



## **CompactFlash® 5.0 Streaming Performance Control Features**

---

Revolutionary Features That Enables Predictable Performance for Applications That Need Guaranteed Minimum Levels of Performance



Authors: Yishai Kagen (SanDisk)  
Patrick Hanlon (SanDisk)

## Summary

The recently announced CF5.0 Specification adds an optional set of features called **Streaming Performance Control** that is original to the CompactFlash standard. The new feature is designed to deliver predictable performance and extend the endurance of flash memory cards. While the feature set is optional, both the CompactFlash memory card and the host must support this feature set to be effective.

The first improvement is related to a mechanism that enables the host to become aware of the internal structure of the memory card. This enables the host to structure its command sequence and payload size to optimize memory card performance and endurance.

The second optional feature set builds on the host having knowledge of the card memory structure. This feature lets the card have information about the data to be accessed (metadata) in advance, allowing the card to setup its internal operations to optimize card performance and endurance. When this second optional feature is used, the card is aware of the type of data that will be written or accessed and there is an optimization in how the data is handled by the card (stream data, random data, file system and directory updates).

A card and a host that take advantage of all the information provided, everything else being equal, are expected to produce better performance and endurance than cards and hosts that do not use the optional feature sets mentioned above.

The most advanced feature, **Streaming Performance Guarantee**, builds on these capabilities and guarantees specific performance levels provided that both the host and the card support the specific Performance Guarantee parameters.



## Introduction

CompactFlash memory cards were one of the instrumental components in the revolutionary transition from analog/film photography to digital photography that we know today. The CompactFlash memory card platform has also found uses in many industrial applications. More recently, CompactFlash has been adopted for professional grade video camcorder applications where minimum performance of the storage media is critical to quality video recordings.

Every storage medium (Hard Disk, Tape, Optical, and Flash Memory) has system characteristics that must be well understood to optimize the host/storage interaction for optimal performance to meet the application requirements. We will focus on NAND Flash Memory characteristics and applications that require a minimum performance level such as professional video camcorders.

Without an understanding of NAND Flash Memory characteristics, a host/card system may have very poor and/or unpredictable performance. Unlike Hard Disks, NAND Flash memory devices cannot erase small amounts of data. In current memory technology, the minimum block size that can be erased at one time is measured in Megabytes. Additionally, there is a minimum write size to current flash memory technology that can be 8KB or larger. The minimum write size tends to increase significantly for more advanced NAND Flash Memories and higher performance memory cards. In some cases, the minimum write size is measured in Megabytes.

