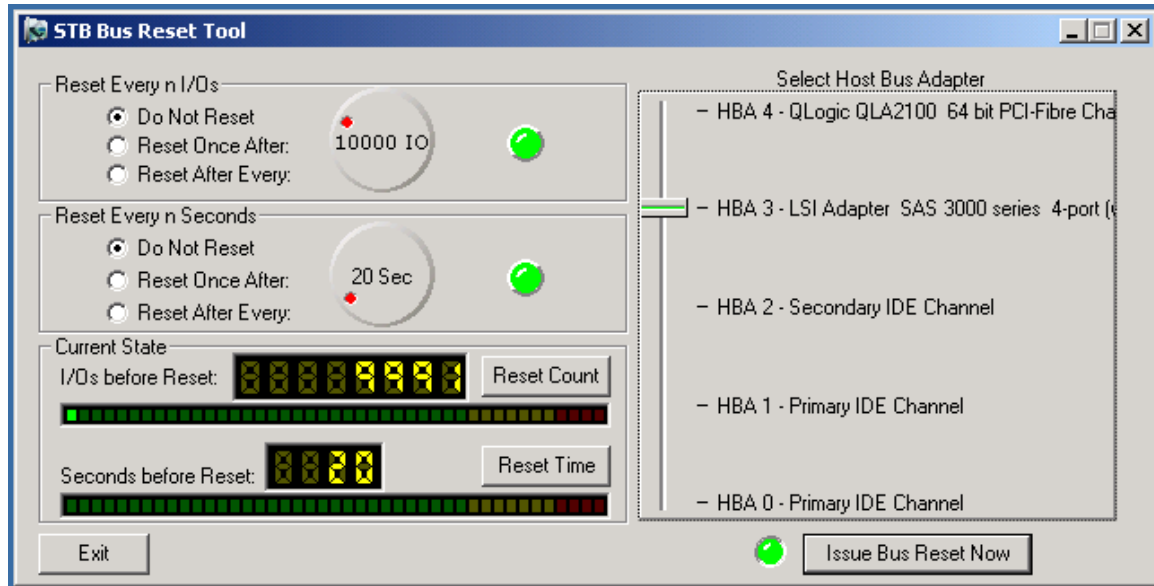
The Bus Reset choices in the Quick Commands menus allow you to issue a hard bus reset to the bus that the device is attached to. You can confirm the Bus Reset by using the Bus Analyzer Module (BAM) being sure to select capture of "Reset" phases.

Bus Reset Control

The Bus Reset control is accessed by the top menu "Bus Reset Choice" as shown here:



The Bus Reset Control:



As shown above, you may select any bus that is displayed in the right side “Select Host Bus Adapter” switch section. Simply drag the vertical switch control to the HBA that you wish to work with. The default switch setting will be the HBA that is currently selected in STB.

Manual Reset

Clicking on the “Issue Bus Reset Now” button immediately causes a hard bus reset on the selected HBA.

Reset by Time

Use the “Reset Every n IOs section of the tool to automatically issue bus resets after n IOs have been generated by STB. Change the number of seconds between resets by rotating the seconds knob control.

Use the radio buttons to the left of the rotary seconds control to select reset behavior.

The default setting “Do Not Reset” does nothing when the timer ends.

The “Reset Once After:” selection will issue one and only one reset when the timer expires.

The “Reset After Every:” selection will issue a reset each time the timer expires.

Reset by I/Os

Use the “Reset Every n IOs section of the tool to automatically issue bus resets after n IOs have occurred. Change the number of IOs between resets by rotating the IO knob control.

Use the radio buttons to the left of the rotary IOs control to select reset behavior.

The default setting "Do Not Reset" does not issue a reset.

The "Reset Once After:" selection will issue one and only one reset after n IOs.

The "Reset After Every:" selection will issue a reset each time n IOs have occurred.

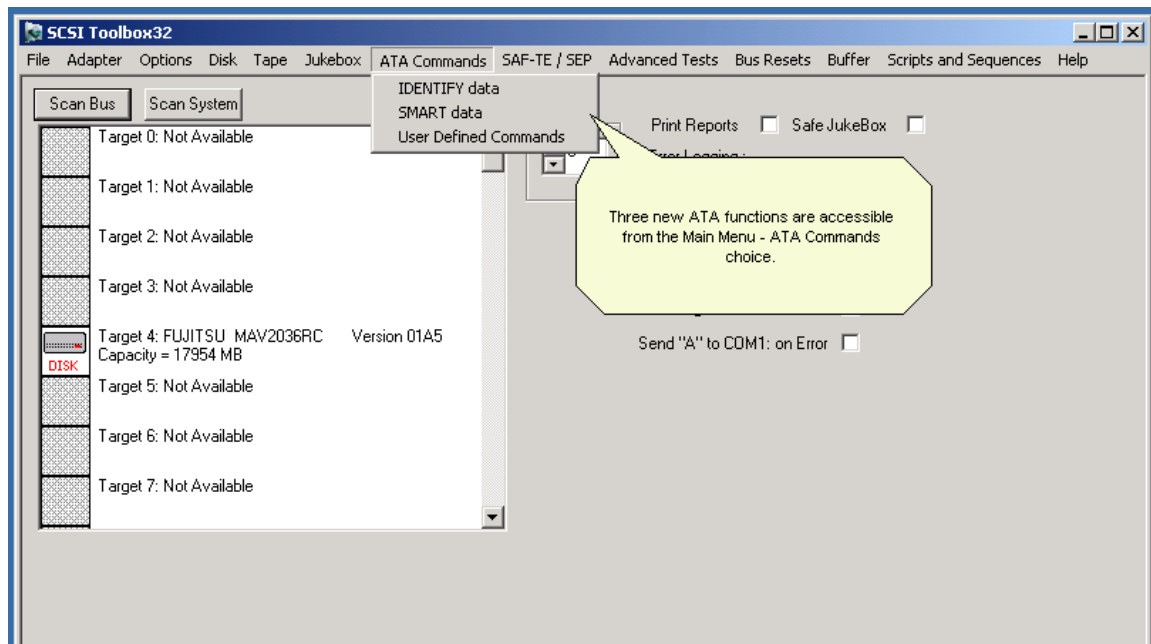
ATA Task Register Commands

Introducing to the SCSItoolbox ATA Task Register Commands

A new feature in the SCSItoolbox Suite Version 6.0 is the ability to issue any task register ATA command to any ATA/SATA storage device. This allows test scenarios that include ATA commands that are not implemented in ATAPI such as retrieving SMART data, ATA IDENTIFY data, SEEK, SET MAX ADDRESS, and more.

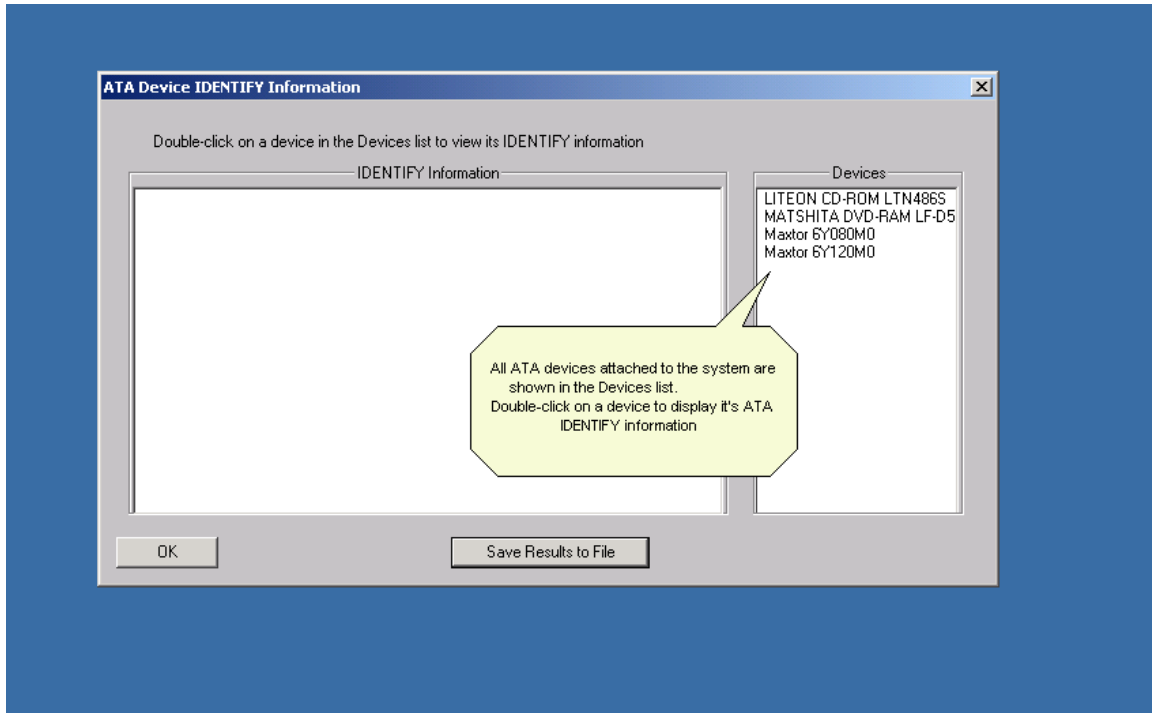
Accessing the ATA Commands

Use the Main menu ATA Commands choice to choose an ATA function

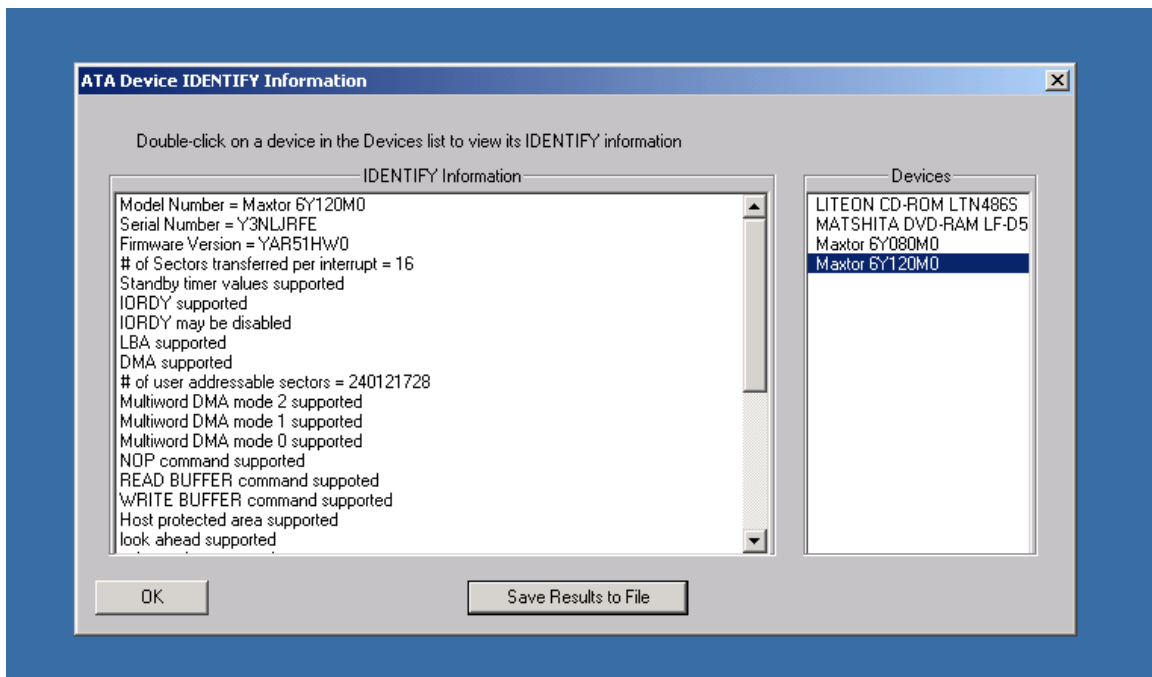


The ATA IDENTIFY command

Choose an ATA device by double-clicking on it in the Devices List

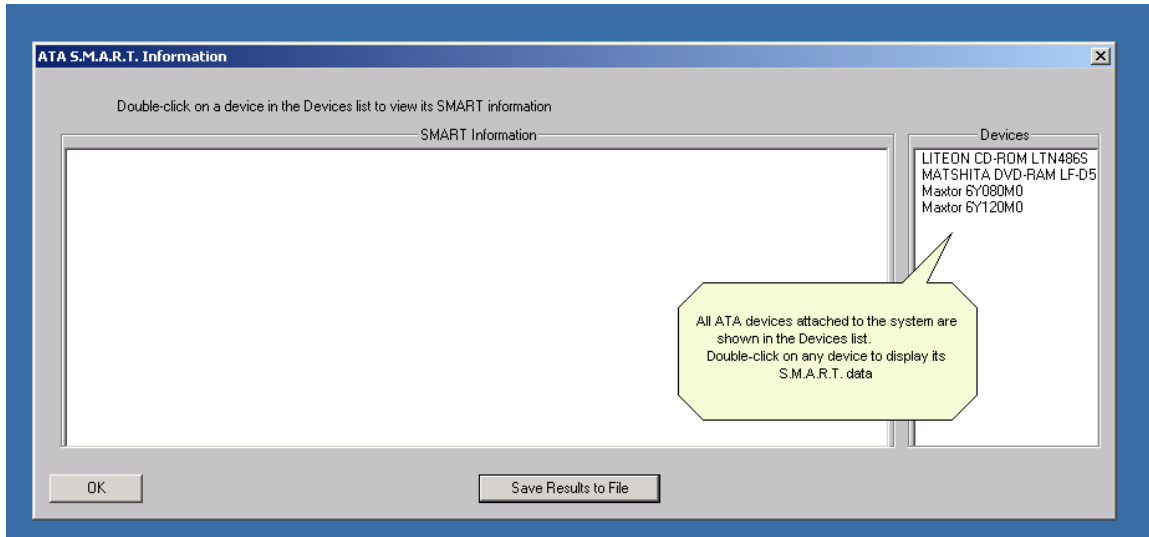


The ATA IDENTIFY information will be displayed / interpreted

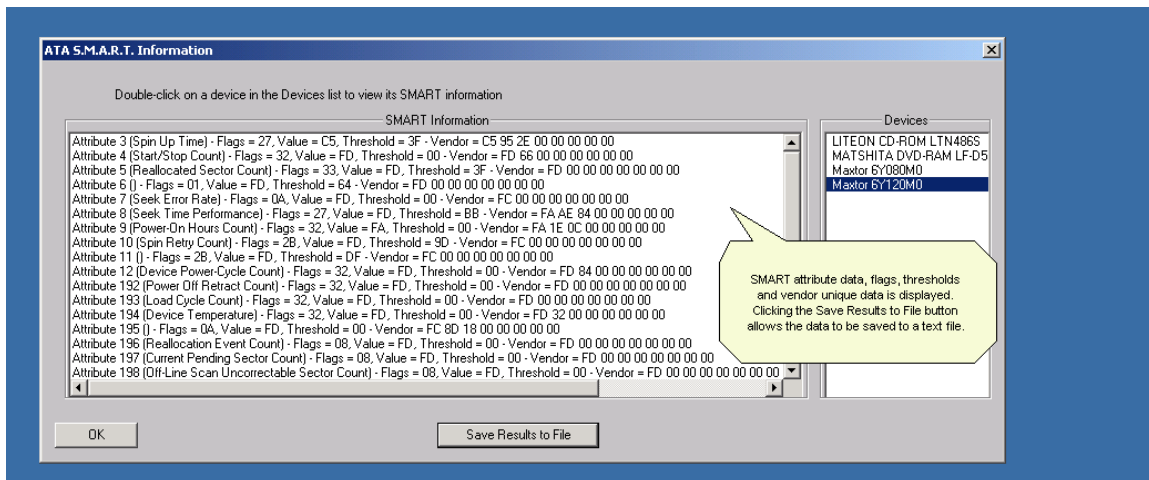


The ATA SMART command

Choose an ATA device by double-clicking on it in the Devices List



SMART attribute data will be displayed / interpreted

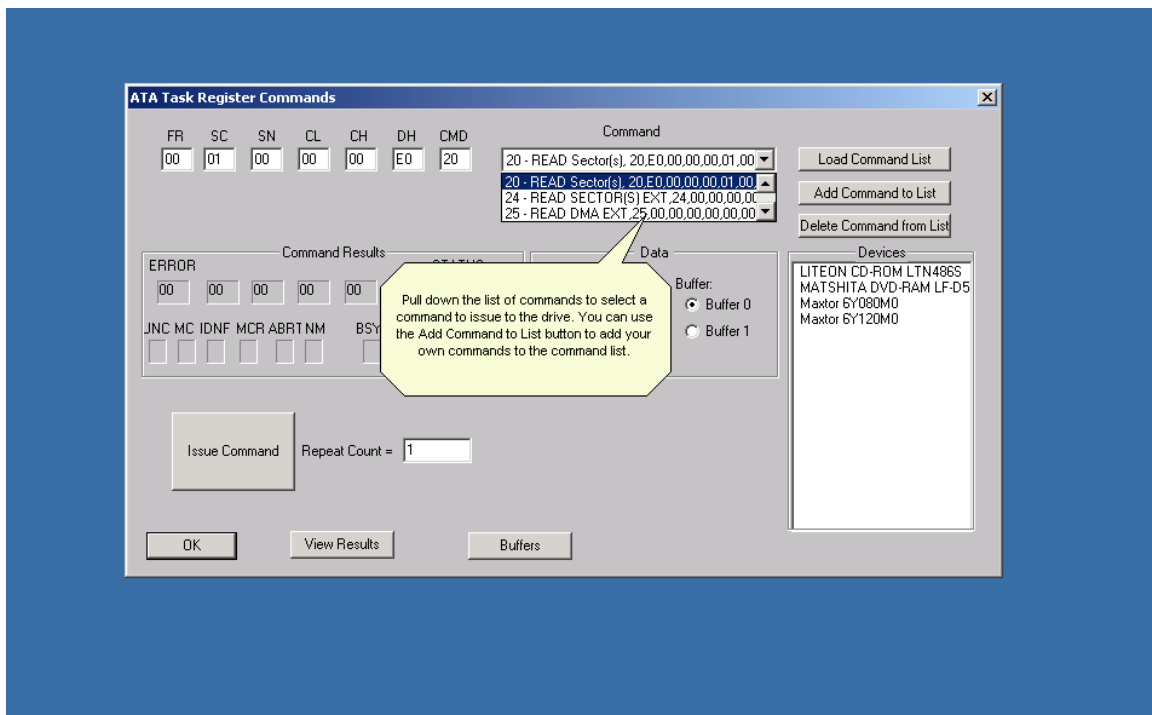


The ATA User-Defined Command menu

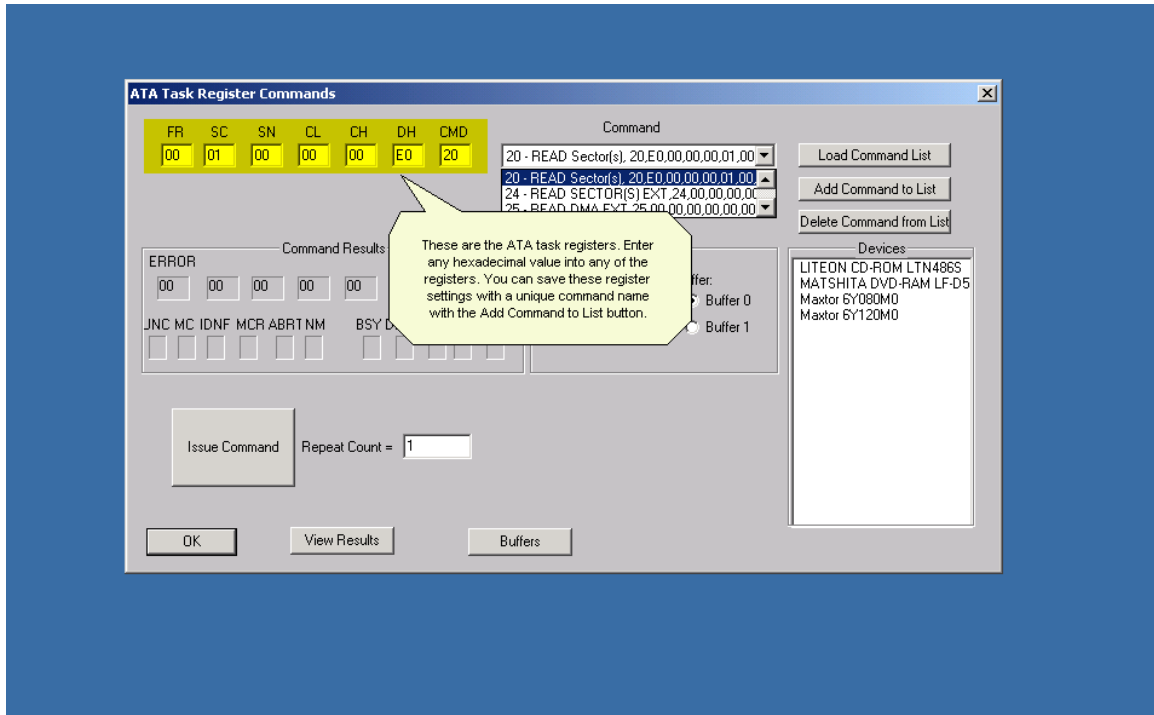
The User Defined Command menu lets you :

- choose commands from a list
- add new or custom commands to the command list
- issue commands to any ATA device
- view all status / error bytes
- view data

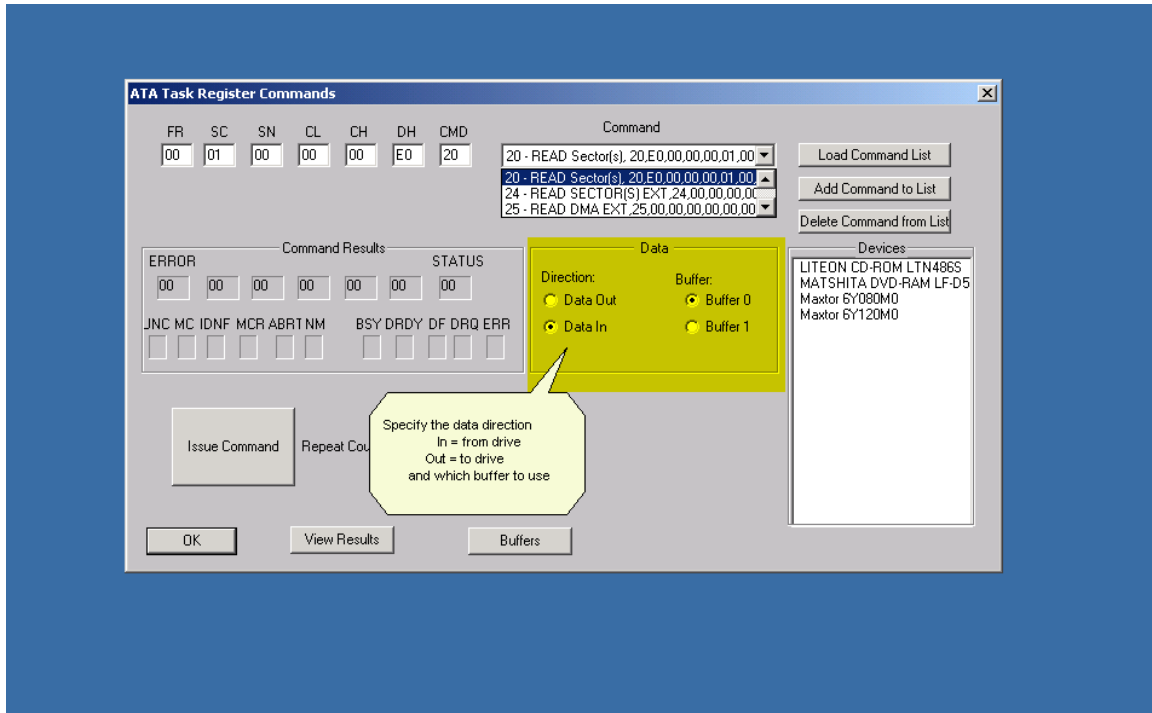
Commands are accessed from the pull-down Command list. When you click on a command in the list the seven ATA task registers are filled with the data that defines that command.



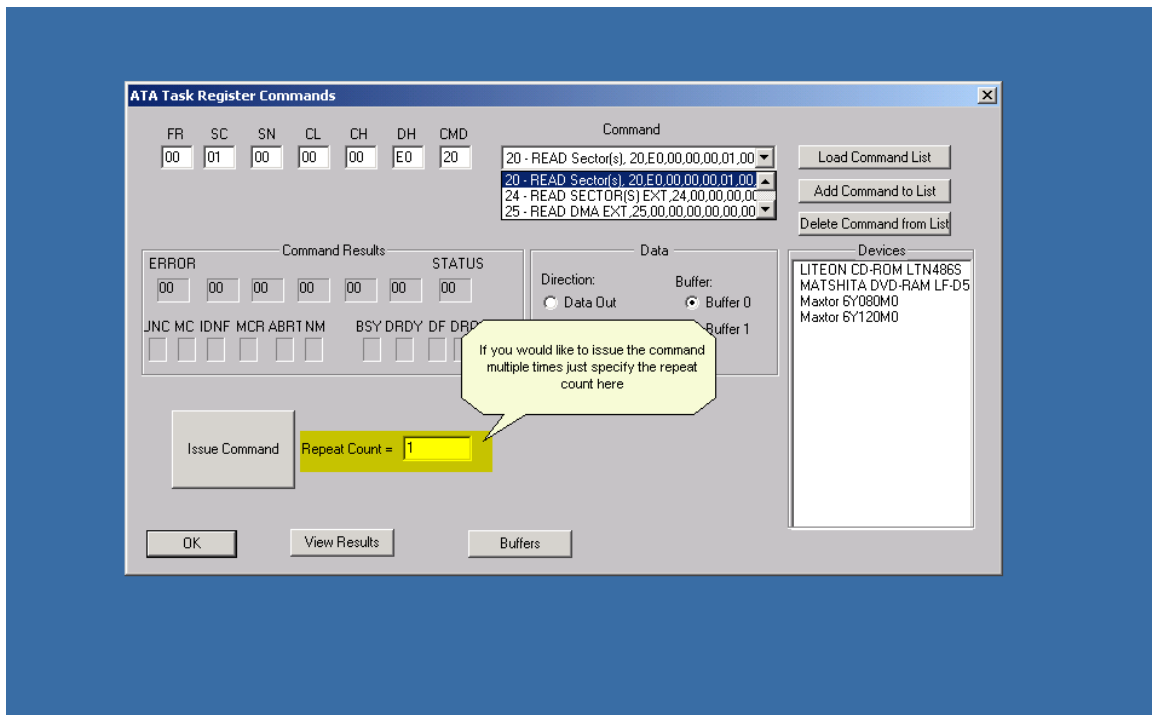
You can also enter any data you wish in the seven ATA task registers, and then add the new command to the Command list with a unique name. This allows any command, standard or vendor unique to be defined and saved for easy access.



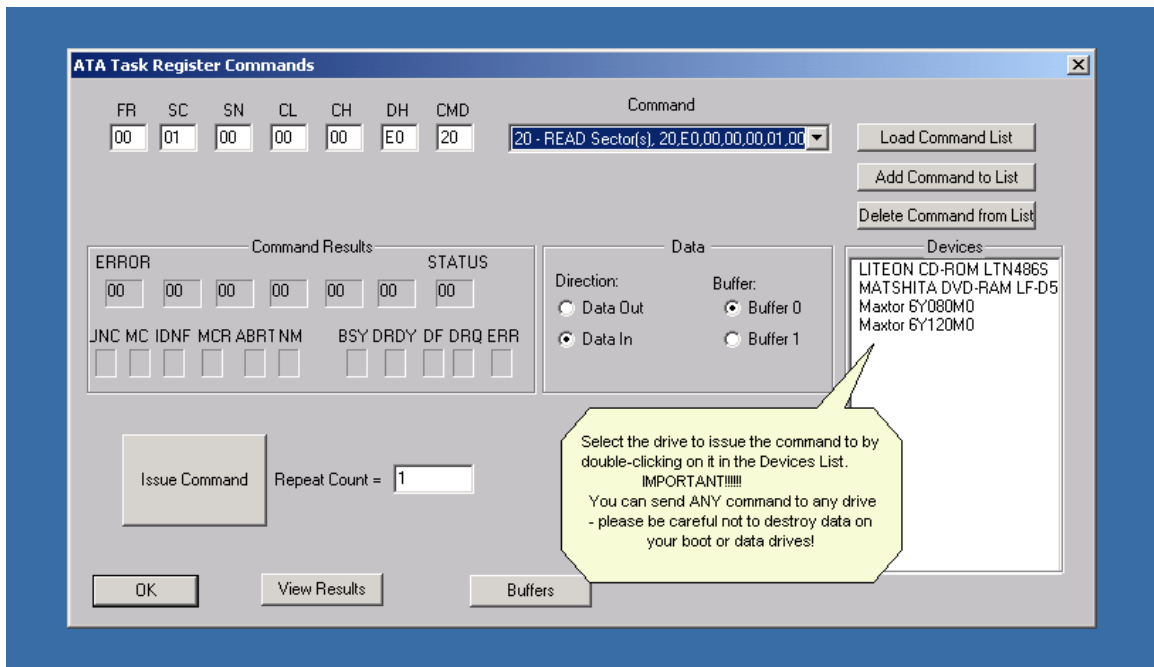
Specify the data direction (**In** from drive or **Out** to drive) and which of the two data buffers you wish to use.



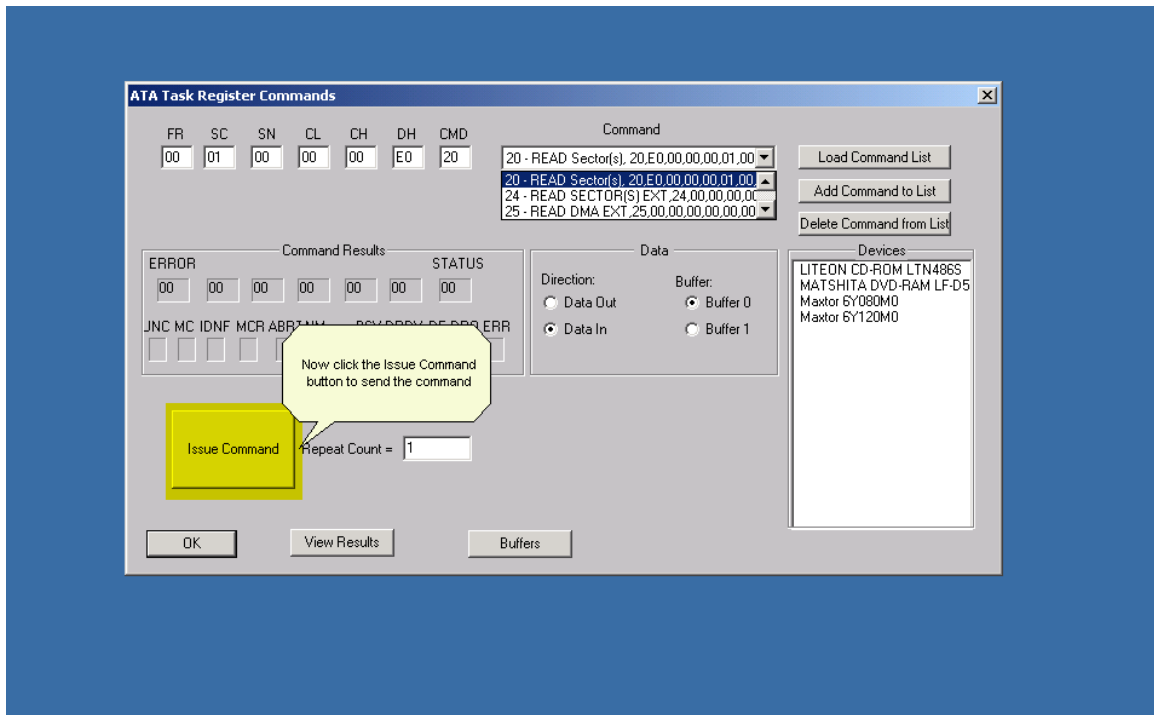
If you want the command issued multiple times specify the repeat count



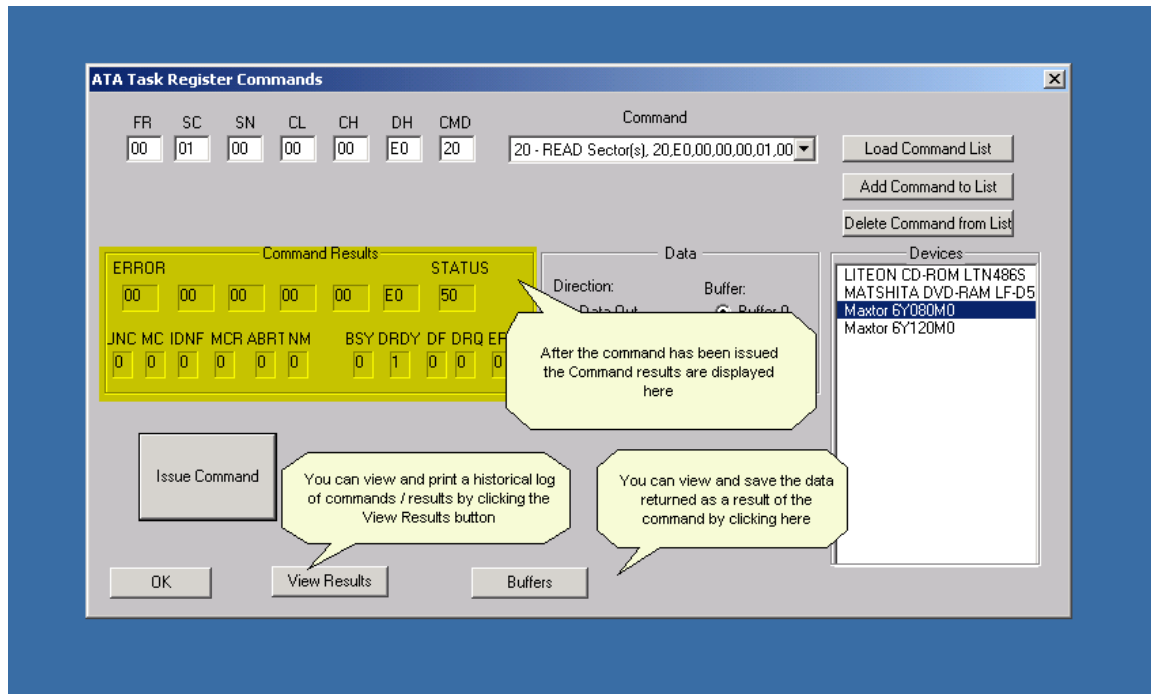
But first – you need to select which ATA device you want to send the command to. Do this by double-clicking on the device. Please – don't destroy data on a drive that you care about! You will be warned if you choose "Data Out" – don't be hasty!



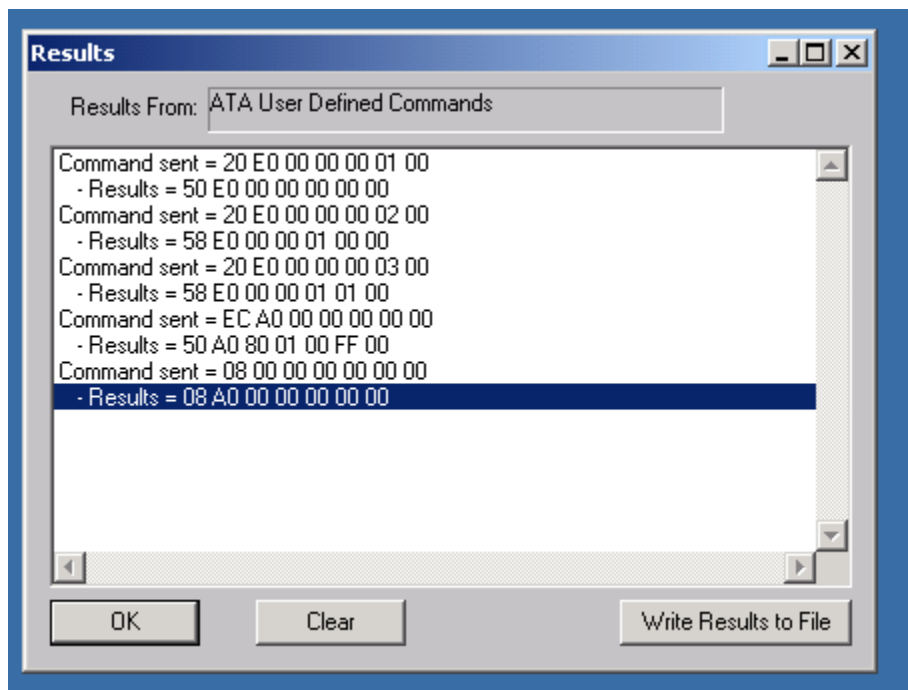
Now you are ready to send the command – click the Issue Command button to do it



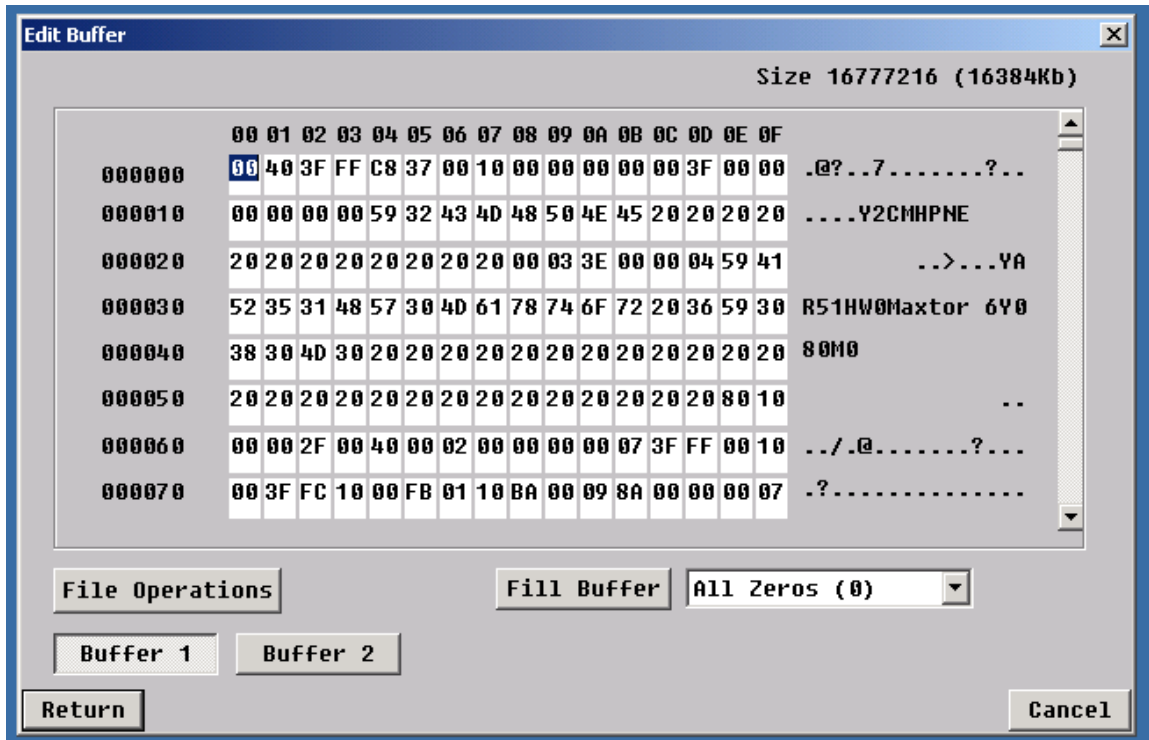
When you issue a command it can succeed or it can fail. The Error and Status indicators let you know what happened.



You can view or save a history of the commands you have issued and their results by clicking the View Results button



If you would like to view the data resulting from a command click the Buffers button – below is data from an ATA IDENTIFY command



Summary

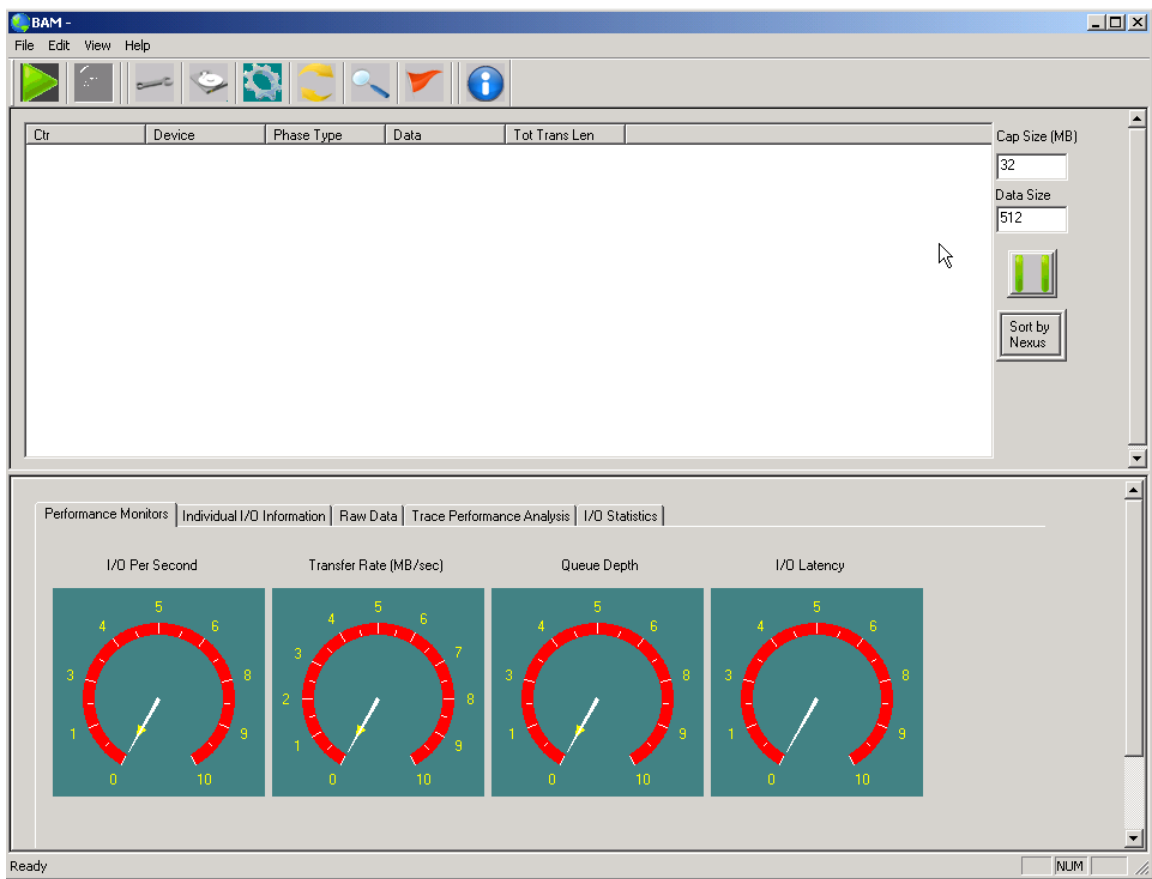
The ATA Commands addition to the SCSI toolbox Suite brings a new level of testing functionality – hardware-independent ATA/SATA testing.

In addition to the GUI-based ATA Commands, the ATA User Defined Command functionality has been added to the Developers Toolbox api library. This enables any user test written with Visual Basic or Visual C++ to have complete ATA task register level access to any ATA device.

Bus Analyzer Module

Description of BAM

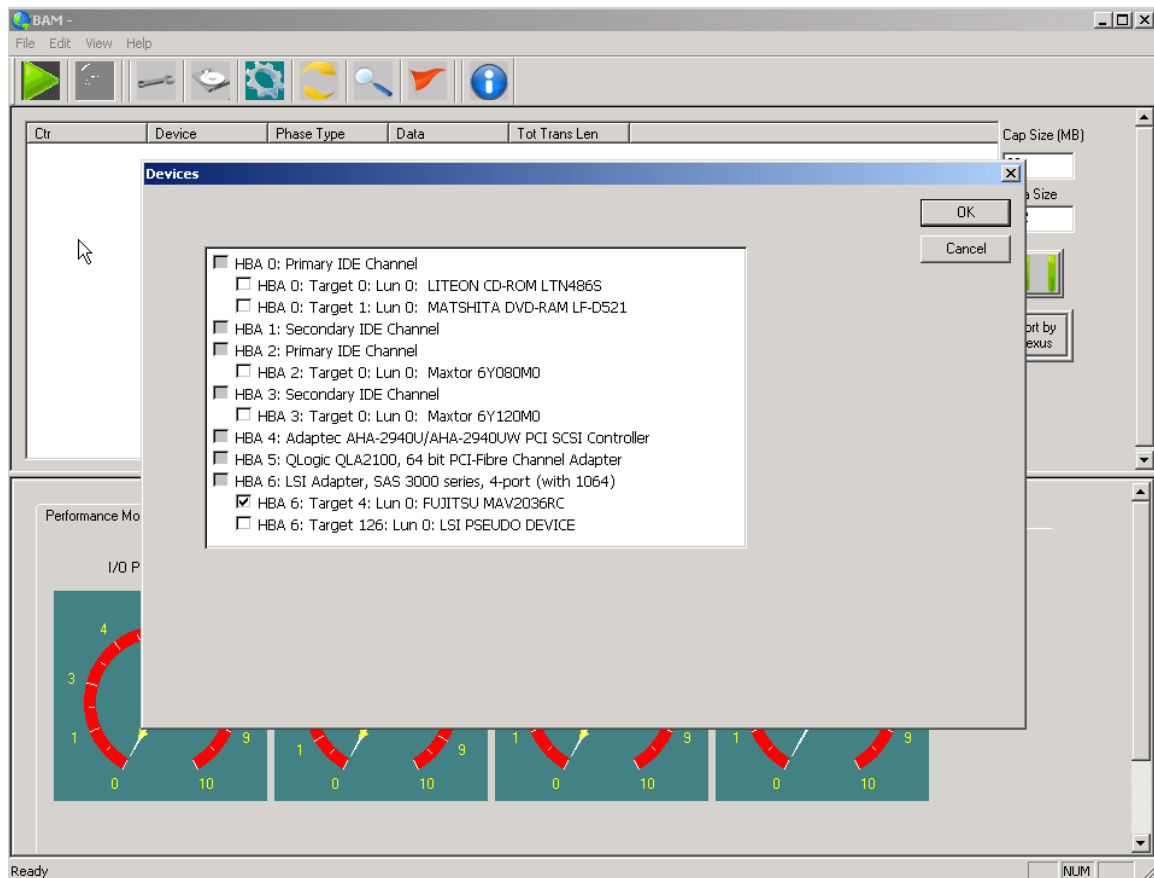
BAM is a software bus analyzer that can capture, display, and analyze trace data from any peripheral bus, including SCSI, Fibre Channel, IDE, ATA, SATA, and SAS. BAM offers complete versatility as far as choice of phases that are captured and displayed, capture modes to minimize I/O impact, buffer size and capture size, and device(s) to capture trace data from.



Pre-Capture Setup

Identifying devices to capture

The Device screen displays all devices attached to all adapters on the system. In performance mode you can select one device, in capture mode you can select as many devices as you wish.



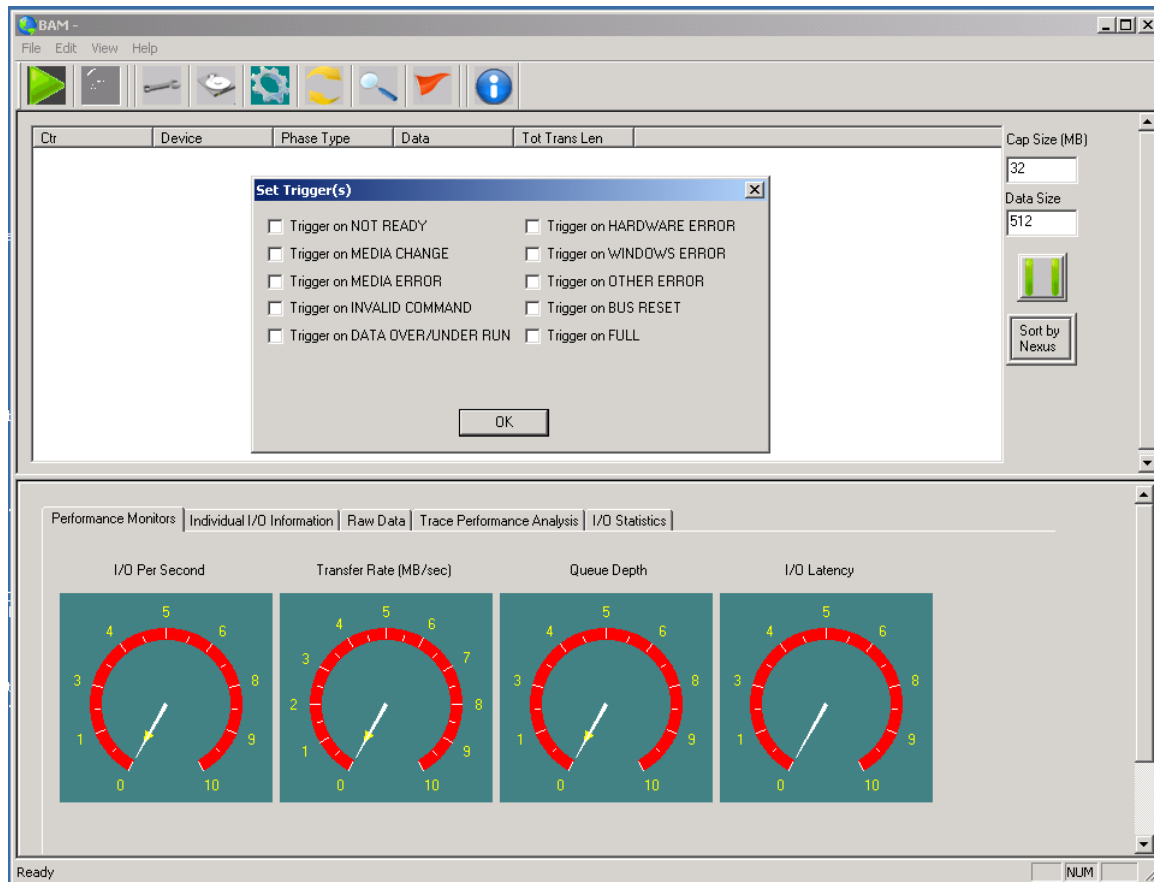
Identifying Phases to capture

Select the phases that you wish to record during capture by using the Phases screen. The phases that can be captured are:

CMD	- SCSI or ATAPI commands
IN/OUT	- Data in or out phase
OK	- Successful completion of a command that does not have a data phase
SENSE	- Sense data captured when a Check Condition error occurs
ATA	- ATA task register command
ASTS	- ATA task register command status
SPT	- SCSI Pass through block
SRB	- SCSI Request block
SSTS	- SCSI request block status
IRP	- Windows I/O Request Packet

Selecting Triggers

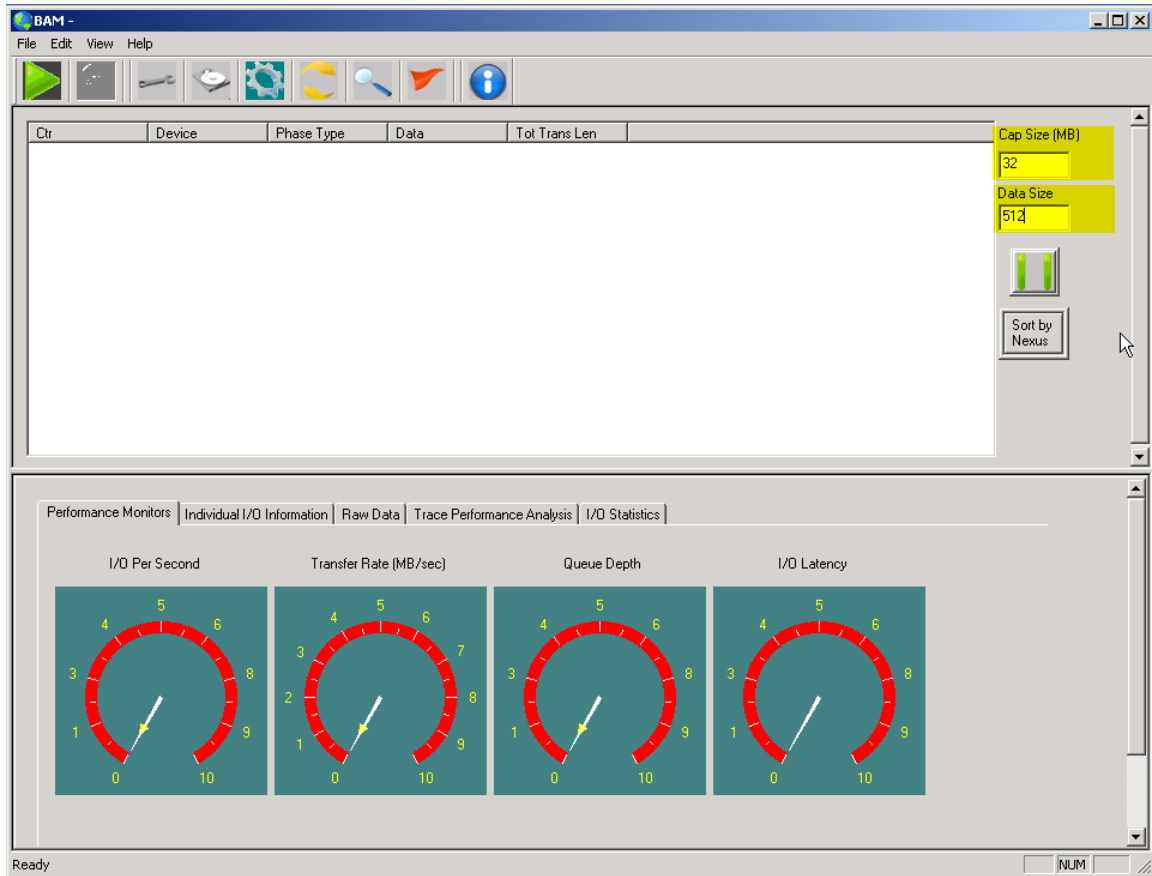
Use the Trigger Screen to select trigger conditions to stop trace capture. When a trigger condition occurs during data capture the capture will stop and a message box will inform you of which trigger condition caused the stop.



Setting Capture Buffer size and Phase Data Size

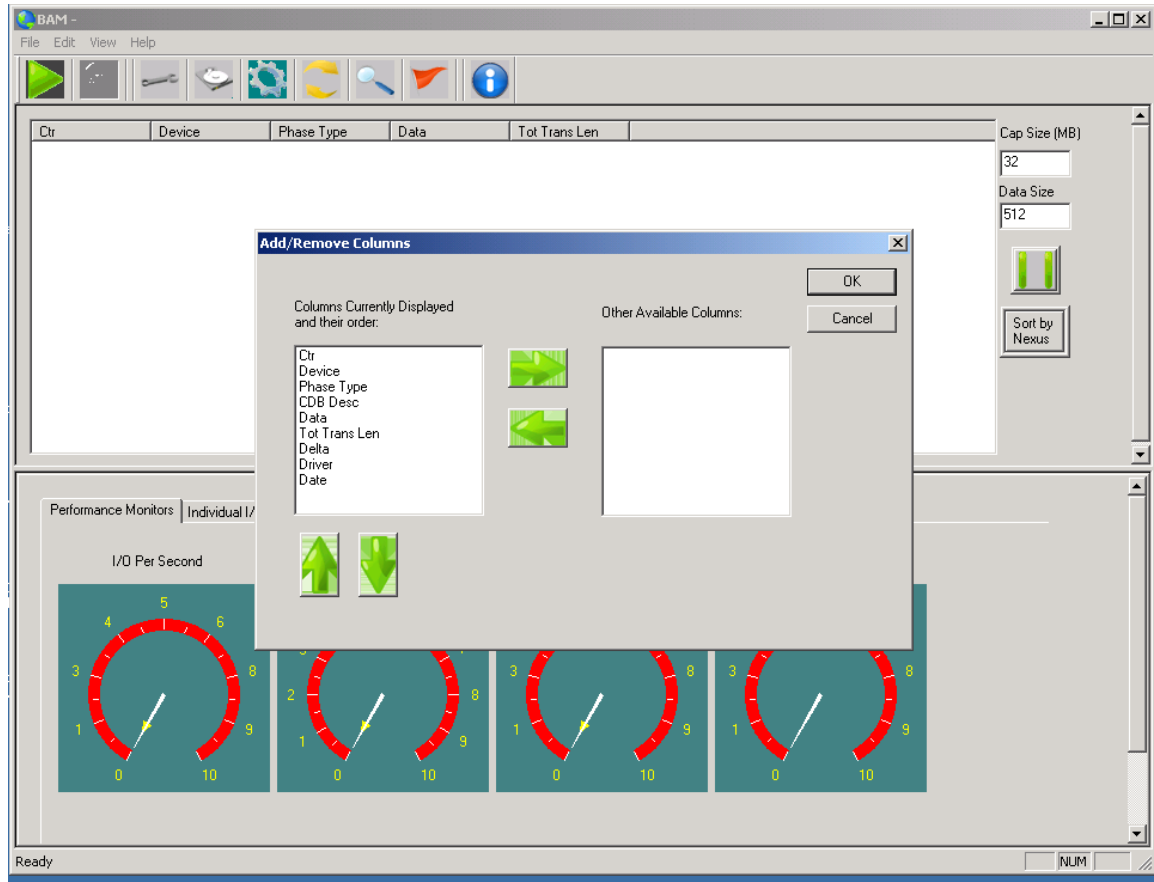
The amount of trace data you will be able to capture is determined by the size of the Capture Buffer. The Capture Buffer uses system memory – enter the amount of memory you can allocate to trace memory in the Cap Size field. Enter this number in Megabytes.

Enter the amount of data (in bytes) from each phase that you wish to capture in the Data Size field – up to 64K.



Identifying columns to display

Use the Columns screen to select the data you wish displayed on the screen, and the order you would like the data columns to be in. In our example we will show all data columns.



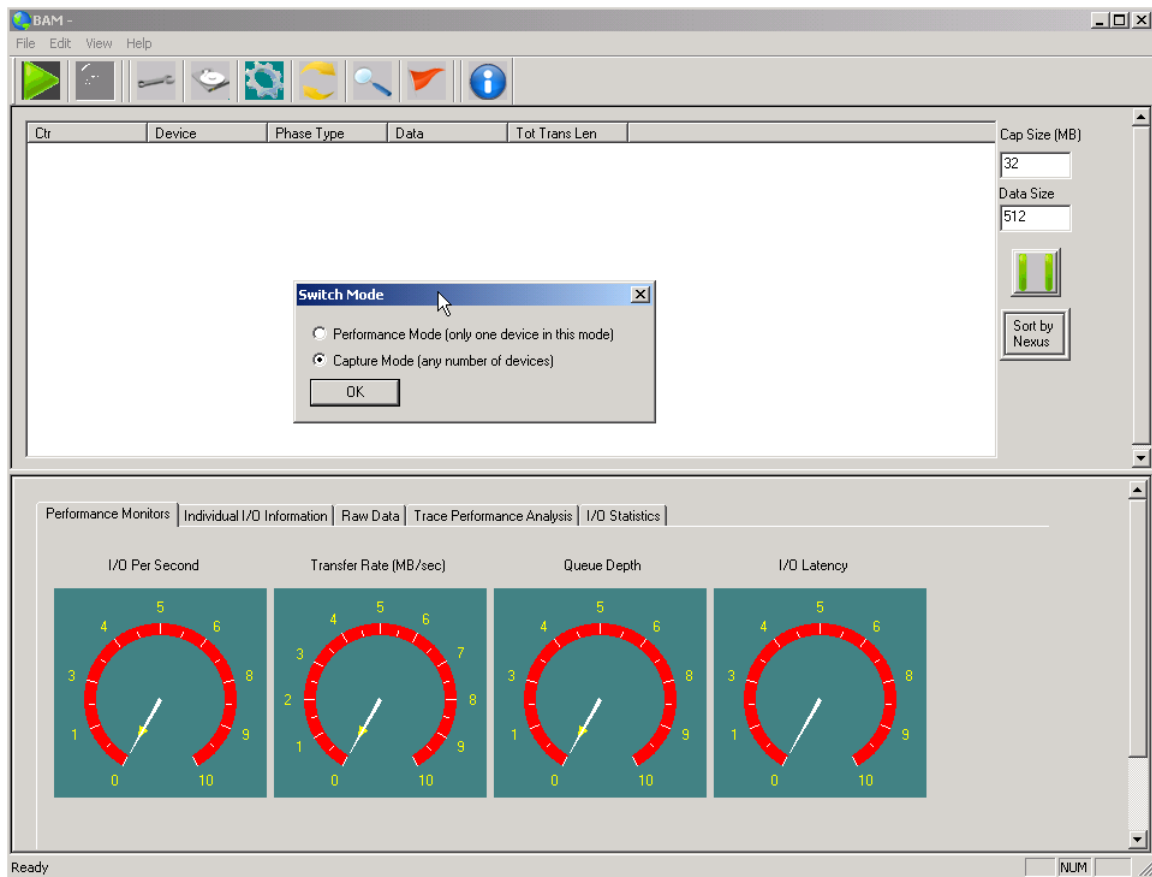
Selecting the mode

Use the Mode Selection screen to choose between Performance mode and Capture Mode.

Performance mode allows data from one device to be captured, and the phases displayed in real-time during capture will be limited to not impact system performance. Once a Performance Mode capture is stopped all phases will be displayed on the screen.

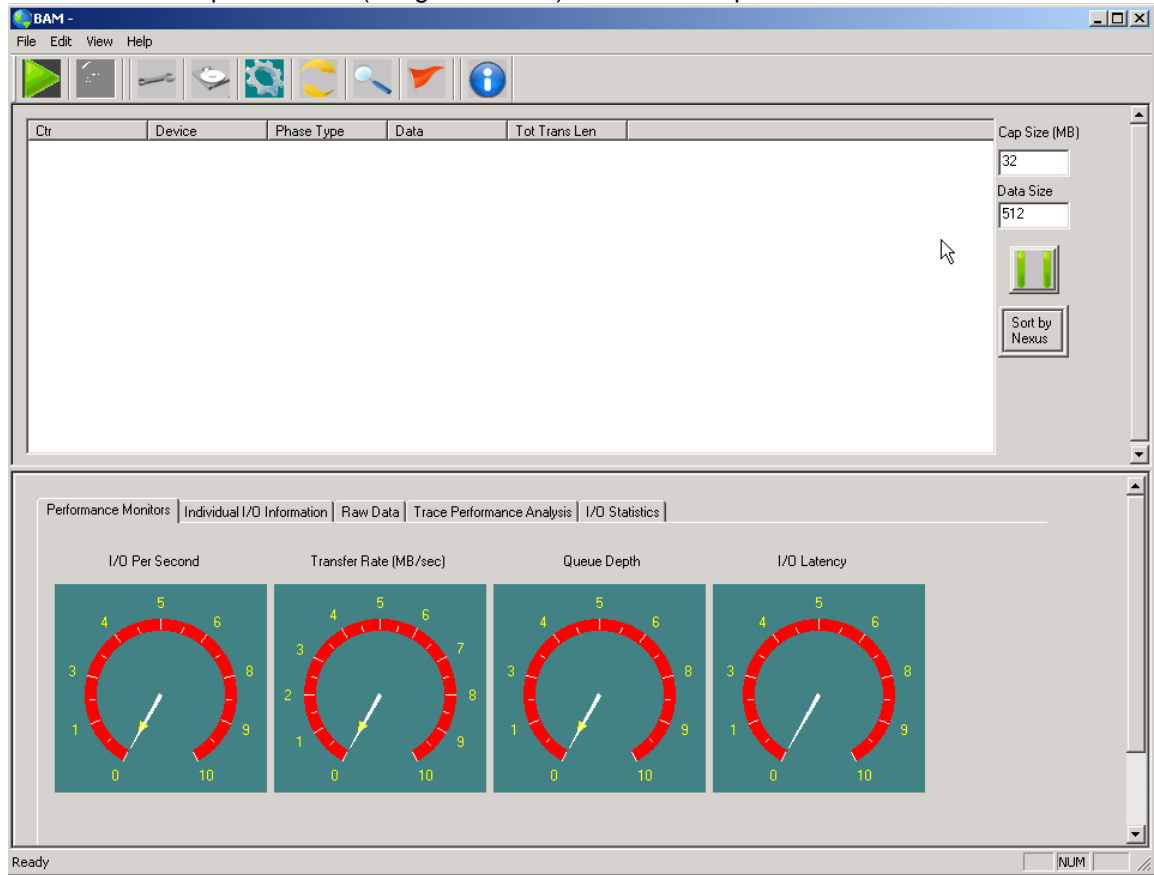
In Capture mode you may capture data from multiple devices, and all phases are displayed in real-time during capture.

Note: - during Capture Mode capture you may use the Pause/Continue button to the right of the phase display to momentarily pause phase data scrolling.



Starting and stopping capture

Click the Start Capture button (the green arrow) to start the capture



And click the Stop button (the red stop sign) to stop capture

The screenshot displays the BAM (Bus Analyzer Monitor) software interface. The top window shows a table of captured data with columns for Ctr, Device, Phase Type, CDB Desc, Data, Tot Trans Len, Delta, and Driver. A red stop sign button is visible in the toolbar. The bottom window shows performance monitors with four gauges: I/O Per Second, Transfer Rate (MB/sec), Queue Depth, and I/O Latency.

Ctr	Device	Phase Type	CDB Desc	Data	Tot Trans Len	Delta	Driver
0	6:4:0	CDB	Read Capacity	25 00 00 00 00 ...	10 Bytes	58 us	stsclass
1	6:4:0	CDB	Write (10)	2A 08 00 83 27 ...	10 Bytes	117.8 ms	stsclass

Performance Monitors: Individual I/O Information | Raw Data | Trace Performance Analysis | I/O Statistics

- I/O Per Second: 138
- Transfer Rate (MB/sec): 12
- Queue Depth: 159
- I/O Latency: 5

Ready NUM

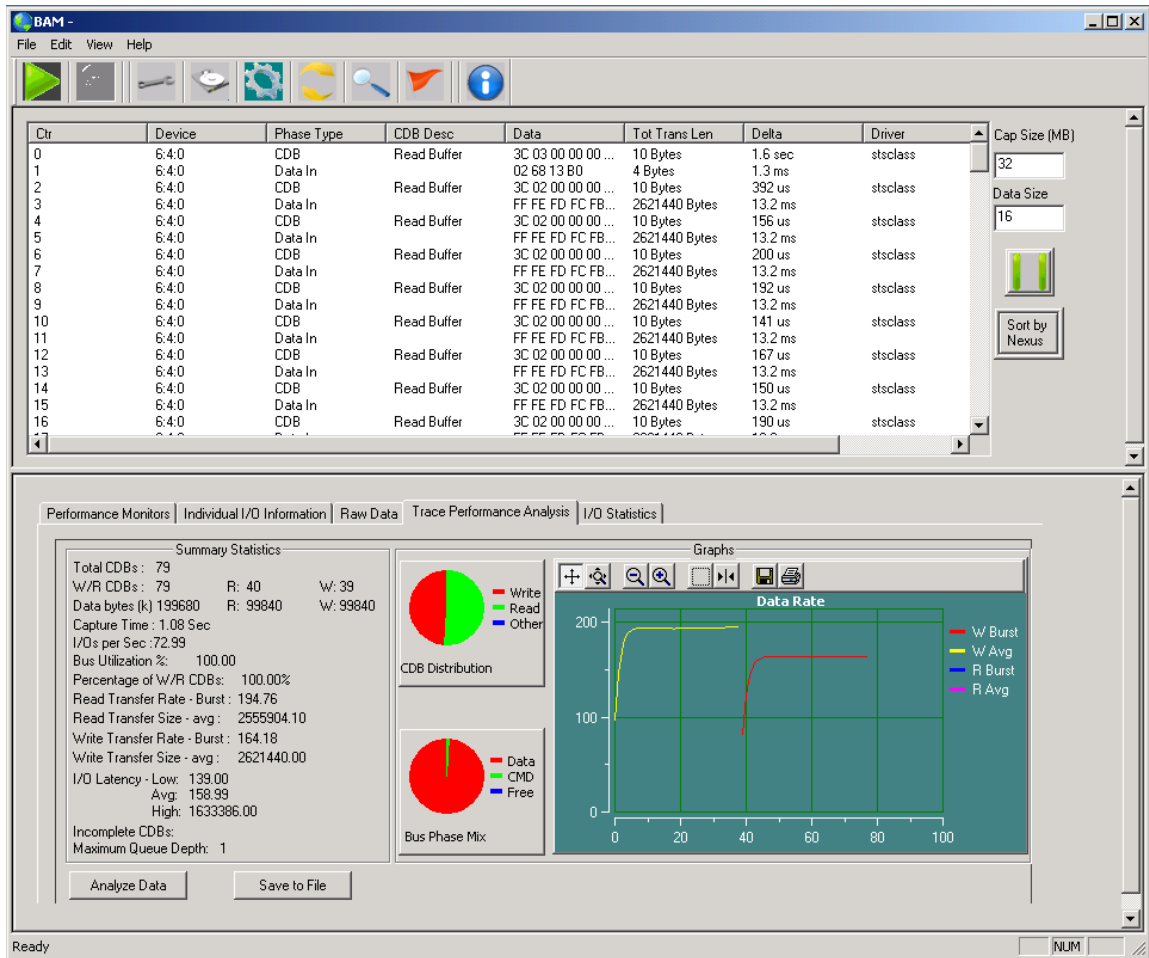
Monitoring while tracing

During trace capture the Performance Monitor tab will show real-time performance data such as Number of I/Os per second, Transfer rate in MB/second, Maximum Queue Depth achieved, and I/O Latency. These displays show the maximum, average, and minimum values.

Post Capture Analysis

Trace Performance Analysis

The Trace Performance Analysis tab displays statistical information about the trace data



Information on type of I/O, data size, transfer rates, I/O rates, and intercommand I/O latency are displayed, as well as graphs of CDB type distribution, and Bus Phase mixture over the trace capture period. Read and Write Transfer rates are displayed in a scrollable graph.

If you start another trace capture be sure to click the **Analyze Data** button once the capture has been stopped in order to recalculate and display the current data.

The Data Rate graph can be scrolled by clicking on the axis that you wish to scroll, then using the keyboard arrow keys to scroll the graph.

You may zoom on any graph axis by clicking on the zoom control (the second tool from the left at the top of the graph) then using the + and/or - buttons to zoom.

Data Analysis

The top section of the tool displays trace data, including the device ID, the Phase Type, CDB Description, Data, Data Length, Delta Time, what OS driver issued the IO, and Date/Time.

Note: - trace data can be displayed in two modes:

1. Sorted by time
2. Sorted by Nexus

Time sorting displays the trace data in the order it occurred on the bus.

Nexus sorting will group phase data for a given nexus together. For example, nexus mode will group commands and data together even if other commands had occurred between the command and data phase, such as would be the case with command tag queuing or with interleaved capture of multiple device data.

The default mode is Sorted by Time – change modes by clicking the “Sort by” button

The screenshot shows the BAM software interface. The top window displays a table of trace data with columns: Ctr, Device, Phase Type, CDB Desc, Data, Tot Trans Len, Delta, and Driver. The data is sorted by time. Below the table, there is a 'Sort by' button with a dropdown menu showing 'Nexus' selected. The bottom window shows the 'Individual I/O Information' tab, displaying details for a specific I/O operation:

Field	Value
Originator	stsclass
Device	6:4:0
Operation Code	3C
CDB	3C 02 00 00 00 28 00 00 00
Bytes Transferred	10
SRB Status	01
Target Status	00
Sense Key	-
Sense ASC	-
Sense ASCQ	-

The status bar at the bottom indicates 'Ready' and 'NUM'.

Searching through Trace Data

The Search tool allows you to specify a data pattern to search for within the trace data

The screenshot displays the BAM software interface. The top section is a table of trace data with columns: Ctr, Device, Phase Type, CDB Desc, Data, Tot Trans Len, Delta, Driver, and Cap Size (MB). A 'Search Trace Data' dialog box is open, showing a search for 'Sync Cache'. The bottom section shows performance monitors for I/O Per Second, Transfer Rate (MB/sec), Queue Depth, and I/O Latency, each represented by a circular gauge.

