SCSI*toolbox*™ 32

What's New in Version 7

Copyright © 2003, 2004 SCSI Toolbox, LLC.
All rights reserved.

What's new in Version 7

SCSItoolbox32 (STB) Additional Features & Tests

Disk Performance/Error Graphing Test

This test is accessed from the ***Disk->Tests->Performance/Error Graph*** menu choice.

This test enables you to quantify the real-world I/O performance of a disk drive. It supplies a macro view of the entire drive that also provides a micro view of all problem areas. In the case of disk read performance, you need to read all blocks on the drive and record the time each read takes to complete. The shorter the completion times the better. To visualize this data it is graphed as block-number versus I/O Completion time.

Look at the graph in Figure 1 – the X axis is the block number read and the Y axis is the time it took the READ I/O to complete. The Red plot shows the slowest I/O time out the past 100 reads. The Green plot shows the average completion time of the last 100 reads, and the Blue plot shows the average of all reads.

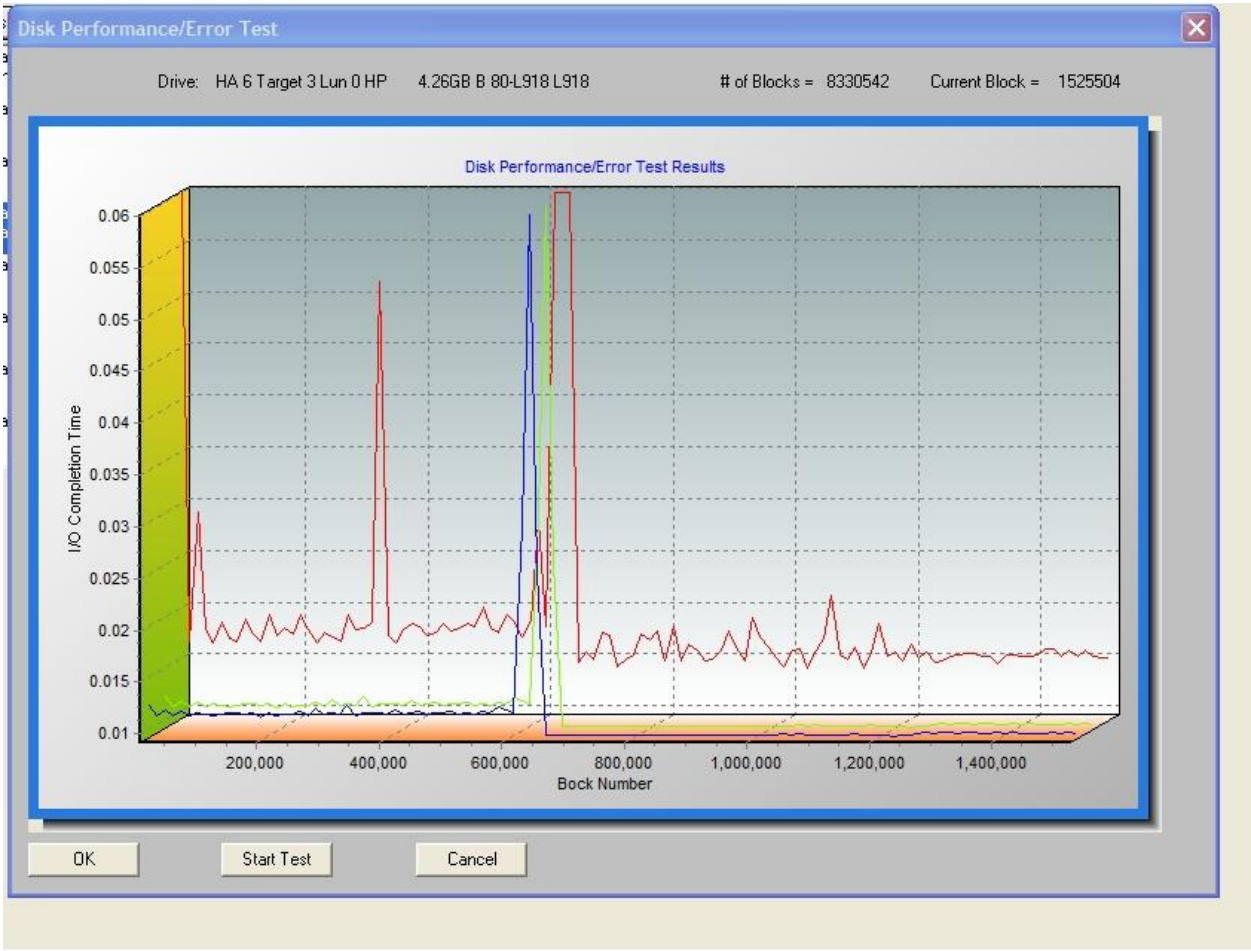


Figure 1

Notice the large difference between the slowest I/O completion time and the average. In this case something is causing the drive to take much longer to complete the read. Also notice the correlation between the Red and Green plots. Since the Green plots show the average completion time across 100 I/O's these plots will indicate how many of the I/O's are slow.

For example at block number 400,000 the **Red** plot showed .05 seconds, and the Green shows around .012. This tells us that there were just a few blocks within a 100 block range that had trouble. On the other hand, at block 600,000 we see the **Red** plot peaking again around .06, while the Green average is almost the same. This would indicate that most if not all the blocks in this region are having trouble.

What causes slow I/O completion times? Usually it stems from the drive having to do varying degrees of error correction. We recommend looking at two suspect areas, *grown defects* and *error correction* counts.

Figure 2 shows the Primary defect list of the drive, Figure 3 shows there are no Grown defects