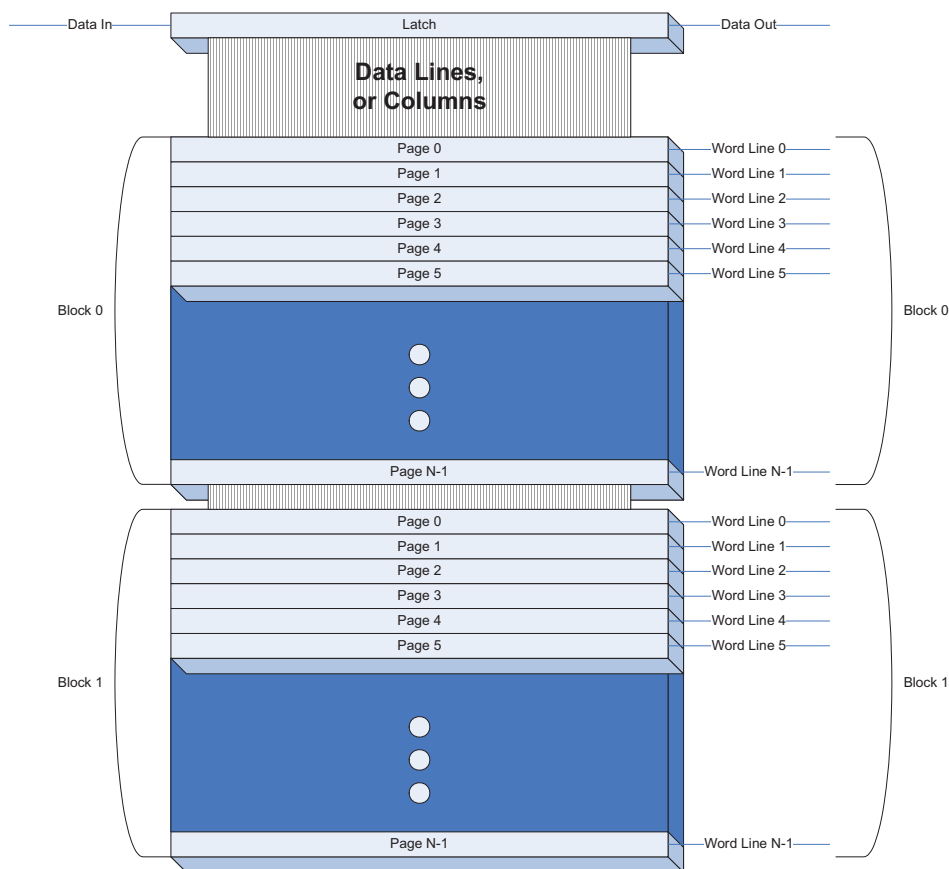
Optimizing writing and reading to NAND Flash Memory storage devices requires that the logical data access patterns be aligned to the underlying physical memory. Alignment of the logical to physical structure reduces the amount of data copy operations. Aligned writes also minimize the erase cycles within the memory card system. This translates into improved performance and improved endurance of the memory card system.

The CompactFlash interface leverages the well known and widely adopted ATA standard that has evolved from the hard disk drive industry. The ATA standard recognizes a minimum data block size of 512 bytes (one traditional sector), which was the optimal access size when the ATA standard was originally written. Over recent years there were attempts to increase the ATA sector size to 4KB. This should help, but will not solve the NAND Flash Memory performance issue for applications such as professional video camcorders.

## Basics of Flash Memory Systems

Dr. Masuoka invented flash memory technology at Toshiba when he decided that handling non-volatile memories in larger blocks, rather than on a per-bit basis as was common at that time, was much more efficient and enabled high density non-volatile memory. The name “flash” was given to the memory, according to various Internet sources attributing their source to Toshiba, by Dr. Masuoka's colleague, Mr. Shoji Ariizumi, because the erasure process of the memory contents reminded him of a flash of a camera. Intel was the first mass manufacturer of flash devices.

**Figure 1 – Simplified NAND flash structure**



Each page is written to, or programmed, at once. It is theoretically possible to program pages in parts (referred to as partial page programming), but this practice is discouraged by flash manufacturers and may not work reliably in the future. This explanation is simplified. Modern flash memory structure is more complex.



## NAND Limitations

Like every recording mechanism, NAND has its advantages and its limitations. Here is a summary of some of the key NAND limitations:

1. It is not possible to erase any data written to flash unless the whole block is erased.
2. The flash memory is written one page at a time. Each Write operation writes exactly one logical page. Not more and not less.
3. Once a page is written it cannot be changed unless the whole block is erased.
4. Each physical page contains N logical pages; N is the number of bits per cell that the memory supports. For example, SLC supports 1 bit per cell. MLC usually supports 2 bits per cell, and some advanced memories support 3 and 4 bits per cell.
5. Flash blocks can endure a limited number of Program and Erase cycles. The number of cycles each block can endure is statistically distributed.

To overcome these limitations, NAND Memory System companies have developed various techniques. To illustrate, we can look at a very simple addressing scheme is to divide the LBA space into block size chunks, and map these chunks into physical blocks. **Error! Reference source not found.** shows a simple mapping scheme for a 16MB CompactFlash card using a device containing 16 blocks + 3 spare blocks, after the device has been completely written into exactly once.