Ctr	Device	Phase Type	CDB Desc	Data	Tot Trans Len	Delta	Driver	Cap Size (MB)
0	2:0:0	CDB	Write (10)	2A 08 00 08 65 ...	10 Bytes	771.3 ms	unknown	32
1	2:0:0	Data Out		49 4E 44 58 28 ...	4096 Bytes	134 us	unknown	512
2	2:0:0	CDB	Write (10)	2A 08 00 45 C7 ...	10 Bytes	359.3 ms	unknown	
3	2:0:0	Data Out		52 43 52 44 28 ...	8192 Bytes	145 us	unknown	
4	2:0:0	CDB	Write (10)	2A 08 00 45 16 ...	10 Bytes	56 us	unknown	
5	2:0:0	Data C				143 us	unknown	
6	2:0:0	CDB				89 us	unknown	
7	2:0:0	Data C				136 us	unknown	
8	2:0:0	CDB				1.6 sec	unknown	
9	2:0:0	Data C				48 us	unknown	
10	2:0:0	CDB				899.8 ms	unknown	
11	2:0:0	Data C				134 us	unknown	
12	2:0:0	CDB				899.9 ms	unknown	
13	2:0:0	Data Out		49 4E 44 58 28 ...	4096 Bytes	136 us	unknown	
14	2:0:0	CDB	Write (10)	2A 08 00 45 16 ...	10 Bytes	218.7 ms	unknown	
15	2:0:0	Data Out		52 43 52 44 28 ...	4096 Bytes	149 us	unknown	
16	2:0:0	CDB	Sync Cache	35 00 00 00 00 ...	10 Bytes	61 us	classprp	

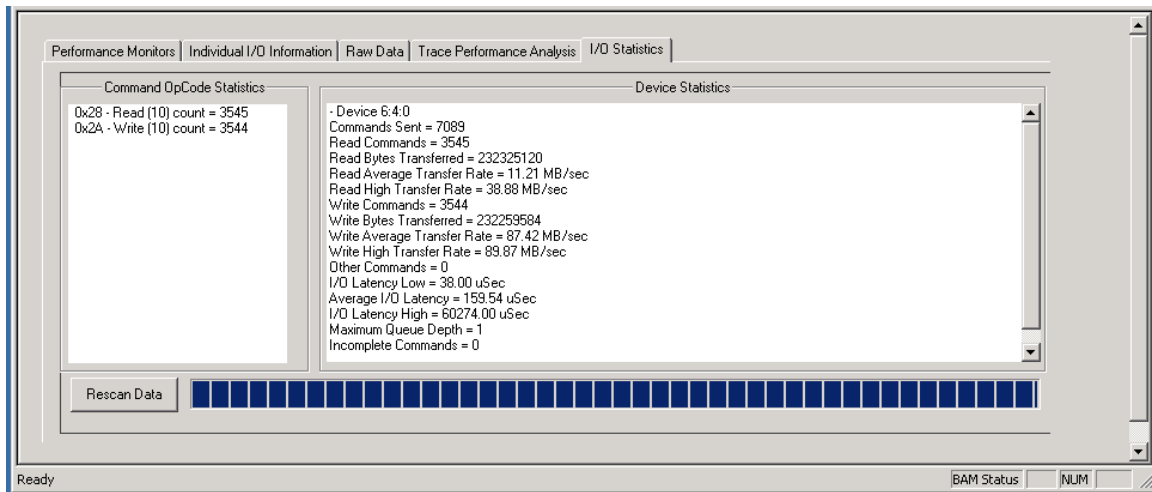
Performance Monitors: Individual I/O Information | Raw Data | Trace Performance Analysis | I/O Statistics

- I/O Per Second: Gauge showing values 0, 10, 19, 29, 39, 48, 58, 68, 78.
- Transfer Rate (MB/sec): Gauge showing values 0, 4, 8, 11, 15, 19, 23, 26, 30, 34, 38.
- Queue Depth: Gauge showing values 0, 1, 3, 4, 5, 6, 8, 9, 10.
- I/O Latency: Gauge showing values 0, 1, 3, 4, 5, 6, 8, 9, 10.

Ready NUM

IO Statistics

Statistics about the type and amount of commands captured may be viewed with the IO Statistics tab



If you start another trace capture be sure to click the **Rescan Data** button once the capture has been stopped in order to recalculate and display the current data.

Check Conditions

If an IO fails resulting in a check condition the IO will be marked red in the phase display, as shown here:

Ctrl	Device	Phase Type	CDB Desc	Data	Data Length	Delta	Drive	Cap Size (MB)
0	6:4:0	Reset			0 Bytes	12 us	stbt	32
1	6:4:0	CDB	Read (10)	28 00 00 00 00 00 00 01 00	10 Bytes	8.6 sec	stclas	Data Size
2	6:4:0	Data In		00 83 00 0C 61 93 00 08 50 00 00 E0 00 03 F2 50	512 Bytes	1.2 ms		512
3	6:4:0	Sense		70 00 06 00 00 00 00 28 00 00 00 00 29 02 00 00	18 Bytes	3 us		

Sort by
Nexus

Performance Monitors | Individual I/O Information | Raw Data | Trace Performance Analysis | I/O Statistics

Originator: stsclass
 Device: 6:4:0
 Operation Code: 28

CDB: 28 00 00 00 00 00 00 01 00
 Bytes Transferred: 10

Error Information

SRB Status: C4	Sense Key: 06
Target Status: 02	Sense ASC: 29
	Sense ASCQ: 02

Disk Tests

Disk Test Options

DriveWatch

DriveWatch is a real-time performance display that is accessible from any of the SCSItoolbox32 tests.

Information displayed:

- Average, minimum, and maximum read and write transfer rates, along with their respective LBA.
- Total number of blocks read and written
- Blocks per transfer currently in use
- Number of hard (uncorrectable) and soft (correctable) errors
- Number of read and write I/O's per second
- DriveWatch will graphically display performance and error information

SCSI DriveWatch ✖

SCSI Target = LUN =

io's per second

Read Data Rates			
Average Transfer Rate =	<input type="text" value="0.000"/> KB/Sec		<input type="text" value="0.0"/>
High Transfer Rate =	<input type="text" value="0.000"/> KB/Sec , at Block number	<input type="text" value="0"/>	<input type="text" value="0.0"/>
Low Transfer Rate =	<input type="text" value="0.000"/> KB/Sec , at Block number	<input type="text" value="0"/>	<input type="text" value="0.0"/>

Write Data Rates			
Average Transfer Rate =	<input type="text" value="45976.117"/> KB/Sec		<input type="text" value="718.4"/>
High Transfer Rate =	<input type="text" value="55349.332"/> KB/Sec , at Block number	<input type="text" value="413440"/>	<input type="text" value="864.8"/>
Low Transfer Rate =	<input type="text" value="5968.396"/> KB/Sec , at Block number	<input type="text" value="256"/>	<input type="text" value="93.3"/>

Blocks per Transfer =

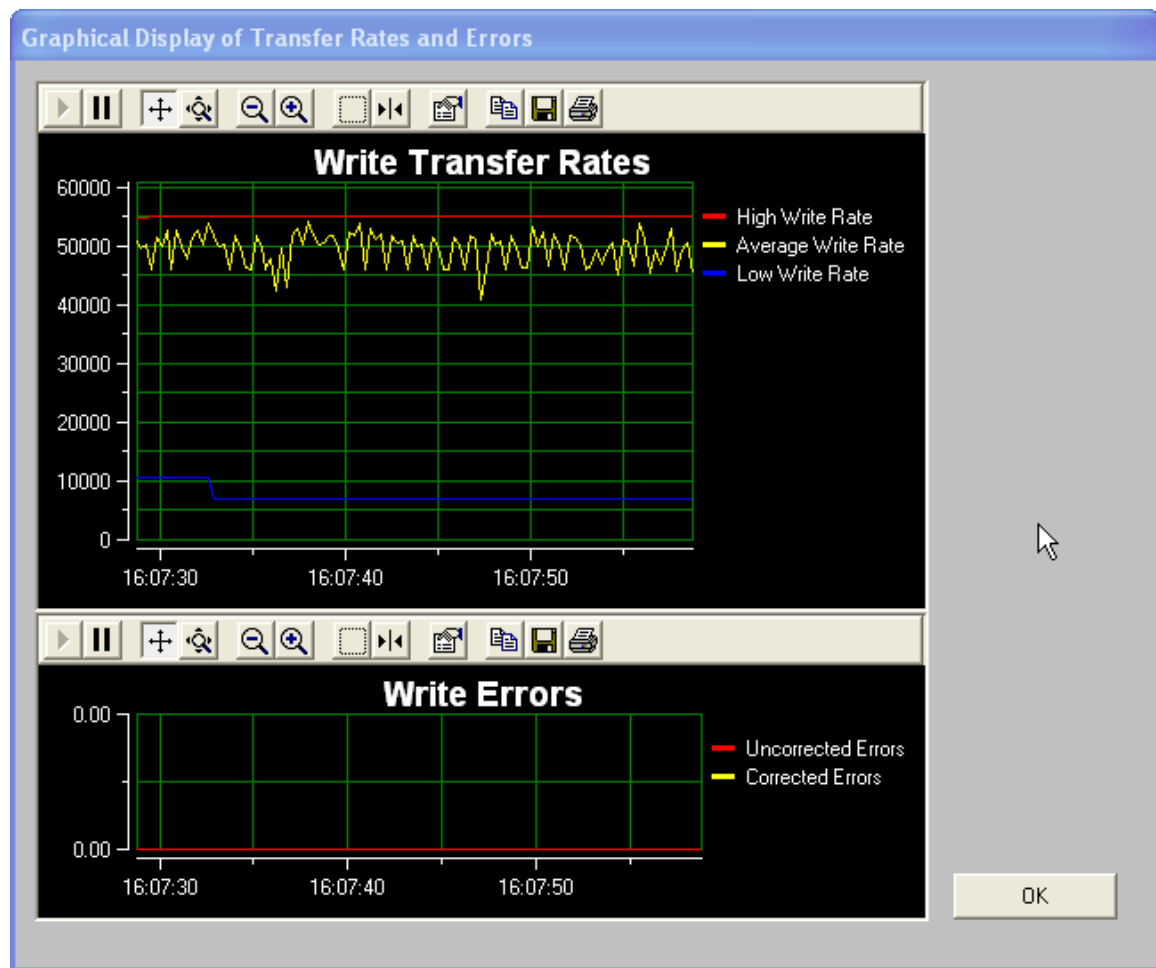
Hard Error Count = last error at Block number

Soft Error Count = last error at Block number

Total Blocks Written =

Total Blocks Read =

The New Graphing display within drivewatch:



Test Alarms

The test alarms allow all tests to be aborted upon the following conditions:

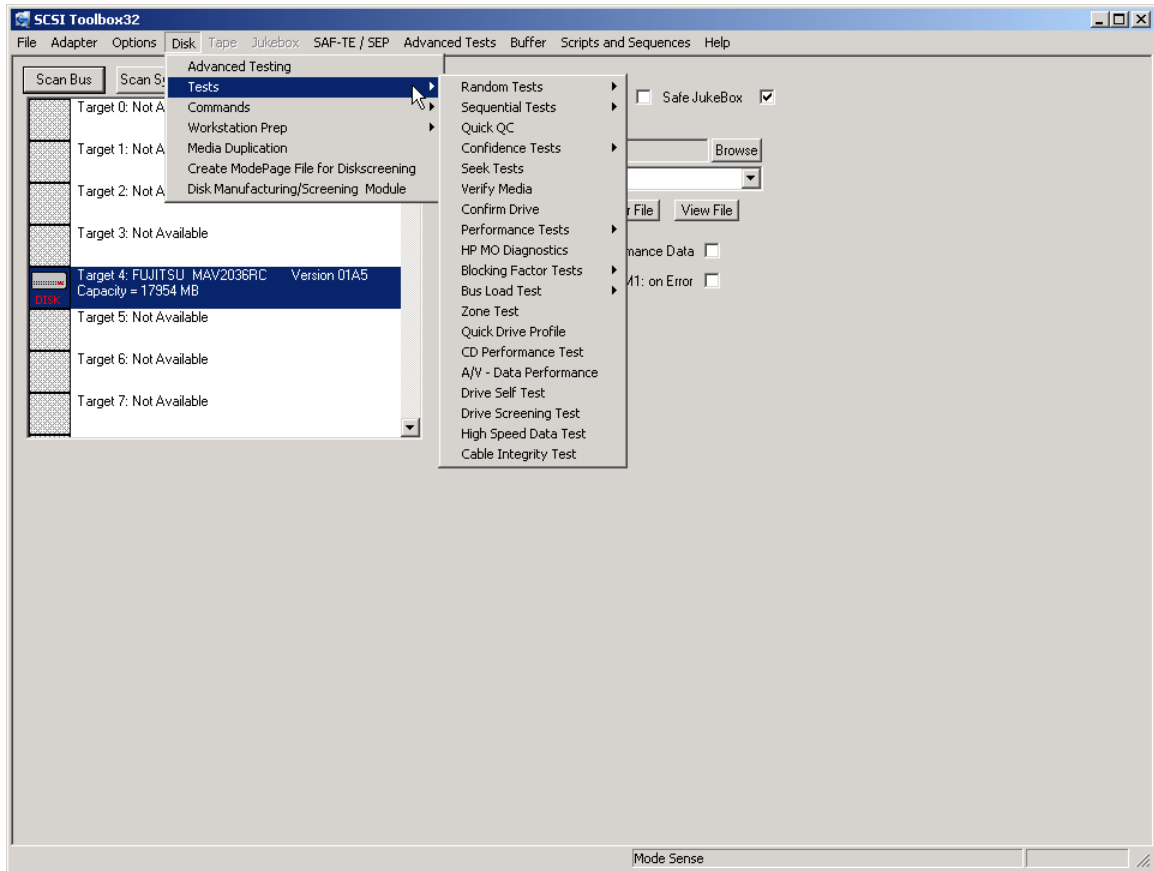
- If the number of hard errors exceeds a threshold
- If the number of soft errors exceeds a threshold
- If the high, low, or average read or write transfer rate falls below a threshold

Test Alarms [X]

Stop Test if:

- Hard errors are greater than
- Soft errors are greater than
- Read transfer rate is less than KB/sec Avg Hi Low
- Write transfer rate is less than KB/sec Avg Hi Low

The following tests may be accessed from the Test Menu SCSItoolbox32 (details on each test follow):



Sequential Read

This test will sequentially read all blocks between the Start Block and the End block. If a read fails the test will stop and the details of the error will be reported.

SCSI Disk Device Tests

Test Name: Sequential Read Test (Non-Destructive) Ignore Blank Checks

Total Blocks: 4109999

Current Block: 0

Last Error at Block: 0

Time to Complete: --:--:--

Average Transfer Rate (KB / Sec.): 0

Start Block: 0

End Block: 4109999

Blocks per Transfer: 128

Number of Passes: 1

Use Block Number

Test Results :
Click Start to begin test - Return to exit

Return Start Test Drive Watch Set Alarms

User Test Parameters:	
Parameter	
Start Block	Enter the LBA to start the sequential read test at Default = 0
End Block	Enter the LBA where the test should stop Default = highest LBA as reported by the READ CAPACITY command.
Blocks per Transfer	Enter number of blocks per CDB Default = 128 blocks (64K per transfer)
Number of Passes	Enter the number of test passes Default = 1
Ignore Blank Checks	Check this selection if you are testing WORM or MO media that has not had all blocks within the test range previously written.
Use Block Number	If checked the read test will assume that the blocks being read have been written by the Sequential Write Test with this option checked. Each block will be checked to see that it contains a data pattern based on the LBA. If the data pattern does not compare the test will fail with a data compare error.

Sequential Write

This test will sequentially write all blocks between the Start Block and the End block. If a write fails the test will stop and the details of the error will be reported.

The screenshot shows the 'SCSI Disk Device Tests' window. The 'Test Name' is 'Sequential Write Test (Destructive)'. The 'Ignore Blank Checks' checkbox is unchecked. The 'Total Blocks' is 4109999, 'Current Block' is 0, and 'Last Error at Block' is 0. The 'Time to Complete' is shown as '---:--'. The 'Average Transfer Rate (KB / Sec.)' is 0. The 'Start Block' is 0 and the 'End Block' is 4109999. The 'Blocks per Transfer' is 128 and the 'Number of Passes' is 1. There is a 'Use Block Number' checkbox which is unchecked, and a 'Set Data Pattern' button followed by a row of buttons for data patterns: 00, 3E, B6, AF, 00, 00, 02, 00. Below this is a large empty text area. The 'Test Results' section contains the text 'Click Start to begin test - Return to exit'. At the bottom are four buttons: 'Return', 'Start Test', 'Drive Watch', and 'Set Alarms'.

User Test Parameters:	
Parameter	
Start Block	Enter the LBA to start the sequential write test at Default = 0
End Block	Enter the LBA where the test should stop Default = highest LBA as reported by the READ CAPACITY command.
Blocks per Transfer	Enter number of blocks per CDB Default = 128 blocks (64K per transfer)
Number of Passes	Enter the number of test passes Default = 1
Ignore Blank Checks	Used if testing WORM or MO disks.
Use Block Number	Checking this option will write a data pattern that contains the LBA to each block.
Set Data Pattern	Used to specify a data pattern to be written. Data patterns can also be set using the Buffer Menu choice.

Sequential Write/Read

This test will sequentially write, then read all blocks between the Start Block and the End block. If a write or read fails the test will stop and the details of the error will be reported.

User Test Parameters:

Parameter	
Start Block	Enter the LBA to start the test Default = 0
End Block	Enter the LBA where the test should stop Default = highest LBA as reported by the READ CAPACITY command.
Blocks per Transfer	Enter number of blocks per CDB Default = 128 blocks (64K per transfer)
Number of Passes	Enter the number of test passes Default = 1
Ignore Blank Checks	Used if testing WORM or MO disks
Use Block Number	Checking this option will write a data pattern that contains the LBA to each block
Set Data Pattern	Used to specify a data pattern to be written. Data patterns can also be set using the Buffer Menu choice

Map Drive Defects

This test sequentially reads all blocks between Start Block and End Block. The error recovery information reported by the drive via Sense Data and Log Pages is displayed and may be recorded to a log file using the Log Results to Disk option.

Map Drive Defects

Starting Block = Blocks Scanned =

Ending Block = Log Results to Disk

Errors reported via Sense Data

Blocks Recovered	=	<input type="text" value="0"/>	Blocks Recovered using positive offset	<input type="text" value="0"/>
Blocks Recovered with no ECC	=	<input type="text" value="0"/>	Blocks Recovered using negative offset	<input type="text" value="0"/>
Blocks Recovered with retries	=	<input type="text" value="0"/>	Blocks Recovered Data AutoReallocated	<input type="text" value="0"/>
Blocks Recovered with ECC	=	<input type="text" value="0"/>	Blocks Recovered Previous Sector ID	<input type="text" value="0"/>
Blocks Recovered with ECC + Retries	=	<input type="text" value="0"/>	Blocks Recovered w/ ECC - AutoReallocated	<input type="text" value="0"/>
Unrecovered Blocks	=	<input type="text" value="0"/>	Blocks Recovered w/ ECC and/or Retries	<input type="text" value="0"/>

Errors reported via Log Pages

Errors Recovered without delay	=	<input type="text" value="0"/>	Total times recovery invoked	=	<input type="text" value="0"/>
Errors Recovered with retries	=	<input type="text" value="0"/>	Total unrecovered errors	=	<input type="text" value="0"/>
Errors Recovered with ECC	=	<input type="text" value="0"/>	Total errors recovered	=	<input type="text" value="0"/>

User Test Parameters:

Parameter	
Start Block	Enter the LBA to start the test Default = 0
End Block	Enter the LBA where the test should stop Default = highest LBA as reported by the READ CAPACITY command.
Log Results to Disk	Check this box to specify a log file to record the results of the test

Sequential Write/Read

This test will sequentially write and verify all blocks between the Start Block and the End block. If a write or verify fails the test will stop and the details of the error will be reported.

User Test Parameters:	
Parameter	
Start Block	Enter the LBA to start the test Default = 0
End Block	Enter the LBA where the test should stop Default = highest LBA as reported by the READ CAPACITY command.
Blocks per Transfer	Enter number of blocks per CDB Default = 128 blocks (64K per transfer)
Number of Passes	Enter the number of test passes Default = 1
Ignore Blank Checks	Used if testing WORM or MO disks
Use Block Number	Checking this option will write a data pattern that contains the LBA to each block
Set Data Pattern	Used to specify a data pattern to be written. Data patterns can also be set using the Buffer Menu choice

Random Read

This test will read random blocks between the Start Block and the End block. If a read fails the test will stop and the details of the error will be reported. This test will run until stopped by pressing the Cancel button.

User Test Parameters:	
Parameter	
Start Block	Enter the LBA to start the test Default = 0
End Block	Enter the LBA to define the high range of the test Default = highest LBA as reported by the READ CAPACITY command.
Blocks per Transfer	Enter number of blocks per CDB Default = 128 blocks (64K per transfer)
Number of Passes	Enter the number of test passes Default = 1
Ignore Blank Checks	Used if testing WORM or MO disks
Use Block Number	If checked the read test will assume that the blocks being read have been written by the Sequential Write Test with this option checked. Each block will be checked to see that it contains a data pattern based on the LBA. If the data pattern does not compare the test will fail with a data compare error

Random Write

This test will write random blocks between the Start Block and the End block. If a write fails the test will stop and the details of the error will be reported. This test will run until stopped by pressing the Cancel button.

SCSI Disk Device Tests

Test Name: Random Write Test (Destructive) Ignore Blank Checks

Total Blocks: 4109999

Current Block: 0

Last Error at Block: 0

Time to Complete: --:--:--

Average Transfer Rate (KB / Sec.): 0

Start Block: 0

End Block: 4109999

Blocks per Transfer: 128

Number of Passes: 1

Use Block Number 00 3E B6 AF 00 00 02 00

Test Results :
Click Start to begin test - Return to exit

User Test Parameters:	
Parameter	
Start Block	Enter the LBA to define the low range of the test Default = 0
End Block	Enter the LBA to define the high range of the test Default = highest LBA as reported by the READ CAPACITY command.
Blocks per Transfer	Enter number of blocks per CDB Default = 128 blocks (64K per transfer)
Number of Passes	Enter the number of test passes Default = 1
Ignore Blank Checks	Used if testing WORM or MO disks
Use Block Number	Check this selection if you are testing WORM or MO media that has not had all blocks within the test range previously written

Random Write/Read

This test will write and read random blocks between the Start Block and the End block. If a write or read fails the test will stop and the details of the error will be reported. This test will run until stopped by pressing the Cancel button.

User Test Parameters:

Parameter	
Start Block	Enter the LBA to define the low range of the test Default = 0
End Block	Enter the LBA to define the high range of the test Default = highest LBA as reported by the READ CAPACITY command.
Blocks per Transfer	Enter number of blocks per CDB Default = 128 blocks (64K per transfer)
Number of Passes	Enter the number of test passes Default = 1
Ignore Blank Checks	Used if testing WORM or MO disks
Use Block Number	Checking this option will write a data pattern that contains the LBA to each block. The read operation will confirm this data pattern and stop with a data compare error if the data is not correct
Set Data Pattern	Used to specify a data pattern to be written. Data patterns can also be set using the Buffer Menu choice.

Random Write/Verify

This test will write and verify random blocks between the Start Block and the End block. If a write or verify fails the test will stop and the details of the error will be reported. This test will run until stopped by pressing the Cancel button.

User Test Parameters:

Parameter	
Start Block	Enter the LBA to define the low range of the test Default = 0
End Block	Enter the LBA to define the high range of the test Default = highest LBA as reported by the READ CAPACITY command.
Blocks per Transfer	Enter number of blocks per CDB Default = 128 blocks (64K per transfer)
Number of Passes	Enter the number of test passes Default = 1
Ignore Blank Checks	Used if testing WORM or MO disks
Use Block Number	Checking this option will write a data pattern that contains the LBA to each block. The read operation will confirm this data pattern and stop with a data compare error if the data is not correct
Set Data Pattern	Used to specify a data pattern to be written. Data patterns can also be set using the Buffer Menu choice.

Quick QC

This test will write and read 1000 blocks at the beginning, middle, and end of the drive. Any write or read errors will be reported.

SCSI Disk Device Tests

Test Name: Ignore Blank Checks

Total Blocks:

Current Block:

Last Error at Block:

Time to Complete:

Average Transfer Rate (KB / Sec.):

Start Block:

End Block:

Blocks per Transfer:

Number of Passes:

Use Block Number

Test Results :

Quick Drive Profile

Quick Drive Profile [X]

Vendor = Product = F/W =

Read Transfer Rate Average = High = KB/S Number of Blocks Read =

Write Transfer Rate Average = High = KB/S Number of Blocks Written =

Read Blocks 2 = 8 = 64 = 128 = 256 = 512 = KB/S

Write Blocks 2 = 8 = 64 = 128 = 256 = 512 = KB/S

Average Seek = Ms Write Cache = Read Cache = Number of Segments =

Number of Primary Defects = Number of Grown Defects = Hard Errors = Soft Errors =

Status:

Enable Destructive Tests

User Test Parameters:

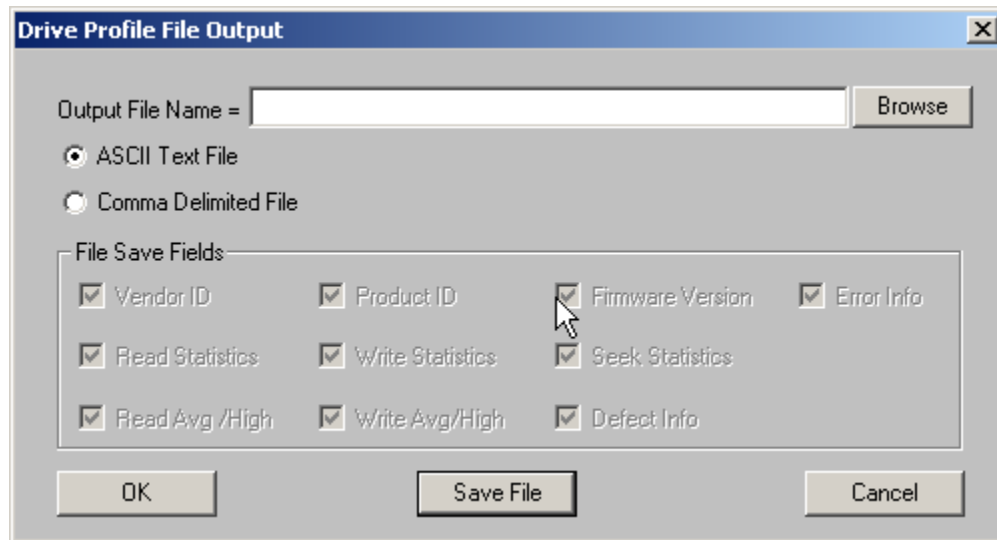
Parameter	
Enable Destructive Test	Checking this option will allow writes to the drive during this test. If not check only read operations will be executed.

Button	Functionality
Profile Drive	Executes a series of tests to measure the drives performance parameters. Parameters measured are Read and Write transfer rates at various blocks-per-transfer, and seek time. Various MODE PAGE settings are displayed, along with the number of Primary and Grown defects, and the number of correctable and uncorrectable errors.
File Output	Displays the Drive Profile File Output dialog box (see the next section).

Drive Profile File Output

The options on the Drive Profile File output dialog allow the performance metrics data to be saved to a file. This file can be either an ASCII text file, or a comma-delimited file that can be used to graph data using Microsoft Excel compatible spreadsheets.

When outputting data to a comma-delimited file the fields of data that are saved can be specified with the check boxes.



The screenshot shows a dialog box titled "Drive Profile File Output" with a close button (X) in the top right corner. The dialog contains the following elements:

- An "Output File Name =" text box with a "Browse" button to its right.
- Two radio buttons: "ASCII Text File" (selected) and "Comma Delimited File".
- A "File Save Fields" section containing a grid of nine checked checkboxes:
 - Vendor ID
 - Product ID
 - Firmware Version
 - Error Info
 - Read Statistics
 - Write Statistics
 - Seek Statistics
 - Read Avg /High
 - Write Avg/High
 - Defect Info
- Three buttons at the bottom: "OK", "Save File", and "Cancel".

AV/Data Performance Comparison

Drive Optimization Test

	Not Optimized .0002% Data Errors	Optimized for A/V .0002% Data Errors	Optimization Increase
Read Speed:	<input type="text"/> MB/s	<input type="text"/> MB/s	<input type="text"/> %
Write Speed:	<input type="text"/> MB/s	<input type="text"/> MB/s	<input type="text"/> %

Status:

This test will measure the data transfer rate of your drive with and without A/V optimization. A 10 MB data pattern that contains 24 bytes of errors is transferred in each test phase.

This test will write data to this disk drive

STARTING THIS TEST WILL DESTROY ANY DATA!!!

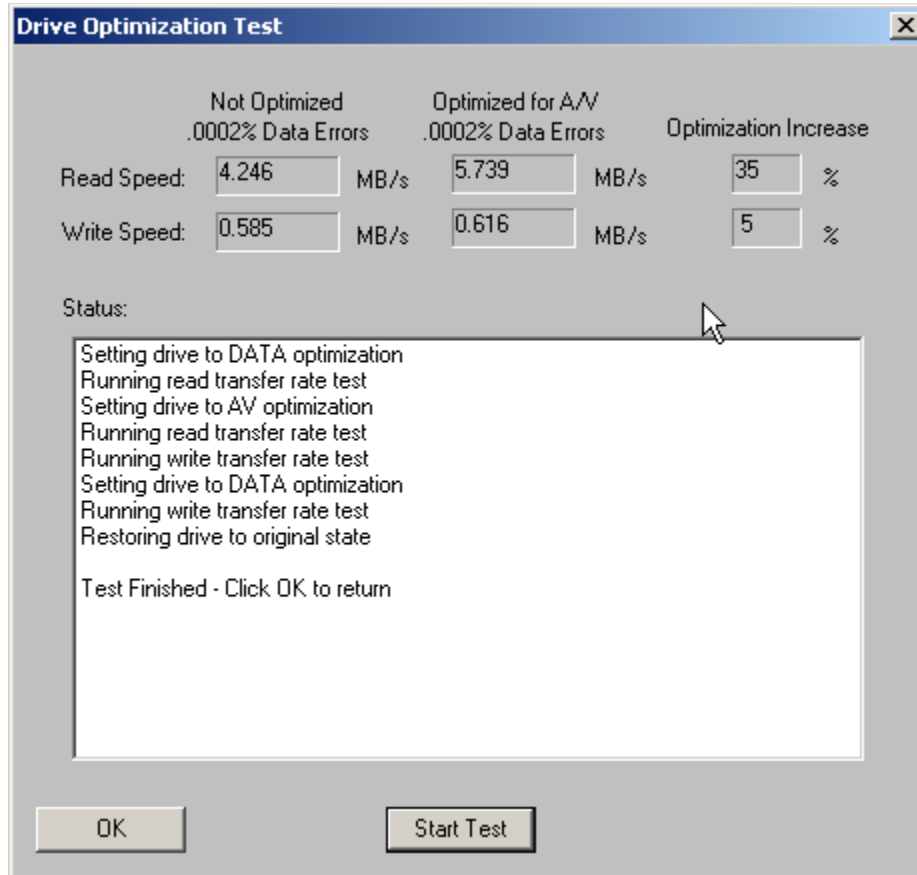
Click START TEST to test, or OK to return

OK Start Test

User Test Parameters:	
Button	Functionality
Start Test	Introduces correctable (ECC) errors onto the drive. Then the test will configure the drive for A/V optimization, then execute a write and a read test. Then the drive will be configured for Data use and the tests run again. Then the correctable errors will be removed from the drive. The differences in read and write performance will be displayed in the Drive Optimization Test Dialog Box (see the next page).

Drive Optimization Test Dialog

The Drive Optimization Test Dialog Box displays differences in read and write performance.



The dialog box, titled "Drive Optimization Test", displays performance metrics for "Not Optimized" and "Optimized for A/V" states. It includes input fields for Read Speed, Write Speed, and Optimization Increase, along with a status log and "OK" and "Start Test" buttons.

	Not Optimized		Optimized for A/V		Optimization Increase
	.0002% Data Errors		.0002% Data Errors		
Read Speed:	<input type="text" value="4.246"/>	MB/s	<input type="text" value="5.739"/>	MB/s	<input type="text" value="35"/> %
Write Speed:	<input type="text" value="0.585"/>	MB/s	<input type="text" value="0.616"/>	MB/s	<input type="text" value="5"/> %

Status:

```
Setting drive to DATA optimization
Running read transfer rate test
Setting drive to AV optimization
Running read transfer rate test
Running write transfer rate test
Setting drive to DATA optimization
Running write transfer rate test
Restoring drive to original state

Test Finished - Click OK to return
```

Buttons:

Drive Confidence Test 1

Disk Test Information

Test Name:

DiskStart Block: Number of passes: Starting Blocks Per Xfr:

Disk Ending Block: Percent to test: Ending Blocks Per Xfr:

Current Test Block = End of Test = Ignore Errors

Data Pattern: Log Errors

Information / Test Results

Enter data pattern and percent of disk to test then click Start Test

Information / Test Results

Enter data pattern and percent of disk to test then click Start Test

User Test Parameters:

Button	Functionality
Start Test	Checks drive functionality by writing data patterns to the disk, executing random seeks, and starting/stopping the drive.

This test checks drive functionality by writing data patterns to the disk, executing random seeks, and starting/stopping the drive. The number of test passes, the data pattern, and the percentage of the disk to be tested may be specified. The specified data pattern will be written to the percentage of the disk that was specified (starting from the first block of the disk). The data then will be read back and compared for integrity. Following this, 5,000 random disk seeks will be performed on the drive. The final step of this test will start and stop the disk five times.

Drive Confidence Test 2

Disk Test Information

Test Name:

DiskStart Block
 Number of passes :
 Starting Blocks Per Xfr

Disk Ending Block
 Percent to test:
 Ending Blocks Per Xfr

Current Test Block =
 End of Test =
 Ignore Errors

Log Errors

Data Pattern:

Information / Test Results

This test first sets the PER bit (Post Error, MODE SELECT page 1). With this bit set the drive will report any corrected errors to the host. Then the entire drive is written with two worst-case data patterns and the data is read back and compared. Finally, 10,000 random seeks with reads are executed. The results of this test are written to a disk file called "Fujitsu.log".

Seek Tests

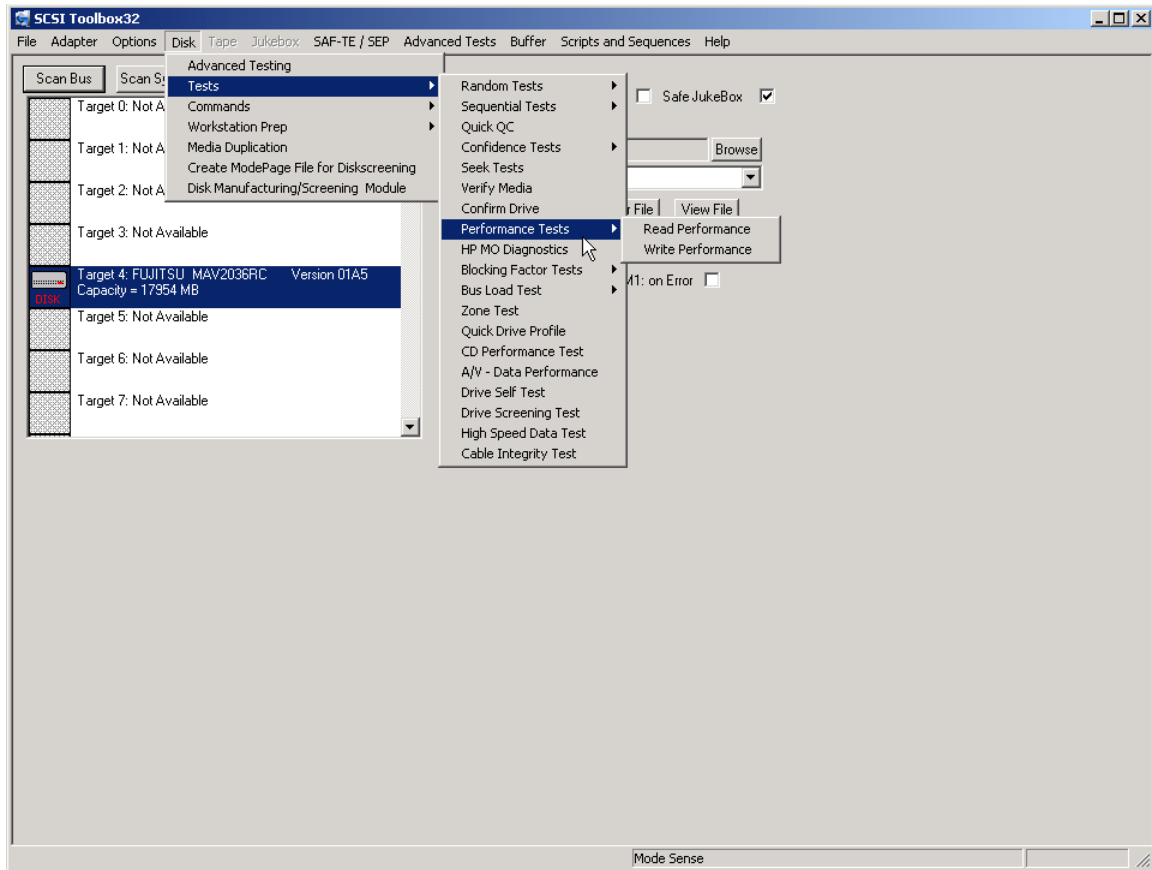
The screenshot shows a software window titled "Disk Seek Tests". At the top left, "Device Information:" is followed by a text box containing "Target 6 IBM DDYF-T18350S". Below this, a "Seek Test Type:" section contains five radio buttons: "Random Seek" (selected), "1/3 Stroke Seek", "Single-Track Seek", "Butterfly Seek", and "User Defined Seek". To the right of these are three input fields: "Block One" with "0", "Block Two" with "35566477", and "Number of seeks" with "1000". Below these is an "Average Seek Time:" field with "0". A large empty box contains the text "Choose test, then Click START TEST". On the right side, a "Select Devices" panel has a "Select All Targets" button and an empty list area. At the bottom are "Return" and "Start Test" buttons, with a mouse cursor pointing at "Start Test".

This test menu gives you access to several different seek test algorithms. Each seek test will first calculate the system overhead time for processing SCSI commands. Then it will perform 10,000 seeks of the type specified and calculate the total and average seek times. You may select one of the following seek tests:

- | | |
|-------------------|--|
| Random | Seeks to random blocks across the block range of the device |
| 1/3 Stroke Seek | Seeks from block 0 to a block 1/3 of the total block range of the device then back to block 0 |
| Single Track Seek | Seeks from a block 0 to a block calculated to be within the same track then back to block 0 |
| Butterfly | This test seeks between two initially specified blocks, subtracting one block from the high block and adding one block to the low block and seeking again until the high and low block are the same. |
| User Defined | You are allowed to specify two block numbers to seek back and forth between and the number of times to perform the seek. |

Performance Tests

Both Read and Write Performance tests are available to measure drive transfer rate



The Read Transfer Rate Test

This test reads a specified amount of data (default: 10 MB) from the selected device with or without incrementing the block number and reports the sustained transfer rate during the transfer.

Reading 10 Mbytes from Target 2 Host Adapter 0
(156 Reads of 128 Blocks using 512 byte blocks)
Elapsed Transfer Time = 31 seconds
Sustained Average Transfer Rate = 1651 KB/sec

The Write Transfer Rate Test

This test writes a specified amount of data (default: 10 MB) from the selected device with or without incrementing the block number and reports the sustained transfer rate during the transfer.

Writing 10 Mbytes from Target 2 Host Adapter 0
(156 Reads of 128 Blocks using 512 byte blocks)
Elapsed Transfer Time = 31 seconds
Sustained Average Transfer Rate = 1651 KB/sec

Disk Drive Performance Information

Read Transfer Rate Performance Test

MBytes to Transfer: Ignore Errors

Starting Block: Increment Blocks

Blocks per Transfer:

Test Results:

Elapsed Transfer Time = 2754 milliseconds

Sustained Average Transfer Rate = 18591 KB/sec

Disk Drive Performance Information

Write Transfer Rate Performance Test

MBytes to Transfer: Ignore Errors

Starting Block: Increment Blocks

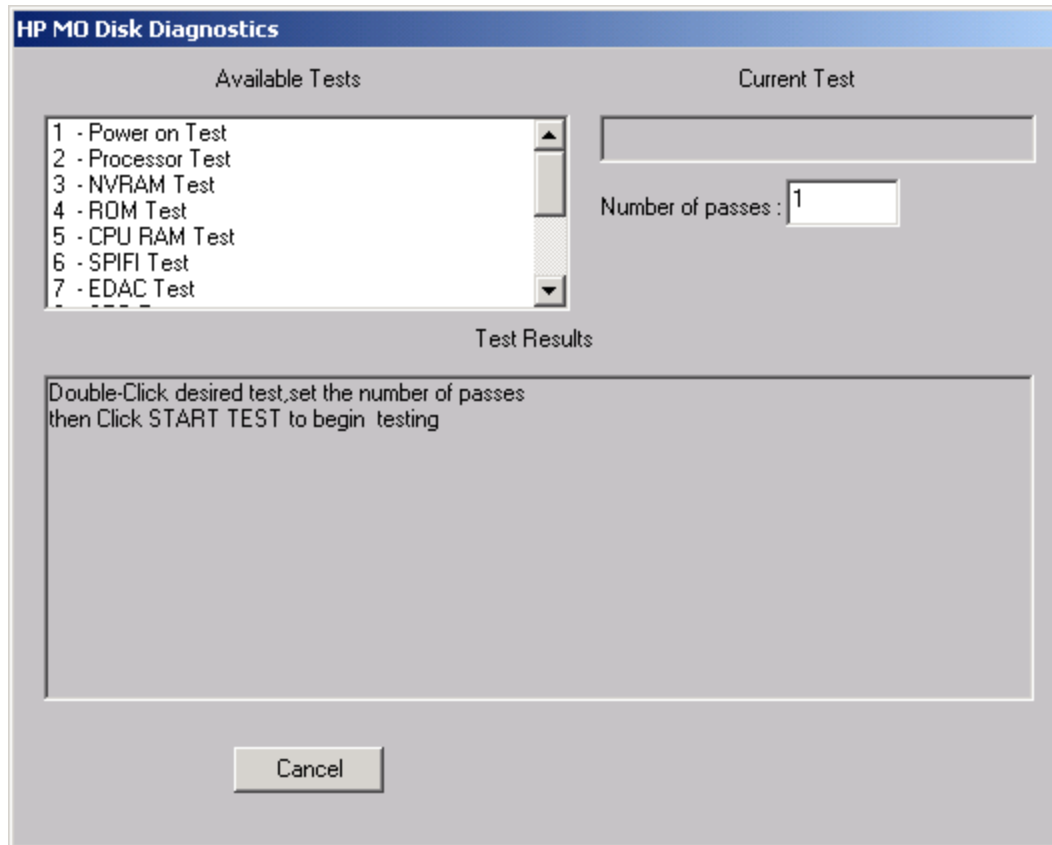
Blocks per Transfer:

Test Results:

Elapsed Transfer Time = 3.000 seconds

Sustained Average Transfer Rate = 17066.667 KB/sec

HP MO (Magneto Optical) Tests



This selection is similar to the tests and utilities in the dosdass2 program.

Blocking Factor Tests


Disk Test Information

Test Name:

DiskStart Block: Starting Blocks Per Xfr:

Disk Ending Block: Ending Blocks Per Xfr:

Current Test Block = End of Test =



Information / Test Results

```
Target 4 HITACHI DK32DJ-72FC
Read transfer rate using 8 block transfers = 5163.893 KB/sec
Read transfer rate using 16 block transfers = 9733.840 KB/sec
Read transfer rate using 32 block transfers = 17326.565 KB/sec
Read transfer rate using 64 block transfers = 29210.331 KB/sec
Read transfer rate using 128 block transfers = 44258.078 KB/sec
Read transfer rate using 256 block transfers = 46470.982 KB/sec
Read transfer rate using 512 block transfers = 48683.886 KB/sec
Read transfer rate using 1024 block transfers = 47210.768 KB/sec
Read transfer rate using 2048 block transfers = 47185.920 KB/sec
Test finished - Click Return to continue
```

This test performs a series of 10MB reads with varying blocking factors to profile the I/O rates of the drive for various transfer sizes. The test starts with a transfer size of 1 block and increments that transfer rate through: 2, 4, 8, 16, 32, 64, 128, 256, and 512 block transfers and times the 10MB I/O. The results are displayed in text form and as a bar graph.

Write Performance

This test is the same as the Read Performance except that write commands are issued.

Disk Test Information

Test Name

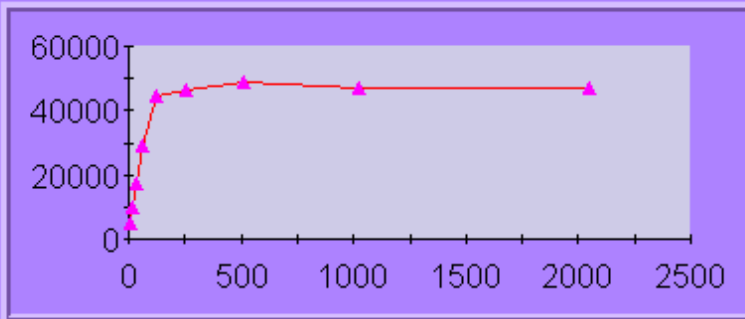
DiskStart Block Starting Blocks Per Xfr

Disk Ending Block Ending Blocks Per Xfr

Current Test Block = End of Test =



Information / Test Results



Bus Load Test

Disk Test Information

Test Name: Bus Write Transfer Rate Performance Test

DiskStart Block: 0

Disk Ending Block: 19531

Starting Blocks Per Xfr: 8

Ending Blocks Per Xfr: 128

Current Test Block = 16000

End of Test =

Ignore Errors

Select All Targets

- Target 4
- Target 5
- Target 6

Information / Test Results

Click START TEST to begin test
This test will cause slow system response

LUN 0

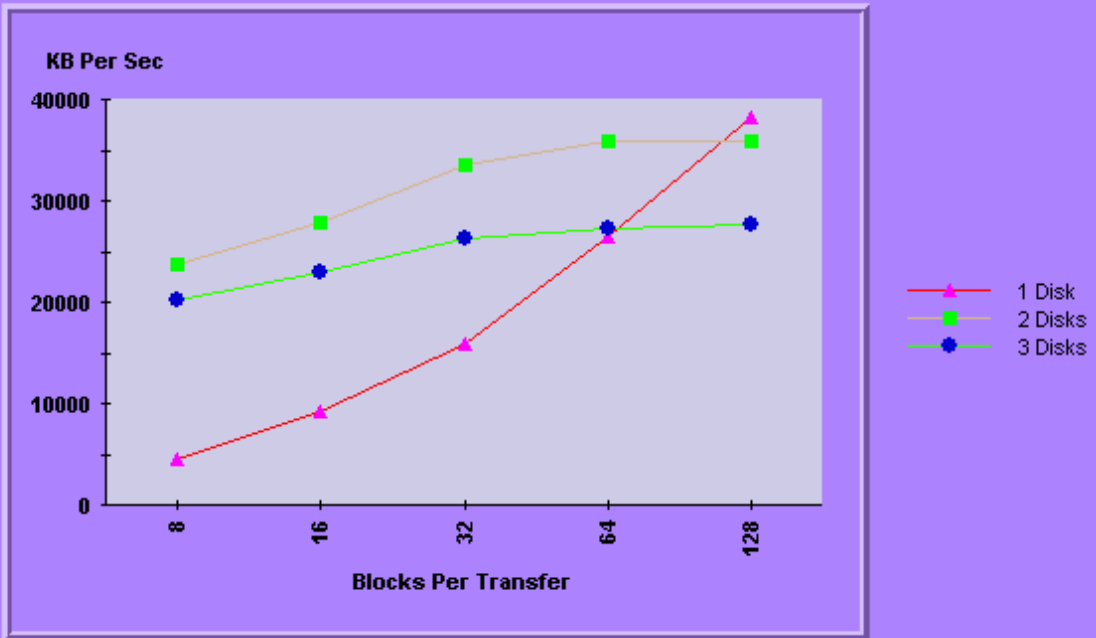
Select All LUNs

Return Start Test

The bus load test will read or write data to one or more disk drives, graphing the transfer rates. With multiple disk drives, the test is performed on a single drive first, then on two drives, and so on, until all drives selected have been run. The average transfer rate to individual drives are graphed for each set of tests. Each drive tested will be run on its own worker thread, utilizing the multi-tasking ability of Win32. The starting and ending blocks per transfer, starting and ending block are all user-definable. When the test is first started a file box is displayed for logging the numerical results of the test. At this point it cancel is clicked no logs will be saved.

Disk Test Information

Bus Load Test



Testing Complete

Return

Start Test

Text

Disk Test Information

Test Name

Select All Targets

DiskStart Block

Starting Blocks Per Xfr

Disk Ending Block

Ending Blocks Per Xfr

Current Test Block = End of Test =

Target 4
Target 5
Target 6

Information / Test Results

Write transfer rate on device 5, using 16 block transfers = 17372.068 KB/sec
Write transfer rate on device 6, using 16 block transfers = 13031.421 KB/sec
Total bus throughput = 68619.133 KB/sec
Average device throughput = 22873.045 KB/sec

Write transfer rate on device 4, using 32 block transfers = 38215.645 KB/sec
Write transfer rate on device 5, using 32 block transfers = 27397.873 KB/sec
Write transfer rate on device 6, using 32 block transfers = 13031.421 KB/sec
Total bus throughput = 78644.938 KB/sec
Average device throughput = 26214.979 KB/sec

Write transfer rate on device 4, using 64 block transfers = 38215.645 KB/sec
Write transfer rate on device 5, using 64 block transfers = 30653.377 KB/sec

LUN 0

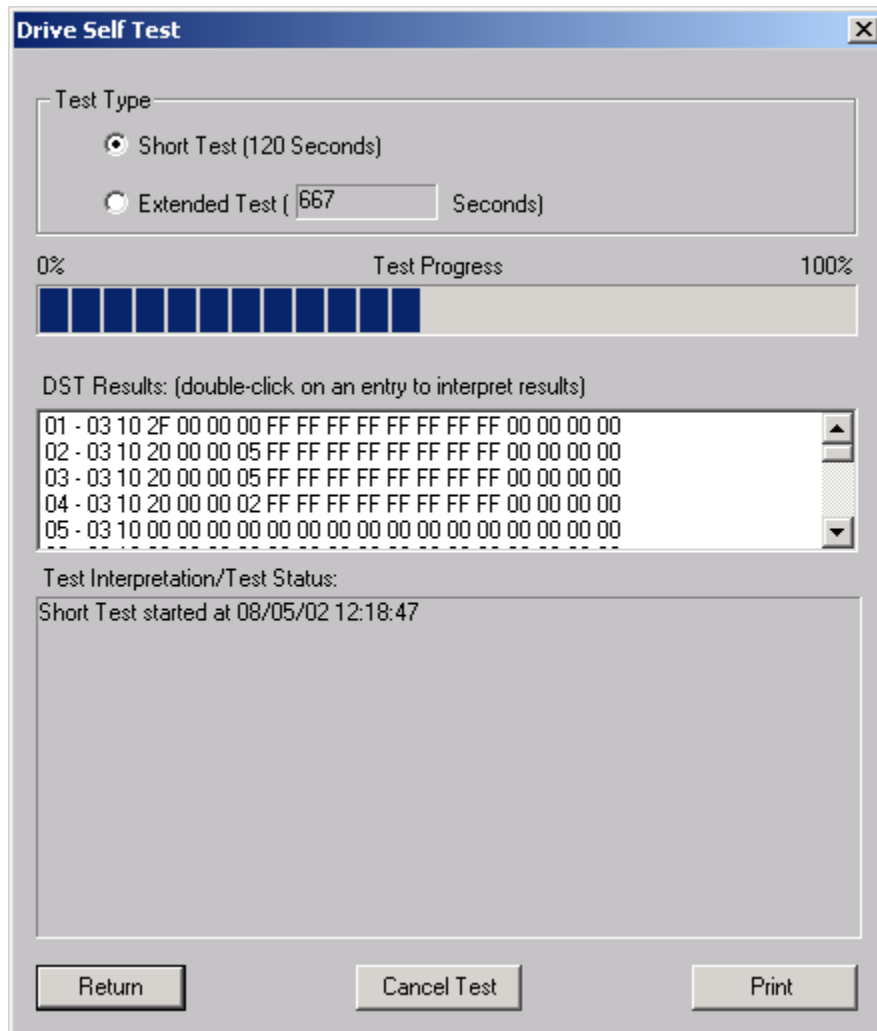
Return

Start Test

Graph

Select All LUNs

Drive Self Test



The Drive Self Test (DST) runs the drives built-in diagnostic tests in background mode. Once a self test is complete you can double-click on it's entry in the DST Results list box and the details of the results will be displayed in the Test Interpretation/Test Status window.


Drive Self Test [X]

Test Type

Short Test (120 Seconds)

Extended Test (Seconds)

