

Performance Evaluation

*Integrated Systems
Advanced Technology Group*

BlockReader 1.0

Justin Bishop

April 10, 1995

BlockReader is an I/O test program which reads or writes a file and collects and reports the sum of times between file read and write request issue and request completion. BlockReader menu options permit the user to specify the number of sectors to be transferred on an I/O request, the interval between successive I/O requests and to position the disk to the start of a file so that read time measurements do not include an initial seek delay.

1. Introduction

BlockReader is a Macintosh application that reports the time taken to execute PBRead and PBWrite requests (executed synchronously). It permits the user to set the size of request and delay between requests. BlockReader can execute six different test operations, including reading and writing sequentially through a file, reading and writing randomly through a file, and alternating read and write operations to a file.

BlockReader can be configured to bypass or access the HFS Cache. When the cache is bypassed all requests access the disk drive. When the cache is accessed the cache will be checked for requested sectors on read requests and some write requests will be copied into the cache instead of being written to the disk at request time.

2. The BlockReader Application

BlockReader has three menus: *File*, *Parameters*, and *Execute*. The *File* menu has three options: *Create File*, *Select File*, and *Quit*. The *Create File* option prompts the user to enter the length of the file to create (in megabytes) and presents a Standard File dialog for file creation. Disk space for the file is allocated using the *AllocContig* File Manager call. This guarantees that the created file will be contiguous on the disk (no fragmentation). If the file create fails, the user will be notified by an Alert. The *Select File* menu item allows the user to select the file to read or write. When the menu item is selected, a Standard File Dialog is presented allowing the user to select the file. When a

file is selected the file name will appear on the first line of the window of the BlockReader application (Figure 1). When BlockReader is first launched, it will search in the current folder for a file named *Sector R/W Test File*. If the file is found, BlockReader will select the file. If the file is not found, a file must be selected or created using the menu items. The *Quit* menu item exits BlockReader.

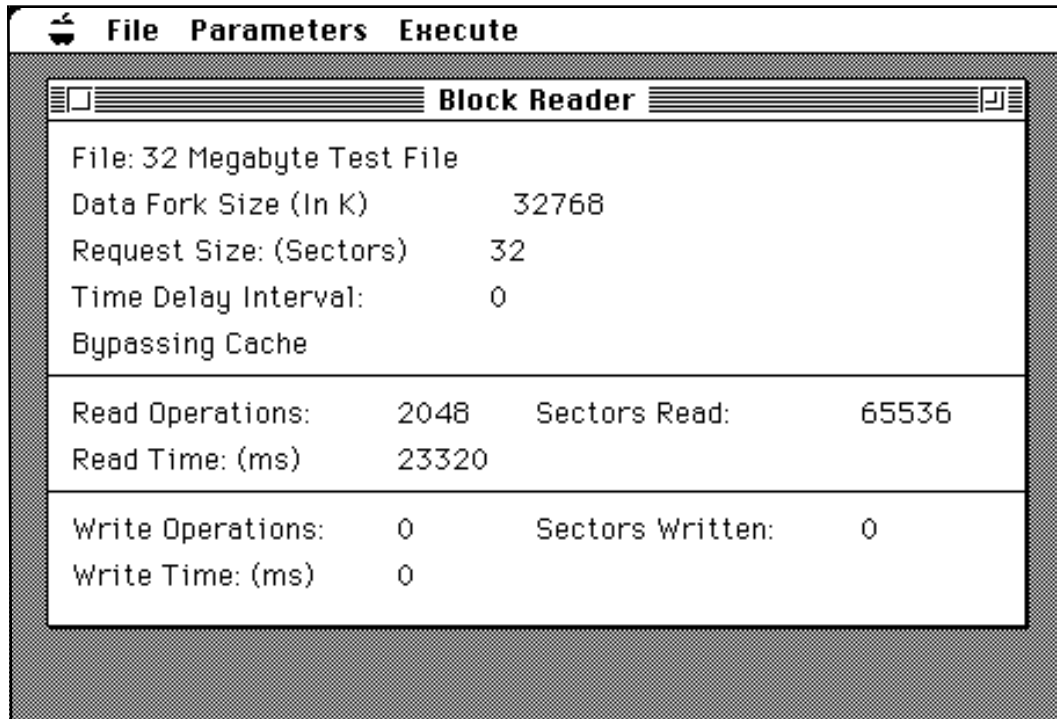


Figure 1. BlockReader Read and Write Operations.

