

STORAGWERKS

DataMover Storage

User Guide

DMS Development Group, v3.3.8

4/15/2021

Contents

Introduction	12
Screen Shot	12
Quick Start Guide	13
Basic Dialog Operation.....	14
Menu Items	14
Select Logical Drives.....	14
Select Physical Drives.....	14
Advanced Patterns	14
Advanced Features Logic	14
Advanced Features Physical.....	14
GUI Items	14
Status	14
Location.....	14
I/O Direction.....	15
Threads.....	15
Pattern	15
Start.....	15
Stop	15
Help	15
Advanced Dialog Operation	15
Menu Items.....	15
License Import.....	15
License Register	15
License Unregister.....	16
Options Import Data File.....	16
Options Performance Test	16
Options Quick Performance Test	16
More Performance Tests Quick Walking Test.....	16
More Performance Tests Long Walking Test	16
More Performance Tests Custom Profile Test.....	16
Options Clear Modifications	16

Options Target Test Standalone Mode	16
Options Target Test Cluster Mode	16
Options Target Test Random Standalone Mode.....	16
Options Target Test Random Cluster Mode	16
Options Target Test Clear Target	16
Options Custom Tests Random I/O Test	16
Options Custom Tests Clear Tests.....	16
Options Custom Tests Mercury (w/ Init).....	16
Options Custom Tests Mercury (w/o Init)	16
Options Custom Tests Venus (w/Init)	17
Options Custom Tests Venus (w/o Init)	17
Options Custom Tests Earth (w/ Init).....	17
Options Custom Tests Earth (w/o Init).....	17
Options Custom Tests Jupiter (w/ Init)	17
Options Custom Tests Saturn (w/Init).....	17
Options Custom Test Saturn (w/o Init)	17
Options Custom Test Uranus (w/o Init)	17
Options Custom Profile	17
Options Recall Last Performance Data	17
Options Performance Workload Monitor	17
Options Patterns	17
Options Save Configuration	18
Options Import Configuration.....	18
Options Custom Profiles Create Profiles.....	18
Options Custom Profiles Map Profiles	18
Options Preferences.....	18
Select Logical Drives.....	18
Select Physical Drives.....	18
View Performance History Bytes Write (Red).....	18
View Performance History Bytes Read (Yellow)	18
View Performance History Bytes Total (Green)	18
View Thread Details	18

View Custom Details	18
GUI Items	19
Threads.....	19
Transfer Size.....	19
Buffer Size	19
Operation	19
Pattern	19
Status	19
I/O Depth	19
Timeout.....	19
Error Injection	20
Latency.....	20
Read Latency.....	20
Write Latency.....	20
I/O Direction.....	20
Operation	20
Elapsed Time	20
Performance	20
IOPS.....	20
Extended	20
Location.....	20
Performance Real-Time Graph	20
IOPS Real-Time Graph.....	20
Target Window.....	20
Error Checking Check Box	22
Performance Check Box.....	22
Asynchronous Check Box	22
Random Check Box	22
Start.....	22
Stop	22
Modify.....	22
License Operations.....	22

HASP USB Key License Procedure	24
Server License Key Procedure	24
Asynchronous and Synchronous I/O Description	25
Functional Tests	26
General Tests	27
Target Tests	31
Target Test Standalone Mode [TARGSTEST]	33
Target Test Random Standalone [TARGSTESTR]	34
Target Test Cluster Mode [TARGCTEST]	36
Target Test Random Cluster Mode [TARGCTESTR]	37
Target Test Standalone Mode with Variable Transfer Size [TARGVTEST]	38
Target Test Cluster Mode with Variable Transfer Size [TARGCVTEST]	40
Target Test Target Verify [TSVERIFYLIVE]	41
Target Test Target Verify Cold Storage Initialization [TSCOLDINIT]	42
Target Test Target Verify Cold Storage [TSCOLDVERIFY]	43
Custom Tests	44
Random I/O [RANDIOTEST]	44
Variations on the Sequential Workload Theme	45
Mercury [MERCURYA, MERCURYB]	45
Venus [VENUSA, VENUSB]	45
Earth [EARTHA, EARTH B]	46
Jupiter [JUPITER]	47
Saturn [SATURNA, SATURNB]	47
Uranus [URANUS]	48
Custom Profile [CUSTOM]	49
Saturn Test Results	50
GUI Items	50
Target	50
Display	50
Throughput	50
IOPS	51
Latency	51

LBA Start.....	51
LBA Stop	51
DataMover Preferences	52
Menu Items	53
Network Settings.....	53
HTTP Proxy	53
Port.....	53
Enable.....	53
Advanced I/O Settings.....	53
Logical Target Size	53
Async Timeout Value.....	53
Max Thread Count	53
Max I/O Depth	53
Enable Extended Perf.....	53
Disable Perf Streaming.....	53
Enable T10 PI.....	53
Stop All on Error	53
Enable PRNG Seed ReUse	54
Enable Data Tags.....	54
Atomizer.....	54
Enable Non-Sector Aligned I/O	56
Disable Default Workloads.....	56
Increment By	56
Enable Perf Bypass	56
Single Shot.....	56
Stop on Error	57
Strip Chart Recorder	57
Perf Results	57
Ramp Time	57
Custom Test	57
Stop Timer	57
Performance Estimator	57

Time Duration	57
Combinations	58
Logical Target Size Estimator	58
Cluster Settings	58
Enable SCSI PR.....	58
Aggressiveness	58
PR Service Action	58
PR Type.....	58
Reset	58
OK.....	58
Cancel.....	58
Performance Tests	58
Introduction	58
Theory and Operation.....	59
Workload Descriptions.....	60
Overlapping and Non-Overlapping I/O Descriptions	61
Sequential I/O Description.....	61
CACHE Workload Tests	62
WALK Workload Tests.....	65
RANDOM Workload Tests	67
Performance Tests on Logical and Physical Targets	68
Default Performance Tests	69
Extended Performance Tests	69
When Should EPT Be Used?	71
Long Test Results	72
Quick Test Results	75
Performance Workload Monitor	76
Menu Items	77
View Performance History Bytes Write (Red).....	77
View Performance History Bytes Read (Yellow)	77
View Performance History Bytes Total (Green)	78
View Graph Type Standard	78

View Graph Type per CPU Utilization	78
View Graph Type Latency vs. Transfer Size.....	78
View Graph Type CPU vs. Transfer Size	78
GUI Items	78
Top Chart.....	78
Bottom Chart	78
Target Selection	78
Refresh Rate	78
Stage.....	78
Workload Completion	79
Current Workload	79
OK.....	79
Performance Bypass Dialog	79
GUI Items	79
Custom Performance Tests	80
Theory of Operation.....	81
Results.....	81
Bus Trigger Operation	82
Windows Event Reporting	83
Physical Target Data Corruption Event	83
General Tab Data	83
Details Tab Data	83
Logical Target Data Corruption Event.....	84
General Tab Data	84
Details Tab Data	84
Windows Event ID 153 Explained	86
General Event ID 153 Message	87
Details of Event ID 153 Message.....	87
Final Analysis.....	90
Host Memory Considerations	90
Custom Data Patterns	91
Practical Knowledge.....	91

How are IOPS measured?	91
Why is Standard Deviation Important?	92
4K Native (4Kn) Drives	92
512e Advanced Format Drives	92
Global Configuration and Preferences Save and Import	92
Automation	94
Multiple Initiator Cluster Emulation Testing.....	94
Persistent Reserve and Release	94
High Performance Resolution Timer.....	94
Automation Support and Scripting	95
Creating Custom Profiles	97
Menu Items	98
File Import Profile	98
GUI Items	98
I/O Parameters.....	98
Percentages.....	99
Target Geometry Access by Percentage	99
Target Geometry Access by Offset	100
Name.....	100
Elements	100
Profile Listing.....	100
Copy	100
Up.....	100
Down	100
Add	100
Delete.....	100
Save	101
Reset	101
Mapping Custom Profiles.....	101
GUI Items	102
Left Pane	102
Right Pane	102

Add	102
Remove	102
Up	102
Down	102
Separator.....	103
Target	103
Threads.....	103
Loop.....	103
Delay	103
All Targets	103
Link	103
OK.....	103
Cancel.....	103
Reset	103
DataMover I/O Model.....	104
DataMover Data Integrity Error Analysis Example	106
Data Mismatch Bus Trigger	106
Data Mis compare I/O Example	107
T10 Protection Information Testing.....	109
Protection Information Fields	109
Methodology and Verification	110
PI Type 1.....	110
PI Type 2.....	110
PI Type 3.....	110
WRITE and READ API Errors	110
SCSI Errors and Warnings.....	111
Variable Transfer Sizes	112
Issue Replication Technique	112
Logging	113
Thread Details Dialog (TDD).....	113
Appendix	114
Advanced Dialog Real-Time Graph Mouse over Behavior	114

Performance Monitor Dialog Real-Time Chart Mouse over Behavior	114
Glossary of Terms.....	115

Introduction

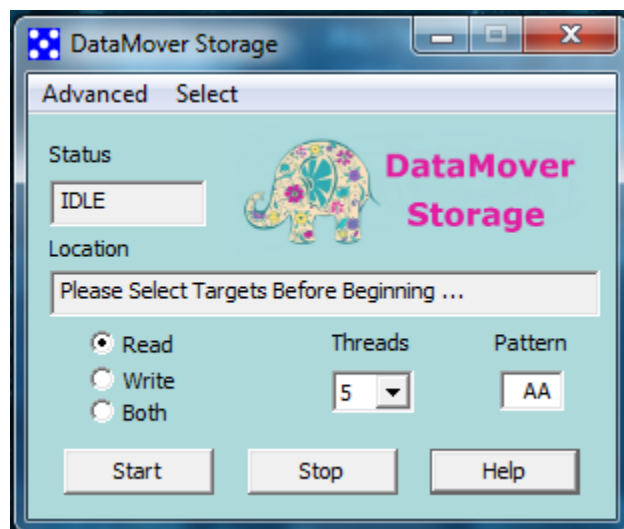
DataMover Storage (DMS) is not a command line application; it is a native graphical user interface (GUI) host-based application with no JAVA or .NET components. DMS's main purpose is to generate I/O to storage and networking interfaces to validate and characterize the subsystems that provide these resources. DMS is available in 32bit and 64bit versions. DMS will not bypass any native in-box components provided by Microsoft (MS) or a third-party vendor used to enable these resources on the host. Examples of these components include the Microsoft class and port driver, and vendor miniport and NDIS driver.

DMS provides functional and performance workload testing. DMS's performance workloads are designed to ensure that three basic workload types are implemented correctly – CACHE only, SEQUENTIAL only and RANDOM only. Functional tests are feature rich, with data integrity, bus triggers, and windows event logging.

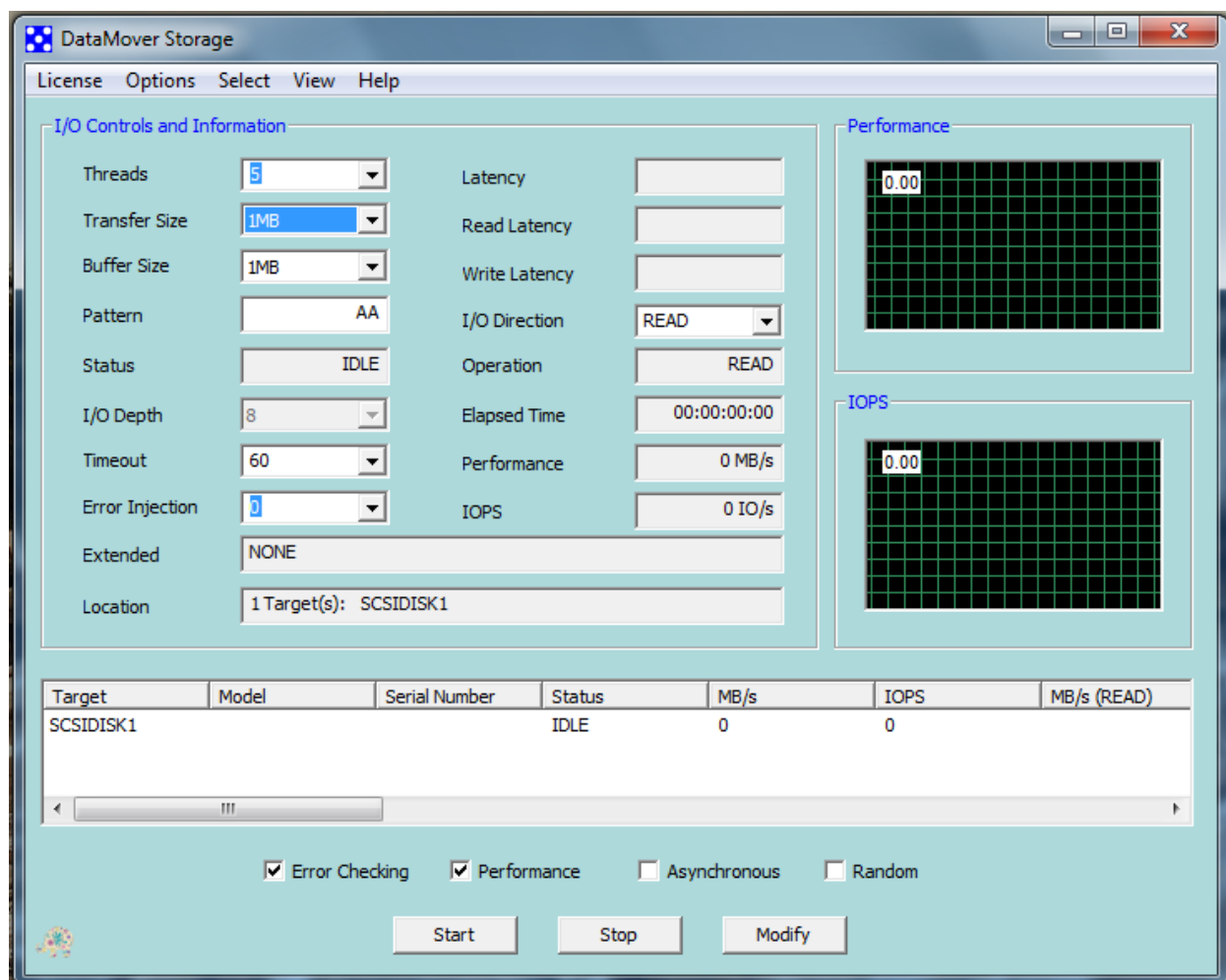
Screen Shot

DMS consists of two dialogs – the Basic Dialog (BD) and the Advanced Dialog (AD). As the name implies, the BD provides basic functionality and is not as feature rich as the AD. Users not wishing to learn AD features and requiring simple I/O generation, typically use the BD instead.

Picture of the Basic Dialog:



Picture of the Advanced Dialog:



Quick Start Guide

Download the latest version of DMS from www.storagewerks.net and install the application.

To begin any type of I/O operation, the target(s) must be selected. Targets fall under two categories – logical and physical. Logical targets can be logical drive letters (C: / for example) representing local drives or file share points. Physical targets are raw block devices presented to the host that are SCSI compliant. DMS will prevent I/O on physical targets that are associated with a logical drive. Target selection is accomplished using the menu item “Select->Logical Drives,” or “Select->Physical Drives.” Multiple targets may be selected using the SHIFT or CTRL keys. After target selection is complete, select START and I/O operations in the READ direction using the default data pattern of 0xAAh will begin.

Basic Dialog Operation

The BD consists of menu items, GUI buttons, windows and dropdowns. The BD provides basic I/O generation functionality for those users not requiring the AD extended feature set. What are the characteristics of this functionality?

- Synchronous I/O only
- Fixed I/O transfer size of 1 MiB
- READ, WRITE and BOTH I/O directions
- Selectable data patterns
- No Error checks
- No Performance metrics
- No Advanced data patterns

Menu Items

Select Logical Drives - Selects the logical targets to perform I/O to.

Select Physical Drives - Selects the physical targets to perform I/O to.

Advanced Patterns – Enables advanced data patterns used for BD and AD operations.

CRPAT, CSPAT and CJTPAT - CSPAT, CRPAT and CJTPAT are modified versions of the test patterns defined by *FC-MJSQ Specification* on www.t11.org. They differ in two key areas: 1.) The pattern is repeated until it fills the buffer size selected and 2.) Valid fibre channel headers are used vs. using these 24 bytes for data pattern. As a third-party vendor application, it is impossible for DMS to modify fibre channel frame formatting and if it were done, the frame would not be recognizable to another fibre channel device or fabric.

RANDOM – Enables random data patterns, “RND” will be displayed in BD and AD “Pattern” window control.

RESET - Sets the pattern back to 0xAAh, “AA” will be displayed in “Pattern” window control.

Advanced Features Logic - Opens the AD in logical target mode. Logical targets must have been selected prior to this operation. If BD is in “RUN” state when this item is selected, all I/O will automatically stop before AD is opened.

Advanced Features Physical - Opens the AD in physical target mode. Physical targets must have been selected prior to this operation. If BD is in “RUN” state when this item is selected, all I/O will automatically stop before AD is opened.

GUI Items

Status – Indicates status of current I/O operations – IDLE, RUN or STOP.

Location – Displays list of targets selected. For example, “1 Target(s): C:\.”

I/O Direction – Grouping of radio buttons that allow the user to select I/O direction, default in READ. What do these directions mean? READ – the host receives data from the target; WRITE – the host transmits data to the target; BOTH – the host simultaneously receive data from target and transmit data to the target.

Threads – A dropdown window that allows the user to select the number of concurrent I/O threads per target per I/O direction, default is 5. Maximum value is 25 threads per target for each I/O direction. For example, if BOTH I/O Direction is selected with thread count equal to 5, five READ threads and five WRITE threads will be spawned on each target.

Pattern – Edit window that allows the user to enter any byte pattern in hexadecimal format. Field holds a maximum of two characters; each character must be either 0-9 or A-F. For example, if the user enters “EB,” the byte pattern in binary format will be 1110 1011. This pattern will be used in all I/O operations for BD and AD operations.

Start – Begins I/O.

Stop – Ends all I/O.

Help – Launches rudimentary help file.

Advanced Dialog Operation

The checkbox’s at the bottom of the AD enable or disable basic features related to error checking, performance gathering, and type of I/O generated. When the “Error Checking” checkbox is enabled, DMS performs data integrity checks on all I/O, detailed verification on Application Programming Interface (API) return status and operation, I/O completion timeout monitoring, bus trigger activation on error detection, and error injection. When the “Performance” checkbox is enabled, DMS calculates throughput, IOPS and latency metrics for every I/O and displays this information; when unchecked these parameters will not be updated. When the “Asynchronous” checkbox is enabled, DMS performs asynchronous I/O, when unchecked, synchronous I/O is performed. When the “Random” checkbox is enabled, DMS performs random I/O generation, when unchecked, sequential I/O is enabled. Any combination of these checkboxes is valid. When certain tests are selected, these checkbox’s may be automatically configured by the application and appear grayed out.

Menu Items

License Import - Imports the license. This procedure is not required if using the license USB key and is only required when using the licensing server method.

License Register - Registers the application with the licensing server. Use this selection to renew the license lease. This procedure is not required if using the license USB key and is only required when using the licensing server method.

License Unregister - Removes the registration from the licensing server. Use this selection when preparing to move a license seat to a new machine. This procedure is not required if using the license USB key and is only required when using the licensing server method.

Options Import Data File - Imports a custom data pattern.

Options Performance Test - Selects CACHE workload performance test when Random checkbox unselected. If Random checkbox selected, RANDOM workload will be used. Duration: > 24 hrs.

Options Quick Performance Test - Selects CACHE or RANDOM workload quick performance test. If Random checkbox unselected, CACHE workload will be used. If Random checkbox selected, RANDOM workload will be used. Duration: < 30 minutes.

More Performance Tests Quick Walking Test - Selects WALK workload quick performance test. Duration: < 1 hr.

More Performance Tests Long Walking Test - Selects WALK workload performance test. Duration: > 24 hrs.

More Performance Tests Custom Profile Test

Selects the Custom Profile performance test. Duration: Variable.

Options Clear Modifications - Clears per target modifications made in physical or logical mode.

Options Target Test Standalone Mode - Selects the Standalone Target test mode.

Options Target Test Cluster Mode - Selects the Cluster Target test mode.

Options Target Test Random Standalone Mode - Selects the random Standalone Target test mode.

Options Target Test Random Cluster Mode - Selects the random Cluster Target test mode.

Options Target Test Clear Target - Disables target mode testing and restores previous AD workload and test settings.

Options Custom Tests Random I/O Test - Selects the random I/O test mode.

Options Custom Tests Clear Tests - Disables custom test and restores previous AD workload and test settings.

Options Custom Tests Mercury (w/ Init) – Selects the Mercury sequential workload, available on physical targets only. Data pattern will be initialized for READ I/O direction.

Options Custom Tests Mercury (w/o Init) – Selects the Mercury sequential workload, available on physical targets only. Data pattern will not be initialized for READ I/O direction; data read from target will not be controlled by DMS.

Options Custom Tests Venus (w/Init) – Selects the Venus sequential workload, available on physical targets only. Data pattern will be initialized for READ I/O direction.

Options Custom Tests Venus (w/o Init) – Selects the Venus sequential workload, available on physical targets only. Data pattern will not be initialized for READ I/O direction; data read from target will not be controlled by DMS.

Options Custom Tests Earth (w/ Init) – Selects the Earth sequential workload, available on physical targets only. Data pattern will be initialized for READ I/O direction.

Options Custom Tests Earth (w/o Init) – Selects the Earth sequential workload, available on physical targets only. Data pattern will not be initialized for READ I/O direction; data read from target will not be controlled by DMS.

Options Custom Tests Jupiter (w/ Init) – Selects Juniper sequential workload, available on logical targets only. Data pattern will be initialized for READ I/O direction.

Options Custom Tests Saturn (w/Init)

Selects the Saturn sequential workload, available on physical targets only; the data pattern will be initialized for READ I/O directions.

Options Custom Test Saturn (w/o Init)

Selects the Saturn sequential workload, available on physical targets only; data pattern will not be initialized for READ I/O direction, data read from target will not be controlled by DMS.

Options Custom Test Uranus (w/o Init)

Selects the Uranus custom profile test, available on physical targets only; sequential custom profiles only supported, random custom profiles not supported.

Options Custom Profile

Selects the Custom Profile test, available on physical targets only; if a custom profile has not been mapped to all targets, test selection will be aborted.

Options Recall Last Performance Data - Displays results of last performance test that was run.

Options Performance Workload Monitor – Displays the Performance Monitor Dialog (PMD), will not display if any thread is in INIT state.

Options Patterns - Enables advanced data patterns used for BD and AD operations.

CRPAT, CSPAT and CJTPAT - CSPAT, CRPAT and CJTPAT are modified versions of the test patterns defined by *FC-MJSQ Specification* on www.t11.org. They differ in two key areas: 1.) The pattern is repeated until it fills the buffer size selected and 2.) Valid fibre channel headers are used vs. using these 24 bytes for data pattern. As a third-party vendor application, it is impossible for DMS to modify fibre channel frame formatting and if it were done, the frame would not be recognizable to another fibre channel device or fabric.

RANDOM – Enables random data patterns, “RND” will be displayed in BD and AD “Pattern” window control.

RESET - Sets the pattern back to 0xAAh, “AA” will be displayed in “Pattern” window control.

Options Save Configuration

This menu item allows the user to save the current DMS configuration to a file. The file will be given a date/time stamp and pre-appended with ‘_LOGICAL_’ or ‘_PHYSICAL_’ depending on the target type selection present during the save.

Options Import Configuration

This menu item allows the user to import a previously saved DMS configuration profile. The currently selected target type must match the target type saved in the configuration file. If they do not, the configuration cannot be imported.

Options Custom Profiles Create Profiles

Opens the DataMover Custom Profile Dialog (DCPD).

Options Custom Profiles Map Profiles

Opens the DataMover Custom Mapping Dialog (DCMD) used to assign custom profiles to one or more targets.

Options Preferences - This will open the DataMover Preferences Dialog (DPD) that allows the user to adjust specific settings related to internet access for registration purposes, performance testing behavior, I/O behavior and logical target free space consumption.

Select Logical Drives - Selects the logical targets to perform I/O to.

Select Physical Drives - Selects the physical targets to perform I/O to.

View Performance History Bytes Write (Red) - Enable or disable Write Performance and IOPS graph data.

View Performance History Bytes Read (Yellow) - Enable or disable Read Performance and IOPS graph data.

View Performance History Bytes Total (Green) - Enable or disable Total Performance and IOPS graph data.

View Thread Details

Opens Thread Details Dialog (TDD).

View Custom Details

Opens Custom Profile Details Dialog (CPDD).

GUI Items

Threads - A dropdown window that allows the user to select the number of concurrent I/O threads per target per I/O direction, default is 5. Maximum value is 25 threads per target for each I/O direction. For example, if BOTH I/O Direction is selected with thread count equal to 5, five READ threads and five WRITE threads will be spawned on each target.

Transfer Size – A dropdown window that allows the user to select the maximum transfer size used for each I/O exchange, range is 512B to 4MiB. This value may differ from the maximum transfer size used by the hardware. Maximum transfer size settings are determined by the HBA manufacturer or Microsoft. For example, if you are using the Microsoft iSCSI Initiator on an NDIS driver, the default hardware maximum transfer size is 256KiB. If the transfer size selected is greater than this value, 512KiB for example, two 256KiB I/O hardware exchanges will be required to complete the transfer. The transfer size must be less than or equal to buffer size.

Buffer Size - A dropdown window that allows the user to select the maximum buffer size used for each I/O exchange, range is 512B to 4MiB. What is the purpose of buffer size? Buffer size determines the number of consecutive I/O exchanges that will take place. For example, if the transfer size is 512B, and the buffer size is 4MiB, 8192 I/O exchanges will be required to complete the transfer. The buffer size must be greater than or equal to transfer size.

Operation – Displays global I/O direction setting, possible values are READ, WRITE or WRITE/READ. READ indicates that the target is transmitting data to the host; WRITE indicates that the host is transmitting data to the target and WRITE/READ indicates that the host and target are simultaneously transmitting data to each other.

Pattern - This can be the data pattern selected in the BD, AD or the pattern imported by the AD.

Status – Displays the global status of the AD, possible values are RUN, STOP or FAIL. RUN indicates that all threads have started; STOP indicates that all threads have stopped or are in the process of stopping; FAIL indicates one or more threads have failed. For FAIL condition, see “Extended” window for additional information on failure.

I/O Depth – A dropdown that allows the user to select the number of maximum outstanding I/Os per thread per target, maximum value is 64. DMS does not guarantee that there will always be this number of outstanding I/Os on the target. The number of outstanding IOs on a target will be dependent on the number of threads per target, transfer size, buffer size, I/O depth selection and the maximum transfer size of the host adapter. This control will be disabled when the asynchronous check box is unchecked.

Timeout - I/O timeout is adjustable if performing I/O to a physical target in synchronous mode. The default setting will be determined by class driver timeout parameter setting in the system registry. If it does not exist, 16 seconds will be used as the default. This control will be disabled when the asynchronous checkbox is checked.

Error Injection - Error Injection is available when performing I/O on physical targets in synchronous or asynchronous modes. The default setting is zero seconds. By selecting any value other than zero, bus resets will be performed on the selected targets periodically. For example, if the user selects 20 in this dropdown window, a bus reset will be performed once every 20 seconds. Bus resets test the system's ability to perform error recovery and I/O retries.

Latency - The total average latency in milliseconds per IO across all targets for all I/O directions.

Read Latency - The total average latency in milliseconds per IO across all targets for READ I/O direction only.

Write Latency - The total average latency in milliseconds per IO across all targets for WRITE I/O direction only.

I/O Direction – A dropdown window that allows the user to select I/O direction, possible values are READ, WRITE or BOTH. Default direction is READ.

Operation – Displays the type of operation or test in progress. Some of the possible values are READ, WRITE, WRITE/READ, PERFTEST, QPERFTEST, PERFWLKTST, QPERFWLKTST, TARGSTEST and TARGSTESTR.

Elapsed Time – The window displays a timer that starts counting when “Start” is selected and stops counting when “Stop” is selected. The format is as follows: DAYS: HOURS: MINUTES: SECONDS.

Performance – The total throughput in MiB/s across all targets for all I/O directions.

IOPS – The total IOPS (Input/output/sec) across all targets for all I/O directions.

Extended - If error checking is enabled and a thread experiences a failure, a more detailed description of the failure will be displayed here. In addition, an application event will be logged providing the most detail on the error condition.

