SCSI Toolbox, LLC
PO Box 620520
Littleton, CO 80162
www.scsitools.com
303.972.2072

SCSI Toolbox LLC
SCSItoolbox32 Manual
Version 4.1

Introduction

The SCSItolbox32 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 9x, Windows NT, Windows 2000, and Windows XP 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 includes a complete Manufacturing environment, as well as development tools to develop test routines using Microsoft Visual Basic or Visual C++.

Installation

Insert the SCSItoolbox32 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".

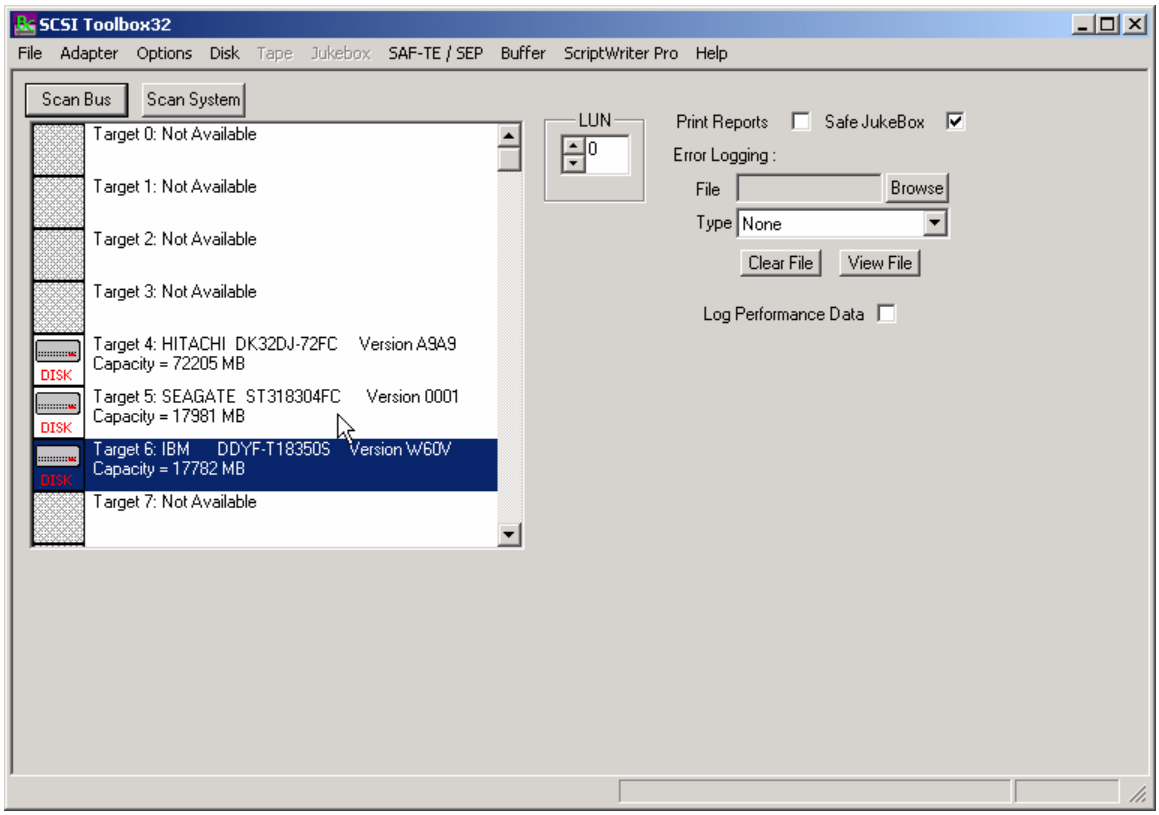
Upon completion of the installation process you will be prompted to reboot your computer. Install the hardware key (dongle) onto the printer port of your computer and reboot.

Windows 98 Users

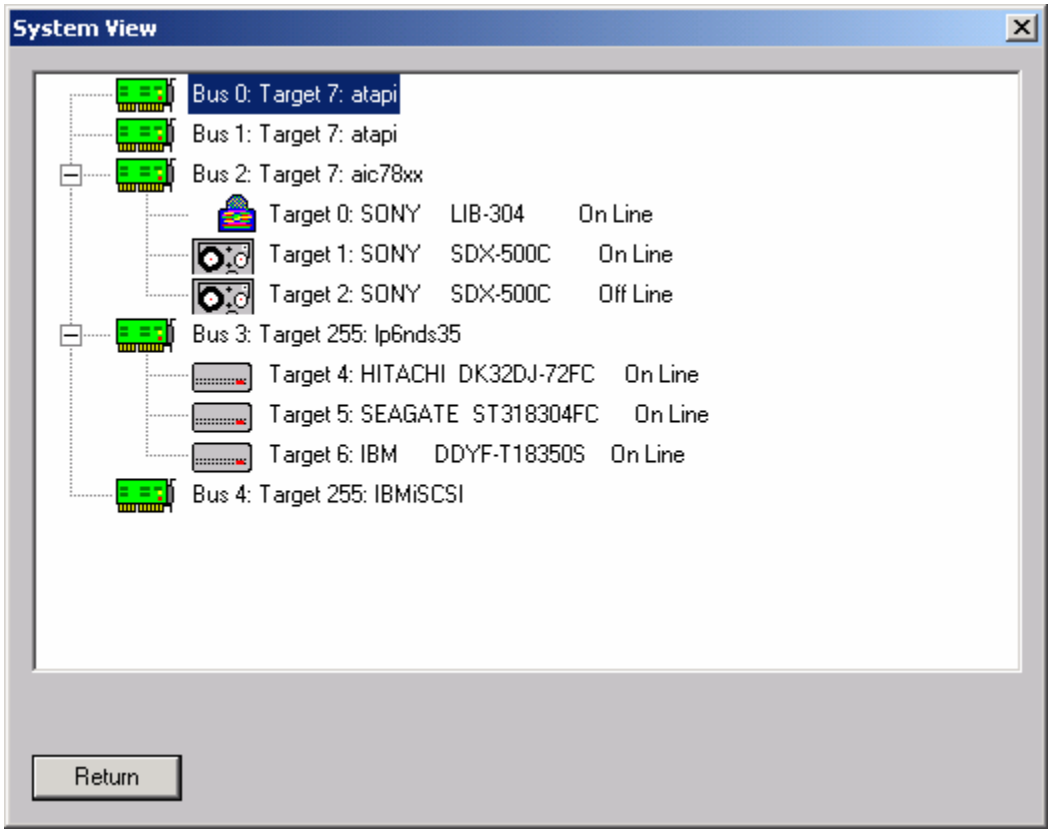
PLEASE NOTE!!!!

If you are installing the SCSItoolbox under Windows 98 or ME you will need to rename the file "PTI2GBASE-98.dll" to "PTI2GBASE.DLL". This file is located in the SCSItoolbox folder.

Main Menu

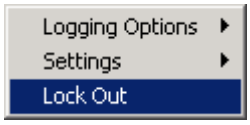


The System View shows all HBA' s, all targets, and all LUN's attached to the system:



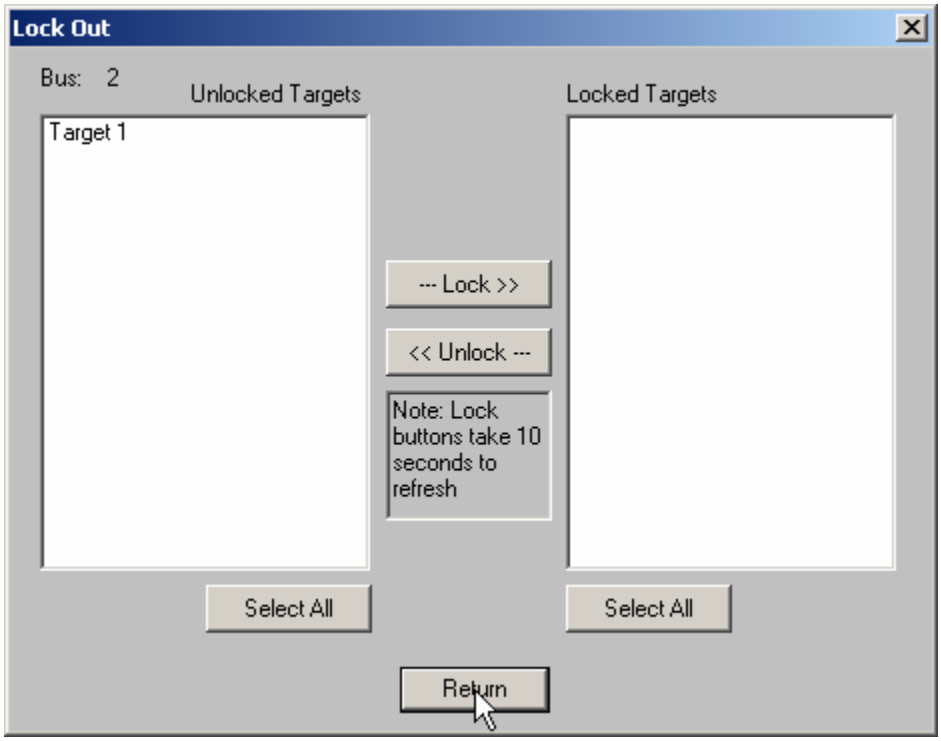
Options

The following options are available from the Options menu choice:

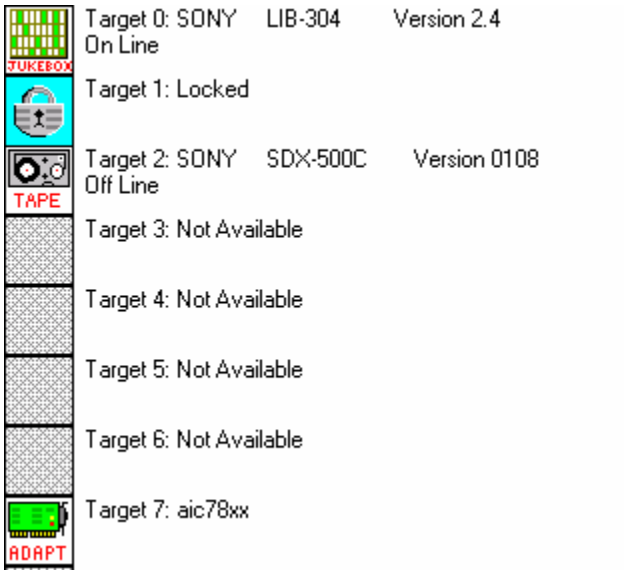


Lock Out (protecting a drive)

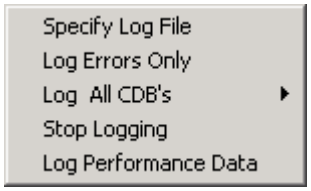
The Drive Locking option allows you to specify which drives can be accessed for testing or commands. Select the drive to lock by clicking on it's target number in the left (Unlocked) box. Then click the Lock button to move this drive to the right (Locked) box.



When a drive is locked it will appear like this:



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.

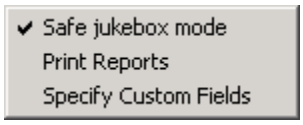
Logging options are:

- *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 and 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

Todays Date :%d

Manufacturer :%v

Product :%p

Firmware :%r

SCSI Target :%a

Test results for %v %p

Drive PASSED %n

Tested by (Tech) %d

%f

Specifiy 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 denined field 1 or 2.

Disk Tests

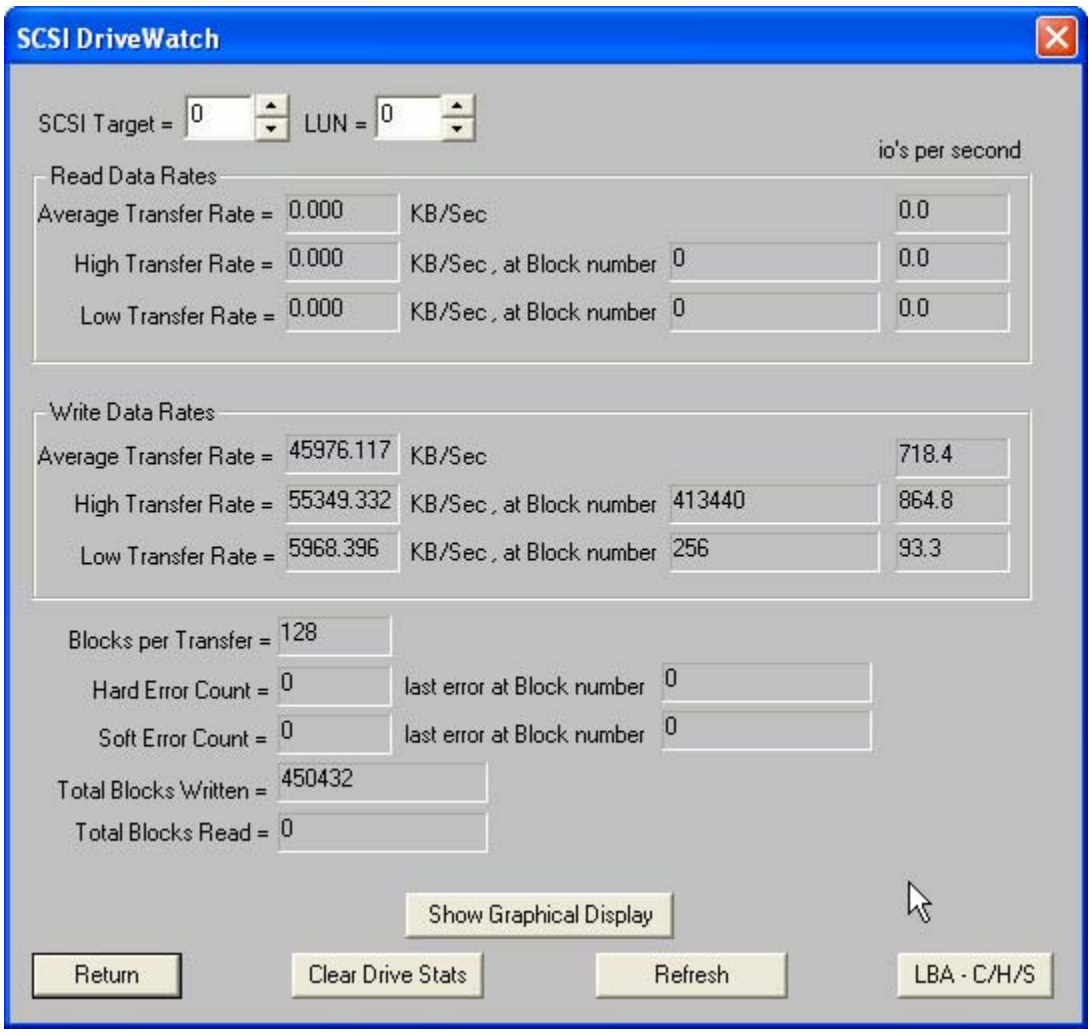
Disk Test Options

DriveWatch

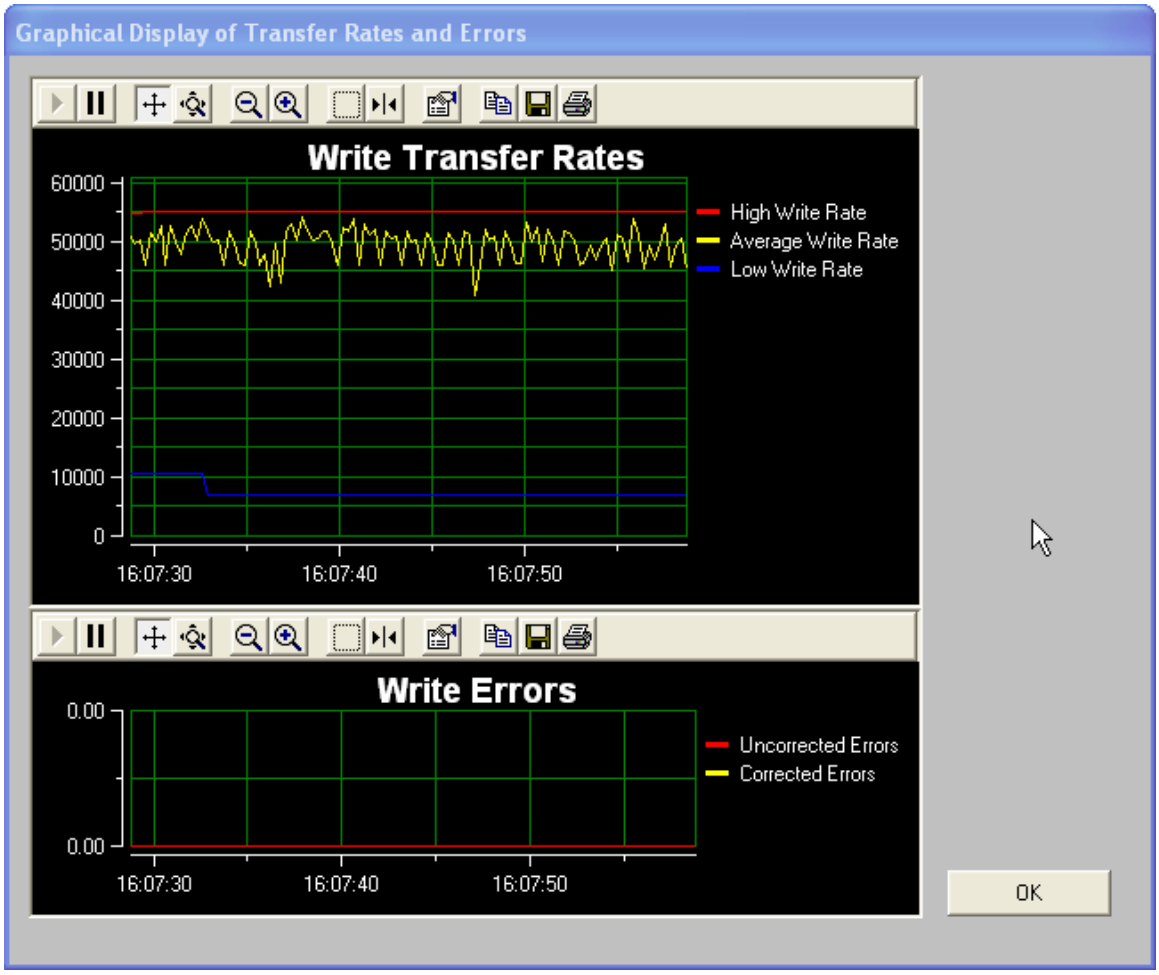
DriveWatch is a real-time performance display that is accessible from any of the SCSItoolbox32 tests.

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



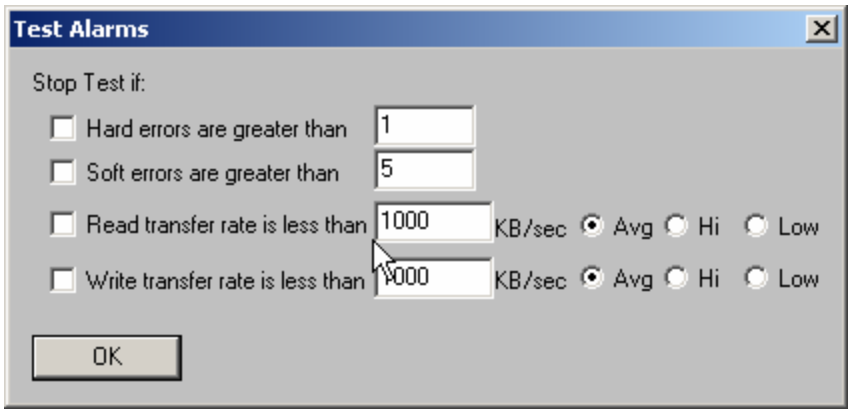
The New Graphing display within drivewatch:



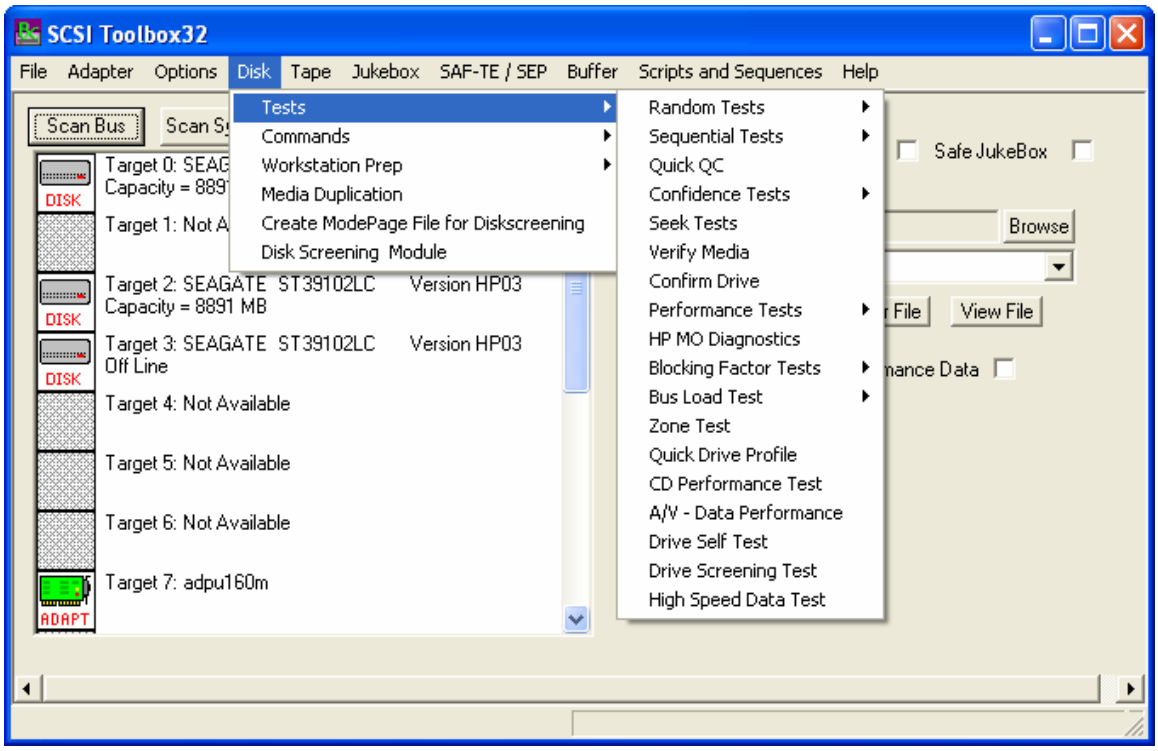
Test Alarms

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

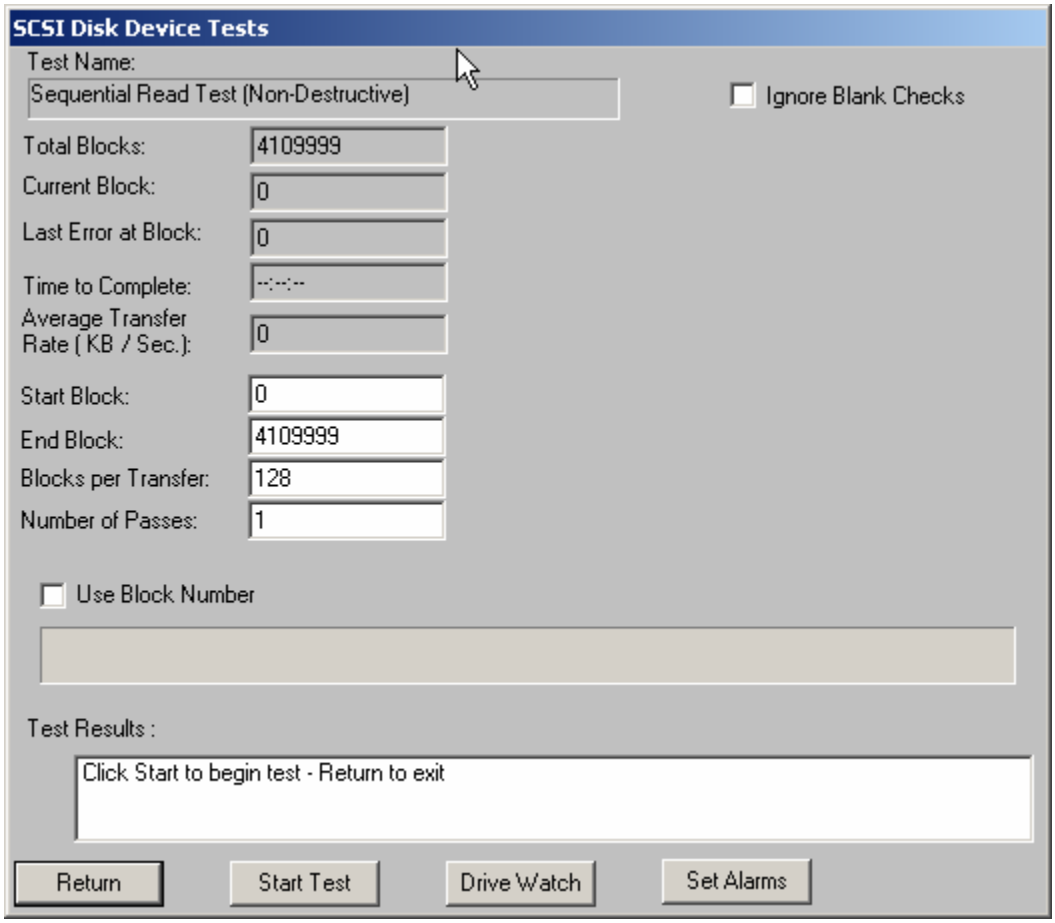


The following tests may be accessed from the Test Menu SCSItoolbox32 (details on each test follow):



Sequential Read

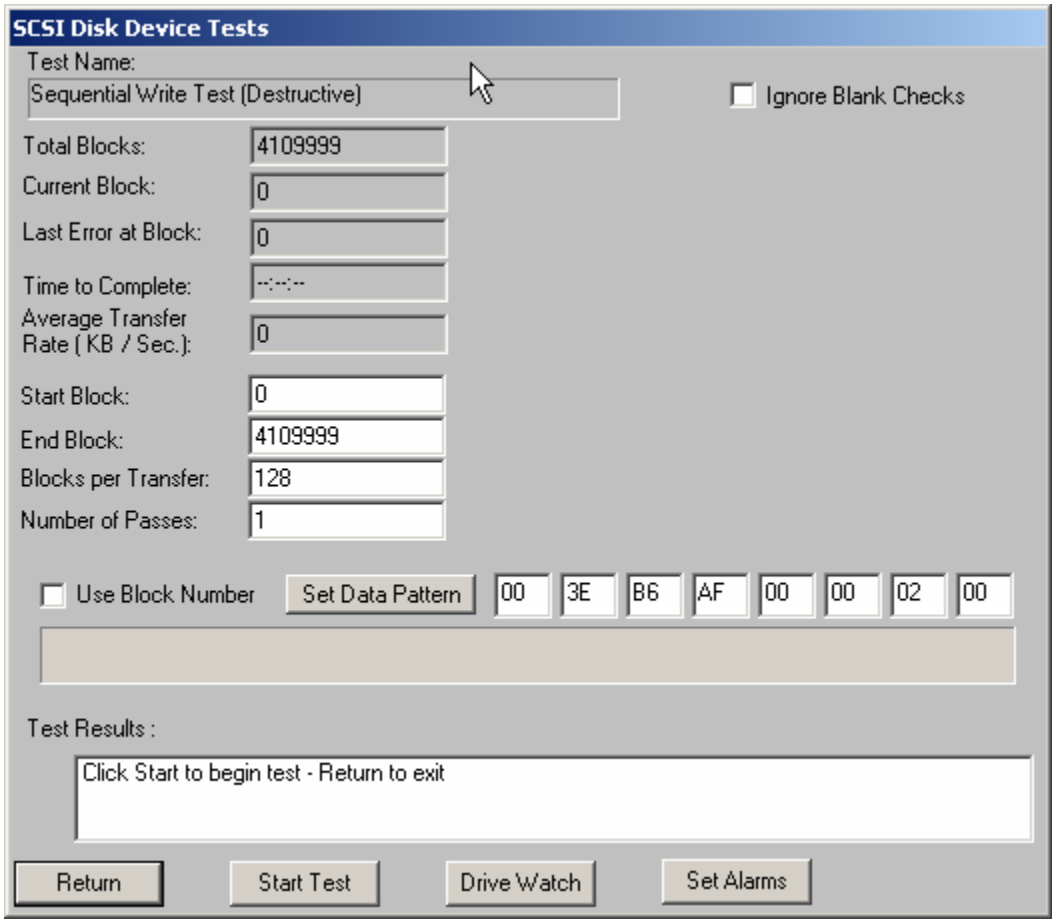
This test will sequentially read all blocks between the Start Block and the End block. If a read fails the test will stop and the details of the error will be reported.



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

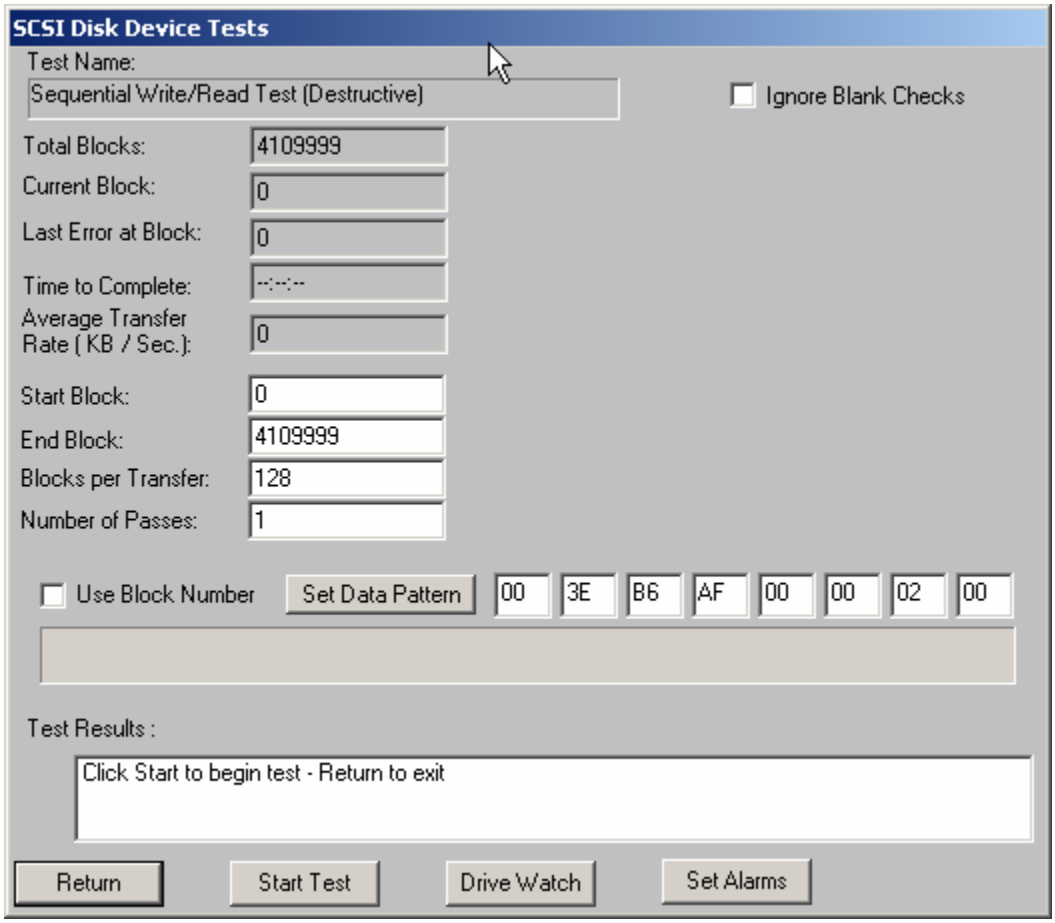
This test will sequentially write all blocks between the Start Block and the End block. If a write fails the test will stop and the details of the error will be reported.



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

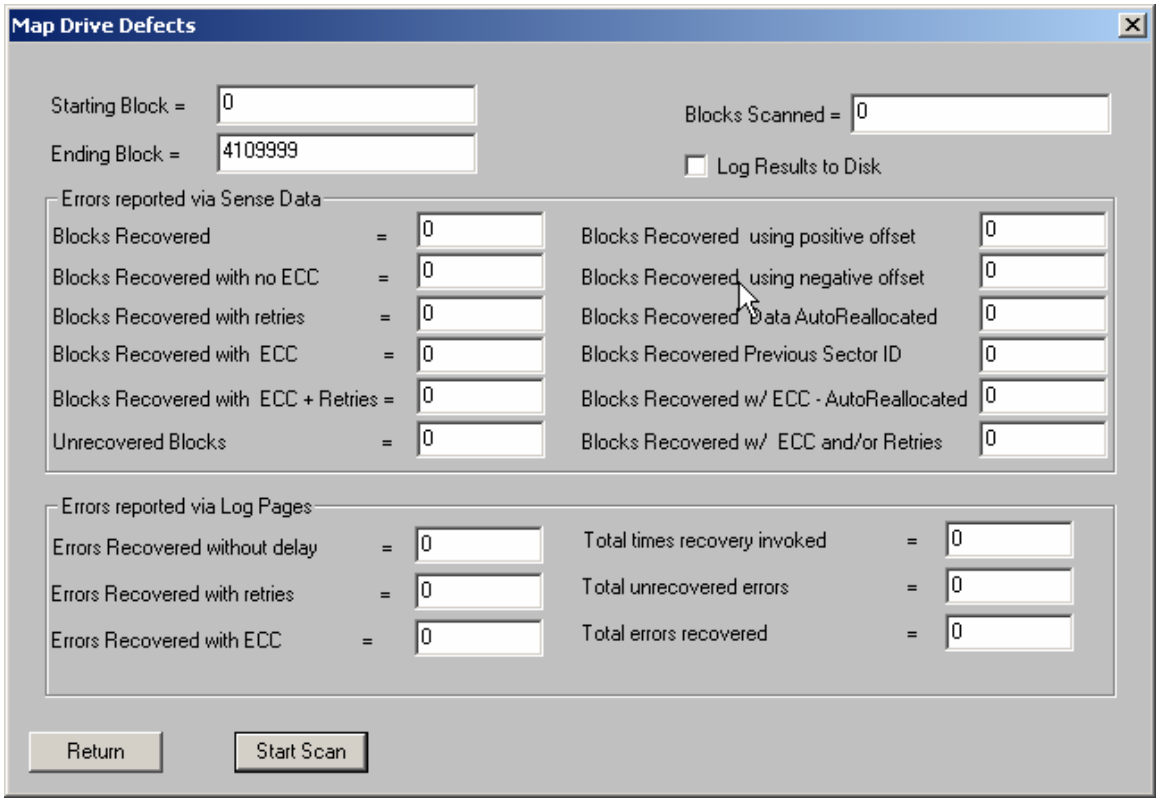
This test will sequentially write, then read all blocks between the Start Block and the End block. If a write or read fails the test will stop and the details of the error will be reported.



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

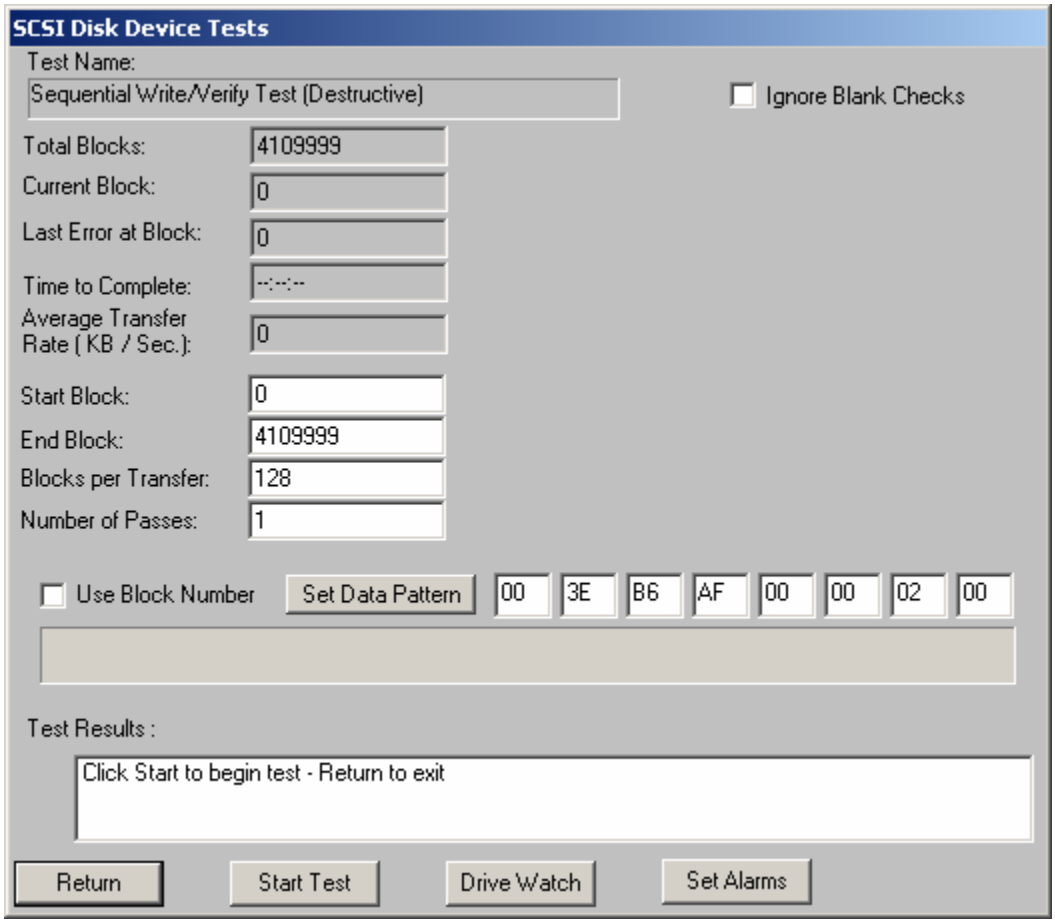
This test sequentially reads all blocks between Start Block and End Block. The error recovery information reported by the drive via Sense Data and Log Pages is displayed and may be recorded to a log file using the Log Results to Disk option.



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

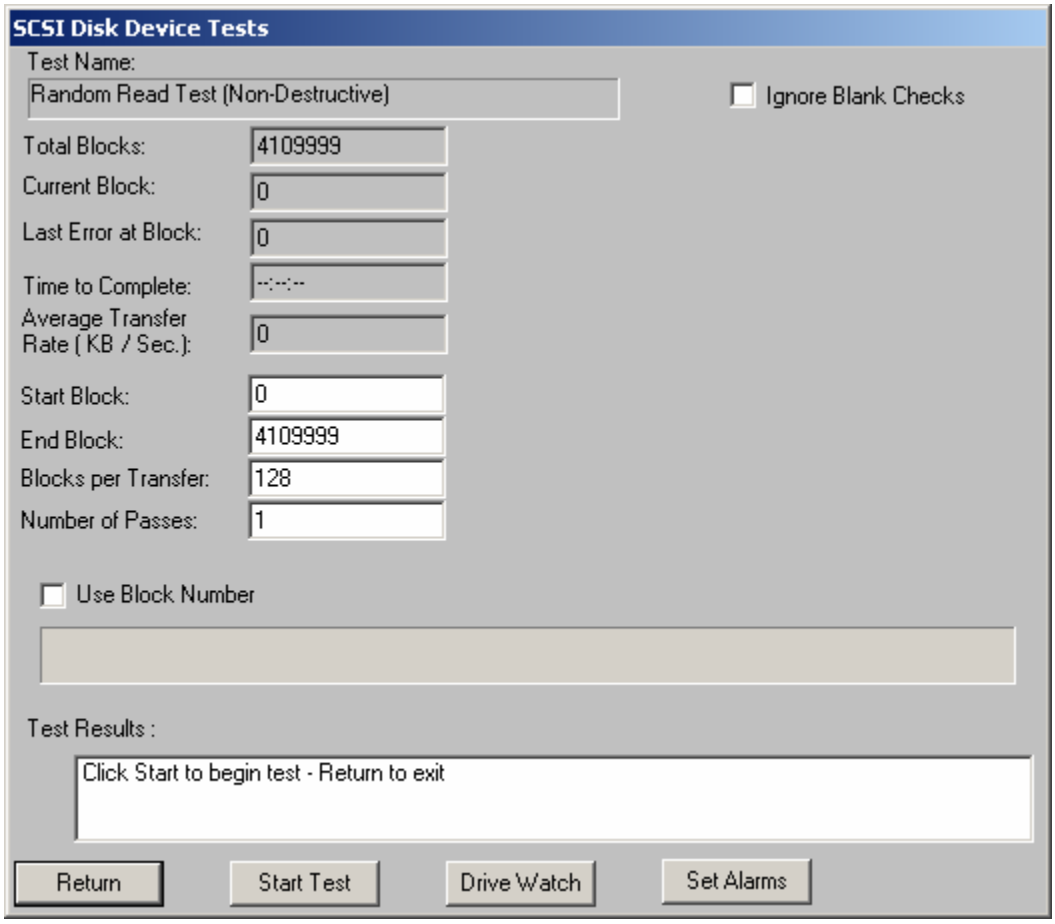
This test will sequentially write and verify all blocks between the Start Block and the End block. If a write or verify fails the test will stop and the details of the error will be reported.



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

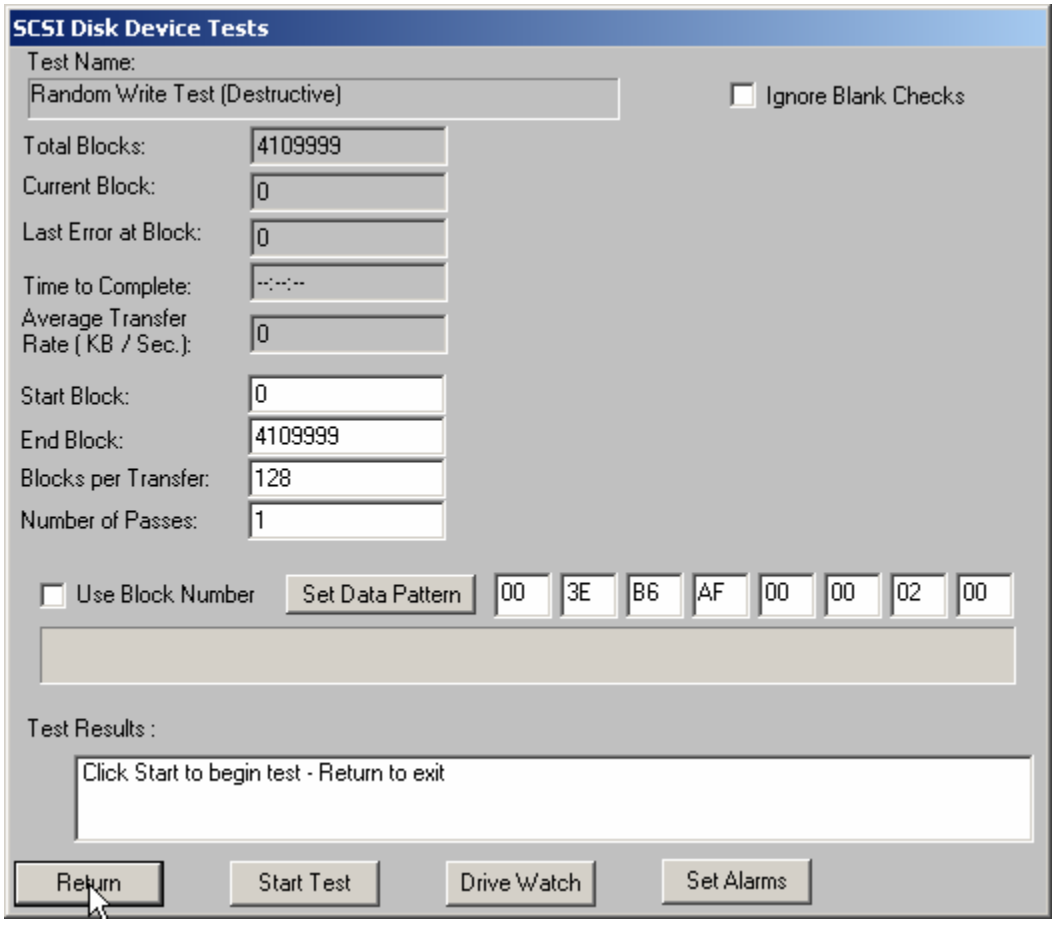
This test will read random blocks between the Start Block and the End block. If a read fails the test will stop and the details of the error will be reported. This test will run until stopped by pressing the Cancel button.



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

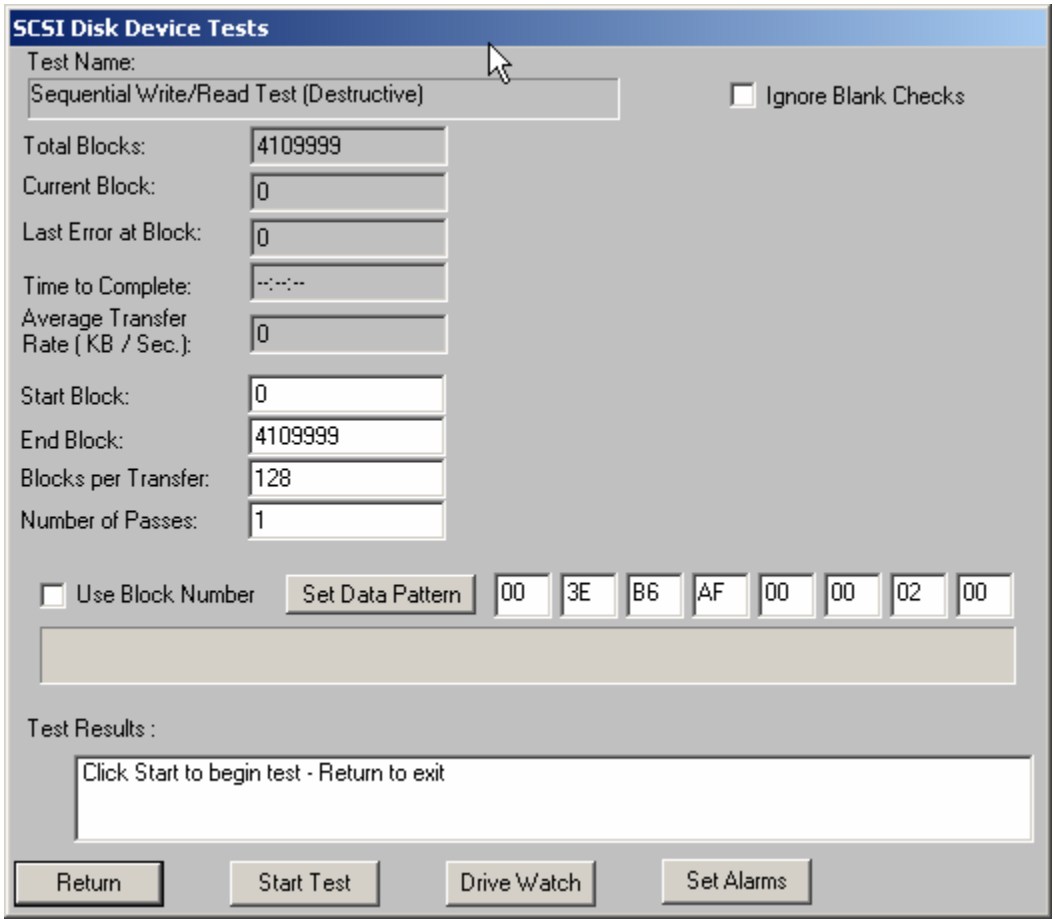
This test will write random blocks between the Start Block and the End block. If a write fails the test will stop and the details of the error will be reported. This test will run until stopped by pressing the Cancel button.



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

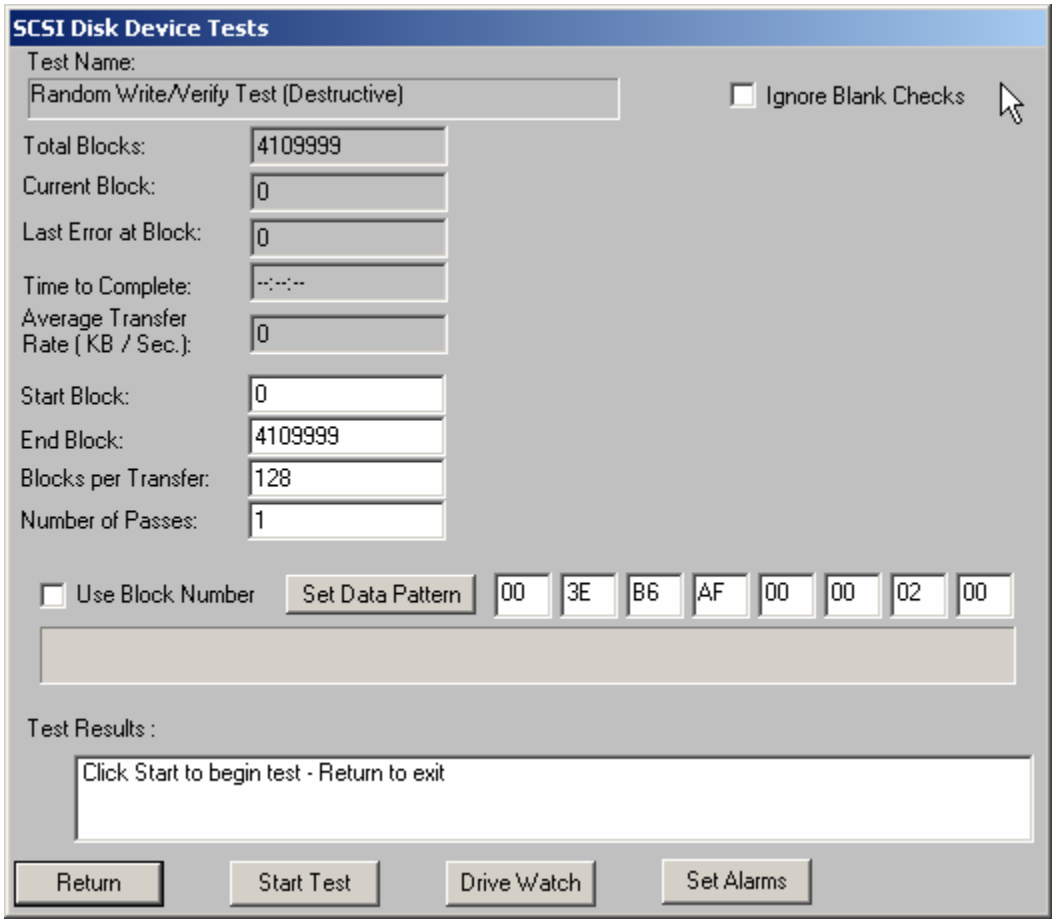
This test will write and read random blocks between the Start Block and the End block. If a write or read fails the test will stop and the details of the error will be reported. This test will run until stopped by pressing the Cancel button.



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

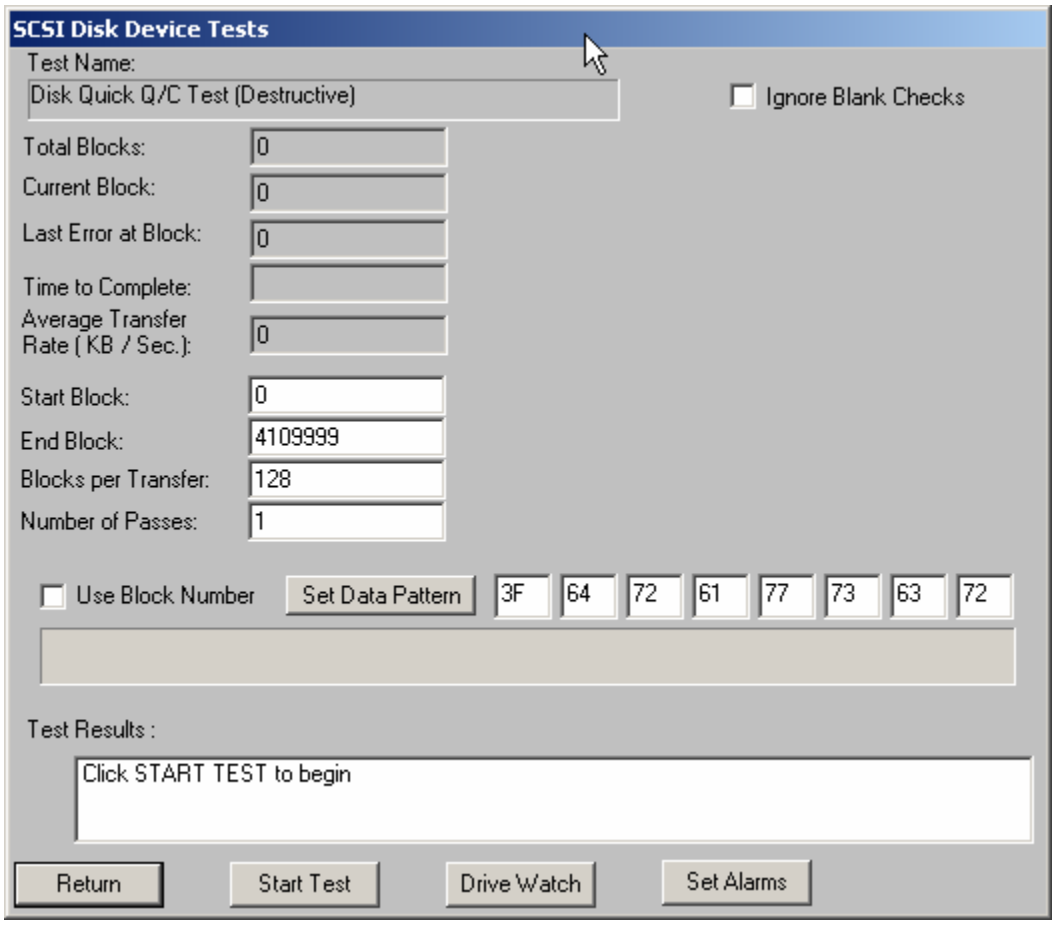
This test will write and verify random blocks between the Start Block and the End block. If a write or verify fails the test will stop and the details of the error will be reported. This test will run until stopped by pressing the Cancel button.