Table 1 – Flash address management example, initial memory map

Block Number	Physical Block Address (PBA)	Logical Block Address (LBA)
0	0000h - 07ffh	0000h - 07ffh
1	0800h - 0fffh	0800h - 0fffh
2	1000h - 17ffh	1000h - 17ffh
3	1800h - 1fffh	1800h - 1fffh
4	2000h - 27ffh	2000h - 27ffh
5	2800h - 2fffh	2800h - 2fffh
6	3000h - 37ffh	3000h - 37ffh
7	3800h - 3fffh	3800h - 3fffh
8	4000h - 47ffh	4000h - 47ffh
9	4800h - 4fffh	4800h - 4fffh
10	5000h - 57ffh	5000h - 57ffh
11	5800h - 5fffh	5800h - 5fffh
12	6000h - 67ffh	6000h - 67ffh
13	6800h - 6fffh	6800h - 6fffh
14	7000h - 77ffh	7000h - 77ffh
15	7800h - 7fffh	7800h - 7fffh
16	8000h - 87ffh	Spare1
17	8800h - 8fffh	Spare2
18	9000h - 97ffh	Spare3

But, what happens if the host device does not take into consideration the physical structure of the card? What happens if we want to write a 128Kbyte (131,072 bytes) block, the starting LBA of which is at address 4780h?

First, the card controller accepts the data and writes it into one of the spare blocks, for example Spare1, and changes its internal tables to point to the new physical location for this range. At some point the card goes through a cycle of what is called “garbage collection”. If the card writes new data to new locations all the time it will run out of spares. If it does not reorganize the data in blocks it will run out of controller memory for table storage.

***Garbage Collection = the internal NAND management process where the NAND system reorganizes and reclaims data blocks for future writes***

A simple algorithm may, at this point, copy data and consolidate addresses until we again have each 1MB block of contiguous LBA space mapped to a 1MB physical block.

In order to achieve this, the card will have to do the following:

1. Copy LBA 4000h-4779h from block 8 to the Spare2 block.
2. Copy LBA 4780h-47FFh from Spare1 to Spare2.
3. Copy LBA 4800h-4879h from Spare1 to Spare3.
4. Copy LBA 4880h-4FFFh from block 9 to Spare3.
5. Erase the original blocks 8, 9. Block Spare1 will continue to be written into previously unwritten pages until it is completely used up, at which point it will also be erased.

Table 2 – Resulting memory map after writing 128KB at LBA 4780h

Block Number	Physical Block Address (PBA)	Logical Block Address (LBA)
0	0000h – 07ffh	0000h - 07ffh
1	0800h - 0fffh	0800h - 0fffh
2	1000h - 17ffh	1000h - 17ffh
3	1800h - 1fffh	1800h - 1fffh
4	2000h - 27ffh	2000h - 27ffh
5	2800h - 2fffh	2800h - 2fffh
6	3000h - 37ffh	3000h - 37ffh
7	3800h - 3fffh	3800h - 3fffh
8	4000h - 47ffh	Spare2
9	4800h - 4fffh	Spare3
10	5000h - 57ffh	5000h - 57ffh
11	5800h - 5fffh	5800h - 5fffh
12	6000h - 67ffh	6000h - 67ffh
13	6800h - 6fffh	6800h - 6fffh
14	7000h - 77ffh	7000h - 77ffh
15	7800h - 7fffh	7800h - 7fffh
16	8000h - 87ffh	Spare1, partially used
17	8800h - 8fffh	4000h - 47ffh
18	9000h - 97ffh	4800h - 4fffh



The amount of activity resulting from writing 128KB to an address which is not aligned with the physical structure of the flash is the following:

- Writing 2,228,224 bytes (two complete blocks + the original 128Kbyte)
- Erasing 2 blocks, and using up 1/8 of another block that will eventually have to be erased.