Location - Number of targets selected and their string identifiers. For example, “10 Target(s): SCSIDISK1 SCSIDISK2 SCSIDISK3 SCSIDISK4 SCSIDISK5 SCSIDISK6 SCSIDISK7 SCSIDISK8 SCSIDISK9 SCSIDISK10.”

Performance Real-Time Graph - Displays the total throughput across all targets for READ, WRITE and BOTH I/O directions. Double clicking on the plot area will reset values back to zero. Default display is READ I/O direction only

IOPS Real-Time Graph - Displays the total IOPS across all targets for READ, WRITE and BOTH I/O directions. Double clicking on the plot area will reset values back to zero. Default display is READ I/O direction only

Target Window

Target – Target string identifier.

Status – Displays current thread status, possible values are INIT, RUN, STOP and FAIL. INIT will be displayed when the thread is initializing and has not started I/O to the target. RUN will be displayed when at least one thread has reached RUN state; STOP will be displayed when at least one thread exits or stops. If error checking is enabled and a thread experiences a failure, FAIL will be displayed.

MB/s - The total throughput in MiB/s per target for all I/O directions.

IOPS - The total IOPS per target for all I/O directions.

MB/s (READ) - The total throughput in MiB/s per target for READ I/O direction.

IOPS (READ) - The total IOPS per target for READ I/O direction.

MB/s (WRITE) - The total throughput in MiB/s per target for WRITE I/O direction

IOPS (WRITE) - The total IOPS per target for WRITE I/O direction.

Error - If error checking is enabled and a thread experiences a failure, a more detailed description of the failure will be displayed here along with an application event being logged providing the most detail on the error condition.

Operation – Displays the type of I/O per target, possible values are ASYNC and SYNC; ASYNC designates asynchronous I/O, SYNC designates synchronous I/O.

Chunk -Transfer size selected for target.

Buffer - Buffer size selected for target.

Performance – Indicates if performance metrics are enabled for target, possible values are ENABLED and DISABLED.

Error Checking - Indicates if error checking is enabled for target, possible values are ENABLED and DISABLED.

I/O Depth - Queue depth per thread per target associated with asynchronous I/O.

Threads - The number of threads per I/O direction per target.

Direction - I/O direction selected for target (READ, WRITE, or READ/WRITE).

Latency - Average latency value for target, READ and WRITE direction (in milliseconds per IO).

Latency (READ) - Average Latency value for target, READ only direction (in milliseconds per IO).

Latency (WRITE) - Average latency value for target, WRITE only direction (in milliseconds per IO).

LBA Ops – Displays the type of LBA access in progress, possible values RANDOM or SEQUENTIAL.

File Ops – Displays the type of file access in progress, possible values are ALIGNED and NON-ALIGNED.

Error Checking Check Box - If selected, error checking will be enabled.

Performance Check Box - If selected, performance monitoring will be enabled.

Asynchronous Check Box - If selected, asynchronous I/O will be enabled.

Random Check Box - If selected, random LBA (physical targets) or random file access (logical target) will be enabled.

Start – Begins I/O generation

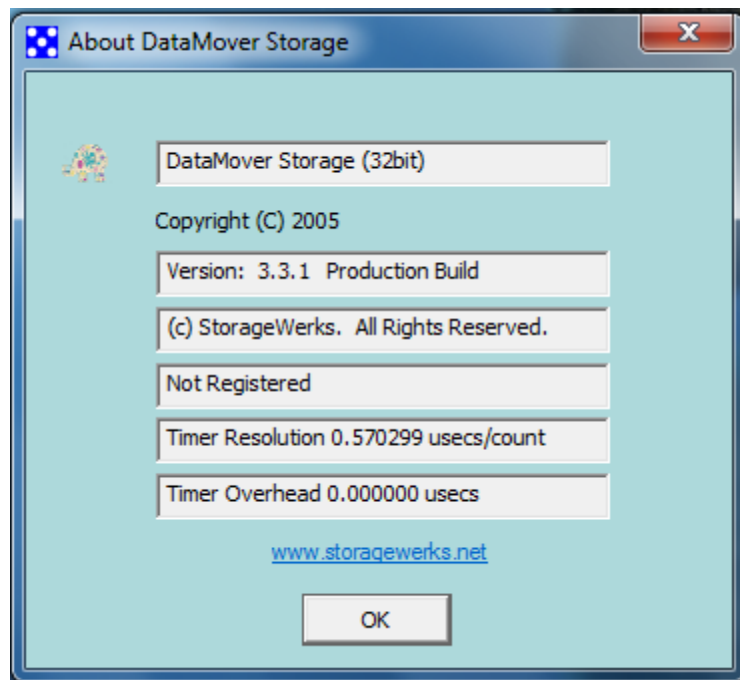
Stop – Stops all I/O

Modify – By default, all targets will use the same workload and test parameter settings, but this can be bypassed using the “Modify” button. The user may define separate workload and test parameters on a per target basis by selecting the new settings in the AD, highlighting the target in target window and selecting “Modify.” This button will remain disabled once “Run” is selected and will not be re-enabled until all I/O is stopped, and “Clear Modifications” is performed from the menu.

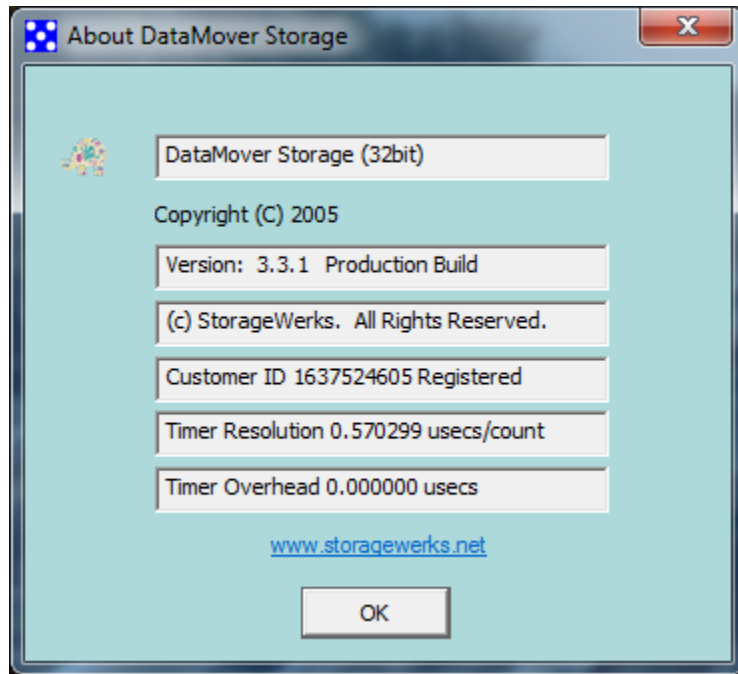
License Operations

There are two licensing methods DMS supports – license server method and HASP USB license key method. If using the licensing server method, a license key must be imported for activation, this step is not required if using the USB license key method. To determine if a license has been activated, open the “About DataMover” menu item from the BD. Both licensing methods require the machine to have at least one network interface card (NIC) enabled.

Screenshot before license activation:



Screenshot after license activation:



Notice how the bottom control change's, displaying the license identification number.

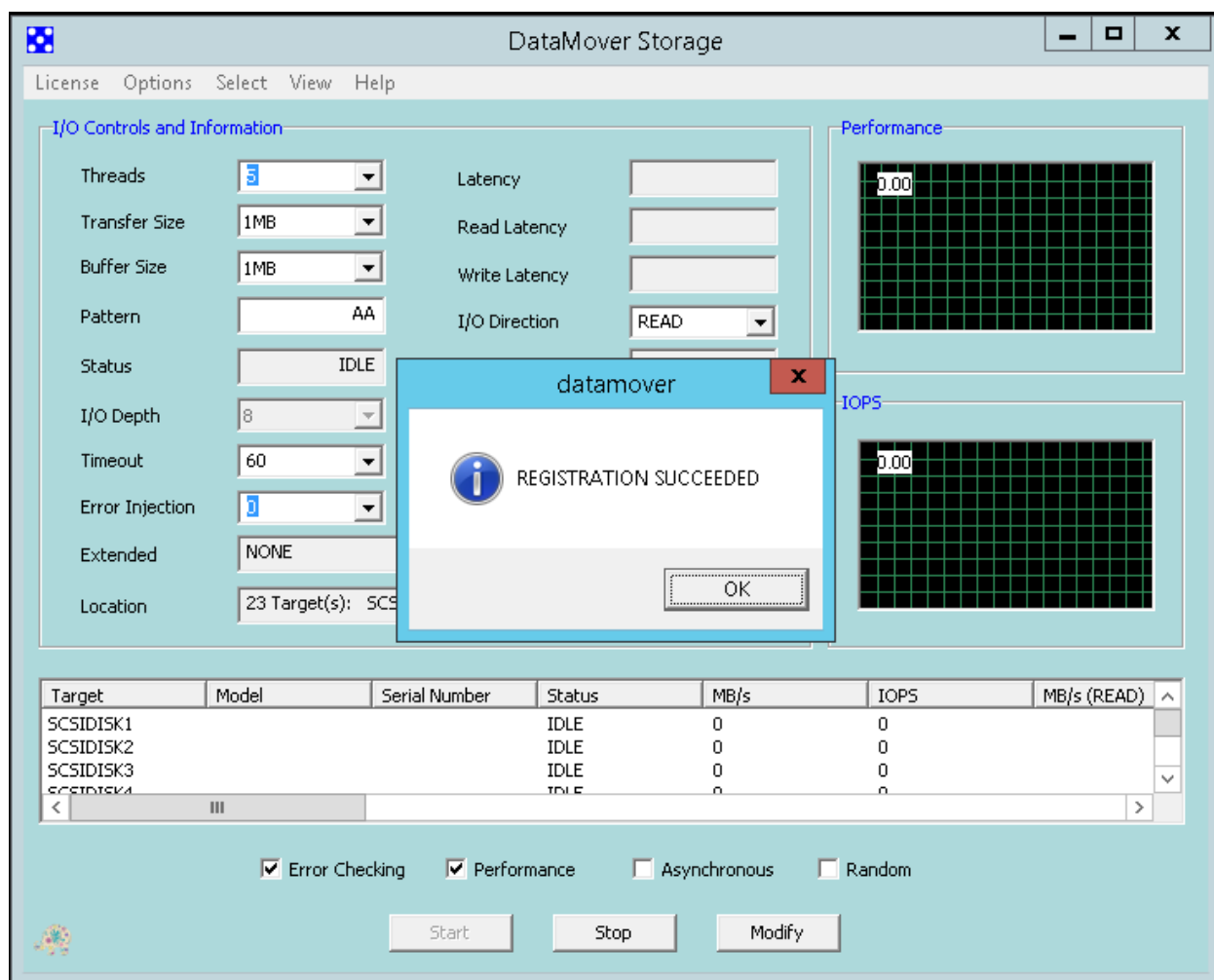
HASP USB Key License Procedure

The license activation is simple. After installing the application, install the HASP key before starting the application. You are done! The time remaining on the key will be posted in the windows application event log each time the application is opened. The key may be used on multiple machines, move the key to a new machine to activate DMS.

Server License Key Procedure

Open the AD and perform the license import procedure using the "License->Import" menu item. Using the dialog, point to the license file location and select "OK."

Successful license key activation response:



When a license is registered, it will be given a 10-day lease for that machine. When the lease expires, the user must perform a license registration by selecting the “License->Register” menu item in the AD. This will renew the lease for another 10 days on this machine. If the user wishes to move the license from one machine to another, perform the following steps:

1. Unregister the license from the current machine by selecting the “License->UnRegister” menu item in the AD. This will release the license seat back into the general pool managed by the license server.
2. On the new machine, perform a license import if this machine has never been used before. If DMS has been used on this machine before, perform a license registration by selecting the “License->Registration” menu item in the AD. This will allocate a license seat from the general pool managed by the license server to this new machine.

Asynchronous and Synchronous I/O Description

DMS performs two basic types of I/O – synchronous and asynchronous. There can be random synchronous and non-random (or sequential) synchronous I/O; and there can be random asynchronous

and non-random (or sequential) asynchronous I/O. All the other I/O workload parameter attributes such as the ones just mentioned will fall under these two categories. What is the difference between these types of I/O? When synchronous I/O has been selected, the DMS thread(s) responsible for the I/O generation will submit only one I/O request at a time and wait for its completion before sending the next I/O request. When asynchronous I/O has been selected, the DMS thread(s) responsible for the I/O generation will submit one or more I/O requests at a time and wait for all outstanding I/O's to complete before sending the next grouping of I/O requests. The "I/O Depth" control in the AD determines the number of outstanding I/O requests each thread will send. For example, if this setting is 8, each I/O thread will send 8 I/O requests at a time and wait for all 8 to complete before sending the next grouping of 8. The higher the I/O depth setting, the more outstanding I/O requests will be sent to the target. Asynchronous I/O tests the system's ability to process many queued I/O requests both on the host and target side. Asynchronous I/O typically provides the best performance, but sometimes this is not always the case.

Functional Tests

What attributes defines a functional test and makes them unique? The short answer is error checking. Functional testing is only performed by the AD and can be enabled using AD menu items and AD controls.

General Tests

Functional tests can be enabled from the main AD GUI by checking the “Error Checking” checkbox. When performing functional tests, the user must consider the type of test coverage taking place which will be determined by I/O direction, and whether sequential or random I/O is enabled. The table below explains test coverage based on feature selections and target coverage which explains the amount of the target’s geometry that may be accessed.

I/O Direction	Random or Sequential	Target Type	Target Coverage	Explanation
READ	SEQUENTIAL	LOGICAL	VERY SMALL	Each I/O thread will create a small file large enough to meet the requirements of the workload parameters. The file will be written to once initializing with the data pattern selected. After file initialization is complete, the file will be read from many times, after each read pass, error checks will be performed, this process will repeat until the test is stopped. Since the files are small, this will produce a cache only type workload response.
WRITE	SEQUENTIAL	LOGICAL	VERY SMALL	Each I/O thread will create a small file large enough to meet the requirements of the workload parameters. The file will be initialized with data pattern selected. After file initialization complete, a write->read->compare process begins and will repeat until the test is stopped. Since the files are small, this will produce a cache only type workload response for READ I/O direction.
BOTH	SEQUENTIAL	LOGICAL	VERY SMALL	Each I/O thread will create a small file large enough to meet the requirements of the workload parameters. The file will be initialized

				with data pattern selected. After file initialization complete, a write->read->compare process begins and will repeat until the test is stopped. Since the files are small, this will produce a cache only type workload response for READ I/O direction.
READ	RANDOM	LOGICAL	<p>VERY SMALL</p> <p>By default, free space usage is set to 80% of available for logical targets. The amount of free space consumed for this test will be dependent on "Logical Target Size" setting in the DPD.</p>	Each I/O thread will create a file large enough to meet the "Logical Target Size" setting in the DPD. The file will be written to once initializing with the data pattern selected; time duration for this process will be dependent on workload parameters and "Logical Target Size." After file initialization is complete, the file will be read from many times, after each read pass, error checks will be performed, this process will repeat until the test is stopped. Since the same random file offsets are accessed repeatedly, this will produce a cache only type workload response.
WRITE	RANDOM	LOGICAL	<p>80%</p> <p>By default, free space usage is set to 80% of available for logical targets. The amount of free space consumed for this test will be dependent on "Logical Target Size" setting in the DPD.</p>	Each I/O thread will create a file large enough to meet the "Logical Target Size" setting in the DPD. The file will be initialized with the data pattern selected; time duration for this process will be dependent on workload parameters and "Logical Target Size." After file initialization is complete, a random offset into the file are chosen and a write->read->compare operation performed. This process will repeat until the test is stopped.

				Since the files are potentially large and file offsets are chosen randomly, cache hits on READ are minimized.
BOTH	RANDOM	LOGICAL	80% By default, free space usage is set to 80% of available for logical targets. The amount of free space consumed for this test will be dependent on "Logical Target Size" setting in the DPD.	Each I/O thread will create a file large enough to meet the "Logical Target Size" setting in the DPD. The file will be initialized with the data pattern selected; time duration for this process will be dependent on workload parameters and "Logical Target Size." After file initialization is complete, a random offset into the file are chosen and a write->read->compare operation performed. This process will repeat until the test is stopped. Since the files are potentially large and file offsets are chosen randomly, cache hits on READ are minimized.
READ	SEQUENTIAL	PHYSICAL	VERY SMALL	Each I/O thread will occupy its own range of contiguous LBA's, range size dependent on workload parameters. The storage locations will be initialized with the data pattern selected. After initialization is complete, the storage locations will be read from many times, after each read pass, error checks will be performed, this process will repeat until the test is stopped. Since the LBA ranges are small, this will produce a cache only type workload response.
WRITE	SEQUENTIAL	PHYSICAL	VERY SMALL	Each I/O thread will occupy its own range of contiguous LBA's, range size dependent on workload parameters.

				<p>The storage locations will be initialized with the data pattern selected. After initialization is complete, a write->read->compare process begins and will repeat until the test is stopped. Since the LBA ranges are small, this will produce a cache only type workload response for READ I/O direction.</p>
BOTH	SEQUENTIAL	PHYSICAL	VERY SMALL	<p>Each I/O thread will occupy its own range of contiguous LBA's, range size dependent on workload parameters. The storage locations will be initialized with the data pattern selected. After initialization is complete, a write->read->compare process begins and will repeat until the test is stopped. Since the LBA ranges are small, this will produce a cache only type workload response for READ I/O direction.</p>
READ	RANDOM	PHYSICAL	VERY SMALL	<p>Each I/O thread will occupy its own group of randomly selected LBA ranges, range size dependent on workload parameters. The storage locations will be initialized with the data pattern selected. After initialization is complete, the storage locations will be read from many times, after each read pass, error checks will be performed, this process will repeat until the test is stopped. Since the same LBA's are accessed repeatedly, this will produce a cache only type workload response.</p>

WRITE	RANDOM	PHYSICAL	100%	Each I/O thread randomly selects an LBA range large enough to complete the I/O request. Each thread gets exclusive ownership, producing non-overlapping I/O. After LBA selection, a write->read->compare operation is performed and the process repeats on a new randomly selected LBA range until the test is stopped. Since the I/O will span the entire target geometry, cache hits on READ are minimized.
BOTH	RANDOM	PHYSICAL	100%	Each I/O thread randomly selects an LBA range large enough to complete the I/O request. Each thread gets exclusive ownership, producing non-overlapping I/O. After LBA selection, a write->read->compare operation is performed and the process repeats on a new randomly selected LBA range until the test is stopped. Since the I/O will span the entire target geometry, cache hits on READ are minimized.

Target Tests

These functional tests use the phrase “Target” because they will always access the full target geometry and perform write->read->compare operations most of the time. As always, logical target coverage will be dependent on the “Logical Target Size” setting in the DPD which defaults to 80% of available free space. There are four types of target test – standalone, random standalone, cluster and random cluster. The standalone tests refer to a single DMS host having access to a target; cluster refers to multiple DMS hosts having access to the same target. The cluster tests are available only on physical targets and are meant to simulate a storage cluster environment where multiple cluster nodes access the same block device. Exclusive access to the device is controlled by SCSI reserve and release commands and the target’s ability to process these control mechanisms correctly.

Each section will describe the test in terms of I/O direction, workload type, bus trigger availability, automation level, data comparison operation, target type and target coverage. Each section also includes comments on AD fixed settings, adjustable settings, test purpose, workload descriptions and user notes.

Target Test Standalone Mode [TARGSTEST]

I/O Direction	Workload Type	Bus Trigger	Automation	Data Comparison	Target Type	Target Coverage	T10 PI	Error Injection
Write followed by Read	Sequential or Random, asynchronous or synchronous	Yes	None, settings manually chosen by user, same workload used on all targets	Always On	Logical and Physical	Full	Physical targets only	Physical targets only

Fixed Settings

Error On, Performance On, I/O Direction

Adjustable Settings

Thread Count, Transfer Size, Buffer Size, I/O Depth, Asynchronous On/Off, I/O Timeout Duration (physical target only running synchronous workload), Error Injection Timer (physical targets only), Data Pattern, Non-Aligned or Aligned I/O (logical targets only), Random On/Off, T10 PI On/Off, Data Tags On/Off, PRNG Seed Re-use On/Off, Atomizer On/Off.

Purpose

Perform I/O with error checking enabled to the entire target geometry using the same workload parameters across all targets.

Sequential Workload Description

For sequential workloads, odd numbered threads will start at the beginning of the target's geometry, and even numbered threads will start at the end of the target's geometry. Threads will use non-overlapping storage locations and work their way up and down the target geometry. After the thread completes each pass (up or down) the process will repeat.

Random Workload Description

For random workloads, each I/O thread will randomly select a region of the target that is big enough to complete the I/O request.

Operational Notes

Logical targets will use the class disk driver timeout value defined in the registry for most circumstances. Physical targets running asynchronous workloads will use the class disk driver timeout value defined in the registry. Physical targets running synchronous workloads will default to the class disk driver timeout value defined in the registry, however, value can be manually adjusted in the AD.

Target Test Random Standalone [TARGSTESTR]

The term random in this test does not refer to random workload, instead it refers to the workload parameters which are randomly selected for each target.

I/O Direction	Workload Type	Bus Trigger	Automation	Data Comparison	Target Type	Target Coverage	T10 PI	Error Injection
Write followed by Read	Sequential or Random, asynchronous or synchronous	Yes	Yes	Always On	Logical and Physical	Full	Physical targets only	Physical targets only

Fixed Settings

Error On, Performance On, I/O Direction

Adjustable Settings

I/O Timeout Duration (physical target only running synchronous workload), Error Injection Timer (physical targets only), Data Pattern, Non-Aligned or Aligned I/O (logical targets only), T10 PI On/Off, Data Tags On/Off, PRNG Seed Re-use On/Off, Atomizer On/Off.

Purpose

Perform I/O with error checking enabled to the entire target geometry while maximizing workload variability across one or more targets.

Automation Behavior

Physical Targets

Some of the randomly selected attributes will apply to all I/O threads running on the target, these are major workload classification (asynchronous or synchronous) and lba access methods (sequential or random). All threads will be either asynchronous or synchronous and will access the geometry using sequential or random-access methods. If communication with the target is asynchronous, all I/O threads will operate using the same IODepth, also randomly calculated. In summary, the same workload classification, IODepth (if asynchronous) and lba access method will be common across all threads on any given target and will vary across multiple targets.

The number of I/O threads per target will be randomly determined. Each I/O thread can be one of the following randomly selected target test types: TARGSTEST, TARGSTESTR or TARGVTEST.

- TARGSTEST – All I/O threads of this type will share same transfer and buffer size and will run in Target Test Standalone Mode.
- TARGSTESTR – All I/O threads of this type will have randomly selected transfer and buffer size and will perform a modified version of TARGSTEST. The modifications include random lba skips, read bypass, delays and read overlays on any part of the target's geometry.

- TARGVTEST – All I/O threads of this type will perform variable length transfers and will run in Target Test Standalone Mode with Variable Transfer Size mode. The number and length of variable transfers will be randomly determined and will be sufficient to satisfy the randomly selected buffer size.

Logical Targets

Some of the randomly selected attributes will apply to all I/O threads running on the target, these are major workload classification (asynchronous or synchronous) and file access methods (sequential or random). All threads will be either asynchronous or synchronous and will access the file using sequential or random-access methods. If communication with the target is asynchronous, all I/O threads will operate using the same IODepth, also randomly calculated. In summary, the same workload classification, IODepth (if asynchronous) and file access method will be common across all threads on any given target and will vary across multiple targets.

The number of I/O threads per target will be randomly determined. Unique to each I/O thread will be randomly selected transfer and buffer size. All threads will perform the basic target sequential or random workload.

Sequential Workload Description

For sequential workloads, odd numbered threads will start at the beginning of the target's geometry, and even numbered threads will start at the end of the target's geometry. Threads will use non-overlapping storage locations and work their way up and down the target geometry. After the thread completes each pass (up or down) the process will repeat.

Random Workload Description

For random workloads, each I/O thread will randomly select a region of the target that is big enough to complete the I/O request.

Operational Notes

Logical targets will use the class disk driver timeout value defined in the registry for most circumstances. Physical targets running asynchronous workloads will use the class disk driver timeout value defined in the registry. Physical targets running synchronous workloads will default to the class disk driver timeout value defined in the registry, however, value can be manually adjusted in the AD.

Target Test Cluster Mode [TARGCTEST]

I/O Direction	Workload Type	Bus Trigger	Automation	Data Comparison	Target Type	Target Coverage	T10 PI	Error Injection
Write followed by Read	Sequential or Random, asynchronous or synchronous	Yes	None, settings manually chosen by user, same workload used on all targets	Always On	Physical targets only	Full	Yes	No

Fixed Settings

Error On, Performance On, I/O Direction, Error Injection Off

Adjustable Settings

Thread Count, Transfer Size, Buffer Size, I/O Depth, Asynchronous On/Off, I/O Timeout Duration (applies to synchronous workloads only), Data Pattern, Random On/Off, T10 PI On/Off, Data Tags On/Off, and PRNG Seed Re-use On/Off, Atomizer On/Off.

Purpose

Perform I/O with error checking enabled to the entire target geometry using the same workload parameters across all targets, sharing the targets across multiple hosts using SCSI reserve/release or persistent reservations to control access.

Sequential Workload Description

For sequential workloads, odd numbered threads will start at the beginning of the target's geometry, and even numbered threads will start at the end of the target's geometry. Threads will use non-overlapping storage locations and work their way up and down the target geometry. After the thread completes each pass (up or down) the process will repeat.

Random Workload Description

For random workloads, each I/O thread will randomly select a region of the target that is big enough to complete the I/O request.

Operational Notes

Please see the DPD section regarding SCSI reserve/release functionality. Physical targets running asynchronous workloads will use the class disk driver timeout value defined in the registry. Physical targets running synchronous workloads will default to the class disk driver timeout value defined in the registry, however, value can be manually adjusted in the AD.

Target Test Random Cluster Mode [TARGCTESTR]

The term random in this test does not refer to random workload, instead it refers to the workload parameters which are randomly selected for each target.

I/O Direction	Workload Type	Bus Trigger	Automation	Data Comparison	Target Type	Target Coverage	T10 PI	Error Injection
Write followed by Read	Sequential or Random, asynchronous or synchronous	Yes	Yes	Always On	Physical targets only	Full	Yes	No

Fixed Settings

Error On, Performance On, I/O Direction, Error Injection Off

Adjustable Settings

I/O Timeout Duration (applies to synchronous workloads only), Data Pattern, T10 PI On/Off, Data Tags On/Off, PRNG Seed Re-use On/Off, Atomizer On/Off.

Purpose

Perform I/O with error checking enabled to the entire target geometry while maximizing workload variability across one or more targets, sharing the targets across multiple hosts using SCSI reserve/release or persistent reservations to control access.

Automation Behavior

Physical Targets

Some of the randomly selected attributes will apply to all I/O threads running on the target, these are major workload classification (asynchronous or synchronous) and lba access methods (sequential or random). All threads will be either asynchronous or synchronous and will access the geometry using sequential or random-access methods. If communication with the target is asynchronous, all I/O threads will operate using the same IODepth, also randomly calculated. In summary, the same workload classification, IODepth (if asynchronous) and lba access method will be common across all threads on any given target and will vary across multiple targets.

The number of I/O threads per target will be randomly determined. Each I/O thread can be one of the following randomly selected target test types: TARGCTEST, TARGCTESTR or TARGCVTEST.

- TARGCTEST – All I/O threads of this type will share same transfer and buffer size and will run in Target Test Cluster Mode.
- TARGCTESTR – All I/O threads of this type will have randomly selected transfer and buffer size and will perform a modified version of TARGCTEST. The modifications include random lba skips, read bypass, delays and read overlays on any part of the target's geometry.

- **TARGCVTEST** – All I/O threads of this type will perform variable length transfers and will run in Target Test Cluster Mode with Variable Transfer Size mode. The number and length of variable transfers will be randomly determined and will be sufficient to satisfy the randomly selected buffer size.

Logical Targets

Not applicable.

Sequential Workload Description

For sequential workloads, odd numbered threads will start at the beginning of the target's geometry, and even numbered threads will start at the end of the target's geometry. Threads will use non-overlapping storage locations and work their way up and down the target geometry. After the thread completes each pass (up or down) the process will repeat.

Random Workload Description

For random workloads, each I/O thread will randomly select a region of the target that is big enough to complete the I/O request.

Operational Notes

Logical targets will use the class disk driver timeout value defined in the registry for most circumstances. Physical targets running asynchronous workloads will use the class disk driver timeout value defined in the registry. Physical targets running synchronous workloads will default to the class disk driver timeout value defined in the registry, however, value can be manually adjusted in the AD.