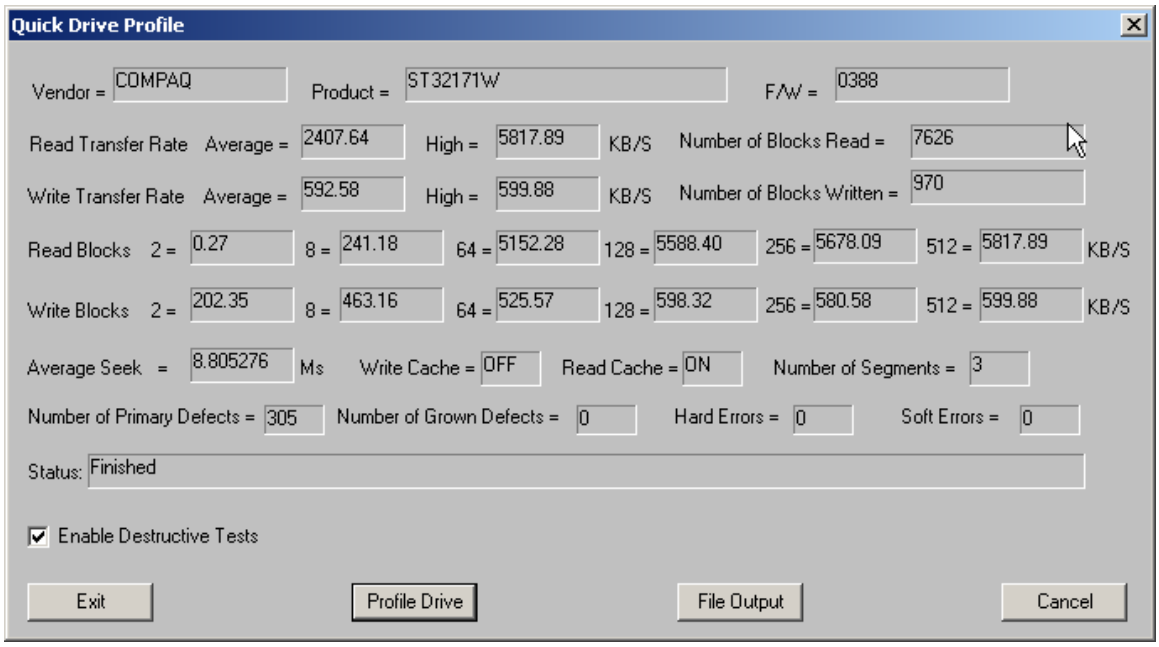
User Test Parameters:	
Parameter	
Start Block	Enter the LBA to define the low range of the test
	Default = 0
End Block	Enter the LBA to define the high range of the test
	Default = highest LBA as reported by the READ CAPACITY command.
Blocks per Transfer	Enter number of blocks per CDB
	Default = 128 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.



Quick Drive Profile



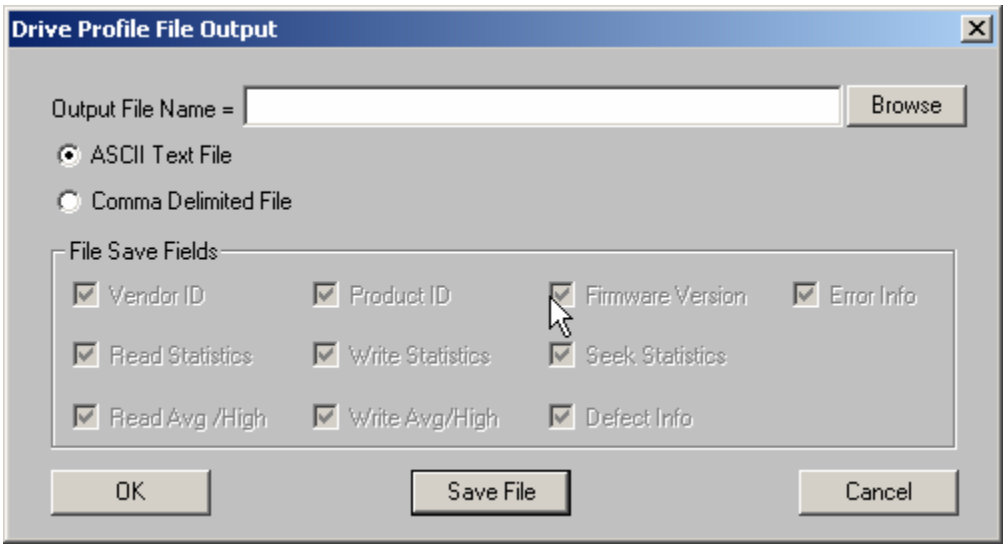
User Test Parameters:	
Parameter	
Enable Destructive Test	Checking this option will allow writes to the drive during this test. If not check only read operations will be executed.

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).

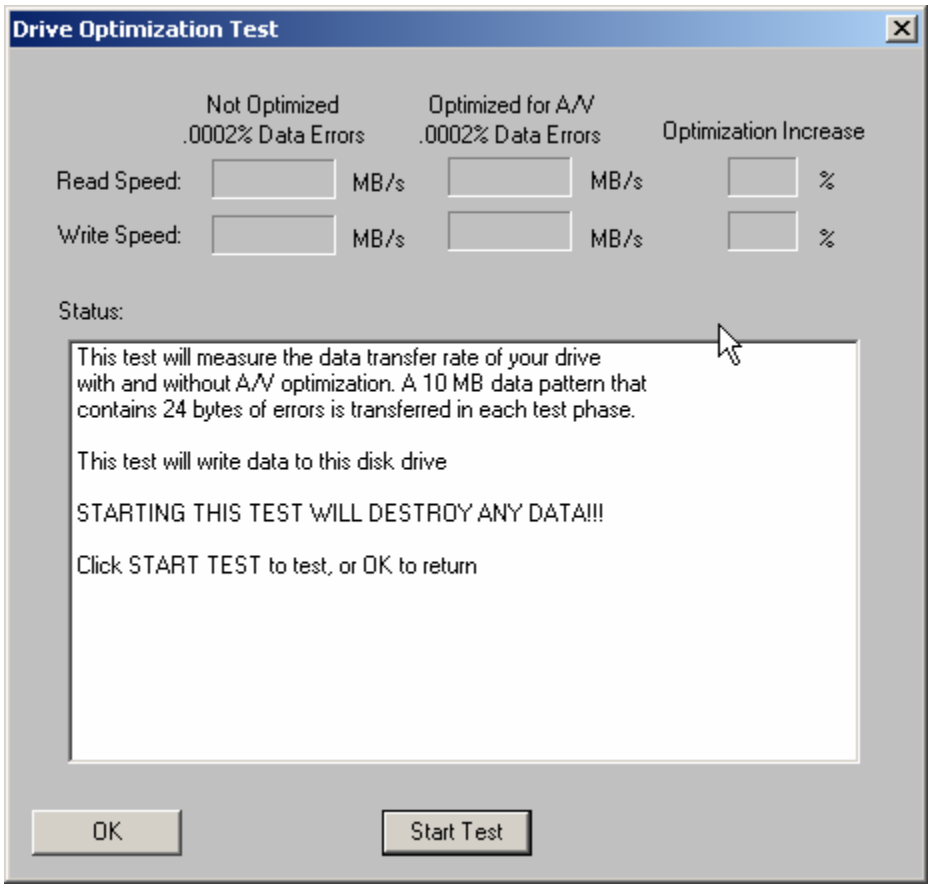
Drive Profile File Output

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.



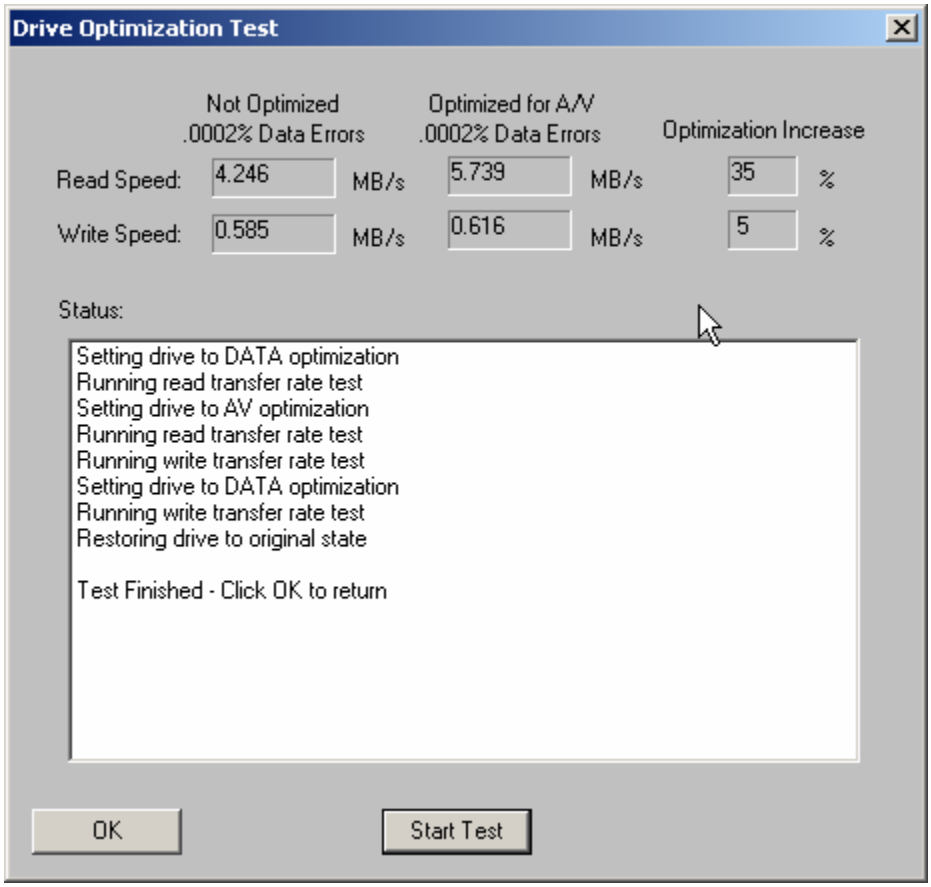
AV/Data Performance Comparison



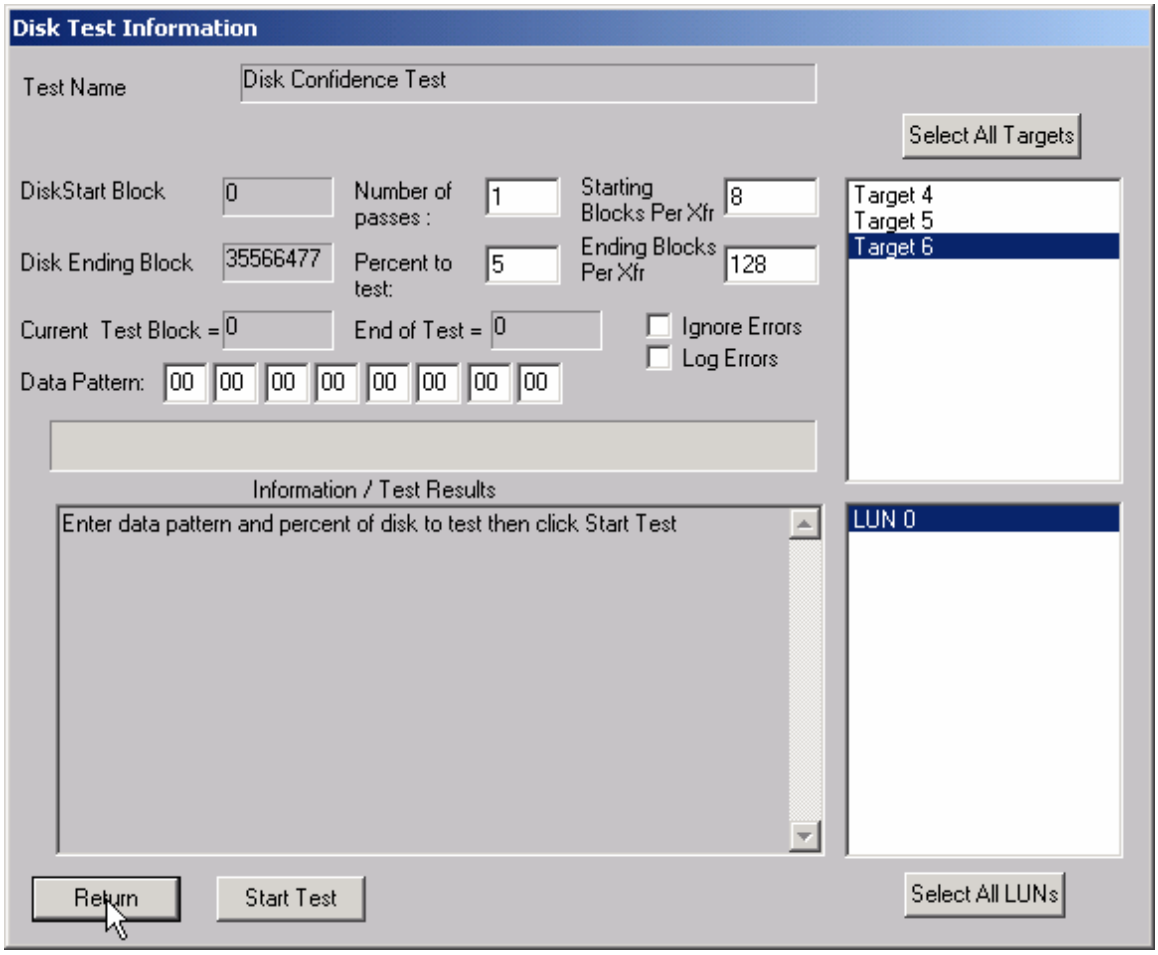
User Test Parameters:	
Button	**Functionality**
Start Test	Introduces correctable (ECC) errors onto the drive. Then the test will configure the drive for A/V optimization, then execute a write and a read test. Then the drive will be configured for Data use and the tests run again. Then the correctable 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.



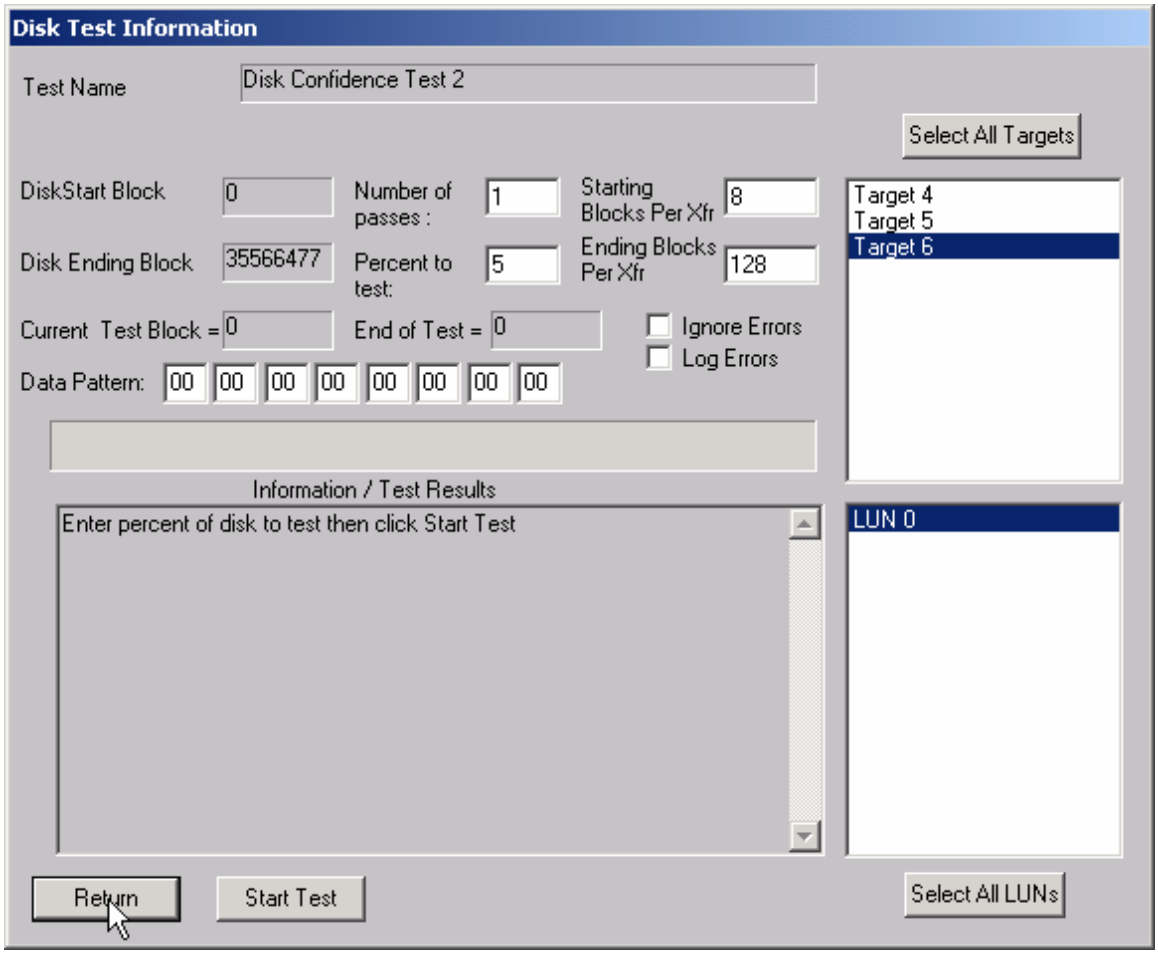
Drive Confidence Test 1



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.

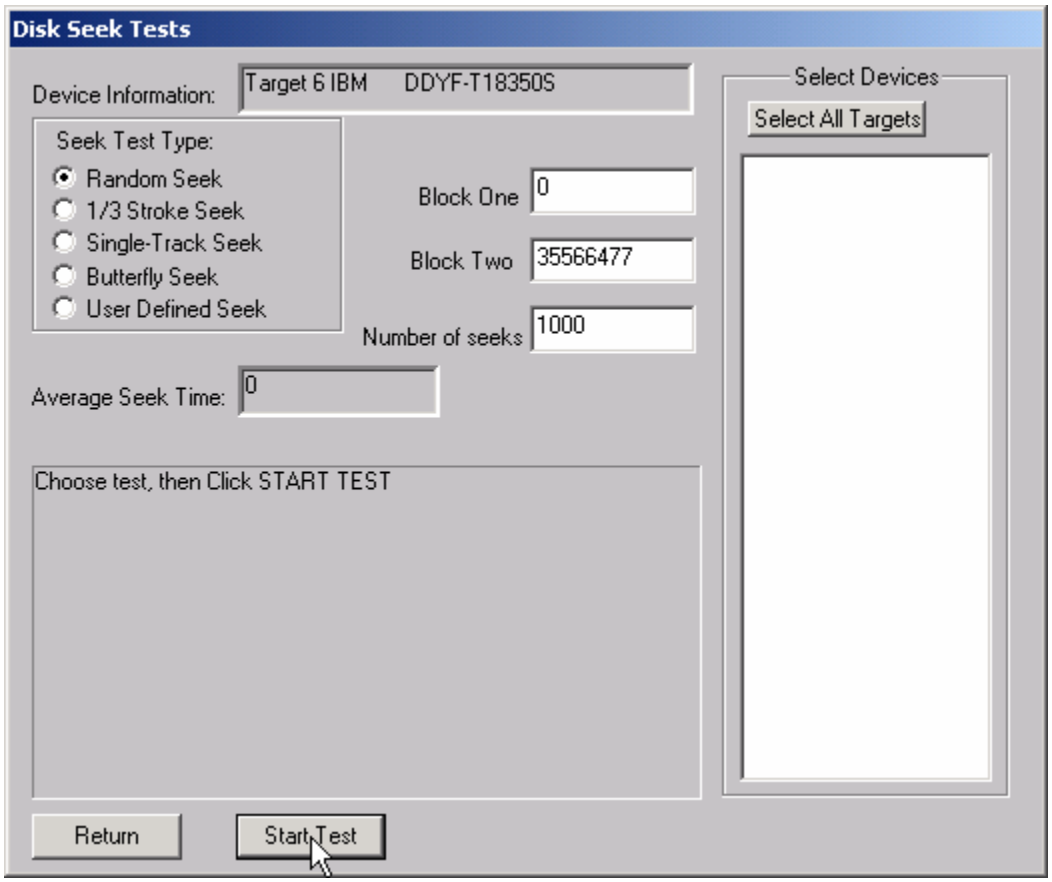
This test checks drive functionality by writing data patterns to the disk, executing random seeks, and starting/stopping the drive. The number of test passes, the data pattern, and the percentage of the disk to be tested may be specified. The specified data pattern will be written to the percentage of 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



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

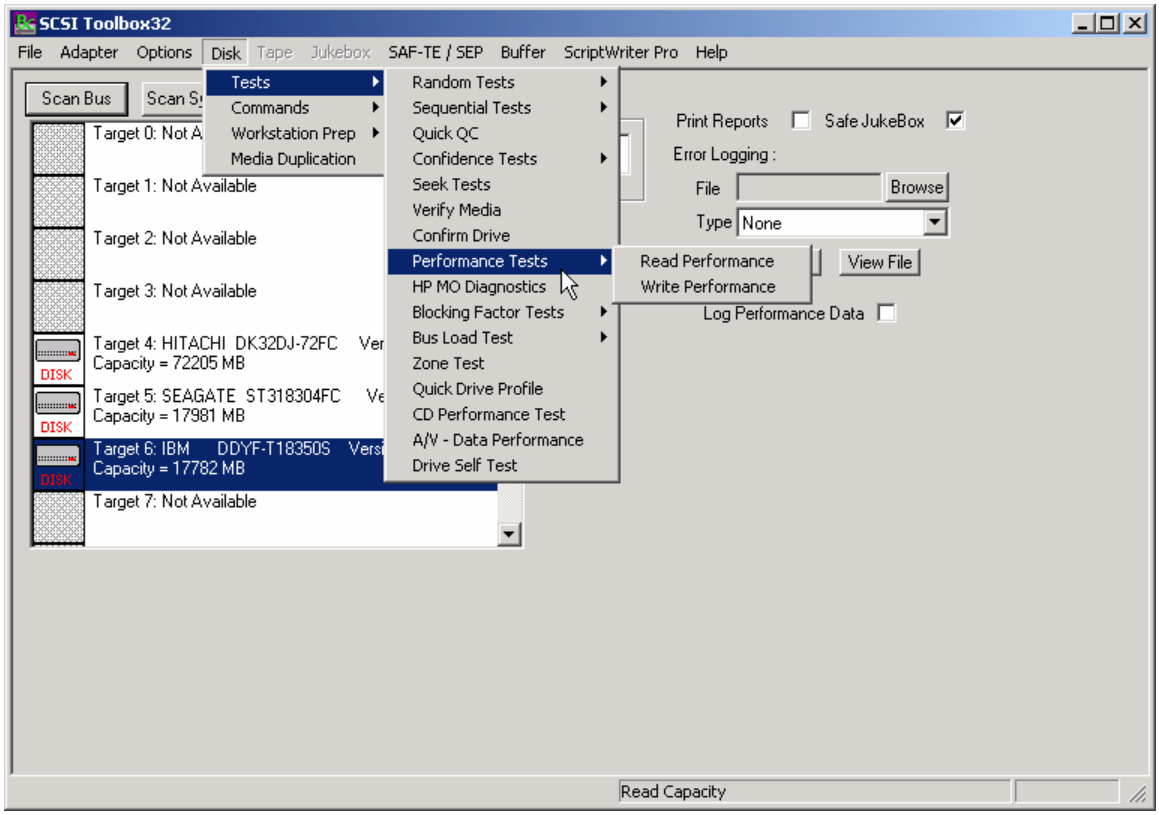


This test menu gives you access to several different seek test algorithms. Each seek test will first calculate the system overhead time for processing SCSI commands. Then it will perform 10,000 seeks of the type specified and calculate the total and average seek times. You may select one of the following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

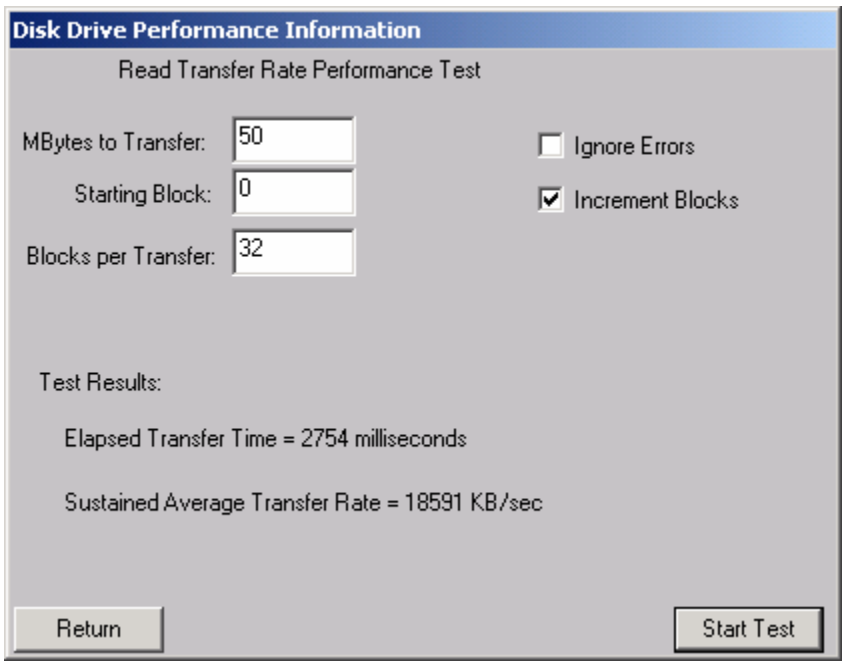
This test reads a specified amount of data (default: 10 MB) from the selected device with or without incrementing the block number and reports the sustained transfer rate during the transfer.

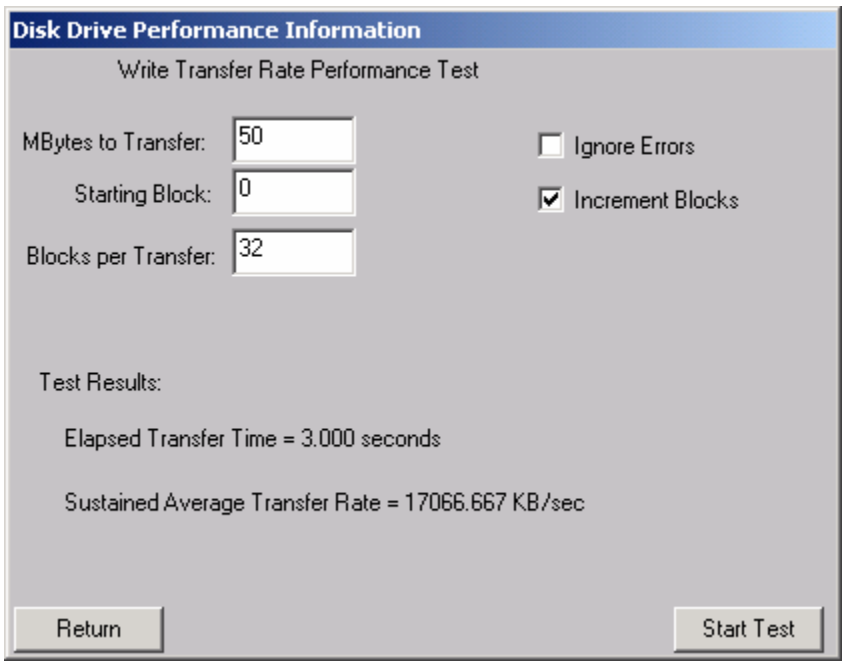
Reading 10 Mbytes from Target 2 Host Adapter 0
(156 Reads of 128 Blocks using 512 byte blocks)
Elapsed Transfer Time = 31 seconds
Sustained Average Transfer Rate = 1651 KB/sec

The Write Transfer Rate Test

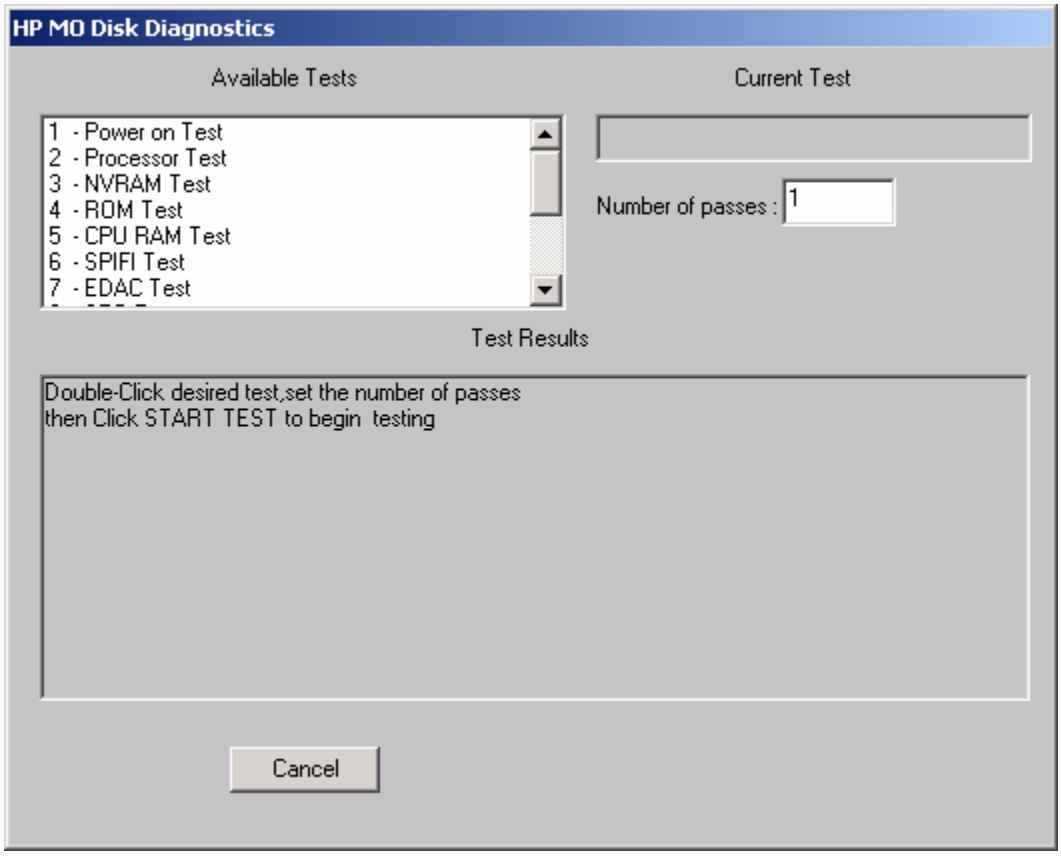
This test writes a specified amount of data (default: 10 MB) from the selected device with or without incrementing the block number and reports the sustained transfer rate during the transfer.

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



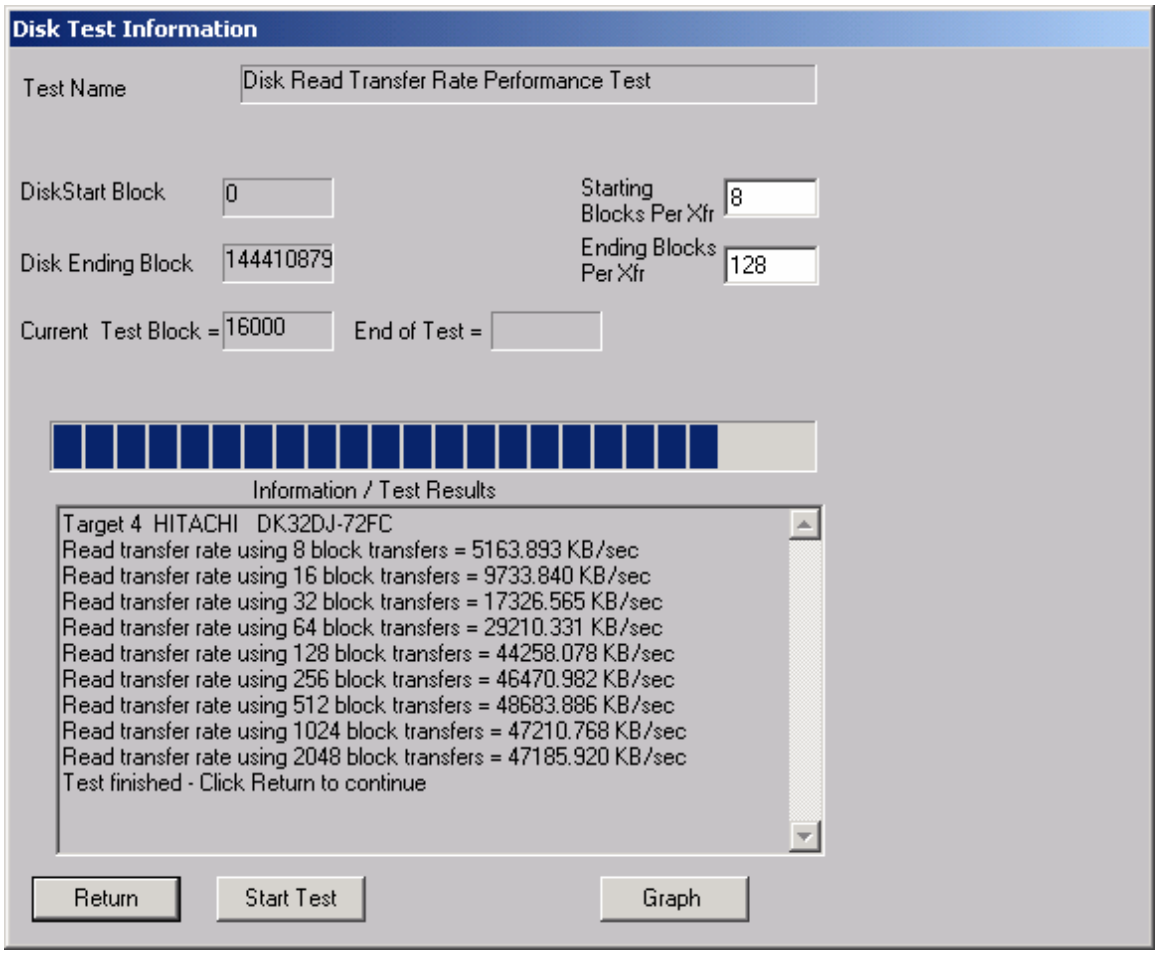


HP MO (Magneto Optical) Tests



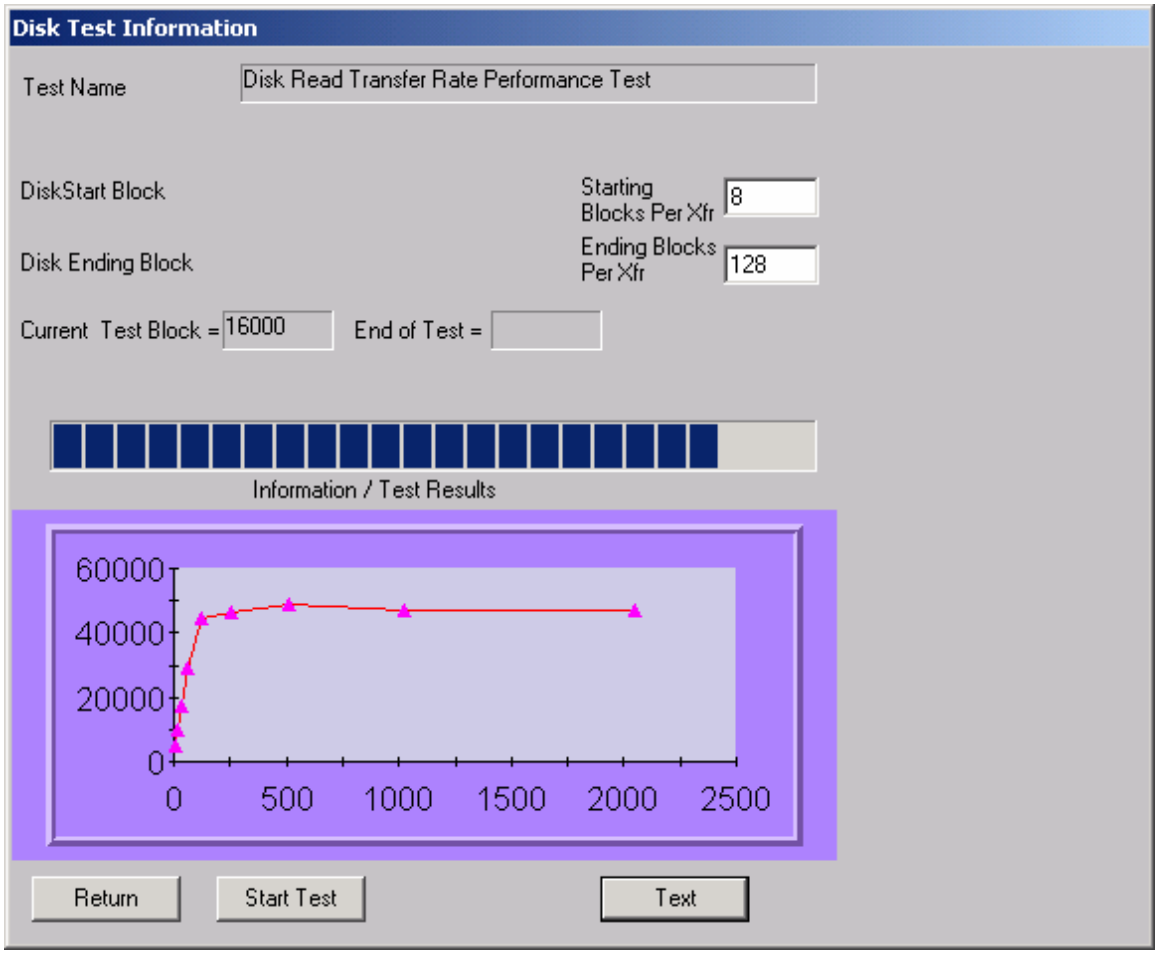
This selection is similar to the tests and utilities in the dosdass2 program.

Blocking Factor Tests

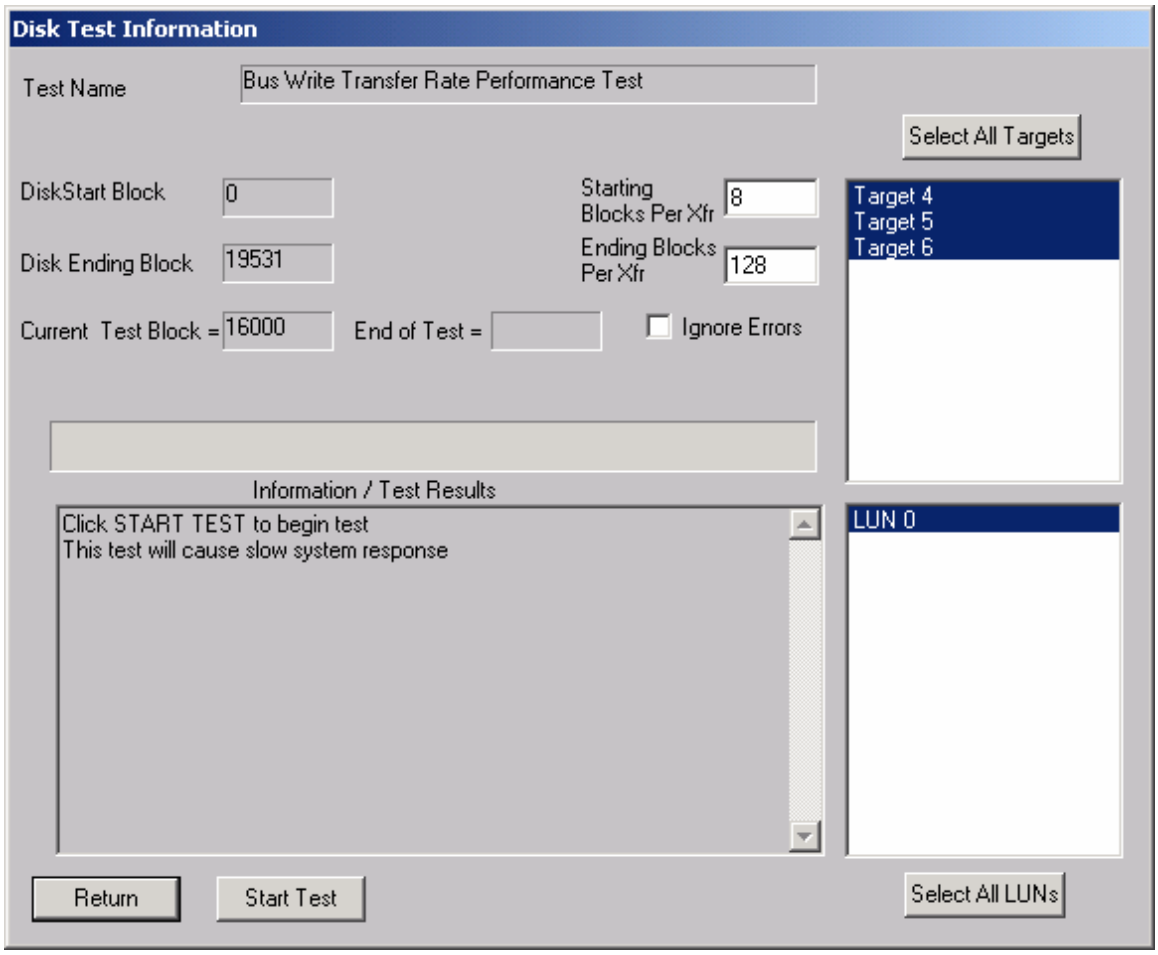


This test performs a series of 10MB reads with varying blocking factors to profile the I/O rates of the drive for various transfer sizes. The test starts with a transfer size of 1 block and increments that transfer rate through: 2, 4, 8,16,32,64,128,256, and 512 block transfers and times the 10MB I/O.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

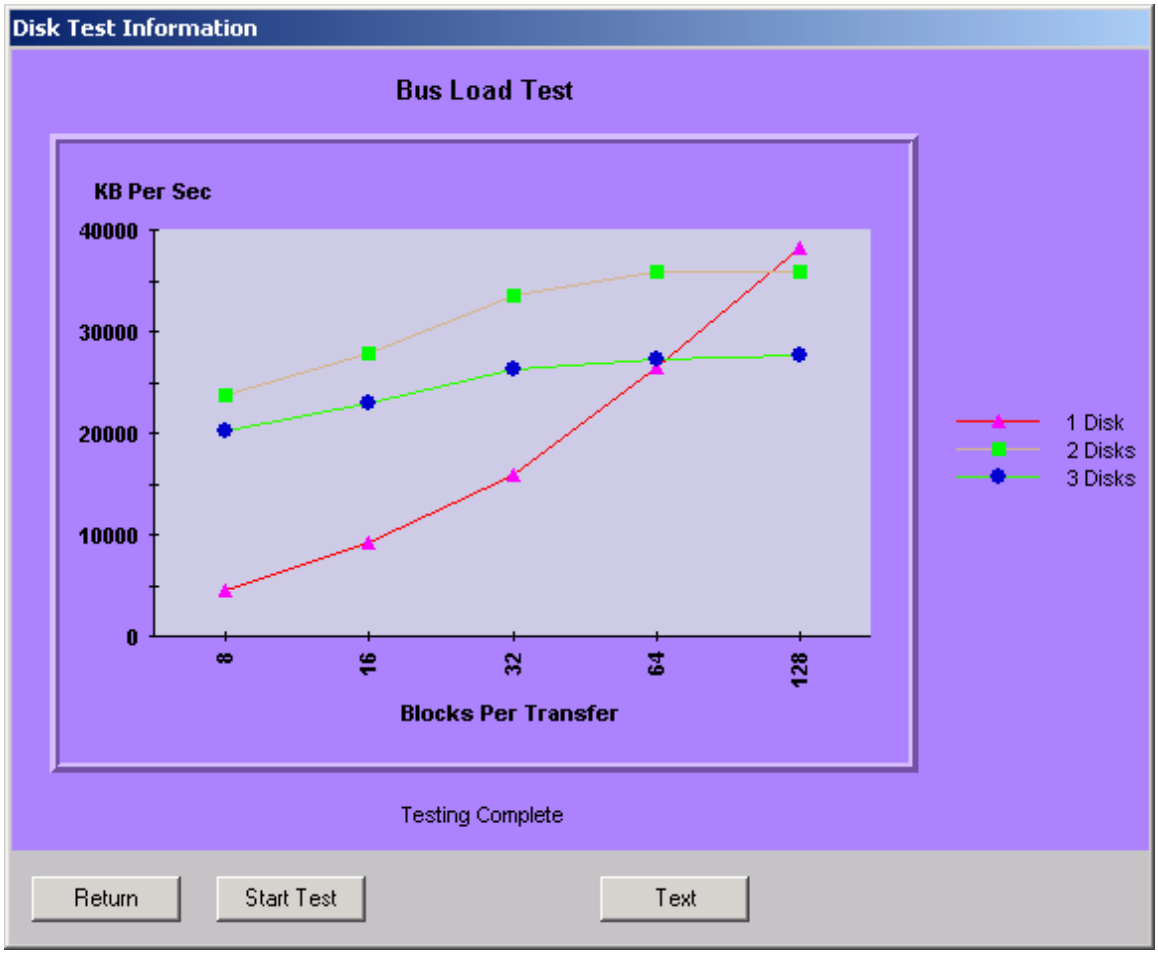
Write Performance
This test is the same as the Read Performance except that write commands are issued.

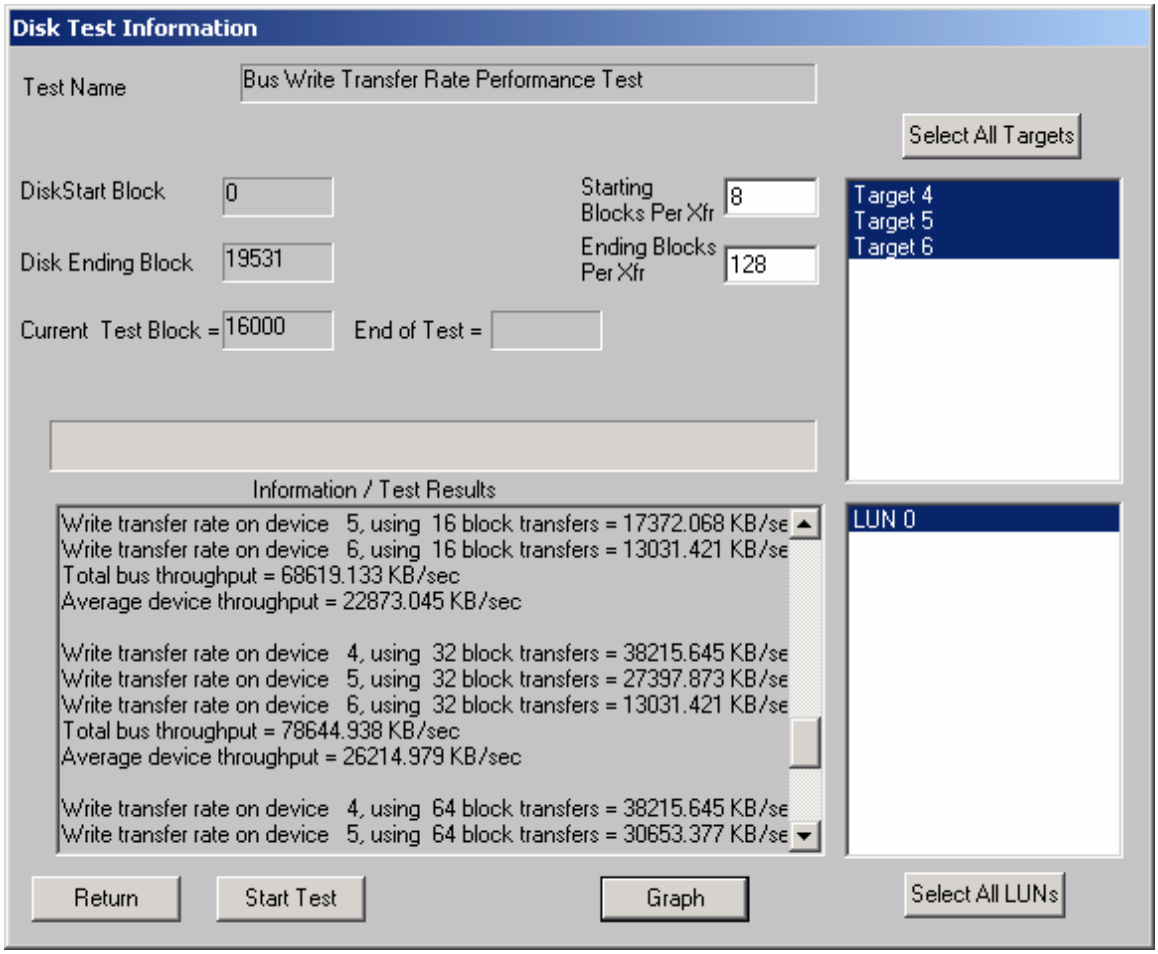


Bus Load Test

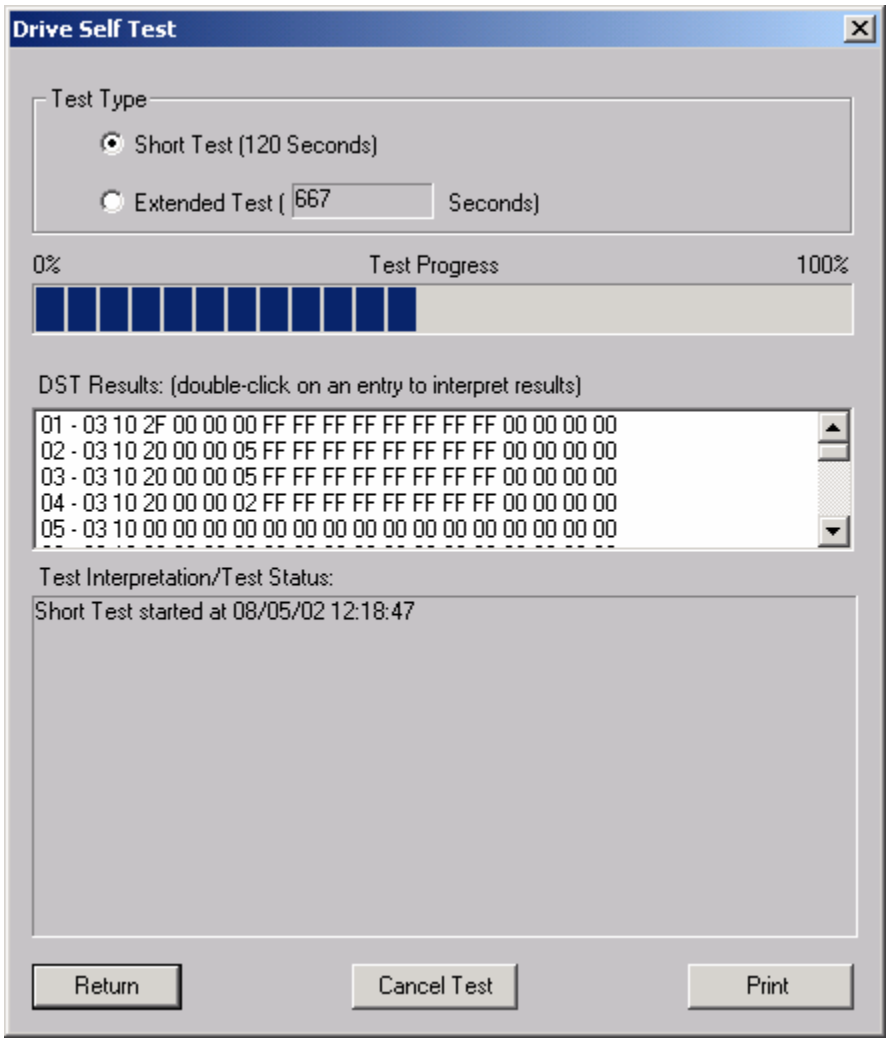


The bus load test will read or write data to one or more disk drives, graphing the transfer rates. With multiple disk drives, the test is performed on a single drive first, then on two drives, and so on, until all drives selected have been run. The average transfer rate to individual drives are graphed 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.

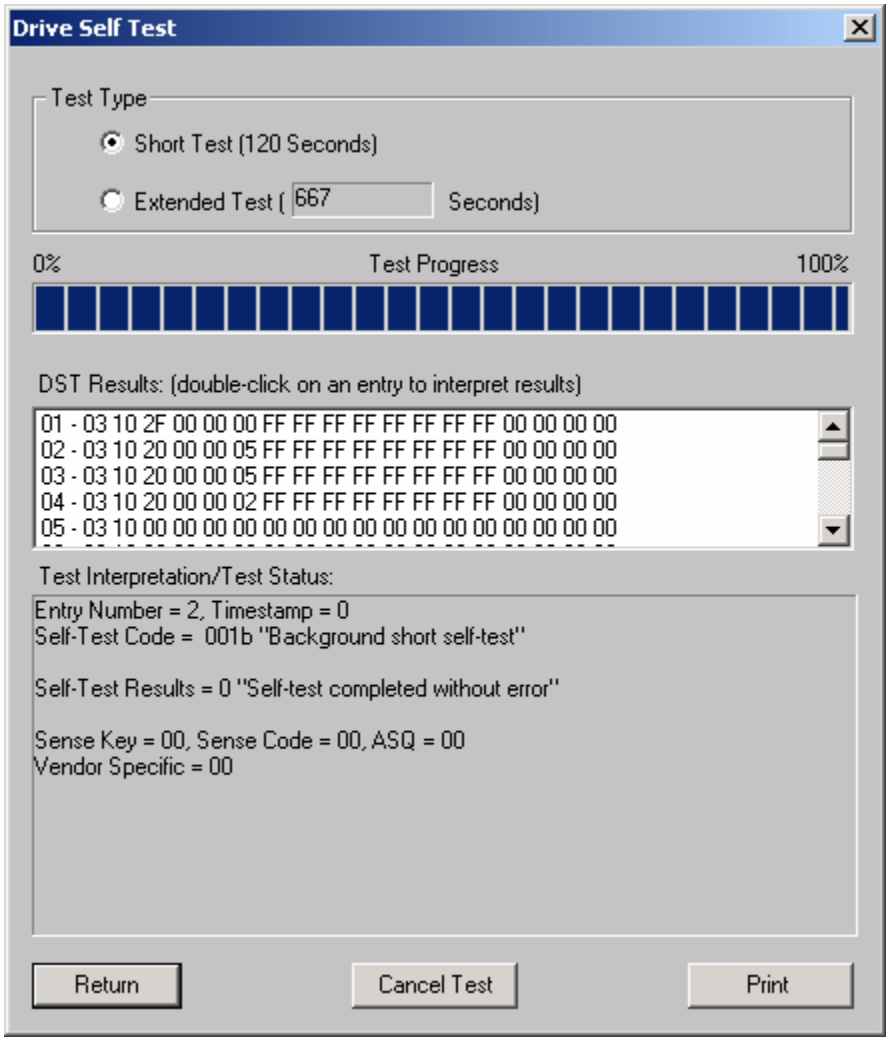




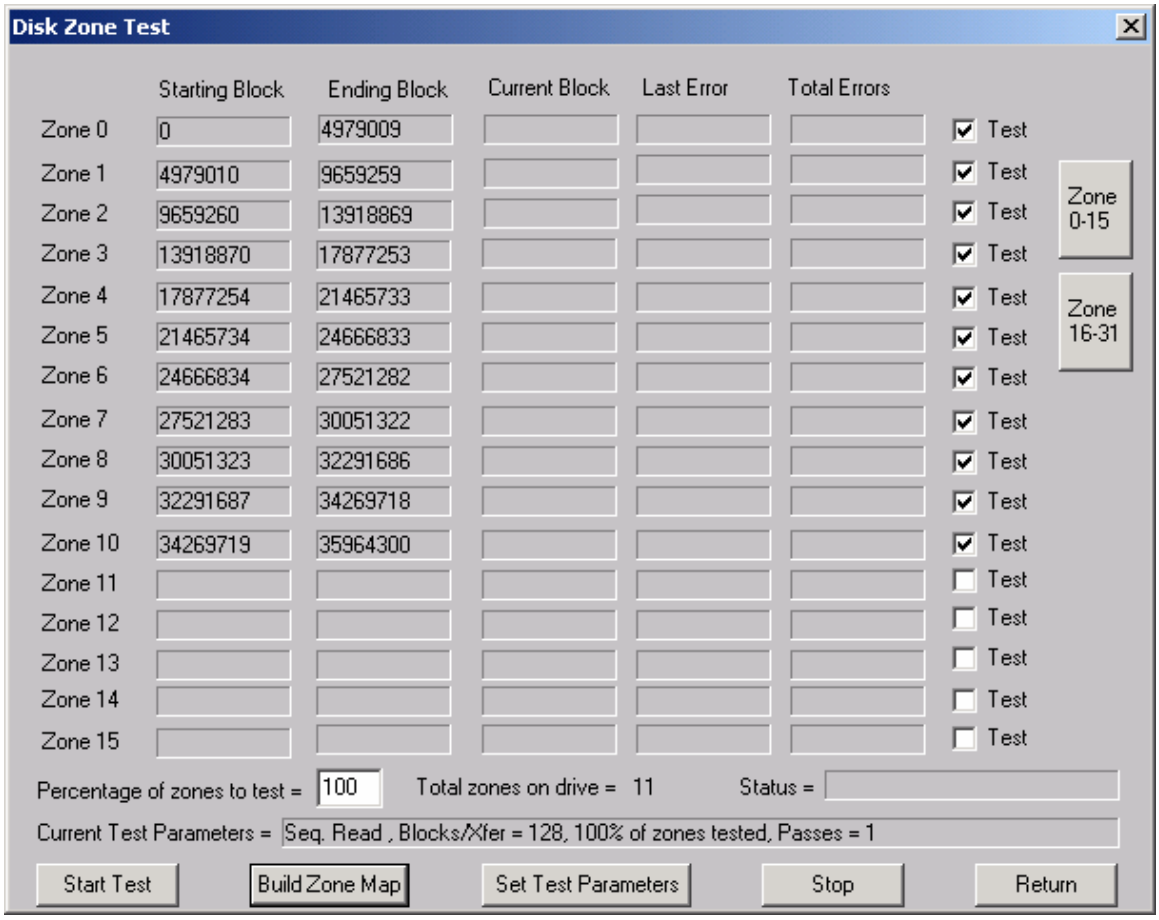
Drive Self Test



The Drive Self Test (DST) runs the drives built-in diagnostic tests in background mode. Once a self test is complete you can doulble-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.