0% Test Progress 100%



DST Results: (double-click on an entry to interpret results)

```
01 - 03 10 2F 00 00 00 FF FF FF FF FF FF FF FF 00 00 00 00
02 - 03 10 20 00 00 05 FF FF FF FF FF FF FF FF 00 00 00 00
03 - 03 10 20 00 00 05 FF FF FF FF FF FF FF FF 00 00 00 00
04 - 03 10 20 00 00 02 FF FF FF FF FF FF FF FF 00 00 00 00
05 - 03 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Test Interpretation/Test Status:

Entry Number = 2, Timestamp = 0
Self-Test Code = 001b "Background short self-test"

Self-Test Results = 0 "Self-test completed without error"

Sense Key = 00, Sense Code = 00, ASQ = 00
Vendor Specific = 00

Disk Zone Test

Zone	Starting Block	Ending Block	Current Block	Last Error	Total Errors	Test
Zone 0	0	4979009				<input checked="" type="checkbox"/> Test
Zone 1	4979010	9659259				<input checked="" type="checkbox"/> Test
Zone 2	9659260	13918869				<input checked="" type="checkbox"/> Test
Zone 3	13918870	17877253				<input checked="" type="checkbox"/> Test
Zone 4	17877254	21465733				<input checked="" type="checkbox"/> Test
Zone 5	21465734	24666833				<input checked="" type="checkbox"/> Test
Zone 6	24666834	27521282				<input checked="" type="checkbox"/> Test
Zone 7	27521283	30051322				<input checked="" type="checkbox"/> Test
Zone 8	30051323	32291686				<input checked="" type="checkbox"/> Test
Zone 9	32291687	34269718				<input checked="" type="checkbox"/> Test
Zone 10	34269719	35964300				<input checked="" type="checkbox"/> Test
Zone 11						<input type="checkbox"/> Test
Zone 12						<input type="checkbox"/> Test
Zone 13						<input type="checkbox"/> Test
Zone 14						<input type="checkbox"/> Test
Zone 15						<input type="checkbox"/> Test

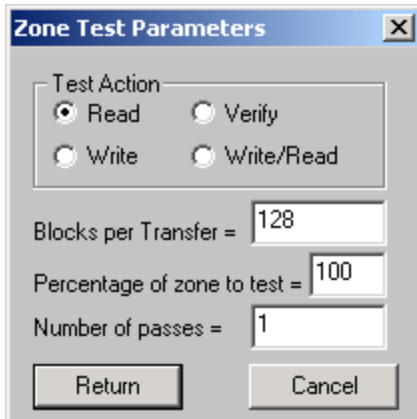
Percentage of zones to test = Total zones on drive = 11 Status =

Current Test Parameters =

Zone 0-15
Zone 16-31

The Zone Test determines the starting and ending blocks of each zone of a disk drive, and allows various types of tests to be run on any or all of the zones.

Click the Build Zone Map button to determine the zone boundaries of the drive. As each zone is processed the beginning and ending blocks of the zone are displayed. This process can take several minutes to complete.



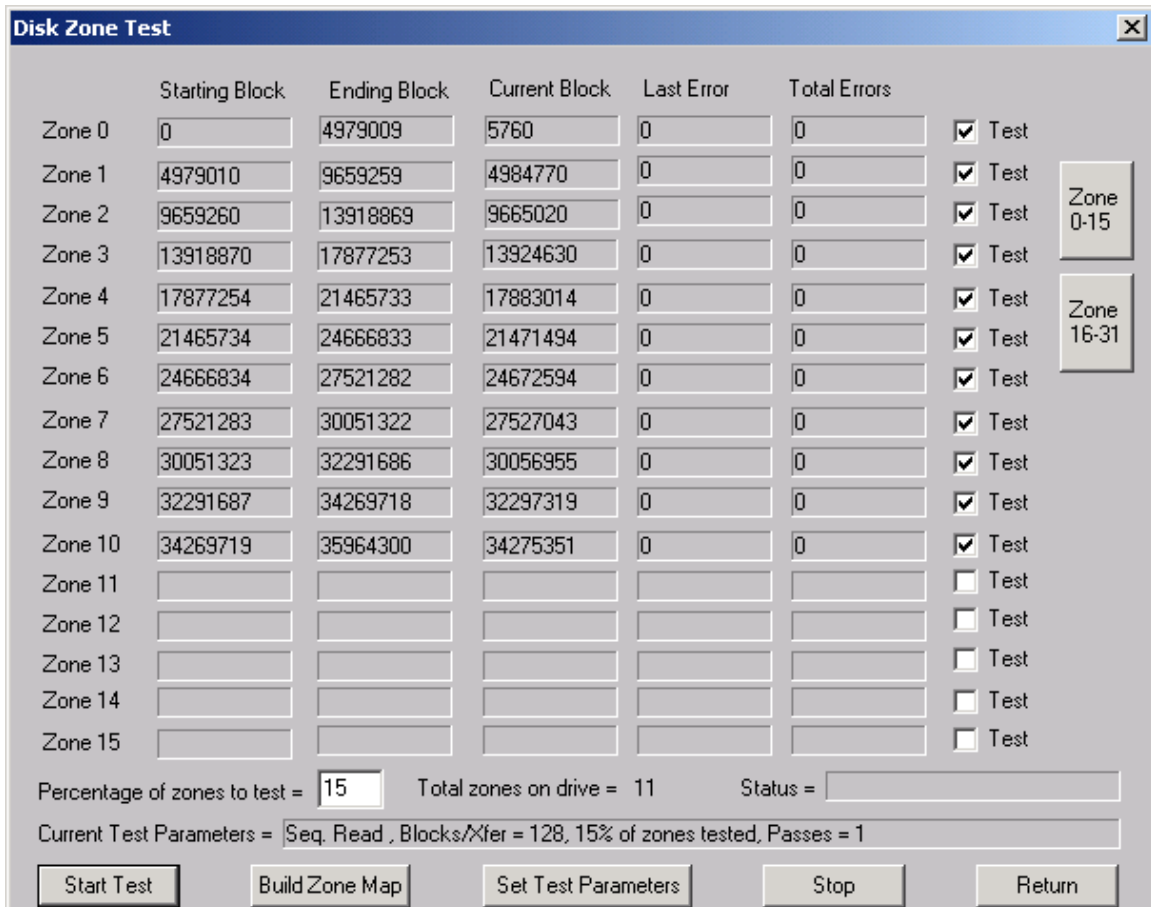
Set Zone Test Parameters – Clicking this button brings up the above dialog. You can select the following choices :

Test Action – Read Only, Write and Read, Write Only, or Verify

Blocks per Transfer – how many blocks are transferred for each SCSI command

Percentage of Zone to test – Specifies from 1 to 100% of each zone to test

Number of Passes – how many times the test is run



CD Performance Test

This test will check the read, seek, and start/stop functions. It will also measure drive performance.

Read Transfer Rates:	Speed	Average	High	Low	
Inner Tracks:	32X	5649.60	6241.58	2128.15	KB/s
Outer Tracks:	32X	5621.95	6228.00	343.64	KB/s

Stop Time : 18.254 sec.

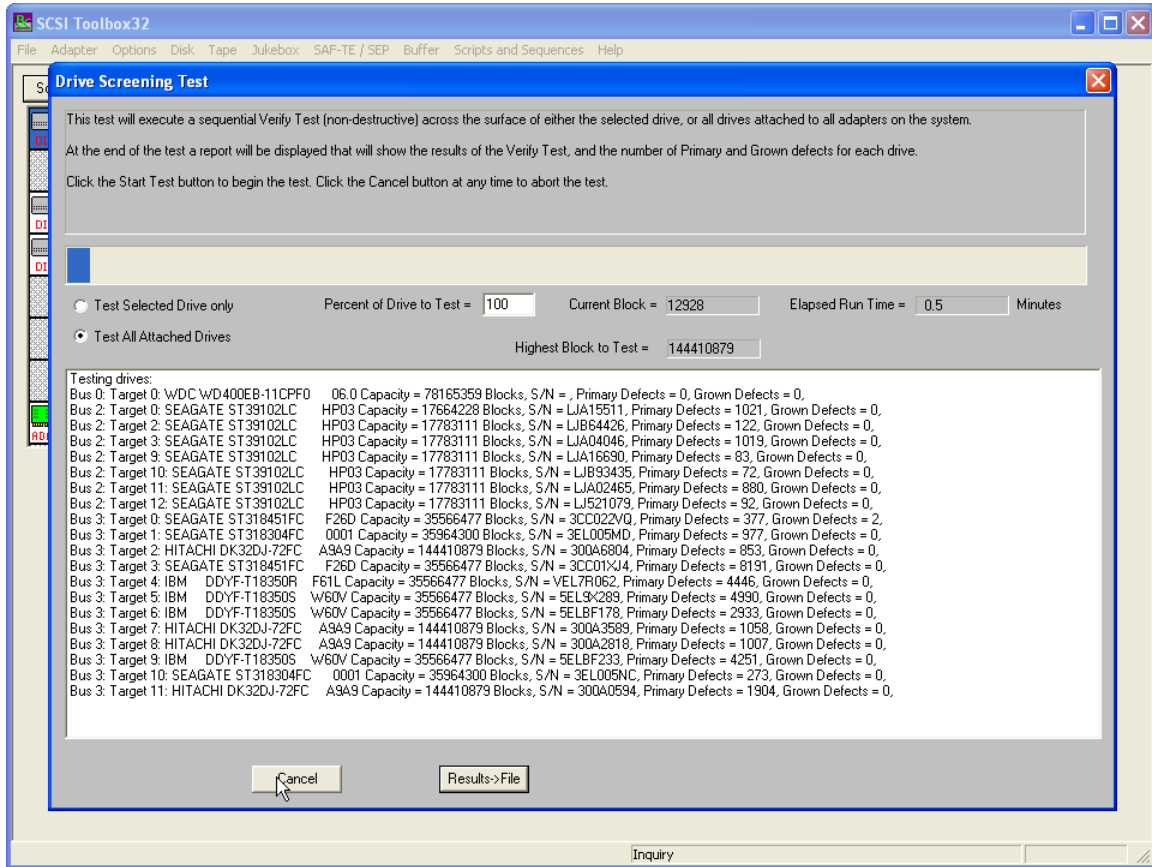
Start Time : 13.178 sec.

Status :
Testing Finished

OK Start Test

This test issues read commands to the inner and outer tracks of the device, and reports the time needed to start and stop the drive.

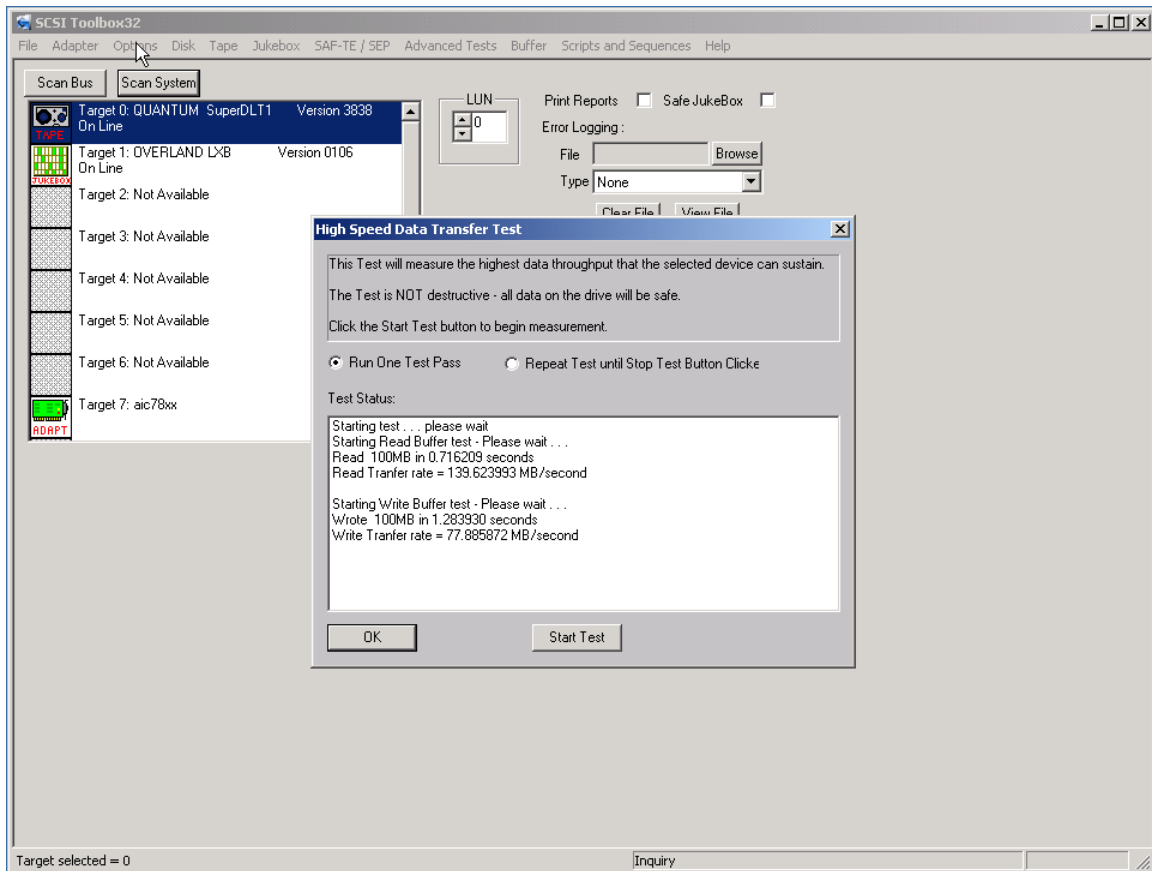
Drive Screen/Verify Test



This test will verify any percentage of either the currently selected disk or all disks attached to all host bus adapters.

This test is not threaded, so you may want to use the Disk Screening Module which is documented in a separate chapter at the end of this manual.

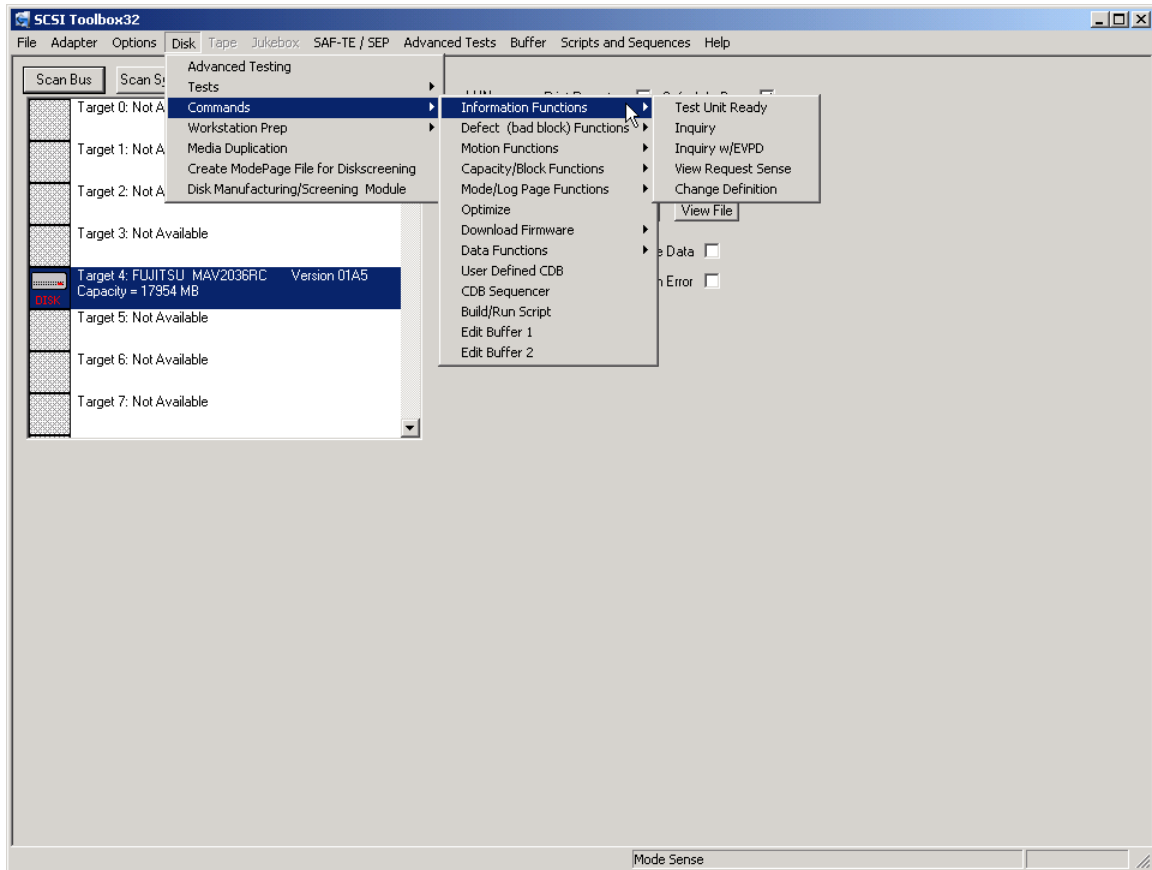
High Speed Data Test



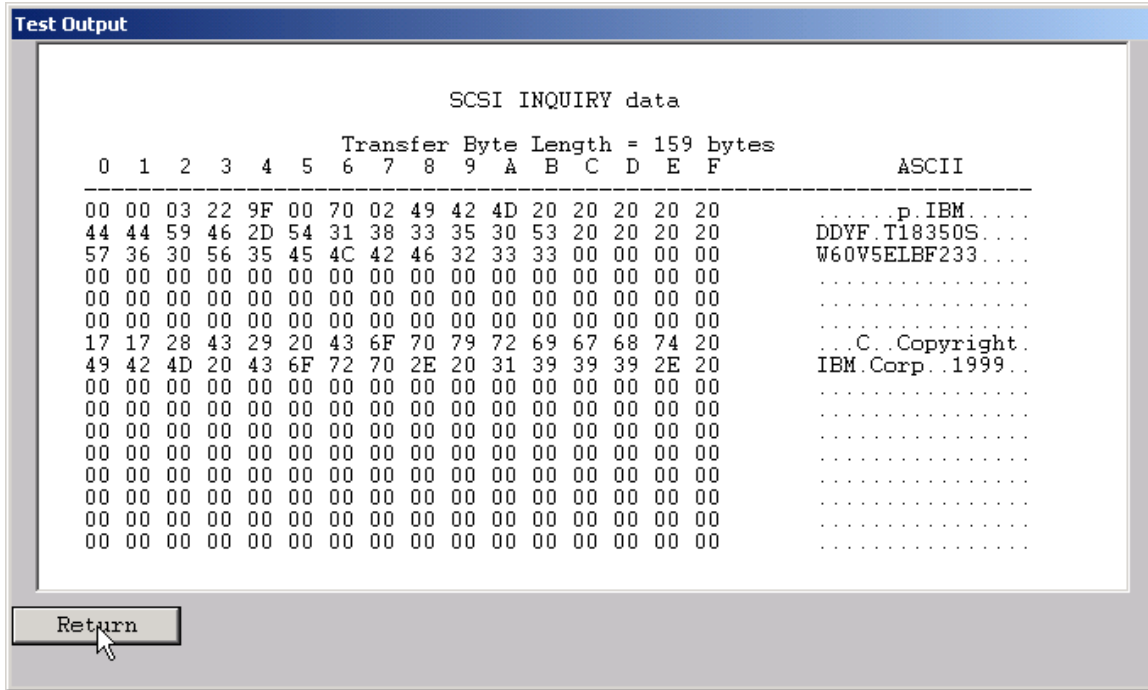
The High Speed Data test transfers data to and from the drive without transferring to the actual drive media. This allows the data transfer speed of the drive and host bus adapter to be determined.

Disk Commands

The following commands may be accessed from the Commands Menu within SCSItoolbox32.

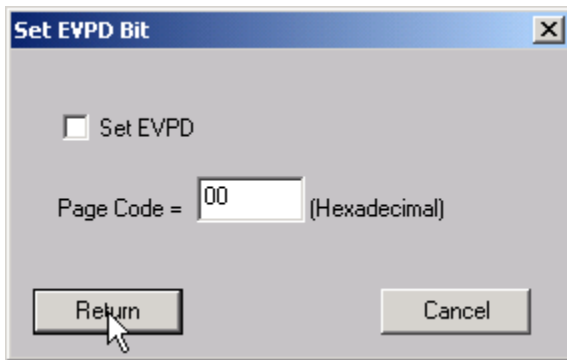


INQUIRY



INQUIRY data is displayed in hexadecimal on the left side of the screen, and in ASCII on the right

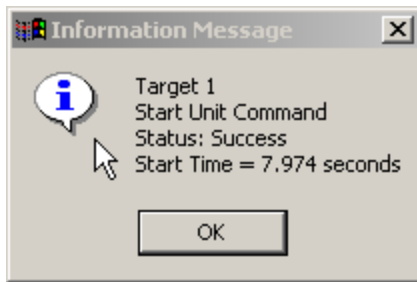
INQUIRY with EVPD



This option lets you specify whether to set the EVPD bit in the INQUIRY command, and which EVPD page you would like to display.

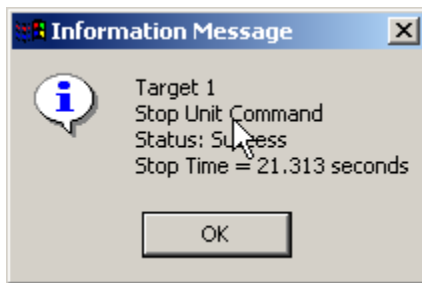
Start Drive (Motion Commands)

This command will start the disk drive and report the time needed for the drive to report as on line.



Stop Drive (Motion Commands)

This command will stop the drive and report the time taken to stop.



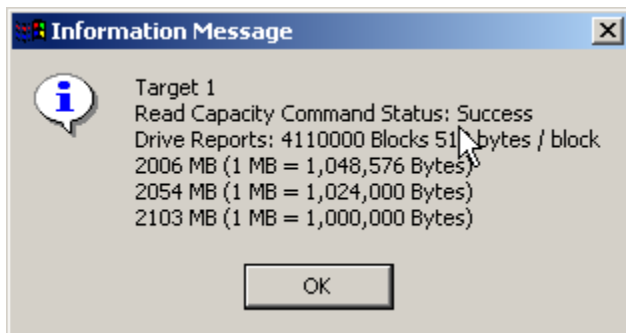
Start All (Motion Commands)

This command will start all drives on the selected HBA.

Stop All (Motion Commands)

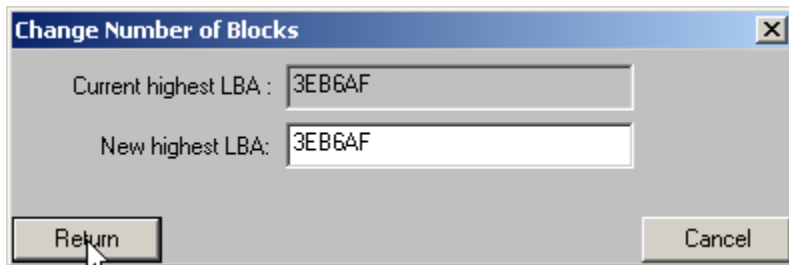
This command will stop all drives on the selected HBA.

Read Capacity (Capacity/Block Functions)



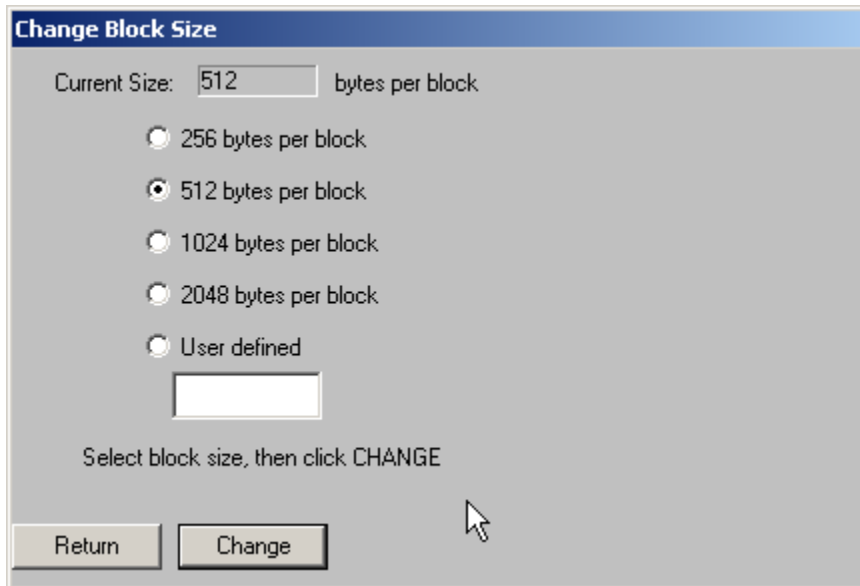
Change Capacity (Capacity/Block Functions)

This command allows you to change the capacity (highest LBA) of the drive. The LBA is specified in hexadecimal. This command will only change the capacity of the drive if the drive firmware allows changing capacity.

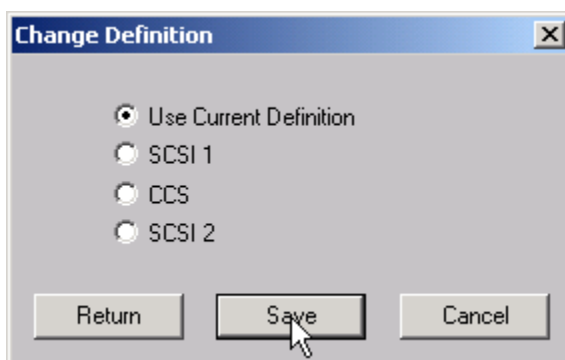


Change Blocksize (Capacity/Block Functions)

This command allows the blocksize of the drive to be changed (assuming that the drive firmware supports changing block size). THIS IS A DESTRUCTIVE OPERATION AND WILL REQUIRE REFORMATTING THE DRIVE!



Change Definition



View/Edit Mode Pages (Mode/Log Page Functions)

The following MODE PAGES can be viewed and edited:

Choose Mode Select Page

- 00 - Unit Attention Page
- 00 - IBM Vendor Unique Page
- 01 - Error Recovery
- 02 - Disconnect/Reconnect
- 04 - Ridged Disc Drive Parameters
- 07 - Verify Error Recovery
- 08 - Caching
- 0x0A - Control Mode
- 0x10 - XOR Control
- 0x14 - Enclosure Service Management
- 0x19 - Fibre Channel Interface Control
- 0x19 - SAS SSP Page
- 0x1C - IE Control
- 0x25 - Digital Special Function
- Hex Edit
- Display all available pages

Edit Page

Return

Select the MODE PAGE to view/edit, then click the Edit Page button. The example below shows the Caching Page:

Caching - Page Code 0x08

RCD # of segments

MF

WCE

Disable PreFetch Length - MSB	<input type="text" value="FF"/>	Maximum PreFetch - MSB	<input type="text" value="FF"/>
Disable PreFetch Length - LSB	<input type="text" value="FF"/>	Maximum PreFetch - LSB	<input type="text" value="FF"/>
Minimum PreFetch - MSB	<input type="text" value="00"/>	Maximum PreFetch Ceiling - MSB	<input type="text" value="FF"/>
Minimum PreFetch - LSB	<input type="text" value="00"/>	Maximum PreFetch Ceiling - LSB	<input type="text" value="FF"/>

Display Values

Current Saved

Default Changeable

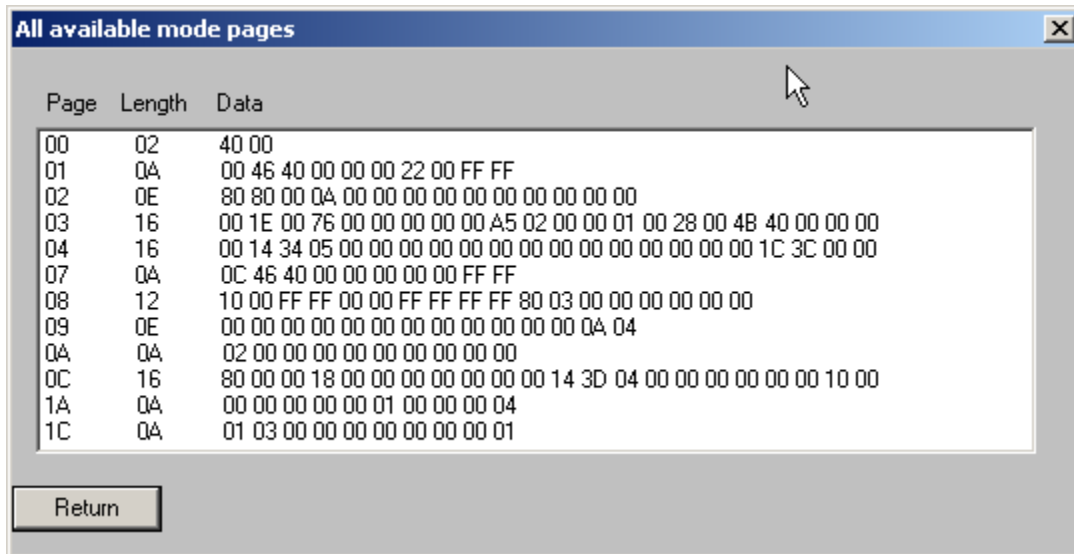
Any of the MODE PAGE parameters may be changed/edited. To save the page data (MODE SELECT) click the Save Page button.

The Restore Defaults button will restore the factory default values

The Display Values radio buttons allow you to view the current, default, saved, or changeable mask values.

View All Mode Pages (Mode/Log Page Functions)

Choose the Display all available pages choice of the Choose Mode Select Page dialog to display all of the MODE PAGES of a drive



The screenshot shows a window titled "All available mode pages" with a close button in the top right corner. The window contains a table with three columns: "Page", "Length", and "Data". The data is as follows:

Page	Length	Data
00	02	40 00
01	0A	00 46 40 00 00 00 22 00 FF FF
02	0E	80 80 00 0A 00 00 00 00 00 00 00 00 00
03	16	00 1E 00 76 00 00 00 00 00 A5 02 00 00 01 00 28 00 4B 40 00 00 00
04	16	00 14 34 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 1C 3C 00 00
07	0A	0C 46 40 00 00 00 00 00 FF FF
08	12	10 00 FF FF 00 00 FF FF FF FF 80 03 00 00 00 00 00 00
09	0E	00 00 00 00 00 00 00 00 00 00 00 00 0A 04
0A	0A	02 00 00 00 00 00 00 00 00 00
0C	16	80 00 00 18 00 00 00 00 00 00 00 00 14 3D 04 00 00 00 00 00 00 10 00
1A	0A	00 00 00 00 00 01 00 00 00 04
1C	0A	01 03 00 00 00 00 00 00 00 01

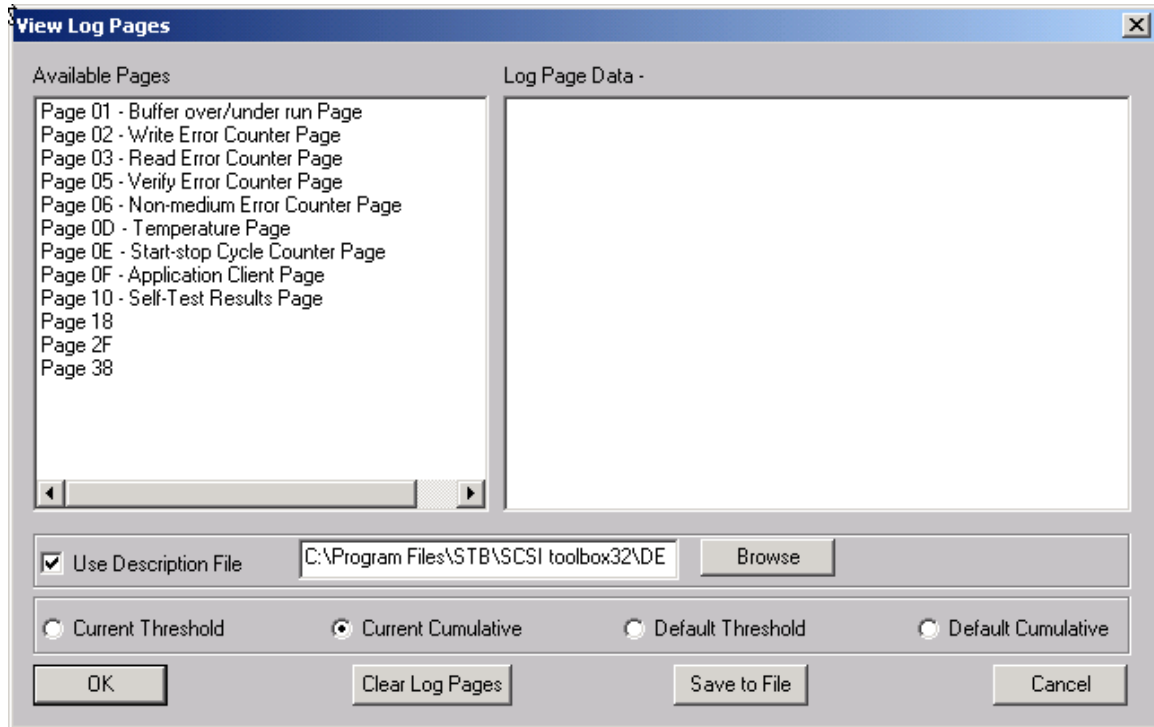
At the bottom left of the window, there is a button labeled "Return".

Restore Default Mode Pages

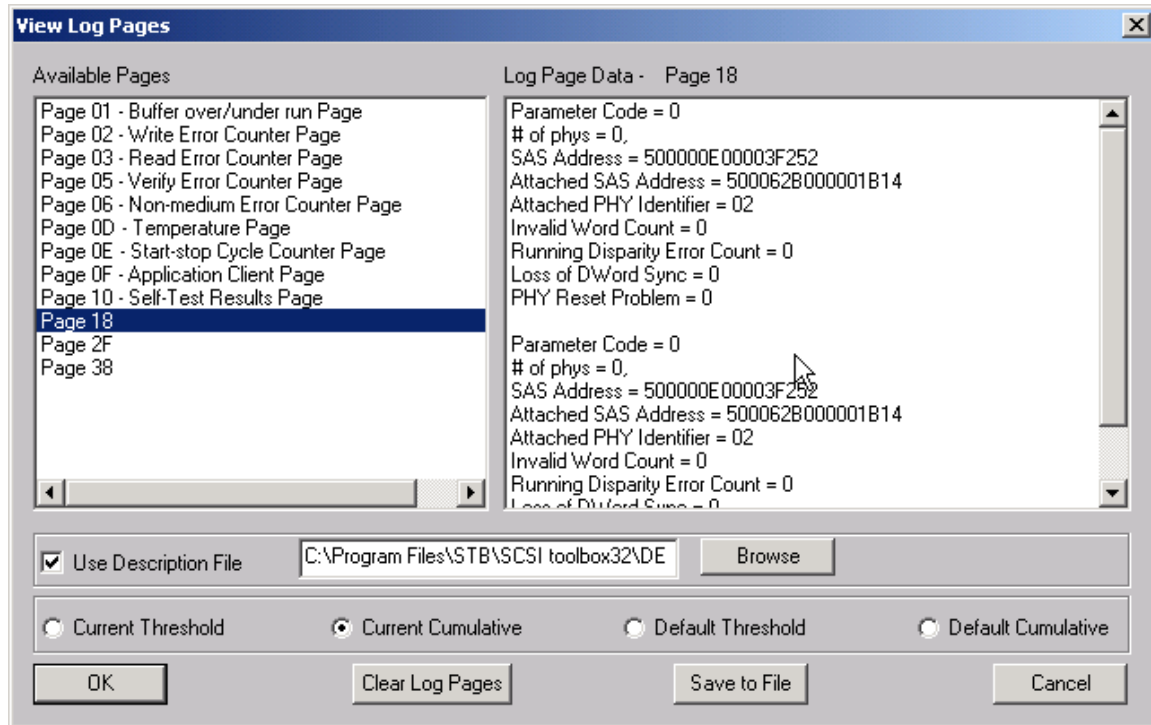
This command will restore all of the drives MODE PAGES to their factory default values.

View Log Pages (Mode/Log Page Functions)

This commands displays the LOG PAGE data from the drive. The raw (uninterrupted) data is displayed as below:

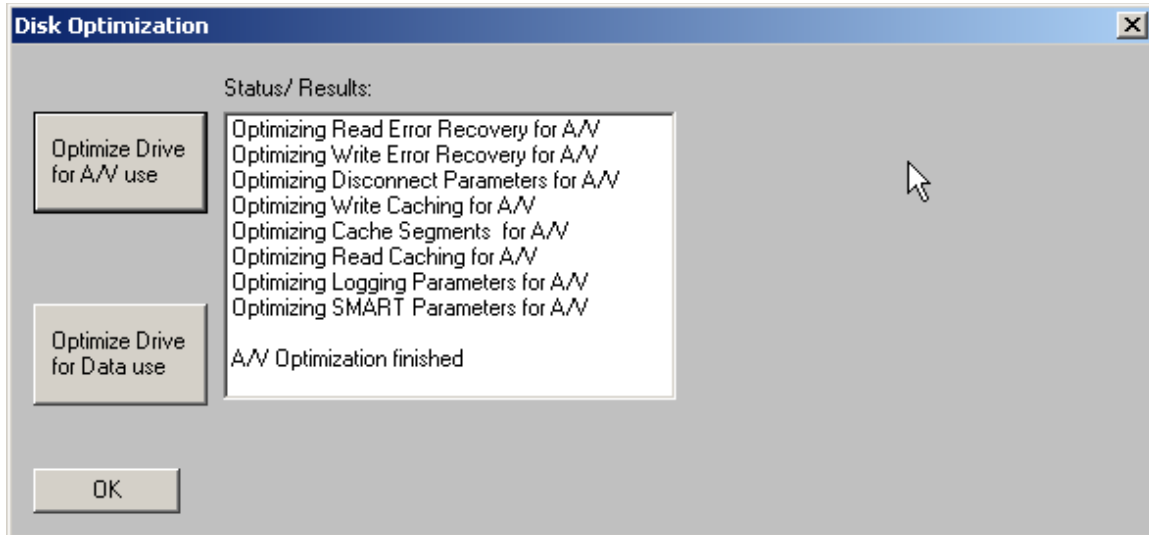


Clicking on the Use Descriptor File will use a text file (disk.def) to interpret the log data, as below:



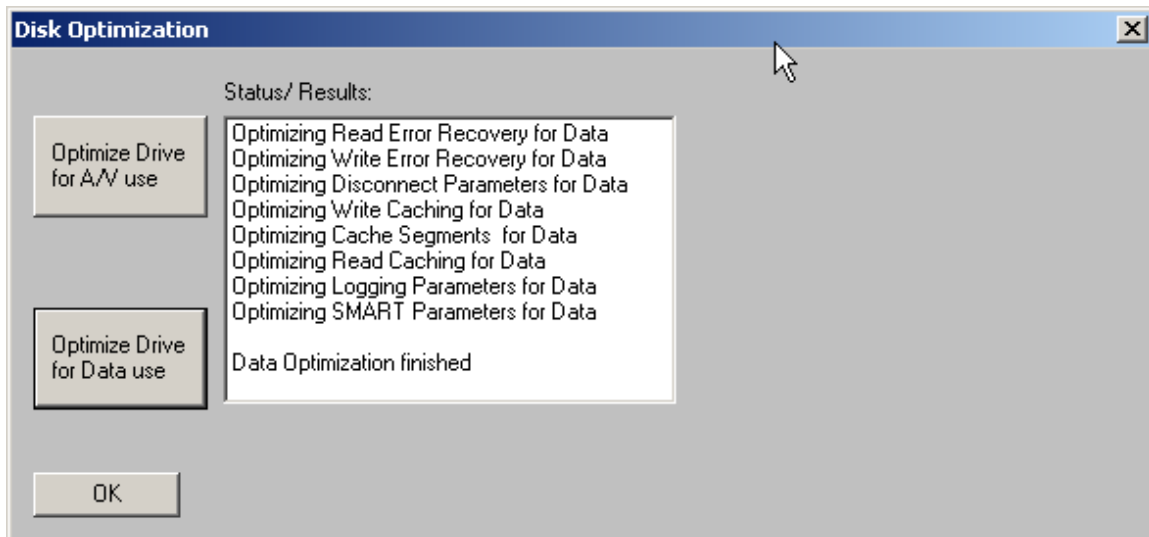
Optimize drive for A/V use

This command sets the specified MODE PAGE variables to optimize the drive for highest A/V performance.



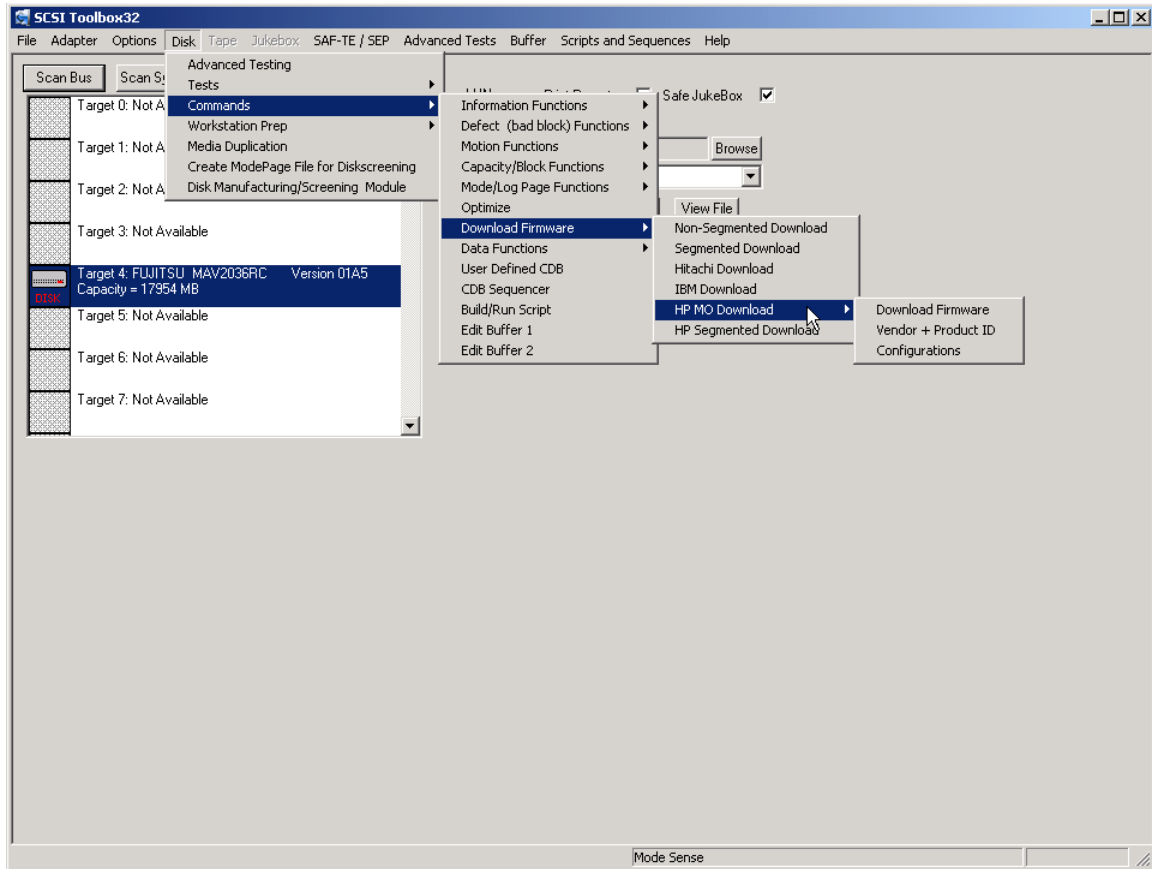
Optimize drive for Data use

This command sets the specified MODE PAGE variables to optimize the drive most reliable data operation.



Download firmware

Firmware can be downloaded in drives via five Download Firmware commands.



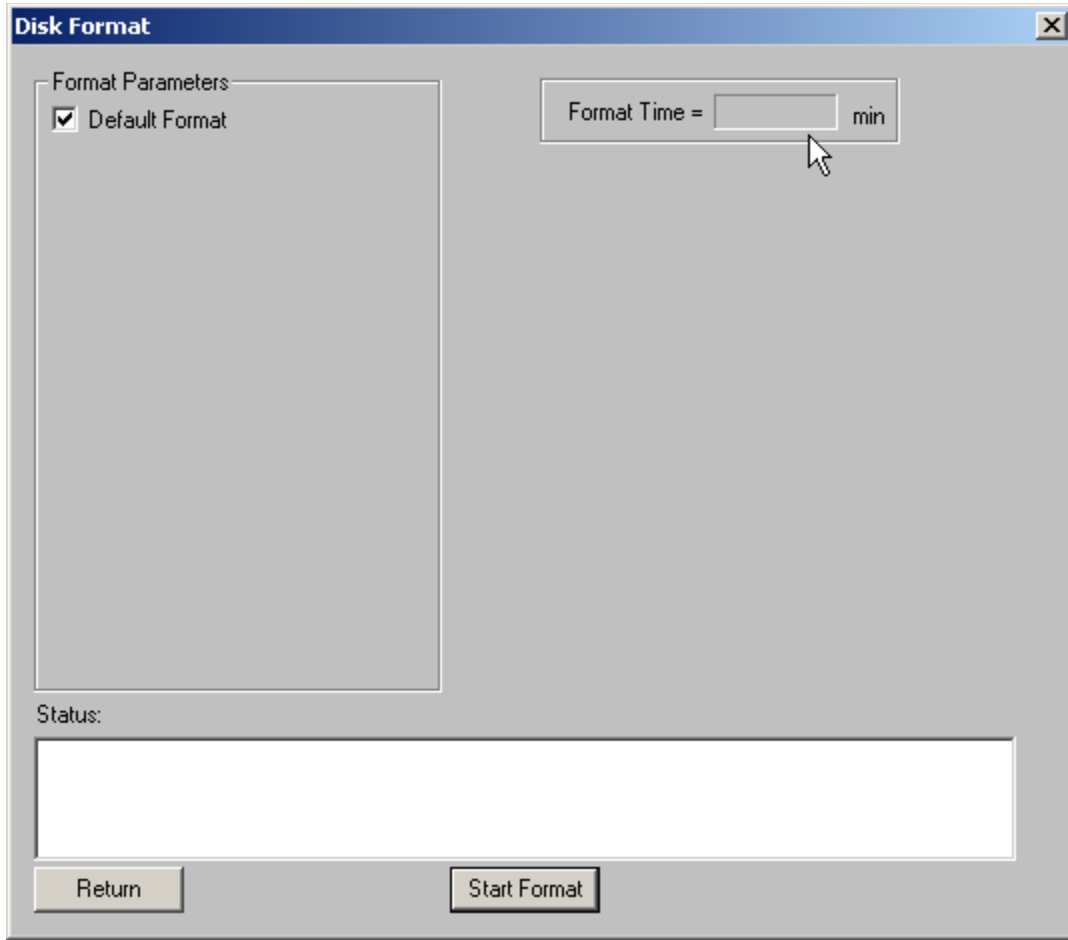
Use the Browse button to specify the firmware file name, then click the Download button.

Most disk drives support the Segmented Download choice. If the disk is a Hitachi or IBM drive use the proper choice.

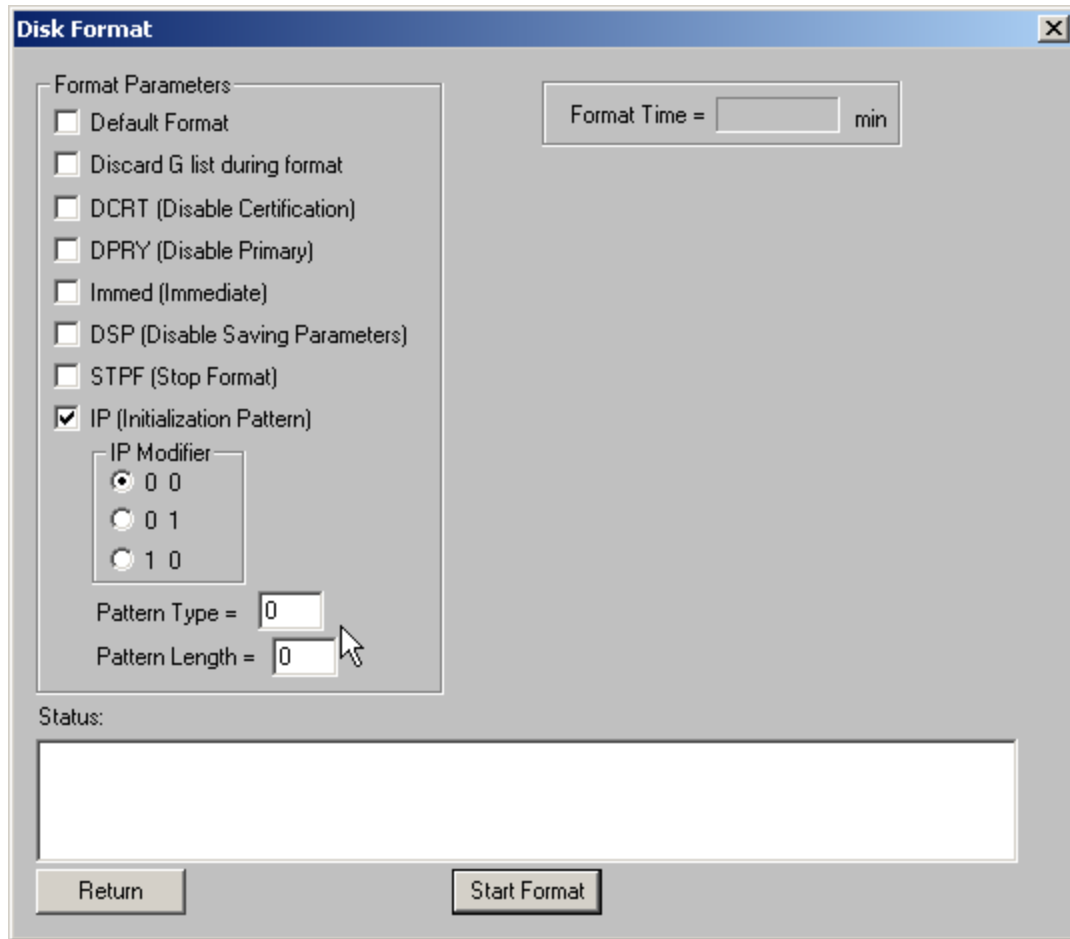
If the drive requires a non-segmented download use this choice.

Format

This command issues a “low-level” format command to the disk drive. The default format command will discard the Grown defect list.

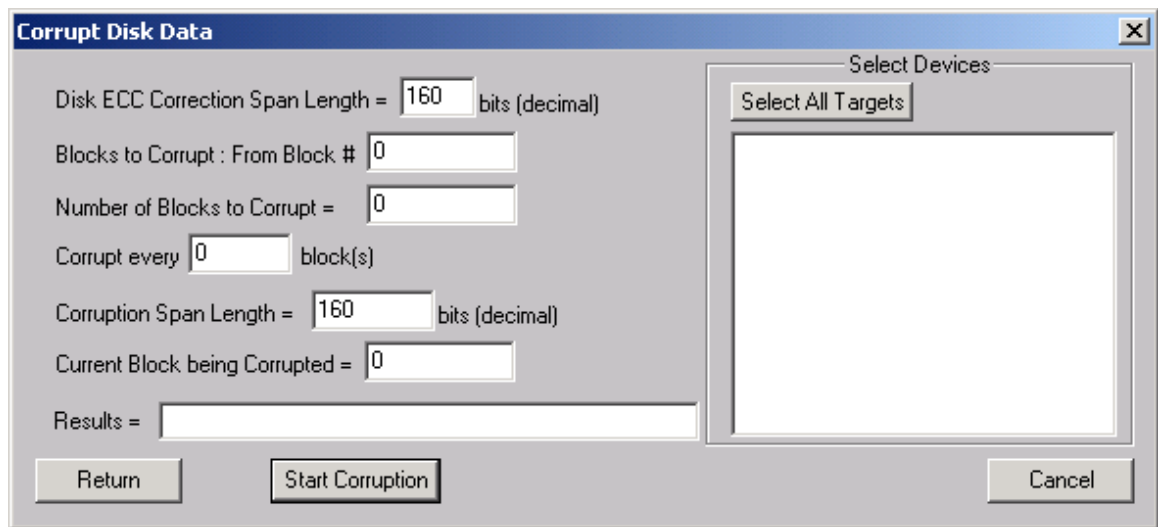


If the Default Format check box is “un-checked” the following format options can be specified:



Please note – not all disk drives support all format options.

Corrupt Data



The corrupt data choice allows you to create soft ECC errors in any block of the drive. Note – these errors can be corrected or removed by writing over the corrupted blocks. The Sequential Write test is used to do this.

Specify the following:

Blocks to Corrupt – From Block # - the starting block to corrupt

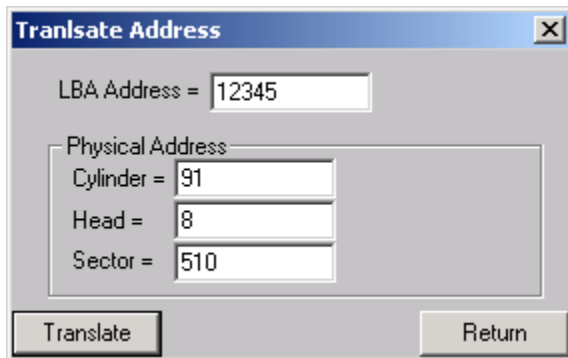
Number of Blocks to Corrupt – how many blocks should be corrupted? Must be ≥ 1

Corrupt every – allows you to skip blocks between corrupt blocks. Must be ≥ 1

Corruption Span Length – set this to $>$ the ECC span length to create unrecoverable errors, or $<$ the ECC span to create correctable errors.

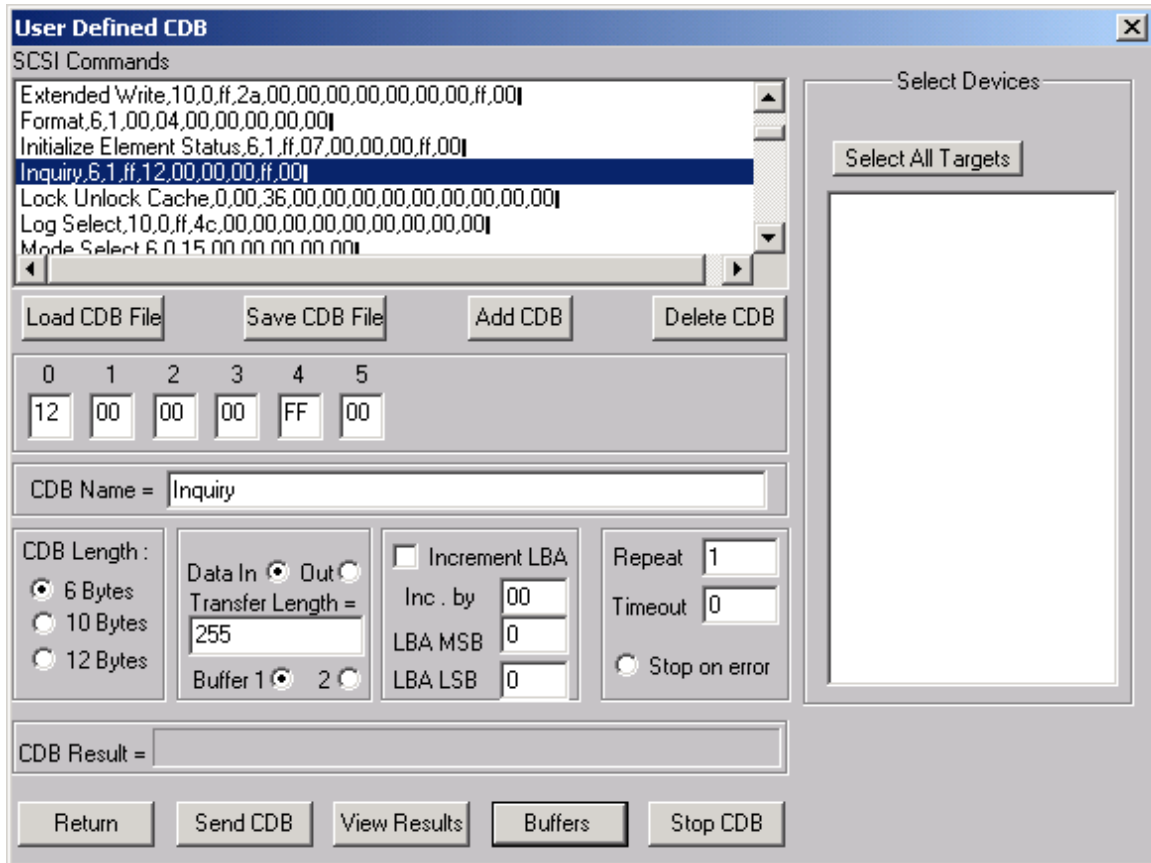
Once you have filled in the above selections, click on the Start Corruption button to begin.

Translate Address



Enter a LBA address, click the Translate button and the CHS address will be displayed

User Defined CDB



The User-Defined CDB functions of the SCSI toolbox allow the operator to construct an type of SCSI CDB and issue that CDB to the currently select SCSI TARGET. Any command created may also be added into a CDB file. Customized files of CDB's can be created and maintained.

Defining or Modifying a SCSI CDB

The following parameters define a SCSI CDB and are accessible/modifiable from the left side of the dialog box:

- Command name
- Individual bytes of the SCSI CDB
- Command length (6,10,12 bytes)
- Data Direction
- Data transfer length
- Data buffer (buffer 1 or 2)
- Command timeout value
- Command repeat count
- Increment information

Once a SCSI CDB has been defined that command can be issued to the currently selected TARGET, stored to the current CDB list file, or added to the script window.

Important Issues

It is possible to create an incorrect SCSI CDB! If you are not intending to create an illegal CDB, some common mistakes to watch out for are:

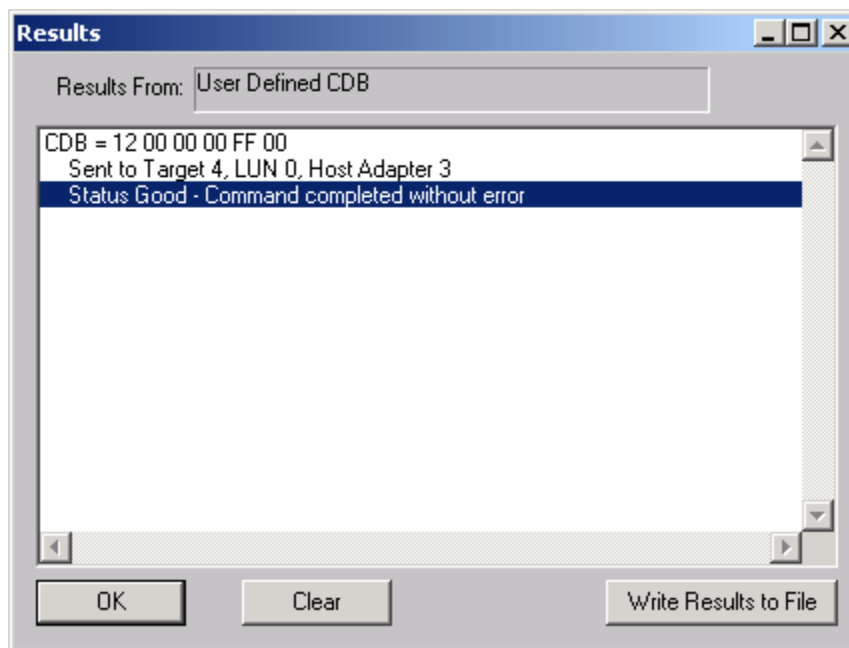
- Data direction incorrect
- Data length specified in CDB does not match Data Transfer Length

Incrementing Issues

It is possible to increment a field within the CDB each time the CDB is issued. This is commonly used to increment the block number specified in the CDB. Set the Increment portion of the dialog as follows to use the feature:

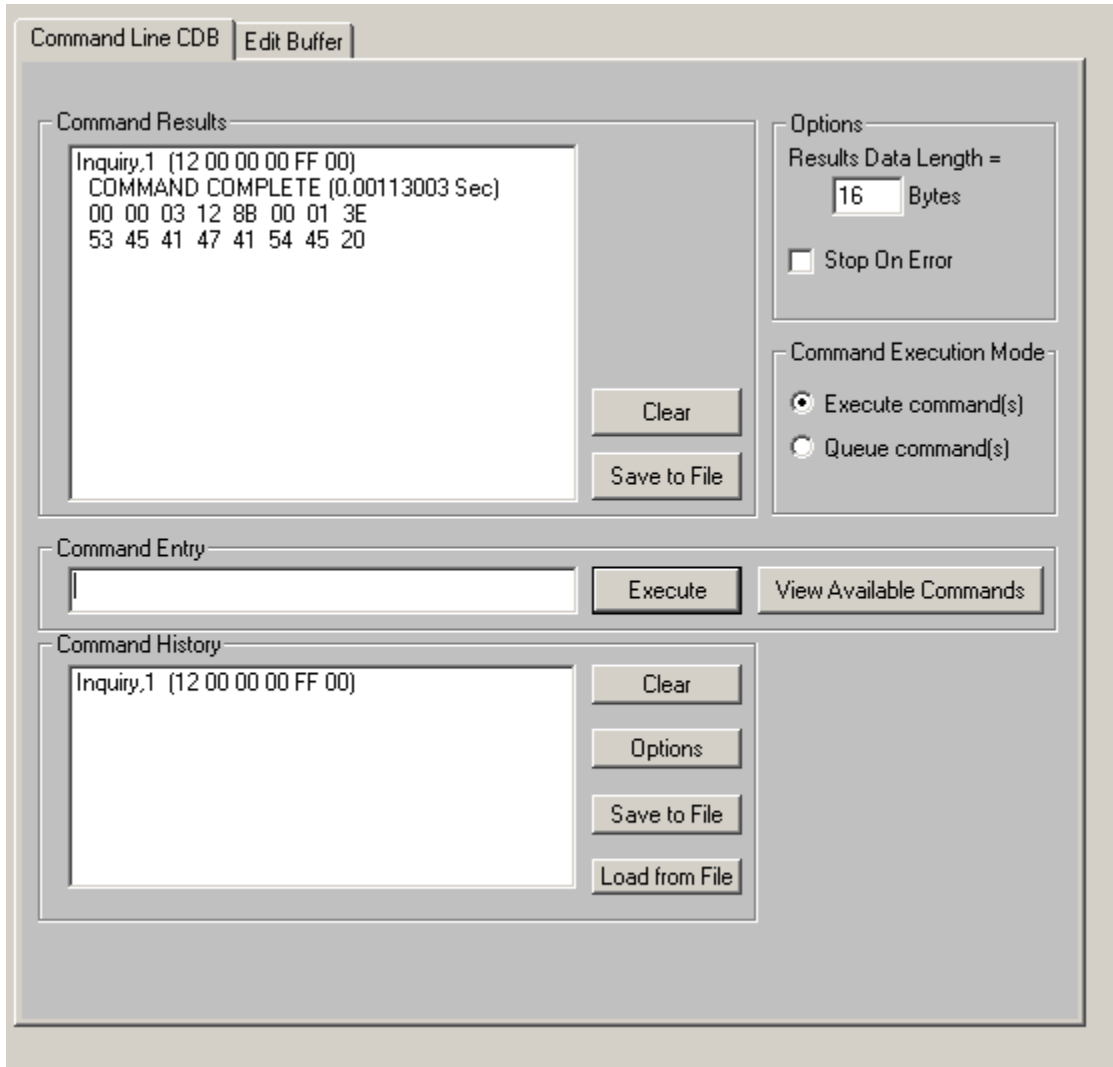
- Check the Increment LBA box
- Specify which CDB byte is the LSB of the SCSI CDB
- Specify which CDB byte is the MSB
- Specify how much you want to increment by

For instance, if you define a 10 byte READ EXTENDED CDB, byte 2 is the LSB and byte 5 is MSB of the Logical Block Address. In this example we will start reading a block 0 by setting CDB bytes 2 through 5 to zero (0). If you define this CDB to read 128 blocks per command (0x80 hex in byte 8 of the CDB) you will set the "increment by" field to 128. By setting the "CDB Repeat Count" field to 100, then clicking the "Send CDB" button, the READ EXTENDED CDB will be issued 100 times, and each time it is issued, 128 blocks of data will be read and the LBA will be incremented by 128.



After issuing a User-defined CDB you can click on the Display Results button to view the status of each CDB that has been sent. You can clear this display with the Clear button, or you can save the contents of the display to a disk file.

CDB Sequencer



This function lets you type in the name of a CDB on the command line. When you press *Enter*, the function looks for a matching CDB name in the COMMANDS.DAT file, and issues the CDB to the selected target.

For example, typing "INQ", followed by pressing *Enter*, matches with the INQUIRY CDB in the sample COMMANDS.DAT file and issues that CDB. Note that the first match found will be issued. For example, if you had two read commands defined, Read6 and Read10, simply typing "read" will match Read10, since that CDB is first in the file. The file is sorted alphabetically.

If a match is not found in the CDB file, an error message is returned. If you are using multiple CDB files, you must first rename (or copy) the desired file to COMMANDS.DAT.

You can use the "View Available Commands" button to display a list of all commands in the commands.dat file. This display can also be used to select a command from the list for execution.

Command Results Window

When the CDB is issued, the status of the command is shown in the top list box (Command Results).

This status shows:

- the bytes of the CDB that were sent
- the time spent executing the CDB,
- several bytes of data, depending on the Results Data Length setting
- whether the command completed successfully, or
- if the result is a check condition, the sense data is shown.

The contents of the Command results window can be erased or saved to a file.

Command History Window

The CDB issued is also recorded in the lower “Command History” window.

Commands in this window can be selected and re-issued using the “Execute” button.

Commands in this window can also be saved and/or reloaded from a file to facilitate reusing CDB sequences.

Options

The “Options” buttons allow choosing to execute commands upon pressing *Enter*, or Queuing the commands in the Command History window for batch execution.

This allows a series of CDB’s to be set up, then sent sequentially with no delays.

The “Stop on Error” checkbox will stop a multi-pass CDB on error if checked. If not checked, a multi-pass CDB will execute all passes, regardless of any error conditions.

Command line arguments

Command line arguments are available to specify the number of times to issue the cdb, and to change the values in bytes 1 and 2 of the cdb.

Multi-pass CDB’s are issued by typing in the CDB name, a comma, then the number of passes.

For example, “READ,1000” will issue 1000 reads.

To issue a READ EXTENDED cdb and set the FUA bit, you would enter READ_10,1, 80

These arguments tell the program to:

- issue the READ_10 cdb
- issue the cdb 1 time
- modify byte 1 to the hex value 0x80 (this sets the FUA bit in the cdb)

Continuous command execution

Entering a pass count of less than 0 will cause the cdb to be issued continuously, until the “Stop” button is clicked.

Results Data Length

The “Results Data Length” allows you to specify how many bytes of data are displayed in the “Command Results” window. If a command transfers data, this many bytes will be displayed along with the command results. If the command terminates in a check condition, this many bytes of sense data will be displayed.

The entire data buffer can be viewed, modified, and saved/loaded from a file in the ***Edit Buffer*** tab page.

Re-issuing commands

Any command or commands can be re-issued by selecting them (clicking on them) in the “Command History” window. Once the commands you want to reissue are highlighted, click the “Execute” button and they will be reissued.

Build Run Script

SCSI Script

CDB Functions

STB Test Functions

Target/LUN/Adapter Functions

Script Control Functions

Mode Page Functions

Buffer Functions

Edit Item

Delete Item from Script

Repeat Script times

Single Step -

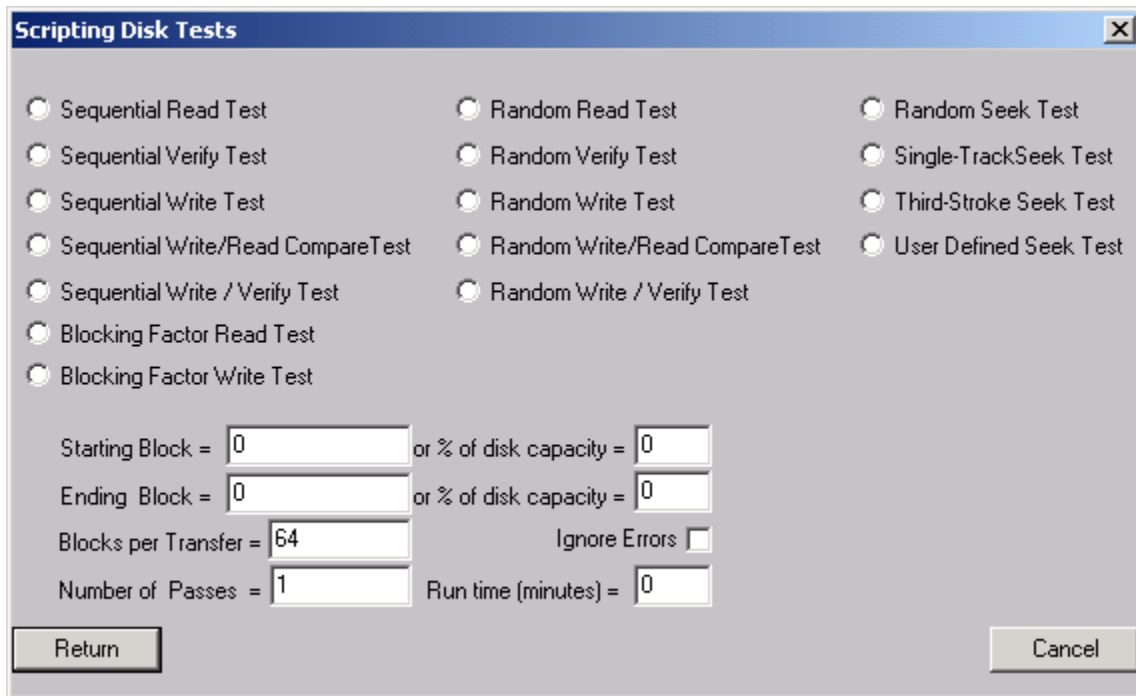
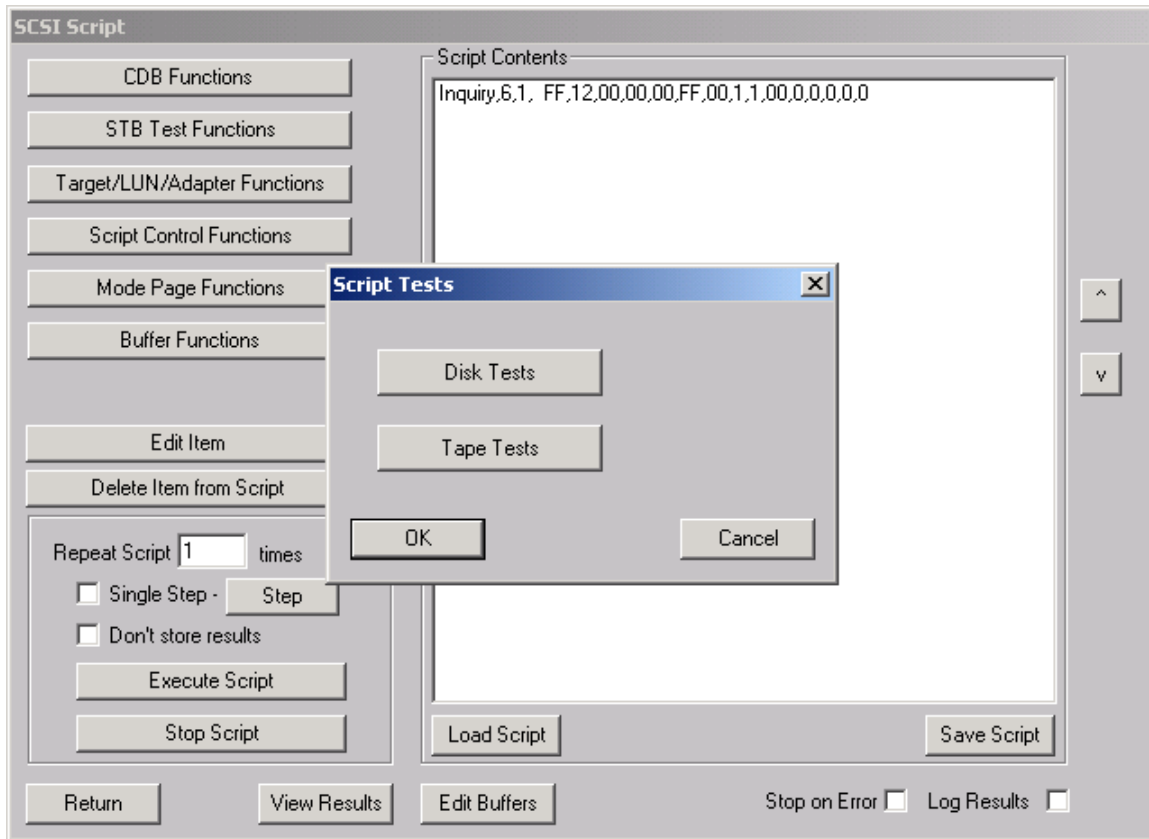
Don't store results

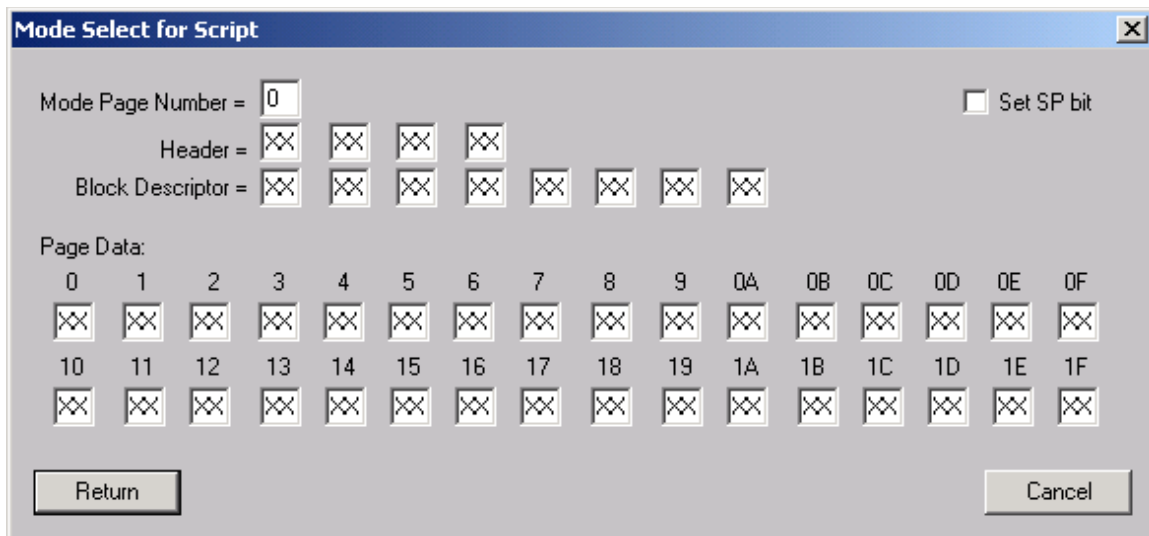
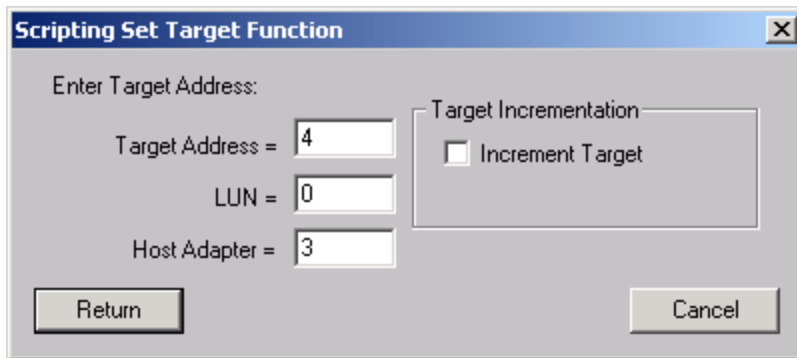
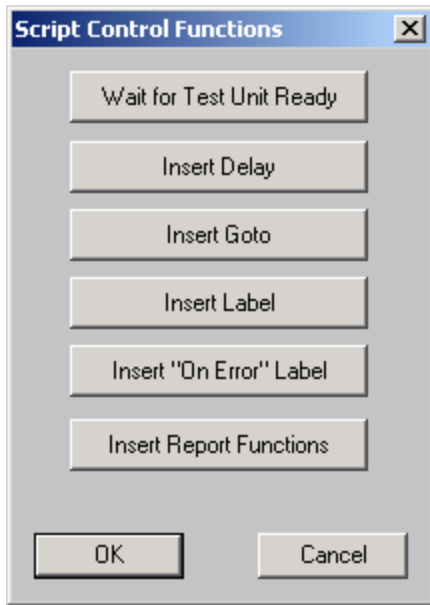
Script Contents

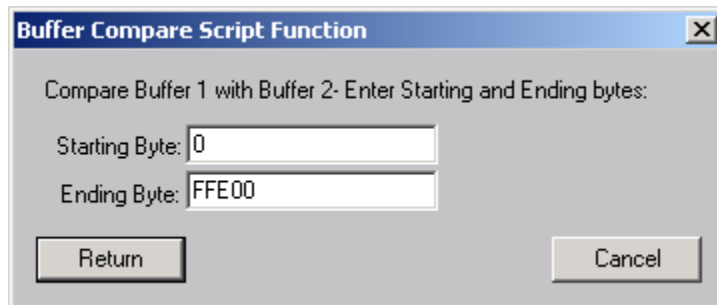
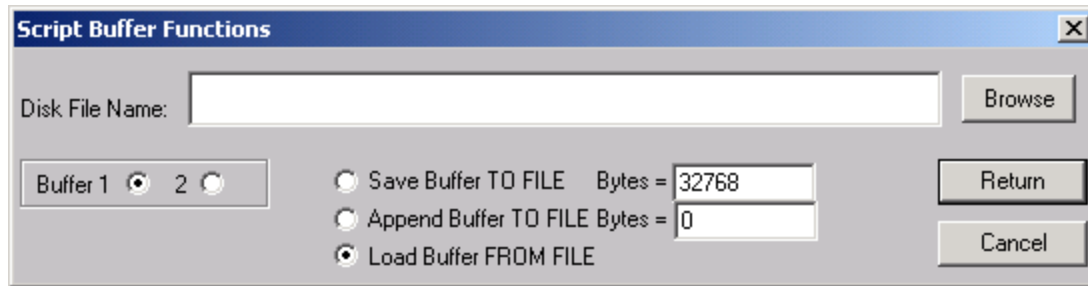
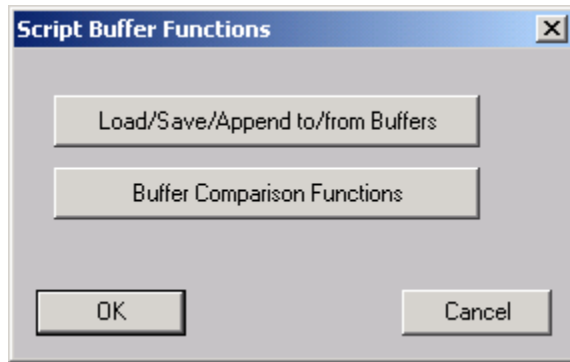
^

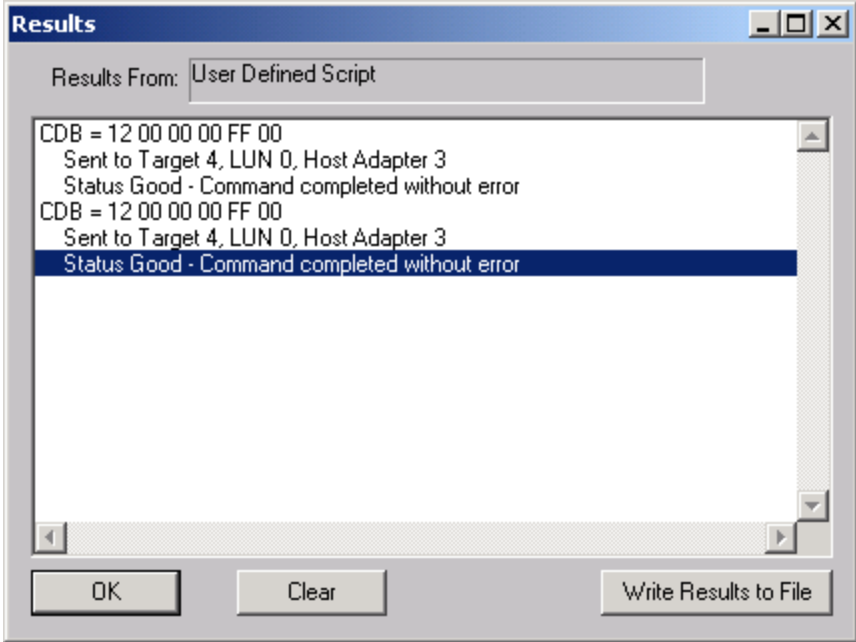
v

Stop on Error Log Results



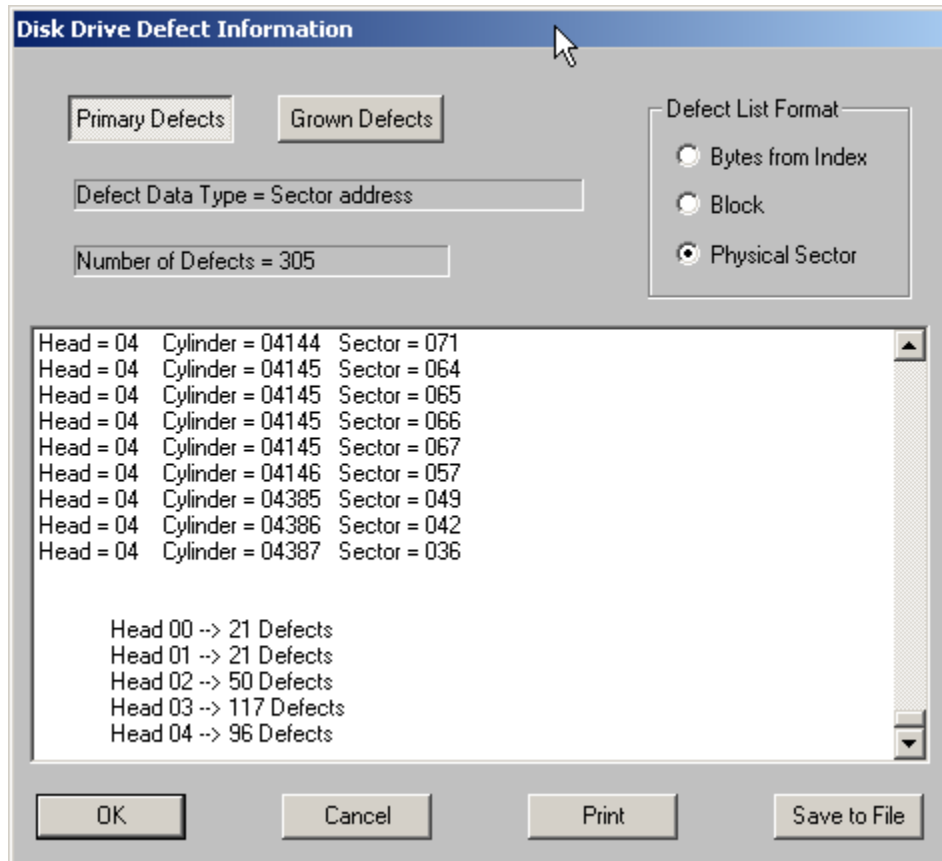






Display Defect Data

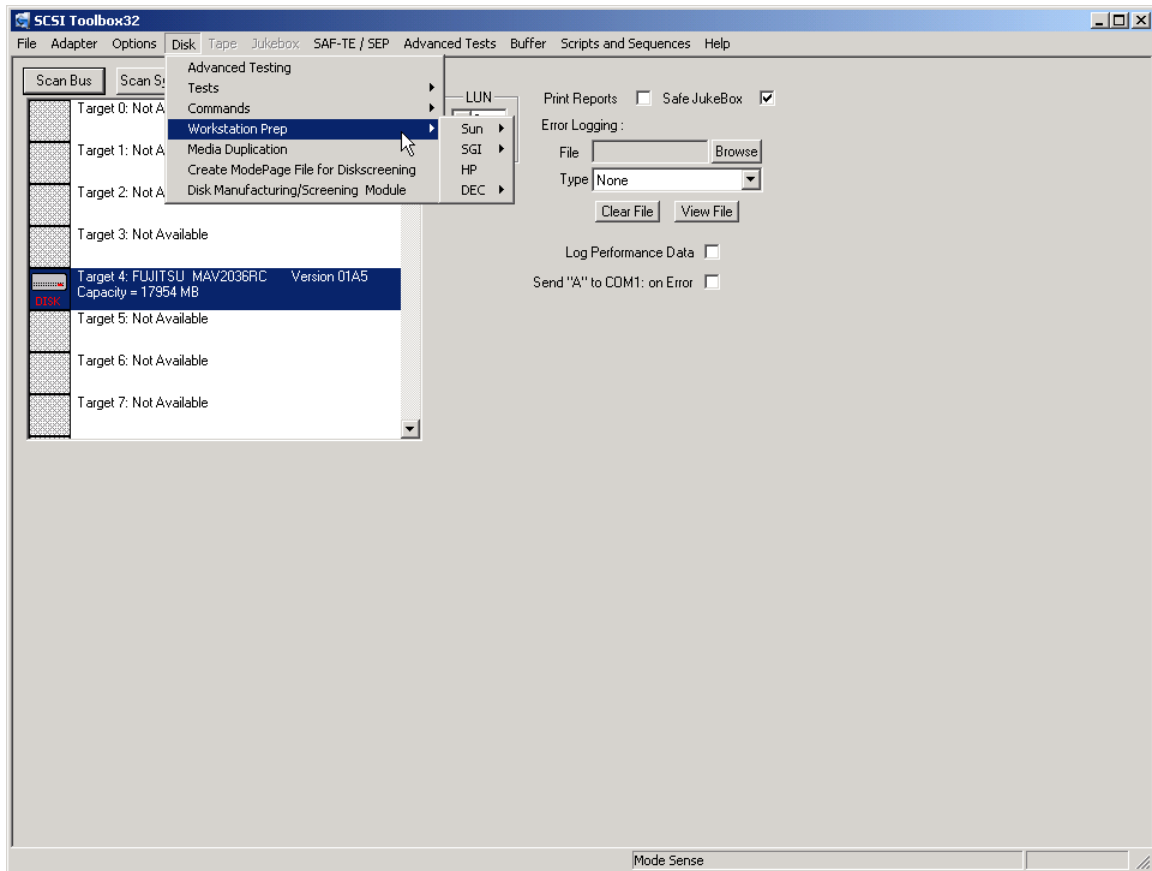
This command allows both the Primary and Grown defect data to be displayed, printed, or saved to a file. The number of defects is shown as well as each individual defect. A summary of defects per head is shown at the end of the defect data. The Defect List Format may be specified using the radio buttons. Click the Primary Defects button to display the primary (factory) defects, or the Grown Defects button to view the grown defect list.



Please note - not all disk drives will support all three formats



Workstation Prep



The Workstation Prep Module allows you to quickly prepare any disk device for use on many Unix-based workstations. The type of prep and what it entails depends on which type of workstation is being used, and ranges from supplying drive-specific information for entry into a disktab file, to rewriting defect information and calculating "magic numbers." Through whatever means, the results are the sam--drives can be quickly prepared to work on a number of different workstations.

One main advantage of using the SCSI *toolbox32* for preparing drives for workstations is the intelligence built into the SCSI *toolbox32*. As soon as you receive new disk devices from drive vendors, the SCSI *toolbox32* will show you how to optimally configure them for workstation use.

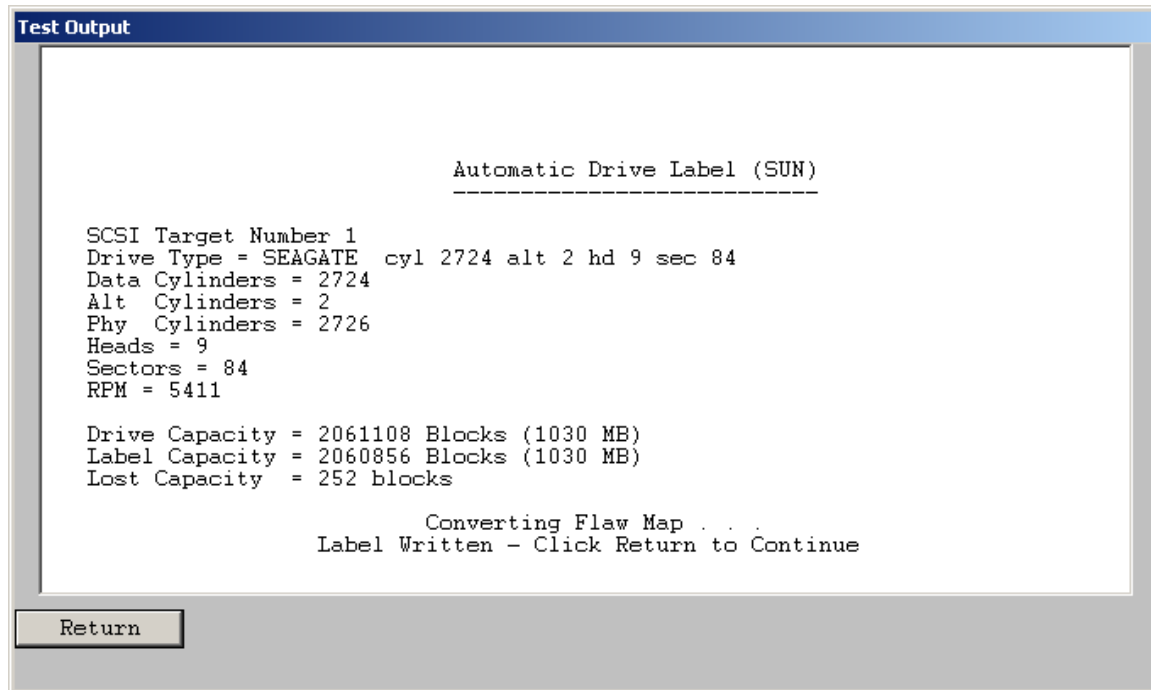
A good strategy for preparing disks is to first qualify the drive as functional. This can be quickly ascertained by running the Quick QC Test. If you are more comfortable completely reformatting the drive, you can do so using the Format Command. If you want to test the drive for an extended period of time, the Random Write/Read Test can be run for as long as you like. Of course, saving the screens from the above tests and functions, and sending the test results to a log file will allow you to print out all results and include them with your disk drives, so that the end user of the drive will know that the drive was indeed tested.

You might also want to take the time to print the drive's flaw list to a file, using the Read Defect Data command. The flaw list can be printed and included with the drive.

The Sun Workstation Menu

Read Existing Label

This choice lets you confirm whether a drive has an existing, valid Sun label and partition table. If the selected and the drive already has a Sun label, the label information will be displayed.



Write New Label

Pick this choice and the SCSI *toolbox32* will analyze the drive, calculate the optimal geometry for a Sun, and then create a label based on this geometry. It will create eight partitions, with the C partition the size of the entire drive.

Test Output

```
Automatic Drive Label (SUN)
-----

SCSI Target Number 1
Drive Type = SEAGATE cyl 2724 alt 2 hd 9 sec 84
Data Cylinders = 2724
Alt Cylinders = 2
Phy Cylinders = 2726
Heads = 9
Sectors = 84
RPM = 5411

Drive Capacity = 2061108 Blocks (1030 MB)
Label Capacity = 2060856 Blocks (1030 MB)
Lost Capacity = 252 blocks

Converting Flaw Map . . .
Label Written - Click Return to Continue
```

Return

Calculate Format.dat Entry

This choice will calculate and display the lines which can be added to the format.dat file which will correspond to this drive.

Test Output

```
Create format.dat Entry
-----

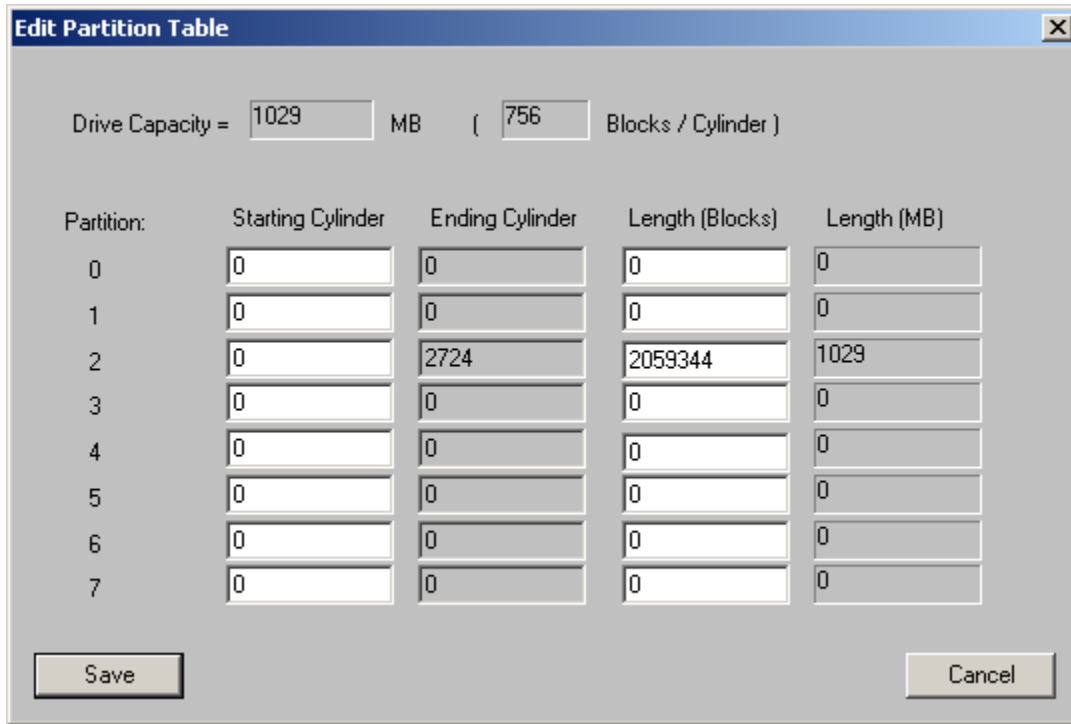
Add the following entry to your /etc/format.dat file:

disk_type = "SEAGATE ST31200W" \
: ctrl = SCSI : fmt_time = 9 \
: ncyl = 2724 : acyl = 2 : pcyl = 2726 : nhead = 9 : nsect = 84 \
: rpm = 5411 : bpt = 50148

partition = "SEAGATE ST31200W" \
: disk = "SEAGATE ST31200W" : ctrl = SCSI \
: a = 0, 15876 : b = 21, 65016 : c = 0, 2059344 \
: g = 107, 1978452
```

Return

Edit Sun Partition Table



Drive Capacity = MB (Blocks / Cylinder)

Partition:	Starting Cylinder	Ending Cylinder	Length (Blocks)	Length (MB)
0	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
1	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
2	<input type="text" value="0"/>	<input type="text" value="2724"/>	<input type="text" value="2059344"/>	<input type="text" value="1029"/>
3	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
4	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
5	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
6	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
7	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>

This function allows you to edit the partition table, creating new partitions or editing the size of existing partitions.

The SGI Workstation Menu

Read Current Label

This option will check a drive for the presence of a valid SGI label. If found the details of the label will be displayed.

Test Output

```
Read SGI Drive Label
-----
Partition      Size (blocks)  Size (MB)
-----
0              32508          16
1              80136          40
2              0              0
3              0              0
4              0              0
5              0              0
6             1943676        971
7             2056320        1028
8              4536           2
9              0              0
10            2060856        1030

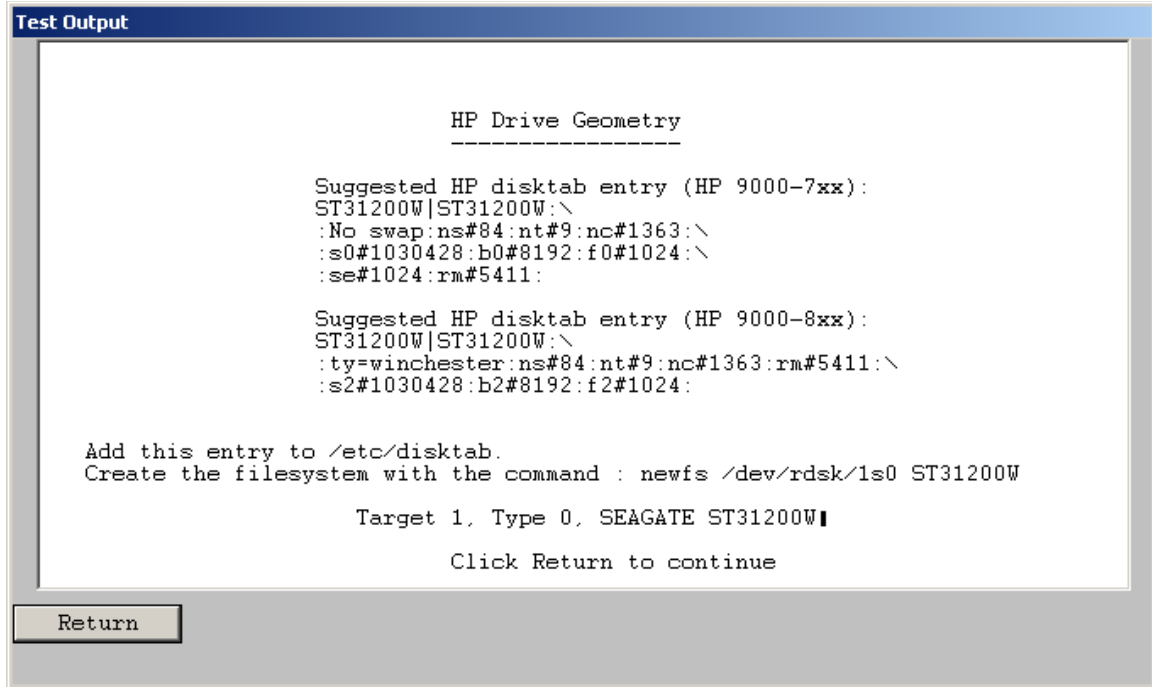
Click Return to continue
```

Return

Write Label

This option lets you write a Silicon Graphics label on the current disk drive. This process is entirely automatic, and will return a status showing whether the label was successfully written or not. The drive is analyzed and a geometry is calculated and written to the drive, along with SGI standard partitions.

The HP Workstation Menu



This choice will generate an entry to be added to the system's disktab file. This file is located in the /etc directory on the HP machine. The following figure shows an example of an HP entry.

HP Drive Geometry

Suggested HP disktab entry:

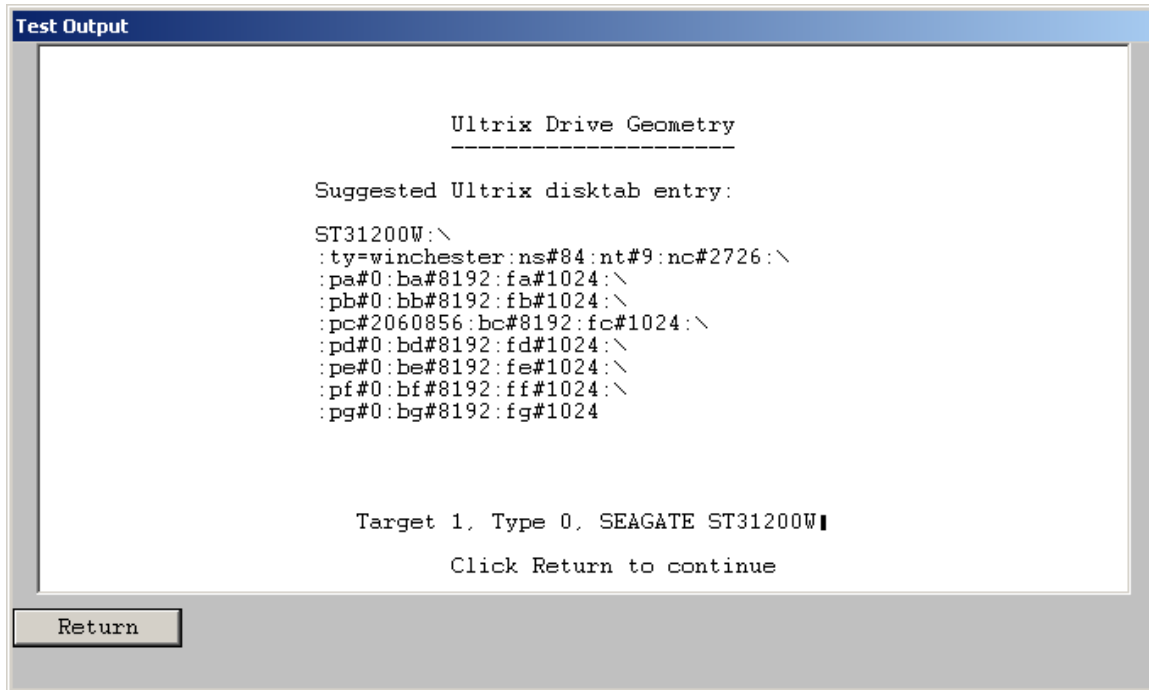
```
IBM_OEM_|IBM_OEM_:\
:No swap:ns#63:nt#15:nc#1038:\
s0#980910:b0#8192:f0#1024:\
se#1024:rm#3600:
```

Add this entry to /etc/disktab.

Create the filesystem with the command : newfs /dev/rdisk/1s0 IBM_OEM

The DEC Workstation Menu

DEC Ultrix



This choice will generate an entry to be added to the system's disktab file. This file is located in the /etc directory on the DEC machine. The following figure shows an example of a DEC entry.