Target Test Standalone Mode with Variable Transfer Size [TARGVTEST]

I/O Direction	Workload Type	Bus Trigger	Automation	Data Comparison	Target Type	Target Coverage	T10 PI	Error Injection
Write followed by Read	Sequential or Random, asynchronous or synchronous	Yes	None, settings manually chosen by user, same workload used on all targets	Always On	Physical targets only	Full	Yes	Physical targets only

Fixed Settings

Error On, Performance On, I/O Direction

Adjustable Settings

Thread Count, Transfer Size, Buffer Size, I/O Depth, Asynchronous On/Off, I/O Timeout Duration (physical target only running synchronous workload), Error Injection Timer (physical targets only), Data

Pattern, Non-Aligned or Aligned I/O (logical targets only), Random On/Off, T10 PI On/Off, Data Tags On/Off, PRNG Seed Re-use On/Off, Atomizer On/Off.

Purpose

Perform I/O with error checking enabled to the entire target geometry using the same workload parameters across all targets using variable transfer sizes.

Variable Transfer Sizes

The individual I/O thread will randomly calculate the number of transfers required to fulfill the buffer size which is manually set by the user. The length of these transfers will be randomly calculated. In general, the larger the buffer size, the greater the number of variable transfers will be required.

Sequential Workload Description

For sequential workloads, odd numbered threads will start at the beginning of the target's geometry, and even numbered threads will start at the end of the target's geometry. Threads will use non-overlapping storage locations and work their way up and down the target geometry. After the thread completes each pass (up or down) the process will repeat.

Random Workload Description

For random workloads, each I/O thread will randomly select a region of the target that is big enough to complete the I/O request.

Operational Notes

Physical targets running asynchronous workloads will use the class disk driver timeout value defined in the registry. Physical targets running synchronous workloads will default to the class disk driver timeout value defined in the registry, however, value can be manually adjusted in the AD.

Target Test Cluster Mode with Variable Transfer Size [TARGCVTEST]

I/O Direction	Workload Type	Bus Trigger	Automation	Data Comparison	Target Type	Target Coverage	T10 PI	Error Injection
Write followed by Read	Sequential or Random, asynchronous or synchronous	Yes	None, settings manually chosen by user, same workload used on all targets	Always On	Physical targets only	Full	Yes	No

Fixed Settings

Error On, Performance On, I/O Direction, Error Injection Off

Adjustable Settings

Thread Count, Transfer Size, Buffer Size, I/O Depth, Asynchronous On/Off, I/O Timeout Duration (applies to synchronous workloads only), Data Pattern, Random On/Off, T10 PI On/Off, Data Tags On/Off, and PRNG Seed Re-use On/Off, Atomizer On/Off.

Purpose

Perform I/O with error checking enabled to the entire target geometry using the same workload parameters across all targets, sharing the targets across multiple hosts using SCSI reserve/release or persistent reservations to control access using variable transfer sizes.

Variable Transfer Sizes

The individual I/O thread will randomly calculate the number of transfers required to fulfill the buffer size which is manually set by the user. The length of these transfers will be randomly calculated. In general, the larger the buffer size, the greater the number of variable transfers will be required.

Sequential Workload Description

For sequential workloads, odd numbered threads will start at the beginning of the target's geometry, and even numbered threads will start at the end of the target's geometry. Threads will use non-overlapping storage locations and work their way up and down the target geometry. After the thread completes each pass (up or down) the process will repeat.

Random Workload Description

For random workloads, each I/O thread will randomly select a region of the target that is big enough to complete the I/O request.

Operational Notes

Please see the DPD section regarding SCSI reserve/release functionality. Physical targets running asynchronous workloads will use the class disk driver timeout value defined in the registry. Physical

targets running synchronous workloads will default to the class disk driver timeout value defined in the registry, however, value can be manually adjusted in the AD.

Target Test Target Verify [TSVERIFYLIVE]

I/O Direction	Workload Type	Bus Trigger	Automation	Data Comparison	Target Type	Target Coverage	T10 PI	Error Injection
Write followed by Read	Sequential, asynchronous or synchronous	Yes	None, settings manually chosen by user, same workload used on all targets	Always On	Physical targets only	Full	No	No

Fixed Settings

Error On, Performance On, I/O Direction

Adjustable Settings

Thread Count, Transfer Size, Buffer Size, I/O Depth, Asynchronous On/Off, Data Pattern, Data Tags On/Off, Error Injection On/Off, percentage of geometry coverage per initialization loop.

Purpose

Initialize a portion of the target with a data pattern (percentage of target covered each pass can be set in the DPD), after initialization, verify data has not changed. After verification, perform initialization on next target area, if random data pattern selected, new random data pattern will be used for each initialization loop. Repeat process indefinitely until manually stopped or STOP timer reached.

Variable Transfer Sizes

Not used.

Sequential Workload Description

Each thread will be given its own area of target geometry to operate in. The number of blocks initialized per loop will be dependent on percentage setting used in Logical Target Size control in DPD.

Random Workload Description

Not used.

Operational Notes

If a purely sequential workload is desired, only use one thread per target.

Target Test Target Verify Cold Storage Initialization [TSCOLDINIT]

I/O Direction	Workload Type	Bus Trigger	Automation	Data Comparison	Target Type	Target Coverage	T10 PI	Error Injection
Write	Sequential, asynchronous or synchronous	No	None, settings manually chosen by user, same workload used on all targets	No	Physical targets only	Full	No	No

Fixed Settings

Error On, Performance On, I/O Direction

Adjustable Settings

Thread Count, Transfer Size, Buffer Size, I/O Depth, Asynchronous On/Off, Data Pattern, Data Tags On/Off, Error Injection On/Off, percentage of geometry coverage per initialization loop.

Purpose

Initialize target with data, metadata for each write transaction is saved to file for later use in Target Test Target Cold Verify. Test will run until end of target geometry is reached or STOP timer ends. Test may be used to verify targets ability to maintain data over long periods of cold storage (i.e., powered off), or targets ability to record data correctly during an abrupt power off event.

Variable Transfer Sizes

Not used.

Sequential Workload Description

Each thread will be given its own area of target geometry to operate in. The number of blocks initialized per loop will be dependent on percentage setting used in Logical Target Size control in DPD.

Random Workload Description

Not used.

Operational Notes

If a purely sequential workload is desired, only use one thread per target. Target may be powered off during initialization process, only metadata for valid transactions will be saved to file. Cold storage files saved to Perf Results directory defined in DPD.

Target Test Target Verify Cold Storage [TSCOLDVERIFY]

I/O Direction	Workload Type	Bus Trigger	Automation	Data Comparison	Target Type	Target Coverage	T10 PI	Error Injection
Read	Sequential, asynchronous or synchronous	Yes	None, settings manually chosen by user, same workload used on all targets	Yes, data compared against metadata files created in TSCOLDINIT.	Physical targets only	Full	No	No

Fixed Settings

Error On, Performance On, I/O Direction

Adjustable Settings

Thread Count, Transfer Size, Buffer Size, I/O Depth, Asynchronous On/Off, Data Pattern, Data Tags On/Off, Error Injection On/Off, percentage of geometry coverage per initialization loop.

Purpose

Initialize target with data, metadata for each write transaction is saved to file for later use in Target Test Target Cold Verify. Test will run until end of target geometry is reached or STOP timer ends. Test may be used to verify targets ability to maintain data over long periods of cold storage (i.e., powered off), or targets ability to record data correctly during an abrupt power event.

Variable Transfer Sizes

Not used.

Sequential Workload Description

Each thread will be given its own area of target geometry to operate in. The number of blocks initialized per loop will be dependent on percentage setting used in Logical Target Size control in DPD.

Random Workload Description

Not used.

Operational Notes

If a purely sequential workload is desired, only use one thread per target. Target may be powered off during initialization process, only metadata for valid transactions will be saved to file. Cold storage files must be in Perf Results directory defined in DPD.

Custom Tests

Random I/O [RANDIOTEST]

This functional test randomly selects workload parameters and periodically updates them on a per target basis. Full target coverage is not guaranteed.

I/O Direction	Workload Type	Bus Trigger	Automation	Data Comparison	Target Type	Target Coverage
Write followed by Read	Sequential or Random	Yes	Full	Always On	Logical and Physical	Partial to Full

Fixed Settings

Error On, Performance On, I/O Direction

Adjustable Settings

I/O Timeout Duration (Physical Only), Error Injection Timer (Physical Only), Data Pattern, Non-Aligned or Aligned I/O (Logical Only and must be enabled in the DPD) , Atomizer On/Off.

Purpose

Randomly select I/O parameters with error checking and performance monitoring always on. For logical targets (a.k.a file system or network shares), the default space usage is 80% of available free space but this setting can be changed in the DPD. Full target geometry will only be used with random workloads.

Automation

The following parameters will be randomly chosen on a per target basis: workload type, thread count, transfer size, buffer size, asynchronous on or off, random on or off, and I/O depth. These parameters will be updated once an hour with new values randomly selected. For logical targets, the volume may need to be initialized first before the test begins, this may increase the test time beyond one hour depending on the “Logical Target Size” setting.

Sequential Workload Description

For sequential workloads, a small area of the target will be used. This behavior will generate a mostly cache workload.

Random Workload Description

For random workloads, each I/O thread will randomly select a region of the target that is big enough to complete the I/O request.

Operational Notes

It is important to know the disk timeout value used by the disk class driver or your vendor's DSM driver. The DMS timeout value should be greater than or equal to this setting.

Variations on the Sequential Workload Theme

The AD provides four types of sequential workloads – Mercury, Venus, Earth and Jupiter. The first three are available on physical targets only, the last, logical targets only. The differences between Mercury, Venus and Earth lay in how the target geometry is accessed.

Mercury [MERCURYA, MERCURYB]

I/O Direction	Workload Type	Bus Trigger	Automation	Data Comparison	Target Type	Target Coverage
Write only, Read Only or Both	Sequential	No	None	Always Off	Physical	Full

Fixed Settings: Error Off, Performance On, Asynchronous On, Error Injection Timer Off, I/O Timeout Duration Off

Adjustable Settings: Thread count, Transfer Size, Buffer Size, I/O Depth, Data Pattern, and I/O Direction.

Purpose: Evenly divide the target geometry into equal segments based on I/O direction, buffer size, thread count and queue depth. I/O will be sequential starting at the beginning of the segment and walk the segment until the end is reached. When the end is reached, I/O will start at the beginning again. For example, if I/O direction = read, thread count = 10 and queue depth = 2, the target will be divided up into 10 equal segments; if I/O direction changed to both, the target would be divided into 20 equal segments. This setting may emulate an Exchange Mail configuration with each I/O thread representing a user mailbox.

Sequential Workload Description: Dividing the target into many segments and performing I/O to these segments simultaneously will induce a fair amount of thrashing while generating a low cache hit profile on read.

Random Workload Description: Not Supported.

Operational Notes: There are two versions of this test: with initialization (MERCURYA) and without initialization (MERCURYB). This refers to data pattern initialization. The data pattern will always be used on write direction, but on reads it will not be used unless initialization was selected. When initialization is selected, the data pattern will first be written to the entire target geometry before any read operation.

Venus [VENUSA, VENUSB]

I/O Direction	Workload Type	Bus Trigger	Automation	Data Comparison	Target Type	Target Coverage
Write only, Read Only or Both	Sequential	No	None	Always Off	Physical	Full

Fixed Settings: Error Off, Performance On, Asynchronous On, Error Injection Timer Off, I/O Timeout Duration Off

Adjustable Settings: Thread count, Transfer Size, Buffer Size, I/O Depth, Data Pattern, and I/O Direction.

Purpose: Threads work together to perform I/O to the entire target geometry using adjacent non-overlapping regions, incrementally moving up the geometry until the end is reached.

Sequential Workload Description: Each thread requests a target region large enough to complete its I/O transfer. When the I/O transfer is finished, the thread will request a new and higher region that is as close as possible to its current location. By doing so, the threads will work together to walk the entire target geometry from beginning to end using non-overlapping adjacent regions. The same region may be used more than once as one thread releases an area just before another thread requests the same location, this will produce some cache hits on READ.

Random Workload Description: Not Supported.

Operational Notes: There are two versions of this test: with initialization (VENUSA) and without initialization (VENUSB). This refers to data pattern initialization. The data pattern will always be used on write direction, but on reads it will not be used unless initialization was selected. When initialization is selected, the data pattern will first be written to the entire target geometry before any read operation. This workload may be like Microsoft's SQLIO application set to sequential mode.

Earth [EARTHA, EARTHB]

I/O Direction	Workload Type	Bus Trigger	Automation	Data Comparison	Target Type	Target Coverage
Write only, Read Only or Both	Sequential	No	None	Always Off	Physical	Full

Fixed Settings: Error Off, Performance On, Asynchronous On, Error Injection Timer Off, I/O Timeout Duration Off

Adjustable Settings: Thread count, Transfer Size, Buffer Size, I/O Depth, Data Pattern, and I/O Direction.

Purpose: Threads work together to perform I/O to the entire target geometry using adjacent non-overlapping regions, incrementally moving up the geometry until the end is reached.

Sequential Workload Description: Each thread requests a target region large enough to complete its I/O transfer. When the I/O transfer is finished, the thread will request a new and higher region that is as close as possible to its current location. By doing so, the threads will work together to walk the entire target geometry from beginning to end using non-overlapping adjacent regions. The same region will not be used more than once as the threads traverse their way up the target geometry. This is the only difference between the Venus and Earth test. By doing so, cache hits (excluding pre-fetch) should not occur on READ.

Random Workload Description: Not Supported.

Operational Notes: There are two versions of this test: with initialization (EARTHA) and without initialization (EARTH B). This refers to data pattern initialization. The data pattern will always be used on write direction, but on reads it will not be used unless initialization was selected. When initialization is selected, the data pattern will first be written to the entire target geometry before any read operation.

Jupiter [JUPITER]

I/O Direction	Workload Type	Bus Trigger	Automation	Data Comparison	Target Type	Target Coverage
Write only, Read Only or Both	Sequential	No	None	Always Off	Logical	Full

Fixed Settings: Error Off, Performance On, Asynchronous On, Error Injection Timer Off, I/O Timeout Duration Off

Adjustable Settings: Thread count, Transfer Size, Buffer Size, I/O Depth, Data Pattern, I/O Direction, and Percentage of Target Used.

Purpose: Threads work together to perform I/O to 80% of the target geometry using individual files. Each I/O thread file will be of equal size; together they will consume 80% of available free space on the target. Space consumption can be adjusted higher or lower using the “Logical Target Size” DPD setting, 80% is default value. Each thread will increment the file pointer after each I/O until end of file is reached, then repeat from the beginning.

Sequential Workload Description: Each thread creates its own file and initializes with selected data pattern. When initialization complete, the thread will begin I/O at file offset equal to zero. After each I/O, the file pointer will automatically increment until end of file is reached at which time pointer position will be reset to zero and the process will begin again. This workload will not generate cache hits on READ (excluding pre-fetch).

Random Workload Description: Not Supported.

Operational Notes: There is only one version of this test – with initialization. This refers to data pattern initialization. The data pattern will always be used on write and read directions. For logical targets, a file must always be written to before it can be read.

Saturn [SATURNA, SATURNB]

I/O Direction	Workload Type	Bus Trigger	Automation	Data Comparison	Target Type	Target Coverage
Write only, Read Only	Sequential	No	None	Always Off	Physical	Full

Fixed Settings: Error Off, Performance On, Asynchronous On, Error Injection Timer Off, I/O Timeout Duration Off

Adjustable Settings: Thread count, Transfer Size, Buffer Size, I/O Depth, Data Pattern, Custom Timer and I/O Direction.

Purpose: Characterize sequential performance as a function of LBA position. Threads work together to perform I/O to the entire target geometry using adjacent non-overlapping regions, incrementally moving up the geometry until the end is reached. Data samples of Throughput, IOPS, latency and LBA are taken periodically based on Custom timer setting. For example, if custom test timer set to 2 minutes, data samples will be captured every 2 seconds over a 2-minute interval and repeat. When the end of the geometry is reached, the test stops and displays the following results which can also be saved to comma delimited file:

- Maximum throughput and associated LBA range
- Minimum throughput and associated LBA range
- Maximum IOPS and associated LBA range
- Minimum IOPS and associated LBA range
- Maximum latency and associated LBA range
- Minimum latency and associated LBA range

Sequential Workload Description: Same as EARTH.

Random Workload Description: Not Supported.

Operational Notes: There are two versions of this test: with initialization (SATURNA) and without initialization (SATURNB). This refers to data pattern initialization. The data pattern will always be used on write direction, but on reads it will not be used unless initialization was selected. When initialization is selected, the data pattern will first be written to the entire target geometry before any read operation.

Uranus [URANUS]

I/O Direction	Workload Type	Bus Trigger	Automation	Data Comparison	Target Type	Target Coverage
Custom	Sequential	No	None	Always Off	Physical	Custom

Fixed Settings: Error Off, Performance On, Asynchronous On, Error Injection Timer Off, I/O Timeout Duration Off

Adjustable Settings: Thread count, Transfer Size, Buffer Size, I/O Depth, Data Pattern, Custom Timer and I/O Direction defined in custom profile mapped to target.

Purpose: Characterize sequential performance as a function of LBA position; the sequential workloads used are defined by custom profile mapped to target. When the end of the geometry is reached, the test stops and displays the following results which can also be saved to comma delimited file:

- Maximum throughput and associated LBA range
- Minimum throughput and associated LBA range
- Maximum IOPS and associated LBA range
- Minimum IOPS and associated LBA range
- Maximum latency and associated LBA range
- Minimum latency and associated LBA range

Sequential Workload Description: Defined in custom profile.

Random Workload Description: Not Supported.

Operational Notes: Custom profiles with a random workload component cannot be mapped to target selected for this test. Test results will be identical to Saturn.

Custom Profile [CUSTOM]

I/O Direction	Workload Type	Bus Trigger	Automation	Data Comparison	Target Type	Target Coverage
Custom	Combination of Sequential and Random	No	None	Always Off	Physical	Custom

Fixed Settings: Error Off, Performance On, Asynchronous On, Error Injection Timer Off, I/O Timeout Duration Off

Adjustable Settings: Thread count, Transfer Size, Buffer Size, I/O Depth, Data Pattern, Custom Timer and I/O Direction defined in custom profile mapped to target.

Purpose: Generate custom workloads that have varying degree of read and write I/O direction and varying degree of sequential and random components.

Sequential Workload Description: Defined in custom profile.

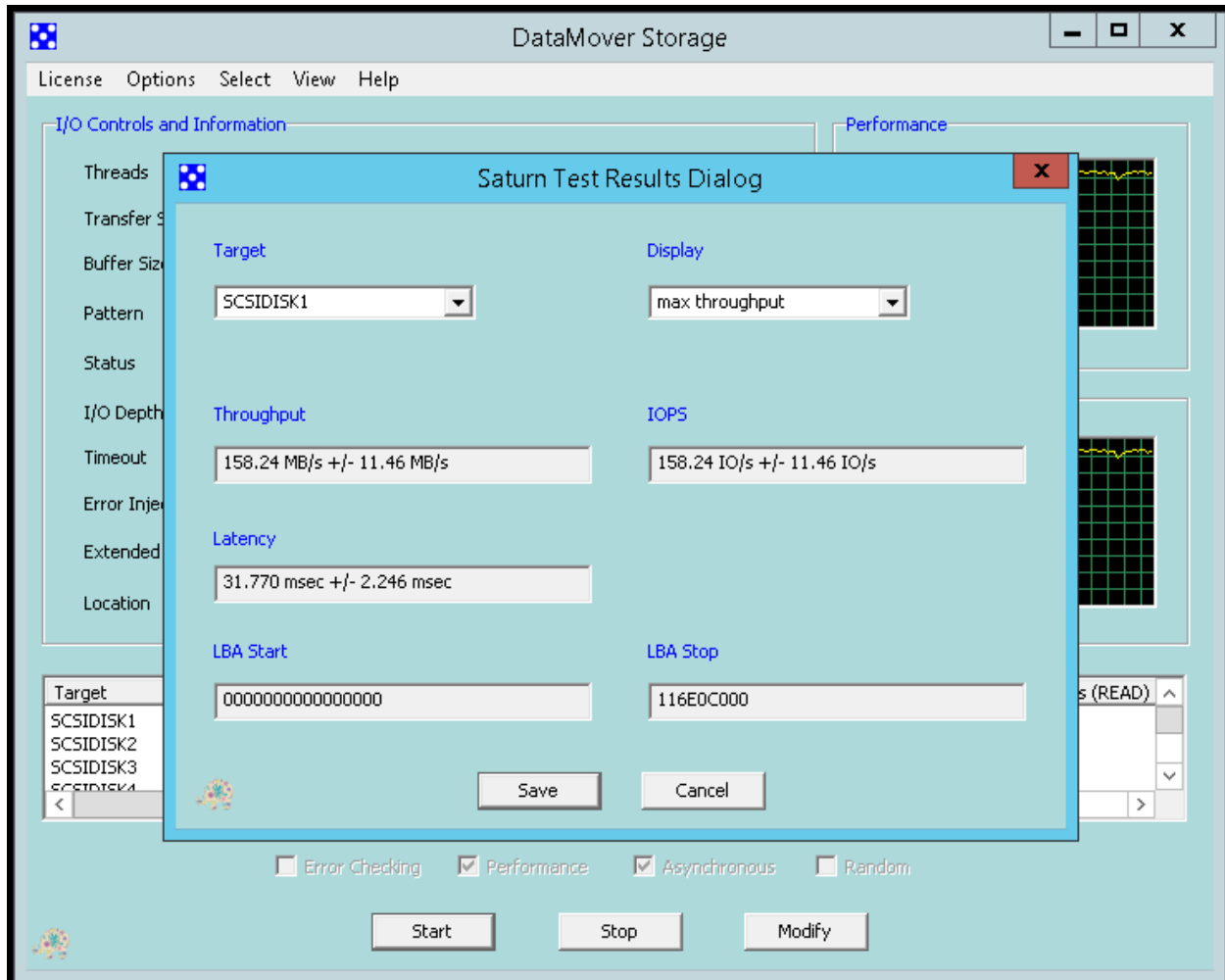
Random Workload Description: Defined in custom profile.

Operational Notes: All targets must have at least one custom profile mapped to it. If this is not the case, test selection will be aborted.

Saturn Test Results

The Saturn Test Results Dialog (STRD) displays the results of test Saturn when it finished.

Screenshot of STRD:



GUI Items

Target

Dropdown control allowing the user to select target results to be displayed.

Display

Dropdown control allowing the user to select test results to view max or min throughput, max or min IOPS, max or min latency.

Throughput

Mean throughput showing associated standard deviation.

IOPS

Mean IOPS showing associated standard deviation

Latency

Mean IOPS showing associated standard deviation.

LBA Start

The start LBA of range (in 8-byte hexadecimal format) for the given test result.

LBA Stop

The stop LBA of range (in 8-byte hexadecimal format) for the given test result

DataMover Preferences

The DataMover Preferences Dialog (DPD) allows the users to make persistent settings related to web proxy, general I/O behavior and performance workload operation.

Screenshot of DPD:

DataMover Preferences

Options

Network Settings

HTTP Proxy: J Port: 0

☐ Enable

Performance Estimator

Time Duration: 9 hrs Combinations: 1176

Advanced I/O Settings

Logical Target Size: 80%

Async Timeout Value: 4 min

Max Thread Count: 10

Max I/O Depth: 16

☐ Enable Extended Perf ☐ Enable Perf Bypass

☐ Enable Non-Sector Aligned I/O ☐ Single Shot

☐ Disable Default Workloads ☐ Stop On Error

☐ Disable Perf Streaming ☐ Strip Chart Recorder

☐ Enable T10 PI

Increment By: 1 C:\

Ramp Time: 30 sec **Perf Results**

Custom Test: 30 min (0.5 hr)

Stop Timer: DISABLE

Logical Target Size Estimator

Target	Type	Free Space Used
C:\	FIXED DISK	25.9 GB

Cluster Settings

☐ Enable SCSI PR PR Service Action: Reg&Ignore

PR Type: WrExRegOnly

Aggressiveness: Low 100% High

OK Cancel Reset

Menu Items

Options Performance Bypass Parameters - Displays the Performance Bypass Dialog (PBD).

Network Settings

HTTP Proxy – The DNS name of the proxy server.

Port – The TCP port used by the proxy server.

Enable – Checking this checkbox instructs DMS to automatically use the proxy server when trying to contact the StorageWerks license server. For newer web browsers, this setting should not be required; the browser will contact the proxy automatically making this operation transparent to the application.

Advanced I/O Settings

Logical Target Size – This slider control allows the user to adjust the amount of free space DMS will consume on logical targets. Default setting is 80%. This setting will be used for most performance and functional tests excluding CACHE only workloads where file size will always be minimized by default. Reducing the size of free space used reduces the time required to initialize the file.

Async Timeout Value – This slider control allows the user to adjust the amount of time DMS waits for asynchronous operations to complete. Default setting is 4 minutes. This is not the same as I/O completion time which is defined by the class disk driver timeout value setting in the registry. This value should be set larger than the class disk driver timeout.

Max Thread Count – This slider control allows the user to change the maximum thread count value used by the extended performance tests. Default value is 10.

Max I/O Depth – This slider control allows the user to change the maximum I/O depth value used by the extended performance tests. Default value is 16.

Enable Extended Perf – This checkbox control enables or disables the extended performance tests. Default setting is disabled.

Disable Perf Streaming – This checkbox control enables or disables the large buffer setting used in performance workloads. By default, a buffer size of 4MiB will be used for performance testing; disabling this feature causes transfer size and buffer size to always be equal. Using a smaller buffer size reduces the number of contiguous storage locations required to complete a transfer.

Enable T10 PI – This checkbox control enables or disables T10 Protection Information testing on physical targets when any of the Target Tests are selected. Default setting is disabled.

Stop All on Error – This checkbox control enables or disables the ability for all I/O threads to stop on error. Default setting is disabled, normally, only the I/O thread that identifies the error will stop while all other I/O threads continue to run. This setting can be helpful in troubleshooting.

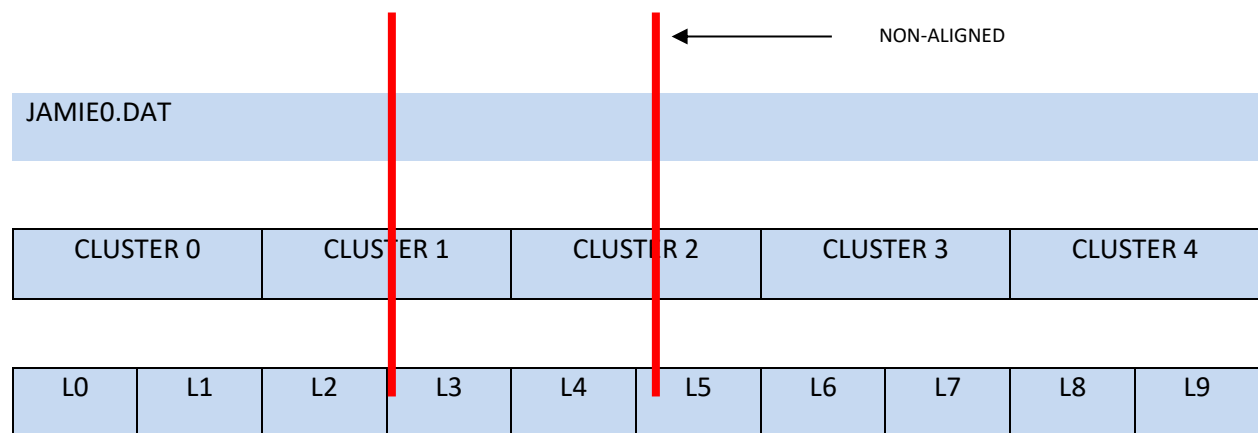
Enable PRNG Seed ReUse – This checkbox control enables or disables pseudo random number generator seed re-use. Default setting is disabled. Tests such as Random I/O and Random Target Standalone will randomly select workload parameters and it may be helpful to repeat the same settings in troubleshooting. In general, most of the DMS operations that rely on randomization will be repeatable when this feature is enabled. Random lbas and file pointer offsets as an example. One area that cannot be reproduced run-to-run are the random data patterns, DMS uses Microsoft cryptographic services which do not allow the seed values to be adjusted.

Enable Data Tags – This checkbox control enables or disables unique tags that can be added to the data buffer. Default setting is disabled. The format of the tag is [hostname]:[target number]:[thread number]:[data offset number] for synchronous I/O and [hostname]:[target number]:[thread number]:[iodepth number]:[data offset number]. When this feature is enabled, a tag will be inserted every 128 bytes in the data buffer, format is human readable, not hexadecimal.