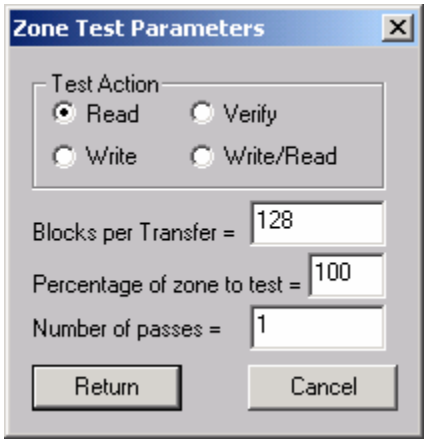


Disk Zone Test



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 bondaries 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.



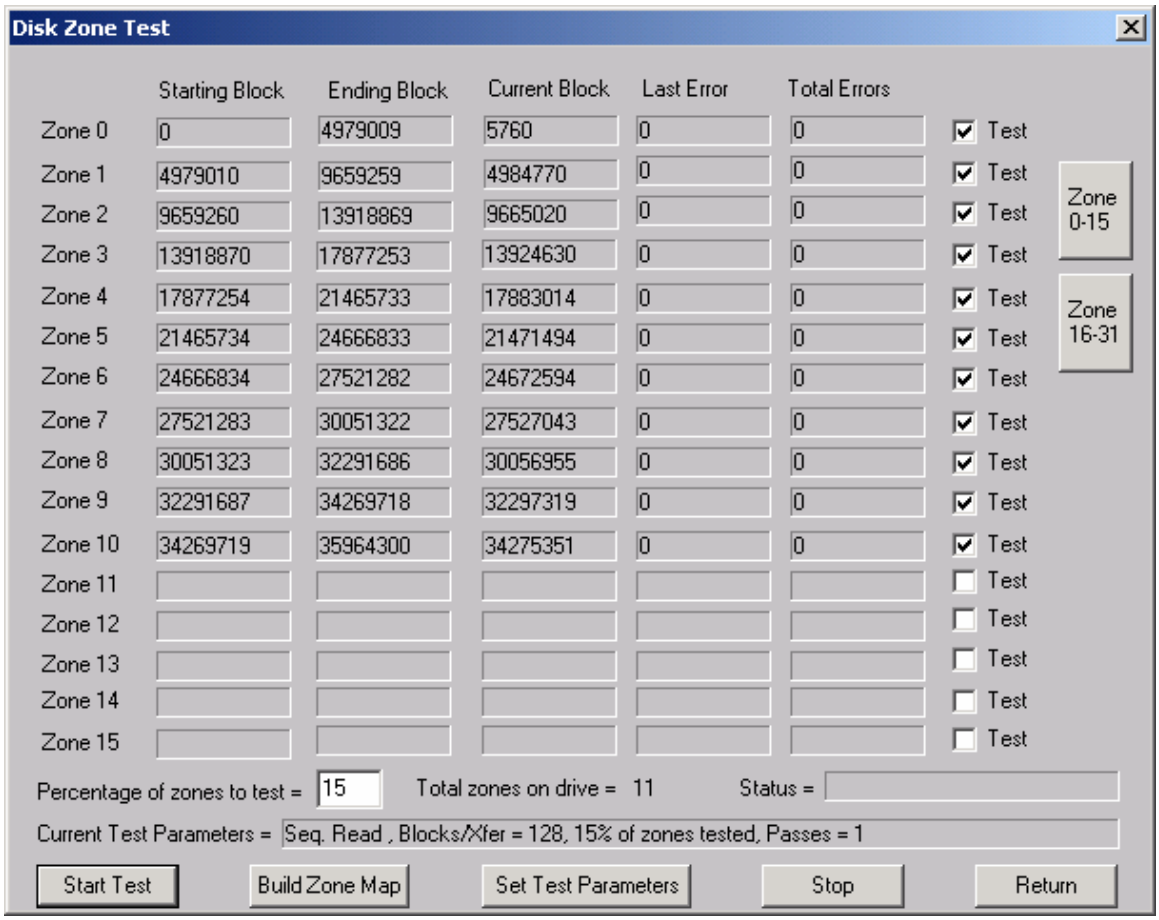
Set Zone Test Parameters – Clicking this button brings up the above dialog. You can select the following choices :

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

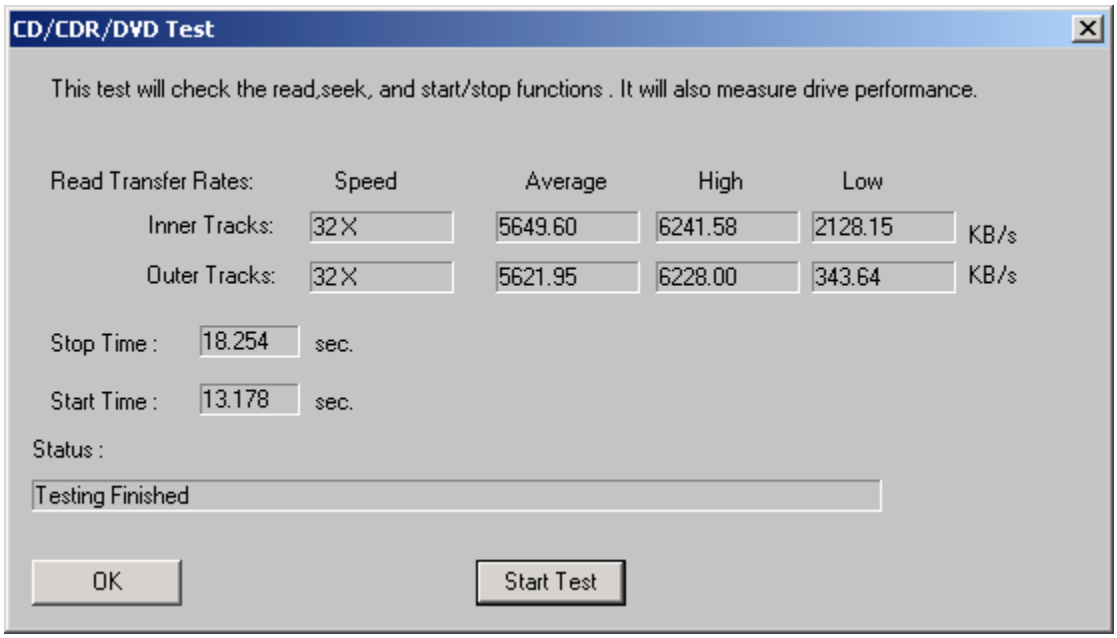
Blocks per Transfer – how many blocks are transferred for each SCSI command

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

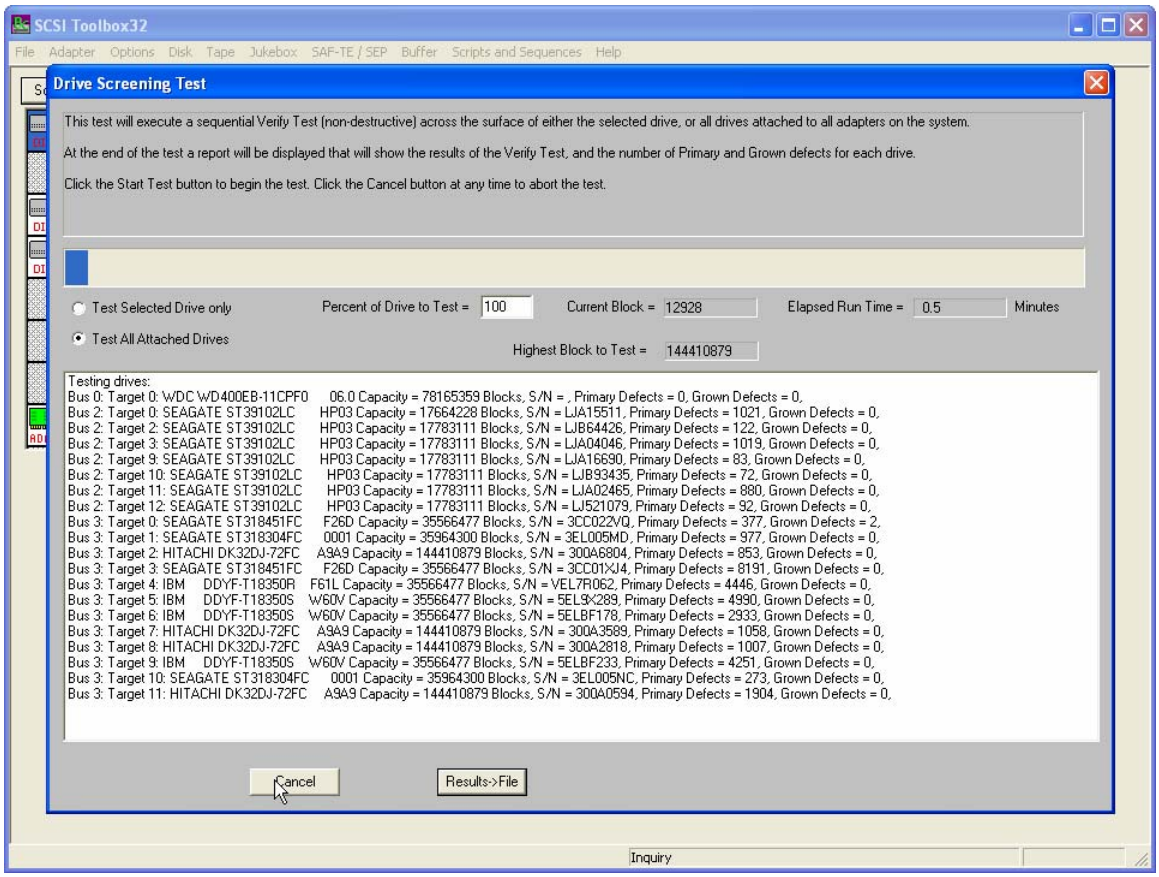


CD Performance Test



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

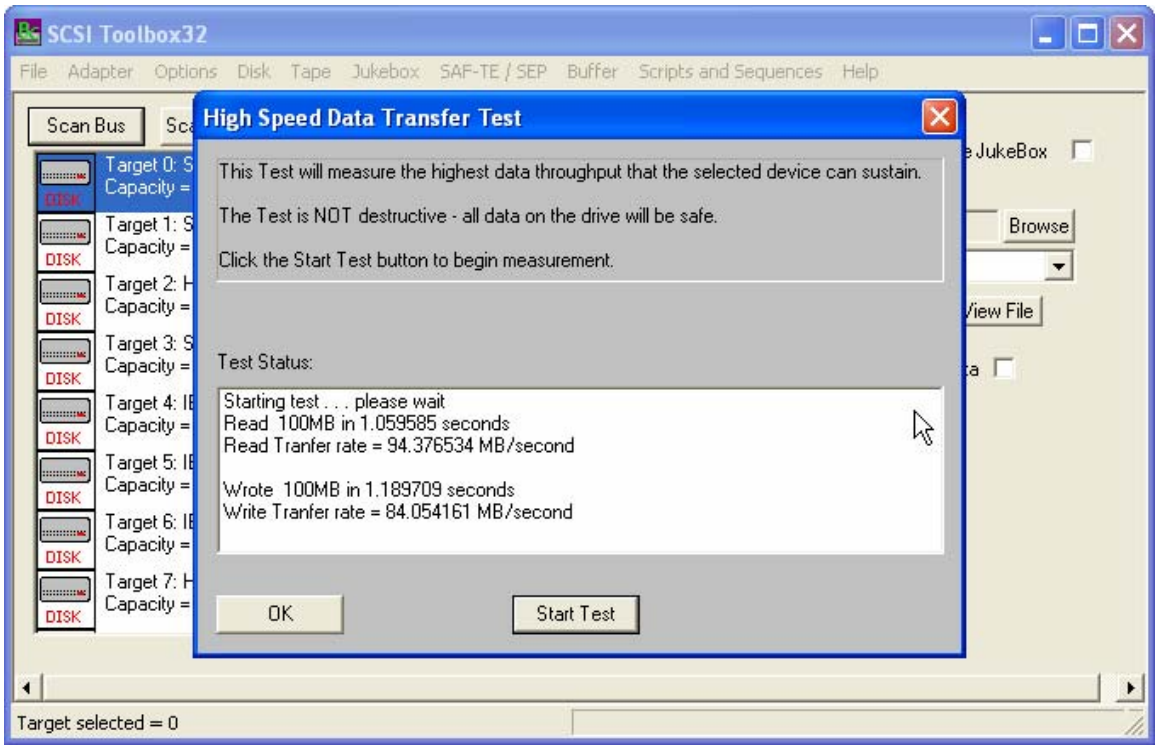
Drive Screen/Verify Test



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.

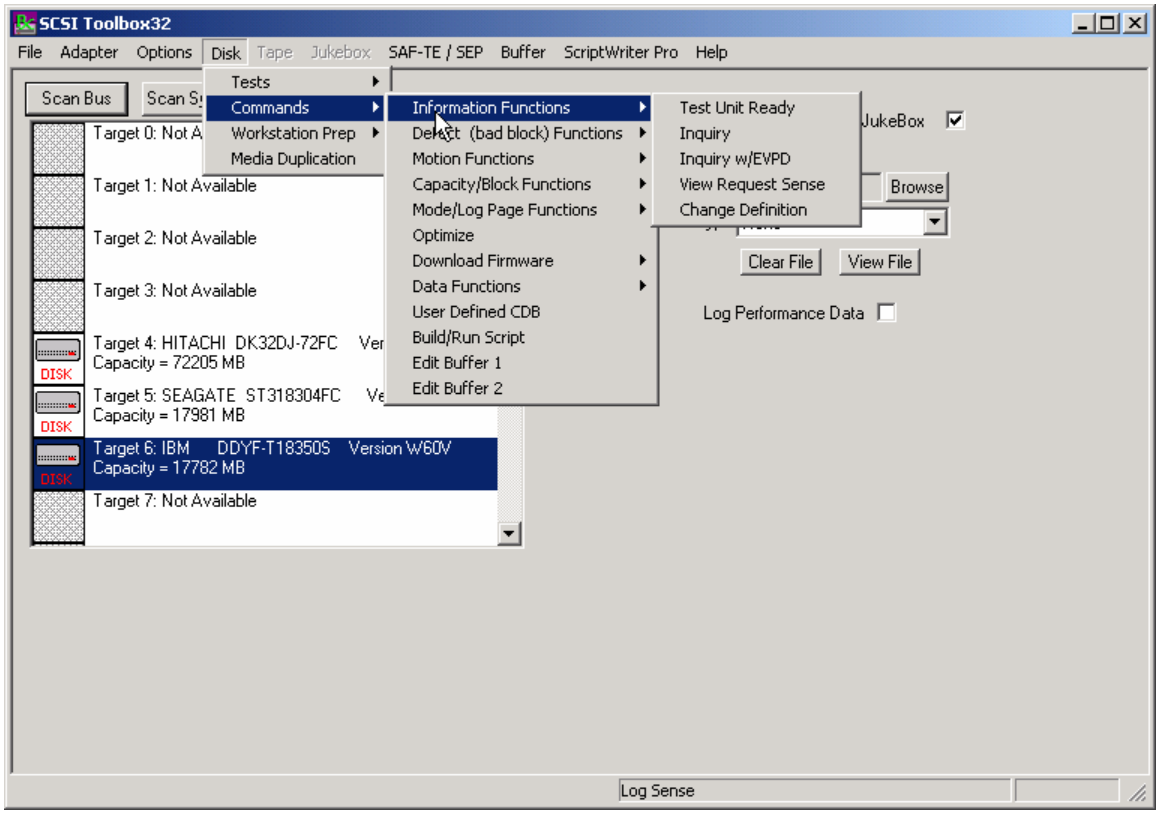
High Speed Data Test



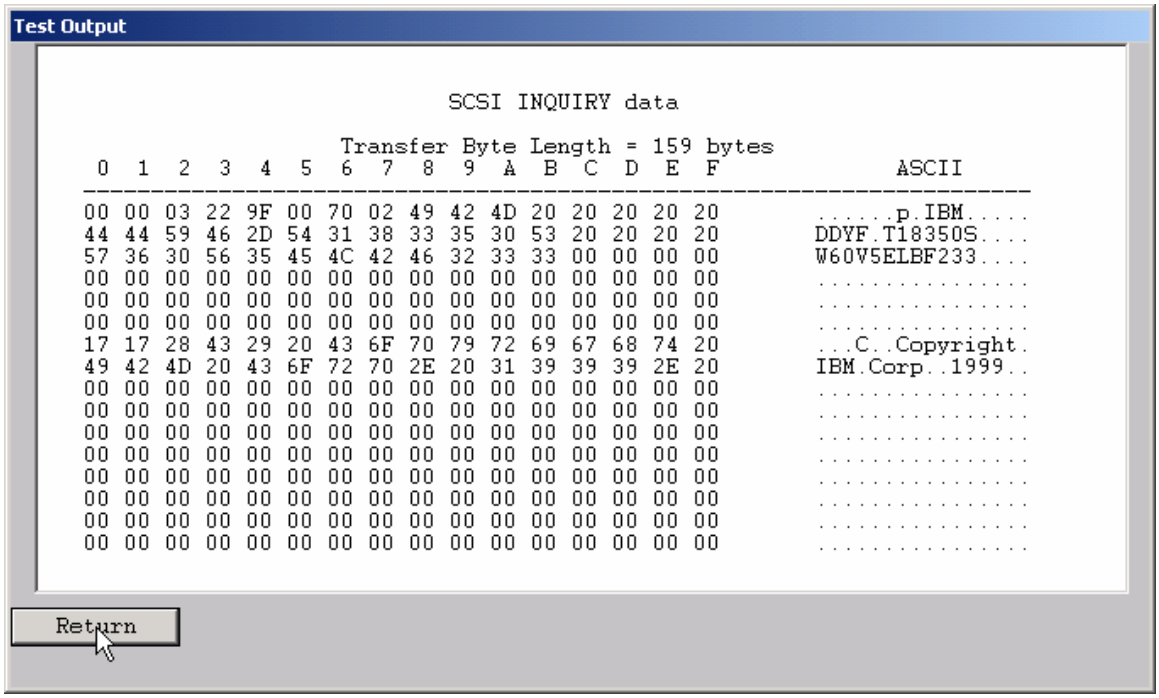
The High Speed Data test transfers data to and from the drive without transferring to the actual drive media. This allows the data transfer speed of the drive and host bus adapter to be determined.

Disk Commands

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

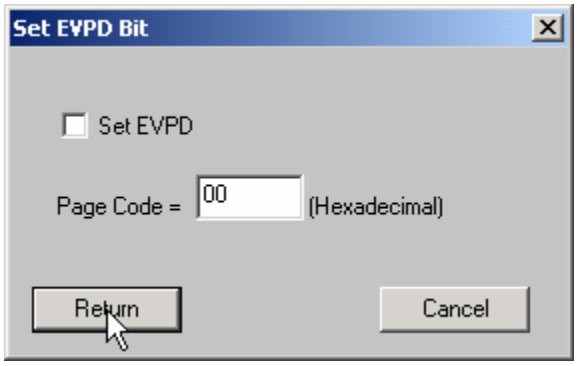


INQUIRY



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

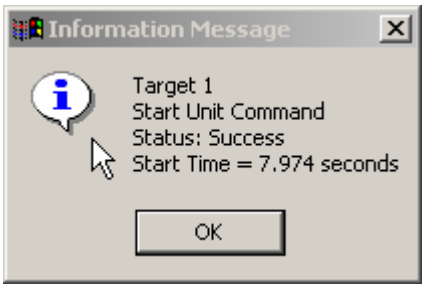
INQUIRY with EVPD



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

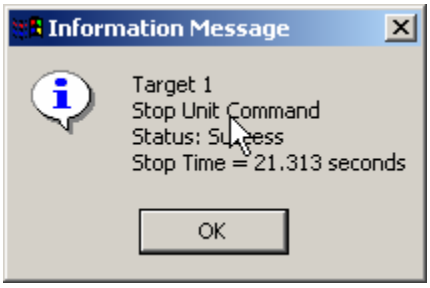
Start Drive (Motion Commands)

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.



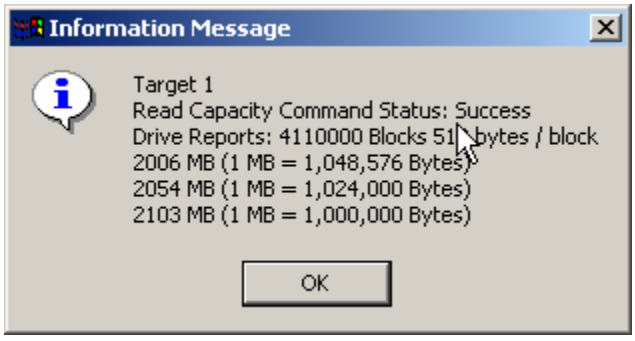
Start All (Motion Commands)

This command will start all drives on the selected HBA.

Stop All (Motion Commands)

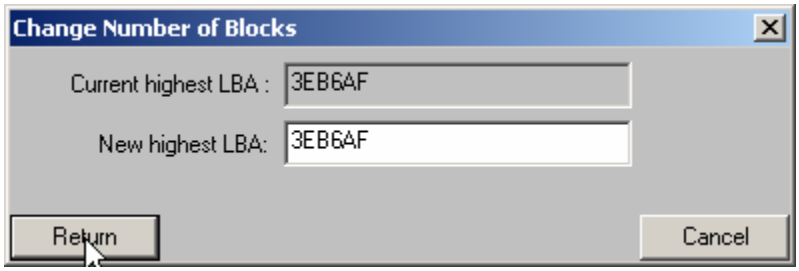
This command will stop all drives on the selected HBA.

Read Capacity (Capacity/Block Functions)



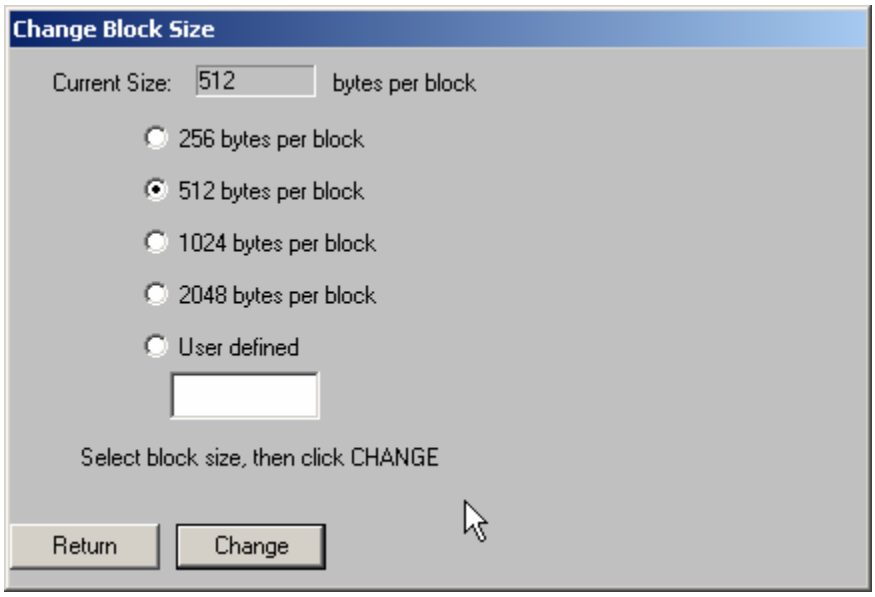
Change Capacity (Capacity/Block Functions)

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.

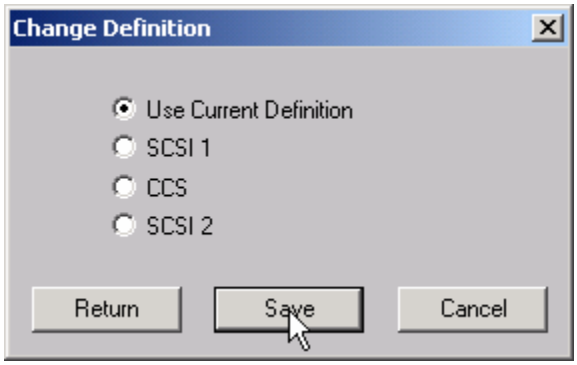


Change Blocksize (Capacity/Block Functions)

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!

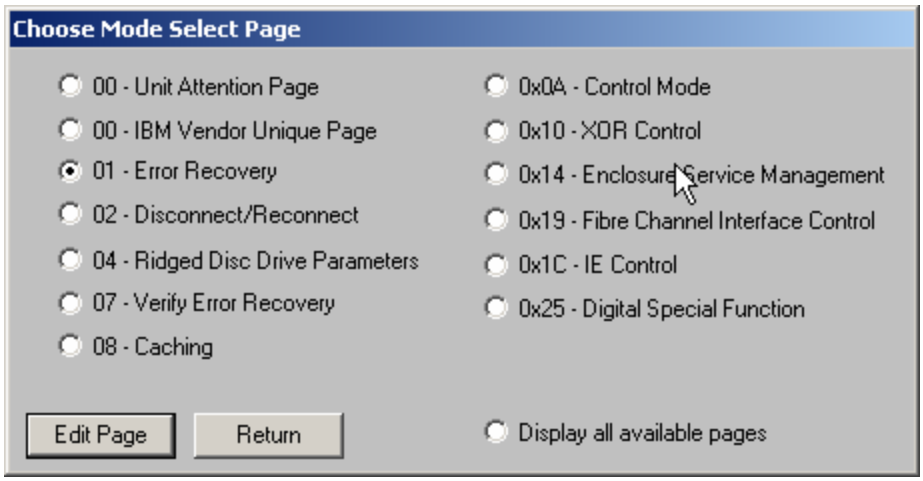


Change Definition

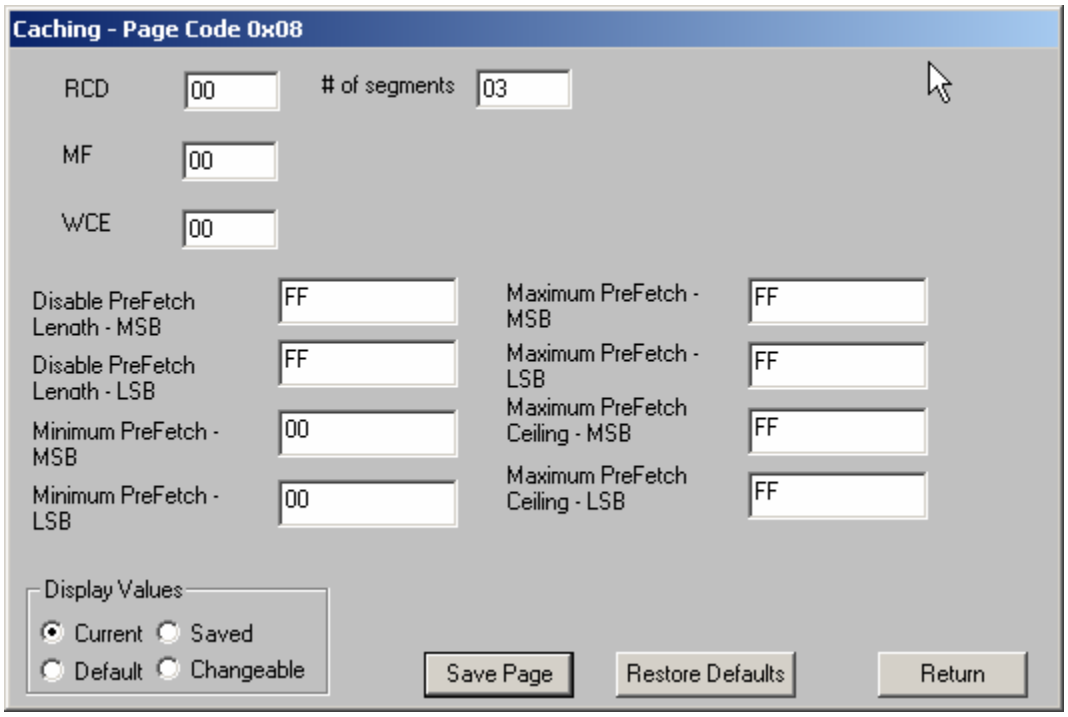


View/Edit Mode Pages (Mode/Log Page Functions)

The following MODE PAGES can be viewed and edited:



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



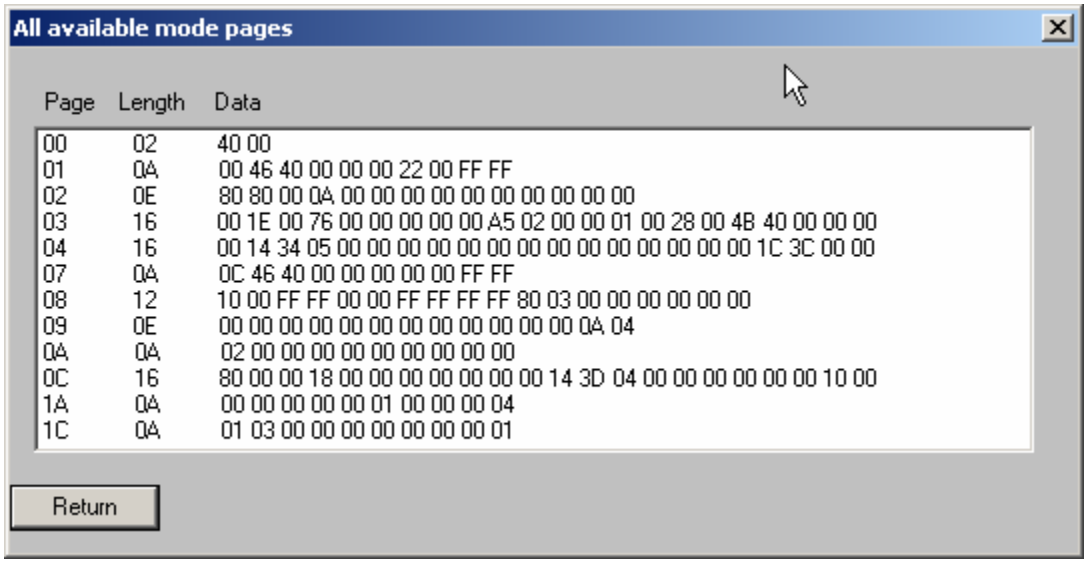
Any of the MODE PAGE parameters may be changed/edited. To save the page data (MODE SELECT) click the Save Page button.

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

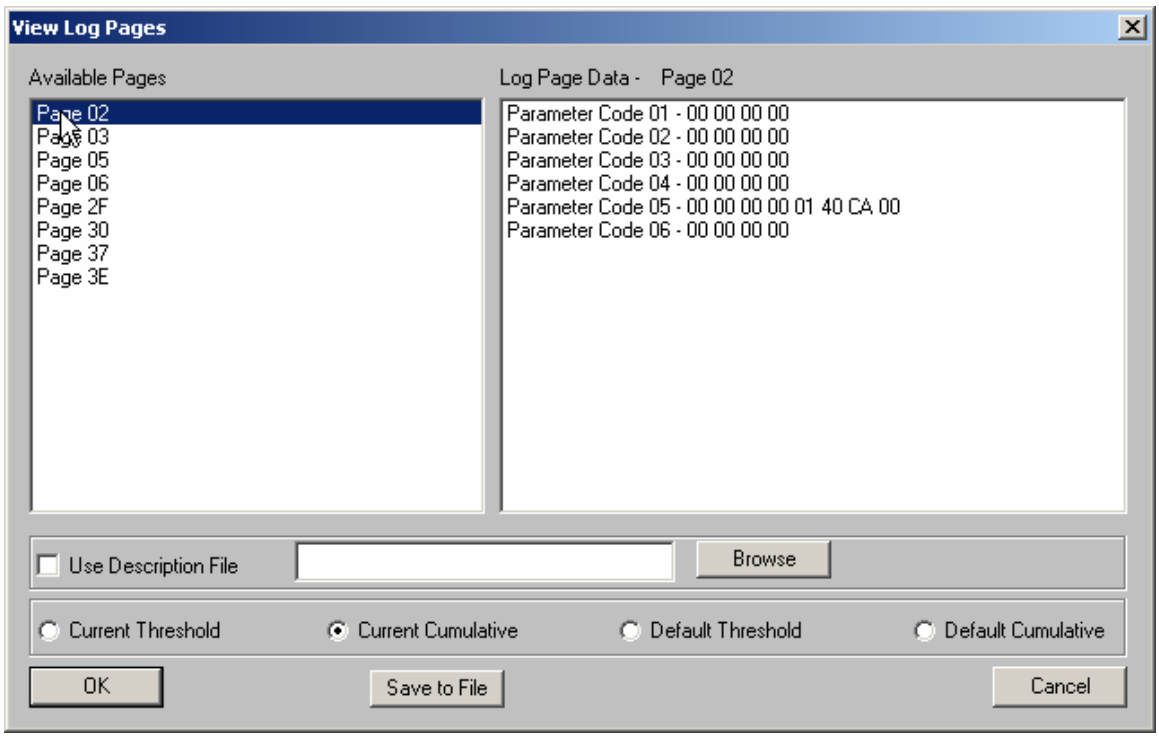


Restore Default Mode Pages

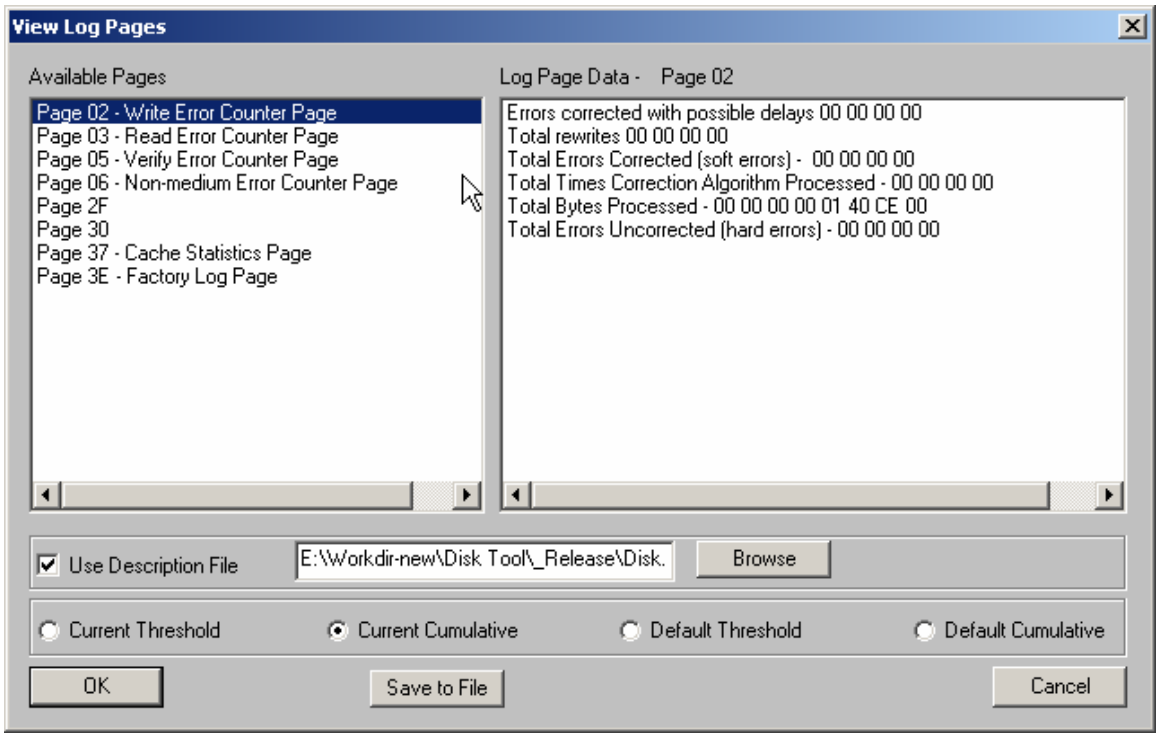
This command will restore all of the drives MODE PAGES to their factory default values.

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:

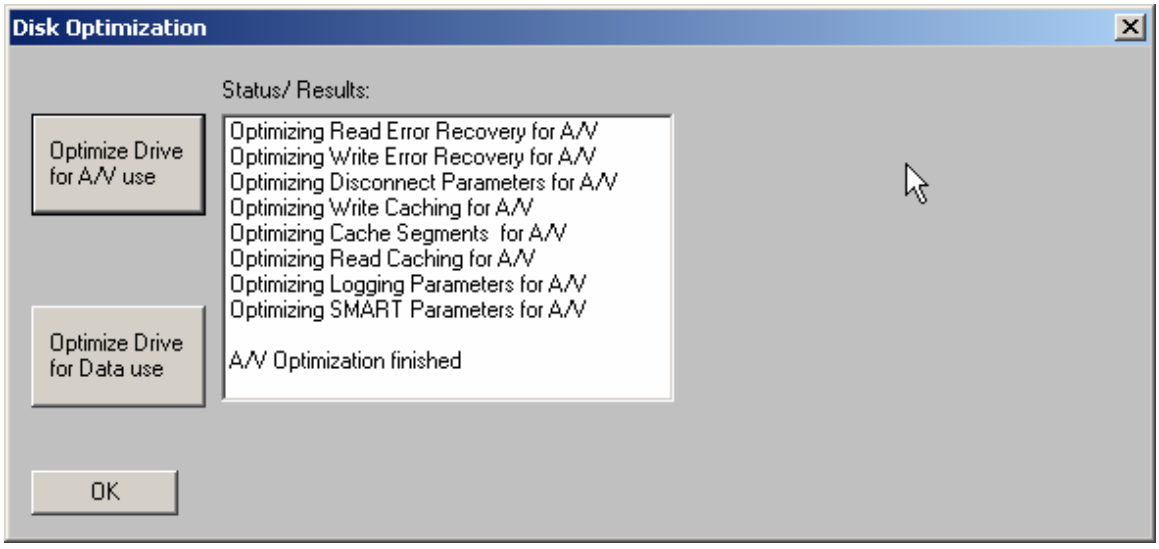


Clicking on the Use Descriptor File will use a text file (disk.def) to interpret the log data, as below:



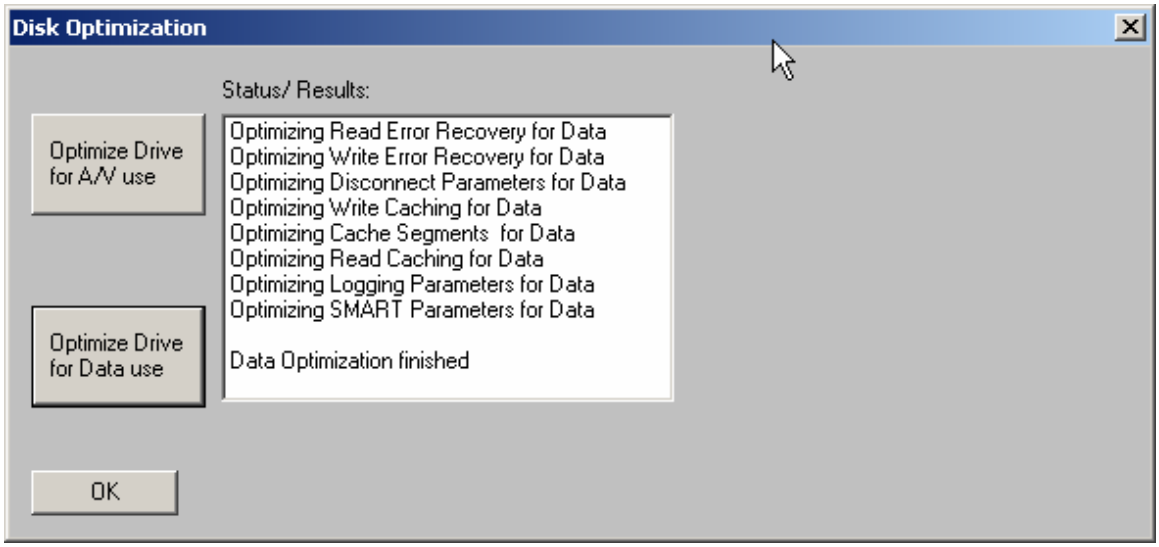
Optimize drive for A/V use

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



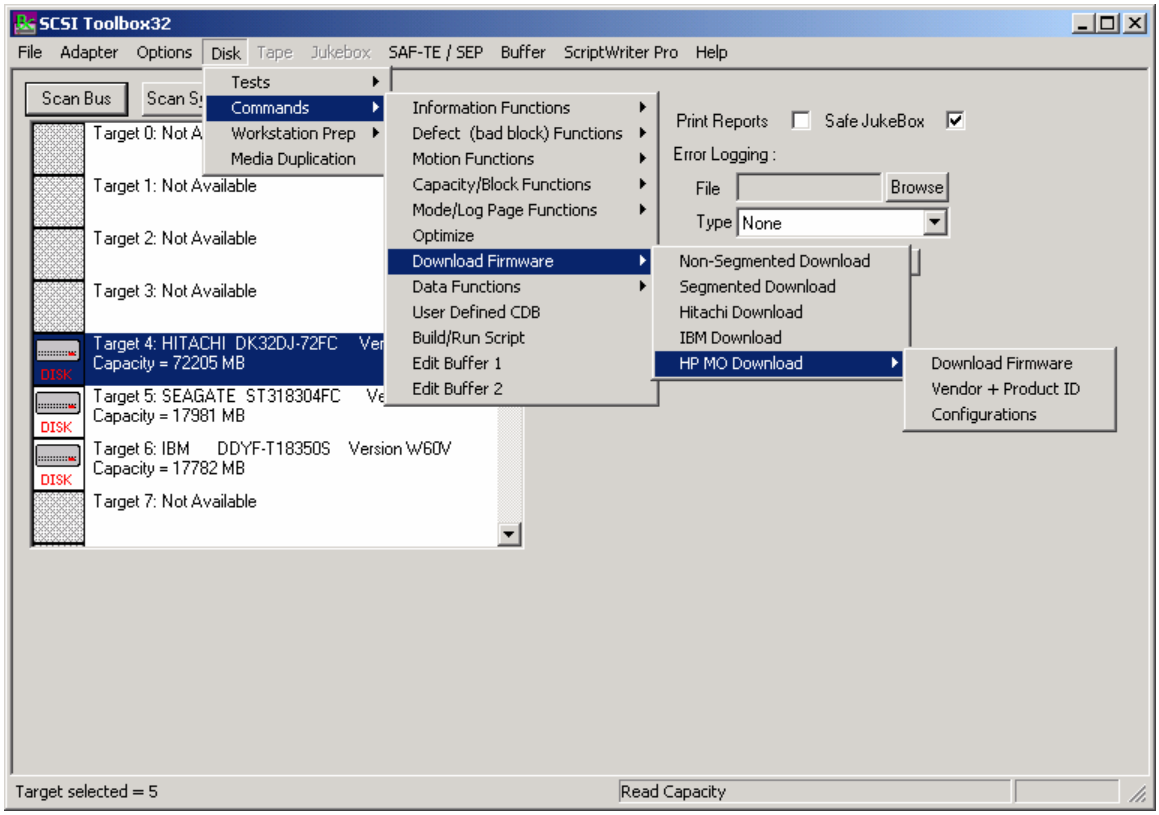
Optimize drive for Data use

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.



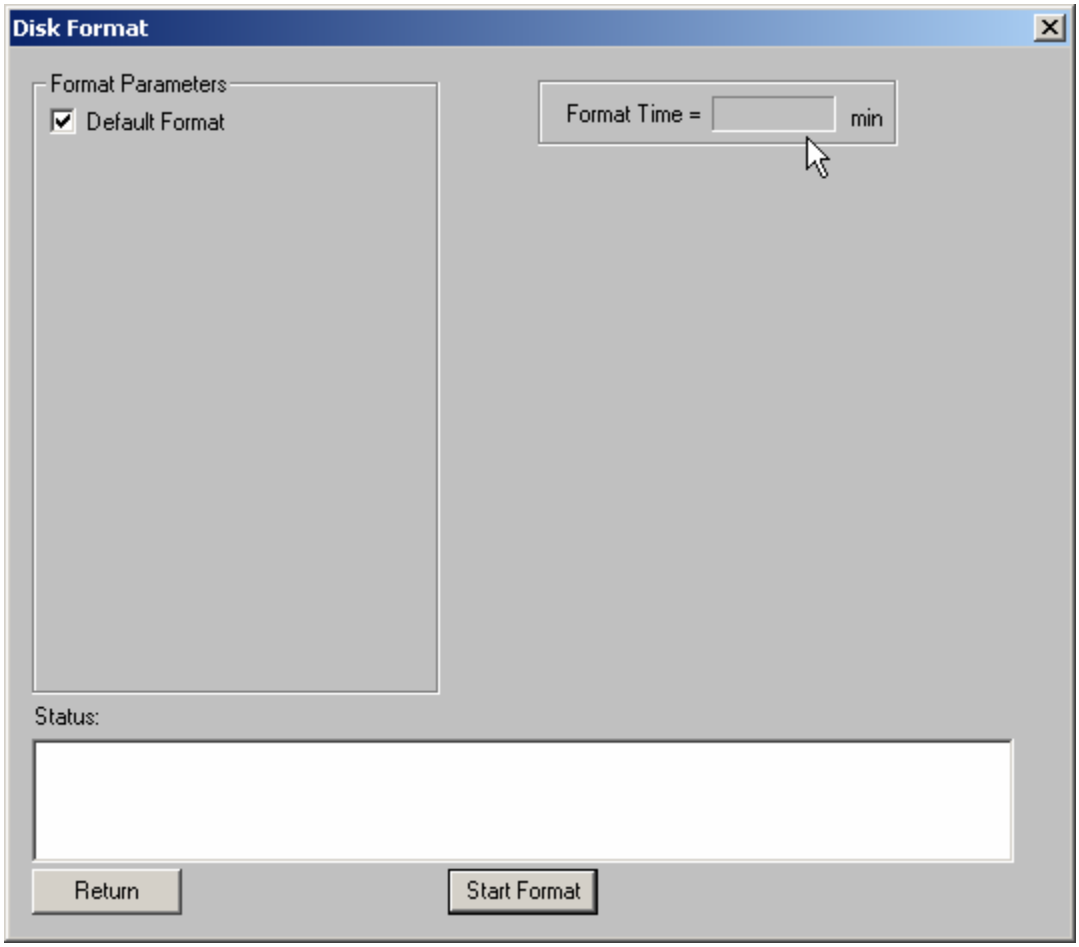
Use the Browse button to specify the firmware file name, then click the Download button.

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

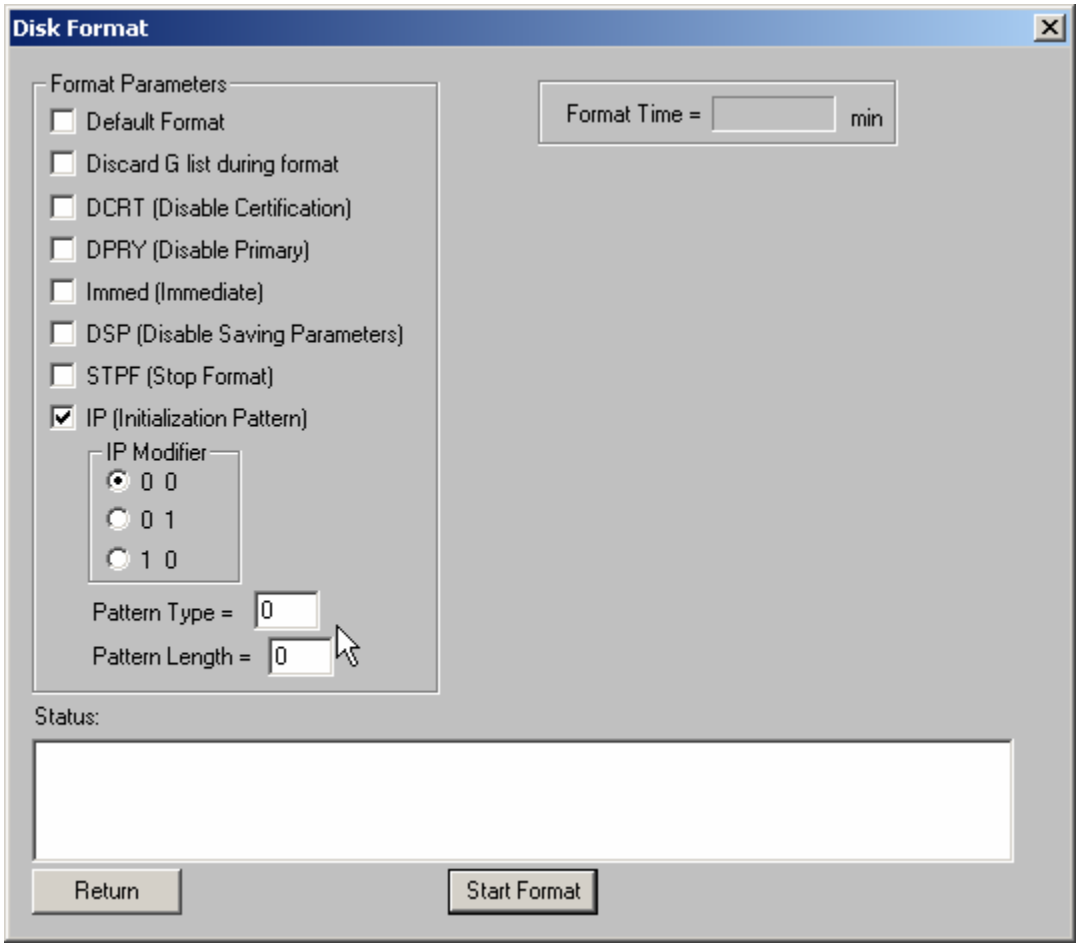
If the drive requires a non-segmented download use this choice.

Format

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

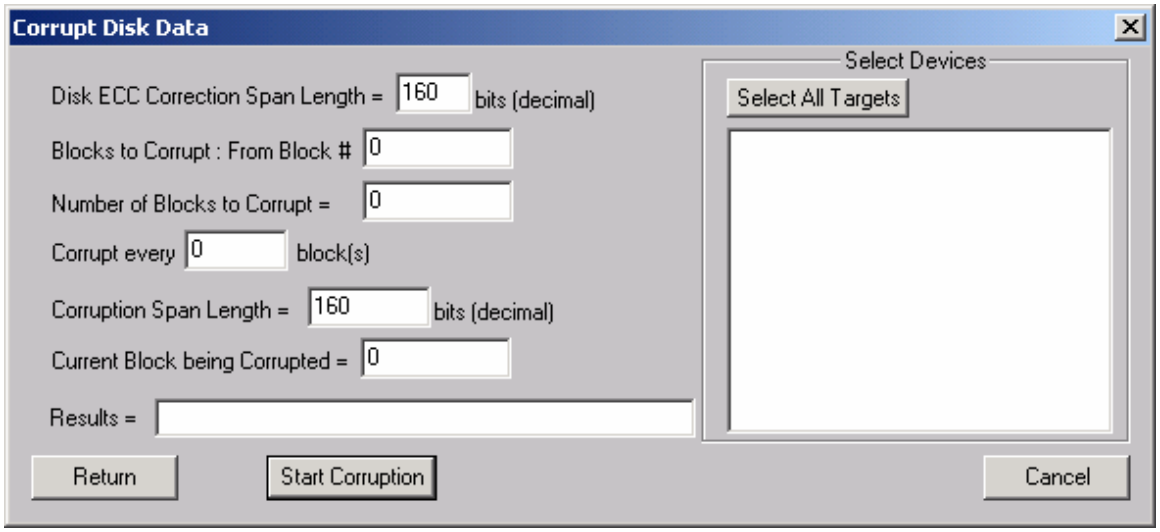


If the Default Format check box is “un-checked” the following format options can be specified:



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

Corrupt Data

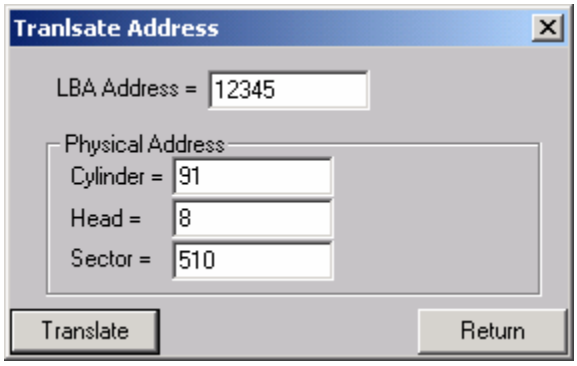


The corrupt data choice allows you to create soft ECC errors in any block of the drive. Note – these errors can be corrected or removed by writing over the corrupted blocks. The Sequential Write test is used to do this.

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.