```
Ultrix Drive Geometry
Suggested Ultrix disktab entry:
IBM_OEM_\
:ty=winchester:ns#63:nt#15:nc#2076:\
:pa#0:ba#8192:fa#1024:\
:pb#0:bb#8192:fb#1024:\
:pc#1961820:bc#8192:fc#1024:\
:pd#0:bd#8192:fd#1024:\
:pe#0:be#8192:fe#1024:\
:pf#0:bf#8192:ff#1024:\
:pg#0:bg#8192:fg#1024
```

Add this entry to /etc/disktab. Then perform a newfs with the following command: newfs /dev/rrz1c IBM_OEM_

DEC OSF

This choice will generate an entry to be added to the system's disktab file. This file is located in the /etc directory on the DEC machine. The following figure shows an example of a DEC entry.

```
OSF Drive Geometry
Suggested OSF disktab entry:
IBM_OEM_\
:ty=winchester:ns#63:nt#15:nc#2076:\
:pa#0:ba#8192:fa#1024:\
```

```
:pb#0:bb#8192:fb#1024:\  
:pc#1961820:bc#8192:fc#1024:\  
:pd#0:bd#8192:fd#1024:\  
:pe#0:be#8192:fe#1024:\  
:pf#0:bf#8192:ff#1024:\  
:pg#0:bg#8192:fg#1024
```

Add this entry to /etc/disktab.

Then perform a newfs with the following command: newfs /dev/rrz1c IBM_OEM_

Media Duplication

The optional Media Module lets you copy or clone media from one source to up to six destinations. This copying is done on a byte-by-byte basis, so the copy will be an exact replica of the original. The Media Module will copy disks to disks, and tapes to tapes, and can be used to copy a marginal disk or transcribe tapes from one media type to another. In the case of a tape-to-tape copy, the destination tape will be written in the same format as the source tape, in regard to block size, etc. All filemarks will be copied as well. A tape copy will end when the End of Tape mark is encountered on the source tape.

Disk copies will proceed until the last block is encountered on either the source or the destination disk. This allows you to copy part of a large disk to a smaller one, or to copy all of a small disk to a larger one. The copy operation of the Media Module can be interrupted at any time by pressing the "Cancel" button.

The screenshot shows a dialog box titled "Multiple Adapter Media Duplication". It has a "Source Device:" dropdown menu currently showing "Bus 2 TID 1 LUN 0 SEAGATE ST31200W SUN1.05". Below this are three checkboxes: "Ignore Errors", "Write with Verify", and "Checksum Verification". To the right of these are "Start Block:" (0) and "End Block:" (2061106) input fields. A "Copy Progress" section contains four input fields: "Current Block:" (0), "System ID Rate (KB/Sec)" (.....), "Time to Complete" (.....), and "Error Count:" (0). On the right side, there is a "Destination Devices:" list box containing four entries: "SOURCE: Bus 2 TID 1 LUN 0 SEAGATE ST31200W SU", "Bus 2 TID 2 LUN 0 SEAGATE ST31200W SUN1.05", "Bus 2 TID 3 LUN 0 SEAGATE ST31200W SUN1.05", and "Bus 2 TID 5 LUN 0 HP Ultrium 1-SCSI". Below the list box is an "Eject when finished" checkbox and a "Select All Destinations" button. At the bottom, there is a "Duplication Status:" label above a large empty text area. The bottom of the dialog features four buttons: "Return", "Start Copy", "Calc. Checksum", and "Cancel".

Choosing the Source Device

Use the Source Device pull-down list to pick which device to copy from. Click on the arrow in the selection box to display a list of possible source devices. Clicking on one of those devices will select it. If the device is a disk, it will default the start and end blocks to the full capacity of the disk.

Choosing the Destination Drive

Destination devices are chosen from the Destination Device List box. This list box supports the standard Windows method of selecting multiple items from a list. You may select 1 or more of the available devices displayed.

Copying Rules

You must be copying to and from the same type of device, i.e., disk-to-disk and tape-to-tape. The source and destination devices must be different SCSI targets. If an error is encountered during a copy operation, the copy will be aborted. In the case of multiple destinations, if the copy encounters an error to a particular device, the copy to that device will be aborted; all others will continue. Multiple-device copies are implemented by:

1. Reading a block (or blocks) of data from the source.
2. Writing that data to each destination device. (This eliminates reading the source device multiple times.)

Keep in mind that throughput is important. Faster bus adapters will provide the highest levels of performance.

Choosing the Copy Range

In the case of a disk-to-disk copy, you may specify the range of blocks that are copied. The starting block will be defaulted to block zero, and the ending block will default to the highest block number of the source disk. You may enter any other value into the starting and ending block in order to copy less than the whole source disk.

Write with Verify

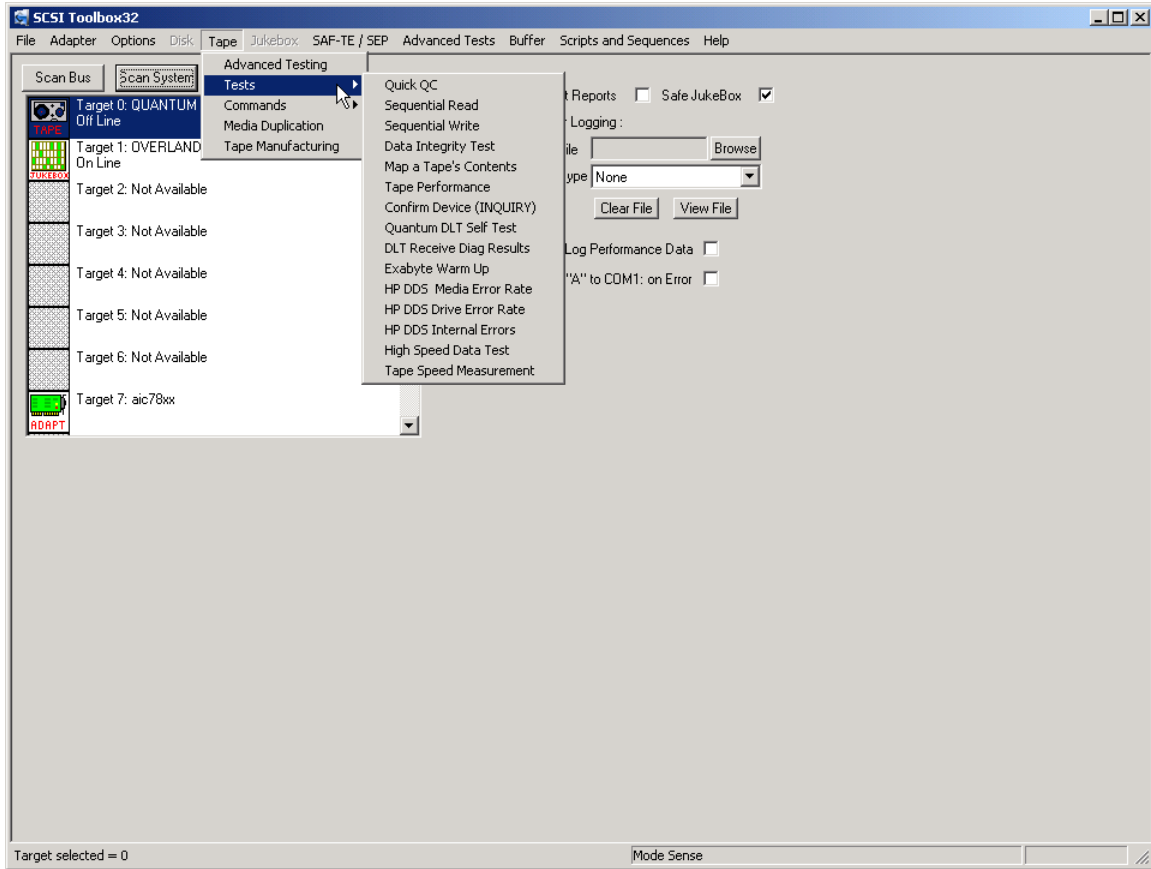
Selecting this check box will cause all write commands to be written with verify commands. This will cause the destination drive to compare the CRC and ECC data for each block after it is written. **Note: Selecting this option will significantly reduce the throughput of the copy operation.**

Ignore Errors

Selecting this check box will cause the copy operation to continue on a device after an error has occurred. You may want to use this to skip over bad blocks on a device.

Tape Tests

The following tests are available via the Tape Test menu



TapeWatch

Drive Watch [X]

Read Data Rates

Average Transfer Rate =	0.000	KB/Sec
High Transfer Rate =	0.000	KB/Sec
Low Transfer Rate =	0.000	KB/Sec

SCSI Target = 3

LUN = 0

Write Data Rates

Average Transfer Rate =	8011.622	KB/Sec
High Transfer Rate =	9149.565	KB/Sec
Low Transfer Rate =	7169.560	KB/Sec

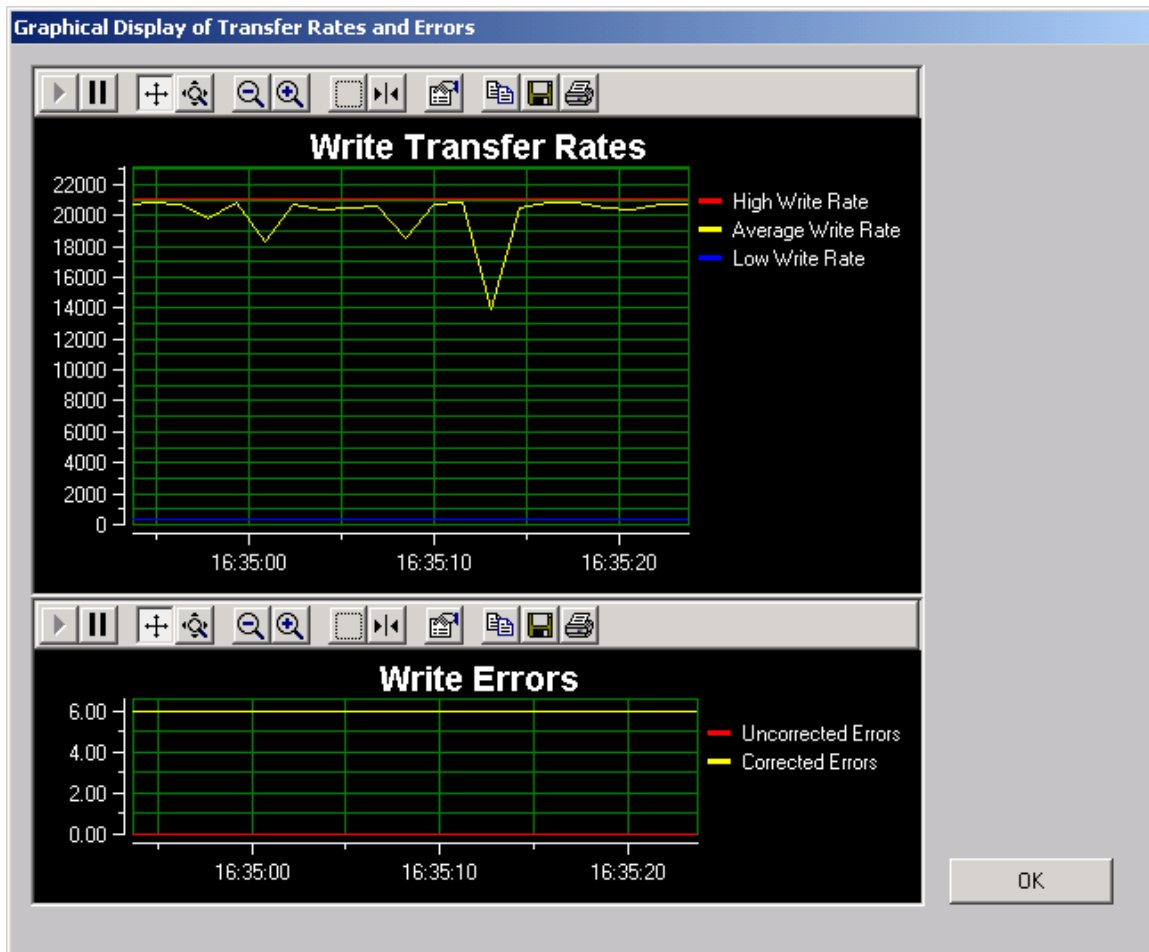
Number of hard errors = 0

Number of soft errors = 0

Show Graphical Display

OK Refresh Clear Drive Stats

Tape Drive Watch allows real-time monitoring of tape drive performance data as any test is running. The Show Graphical Display button will show performance and error information graphically:



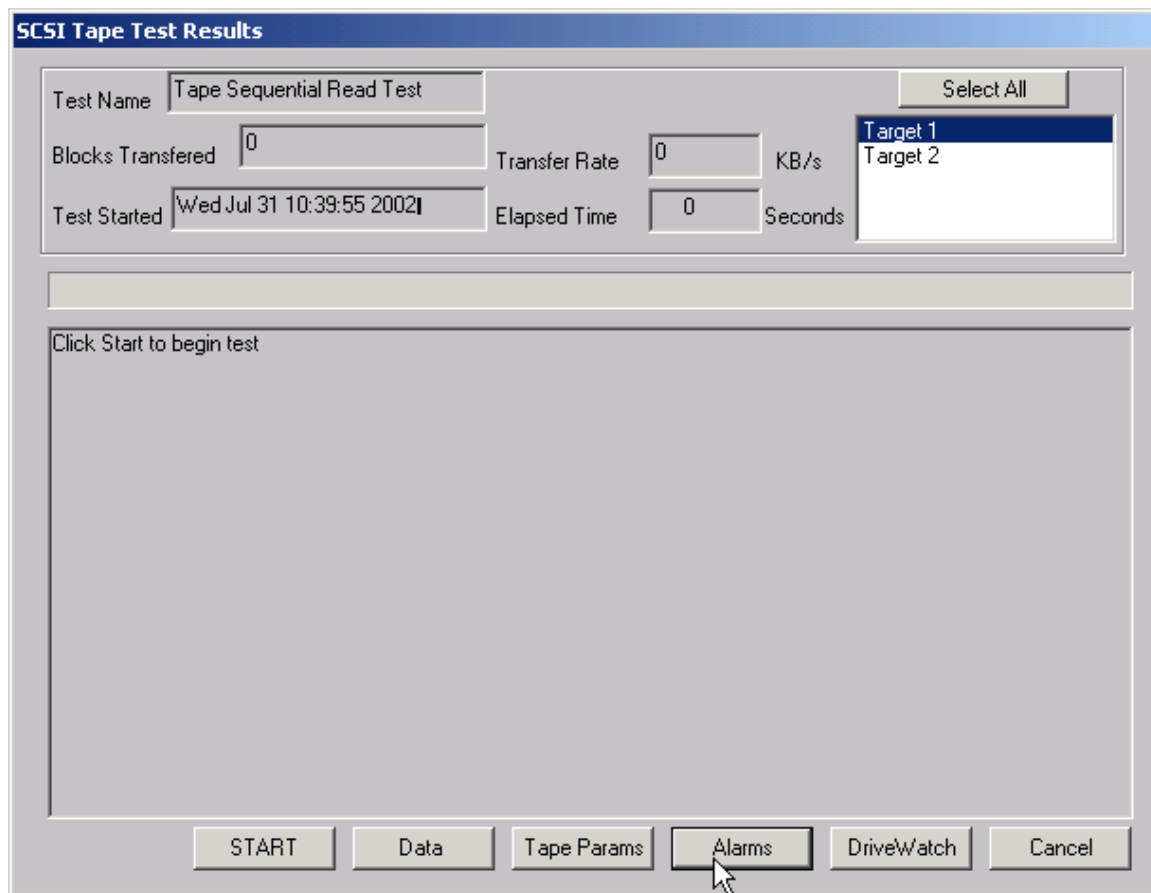
Quick QC Test

The screenshot shows a software window titled "SCSI Tape Test Results". At the top, there is a blue header bar with the title. Below the header, the interface is divided into several sections. On the left, there are four input fields: "Test Name" containing "Tape Quick QC Test", "Blocks Transferred" containing "0", "Test Started" containing "Wed Jul 31 10:39:40 2002", and "Transfer Rate" containing "0" with "KB/s" to its right. Below these is an "Elapsed Time" field with "Seconds" to its right. To the right of these fields is a "Select All" button and a list box containing "Target 1" (highlighted) and "Target 2". Below the input fields is a large grey rectangular area with the text "Click START to begin test". At the bottom of the window, there are four buttons: "Return", "START", "Data", and "Tape Params". A mouse cursor is visible over the "Tape Params" button.

The Quick QC test exercises every function of the tape drive. The test Progresses in the following order:

- Rewind tape
- Write blocks of data to tape
- Write a File Mark
- Rewind tape
- Read blocks of data from tape
- Space Reverse to beginning of tape
- Space Forward to File Mark
- Rewind tape

Sequential Read Test



The image shows a software dialog box titled "SCSI Tape Test Results". It contains several input fields and a list box. The "Test Name" field is set to "Tape Sequential Read Test". The "Blocks Transferred" field shows "0". The "Transfer Rate" field shows "0" KB/s. The "Test Started" field shows "Wed Jul 31 10:39:55 2002". The "Elapsed Time" field shows "0" Seconds. A "Select All" button is located to the right of the "Test Name" field. A list box on the right contains "Target 1" (selected) and "Target 2". A large text area in the center contains the instruction "Click Start to begin test". At the bottom, there are six buttons: "START", "Data", "Tape Params", "Alarms", "DriveWatch", and "Cancel". A mouse cursor is pointing at the "Alarms" button.

Test Name	Tape Sequential Read Test	Select All		
Blocks Transferred	0	Transfer Rate	0	KB/s
Test Started	Wed Jul 31 10:39:55 2002	Elapsed Time	0	Seconds

Click Start to begin test

START Data Tape Params Alarms DriveWatch Cancel

This test asks if you want to rewind the tape or not then sequentially reads every block on the tape, until either a Blank Check (End of Data) or End of Media is encountered. In the case of tape drives that can report remaining tape and error information, this information is displayed and updated every 1 MB of data. In this case, the Total Error Count, Errors since last Error, and Remaining Tape data is displayed.

Sequential Write Test

SCSI Tape Test Results

Test Name: Tape Sequential Write Test

Blocks Transferred: 0

Transfer Rate: 0 KB/s

Test Started: Wed Jul 31 10:40:12 2002

Elapsed Time: 0 Seconds

Select All

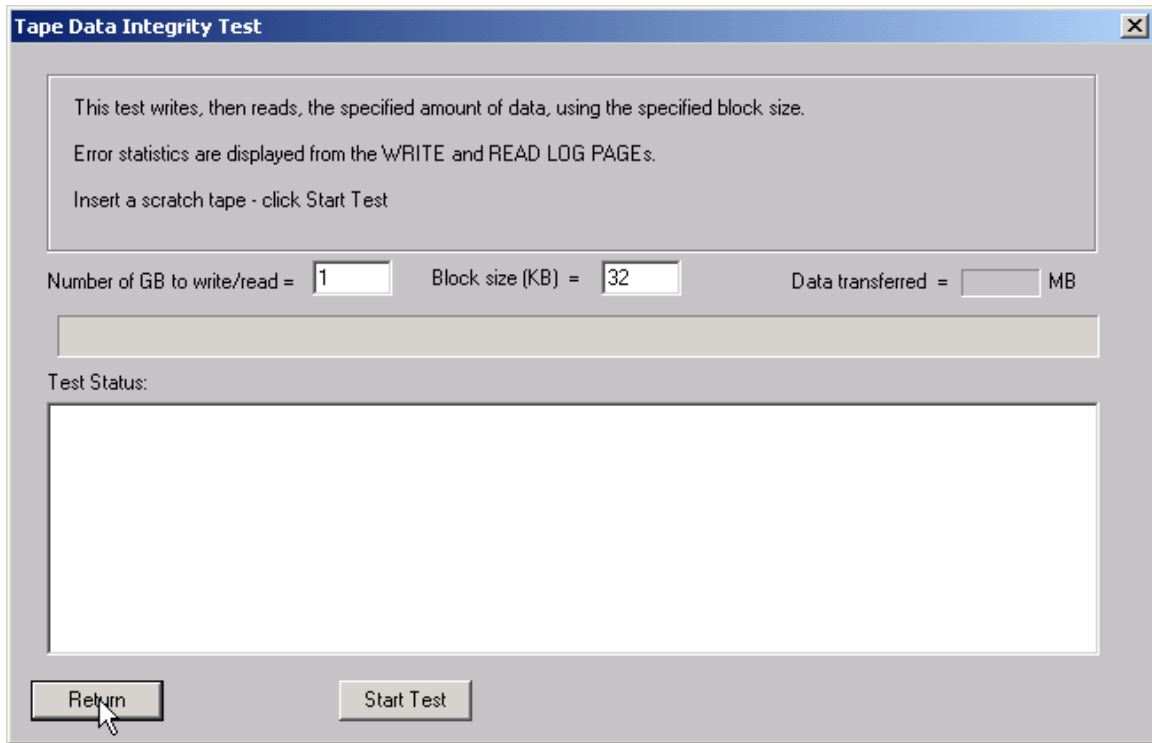
Target 1
Target 2

Click Start to begin test

START Data Tape Params Alarms DriveWatch Cancel

The Sequential Write Test asks if you want to rewind the tape or not then writes to every block on the tape media until End of Media is encountered. In the case of Exabyte 8mm drives and most 4mm DAT drives, the remaining tape and error statistics are displayed on the screen after each one megabyte of testing is complete.

Data Integrity Test



The screenshot shows a window titled "Tape Data Integrity Test" with a close button in the top right corner. Inside the window, there is a text box containing the following instructions:

This test writes, then reads, the specified amount of data, using the specified block size.
Error statistics are displayed from the WRITE and READ LOG PAGEs.
Insert a scratch tape - click Start Test

Below the text box, there are three input fields:

Number of GB to write/read = Block size (KB) = Data transferred = MB

Below the input fields is a horizontal progress bar. Underneath the progress bar is the label "Test Status:" followed by a large empty rectangular area for displaying test results.

At the bottom of the window, there are two buttons: "Return" and "Start Test". A mouse cursor is pointing at the "Return" button.

This test will rewind the tape, then write the specified amount of data to the selected drive. At the end of the write phase the error information collected from the drive's log pages will be displayed.

Map a Tape's Contents

Map Tape Contents

General Tape Information

Vendor = Firmware =

Product =

Read Errors (when test started) = Currently =

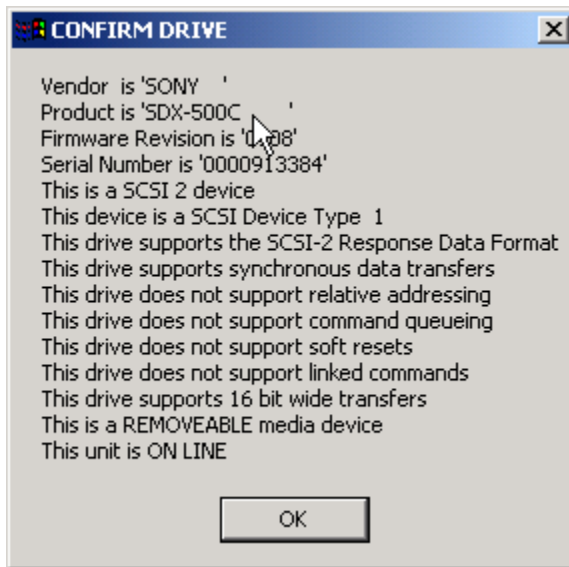
Number of blocks read =

Map of Tape Contents

Status:

Click the Map Tape button and the tape will be rewound, then read. The contents of the tape (data, filemark, etc) will be displayed in the Map of Tape Contents window. For each section of data the read error rates will be displayed from the drive's log pages.

Confirm Device (Inquiry)



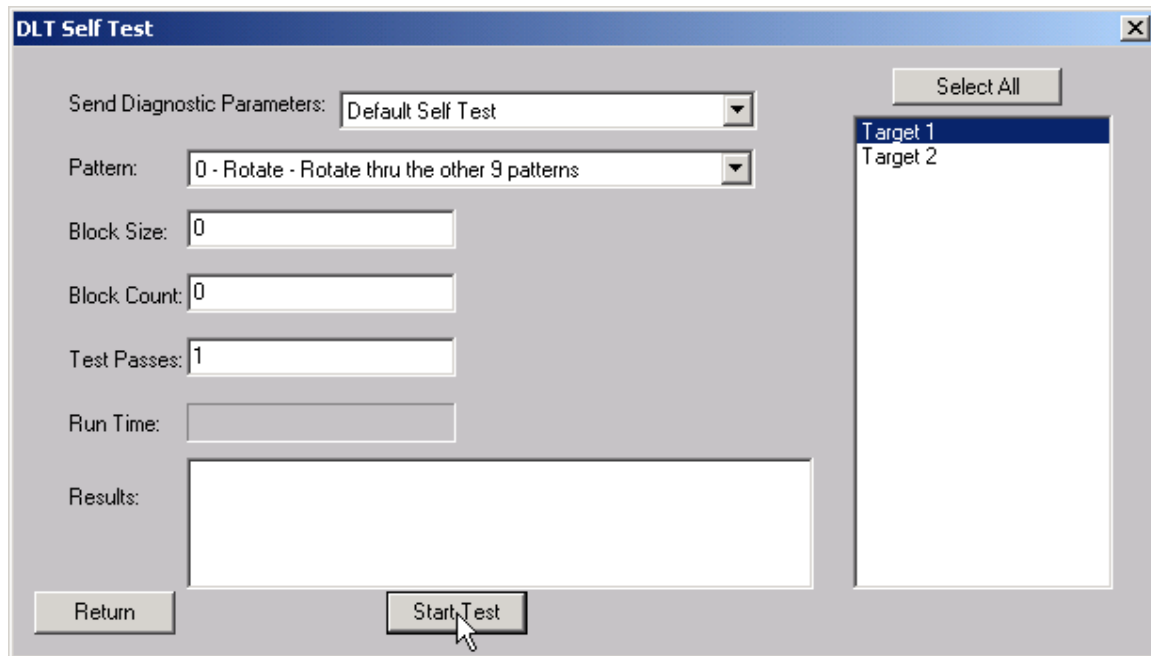
This test will report as much information as is available about the currently selected tape drive. In all cases, the VENDOR, PRODUCT, and REVISION fields returned by a SCSI INQUIRY command will be displayed, along with the ANSI SCSI level of the device, and the device's ability to support any of the following:

- Synchronous data transfers
- Relative addressing
- Command queuing
- Soft resets
- Linked commands, and

Whether the device is a removable media type or not

Some devices will report back with much more information than indicated above. In these cases the information is often in a vendor-unique format, which the SCSI *toolbox* will interpret for you. If additional data beyond that described above is available that data will displayed.

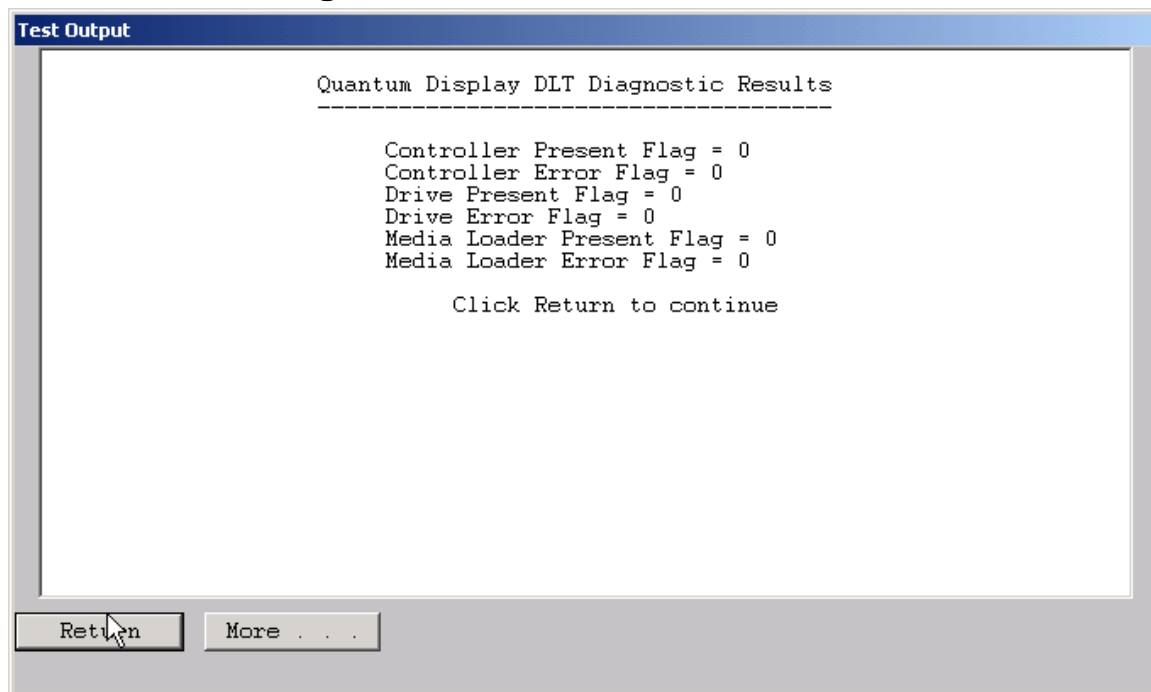
Quantum DLT Self Test



The image shows a Windows-style dialog box titled "DLT Self Test". It contains several input fields and buttons. On the left side, there are labels and input boxes for "Send Diagnostic Parameters" (set to "Default Self Test"), "Pattern" (set to "0 - Rotate - Rotate thru the other 9 patterns"), "Block Size" (0), "Block Count" (0), "Test Passes" (1), and "Run Time" (empty). Below these is a large empty box labeled "Results". At the bottom left is a "Return" button, and at the bottom center is a "Start Test" button with a mouse cursor over it. On the right side, there is a "Select All" button above a list box containing "Target 1" and "Target 2".

This test allows you to set all the user defined parameters, or use the default self test. This test supports disconnect-reconnect with Adaptec or Q-Logic SCSI adapters in the multi-drive mode.

DLT Receive Diagnostic Results



The image shows a Windows-style dialog box titled "Test Output". It contains a large text area with the following text:

```
Quantum Display DLT Diagnostic Results
-----
Controller Present Flag = 0
Controller Error Flag = 0
Drive Present Flag = 0
Drive Error Flag = 0
Media Loader Present Flag = 0
Media Loader Error Flag = 0