Atomizer

This checkbox control enables or disables the atomizer feature. Atomizer applies to random workloads only. When enabled, the starting LBA location and number of contiguous blocks selected will satisfy the transfer size; when disabled, the starting LBA location and number of contiguous blocks selected will satisfy the buffer size. For example, if a workload has a transfer size equal to 512B and a buffer size equal to 4MB, when this feature is enabled, there will be 8192 randomly selected LBA locations of length equal to 1 block that will be accessed to satisfy this buffer size; when this feature is disabled, there will be one contiguous region of 8192 blocks that will be accessed to satisfy this buffer size.

Enable Non-Sector Aligned I/O – This checkbox control enables or disables non-sector aligned I/O. Default setting is disabled. This setting will only be used when performing RANDOM I/O to logical targets and will allow DMS to perform I/O to file locations that do not align with file system cluster boundaries and/or overlap cluster boundaries depending on the transfer size. What is RMW? Read first, modify, then write occurs when I/O is not cluster aligned. This behavior is illustrated below. The file JAMIE0.DAT consumes file system clusters 0 thru 4; in turn, each cluster consumes two blocks of physical storage.



Process A indexes into file JAMIE0.DAT to perform a write on the file location between the red lines which overlaps clusters 1 and 2. To accomplish this task, the file system driver will need to read in cluster locations 1 and 2; modify the data in system memory with the new information, then write back the entire clusters 1 and 2. We can see that the information updated is less than the final information written. This behavior demonstrates read-modify-write. When this feature is enabled, all the host's memory may be consumed depending on the size of the files; Microsoft refers to this as buffered I/O – the operating system will attempt to store all file contents in host system memory. This is not a memory leak induced by DMS. RMW behavior is great way to stress a file system driver and underlying block storage device.

Disable Default Workloads – This checkbox enables or disables the default performance workloads. Default setting is unchecked, or default workloads enabled.

Increment By – This dropdown control allows the user to select the increment setting used in the extended performance tests. Default value is 5. A larger value reduces the number of combinations processed and reduces test time.

Enable Perf Bypass - This checkbox enables or disables the performance bypass feature. At least one transfer size in the Performance Bypass Dialog (BPD) must be selected for this feature to be enabled. Default setting is disabled.

Single Shot

This checkbox enables a single iteration on some tests that would normally loop indefinitely. This setting can be used on the following combinations of settings for logical and physical targets:

- Logical target, performance on, error off, synchronous and asynchronous, any IO direction
- Physical target, performance on, error off, synchronous and asynchronous, any IO direction

Default setting is disabled.

Stop on Error

This checkbox allows the user to stop DMS on error. By default, individual IO threads always stop on error. This setting does not change this behavior; however, it does stop any of the automated tests from continuing their next iteration if any FAIL condition has been reported. For example, if Target Test Random Standalone Mode is running and this feature is enabled, the next iteration loop of this test will stop if a FAIL condition status is present. Default setting is disabled.

Strip Chart Recorder

This checkbox allows the user to enable and disable the strip chart recorder function. When strip chart recorder is enabled, DMS will log to a file all performance metrics until the application is stopped. Default setting is disabled.

Perf Results

This button, when pressed, allows the user to select a folder destination for the performance test results. The location of the folder is displayed in the window above the control. By default, if no folder has been selected, the destination will be the DMS installation directory. This setting is useful for performance tests which are run in automation mode.

Ramp Time - This dropdown control allows the user to select the performance test ramp time. Ramp time is the time DMS waits before collecting performance data. This timer does not start until all threads have reached RUN state. Default setting is 30 seconds.

Custom Test - This dropdown control allows the user to select the custom test time duration. The timer does not start until all threads have reached RUN state. This timer is used by the Target Tests Random Standalone Mode and Random Cluster Mode, and Custom Tests Random I/O Test. When the timer expires, these tests will restart with newly selected I/O parameters. Default setting is 30 minutes.

Stop Timer

This dropdown control allows the user to select the amount of time any test can run (if any performance test is selected, this timer is ignored). Timer resolution is in 1-minute intervals, ranging from 1 min to 72 hours. When the timer is reached, DMS will automatically stop the test. This setting comes in handy if using DMS in automation mode. Default setting is DISABLED.

Performance Estimator

This section estimates the time it will take to complete the performance test based on the workload parameters selected in Advanced I/O Settings and displays the number of workload iterations that will be executed.

Time Duration – The estimated time in hours that a given performance test will take to complete.

Combinations – The number of workload parameter combinations that will be executed for a given performance test based on the Advanced I/O Settings.

Logical Target Size Estimator

When logical targets have been selected in the BD or AD, this GUI item will calculate and display the amount of free space consumed on each target based on Logical Target Size setting made in Advanced I/O Settings section. When physical targets have been selected, it will display the targets showing their SCSI Inquiry information; currently, there is no calculator associated with physical target type.

Cluster Settings

Enable SCSI PR

This checkbox enables or disables SCSI persistent reserve/release (R/R) functionality. By default, this control is unchecked, meaning that old SPC-2 style reserve/release is enabled.

Aggressiveness

This slider control provides a mechanism to throttle the reserve/release requests coming from DMS, ranging from 0 to 100%. The slide range determines the maximum wait time that the DMS will stall prior to making the next R/R request. By default, this setting is 100% (most aggressive) which means DMS will not insert wait states prior to the next R/R request. StorageWerks recommends lowering the aggressive value in relation to the number of DMS hosts accessing same LUN. For example, if five DMS nodes are accessing same LUN, an aggressive setting of 20% for each DMS is suggested. This prevents one instance of DMS from monopolizing the ownership of the LUN.

PR Service Action

This dropdown control allows the user to select Register or Register and Ignore Existing Key service action types. The default setting is Register and Ignore Existing Key.

PR Type

This dropdown control allows the user to select persistent reservation type. The default setting is Write Exclusive – Registrants Only.

Reset – This button resets all controls in the DPD back to their default settings.

OK – This button saves any changes made in the DPD.

Cancel – This button cancels any changes made in the DPD.

Performance Tests

Introduction

DMS supports three performance workload type categories – CACHE, WALK and RANDOM. Each category offers a quick and long version of the test. The quick tests typically last less than 30 minutes, while the long tests may take more than twenty-four hours to complete. The quick tests are not meant

to provide the best results, but in some cases they may, their purpose is to provide a short duration test providing a general indication of system performance.

Theory and Operation

The three workload type categories will be explained in this section. Each workload type is meant to test a specific characteristic of the target device. These characteristics may include minimizing target controller cache memory hits, maximizing target controller cache memory hits and maximizing target controller pre-fetch cache memory hits while minimizing target controller cache memory hits. These attributes will help characterize the system and provide insight into the maximum throughput and IOPS possible for all workload category types. The automated DMS performance tests will try many different workload parameter combinations to achieve the best results. In this case, best means highest throughput (in MiB/s) and IOPS measured. Workload parameters that DMS adjusts include I/O type (synchronous or asynchronous), thread count per target, I/O depth per target and buffer size.

Each workload category test provides the same measurement results which can be saved to a comma-delimited file when the test ends. Three basic I/O directions are tested: Half-duplex READ, half-duplex WRITE and full duplex. What is meant by half-duplex and full duplex? Modern I/O technologies support the ability to transmit and receive data at the same time; this is what is meant by full duplex. Half-duplex implies that only one direction (transmit or receive) is being used at one time. Half-duplex READ means that the host is performing 100% reads from the target; half-duplex WRITE means that the host is performing 100% writes to the target and full duplex means the host is performing 50% reads from the target and 50% writes to the target simultaneously. DMS provides the following metrics over fourteen different transfer sizes ranging from 512B to 4MiB:

- TP (MB/s) - The throughput mean value derived over many samples for given transfer size. For full duplex, this will be total throughput for both I/O directions.
- TP Standard Deviation (MB/s) – The standard deviation of the throughput mean value for a given transfer size. This value will give an indication of variation in data samples.
- IOPS - The IOPS mean value derived over many samples for given transfer size. For full duplex, this will be total IOPS for both I/O directions.
- IOPS Standard Deviation - The standard deviation of the IOPS mean value for a given transfer size. This value will give an indication of variation in data samples.
- LAT (ms) - The latency mean value derived over many samples for given transfer size. For full duplex, this will be total latency for both I/O directions.
- LAT Standard Deviation (ms) - The standard deviation of the latency mean value for a given transfer size. This value will give an indication of variation in data samples.
- LATR (ms) - The latency mean value for I/O READ direction only derived over many samples for given transfer size. This parameter only available with full duplex.
- LAT Standard Deviation (ms) - latency mean value for I/O READ direction only derived over many samples for given transfer size. This parameter only available with full duplex.
- LATW (ms) - The latency mean value for I/O WRITE direction only derived over many samples for given transfer size. This parameter only available with full duplex.

- LAT Standard Deviation (ms) - latency mean value for I/O WRITE direction only derived over many samples for given transfer size. This parameter only available with full duplex.
- CPU (%) - The “CPU” utilization mean value derived over many samples for given transfer size.
- CPU Standard Deviation (%) - The standard deviation of the “CPU” utilization mean value for a given transfer size.

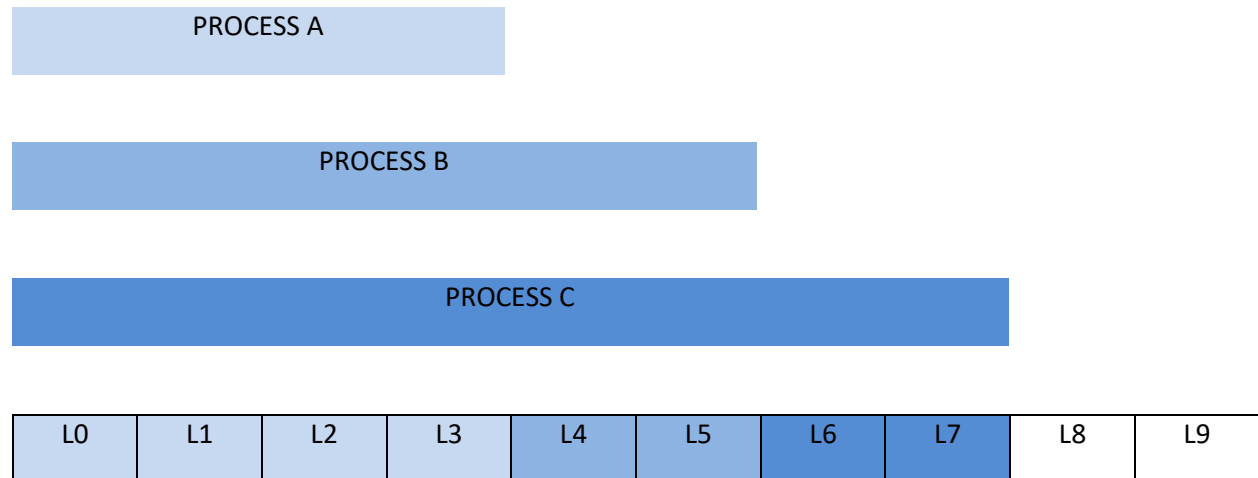
For any given iteration of a performance workload, performance metric measurements will not start until all threads have reached RUN phase. This attribute guarantees that the correct workload will be applied and measured.

Workload Descriptions

What is meant by the terms CACHE, WALK and RANDOM and are they the same for logical and physical targets? To answer this question, we must first define what is meant by overlapping, non-overlapping and sequential I/O. Overlapping I/O occurs when the same location in a file or on a block device (a.k.a hard disk, iSCSI LUN, Fibre Channel LUN, etc.) is accessed by more than one thread or process at the same time. Non-overlapping, I/O occurs when the same location in a file or on a block device (a.k.a hard disk, iSCSI LUN, Fibre Channel LUN, etc.) is not accessed by more than one thread or process at the same time. Sequential I/O occurs when one or more consecutive locations in a file or a block device are accessed serially to complete a transfer.

Overlapping and Non-Overlapping I/O Descriptions

The graphic below illustrates what is meant by overlapping and non-overlapping I/O. In this example, process A, B and C access locations L0-L3 simultaneously; process B and C access locations L4-L5 simultaneously, and process C has sole access to locations L6-L7.



Overlapping I/O occurs on locations L0-L3 and L4-L5, non-overlapping I/O occurs on L6-L7. Modern storage systems use memory (referred to as cache) to increase performance on overlapping I/O operations. Typically, the example about to be given applies to the READ I/O direction only; cache implementations for the WRITE I/O direction are sometimes referred to as write-through and write-back implementations and behave in a different manner. For the READ case, when locations L0-L7 are first accessed, the storage target will copy the contents of these locations to its memory. Subsequent access to locations L0-L7 will reside in memory, and access to the block device media to retrieve the data will not be required. This type of operation greatly enhances the performance of the storage system because the memory has smaller access retrieval times compared to the block device media. DMS takes advantage of this behavior in its CACHE performance tests.

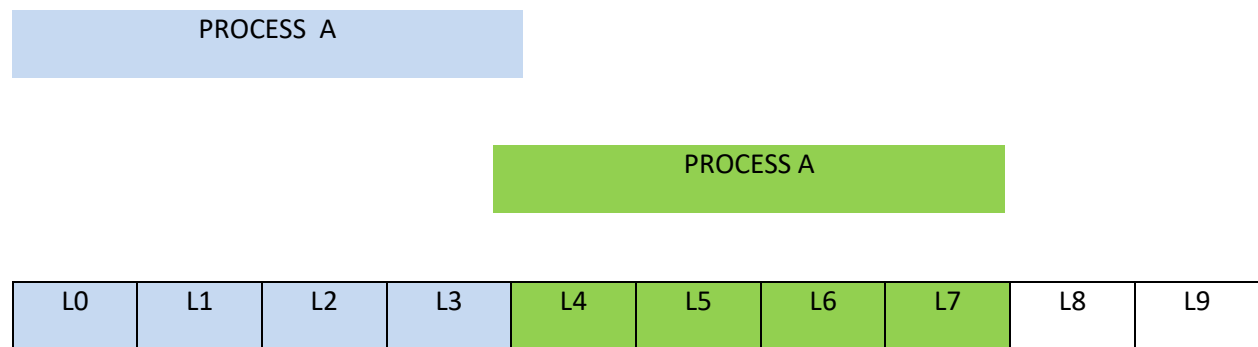
Sequential I/O Description

In essence, I/O to a file or block device is always sequential regardless of the workload being of type CACHE, WALK or RANDOM. As stated previously, sequential I/O occurs when one or more consecutive locations in a file or a block device are accessed serially to complete a transfer. This is true for all workload types. When performing RANDOM workloads, the randomness of the I/O will be determined by the transfer size and the starting location of the I/O. A smaller transfer size requires fewer consecutive storage locations to complete the transfer, and selection of the starting location determines the randomness of the I/O.

Consider a host bus adapter (HBA) that has a maximum transfer size of 256KiB, and we want to perform two random I/O operations using a 4MiB buffer size which we will refer to as IOP1 and IOP2. IOP1 will use a transfer size of 512B, and IOP2 will use a transfer size 4MiB. IOP1 will require 8192 512B transfers to fulfill the 4MiB buffer; IOP2 will require $(4\text{MiB} * 1024) \div 256\text{KiB}$ or 16 256KiB transfers to fulfill the

4MiB buffer. If each transfer for IOP1 and IOP2 will start at a randomly selected location, we see that IOP1 is 99% more random than IOP2 because it requires 8176 more transfers to complete the 4MiB buffer compared to IOP2. Also, since we know that all I/O is essentially sequential once the starting location has been chosen, IOP2 with its larger transfer size will benefit from the storage system's pre-fetch caching algorithms on READ I/O direction. Modern storage systems use a process called pre-fetch that tries to guess the next locations that will be accessed, pre-populating the storage cache with this data. If the guess is correct, the data will be accessed from the cache not the block device, increasing performance. DMS's WALK performance test takes advantage of this behavior to isolate the pre-fetch characteristics of the system while minimizing controller cache hits caused by overlapping I/O.

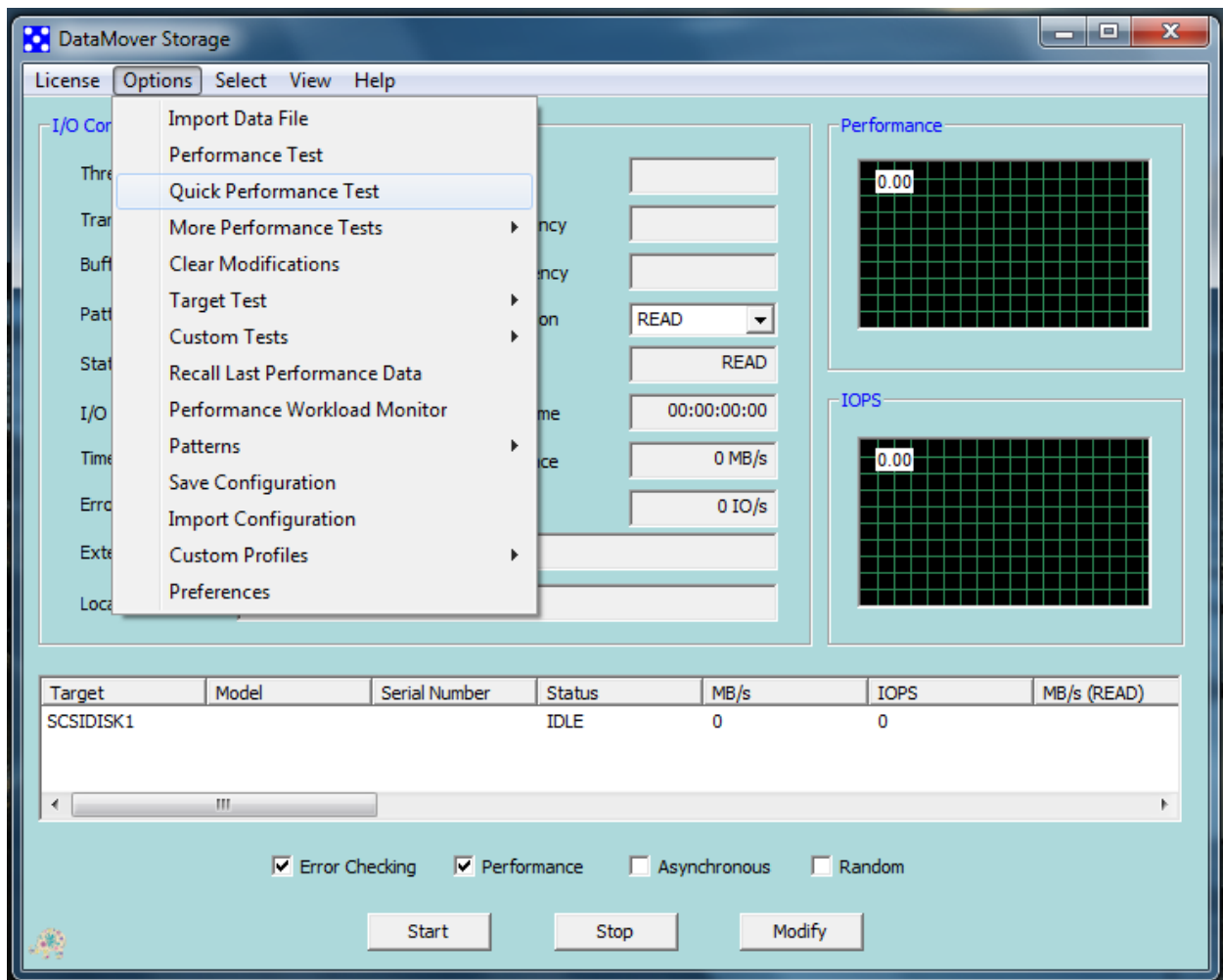
The graphic below illustrates sequential I/O and pre-fetch operations. Process A accesses four locations L0-L3 using sequential I/O with L0 as the starting location. In this example, the storage device's pre-fetch algorithm guesses that the next four locations (L4-L7) will be accessed and pre-populates its memory with this data. When process A accesses the next four locations, the storage device will retrieve this data from memory not the block device media improving performance.



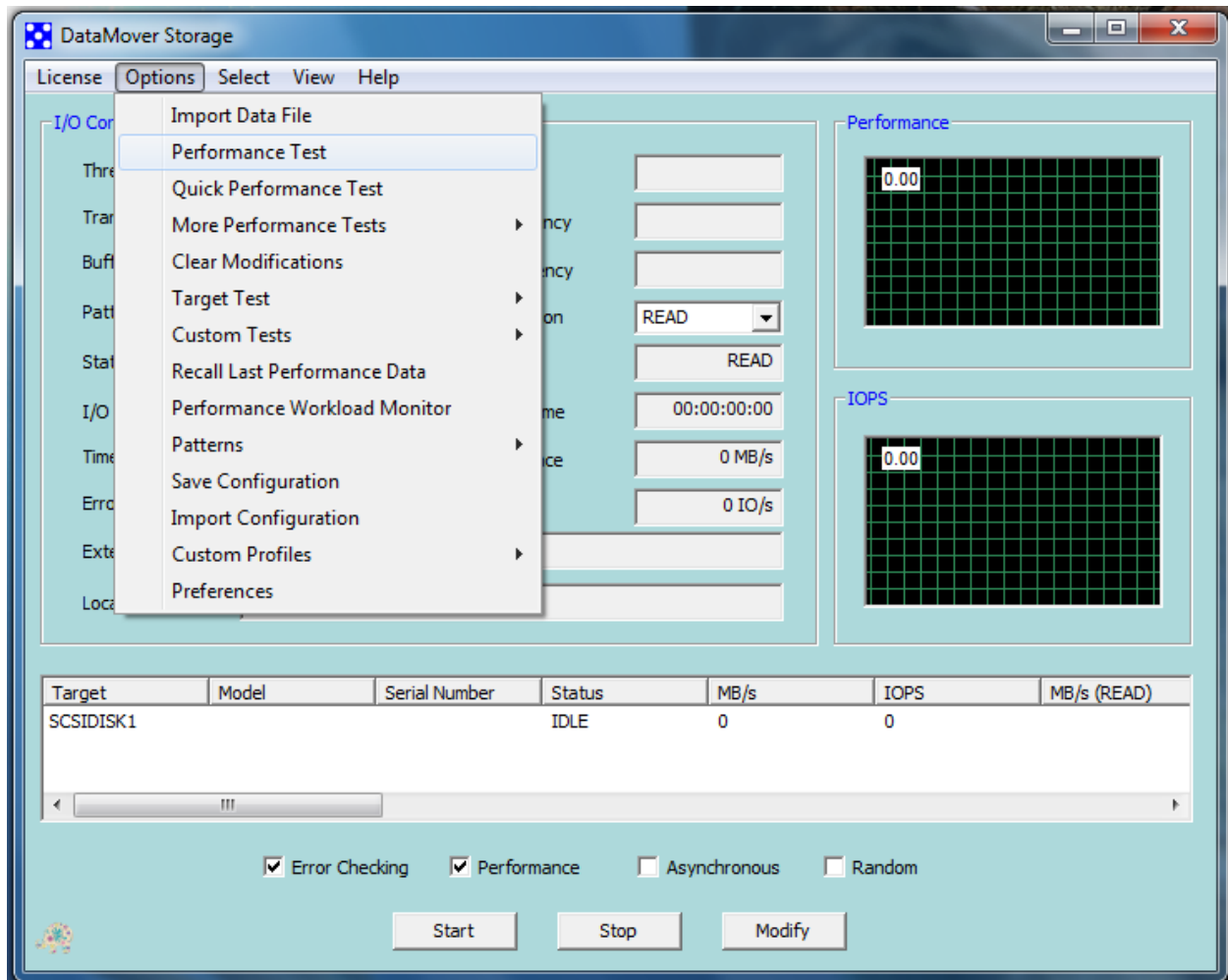
DMS takes advantage of this behavior in its WALK performance test by using non-overlapping I/O to isolate the pre-fetch characteristics of the target device.

CACHE Workload Tests

The CACHE workload uses repeating, non-overlapping I/O to maximize cache hits on the target storage device. The quick test collects data on 512B and 1MiB transfer sizes for all I/O directions using a limited combination of workload parameters. Test duration is typically less than 30 minutes. The small transfer size is best for maximum IOPS, and the larger transfer size is best for maximum throughput. Start this test by selecting the "Options->Quick Performance Test" menu item in the AD and clicking "Start," "QPERFTEST" will be displayed in Operation window.

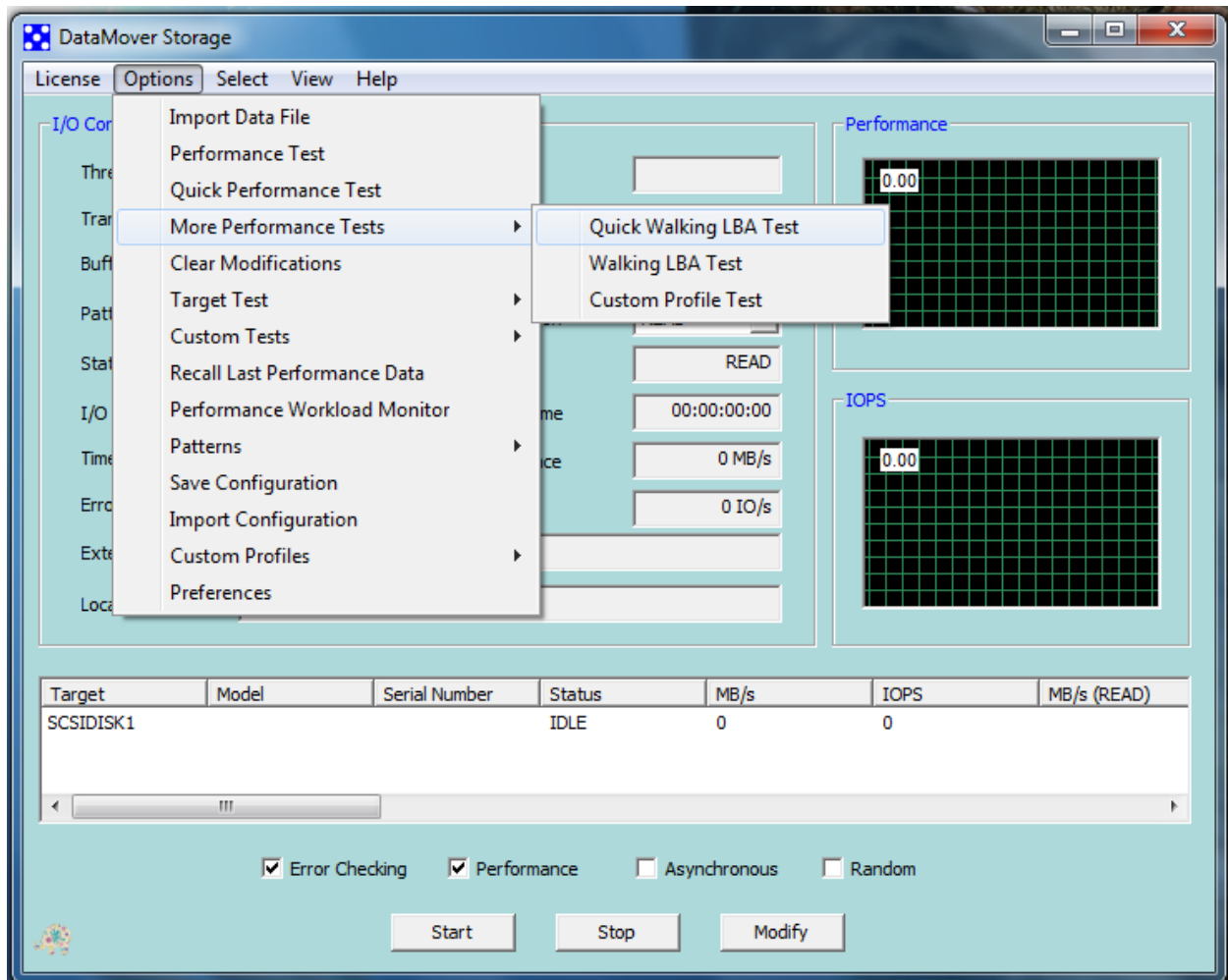


The long version of the test uses similar algorithms but test hundreds of combinations of workload parameters over many different transfer sizes. Test duration is typically > 24 hours. The long version may be started by selecting the “Options->Performance Test” menu item in the AD and clicking “Start,” “PERFTEST” will be displayed in Operation window.