The number of bytes actually written to the flash divided by the number of bytes the host wrote to the device is called Write Amplification. In this case we have a **Write Amplification of 17** (2,228,224 bytes actually written divided by 131,072 bytes the host actually was trying to write). All of this activity takes time (reducing performance) and uses up several flash Write / Erase cycles, reducing the flash endurance and reliability.

$$\frac{\text{Number of Bytes Actually Written}}{\text{Number of Bytes Host Wrote}} = \text{Write Amplification}$$

The most useful information for the host in our example regarding the card is the size of the internal blocks and the offset of the LBAs relative to the PBAs. If the host knew that the card blocks are mapped so that each 1MB (2048 LBAs) occupies exactly one physical block, and the offset is zero (meaning LBA 0 corresponds to PBA 0), it could, for example, have written the 128KB, if it had the available space, at address 4800h. This would have triggered at most a **Write Amplification ratio of 9**, and only one block erasure.

**LBA (Logical Block Address) Offset to align with PBA (Physical Block Address) significantly reduces Write Amplification which increases performance and endurance.**

If, for example, this 128Kbyte is a part of a much larger file, and the host supported the 48-bit address feature set, it could write 1MByte (2048 LBAs) at once. That way a whole new block would be written, and all the card needs to do is adjust its block mapping and erase the original block. In this case the **Write Amplification ratio is 1**. It is not possible to have a Write Amplification ratio better than 1 without data compression.

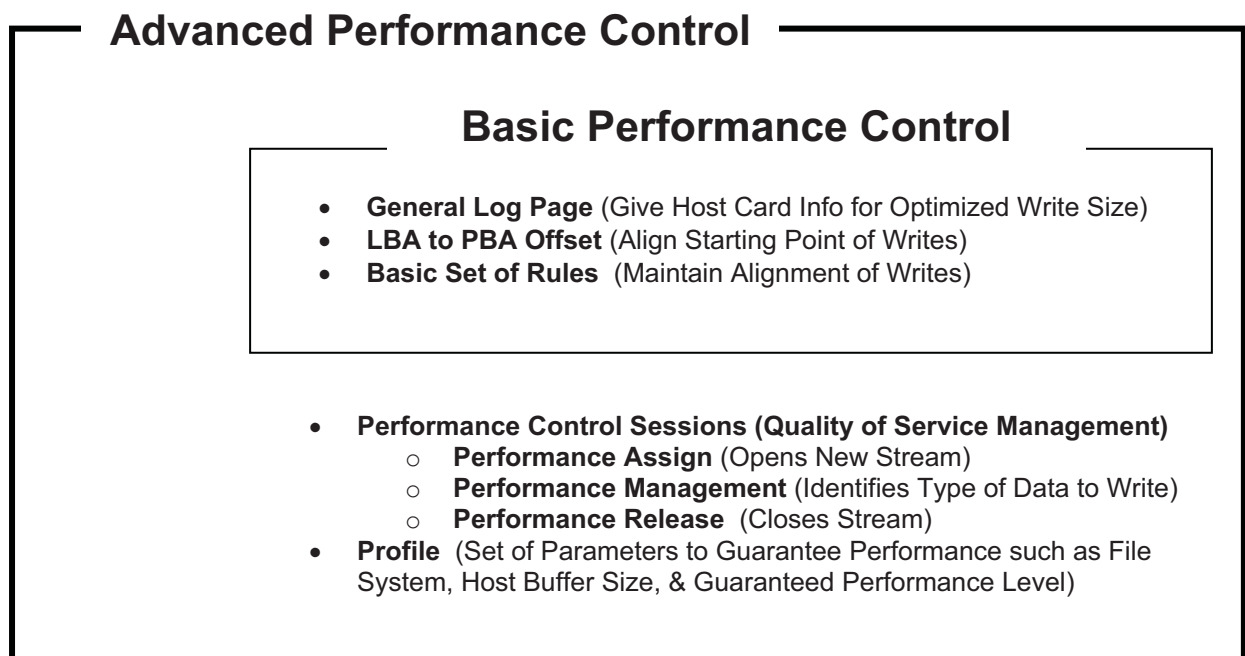
**48 Bit Addressing (New to CF5) also increases performance by increasing payload of data transfers from 128KB to 32MB.**

Similarly, data that is not written on a page boundary and not a multiple of a page size causes many more write/erase cycles than data that is written on a page boundary. The cause is increased Write Amplification.