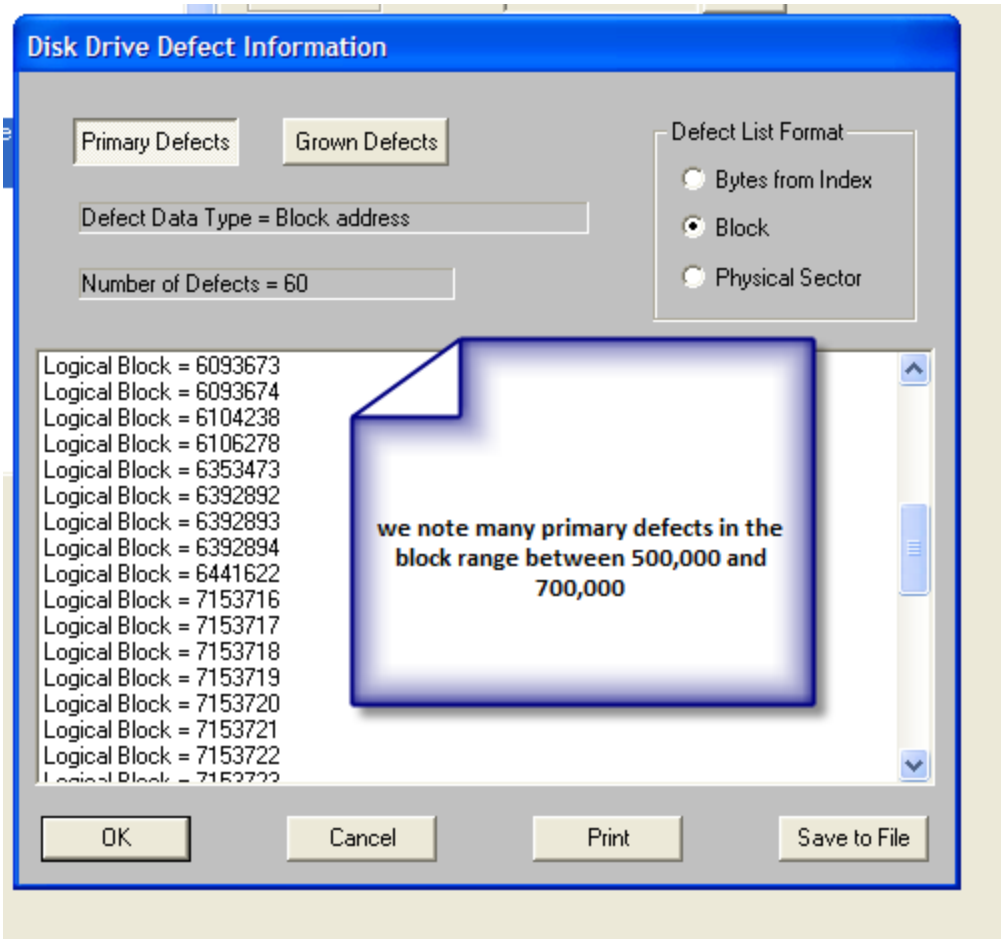


Figure 2

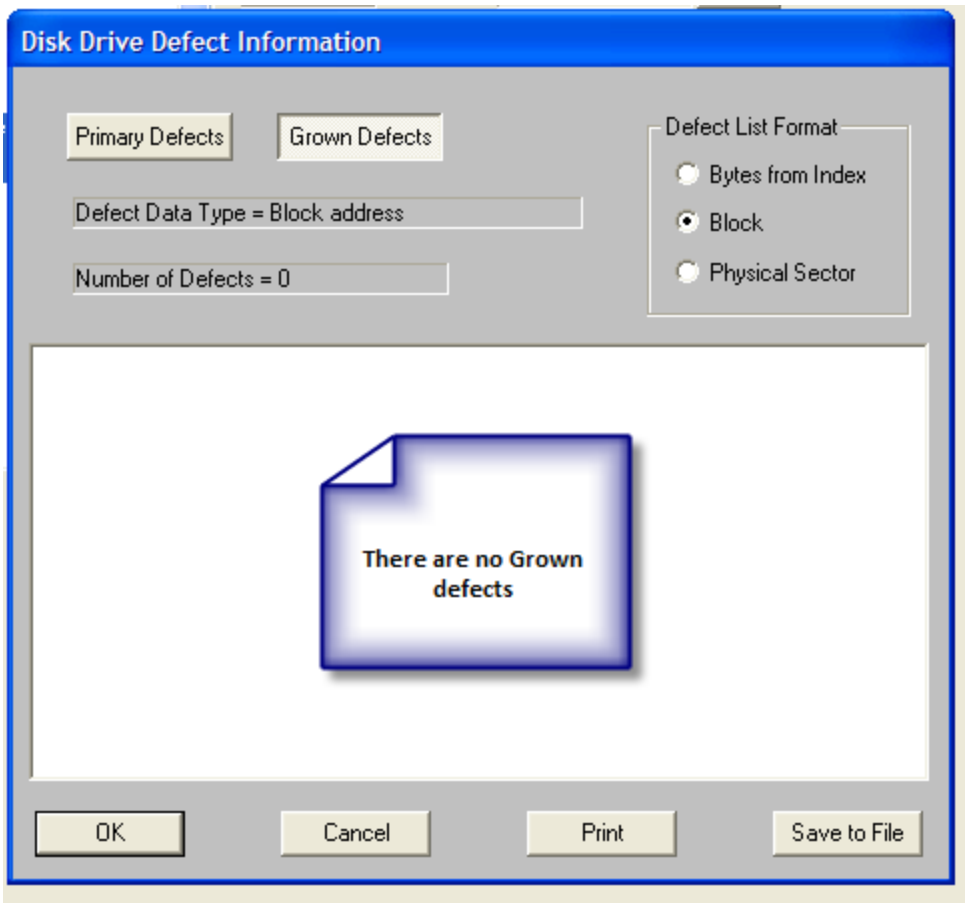


Figure 3

This implies that the problem is *not* related to the drive accumulating defects. That leaves the question of is the drive having to do excessive amounts of error correction which in turn is slowing command execution? To confirm this you must examine the Error Correction logs for the drive. A summary of these logs is shown in Figure 4.

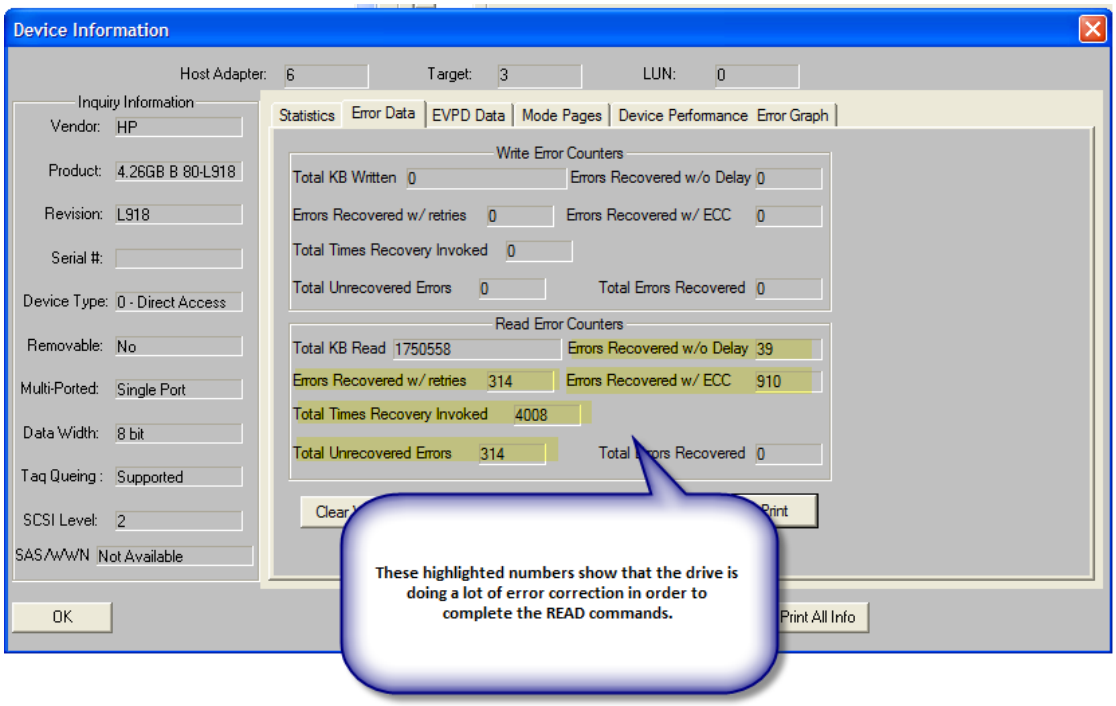


Figure 4

As noted in Figure 4, the drive is recording many read retry efforts.