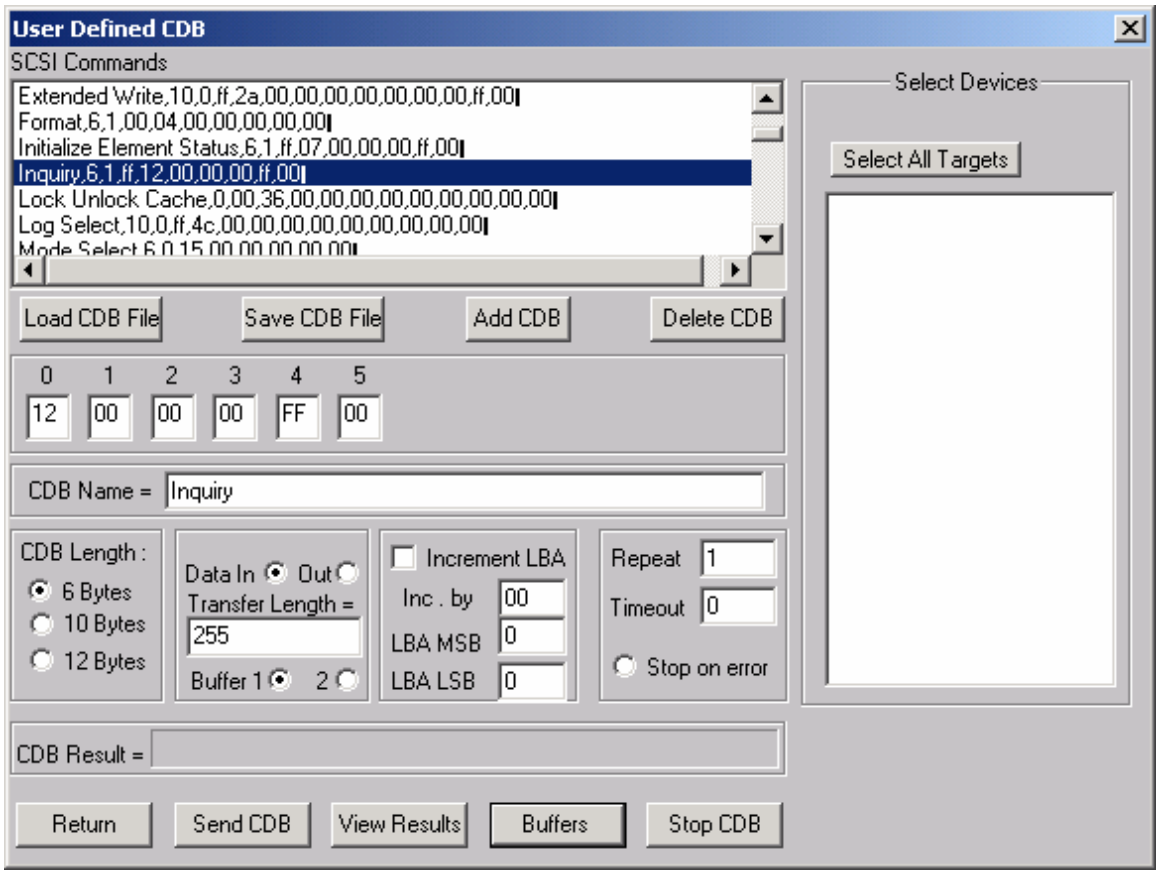
Once you have filled in the above selections, click on the Start Corruption button to begin.

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 SCSItoolbox 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.

Definining 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:

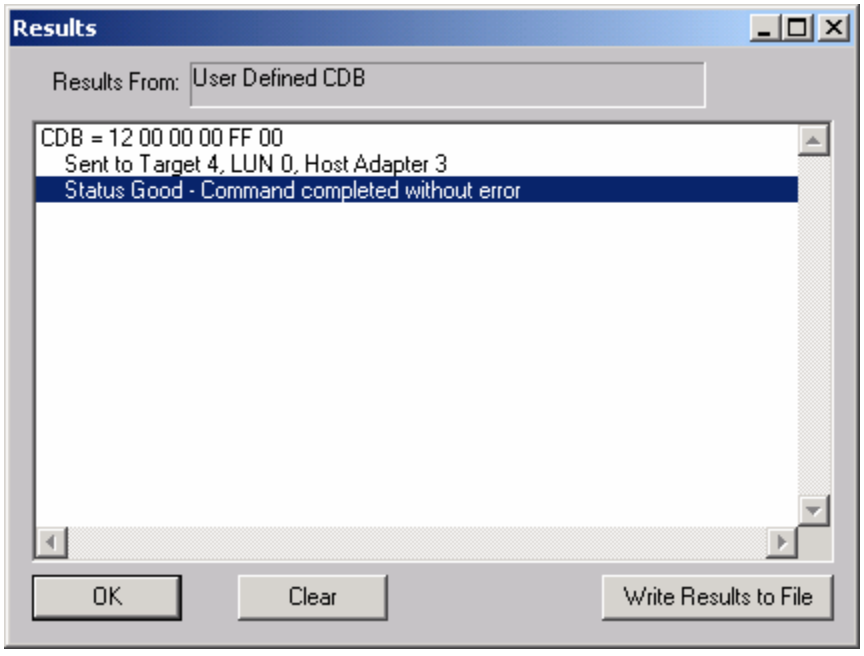
- Data direction incorrect
- Data length specified in CDB does not match Data Transfer Length

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:

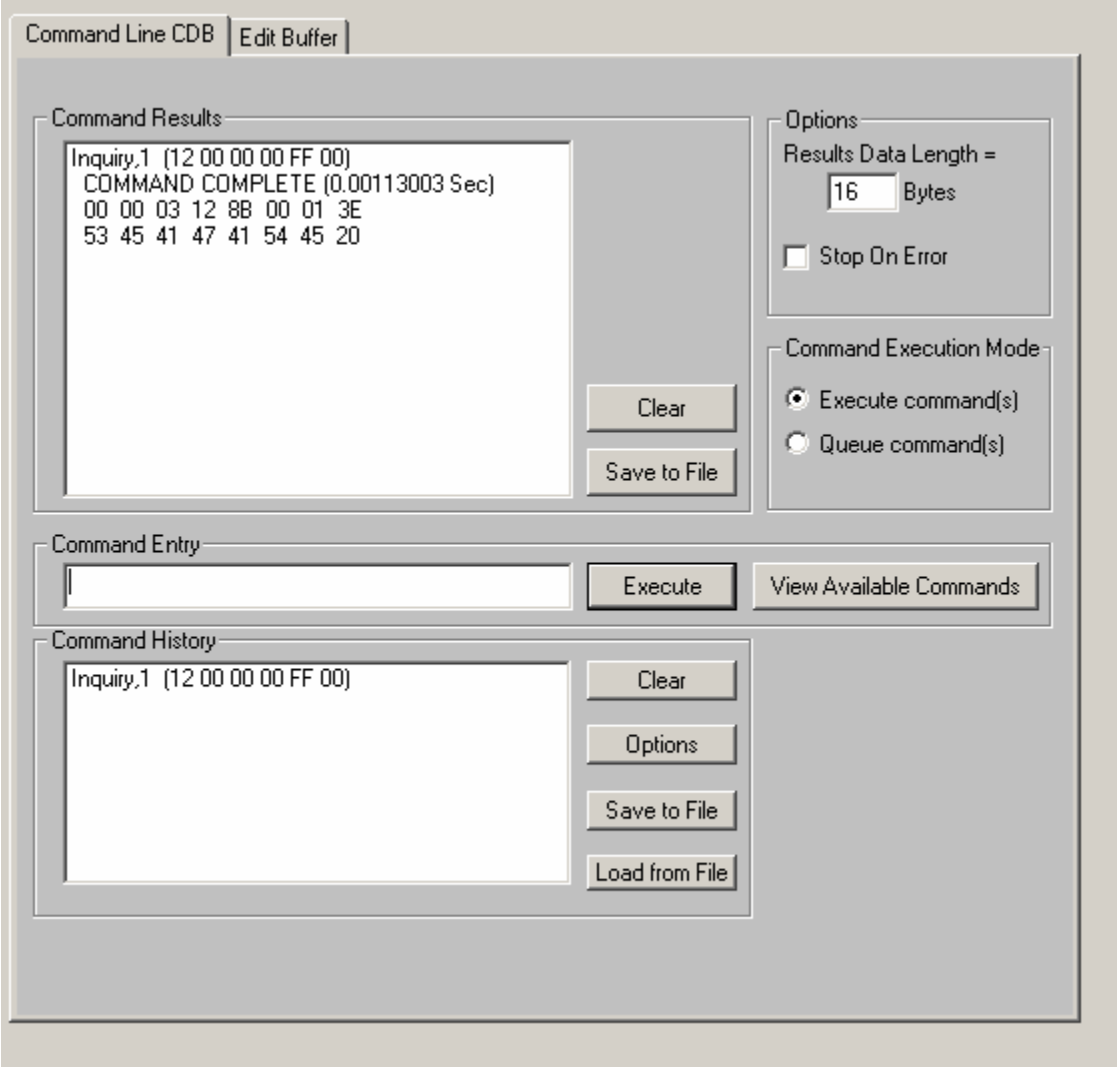
- Check the Increment LBA box
- Specify which CDB byte is the LSB of the SCSI CDB
- Specify whick CDB byte is the MSB
- Specify how much you want to increment by

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

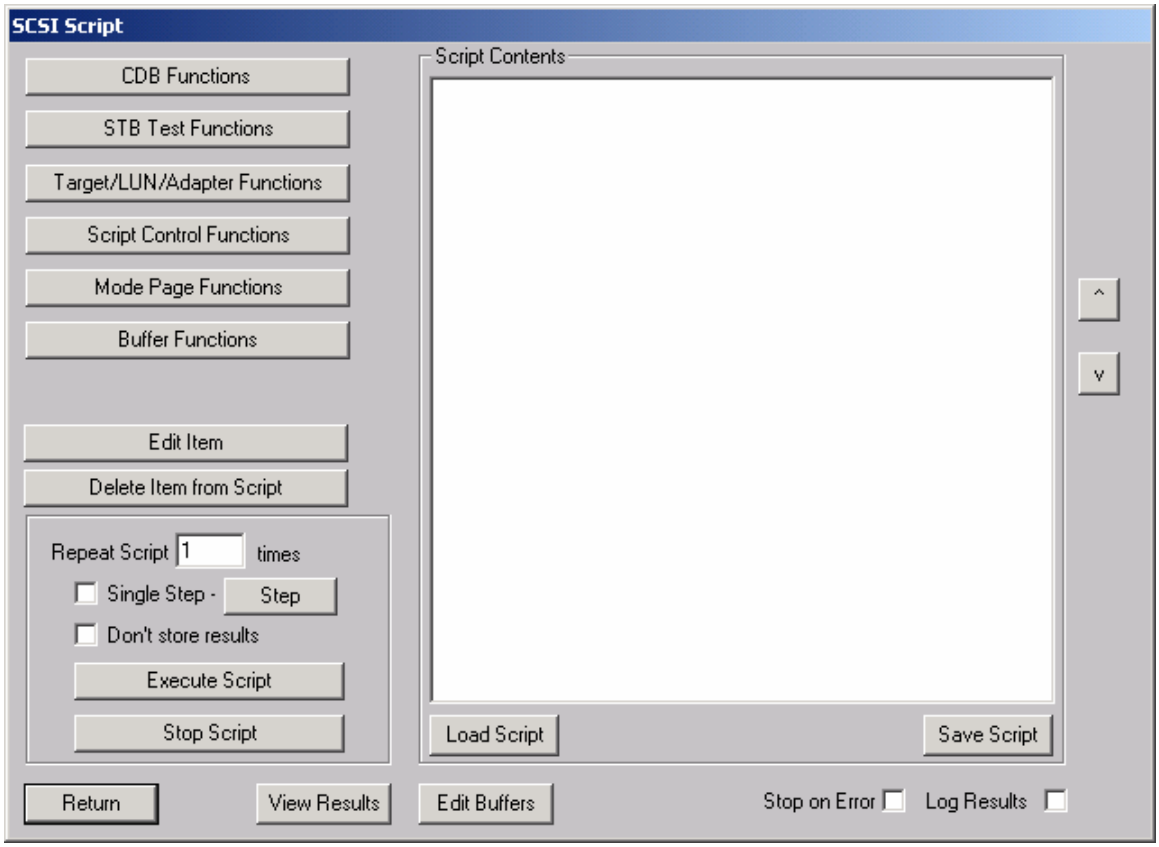
The “Results Data Length” allows you to specify how many bytes of data are displayed in the “Command Results” window. If a command transfers data, this many bytes will be displayed along with the command results. If the command terminates in a check condition, this many bytes of sense data will be displayed.

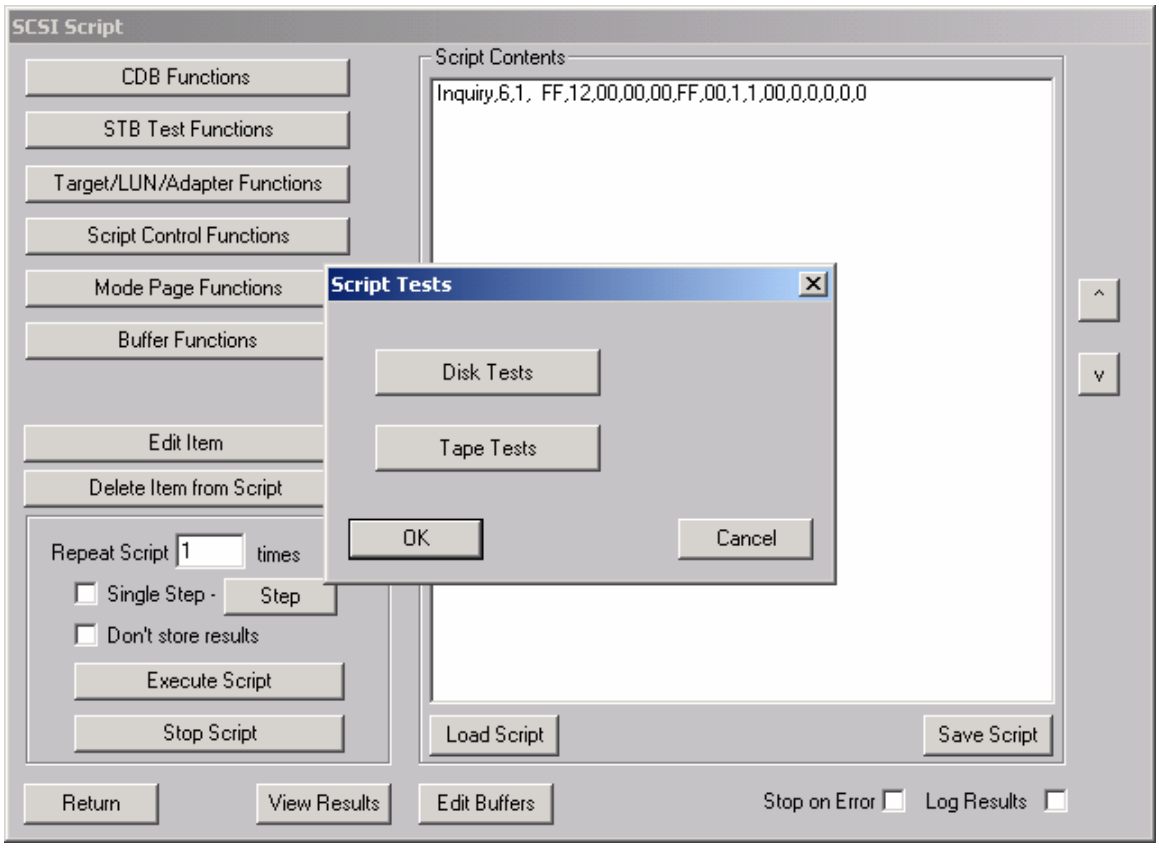
The entire data buffer can be viewed, modified, and saved/loaded from a file in the ***Edit Buffer*** tab page.

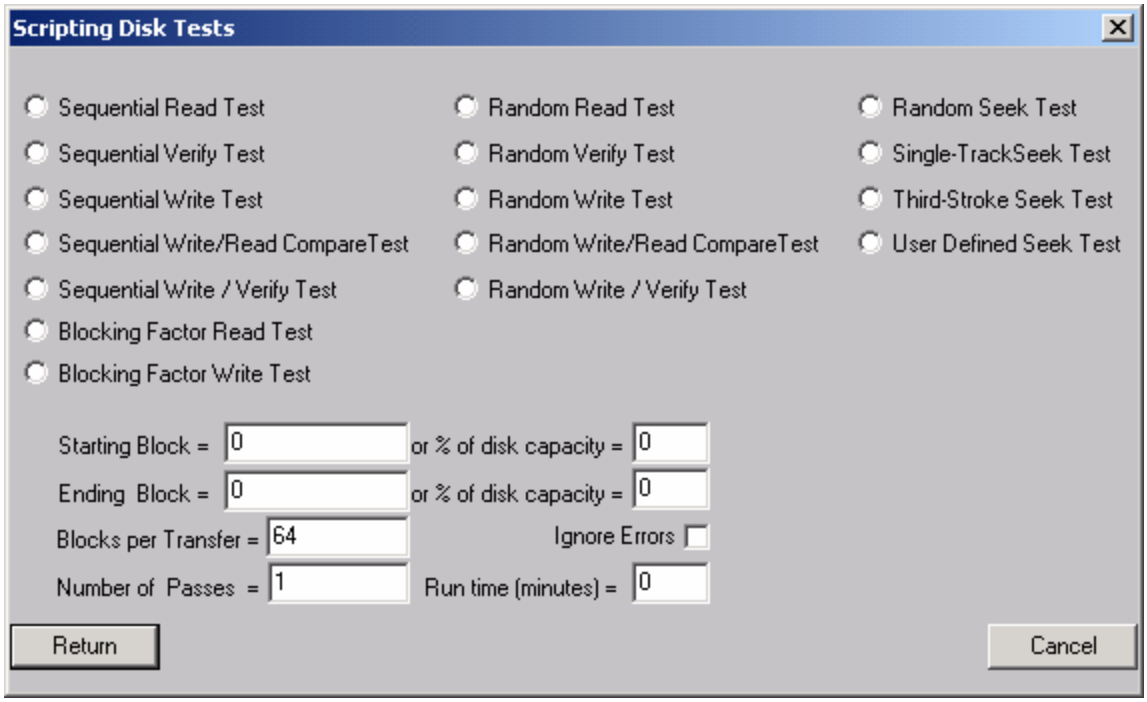
Re-issuing commands

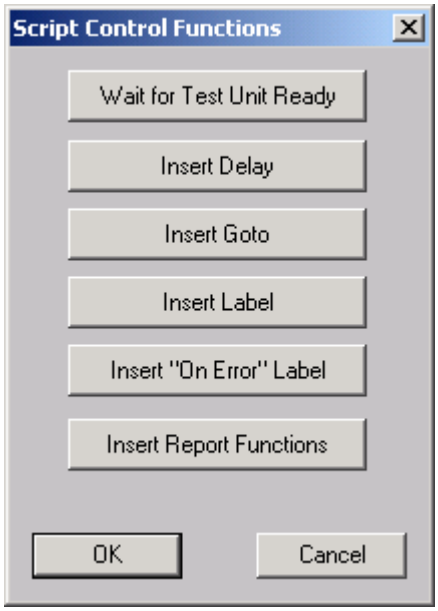
Any command or commands can be re-issued by selecting them (clicking on them) in the “Command History” window. Once the commands you want to reissue are highlighted, click the “Execute” button and they will be reissued.

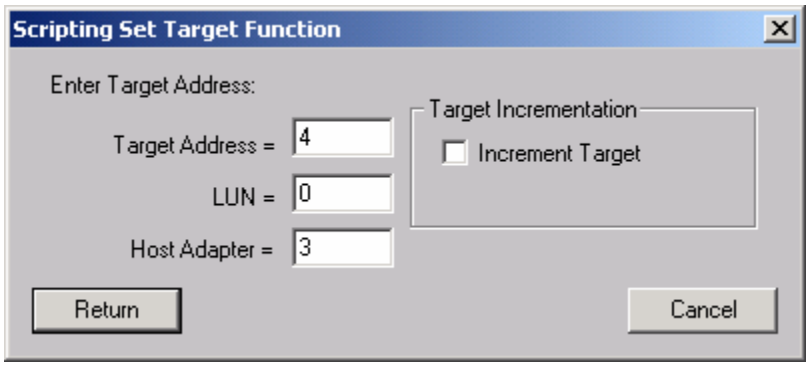
Build Run Script

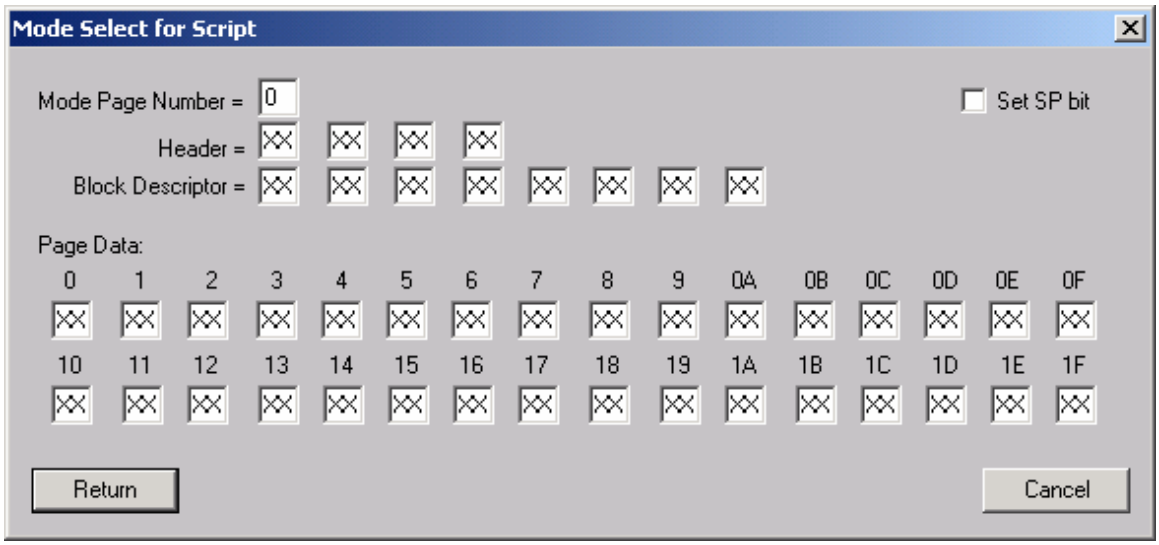


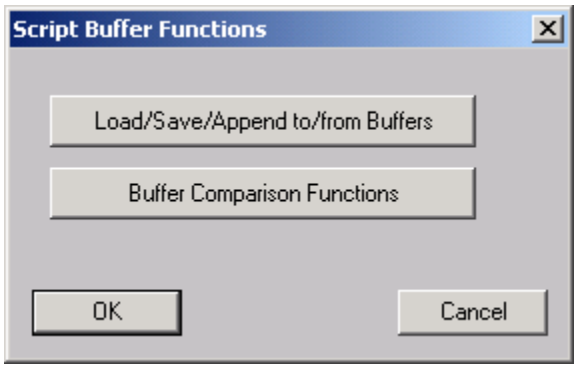


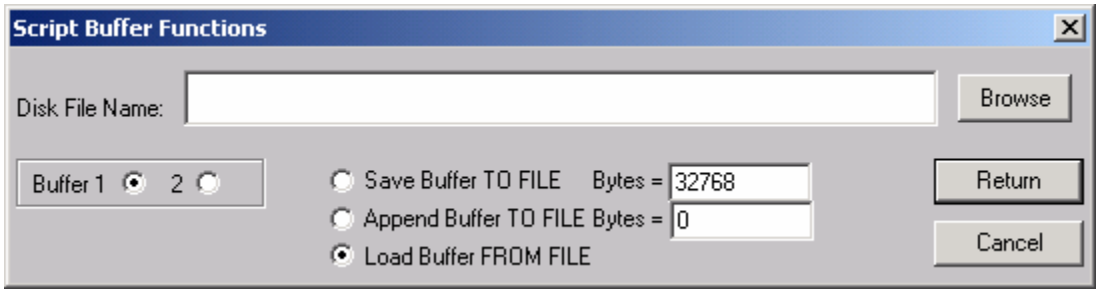


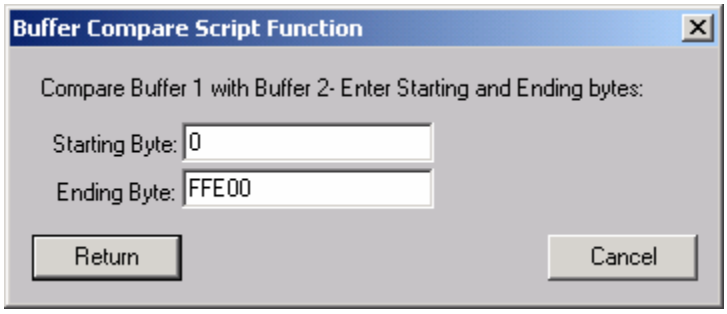


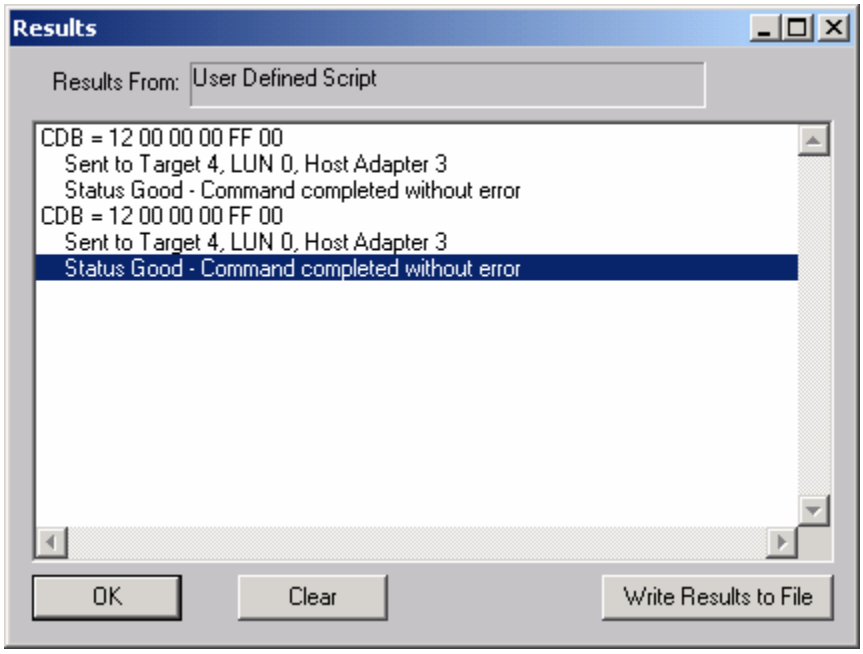






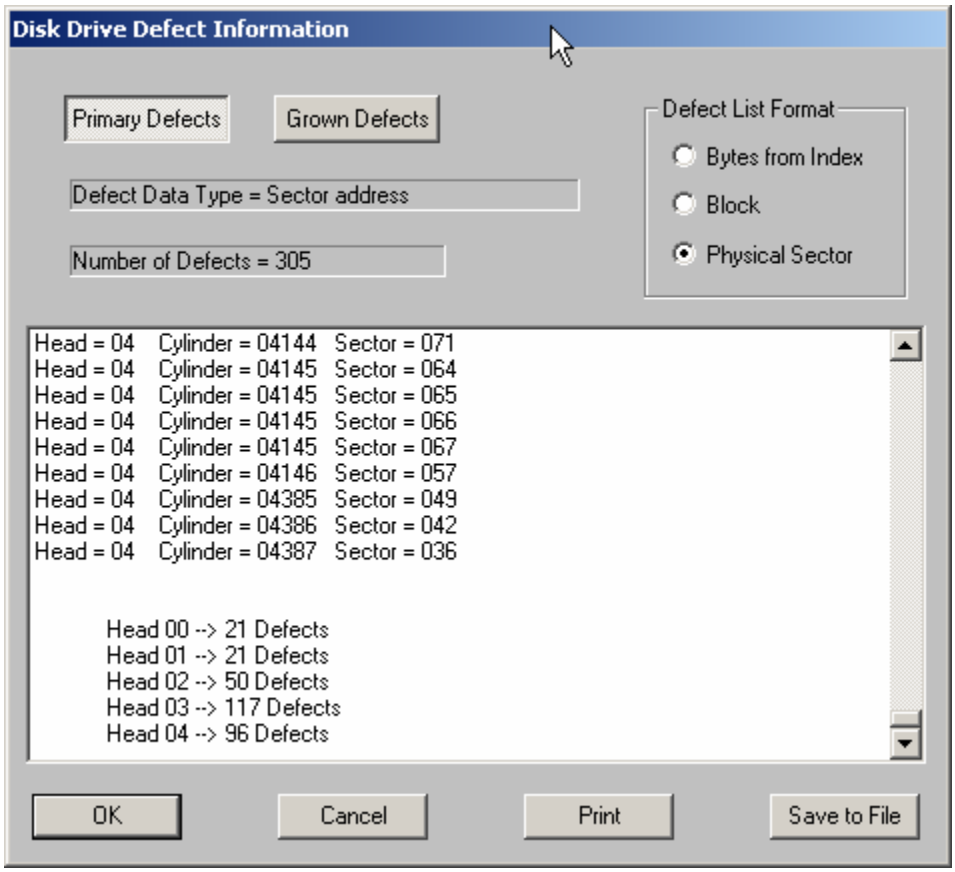






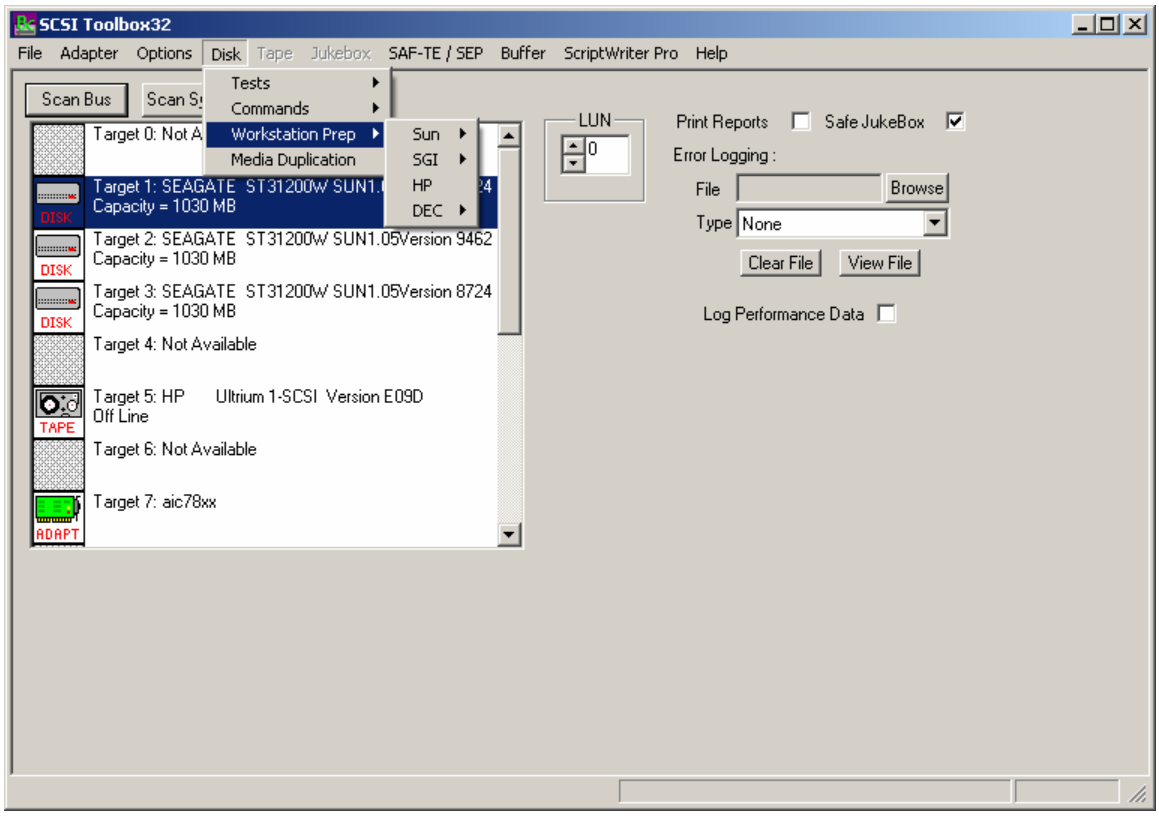
Display Defect Data

This commands 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.

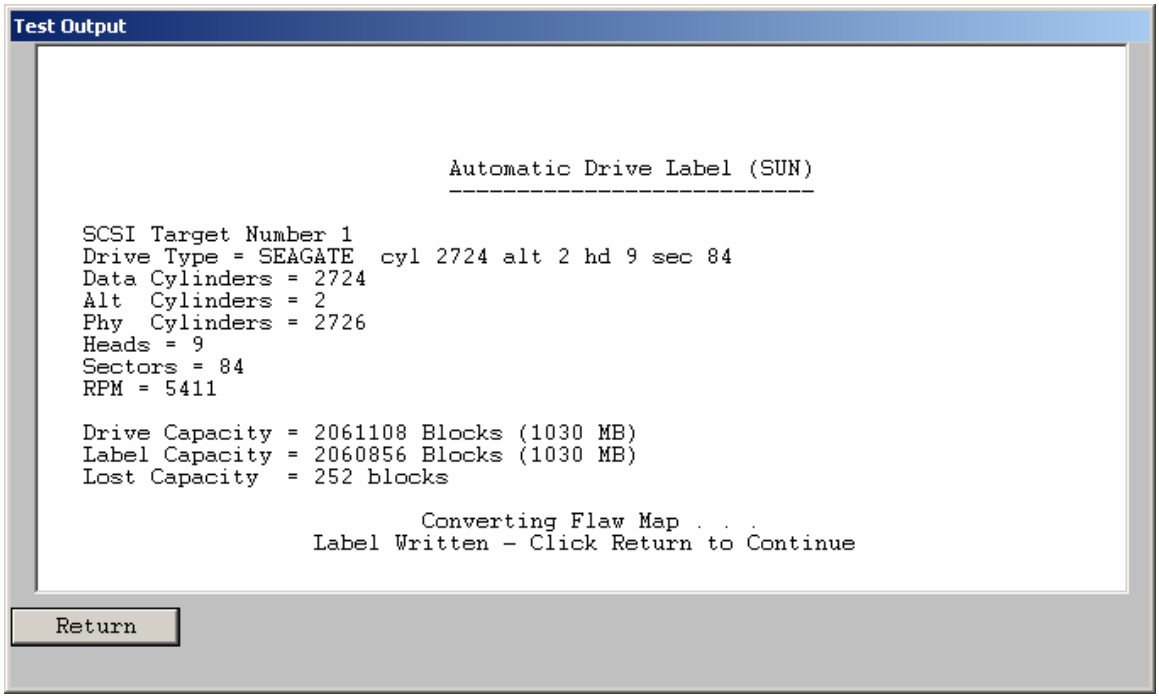
A good strategy for preparing disks is to first qualify the drive as functional. This can be quickly ascertained by running the Quick QC Test. If you are more comfortable completely reformatting the drive, you can do so using the Format Command. If you want to test the drive for an extended period of 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

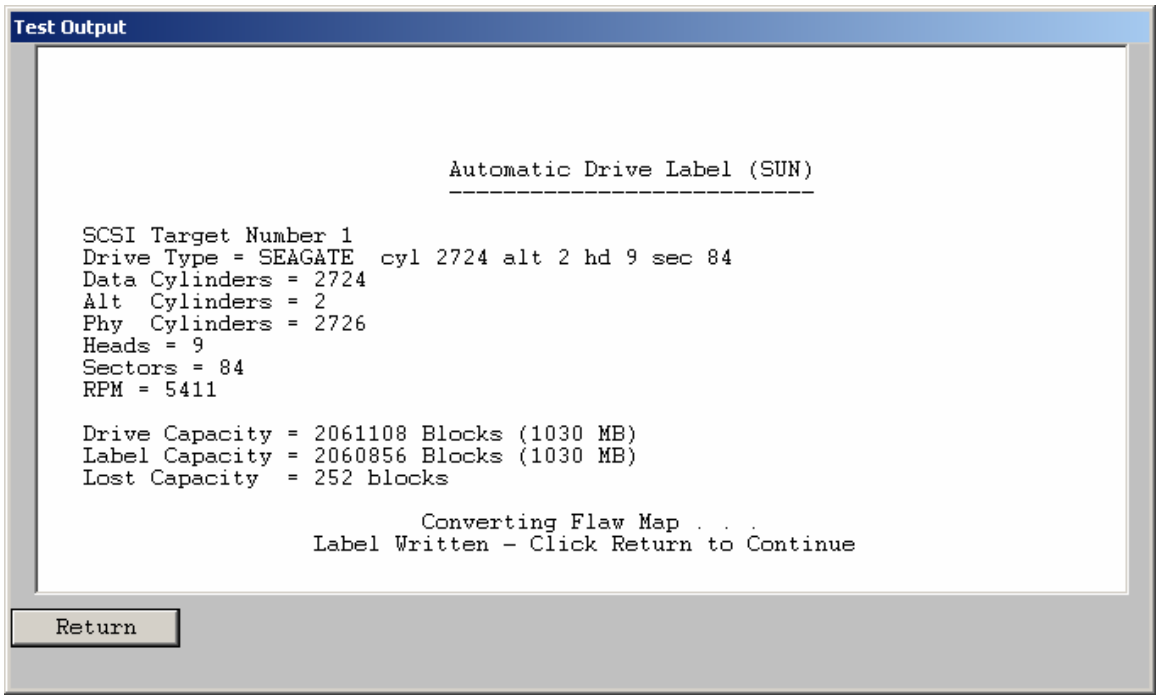
Read Existing Label

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.



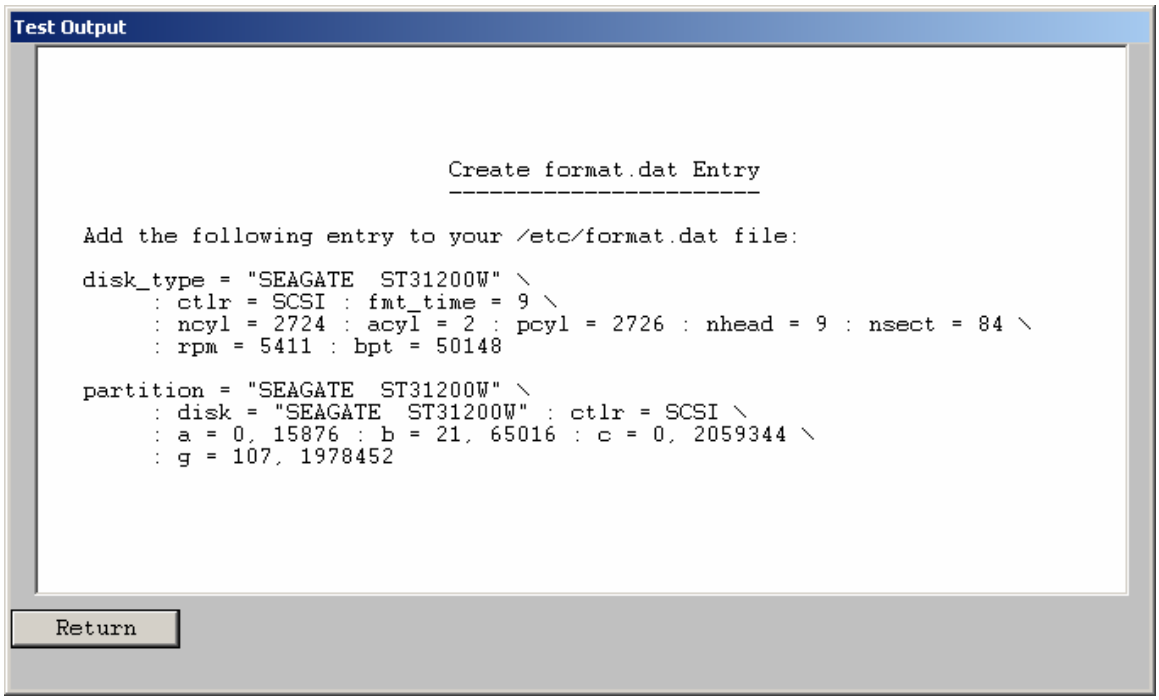
Write New Label

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.

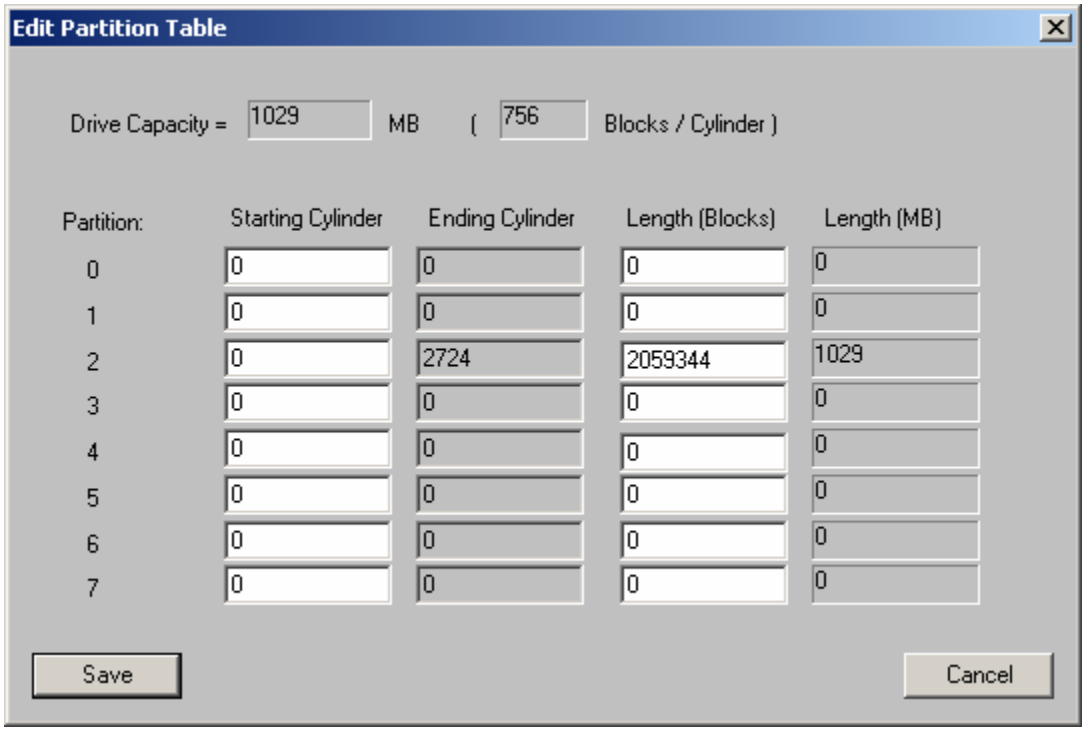


Calculate Format.dat Entry

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



Edit Sun Partition Table

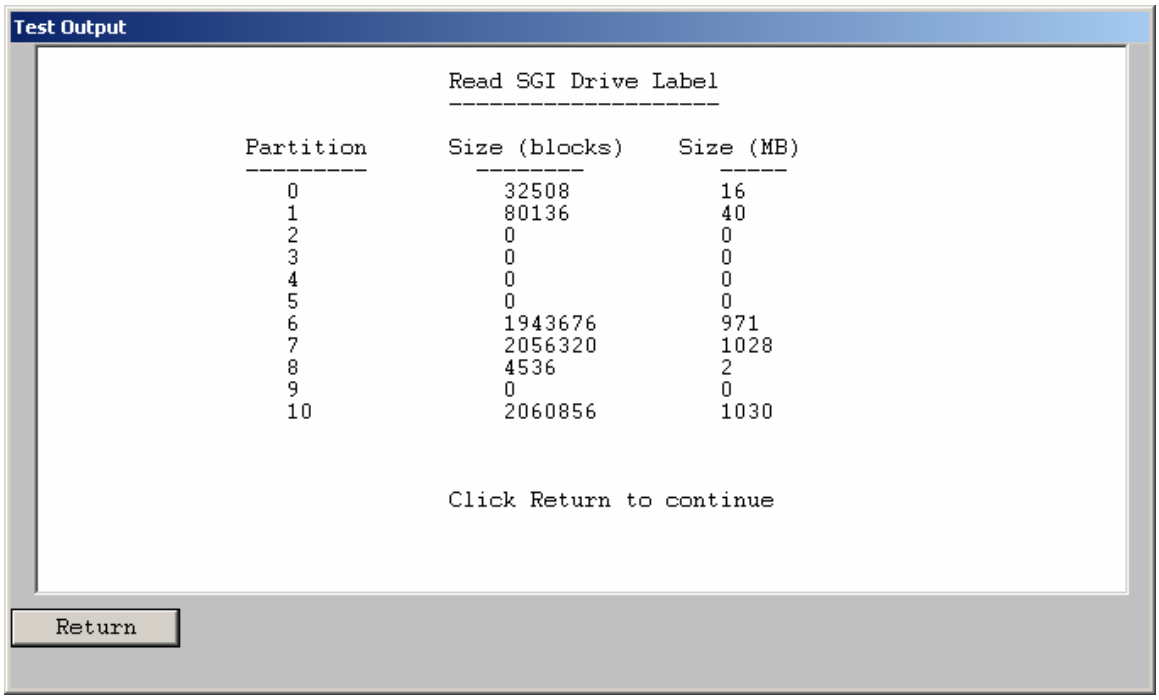


This function allows you to edit the partition table, creating new partitions or editing the size of existing partitions.

The SGI Workstation Menu

Read Current Label

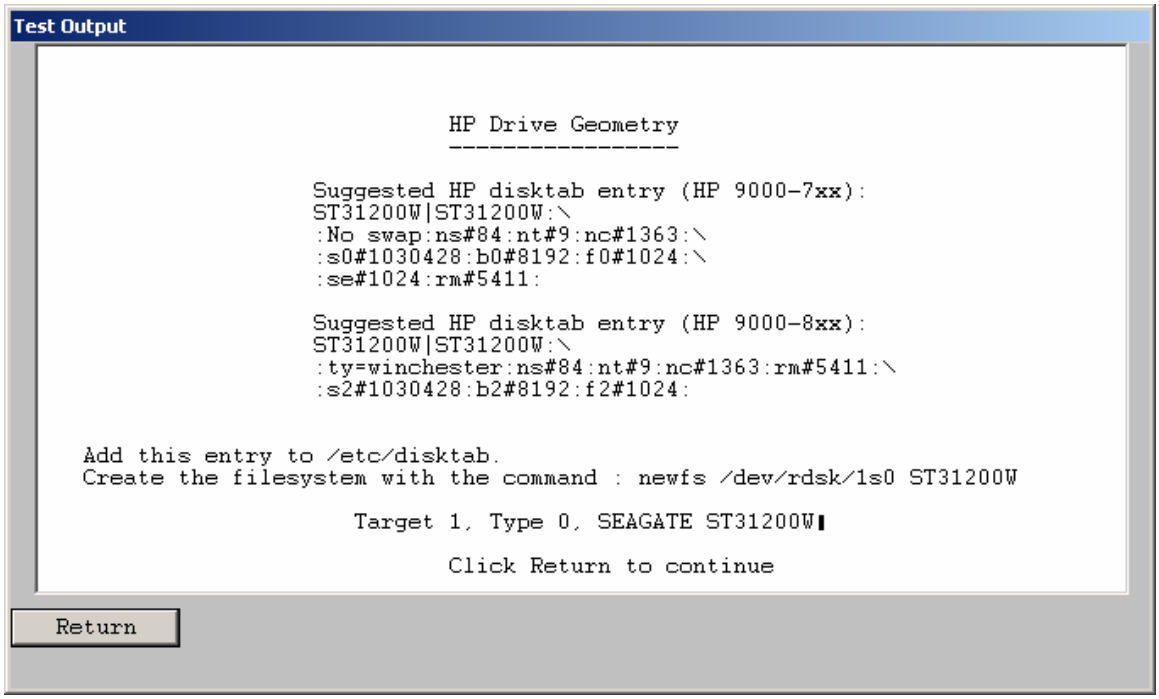
This option will check a drive for the presence of a valid SGI label. If found the details of the label will be displayed.



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

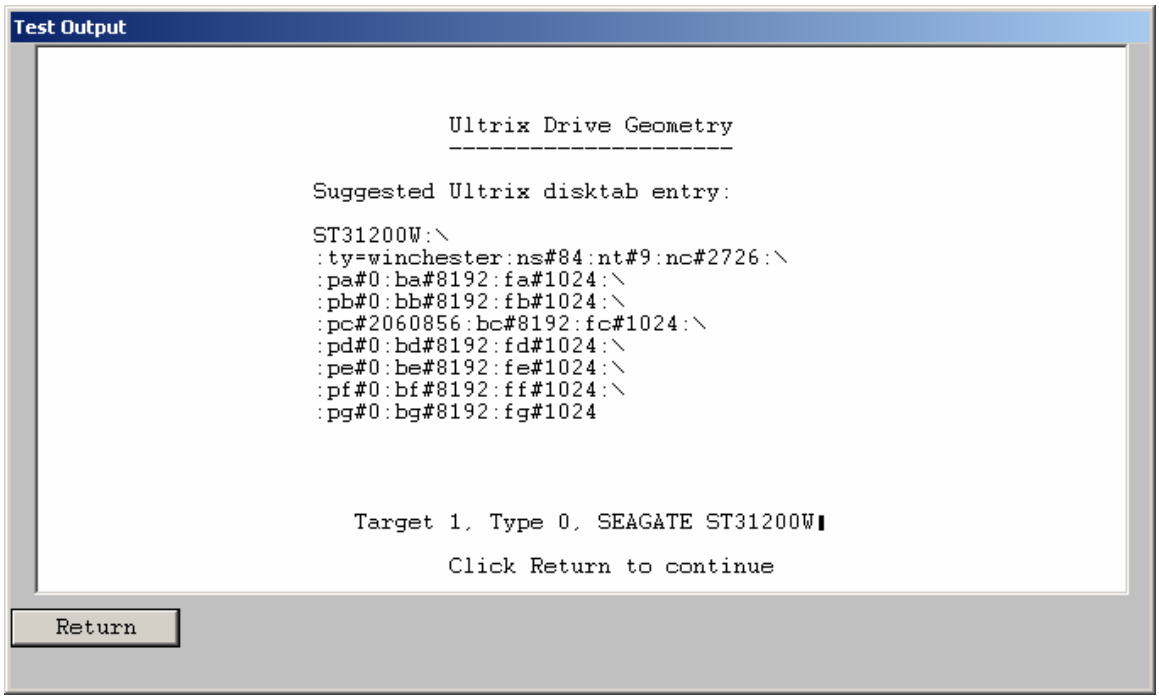
Suggested HP disktab entry:

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/rdsk/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.