## Performance Control – A Layered Approach

Now that we understand some background and fundamental limitations of NAND Flash Memory, we can look at how the new Performance Control Features of CF5 can be used to overcome these limitations. To reiterate, the Performance Control Features are optional and to be effective both host and card must support the features. Additionally, the Performance Control Feature Set is a layered approach so that a **Basic** approach or **Advanced** approach can be used.

Figure 2–Layered Approach to Performance Control



## Basic Performance Control

The Streaming Performance Control feature set requires support for the General Log Pages feature set. Key information regarding the card is available to the host in page 1 of address 05h, through the Read Log Ext or Read Log DMA Ext commands. This mechanism enables the card to declare underlying attributes about the physical memory to the host so that the host for the optimal LBA offset and aligned writes to reduce Write Amplification and Garbage Collection.



The following is an excerpt from the CFA specification, version 5.0

“The CFA log shall contain up to 10 Performance Control records at page #1. The records are written consecutively, at word locations 0, 24, 48, 72, 96, 120, 144, 168, 192, 216 of the page. Each record shall describe the performance of an LBA range. There is no guarantee that all LBAs will be included in the table. See Table 3 for the Performance Control Description record structure.

**Table 3 - Performance Control Description Record Structure**

Byte	Size	Name	Description
0 to 1	Word	Type	0 – Invalid record, 1 – Write performance record. 2 – Read performance record.
2 to 3	Word	S <sub>Tm</sub>	The maximum number of concurrent streams. 0 if the Performance Control functionality is not supported. There is no limit to the number of streams that can be simultaneously allocated.
4 to 5	Word	S <sub>Ta</sub>	Number of available unallocated streams. This value must be 0 or a positive number not greater than S <sub>Tm</sub> .
6 to 7	Word	RU	Size of a Recording Unit (RU) in 512 bytes blocks. For example, if the RU is 128KB this number will be 256.
8 to 9	Word	AU	Number of RUs in an Allocation Unit (AU). Continuing the previous example, if the size of the RU is 128KB, and the size of the AU is 8MB, then this number will be 64 (64*256*512 bytes = 8MB, or 8,388,608 Bytes)
10 to 11	Word	PR	The profile code for a specific performance level and specific file system usage. Profile codes are assigned in a separate document.
12 to 19	QWord	OFS	The LBA of the first sector in the first RU in the first AU this record is associated with.
20 to 23	Dword	T <sub>F</sub>	Maximum time for reading and / or writing standard file system information as defined for PR (the Profile code) required for processing one AU plus the time the card needs to transition to the next AU, in microseconds. If the value of T <sub>F</sub> , for example, is 120 milliseconds, then this field will contain the value 120,000. This field may contain 0. A value of 0 means T <sub>F</sub> is undefined. If PR is 0 then the definition of T <sub>F</sub> shall be vendor specific.
24 to 27	Dword	T <sub>AU</sub>	Maximum net time to read or write one AU, one RU at a time, or using the method defined by the profile as specified by the PR field, in microseconds. For example, if writing an 8MB AU lasts 280 milliseconds at most, this field will contain a value of 280,000. This field may contain 0. A value of 0 means T <sub>AU</sub> is undefined.
28 to 31	Dword	N <sub>AU</sub>	Number of consecutive AUs in this range. The range size is N <sub>AU</sub> *AU*RU*(LBA size).
32 to 47		Reserved	Set to 0

The important numbers for a Performance Control aware host are **RU**, **AU**, and **OFS**.