What causes retries? Why would the drive need to retry the read operation? It could be a number of issues, from media damage due to a head crash, wear and aging of the media, servo tracking problems possibly cause by temperature extremes or power loss during writes, etc. Retries over a very large block range may indicate disk head or preamp problems.

To verify if the problem stems from a permanent problem or not we recommend two steps –

1. Reformat the drive – this will erase any grown defects and replace them if the format media scan shows a problem.
2. Write to all blocks of the drive. This will force the drive to update both sector data and error correction code data.

After performing these two steps run the Performance/Error graph test again and note any differences – as Figure 5 shows the problem area still has a few blocks that are reading slow, but the average I/O completion time is now low. Again, this indicates that only a few blocks are having problems.

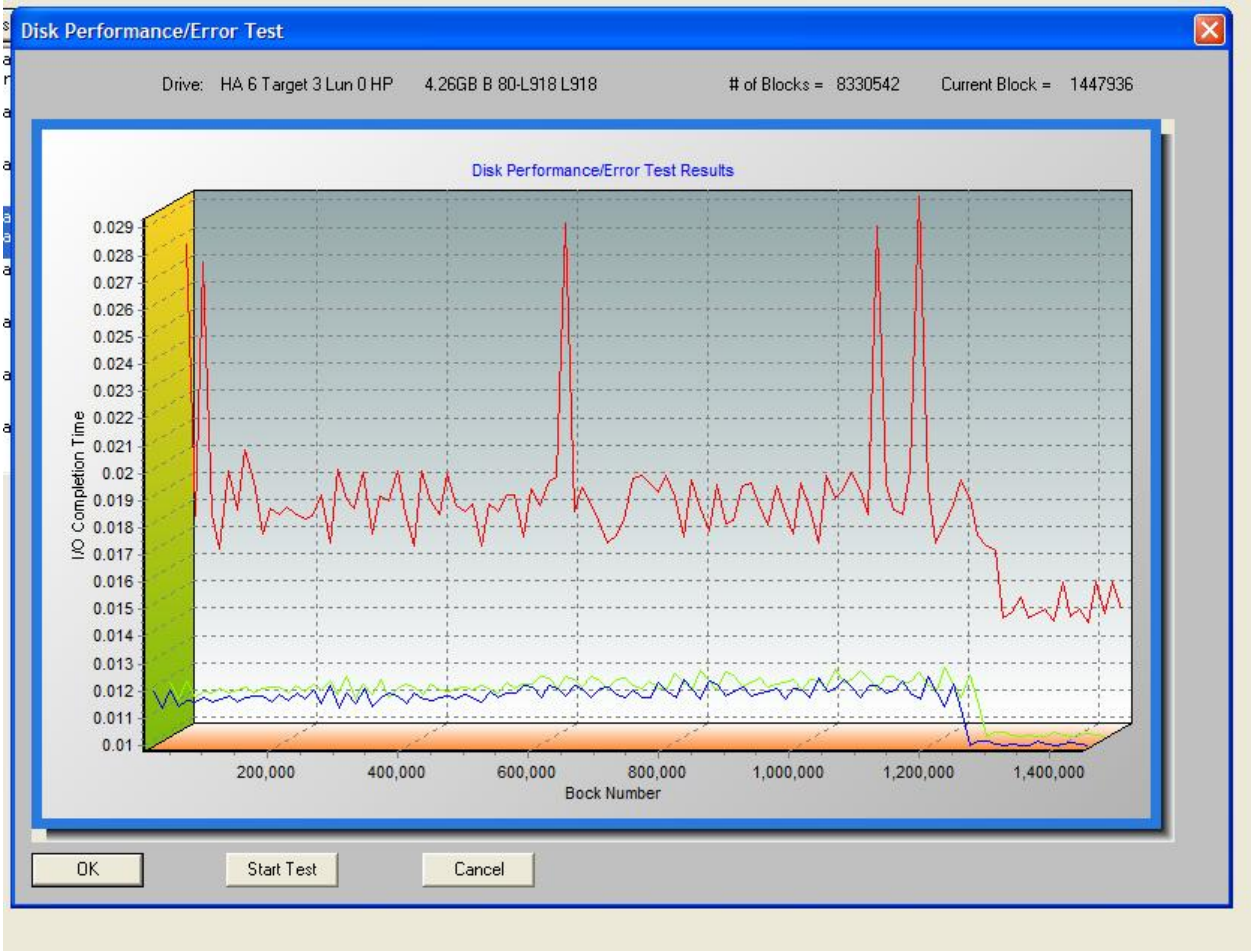
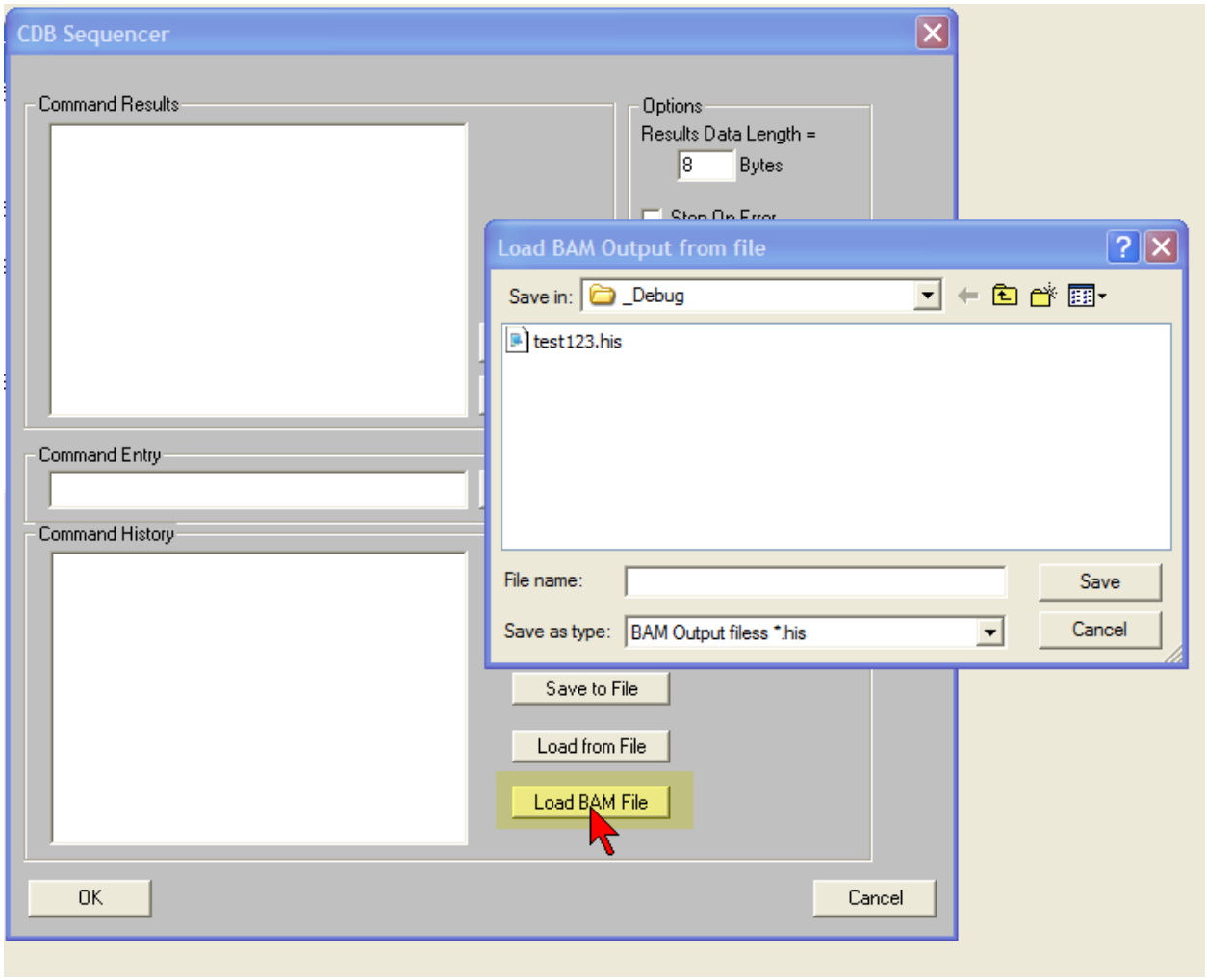