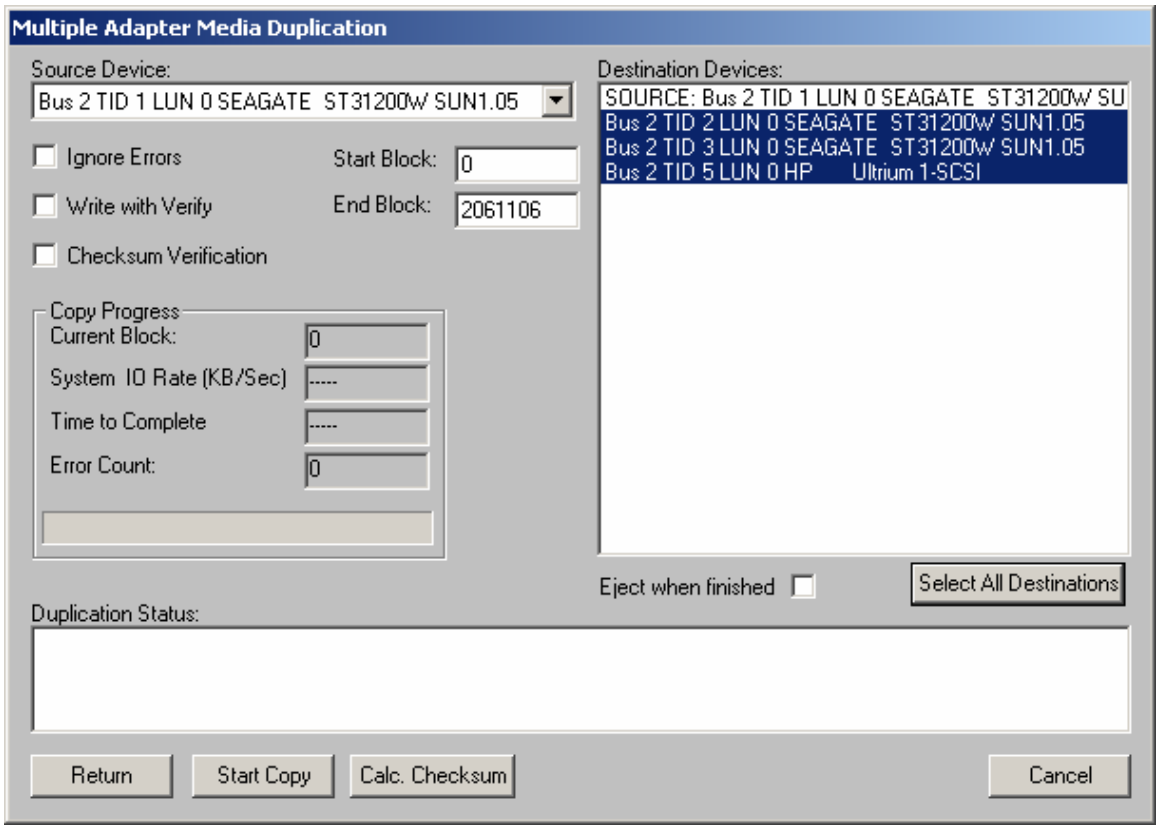

```
                OSF Drive Geometry
  Suggested OSF disktab entry:
          IBM_OEM_:\
           :ty=winchester:ns#63:nt#15:nc#2076:\
           :pa#0:ba#8192:fa#1024:\
           :pb#0:bb#8192:fb#1024:\
           :pc#1961820:bc#8192:fc#1024:\
           :pd#0:bd#8192:fd#1024:\
           :pe#0:be#8192:fe#1024:\
           :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 copies 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.



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 cop 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.

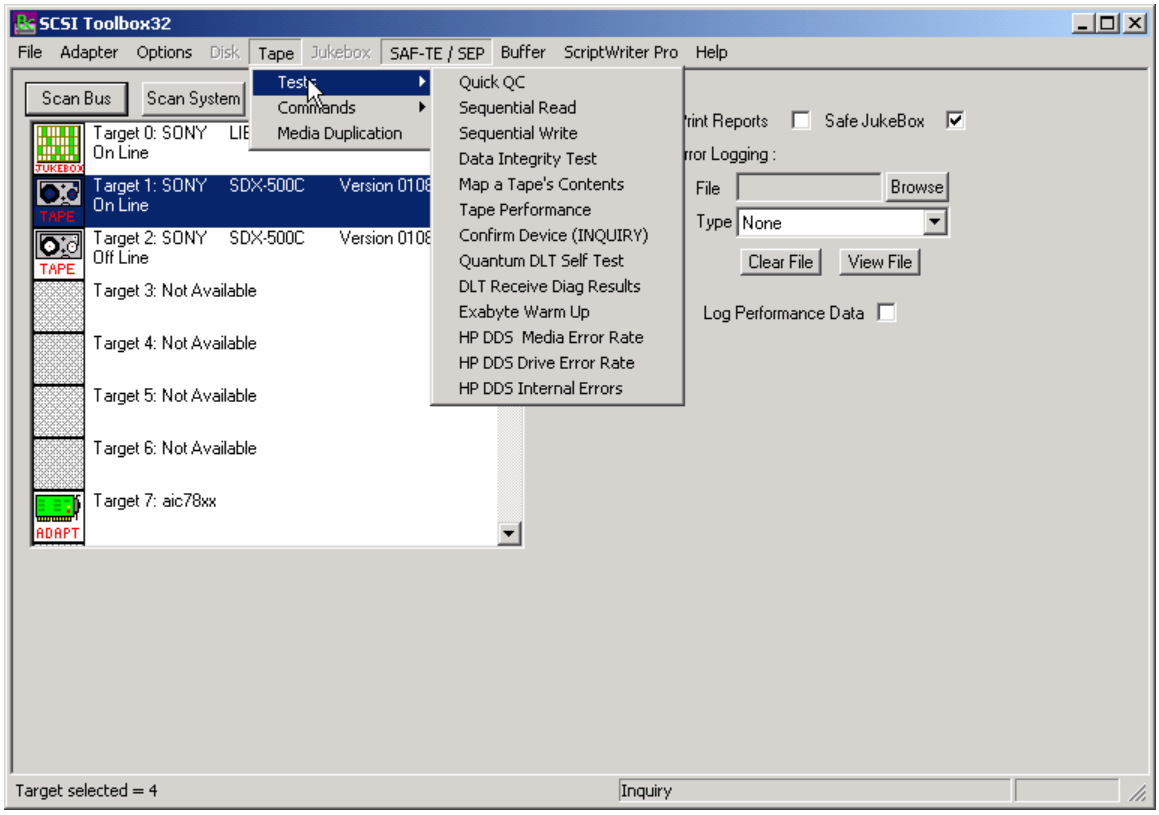
Choosing the Copy Range
In the case of a disk-to-disk copy, you may specify the range of blocks that are copied. The starting block will be defaulted to block zero, and the ending block will default to the highest block number of the source disk. You may enter any other value into the starting and ending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

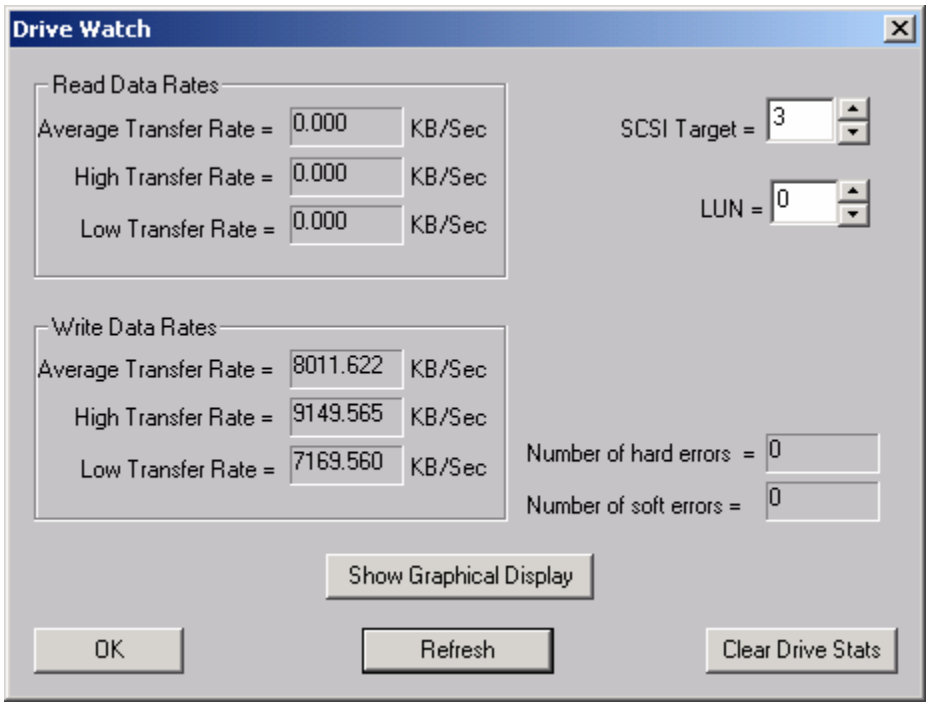
Ignore Errors
Selecting this check box will cause the copy operation to continue on a device after an error has occurred. You may want to use this to skip over bad blocks on a device.

Tape Tests

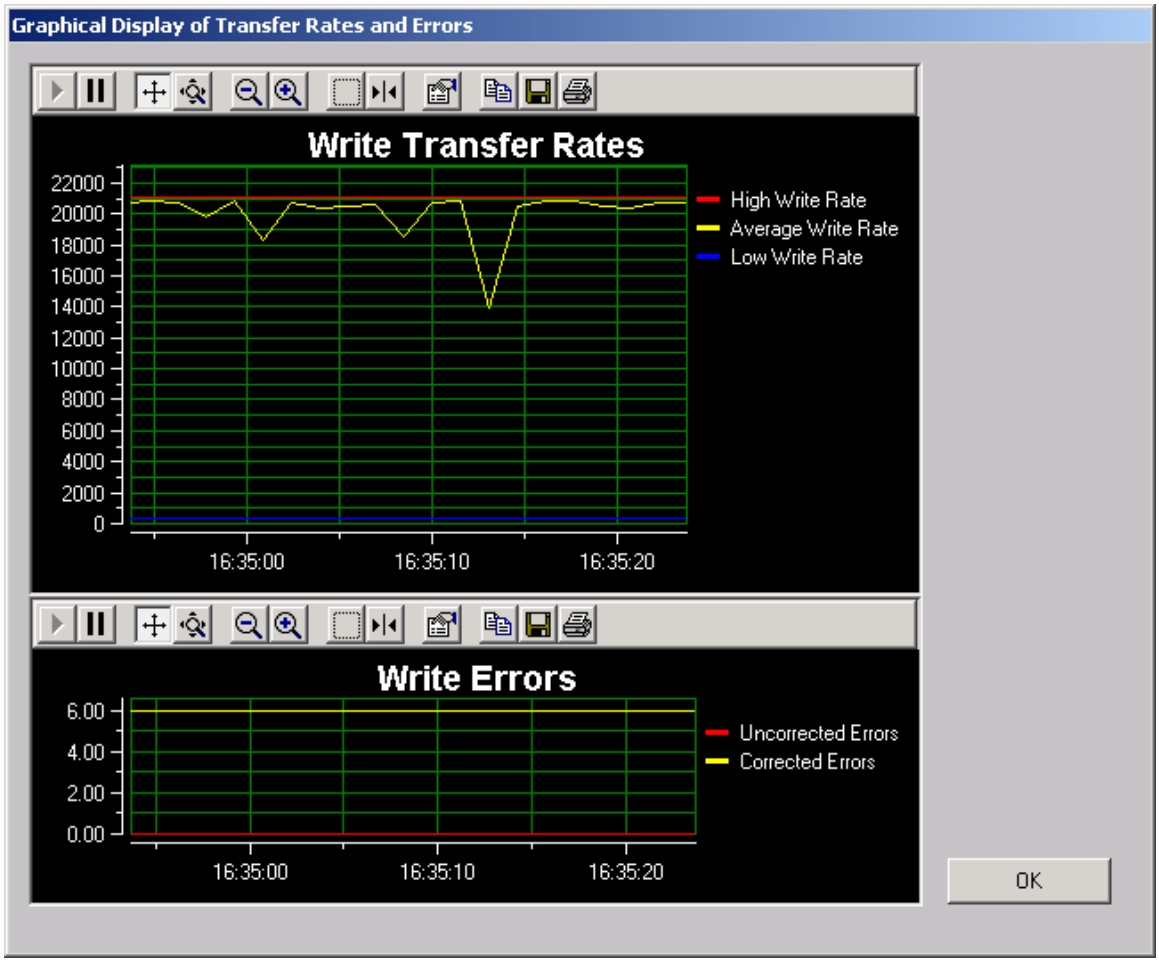
The following tests are available via the Tape Test menu



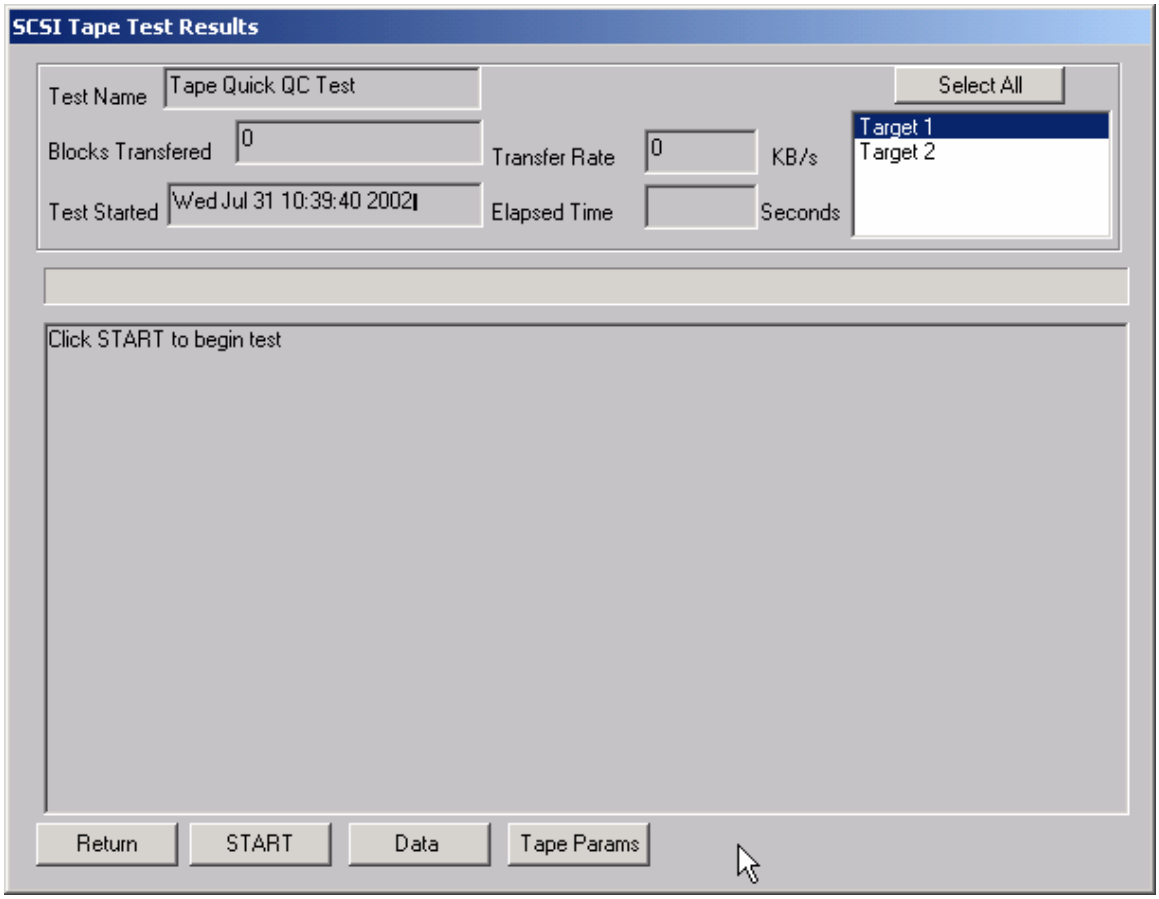
TapeWatch



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:



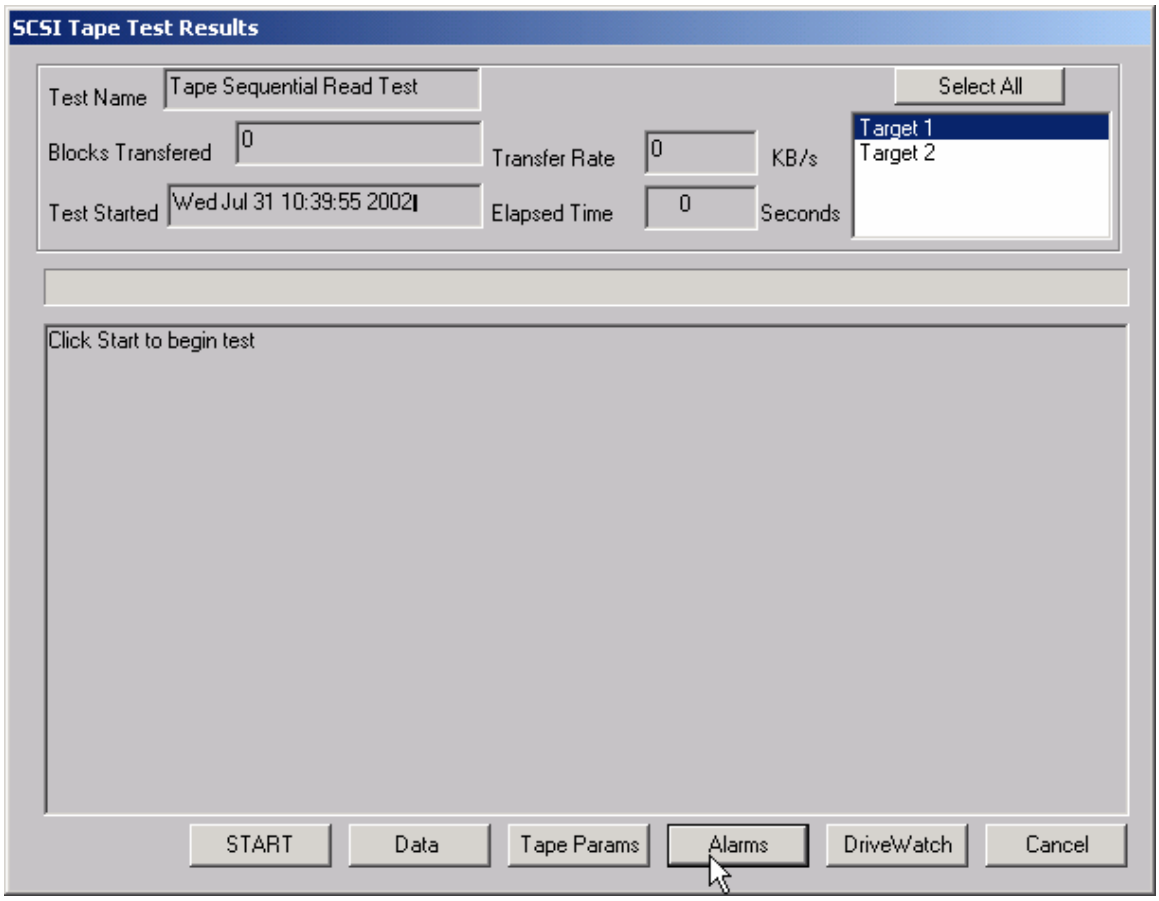
Quick QC Test



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

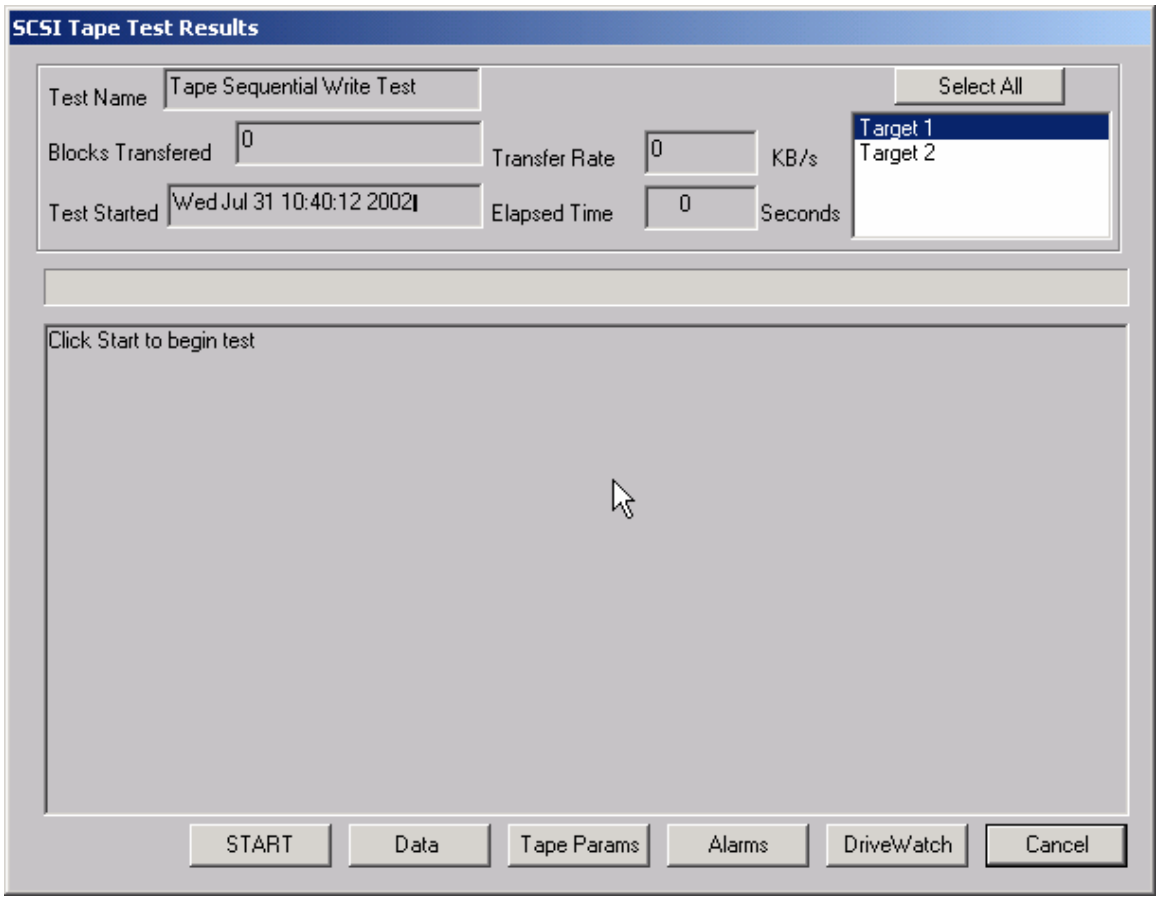
-Rewind tape
-Write blocks of data to tape
-Write a File Mark
-Rewind tape
-Read blocks of data from tape
-Space Reverse to beginning of tape
-Space Forward to File Mark
-Rewind tape

Sequential Read Test



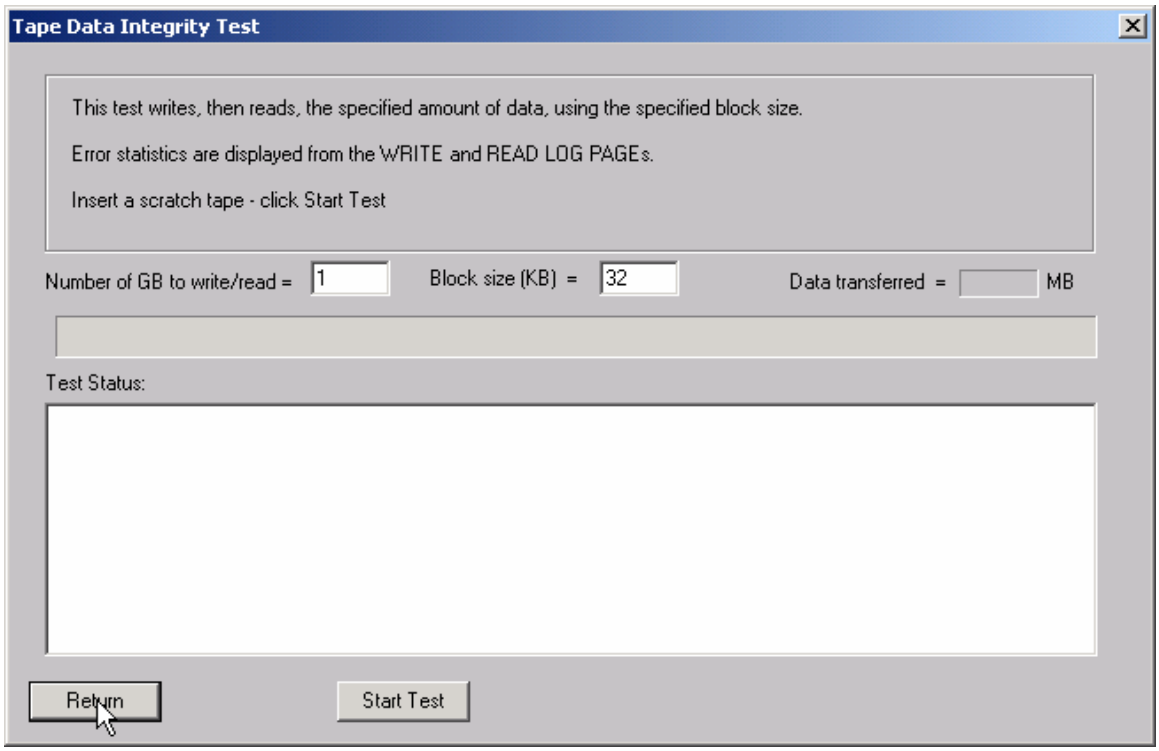
This test asks if you want to rewind the tape or not then sequentially reads every block on the tape, until either a Blank Check (End of Data) or End of Media is encountered. In the case of tape drives that can report remaining tape and error information, this information is displayed and updated every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

Sequential Write Test



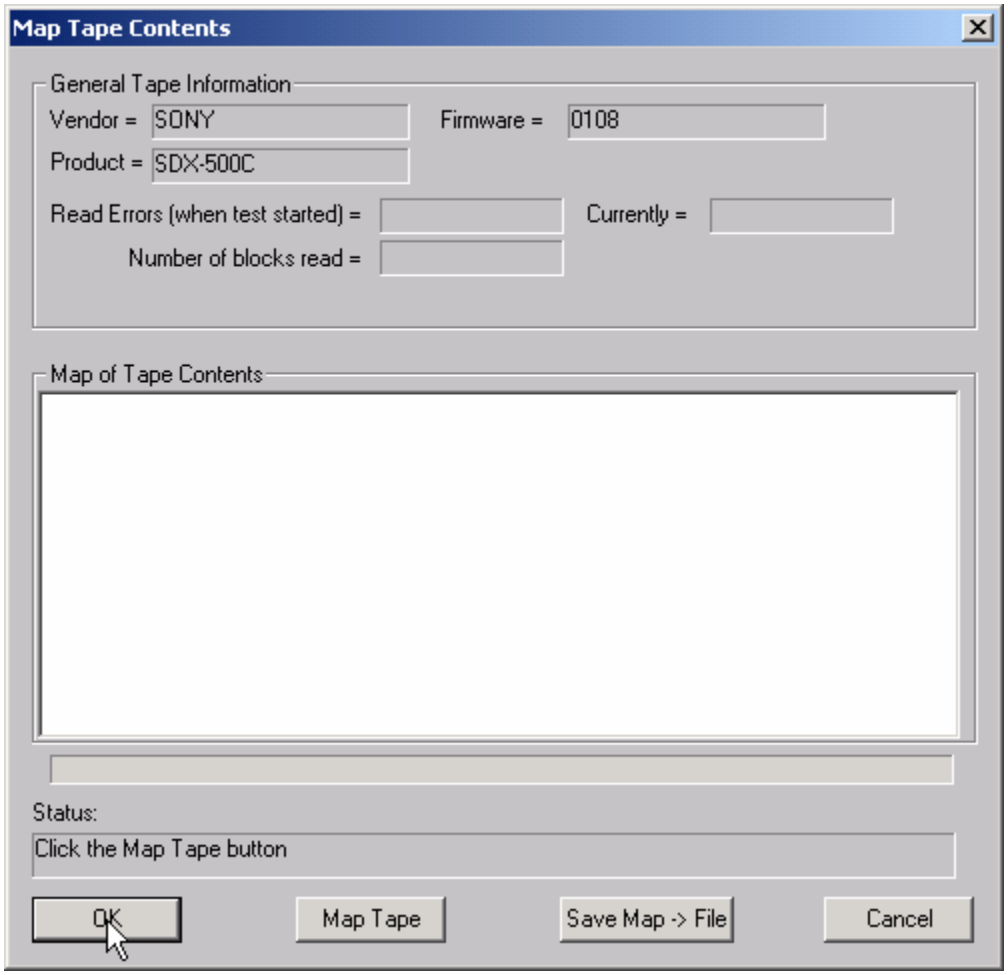
The Sequential Write Test asks if you want to rewind the tape or not then writes to every block on the tape media until End of Media is encountered. In the case of Exabyte 8mm drives and most 4mm DAT drives, the remaining tape and error statistics are displayed on the screen after each one megabyte of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

Data Integrity Test



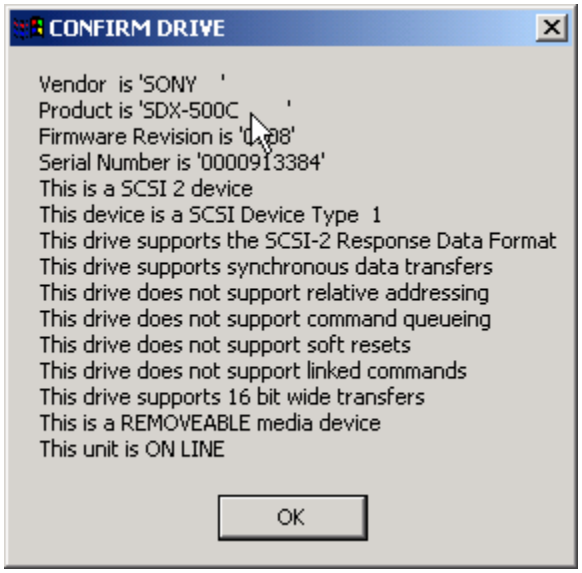
This test will rewind the tape, then write the specified amount of data to the selected drive. At the end of the write phase the error information collected from the drive's log pages will be displayed.

Map a Tape's Contents



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)

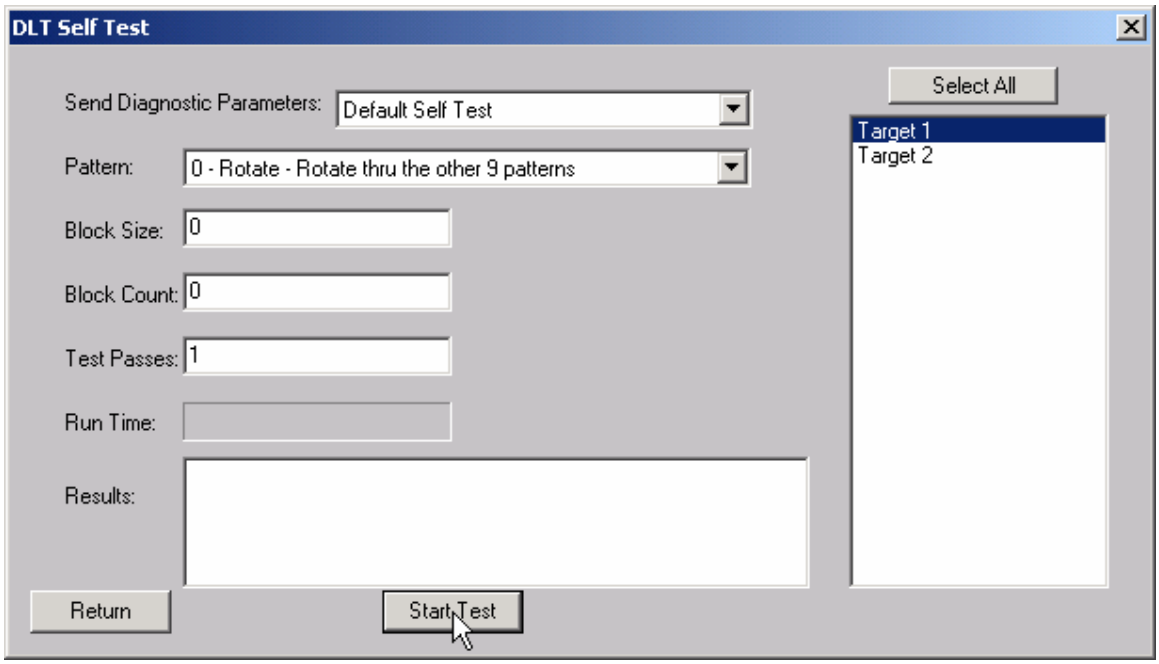


This test will report as much information as is available about the currently selected tape drive. In all cases, the VENDOR, PRODUCT,
and REVISION fields returned by a SCSI INQUIRY command will be displayed, along with the ANSI SCSI level of the device, and the device's ability to support any of the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

- Synchronous data transfers
- Relative addressing
- Command queuing
- Soft resets
- Linked commands, and
- Whether the device is a removable media type or not

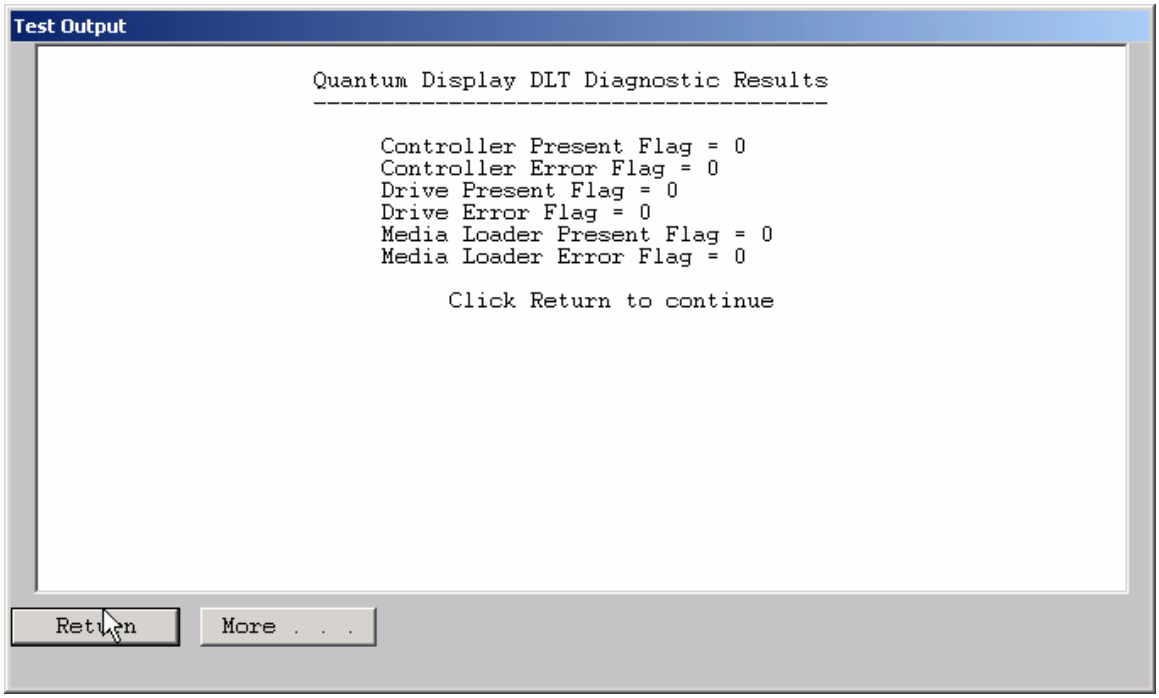
Some devices will report back with much more information than indicated above. In these cases the information is often in a vendor-unique format, which the SCSI *toolbox* will interpret for you. If additional data beyond that described above is available that data will displayed.

Quantum DLT Self Test



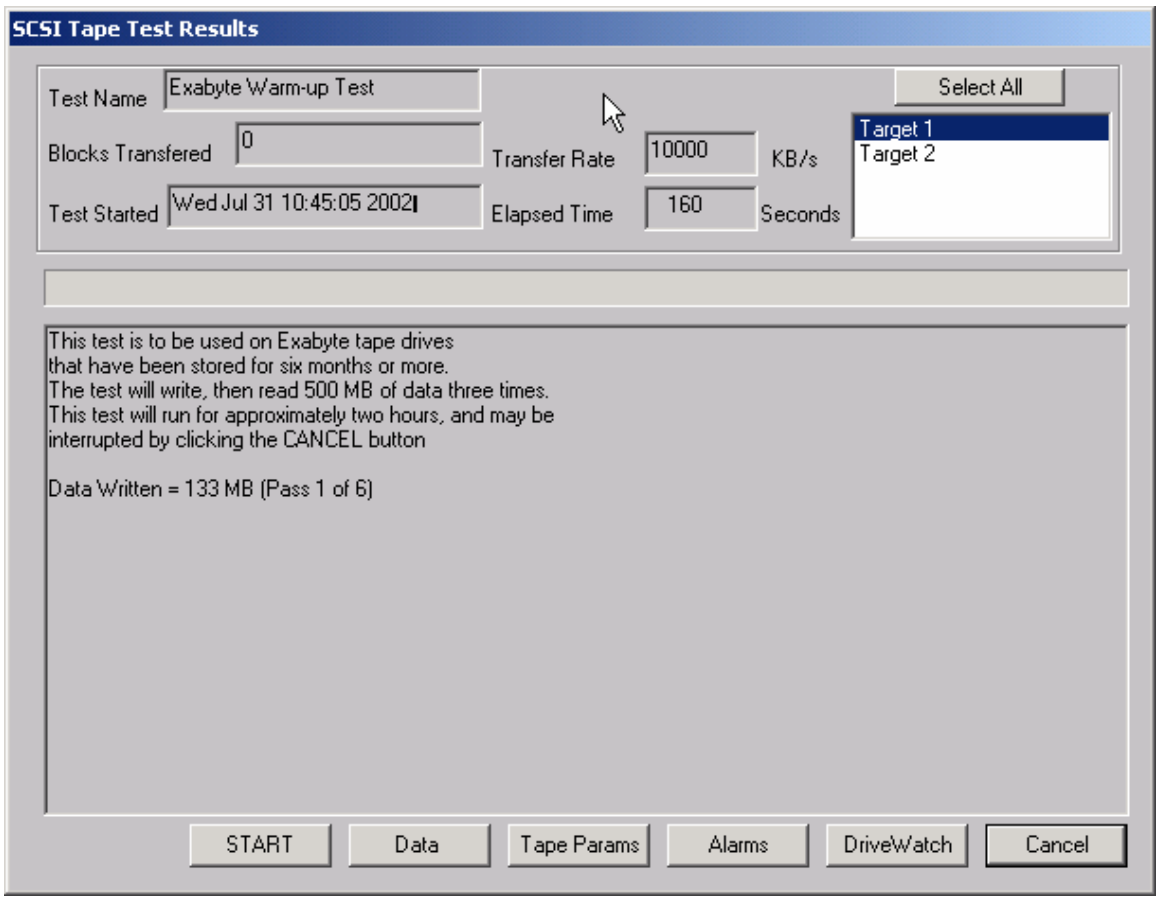
This test allows you to set all the user defined parameters, or use the default self test. This test supports disconnect-reconnect with Adaptec or Q-Logic SCSI adapters in the multi-drive mode.

DLT Receive Diagnostic Results



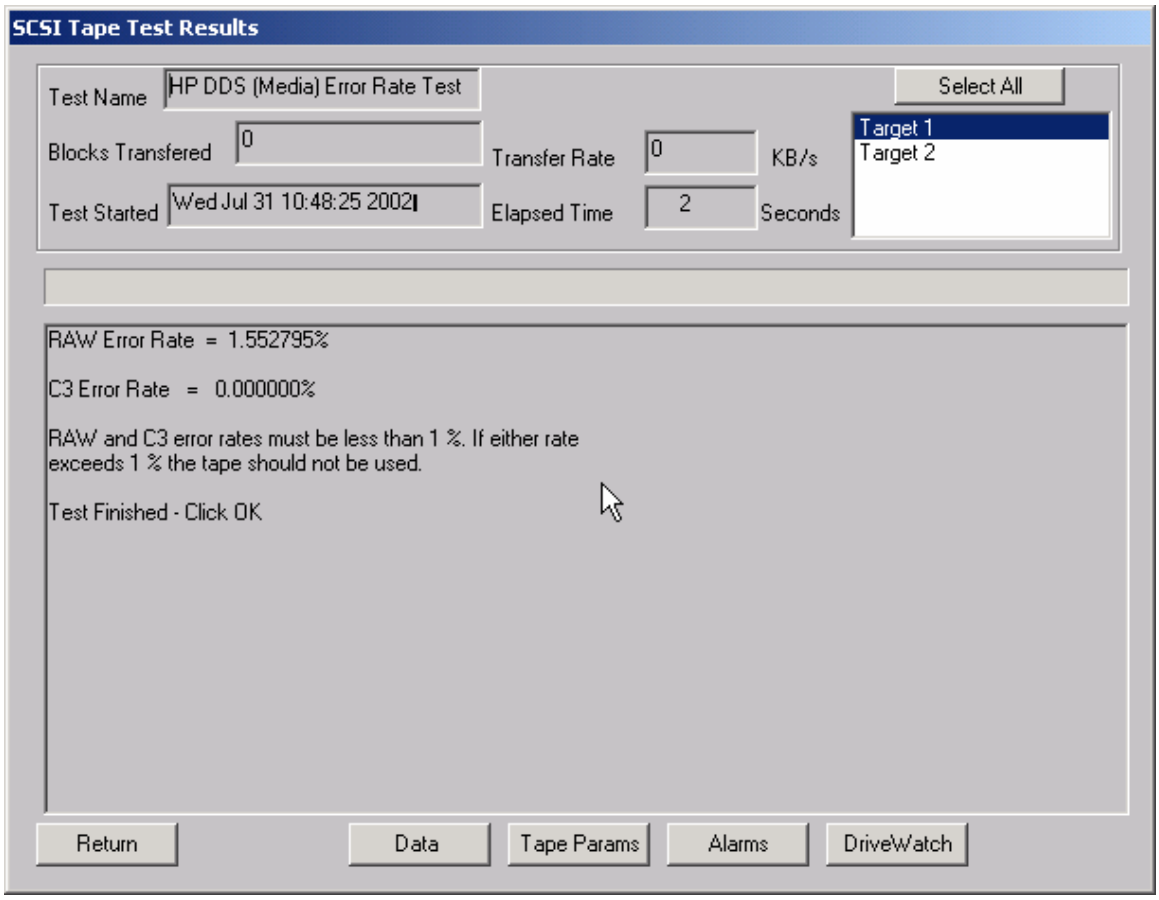
DLT Receive Diagnostic Results
This function displays the results of a recv diagnostic command for the controller, drive and medium on any DLT-type device.

Exabyte Warm Up Test



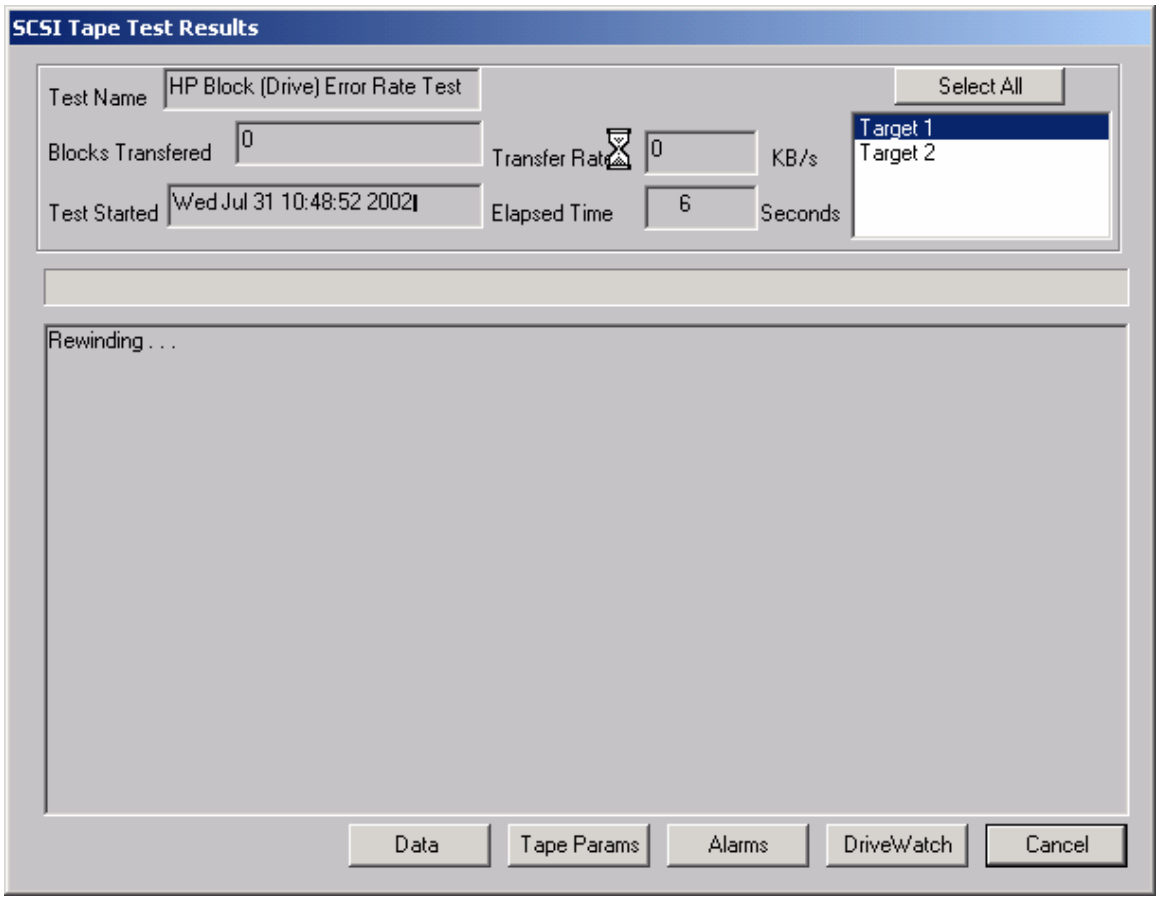
This test is specific to Exabyte 8mm tape drives. If the drive is inactive for long periods of time then is used it may fail. This test simply moves the tape forward and back writes data etc to break in a drive that has been inactive. This helps get all the moving parts moving smoothly again and ready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

DDS (4mm) Media Error Rate Test



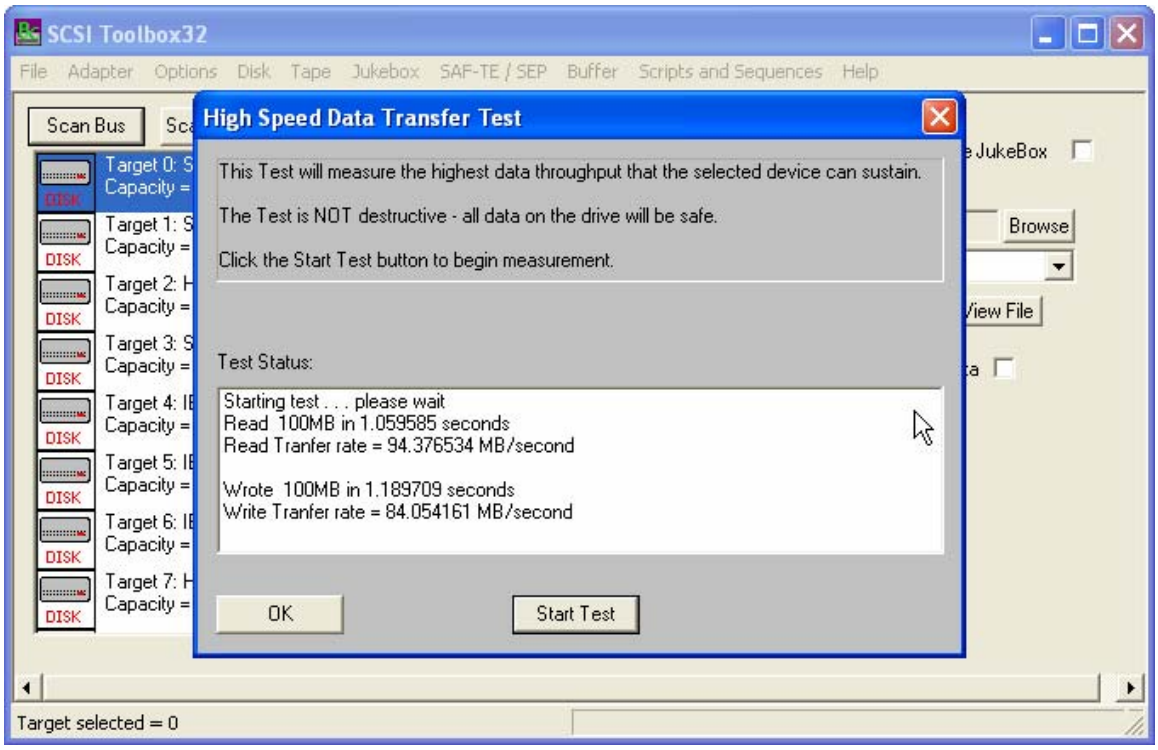
This test reads information from the header of a DDS (4mm) tape and interprets it into tape error statistics. This information is obtained by reading the TAPE LOG PAGE of the LOG SENSE command.

DDS (4mm) Drive Error Rate Test



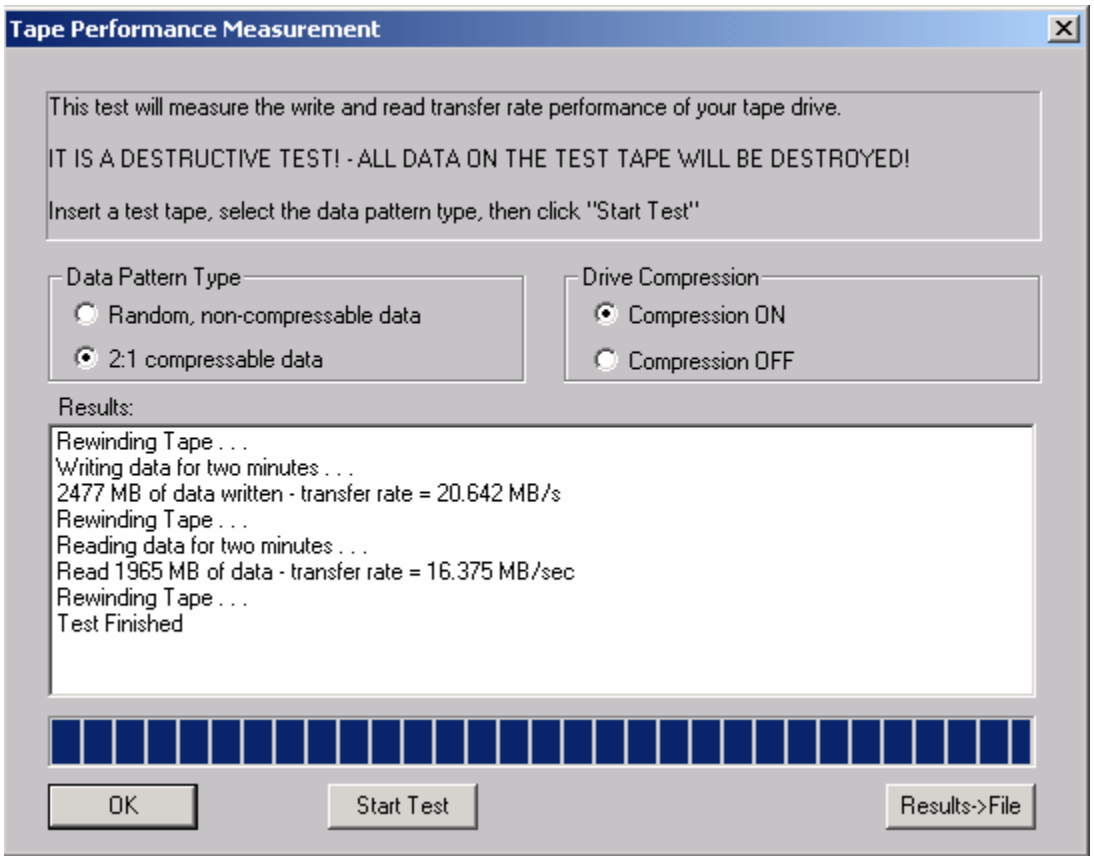
This test will space to the End of Data (EOD) on the tape, then will write 50 MB of data. The Block Error Rates from LOG SENSE PAGES 34 and 35 will then be read and interpreted. This test will show you the error rate of the tape drive itself, where the HP Media Error Rate test will show the error rate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



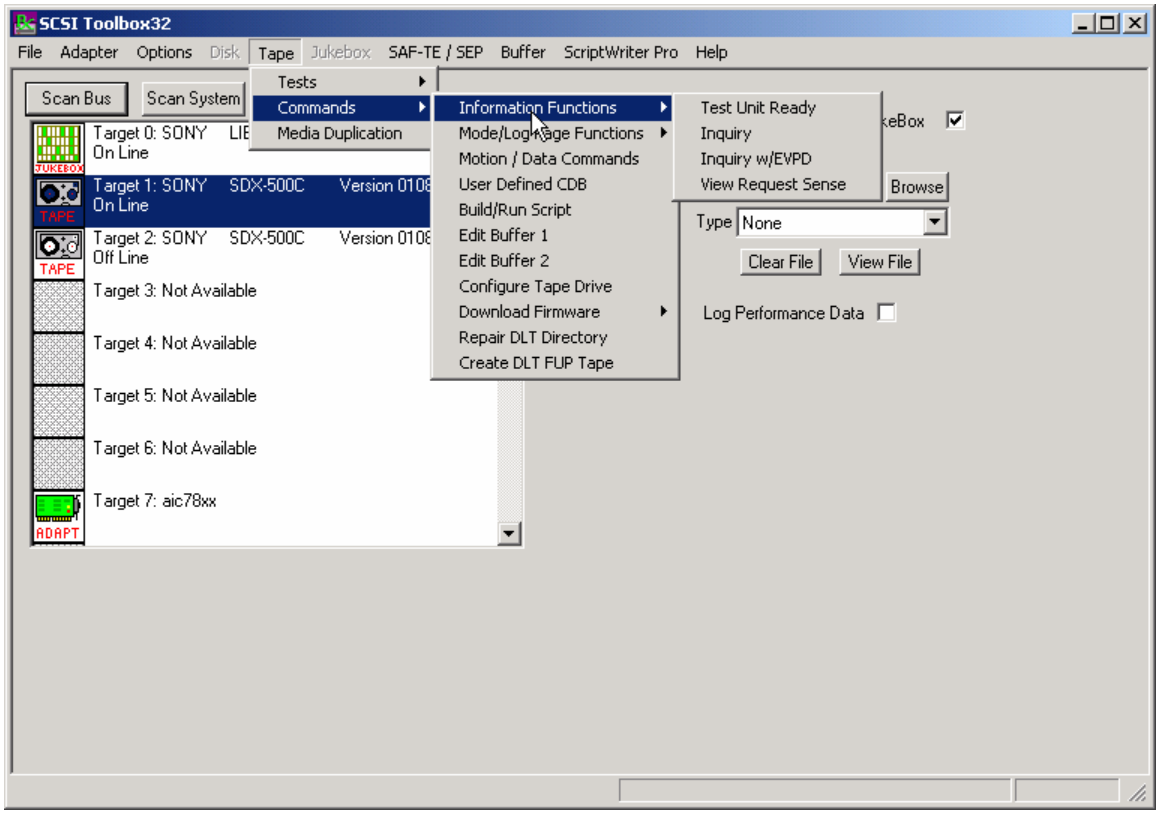
The Tape Performance Measurement test transfers data to and from the drive for two minutes, measuring the amount of data that can be transferred in that time to determine the “real –world” data throughput of the drive.

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

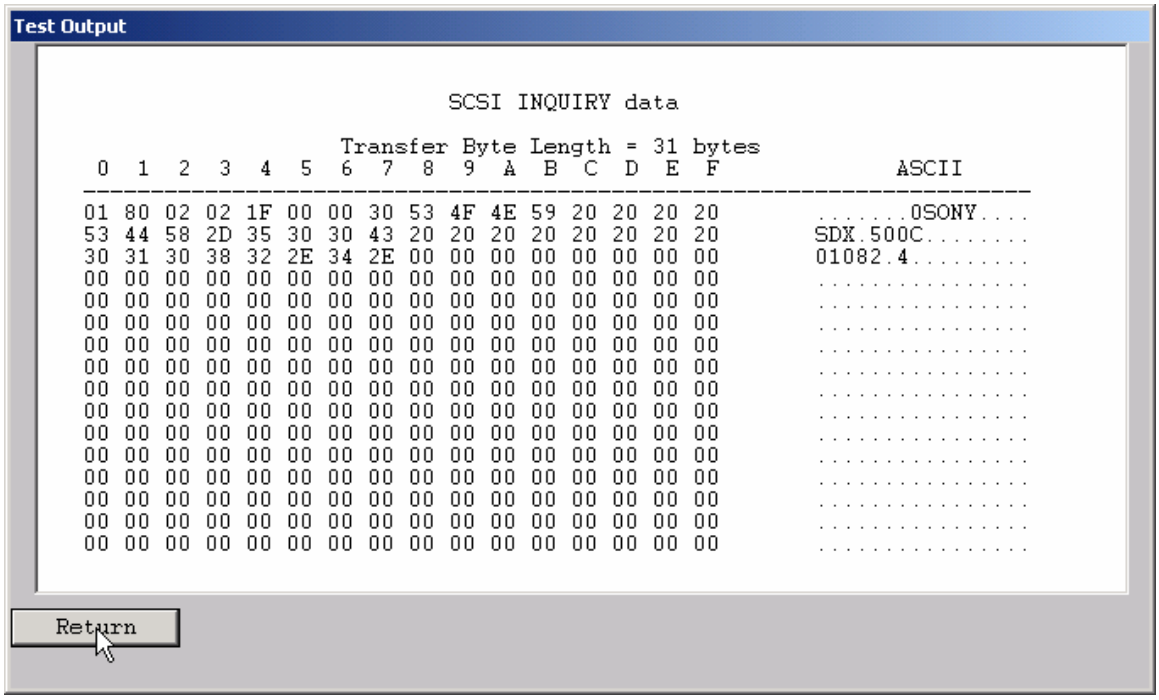
The following commands are available via the Tape Commands menu:



Information Functions – Test Unit Ready

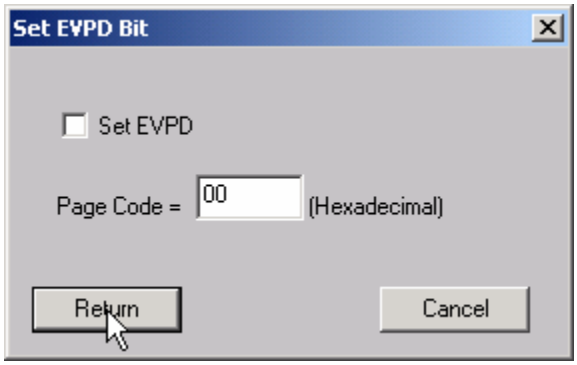
This command displays the return value from a SCSI Test Unit Ready command. A value of zero indicates a successful command completed.

Inquiry



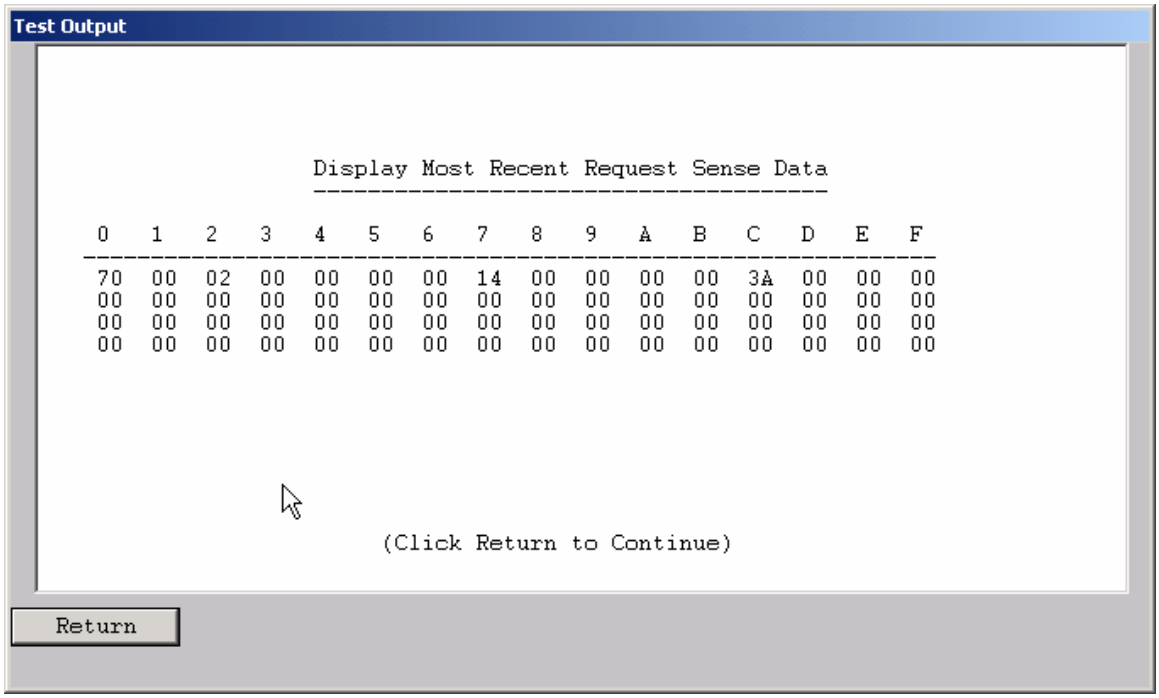
This command displays the data returned from the SCSI INQUIRY command. The data is shown in hexadecimal format on the left of the screen and in ASCII format on the right side of the screen.

Inquiry with EVPD



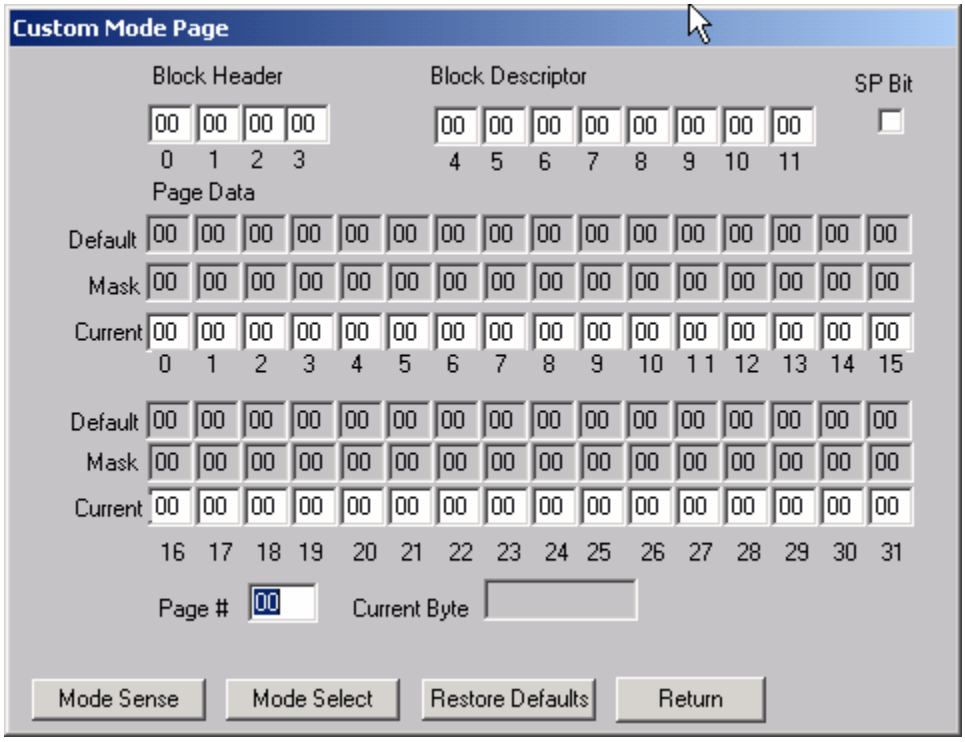
This option lets you specifiy whether to set the EVPD bit in the INQUIRY command, and which EVPD page you would like to display.

View most recent Sense Data



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



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

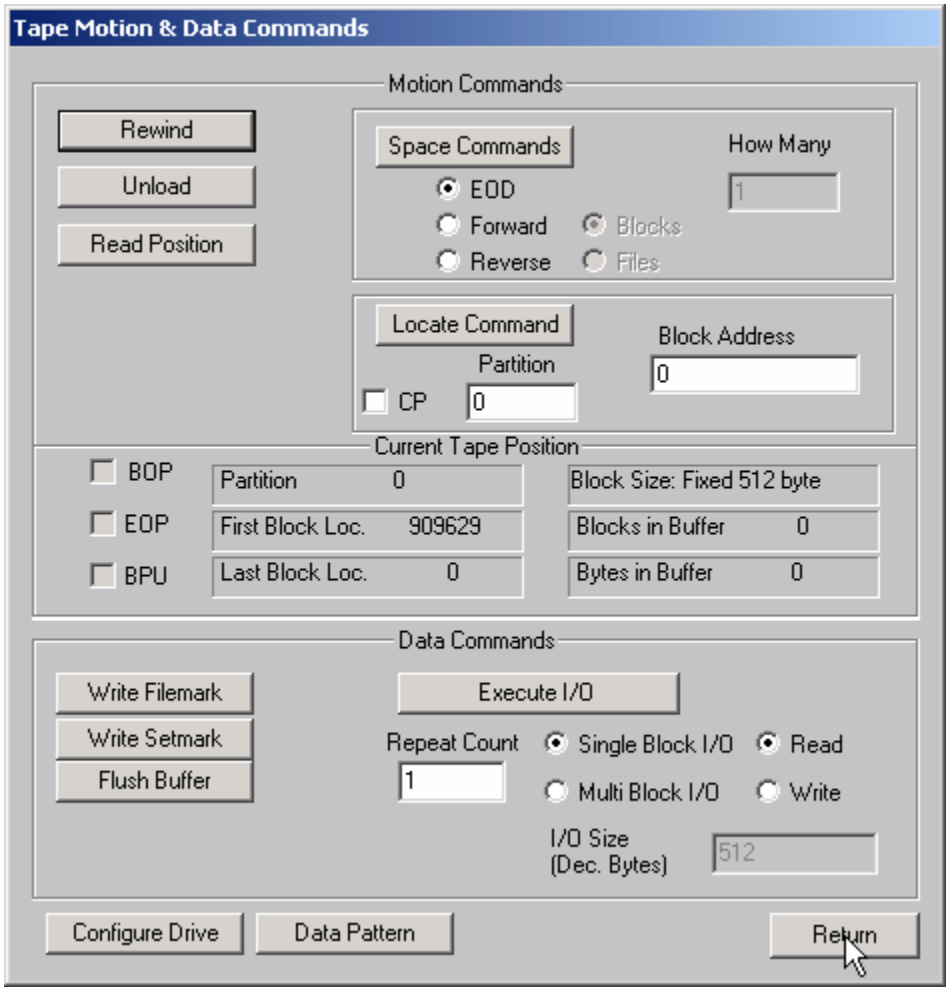
DDS Log Pages

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



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
  - 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
  - This command allows you to space forware 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
  - This command spaceds 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 wit 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.

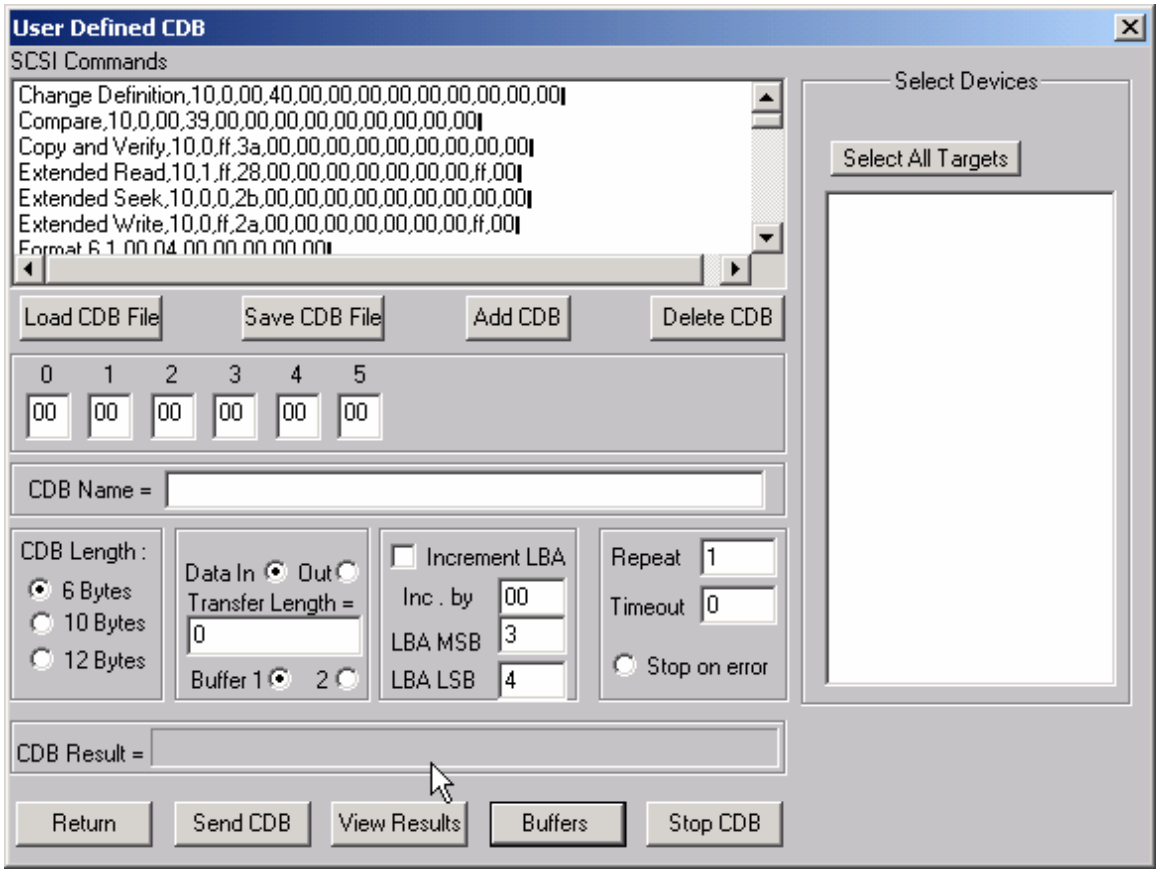
read block(s)

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 User-Defined CDB functions of the SCSItoolbox 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.

Definining 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:

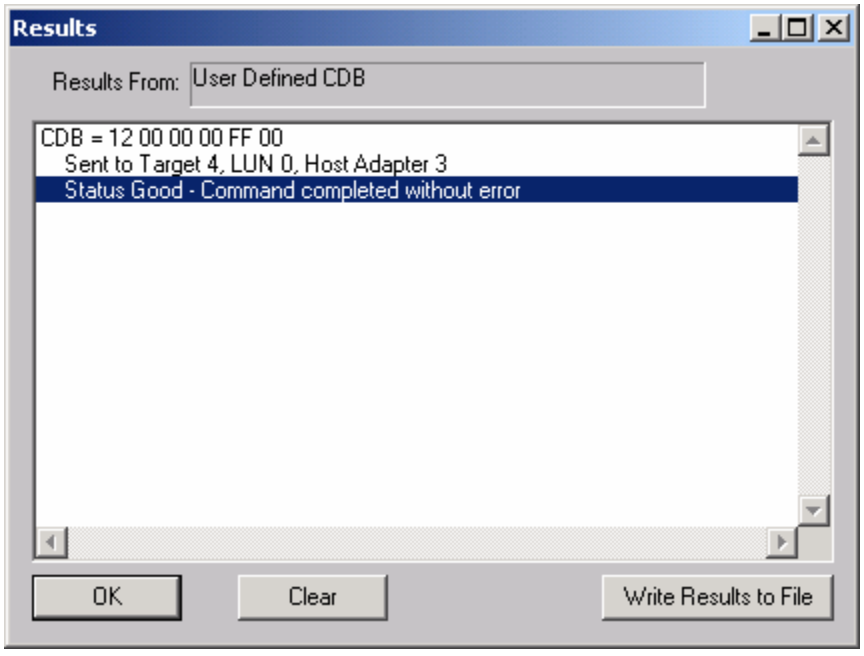
- Data direction incorrect
- Data length specified in CDB does not match Data Transfer Length

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:

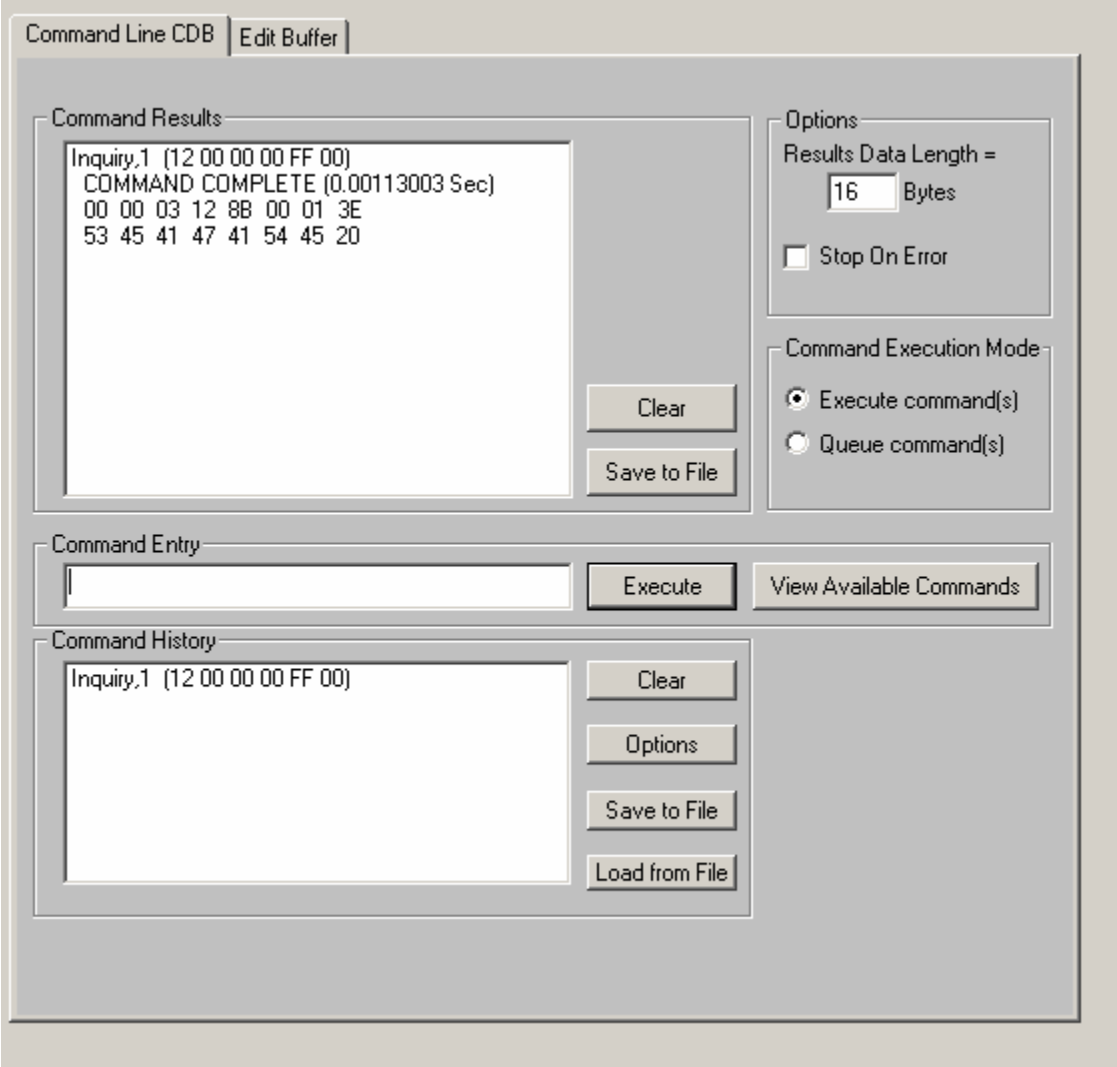
- Check the Increment LBA box
- Specify which CDB byte is the LSB of the SCSI CDB
- Specify whick CDB byte is the MSB
- Specify how much you want to increment by

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

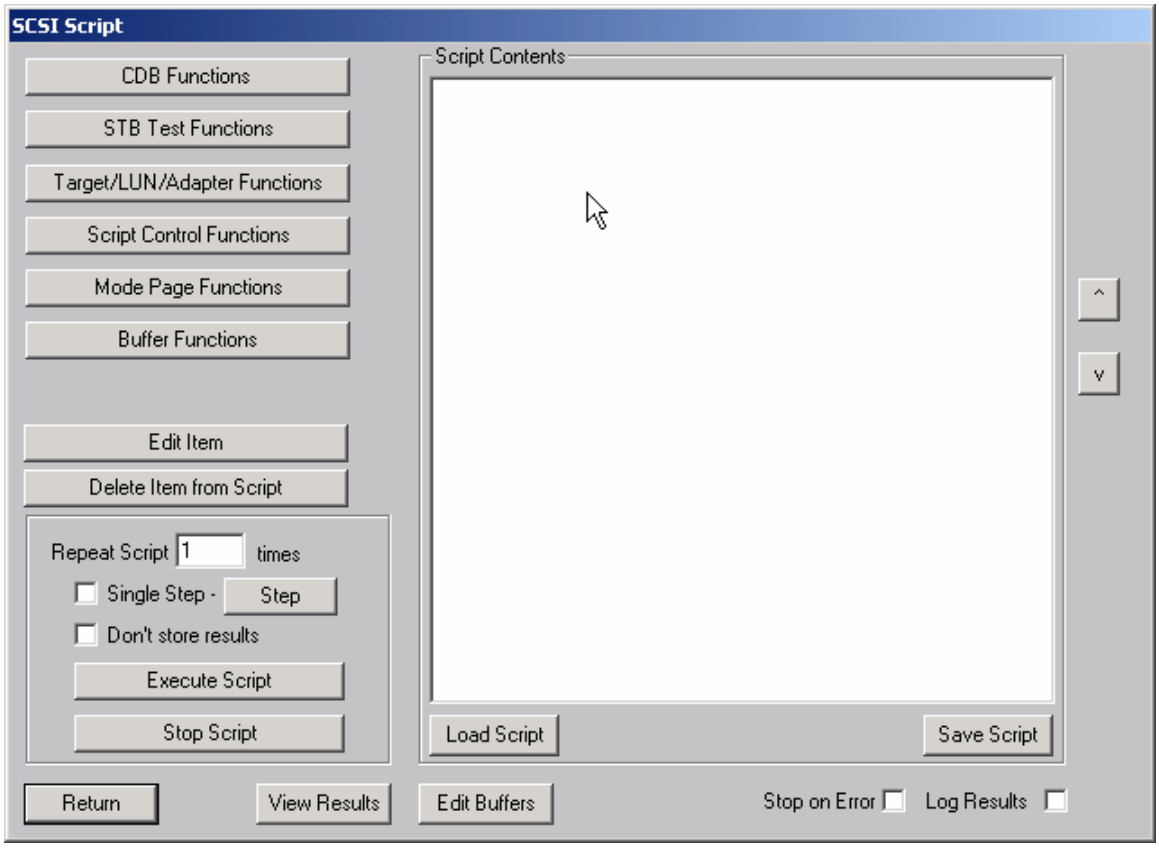
The “Results Data Length” allows you to specify how many bytes of data are displayed in the “Command Results” window. If a command transfers data, this many bytes will be displayed along with the command results. If the command terminates in a check condition, this many bytes of sense data will be displayed.

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

Re-issuing commands

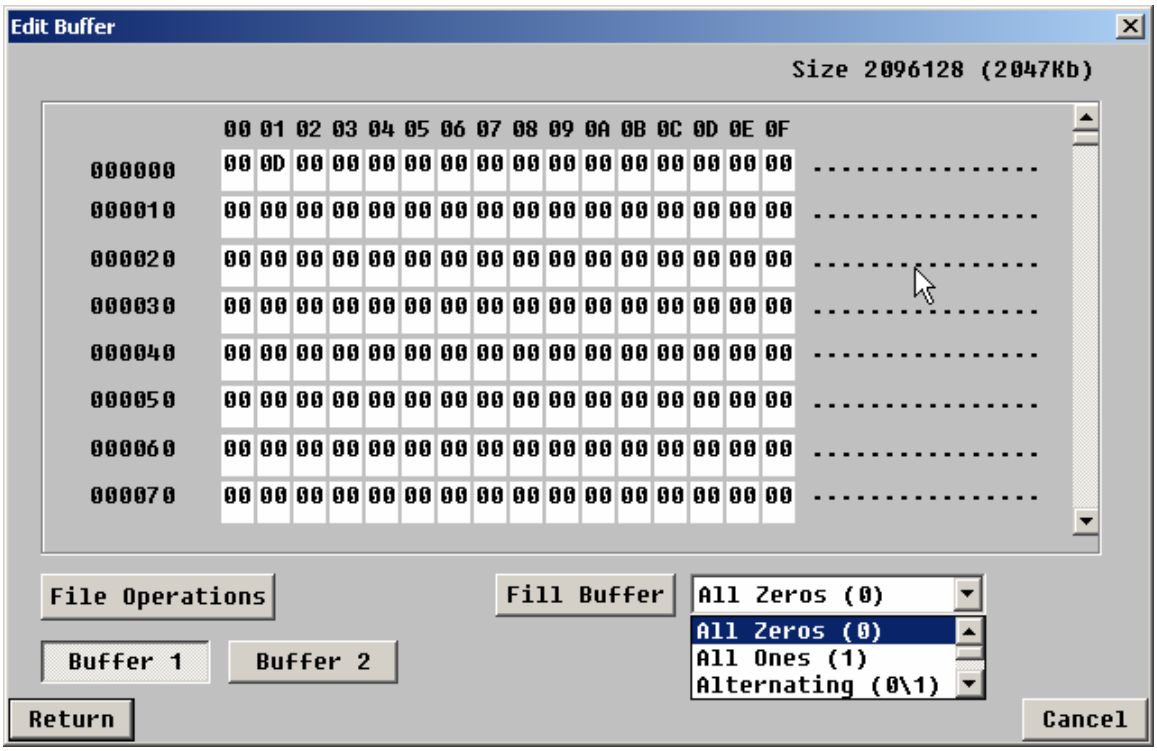
Any command or commands can be re-issued by selecting them (clicking on them) in the “Command History” window. Once the commands you want to reissue are highlighted, click the “Execute” button and they will be reissued.