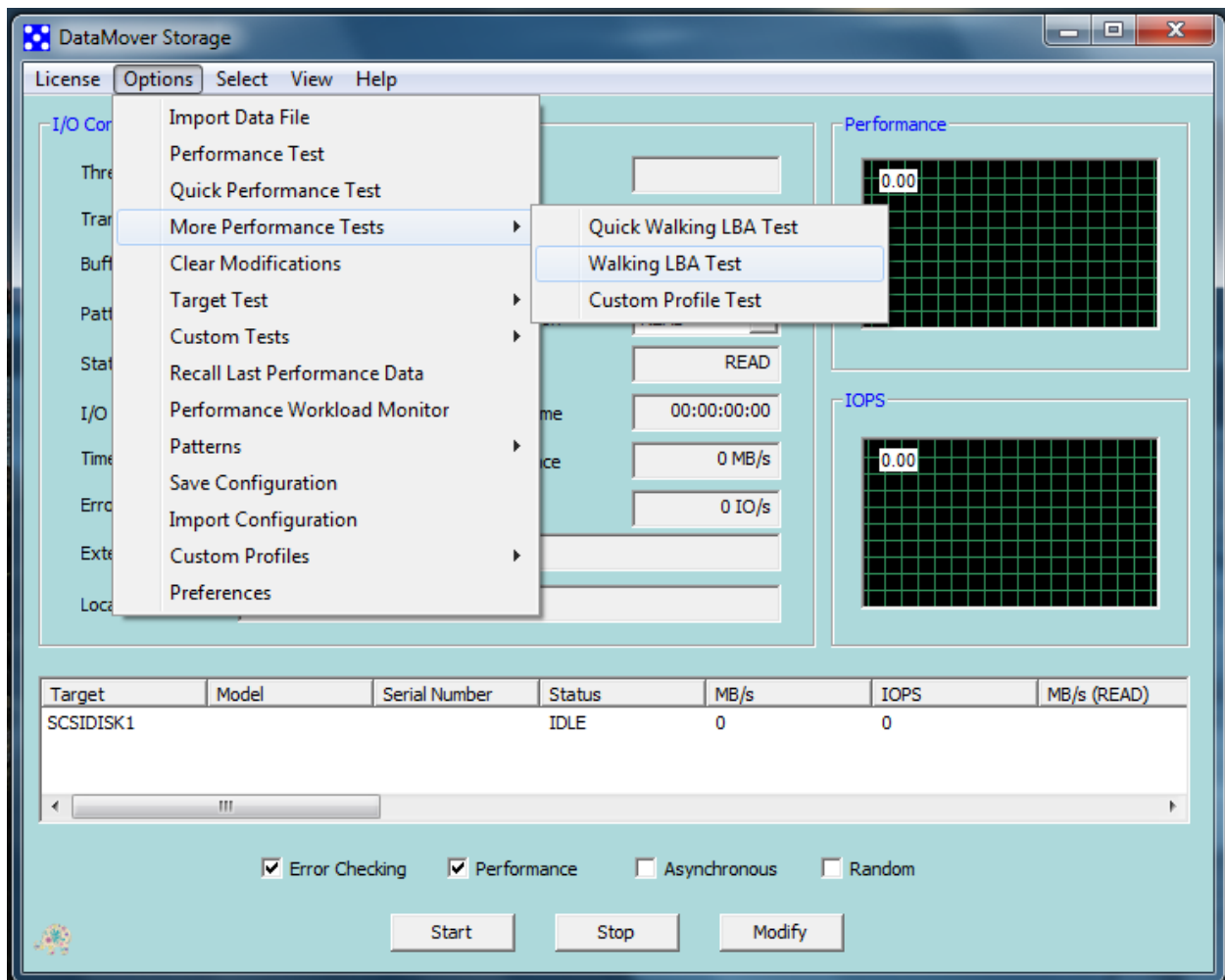


WALK Workload Tests

The WALK workload uses non-repeating, non-overlapping I/O to maximize pre-fetch cache hits on the target storage device. The quick test collects data on 512B and 1MiB transfer sizes for all I/O directions using a limited combination of workload parameters. Test duration is typically less than 30 minutes. The small transfer size is best for maximum IOPS, and the larger transfer size is best for maximum throughput. Start this test by selecting the “Options->More Performance Tests->Quick Walking LBA Test” menu item in the AD and clicking “Start,” “QPERFWLKTST” will be displayed in Operation window.

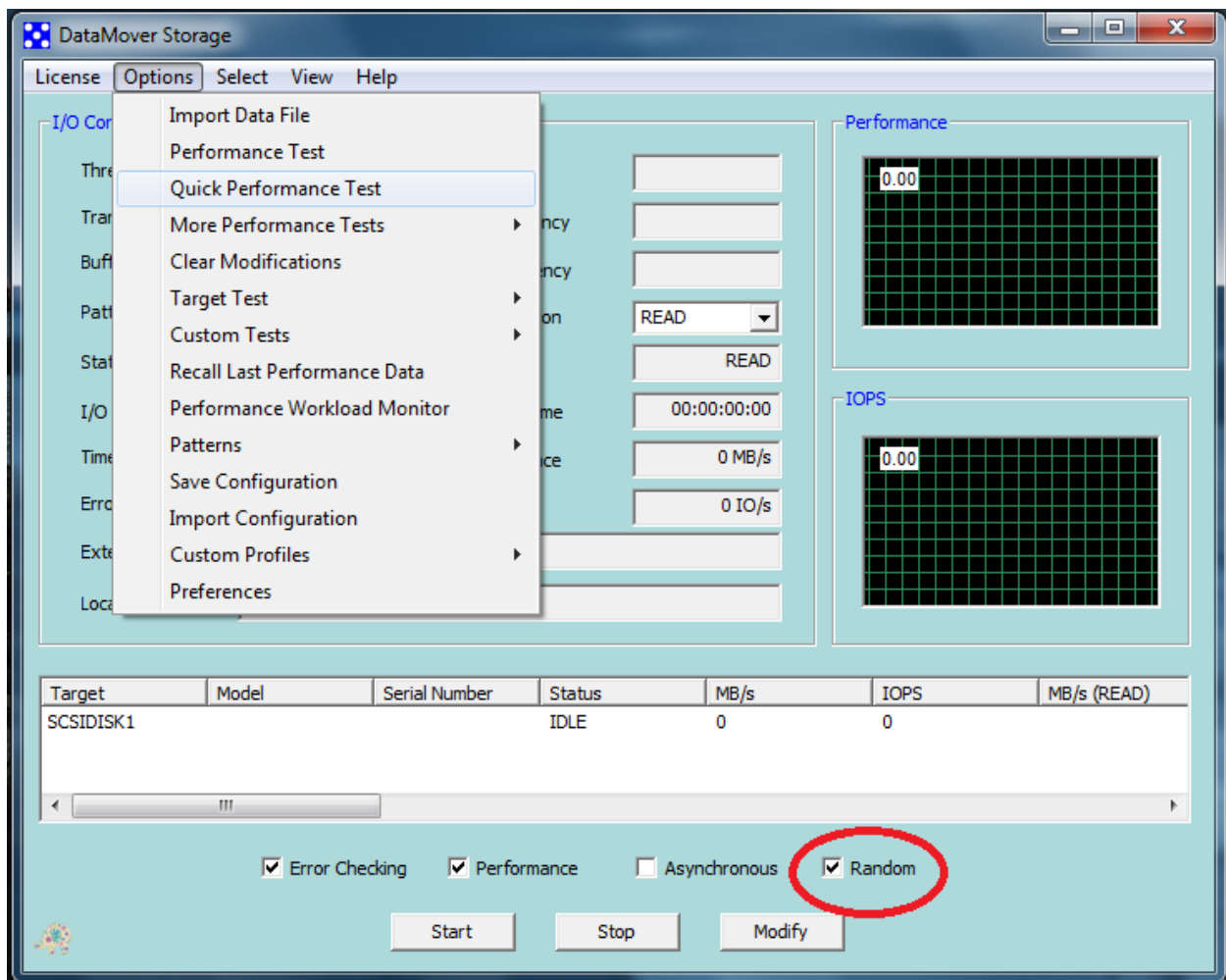


The long version of the test uses similar algorithms but test hundreds of combinations of workload parameters over many different transfer sizes. Test duration is typically > 24 hours. The long version may be started by selecting the “Options->More Performance Tests->Walking LBA Test” menu item in the AD and clicking “Start,” “PERFWLKTST” will be displayed in Operation window.

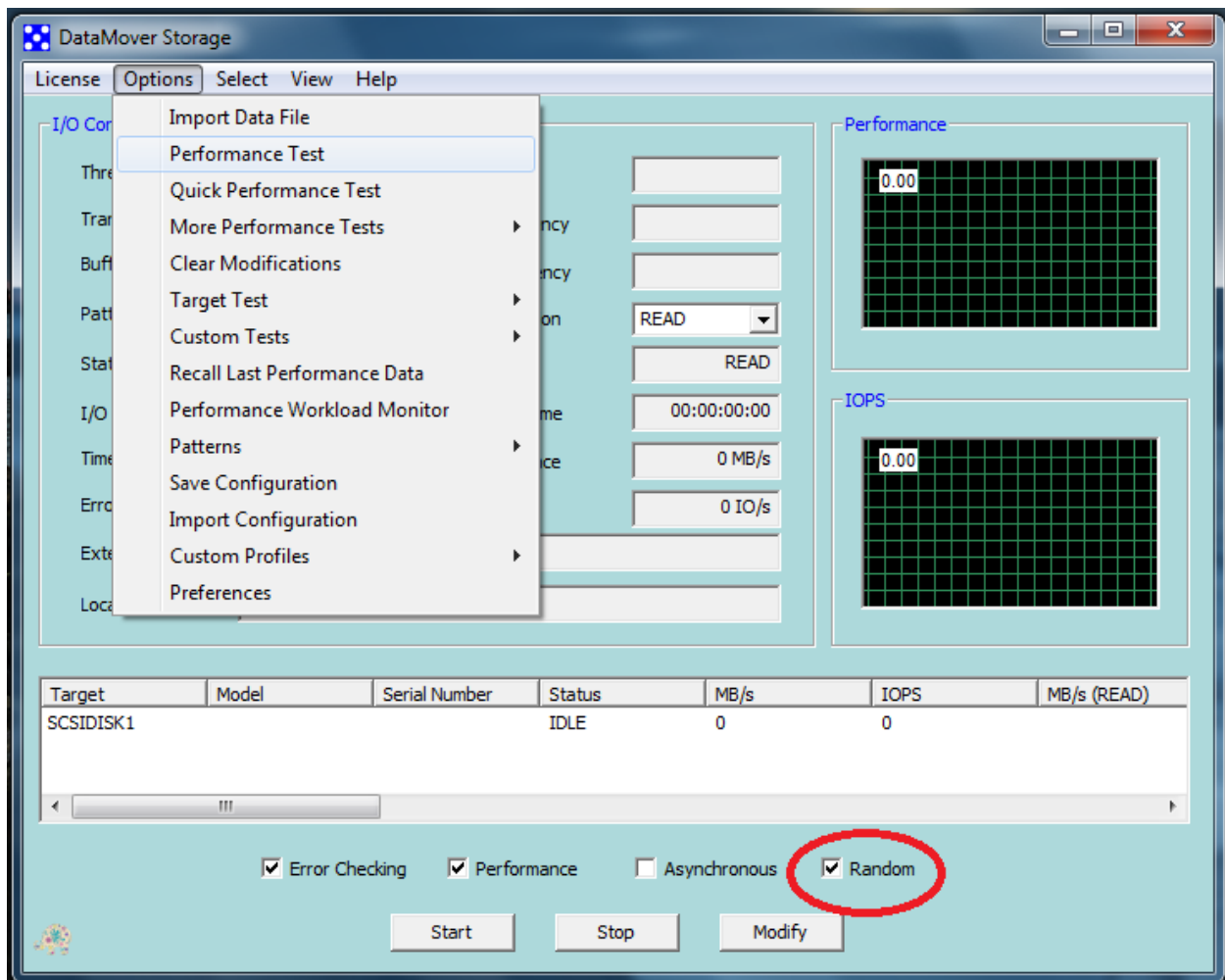


RANDOM Workload Tests

The RANDOM workload uses non-repeating, non-overlapping I/O to minimize controller and pre-fetch cache hits on the target storage device. By randomly selecting the start location for the I/O, cache effects are minimized, and block device access are maximized. The quick test collects data on 512B and 1MiB transfer sizes for all I/O directions using a limited combination of workload parameters. Test duration is typically less than 30 minutes. The small transfer size is best for maximum IOPS, and the larger transfer size is best for maximum throughput. Start this test by selecting the “Options->Quick Performance Test” menu item in the AD, check the “Random” check box and clicking “Start,” “QPERFTEST” will be displayed in Operation window.



The long version of the test uses similar algorithms but test hundreds of combinations of workload parameters over many different transfer sizes. Test duration is typically > 24 hours. The long version may be started by selecting the “Options->Performance Test” menu item in the AD, check the “Random” check box and clicking “Start,” “PERFTEST” will be displayed in Operation window.



Performance Tests on Logical and Physical Targets

There is one noticeable difference between logical and physical targets when speaking about I/O generation. For logical targets, DMS will always use one or more files per target; for physical targets, DMS does not use files and instead talks to the physical locations on the target. The physical locations are called Logical Block Addresses (LBAs) or sectors which represent the small data unit on the storage device. Even when files are used, eventually this I/O lands on an LBA on a physical target, but the location of the LBA(s) that will be used is not under DMS's control. But when DMS performs I/O to a physical target, it has complete control over what LBA(s) will be used. This is an important fact because a file may be broken up over non-contiguous LBA regions which make it impossible to guarantee that the pre-fetch cache hits will be maximized. Based on the fragmented nature of file systems, StorageWorks always suggests that physical targets be used for performance testing to guarantee the best metrics are obtained.

The CACHE, WALK and RANDOM workloads are the same for logical and physical targets.

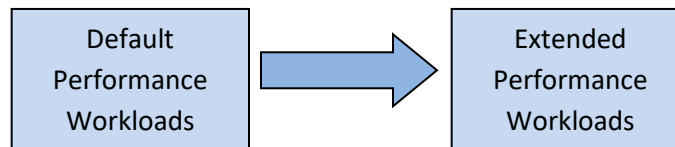
Default Performance Tests

The Default Performance Tests (DPT) executes many I/O parameter combinations to obtain the best results. The DPT may also be disabled by checking the “Disable Default Workloads” checkbox in the DPD.

Extended Performance Tests

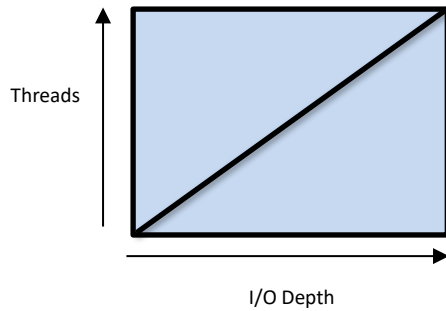
The Extended Performance Tests (EPT) allows the user to bypass or append additional workload combinations to the default performance tests. Extended performance workload tests are enabled by checking the “Enable Extended Perf” checkbox in the DPD. If default performance workloads are also enabled, the extended performance workloads will not start until the default performance workloads complete.

Performance Workload Phase Flow:



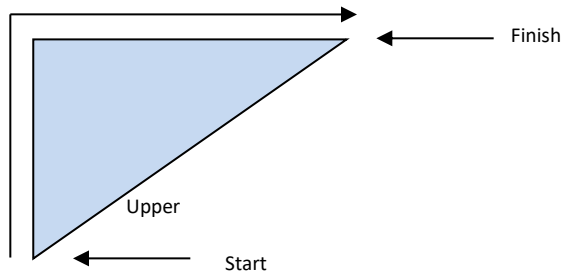
Using the DPD, extended performance workload behavior and duration will be determined by three settings: “Max Thread Count,” “Max I/O Depth” and “Increment By.” The EPT uses a four-sided shape using thread count for height and I/O depth for width. If thread count and I/O depth are the same, then it is a square, if they are different it is a rectangle. The EPT divides the object into four regions: upper triangle, lower triangle, diagonal and rectangle. Each region represents a different stage of the EPT starting with upper triangle and ending with rectangle.

Upper Triangle -> Lower Triangle -> Diagonal -> Rectangle



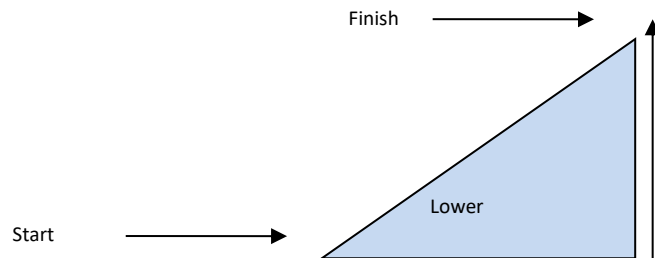
Upper triangle executed as follows:

Vertical side executed first, start equals thread count = 1, I/O depth = 1; horizontal side executed last, finish equals thread count = thread max., I/O depth = I/O depth max. Diagonal and interior combinations not tested.



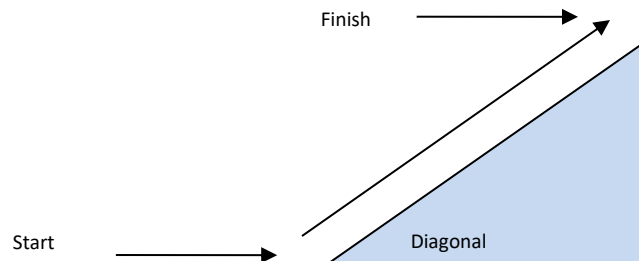
Lower triangle executed as follows:

Horizontal side executed first, start equals thread count = 1, I/O depth = 1; vertical side executed last, finish equals thread count = 1, I/O depth = I/O depth max. Diagonal and interior combinations not tested.



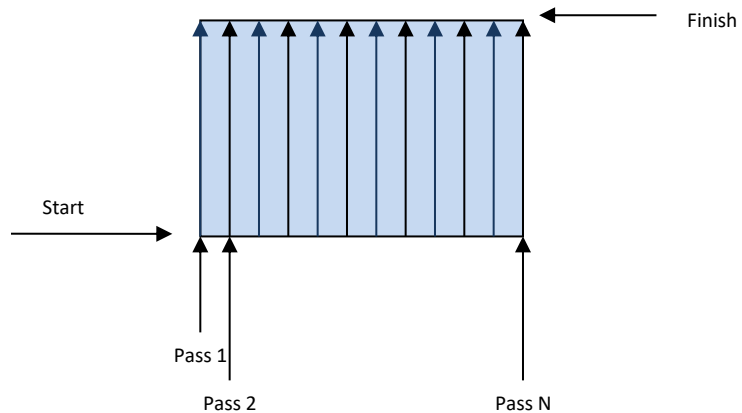
Diagonal executed as follows:

Diagonal side executed solely, start equals thread count = 1, I/O depth = 1; finish equals thread count = thread max., I/O depth = I/O depth max. The thread count and I/O depth incremented equally by "Increment By" setting.



Rectangle executed as follows:

Lower left-hand corner
executed first, start equals
thread count = 1, I/O depth =
1. On each pass, thread count
incremented while I/O depth
stays unchanged until thread
max. Then thread count reset
to 1 and I/O depth
incremented, process repeats
until finish where thread count
= thread max and I/O depth =
I/O depth max.



EPT example:

Using the DPD, thread count equals 22, I/O depth equals 16 and increment equals 5. We have a rectangle of height 22 and width 16; upper triangle will be executed first, followed by lower triangle, diagonal and lastly rectangle. The combinations of thread and I/O depth used to complete the upper triangle will be (in thread/depth and order) 1/1, 6/1, 11/1, 16/1, 21/1, 22/1, 22/6, 22/11 and 22/16. The combinations of thread and I/O depth used to complete the lower triangle will be 1/1, 1/6, 1/11, 1/16, 6/16, 11/16, 16/16, 21/16 and 22/16. The combinations of thread and I/O depth used to complete the diagonal will be 1/1, 6/6, 11/11, 16/16, 21/16 and 22/16. And lastly, the combinations of thread and I/O depth used to complete the rectangle will be 1/1, 6/1, 11/1, 16/1, 21/1, 22/1, 1/6, 6/6, 11/6, 16/6, 21/6, 22/6, 1/11, 6/11, 11/11, 16/11, 21/11, 22/11, 1/16, 6/16, 11/16, 16/16, 21/16 and 22/16.

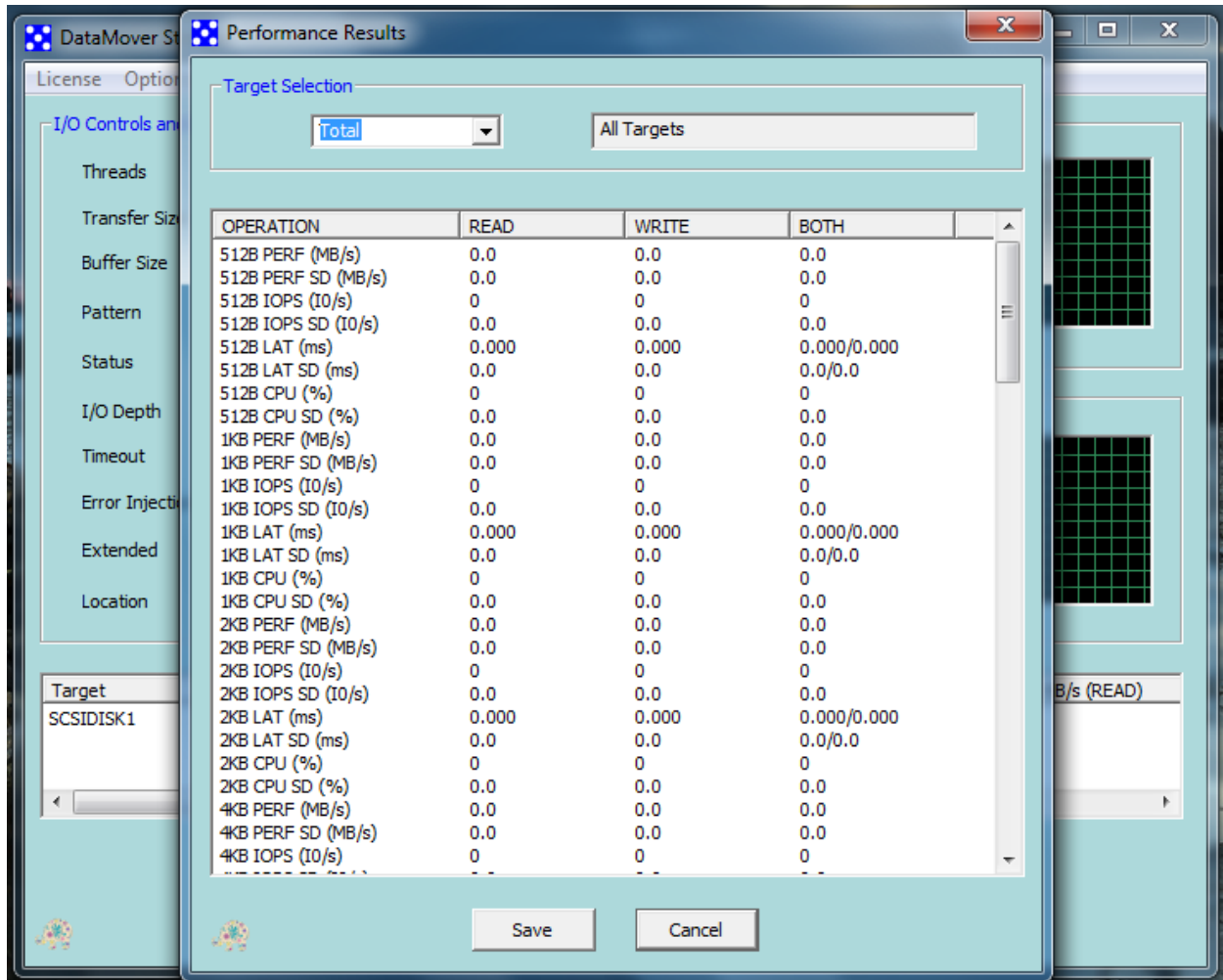
The EPT can be stopped at any time and it may not be necessary to execute all stages of the test. This determination can be made by monitoring the real-time data.

When Should EPT Be Used?

StorageWerks suggests the following guidelines when making the determination if EPT should be enabled. Manually test various transfer sizes and I/O directions, does it take a thread count > 10 or an I/O depth > 16 to achieve the best results for CACHE or RANDOM workloads? If the answer is yes, EPT should be used. Does it take > 25 threads or an I/O depth > 16 to achieve the best results for WALK workloads? If the answer is yes, EPT should be used. Make a note of thread count and I/O depth used in the manual testing and using the DPD set thread count and I/O depth to these values.

Long Test Results

When a long performance test completes the Performance Results Dialog (PRD) will be displayed.



By selecting “Save,” the results may be saved to a comma-delimited file. The results file consists of five main sections – global test settings, global performance results, global workload parameters, per target performance results and per target workload parameters. The global test settings will be first displayed, followed by global results, and then per target results. Global test setting description is as follows:

- Default Workloads – Possible values ‘ON’ or ‘OFF’
- Extended Workloads – Possible values ‘ON’ or ‘OFF’
- Streaming – Possible values ‘ON’ or ‘OFF’
- Default Maximum Thread – Possible values 1 thru 25
- Default Maximum IO Depth – Possible values 1 thru 64
- Extended Maximum Thread – Possible values 1 thru 25
- Extended Maximum IO Depth – Possible values 1 thru 64
- Ramp Time – Time in seconds
- Extended Perf Increment – Possible values 1 thru 64

- Performance Bypass – Possible values ‘ON’ or ‘OFF’
- Performance Bypass Mask – Bit mask showing the transfer sizes bypassed. If the bit position is set to “1” the transfer size is bypassed.

Bit Position	Transfer Size
0	512 B
1	1 KiB
2	2 KiB
3	4 KiB
4	8 KiB
5	16 KiB
6	32 KiB
7	64 KiB
8	128 KiB
9	256 KiB
10	512 KiB
11	1024 KiB
12	2048 KiB
13	4096 KiB
14:63	Reserved

- Performance Timer Resolution – This value indicates the number of counts per second representing the resolution of the system’s timing resolution. For example, if 50,000 is reported, the system is capable of measuring time resolution down to 1/50,000 seconds, or 20 micro-seconds.

The workload parameter section is broken down by I/O direction and performance measurement type (throughput, IOPS, latency and CPU %).

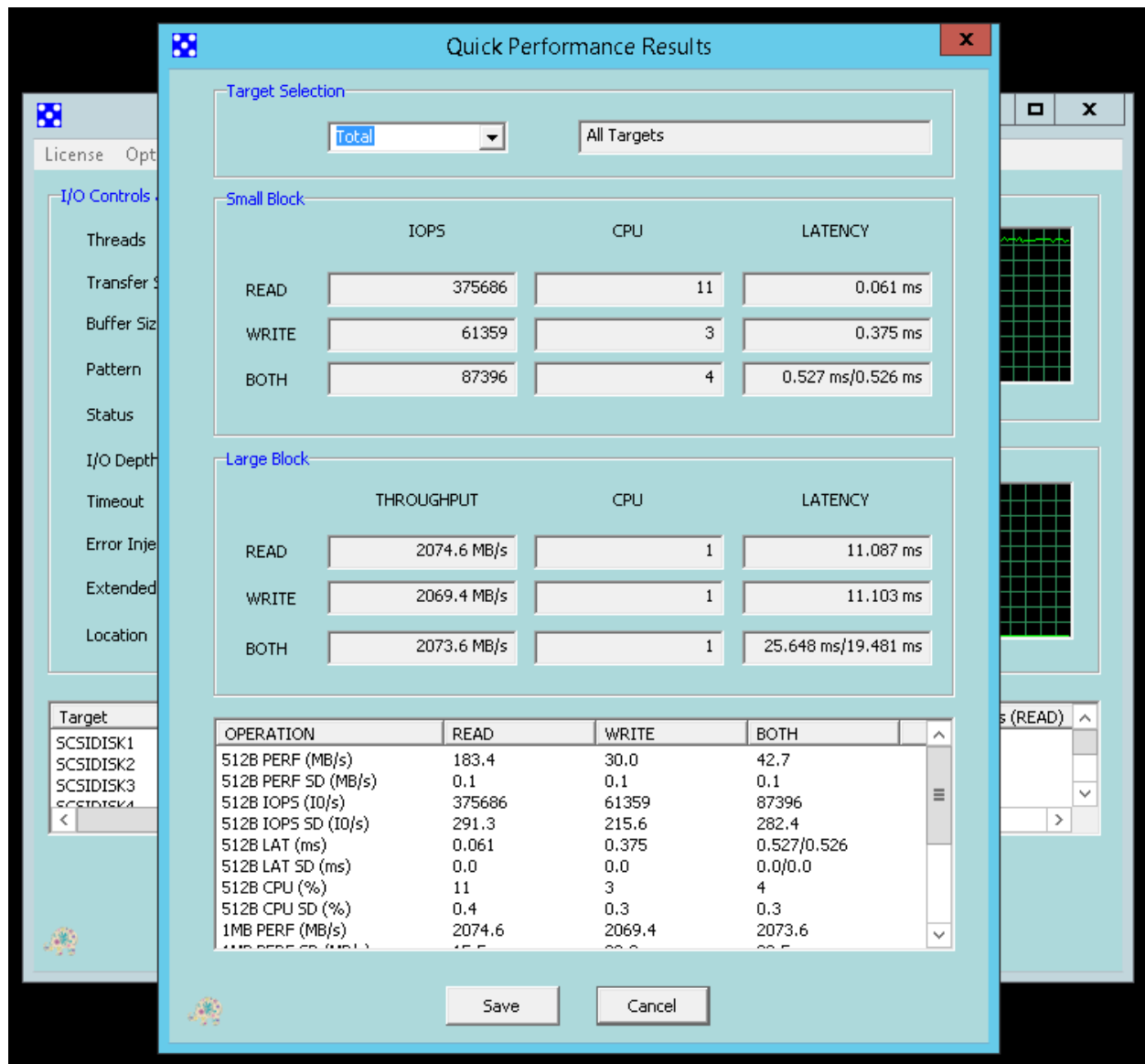
- Half Duplex Read, Write and Full Duplex TP (MB/s)
- Half Duplex Read, Write and Full Duplex IOPS

- Half Duplex Read, Write and Full Duplex LAT (ms)
- Full Duplex LATR (ms)
- Full Duplex LATW (ms)
- Half Duplex Read, Write and Full Duplex CPU (%)

For each data sample, the I/O operation (asynchronous or synchronous), thread and I/O depth setting will be saved. Possible I/O operation values are 1 and 2, 1 equal synchronous I/O and 2 equals asynchronous I/O. Possible values for thread are 1-25, and I/O depth is 1-64. These workload parameters will provide a historical record of what workload parameter combination obtained the best results. StorageWerks provides a sample excel spreadsheet in the data patterns download on www.storagewerks.net where the performance results section can be directly copied and pasted in the “data” tab automatically creating performance charts.