Figure 5

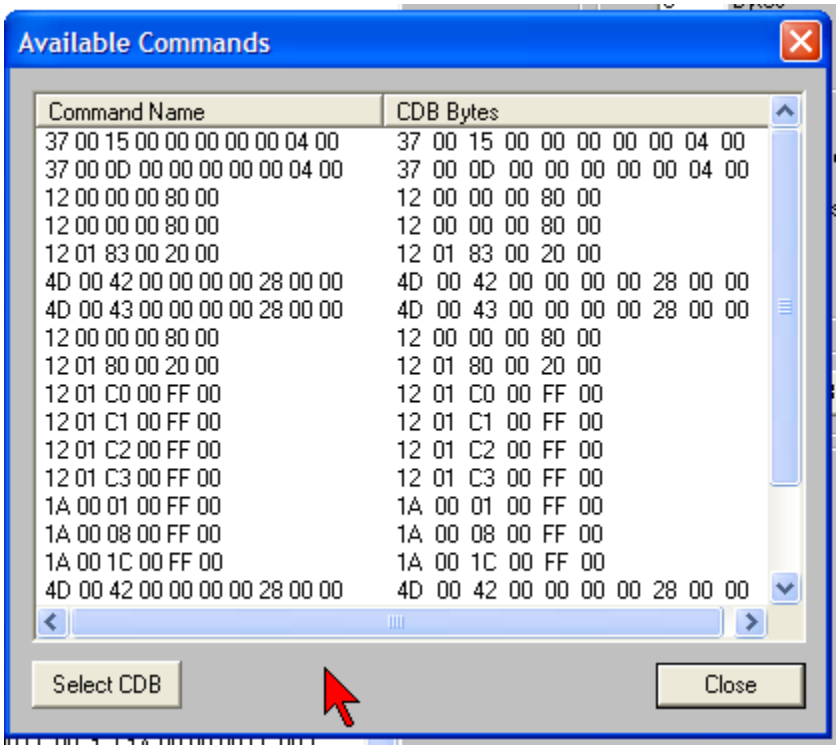
Checking the drive Error Correction logs in Figure 6 now shows fewer corrections are happening – the format/write process was able to repair the problem. It would still be prudent to watch the drive – occasionally run the same test and log examination sequence to be sure that the problem was indeed temporary.

Load/Execute BAM Trace in CDB Sequencer

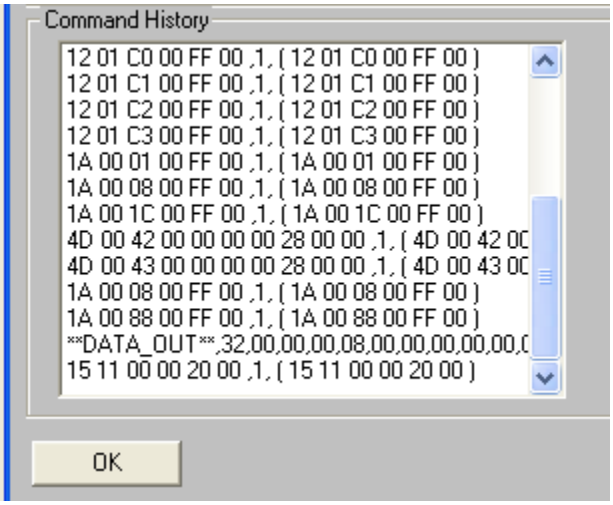
This new feature gives you the ability to issue a series of CDB's which were captured using BAM. It is accessed by the new CDB Sequencer ***Load BAM File*** choice – specify a file previously made using the BAM ***Save to CDB Seq File*** option:



This will create a new command definition file and new history entries in the Command History Window. An example of the new command definition file is here:



An example of the new history entry is here:



Select the entire CDB sequence by first clicking on the first entry in the Command History window, then Shift-Clicking on the final entry in the Command History window.