**RU = Recording Unit**

(In our example, is equivalent to one page and we set it to 8Kbyte)

**AU = Allocation Unit**

(In our example, it is 1MByte)

**OFS = LBA Offset to Align with Card PBA**

The entries for **OFS**, **RU** and **AU** for the example would be the following:

**OFS=8192**

**RU=16** (8KB = 16 X 512 byte blocks)

**AU=128** (there are 128 8Kbyte RUs in one AU)

The relationships to the physical address space of the card appear in figures 3 and 4.

Figure 3 – Illustration of Card with Defined AU's

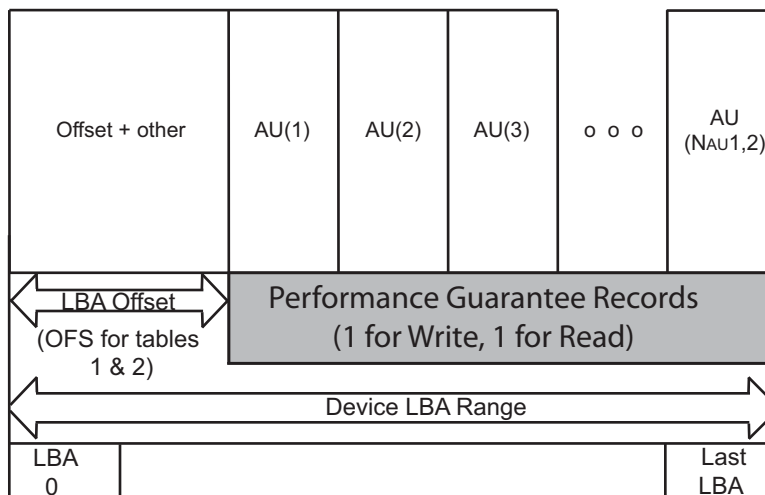
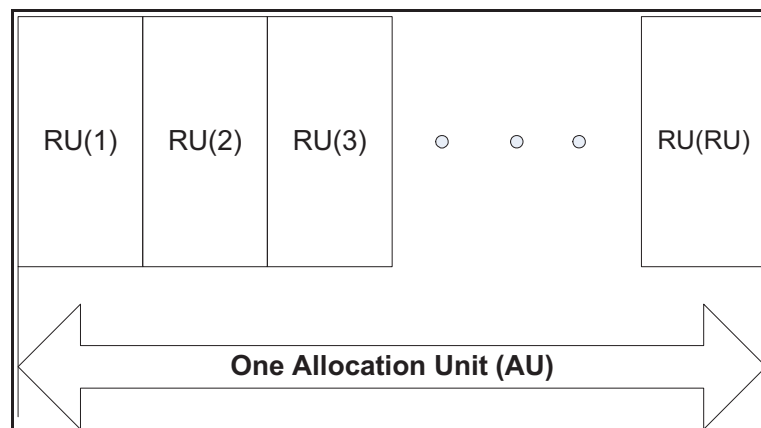


Figure 4 – Division of AU's into RU's





A set of rules for writing is provided next. There are many methods of managing flash memory, and following the rules given here may produce better results on some cards than on others. Following some of the rules may not cause any improvement on some cards; however, following them should not reduce performance or reliability.

- This set of rules apply only to the actual data, not to file system management. The rules do not affect file allocation tables, subdirectories, and other file system entries.
- The starting LBA of a Write command is the first LBA on an RU.
- The length parameter is equal to an integer number of RUs. This only helps performance if the previous rule is met.
- When writing long sequential files, allocate LBAs for the write that span exactly one or an integer number of currently free allocation units. The allocation units do not have to be allocated sequentially to each other, only the data written into each allocation unit needs to be written sequentially.