Click Return to continue
```

At the bottom of the dialog box, there are two buttons: "Return" and "More . . .". A mouse cursor is pointing at the "Return" button.

DLT Receive Diagnostic Results

This function displays the results of a recv diagnostic command for the controller, drive and medium on any DLT-type device.

Exabyte Warm Up Test

The screenshot shows a software window titled "SCSI Tape Test Results". At the top, the test name is "Exabyte Warm-up Test". Below this, there are several input fields: "Blocks Transferred" is set to 0, "Transfer Rate" is 10000 KB/s, and "Test Started" is "Wed Jul 31 10:45:05 2002". The "Elapsed Time" is 160 Seconds. To the right of these fields is a "Select All" button and a list box containing "Target 1" and "Target 2", with "Target 1" selected. Below the input fields is a large text area containing the following text: "This test is to be used on Exabyte tape drives that have been stored for six months or more. The test will write, then read 500 MB of data three times. This test will run for approximately two hours, and may be interrupted by clicking the CANCEL button. Data Written = 133 MB (Pass 1 of 6)". At the bottom of the window, there are six buttons: "START", "Data", "Tape Params", "Alarms", "DriveWatch", and "Cancel".

This test is specific to Exabyte 8mm tape drives. If the drive is inactive for long periods of time then is used it may fail. This test simply moves the tape forward and back writes data etc to break in a drive that has been inactive. This helps get all the moving parts moving smoothly again and ready for normal use.

DDS (4mm) Media Error Rate Test

SCSI Tape Test Results

Test Name	HP DDS (Media) Error Rate Test	<input type="button" value="Select All"/>	
Blocks Transferred	0	Transfer Rate	0 KB/s
Test Started	Wed Jul 31 10:48:25 2002	Elapsed Time	2 Seconds

RAW Error Rate = 1.552795%

C3 Error Rate = 0.000000%

RAW and C3 error rates must be less than 1 %. If either rate exceeds 1 % the tape should not be used.

Test Finished - Click OK

<input type="button" value="Return"/>	<input type="button" value="Data"/>	<input type="button" value="Tape Params"/>	<input type="button" value="Alarms"/>	<input type="button" value="DriveWatch"/>
---------------------------------------	-------------------------------------	--	---------------------------------------	---

This test reads information from the header of a DDS (4mm) tape and interprets it into tape error statistics. This information is obtained by reading the TAPE LOG PAGE of the LOG SENSE command.

DDS (4mm) Drive Error Rate Test

SCSI Tape Test Results

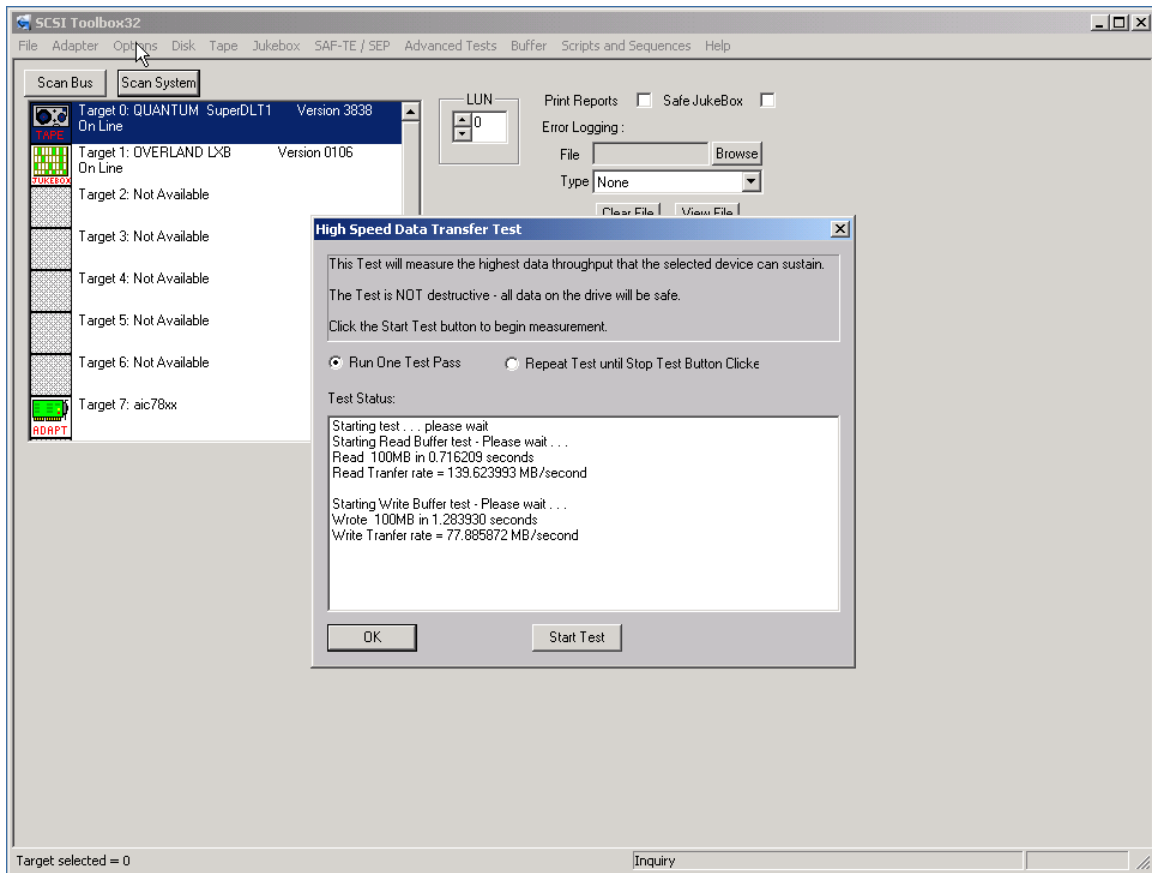
Test Name	HP Block (Drive) Error Rate Test	<input type="button" value="Select All"/>	
Blocks Transferred	0	Transfer Rate	0 KB/s
Test Started	Wed Jul 31 10:48:52 2002	Elapsed Time	6 Seconds

Target 1
Target 2

Rewinding . . .

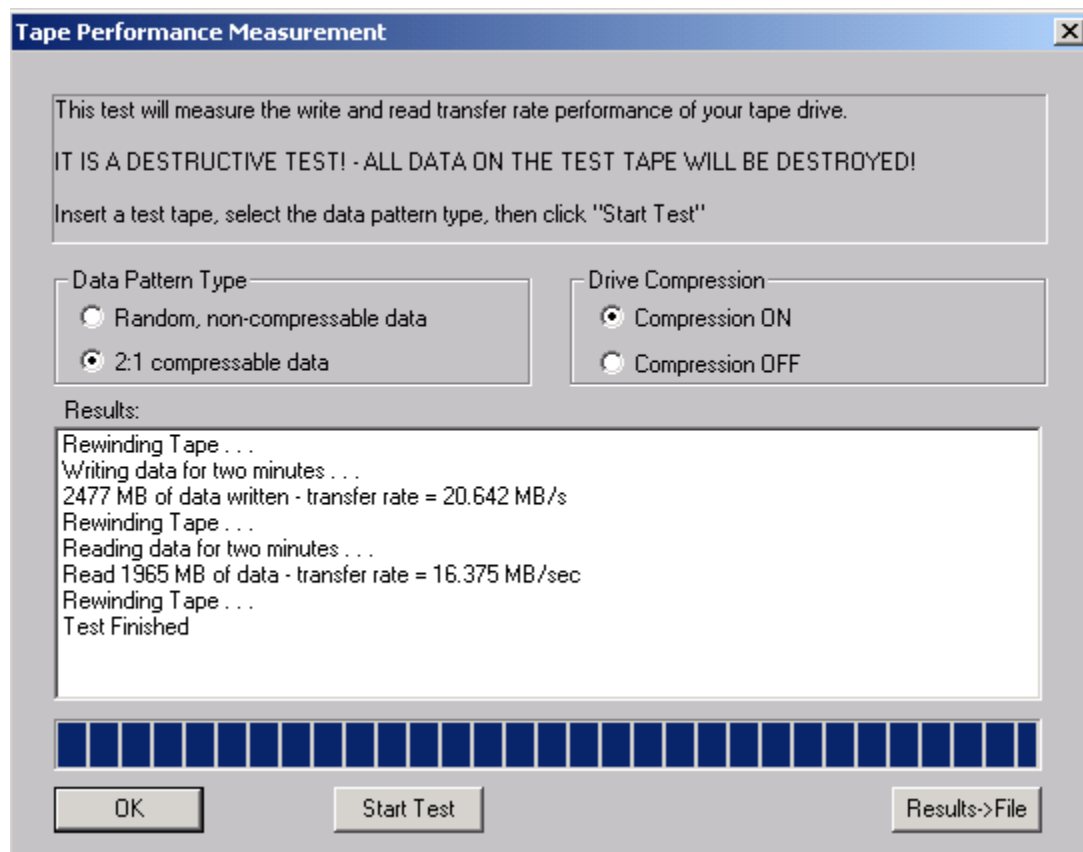
This test will space to the End of Data (EOD) on the tape, then will write 50 MB of data. The Block Error Rates from LOG SENSE PAGES 34 and 35 will then be read and interpreted. This test will show you the error rate of the tape drive itself, where the HP Media Error Rate test will show the error rate of the tape media.

High Speed Data Test



The High Speed Data test transfers data to and from the drive without transferring to the actual drive media. This allows the data transfer speed of the drive and host bus adapter to be determined.

High Performance Measurement Test



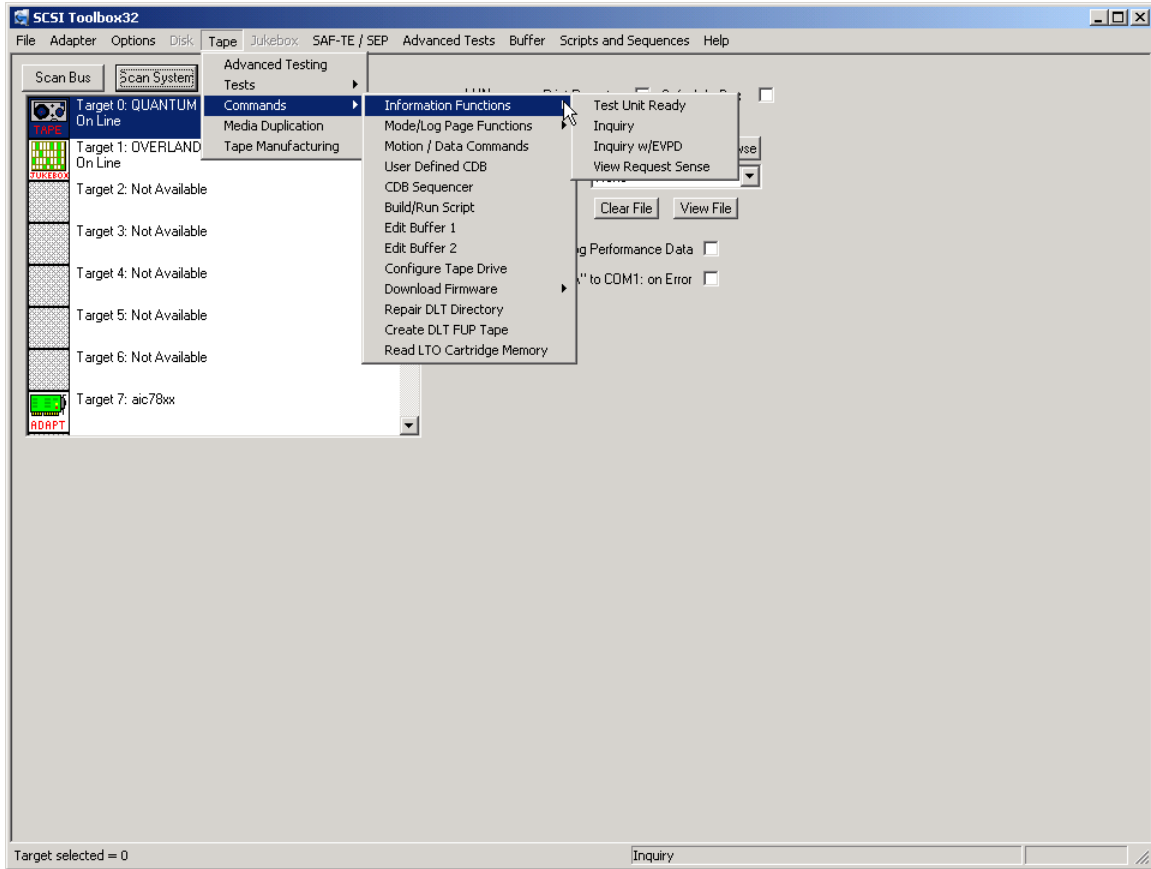
The Tape Performance Measurement test transfers data to and from the drive for two minutes, measuring the amount of data that can be transferred in that time to determine the “real –world” data throughput of the drive.

The data pattern can be selected to be a non-compressable format, or a format that will be able to be compressed to a 2:1 ratio.

The drive’s compression can be turned on or off for this test.

Tape Commands

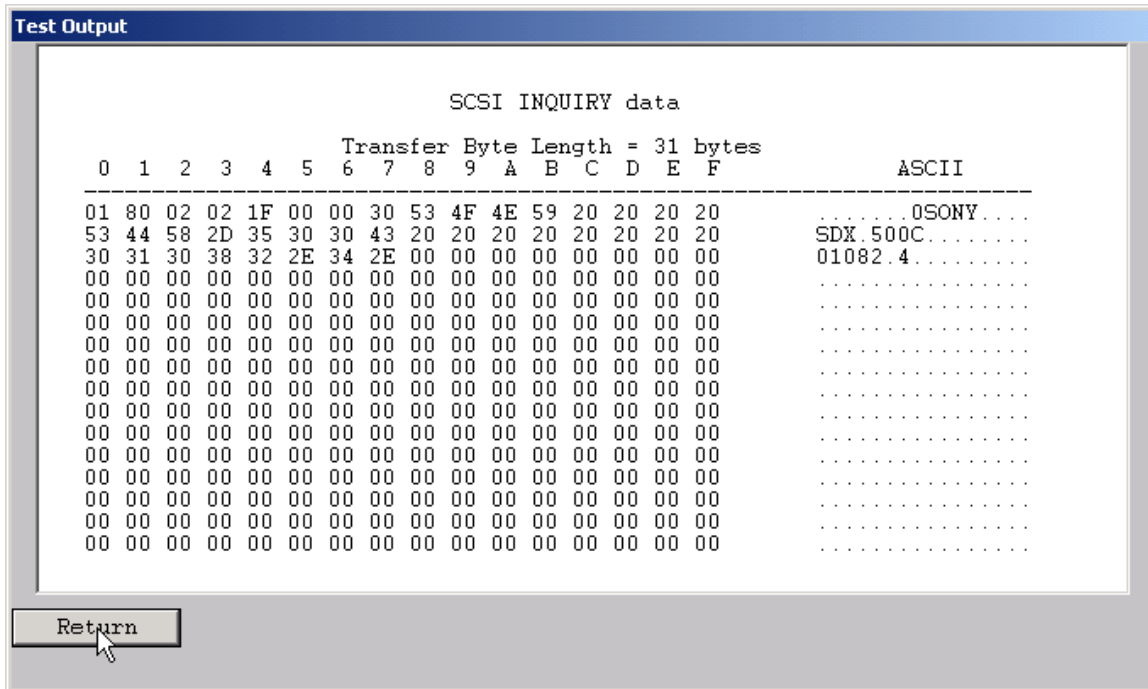
The following commands are available via the Tape Commands menu:



Information Functions – Test Unit Ready

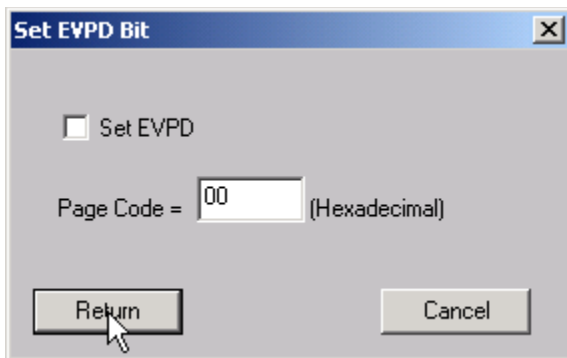
This command displays the return value from a SCSI Test Unit Ready command. A value of zero indicates a successful command completed.

Inquiry



This command displays the data returned from the SCSI INQUIRY command. The data is shown in hexadecimal format on the left of the screen and in ASCII format on the right side of the screen.

Inquiry with EVPD



This option lets you specify whether to set the EVPD bit in the INQUIRY command, and which EVPD page you would like to display.

View most recent Sense Data

Test Output

Display Most Recent Request Sense Data

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
70	00	02	00	00	00	00	14	00	00	00	00	3A	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

(Click Return to Continue)

Return

This command will display the most recent request sense data. This data will have been generated by the most recent check condition that occurred.

Mode Page Edit

Custom Mode Page

Block Header: 00 00 00 00 (0 1 2 3)

Block Descriptor: 00 00 00 00 00 00 00 00 (4 5 6 7 8 9 10 11)

SP Bit:

Page Data

Default: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Mask: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Current: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)

Default: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Mask: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Current: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31)

Page #: 00 Current Byte:

Mode Sense Mode Select Restore Defaults Return

This command allows you to select one of several standard mode pages and view and or change the values by their acromymns. For example selecting the Error Recovery page you can view the setting for the PER bit and change it.

This is a "Hex Edit" selection which displays a dialog box with which you can select the mode page number, and issue a mode sense for that page. The data is then displayed as a series of hexadecimal byte values with the changeable mask value for that byte. You may then edit any byte to specify a new value then issue a mode select to store the new values to the device.

DLT Log Pages

This command will read, format and display the log page data for DLT drives from the following log pages:

1. Write Error Counter Page
2. Read Error Counter Page
3. Tape Log Page
4. Tape Capacity Page
5. Write Frames Error Counter Page
6. Read Frames Error Counter Page
7. Data Compression Page

DDS Log Pages

This command will read, format and display the log page data for DDS drives from the following log pages:

1. Write Error Counter Page
2. Read Error Counter Page
3. Tape Log Page
4. Tape Capacity Page
5. Write Frames Error Counter Page
6. Read Frames Error Counter Page
7. Data Compression Page

View Log Pages

Motion/Data Functions

Tape Motion & Data Commands

Motion Commands

Rewind
Unload
Read Position

Space Commands How Many
 EOD
 Forward Blocks
 Reverse Files

Locate Command Block Address
Partition
 CP

Current Tape Position

<input type="checkbox"/> BOP	Partition	<input type="text" value="0"/>	Block Size: Fixed 512 byte	
<input type="checkbox"/> EOP	First Block Loc.	<input type="text" value="909629"/>	Blocks in Buffer	<input type="text" value="0"/>
<input type="checkbox"/> BPU	Last Block Loc.	<input type="text" value="0"/>	Bytes in Buffer	<input type="text" value="0"/>

Data Commands

Write Filemark
Write Setmark
Flush Buffer

Execute I/O
Repeat Count Single Block I/O Read
 Multi Block I/O Write
I/O Size (Dec. Bytes)

Configure Drive Data Pattern Return

This dialog provides access to the basic motion and data commands normally used to diagnose tape devices.

rewind

This command executes a SCSI REWIND command, rewinding the tape device under test to Beginning of Media. The time to complete the rewind operation is displayed.

unload

Clicking this button issues an UNLOAD command to the selected drive. The time to complete the unload operation is displayed.

space

Clicking this button will issue the space command as specified by the radio buttons and edit field shown below the button. Each space type is described as follows:

- EOD
 - o This command will space forward on the tape to the logical end of data mark which is written on the media. No other settings are valid with this type of space command.
- Forward
 - o This command allows you to space forward through the tape media by file marks. If End of Media or Blank Check (End of Data) is encountered before a file mark has been spaced over, the command will be aborted and an error message will be displayed.
- Reverse
 - o This command spaces from the current position toward Beginning of Tape (BOT). The user can select either blocks or files, and indicate how many to sequentially space over.

read position

The Read Position command will read the current block number from the drive and display it in the Block Address window.

Locate

Enter the block number and position that you want to position the tape to in the Block Address and Partition windows. Then Click the Locate button and the tape will be positioned to this block number.

write filemark

Clicking this button will issue a WRITE FILEMARK command to the tape drive with the NUMBER OF FILEMARKS field set to 1.

write setmark

Clicking this button will issue a WRITE FILEMARK command to the tape drive with the NUMBER OF FILEMARKS field set to 1 and the Wsmk bit set to 1.

Note: not all tape drives support this command

flush buffer

Clicking this button will issue a WRITE FILEMARK command to the tape drive with the NUMBER OF FILEMARKS field set to 0. This command will cause the drive to flush any data that is in the buffer to the physical tape media.

read block(s)

Data blocks can be read from the tape drive by clicking the Read radio button, then clicking the Execute IO button. The number of blocks read specified in the Repeat Count window.

write block(s)

Data blocks can be written to the tape drive by clicking the Write radio button, then clicking the Execute IO button. The number of blocks written is specified in the Repeat Count window.

User Defined CDB

The screenshot shows the 'User Defined CDB' dialog box. The 'SCSI Commands' list includes: Change Definition, Compare, Copy and Verify, Extended Read, Extended Seek, Extended Write, and Format. The 'CDB Length' is set to 6 Bytes. The 'Data In' radio button is selected. The 'Transfer Length' is 0. The 'Buffer 1' radio button is selected. The 'Increment LBA' checkbox is unchecked. The 'Inc. by' field is 00. The 'LBA MSB' field is 3. The 'LBA LSB' field is 4. The 'Repeat' field is 1. The 'Timeout' field is 0. The 'Stop on error' radio button is selected. The 'CDB Result' field is empty. The 'View Results' button is highlighted by a mouse cursor.

The User-Defined CDB functions of the SCSI toolbox allow the operator to construct an type of SCSI CDB and issue that CDB to the currently select SCSI TARGET. Any command created may also be added into a CDB file. Customized files of CDB's can be created and maintained.

Defining or Modifying a SCSI CDB

The following parameters define a SCSI CDB and are accessible/modifiable from the left side of the dialog box:

- Command name
- Individual bytes of the SCSI CDB
- Command length (6,10,12 bytes)
- Data Direction
- Data transfer length
- Data buffer (buffer 1 or 2)
- Command timeout value
- Command repeat count
- Increment information

Once a SCSI CDB has been defined that command can be issued to the currently selected TARGET, stored to the current CDB list file, or added to the script window.

Important Issues

It is possible to create an incorrect SCSI CDB! If you are not intending to create an illegal CDB, some common mistakes to watch out for are:

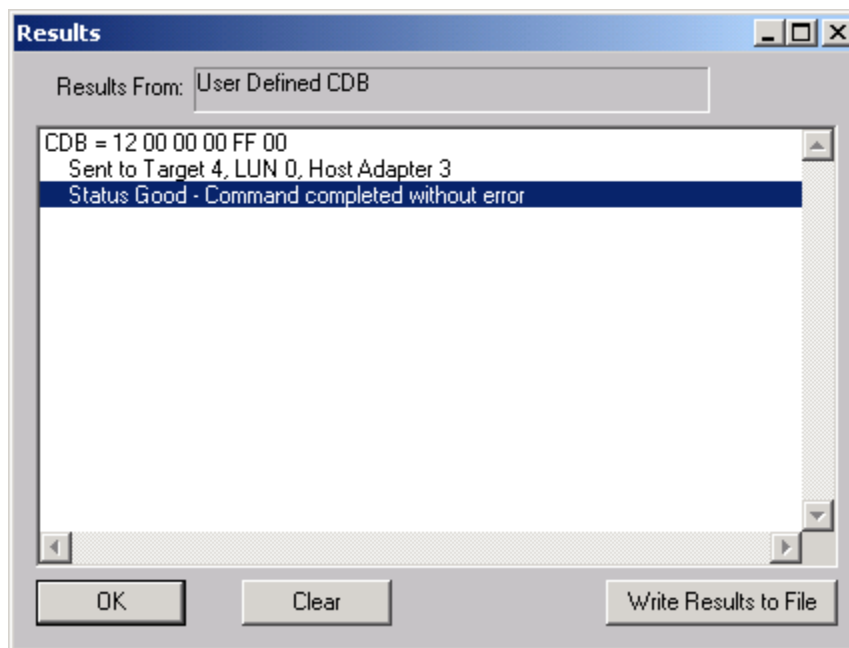
- Data direction incorrect
- Data length specified in CDB does not match Data Transfer Length

Incrementing Issues

It is possible to increment a field within the CDB each time the CDB is issued. This is commonly used to increment the block number specified in the CDB. Set the Increment portion of the dialog as follows to use the feature:

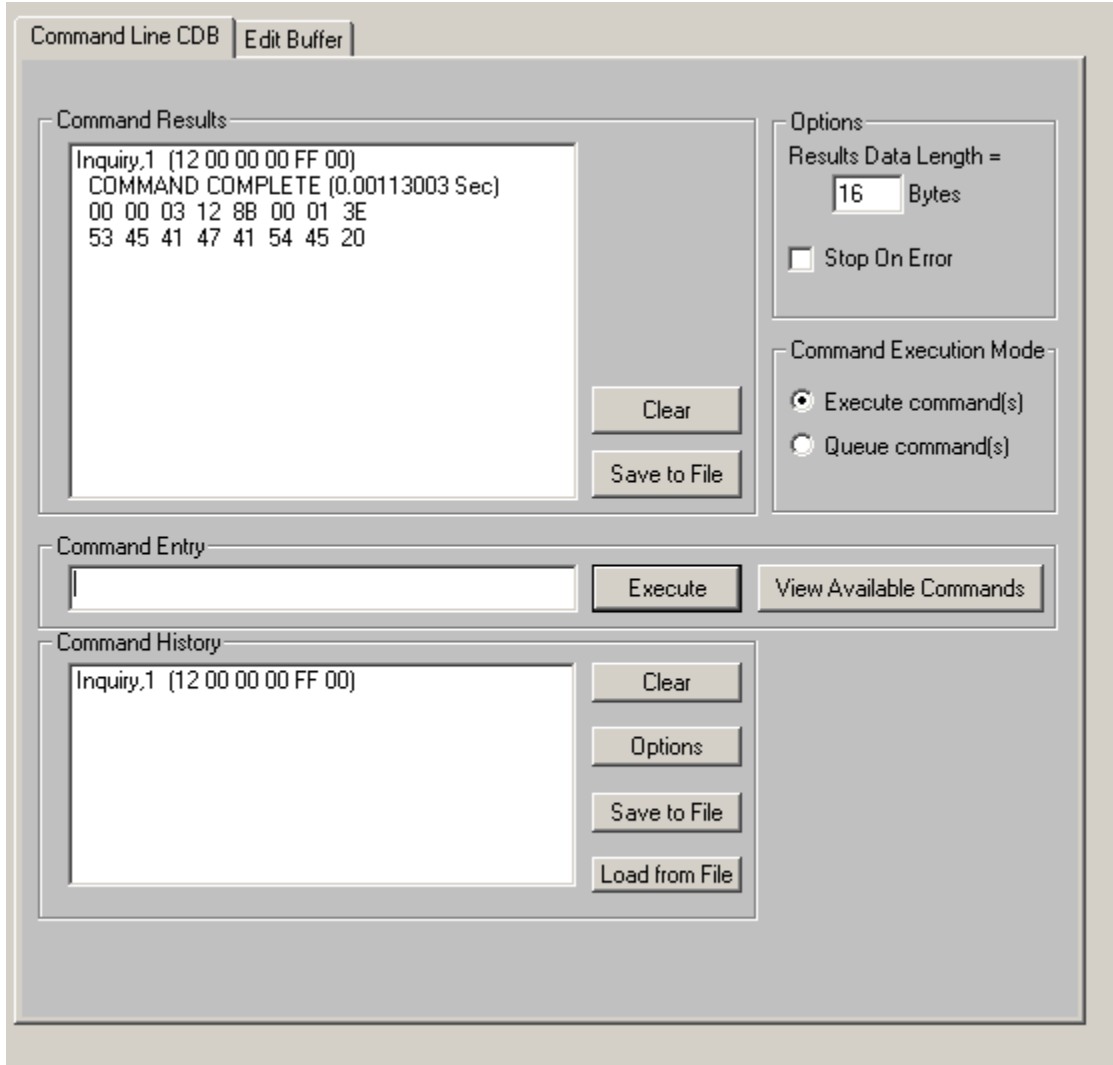
- Check the Increment LBA box
- Specify which CDB byte is the LSB of the SCSI CDB
- Specify which CDB byte is the MSB
- Specify how much you want to increment by

For instance, if you define a 10 byte READ EXTENDED CDB, byte 2 is the LSB and byte 5 is MSB of the Logical Block Address. In this example we will start reading a block 0 by setting CDB bytes 2 through 5 to zero (0). If you define this CDB to read 128 blocks per command (0x80 hex in byte 8 of the CDB) you will set the "increment by" field to 128. By setting the "CDB Repeat Count" field to 100, then clicking the "Send CDB" button, the READ EXTENDED CDB will be issued 100 times, and each time it is issued, 128 blocks of data will be read and the LBA will be incremented by 128.



After issuing a User-defined CDB you can click on the Display Results button to view the status of each CDB that has been sent. You can clear this display with the Clear button, or you can save the contents of the display to a disk file.

CDB Sequencer



This function lets you type in the name of a CDB on the command line. When you press *Enter*, the function looks for a matching CDB name in the COMMANDS.DAT file, and issues the CDB to the selected target.

For example, typing "INQ", followed by pressing *Enter*, matches with the INQUIRY CDB in the sample COMMANDS.DAT file and issues that CDB. Note that the first match found will be issued. For example, if you had two read commands defined, Read6 and Read10, simply typing "read" will match Read10, since that CDB is first in the file. The file is sorted alphabetically.

If a match is not found in the CDB file, an error message is returned. If you are using multiple CDB files, you must first rename (or copy) the desired file to COMMANDS.DAT.

You can use the "View Available Commands" button to display a list of all commands in the commands.dat file. This display can also be used to select a command from the list for execution.

Command Results Window

When the CDB is issued, the status of the command is shown in the top list box (Command Results).

This status shows:

- the bytes of the CDB that were sent
- the time spent executing the CDB,
- several bytes of data, depending on the Results Data Length setting
- whether the command completed successfully, or
- if the result is a check condition, the sense data is shown.

The contents of the Command results window can be erased or saved to a file.

Command History Window

The CDB issued is also recorded in the lower “Command History” window.

Commands in this window can be selected and re-issued using the “Execute” button.

Commands in this window can also be saved and/or reloaded from a file to facilitate reusing CDB sequences.

Options

The “Options” buttons allow choosing to execute commands upon pressing *Enter*, or Queuing the commands in the Command History window for batch execution.

This allows a series of CDB’s to be set up, then sent sequentially with no delays.

The “Stop on Error” checkbox will stop a multi-pass CDB on error if checked. If not checked, a multi-pass CDB will execute all passes, regardless of any error conditions.

Command line arguments

Command line arguments are available to specify the number of times to issue the cdb, and to change the values in bytes 1 and 2 of the cdb.

Multi-pass CDB’s are issued by typing in the CDB name, a comma, then the number of passes.

For example, “READ,1000” will issue 1000 reads.

To issue a READ EXTENDED cdb and set the FUA bit, you would enter READ_10,1, 80

These arguments tell the program to:

- issue the READ_10 cdb
- issue the cdb 1 time
- modify byte 1 to the hex value 0x80 (this sets the FUA bit in the cdb)

Continuous command execution

Entering a pass count of less than 0 will cause the cdb to be issued continuously, until the “Stop” button is clicked.

Results Data Length

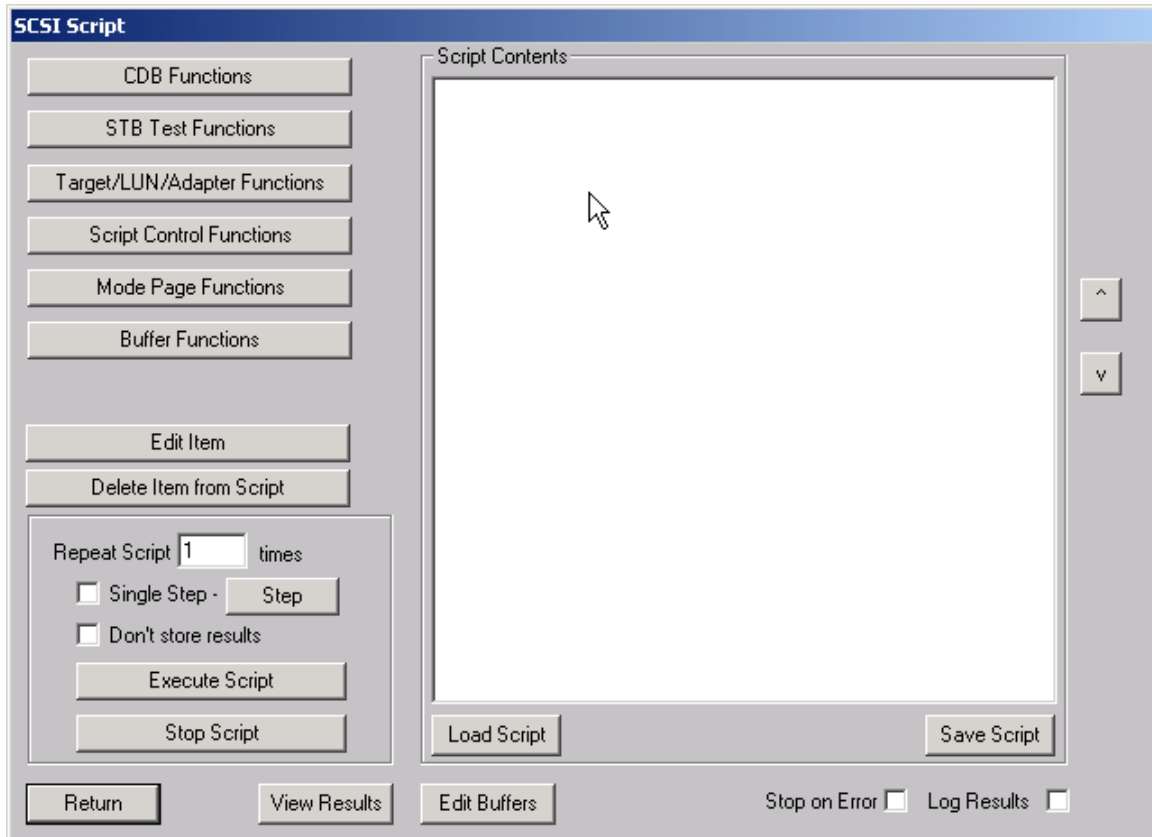
The “Results Data Length” allows you to specify how many bytes of data are displayed in the “Command Results” window. If a command transfers data, this many bytes will be displayed along with the command results. If the command terminates in a check condition, this many bytes of sense data will be displayed.

The entire data buffer can be viewed, modified, and saved/loaded from a file in the ***Edit Buffer*** tab page.

Re-issuing commands

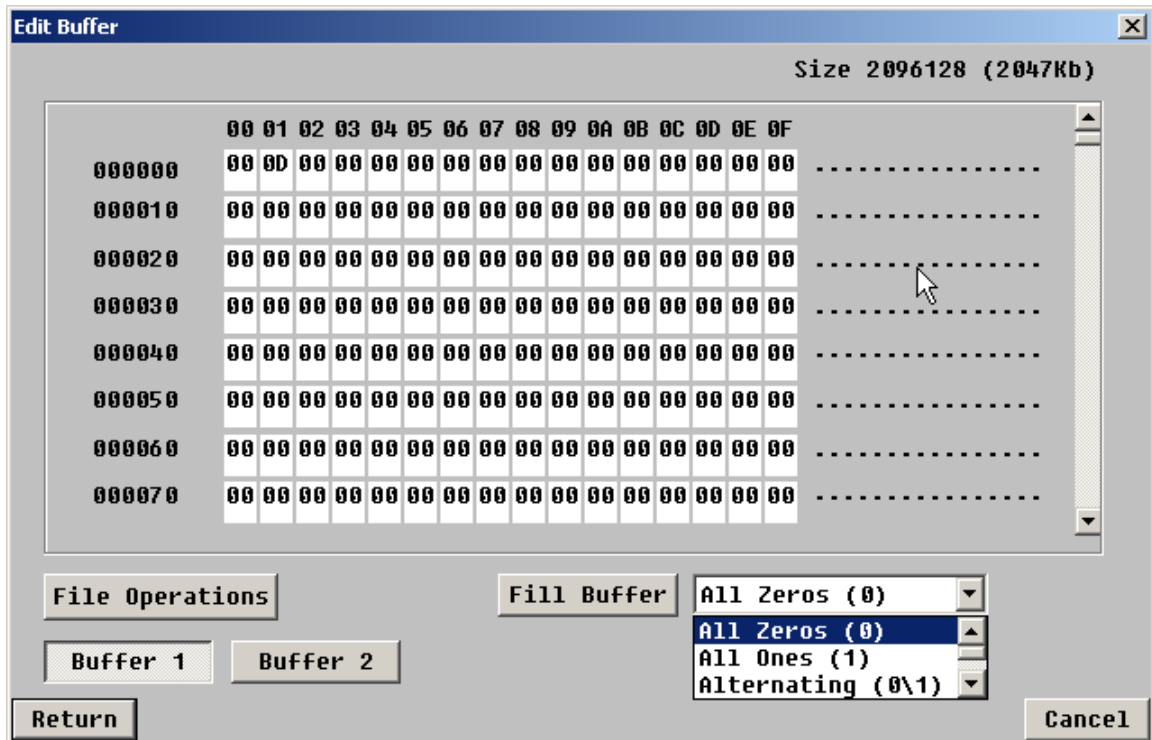
Any command or commands can be re-issued by selecting them (clicking on them) in the “Command History” window. Once the commands you want to reissue are highlighted, click the “Execute” button and they will be reissued.

Build/Run Script

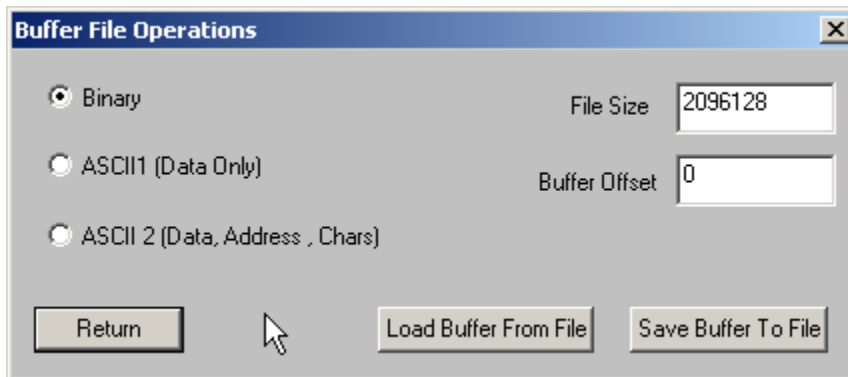


Edit Buffer 1/ Edit Buffer 2

The buffer functions allow the buffers used by the SCSI toolbox32 to be viewed, edited, saved to a disk file, or loaded from a disk file. The data in buffer 1 is used for all write tests. Buffer 1 will also contain all data read from the device.

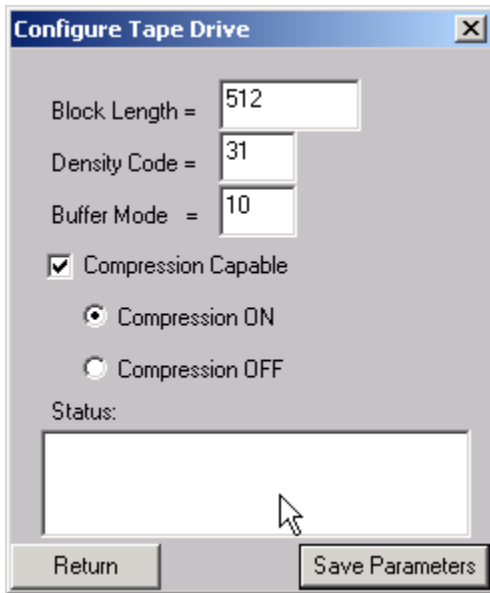


The drop box next to the Fill Buffer button allows you to choose a pattern to load into the buffer. Select the data pattern, then click the Fill Buffer button to load the data into the buffer. The File Operations button allows the buffer data to be saved to or loaded to a disk file.



User choice allow the data to be saved either as binary data, or as ASCII data with or without the hex address of the data. The File size specifies how may bytes of data to read or write. The Buffer Offset choice allows data to be loaded starting at a specific address of the buffer.

Configure TapeDrive



Configure Tape Drive

Block Length = 512

Density Code = 31

Buffer Mode = 10

Compression Capable

Compression ON

Compression OFF

Status:

Return Save Parameters

This command allows the user to set the standard variable parameters in the MODE SELECT parameter list header and block descriptor. The fields that may be assigned values are:

- Block Length
 - o Sets the block length field in the block descriptor, used to set block length for fixed block I/O block descriptor bytes 5-7. Setting this value to zero 0 sets the drive to write variable length records.
- Density Code
 - o Some tape drives will read and write different with different data densities and provide values in the block descriptor byte 0 for those different settings.
- Buffer Mode
 - o Assigns the value to parameter list header byte 2 to set buffer mode on or off

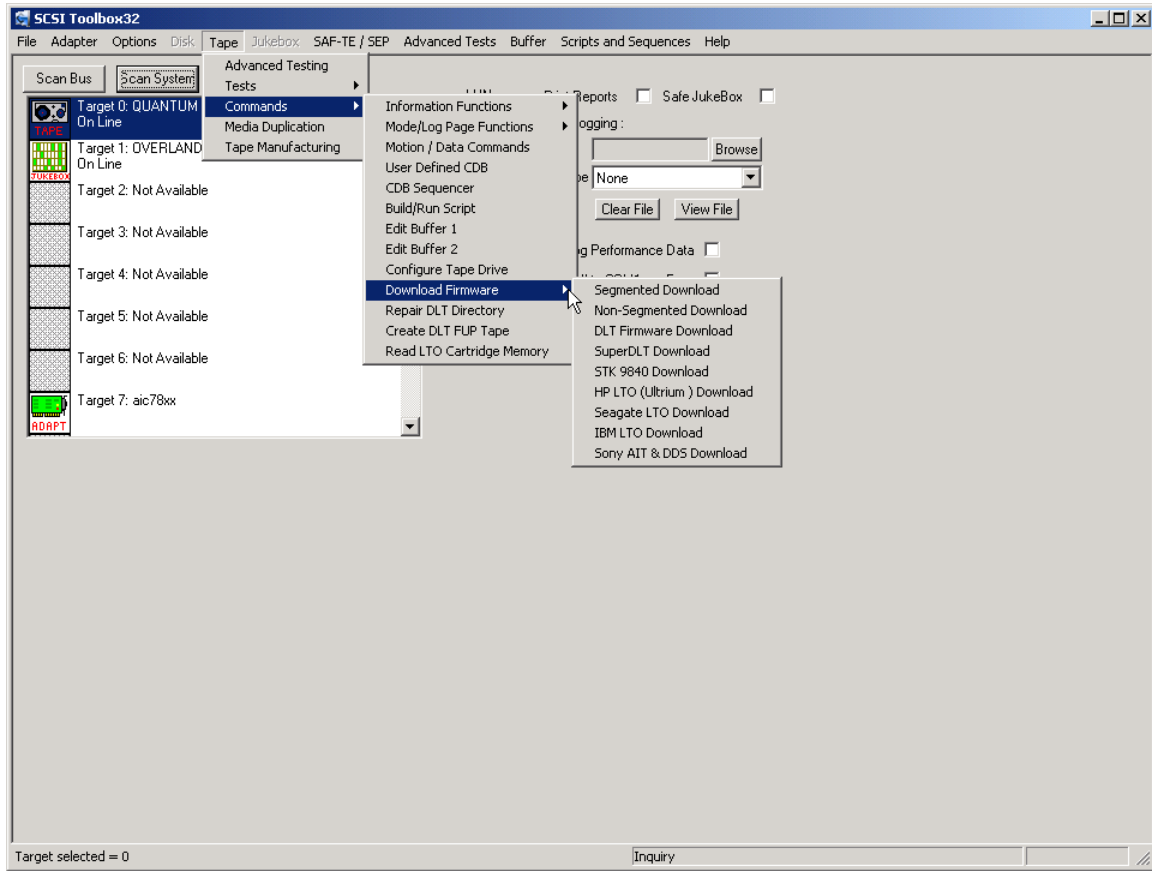
If the drive supports compression, the Compression Capable check box will be checked and the user can turn compression on or off with the radio button below it.

When the Save Parameters button is clicked the mode select command is issued to the tape drive and the results of that command are displayed in the box at the bottom of the window.

Note: Not all tape drives support all of these settings. The user must be familiar with the drive and have a SCSI manual available to determine support and valid values for these parameters.

Download Firmware

Several types of firmware download options are available as shown below:



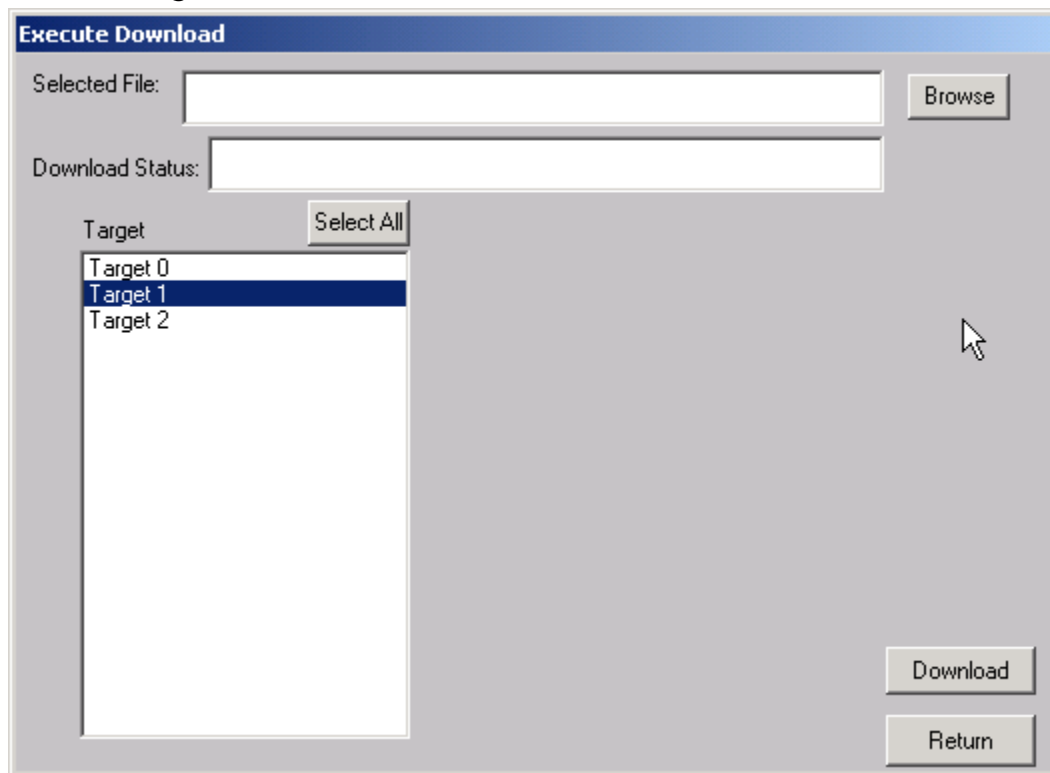
Segmented Download

The screenshot shows a dialog box titled "Execute Download". At the top, there is a blue title bar with the text "Execute Download". Below the title bar, there are two text input fields: "Selected File:" and "Download Status:". To the right of the "Selected File:" field is a "Browse" button. Below the "Download Status:" field, there is a "Target" list box containing three items: "Target 0", "Target 1", and "Target 2". The "Target 1" item is highlighted in blue. To the right of the "Target" list box is a "Select All" button. At the bottom right of the dialog box, there are two buttons: "Download" and "Return".

The Segmented firmware download command will download new firmware into the tape drive using the SCSI WRITE BUFFER command. The firmware will be downloaded in 32K segments. Use the Browse button to select the download file name.

If your tape drive does not support segmented downloads, use the Non-Segmented download command.

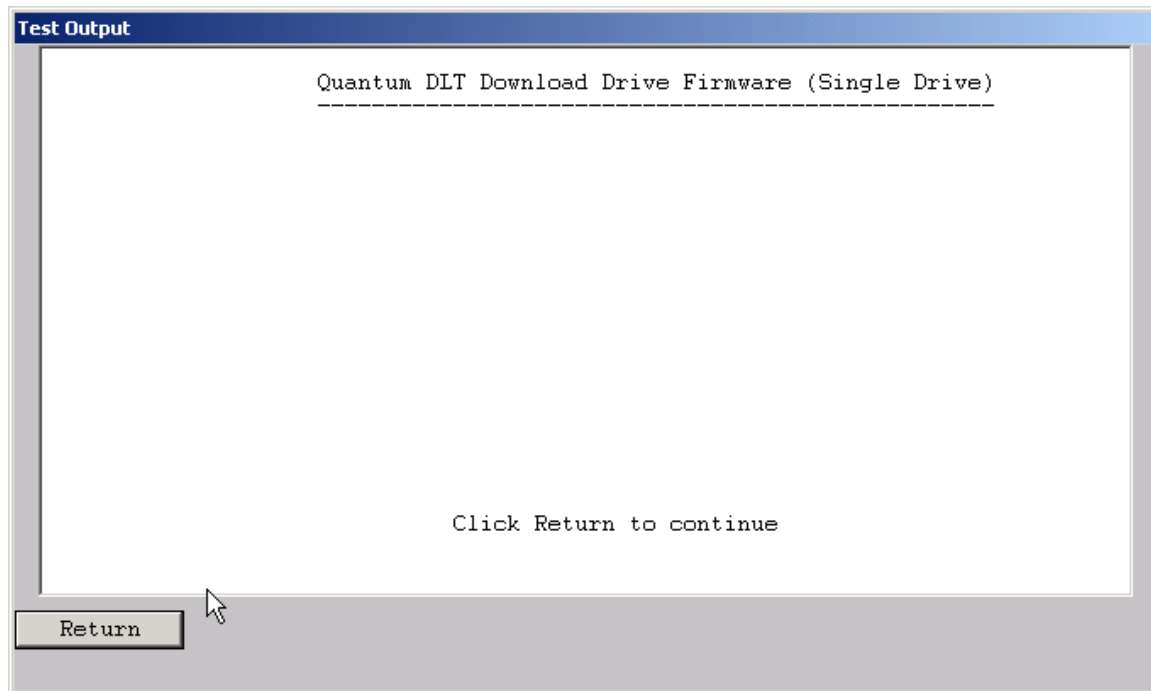
Non-Segmented Download



The Non-Segmented firmware download command will download new firmware into the tape drive using the SCSI WRITE BUFFER command. Use the Browse button to select the file to download into the drive.

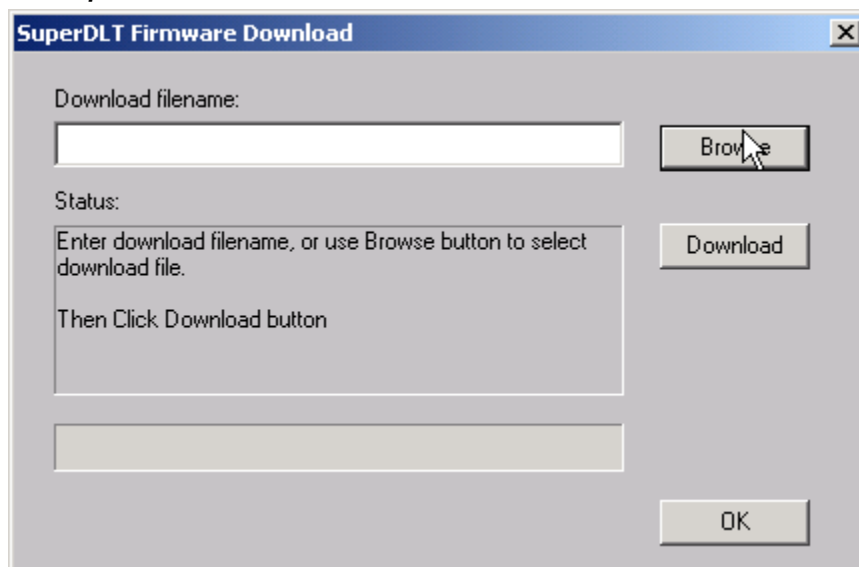
If the available buffer space is not large enough to hold the file the download will fail. The "About SCSI toolbox" choice of the "Help" menu displays the available buffer space.

DLT Firmware Download



This command will download new firmware into a DLT or DLT-compatible tape drive.

SuperDLT Download

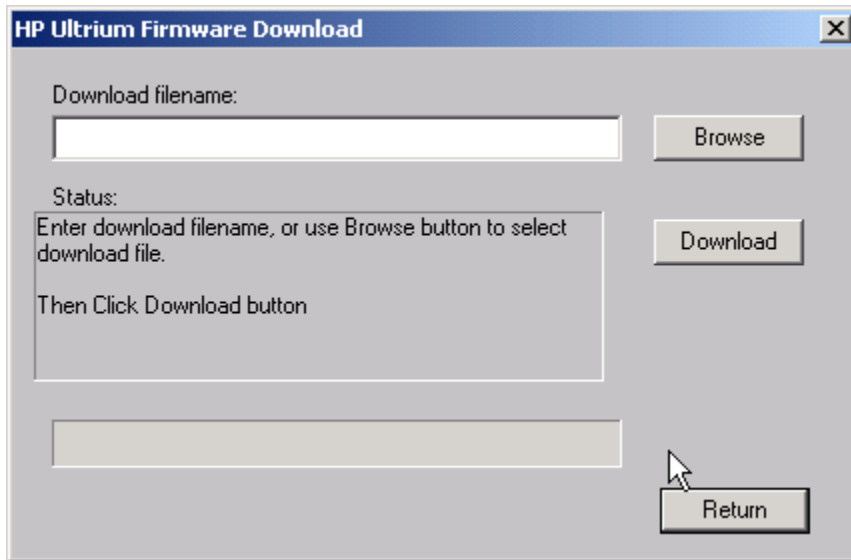


This command will download new firmware into a SuperDLT drive. Use the Browse button to select the download file, then click the Download button.

STK 9840 Download

This command will download new firmware into a STK 9840/9940 drive. Use the Browse button to select the download file, then click the Download button.

HP LTO Download

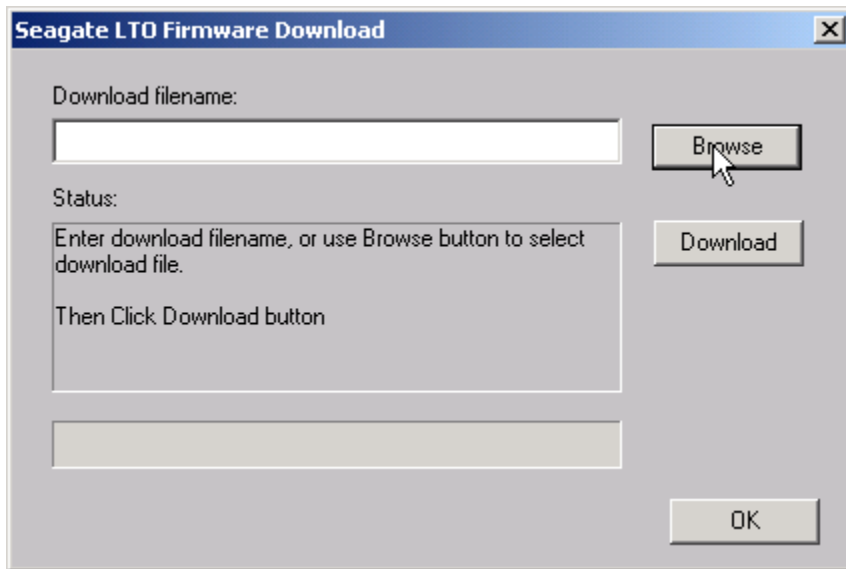


The screenshot shows a dialog box titled "HP Ultrium Firmware Download" with a close button (X) in the top right corner. The dialog contains the following elements:

- A label "Download filename:" followed by a text input field.
- A "Browse" button to the right of the input field.
- A "Status:" label above a text area containing the instructions: "Enter download filename, or use Browse button to select download file." and "Then Click Download button".
- A "Download" button to the right of the status text area.
- A horizontal progress bar at the bottom of the dialog.
- A "Return" button at the bottom right, with a mouse cursor hovering over it.

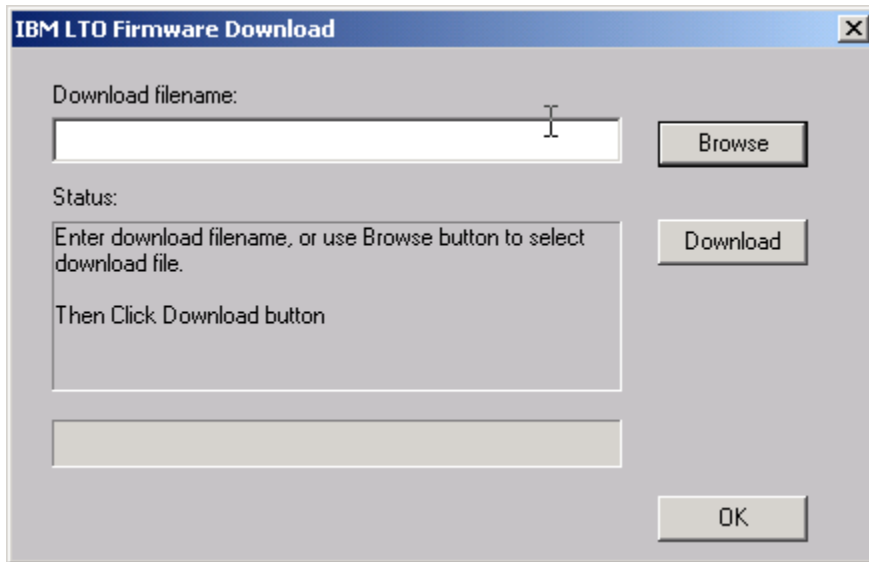
This command will download new firmware into a HP LTO (Ultrium) drive. Use the Browse button to select the download file, then click the Download button.

Seagate LTO Download



This command will download new firmware into a Seagate LTO drive. Use the Browse button to select the download file, then click the Download button.

IBM LTO Download

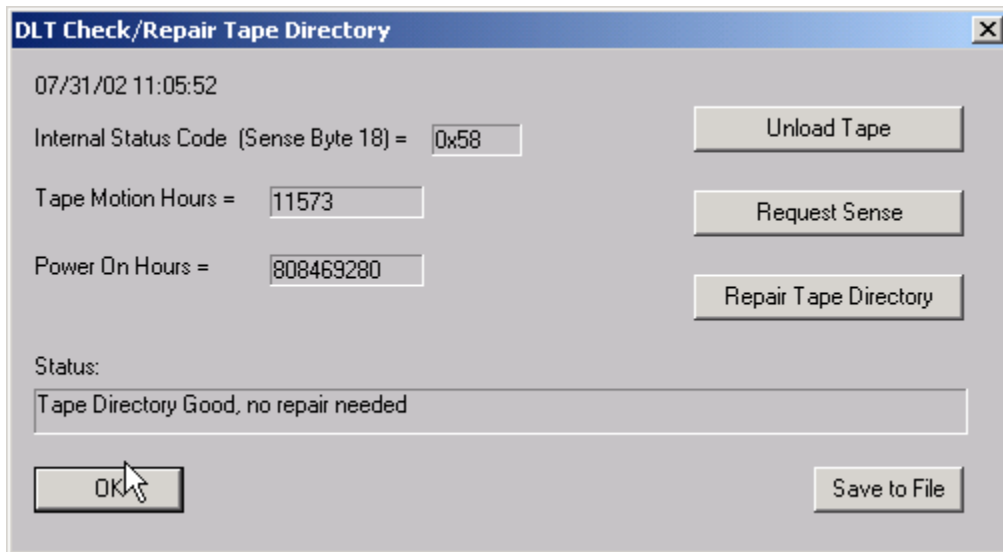


This command will download new firmware into a IBM LTO drive. Use the Browse button to select the download file, then click the Download button.

Sony AIT & DDS Download

This command will download new firmware into a Sony AIT and DDS drives. Use the Browse button to select the download file, then click the Download button.

Repair DLT Tape Directory



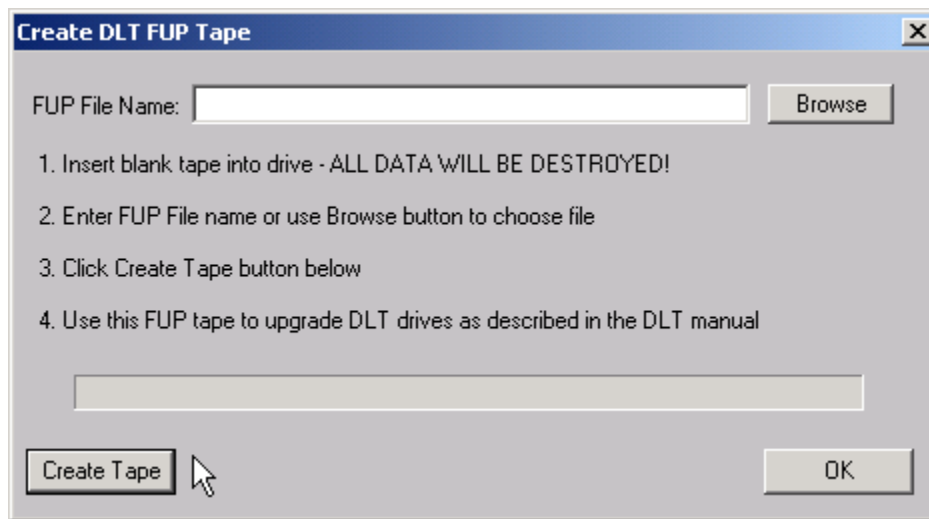
The screenshot shows a dialog box titled "DLT Check/Repair Tape Directory" with a close button (X) in the top right corner. The dialog displays the following information and controls:

- Timestamp: 07/31/02 11:05:52
- Internal Status Code (Sense Byte 18) =
- Tape Motion Hours =
- Power On Hours =
- Status:
-

DLT tapes contain a directory at the beginning of the tape. This directory can become damaged if the tape drive is powered down before the tape has been unloaded. This command will determine the integrity of the DLT directory, and if it is damaged it can repair or re-write the directory.

Note: If the directory needs repair it can take up to several hours for the operation to complete.

Create DLT FUP (Firmware Update) Tape



Create DLT FUP Tape [X]

FUP File Name:

1. Insert blank tape into drive - ALL DATA WILL BE DESTROYED!
2. Enter FUP File name or use Browse button to choose file
3. Click Create Tape button below
4. Use this FUP tape to upgrade DLT drives as described in the DLT manual

This command allows you to create an update tape from a firmware file. Use the Browse button to specify the firmware file name, then click the Create Tape button. You can then use this update tape to upgrade the firmware in other DLT tape drives.

LTO Cartridge Memory Display

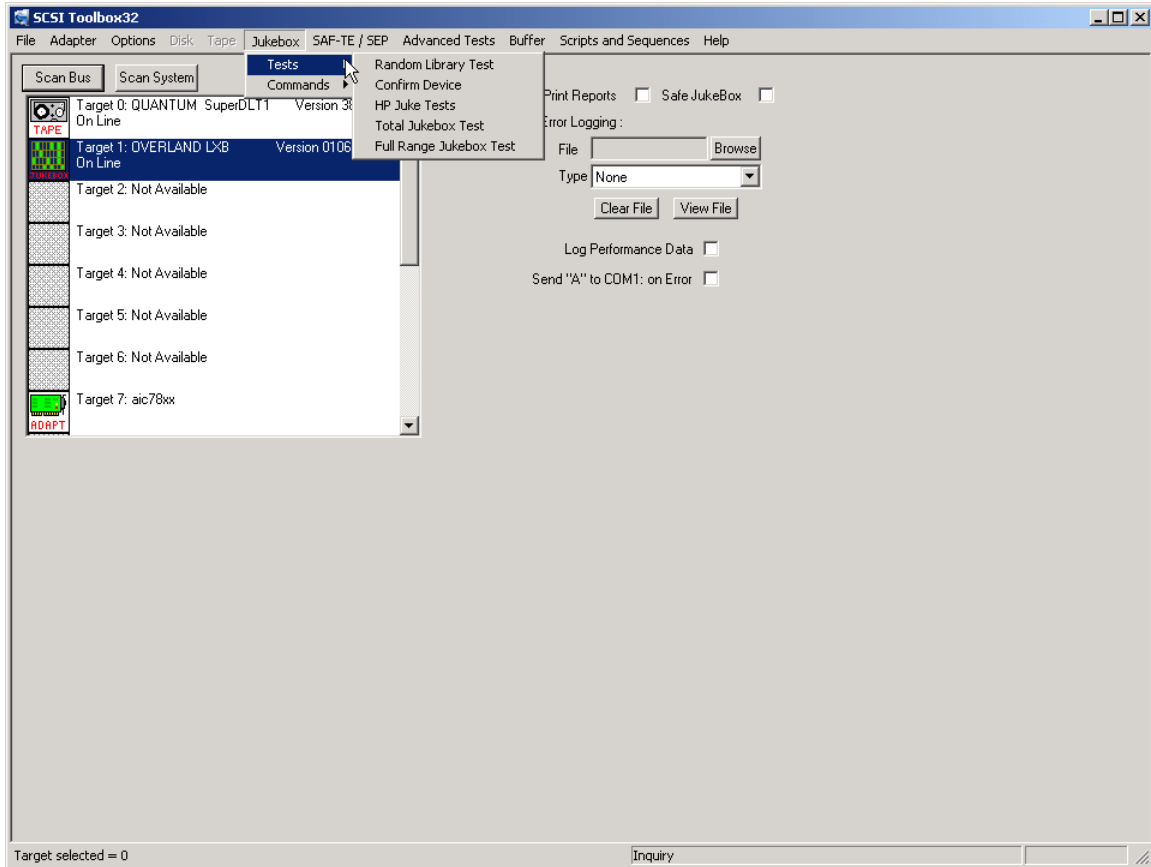
This command allows you to display the cartridge memory stored within LTO tape cartridges.

Usage	Medium Common	Device Common	Host Common
Medium Manufacturer =	HP	Medium Serial Number =	AAAH00005B
Medium Length =	16	Medium Width =	127
Assigning Organization =	LTO_CVE		
Medium Density Code =	64		
Medium Manufacture Date =	20000706		
MAM Capacity =	4096		
Medium Type =	0	Medium Type Information =	0

Usage	Medium Common	Device Common	Host Common
Remaining Capacity in Partition =	95367	Max Capacity in Partition =	95367
TapeAlert Flags =	0	Formatted Density Code =	64
Load Count =	57		
MAM Space Remaining =	1012	Assigning Organization =	LTO_CVE
Device Make/Serial # at Last Load =	IBM		
Device Make/Serial # at Last Load -1 =	IBM		
Device Make/Serial # at Last Load -2 =	IBM		
Device Make/Serial # at Last Load -3 =	IBM		
Total MB Written in Medium Life =	3933661		
Total MB Read in Medium Life =	-122226291		
Total MB Written in Current/Last Load =	0		
Total MB Read in Current/Last Load =	0		

Jukebox Tests

The following tests are available via the Jukebox Test menu:



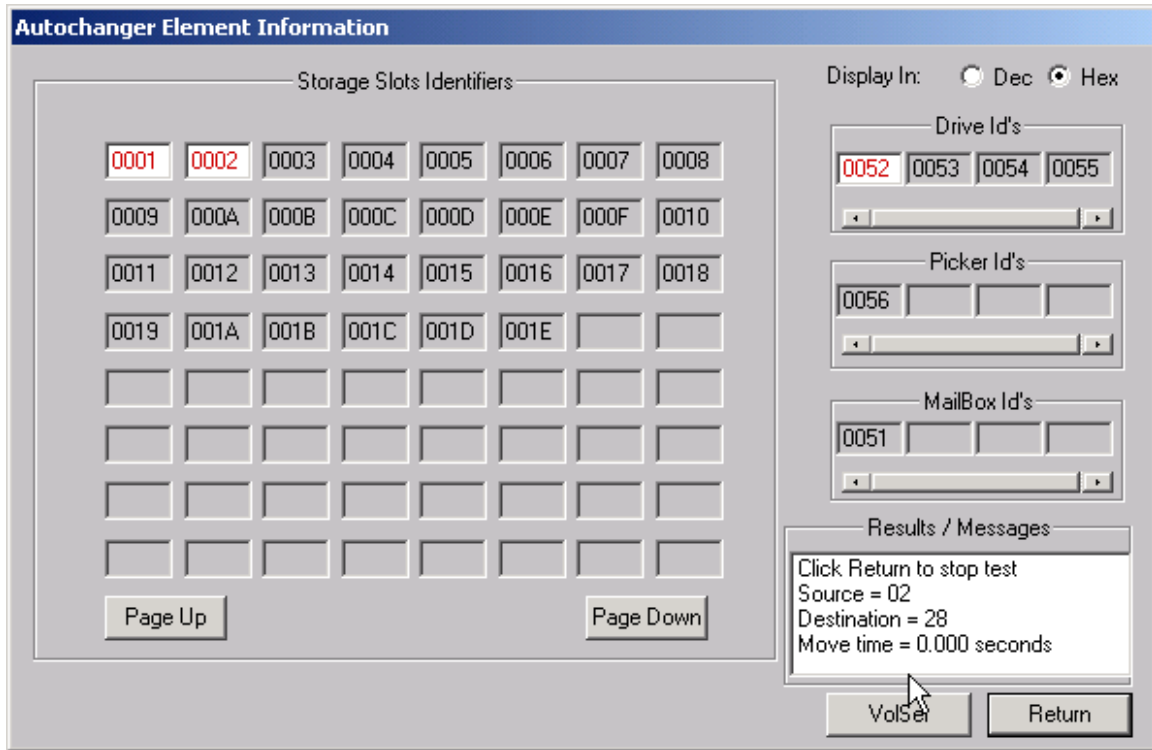
The jukebox module uses a display screen to show the configuration of the jukebox under test. The eight by eight matrix represents the storage elements of the jukebox, allowing the display of up to 64 storage elements at a time. If a storage element is present, its storage element address is displayed. If a given storage element does not exist on the jukebox under test, that position will be blank. If there is media in a particular storage element, its address will be displayed in a highlighted color. This allows you to see the addresses of each storage element and see if media is present in any of the storage elements. "PAGE UP" and "PAGE DOWN" buttons are provided to scroll the display across the valid storage element range.

This screen also displays the Media Transport Elements (the jukebox "picker"). The same display conventions are used here. The element address of the element will be displayed, and if there is media in the element the address will be displayed in highlighted color. Please note that it is possible to have a jukebox that does not have an Import/Export element. Horizontal scroll bars are provided for devices which have more elements than are displayed at one time.

Data Transfer Elements (the drives in the jukebox) are displayed in the same manner. Please note that the drive element address is not necessarily the same as the drive SCSI address.

This display is seen when using the Initialize Element Status, Read Element Status, Position to Element, and Move Media commands.

Random Library Test



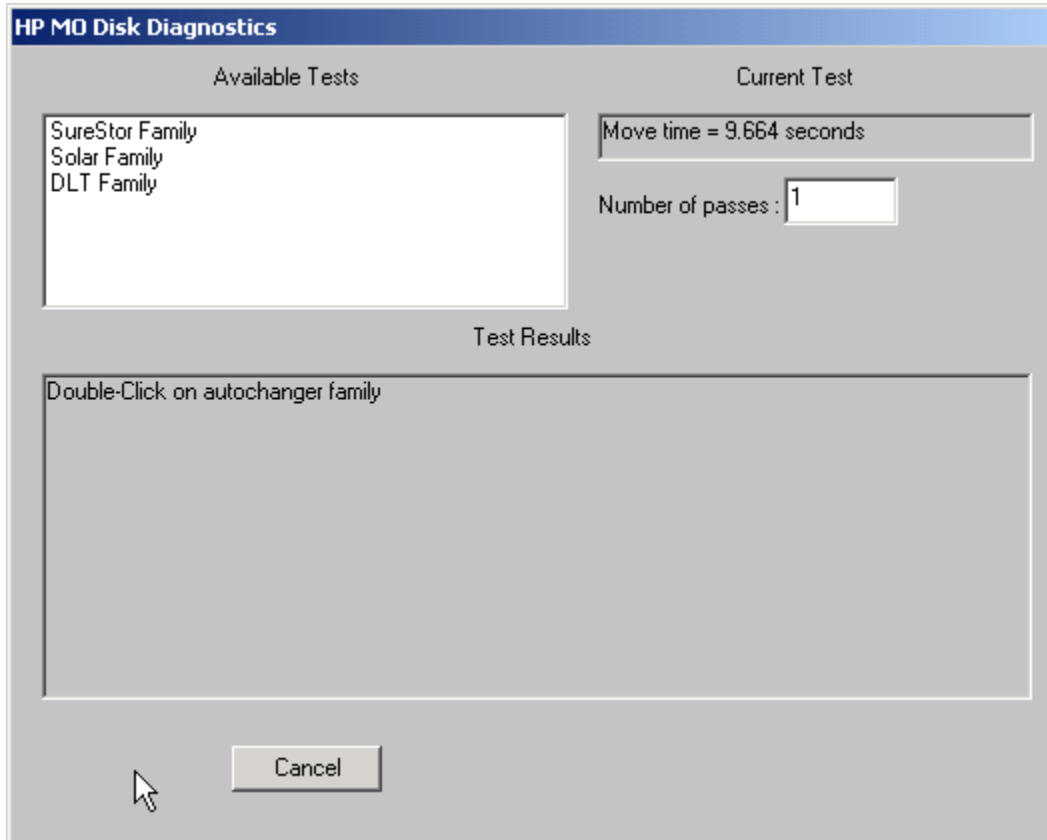
This test will move a piece of media randomly between the jukebox storage elements. It requires at least one empty storage element, and at least one storage element with media in it. As the test runs, it will randomly choose a full storage element and randomly choose an empty storage element. Then it will execute a MOVE MEDIUM command from the full element to the empty element. The Source and Destination storage elements will be displayed at the bottom of the screen. This test will run until interrupted.

Confirm Device



This choice will display pertinent SCSI information about the jukebox, such as firmware revisions, etc.

HP Juke Tests



SureStore Family Tests
Solar Family Tests

Choosing one of these selections will display a menu of choices similar to the dosdass2 diagnostic menu. Within this menu, a diagnostic may be specified and the number of test passes may be specified.

This menu will vary depending upon which HP autochanger is being tested. As the tests run, the current loop count is displayed and decremented as each pass completes.

Total Jukebox Test

Test	Drive	Slot Element	Picker Element	Unload	SCSI Address	Moves	Status
<input checked="" type="checkbox"/>	82	Random	86	<input type="checkbox"/>			
<input checked="" type="checkbox"/>	83	Random	86	<input type="checkbox"/>			
<input checked="" type="checkbox"/>	84	Random	86	<input type="checkbox"/>			
<input checked="" type="checkbox"/>	85	Random	86	<input type="checkbox"/>			
<input type="checkbox"/>				<input type="checkbox"/>			
<input type="checkbox"/>				<input type="checkbox"/>			
<input type="checkbox"/>				<input type="checkbox"/>			

Delay between moves: Number of passes: Stop on error

The Total Jukebox test is designed to stress moving media between storage elements and drives. This is the most common failure point in jukeboxes.

The element address of the drive is shown on the right. Click the "TEST" box to select each drive to test. Use the Slot Element and Picker Element choices to determine where the tape will be moved FROM.

Check the Unload box if the tape drive needs an implicit unload command sent to it before executing a move from the drive. In this case, specify the SCSI address of the tape drive in the SCSI address field. If your jukebox can execute a move from a loaded tape drive without needing an explicit unload leave the Unload box unchecked.

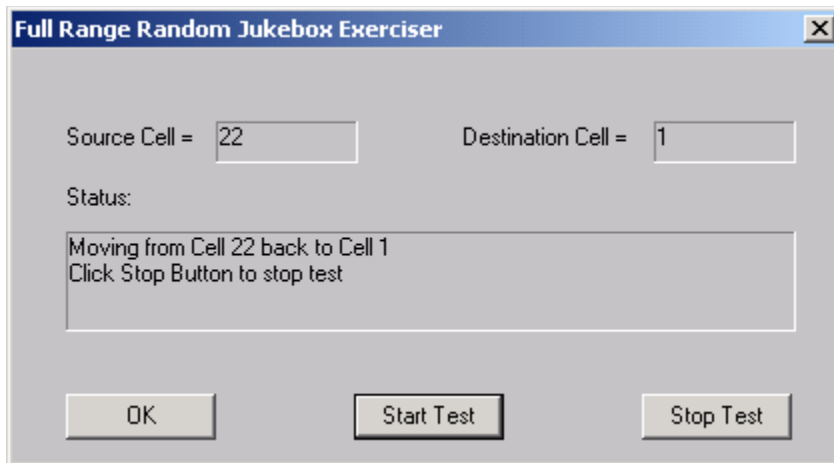
You may specify an amount of time to delay between moves with the Delay between moves field. Some jukeboxes require some "settling time" between moves.

Specify the number of moves in the Number of Passes field.

Click the Start box to begin the test.

JukeBox Full Range Test

This test will include moves from any location within the jukebox



The image shows a dialog box titled "Full Range Random Jukebox Exerciser". It contains two input fields: "Source Cell =" with the value "22" and "Destination Cell =" with the value "1". Below these is a "Status:" label and a text area containing the message "Moving from Cell 22 back to Cell 1" and "Click Stop Button to stop test". At the bottom, there are three buttons: "OK", "Start Test", and "Stop Test".

Full Range Random Jukebox Exerciser

Source Cell = 22 Destination Cell = 1

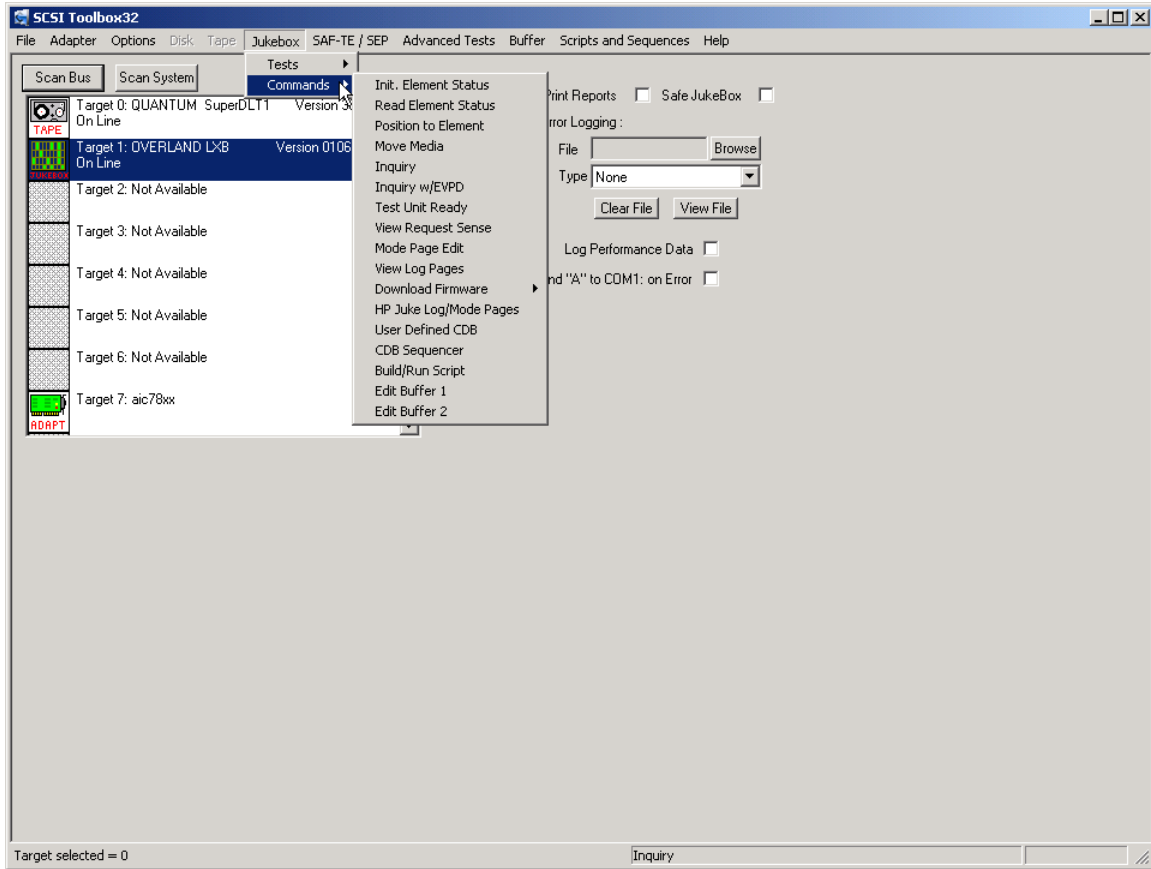
Status:

Moving from Cell 22 back to Cell 1
Click Stop Button to stop test

OK Start Test Stop Test

Jukebox Commands

The following commands are available via the Jukebox Commands menu:



Initialize Element Status

Autochanger Element Information

Storage Slots Identifiers

0001	0002	0003	0004	0005	0006	0007	0008
0009	000A	000B	000C	000D	000E	000F	0010
0011	0012	0013	0014	0015	0016	0017	0018
0019	001A	001B	001C	001D	001E		

Page Up Page Down

Display In: Dec Hex

Drive Id's

0052	0053	0054	0055
------	------	------	------

Picker Id's

0056			
------	--	--	--

MailBox Id's

0051			
------	--	--	--

Results / Messages

VolSer Return

This command issues a SCSI INITIALIZE ELEMENT STATUS command to the jukebox. This command causes the jukebox to check its configuration and determine the full or empty status of all of its elements. It is good practice to ALWAYS run this command before running any other commands or tests.

Read Element Status

Autochanger Element Information

Storage Slots Identifiers

0001	0002	0003	0004	0005	0006	0007	0008
0009	000A	000B	000C	000D	000E	000F	0010
0011	0012	0013	0014	0015	0016	0017	0018
0019	001A	001B	001C	001D	001E		

Page Up Page Down

Destination Id: 0 Picker Id: 56 Flip Media Position VolSer Return

Display In: Dec Hex

Drive Id's: 0052 0053 0054 0055

Picker Id's: 0056

MailBox Id's: 0051

Results / Messages

This command will show the status of each of the jukebox elements, as described at the beginning of this chapter.

Position to Element

Autochanger Element Information

Storage Slots Identifiers

0001	0002	0003	0004	0005	0006	0007	0008
0009	000A	000B	000C	000D	000E	000F	0010
0011	0012	0013	0014	0015	0016	0017	0018
0019	001A	001B	001C	001D	001E		

Page Up Page Down

Display In: Dec Hex

Drive Id's

0052	0053	0054	0055
------	------	------	------

Picker Id's

0056			
------	--	--	--

MailBox Id's

0051			
------	--	--	--

Results / Messages

Destination Id: Picker Id:

Flip Media

Position VolSer Return

This command will allow you to position a transport element at a destination element. Assuming valid addresses, the transport element will be positioned to the destination element address.

Note: Not all jukeboxes support this command.

Move Media

Autochanger Element Information

Storage Slots Identifiers

0001	0002	0003	0004	0005	0006	0007	0008
0009	000A	000B	000C	000D	000E	000F	0010
0011	0012	0013	0014	0015	0016	0017	0018
0019	001A	001B	001C	001D	001E		

Page Up Page Down

Display In: Dec Hex

Drive Id's: 0052 0053 0054 0055

Picker Id's: 0056

MailBox Id's: 0051

Results / Messages

Source Id: 0 Destination Id: 0 Picker Id: 56 Flip Media Move Media VolSer Return

This command will prompt you to enter the source, destination, and transport address, and then will attempt to move the piece of media from the source element address to the destination element address via the transport element address. This assumes the following:

1. You entered a valid source, destination, and transport address (in hexadecimal or decimal, depending on the "Display In" setting).
2. The source element you entered contains media.
3. The destination element you entered is empty.
4. The transport element is empty. If any of the above conditions are not met, the command will fail, probably with an ILLEGAL REQUEST error. You can use the View Most Recent Sense command in the Command MENU to examine the request sense data in case of an error.

You may also use the drag and drop method to pick up a source and drag to a destination. When you drop on the destination the values will be filled in on the edit boxes. Simply confirm that the desired transport address is selected and press the move button to execute the move.

View Request Sense

Test Output

Display Most Recent Request Sense Data

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
70	00	05	00	00	00	00	0A	00	00	00	00	24	00	00	C0
00	02	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

(Click Return to Continue)

Return

This command will display the most recent request sense data.

Mode Page Edit

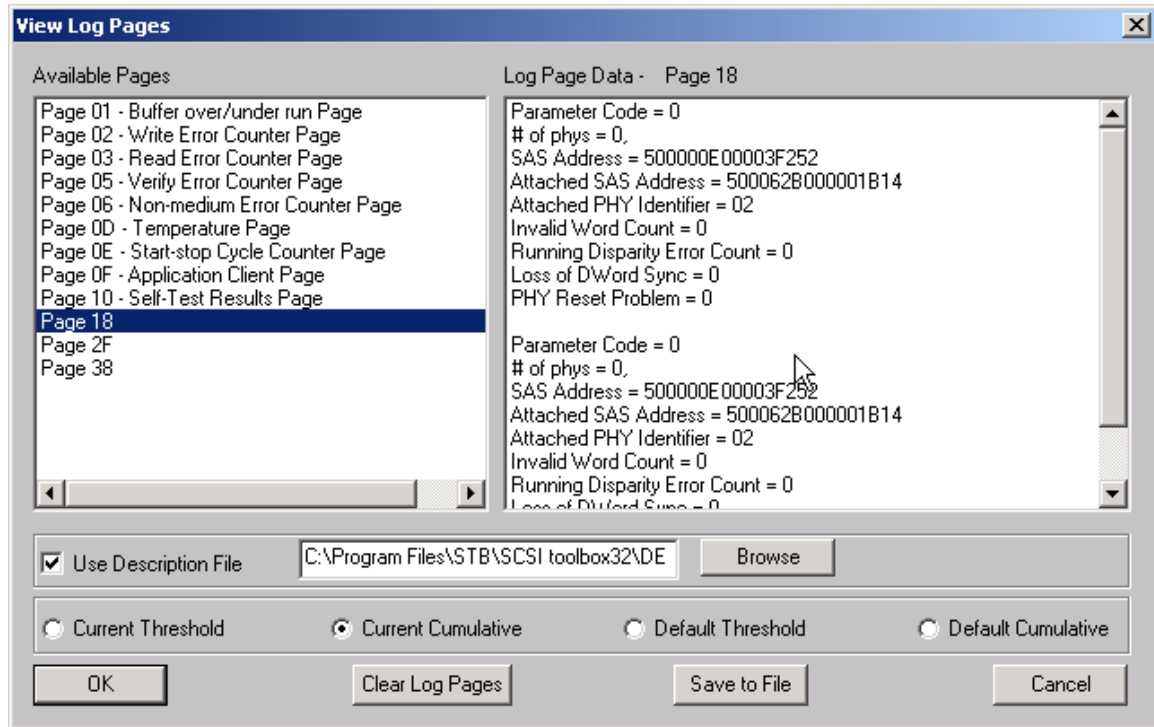
Custom Mode Page

Block Header	Block Descriptor	SP Bit
<input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/>	<input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/>	<input type="checkbox"/>
0 1 2 3	4 5 6 7 8 9 10 11	
Page Data		
Default	<input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/>	
Mask	<input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/>	
Current	<input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/>	
	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	
Default	<input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/>	
Mask	<input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/>	
Current	<input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/>	
	16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	
Page #	<input type="text" value="0C"/>	Current Byte <input type="text"/>

This command allows you to select one of several standard mode pages and view and or change the values by their acromymns. For example selecting the Error Recovery page you can view the setting for the PER bit and change it.

This is a "Hex Edit" selection which displays a dialog box with which you can select the mode page number, and issue a mode sense for that page. The data is then displayed as a series of hexadecimal byte values with the changeable mask value for that byte. You may then edit any byte to specify a new value then issue a mode select to store the new values to the device.

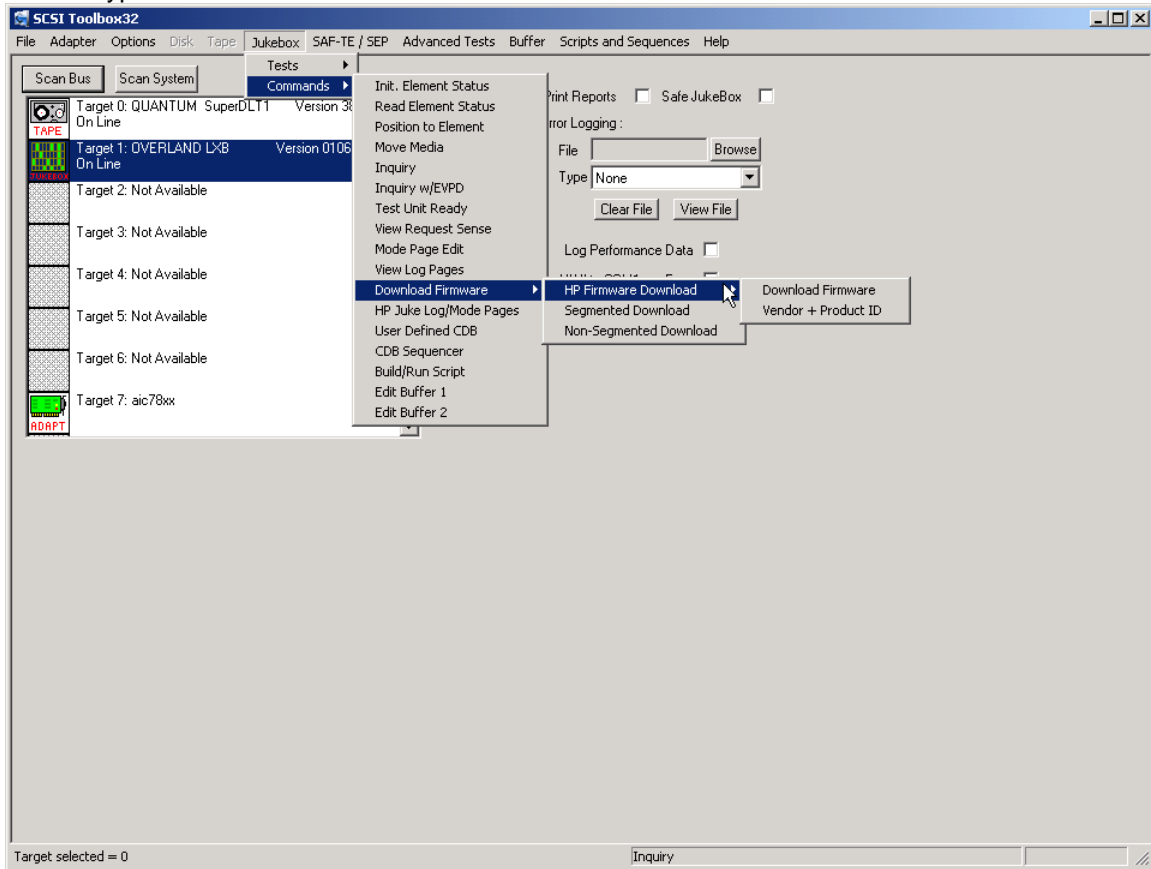
View Log Pages



This command allows you to select the display of either standard mode pages or standard log pages, you can then select from the list of pages and the data will be retrieved from the drive and interpreted into English.

Download Firmware

Several types of firmware download methods are available as shown below:



HP Firmware Download

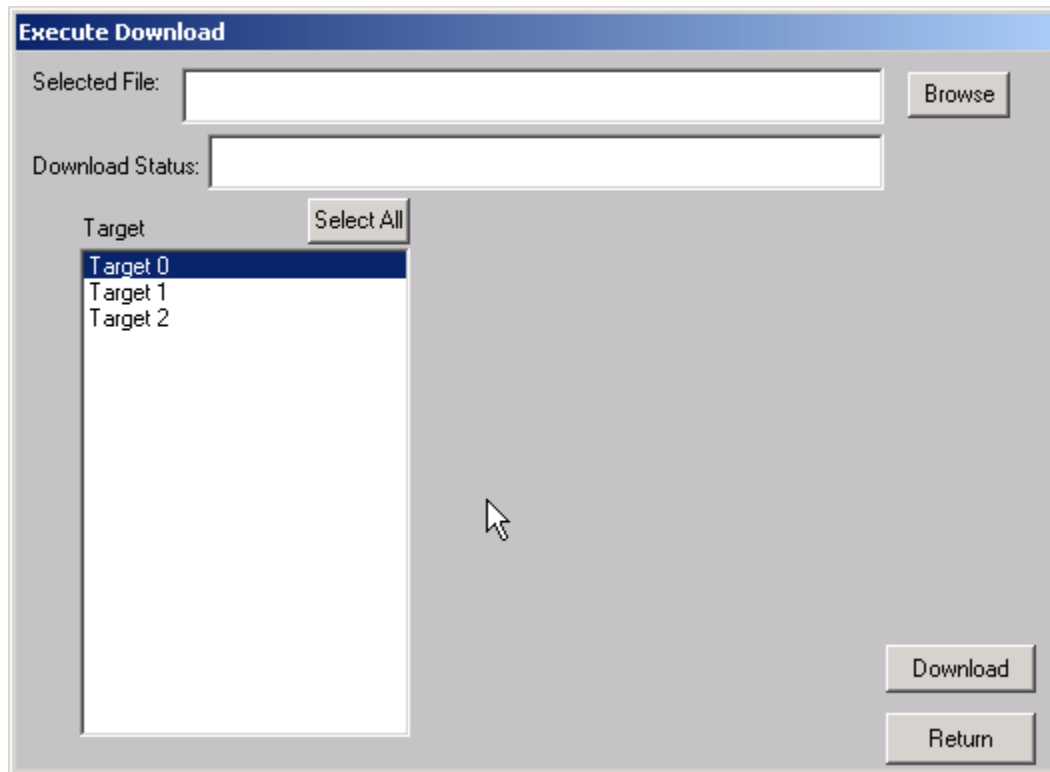
Download Firmware

This command duplicates the functionality of hp-dload.exe. The user is prompted to specify the first download filename. The program then reads that file sends it to the drive then increments the extension by 1 and looks for a file which matches. If it finds a matching file it sends that file to the drive. This process will continue until the file is not found. This allows you to download a standard set of file with names like: File.001 File.002, ... File.008.

Download Vendor & Product ID

This selection is used to download both firmware and Vendor ID and Product ID data to the autochanger. The operator will be asked to specify the filename of the download file, then the download will be executed. The file names conform to the HP standard (i.e., a:\ac_base\dnld_001).

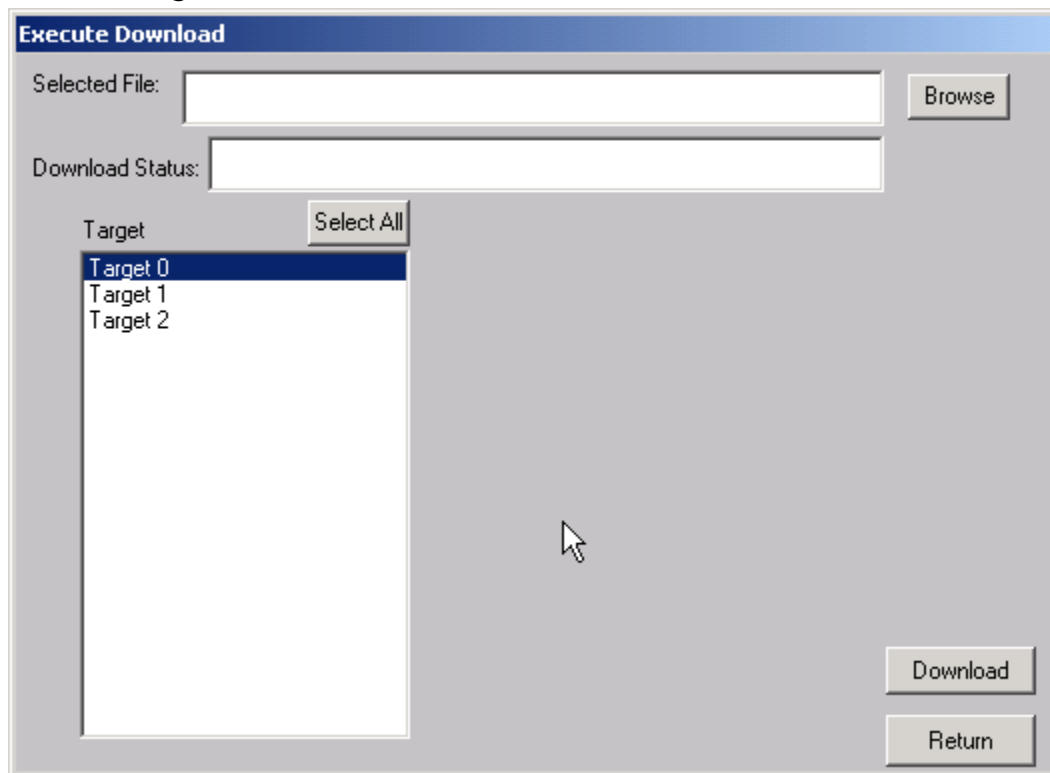
Segmented Download



The Segmented firmware download command will download new firmware into the jukebox using the SCSI WRITE BUFFER command. The firmware will be downloaded in 32K segments. Use the Browse button to select the download file name.

If your jukebox does not support segmented downloads, use the Non-Segmented download command.

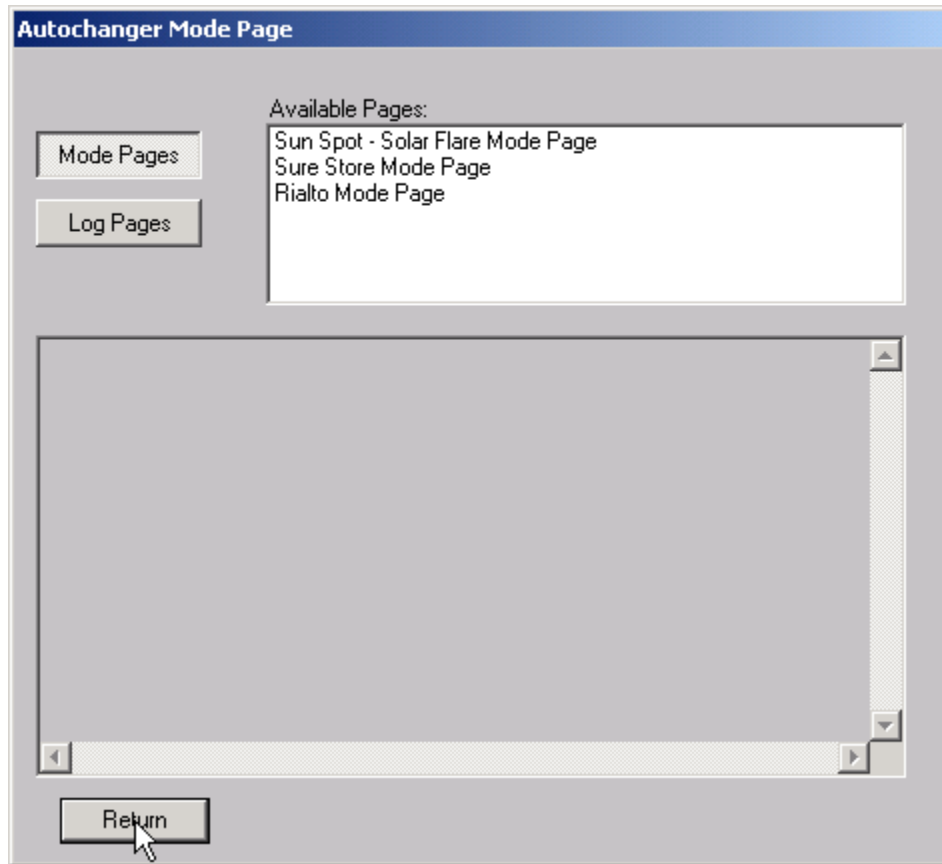
Non-Segmented Download



The Non-Segmented firmware download command will download new firmware into the jukebox using the SCSI WRITE BUFFER command. Use the Browse button to select the file to download into the drive.

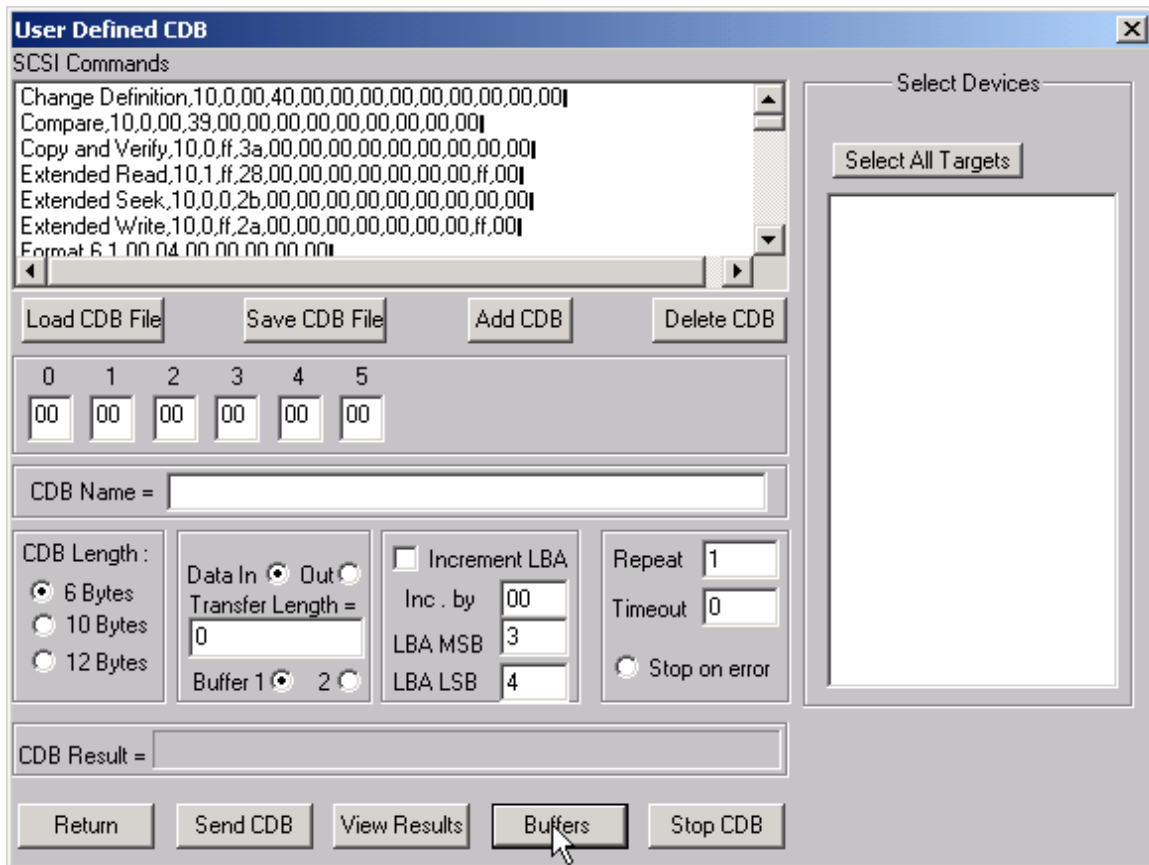
If the available buffer space is not large enough to hold the file the download will fail. The "About SCSI toolbox" choice of the "Help" menu displays the available buffer space.

HP Jukebox Mode Pages



This command allows you to select the display of either standard mode pages or standard log pages, you can then select from the list of pages and the data will be retrieved from the drive and interpreted into English.

User Defined CDB



The User-Defined CDB functions of the SCSI toolbox allow the operator to construct an type of SCSI CDB and issue that CDB to the currently select SCSI TARGET. Any command created may also be added into a CDB file. Customized files of CDB's can be created and maintained.

Defining or Modifying a SCSI CDB

The following parameters define a SCSI CDB and are accessible/modifiable from the left side of the dialog box:

- Command name
- Individual bytes of the SCSI CDB
- Command length (6,10,12 bytes)
- Data Direction
- Data transfer length
- Data buffer (buffer 1 or 2)
- Command timeout value
- Command repeat count
- Increment information

Once a SCSI CDB has been defined that command can be issued to the currently selected TARGET, stored to the current CDB list file, or added to the script window.

Important Issues

It is possible to create an incorrect SCSI CDB! If you are not intending to create an illegal CDB, some common mistakes to watch out for are:

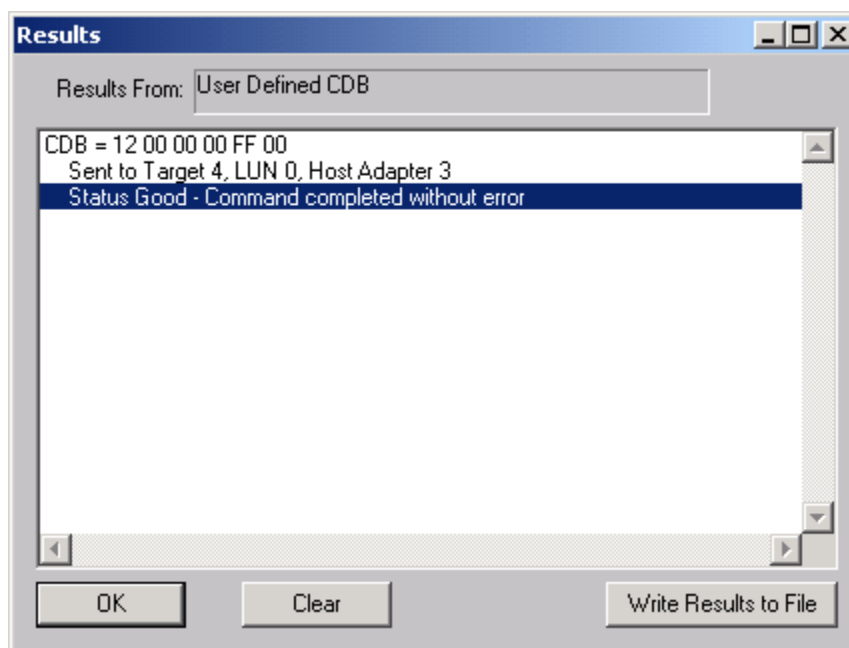
- Data direction incorrect
- Data length specified in CDB does not match Data Transfer Length

Incrementing Issues

It is possible to increment a field within the CDB each time the CDB is issued. This is commonly used to increment the block number specified in the CDB. Set the Increment portion of the dialog as follows to use the feature:

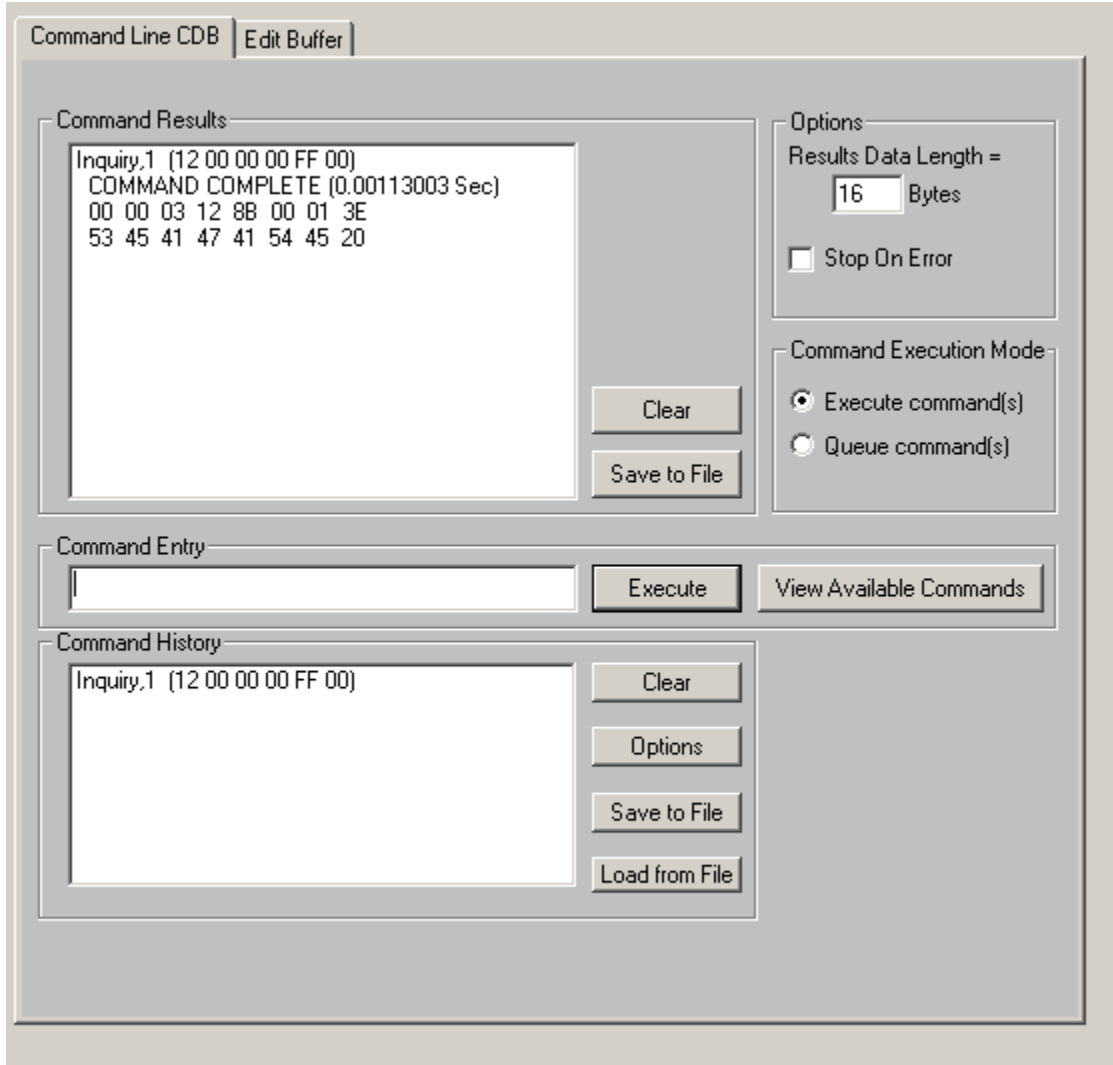
- Check the Increment LBA box
- Specify which CDB byte is the LSB of the SCSI CDB
- Specify which CDB byte is the MSB
- Specify how much you want to increment by

For instance, if you define a 10 byte READ EXTENDED CDB, byte 2 is the LSB and byte 5 is MSB of the Logical Block Address. In this example we will start reading a block 0 by setting CDB bytes 2 through 5 to zero (0). If you define this CDB to read 128 blocks per command (0x80 hex in byte 8 of the CDB) you will set the "increment by" field to 128. By setting the "CDB Repeat Count" field to 100, then clicking the "Send CDB" button, the READ EXTENDED CDB will be issued 100 times, and each time it is issued, 128 blocks of data will be read and the LBA will be incremented by 128.



After issuing a User-defined CDB you can click on the Display Results button to view the status of each CDB that has been sent. You can clear this display with the Clear button, or you can save the contents of the display to a disk file.

CDB Sequencer



This function lets you type in the name of a CDB on the command line. When you press *Enter*, the function looks for a matching CDB name in the COMMANDS.DAT file, and issues the CDB to the selected target.

For example, typing "INQ", followed by pressing *Enter*, matches with the INQUIRY CDB in the sample COMMANDS.DAT file and issues that CDB. Note that the first match found will be issued. For example, if you had two read commands defined, Read6 and Read10, simply typing "read" will match Read10, since that CDB is first in the file. The file is sorted alphabetically.

If a match is not found in the CDB file, an error message is returned. If you are using multiple CDB files, you must first rename (or copy) the desired file to COMMANDS.DAT.

You can use the "View Available Commands" button to display a list of all commands in the commands.dat file. This display can also be used to select a command from the list for execution.

Command Results Window

When the CDB is issued, the status of the command is shown in the top list box (Command Results).

This status shows:

- the bytes of the CDB that were sent
- the time spent executing the CDB,
- several bytes of data, depending on the Results Data Length setting
- whether the command completed successfully, or
- if the result is a check condition, the sense data is shown.

The contents of the Command results window can be erased or saved to a file.

Command History Window

The CDB issued is also recorded in the lower “Command History” window.

Commands in this window can be selected and re-issued using the “Execute” button.

Commands in this window can also be saved and/or reloaded from a file to facilitate reusing CDB sequences.

Options

The “Options” buttons allow choosing to execute commands upon pressing *Enter*, or Queuing the commands in the Command History window for batch execution.

This allows a series of CDB’s to be set up, then sent sequentially with no delays.

The “Stop on Error” checkbox will stop a multi-pass CDB on error if checked. If not checked, a multi-pass CDB will execute all passes, regardless of any error conditions.

Command line arguments

Command line arguments are available to specify the number of times to issue the cdb, and to change the values in bytes 1 and 2 of the cdb.

Multi-pass CDB’s are issued by typing in the CDB name, a comma, then the number of passes.

For example, “READ,1000” will issue 1000 reads.

To issue a READ EXTENDED cdb and set the FUA bit, you would enter READ_10,1, 80

These arguments tell the program to:

- issue the READ_10 cdb
- issue the cdb 1 time
- modify byte 1 to the hex value 0x80 (this sets the FUA bit in the cdb)

Continuous command execution

Entering a pass count of less than 0 will cause the cdb to be issued continuously, until the “Stop” button is clicked.

Results Data Length

The “Results Data Length” allows you to specify how many bytes of data are displayed in the “Command Results” window. If a command transfers data, this many bytes will be displayed along with the command results. If the command terminates in a check condition, this many bytes of sense data will be displayed.

The entire data buffer can be viewed, modified, and saved/loaded from a file in the ***Edit Buffer*** tab page.

Re-issuing commands

Any command or commands can be re-issued by selecting them (clicking on them) in the “Command History” window. Once the commands you want to reissue are highlighted, click the “Execute” button and they will be reissued.

Build/Run Script

SCSI Script

CDB Functions

STB Test Functions

Target/LUN/Adapter Functions

Script Control Functions

Mode Page Functions

Buffer Functions

Edit Item

Delete Item from Script

Repeat Script times

Single Step -

Don't store results

Script Contents

Load Script

Save Script

Return

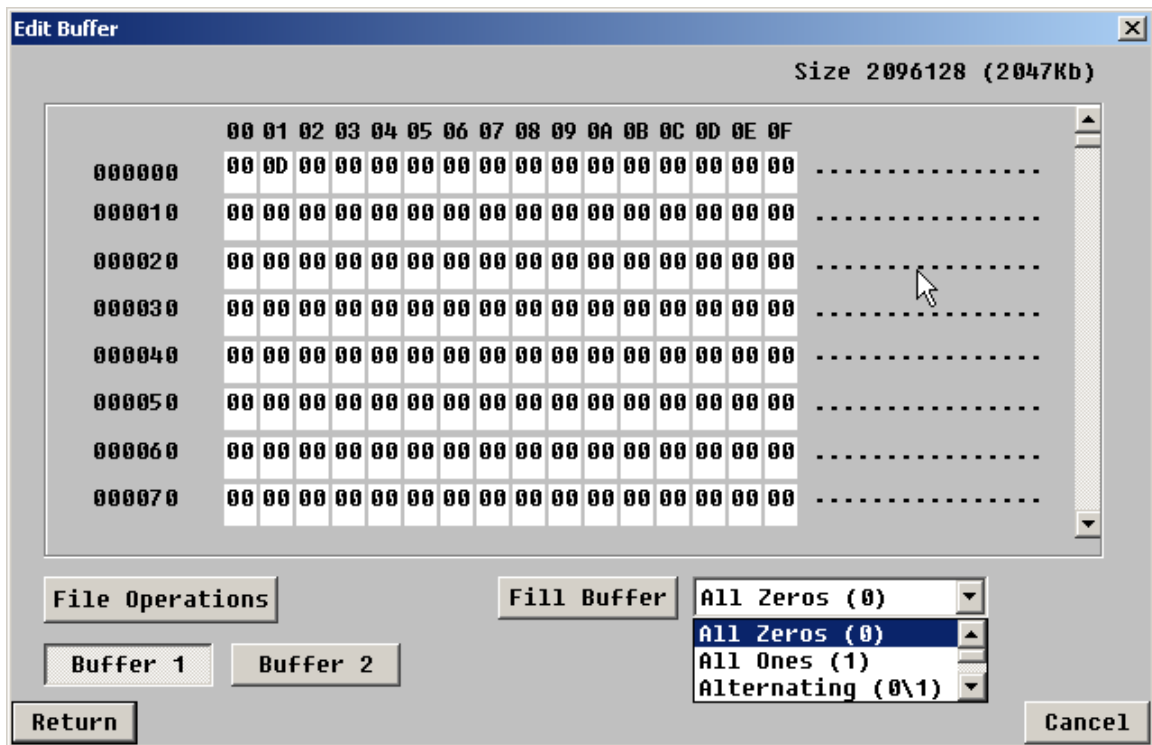
View Results

Edit Buffers

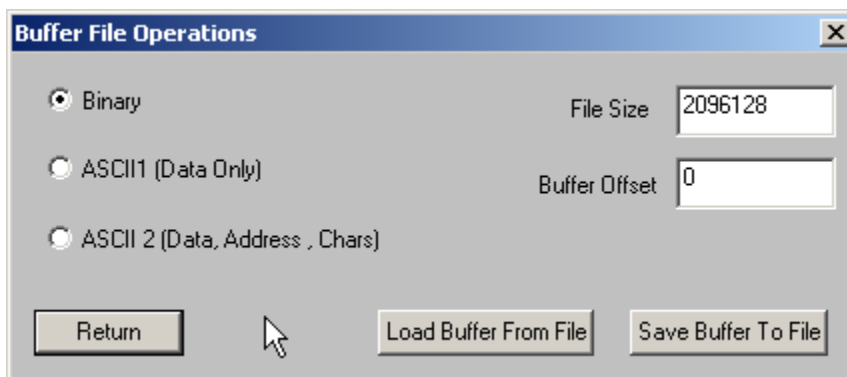
Stop on Error Log Results

Edit Buffers

The buffer functions allow the buffers used by the SCSI toolbox32 to be viewed, edited, saved to a disk file, or loaded from a disk file. The data in buffer 1 is used for all write tests. Buffer 1 will also contain all data read from the device.



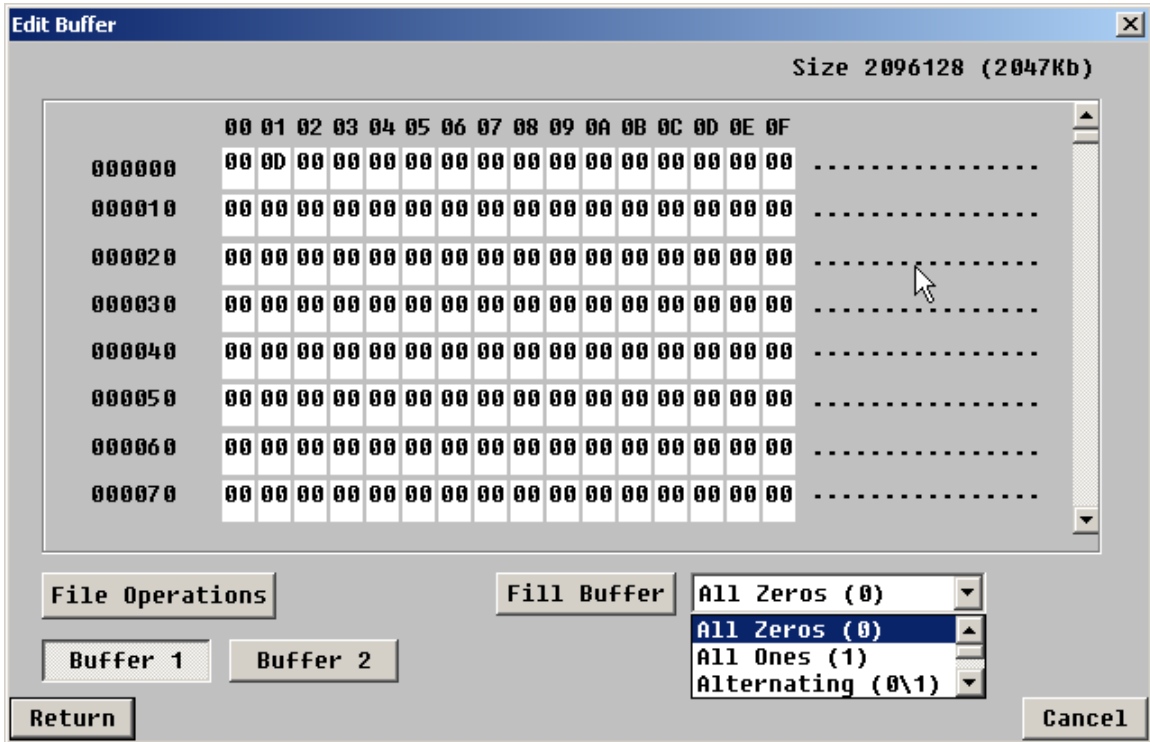
The drop box next to the Fill Buffer button allows you to choose a pattern to load into the buffer. Select the data pattern, then click the Fill Buffer button to load the data into the buffer. The File Operations button allows the buffer data to be saved to or loaded to a disk file.



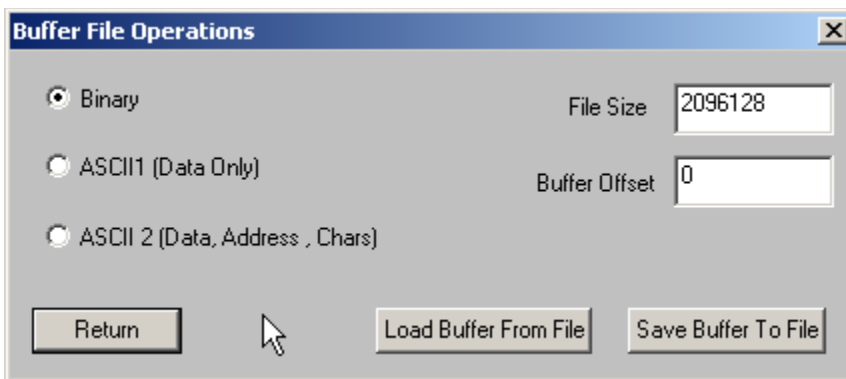
User choice allow the data to be saved either as binary data, or as ASCII data with or without the hex address of the data. The File size specifies how may bytes of data to read or write.

The Buffer Offset choice allows data to be loaded starting at a specific address of the buffer. Buffer Functions

The buffer functions allow the buffers used by the SCSI toolbox32 to be viewed, edited, saved to a disk file, or loaded from a disk file. The data in buffer 1 is used for all write tests. Buffer 1 will also contain all data read from the device.



The drop box next to the Fill Buffer button allows you to choose a pattern to load into the buffer. Select the data pattern, then click the Fill Buffer button to load the data into the buffer. The File Operations button allows the buffer data to be saved to or loaded to a disk file.



User choice allow the data to be saved either as binary data, or as ASCII data with or without the hex address of the data. The File size specifies how many bytes of data to read or write. The Buffer Offset choice allows data to be loaded starting at a specific address of the buffer.

SCSI Tutorial

Introduction to SCSI Philosophy

The purpose of this chapter is to give an overview of SCSI. SCSI encompasses hardware interfaces as well as software protocol specifications. The overall goal of the SCSI specification is to free system and peripheral designers from the physical specifications of the hardware they are working with, and to allow intelligence to be embedded within peripherals. This embedded intelligence should allow the host, or main processor, to concentrate its power on running application code, rather than mundane tasks needed to operate the peripheral.

The SCSI specification allows the system designer to view all peripherals as black boxes that contain blocks of data. This data may be accessible randomly, as in a disk drive, or it may be sequentially accessible, as in a tape drive or a scanner.

SCSI does away with dealing with heads, sectors, and cylinders on disk drives, because SCSI only deals with logical blocks of data in the device. In the case of disk drives, the SCSI specification allows the drive itself to deal with defective blocks, keeping track of spare blocks and remapping internally, that is, within the SCSI black box. This means that even though a drive may have defective blocks, the disk appears as a contiguous number of flawless blocks to the SCSI world.

This black box approach to data storage and retrieval makes the job of writing a universal device driver a fairly simple task. For instance, a disk device driver can accommodate drives of virtually any size, because the SCSI specification lets you ask a disk drive how many blocks of data it can hold.

And reading or writing data to a SCSI disk consists of simply telling the drive how many blocks you want to write, what block number to start writing at, then sending the data to the drive as the drive asks for it. The device driver never knows physically where on the drive the data is going because the SCSI peripheral takes care of all logical block to physical geometry mapping. As long as you do not try to read or write data beyond the highest accessible block number, everything is fine.

As SCSI specifications have evolved, support of input and output devices other than disk and tape have been added. The SCSI-2 specification defines standard methods for dealing with scanners, printers, and jukebox robotics. The SCSI-2 specification also defines a hardware upgrade path that allows higher data transfer rates and several wider data transfer paths.

SCSI Hardware

The SCSI bus is made up of eighteen lines that carry data: in the case of single-ended SCSI, several lines representing signal ground; in the case of differential SCSI, eighteen differential signal lines. These 18 signal lines can be thought of as two groups, nine (9) control lines and nine (9) data lines.

Data Lines

The data lines consist of eight bits of data and one bit of parity data. The SCSI-2 specification allows for a wide SCSI data path of 16 or 32 bits.

Control Lines

The nine control lines consist of two lines that control bus handshaking, six lines that define what state or phase the bus is in, and a reset line. The data and control lines that make up the physical SCSI bus can use two different electrical methods to convey information across the bus: single-ended and differential.

Single-ended SCSI

With the single-ended method, binary values on the SCSI bus are represented by two voltage levels on the bus lines. A high level is a voltage over 2.5 volts (+ - 5%) and a low level is a voltage that is less than .4 volts. The single-ended SCSI specification is an active low definition. That means that a logical one is represented by a low voltage level, and a logical zero is represented by a high voltage level. The single-ended SCSI specification allows for a bus cable length of up to 6 meters, although the maximum single-ended cable length is dependent upon the number of devices on the bus speeds at which they run.

Single-ended Termination

The SCSI bus must be terminated properly to prevent ringing and false signals. Single-ended termination consists of a resistor network on each signal line. This resistor network presents a constant impedance to the SCSI bus. The network consists of a 220 ohm resistor connected between 5 volts (the term power line on the SCSI bus) and the signal line, and a 330 ohm resistor connected between the signal line and ground. It is essential to proper termination that the terminating resistor network is supplied with +5 volts on the term power line.

Unfortunately, it is a common occurrence for the device supposed to be providing term power to have blown the fuse in line with the term power line. Unless you have a SCSI terminator with an LED to show that term power is valid, you may never know if the proper +5 volts is being supplied to your termination. Recalling that the SCSI bus is an active low bus, you can see why the lack of this +5 volts can lead to very unpredictable results! It is well worth using only terminators with term power indicators to avoid hours of troubleshooting headaches.

Differential SCSI

Differential SCSI consists of a pair of signal lines for each signal. The voltage levels on these pairs of lines move in opposite directions, and both the high and the low lines are checked in order for their data to be valid. This method provides its own noise cancellation and allows for cable lengths up to 25 meters. Single-ended SCSI to Differential SCSI converters are available, otherwise you can not mix single-ended and differential devices on the same SCSI bus.

Cables and Connectors

Typically, a raw SCSI device will have a 50 pin Dual in-line male connector for its SCSI bus connection. This type of connector has two rows of 25 pins spaced .1" apart. The active SCSI signals are connected to the even numbered pins, and the odd numbered pins are signal ground. Most systems use a connection other than the 50 pin dual in-line type. Some use a DB-50 type connector, some a 50 pin variation on the Centronics type connector, while others use a miniature version of the D-type connector. Some use only female gender connectors on their enclosures, while others use male gender. In other words, there is not one standard for connections and cables in the SCSI world!

SCSI Software

The SCSI specification describes a software protocol as well as a hardware definition. An understanding of this software aspect of SCSI will be helpful when interpreting SCSI problems and errors. A SCSI transaction happens between two SCSI devices, at which time each of the

devices takes on one of two roles: Initiator or Target. The initiator is typically the host computer, and the target is typically the peripheral device. The completion of the series of transactions that make up a full SCSI transaction between an initiator and a target is called a nexus. The intermediate steps that happen to complete a nexus are called **SCSI Bus Phases**. These phases are:

- Bus Free Phase
- Arbitration Phase
- Selection Phase
- Reselection Phase
- Information Transfer Phases, which consist of:
 - Command Phase
 - Data In Phase
 - Data Out Phase
 - Status Phase
 - Message Phase

The current bus phase is defined by the state of several of the control signal lines. Some phases can be entered from any other phase, while others must progress between phases in a specific order. Phase order, and transition between phases is controlled by the SCSI controller chips on the host adapter and on the SCSI peripherals. These chips are dedicated micro-processors that control all of the timing and signal transitions necessary to assure conformance with the SCSI specifications.

The Bus Free Phase

The bus free phase refers to the time when no SCSI activity is occurring on the bus. No devices are selected, no transfers are taking place during this phase. From this phase the SCSI bus can go to the arbitration phase, or in the case of a SCSI subsystem that does not support arbitration it can go to either the selection or reselection phase.

The Arbitration Phase

In this phase SCSI devices can negotiate for access to the bus. SCSI devices arbitrate by setting their ID number and the SCSI BUSY line. If more than one device wants the bus, the device with the highest SCSI address wins the arbitration and sets the SELECT line. Arbitration is required in SCSI II implementations.

The Selection\Reselection Phase

In this phase the SCSI initiator selects which SCSI target it will talk to. The SCSI initiator puts the bus in select phase, and sets the address of the target. Then it waits for the target to respond that it is there. Once the target has responded the target device will control the SCSI bus until the transaction is completed or until the target disconnects from the bus.

If the SCSI target determines that it will need some time before it fulfills the request, it will disconnect itself from the bus, allowing the bus to be used by other peripherals. For instance, if a SCSI tape drive receives a command to space forward it will know that this is a time consuming operation. Rather than make the host computer and other peripherals wait for it the tape drive can send a disconnect message to the host. This message effectively says, "I'm going to be busy for a while, I'll let you know when I'm ready to do something else." The host remembers that it started an operation with this peripheral, but that the operation is not finished yet. When the tape drive is ready, it will use the SCSI reconnect phase to tell the host, "Hey, I'm back now." The host will then pick up where it left off, finishing the command.

The Command Phase

The command phase is used to send a command code from the initiator to the target. This command code is made up of a number of bytes grouped into a data packet called a Command Descriptor Block (CDB).

Command Descriptor Blocks

A CDB can be 6, 10, or 12 bytes long. These bytes describe the command to the target, including information about any parameters that the target needs to execute the command. If more information is needed than can fit into the CDB, additional data blocks can be sent to the target. These additional data blocks are sent using the data phase of the SCSI bus.

CDB Format

The basic format of a six-byte CDB is as follows:

Byte	Bit 7	6	5	4	3	2	1	0	
0		Operation Code							
1	LUN	Logical Block (MSB)							
2		Logical Block Address							
3		Logical Block Address (LSB)							
4		Transfer Data Length							
5		Control Byte				Flag Link			

The operation code defines the command. The most basic SCSI command is the Test Unit Ready command, and its operation code is 0. The CDB for the Test Unit Ready command is