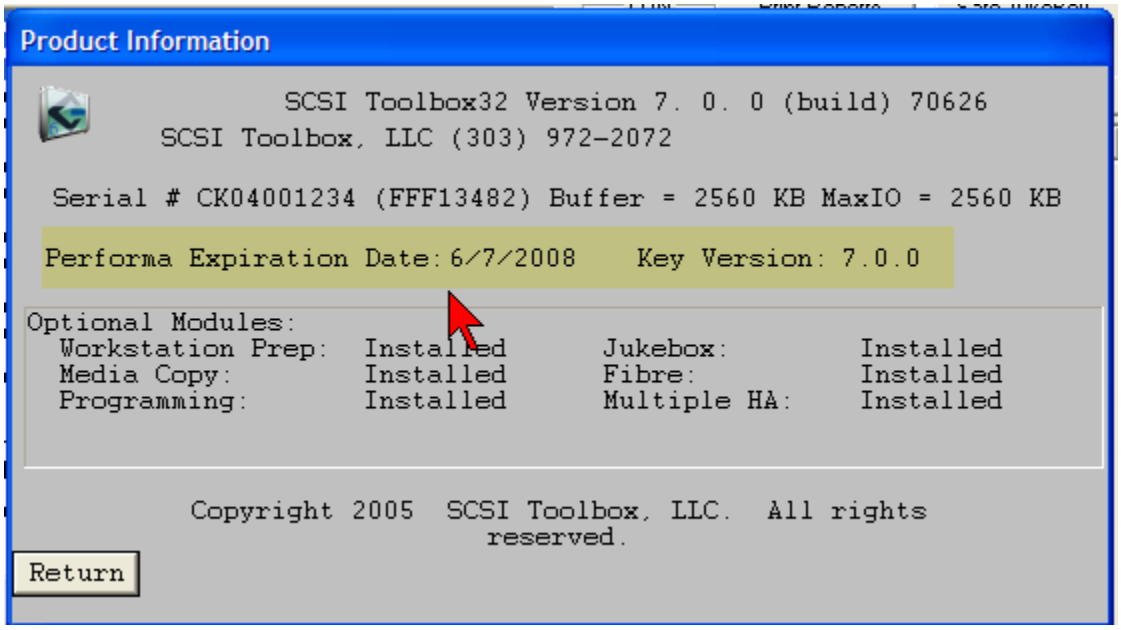
Once the commands have been selected, click the ***Execute*** button to send the sequence of commands to the selected device.

The results of each CDB will be displayed in the Command Results window. I/O completion time, status, and REQUEST SENSE data if a CHECK CONDITION occurs will be displayed.

See the description of Added Save to CDB Sequencer File option in this section for instructions on creating the BAM file.

Display Performa Expiration Date

The Performa Expiration Date and version number are now displayed when you click the ***Help->About SCSItoolbox*** main menu choice as shown here:



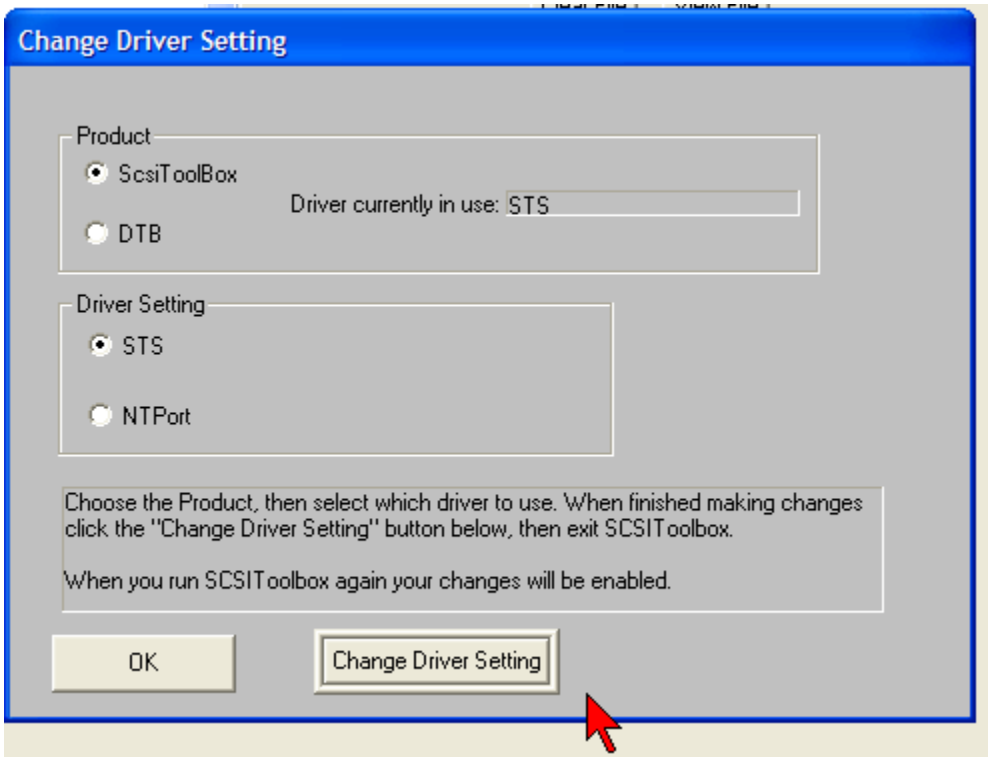
Be sure to keep your Performa subscription up to date so you will have access to the latest features of the SCSItoolbox Suite!

Default Driver Selection Option

The main menu Options->Change Driver choice allows you to easily change between the available drivers for SCSItoolbox and Developers Toolbox.

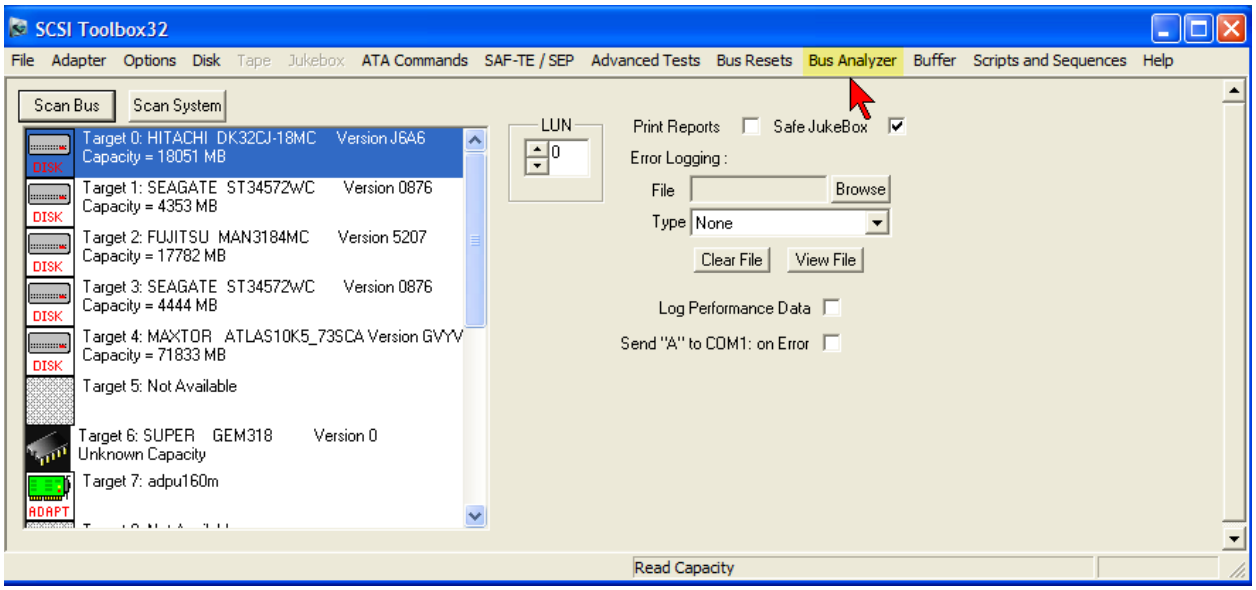
You may want to try changing the default driver if you are having issues where certain devices are not showing up in the device display.