Quick Test Results

When a quick performance test completes the Quick Performance Results Dialog (QPRD) will be displayed.



Selecting "Save," will save the results to a text file not comma delimited. The results are not intended to be as detailed as the long performance test since this is a quick test. The QPRD consists of a Small and Large Block section representing data associated with the smallest transfer size of 512B and larger 1MiB. The small block section provides performance data for IOPS, CPU and latency; large block differs slightly by providing throughput instead of IOPS. This distinction is made because users are most likely to be interested in IOPS for the small transfer sizes.

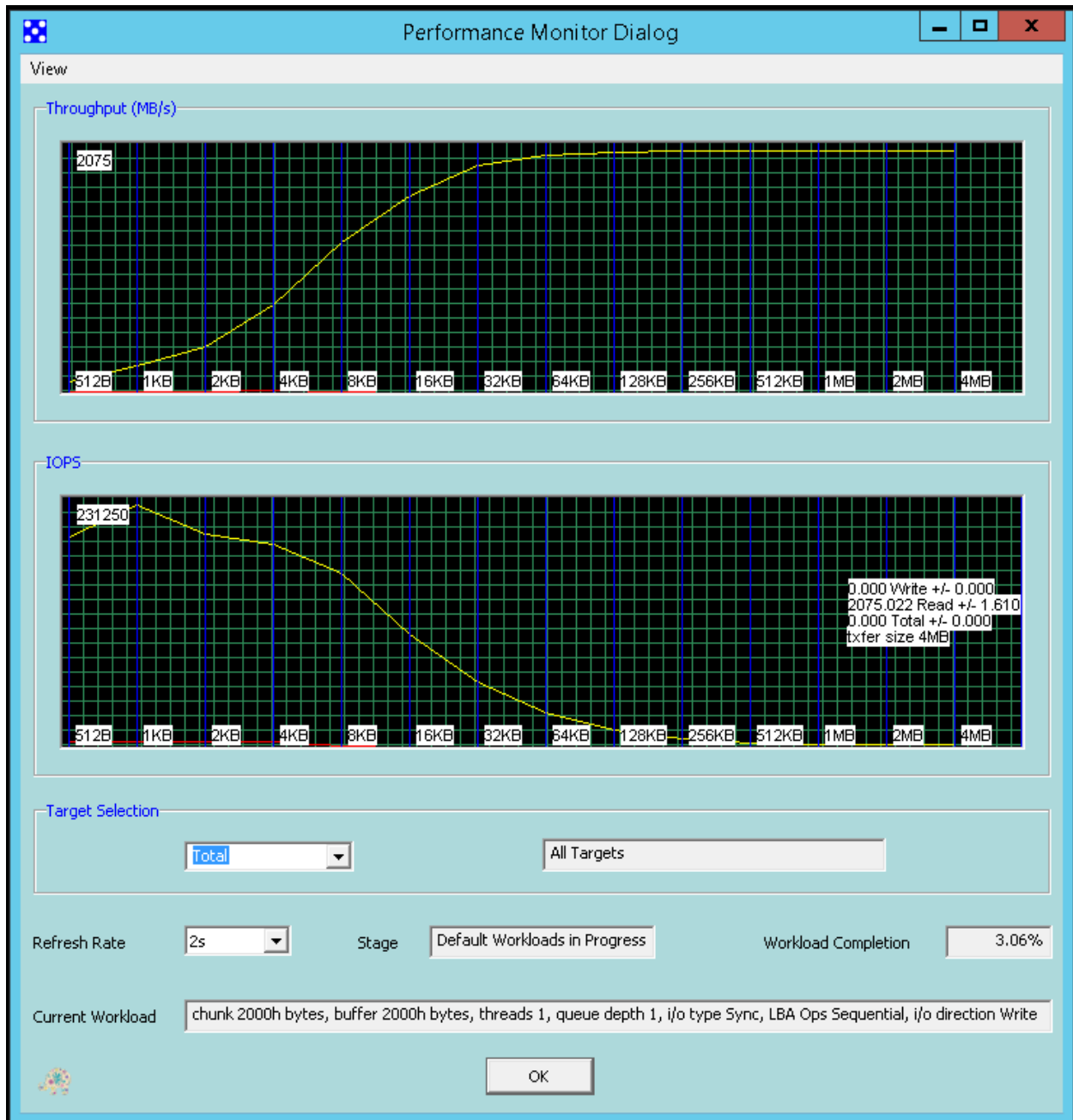
Performance Workload Monitor

DMS provides the ability to view the performance metric graphs real-time as the tests are running. The Performance Monitor Dialog (PMD) allows the user to view any combination of I/O direction and provides the following graph types:

- Standard – Throughput (in MiB/s) and IOPS displayed vs. transfer size for all I/O directions, default I/O direction is READ only.
- Per CPU Utilization – Ratio of throughput and IOPS per CPU utilization displayed vs. transfer size for all I/O directions, default I/O direction is READ only.
- Latency vs. Transfer Size – Latency (in milliseconds) displayed vs. transfer size for all I/O directions, default I/O direction is READ only.
- CPU vs. Transfer Size - CPU utilization displayed vs. transfer size for all I/O directions, default I/O direction is READ only.

To access the PMD, select the “Options->Performance Workload Monitor” menu item in the AD. The PMD data will be active only when a performance test is in progress. The PMD will not open if any threads are still in INIT phase.

Default PMD screenshot:



This PMD example shows a test in progress.

Menu Items

View Performance History Bytes Write (Red) - Enable or disable write I/O direction graph data.

View Performance History Bytes Read (Yellow) - Enable or disable read I/O direction graph data.

View Performance History Bytes Total (Green) - Enable or disable full-duplex I/O direction graph data.

View Graph Type Standard - Default selection on PMD open. Sets top graph as throughput (in MiB/s) vs. transfer size and bottom graph as IOPS vs. transfer size.

View Graph Type per CPU Utilization - Sets top graph as throughput per CPU utilization (in MiB/s/%) vs. transfer size and bottom graph as IOPS per CPU utilization (in IO/s/%) vs. transfer size.

View Graph Type Latency vs. Transfer Size - Sets top graph as throughput latency (in milliseconds) vs. transfer size and bottom graph as IOPS latency (in milliseconds) vs. transfer size. Latency graphs will be the same for throughput and IOPS.

View Graph Type CPU vs. Transfer Size - Sets top graph as throughput CPU utilization vs. transfer size and bottom graph as IOPS CPU utilization vs. transfer size. CPU utilization graphs will be the same for throughput and IOPS.

GUI Items

Top Chart – The type of data displayed is determined by graph type selection setting. Default graph type is throughput. Horizontal axis displays transfer sizes ranging from 512B to 4MiB indicated by blue vertical lines. Upper left-hand corner displays the maximum value for all data points to provide scale.

Bottom Chart - The type of data displayed is determined by graph type selection setting. Default graph type is IOPS. Horizontal axis displays transfer sizes ranging from 512B to 4MiB indicated by blue vertical lines. Upper left-hand corner displays the maximum value for all data points to provide scale.

Target Selection – This dropdown allows the user to select the data source for the top and bottom charts which can be an individual target, or all targets combined.

Refresh Rate – A dropdown window that allows the user to select the refresh rate of the top and bottom graphs. Possible values are (in seconds) 2s, 5s, 10s, 20s, 30s, 60s, 300s and disable. Default selection is 2s.

Stage – Displays what phase the performance test is executing. Possible descriptions are:

- “Default Workloads in Progress” – Indicates the default automated performance workload iterations are being processed.
- “Upper Triangle in Progress” – Indicates the extended automated performance workload iterations called upper triangle are being processed.
- “Bottom Triangle in Progress” - Indicates the extended automated performance workload iterations called bottom triangle are being processed.
- “Diagonal in Progress” - Indicates the extended automated performance workload iterations called diagonal are being processed.
- “Rectangle in Progress” - Indicates the extended automated performance workload iterations called rectangle are being processed.

Workload Completion – Percentage of all workload iterations completed including default and extended performance test.

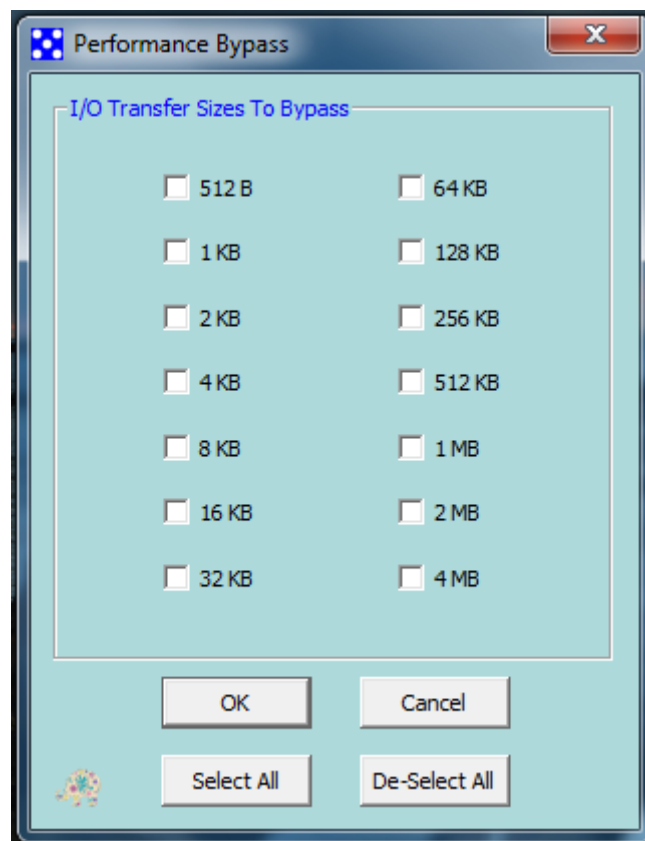
Current Workload – Displays the current workload parameters being used (transfer and buffer size, thread count and queue depth), I/O type (synchronous or asynchronous), LBA operation type (sequential or random) and I/O direction.

OK – Selecting this button closes the PMD.

Performance Bypass Dialog

By default, the DMS performance tests characterize fourteen transfer sizes. If the user does not wish to characterize all transfer sizes, any transfer size used by the performance test may be bypassed using the Performance Bypass Dialog (PBD). The PBD is accessed from the Preferences Dialog menu. At least one transfer size must remain un-checked (not bypassed) to allow the performance bypass feature to be enabled. By removing unwanted transfer sizes, the performance test times will be shorter.

Default PBD Screenshot:



GUI Items

512 B Checkbox – If checked, transfer size will be bypassed

1 KB Checkbox – If checked, transfer size will be bypassed

2 KB Checkbox – If checked, transfer size will be bypassed

4 KB Checkbox – If checked, transfer size will be bypassed

8 KB Checkbox – If checked, transfer size will be bypassed

16 KB Checkbox – If checked, transfer size will be bypassed

32 KB Checkbox – If checked, transfer size will be bypassed

64 KB Checkbox – If checked, transfer size will be bypassed

128 KB Checkbox – If checked, transfer size will be bypassed

256 KB Checkbox – If checked, transfer size will be bypassed

512 KB Checkbox – If checked, transfer size will be bypassed

1 MB Checkbox – If checked, transfer size will be bypassed

2 MB Checkbox – If checked, transfer size will be bypassed

4 MB Checkbox – If checked, transfer size will be bypassed

OK – Saves the selections and exits the dialog

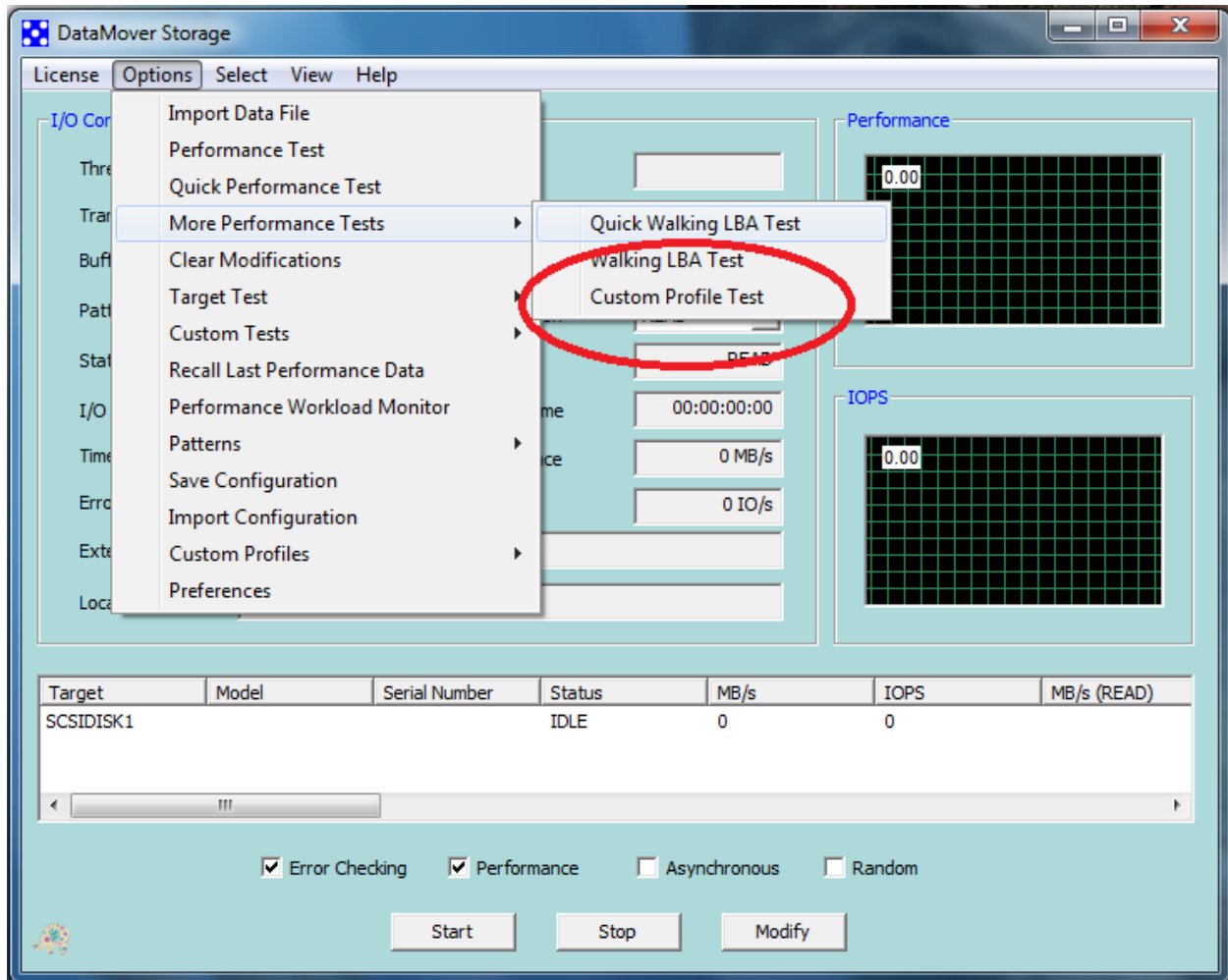
Select All – Checks all transfer size check boxes

Cancel – Does not save the selections and exits the dialog

De-Select All – Un-checks all transfer size check boxes

Custom Performance Tests

In addition to the standard performance workload type categories – CACHE, WALK and RANDOM, DMS also supports CUSTOM PROFILE performance workload analysis and testing. A prerequisite for this test is a minimum of one custom profile mapped to every target. If a target does not have a custom profile mapped, the test cannot be selected. To select this performance test, the user must navigate to the Options->More Performance Tests->Custom Profile Test, 'CUSTPERF' will be displayed Operation window:



Theory of Operation

All custom profiles will be run in single thread mode regardless of settings chosen by the user. And any linking created during the custom profile target map operation will be disabled to allow the single profile to be characterized. The custom profile will be characterized with increasing IODepth using the following thread-IODepth succession: 1-1, 1-2, 1-4, 1-8, 1-16, 1-32 and 1-64. The number of data samples captured will be dependent on the Custom Timer setting in the DPD, as the Custom Timer increases, more data samples for the given thread-IODepth combination will be captured and analyzed.

Results

Like other DMS tests, results will be saved to a comma delimited file with a breakdown of each custom profile's performance data. Performance data includes overall throughput, IOPS and latency, throughput, IOPS and latency for READ component and throughput, IOPS and latency for WRITE component for every thread-IODepth combination. The maximum and minimum results for each of these subcategories will be calculated and appended to the section results.

Bus Trigger Operation

When a hard error is detected (data mis compare, I/O timeout, application interface error, SCSI error) DMS will write one of these signatures to the target.

ASCII	HEX
"MOOJIT GOT CRC ERROR ON LAST I/O "	4D4F4F4A 49542047 4F542043 52432045 52524F52 204F4E20 4C415354 20492F4F 20202020 20202020
"MOOJIT GOT API ERROR ON LAST I/O "	4D4F4F4A 49542047 4F542041 50492045 52524F52 204F4E20 4C415354 20492F4F 20202020 20202020
"MOOJIT GOT TIMEOUT ERROR ON LAST I/O "	4D4F4F4A 49542047 4F542054 494D454F 55542045 52524F52 204F4E20 4C415354 20492F4F 20202020

Since this is a write operation, the signature may be embedded with the first write command (e.g., iSCSI if immediate data is supported), otherwise, it will be included in the first data-out frame associated with the command. The data-out behavior is the most prevalent and it will add unwanted time between the original error event and the trigger. To try and remove this unwanted latency, DMS will send two bus triggers to physical targets that have been classified as SCSI (SAS, Fibre Channel, iSCSI) or SATA – signature (1) followed by signature (2). Signature (2) are referenced in the table above, signature (1) will vary depending on the technology. This double bus signature only applies to physical targets, logical targets will only use signature (2) (remember that logical targets are filesystems or SMB mount points). Signature (1) will vary as follows:

	SCSI	SATA
Signature (1a)	Any CDB length with CDB[0] = 0xEFh	N/A
Signature (1b)	N/A	CHECK POWER MODE or SANITIZE STATUS EXT

The sequence of bus events for SCSI will be signature (1a) -> signature (2), and for SATA will be signature (1b) -> signature (2). The advantage of signature (1) it does not require a data-out phase and eliminates the time between command and data phase. NVMe devices will use signature (2) only. This is due the small latency between the command fetch and first PCIe TLP with the signature. In NVMe, most of the latency occurs between the command doorbell and subsequent command fetch operation, there is

nothing that DMS can do to address this since the doorbell operation is handled by the NVMe miniport driver.

In most cases, the first DWORD (4D4F4F4A) should be sufficient to trigger the analyzer.

Windows Event Reporting

Windows events relating to DMS are posted to the Windows application event log. The AD is responsible for most of the event generation; the BD only posts a few. DMS posts events related to start and stop of I/O, functional and performance testing. DMS application events provide time correlation with other system events to facilitate troubleshooting. Most of the events are self-explanatory but a few will be described in more detail here.

Physical Target Data Corruption Event

When a data corruption is detected by DMS, information will be posted in the General and Details tab of the application event.

General Tab Data

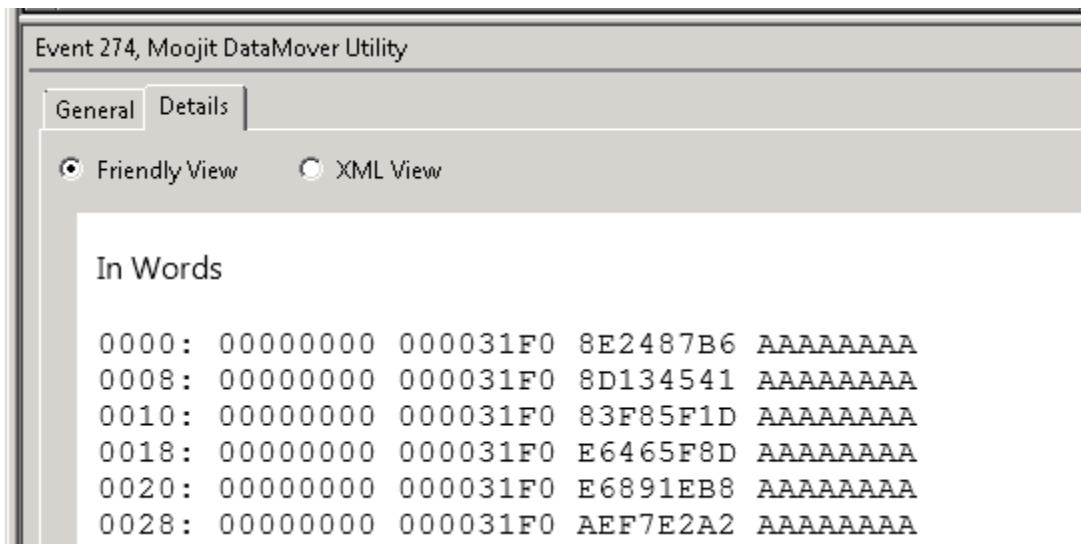
Using the example below, five pieces of information relevant to the target and error will be included in the message. Reading from left to right, they are target name, SCSI vendor and model name, target SCSI bus-target-lun designator, thread which detected the error, and number of DWORDs corrupted.

Data mis compare on Target SCSIDISK5: MOOJIT MJT5200 , B-T-L = 0-0-4, Thread 25, 2047 DWORD error(s) detected. Each mis compare sample consists of four DWORDs - Reading left to right, the first two DWORDs represent the 64bit LBA where the data mis compare was detected, the last two DWORDs are the expected and actual value - LBA0 (MSB), LBA0 (LSB), EXPECTED0, ACTUAL0 , ... , LBA9 (MSB), LBA9 (LSB), EXPECTED9, ACTUAL9, etc.

In this example, thread #25 reported 2047 DWORDs (or 8188 bytes) of data corrupted on the “SCSIDISK5” target “MOOJIT MJT5200” located at B-T-L 0-0-4.

Details Tab Data

Using the example below, for each DWORD corruption, the storage location on the target where corruption occurred (in 64-bit LBA format), expected and actual data values (in HEX) are provided.



Looking at the first entry, the expected DWORD value of 0x8E2487B6 returned an actual value of 0xAAAAAAAA from LBA 0x0000000000000031F0.

Logical Target Data Corruption Event

When a data corruption is detected by DMS, information will be posted in the General and Details tab of the application event.

General Tab Data

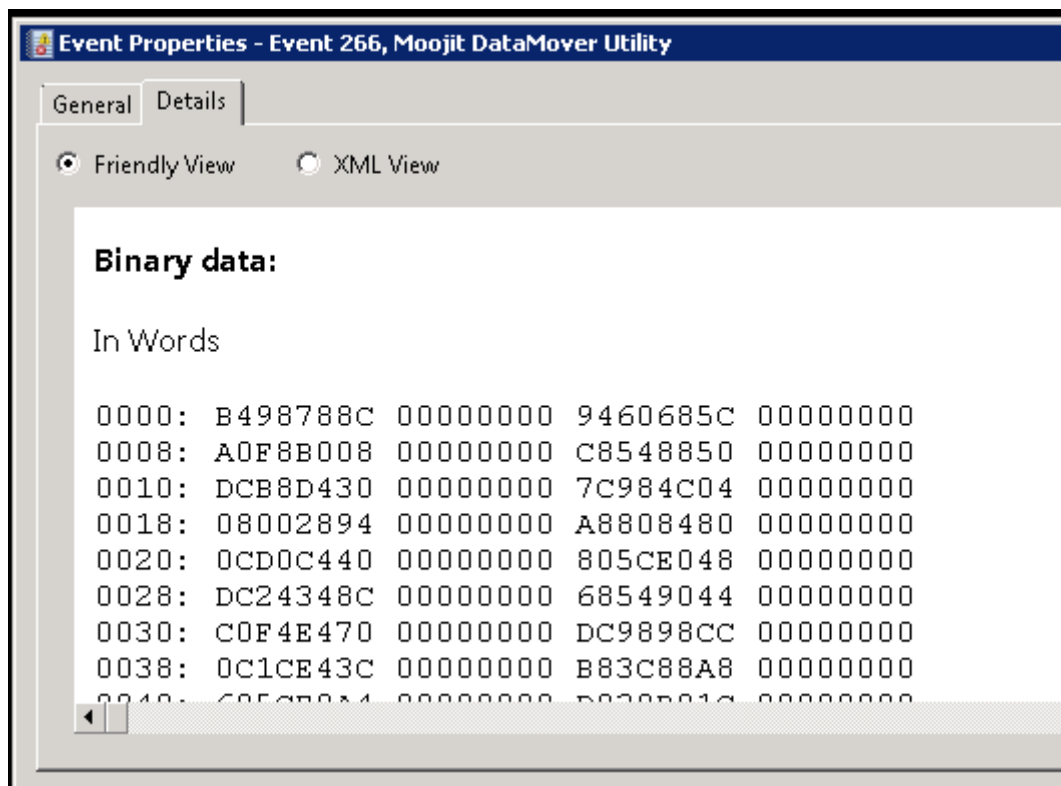
Using the example below, four pieces of information relevant to the target and error will be included in the message. Reading from left to right, they are filename name, target name, thread which detected the error, and number of DWORDs corrupted.

Data mis compare on file JAMIE3.DAT, Target C:\, thread 5, 83 DWORD error(s) detected. The first DWORD (reading left to right) is the expected value, followed by the actual value - EXPECTED0, ACTUAL0, ... , EXPECTED9, ACTUAL9, etc. Select the 'WORD' radio button to view data in DWORD format.

In this example, thread #5 reported 83 DWORDs (or 332 bytes) of data corrupted in file "JAMIE3.DAT", located on target "C:\".

Details Tab Data

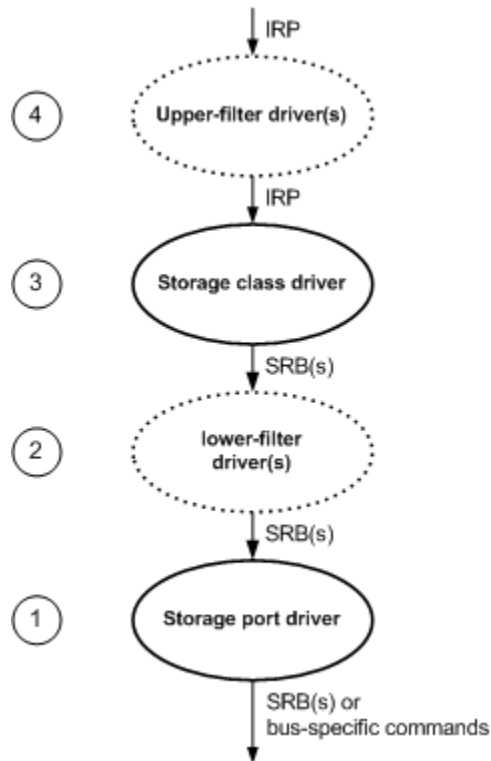
Using the example below, for each DWORD corruption, the expected and actual data values (in HEX) are provided.