Build/Run Script

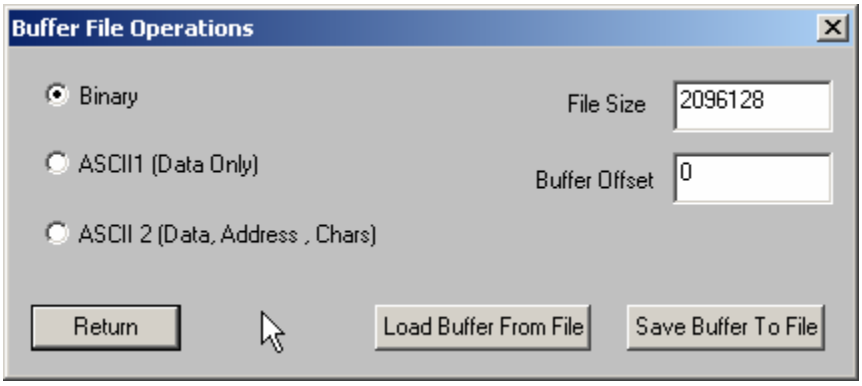


Edit Buffer 1/ Edit Buffer 2

The buffer functions allow the buffers used by the SCSItoolbox32 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.

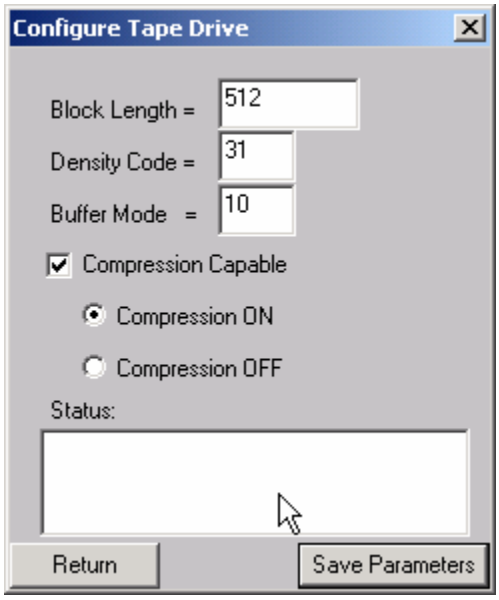


The drop box next to the Fill Buffer button allows you to choose a pattern to load into the buffer. Select the data pattern, then click the Fill Buffer button to load the data into the buffer. The File Operations button allows the buffer date to be saved to or loaded to a disk file.



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



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
  - 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
  - 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
  - Assigns the value to parameter list header byte 2 to set buffer mode on or off

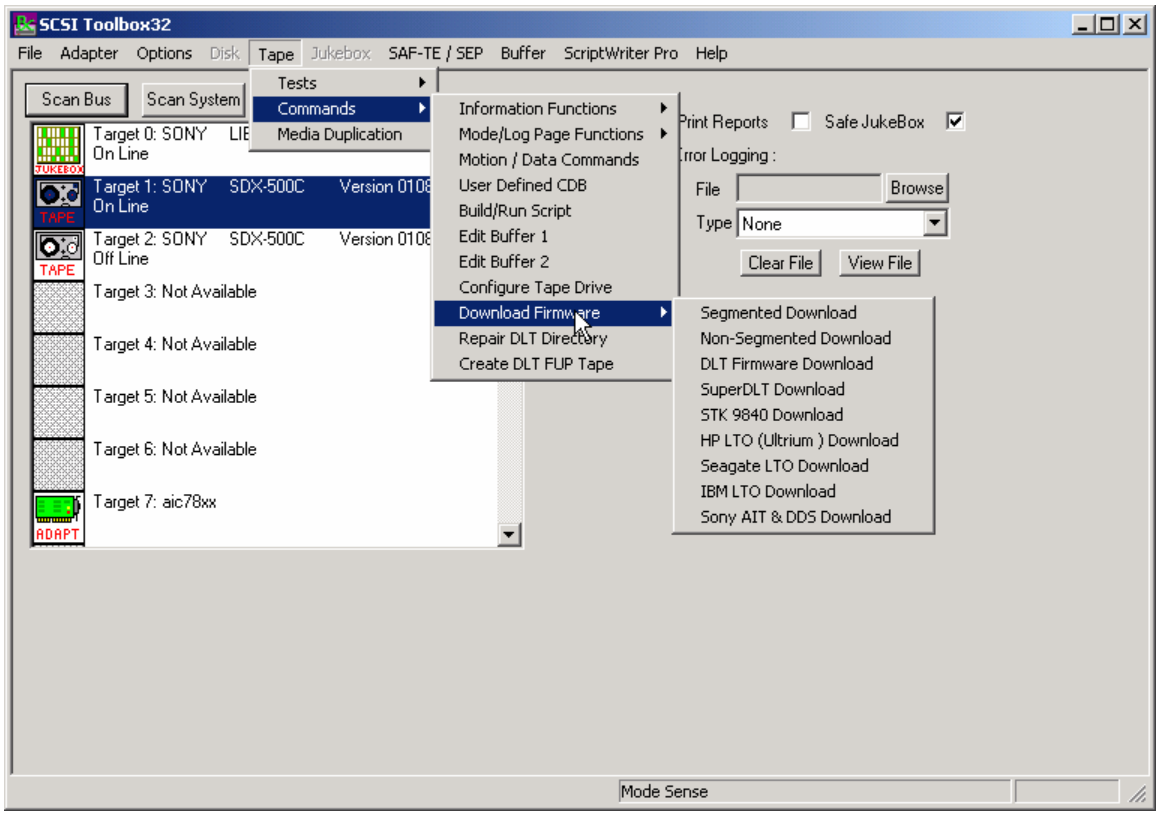
If the drive supports compression, the Compression Capable check box will be check and the user can turn compression on or off with the radio button below it.

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.

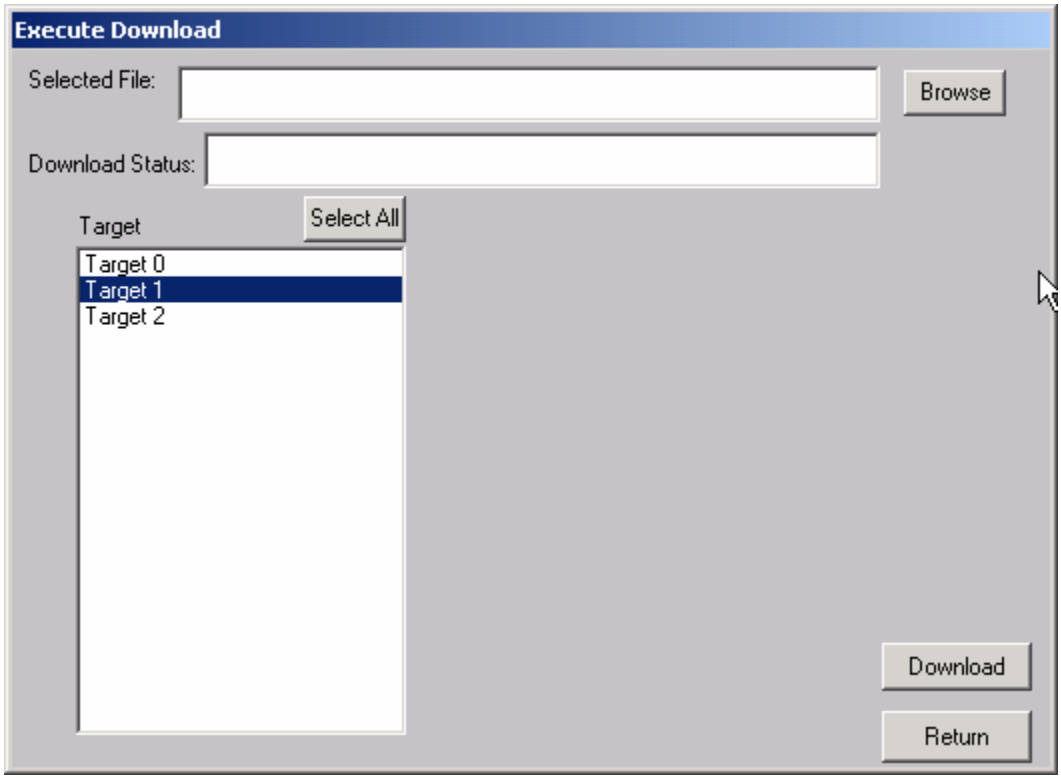
Note: Not all tape drives support all of these settings. The user must be familiar with the drive and have a SCSI manual available to determine support and valid values for these parameters.

Download Firmware

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



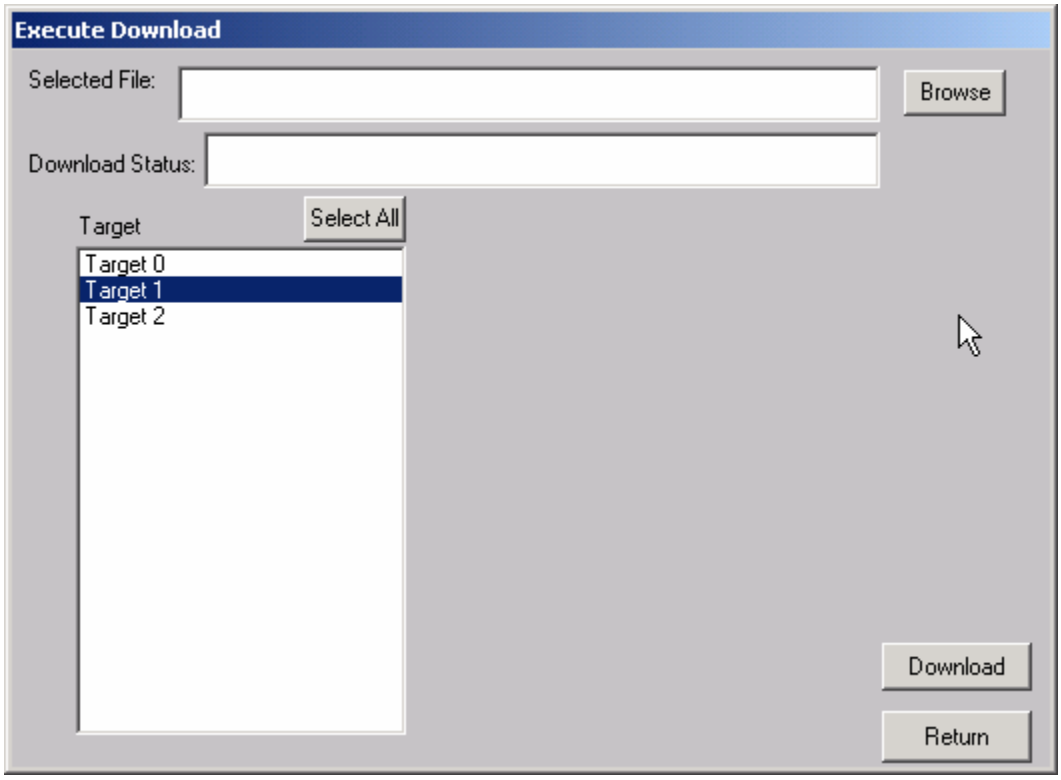
Segmented Download



The 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.

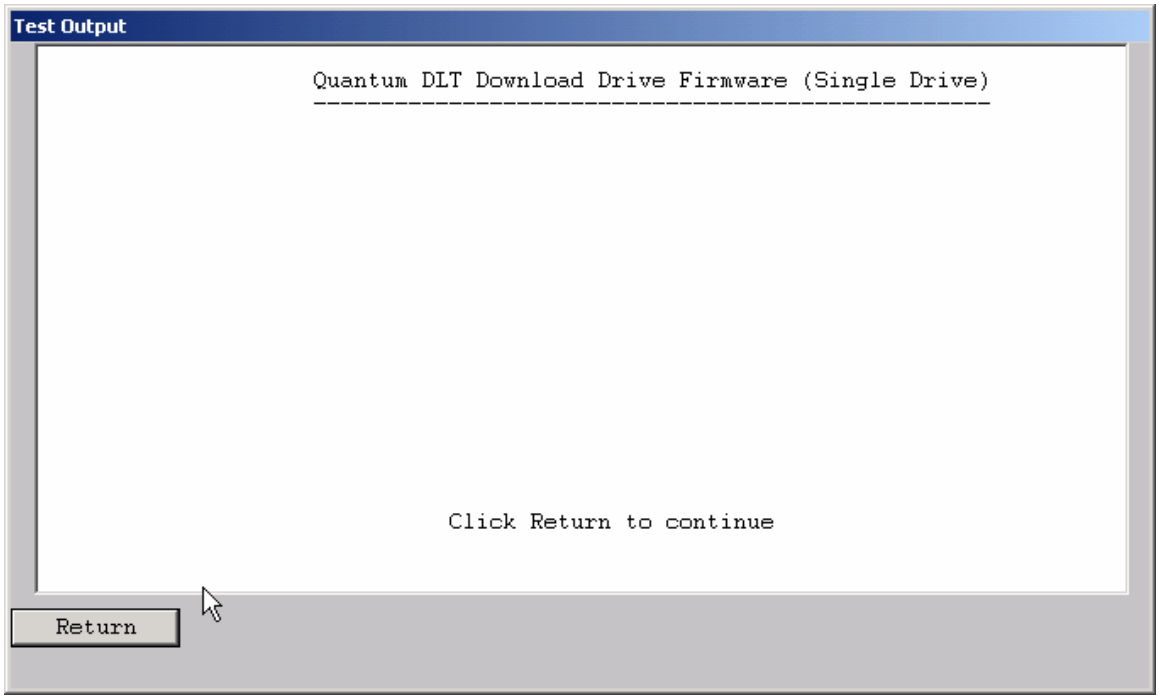
Non-Segmented Download



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.

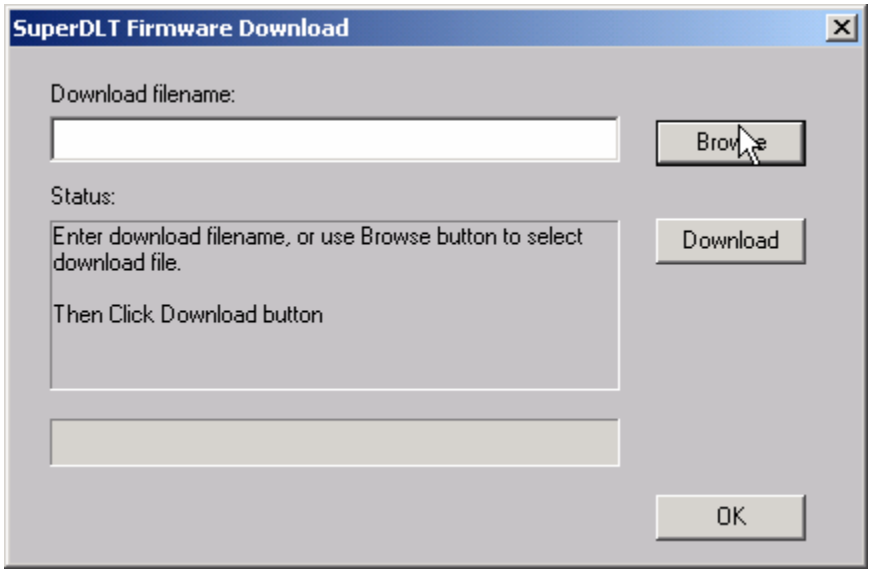
If the available buffer space is not large enough to hold the file the download will fail. The "About SCSI toolbox" choice of the "Help" menu displays the available buffer space.

DLT Firmware Download



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

SuperDLT Download

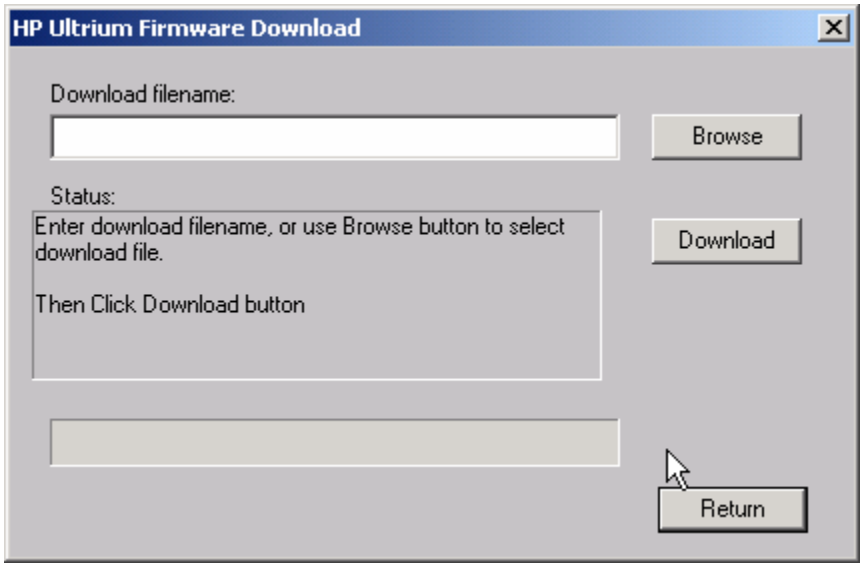


This command will download new firmware into a SuperDLT drive. Use the Browse button to select the download file, then click the Download button.

STK 9840 Download

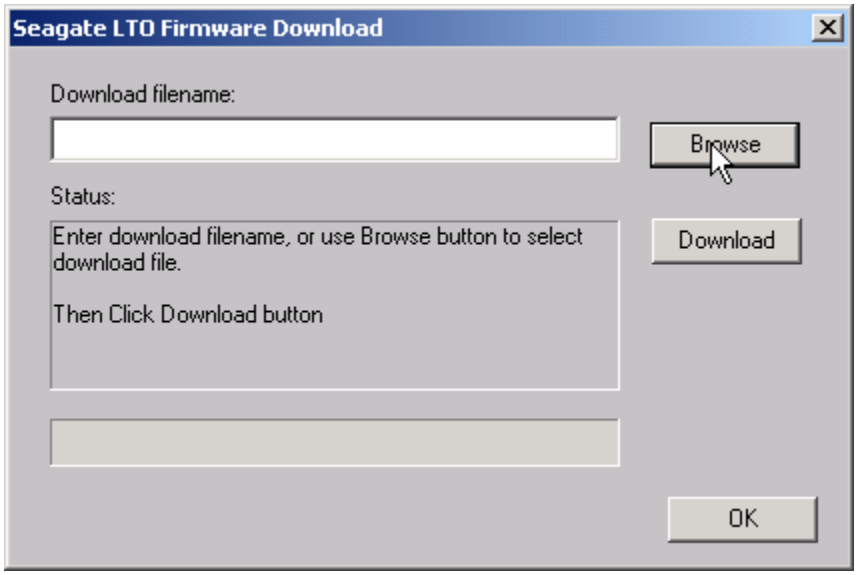
This command will download new firmware into a STK 9840/9940 drive. Use the Browse button to select the download file, then click the Download button.

HP LTO Download



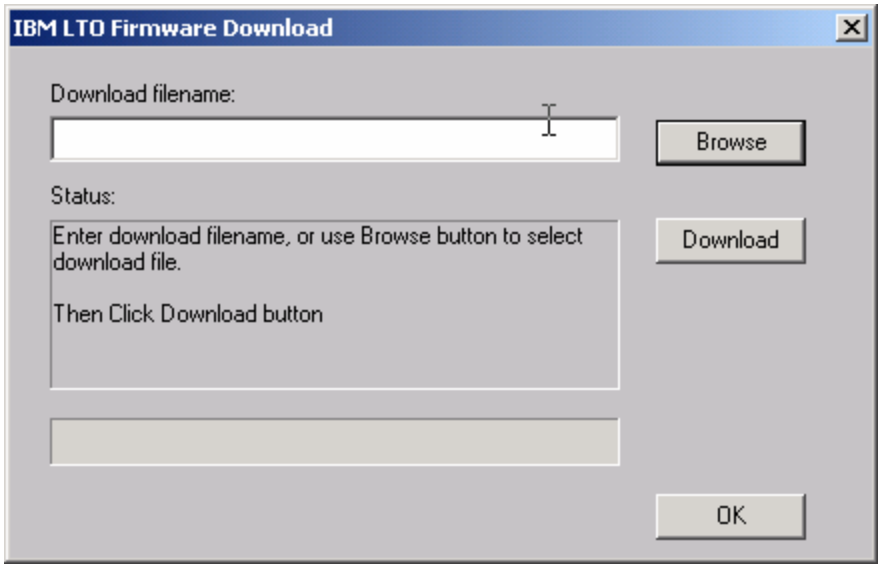
This command will download new firmware into a HP LTO (Ultrium) drive. Use the Browse button to select the download file, then click the Download button.

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

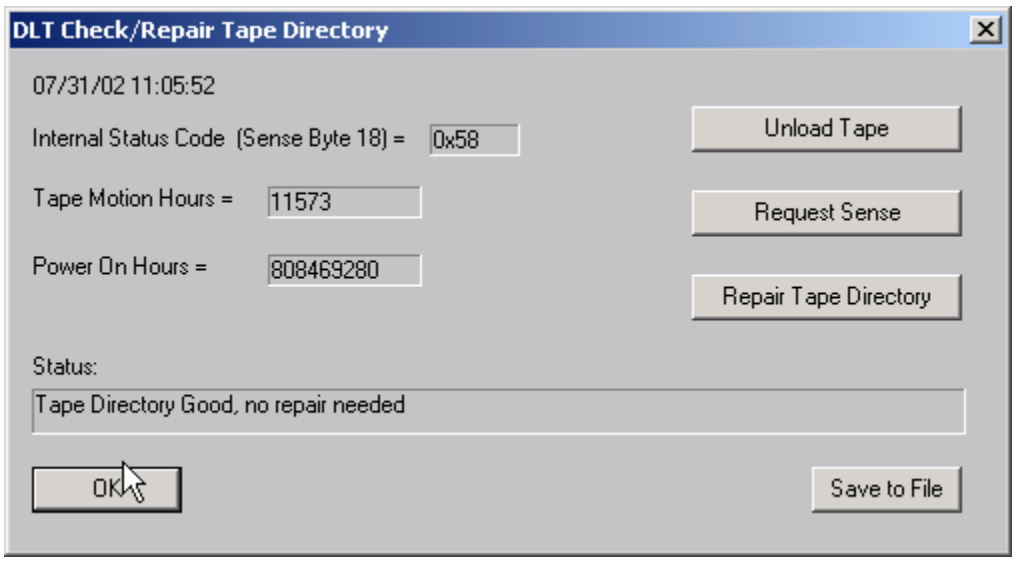


This command will download new firmware into a IBM LTO drive. Use the Browse button to select the download file, then click the Download button.

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.

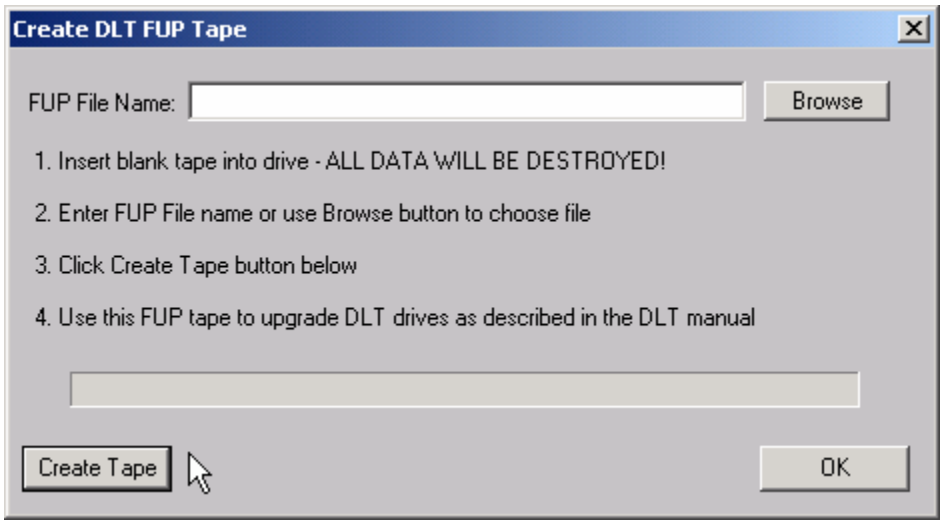
Repair DLT Tape Directory



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.

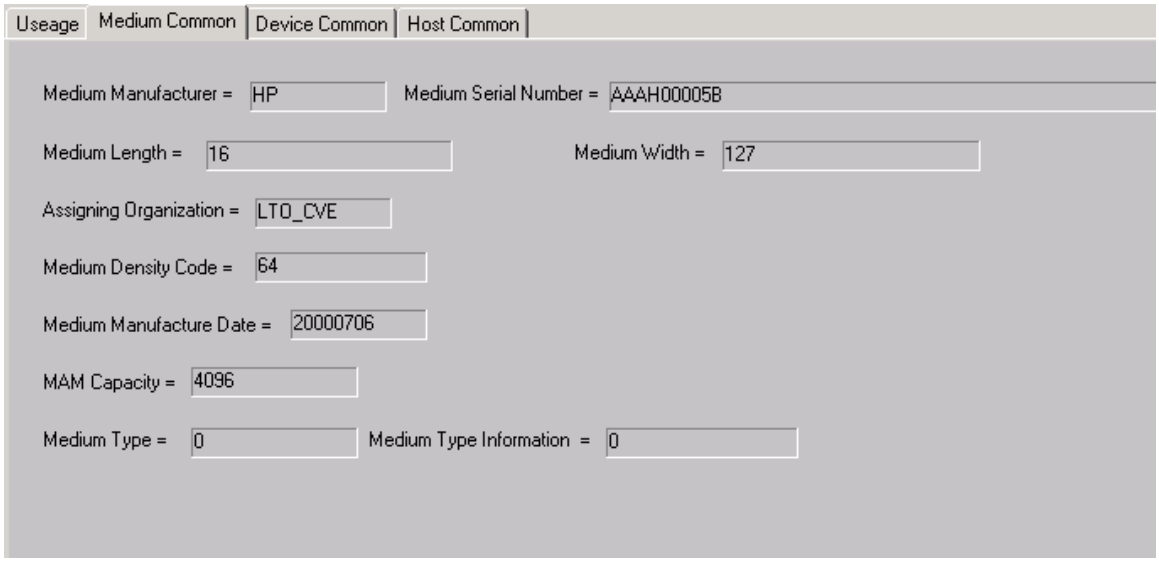
Create DLT FUP (Firmware Update) Tape

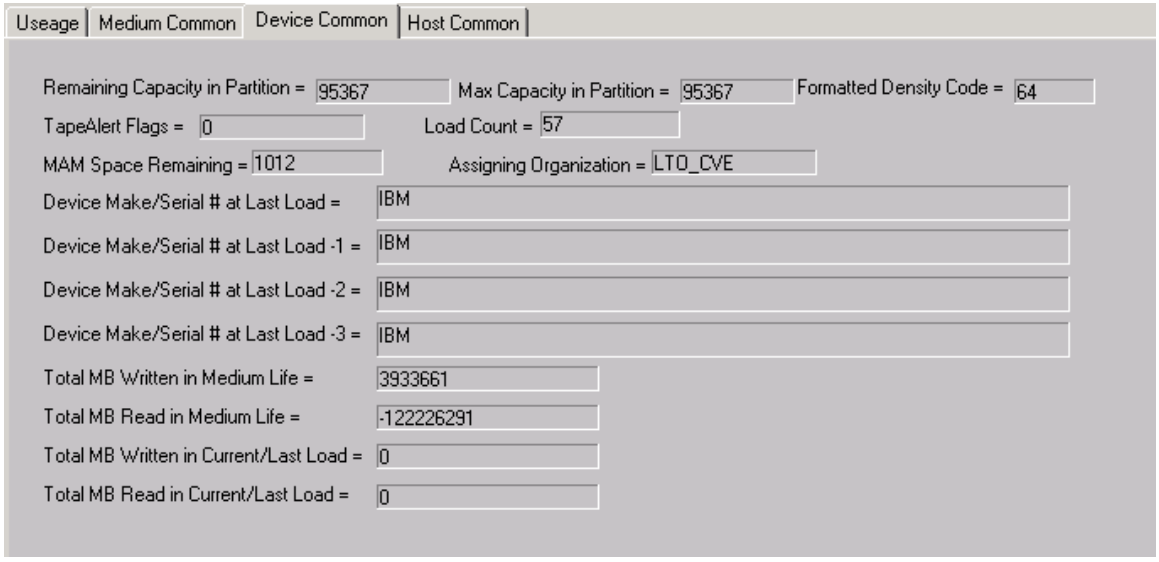


This command allows you to create an update tape from a firmware file. Use the Browse button to specify the firmware file name, then click the Create Tape button. You can then use this update tape to upgrade the firmware in other DLT tape drives.

LTO Cartridge Memory Display

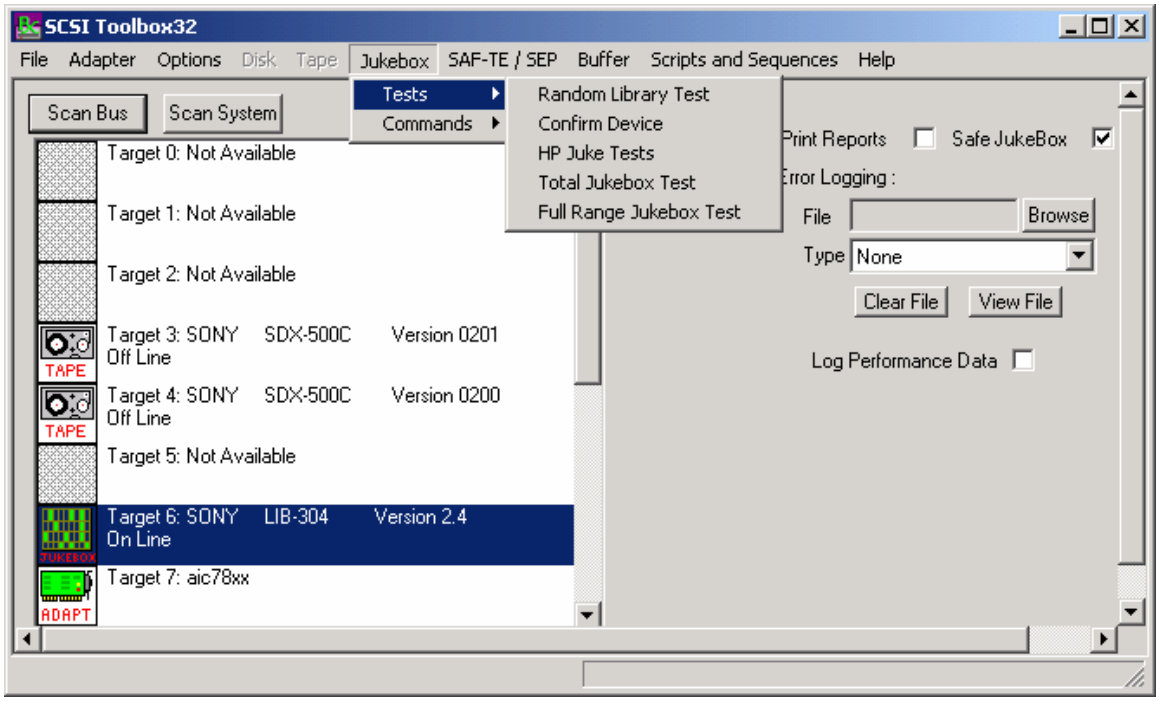
This command allows you to display the cartridge memory stored within LTO tape cartridges.





Jukebox Tests

The following tests are available via the Jukebox Test menu:



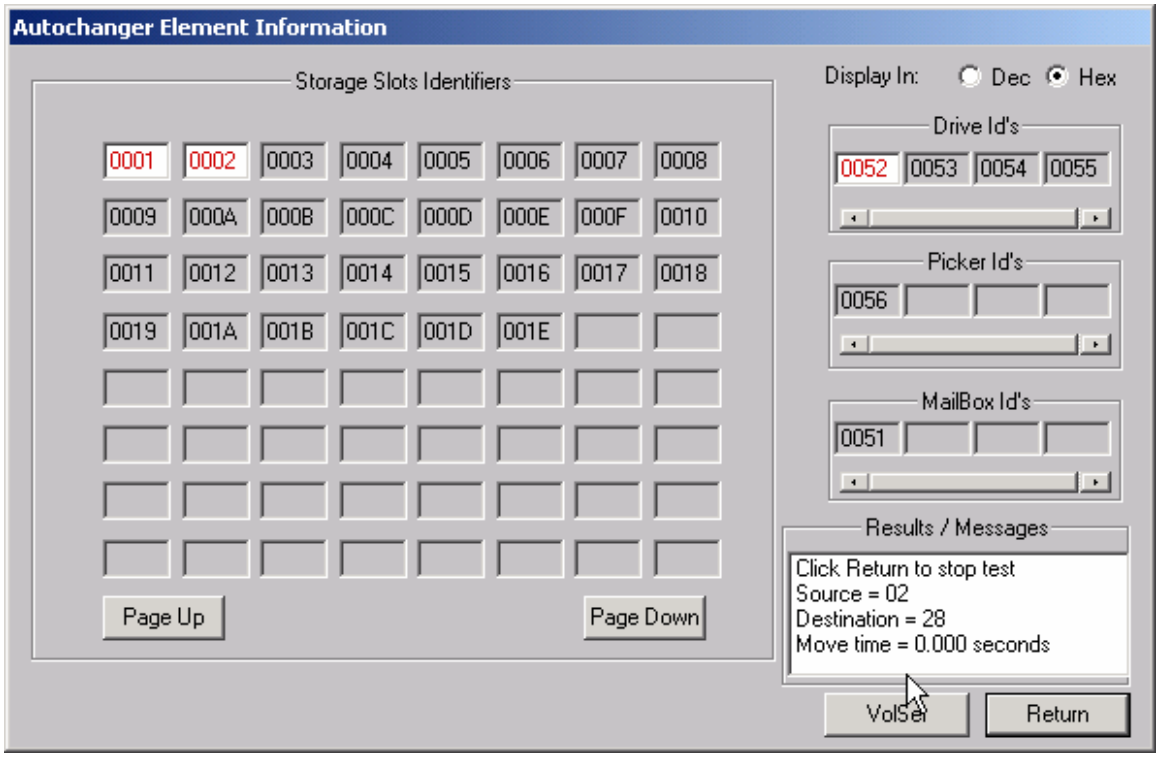
The jukebox module uses a display screen to show the configuration of the jukebox under test. The eight by eight matrix represents the storage elements of the jukebox, allowing the display of up to 64 storage elements at a time. If a storage element it present, it's storage element address is displayed. 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, it's 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. Ther 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

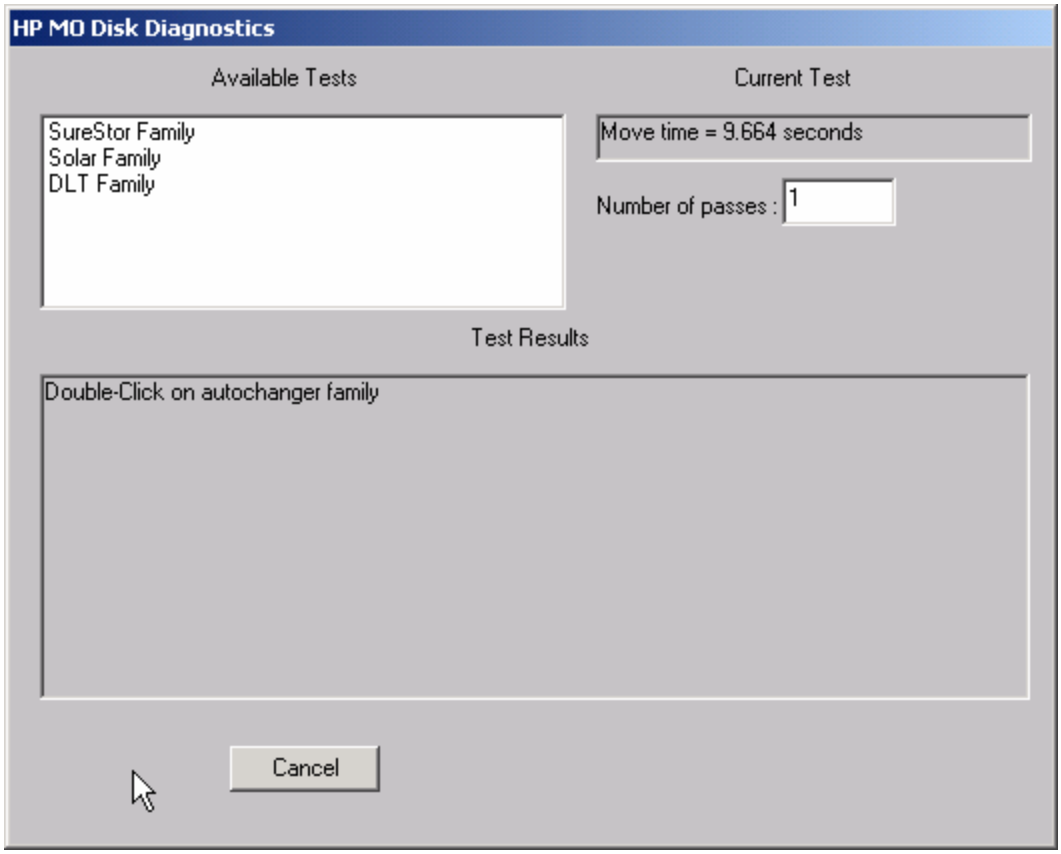


This test will move a piece of media randomly between the jukebox storage elements. It requires at least one empty storage element, and at least one storage element with media in it. As the test runs, it will randomly choose a full storage element and randomly choose an empty storage element. Then it 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.

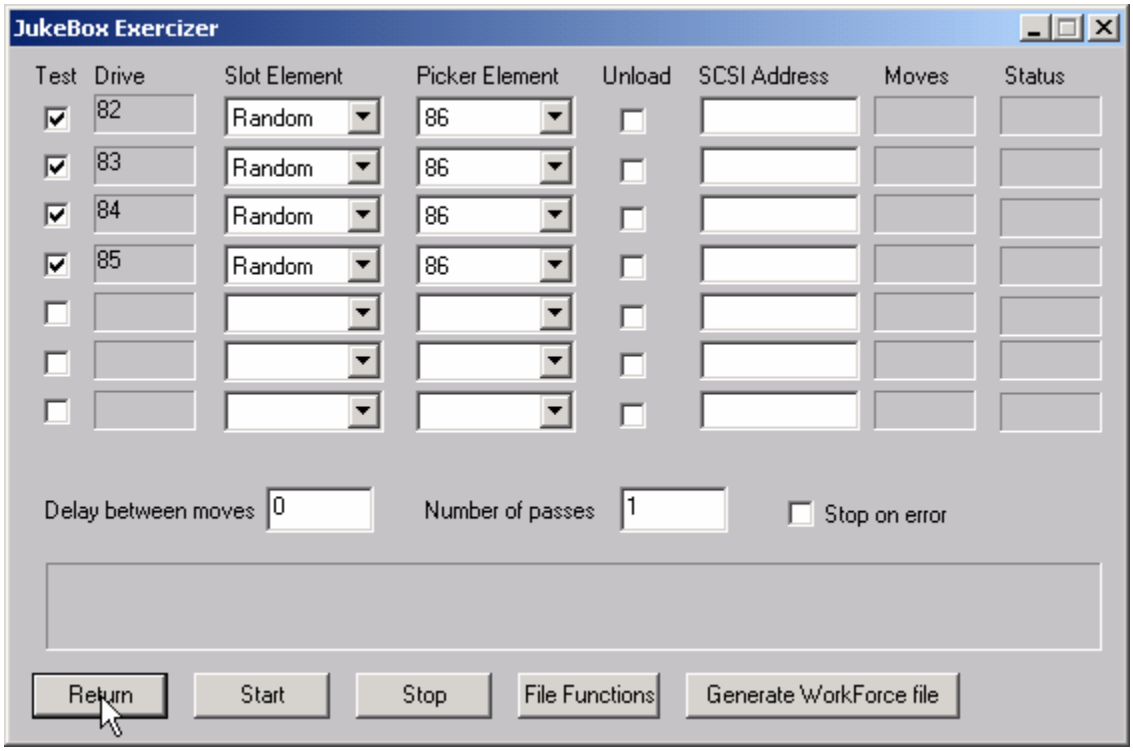
HP Juke Tests



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 decrimented as each pass completes.

Total Jukebox Test



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.

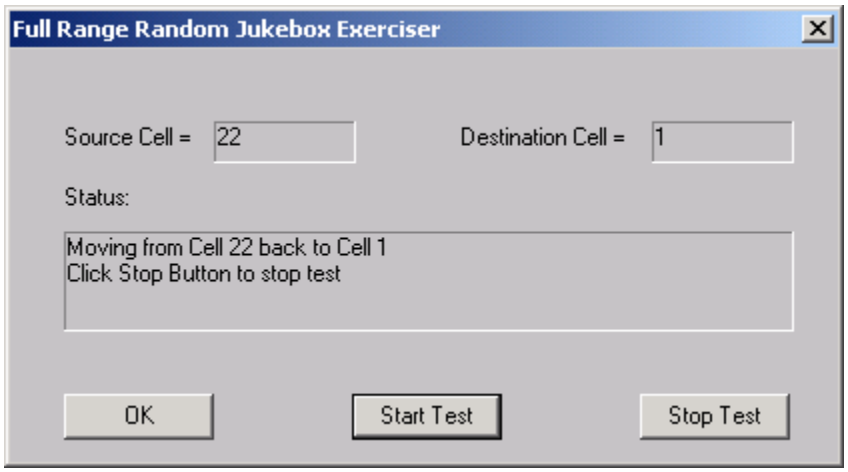
You may specify an amount of time to delay between moves with the Delay between moves field. Some jukeboxes require some “settling time” between moves.

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

Click the Start box to begin the test.

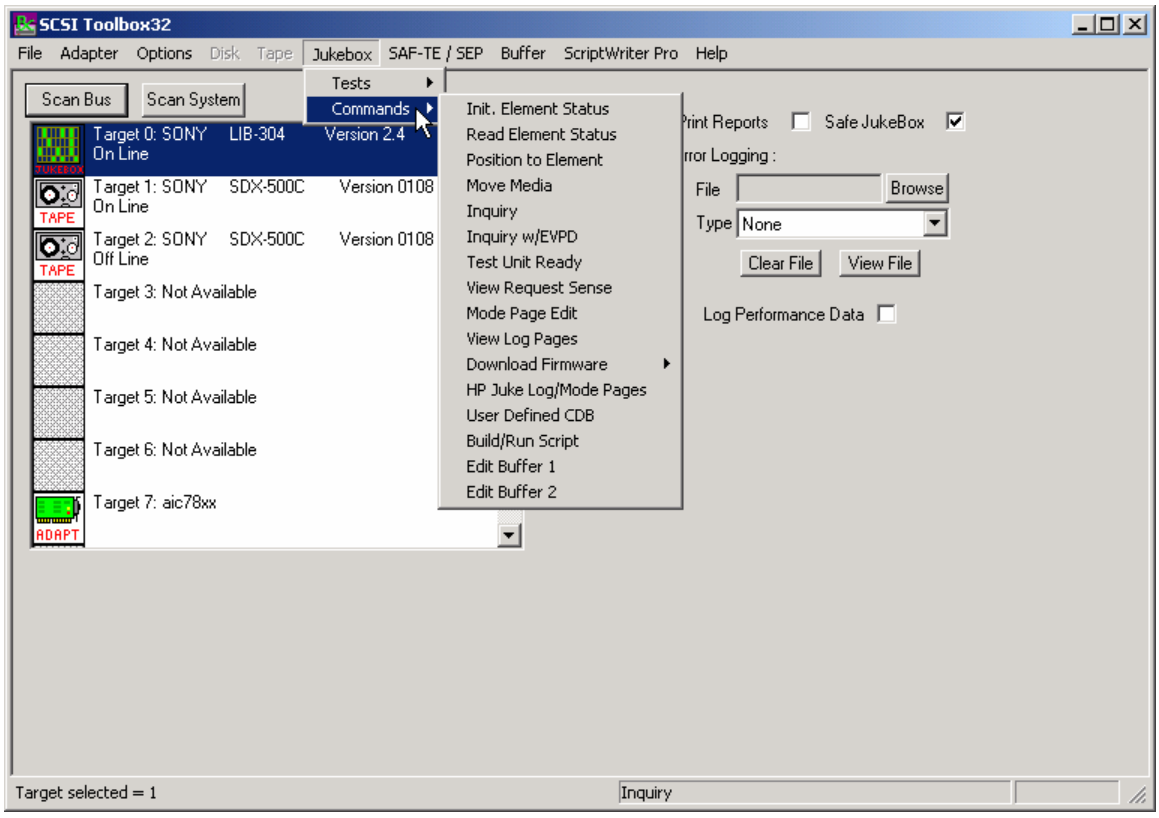
JukeBox Full Range Test

This test will include moves from any location within the jukebox

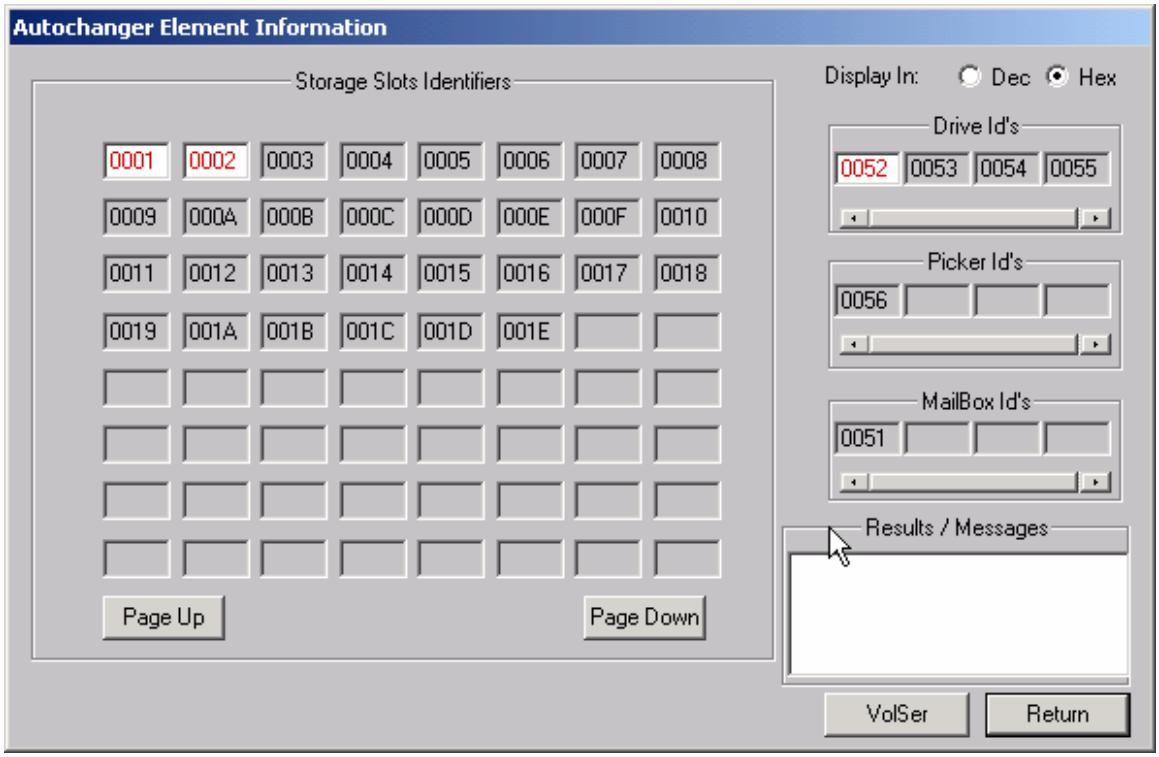


Jukebox Commands

The following commands are available via the Jukebox Commands menu:

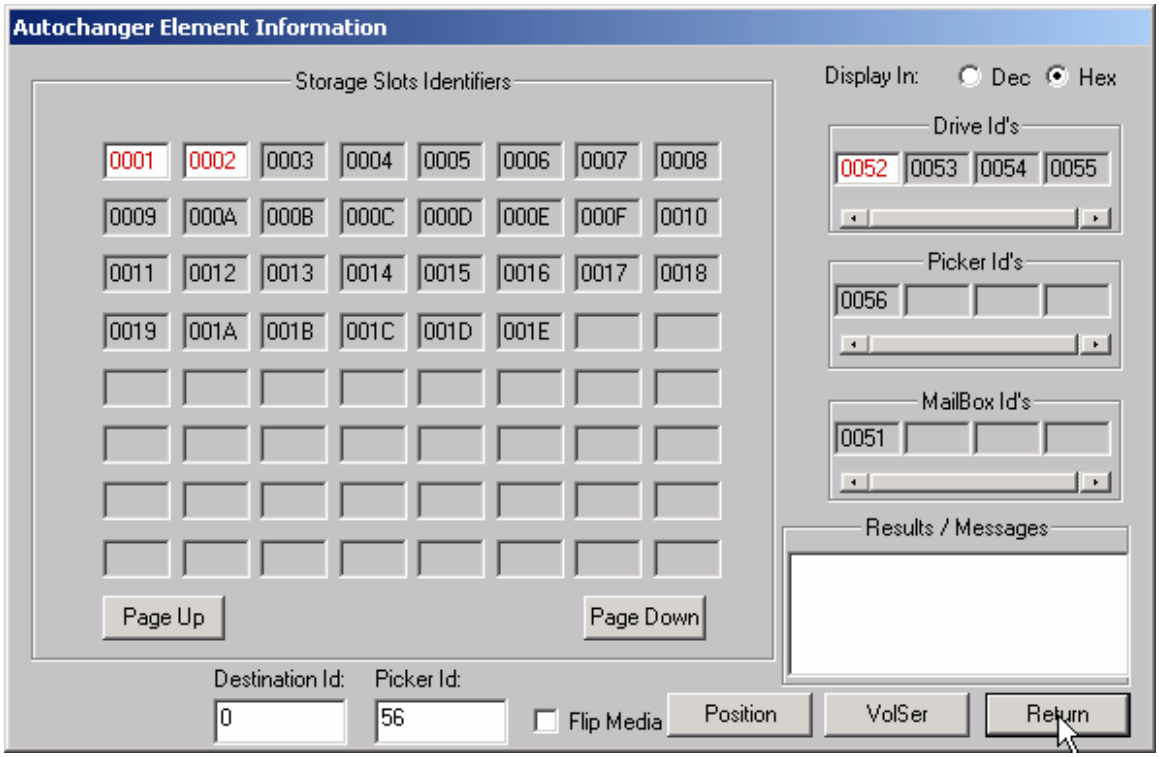


Initialize Element Status



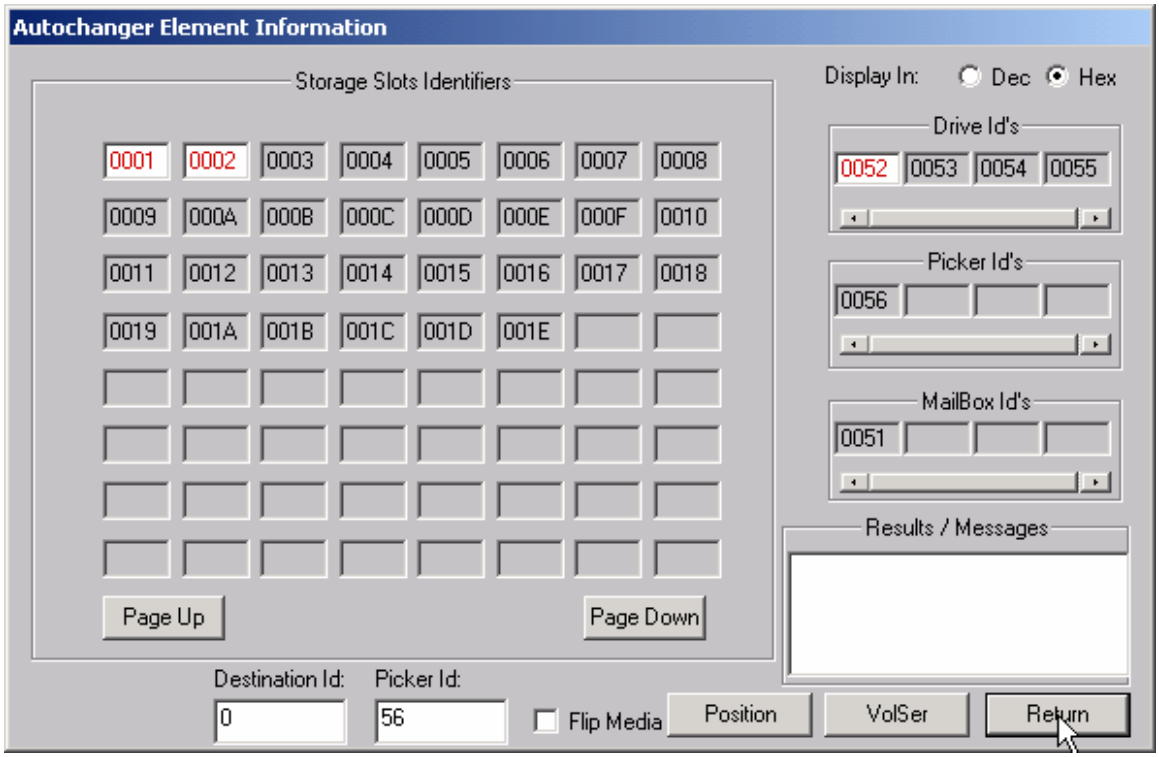
This command issues a SCSI INITIALIZE ELEMENT STATUS command to the jukebox. This command causes the jukebox to check its configuration and determine the full or empty status of all of its elements. It is good practice to ALWAYS run this command before running any other commands or tests.

Read Element Status



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

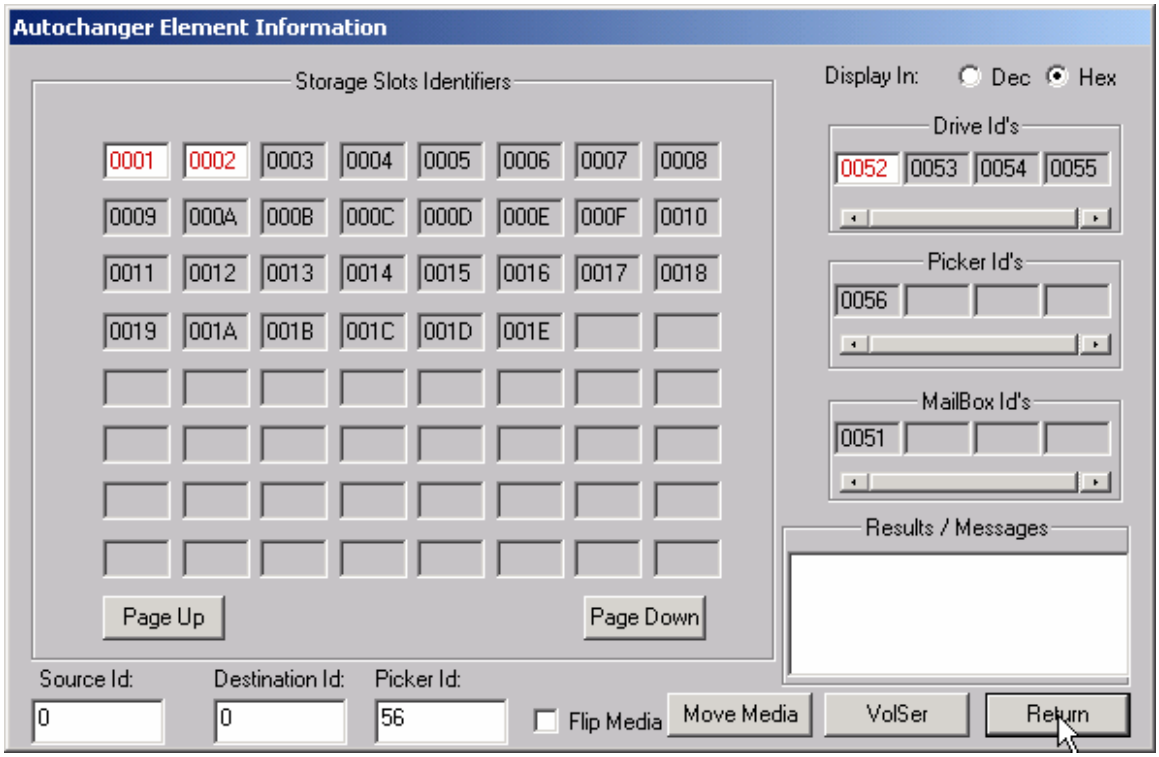
Position to Element



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

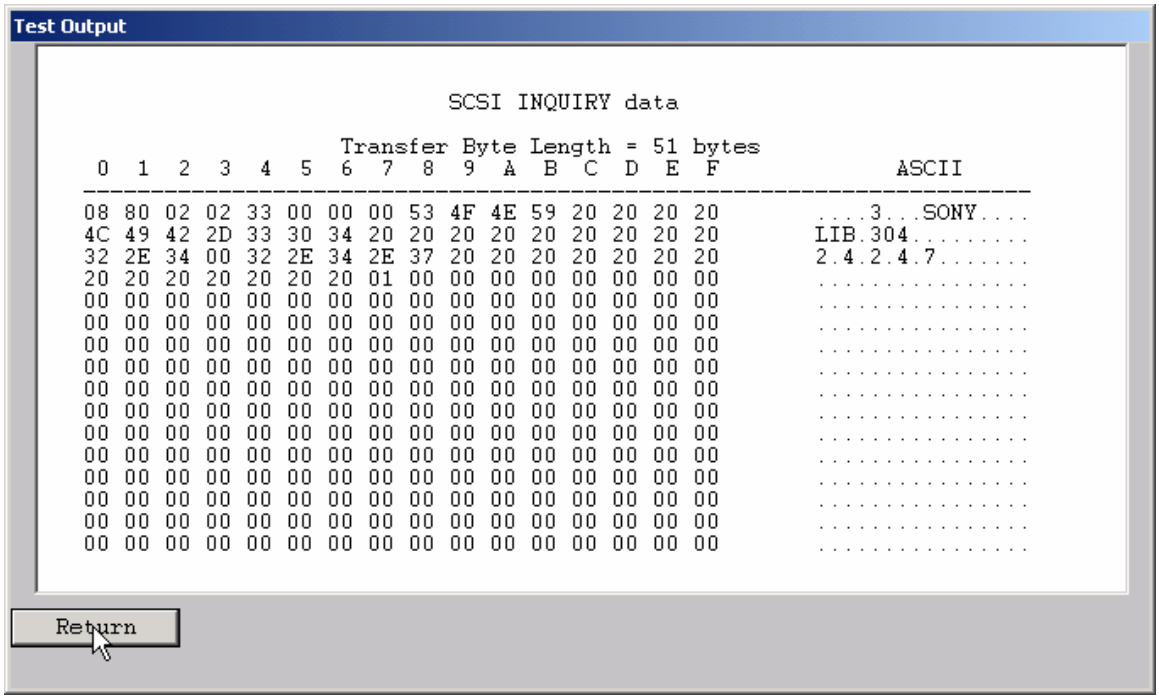


This command will prompt you to enter the source, destination, and transport address, and then will attempt to move the piece of media from the source element address to the destination element address via the transport element address. This assumes the following:

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.

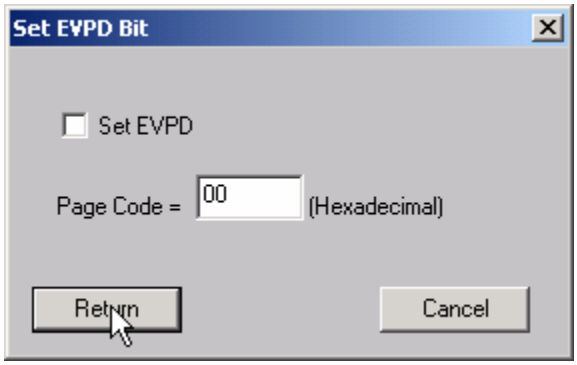
You may also use the drag and drop method to pick up a source and drag to a destination. When you drop on the destination the values will be filled in on the edit boxes. Simply confirm that the desired transport address is selected and press the move button to execute the move.

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

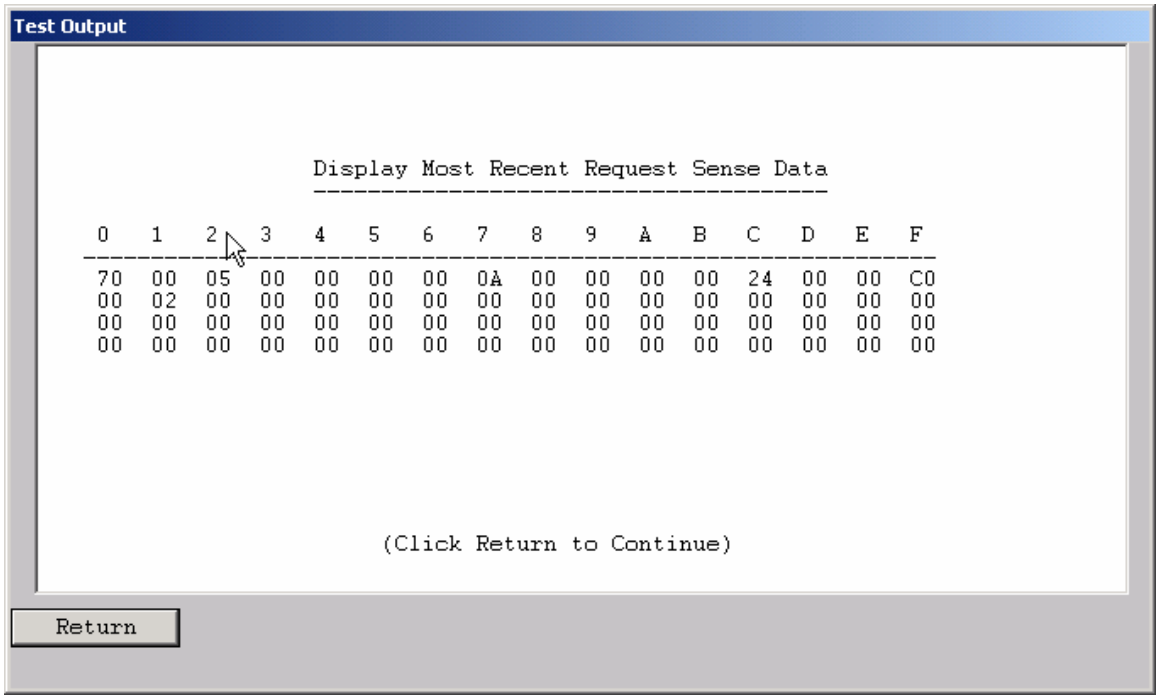


This command allows you to specify setting the EVPD bit and Page Code for the INQUIRY command.

Test Unit Ready

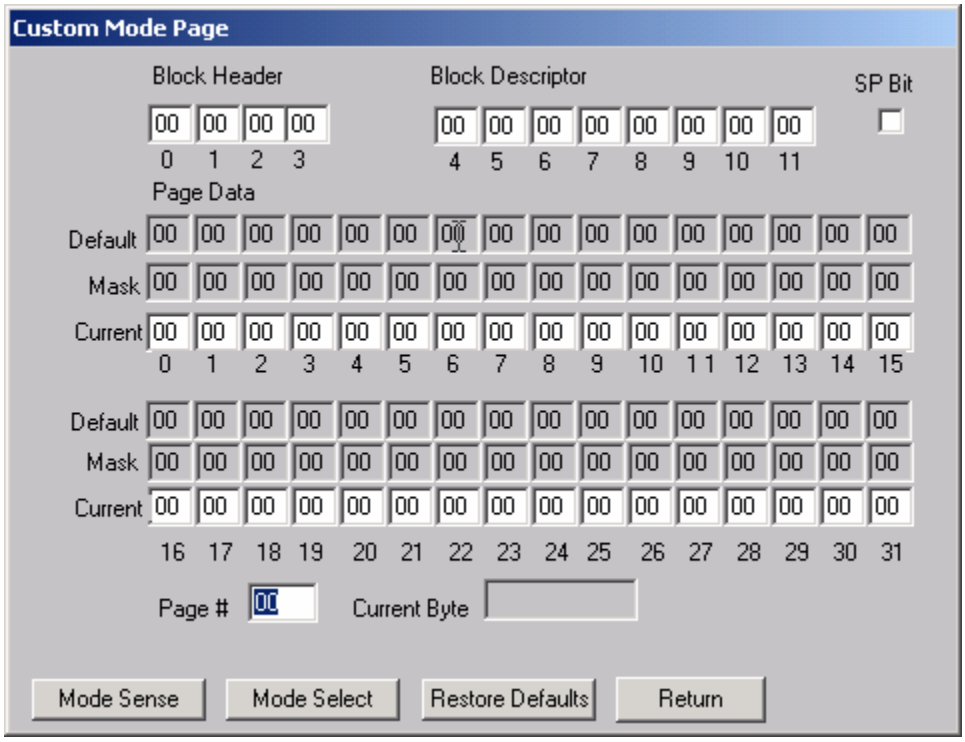
This command displays the return value from a SCSI Test Unit Ready command. A value of zero indicates a successful Command Complete.

View Request Sense



This command will display the most recent request sense data.

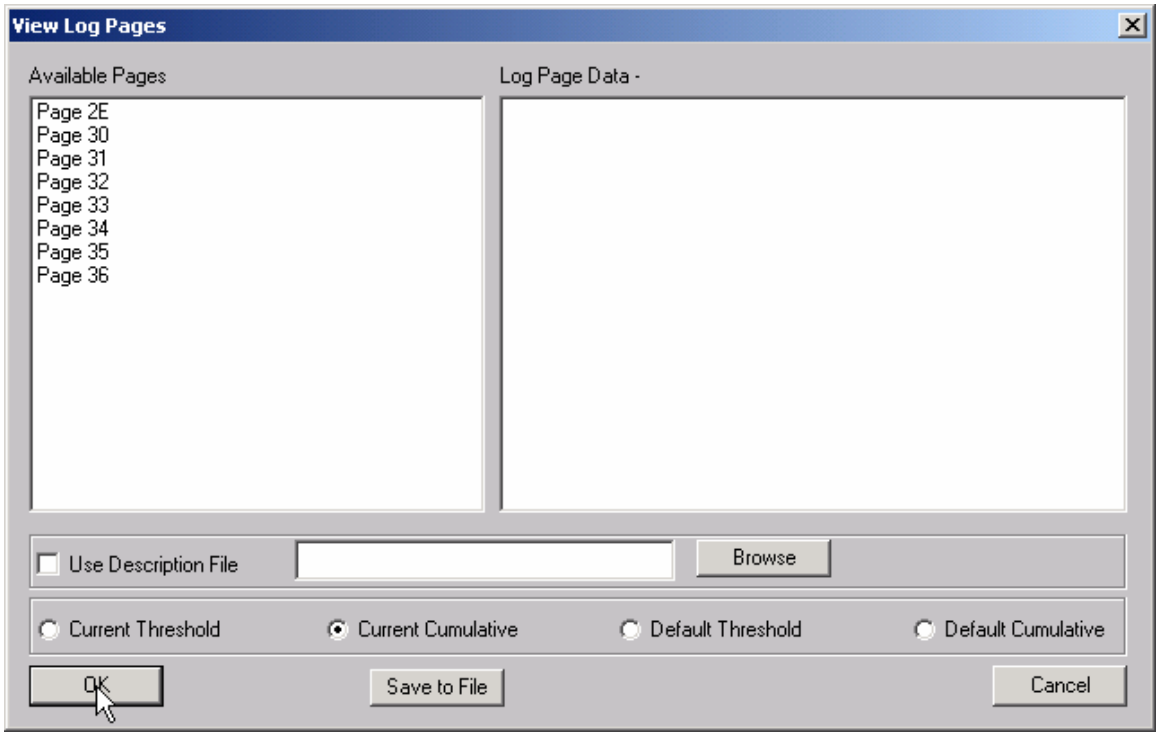
Mode Page Edit



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.

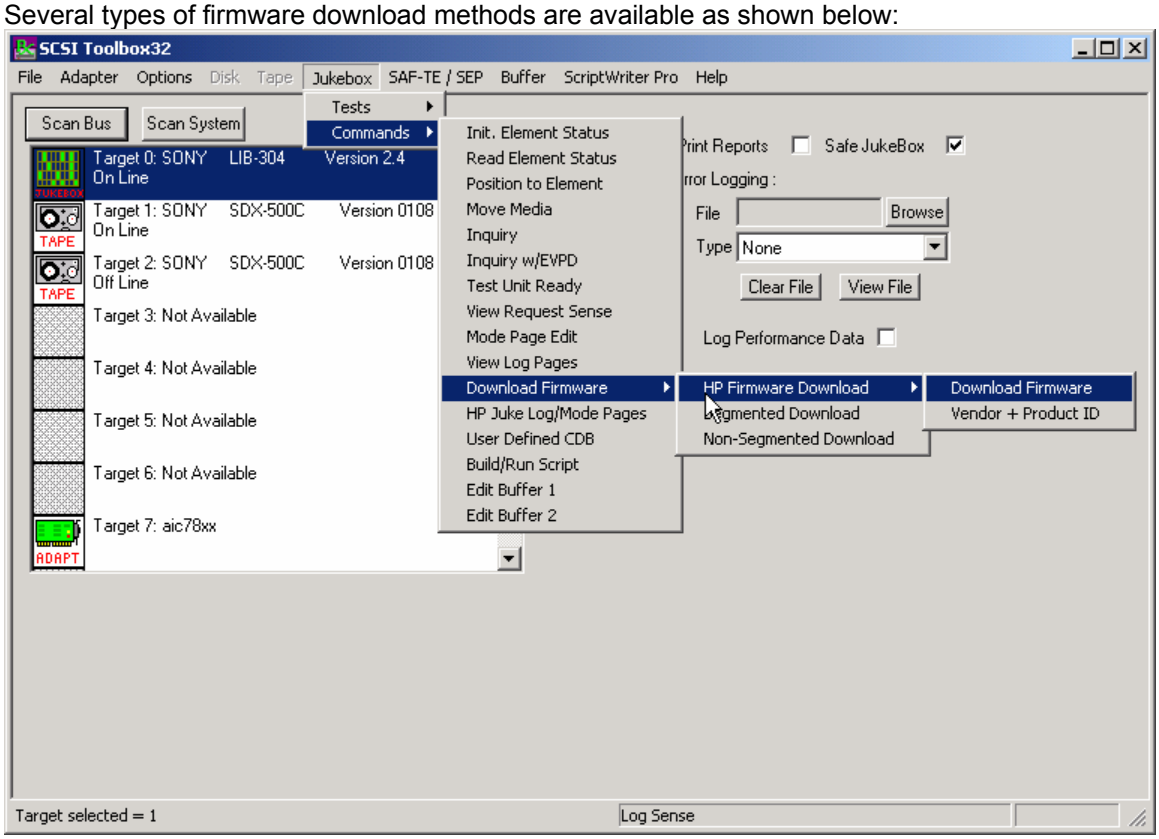
View Log Pages



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

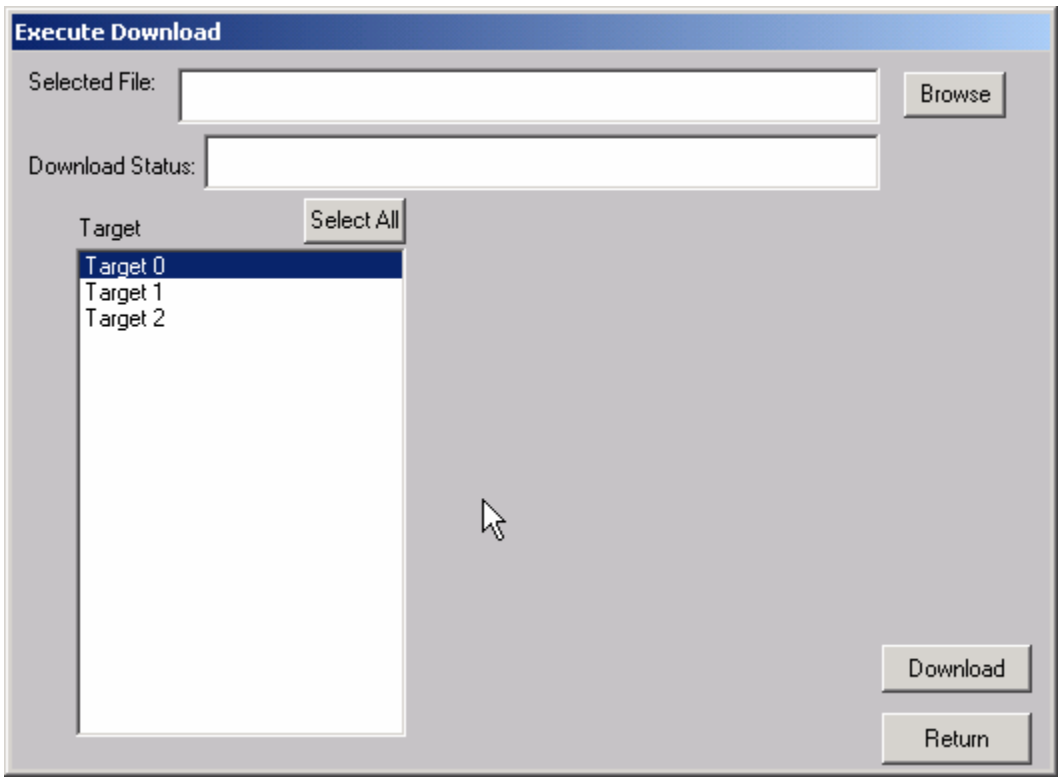
This command duplicates the functionality of hp-dload.exe. The user is prompted to specify the first download filename. The program then reads that file sends it to the drive then increments the extension by 1 and looks for a file which matches. If it finds a matching file it sends that file to the drive. 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).

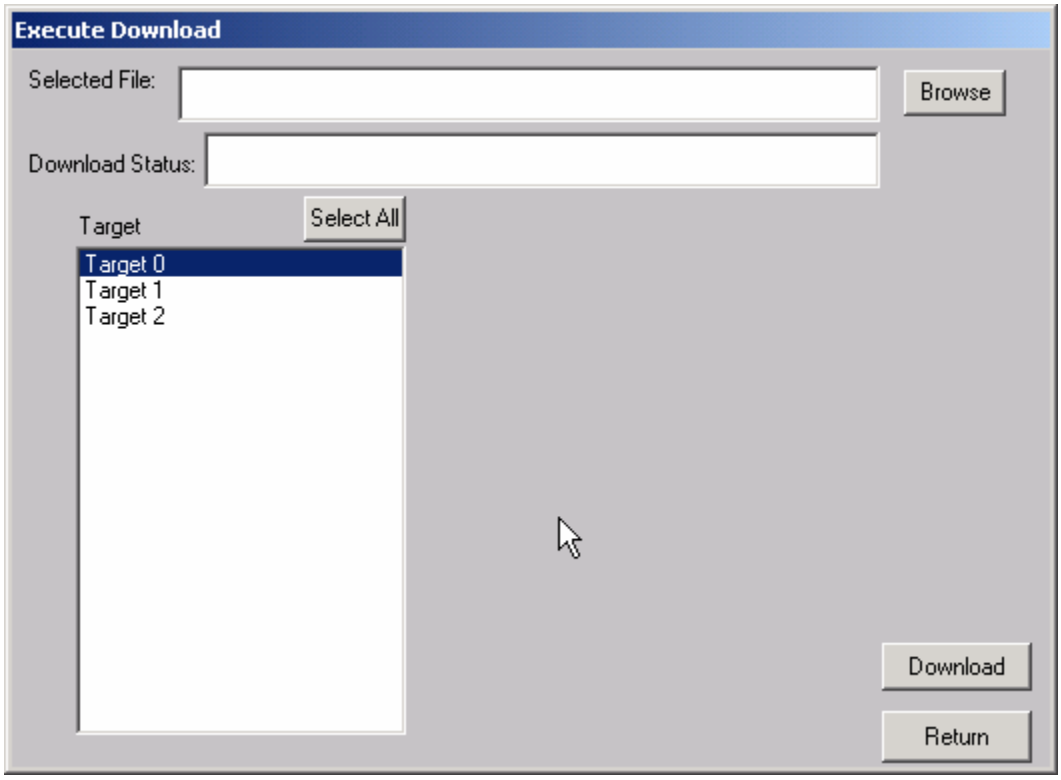
Segmented Download



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.

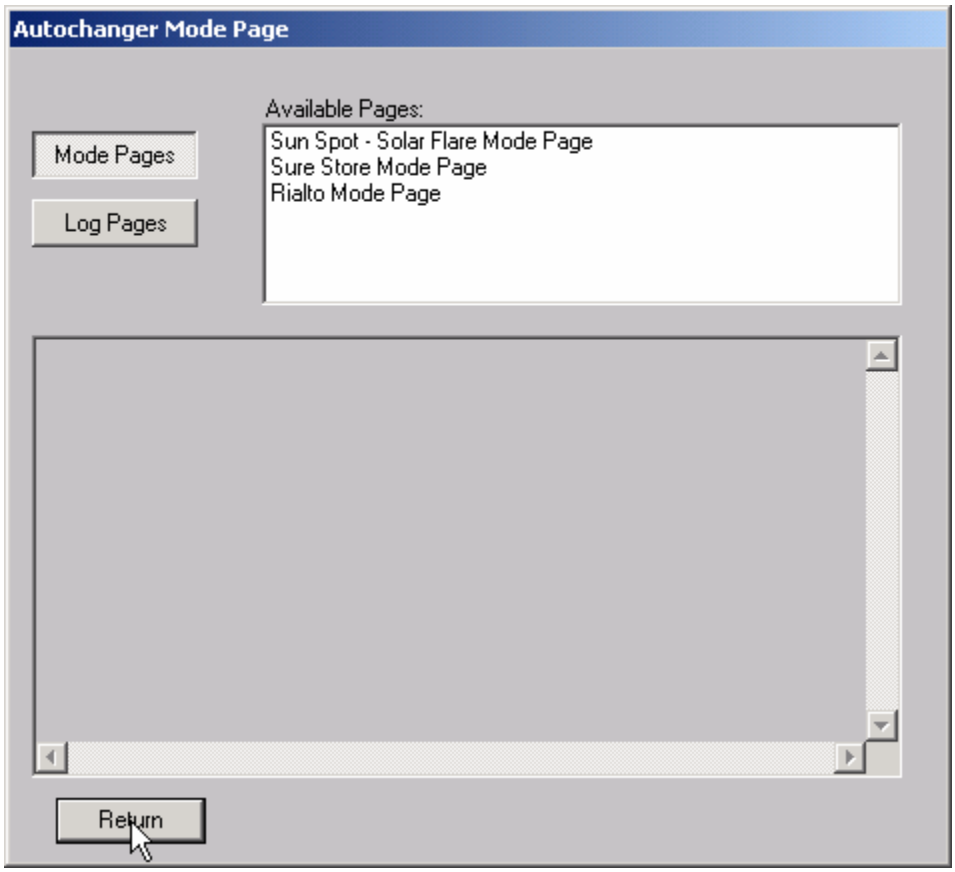
Non-Segmented Download



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.

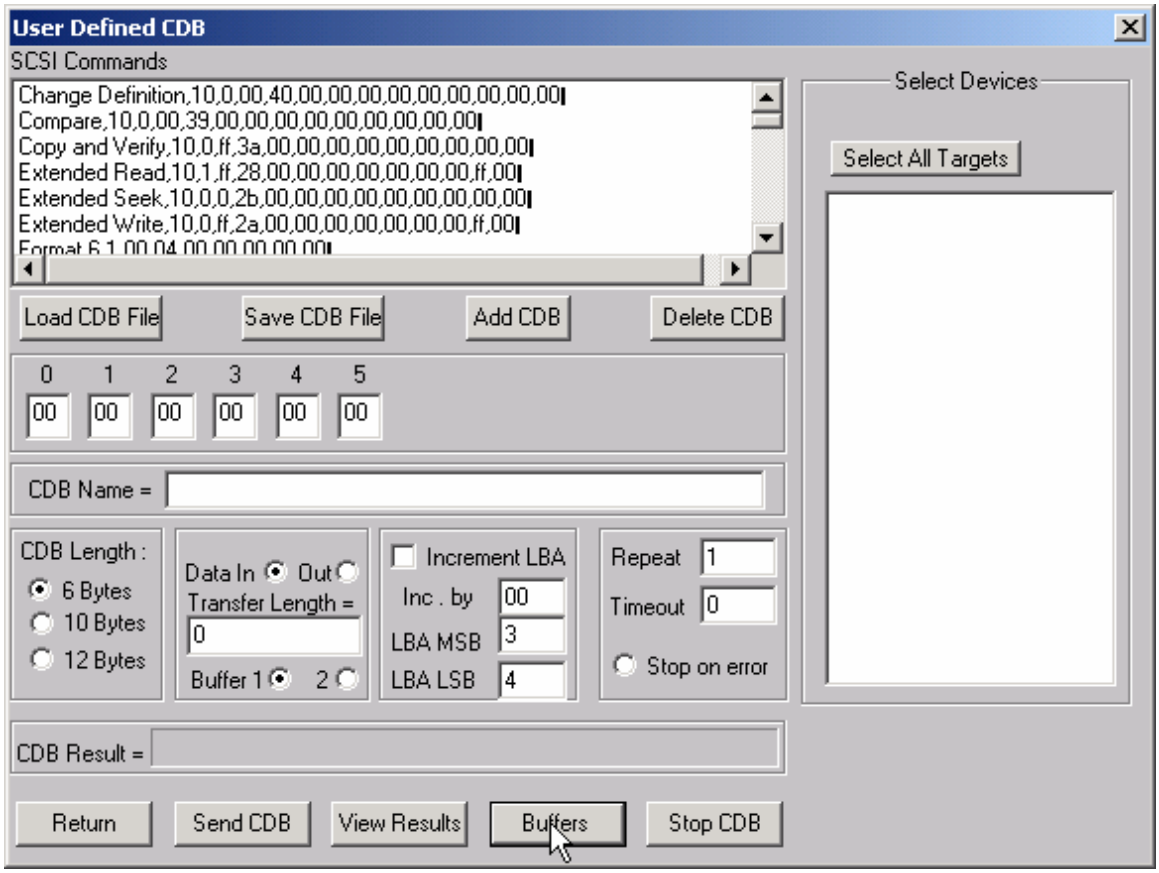
If the available buffer space is not large enough to hold the file the download will fail. The "About SCSI toolbox" choice of the "Help" menu displays the available buffer space.

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 SCSItoolbox 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.

Definining 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:

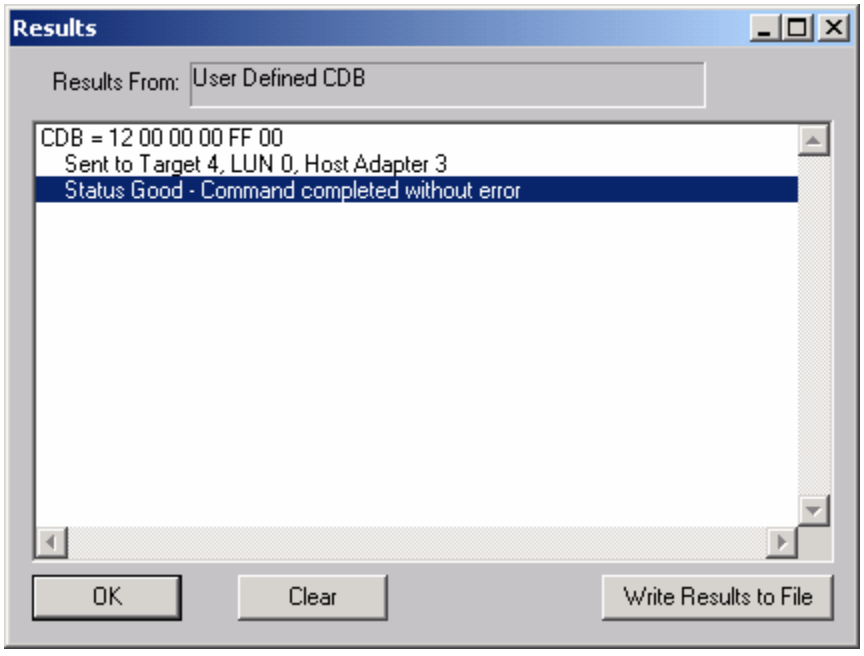
- Data direction incorrect
- Data length specified in CDB does not match Data Transfer Length