If you change drivers using this option you will need to exit and then re-launch SCSItoolbox for the change to take effect.



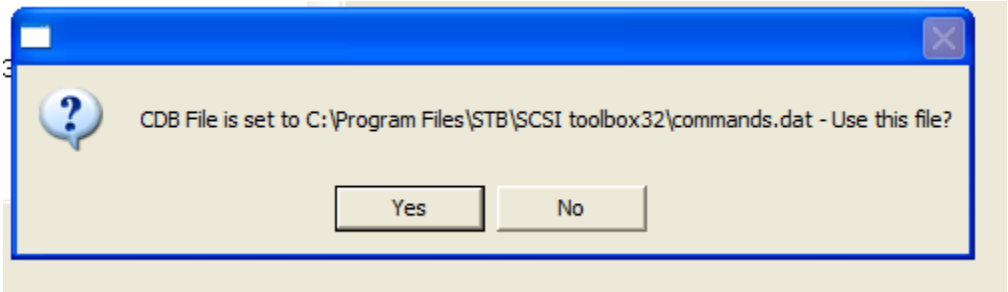
Launch Bus Analyzer from within STB

Use this main menu choice to launch BAM while running the SCSItoolbox

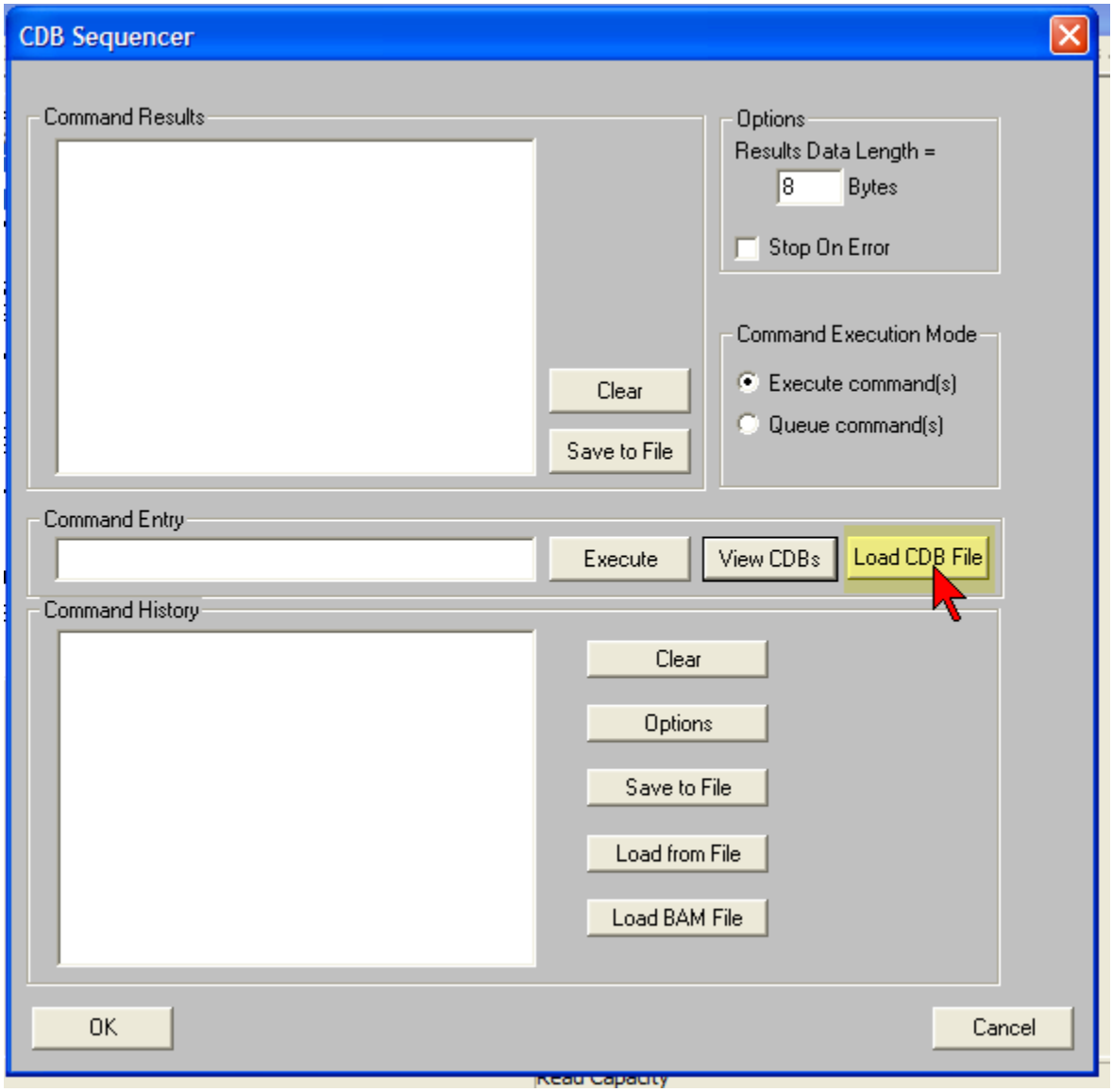


Command description file choice added to CDB Sequencer

When first entering the CDB Sequencer you will see the name of the command description file that will be used. You can accept this file, or you can specify another file –



Within the CDB Sequencer you may also change the command description file that is used –