```
00 00 00 00 00 00
```

This command does not transfer any data, therefore the logical block address and the transfer data length are set to zero. Most SCSI commands will deal with logical unit 0, therefore the LUN field is also set to zero. It is very hard to set an illegal parameter in the Test Unit Ready command, since all of the parameters are zero!

A more complex command, The SCSI INQUIRY command will transfer data from the device, but it does not transfer the data from a data block on the device. Instead, it transfers the data from the device's controller. The operation code for the INQUIRY command is 0x12. The CDB for this command is:

```
12 00 00 00 ff 00
```

The 0xff value in byte 4 of the above CDB specifies that we want to transfer up to 255 bytes of information. If the device has fewer than 255 bytes to transfer, this will not result in an error. The device will send all of the data it has and then signal that it is finished. If we specified a transfer length of 5 bytes, and the device could have sent us 32 bytes, it will only transfer the 5 bytes we requested. Many of the SCSI commands that have a data phase will return a data header or special information about how many bytes were transferred.

The basic CDB structure is the same for the 10 and 12 byte CDBs— there is just more room in the command to specify a larger logical block address and a large transfer byte length. The more complex SCSI commands will have additional data phases that will send additional information to the device, as well as receive data from the device.

The Data In\Out Phase

The data phases are used to transfer data to or from the initiator. This data can be additional command parameters, such as Mode Select data or defect data, or it can be actual user data. Data transfers can happen in two different ways: Asynchronously or Synchronously.

Asynchronous Transfers

When a SCSI subsystem transfers data from a peripheral to a host there

needs to be a way for the devices to signal each other when they are ready for data or when they cannot receive data. This process is called handshaking. The peripheral that is receiving the data tells the sender when it is ready, and the sender always waits until it is told to send data. With Asynchronous SCSI, this handshaking process happens with each and every byte of data. The sender says, "I've got some data for you," then waits for the receiver to say, "O.K., send it over." This process happens with every byte sent, including SCSI commands, messages, user data, etc.

Synchronous Transfers

With Synchronous transfers, the target and initiator decide beforehand how fast they can talk. Using the SCSI message phase, a device asks another device, "Can you do sync transfers?" If the answer is yes, one device will ask the other, "How fast can you transfer?" If the reply is, "Five megabytes per second," but the first device can only run at 4.6 MB/sec, the dialog will continue until a transfer rate is agreed upon between the two devices.

Once a transfer rate has been negotiated, whenever data is transferred it is clocked out at that data rate. The two devices know how many bytes of data were supposed to transfer, and each device will toggle its handshaking line that many times while counting the handshaking line from the other device. As long as the number of handshakes agrees with the number of bytes transferred, everything is fine and the data transfer happens more quickly. In order to do sync-type transfers, both the SCSI hardware and the host SCSI device driver must be designed to do so.

The Status Phase

The status phase is used for the target to let the initiator know if a command was completed successfully or not. If a command does not complete successfully, the initiator can issue a REQUEST SENSE command to ask the target what went wrong. The sense data returned will describe the problem and hopefully point the way to a solution.

The Message Phase

The message phase is used for several functions, such as sync transfer negotiations, signaling command complete, telling the host to save data pointers, disconnecting/reconnecting, and many others. This allows initiators and targets to learn about each other and optimize performance.

SCSI Commands

While the SCSI controller chips take care of switching between bus phases and handling all of the handshaking, it is up to the system device driver to assemble CDBs, send them in the proper order, and interpret the data returned by them. Understanding which SCSI commands are available, and what those commands do is essential to successful SCSI troubleshooting. Reading the Commands chapter of this manual, and browsing through SCSI software manuals for the peripherals that you work with will give you a good overview of how these commands work.

SCSI Bus Operation—An Example

Typical SCSI Operation

This example describes the typical SCSI bus sequence between a SCSI host initiator and a target, in detail:

1. The host arbitrates for the SCSI bus by asserting BSY and the data line corresponding to its bus ID. If any other devices wish to compete for the bus, they also assert BSY and the appropriate data line. Each arbitrating device then inspects the data bus and the device with the highest ID wins it. All the other devices must release BSY and their data lines.
2. The host attempts to select the target by asserting SEL and releasing BSY. The host maintains its ID and asserts the target's ID on the data bus. Each target then checks the data

lines. If the target ID matches that on the data bus, it accepts selection by asserting BSY. Once the host has detected BSY being asserted, it asserts ATN to indicate that it will want the target to go to the MESSAGE OUT phase. The host releases SEL.

3. The target now has control of the SCSI bus and it is the target which switches between phases. The target responds to the ATTENTION condition and initiates the MESSAGE OUT phase. The host sends an IDENTIFY message which tells the target which logical unit the host wishes to talk to. The fact that the target responds to the ATN indicates to the host that the target can accommodate more than just a COMMAND COMPLETE message.

4. The target initiates the COMMAND phase and transfers the Command Descriptor Block from the host. In the COMMAND phase, the target decodes the command and either executes the command (TEST UNIT READY) or sets itself up for a data transfer to the host (for example, READ, WRITE, INQUIRY). The target then either switches to the STATUS phase if the command is complete, or DATA phase if it is ready to transfer data.

5. The data transfer length is set by the host in the Command Descriptor Block. The target remains in the DATA phase until all the data is transferred.

6. The target then initiates a STATUS phase and transfers one byte to the host to indicate whether it has completed the command successfully. If the target has detected an error, the next command that the host is expected to send is REQUEST SENSE. This allows the target to return further status information to the host.

7. The target completes the SCSI sequence by going to the MESSAGE IN phase and transferring a COMMAND COMPLETE message to the host. The target then releases BSY, allowing the bus to go to the BUS FREE state.

Disconnect

In order to improve bus usage and performance, many SCSI devices are capable of disconnecting from the host in order to free the bus to allow other requests to be sent to other targets. To do so, however, the host needs to support Disconnect/Reselect. If Disconnect is implemented, the procedure is as follows:

1. The host arbitrates for the SCSI bus and if it wins it, selects the target device. Before releasing SEL and completing the selection phase, the host asserts the ATN line. The host then releases SEL and BSY. The target now has control of the SCSI bus. By asserting ATN, the host has indicated that the target should go to a MESSAGE OUT phase.

2. After the SELECTION phase is completed, the target responds to the host's ATTENTION condition by initiating a MESSAGE OUT phase. It receives a message from the host which tells it if the host can support Disconnect/Reselect and the desired logical unit number on the target.

3. The I/O activity from this point is controlled entirely by the target. The target initiates the COMMAND phase and reads in the Command Descriptor Block from the host. After decoding the command, the target determines whether it should disconnect from the bus. The target disconnects from the bus for any non-trivial commands.

4. The disconnect process is when the target initiates a MESSAGE IN phase and sends the host a SAVE DATA POINTERS (during a DATA phase only) and a DISCONNECT message. Following the MESSAGE IN phase, the target releases BSY, freeing the bus which then enters

the BUS FREE state. The host can now select another target, or allow another target to win the bus and reselect the host.

5. Although the host and the target are physically disconnected, they are still logically connected. Both know that they have a command to finish and will return to that job later. This principle allows many I/O commands to be executed simultaneously using a single peripheral bus. Once the target has completed a task and is ready to communicate with the host, it must reestablish the physical path. The reselection process involves the target arbitrating for the bus and reselecting the host. After the physical reconnection is made, the target sends an IDENTIFY message to the host to indicate which target initiates the next appropriate phase for the command, usually a DATA phase.

6. During a large data transfer, the target may disconnect at intervals depending on its use of the bus. The drive optimizes its use of the bus so as to maximize the transfer rate when it is connected to the host, and to minimize the time for which it holds the bus without handshakes. If the target disconnects, during a data transfer, the target initiates a MESSAGE IN phase and send the host a SAVE DATA POINTERS message and a DISCONNECT message. The host responds to the SAVE DATA POINTERS message by saving the current data pointer. After transmission of the DISCONNECT message the target releases BSY, freeing the bus.

7. Once the target is again ready to reselect the host, it goes through the same process as before—arbitrating for the bus, reselecting the host and sending an IDENTIFY message. However, the host's response is slightly different in this case since the disconnect was during a data transfer. Host acceptance of the IDENTIFY message also implies a RESTORE DATA POINTERS message to the host. The data transfer can now be resumed.

8. After completion of the data transfer, the target initiates a STATUS phase and sends a single status byte to the host. The final action of the target is to initiate a MESSAGE IN phase and send a COMMAND COMPLETE message to the host.

Disconnection Notes

Certain devices will disconnect on completion of a data transfer if the final transfer occurs on a disconnect boundary, before initiating the STATUS phase. This is intended to optimize bus usage. Most devices do not disconnect on receipt of the following commands:

INQUIRY

REQUEST SENSE

TEST UNIT READY

The drive will disconnect on other commands if it is programmed or configured to do so.

Miscellaneous SCSI Issues

SCSI Standards

The SCSI standard describes the physical and electrical characteristics of a parallel I/O bus used when connecting computers and peripherals in a daisy-chained manner. The connection of devices such as disk drives, tape drives, optical drives, printers, CD-ROM drives, and other devices without hardware modification is specified by the standard.

The SCSI bus provides two electrical specifications: single-ended and differential. The single-ended driver and receiver configuration uses TTL (Transistor-Transistor Logic levels and is primarily designed for applications within a cabinet. The single-ended version uses cable lengths up to 6 meters (19.68 feet). The differential driver and

receiver configuration uses EIA RS-485 signals and is primarily designed for applications requiring longer cable lengths. The differential version uses cable lengths of up to 25 meters (82.02 feet). The original SCSI standard was approved in 1986 and was called SCSI-1.

The SCSI-2 revision was released in 1992, and incorporates wide SCSI which permits 16 or 32 bit parallel transfer using two cables. Combined with the fast SCSI option, data transfer rates up to 40 megabytes per second are possible. Active termination was also specified for this standard. The SCSI-3 revision is currently being released. New features have been suggested and are being implemented to improve the SCSI-2 revision. Improvements include: the ability to address 32 devices, a single 16 bit data bus cable, and a serial SCSI protocol.

SCSI Cabling

The Small Computer System Interface (SCSI) is now being used by faster and more complex devices. The SCSI interface and associated ANSI standards are very flexible and allow for faster transmission rates. A major limiting factor to higher throughput and data integrity is SCSI cabling. SCSI cables must be engineered correctly in order to handle the increased transmission rates of today's SCSI devices.

Over the years that PTI has been involved with SCSI products and enhancements, experience shows that more than 80% of the problems with a new installation of external SCSI-based devices (disk drives, tapes, optical drives, etc.) have been related to the SCSI cables used in the installations. Ten percent of the problems relate to improper termination, and five percent to software improperly installed. With modern SCSI devices, only a small percentage actually involve true device failure (and even in these instances the power supply that powers the SCSI peripheral is often the problem source). When there is a problem with the cable, the symptoms vary greatly. The system may not operate at all, or there will be intermittent SCSI communication failures. In many cases the symptoms are initially thought to be due to the devices on the SCSI bus or the software drivers running the devices, resulting in excessive system installation delays and costs. Most SCSI integrators are using some form of shielded round cables when daisy-chaining external SCSI devices. Shielded round SCSI cables came into being because of the problems that unshielded flat ribbon cables have with electromagnetic interference (EMI).

Unshielded cable can not pass FCC requirements, whereas properly shielded round cable can function within FCC specifications. However, unlike poorly constructed round cabling, crosstalk noise in the bus is not a problem with flat ribbon cable. Crosstalk noise is best controlled in round cabling by careful conductor placement (clocks in the center, data around the periphery), and by using twisted-pair cables. The combination of cabling types (round versus flat) and quality (shielded, grounded, etc) within a single SCSI bus can create cable impedance mismatches. The 3 meter and 6 meter length limits suggested for 10 and 5 megabytes per second transfers (Fast SCSI-2 and SCSI-1 respectively), are aimed at eliminating potential data problems on the SCSI bus. Where circumstances present a need for longer bus operation, special emphasis must be paid to the guidelines suggested above and in the standard.

With many integrators not readily able to provide specifications as to how their cables are constructed, it is usually very hard to determine if the purchased round SCSI cable will work error-free in the target SCSI environment. There are SCSI round cable suppliers providing cables with connectors that, though specified as SCSI 50-pin cables, may have as few as 25 lines wired from connector to connector.

Differential applications will not operate under these conditions, and many single-ended SCSI applications will also have trouble maintaining uncorrupted signal transmissions with cable more than 6 feet long, as most of the ground lines are not connected. Even if all of the wires are connected, and even if twisted pair cable made specifically for SCSI is

used, depending upon how the cable is wired to the connector, there can also be problems. The twisted pairs on the cable must be matched with the correct pin numbers. Otherwise it is possible to have the plus side of two signal lines going through the same wire pair, inducing interference between the signals. This causes problems on fast SCSI-based systems, and systems with longer SCSI cable lengths. As is the case also with SCSI termination issues, PTI recommends careful examination of cabling specifications as part of a properly engineered SCSI environment. Cabling problems present challenging and frustrating problems in the real-world of SCSI. Savings on poorly constructed cables is rapidly lost when the cable presents data integrity problems.

Simple Peripheral Installation Guidelines

SCSI gives new meaning to the term "plug-and-play." A single peripheral attached to a matching host adapter is usually easy to install, but when you plug several into a single host, things can get more complicated. The SCSI-2 draft specification, universal drivers, and easier installation routines go a long way toward making things easier. But even so, it's likely that you will come up against at least one or two problems along the way. Here are eight tips for getting your setup to go as smoothly as possible.

1. Start simple. Start with the basics. Rather than plugging a chain of peripherals into a single SCSI card and then booting up (and confronting the confusion and possible conflicts that are likely to result), get the host adapter installed first and then install the first hard disk. You will want to continue installing devices one at a time, checking to see that they are working before moving to the next. This may seem like a bit of extra work (you will have to install and reinstall device drivers for most of the peripherals, for example), but in a complex setup, this is often a shortcut in disguise.

2. Make a list. When adding a new device, make sure that you know all of the existing SCSI identification numbers on the chain before you start installation. If you have a host adapter that reports the IDs during bootup (as the Adaptec AHA-1542C does), the task is easy; otherwise prepare to check the jumper or switch settings on each peripheral in the chain. Or better yet, keep an up-to-date, easily accessible list of all the SCSI peripheral IDs on the chain, as well as the I/O address and BIOS (Basic Input/Output System) location of the host adapter. Devices with low number settings have a higher priority. As a rule, you'll want to set your first hard disk at 0 and your second hard disk at 1. Host adapters are generally set at 7, and some operating systems (such as those that are Unix-based) expect to find them set there. In certain cases, some operating systems also expect other peripherals, such as CD-ROMs, to hold specific IDs; check your software and adapter manuals before installation. SCO Unix, for example, expects a CD-ROM to be located at ID5. If you ignore such rules, the system may ignore your device.

3. Internals first. Try to install all your internal units first (starting with hard disks). But install them one at a time and before you start experimenting with external devices and termination. Adding an external device means you need to make a change in the devices that are terminated.

4. Observe proper termination. In a SCSI set up, each end of the SCSI chain must be terminated. If you have only internal or external devices on the bus, the host adapter and last device on the chain should be terminated. If you have external and internal devices on the chain, you will generally terminate the first and last of these devices but not the SCSI host adapter.

5. Making connections. There are a variety of SCSI cable connectors: 25-pin for SCSI-1, 50-pin for SCSI-2, and 68-pin for the Wide-SCSI. Look at the connectors on the items you need to attach before you buy, and be sure to obtain quality cables with good shielding. Your best bet is to buy internal and external cables as short as possible, but no shorter than one foot long (to avoid signal noise.)

6. Check and test. Turn on any external devices before booting your PC; most external devices must be running to be recognized by their driver when the PC boots. If a unit (or the whole chain) fails, try the following: Check the ID numbers and cable connections to make sure that peripherals requiring specific IDs are set for them and that all devices are appropriately connected. Then power-down and power-up the system, watching the messages during the initialization of the configuration files. Make sure all drivers are loaded properly. If you can isolate an offending device, reposition it elsewhere on the chain. If that fails, change its ID number; perhaps the device is just unable to work at the ID number for which it was set. Next, try swapping cables between units and experiment with different combinations of termination, changing one element at a time. There may be a conflict, or the term power level may not be sufficient. In the latter case, you may have to add a terminator to the host adapter or even to another device on the chain.

Troubleshooting SCSI toolbox32 Problems

This chapter will describe some common problems that can be addressed with the SCSI *toolbox*. It will also help with troubleshooting the SCSI *toolbox* itself if the need arises.

Problem: The SCSI *toolbox32* does not see any of the peripherals attached.

Solution: First, make sure that the SCSI bus has termination power. The SCSI *toolbox* hardware gets its power from the SCSI term power line. This power must be supplied either by one of the attached SCSI peripherals or by the PTI TermPower unit. Second, be sure that your SCSI cable is attached properly and alright. Last, be sure that the last attached SCSI peripheral is terminated, and that only the last one is terminated.

Problem: The attached SCSI peripheral shows up at all addresses when using the Device Menu.

Solution: The peripheral is probably set to SCSI address seven. The SCSI *toolbox* hardware adapter uses address seven, so be sure that none of the peripherals are set to seven.

Problem: Strange or intermittent data is displayed when using the Device Menu.

Solution: Be sure that each attached peripheral is set to a separate address. A defective SCSI cable or terminator can also show this symptom.

Problem: The SCSI *toolbox32* hangs.

Solution: Reboot the computer that is running the *toolbox*. (A variety of problems can cause "hangs.")

Problem: You have a dead SCSI disk drive.

Solution: Always check the block size and capacity of a SCSI disk. Either look at the DEVICE menu or use the READ CAPACITY command in the COMMAND menu. If the block size or capacity numbers seem strange (too large or too small) the problem is that the low-level SCSI format on the drive has been damaged. Use the FORMAT command in the COMMAND menu to reformat the drive.

Problem: You need to test a CD-ROM.

Solution: To test a read-only device like a CD-ROM, use any of the tests that are read only, such as the sequential read test or the random read test.

Problem: You think that a disk drive may have had a head crash.

Solution: Examine the Grown Defect List using the READ DEFECT DATA command. Take note of many grown defects in the same general block range. Also, reformat the disk. Most disk drives will verify as they format, adding additional flaws to the Grown defect list. You can also run the VERIFY TEST, again noting if there are many flaws grouped together.

Problem: You think your SCSI cable may have a short or open connection.

Solution: The data lines of the SCSI bus are used both for addressing and for data transmission. Therefore, if the cable has a problem with the data lines it will be seen as either an addressing problem, or the INQUIRY data displayed in the DEVICE menu will be wrong. If the INQUIRY data is wrong, look in an ASCII table to see which bit of the data lines is stuck. If the problem is with the control lines, the program may hang in an unpredictable way.

Problem: A SCSI disk will not come on line, or it is on line but no capacity is reported. This is usually caused by a corrupted disk format.

Solution: The FORMAT command in the Command Menu will attempt to run even if the disk is off line. Try using the FORMAT command to restore the media format.

Problem: Still having problems?

Solution: Call our **Technical Support Line** at 303-763-7488.

Request Sense & Sense Key Interpretation

Sense Description

Key

0hNO SENSE. Indicates that there is no specific sense key information to be reported for the designated logical unit. This would be the case for a successful command or a command that received CHECK CONDITION or COMMAND TERMINATED status because one of the filemark, EOM, or ILI bits is set to one.

1hRECOVERED ERROR. Indicates that the last command completed successfully with some recovery action performed by the target. Details may be determined by examining the additional sense bytes and the information field. When multiple recovered errors occur during one command, the choice of error to report (first, last, most severe, etc.) is device specific.

2h NOT READY. Indicates that the logical unit addressed cannot be accessed. Operator intervention may be required to correct this condition.

3hMEDIUM ERROR. Indicates that the command terminated with a non-recovered error condition which was probably caused by a flaw in the medium or an error in the recorded data. This sense key may also be returned if the target is unable to distinguish between a flaw in the medium and a specific hardware failure (sense key 4h).

4hHARDWARE ERROR. Indicates that the target detected a non-recoverable hardware failure (for example, controller failure, device failure, parity error, etc.) while performing the command or during a self test.

5hILLEGAL REQUEST. Indicates there was an illegal parameter in the command descriptor block or in the additional parameters supplied as data for some commands (FORMAT UNIT, SEARCH DATA, etc.). If the target detects an invalid parameter in the command descriptor block, then it shall terminate the command without altering the medium. If the target detects an invalid parameter in the additional parameters supplied as data, the target may have already altered the medium. This sense key may also indicate that an invalid IDENTIFY message was received.

6hUNIT ATTENTION. Indicates that the removable medium may have been changed or the target has been reset.

7hDATA PROTECT. Indicates that a command that reads or writes the medium was attempted on a block that is protected from this operation. The read or write operation is not performed.

8hBLANK CHECK. Indicates that a write-once device or a sequential-access device encountered blank medium; or format-defined end-of-data indication while reading; or a write-once device encountered anon-blank medium while writing.

9hVendor Specific. This sense key is available for reporting vendor specific conditions.

AhCOPY ABORTED. Indicates a COPY, COMPARE, or COPY AND VERIFY command was aborted due to an error condition on the source device, the destination device, or both.

BhABORTED COMMAND. Indicates that the target aborted the command. The initiator may be able to recover by trying the command again.

ChEQUAL. Indicates a SEARCH DATA command has satisfied an equal comparison.

DhVOLUME OVERFLOW. Indicates that a buffered peripheral device has reached the end-of-partition and data may remain in the buffer that has not been written to the medium. A RECOVER BUFFERED DATA command(s) may be issued to read the unwritten data from the buffer.

EhMISCOMPARE. Indicates that the source data did not match the data read from the medium.

FhRESERVED.

Additional Sense Code/ Qualifier Interpretation

The additional sense codes and additional sense code qualifiers are defined below:

```

=====
D - DIRECT ACCESS DEVICE
.T - SEQUENTIAL ACCESS DEVICE
.L - PRINTER DEVICE
.P - PROCESSOR DEVICE
.W - WRITE ONCE READ MULTIPLE DEVICE
.R - READ ONLY (CD-ROM) DEVICE
.S - SCANNER DEVICE
.O - OPTICAL MEMORY DEVICE
.M - MEDIA CHANGER DEVICE
.C - COMMUNICATION DEVICE
  
```

ASC	ASQC	DTLPWRSOMC	DESCRIPTION
13h	00h	D W O	ADDRESS MARK NOT FOUND FOR DATA FIELD
12h	00h	D W O	ADDRESS MARK NOT FOUND FOR ID FIELD
00h	11h	R	AUDIO PLAY OPERATION IN PROGRESS
00h	12h	R	AUDIO PLAY OPERATION PAUSED
00h	14h	R	AUDIO PLAY OPERATION STOPPED DUE TO ERROR
00h	13h	R	AUDIO PLAY OPERATION SUCCESSFULLY COMPLETED
00h	04h	T S	BEGINNING-OF-PARTITION/MEDIUM DETECTED
14h	04h	T	BLOCK SEQUENCE ERROR
30h	02h	DT WR O	CANNOT READ MEDIUM - INCOMPATIBLE FORMAT
30h	01h	DT WR O	CANNOT READ MEDIUM - UNKNOWN FORMAT
52h	00h	T	CARTRIDGE FAULT
3Fh	02h	DTLPWRSOMC	CHANGED OPERATING DEFINITION
11h	06h	WRO	CIRC UNRECOVERED ERROR
30h	03h	DT	CLEANING CARTRIDGE INSTALLED
4Ah	00h	DTLPWRSOMC	COMMAND PHASE ERROR
2Ch	00h	DTLPWRSOMC	COMMAND SEQUENCE ERROR
2Bh	00h	DTLPWRSO	COPY CANNOT EXECUTE SINCE HOST CANNOT DISCONNECT
41h	00h	D	DATA PATH FAILURE (SHOULD USE 40 NN)
4Bh	00h	DTLPWRSOMC	DATA PHASE ERROR
11h	07h	W O	DATA RESYNCHRONIZATION ERROR
16h	00h	D W O	DATA SYNCHRONIZATION MARK ERROR
19h	00h	D O	DEFECT LIST ERROR
19h	03h	D O	DEFECT LIST ERROR IN GROWN LIST
19h	02h	D O	DEFECT LIST ERROR IN PRIMARY LIST
19h	01h	D O	DEFECT LIST NOT AVAILABLE
1Ch	00h	D O	DEFECT LIST NOT FOUND
32h	01h	D W O	DEFECT LIST UPDATE FAILURE
40h	NNh	DTLPWRSOMC	DIAGNOSTIC FAILURE ON COMPONENT NN (80H-FFH)
63h	00h	R	END OF USER AREA ENCOUNTERED ON THIS TRACK
00h	05h	T S	END-OF-DATA DETECTED
14h	03h	T	END-OF-DATA NOT FOUND
00h	02h	T S	END-OF-PARTITION/MEDIUM DETECTED
51h	00h	T O	ERASE FAILURE
0Ah	00h	DTLPWRSOMC	ERROR TOO LONG TO CORRECT
03h	02h	T	EXCESSIVE WRITE ERRORS
3Bh	07h	L	FAILED TO SENSE BOTTOM-OF-FORM
3Bh	06h	L	FAILED TO SENSE TOP-OF-FORM
00h	01h	T	FILEMARK OR SETMARK NOT FOUND
09h	02h	WR O	FOCUS SERVO FAILURE
31h	01h	DL O	FORMAT COMMAND FAILED
58h	00h	O	GROWN DEFECT LIST NOT FOUND
00h	06h	DTLPWRS OMC	I/O PROCESS TERMINATED
10h	00h	D W O	ID CRC OR ECC ERROR
22h	00h	D	LEGAL FUNCTION(SHOULD USE 20 00, 24 00, OR 26 00)
64h	00h	R	ILLEGAL MODE FOR THIS TRACK
28h	01h	M	IMPORT OR EXPORT ELEMENT ACCESSED
30h	00h	DT WR O M	INCOMPATIBLE MEDIUM INSTALLED
11h	08h	T	INCOMPLETE BLOCK READ
48h	00h	DTLPWRSO MC	INITIATOR DETECTED ERROR MESSAGE RECEIVED
3Fh	03h	DTLPWRSO MC	INQUIRY DATA HAS CHANGED
44h	00h	DTLPWRSO MC	INTERNAL TARGET FAILURE
3Dh	00h	DTLPWRSO MC	INVALID BITS IN IDENTIFY MESSAGE
2Ch	02h	S	INVALID COMBINATION OF WINDOWS SPECIFIED
20h	00h	DTLPWRSO MC	INVALID ELEMENT ADDRESS
24h	00h	DTLPWRSO MC	INVALID FIELD IN CDB
26h	00h	DTLPWRSO MC	INVALID FIELD IN PARAMETER LIST
49h	00h	DTLPWRSO MC	INVALID MESSAGE ERROR
11h	05h	WR O	LAMP FAILURE
5Bh	02h	DTLPWRSO M	LOG COUNTER AT MAXIMUM
5Bh	00h	DTLPWRSO M	LOG EXCEPTION
5Bh	03h	DTLPWRSO M	LOG PARAMETERS CHANGED
21h	00h	DT WR O M	LOGICAL BLOCK ADDRESS OUT OF RANGE
08h	00h	DTL WRS O MC	LOGICAL UNIT COMMUNICATION FAILURE
08h	02h	DTL WRS O MC	LOGICAL UNIT COMMUNICATION PARITY ERROR
08h	01h	DTL WRS O MC	LOGICAL UNIT COMMUNICATION TIME-OUT
4Ch	00h	DTLPWRSO M	LOGICAL UNIT FAILED SELF-CONFIGURATION
3Eh	00h	DTLPWRSO MC	LOGICAL UNIT HAS NOT SELF-CONFIGURED YET

04h 01h	DTLPWRSO MC	LOGICAL UNIT IS IN PROCESS OF BECOMING READY
04h 00h	DTLPWRSO MC	LOGICAL UNIT NOT READY, CAUSE NOT REPORTABLE
04h 04h	DTL O	LOGICAL UNIT NOT READY, FORMAT IN PROGRESS
04h 02h	DTLPWRSO MC	LOGICAL UNIT NOT READY, INITIALIZINGCOMMAND REQUIRED
04h 03h	DTLPWRSO MC	LOGICAL UNIT NOT READY, MANUAL INTERVENTION REQUIRED
25h 00h	DTLPWRSO MC	LOGICAL UNIT NOT SUPPORTED
15h 01h	DTL WRS O M	MECHANICAL POSITIONING ERROR
53h 00h	DTL WRS O M	MEDIA LOAD OR EJECT FAILED
3Bh 0Dh	M	MEDIUM DESTINATION ELEMENT FULL
31h 00h	DT W O	MEDIUM FORMAT CORRUPTED
3Ah 00h	DTL WRS O M	MEDIUM REMOVAL PREVENTED
3Bh 0Eh	M	MEDIUM SOURCE ELEMENT EMPTY
43h 00h	DTLPWRSOMC	MESSAGE ERROR
3Fh 01h	DTLPWRSOMC	MICROCODE HAS BEEN CHANGED
1Dh 00h	D W O	MISCOMPARE DURING VERIFY OPERATION
11h 0Ah	DT O	MISCORRECTED ERROR
2Ah 01h	DTL WRSOMC	MODE PARAMETERS CHANGED
07h 00h	DTL WRS OM	MULTIPLE PERIPHERAL DEVICES SELECTED
11h 03h	DT W S O	MULTIPLE READ ERRORS
00h 00h	DTLPWRSOMC	NO ADDITIONAL SENSE INFORMATION
00h 15h	R	NO CURRENT AUDIO STATUS TO RETURN
32h 00h	D W O	NO DEFECT SPARE LOCATION AVAILABLE
11h 09h	T	NO GAP FOUND
01h 00h	D W O	NO INDEX/SECTOR SIGNAL
06h 00h	D WR OM	NO REFERENCE POSITION FOUND
02h 00h	D WR OM	NO SEEK COMPLETE
03h 01h	T	NO WRITE CURRENT
28h 00h	DTLPWRSOMC	NOT READY TO READY TRANSITION, MEDIUM MAY HAVE CHANGED
5Ah 01h	DT WR OM	OPERATOR MEDIUM REMOVAL REQUEST
5Ah 00h	DTLPWRSOM	OPERATOR REQUEST OR STATE CHANGE INPUT (UNSPECIFIED)
5Ah 03h	DT W O	OPERATOR SELECTED WRITE PERMIT
5Ah 02h	DT W O	OPERATOR SELECTED WRITE PROTECT
61h 02h	S	OUT OF FOCUS
4Eh 00h	DTLPWRSOMC	OVERLAPPED COMMANDS ATTEMPTED
2Dh 00h	T	OVERWRITE ERROR ON UPDATE IN PLACE
3Bh 05h	L	PAPER JAM
1Ah 00h	DTLPWRSOMC	PARAMETER LIST LENGTH ERROR
26h 01h	DTLPWRSOMC	PARAMETER VALUE INVALID
2Ah 00h	DTL WRSOMC	PARAMETERS CHANGED
03h 00h	DTL W S O	PERIPHERAL DEVICE WRITE FAULT
50h 02h	T	POSITION ERROR RELATED TO TIMING
3Bh 0Ch	S	POSITION PAST BEGINNING OF MEDIUM
3Bh 0Bh	S	POSITION PAST END OF MEDIUM
15h 02h	DT WR O	POSITIONING ERROR DETECTED BY READ OF MEDIUM
29h 0	DTLPWRSOMC	POWER ON, RESET, OR BUS DEVICE RESET OCCURRED
42h 00h	D	POWER-ON OR SELF-TEST FAILURE (SHOULD USE 40NN)
1Ch 01h	D O	PRIMARY DEFECT LIST NOT FOUND
40h 00h	D	RAM FAILURE (SHOULD USE 40 NN)
15h 00h	DTL WRSOM	RANDOM POSITIONING ERROR
3Bh 0Ah	S	READ PAST BEGINNING OF MEDIUM
3Bh 09h	S	READ PAST END OF MEDIUM
11h 01h	DT W S O	READ RETRIES EXHAUSTED
14h 01h	DT WR O	RECORD NOT FOUND
14h 00h	DTLWRS O	RECORDED ENTITY NOT FOUND
18h 02h	D WR O	RECOVERED DATA - DATA AUTO-REALLOCATED
18h 05h	D WR O	RECOVERED DATA - RECOMMEND REASSIGNMENT
18h 06h	D WR O	RECOVERED DATA - RECOMMEND REWRITE
17h 05h	D WR O	RECOVERED DATA USING PREVIOUS SECTOR ID
18h 03h	R	RECOVERED DATA WITH CIRC
18h 01h	D WR O	RECOVERED DATA WITH ERROR CORRECTION & RETRIES APPLIED
18h 00h	DT WR O	RECOVERED DATA WITH L-EC
17h 03h	DT WR O	RECOVERED DATA WITH NEGATIVE HEAD OFFSET
17h 00h	DT WRSO	RECOVERED DATA WITH NO ERROR CORRECTION APPLIED
17h 02h	DT WR O	RECOVERED DATA WITH POSITIVE HEAD OFFSET
17h 01h	DT WRS O	RECOVERED DATA WITH RETRIES AND/OR CIRC APPLIED
17h 06h	D W O	RECOVERED DATA WITHOUT ECC - DATA AUTO-REALLOCATED
17h 07h	D W O	RECOVERED DATA W/O ECC - RECOMMEND REASSIGNMENT
17h 08h	D W O	RECOVERED DATA WITHOUT ECC -RECOMMEND REWRITE
1Eh 00h	D W O	RECOVERED ID WITH ECC CORRECTION
3Bh 08h	T	REPOSITION ERROR
36h 00h	L	RIBBON, INK, OR TONER FAILURE
37h 00h	DTL WRSOMC	ROUNDED PARAMETER
5Ch 00h	D O	RPL STATUS CHANGE
39h 00h	DTL WR SOMCSAVING	PARAMETERS NOT SUPPORTED
62h 00h	S	SCAN HEAD POSITIONING ERROR
47h 00h	DTLPWRSOMCSCSI	PARITY ERROR
54h 00h	P	SCSI TO HOST SYSTEM INTERFACE FAILURE
45h 00h	DTLPWRSOM	SELECT OR RESELECT FAILURE
3Bh 00h	TL	SEQUENTIAL POSITIONING ERROR
00h 03h	T	SETMARK DETECTED
3Bh 04h	L	SLEW FAILURE
09h 03h	WR O	SPINDLES SYNCHRONIZED
1Bh 00h	DTLPWRSOMC	SYNCHRONOUS DATA TRANSFER ERROR
55h 00h	P	SYSTEM RESOURCE FAILURE
33h 00h	T	TAPE LENGTH ERROR
3Bh 03h	L	TAPE OR ELECTRONIC VERTICAL FORMS UNIT NOT READY
3Bh 01h	T	TAPE POSITION ERROR AT BEGINNING-OF-MEDIUM
3Bh 02h	T	TAPE POSITION ERROR AT END-OF-MEDIUM
3Fh 00h	DTLPWRSOMC	TARGET OPERATING CONDITIONS HAVE CHANGED
5Bh 01h	DTLPWRSOM	THRESHOLD CONDITION MET
26h 03h	DTLPWRSOMC	THRESHOLD PARAMETERS NOT SUPPORTED
2Ch 01h	S	TOO MANY WINDOWS SPECIFIED
09h 00h	DT WR O	TRACK FOLLOWING ERROR

09h 01h	WR O	TRACKING SERVO FAILURE
61h 01h	S	UNABLE TO ACQUIRE VIDEO
57h 00h	R	UNABLE TO RECOVER TABLE-OF-CONTENTS
53h 01h	T	UNLOAD TAPE FAILURE
11h 00h	DT WR S O	UNRECOVERED READ ERROR
11h 04h	D W O	UNRECOVERED READ ERROR - AUTO REALLOCATE FAILED
11h 0Bh	D W O	UNRECOVERED READ ERROR - RECOMMEND REASSIGNMENT
11h 0Ch	D W O	UNRECOVERED READ ERROR-RECOMMEND REWRITE DATA
46h 00h	DTLPWRS OMC	UNSUCCESSFUL SOFT RESET
59h 00h	O	VIDEO ACQUISITION ERROR
50h 00h	T	WRITE APPEND ERROR
50h 01h	T	WRITE APPEND POSITION ERROR
0Ch 00h	T S	WRITE ERROR
0Ch 02h	D W O	WRITE ERROR - AUTO REALLOCATION FAILED
0Ch 01h	D W O	WRITE ERROR RECOVERED WITH AUTO REALLOCATION
27h 00h	DT W O	WRITE PROTECTED
80h XXh\THRU > FFh XX/		VENDOR SPECIFIC
XXh 80h\THRU > XXh FFh/		VENDOR SPECIFIC QUALIFICATION OF STANDARD ABC.

ALL CODES NOT SHOWN ARE RESERVED.

SCSI Command Definitions

The following table is a numerical order listing of the command operation codes.

SCSI-2 Operation Codes

=====	
D - DIRECT ACCESS DEVICE	Device Column Key
.T - SEQUENTIAL ACCESS DEVICE	M = Mandatory
.L - PRINTER DEVICE	O = Optional
.P - PROCESSOR DEVICE	V = Vendor Specific
.W - WRITE ONCE READ MULTIPLE DEVICE	R = Reserved
.R - READ ONLY (CD-ROM) DEVICE	
.S - SCANNER DEVICE	
.O - OPTICAL MEMORY DEVICE	
.M - MEDIA CHANGER DEVICE	
.C - COMMUNICATION DEVICE	

OP DTLPWRSOMC DESCRIPTION

00	MMMMMMMMMM	TEST UNIT READY
01	M	REWIND
01	O V OO OO	REZERO UNIT
02	VVVVVV V	
03	MMMMMMMMMM	REQUEST SENSE
04	O	FORMAT
04	M O	FORMAT UNIT
05	VMVVVV V	READ BLOCK LIMITS
06	VVVVVV V	\
07	O	INITIALIZE ELEMENT STATUS
07	OVV O OV	REASSIGN BLOCKS
08	M	GET MESSAGE(06)
08	OMV OO OV	READ(06)
08	O	RECEIVE
09	VVVVVV V	
0A	M	PRINT
0A	M M	SEND MESSAGE(06)
0A	M	SEND(06)
0A	OM O OV	WRITE(06)
0B	O OO OV	SEEK(06)
0B	O	SLEW AND PRINT
0C	VVVVVV V	
0D	VVVVVV V	
0E	VVVVVV V	
0F	VOVVVV V	READ REVERSE
10	O O	SYNCHRONIZE BUFFER
10	VM VVV	WRITE FILEMARKS
11	VMVVVV	SPACE
12	MMMMMMMMMM	INQUIRY
13	VOVVVV	VERIFY(06)
14	VOOVVV	RECOVER BUFFERED DATA
15	OMO OOOOOO	MODE SELECT(06)
16	M MM MO	RESERVE
16	MM M	RELEASE
17	MM M	RELEASE UNIT
18	OOOOOOOO	COPY
19	VMVVVV	MODE SENSE(06)
1B	O	LOAD UNLOAD
1B	O	SCAN
1B	O	STOP PRINT
1B	O OO O	STOP START UNIT
1C	OOOOOOOOOO	RECEIVE DIAGNOSTIC RESULTS
1D	MMMMMMMMMM	SEND DIAGNOSTIC
1E	OO OO OO	PREVENT ALLOW MEDIUM REMOVAL
1F		
20	V VV V	
21	V VV V	
22	V VV V	
23	V VV V	
24	V VVM	SET WINDOW
25	O	GET WINDOW
25	M M M	READ CD-ROM CAPACITY
26	V VV	
27	V VV	
28	O	GET MESSAGE(10)

28	M	MMMM	READ(10)
29	V	VV O	READ GENERATION
2A		O	SEND MESSAGE(10)
2A		O	SEND(10)
2A	M	M M	WRITE(10)
2B		O	LOCATE
2B		O	POSITION TO ELEMENT
2B	O	OO O	ERASE(10)
2D	V	O O	WRITE AND VERIFY(10)
2F	O	OO O	VERIFY(10)
30	O	OO O	SEARCH DATA HIGH(10)
31		O	OBJECT POSITION
31	O	OO O	SEARCH DATA EQUAL(10)
32	O	OO O	SET LIMITS(10)
34		O	GET DATA BUFFER STATUS
34	O	OO O	READ POSITION
35	O	OO O	SYNCHRONIZE CACHE
36	O	OO O	LOCK UNLOCK CACHE
37	O	O	READ DEFECT DATA(10)
38		O O	MEDIUM SCAN
39		OOOOOOOO	COMPARE
3A		OOOOOOOO	COPY AND VERIFY
3B		OOOOOOOOOO	WRITE BUFFER
3C		OOOOOOOOOO	READ BUFFER
3D		O O	UPDATE BLOCK
3E	O	OO O	READ LONG
3F	O	O O	WRITE SAME
42		O	READ TOC
44		O	PLAY AUDIO(10)
46			
47		O	PLAY AUDIO MSF
48		O	PLAY AUDIO TRACK INDEX
49		O	PLAY TRACK RELATIVE(10)
4A			
4B		O	PAUSE RESUME
4C		OOOOOOOOOO	LOG SENSE
4E			
4F			
50			
51			
52			
53			
54			
55	OOO	OOOOOO	MODE SELECT(10)
56			
57			
58			
59			
5A	OOO	OOOOOO	MODE SENSE(10)
5B			
5C			
5D			
5E			
5F			
A0			
A1			
A2			
A3			
A4			
A5		M	MOVE MEDIUM
A5		O	PLAY AUDIO(12)
A6		O	EXCHANGE MEDIUM
A7			
A8		O	GET MESSAGE(12)
A8		OO O	READ(12)
A9		O	PLAY TRACK RELATIVE(12)
AA		O	WRITE(12)
AB			
AC		O	ERASE(12)
AD			
AE		O O	WRITE AND VERIFY(12)
AF		OO O	VERIFY(12)
B0		OO O	SEARCH DATA HIGH(12)
B1		OO O	SEARCH DATA EQUAL(12)
B2		OO O	SEARCH DATA LOW(12)
B3		OO O	SET LIMITS(12)
B4			
B5			
B5		O	REQUEST VOLUME ELEMENT ADDRESS
B6			

B6	O	SEND VOLUME TAG
B7	O	READ DEFECT DATA(12)
B8	O	READ ELEMENT STATUS
B9		
BA		
BB		
BC		

The Disk Manufacturing & Screening Module

Introduction

The Disk Manufacturing & Screening Toolbox is the easiest to use, most versatile, and fastest disk testing product on the market. A true multi-threaded, multiple host bus adapter, multi-drive screening and testing tool, the Disk Screening Toolbox allows you to:

- confirm that the proper drives are in the subsystem
- download new drive firmware if needed
- set all mode pages to your standard
- set block size and capacity
- format
- log and track all error information
- run any type of test, or sequence of tests at **full bus and device speeds**
- log all test process activities, drive information, and any errors to an Access database

All processes and test sequences are created from a graphical user interface – no programming or scripting is required! Time from installation to running a complete test process can be as little as 5 minutes!

The Disk Screening Toolbox will test SCSI, Fibre Channel, iSCSI, and ATAPI interface disk drives. It runs under Windows operating systems, and will work with any SCSI, FC, iSCSI, or ATAPI host bus adapters.

It can simultaneously test all disk drives connected to up to eight HBA's in a given system – up to a maximum of 250 disk drives at a time.

Testing throughput speed will be dependant upon the maximum throughput that the Host Bus Adapters and your system can sustain. The faster your storage subsystem is the better throughput you will obtain. All test speeds can be monitored at any time during testing.

Test Methodology

The Disk Screening Tool looks at the disk test process as three distinct sections or phases,

1. **Pre-Test Actions**
2. **Testing**
3. **Post-Test Actions**

Pre-Test Actions

The pre-test process allows your test to screen the attached drives by the following qualifiers:

Drive Vendor – you can specify that a drive must match a certain vendor code, or the drive will be rejected from the test. Example = “SEAGATE”

Drive Product – you can specify that a drive must match a certain product code, or the drive will be rejected from the test. Example = “ST32123FC”

Drive Firmware Version – you can specify that the firmware version of each drive must match a specified version. If the version does not match the drive under test the drive can either be rejected, or the proper firmware file can be downloaded into the drive.

Number of Primary Defects – a drive can be rejected if it has too many primary (factory) defects.

Number of Grown Defects – a drive can be rejected if it has too many grown defects, or alternatively a low-level format can be issued to try to clear the grown defects.

Drive Mode Pages – all mode page values can be set to conform to a “golden” drive setting.

Drive Block(“Sector”) Size – the current blocksize of the drive can be checked, and the drive can either be rejected if the blocksize is not correct, or the blocksize of the drive can be changed. Perfect for production runs of RAID systems.

Drive Capacity – the number of blocks of the drive under test can be set to match a specified number, or the drive can be rejected if the capacity does not match a specified value.

Clear Log Pages – all Log Page (error and performance) data can be cleared at the beginning of the test process.

Record Mode Page data – all Mode Pages data can be logged at the beginning of the test process.

During the test process, each drive will be individually checked against any or all of the above parameters. Depending upon the setting chosen, a drive that doesn’t meet one of the qualifying parameters can either be removed from the test process at that point, or the condition can be corrected.

Testing

The test process will be made up of one or more tests assembled into test sequences. Each test sequence is defined by selecting test definition parameters from the Test Configuration Menu. Once a test is defined the test is added to the test sequence list.

The following test parameters can be chosen for each test:

Random or Sequential test type – choose whether each drive block is accessed in a random manner, or sequentially through the test block range.

Run test by **Number of Blocks** or by **Time** – the test can be defined to run for a time duration, such as 30 minutes, or by a number of blocks. If specifying a number of blocks, the entire drive may be specified by entering ‘-1’ for the number of blocks to test.

Data Pattern – a pull-down list of data patterns is available to choose from.

Block Overlay on data pattern – the block number can be overlaid on top of any data pattern.

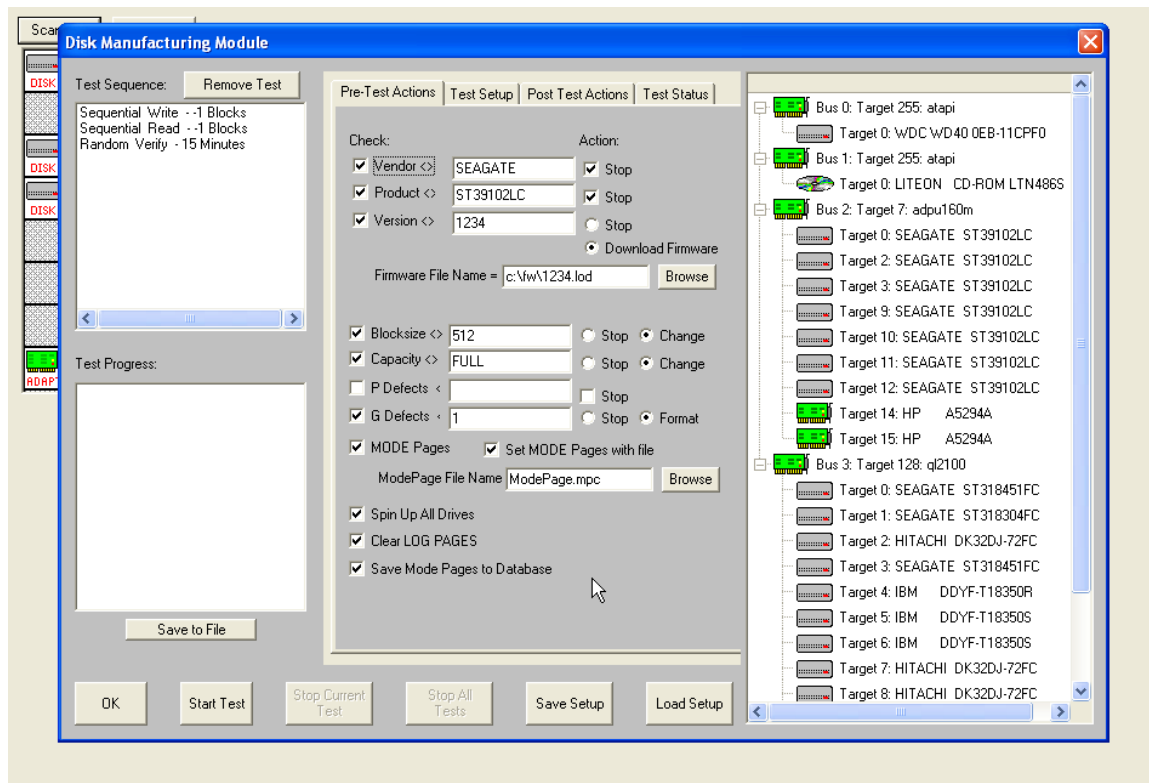
Data Compare on Read – this option can be chosen to confirm that the expected data was read from the disk.

Post-Test Actions

Once a test sequence is finished you may want to perform certain post-test actions. These include spinning all of the disks down, recording each disks MODE PAGE information to the database, and clearing each drives LOG PAGE information.

Getting Started

Choosing the Disk Screening Module from the Disk Menu of the SCSItoolbox will present you with this screen:



The left side of the screen has Test Sequence and Test Progress list boxes, the center part of the screen has several tabbed pages for configuring the test process, and the right side of the screen shows a tree display of all of the adapters and devices attached to the system.

There are several buttons at the bottom of the screen to control the testing process.

Since a test process is made up of **Pre-Test Actions**, **Tests**, and **Post-Test actions**, let's go through each of these sections and see how to configure each.

Pre-Test Action Configuration

All Pre-Test Actions are defined with the following tab page:

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

Check: Action:

Vendor <> SEAGATE Stop

Product <> ST39102LC Stop

Version <> 1234 Stop

Download Firmware

Firmware File Name = c:\fw\1234.lod

Blocksize <> 512 Stop Change

Capacity <> FULL Stop Change

P Defects < Stop

G Defects < 1 Stop Format

MODE Pages Set MODE Pages with file

ModePage File Name ModePage.mpc

Spin Up All Drives

Clear LOG PAGES

Save Mode Pages to Database

Remember that your Pre-Test Actions can be made up of any, all, or none of these choices – it is up to you!

Each Action consists of a **Qualifier**, a **User Input**, and an **Action**. Each Action is described below:

Vendor

Qualifier = INQUIRY VENDOR string

User Input = Enter the Vendor String to select on, for example "SEAGATE" or "IBM"

- please note that all string comparisons will be case sensitive, so if the Vendor string of the disk is SEAGATE you must enter your User Input in all capital letters

Action = Stop – if the User Input does not match the Qualifier the drive will be de-selected and not tested further

Product

Qualifier = INQUIRY PRODUCT string

User Input = Enter the Product String to select on, for example "ST39102".

- please note that all string comparisons will be case sensitive, so if the PRODUCT string of the disk is ST39102 you must enter your User Input in all capital letters. Also note that sub-string matching is implemented, so that if your User Input = ST, this will "match" any Product string containing "ST"

Action = Stop– if the User Input does not match the Qualifier the drive will be de-selected and not tested further

Version

Qualifier = INQUIRY VERSION string

User Input = Enter the Version String to select on, for example 1234.

- please note that all string comparisons will be case sensitive, and that sub-string matching is implemented

Action = Stop– if the User Input does not match the Qualifier the drive will be de-selected and not tested further

Action = Download Firmware – the firmware file specified in the Firmware File Name field will be downloading into the drive before testing. Use the Browse button to search for firmware files on your system. Firmware files must be obtained from your disk drive vendor. All drives that require a firmware download will be downloaded simultaneously.

Blocksize

Qualifier = Block ("Sector") size of the disk drive

User Input = Enter the desired blocksize

Action = Stop– if the User Input does not match the Qualifier the drive will be de-selected and not tested further

Action = Change – the blocksize of the drive will be changed to the blocksize specified by User Input. A low-level format will be run after the block size is changed. All drives that require a format will be formatted simultaneously.

Capacity

Qualifier = Capacity – number of blocks available on the disk drive

User Input = Enter the desired number of blocks – or enter FULL to set drive to it's factory capacity

Action = Stop– if the User Input does not match the Qualifier the drive will be de-selected and not tested further

Action = Change – the capacity of the drive will be changed to that specified by User Input.

P-Defects

Qualifier = Capacity – number of Primary defects on the disk drive

User Input = Enter the maximum number of Primary Defects that you wish to allow

Action = Stop– disk has more Primary Defects than specified by User Input the drive will be de-selected and not tested further

G-Defects

Qualifier = Capacity – number of Grown defects on the disk drive

User Input = Enter the maximum number of Grown Defects that you wish to allow

Action = Stop– disk has more Grown Defects than specified by User Input the drive will be de-selected and not tested further

Action = Format– a low-level format will be run discarding the current Grown Defect List

MODE PAGES

Qualifier = Set the MODE PAGES of the disk drive

User Input = The file name of the Mode Pages file recorded using the Menu Choice "Create ModePage File for DiskScreening"

Action = Each MODE PAGE will be set to the settings recorded from a "golden" drive.

To record a ModePage file for this use, do the following:

1. Select the disk drive to serve as the "golden" drive from the Main drive-selection menu
2. If the MODE PAGE settings of this drive are not set the way you want them, use the SCSItoolbox Mode Page functions to set them the way you want.
3. Choose the "Create ModePage File for DiskScreening" choice from the Disk Menu
4. Specify the file name to save the Mode Page info to, then click the "Save to File" button
5. Use the browse button to select this file for your User Input

Spin Up All Drives

Action = A START command will be sent to each selected disk drive, and the program will wait for the drive to be ready before starting the test sequence

Clear Log Pages

Action = All drive LOG PAGES will be cleared before the test sequence is started. Log Pages contain run-time counters such as read and write error counters, etc. Clearing these pages before testing allows everything to be started "at zero"

Save MODE PAGES to database

Action = All drive MODE PAGE values will be recorded in the database

Test Setup

All Tests are defined with the following tab page:

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

Random or Sequential
 Random Sequential

Type of Test
 Verify Write Write w/ Verify
 Read Write/Read Seek
 FW Download Format Spin Down/Up
 Synchronize External Program

Download File/External Program Executable

Stop Test After
 Time Minutes
 Blocks Blks Start LBA

Fixed/Random Transfers Lengths
 Fixed blk Random max

Test Data
Data Pattern
 Compare on Reads Overlay Block #

This page lets you define test types, and then add the test into the Test Sequence. Simply pick from the various check boxes, then click the “Add this Test to Test Sequence” to quickly define any type of test.

All test definition choices are explained in detail below

Random or Sequential Access

Random

Random Access testing will use a random number generator to choose the starting block of each data transfer. This will create a more strenuous test situation with more acceleration and

deceleration and associated head settling time for each data transfer. There is no guarantee that random access testing will test every block on the disk.

Sequential

Sequential Access testing will process the data blocks of the disk drive in a sequential manner, and will insure that every block on the disk will be tested.

Type of Test

Verify

Uses the SCSI VERIFY command. This command is just like a READ command, with the difference being that no data is actually transferred from the disk to the system. The data is read from the drive platters, and is ECC checked to see if data correction was necessary to accomplish the READ operation.

It can be very efficient to test using the VERIFY command rather than the READ command if you have a large number of drives attached to a bus with a total bus bandwidth of less than the total aggregate bandwidth needed to actually transfer data from all drives. For instance, if you have a 1Gb Fibre Channel bus with 100 drives attached, and each drive can sustain an average read transfer rate of 30MB/sec, using READ commands the Fibre Channel bus will saturate with three drives testing. But using the VERIFY test you can actually have all 100 drives effectively doing read testing at full speed, since no data is being transferred over the FC bus.

Write

The WRITE test will write data to the drive. It is a destructive test – any data previously written on the drive will be lost.

Write with Verify

The Write with Verify issues a write command, then a Verify operation is done to insure that the data is actually readable from the drive. The verify part of the operation does not transfer data across the bus to the host system.

Read

The Read test reads data from the drive and transfers that data back to the host system. This data can be verified by checking the “Compare on Reads” check button.

Write/Read

This test writes data from the host system, then immediately reads the data back to the host. The read data can be compared with the data written using the “Compare on Reads” check button.

FW Download

This test is used to download firmware to the drive during a test sequence. Use the Browse button to specify the file name of the firmware file.

Format

This test is used to format drive(s) during a test sequence. A dialog box allows you to specify a block size change before the format.

Stop Test After

Tests can be run for either an amount of time, or until a number of blocks has been transferred.

Time

This choice allows you to specify the number of minutes to run a test. At the end of this time the actual number of bytes and blocks of data that has been transferred can be seen in the database entry for the test, or can be directly viewed in the Test Status tab page. All drives will test for the same amount of time, but faster drives will have moved more data during the test time.

Blocks

This choice allows you to specify the number of blocks to transfer during the test. Entering -1 for number of blocks will test the entire disk capacity. Faster drives will complete this type of test and move on to the next test in the test sequence while slower drives go at their own pace. All drives will test the specified number of blocks, faster drives will finish sooner.

Data Patterns

13 different data patterns are available for testing – and each pattern can optionally have the data block overlaid, making a total of 26 data patterns ! All Zeros

A repeating pattern of 00000000

All Ones

A repeating pattern of 11111111

Alternating (0/1)

A repeating pattern of 01010101

Alternating (1/0)

A repeating pattern of 10101010

Incrementing

Each block is filled with pattern 00010203040506...

Decrementing

Each block is filled with pattern FFFEFDFCFBFAF9F8...

Block Number

The block number is repeated throughout each block-size buffer

Random

Totally random data using a random number generator with period 16GBytes

User Defined

When the user selects this pattern, the user is prompted to enter a file with the desired pattern. It is recommended this file be created with NotePad. This file can be of any length and the pattern on the file is written to each block. The file must only contain hex digits, namely 0 thru 9, and characters 'a', 'b', 'c', 'd', 'e', 'f'.

Walking Ones

An 8-byte repeating pattern – 00000001 00000010 00000100 00001000 00010000 00100000
01000000 10000000

Walking Zeros

An 8-byte repeating pattern – 11111110 11111101 11111011 11110111 11101111 11011111
10111111 01111111

Alt (0/1) then Alt (1/0)

A two-byte repeating pattern 01010101 10101010

Alt (1/0) then (0/1)

A two-byte repeating pattern 10101010 01010101

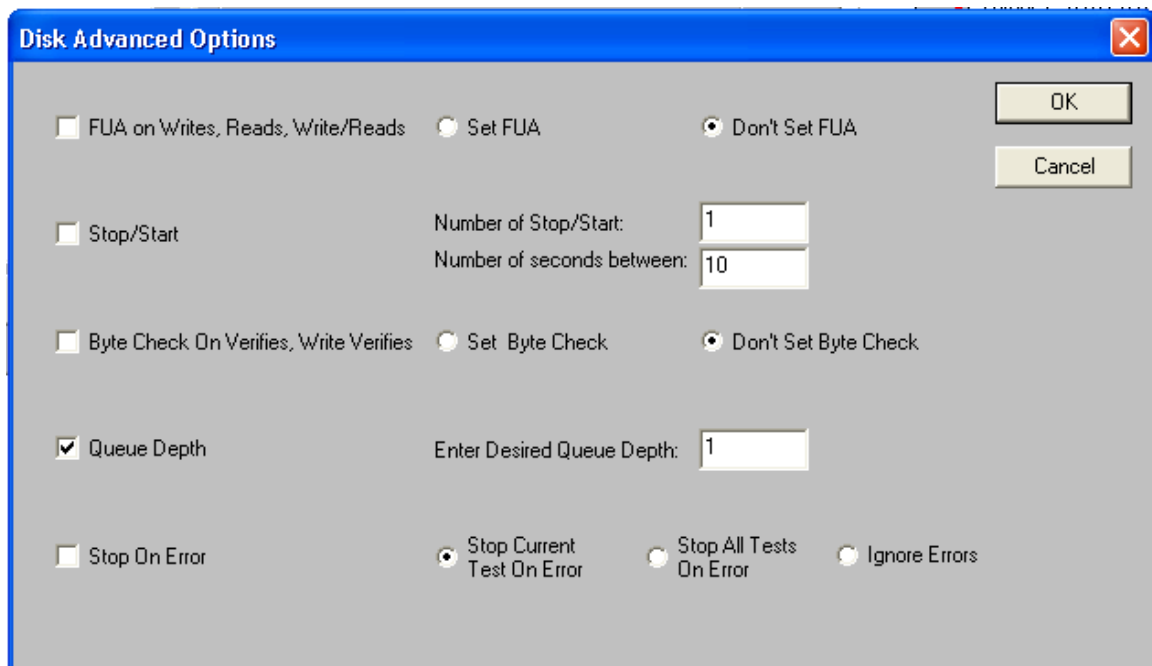
Compare on Reads

Checking this box will cause all data read to be checked that it compares with the expected data. If a miscompare is found the test will be aborted and the miscompare data will be recorded in the database and will be viewable in the Test Status tab page.

Overlay Block Number

Checking this box will overlay the block number over the first n bytes of each block of data.

Advanced Options



The screenshot shows a dialog box titled "Disk Advanced Options" with a blue title bar and a close button (X) in the top right corner. The dialog contains several options and input fields:

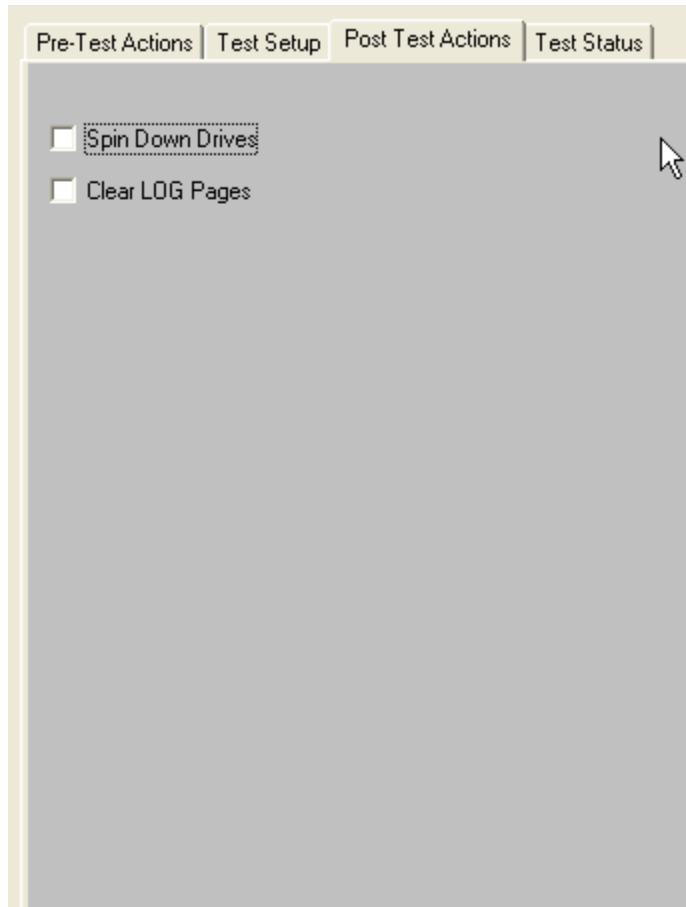
- FUA on Writes, Reads, Write/Reads Set FUA Don't Set FUA
- Stop/Start Number of Stop/Start: Number of seconds between:
- Byte Check On Verifies, Write Verifies Set Byte Check Don't Set Byte Check
- Queue Depth Enter Desired Queue Depth:
- Stop On Error Stop Current Test On Error Stop All Tests On Error Ignore Errors

Buttons for "OK" and "Cancel" are located in the top right corner of the dialog.

The Disk Advanced Options box lets you define test details such as setting Command Tag Queuing depth, turning on FUA (Force Unit Access) on write and read commands, and how DMM should deal with errors that occur during testing.

Post-Test Actions

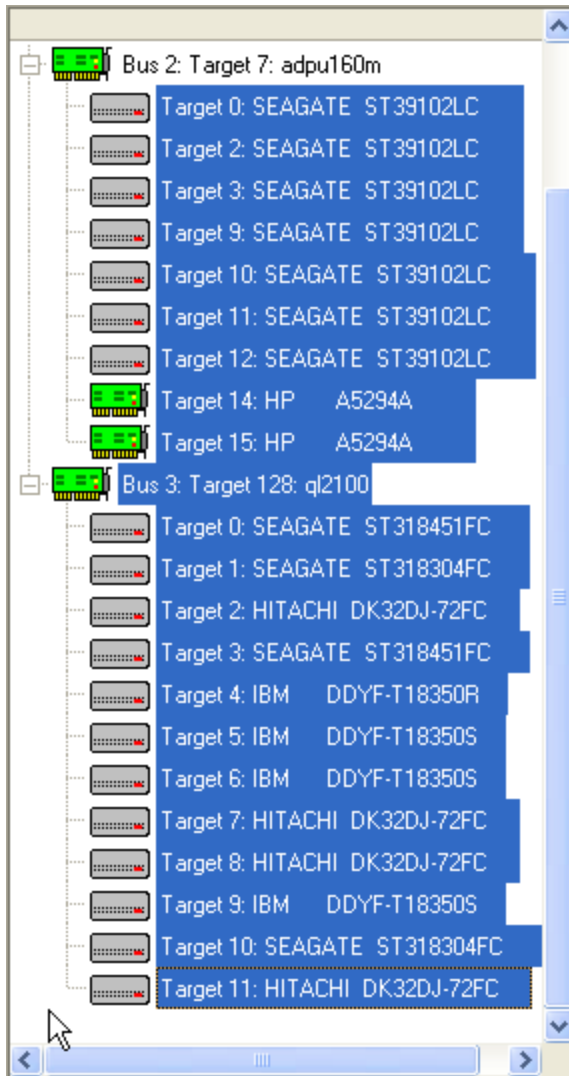
There are currently two choice, as shown below



You can choose to spin all disks down after testing, and to clear all Log Page data.

Selecting Disks to Test

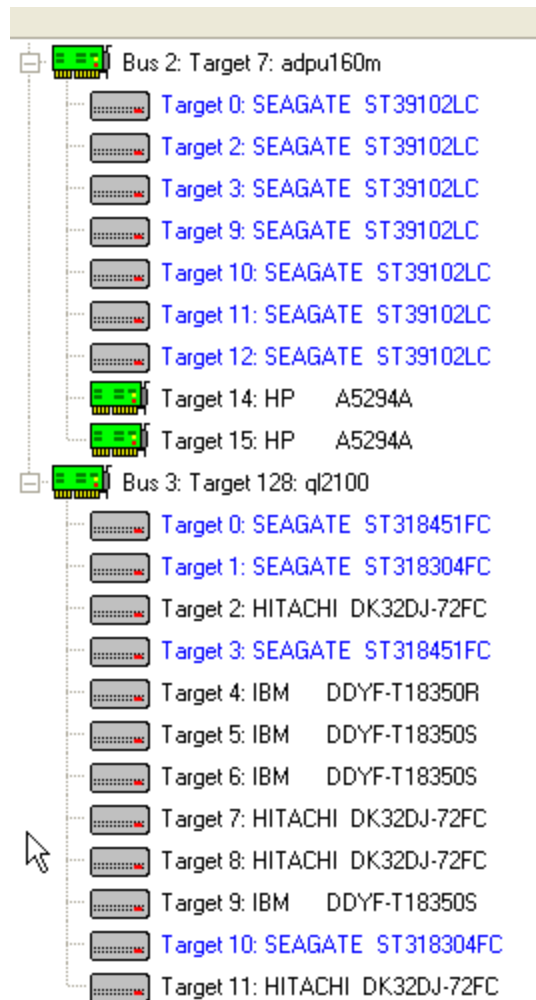
Disks are selected for testing by using the tree display at the right of the Disk Screening Toolbox screen, as shown below



Clicking on a drive selects the drive for testing. Multiple drives may be selected by holding down the Control key while clicking on drives, or a whole series of disks may be selected by clicking on the first drive, then holding down the Shift key while clicking on the last drive.

When a test sequence is started the drives that you have selected will be validated. Any device that is selected, but that is not a disk drive, will be de-selected. If you have specified Pre-Test qualifiers, such as selecting only Seagate drives, all non-qualifying devices will be de-selected.

Here is a picture of the tree display where all drives had been selected, but there was a Pre-Test qualification of only testing SEAGATE drives:



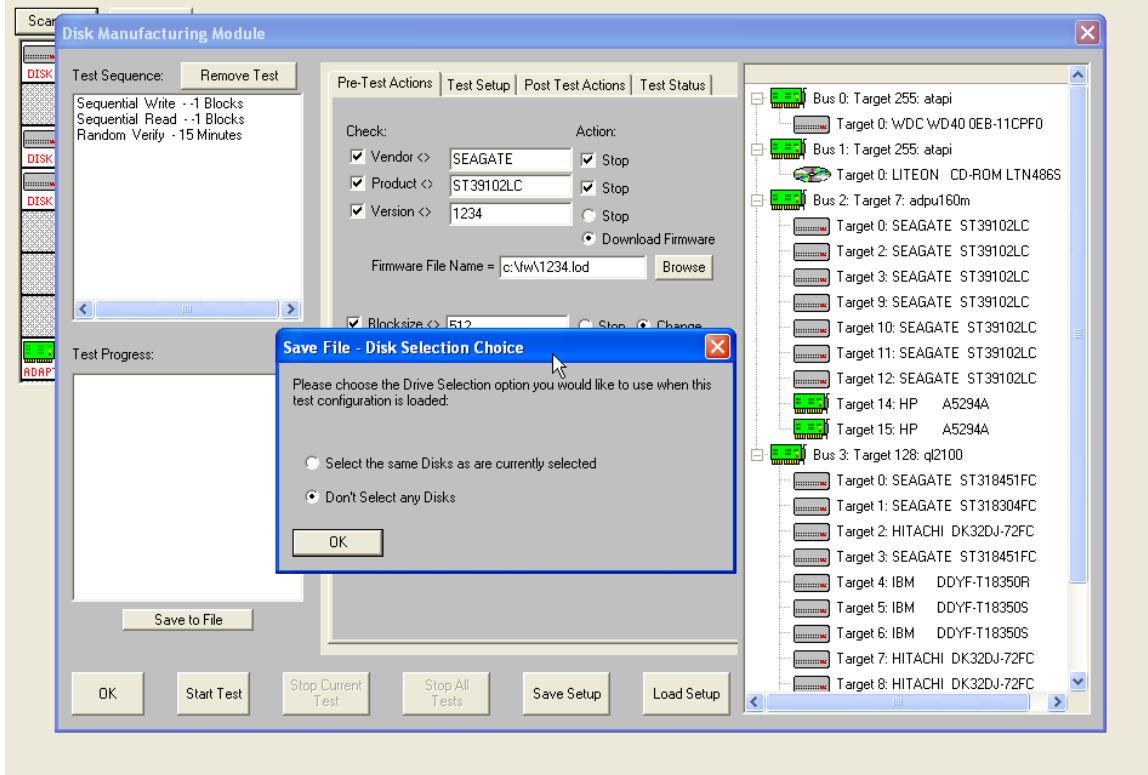
Saving and Loading Test Configurations

Another power feature of the Disk Screening Toolbox is that once a test process has been defined, it can be saved for later recall and use. This is done using the Save Setup and Load Setup buttons.

Saving a Test Configuration

Once you have defined a test process (Pre-Test Actions, Tests, Post-Test Actions, and optionally disk selection) you can save this entire test process by clicking on the Save Setup button.

Once you have specified the file name to save your test definition to, you will be asked if you want to save the disk selection as well – as below:



You choose to not save the disk selection, or you can save the same disks are currently selected.

If you choose to save the currently selected disks, when you load this test definition later it will look for disks at the same addresses. If there was a disk at HA2 Target 3, and there is a disk there now, it will be tested. If it is not there now, that will not cause a problem.

If you choose to not save the disk selection, then when you reload the test definition later you will have to manually select the drives to test again.

Loading a Test Configuration

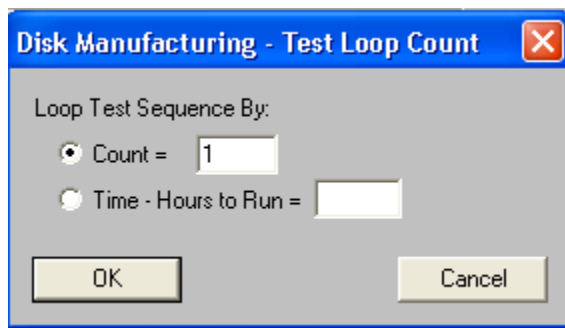
Just click on the Load Setup button to reload a previously defined test configuration.

Starting the Test Process

Once at least one test has been defined and added to the Test Sequence, and once at least one disk drive has been selected, you can start the test process.

Just click on the Start Test button!

The following dialog box will allow you to repeat the test sequence a number of times, or to continually repeat the test sequence until a specified number of hours has passed



The Test Progress box will display a message that the selected drives are being verified, then any Pre-Test Actions will be performed and shown in the Test Progress box.

Once all Pre-Test Actions are finished the Test Sequence will be executed.

Each disk under test has it's own execution thread, and will run the tests in the test sequence as fast as it can. This can lead to faster drives processing through the test sequence faster than others. To check individual drive status, and to pause or stop a test on an individual drive, just right-click on that drive in the tree display

During Testing – individual drive

As a disk is being tested it will be displayed with a Blue highlight in the tree display.

During testing each disk drive can have:

The current test Paused
Testing resumed
The current test stopped
All testing for this drive stopped
The status of the current test displayed

Pausing a Test

Move the cursor over the disk of interest, then click the Right mouse button. Clicking the Pause choice will pause the current test on that drive. The drive will now be displayed with a Magenta highlight in the tree display.

Resuming a Test

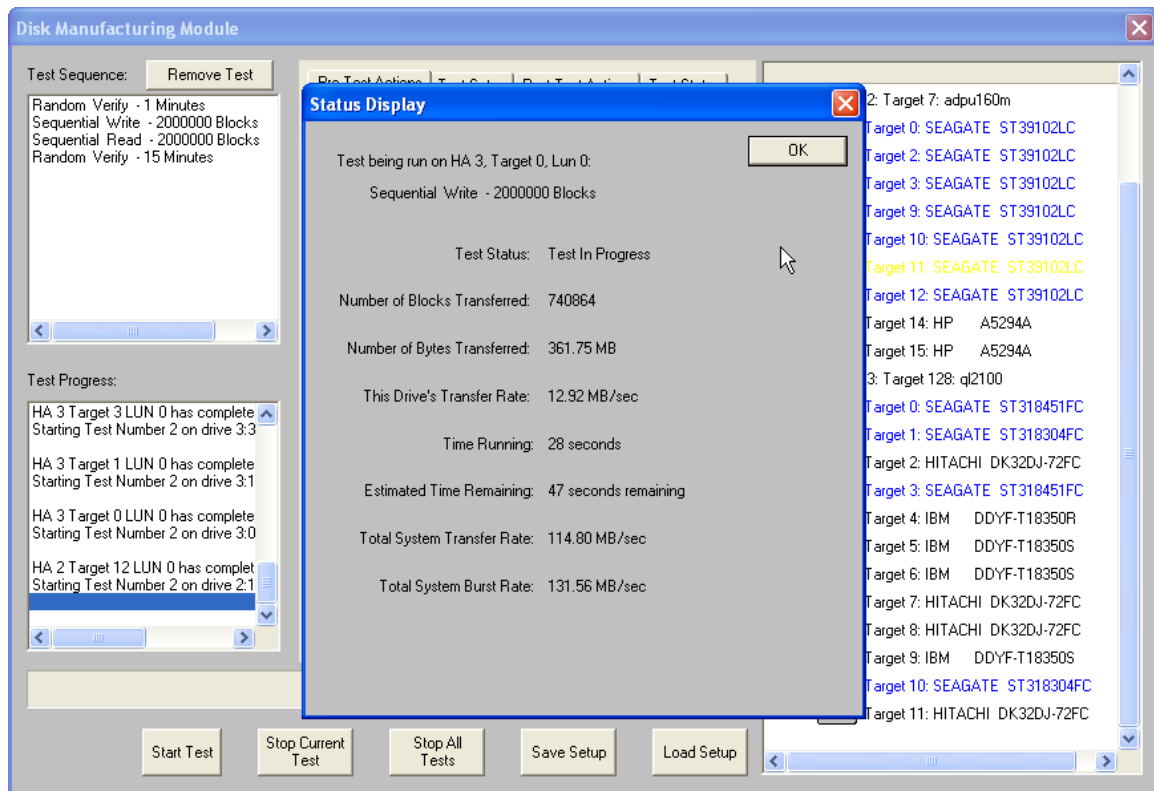
Right click on the paused drive, then click the Resume Test choice to resume testing. The drive is displayed in Blue highlight again

Stopping a Test

Right click on a drive, then choose either the Stop Current Test, or Stop All Tests choice

Viewing Test Status

Right clicking on a drive and choosing the Get Status choice will display the following screen:



Data that is shown is:

- The name of the test that is currently running,
- Test Status
- Number of Blocks Transferred
- Number of Bytes Transferred
- The average transfer rate (over time) achieved
- Time running
- Estimated time remaining
- Total System Transfer Rate
- Total System Burst Rate

All data is updated every few seconds

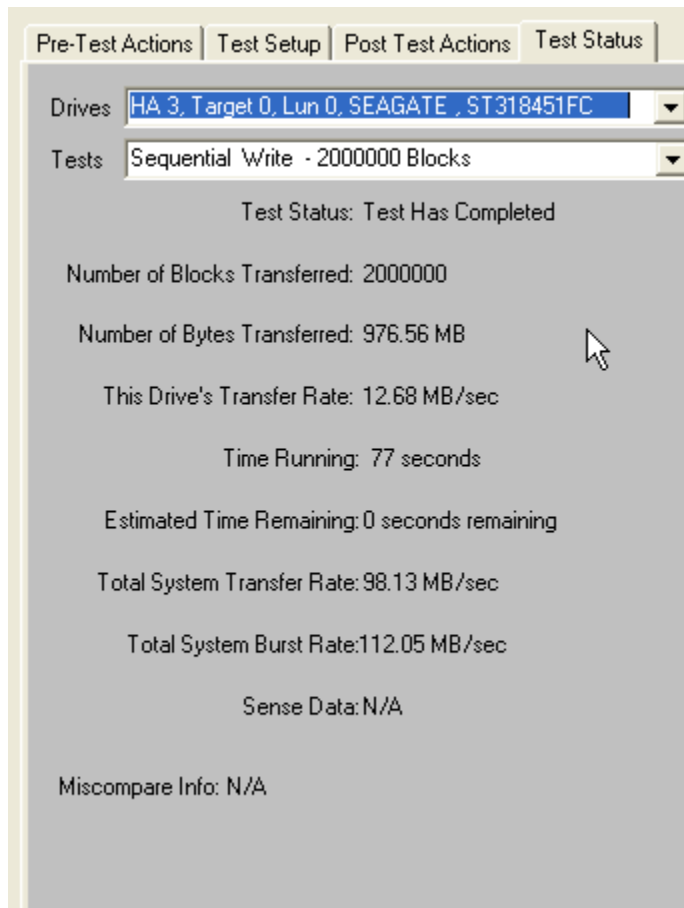
During Testing – All Drives

As the test sequence progresses it is possible to view status on each drive for each test of the sequence. This is done at the Test Status tab page

All performance and error data for each test of each drive is viewable here.

The Test Status Page

The test status pages appears as below:



The Drives Pulldown

Use the Drives Pulldown box to choose which disk

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

Drives: HA 3, Target 0, Lun 0, SEAGATE, ST318451FC

Tests:
HA 2, Target 2, Lun 0, SEAGATE, ST39102LC
HA 2, Target 3, Lun 0, SEAGATE, ST39102LC
HA 2, Target 9, Lun 0, SEAGATE, ST39102LC
HA 2, Target 10, Lun 0, SEAGATE, ST39102LC
HA 2, Target 11, Lun 0, SEAGATE, ST39102LC
HA 2, Target 12, Lun 0, SEAGATE, ST39102LC

Number:
HA 3, Target 0, Lun 0, SEAGATE, ST318451FC
HA 3, Target 1, Lun 0, SEAGATE, ST318304FC
HA 3, Target 3, Lun 0, SEAGATE, ST318451FC
HA 3, Target 10, Lun 0, SEAGATE, ST318304FC

This Drive's Transfer Rate: 12.68 MB/sec

Time Running: 77 seconds

Estimated Time Remaining: 0 seconds remaining

Total System Transfer Rate: 98.13 MB/sec

Total System Burst Rate: 112.05 MB/sec

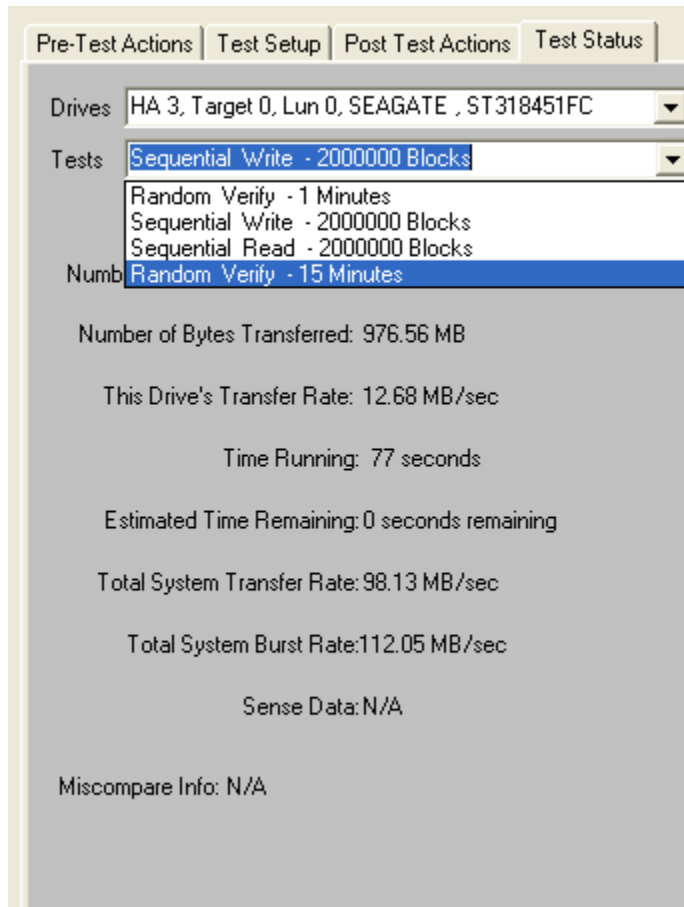
Sense Data: N/A

Miscompare Info: N/A

Just highlight the disk of interest

The Tests Pulldown

Use the Tests Pulldown to choose which test to view



Stopping Tests

Use the **Stop Current Test** or **Stop All Tests** buttons at the bottom of the Disk Screening Toolbox page to stop testing.

Choosing the **Stop Current Test** button will stop the current test running on all drives – if there are more tests in the test sequence each drive will go on to what is the next test for it.

Choosing the **Stop All Tests** button will stop all testing on all drives

Test Results and Output Data

Database data

Test results are written to an Access database – the database file name is DiskScreenDatabase.mdb, and it is located in the folder where SCSItoolbox or the Stand Alone Disk Screening Toolbox is installed.

If you have Access installed on your system you can double-click on this file to view it's contents.

This data is also written as a comma-delimited text file called DiskScreenDatabase.txt

There are two tables – Test and Information.

The Test table stores the following information for each test or test event:

- ComputerName
- DateTimeStamp
- HBA
- TARGET
- Lun
- Vendor
- Product
- SerialNum
- Version
- BlkSz
- Capacity
- Adapter
- Comment
- TestName
- Pattern
- TestResult
- TestStartTime
- TestEndTime
- TimeToDoTest
- NumMBToDev
- NumMBFromDev
- TotMBTrans
- TransRateTo
- TransRateFrom
- TotTransRate
- SenseOnFailure
- MiscompareLBA
- MiscompareOffset
- MiscompareExpected
- MiscompareActual

Here is a (partial!) screen shot of the test table:

Computer	DateTimeStamp	HBA	Tid	Lun	Vendor	Product	SerialNum	Version	BlkSz	Page	Param	Data
gateway	2003 1:11:03 PM	2	0	0	SEAGATE	ST39102LC	LJA15511	HP03	512	1	-1 18 00 E8 00 00 00 00 00 00	00
gateway	2003 1:11:03 PM	2	0	0	SEAGATE	ST39102LC	LJA15511	HP03	512	2	-1 80 80 00 0A 00 00 00 00 00 00 00 00 00	00
gateway	2003 1:11:03 PM	2	0	0	SEAGATE	ST39102LC	LJA15511	HP03	512	3	-1 14 00 00 00 00 0A 00 00 00 06 02 00 00 00	00
gateway	2003 1:11:03 PM	2	0	0	SEAGATE	ST39102LC	LJA15511	HP03	512	4	-1 00 1B 32 0C 00 00 00 00 00 00 00 00 00 00	00
gateway	2003 1:11:03 PM	2	0	0	SEAGATE	ST39102LC	LJA15511	HP03	512	7	-1 00 0B E8 00 00 00 00 00 00 00	00
gateway	2003 1:11:03 PM	2	0	0	SEAGATE	ST39102LC	LJA15511	HP03	512	8	-1 14 00 FF FF 00 00 FF FF FF FF 80 03 00 00	00
gateway	2003 1:11:03 PM	2	0	0	SEAGATE	ST39102LC	LJA15511	HP03	512	10	-1 02 10 00 00 00 00 00 00 00 00	00
gateway	2003 1:11:03 PM	2	0	0	SEAGATE	ST39102LC	LJA15511	HP03	512	26	-1 00 00 00 00 01 00 00 00 04	
gateway	2003 1:11:03 PM	2	0	0	SEAGATE	ST39102LC	LJA15511	HP03	512	28	-1 88 03 00 00 00 00 00 00 01	
gateway	2003 1:11:03 PM	2	0	0	SEAGATE	ST39102LC	LJA15511	HP03	512	0	-1 00 00	
gateway	2003 1:11:03 PM	2	9	0	SEAGATE	ST39102LC	LJA16690	HP03	512	1	-1 18 00 E8 00 00 00 00 00 00 00	
gateway	2003 1:11:03 PM	2	9	0	SEAGATE	ST39102LC	LJA16690	HP03	512	2	-1 80 80 00 0A 00 00 00 00 00 00 00 00 00 00	
gateway	2003 1:11:03 PM	2	9	0	SEAGATE	ST39102LC	LJA16690	HP03	512	3	-1 14 00 00 00 00 0A 00 00 00 06 02 00 00 00	
gateway	2003 1:11:03 PM	2	9	0	SEAGATE	ST39102LC	LJA16690	HP03	512	4	-1 00 1B 32 0C 00 00 00 00 00 00 00 00 00 00	
gateway	2003 1:11:03 PM	2	9	0	SEAGATE	ST39102LC	LJA16690	HP03	512	7	-1 00 0B E8 00 00 00 00 00 00 00	
gateway	2003 1:11:03 PM	2	9	0	SEAGATE	ST39102LC	LJA16690	HP03	512	8	-1 14 00 FF FF 00 00 FF FF FF FF 80 03 00 00	
gateway	2003 1:11:03 PM	2	9	0	SEAGATE	ST39102LC	LJA16690	HP03	512	10	-1 02 10 00 00 00 00 00 00 00 00	
gateway	2003 1:11:03 PM	2	9	0	SEAGATE	ST39102LC	LJA16690	HP03	512	26	-1 00 00 00 00 01 00 00 00 04	
gateway	2003 1:11:03 PM	2	9	0	SEAGATE	ST39102LC	LJA16690	HP03	512	28	-1 88 03 00 00 00 00 00 00 01	
gateway	2003 1:11:03 PM	2	9	0	SEAGATE	ST39102LC	LJA16690	HP03	512	0	-1 00 00	
gateway	2003 1:11:03 PM	2	12	0	SEAGATE	ST39102LC	LJ521079	HP03	512	1	-1 18 00 E8 00 00 00 00 00 00 00	
gateway	2003 1:11:03 PM	2	12	0	SEAGATE	ST39102LC	LJ521079	HP03	512	2	-1 80 80 00 0A 00 00 00 00 00 00 00 00 00 00	
gateway	2003 1:11:03 PM	2	12	0	SEAGATE	ST39102LC	LJ521079	HP03	512	3	-1 14 00 00 00 00 0A 00 00 00 06 02 00 00 00	
gateway	2003 1:11:03 PM	2	12	0	SEAGATE	ST39102LC	LJ521079	HP03	512	4	-1 00 1B 32 0C 00 00 00 00 00 00 00 00 00 00	
gateway	2003 1:11:03 PM	2	12	0	SEAGATE	ST39102LC	LJ521079	HP03	512	7	-1 00 0B E8 00 00 00 00 00 00 00	
gateway	2003 1:11:03 PM	2	12	0	SEAGATE	ST39102LC	LJ521079	HP03	512	8	-1 14 00 FF FF 00 00 FF FF FF FF 80 03 00 00	
gateway	2003 1:11:03 PM	2	12	0	SEAGATE	ST39102LC	LJ521079	HP03	512	10	-1 02 10 00 00 00 00 00 00 00 00	
gateway	2003 1:11:03 PM	2	12	0	SEAGATE	ST39102LC	LJ521079	HP03	512	26	-1 00 00 00 00 01 00 00 00 04	
gateway	2003 1:11:03 PM	2	12	0	SEAGATE	ST39102LC	LJ521079	HP03	512	28	-1 88 03 00 00 00 00 00 00 01	
gateway	2003 1:11:03 PM	2	12	0	SEAGATE	ST39102LC	LJ521079	HP03	512	0	-1 00 00	
gateway	2003 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3C0022VQ	F26D	512	1	-1 18 00 F0 00 00 00 00 00 00 00	
gateway	2003 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3C0022VQ	F26D	512	2	-1 80 80 00 00 00 00 00 00 00 00 00 00 00 00	
gateway	2003 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3C0022VQ	F26D	512	3	-1 02 88 00 00 00 04 00 00 01 5A 02 00 00 01	
gateway	2003 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3C0022VQ	F26D	512	4	-1 00 28 89 0A 00 00 00 00 00 00 00 00 00 00	
gateway	2003 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3C0022VQ	F26D	512	7	-1 00 0B F0 00 00 00 00 00 00 00	
gateway	2003 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3C0022VQ	F26D	512	8	-1 14 00 FF FF 00 00 FF FF FF FF 80 03 00 00	
gateway	2003 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3C0022VQ	F26D	512	10	-1 02 10 00 00 00 00 00 00 02 4B	
gateway	2003 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3C0022VQ	F26D	512	25	-1 00 48 00 00 00 00	
gateway	2003 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3C0022VQ	F26D	512	26	-1 00 00 00 00 01 00 00 00 04	
gateway	2003 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3C0022VQ	F26D	512	28	-1 88 03 00 00 00 00 00 00 01	
gateway	2003 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3C0022VQ	F26D	512	0	-1 00 00 0F 00 00 00	
gateway	2003 1:11:03 PM	3	2	0	HITACHI	DK32DJ-72FC	300A6804	A9A9	512	1	-1 18 00 F0 00 00 00 00 00 01 C2	
gateway	2003 1:11:03 PM	3	2	0	HITACHI	DK32DJ-72FC	300A6804	A9A9	512	2	-1 80 80 00 00 00 00 00 00 00 00 00 00 00	

Text File Output Data

A separate text output file will be created for each device that is tested by the Manufacturing Module. The name of these files is a combination of the device address (bus-target-lun) and the device serial number if available. For instance the example file here is named "4-006-00-3ET06GK5.Log".

This file contains all information about the drive, and about the tests that were run. The results of the Manufacturing test run, as shown below:

```
=====  
>> SCSI Toolbox32, Version 5. 0. 0 (build) 40216, running on \\G64-XP <<  
=====
```

Test Date: 02/23/2004 11:56:44
Device: 4:9:0 SEAGATE ST336732LC Serial: 3ET06GK5

Results:

02/23/2004 11:56:44 TEST 1 of 2:
Write Test; Random Access; for 500000 Blocks
Fixed-Length Transfers of 128 (0x0080) Blocks
Start Block: 0 (0x0)
Data Pattern: Random w/ LBA Overlay

02/23/2004 11:57:04 Test Completed Successfully

02/23/2004 11:57:04 PASSED

02/23/2004 11:57:04 TEST 2 of 2:
Read Test; Random Access; for 500000 Blocks
Fixed-Length Transfers of 128 (0x0080) Blocks
Start Block: 0 (0x0)
Data Pattern: Random w/ LBA Overlay; Data Compare ON

02/23/2004 11:57:35 Test Completed Successfully

02/23/2004 11:57:35 PASSED

```
=====  
>> SCSI Toolbox32, Version 5. 0. 0 (build) 40216, running on \\G64-XP <<  
=====
```

Test Date: 02/23/2004 11:58:13
Device: 4:9:0 SEAGATE ST336732LC Serial: 3ET06GK5

Results:

02/23/2004 11:58:13 TEST 1 of 1:
Read Test; Random Access; for 500000 Blocks
Fixed-Length Transfers of 128 (0x0080) Blocks
Start Block: 0 (0x0)
Data Pattern: Walking Zeros w/ LBA Overlay; Data Compare ON

02/23/2004 11:58:13 CDB = 28 00 03 61 30 3C 00 00 80 00
Miscompare: LBA 0x0361303C at 0x00000000
Exp: 03 61 30 3C EF DF BF 7F FE FD FB F7 EF DF BF 7F
Act: 3C 30 61 03 86 57 E0 06 91 6E 23 2E 0C CC 99 E1
>>> Current Test Aborted <<<

02/23/2004 11:58:13 *** FAIL ***

Other output data

Clicking on the “Save to File” button under the Test Progress window will save the contents of the Test Sequence and Test Progress windows to a file – in text mode.

Here is a sample of this type of output:

Output from Disk Screening Toolbox
The time is Fri Mar 14 10:21:47 2003

Test Sequence:

Random Write/Read - 15 Minutes
Sequential Write - -1 Blocks
Random Read - 60 Minutes

Test Progress:

Verifying selected devices
Device 2:14:0 deselected - not a disk
Device 2:15:0 deselected - not a disk
Device 3:128:0 deselected - not a disk

Starting Pre-Test Actions

Spinning Up drive 2 - 0 - 0
Spinning Up drive 2 - 2 - 0
Spinning Up drive 2 - 3 - 0
Spinning Up drive 2 - 9 - 0
Spinning Up drive 2 - 10 - 0
Spinning Up drive 2 - 11 - 0
Spinning Up drive 2 - 12 - 0
Spinning Up drive 3 - 0 - 0
Spinning Up drive 3 - 1 - 0
Spinning Up drive 3 - 3 - 0
Spinning Up drive 3 - 4 - 0
Spinning Up drive 3 - 5 - 0
Spinning Up drive 3 - 6 - 0
Spinning Up drive 3 - 7 - 0
Spinning Up drive 3 - 8 - 0
Spinning Up drive 3 - 9 - 0
Spinning Up drive 3 - 10 - 0
Spinning Up drive 3 - 11 - 0

Done Pre-Test Actions

Test 1 Started on :

HA 2 Target 0 Lun 0 SEAGATE ST39102LC
HA 2 Target 2 Lun 0 SEAGATE ST39102LC
HA 2 Target 3 Lun 0 SEAGATE ST39102LC
HA 2 Target 9 Lun 0 SEAGATE ST39102LC
HA 2 Target 10 Lun 0 SEAGATE ST39102LC
HA 2 Target 11 Lun 0 SEAGATE ST39102LC
HA 2 Target 12 Lun 0 SEAGATE ST39102LC
HA 3 Target 0 Lun 0 SEAGATE ST318451FC
HA 3 Target 1 Lun 0 SEAGATE ST318304FC
HA 3 Target 3 Lun 0 SEAGATE ST318451FC
HA 3 Target 4 Lun 0 IBM DDYF-T18350R
HA 3 Target 5 Lun 0 IBM DDYF-T18350S
HA 3 Target 6 Lun 0 IBM DDYF-T18350S
HA 3 Target 7 Lun 0 HITACHI DK32DJ-72FC
HA 3 Target 8 Lun 0 HITACHI DK32DJ-72FC

HA 3 Target 9 Lun 0 IBM DDYF-T18350S
HA 3 Target 10 Lun 0 SEAGATE ST318304FC
HA 3 Target 11 Lun 0 HITACHI DK32DJ-72FC

HA 3 Target 8 LUN 0 FAILED test 1 - miscompare on a READ at LBA = 06471482
Starting Test Number 2 on drive 3:8:0

HA 3 Target 11 LUN 0 FAILED test 1 - miscompare on a READ at LBA = 07e426e8
Starting Test Number 2 on drive 3:11:0

HA 2 Target 11 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 2:11:0

HA 2 Target 10 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 2:10:0

HA 2 Target 9 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 2:9:0

HA 2 Target 3 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 2:3:0

HA 2 Target 2 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 2:2:0

HA 2 Target 0 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 2:0:0

HA 3 Target 10 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 3:10:0

HA 3 Target 9 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 3:9:0

HA 3 Target 7 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 3:7:0

HA 3 Target 6 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 3:6:0

HA 3 Target 5 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 3:5:0

HA 3 Target 4 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 3:4:0

HA 3 Target 3 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 3:3:0

HA 3 Target 1 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 3:1:0

HA 3 Target 0 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 3:0:0

HA 2 Target 12 LUN 0 has completed test 1 successfully

Starting Test Number 2 on drive 2:12:0

HA 2 Target 0 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 2:0:0

HA 2 Target 11 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 2:11:0

HA 2 Target 3 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 2:3:0

HA 2 Target 10 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 2:10:0

HA 2 Target 9 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 2:9:0

HA 2 Target 12 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 2:12:0

HA 2 Target 2 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 2:2:0

HA 3 Target 0 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 3:0:0

HA 3 Target 3 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 3:3:0

HA 3 Target 1 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 3:1:0

HA 3 Target 10 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 3:10:0

HA 3 Target 9 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 3:9:0

HA 3 Target 4 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 3:4:0

HA 3 Target 6 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 3:6:0

HA 3 Target 5 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 3:5:0

HA 2 Target 0 LUN 0 has completed test 3 successfully
HA 2 Target 11 LUN 0 has completed test 3 successfully
HA 2 Target 10 LUN 0 has completed test 3 successfully
HA 2 Target 3 LUN 0 has completed test 3 successfully
HA 2 Target 9 LUN 0 has completed test 3 successfully
HA 2 Target 2 LUN 0 has completed test 3 successfully
HA 2 Target 12 LUN 0 has completed test 3 successfully
HA 3 Target 5 LUN 0 was Stopped on test 3
HA 3 Target 6 LUN 0 was Stopped on test 3

HA 3 Target 4 LUN 0 was Stopped on test 3
HA 3 Target 9 LUN 0 was Stopped on test 3
HA 3 Target 10 LUN 0 was Stopped on test 3
HA 3 Target 1 LUN 0 was Stopped on test 3
HA 3 Target 3 LUN 0 was Stopped on test 3
HA 3 Target 0 LUN 0 was Stopped on test 3
HA 3 Target 7 LUN 0 was Stopped on test 2
HA 3 Target 11 LUN 0 was Stopped on test 2
HA 3 Target 8 LUN 0 was Stopped on test 2
ALL TESTS HAVE BEEN COMPLETED

Starting Post-Test Actions
Done Post-Test Actions

The Tape Manufacturing & Screening Module

Introduction

The Tape Manufacturing & Screening Toolbox is the easiest to use, most versatile, and fastest tape testing product on the market. A true multi-threaded, multiple host bus adapter, multi-drive screening and testing tool, the Tape Screening Toolbox allows you to:

- confirm that the proper drives are in the subsystem
- download new drive firmware if needed
- set compression on or off
- create an individual text log file for each device tested
- run **ANY** external program as a part of the test sequence
- log and track all error information
- run any type of test, or sequence of tests at **full bus and device speeds**
- log all test process activities, drive information, and any errors to an Access database

All processes and test sequences are created from a graphical user interface – no programming or scripting is required! Time from installation to running a complete test process can be as little as 5 minutes!

The Tape Screening Toolbox will test SCSI, Fibre Channel, iSCSI, and ATAPI interface tape drives. It runs under Windows operating systems, and will work with any SCSI, FC, iSCSI, or ATAPI host bus adapters.

It can simultaneously test all tape drives connected to up to eight HBA's in a given system – up to a maximum of 250 tape drives at a time.

Testing throughput speed will be dependant upon the maximum throughput that the Host Bus Adapters and your system can sustain. The faster your storage subsystem is the better throughput you will obtain. All test speeds can be monitored at any time during testing.

Test Methodology

The Tape Screening Tool looks at the tape test process as three distinct sections or phases,

4. **Pre-Test Actions**
5. **Testing**
6. **Post-Test Actions**

Pre-Test Actions

The pre-test process allows your test to screen the attached drives by the following qualifiers:

Drive Vendor – you can specify that a drive must match a certain vendor code, or the drive will be rejected from the test. Example = “QUANTUM”

Drive Product – you can specify that a drive must match a certain product code, or the drive will be rejected from the test. Example = “SDLT600”

Drive Firmware Version – you can specify that the firmware version of each drive must match a specified version. If the version does not match the drive under test the drive can either be rejected, or the proper firmware file can be downloaded into the drive.

Rewind All Drives – all drives can be rewound before the test sequence is started.

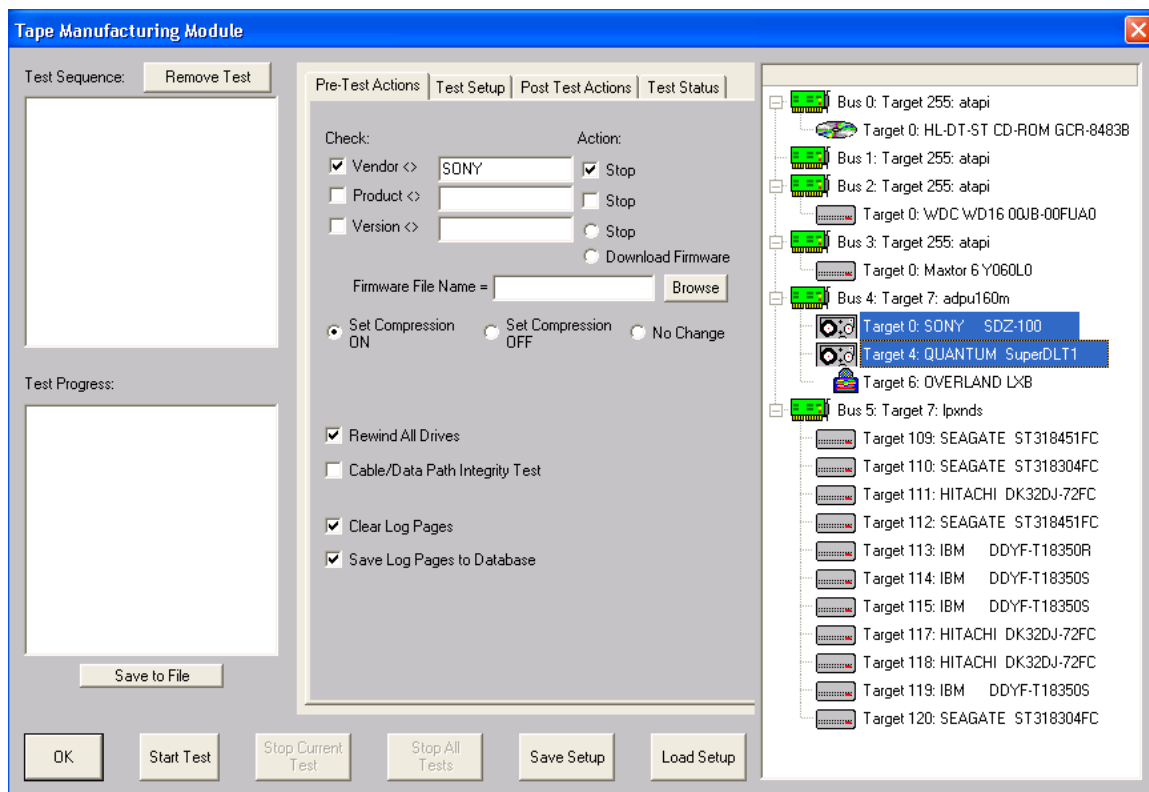
Set Compression On or Off – test with or without data compression enabled.

Clear Log Pages – all Log Page (error and performance) data can be cleared at the beginning of the test process.

Record Mode or Log Page data – all Mode and/or Log page data can be logged at the beginning of the test process.

During the test process, each drive will be individually checked against any or all of the above parameters.

The entire pre-test process definition is accomplished by using the simple GUI choices as show below:



Testing

The test process will be made up of one or more tests assembled into test sequences. Each test sequence is defined by selecting test definition parameters from the Test Configuration Menu. Once a test is defined the test is added to the test sequence list.

The following tests can be used in a test sequence:

Type of Test

- Write
- Read
- Write FileMark
- Read FileMark
- Rewind
- Space Forward by Block
- Space Forward by FileMark
- Space Reverse by Block
- Space Reverse by FileMark
- Space to End Of Data
- Turn Compression On
- Turn Compression Off
- Clear Log Pages
- Save Log Pages
- Synchronize
- Run External Program

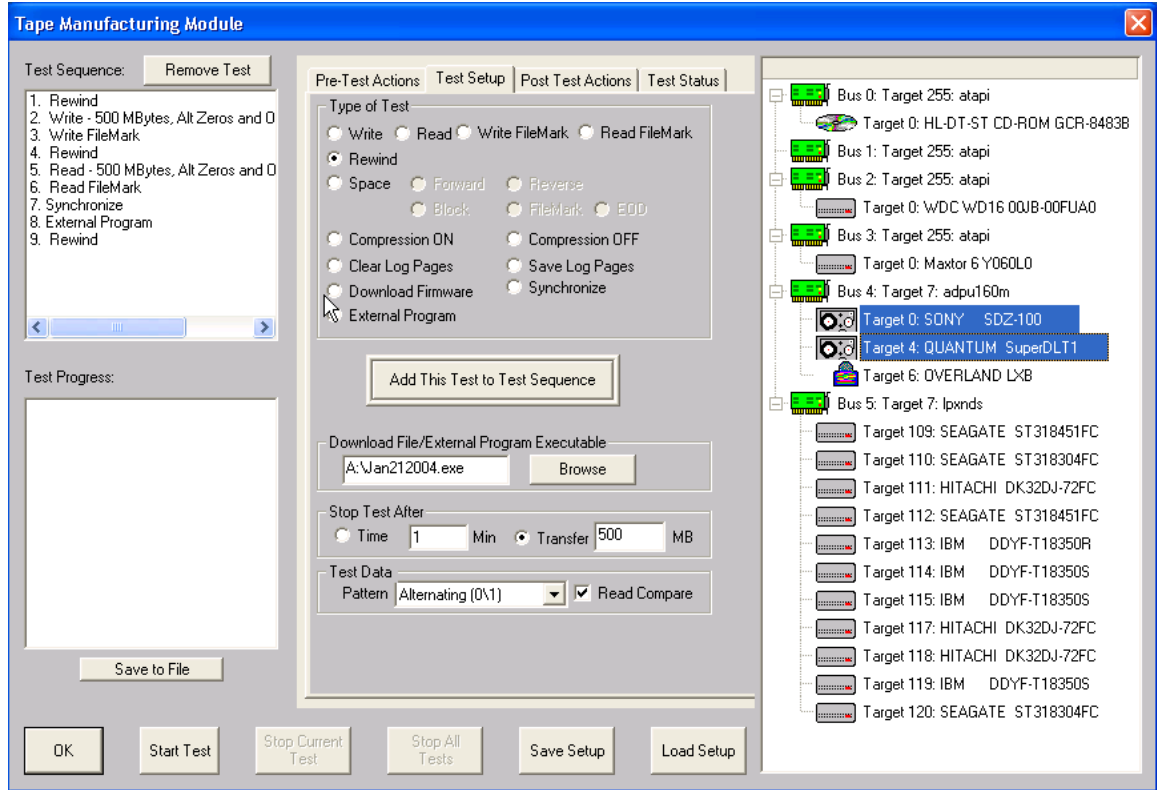
Data Patterns

- 13 data patterns are available
- Data compare on read can be turned on or off

Test Duration

- Read and Write tests can be selected to run for a specified time, or run until a specified amount of data has been transferred.

In the same manner as the pre-test conditions, tests are easily and quickly defined with the GUI as shown below – no editing, no syntax or language to learn, **no way to make a mistake!**

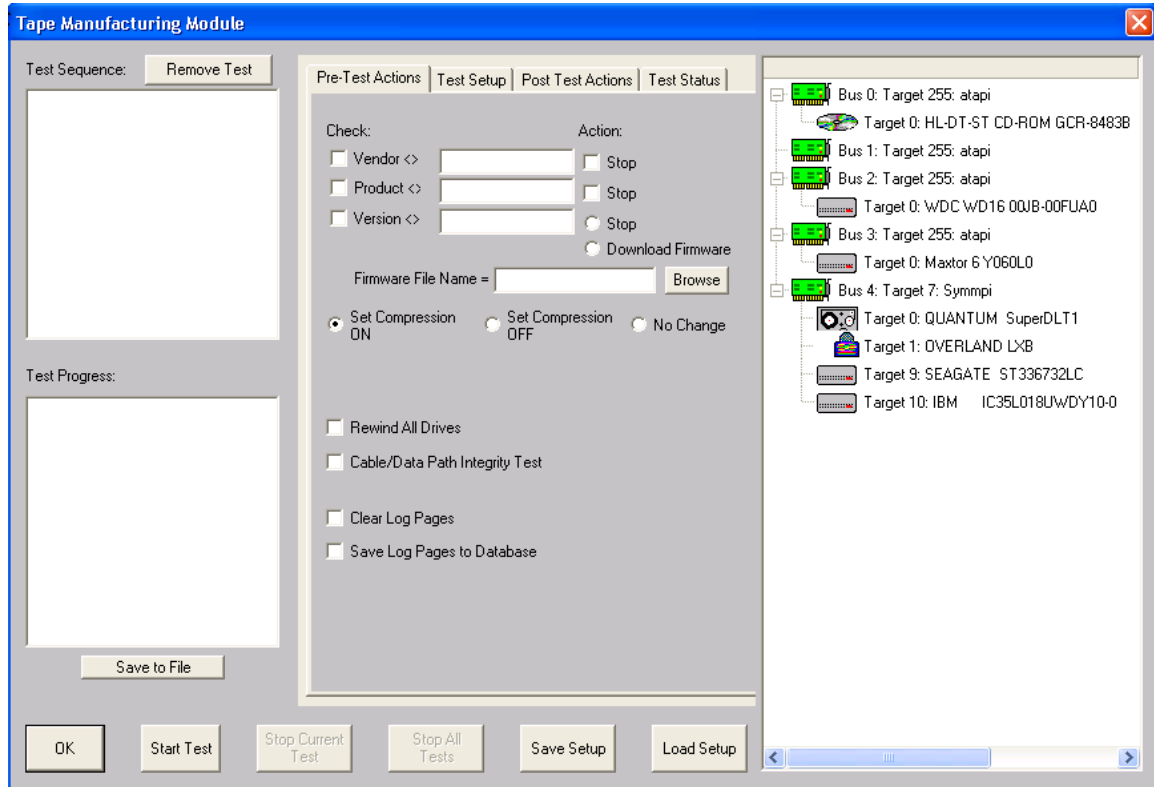


Post-Test Actions

Once a test sequence is finished you may want to perform certain post-test actions. These include rewinding or unloading all of the tapes down, recording each tapes LOG PAGE information to the database, and clearing each drives LOG PAGE information.

Getting Started

Choosing the Tape Screening Module from the Tape Menu of the SCSIttoolbox will present you with this screen:



The left side of the screen has Test Sequence and Test Progress list boxes, the center part of the screen has several tabbed pages for configuring the test process, and the right side of the screen shows a tree display of all of the adapters and devices attached to the system.

There are several buttons at the bottom of the screen to control the testing process.

Since a test process is made up of **Pre-Test Actions, Tests, and Post-Test actions**, let's go through each of these sections and see how to configure each.

Pre-Test Action Configuration

All Pre-Test Actions are defined with the following tab page:

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

Check: Action:

Vendor <> Stop

Product <> Stop

Version <> Stop

Download Firmware

Firmware File Name =

Set Compression ON Set Compression OFF No Change

Rewind All Drives

Cable/Data Path Integrity Test

Clear Log Pages

Save Log Pages to Database

Remember that your Pre-Test Actions can be made up of any, all, or none of these choices – it is up to you!

Each Action consists of a **Qualifier**, a **User Input**, and an **Action**. Each Action is described below:

Vendor

Qualifier = INQUIRY VENDOR string

User Input = Enter the Vendor String to select on, for example "SEAGATE" or "IBM"

- please note that all string comparisons will be case sensitive, so if the Vendor string of the tape is SEAGATE you must enter your User Input in all capital letters

Action = Stop – if the User Input does not match the Qualifier the drive will be de-selected and not tested further

Product

Qualifier = INQUIRY PRODUCT string

User Input = Enter the Product String to select on, for example "ST39102".

- please note that all string comparisons will be case sensitive, so if the PRODUCT string of the tape is ST39102 you must enter your User Input in all capital letters. Also note that sub-string matching is implemented, so that if your User Input = ST, this will "match" any Product string containing "ST"

Action = Stop– if the User Input does not match the Qualifier the drive will be de-selected and not tested further

Version

Qualifier = INQUIRY VERSION string

User Input = Enter the Version String to select on, for example 1234.

- please note that all string comparisons will be case sensitive, and that sub-string matching is implemented

Action = Stop– if the User Input does not match the Qualifier the drive will be de-selected and not tested further

Action = Download Firmware – the firmware file specified in the Firmware File Name field will be downloading into the drive before testing. Use the Browse button to search for firmware files on your system. Firmware files must be obtained from your tape drive vendor. All drives that require a firmware download will be downloaded simultaneously.

Compression

Compression can be explicitly set on, off, or can be left in whatever state the drive is currently in.

Rewind All Drives

Setting this option will rewind all drives before testing begins

Clear Log Pages

Action = All drive LOG PAGES will be cleared before the test sequence is started. Log Pages contain run-time counters such as read and write error counters, etc. Clearing these pages before testing allows everything to be started "at zero"

Save LOG PAGES to database

Action = All drive LOG PAGE values will be recorded in the database

Test Setup

All Tests are defined with the following tab page:

The screenshot shows a software interface with four tabs: "Pre-Test Actions", "Test Setup", "Post Test Actions", and "Test Status". The "Test Setup" tab is active. It contains a "Type of Test" section with several radio button options: "Write" (selected), "Read", "Write FileMark", "Read FileMark", "Rewind", "Space", "Forward", "Reverse", "Block", "FileMark", "EOD", "Compression ON", "Compression OFF", "Clear Log Pages", "Save Log Pages", "Download Firmware", "Synchronize", and "External Program". Below this is a button labeled "Add This Test to Test Sequence". Underneath is a section for "Download File/External Program Executable" with a text input field and a "Browse" button. The "Stop Test After" section has radio buttons for "Time" (selected) and "Transfer", with input fields for "1" Min and "0" MB. The "Test Data" section has a "Pattern" dropdown menu set to "All Zeros (0)" and a "Read Compare" checkbox.

This page lets you define test types, and then add the test into the Test Sequence. Simply pick from the various check boxes, then click the “Add this Test to Test Sequence” to quickly define any type of test.

All test definition choices are explained in detail below

Type of Test

Write

The WRITE test will write data to the drive. It is a destructive test – any data previously written on the drive will be lost.

Read

The Read test reads data from the drive and transfers those data blocks back to the host system. This data can be verified by checking the “Compare on Reads” check button.

Write Filemark

Writes a file mark to the tape at the current position

Read Filemark

Reads a file mark from the tape at the current position

Space Forward Block

Spaces forward by one block

Space Forward FileMark

Spaces forward by one filemark

Space Forward to EOD

Spaces forward to End Of Data

Space Reverse Block

Spaces reverse by one block

Space Reverse FileMark

Spaces reverse by one filemark

Rewind

Rewinds to Beginning Of Tape

FW Download

This test is used to download firmware to the drive during a test sequence. Use the Browse button to specify the file name of the firmware file.

Synchronize

This test is used to pause or stop testing until all drives under test have finished all their tests up to this point. For instance, if you want to test all drives to a certain point, then when all drives have reached that point run an external program to change a temperature chamber, you use the Synchronize test to sync the test steps of all drives.

External Program

This test is used to run any external program as a test step. As in the above example, this could be used to control an environmental chamber, programmable power supplies, etc. Or this could be used to run an external program to extract factory logs or other vendor unique data at a certain point in the test sequence.

Stop Test After

Tests can be run for either an amount of time, or until a number of blocks has been transferred.

Time

This choice allows you to specify the number of minutes to run a test. At the end of this time the actual number of bytes and blocks of data that has been transferred can be seen in the database entry for the test, or can be directly viewed in the Test Status tab page. All drives will test for the same amount of time, but faster drives will have moved more data during the test time.

MB – Megabytes transferred

This choice allows you to specify the number of mega bytes to transfer during the test. Faster drives will complete this type of test and move on to the next test in the test sequence while slower drives go at their own pace. All drives will test the specified number of data, faster drives will finish sooner.

Data Patterns

13 different data patterns are available for testing – and each pattern can optionally have the data block overlaid, making a total of 26 data patterns ! All Zeros

A repeating pattern of 00000000

All Ones

A repeating pattern of 11111111

Alternating (0/1)

A repeating pattern of 01010101

Alternating (1/0)

A repeating pattern of 10101010

Incrementing

Each block is filled with pattern 00010203040506...

Decrementing

Each block is filled with pattern FFFDFDFCFBFAF9F8...

Block Number

The block number is repeated throughout each block-size buffer

Random

Totally random data using a random number generator with period 16GBytes

User Defined

When the user selects this pattern, the user is prompted to enter a file with the desired pattern. It is recommended this file be created with NotePad. This file can be of any length and the pattern on the file is written to each block. The file must only contain hex digits, namely 0 thru 9, and characters 'a', 'b', 'c', 'd', 'e', 'f'.

Walking Ones

An 8-byte repeating pattern – 00000001 00000010 00000100 00001000 00010000 00100000
01000000 10000000

Walking Zeros

An 8-byte repeating pattern – 11111110 11111101 11111011 11110111 11101111 11011111
10111111 01111111

Alt (0/1) then Alt (1/0)

A two-byte repeating pattern 01010101 10101010

Alt (1/0) then (0/1)

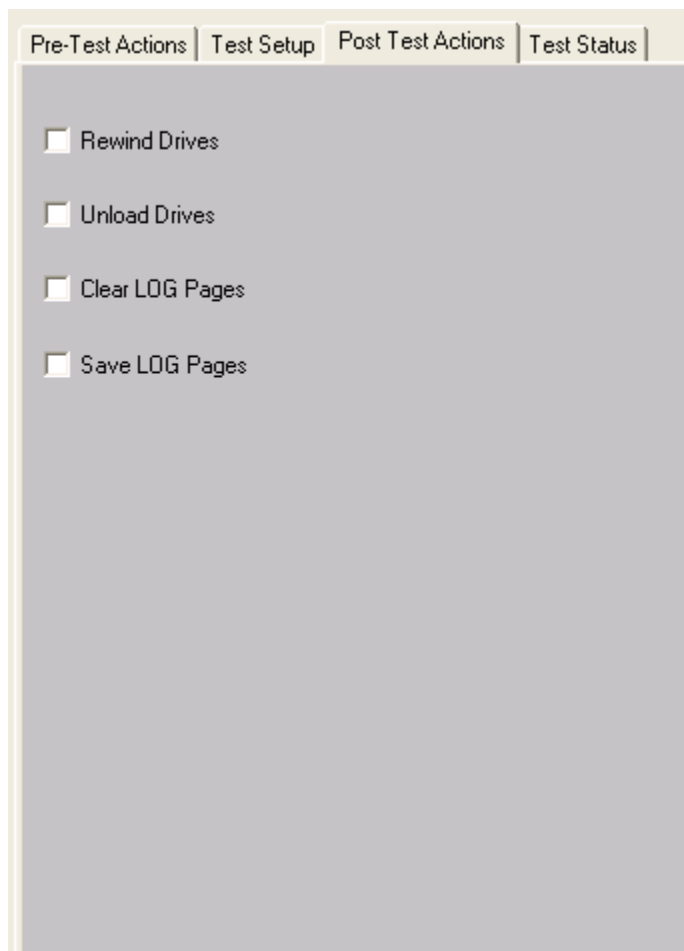
A two-byte repeating pattern 10101010 01010101

Compare on Reads

Checking this box will cause all data read to be checked that it compares with the expected data. If a miscompare is found the test will be aborted and the miscompare data will be recorded in the database and will be viewable in the Test Status tab page.

Post-Test Actions

There are currently four choice, as shown below



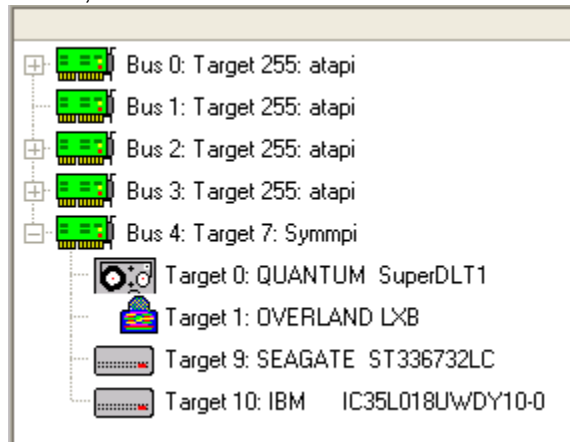
The screenshot shows a software interface with four tabs: "Pre-Test Actions", "Test Setup", "Post Test Actions", and "Test Status". The "Post Test Actions" tab is selected and active. Below the tabs, there is a list of four checkboxes, all of which are currently unchecked:

- Rewind Drives
- Unload Drives
- Clear LOG Pages
- Save LOG Pages

You can choose to rewind or unload all tapes down after testing, and to clear or save all Log Page data.

Selecting Tapes to Test

Tapes are selected for testing by using the tree display at the right of the Tape Screening Toolbox screen, as shown below



Clicking on a drive selects the drive for testing. Multiple drives may be selected by holding down the Control key while clicking on drives, or a whole series of tapes may be selected by clicking on the first drive, then holding down the Shift key while clicking on the last drive. When a test sequence is started the drives that you have selected will be validated. Any device that is selected, but that is not a tape drive, will be de-selected. If you have specified Pre-Test qualifiers, such as selecting only Seagate drives, all non-qualifying devices will be de-selected.

Saving and Loading Test Configurations

Another power feature of the Tape Screening Toolbox is that once a test process has been defined, it can be saved for later recall and use. This is done using the Save Setup and Load Setup buttons.

Saving a Test Configuration

Once you have defined a test process (Pre-Test Actions, Tests, Post-Test Actions, and optionally tape selection) you can save this entire test process by clicking on the Save Setup button.

Once you have specified the file name to save your test definition to, you will be asked if you want to save the tape selection as well .

You choose to not save the tape selection, or you can save the same tapes are are currently selected.

If you choose to save the currently selected tapes, when you load this test definition later it will look for tapes at the same addresses. If there was a tape at HA2 Target 3, and there is a tape there now, it will be tested. If it is not there now, that will not cause a problem.

If you choose to not save the tape selection, then when you reload the test definition later you will have to manually select the drives to test again.

Loading a Test Configuration

Just click on the Load Setup button to reload a previously defined test configuration.

Starting the Test Process

Once at least one test has been defined and added to the Test Sequence, and once at least one tape drive has been selected, you can start the test process.

Just click on the Start Test button!

The Test Progress box will display a message that the selected drives are being verified, then any Pre-Test Actions will be performed and shown in the Test Progress box.

Once all Pre-Test Actions are finished the Test Sequence will be executed.

Each tape under test has it's own execution thread, and will run the tests in the test sequence as fast as it can. This can lead to faster drives processing through the test sequence faster than others. To check individual drive status, and to pause or stop a test on an individual drive, just right-click on that drive in the tree display

During Testing – individual drive

As a tape is being tested it will be displayed with a Blue highlight in the tree display.

During testing each tape drive can have:

- The current test Paused
- Testing resumed
- The current test stopped
- All testing for this drive stopped
- The status of the current test displayed

Pausing a Test

Move the cursor over the tape of interest, then click the Right mouse button. Clicking the Pause choice will pause the current test on that drive. The drive will now be displayed with a Magenta highlight in the tree display.

Resuming a Test

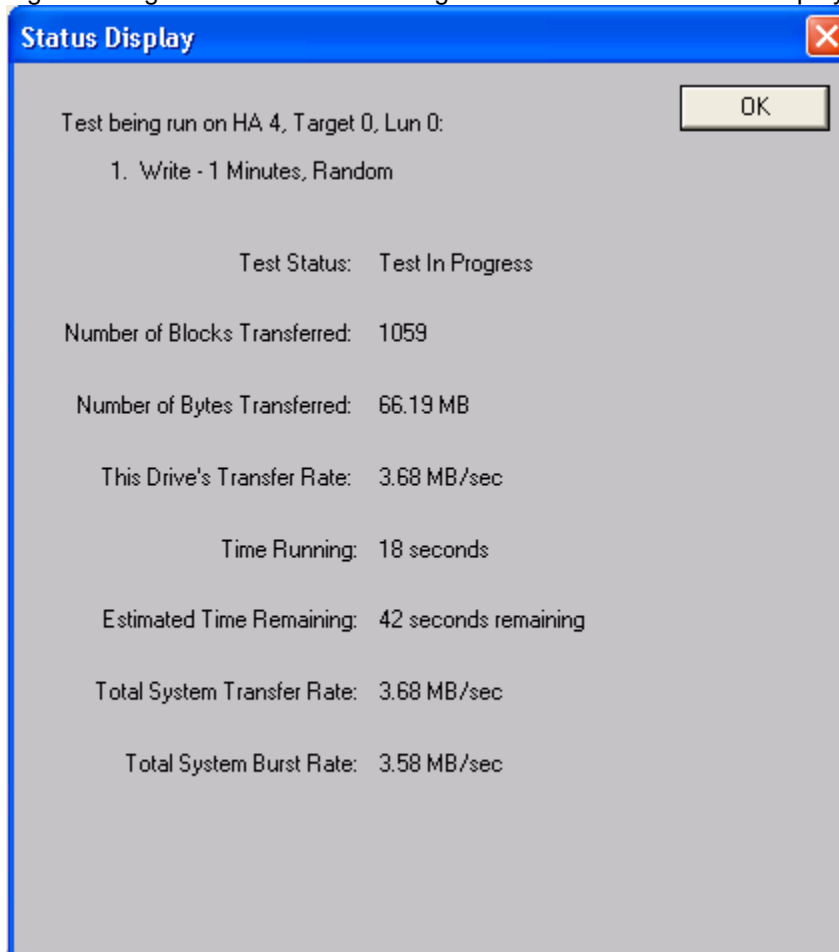
Right click on the paused drive, then click the Resume Test choice to resume testing. The drive is displayed in Blue highlight again

Stopping a Test

Right click on a drive, then choose either the Stop Current Test, or Stop All Tests choice

Viewing Test Status

Right clicking on a drive and choosing the Get Status choice will display the following screen:



Data that is shown is:

The name of the test that is currently running,
Test Status
Number of Blocks Transferred
Number of Bytes Transferred
The average transfer rate (over time) achieved
Time running

Estimated time remaining
Total System Transfer Rate
Total System Burst Rate

All data is updated every few seconds

During Testing – All Drives

As the test sequence progresses it is possible to view status on each drive for each test of the sequence. This is done at the Test Status tab page

All performance and error data for each test of each drive is viewable here.

The Test Status Page

The test status pages appears as below:

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

Drives: HA 4, Target 0, Lun 0, QUANTUM, SuperDLT1

Tests: 1. Write - 1 Minutes, Random

Test Status: Test In Progress

Number of Blocks Transferred: 3856

Number of Bytes Transferred: 241.00 MB

This Drive's Transfer Rate: 6.69 MB/sec

Time Running: 36 seconds

Estimated Time Remaining: 24 seconds remaining

Total System Transfer Rate: 6.69 MB/sec

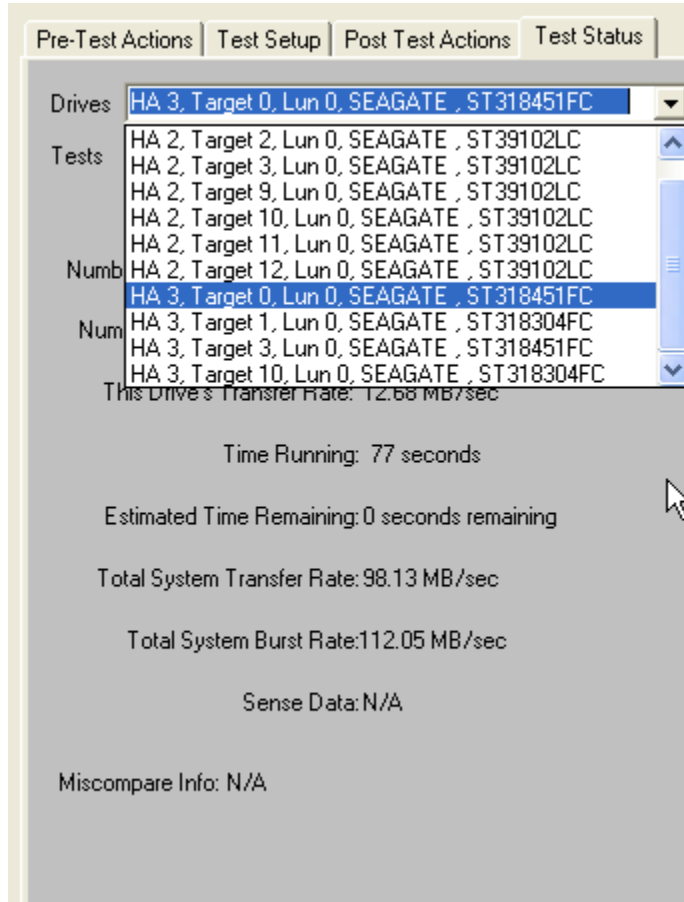
Total System Burst Rate: 6.92 MB/sec

Sense Data: N/A

Miscompare Info: N/A

The Drives Pulldown

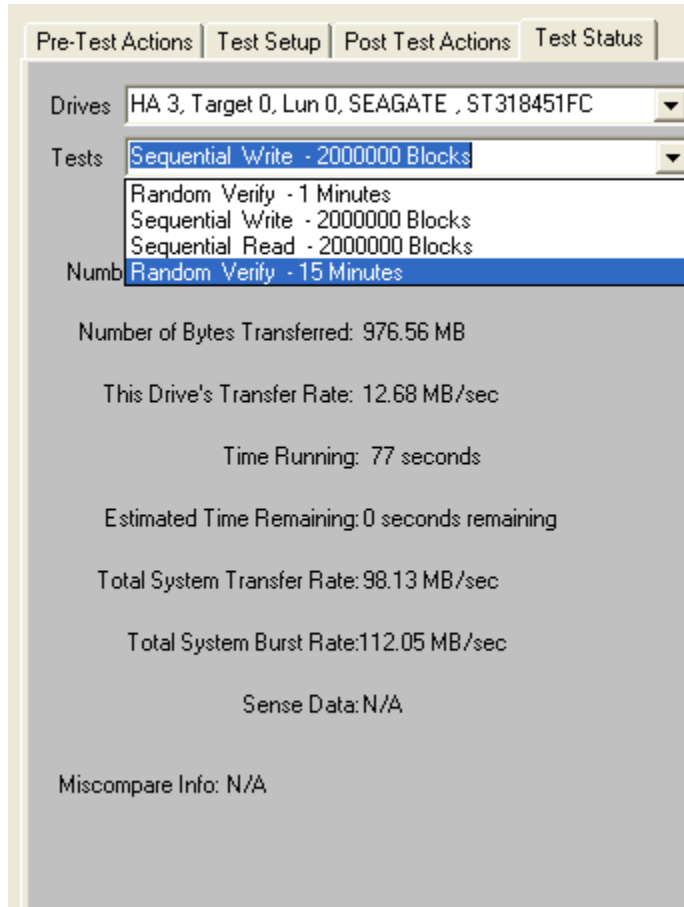
Use the Drives Pulldown box to choose which tape



Just highlight the tape of interest

The Tests Pulldown

Use the Tests Pulldown to choose which test to view



Stopping Tests

Use the **Stop Current Test** or **Stop All Tests** buttons at the bottom of the Tape Screening Toolbox page to stop testing.

Choosing the **Stop Current Test** button will stop the current test running on all drives – if there are more tests in the test sequence each drive will go on to what is the next test for it.

Choosing the **Stop All Tests** button will stop all testing on all drives

Test Results and Output Data

Database data

Test results are written to an Access database – the database file name is TapeScreenDatabase.mdb, and it is located in the folder where SCSItoolbox or the Stand Alone Tape Screening Toolbox is installed.

If you have Access installed on your system you can double-click on this file to view it's contents.

This data is also written as a comma-delimited text file called TapeScreenDatabase.txt

There are two tables – Test and Information.

The Test table stores the following information for each test or test event:

- ComputerName
- DateTimeStamp
- HBA
- TARGET
- Lun
- Vendor
- Product
- SerialNum
- Version
- BlkSz
- Capacity
- Adapter
- Comment
- TestName
- Pattern
- TestResult
- TestStartTime
- TestEndTime
- TimeToDoTest
- NumMBToDev
- NumMBFromDev
- TotMBTrans
- TransRateTo
- TransRateFrom
- TotTransRate
- SenseOnFailure
- MiscompareLBA
- MiscompareOffset
- MiscompareExpected
- MiscompareActual

Here is a (partial!) screen shot of the test table:

Computer	DateTimeStamp	HBA	Tid	Lun	Vendor	Product	SerialNum	Version	BlkSz	Page	Param	Data
gateway	2003 1:11:03 PM	2	0	0	SEAGATE	ST39102LC	LJA15511	HP03	512	1	-1	18 00 E8 00 00 00 00 00 00
gateway	2003 1:11:03 PM	2	0	0	SEAGATE	ST39102LC	LJA15511	HP03	512	2	-1	80 80 00 0A 00 00 00 00 00 00 00 00
gateway	2003 1:11:03 PM	2	0	0	SEAGATE	ST39102LC	LJA15511	HP03	512	3	-1	14 00 00 00 00 0A 00 00 00 D6 02 00 00
gateway	2003 1:11:03 PM	2	0	0	SEAGATE	ST39102LC	LJA15511	HP03	512	4	-1	00 1B 32 0C 00 00 00 00 00 00 00 00
gateway	2003 1:11:03 PM	2	0	0	SEAGATE	ST39102LC	LJA15511	HP03	512	7	-1	00 0B E8 00 00 00 00 00 00 00
gateway	2003 1:11:03 PM	2	0	0	SEAGATE	ST39102LC	LJA15511	HP03	512	8	-1	14 00 FF FF 00 00 FF FF FF FF 80 03 00 00
gateway	2003 1:11:03 PM	2	0	0	SEAGATE	ST39102LC	LJA15511	HP03	512	10	-1	02 10 00 00 00 00 00 00 00 00
gateway	2003 1:11:03 PM	2	0	0	SEAGATE	ST39102LC	LJA15511	HP03	512	26	-1	00 00 00 00 00 01 00 00 00 04
gateway	2003 1:11:03 PM	2	0	0	SEAGATE	ST39102LC	LJA15511	HP03	512	28	-1	88 03 00 00 00 00 00 00 00 01
gateway	2003 1:11:03 PM	2	0	0	SEAGATE	ST39102LC	LJA15511	HP03	512	0	-1	00 00
gateway	2003 1:11:03 PM	2	9	0	SEAGATE	ST39102LC	LJA16690	HP03	512	1	-1	18 00 E8 00 00 00 00 00 00 00
gateway	2003 1:11:03 PM	2	9	0	SEAGATE	ST39102LC	LJA16690	HP03	512	2	-1	80 80 00 0A 00 00 00 00 00 00 00 00
gateway	2003 1:11:03 PM	2	9	0	SEAGATE	ST39102LC	LJA16690	HP03	512	3	-1	14 00 00 00 00 0A 00 00 00 D6 02 00 00
gateway	2003 1:11:03 PM	2	9	0	SEAGATE	ST39102LC	LJA16690	HP03	512	4	-1	00 1B 32 0C 00 00 00 00 00 00 00 00
gateway	2003 1:11:03 PM	2	9	0	SEAGATE	ST39102LC	LJA16690	HP03	512	7	-1	00 0B E8 00 00 00 00 00 00 00
gateway	2003 1:11:03 PM	2	9	0	SEAGATE	ST39102LC	LJA16690	HP03	512	8	-1	14 00 FF FF 00 00 FF FF FF FF 80 03 00 00
gateway	2003 1:11:03 PM	2	9	0	SEAGATE	ST39102LC	LJA16690	HP03	512	10	-1	02 10 00 00 00 00 00 00 00 00
gateway	2003 1:11:03 PM	2	9	0	SEAGATE	ST39102LC	LJA16690	HP03	512	26	-1	00 00 00 00 00 01 00 00 00 04
gateway	2003 1:11:03 PM	2	9	0	SEAGATE	ST39102LC	LJA16690	HP03	512	28	-1	88 03 00 00 00 00 00 00 00 01
gateway	2003 1:11:03 PM	2	9	0	SEAGATE	ST39102LC	LJA16690	HP03	512	0	-1	00 00
gateway	2003 1:11:03 PM	2	12	0	SEAGATE	ST39102LC	LJ521079	HP03	512	1	-1	18 00 E8 00 00 00 00 00 00 00
gateway	2003 1:11:03 PM	2	12	0	SEAGATE	ST39102LC	LJ521079	HP03	512	2	-1	80 80 00 0A 00 00 00 00 00 00 00 00
gateway	2003 1:11:03 PM	2	12	0	SEAGATE	ST39102LC	LJ521079	HP03	512	3	-1	14 00 00 00 00 0A 00 00 00 D6 02 00 00
gateway	2003 1:11:03 PM	2	12	0	SEAGATE	ST39102LC	LJ521079	HP03	512	4	-1	00 1B 32 0C 00 00 00 00 00 00 00 00
gateway	2003 1:11:03 PM	2	12	0	SEAGATE	ST39102LC	LJ521079	HP03	512	7	-1	00 0B E8 00 00 00 00 00 00 00
gateway	2003 1:11:03 PM	2	12	0	SEAGATE	ST39102LC	LJ521079	HP03	512	8	-1	14 00 FF FF 00 00 FF FF FF FF 80 03 00 00
gateway	2003 1:11:03 PM	2	12	0	SEAGATE	ST39102LC	LJ521079	HP03	512	10	-1	02 10 00 00 00 00 00 00 00 00
gateway	2003 1:11:03 PM	2	12	0	SEAGATE	ST39102LC	LJ521079	HP03	512	26	-1	00 00 00 00 00 01 00 00 00 04
gateway	2003 1:11:03 PM	2	12	0	SEAGATE	ST39102LC	LJ521079	HP03	512	28	-1	88 03 00 00 00 00 00 00 00 01
gateway	2003 1:11:03 PM	2	12	0	SEAGATE	ST39102LC	LJ521079	HP03	512	0	-1	00 00
gateway	2003 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3C0022VQ	F26D	512	1	-1	18 00 F0 00 00 00 00 00 00 00
gateway	2003 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3C0022VQ	F26D	512	2	-1	80 80 00 00 00 00 00 00 00 00 00 00
gateway	2003 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3C0022VQ	F26D	512	3	-1	02 B8 00 00 00 04 00 00 01 5A 02 00 00 01
gateway	2003 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3C0022VQ	F26D	512	4	-1	00 28 B9 0A 00 00 00 00 00 00 00 00 00
gateway	2003 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3C0022VQ	F26D	512	7	-1	00 0B F0 00 00 00 00 00 00 00
gateway	2003 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3C0022VQ	F26D	512	8	-1	14 00 FF FF 00 00 FF FF FF FF 80 03 00 00
gateway	2003 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3C0022VQ	F26D	512	10	-1	02 10 00 00 00 00 00 00 02 4B
gateway	2003 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3C0022VQ	F26D	512	25	-1	00 48 00 00 00 00
gateway	2003 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3C0022VQ	F26D	512	26	-1	00 00 00 00 00 01 00 00 00 04
gateway	2003 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3C0022VQ	F26D	512	28	-1	88 03 00 00 00 00 00 00 00 01
gateway	2003 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3C0022VQ	F26D	512	0	-1	00 00 0F 00 00 00
gateway	2003 1:11:03 PM	3	2	0	HITACHI	DK32DJ-72FC	300A6804	A9A9	512	1	-1	18 00 F0 00 00 00 00 00 00 01 C2
gateway	2003 1:11:03 PM	3	2	0	HITACHI	DK32DJ-72FC	300A6804	A9A9	512	2	-1	80 80 00 00 00 00 00 00 00 00 00 00

Text File Output Data

A separate text output file will be created for each device that is tested by the Manufacturing Module. The name of these files is a combination of the device address (bus-target-lun) and the device serial number if available. For instance the example file here is named "5-002-00-0001000125.Log"

This file contains all information about the drive, and about the tests that were run. The results of the Manufacturing test run, as shown below:

```

=====
=====
>> SCSI Toolbox32, Version 5. 0. 0 (build) 40211, running on \\G64-XP <<
=====
=====

```

Test Date: 02/12/2004 14:55:43
Device: 5:2:0 SONY SDZ-100 Serial: 0001000125

Results:
02/12/2004 14:55:43 TEST 1 of 12:
Rewind
02/12/2004 14:55:43 Test Completed Successfully

02/12/2004 14:55:43 PASSED

02/12/2004 14:55:43 TEST 2 of 12:
Write Test; Sequential; for 2000.00 MBytes
Fixed-Length Transfers of 1 (0x0001) Blocks
Data Pattern: Alt Zeros and Ones

02/12/2004 14:56:25 Test Completed Successfully

02/12/2004 14:56:25 PASSED

02/12/2004 14:56:26 TEST 3 of 12:
Write FileMark

02/12/2004 14:56:27 Test Completed Successfully

02/12/2004 14:56:27 PASSED

02/12/2004 14:56:27 TEST 4 of 12:
Write Test; Sequential; for 1000.00 MBytes
Fixed-Length Transfers of 1 (0x0001) Blocks
Data Pattern: Random

02/12/2004 14:57:06 Test Completed Successfully

02/12/2004 14:57:06 PASSED

02/12/2004 14:57:06 TEST 5 of 12:
Write FileMark

02/12/2004 14:57:08 Test Completed Successfully

02/12/2004 14:57:08 PASSED

02/12/2004 14:57:08 TEST 6 of 12:
Rewind

02/12/2004 14:57:21 Test Completed Successfully

02/12/2004 14:57:21 PASSED

02/12/2004 14:57:21 TEST 7 of 12:
Read Test; Sequential; for 2000.00 MBytes
Fixed-Length Transfers of 1 (0x0001) Blocks
Data Pattern: Alt Zeros and Ones

02/12/2004 14:58:07 Test Completed Successfully

02/12/2004 14:58:07 PASSED

02/12/2004 14:58:07 TEST 8 of 12:

Read FileMark

02/12/2004 14:58:07 Test Completed Successfully

02/12/2004 14:58:07 PASSED

02/12/2004 14:58:07 TEST 9 of 12:
Read Test; Sequential; for 1000.00 MBytes
Fixed-Length Transfers of 1 (0x0001) Blocks
Data Pattern: Random

02/12/2004 14:58:47 Test Completed Successfully

02/12/2004 14:58:47 PASSED

02/12/2004 14:58:47 TEST 10 of 12:
Rewind

02/12/2004 14:58:53 Test Completed Successfully

02/12/2004 14:58:53 PASSED

02/12/2004 14:58:53 TEST 11 of 12:
Space Tape Test; Forward to EOD

02/12/2004 14:59:02 Test Completed Successfully

02/12/2004 14:59:02 PASSED

02/12/2004 14:59:02 TEST 12 of 12:
Rewind

02/12/2004 14:59:09 Test Completed Successfully

02/12/2004 14:59:09 PASSED

Other output data

Clicking on the "Save to File" button under the Test Progress window will save the contents of the Test Sequence and Test Progress windows to a file – in text mode.

Here is a sample of this type of output:

Output from Tape Screening Toolbox
The time is Fri Mar 14 10:21:47 2003

Test Sequence:

Random Write/Read - 15 Minutes
Sequential Write - -1 Blocks
Random Read - 60 Minutes

Test Progress:

Verifying selected devices
Device 2:14:0 deselected - not a tape
Device 2:15:0 deselected - not a tape
Device 3:128:0 deselected - not a tape

Starting Pre-Test Actions

Spinning Up drive 2 - 0 - 0
Spinning Up drive 2 - 2 - 0
Spinning Up drive 2 - 3 - 0
Spinning Up drive 2 - 9 - 0
Spinning Up drive 2 - 10 - 0
Spinning Up drive 2 - 11 - 0
Spinning Up drive 2 - 12 - 0
Spinning Up drive 3 - 0 - 0
Spinning Up drive 3 - 1 - 0
Spinning Up drive 3 - 3 - 0
Spinning Up drive 3 - 4 - 0
Spinning Up drive 3 - 5 - 0
Spinning Up drive 3 - 6 - 0
Spinning Up drive 3 - 7 - 0
Spinning Up drive 3 - 8 - 0
Spinning Up drive 3 - 9 - 0
Spinning Up drive 3 - 10 - 0
Spinning Up drive 3 - 11 - 0

Done Pre-Test Actions

Test 1 Started on :

HA 2 Target 0 Lun 0 SEAGATE ST39102LC
HA 2 Target 2 Lun 0 SEAGATE ST39102LC
HA 2 Target 3 Lun 0 SEAGATE ST39102LC
HA 2 Target 9 Lun 0 SEAGATE ST39102LC
HA 2 Target 10 Lun 0 SEAGATE ST39102LC
HA 2 Target 11 Lun 0 SEAGATE ST39102LC
HA 2 Target 12 Lun 0 SEAGATE ST39102LC
HA 3 Target 0 Lun 0 SEAGATE ST318451FC
HA 3 Target 1 Lun 0 SEAGATE ST318304FC
HA 3 Target 3 Lun 0 SEAGATE ST318451FC
HA 3 Target 4 Lun 0 IBM DDYF-T18350R
HA 3 Target 5 Lun 0 IBM DDYF-T18350S
HA 3 Target 6 Lun 0 IBM DDYF-T18350S
HA 3 Target 7 Lun 0 HITACHI DK32DJ-72FC
HA 3 Target 8 Lun 0 HITACHI DK32DJ-72FC
HA 3 Target 9 Lun 0 IBM DDYF-T18350S
HA 3 Target 10 Lun 0 SEAGATE ST318304FC
HA 3 Target 11 Lun 0 HITACHI DK32DJ-72FC

HA 3 Target 8 LUN 0 FAILED test 1 - miscompare on a READ at LBA = 06471482
Starting Test Number 2 on drive 3:8:0

HA 3 Target 11 LUN 0 FAILED test 1 - miscompare on a READ at LBA = 07e426e8

Starting Test Number 2 on drive 3:11:0

HA 2 Target 11 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 2:11:0

HA 2 Target 10 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 2:10:0

HA 2 Target 9 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 2:9:0

HA 2 Target 3 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 2:3:0

HA 2 Target 2 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 2:2:0

HA 2 Target 0 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 2:0:0

HA 3 Target 10 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 3:10:0

HA 3 Target 9 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 3:9:0

HA 3 Target 7 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 3:7:0

HA 3 Target 6 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 3:6:0

HA 3 Target 5 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 3:5:0

HA 3 Target 4 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 3:4:0

HA 3 Target 3 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 3:3:0

HA 3 Target 1 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 3:1:0

HA 3 Target 0 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 3:0:0

HA 2 Target 12 LUN 0 has completed test 1 successfully
Starting Test Number 2 on drive 2:12:0

HA 2 Target 0 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 2:0:0

HA 2 Target 11 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 2:11:0

HA 2 Target 3 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 2:3:0

HA 2 Target 10 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 2:10:0

HA 2 Target 9 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 2:9:0

HA 2 Target 12 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 2:12:0

HA 2 Target 2 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 2:2:0

HA 3 Target 0 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 3:0:0

HA 3 Target 3 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 3:3:0

HA 3 Target 1 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 3:1:0

HA 3 Target 10 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 3:10:0

HA 3 Target 9 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 3:9:0

HA 3 Target 4 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 3:4:0

HA 3 Target 6 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 3:6:0

HA 3 Target 5 LUN 0 has completed test 2 successfully
Starting Test Number 3 on drive 3:5:0

HA 2 Target 0 LUN 0 has completed test 3 successfully

HA 2 Target 11 LUN 0 has completed test 3 successfully

HA 2 Target 10 LUN 0 has completed test 3 successfully

HA 2 Target 3 LUN 0 has completed test 3 successfully

HA 2 Target 9 LUN 0 has completed test 3 successfully

HA 2 Target 2 LUN 0 has completed test 3 successfully

HA 2 Target 12 LUN 0 has completed test 3 successfully

HA 3 Target 5 LUN 0 was Stopped on test 3

HA 3 Target 6 LUN 0 was Stopped on test 3

HA 3 Target 4 LUN 0 was Stopped on test 3

HA 3 Target 9 LUN 0 was Stopped on test 3

HA 3 Target 10 LUN 0 was Stopped on test 3

HA 3 Target 1 LUN 0 was Stopped on test 3

HA 3 Target 3 LUN 0 was Stopped on test 3

HA 3 Target 0 LUN 0 was Stopped on test 3

HA 3 Target 7 LUN 0 was Stopped on test 2

HA 3 Target 11 LUN 0 was Stopped on test 2

HA 3 Target 8 LUN 0 was Stopped on test 2
ALL TESTS HAVE BEEN COMPLETED

Starting Post-Test Actions
Done Post-Test Actions

ATA Command Compliance Testing

INTRODUCTION	369
REQUIREMENTS.....	369
<i>Hardware Requirements</i>	<i>369</i>
<i>User Requirements</i>	<i>370</i>
TEST METHODOLOGY	371
NORMAL AND ERROR STATUS	371
<i>Note:</i>	<i>371</i>
EXCEPTIONS	371
<i>Idle and Standby</i>	<i>371</i>
<i>Sleep</i>	<i>371</i>
<i>Security Freeze</i>	<i>371</i>
DETERMINING THE DRIVE'S CAPABILITIES AND STATE	372
THE STB ATA/SATA COMMAND COMPLIANCE TEST	373
TEST EXECUTION	374
CUSTOMIZING THE TEST SOURCE CODE	377
<i>Opening the project</i>	<i>377</i>
<i>Adding your own commands</i>	<i>377</i>
<i>Defining ATA/SATA commands</i>	<i>377</i>
<i>Special command sequence issues</i>	<i>377</i>
<i>Commands issued during Non-Destructive test.....</i>	<i>378</i>
USING BAM TO CONFIRM TESTING.....	379
THE NEXT STEP – AUTOMATING ATA/SATA COMMAND COMPLIANCE TESTING IN THE STB SUITE DISK MANUFACTURING MODULE (DMM)	381
SUMMARY	383
FURTHER INFORMATION	383

Introduction

Command compliance testing is needed in order to verify proper or expected operation of ATA and SATA storage peripherals. This paper will discuss in particular ATA/SATA disk drive command testing. Testing for packet devices such as CD or DVD drives can be accomplished in the same manner by simply replacing the list of commands used in the test

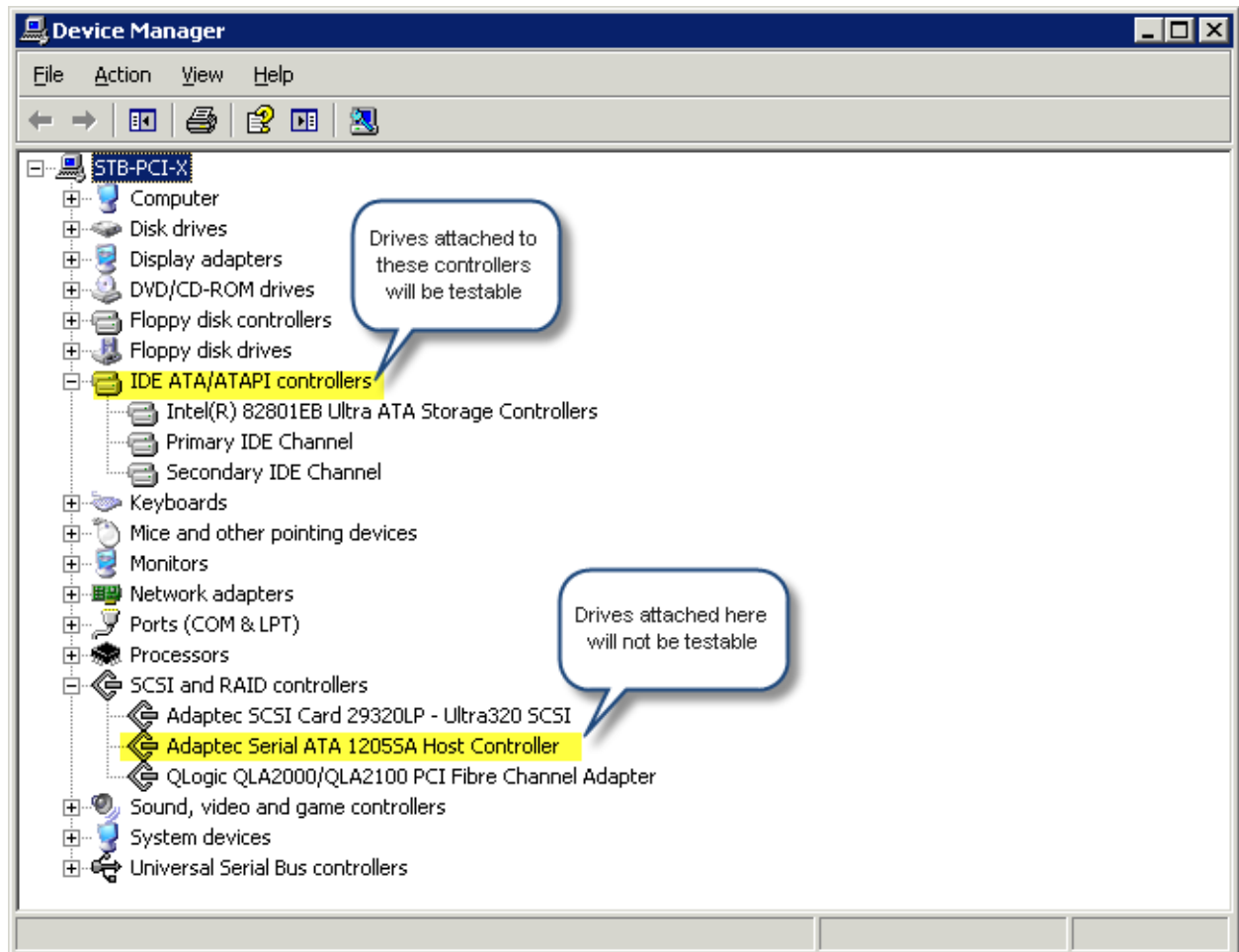
Requirements

Hardware Requirements

To run the example STB ATA and SATA command you will need a computer system running Windows operating system. If you need to test 48-bit ATA or SATA commands you need to be sure that the operating system drivers support these operations. The drivers in Windows Server 2003 do support 48-bit operations, other OS's will need testing to see if they support these commands.

You will also need an ATA or SATA controller which the operating system recognizes as a true ATA Task Register type of controller. Most add-in PCI SATA controllers are seen by Windows as if they are a SCSI host bus adapter rather than an ATA controller.

You can confirm how your operating system views your controller scheme by using Device Manager as shown below – note that the only drives that will be able to be tested for ATA or SATA command compliance **must** be attached to a controller that Windows sees as an *IDE ATA/ATAPI controller*



Note regarding 48-bit commands:

Also, be aware that at this time the Windows operating systems that do support 48-bit operations have a bug in *ATA PASS THROUGH*. After a 48-bit command completes, only the *FIRST* task file is returned (*HOB=0*), but not the second one (*HOB=1*). Thus, *READ NATIVE MAX ADDRESS EXT* can get the low LBA bits 0..23, but not the high LBA bits 24..47.

Once you have disk drives attached to the proper controller you can run the example STB ATA/SATA Command Compliance Test

User Requirements

It is essential that the user of this test be familiar with the ATA and SATA specifications. Documentation can be found at www.t13.org

Test Methodology

A command compliance test must be able to issue any type of ATA or SATA command – including illegal commands. It should give an option to issue data-safe or non-destructive commands or data-destructive commands at the users choice. In order to minimize system hangs it is important that all commands that are issued are able to abort if they have not completed within an expected time.

The test should allow sending any number of commands in sequence. All aspects of these commands must be definable by the user, such as the actual command byte values, the data direction and length (if any), the command length (28-bit or 48-bit commands), and the number of seconds to wait for completion before aborting the command via a timeout.

The test should issue any commands that the user defines, and should gather and display if the command completed normally or with an error. If an error is returned the ATA Task Registers should be displayed and logged for interpretation.

Normal and error status

The results of all commands are returned via the ATA Task Registers. By interpreting the values contained in these registers the test can decide whether a command completed successfully or whether an error occurred. In particular the ERROR and STATUS registers will show error or normal completion. A status value of 0x50 indicates successful completion, a value of 0x51 indicates an error.

Note:

The NOP command always returns an ERROR status. Other commands may as well – be sure to study the ATA and SATA command documentation.

Exceptions

Certain commands will cause the drive to become idle or non-responsive. When a command is issued that results in the drive being left in an IDLE, STANDBY, or SLEEP state other command sequences may be needed to bring the drive back into an ACTIVE state.

Idle and Standby

A drive can be brought back to ACTIVE state from IDLE or STANDBY state by issuing a **READ SECTOR** command. The **CHECK POWER MODE** command can be issued to ascertain the current state of the drive.

Sleep

Once a SLEEP command has been executed by a disk drive it will no longer respond to any commands. In this case either a Bus Reset must be issued, or power to the drive must be cycled.

Security Freeze

A drive which has been placed in a SECURITY FREEZE state must be power cycled to un-freeze the drive. Until this has happened the drive will not respond to any SECURITY FEATURE commands.

Determining the drive's capabilities and state

The IDENTIFY command can be issued to a drive in order to ascertain its capabilities. The data returned by this command will show all information about the drive, such as Model and Serial Number, what version of ATA commands are supported, SATA interface level and speed, whether 48-bit and Security and Power Management features are supported and enabled. The STB Command Compliance will retrieve and interpret all drive IDENTIFY information. The current Power State of the drive can be determined at any point in the test process by issuing the CHECK POWER MODE command and interpreting the results.

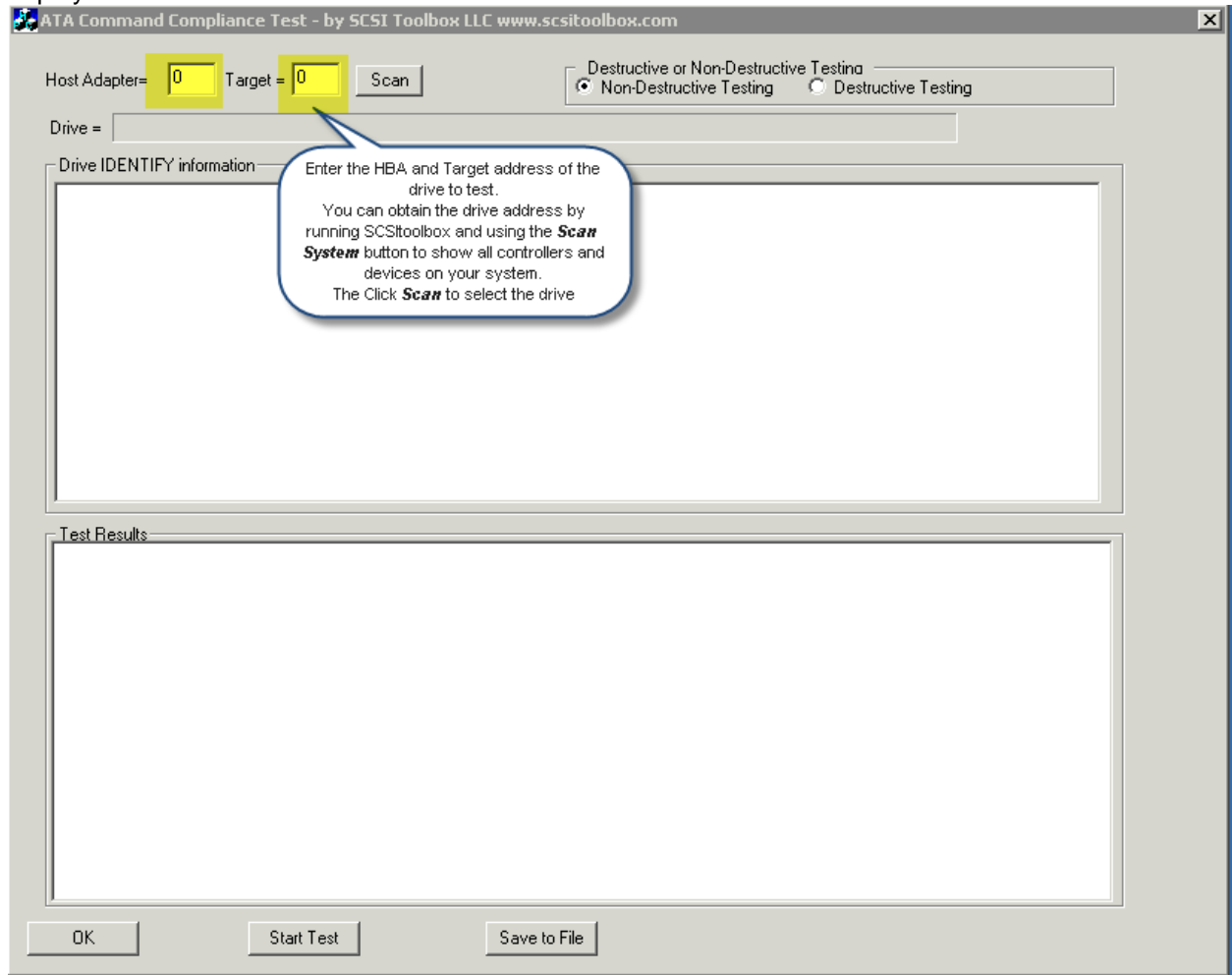
The STB ATA/SATA Command Compliance Test

This test is supplied as a stand-alone executable program which may simply be run. In addition the complete Microsoft Studio vs 6 C++ project is supplied so the user can fully customize the test to their unique requirements. As delivered, the Destructive Testing mode will attempt to preserve data integrity by always issuing a READ command before any WRITE commands, and subsequently using the read data to write back out to the same block. But take care not to run destructive testing on any drive containing important data.

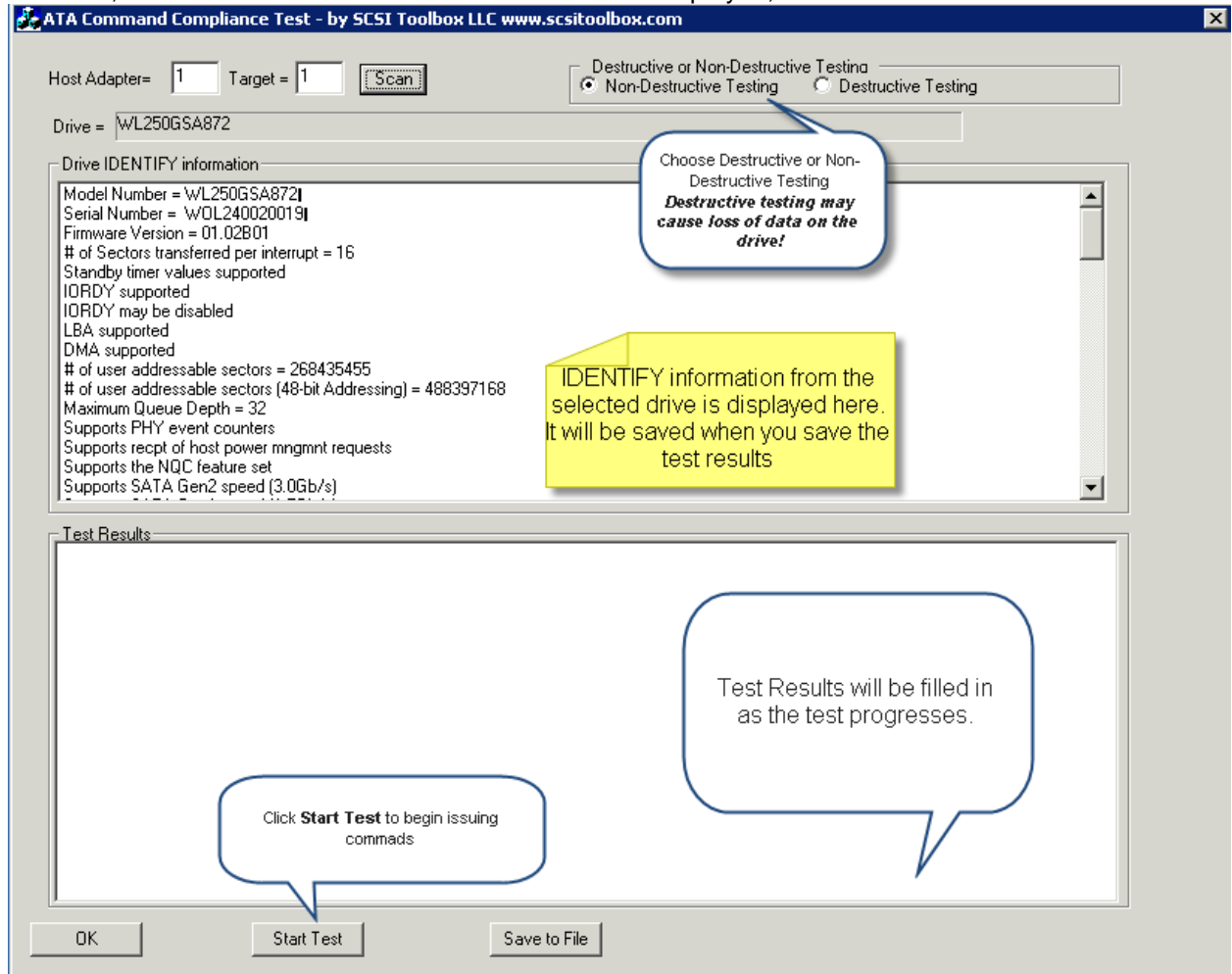
Also – it is **highly recommended** that you do **not** run the compliance test on your system boot drive, or any drives with important data or system files.

Test Execution

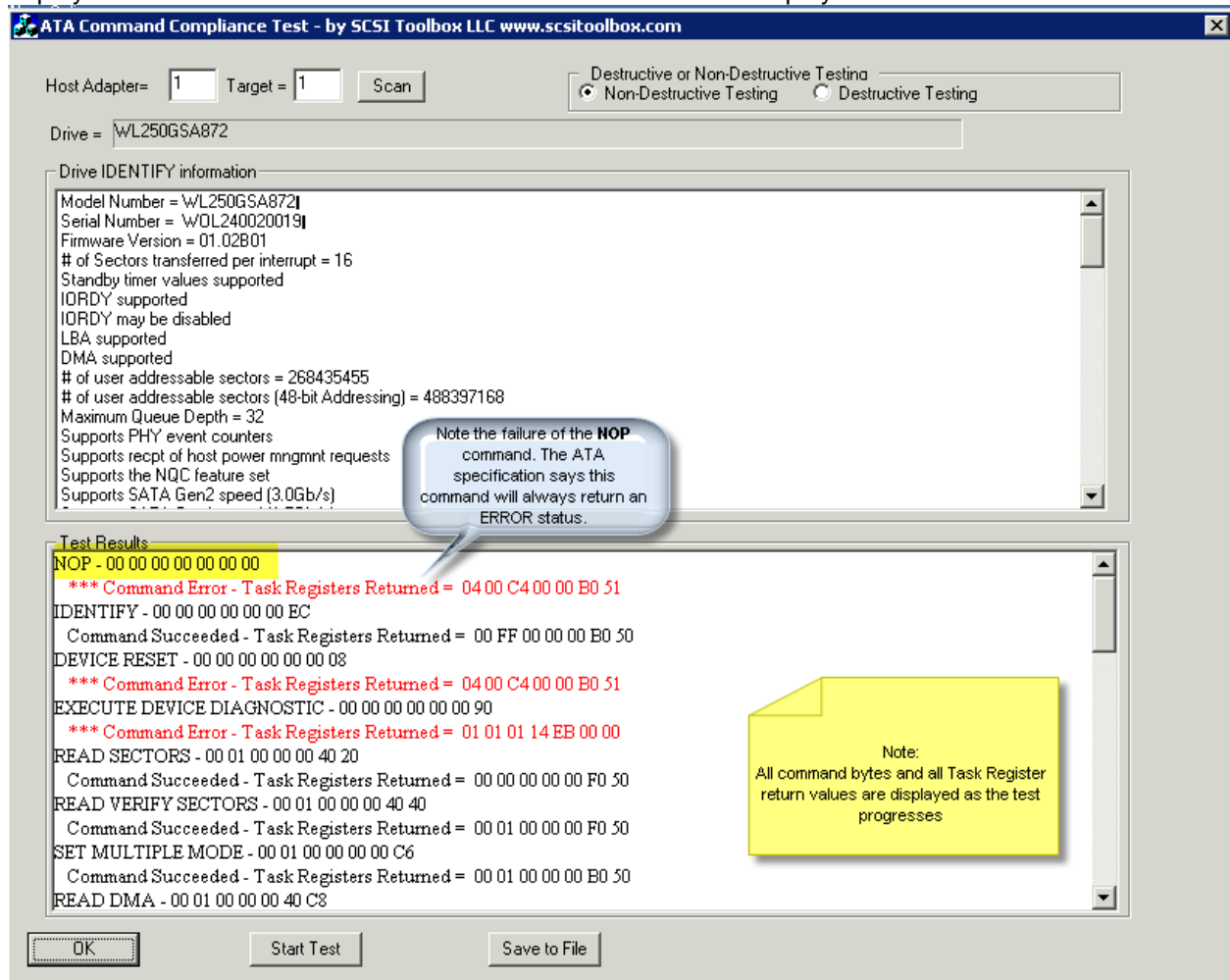
The test application requires that the installed STB Developers Toolbox be version 80428 or above. To start the test simply double-click on the application. The main test dialog will be displayed as below:



Once you have selected a drive and clicked Scan the test will attempt to issue an IDENTIFY command to the drive. If the drive is an ATA or SATA drive, connected to the proper type of controller, the IDENTIFY information for the drive will be displayed, as below:



As the test progresses each command that is issued, and the results of the command are displayed in the Test Results window. If an error occurs it will be displayed in red



Click the Save to File button to store all drive IDENTIFY and test results information to an ascii text file.

Customizing the test source code

Opening the project

The STB ATA/SATA Command Compliance test is written in C++, using Microsoft Visual Studio vs 6.

It uses calls to the STB Developers Toolbox Library dll to communicate with the drive.

Open the project by double-clicking on the project file **ATACompliance.dsw**

The main body of the test code is located in the file **ATAComplianceDlg.cpp**

Adding your own commands

Go to the functions **NonDestructiveCommands()** and **DestructiveCommands()** to customize the commands that are issued during the test.

Defining ATA/SATA commands

The function **BuildCommand()** is used to define commands. The parameters passed to this function are:

```
int CATAComplianceDlg::BuildCommand(  
    int number // the index into the array of commands issued by the test  
    unsigned char b0 – b12 // the bytes that make up the ATA command, in order as defined  
    in the ATA documentation – b0 = function, b1 = sector count...b6 = command, b7-b12  
    the 48-bit command bytes  
  
    int cdlength // either 7 for a 28-bit command or 13 for a 48-bit command  
  
    int datalength // if the command has a data phase – if no data phase this must be set to  
    zero (0) to avoid driver hangs.  
  
    int datadirection // 1 = data in from drive, 0 = data out to drive  
    int timeout // number of seconds after which the command will be aborted if not complete  
)
```

See the pre-defined command descriptions, and modify them to issue whatever command you need to test.

Upon clicking the Start Test button the function **OnStart()** will be called, and will in turn call either **NonDestructiveCommands()** or **DestructiveCommands()** to initialize the array of commands to be sent in the test.

Once the command arrays are filled in the program loops, issuing each command in turn.

Special command sequence issues

As described previously, there are certain sequences of commands which must be issued in order to bring a drive back to active state once it has been put into **STANDBY** or **IDLE** state. You will note as the command sequence is defined there will be a **CHECK POWER MODE** command and a **READ SECTOR(s)** command issued after each type of **IDLE** or **STANDBY** command. After a **SLEEP** command has put the drive into **SLEEP** state a bus reset must be issued. This is handled as a special exception as the test loops through the command list. If the command name == "BUSRESET" DTB will be called to issue a bus reset to the selected drive/bus.

Commands issued during Non-Destructive test

Here is the command sequence issued for the Non-Destructive test

```
BuildCommand(commandcount++, "NOP", 0,0,0,0,0,0,0,0,0,0,0,0,0,7,0,0,5);
// nop will always fail according to spec
BuildCommand(commandcount++, "IDENTIFY", 0,0,0,0,0,0,0xEC,0,0,0,0,0,0,7,512,1,5);
BuildCommand(commandcount++, "DEVICE RESET", 0,0,0,0,0,0,0x08,0,0,0,0,0,0,7,0,0,5);
// devicereset usually fails
BuildCommand(commandcount++, "EXECUTE DEVICE DIAGNOSTIC", 0,0,0,0,0,0,0x90,0,0,0,0,0,0,7,0,1,5);
// read commands
// read dma commands will fail on SATA drives

BuildCommand(commandcount++, "READ SECTORS", 0,0x1,0,0,0,0x40,0x20,0,0,0,0,0,7,512,1,5);
BuildCommand(commandcount++, "READ VERIFY SECTORS", 0,0x1,0,0,0,0x40,0x40,0,0,0,0,0,7,512,1,5);
BuildCommand(commandcount++, "SET MULTIPLE MODE", 0,0x1,0,0,0,0x0,0xC6,0,0,0,0,0,7,0,1,5);
BuildCommand(commandcount++, "READ DMA", 0,0x1,0,0,0,0x40,0xC8,0,0,0,0,0,7,512,1,5);
BuildCommand(commandcount++, "READ DMA QUEUED", 0x01,0,0,0,0,0x40,0xC7,0,0,0,0,0,7,512,1,5);
BuildCommand(commandcount++, "READ MULTIPLE", 0,0x1,0,0,0,0x40,0xC4,0,0,0,0,0,7,512,1,5);
BuildCommand(commandcount++, "READ SECTORS EXT ", 0,0x1,0,0,0,0x40,0x24,0,0,0,0,0,13,512,1,5);
BuildCommand(commandcount++, "READ VERIFY SECTORS EXT", 0,0x1,0,0,0,0x40,0x42,0,0,0,0,0,13,512,1,5);
BuildCommand(commandcount++, "READ DMA EXT", 0,0x1,0,0,0,0x40,0x25,0,0,0,0,0,13,512,1,5);
BuildCommand(commandcount++, "READ MULTIPLE EXT", 0,0x1,0,0,0,0x40,0x29,0,0,0,0,0,13,512,1,5);
BuildCommand(commandcount++, "READ DMA QUEUED EXT", 0x01,0,0,0,0,0x40,0x26,0,0,0,0,0,13,512,1,5);

//idle,standby, and sleep commands –
// with additional commands needed to wake drive back up to active state
BuildCommand(commandcount++, "CHECK POWER MODE", 0,0,0,0,0,0,0xE5,0,0,0,0,0,7,0,1,5);
// sector count == 0xff means drive is active,ready. Sector count == 0x0 means drive is idle,standby, offline

BuildCommand(commandcount++, "IDLE IMMEDIATE", 0,0,0,0,0,0,0xE1,0,0,0,0,0,7,0,1,20);
BuildCommand(commandcount++, "READ SECTORS", 0,0x1,0,0,0,0x40,0x20,0,0,0,0,0,7,512,1,20);
BuildCommand(commandcount++, "CHECK POWER MODE", 0,0,0,0,0,0,0xE5,0,0,0,0,0,7,0,1,5);

BuildCommand(commandcount++, "IDLE ", 0,0,0,0,0,0,0xE3,0,0,0,0,0,7,0,1,20);
BuildCommand(commandcount++, "READ SECTORS", 0,0x1,0,0,0,0x40,0x20,0,0,0,0,0,7,512,1,20);
BuildCommand(commandcount++, "CHECK POWER MODE", 0,0,0,0,0,0,0xE5,0,0,0,0,0,7,0,1,5);

BuildCommand(commandcount++, "STANDBY IMMEDIATE ", 0,0,0,0,0,0,0xE0,0,0,0,0,0,7,0,1,20);
BuildCommand(commandcount++, "READ SECTORS", 0,0x1,0,0,0,0x40,0x20,0,0,0,0,0,7,512,1,20);
BuildCommand(commandcount++, "CHECK POWER MODE", 0,0,0,0,0,0,0xE5,0,0,0,0,0,7,0,1,5);

BuildCommand(commandcount++, "STANDBY ", 0,0,0,0,0,0,0xE2,0,0,0,0,0,7,0,1,20);
BuildCommand(commandcount++, "READ SECTORS", 0,0x1,0,0,0,0x40,0x20,0,0,0,0,0,7,512,1,20);
BuildCommand(commandcount++, "CHECK POWER MODE", 0,0,0,0,0,0,0xE5,0,0,0,0,0,7,0,1,5);

BuildCommand(commandcount++, "SLEEP", 0,0x1,0,0,0,0x40,0xE6,0,0,0,0,0,7,0,1,5);
BuildCommand(commandcount++, "BUSRESET", 0,0,0,0,0,0,0,0,0,0,0,0,0,7,0,0,5);
BuildCommand(commandcount++, "READ SECTORS", 0,0x1,0,0,0,0x40,0x20,0,0,0,0,0,7,512,1,20);
BuildCommand(commandcount++, "CHECK POWER MODE", 0,0,0,0,0,0,0xE5,0,0,0,0,0,7,0,1,5);
```

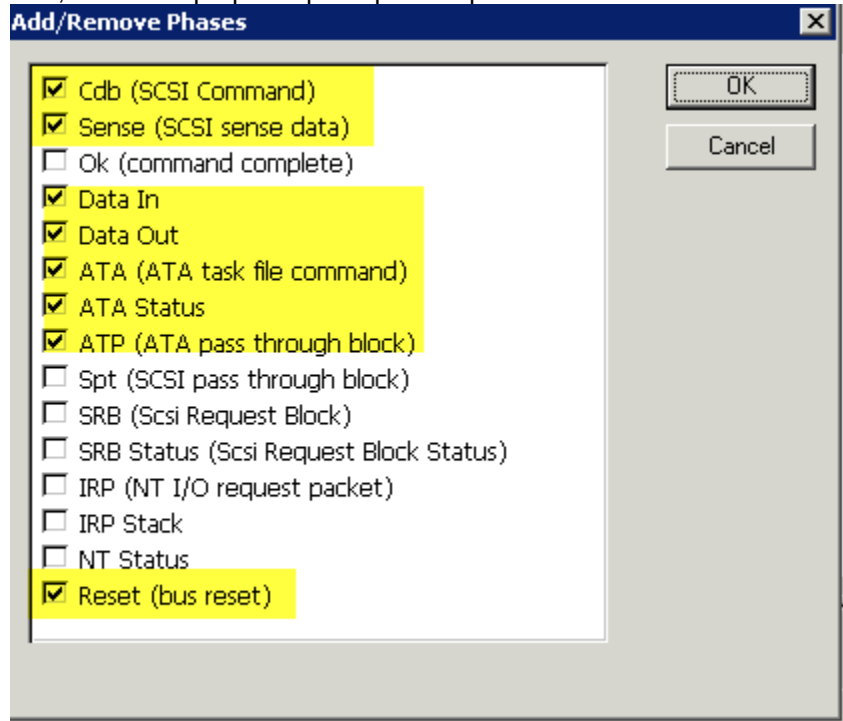
Using BAM to confirm testing

The STB Suite Bus Analyzer Module (BAM) is the perfect tool to confirm the results of the STB ATA/SATA Command Compliance Test.

Simply start BAM, insure that all ATA phase capture options are set, select the drive under test, and start the trace capture.

Here is an example BAM trace showing the Non-Destructive test commands:

First, select the proper capture phase options:



BAM trace capture:

The screenshot shows the 'BAM by SCSI Toolbox, LLC' application window. The interface includes a menu bar (File, Edit, View, Filters, Bus Reset Tool, Help) and a toolbar with various icons. The main area displays a table of trace data. Row 105, representing a 'Reset' phase, is highlighted in yellow. To the right of the table, there are control elements for 'Cap Size (MB)' (set to 32), 'Data Size' (set to 512), a progress indicator, and a 'Sort by Nexus' button.

Ctr	Device	Phase Type	CDB Desc	Data	Data Length	Delta	Cap Size (MB)
102	1:0:1:0	ATA	SLEEP	00 01 00 00 00 40 E6	7 Bytes	5.4 ms	05/03/21
103	1:0:1:0	ATA Status		00 01 00 00 00 F0 50	7 Bytes	8 us	05/03/21
104	1:0:1:0	ATP		28 00 00 00 00 01 00 00 00 00 00 00 05 00 00 00	40 Bytes	1 us	05/03/21
105	1:0:1:0	Reset			0 Bytes	9 us	05/03/21
106	1:0:1:0	ATA	READ	00 01 00 00 00 40 20	7 Bytes	2.4 ms	05/03/21
107	1:0:1:0	ATA Status		00 00 00 00 00 F0 50	7 Bytes	16 us	05/03/21
108	1:0:1:0	Data In		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	512 Bytes	2 us	05/03/21
109	1:0:1:0	ATP		28 00 02 00 00 01 00 00 00 02 00 00 14 00 00 00	40 Bytes	1 us	05/03/21
110	1:0:1:0	ATA	CHECK POWER MODE	00 00 00 00 00 00 E5	7 Bytes	3.3 ms	05/03/21

The next step – automating ATA/SATA Command Compliance testing in the STB Suite Disk Manufacturing Module (DMM)

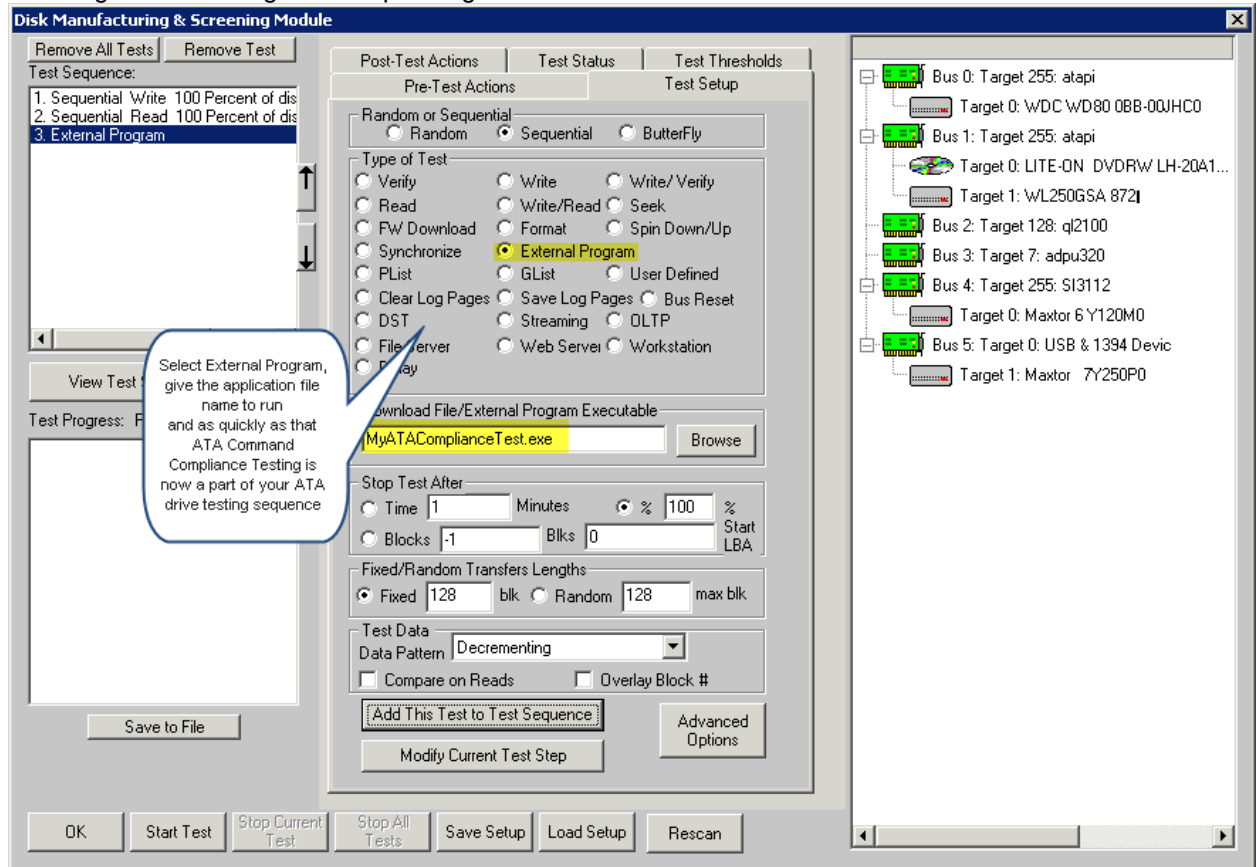
One of the powerful features of the STB Suite Disk Manufacturing Module (DMM) is the ability to call an external program as a test step. When DMM calls an external program by default it passes the address of the drive under test, plus it can also pass any number of other command line parameters.

This means that the sample STB ATA/SATA Command Compliance Test could easily be rewritten to be a Windows command line application with no gui. During a test sequence DMM could run the compliance test for each drive under test. Another very powerful feature of DMM is the ability for any external program or process to signal back to DMM to either log information into each individual drive's logfile, and also to control the further test process on that drive.

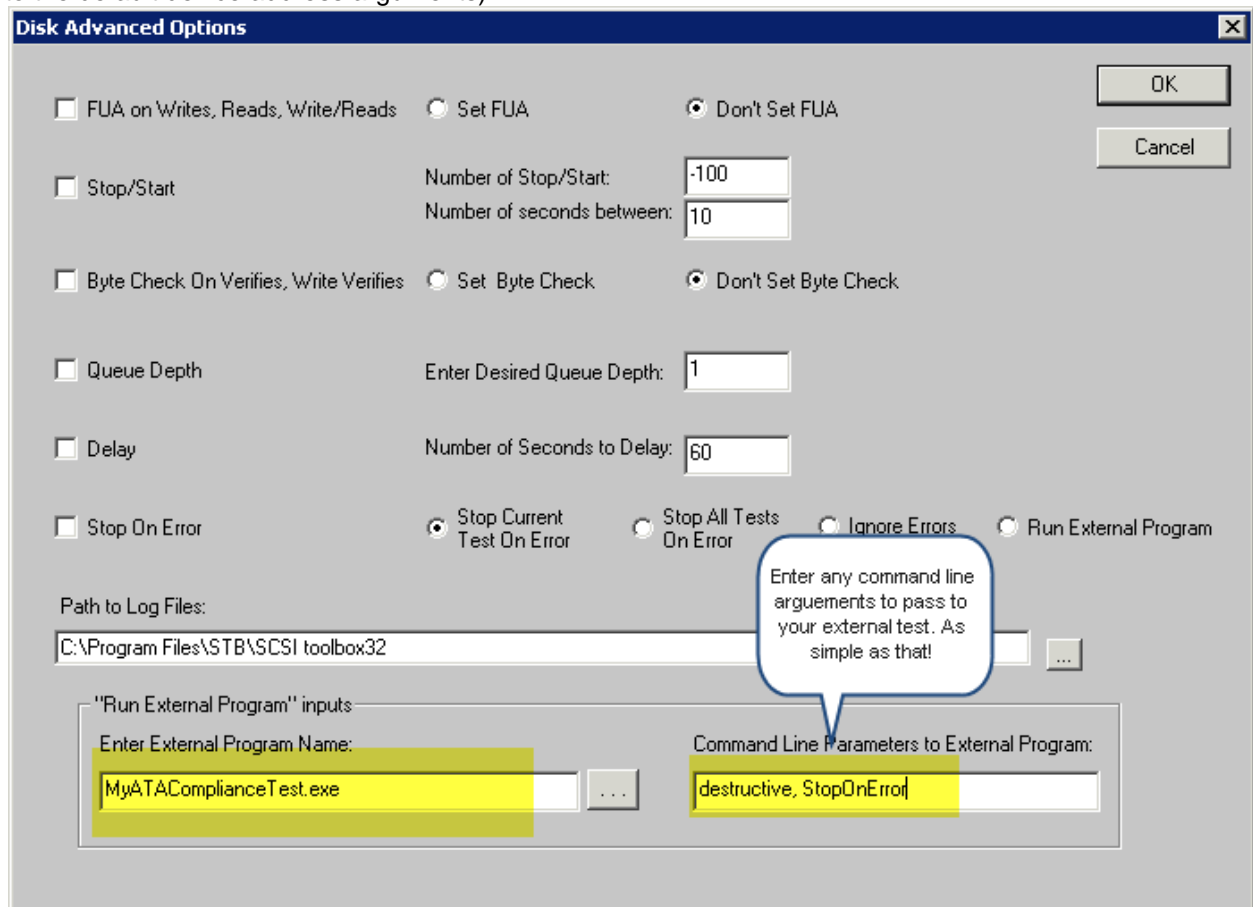
In other words, in a test sequence DMM could start running the ATA/SATA command compliance test, tell the test whether to run destructive or non-destructive (or any other) test options, and the compliance test could log all of its results back into the DMM log files, or, if an error occurs during the compliance test the DMM test sequence for that drive could be aborted.

A second Visual Studio C++ project is supplied which does exactly this. Giving you the ability to instantly build ATA/SATA command compliance testing into your DMM test sequences, with no more than a point- and-click operation. By supplying the complete program project and source code you can easily customize this process to include vendor unique operations, firmware downloads, retrieval and storage of ATA SMART logs – virtually **any** operation that can be accomplished by a powerful ATA pass through command function.

How easy is it to add this type of testing to a complex DMM drive test sequence? As simple as selecting External Program and pointing to the executable to run:



Do you want to pass other command line arguments to your command compliance test? Easy – go to DMM Test Setup Advanced Options and enter the arguments you want passed (in addition to the default device address arguments):



Summary

We have presented a discussion of test methodology to define a complete method of testing ATA and SATA Task Register command compliance.

We have covered issues that must be addressed to accomplish this testing in the Windows operating system environment.

We have described a sample stand-alone application that thoroughly tests ATA or SATA disk drives for any type of command compliance.

We have presented a method to include ATA/SATA disk drive command compliance testing within a larger Disk Manufacturing Module environment.

In addition we have supplied complete project and source files for both the stand-alone compliance test application and the no-gui command-line compliance test.

Further Information

For further information about implementing ATA/SATA, or SCSI/SAS/FC command compliance testing into your test environment please email any questions to support@scsitolbox.com.

We also offer a service to our active Performa customers where we will do hands-on, "live" training via web conferencing. There is no charge for this training – you will be able to meet and discuss any technical issues with members of our development team .

We would be happy to arrange a web conference with your testing team – to set up a web meeting please call Jeremy Wolfe at 720-249-2641.