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:

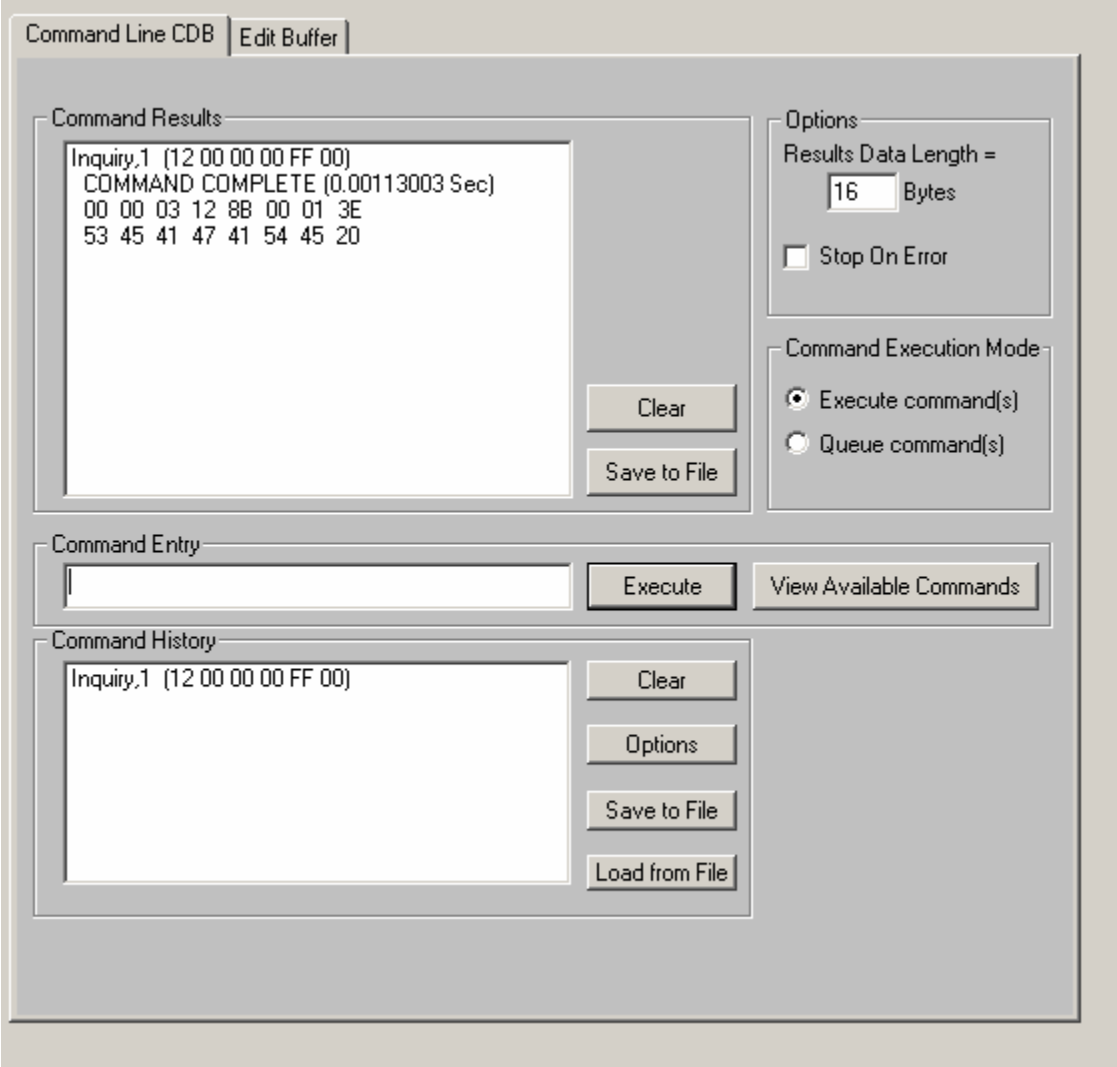
- Check the Increment LBA box
- Specify which CDB byte is the LSB of the SCSI CDB
- Specify whick CDB byte is the MSB
- Specify how much you want to increment by

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

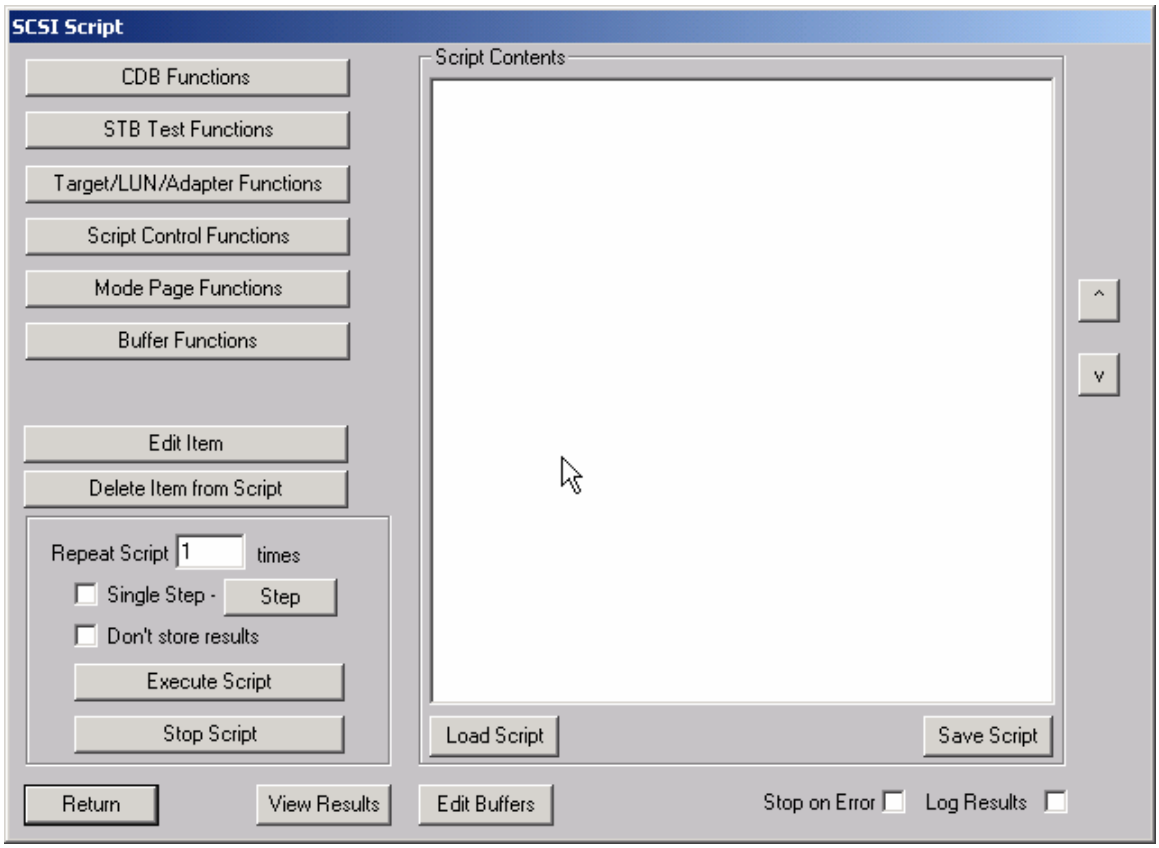
The “Results Data Length” allows you to specify how many bytes of data are displayed in the “Command Results” window. If a command transfers data, this many bytes will be displayed along with the command results. If the command terminates in a check condition, this many bytes of sense data will be displayed.

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

Re-issuing commands

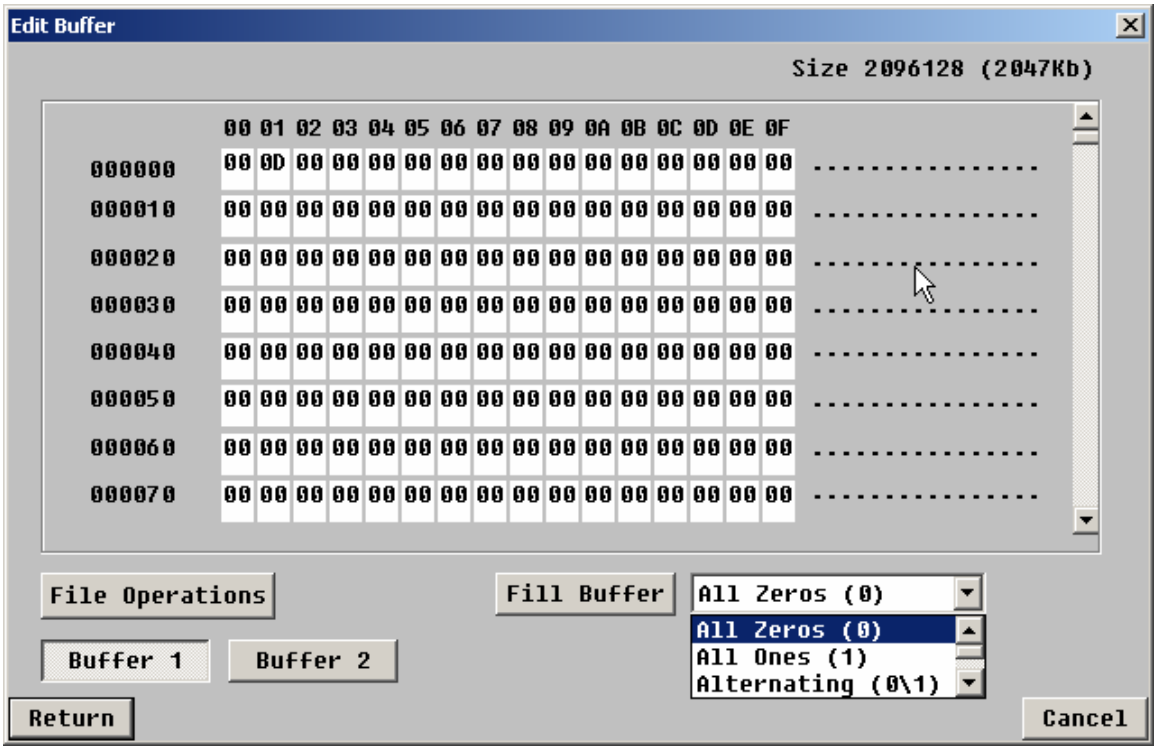
Any command or commands can be re-issued by selecting them (clicking on them) in the “Command History” window. Once the commands you want to reissue are highlighted, click the “Execute” button and they will be reissued.

Build/Run Script

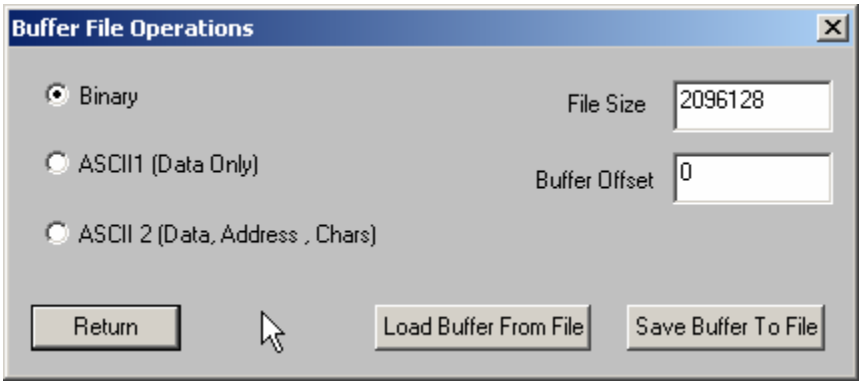


Edit Buffers

The buffer functions allow the buffers used by the SCSItoolbox32 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.



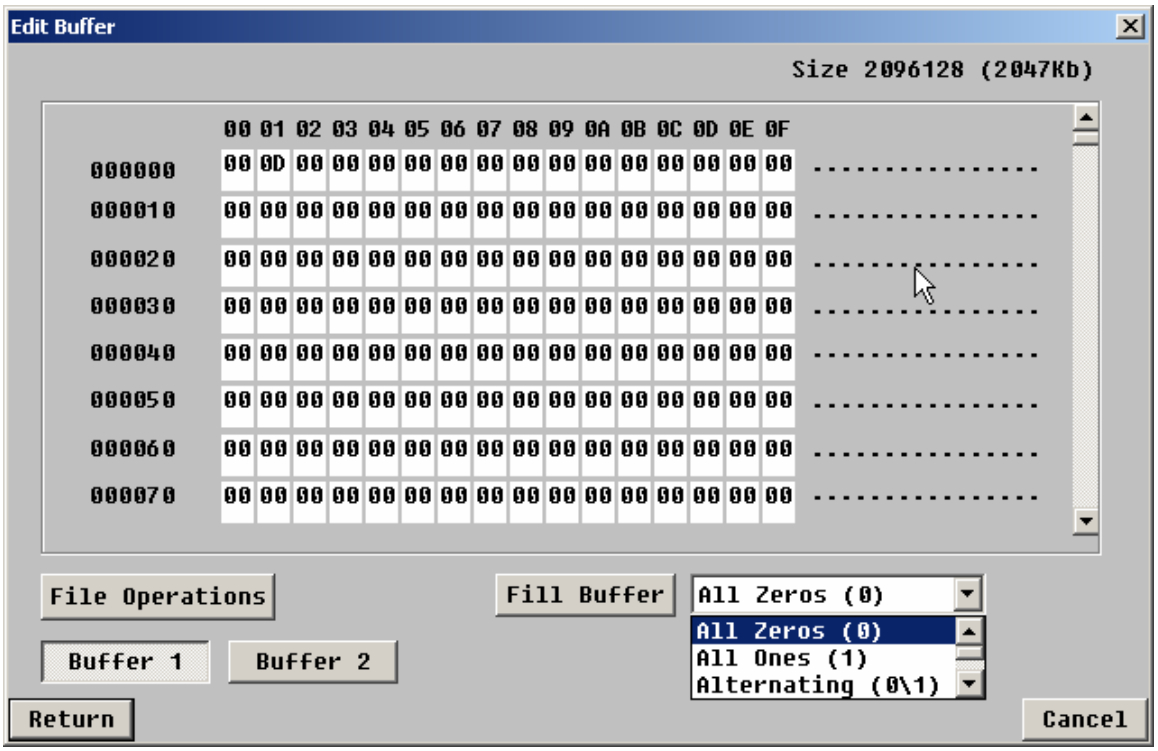
The drop box next to the Fill Buffer button allows you to choose a pattern to load into the buffer. Select the data pattern, then click the Fill Buffer button to load the data into the buffer. The File Operations button allows the buffer date to be saved to or loaded to a disk file.



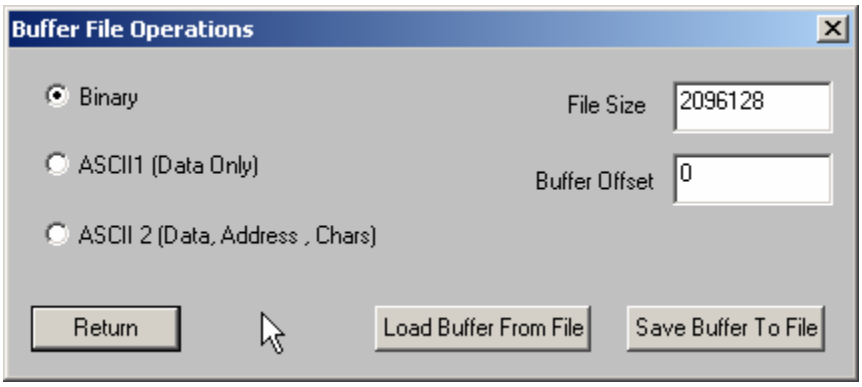
User choice allow the data to be saved either as binary data, or as ASCII data with or without the hex address of the data. The File size specifies how may bytes of data to read or write.

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 SCSItoolbox32 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 date 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.

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 to selection 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 messages 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:

	Bit 7	6	5	4	3	2	1	0
Byte 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 ar 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	ASCQ	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 RESYCHRONIZATION 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	ILEGAL 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      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 OOOOOOOOO    COMPARE
3A OOOOOOOOO    COPY AND VERIFY
3B OOOOOOOOOOWRITE BUFFER
3C OOOOOOOOOOREAD 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.

Installation

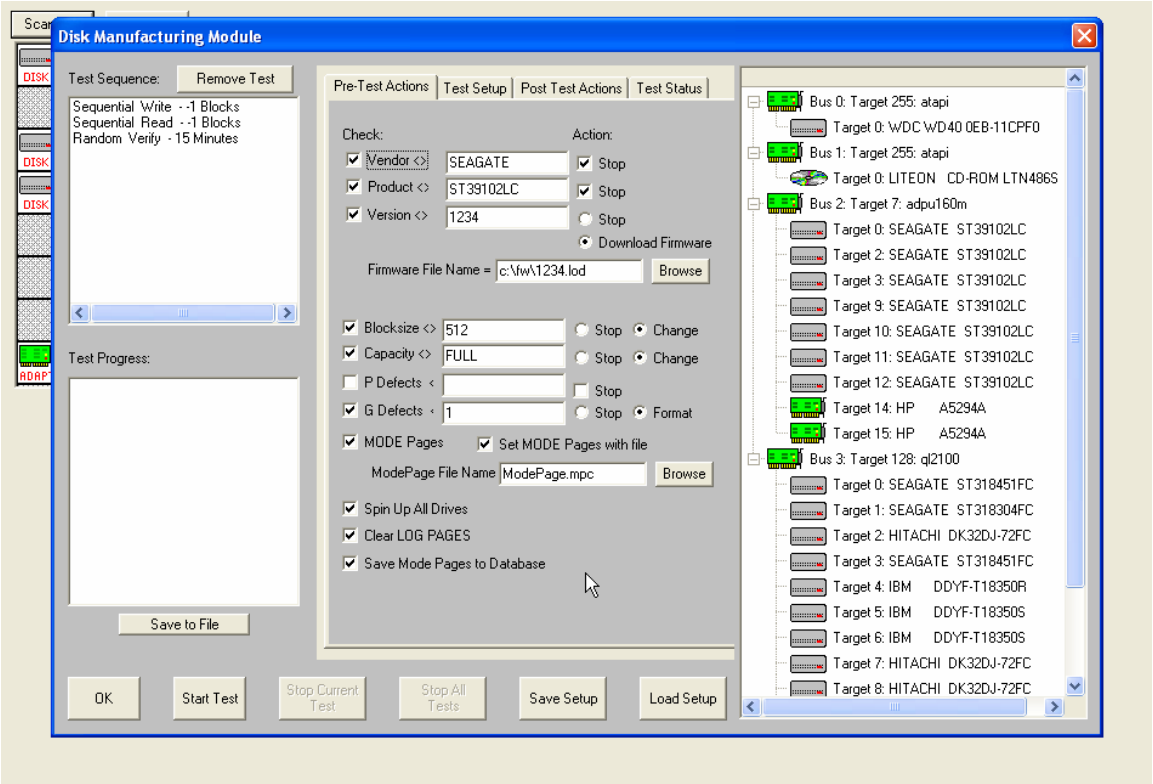
Installation will vary depending on if you purchased the Stand Alone Disk Screening Toolbox or if you purchased the SCSItoolbox, which includes the Disk Screening Module.

In either case, simply insert your CD into your CD drive, and the installation process should begin. If the process does not autorun, use Explorer to navigate to the CD root folder, and double-click on the file Setup.exe to begin the install process.

Plug your hardware “dongle” key into your printer port or USB port.

Getting Started

Starting the Stand Alone Disk Screening Toolbox, or choosing the Disk Screening Module from the Disk Menu of the SCSItoolbox will present you with this screen:



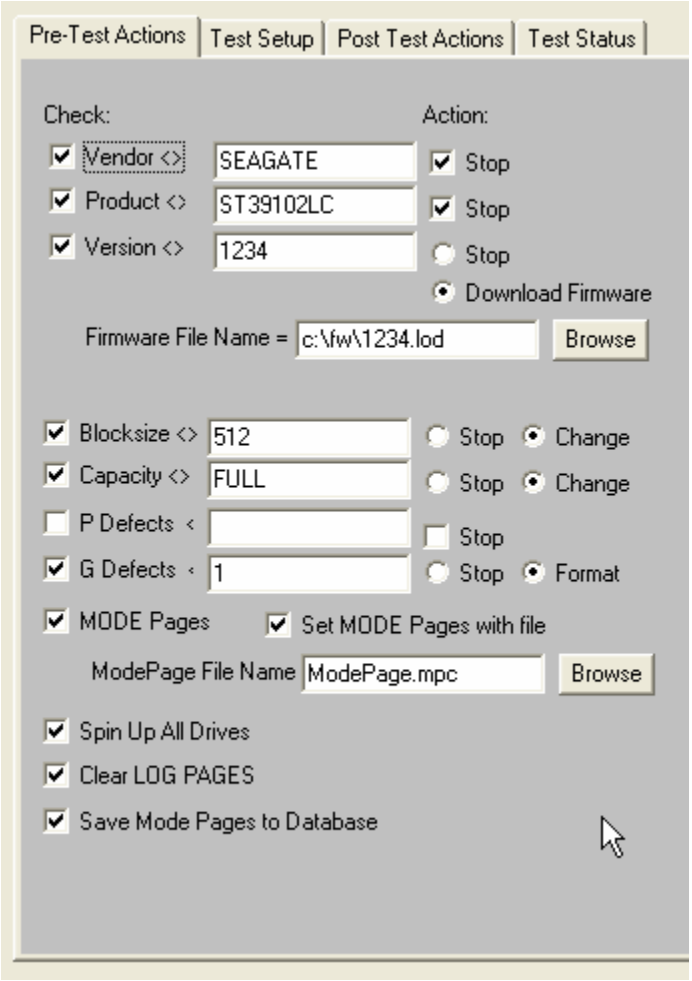
The left side of the screen has Test Sequence and Test Progress list boxes, the center part of the screen has several tabbed pages for configuring the test process, and the right side of the screen shows a tree display of all of the adapters and devices attached to the system.

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:



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 ST39102you 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 seach 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

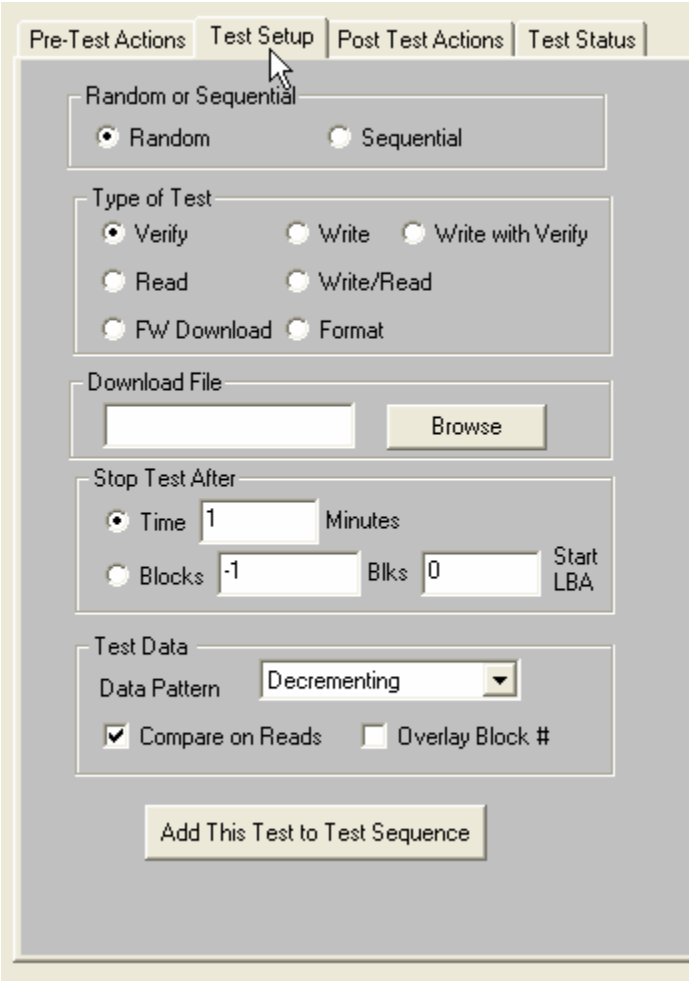
Action = All drive LOG PAGES will be cleared before the test sequence is started. Log Pages contain run-time counters such as read and write error counters, etc. Clearing these pages before testing allows everything to be started “at zero”

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:



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 then 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 a 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

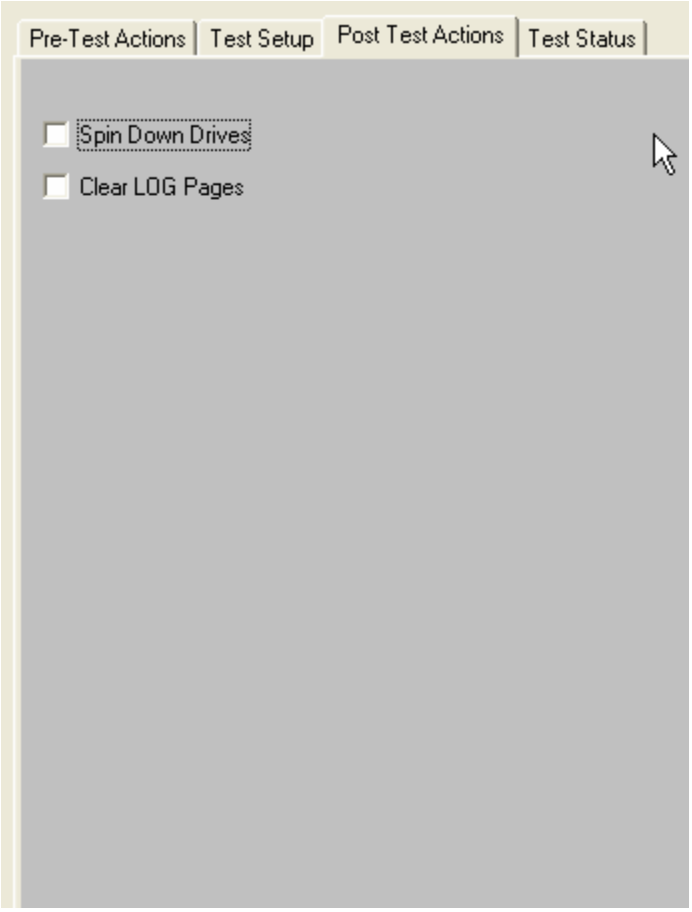
Checking this box will cause all data read to be checked that it compares with the expected data. If a miscompare is found the test will be aborted and the miscompare data will be recorded in the database and will be viewable in the Test Status tab page.

Overlay Block Number

Checking this box will overlay the block number over the first n bytes of each block of data.

Post-Test Actions

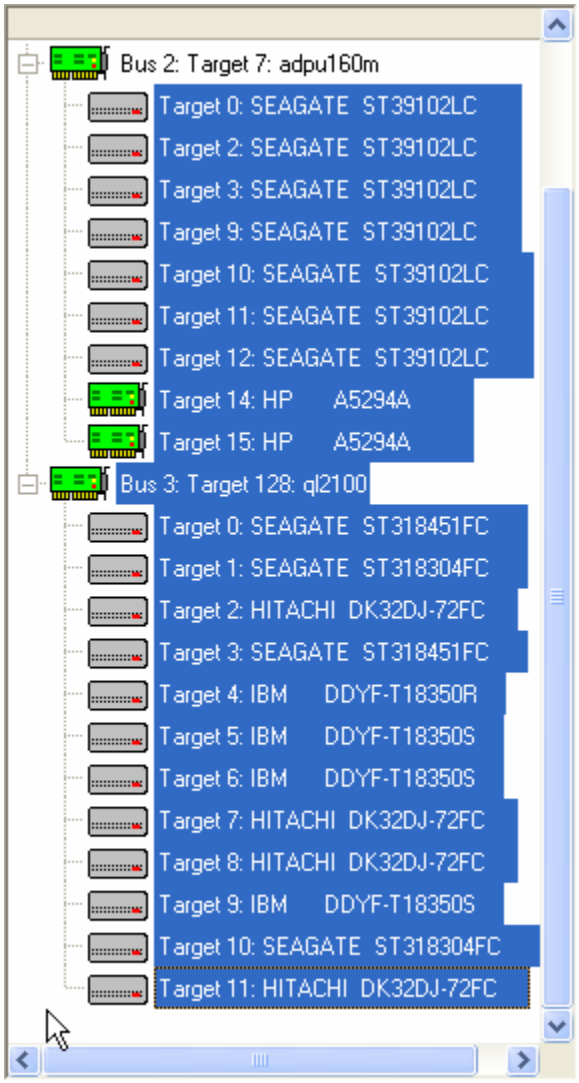
There are currently two choice, as shown below



You can choose to spin all disks down after testing, and to clear all Log Page data.

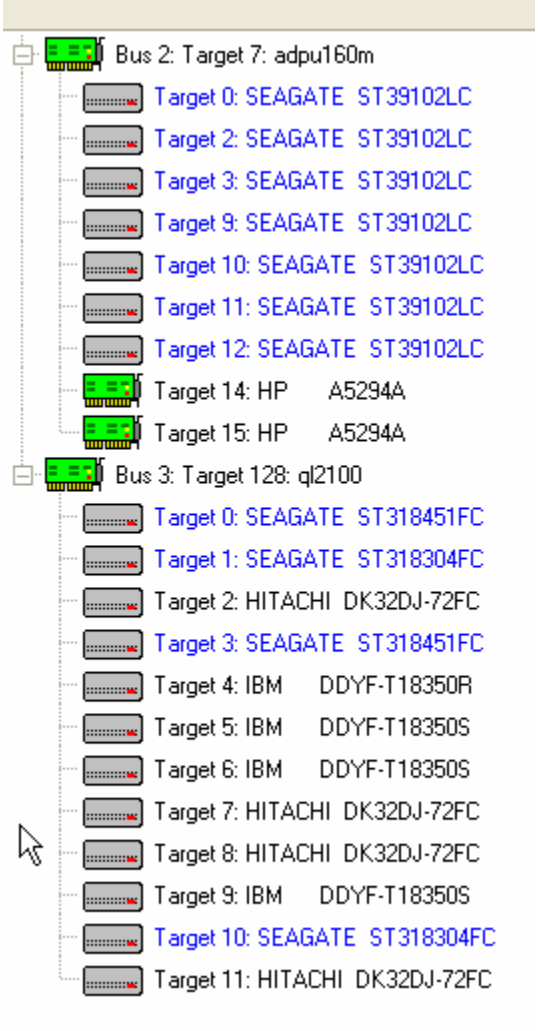
Selecting Disks to Test

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:



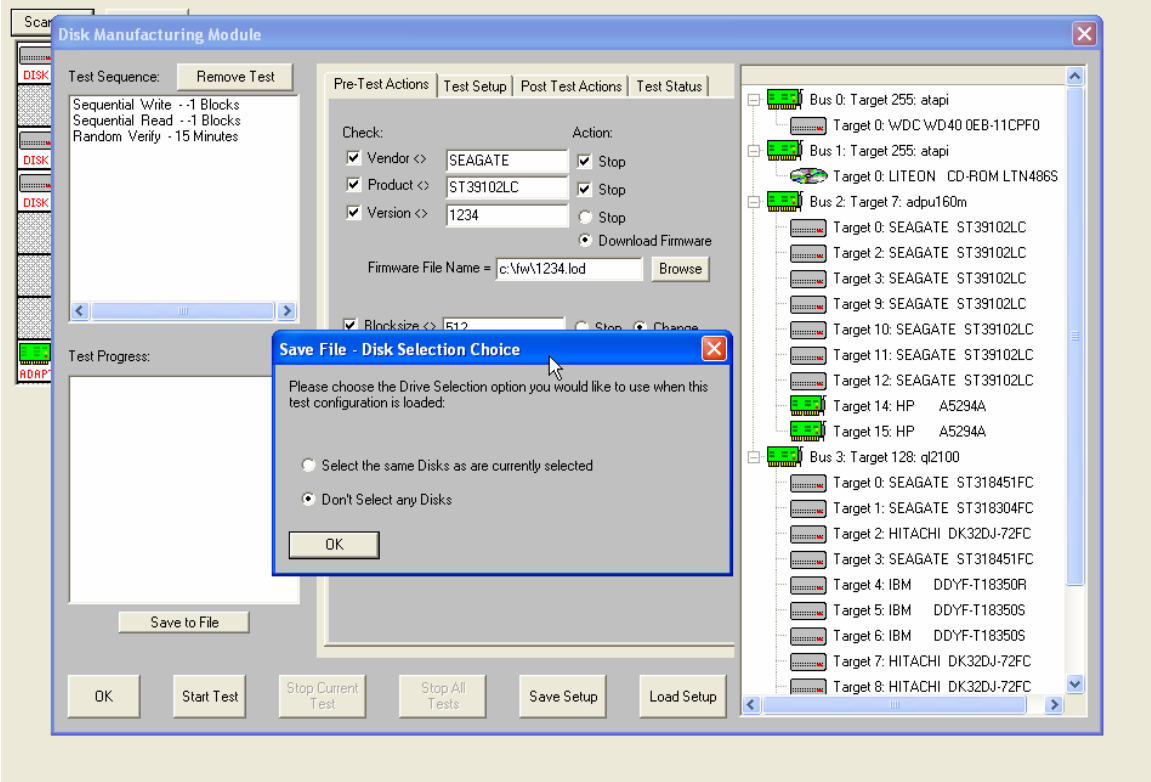
Saving and Loading Test Configurations

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 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 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

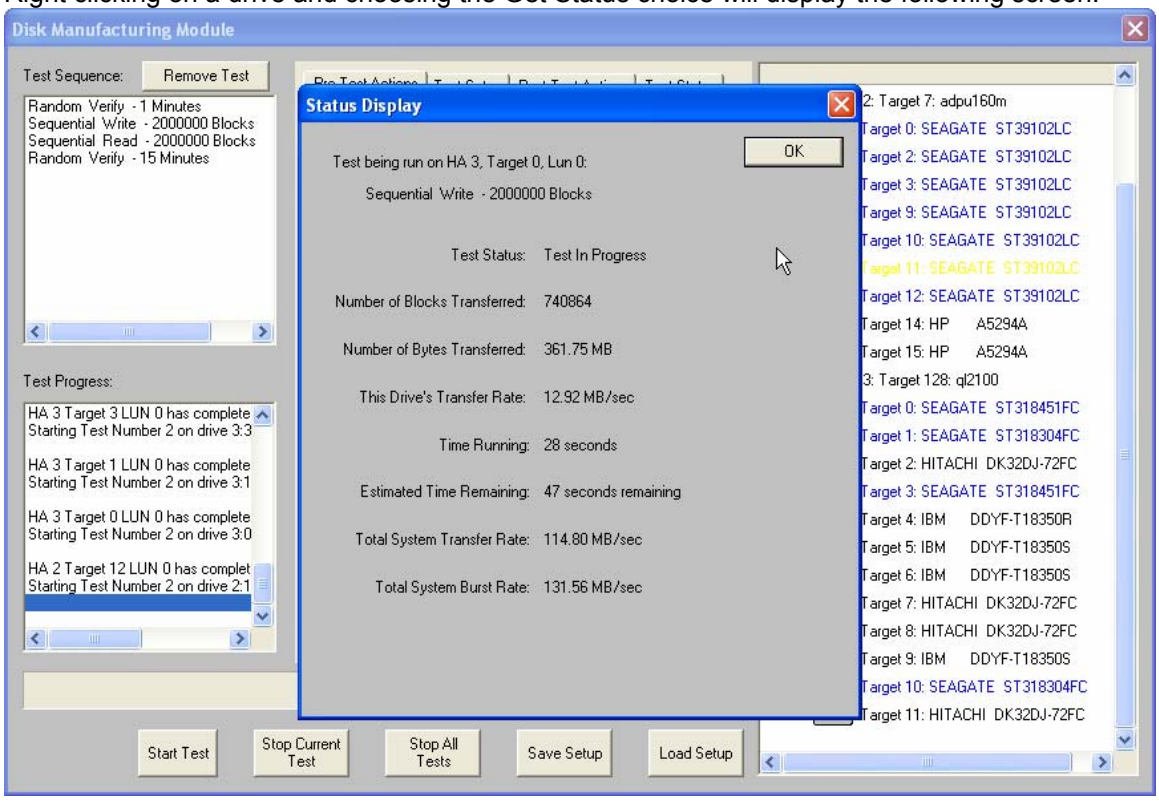
Right click on the paused drive, then click the Resume Test choice to resume testing. The drive is displayed in Blue highlight again

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

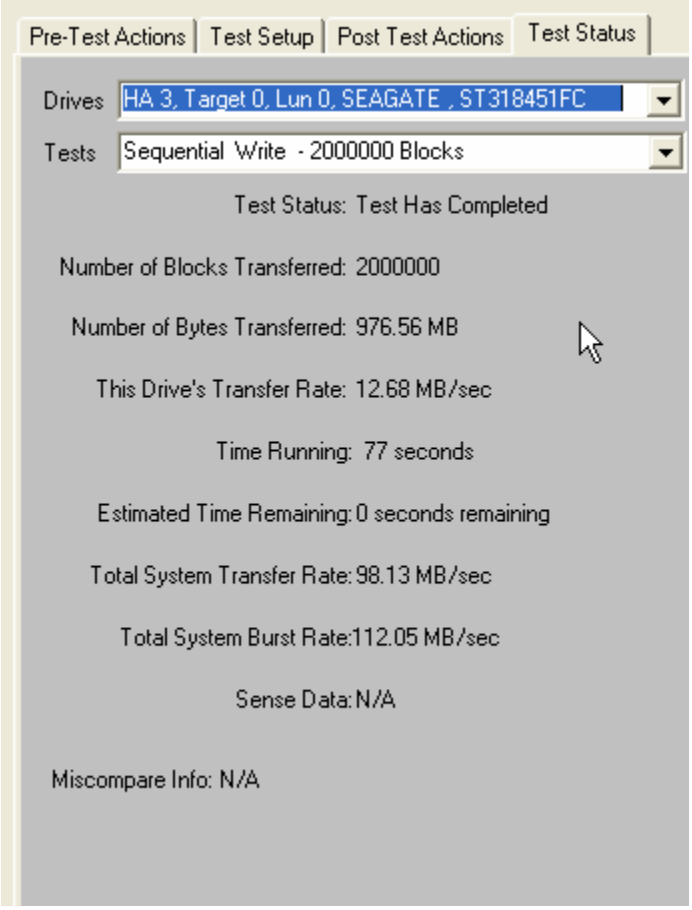
During Testing – All Drives

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.

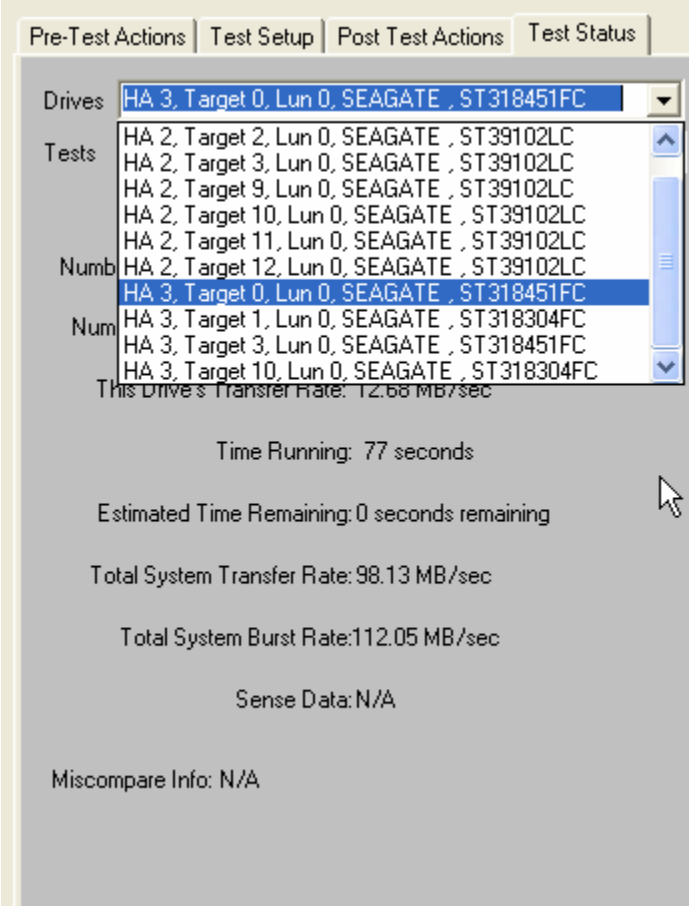
The Test Status Page

The test status pages appears as below:



The Drives Pulldown

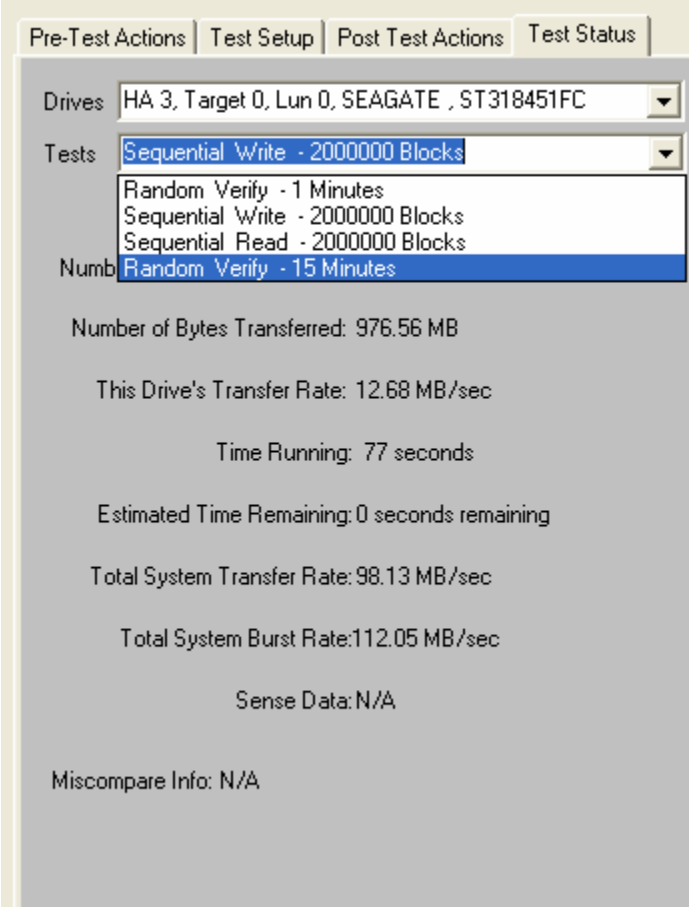
Use the Drives Pulldown box to choose which disk



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:

ComputerNam e
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

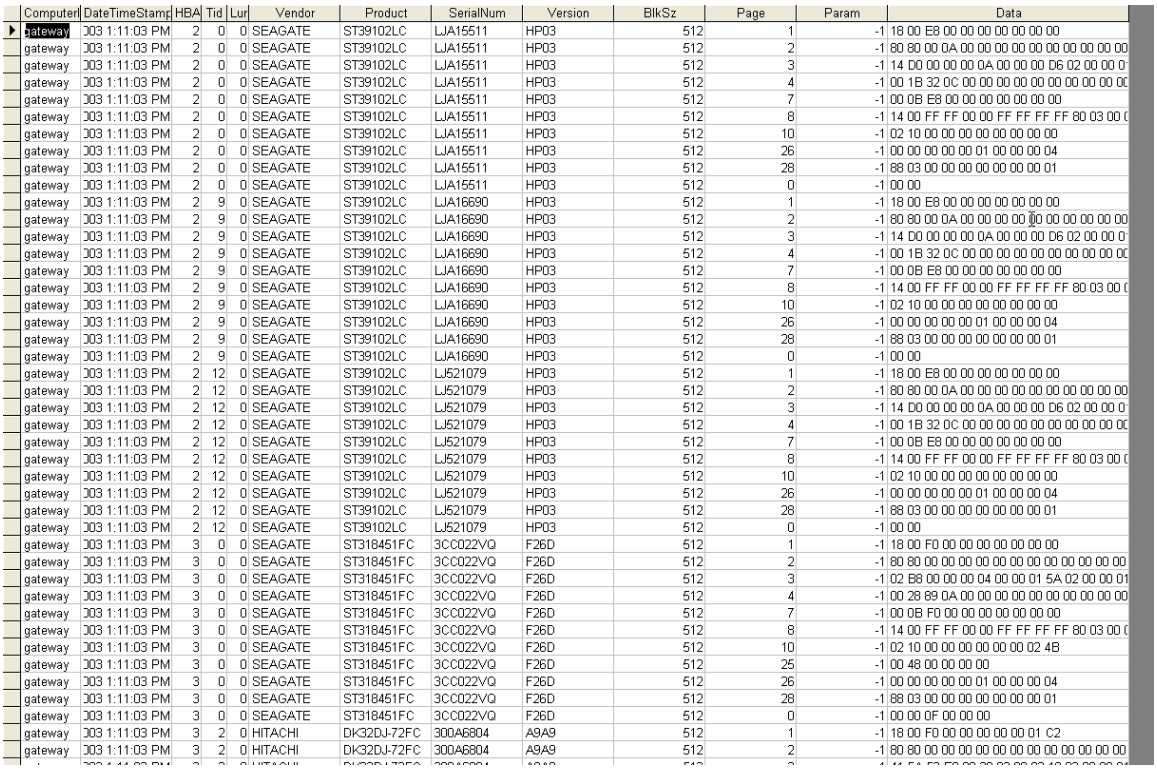
Here is a (partial!) screen shot of the test table:

ComputerName	DateTimeStamp	HBA	Tid	Lun	Vendor	Product	SerialNum	Version	BlkSz	Capacity	Adapter	Comment	TestName	Pattern	T
Starting Test Se	003 1:10:59 PM	-1	-1	-1	****	****	****	****	-1	-1	****	****	****	****	****
gateway	003 1:11:03 PM	2	0	0	SEAGATE	ST39102LC	LJA15511	HP03	512	17664228	adpu160m	Test Started	Random Verify	Decrementing	Tes
gateway	003 1:11:03 PM	2	9	0	SEAGATE	ST39102LC	LJA16690	HP03	512	17783111	adpu160m	Test Started	Random Verify	Decrementing	Tes
gateway	003 1:11:03 PM	2	12	0	SEAGATE	ST39102LC	LJ521079	HP03	512	17783111	adpu160m	Test Started	Random Verify	Decrementing	Tes
gateway	003 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3CC022VQ	F26D	512	35566477	ql2100	Test Started	Random Verify	Decrementing	Tes
gateway	003 1:11:03 PM	3	2	0	HITACHI	DK32DJ-72FC	300A6804	A9A9	512	144410879	ql2100	Test Started	Random Verify	Decrementing	Tes
gateway	003 1:11:38 PM	3	2	0	HITACHI	DK32DJ-72FC	300A6804	A9A9	512	144410879	ql2100	Test Ended	Random Verify	Decrementing	Tes
gateway	003 1:11:38 PM	3	0	0	SEAGATE	ST318451FC	3CC022VQ	F26D	512	35566477	ql2100	Test Ended	Random Verify	Decrementing	Tes
gateway	003 1:11:38 PM	2	12	0	SEAGATE	ST39102LC	LJ521079	HP03	512	17783111	adpu160m	Test Ended	Random Verify	Decrementing	Tes
gateway	003 1:11:38 PM	2	9	0	SEAGATE	ST39102LC	LJA16690	HP03	512	17783111	adpu160m	Test Ended	Random Verify	Decrementing	Tes
gateway	003 1:11:38 PM	2	0	0	SEAGATE	ST39102LC	LJA15511	HP03	512	17664228	adpu160m	Test Ended	Random Verify	Decrementing	Tes
Ending Test Se	003 1:11:39 PM	-1	-1	-1	****	****	****	****	-1	-1	****	****	****	****	****
Starting Test Se	003 1:16:34 PM	-1	-1	-1	****	****	****	****	-1	-1	****	****	****	****	****
gateway	003 1:16:50 PM	2	0	0	SEAGATE	ST39102LC	LJA15511	HP03	512	17664228	adpu160m	Test Started	Random Write/	Random	Tes
gateway	003 1:16:50 PM	2	2	0	SEAGATE	ST39102LC	LJB64426	HP03	512	17783111	adpu160m	Test Started	Random Write/	Random	Tes
gateway	003 1:16:50 PM	2	3	0	SEAGATE	ST39102LC	LJA04046	HP03	512	17783111	adpu160m	Test Started	Random Write/	Random	Tes
gateway	003 1:16:50 PM	2	9	0	SEAGATE	ST39102LC	LJA16690	HP03	512	17783111	adpu160m	Test Started	Random Write/	Random	Tes
gateway	003 1:16:50 PM	2	10	0	SEAGATE	ST39102LC	LJB93435	HP03	512	17783111	adpu160m	Test Started	Random Write/	Random	Tes