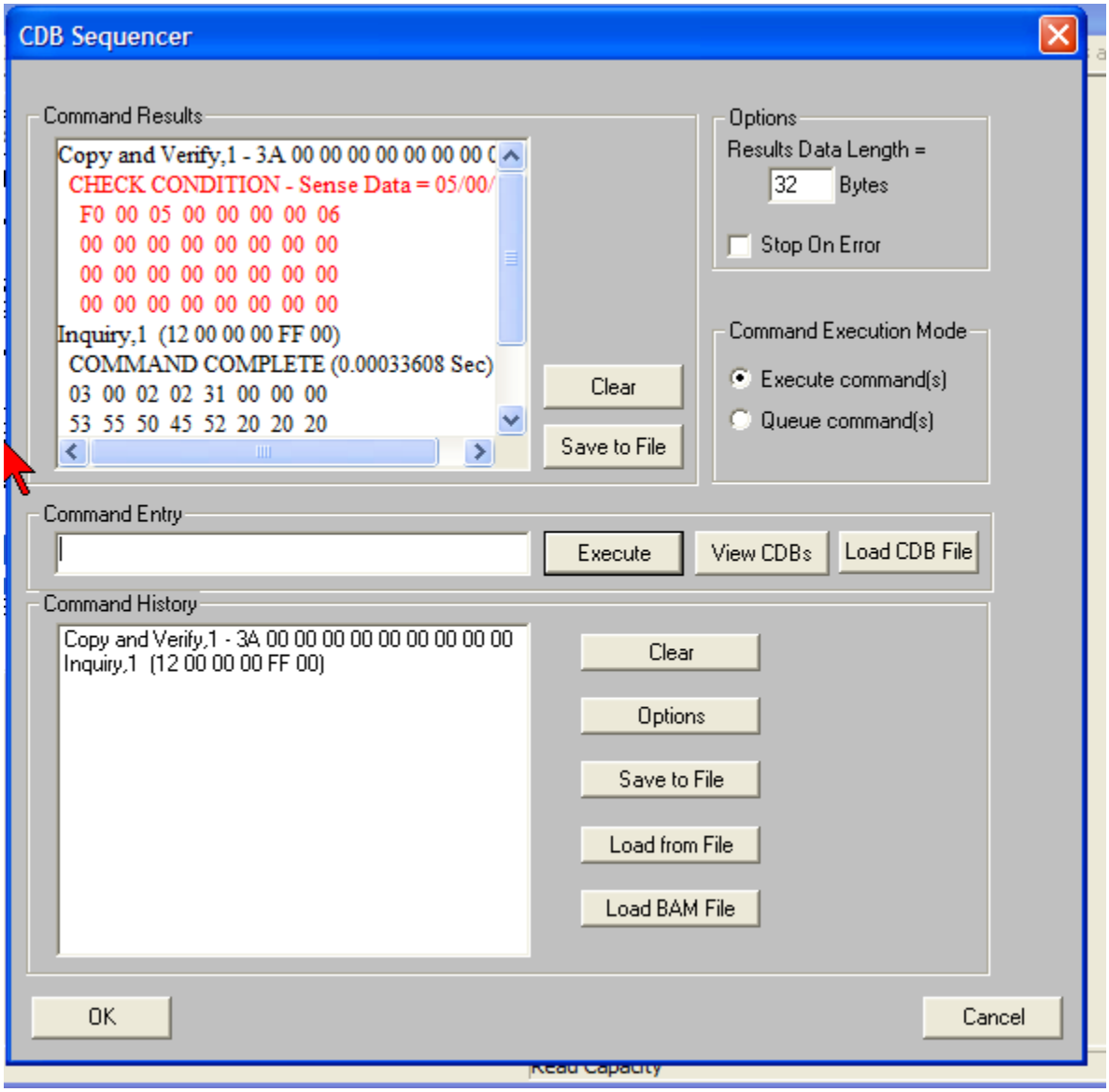
The CDB description files are ascii text files that can be created/edited with any text editor. They can also be created or modified by using the ***User Defined CDB*** command.

Improved Device Discovery method

A newer method of device discover has been implemented that is able to find devices in systems with non-standard host bus adapter configurations.

Display CHECK CONDITION in **Red** in CDB Sequencer

If a command sent in the CDB Sequencer results in a CHECK CONDITION the results in the Command Response window will be displayed in Red, making it easier to see when errors have occurred



SCSItoolbox32 (STB) Bug Fixes

Fixed READ DEFECT DATA by Logical Block

Fixed Dongle Serial Number display

Fixed 16-byte READ CAPACITY

Fixed display of EVPD SERIAL NUMBER data

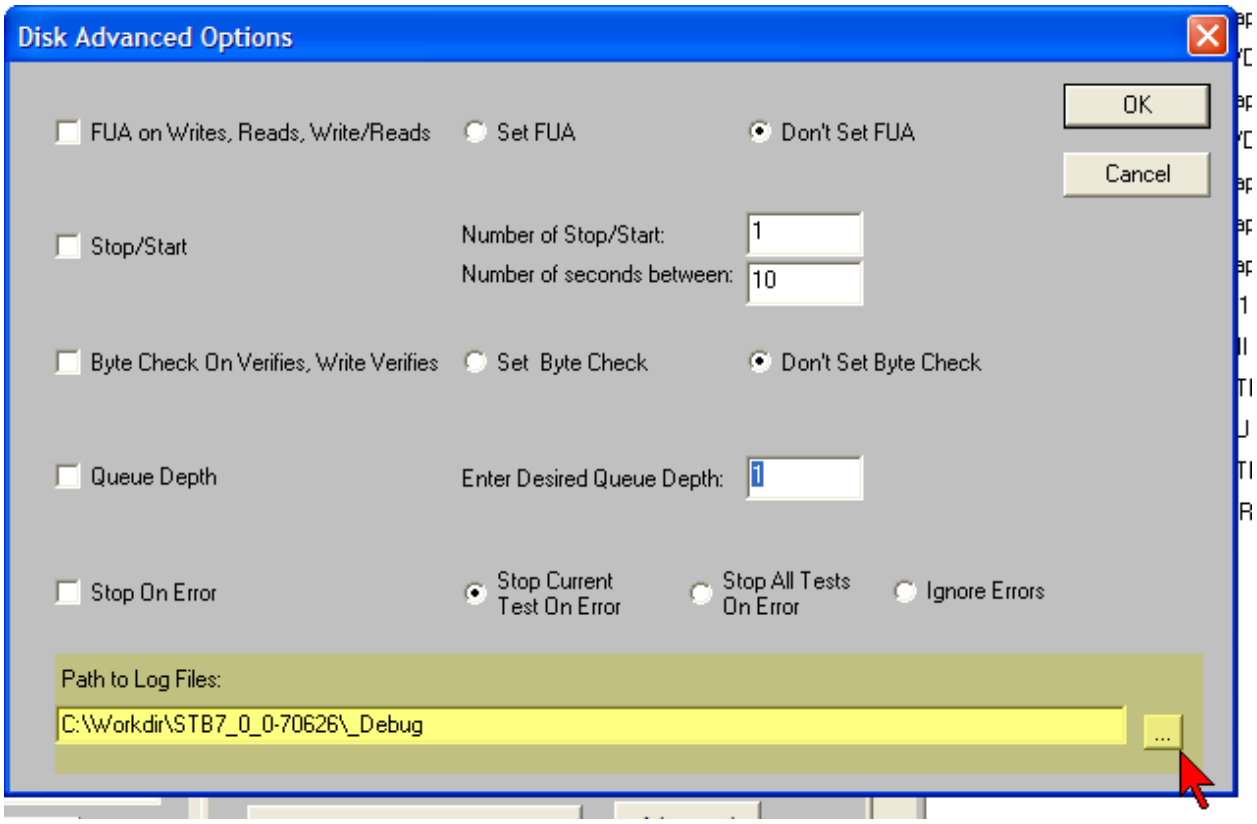
Fixed Log Page description file parsing

Manufacturing Module (DMM,TMM) Additional Features & Tests

Added Advanced Options choice to specify location of log files

Open the Advanced Options page by clicking on the ***Advanced Options*** button in the ***Test Setup*** tab.

The default folder that all log files will be written to is displayed. Click on the Browse button at the far right of the folder display area to specify another location for the log files.



Manufacturing Module (DMM,TMM) Bug Fixes

Changed file share status of DMM text logfile

The file status is changed to *shared* - so other programs can open/print from the text log file while a test sequence is running.