Looking at the first entry, the expected DWORD value of 0xB498788C returned an actual value of 0x00000000.

Windows Event ID 153 Explained

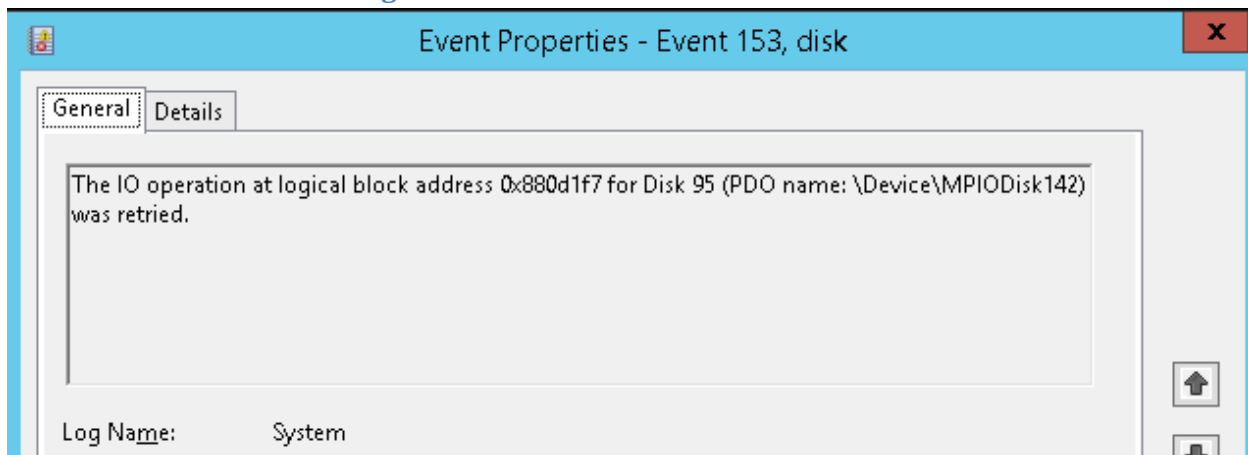
Event 153 is posted by the disk.sys class driver in response to an error reported by the STorport miniport driver. Recall that the windows storage driver architecture consists of multiple drivers in stacked arrangement.



The I/O Request Packet (IRP) is delivered to the class driver by the filesystem kernel component. The class driver converts the IRP into an SCSI Request Block (SRB) which gets passed down the stack through the port driver to eventually the miniport driver which is not shown in this graphic and sits below the port driver. In general terms, the miniport converts the SRB into a bus specific command such as SAS, Fibre Channel or iSCSI, and places the command on the hardware bus adapter (HBA) request queue for processing. When the command completes, the miniport will update several fields in the SRB indicating the completion status and will notify the port driver that the SRB has been processed. Depending on SRB status, the port driver may or may not act, for event ID 153, the port driver does not, and passes the SRB up to the class driver for further processing.

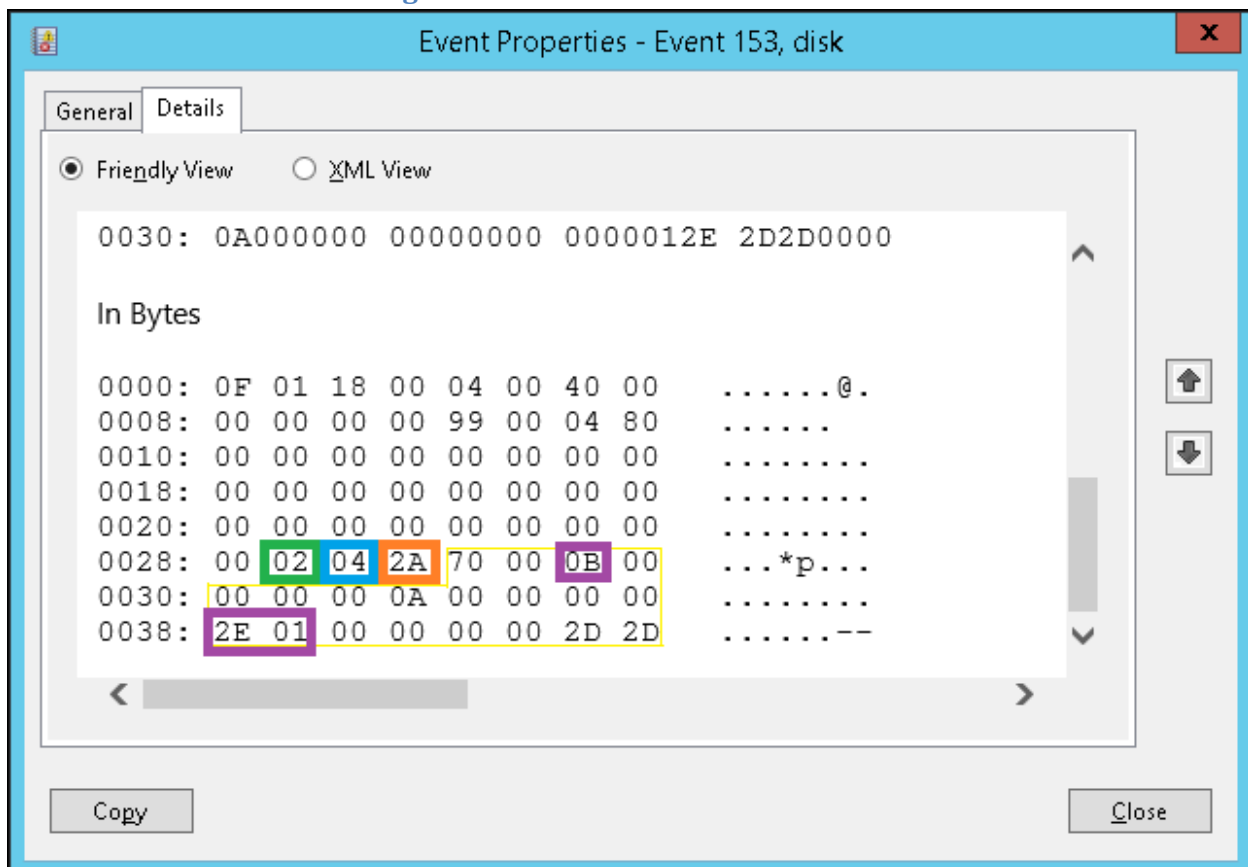
For SRB status not equal to `SRB_STATUS_SUCCESS`, the class driver may perform error recovery by retrying the SRB or it may complete the IRP. For this example, the class driver performed an I/O retry.

General Event ID 153 Message



The general event data tell us the disk and LBA of the I/O that was retried.

Details of Event ID 153 Message



Data bytes starting at offset 0x29h provide the most useful information on what caused the error.

- GREEN = SCSI STATUS field
- BLUE = SRB STATUS field
- ORANGE = SCSI COMMAND OPERATION CODE

- YELLOW = SENSE DATA returned by SCSI device
- PURPLE = SENSE KEY, ASC, ASCQ field

SCSI STATUS

SCSI status returned by device, see T10 SAM-4 for definition of all status responses. In this example, the disk device returned 0x02h (CHECK CONDITION).

SRB STATUS

As the name implies, the status of the SRB, below are the SRB Status return values as defined by Microsoft:

• SRB_STATUS_PENDING	0x00
• SRB_STATUS_SUCCESS	0x01
• SRB_STATUS_ABORTED	0x02
• SRB_STATUS_ABORT_FAILED	0x03
• SRB_STATUS_ERROR	0x04
• SRB_STATUS_BUSY	0x05
• SRB_STATUS_INVALID_REQUEST	0x06
• SRB_STATUS_INVALID_PATH_ID	0x07
• SRB_STATUS_NO_DEVICE	0x08
• SRB_STATUS_TIMEOUT	0x09
• SRB_STATUS_SELECTION_TIMEOUT	0x0A
• SRB_STATUS_COMMAND_TIMEOUT	0x0B
• SRB_STATUS_MESSAGE_REJECTED	0x0D
• SRB_STATUS_BUS_RESET	0x0E
• SRB_STATUS_PARITY_ERROR	0x0F
• SRB_STATUS_REQUEST_SENSE_FAILED	0x10
• SRB_STATUS_NO_HBA	0x11
• SRB_STATUS_DATA_OVERRUN	0x12
• SRB_STATUS_UNEXPECTED_BUS_FREE	0x13
• SRB_STATUS_PHASE_SEQUENCE_FAILURE	0x14
• SRB_STATUS_BAD_SRB_BLOCK_LENGTH	0x15
• SRB_STATUS_REQUEST_FLUSHED	0x16
• SRB_STATUS_INVALID_LUN	0x20
• SRB_STATUS_INVALID_TARGET_ID	0x21
• SRB_STATUS_BAD_FUNCTION	0x22
• SRB_STATUS_ERROR_RECOVERY	0x23
• SRB_STATUS_NOT_POWERED	0x24
• SRB_STATUS_LINK_DOWN	0x25

In this example, the SRB status is 0x04h (SRB_STATUS_ERROR).

SCSI COMMAND OPERATION CODE

There are total of 256 possible SCSI operation codes defined by the T10 standards. Please refer to SPC and SBC T10 standards for definitions. Most likely, the COMMAND OPERATION CODE returned will be one associated with a SCSI READ or WRITE command. In this example, the command is 0x2Ah (WRITE (10)).

SENSE DATA

The first byte at offset 0x2C defines the format of the sense data. This byte will typically be 0x70 (or 0xF0), but it could also be 0x71 (or 0xF1), 0x72, or 0x73. Please refer to the T10 SPC standard for more details. The first byte is important because it will define the structure and meaning of the sense data fields. In this example, 0x70h tells us this fixed format sense data.

Bit Byte	7	6	5	4	3	2	1	0
0	VALID	RESPONSE CODE (70h or 71h)						
1	Obsolete							
2	FILEMARK	EOM	ILI	Reserved	SENSE KEY			
3	(MSB)							
...	INFORMATION							
6	(LSB)							
7	ADDITIONAL SENSE LENGTH (n-7)							
8	(MSB)							
...	COMMAND-SPECIFIC INFORMATION							
11	(LSB)							
12	ADDITIONAL SENSE CODE							
13	ADDITIONAL SENSE CODE QUALIFIER							
14	FIELD REPLACEABLE UNIT CODE							
15	SKSV							
16	SENSE KEY SPECIFIC							
17								
18								
...	Additional sense bytes							
n								

SENSE KEY, ASC, ASCQ

The format of the sense data tells us that the sense key is in byte offset 2, the additional sense code (ASC) at byte offset 12, and additional sense code qualifier (ASCQ) at byte offset 13. When applied to Event ID 153 format, these fields are located at byte offsets 0x2Eh, 0x38h and 0x39h, respectively. These fields are defined in T10 SPC. In this example, we have the following:

- SENSE KEY = 0xBh (ABORTED COMMAND)
- ASC/ASCQ = 2E/01 (COMMAND TIMEOUT BEFORE PROCESSING)

Final Analysis

The disk timed out processing of a command and returned a check condition, indicating the command had been aborted due to an internal timeout. The class driver, when it received this SRB status indication, decided to perform an I/O retry of the same command. This error and subsequent retry operation would not be communicated to the application layer.

Host Memory Considerations

DMS's memory requirements are dependent on the settings chosen by the user. Asynchronous I/O is the worst-case configuration for maximum memory usage. Let us look at an example. The buffer size, I/O

depth and thread count dictate the amount of memory that will be required. If the user selects READ and WRITE operation (BOTH), 25 threads, buffer size equal to 4MiB and I/O depth equal to 64, the memory requirements per target will be as follows: $50 * 4\text{MiB} * 64 = 12.5\text{GiB}$. Based on this calculation, it is easy to see that the user must have a good understanding of the memory configuration in the system under test. Avoid using virtual memory that is in the system's page file, this will have a negative impact on performance. Please refer to the following Microsoft Knowledge Base Article that describes large memory support in Windows [Q283037](#) and how to enable it.

Custom Data Patterns

Custom data pattern files can be created using any hex-binary editor. Each file must begin with a 16-byte header that consists of a NULL terminated variable length ASCII string up to 16 bytes (this includes the NULL value). Data files created by StorageWerks can be downloaded from www.storagewerks.net, please use these as examples when creating your own.

Practical Knowledge

How are IOPS measured?

Most of my users think of this as a simple concept, but it can get downright complicated. The reason why is because it is difficult for a host-based application such as DMS to determine the true number of I/O exchanges required to complete a given transaction. Fundamentally, the number of I/O's required to complete a transaction will be dictated by the size of the transfer and the adapter's maximum transfer size, which is ultimately determined by the adapter's Scatter-Gather (SG) capabilities. SG is a value which represents the maximum number of non-contiguous blocks of system memory that the adapter can access to complete a single I/O transfer. Here are some typical maximum transfer sizes for well-known technologies:

Host Solution	Maximum Transfer Size
Microsoft Software iSCSI Initiator	256KiB
Linux Software iSCSI Initiator	32KiB
Fibre Channel HBA	1MiB
SAS HBA	512KiB
iSCSI HBA	256KiB

The Microsoft class driver will use the maximum transfer size advertised by the adapter to chop an I/O exchange into more than one chunk if it exceeds this value. For example, let us say we want to perform a 2 MiB transfer at the application level using an adapter with a 32KiB maximum transfer size. To complete the transaction, 64 32KiB hardware I/O transactions will be required. The hardware transactions will be transparent to the application, and from the application's perspective, only one I/O transaction (or one IOP) has taken place. This example shows how an application measured IOPS can be misleading for transfer sizes that exceed the maximum transfer.

Why is Standard Deviation Important?

When DMS runs its performance tests, it calculates standard deviation for every calculated mean. When reporting a mean value, standard deviation should always be provided; it tells us the variation of the data samples used to calculate the mean. And the range of values that should be considered centered about the mean value.

4K Native (4Kn) Drives

DMS supports 4KiB block sizes. While running on either target type (logical or physical), if a transfer size is selected that is less than 4KiB, an error message will be displayed in the extended window and each thread will exit with same error. If performing one of the automated tests such as Random I/O custom test, DMS will automatically adjust for 4Kn by choosing transfer sizes greater than or equal to 4KiB. The performance tests will not automatically adjust however but continue as normal, eventually selecting a transfer size that works. If you do not wish to see these errors during performance runs, please enable performance bypass mode and select transfer sizes less than 4KiB to be bypassed.

512e Advanced Format Drives

Since these drives emulate the traditional 512B block size, DMS will operate as normal.

Global Configuration and Preferences Save and Import

When a configuration is saved, the current settings in the AD and DPD will be saved to file. When a configuration is imported, the current settings in the AD and DPD will be updated to reflect the information stored in the file. Having the ability to save and import configurations, allows users to easily share settings. Configuration file naming convention used by DMS is described below:

Example 1: `_LOGICAL_DMS_CFG_08172013_011422.moj`

Example 2: `_PHYSICAL_DMS_CFG_08212013_233801.moj`

The filename consists of three sections beginning with target type (logical or physical), DMS_CFG tag and date and time. All configuration files will use ".moj" extension. Example 1 shows a configuration file for a logical target saved on 8/17/2013 @ 1:14.22 A.M.; Example 2 shows a configuration file for a physical target saved on 8/21/2013 @ 11:38.01 P.M. To provide context, we recommend that the user pre-append before target type a description that represents the settings saved. For example, if Example

1 is a configuration for a random target test, a new description could be
RND_TARGET_TEST_LOGICAL_DMS_CFG_08172013_011422.moj.

Automation

When creating a configuration file to be used in automation, remember to pre-select the targets before saving. If this is not done, the configuration file will fail validation and the test will not start.

Multiple Initiator Cluster Emulation Testing

DMS supports both types of reserve and release – the traditional SPC-2 and the newer SPC-3 persistent reserve and release methods. In multi-initiator environments reserve and release will be used to share common storage resources. Windows cluster, Linux cluster and Oracle RAC are examples of solutions that utilize this resource sharing method. The DMS standalone and random cluster target tests provide this functionality by exercising the storage systems ability to handle multiple initiator resource ownership requests simultaneously. Data integrity checks will always be performed during these tests.

The DPD allows modification to the reserve/release (R/R) configuration. By default, SPC-2 style R/R is enabled, and persistent reserve/release (PR/R) can be enabled in the DPD.

Persistent Reserve and Release

Please refer to SPC-3 for a detailed explanation of PR/R; this section explains the methods supported by DMS regarding registration method and PR/R types. DMS supports Register Key and Register and Ignore Existing Key methods, the default setting are Register and Ignore Existing Key. DMS supports all the various PR/R types supported:

- Write Exclusive
- Exclusive Access
- Write Exclusive – Registrants Only (** default DMS setting **)
- Exclusive Access – Registrants Only
- Write Exclusive – All Registrants
- Exclusive Access – All Registrants

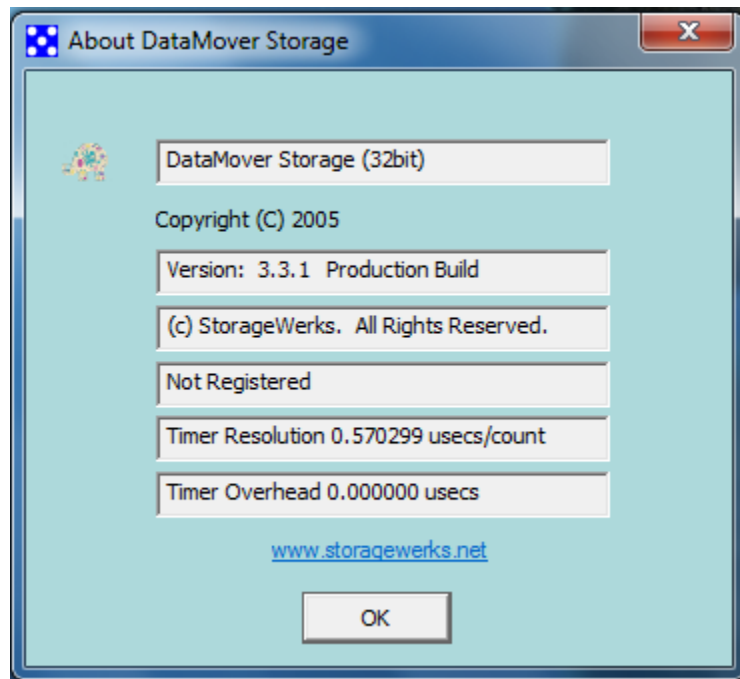
Please see the DPD section for details on DMS PR/R setting adjustments and functionality. PR/R should always be the first choice in multipath environments since it is designed to handle multiple I-T-L nexus combinations between host and target. SPC-2 style R/R may be used in some multipath environments if the vendor DSM driver supports R/R to PR/R translation, but this is rare. If R/R must be used in multipath situation, configure the load balancing policy to be “failover over only” for each target. This will ensure that only one I-T-L nexus is active per target.

High Performance Resolution Timer

DMS uses the system’s native high performance resolution timer if it is enabled. Timing resolution may vary from system to system depending on CPU architecture. If disabled, DMS will default to a 1 millisecond timing resolution. Better timing granularity is important because DMS reports latency values to hundredths of milliseconds or two decimal places. With the introduction of flash storage technology, latencies in this range may be possible. The value of the high-performance timer is reported

in the long performance test results. A value of zero, indicates that this feature is disabled, a value greater than zero indicates the number of counts per second.

Information about the high-performance timer is also displayed in the BD and AD About Dialog shown below:



The Timer Resolution field shows the granularity of the timer in micro-seconds per CPU clock cycle. This value will vary system-to-system; in this example, the system is capable of measuring 70 nanoseconds per CPU clock. The Timer Overhead field shows the time required to perform the system API call to retrieve this information. This value is considered when measuring latency.

Automation Support and Scripting

DMS automation is easy-to-use and does not require the memorization of complicated command line switches. As an alternative, DMS uses a configuration file with the following command line syntax:

`"datamover.exe foo.moj"`

Where "foo.moj" is a configuration file created and saved from the AD.

The creation and use of configuration files are covered in section Global Configuration and Preferences Save and Import, however, it is important to note that when saving configuration files for use in automation the logical or physical targets must be pre-selected. These targets will be used by DMS in automation mode, if no targets are included in the configuration file, DMS will not run and stop at the display of the BD. If the configuration file passes validation, the AD will display with "- Automation Mode" appended to the title bar and the tests will begin automatically.

DataMover Storage - Automation Mode

License Options Select View Help

I/O Controls and Information

Threads: 5

Transfer Size: 1MB

Buffer Size: 1MB

Pattern: AA

Status: RUN

I/O Depth: 8

Timeout: 60

Error Injection: 0

Extended: NONE

Location: 1 Target(s): SCSI DISK3

Latency: 5.989 ms

Read Latency: 5.989 ms

Write Latency: 0.000 ms

I/O Direction: READ

Operation: READ

Elapsed Time: 0:00:00:45

Performance: 396 MB/s

IOPS: 396 IO/s

Performance

IOPS

Error Checking	I/O Depth	Threads	Direction	Latency	Latency (READ)	Latency (WRITE)
ENABLED	1	5	READ	5.989 ms	5.989 ms	0.000 ms

☒ Error Checking
 ☒ Performance
 ☐ Asynchronous
 ☐ Random

Start Stop Modify

If the STOP button is pressed while in automation mode, the AD and BD will exit automatically. When creating configuration files for use in automation, it is highly recommended that the Stop Timer be enabled for non-performance tests in the Preference Dialog - this allows a graceful shutdown of the AD in preparation for the next automation cycle when multiple configuration files are used. Performance tests ignore the Stop Timer and will exit when the complete performance cycle finishes. StorageWerks has provided a free visual basic script that makes automation easy (script provided in DataMover Tools download). To use the script, follows these easy steps:

1. Create automation configuration file(s)
2. Create directory on local hard drive
3. Copy configuration files made in step (1) to directory created in step (2)
4. Copy datamover.vbs file to directory created in step (2)
5. Open command console and navigate to directory created in step (2)
6. Type 'cscript datamover.vbs' at command prompt
7. The datamover.vbs script will automatically cycle through and run all configuration files providing status updates to standard output
8. Automation completes after the last configuration file is run

The configuration files will be used in order of creation date - youngest to oldest.

- ➔ C:\FOO\DATAMOVER.VBS
- ➔ C:\FOO\CONFIG1.MOJ 12/12/2012 8:00 A.M.
- ➔ C:\FOO\CONFIG2.MOJ 12/12/2012 9:00 A.M.
- ➔ C:\FOO\CONFIG3.MOJ 12/12/2012 10:00 A.M.
- ➔ C:\FOO\CONFIG4.MOJ 12/12/2012 11:00 A.M.
- ➔ C:\FOO\CONFIG5.MOJ 12/12/2012 12:00 P.M.
- ➔ C:\FOO\CSCRIPT DATAMOVER.VBS'

RUN ORDER: START -> CONFIG1.MOJ -> CONFIG2.MOJ -> CONFIG3.MOJ -> CONFIG4.MOJ -> CONFIG5.MOJ -> STOP

Creating Custom Profiles

What defines a custom workload? From a DMS perspective, any workload profile created using the DCPD defines a custom workload. Another feature that separates a custom workload from other DMS tests or workloads is the absence of data integrity checks (Error Checking disabled). What this means is that a custom profile workload can be made to be identical to one of the pre-existing workloads DMS uses in various tests, conversely, custom profiles can take it to the extreme by allowing the user to add delays, loop iterations, and LBA locations to simulate storage heat maps.

Screenshot of Custom Profile Dialog (DCPD):

I/O Parameters

Threads: 1 Target: SCSI DISK

I/O Depth: 1 Overlapping: ☐

Transfer Size: 1MB Delay: 0

Buffer Size: 1MB Loop: 1

Percentages

Sequential: 100% / 0% Random

Read: 100% / 0% Write

Target Geometry Access By Percentage

Start: 0% End

Stop: 100% End

☒ Enable

Target Geometry Access By Offset

Start: 0

Stop: 1

☐ Enable

Name:

Elements: 0

Item	Chunk	Buffer	I/O Depth	% R
------	-------	--------	-----------	-----

Copy Up Down Add Delete

Save Reset

Menu Items

File Import Profile

Imports a previously saved profile

GUI Items

I/O Parameters

Threads

Dropdown allowing the user to select the number of threads that will be used for this given workload profile entry.

Target

Not implemented.

I/O Depth

Dropdown allowing the user to select the number of outstanding I/O's that will be used for this given workload profile entry.

Overlapping

Not Implemented.

Transfer Size

Dropdown allowing the user to select the transfer size that will be used for this given workload profile entry.

Buffer Size

Dropdown allowing the user to select the buffer size that will be used for this given workload profile entry.

Delay

The time in (milliseconds) the thread will wait before executing the next workload profile entry.

Loop

The number of times this workload profile entry will be executed before the next entry starts.

Percentages

Sequential / Random

Slider control allowing the user to choose the sequential and random distribution for the workload profile entry

Read / Write

Slider control allowing the user to choose the read and write distribution for the workload profile entry

Target Geometry Access by Percentage

Method that controls the region on the physical target that will be used by the workload profile entry; by default, 100% of the target geometry will be used (start = 0% and stop = 100%). Region can be adjusted using the Start and Stop slider controls.

Start

Window that displays the starting location relative to percentage of total target capacity

Beginning / End

Slider control that allows the user to adjust the starting location on the target

Stop

Window that displays the ending location relative to percentage of total target capacity

Beginning / End

Slider control that allows the user to adjust the ending location on the target (**Note: Ending location must always be greater than starting location**)

Enable

Checkbox that enables 'Target Geometry Access By Percentage' when checked (this is the default setting)

Target Geometry Access by Offset

This method allows the user to enter starting block and ending block, units are in decimal

Start

Edit window allowing user to enter beginning block offset

Stop

Edit window allowing the user to enter ending block offset

Enable

Checkbox that enables 'Target Geometry Access By Offset' when checked (default setting is unchecked)

Name

The name of the workload profile consisting of one or more entries; this window will be auto populated on save if no name has been entered by the user. On workload profile import, this window will display the name of the profile.

Elements

Window displaying the number of workload entries associated with this profile

Profile Listing

List control displaying all the workload entries for this profile

Copy

Makes a copy of the currently highlighted entries and appends to the bottom of the list

Up

Moves highlighted entry up one location in the list

Down

Moves highlighted entry down one location in the list

Add

Adds a new profile entry to the list

Delete

Removes the highlighted entry from the list

Save

Saves the current listing of workload entries to a workload profile file; if the user does not provide a name for the file, one will be automatically generated based on current date timestamp. User will be prompted and asked to save or cancel.