gateway	003 1:16:50 PM	2	11	0	SEAGATE	ST39102LC	LJA02465	HP03	512	17783111	adpu160m	Test Started	Random Write/	Random	Tes
gateway	003 1:16:50 PM	2	12	0	SEAGATE	ST39102LC	LJ521079	HP03	512	17783111	adpu160m	Test Started	Random Write/	Random	Tes
gateway	003 1:16:50 PM	3	0	0	SEAGATE	ST318451FC	3CC022VQ	F26D	512	35566477	ql2100	Test Started	Random Write/	Random	Tes
gateway	003 1:16:50 PM	3	1	0	SEAGATE	ST318304FC	3EL005MD	0001	512	35964300	ql2100	Test Started	Random Write/	Random	Tes
gateway	003 1:16:50 PM	3	2	0	HITACHI	DK32DJ-72FC	300A6804	A9A9	512	144410879	ql2100	Test Started	Random Write/	Random	Tes
gateway	003 1:16:50 PM	3	3	0	SEAGATE	ST318451FC	3CC01XJ4	F26D	512	35566477	ql2100	Test Started	Random Write/	Random	Tes
gateway	003 1:16:50 PM	3	4	0	IBM	DDYF-T18350R		F61L	512	35566477	ql2100	Test Started	Random Write/	Random	Tes
gateway	003 1:16:50 PM	3	5	0	IBM	DDYF-T18350S		W60V	512	35566477	ql2100	Test Started	Random Write/	Random	Tes
gateway	003 1:16:50 PM	3	6	0	IBM	DDYF-T18350S		W60V	512	35566477	ql2100	Test Started	Random Write/	Random	Tes
gateway	003 1:16:50 PM	3	7	0	HITACHI	DK32DJ-72FC	300A3589	A9A9	512	144410879	ql2100	Test Started	Random Write/	Random	Tes
gateway	003 1:16:50 PM	3	8	0	HITACHI	DK32DJ-72FC	300A2818	A9A9	512	144410879	ql2100	Test Started	Random Write/	Random	Tes
gateway	003 1:16:50 PM	3	9	0	IBM	DDYF-T18350S		W60V	512	35566477	ql2100	Test Started	Random Write/	Random	Tes
gateway	003 1:16:50 PM	3	10	0	SEAGATE	ST318304FC	3EL005NC	0001	512	35964300	ql2100	Test Started	Random Write/	Random	Tes
gateway	003 1:16:50 PM	3	11	0	HITACHI	DK32DJ-72FC	300A0594	A9A9	512	144410879	ql2100	Test Started	Random Write/	Random	Tes
gateway	003 1:17:11 PM	3	8	0	HITACHI	DK32DJ-72FC	300A2818	A9A9	512	144410879	ql2100	Test Ended	Random Write/	Random	Mis
gateway	003 1:18:02 PM	3	2	0	HITACHI	DK32DJ-72FC	300A6804	A9A9	512	144410879	ql2100	Test Ended	Random Write/	Random	Mis
gateway	003 1:21:22 PM	3	3	0	SEAGATE	ST318451FC	3CC01XJ4	F26D	512	35566477	ql2100	Test Ended	Random Write/	Random	Tes
gateway	003 1:21:24 PM	3	0	0	SEAGATE	ST318451FC	3CC022VQ	F26D	512	35566477	ql2100	Test Ended	Random Write/	Random	Tes
gateway	003 1:22:31 PM	3	4	0	IBM	DDYF-T18350R		F61L	512	35566477	ql2100	Test Ended	Random Write/	Random	Tes
gateway	003 1:22:35 PM	3	9	0	IBM	DDYF-T18350S		W60V	512	35566477	ql2100	Test Ended	Random Write/	Random	Tes
gateway	003 1:22:36 PM	3	10	0	SEAGATE	ST318304FC	3EL005NC	0001	512	35964300	ql2100	Test Ended	Random Write/	Random	Tes
gateway	003 1:22:36 PM	3	1	0	SEAGATE	ST318304FC	3EL005MD	0001	512	35964300	ql2100	Test Ended	Random Write/	Random	Tes
gateway	003 1:22:44 PM	3	7	0	HITACHI	DK32DJ-72FC	300A3589	A9A9	512	144410879	ql2100	Test Ended	Random Write/	Random	Tes
gateway	003 1:22:50 PM	3	11	0	HITACHI	DK32DJ-72FC	300A0594	A9A9	512	144410879	ql2100	Test Ended	Random Write/	Random	Tes
gateway	003 1:23:02 PM	3	5	0	IBM	DDYF-T18350S		W60V	512	35566477	ql2100	Test Ended	Random Write/	Random	Tes

Here is the rest of the test table:

TimeToDoTest	NumMBToDev	NumMBFromDe	TotMBTrans	TransRateTo	TransRateFrom	TotTransRate	SenseOnFailure	MiscompareLBA	MiscompareOffs	MiscompareExp	MiscompareAct
****	-1	-1	-1	-1	-1	-1	****	-1	-1	****	Starting Test Se
N/A	0	0	0	0	0	0	N/A	0	0		
N/A	0	0	0	0	0	0	N/A	0	0		
N/A	0	0	0	0	0	0	N/A	0	0		
N/A	0	0	0	0	0	0	N/A	0	0		
N/A	0	0	0	0	0	0	N/A	0	0		
35sec	0	124.81	124.81	0	3.57	3.57	N/A	0	0		
35sec	0	231.88	231.88	0	6.63	6.63	N/A	0	0		
35sec	0	153.94	153.94	0	4.4	4.4	N/A	0	0		
35sec	0	154.63	154.63	0	4.42	4.42	N/A	0	0		
35sec	0	154.56	154.56	0	4.42	4.42	N/A	0	0		
****	-1	-1	-1	-1	-1	-1	****	-1	-1	****	Ending Test Se
****	-1	-1	-1	-1	-1	-1	****	-1	-1	****	Starting Test Se
N/A	0	0	0	0	0	0	N/A	0	0		
N/A	0	0	0	0	0	0	N/A	0	0		
N/A	0	0	0	0	0	0	N/A	0	0		
N/A	0	0	0	0	0	0	N/A	0	0		
N/A	0	0	0	0	0	0	N/A	0	0		
N/A	0	0	0	0	0	0	N/A	0	0		
N/A	0	0	0	0	0	0	N/A	0	0		
N/A	0	0	0	0	0	0	N/A	0	0		
N/A	0	0	0	0	0	0	N/A	0	0		
N/A	0	0	0	0	0	0	N/A	0	0		
N/A	0	0	0	0	0	0	N/A	0	0		
N/A	0	0	0	0	0	0	N/A	0	0		
N/A	0	0	0	0	0	0	N/A	0	0		
N/A	0	0	0	0	0	0	N/A	0	0		
N/A	0	0	0	0	0	0	N/A	0	0		
N/A	0	0	0	0	0	0	N/A	0	0		
N/A	0	0	0	0	0	0	N/A	0	0		
N/A	0	0	0	0	0	0	N/A	0	0		
N/A	0	0	0	0	0	0	N/A	0	0		
21sec	50.75	50.69	101.44	2.42	2.41	4.83	N/A	42515868	19	73 1d b5 42 29	fc 1d b5 42 29 4
1min 12sec	183.63	183.56	367.19	2.55	2.55	5.1	N/A	11285515	24	90 38 16 a2 e9	9a 38 16 02 ad
4min 32sec	976.56	976.56	1953.13	3.59	3.59	7.18	N/A	0	0		
4min 34sec	976.56	976.56	1953.13	3.56	3.56	7.13	N/A	0	0		
5min 41sec	976.56	976.56	1953.13	2.86	2.86	5.73	N/A	0	0		
5min 45sec	976.56	976.56	1953.13	2.83	2.83	5.66	N/A	0	0		
5min 46sec	976.56	976.56	1953.13	2.82	2.82	5.64	N/A	0	0		
5min 46sec	976.56	976.56	1953.13	2.82	2.82	5.64	N/A	0	0		
5min 54sec	976.56	976.56	1953.13	2.76	2.76	5.52	N/A	0	0		
6min 0sec	976.56	976.56	1953.13	2.71	2.71	5.43	N/A	0	0		
6min 12sec	976.56	976.56	1953.13	2.63	2.63	5.25	N/A	0	0		

The Information table stores Mode and Log page information – again a screen shot:



ComputerN	DateTimeStamp	HBA	Tid	Lun	Vendor	Product	SerialNum	Version	BlkSz	Page	Param	Data
gateway	303 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	303 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 00 00
gateway	303 1:11:03 PM	2	0	0	SEAGATE	ST39102LC	LJA15511	HP03	512	3	-1	14 D0 00 00 00 0A 00 00 00 D6 02 00 00 0
gateway	303 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 0
gateway	303 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	303 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 0
gateway	303 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	303 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	303 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	303 1:11:03 PM	2	0	0	SEAGATE	ST39102LC	LJA15511	HP03	512	0	-1	00 00
gateway	303 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	303 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 00 00
gateway	303 1:11:03 PM	2	9	0	SEAGATE	ST39102LC	LJA16690	HP03	512	3	-1	14 D0 00 00 00 0A 00 00 00 D6 02 00 00 0
gateway	303 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 0
gateway	303 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	303 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 0
gateway	303 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	303 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	303 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	303 1:11:03 PM	2	9	0	SEAGATE	ST39102LC	LJA16690	HP03	512	0	-1	00 00
gateway	303 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	303 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 00 00
gateway	303 1:11:03 PM	2	12	0	SEAGATE	ST39102LC	LJ521079	HP03	512	3	-1	14 D0 00 00 00 0A 00 00 00 D6 02 00 00 0
gateway	303 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 0
gateway	303 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	303 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 0
gateway	303 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	303 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	303 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	303 1:11:03 PM	2	12	0	SEAGATE	ST39102LC	LJ521079	HP03	512	0	-1	00 00
gateway	303 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3CC022VQ	F26D	512	1	-1	18 00 F0 00 00 00 00 00 00 00
gateway	303 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3CC022VQ	F26D	512	2	-1	80 80 00 00 00 00 00 00 00 00 00 00 00 00 00 00
gateway	303 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3CC022VQ	F26D	512	3	-1	02 B8 00 00 00 04 00 00 01 5A 02 00 00 01
gateway	303 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3CC022VQ	F26D	512	4	-1	00 28 89 0A 00 00 00 00 00 00 00 00 00 00
gateway	303 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3CC022VQ	F26D	512	7	-1	00 0B F0 00 00 00 00 00 00 00
gateway	303 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3CC022VQ	F26D	512	8	-1	14 00 FF FF 00 00 FF FF FF FF 80 03 00 0
gateway	303 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3CC022VQ	F26D	512	10	-1	02 10 00 00 00 00 00 00 02 4B
gateway	303 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3CC022VQ	F26D	512	25	-1	00 48 00 00 00 00
gateway	303 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3CC022VQ	F26D	512	26	-1	00 00 00 00 00 01 00 00 00 04
gateway	303 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3CC022VQ	F26D	512	28	-1	88 03 00 00 00 00 00 00 00 01
gateway	303 1:11:03 PM	3	0	0	SEAGATE	ST318451FC	3CC022VQ	F26D	512	0	-1	00 00 0F 00 00 00
gateway	303 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	303 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 00 00 00

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