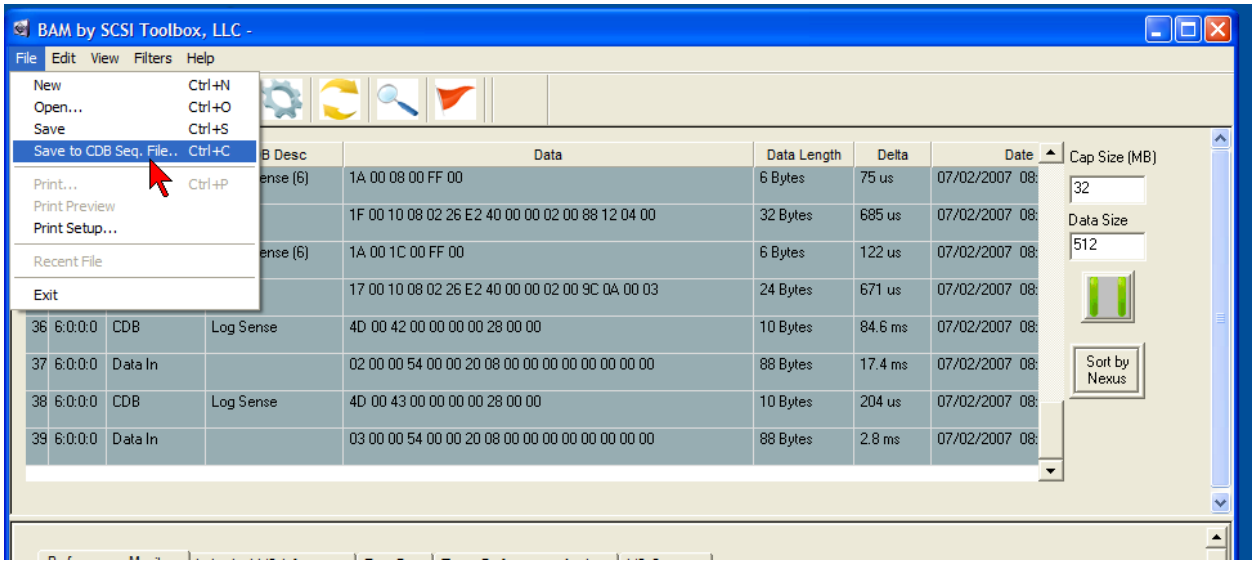
Bus Analyzer Module (BAM) Additional Features & Tests

Added Save to CDB Sequencer File option

You can now record a series of I/O's to/from a device, and later use the STB ***CDB Sequencer*** function to play back the sequence of commands. This is very useful for capturing and recreating problematic I/O streams – you can capture I/O that is generated by any application.

Be sure that you have only captured I/O to one device at a time when using this option to prevent contaminating I/O for one device with those commands sent to another device.

Once you have captured an I/O trace use the ***File->Save to CDB Seq. File*** choice to create the file

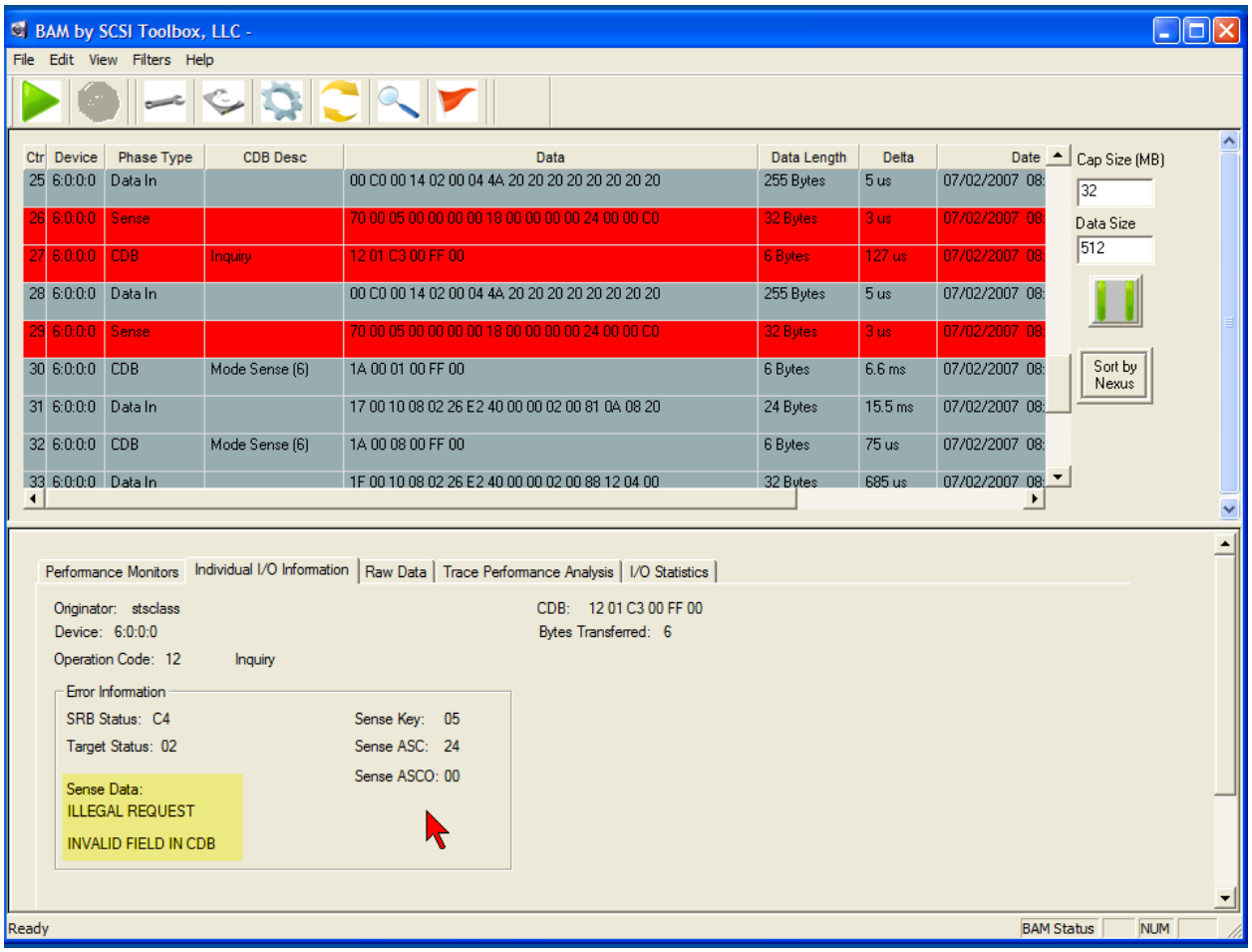


Improved Trace Save operation

The File->Save choice has been updated to save all phase data from a trace capture.

Added Sense Data interpretation to Individual I/O display tab

In the case of a CHECK CONDITION the Sense Key, Sense Code, and Sense ASQ hex data is interpreted as shown –



Bus Analyzer Module (BAM) Bug Fixes

Made all functions (Find, Individual I/O, Raw Data, Analysis) work on saved file data as they do with live data