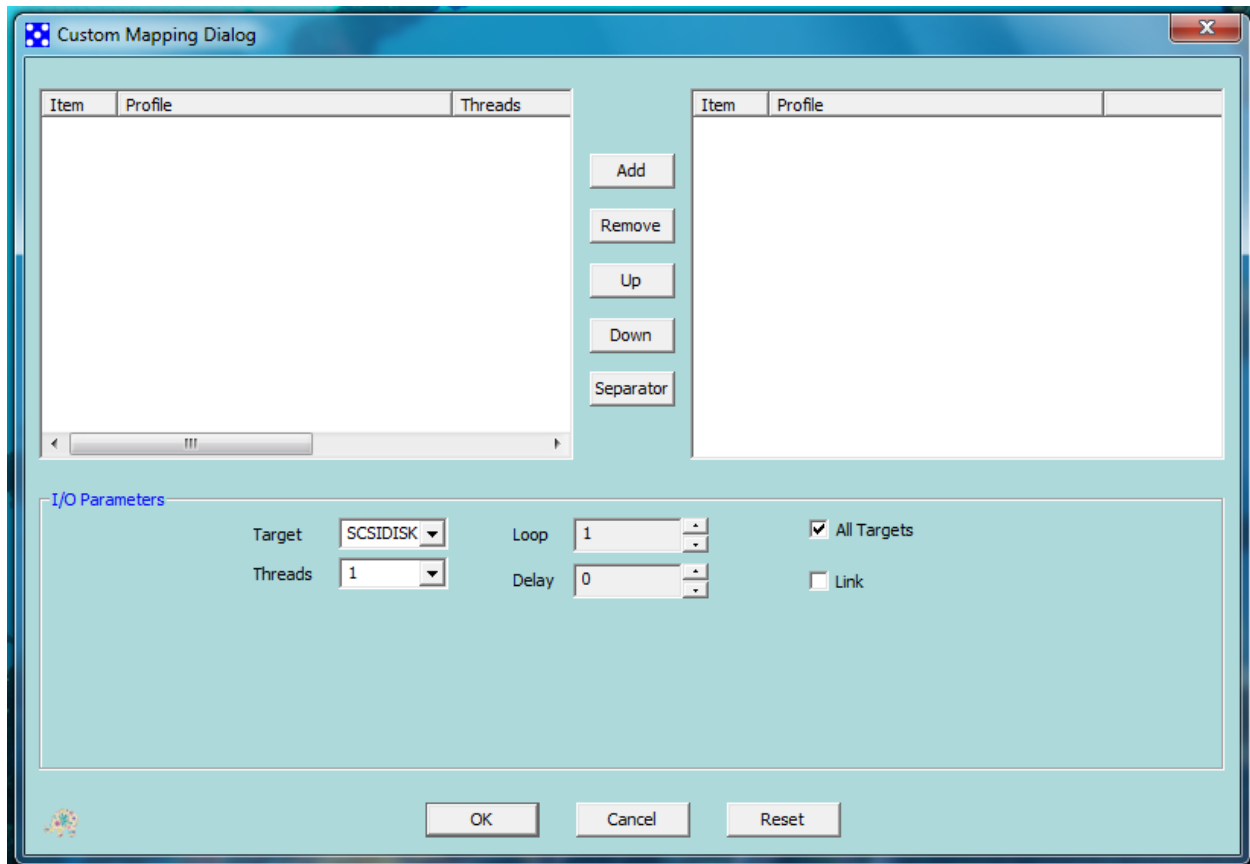
Reset

Deletes all entries from the list and resets all GUI items back to default settings

Mapping Custom Profiles

After creating a custom profile, it must be mapped to a physical target. Custom profiles are not supported on logical targets and cannot be mapped. To facilitate mapping, the DataMover Custom Mapping Dialog (DCMD) allows the user to select multiple custom profiles and map them to multiple targets.

Screenshot of CMD:



GUI Items

Left Pane

Shows the profile elements currently mapped to selected target in dropdown

Right Pane

List of all custom profiles located in directory pointed to by "Perf Results" in DPD

Add

Adds the selected custom profile selected in the right pane to the left pane creating a mapping

Remove

Removes the selected custom profile from the left pane

Up

Moves the selected custom profile in the left pane up one location

Down

Moves the selected custom profile in the left pane down one location

Separator

Adds a separator line to left pane. Separators allow the user to create independent groups of linked custom profiles. Custom profiles that are linked run within the same thread, but it may be desirable to create separate linked groupings, each running within their own thread.

Target

By default, a custom profile will be mapped to all targets per the setting of the 'All Targets' checkbox. When this checkbox is unchecked, the custom profile will only be mapped to the target selected in the target dropdown.

Threads

Dropdown that allows the user to select the number of threads that will be spawned for the custom profile to be added.

Loop

Control that allows the user to select the number of iterations that will occur for the custom profile to be added prior to moving to the next custom profile for execution.

Delay

Control that allows the user to select the delay (in ms) that will occur for the custom profile to be added prior to moving to the next custom profile for execution. If the loop value is greater than 1 for selected custom profile, looping will occur before the delay. For example, if loop = 3 and delay = 10, the custom profile will loop three times and a 10 ms delay will be inserted prior to the next custom profile to be executed.

All Targets

Checkbox that allows the user to map selected custom profile to all targets or just one. Default setting is checked, allowing mapping of custom profile to all targets.

Link

Checkbox that allows multiple custom profiles to be linked together and to run from the same thread. The state of this checkbox when a custom profile is added determines it is linked or not. When linked, the custom profile will follow the behavior of the topmost custom profile in the linked chain. This means that all linked custom profiles will use the same thread count, loop and delay settings.

OK

Closes the dialog.

Cancel

Closes the dialog.

Reset

Delete all entries from left pane and sets all dialog settings back to default.

DataMover I/O Model

A transfer buffer is sub-divided into one or more chunks. A chunk is the transfer size used by DataMover to fill a buffer of variable size. If transfer and buffer size are equal, then only one chunk will be required to fill the buffer (see Note 1); if the buffer size is greater than the transfer size, then multiple chunks will be required to fill the buffer.

Figures A and B depict a sequential and random I/O example, four chunks are required to fill the buffer; each chunk will consist of one or more contiguous blocks (aka LBA's). In this example, each chunk consists of 6 contiguous blocks and the total buffer size is equal to 24 blocks.



If the I/O is sequential, all blocks making up the buffer will be contiguous as shown in Figure A. If the I/O is random, all blocks making up the buffer will not be contiguous as shown in Figure B. The far-right column represents the block numbers associated with each chunk.

Figures C and D represents a simple 80 block device and shows where the blocks associated with each chunk are located for a sequential and random workload, respectively.

80 block drive			
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31
32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47
48	49	50	51
52	53	54	55
56	57	58	59
60	61	62	63
64	65	66	67
68	69	70	71
72	73	74	75
76	77	78	79
Figure C			

80 block drive			
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31
32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47
48	49	50	51
52	53	54	55
56	57	58	59
60	61	62	63
64	65	66	67
68	69	70	71
72	73	74	75
76	77	78	79
Figure D			

Whether the I/O is sequential or random, DataMover will perform four transfers (i.e., chunks 00 thru 03) to fill the buffer.

(1) Conditions exists where this will not always be true. If the transfer size is greater than the maximum transfer size of the adapter, the size of the transfer will decrease to conform to this limitation and the number of transfers required to fill the buffer will increase

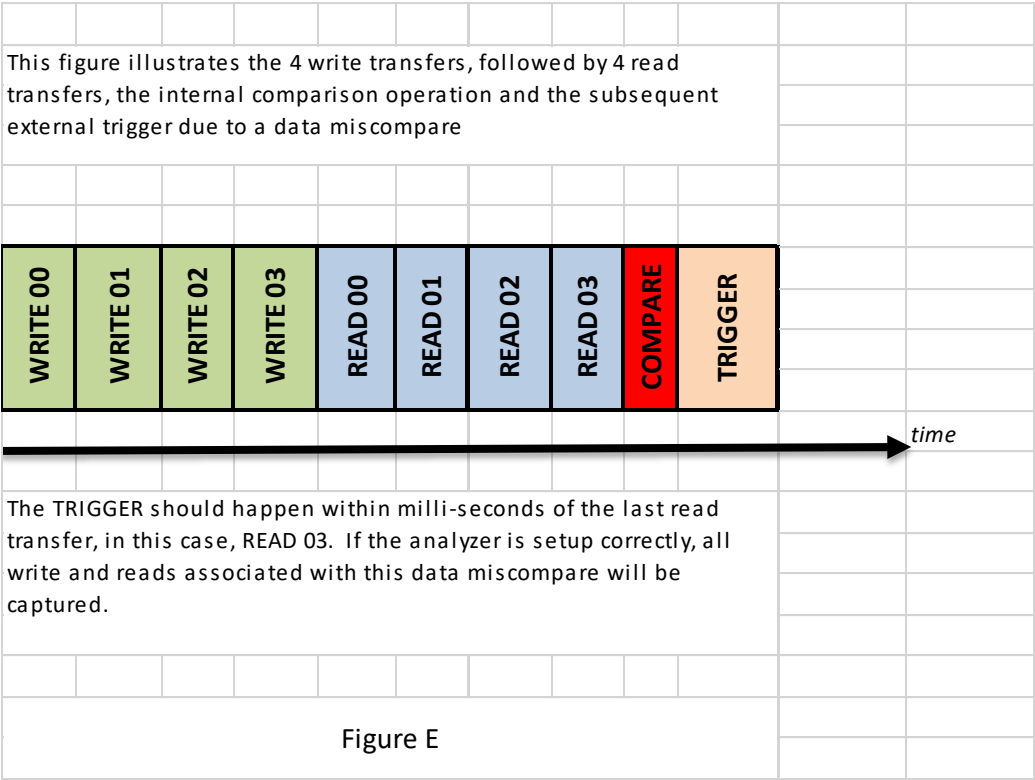
DataMover Data Integrity Error Analysis Example

Data Mismatch Bus Trigger

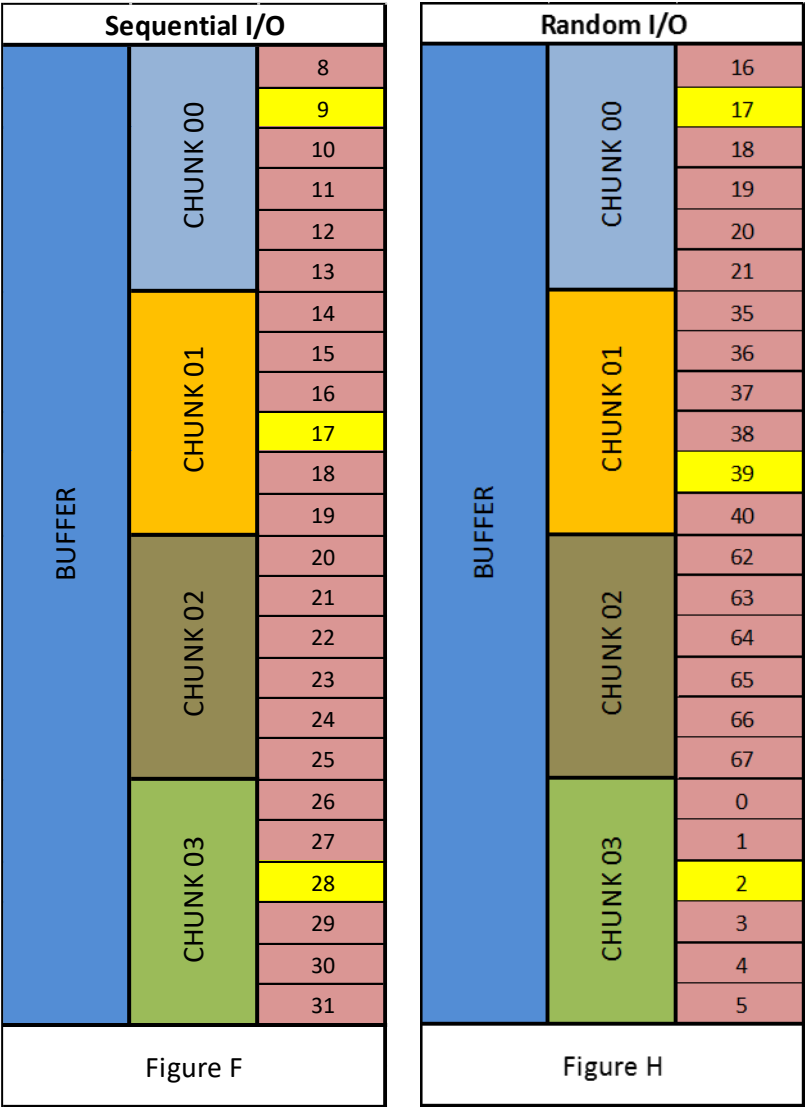
When a data mismatch occurs, DataMover will place a trigger on the bus allowing analyzers to capture protocol information near the failure. The LBA location used by the trigger command will always be the first block of the first chunk using a transfer size equal to chunk size (i.e., workload transfer size).

DataMover performs data integrity checks on the entire buffer after it has been filled. Referencing Figure E, the sequence of events leading to a data mismatch are:

- (1) Write the entire buffer using one or more transfers
- (2) Read the entire buffer using one or more transfers from the same locations
- (3) Check the contents of the buffer for any mismatches
- (4) If a mismatch is found, send a bus trigger, analyze entire buffer recording all mismatches, and post windows event



Referencing Figures F and H, we see that the buffer is comprised of multiple blocks.



Let us use the same block numbering scheme and 80 block device used in previous example (see Figures A thru D). The yellow blocks represent blocks with data mismatch. When a mismatch occurs in these blocks, it is not possible for DataMover to send a bus trigger to each LBA with a mismatch, this would cause a trigger storm and depending on the state of the device, the trigger may never appear on 'the wire.' The simplest approach is to send one trigger even when multiple mismatches occur within a given buffer.

Data Mis compare I/O Example

Data mismatches occur in the highlighted blocks at various offsets within the block (remember that blocks are typically 512B or 4096B in length).

For the sequential workload, DataMover will send the data mismatch trigger using block 8 as the LBA address and transfer length equal to 6. For the random workload, DataMover will send the data

mismatch trigger using block 16 as the LBA address and transfer length equal to 6. The goal of the bus trigger is to eventually find the exact write and read sequence that caused the failure, however, the LBA used by the trigger may or may not be close to mismatch. One important goal of the trigger has been achieved - the analyzer trigger should be close in time to the area of concern. Proceeding backwards in time from the trigger point is the correct direction to look.

To find the I/O with the mismatch, you will need to know the LBA reported by DataMover and the workload transfer size (in blocks). Hint: workload transfer size will be equal to the transfer size used by the trigger command.

The last operation that DataMover performs is to post an application event error, the detail information in the event will help us find the exact sequence in error. The event will reference the LBA, expected and actual data values.

80 block drive			
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31
32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47
48	49	50	51
52	53	54	55
56	57	58	59
60	61	62	63
64	65	66	67
68	69	70	71
72	73	74	75
76	77	78	79
Figure G			

80 block drive			
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31
32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47
48	49	50	51
52	53	54	55
56	57	58	59
60	61	62	63
64	65	66	67
68	69	70	71
72	73	74	75
76	77	78	79
Figure I			

Referencing Figures F and G, LBAs 9, 17, and 28 will be displayed in the event for the sequential workload example; and LBAs 17, 39, and 2 will be displayed in the event for the random workload example (see Figures H and I).

Using the analyzer, have it search backwards from the trigger command for any command of length 6 using an LBA in the range of any single error reported.

For example, the command LBA for the block 39 mismatch in Figure H would be in the range $33 < x \leq 39$.

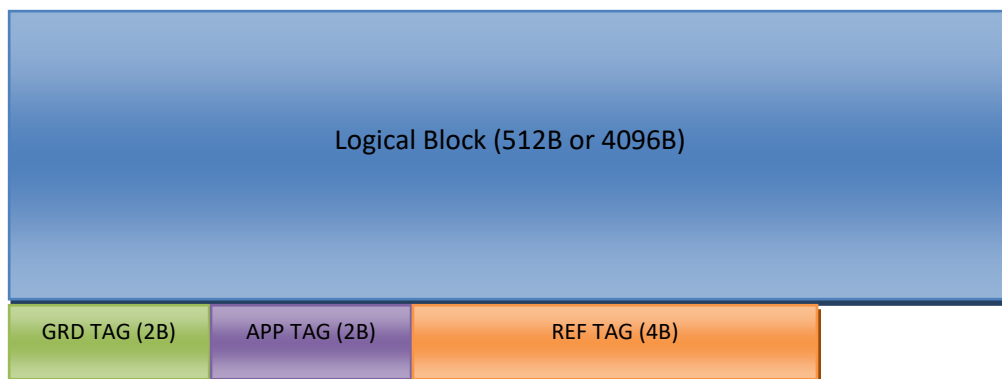
This is how to find the exact write and read sequence that produced the error.

T10 Protection Information Testing

DMS supports all T10 Protection Information (PI) types, specifically, Types 1, 2, and 3 as defined by T10 specification SBC. PI testing will be enabled when the Enable T10 PI check box has been enabled in the DPD and will be available only on the following tests: Target Test Random Standalone Mode and Target Test Standalone Mode. The user does not have to specify the type of test to be performed because DMS will automatically determine the PI type configuration and adjust the tests accordingly. If the block device is configured for PI Type 0 (PI disabled) DMS will default to normal test operations and continue. This is a good time to note that DMS does not SCSI Format the drive to configure PI on the device, it must be configured prior to the start of testing.

Protection Information Fields

This will be a short summary of the T10 PI fields and their definition to help with the explanation of how DMS tests T10 PI. For a detailed explanation of PI operation, please reference the T10 specification SBC. PI consists of 8 bytes that are appended to every logical data block (see figure below).



GRD TAG is short for Logical Block Guard, APP TAG is short for Logical Application Tag, and REF TAG is short for Logical Block Reference Tag as defined in SBC. GRD and APP TAG are 2 bytes in length, REF TAG is twice as long, or 4 bytes in length; logical block size may be 512 or 4096 bytes in length. GRD TAG is a CRC16 of the information contained within the logical block, APP TAG essentially defines ownership of the information in the logical block, and REF TAG is a counter associated with the logical block used to guard against out-of-order data delivery. Big picture summary is that GRD TAG and REF TAG guarantee data integrity guarding against such things as bit rot and faulty system design or operation. APP TAG (when used) can define ownership of the logical block and prevent other applications from over writing data unless desired. Depending on the type of PI used, some of these fields may not be used.

Methodology and Verification

This section describes the verification and operational steps performed by DMS by PI type. Prior to I/O beginning, DMS divides the transfer buffer up into multiple sections that are equal in size to the target's logical block size, calculates the CRC16 and updates the GRD TAG for each section. The APP TAG is set to a value not equal to 0xFFFFh to enable PI checks on data received and transmitted by the target. During I/O thread initialization, DMS will post an application informational event log entry providing status on T10 PI enable or disable, PI Type, and the value used for APP TAG. T10 PI status will display disable if the target does not support PI or if the target has not been PI formatted. To learn how to PI format a block device, see details on FORMAT UNIT in SBC specification. StorageWerks SCSI Utility supports PI formatting and can be used to configure the block storage device prior to DMS testing.

PI Type 1

REF TAG will be updated on every I/O cycle. *GRD TAG and REF TAG will be checked by the target on all read and write transfers.* After data is received, DMS will perform its standard data integrity check.

PI Type 2

REF TAG will be updated on every I/O cycle. *GRD TAG, APP TAG and REF TAG will be checked by the target on all read and write transfers.* After data is received, DMS will perform its standard data integrity check.

PI Type 3

GRD TAG will be checked by the target on all read and write transfers. After data is received, DMS will perform its standard data integrity check.

WRITE and READ API Errors

When error checking is enabled (a default setting for all DMS functional testing), DMS checks for errors at both the Application Programming Interface (API) and storage processing layers. An API can be any function called by a program to perform a given task. There are literally thousands of functions provided by an operating system and they are grouped into different categories based on the type of task to be performed. For example, the functions open (), read () and write () are used to manipulate files among other things. When an API is designed to communicate with a device, such as a file or block storage device, a handle will need to be opened to that device. A handle is an opaque variable provided by the

operating system that provides a communication channel to the device. The application will make all requests to the device using the handle, and multiple handles may be open to the same device from the same application.

When an API error occurs, the basic function call fails, and it is not the same as an error happening at the storage processing layer. For an I/O request to be passed to the storage processing layer, the initial call to the function performing this task must be successful. When an API error happens, Microsoft provides a 32-bit error return value which can be translated into a human readable error message string. As an alternative to returning the 32-bit hexadecimal value, DMS instead, uses another Microsoft API that translates this parameter into a message string which DMS uses to post an error in the application event log. One important thing to remember, it is not DMS coming up with a description of the error, it is Microsoft.

Most times, the message string will not be helpful in deciphering what has happened. The important takeaway for an API error is this, the handle to the device has become non-operational for any number of reasons. For example, the device may be removed from the system which would cause any open handles to the device to be invalidated by the operating system, or perhaps the device has stopped responding and the operating system is performing clean-up steps on any outstanding requests being made on the handle. In the end, only the developers at Microsoft know why a handle would become non-operational, but sometimes we can deduce what has happened by examining underlying bus traces and other system log event entries. DMS will attempt to send a bus trigger when this type of error occurs, for more on bus triggers, please see the “Bus Trigger Operation” section in this manual.

SCSI Errors and Warnings

When a SCSI warning or error message occurs, DMS will post an application event describing the physical target, thread number and sense key, asc and ascq codes in the general tab, the details tab will have the sense data.

Variable Transfer Sizes

The informational event created by a thread using variable transfers will have detail information associated with it describing the transfer lengths used. Below is an example of a thread that is using 11 variable transfers to satisfy its buffer size.

Binary data:				
In Words				
0000:	00063C00	00000000	00010600	00000000
0010:	000DEC00	00000000	0007DA00	00000000
0020:	00082800	00000000	000AD200	00000000
0030:	0004F600	00000000	00085800	00000000
0040:	00018200	00000000	00012C00	00000000
0050:	00000200	00000000		

Transfer sizes are in hexadecimal and occupy 8 bytes. This thread example is using the following transfer sizes:

- 1st transfer = 0x63C00h
- 2nd transfer = 0x10600h
- 3rd transfer = 0xDEC00h
- 4th transfer = 0x7DA00h
- 5th transfer = 0x82800h
- 6th transfer = 0xAD200h
- 7th transfer = 0x4F600h
- 8th transfer = 0x85800h
- 9th transfer = 0x18200h
- 10th transfer = 0x12C00h
- 11th transfer = 0x200h

Issue Replication Technique

When DMS finds an error, this section provides the recommended steps to recreate the problem on the same host or a different host.

1. Save the current configuration to a file, this will save the test parameter settings and the PRNG seed values
2. If DMS has stopped and is in failed state
 - a. Open DPD, turn on PRNG Seed ReUse and Stop All On Error
 - b. Save settings in DPD by clicking OK
 - c. Restart DMS by clicking Start

3. If DMS was closed or is being opened on another host
 - a. Start BD
 - b. Select Target(s)
 - c. Open AD
 - d. Import configuration saved in (1)
 - e. Perform steps 2a-c

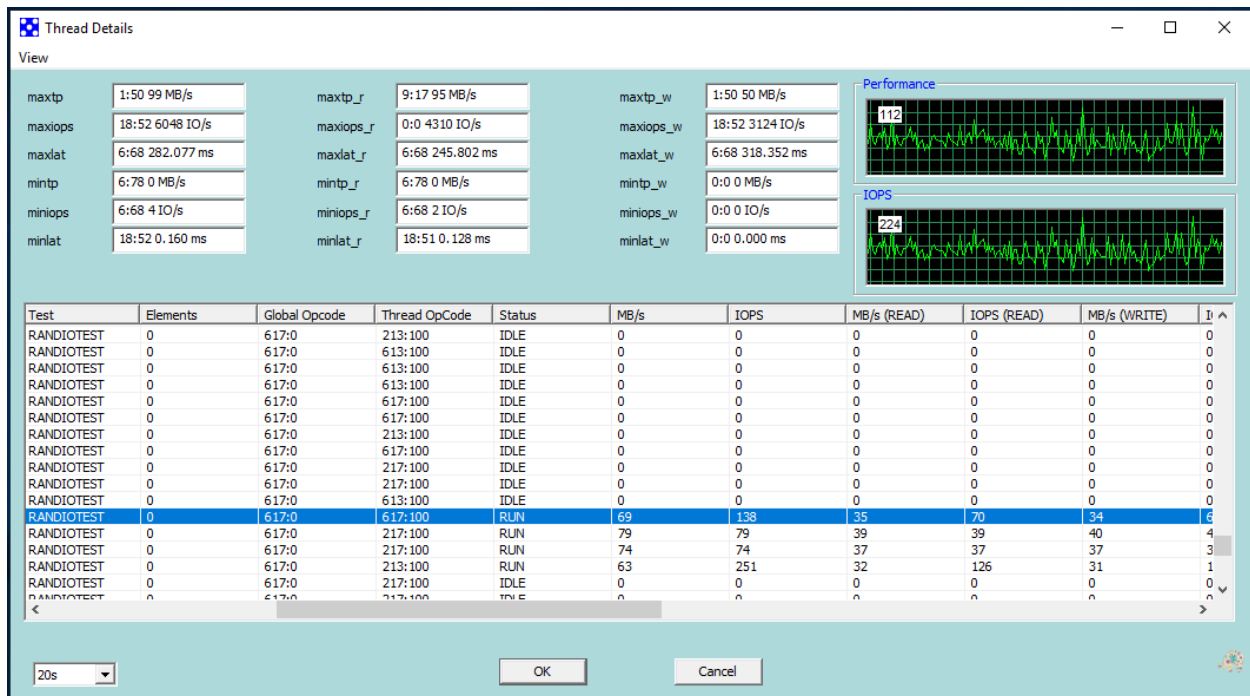
These steps provide a method to replicate the settings used in the original failure.

Logging

DMS uses the standard windows event logging system and will post informational, warning and error messages in the application event log.

Thread Details Dialog (TDD)

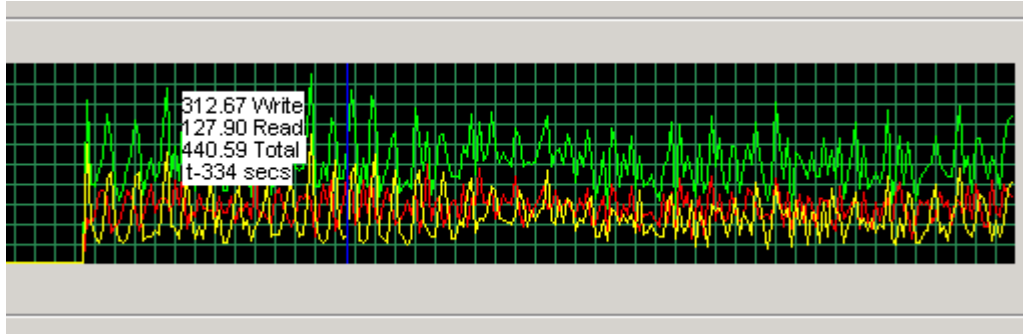
Displays detailed information on all I/O threads, including test name and performance data. When a thread is selected, performance and IOPS graphs will become active.



Appendix

Advanced Dialog Real-Time Graph Mouse over Behavior

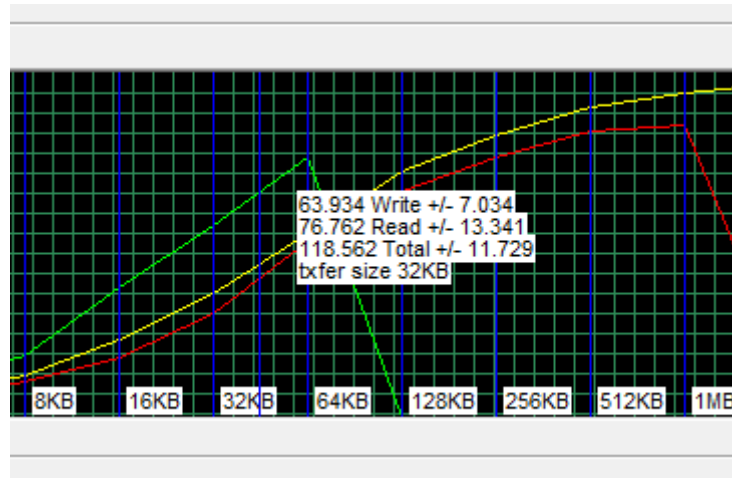
When the mouse pointer is placed over the performance and IOPS real-time graphs, the values of that point in time will be displayed. The amount of data displayed will be dependent on the “View Performance History Bytes” selection made from the AD main menu.



In this example, the vertical blue line represents the location of the mouse pointer where Write = 312.67, Read = 127.90, Total = 440.59 and the time relative to the latest sample (at t-0 sec) is 334 seconds in the past.

Performance Monitor Dialog Real-Time Chart Mouse over Behavior

When the mouse pointer is placed over the real-time charts, the values of that performance sample will be displayed. The amount of data displayed will be dependent on the “View Performance History” selection made from the PDM main menu.



In this example, the fixed vertical blue lines represent the different performance transfer sizes, and the moveable vertical blue line represents the mouse-over measurement point where Write = 63.934 ± 7.034, Read = 76.762 ± 13.341, Total = 118.562 ± 11.729 and the transfer size associated with this information is 32KB. The great thing about this feature - in addition to viewing the real-time performance chart characteristic of the system, the user may also inspect the standard deviation associated with each calculation (the value after ±). Having the ability to know the real-time standard deviation results allows the user to make judgments regarding the data set and whether to continue or end the performance test.

Glossary of Terms

BD – Basic Dialog

AD – Advanced Dialog

DMS – DataMover Storage

DPD – DataMover Preferences Dialog

DSM – Device Specific Module

PMD – Performance Monitor Dialog

EPT – Extended Performance Test

DPT – Default Performance Test

IOPS – Input/output Operations per Second

SAN – Storage Area Network

HBA – Host Bus Adapter

FCoE – Fibre Channel over Ethernet

PBD – Performance Bypass Dialog

STRD - Saturn Test Results Dialog

DCPD - DataMover Custom Profile Dialog

DCMD - Opens the DataMover Custom Mapping Dialog