The *Parameters* menu has four menu items: *Request Length*, *Time Interval*, *Turn Write Protection* and *Bypass Cache*. The *Request Length* option allows the user to set the block size (in sectors) of the size of request BlockReader will use in its read and write requests. The selected read length is shown on the third line of the Application's window. The *Time Interval* menu item allows the user to enter the inter-request delay (in milliseconds). The inter-request delay is the amount of time BlockReader will wait after a request has completed before issuing the next request. The delay time is shown on the fourth line of the application's window. The *Write Protection* menu item enables/disables write operations (by disabling/enabling their menu items in the *Execute* menu). This allows the user to protect against accidentally corrupting files. The *Bypass Cache* menu item controls the use of the HFS (Hierarchical File System) disk cache bypass mechanism in BlockReader's read and write requests. When the menu item contains a check mark, the bypass cache flag will be set in the Parameter Block used in PBRead and PBWrite requests (See Inside Macintosh: Files page 2-89). When the menu item is not checked,

the flag will not be set. The current bypass cache status is written on the fifth line BlockReader's window.

The *Execute* menu has seven options: *Sequential Reads*, *Random Reads*, *Sequential Writes*, *Random Writes*, *Read Write*, *Write Read* and *Position Seek*. Table 1 shows the action taken with each option. The *Execute* menu will be disabled until a file is selected or created. Pressing command period while BlockReader is executing an operation will cause BlockReader to stop execution of the operation.

<i>Operation</i>	<i>Description</i>
<i>Sequential Reads</i>	Read file sequentially. Requests are for <i>Request Length</i> sectors with <i>Time Interval</i> delay between requests.
<i>Random Reads</i>	Read File Non-Sequentially. Requests are for <i>Request Length</i> sectors with <i>Time Interval</i> delay between requests. File is broken up into (FileSize/ <i>Request Length</i>) blocks. Next block to read is selected randomly. Each block in file is read once.
<i>Sequential Writes</i>	Write File Sequentially. Requests are for <i>Request Length</i> sectors with <i>Time Interval</i> delay between requests.
<i>Random Writes</i>	Write File Non-Sequentially. Requests are for <i>Request Length</i> sectors with <i>Time Interval</i> delay between requests. File is broken up into (FileSize/ <i>Request Length</i>) blocks. Next block to write is selected randomly. Each block in file is written once.
<i>Read Write</i>	The test file is broken up into (FileSize/ <i>Request Length</i>) blocks. Starting from the first block in the file, each block is read then written. Requests are for <i>Request Length</i> sectors with <i>Time Interval</i> delay between the two operations on the same block and before moving to the next block. Blocks are accessed consecutively through the file.
<i>Write Read</i>	The test file is broken up into (FileSize/ <i>Request Length</i>) blocks. Starting from the first block in the file, each block is written, then read. Requests are for <i>Request Length</i> sectors with <i>Time Interval</i> delay between the two operations on the same block and before moving to the next block. Blocks are accessed consecutively through the file.
<i>Position Seek</i>	Issues a read for the first sector in the test file. This moves the disk head to the starting cylinder of the file.

Table 1. BlockReader Read and Write Operations.

Figure 1 shows a picture of the BlockReader report window displayed on completion of the test. The window displays the currently selected file (in the figure the file is *32 Megabyte Test File*) and the size of the data fork of the file in the first two lines of the window. BlockReader accesses the data fork of a file, not the resource fork. The window displays the current request length and time interval on lines 3 and 4. The bypass cache status is displayed on line 5. This is followed by the number of read operations executed, the number of sectors read and the time in ms of the PBRead requests issued. The last section of the window reports the number of write operations executed, the number of sectors written and the time taken to execute the PBWrite requests.