The address of the first LBAs on recording units is equal to  $OFS+n*RU$ , when  $n$  is an integer greater from or equal to 0.

The address of the first LBA on allocation units is  $OFS+m*RU*AU$ , where  $m$  is an integer greater from or equal to 0.

It is also beneficial to some degree to follow the RU boundaries when reading, since reading one RU from the flash triggers one read operation, but reading data that resides on two RUs, even if their total length is less than one RU, will trigger two flash read operations, and waste some time.

## Advanced Performance Control

Basic Performance Control dealt with performance and reliability improvement by the host when it has access to information regarding the physical structure of the card. This section will deal with active performance and reliability improvement enabled by dynamic information exchange between the card and the host.

Three new commands were added to the CFA specification version 5. The new commands are:

- **Streaming Performance Assign (code BBh, features 2,3)**
- **Streaming Performance Management (code BBh, feature 4)**
- **Streaming Performance Release (code BBh, feature 8)**

There are several ways to use these commands. One method is to use the **Streaming Performance Management** command only. The purpose of this command is to communicate to the card the nature of the data to be written.



The host writes 3 different kinds of data:

1. **File System Data** that has a fixed logical address. For example, a file allocation table when using a FAT file system will always reside in a fixed logical address once the card has been formatted, and will be used for all card accesses.
2. File system or auxiliary data that has a varying logical address. This includes also actual file data that has to be written in small chunks, for example **sector updates** needed to close a file. Another example is a **Sub-Directory** entry. The entry will be written into as long as the file it points to is being written, but not used otherwise.
3. The **Data** that is being written as a stream, for example a picture or a video stream. The host can communicate to the card the addresses that are immediately afterward written into in each of the above cases using the Streaming Performance Management command. The card may be able to handle the different types of information differently, for example, through an internal cache mechanism.

If the **Performance Management** command is used with a data range **Type of 3** (the actual data to be written), the host should be aware that only complete allocation units may be safely defined using this command. All previous data in an allocation unit that contains addresses defined by this command as data range type 3 may be lost. The command may effectively erase the previous contents of the whole allocation unit, since its function is to prepare the AU to receive new data.

## Performance Control Sessions

It is possible to use the **Streaming Performance Control** commands in a more structured manner. We can, using the **Streaming Performance Assign** and **Streaming Performance Release** commands to set up a write session. The advantage of assigning a file stream ID and releasing it is that the card may temporarily modify its internal behavior to support the highest possible performance for the assigned file stream. Any data range type definition having the assigned file stream ID that occurs between the **Performance Assign** and **Performance Release** commands will not be remembered by the card after the Performance Release command for that file stream ID command is executed.

If the host writes complete allocation units as much as possible during a stream file session, and the card delays its internal maintenance activities as much as possible during the session, the card performance will be as high and consistent as possible.

It is important that the file streams will be released once they are written, and the card given time to finish delayed internal maintenance (example – **Garbage Collection**). If stream files are never released the card may exhibit unpredictable behavior. In some cases it will run out of spare internal resources, at which point it may become slow. Another example is wear leveling. If it is never allowed, card reliability may suffer.



## Streaming Performance Guarantee Based On Profiles

The Performance Control feature also enables the card and host to use Profiles to further optimize performance of the host/card system.

A profile is a document that details host – card access patterns and behavior, so that a certain performance is guaranteed. The support of a specific profile is indicated by the PR word in the Performance Control Description record, see Table 3. If the value of PR is not zero, it means that the card supports a particular profile. A profile is a contract between the card and the host. If the host follows the rules defined in the profile for the host, the card guarantees a certain performance and / or other predictable behavior.

**Profile = Contract between host and card that guarantees performance**

The initial use of a **Profile** is for recording video applications. For a quality video recording, the card needs to guarantee that no data/frames will be lost. If the host supports the same profile as the card, and follows the access patterns and rules defined in that profile in conjunction with the other fields in the Performance Control Description Record, then the card will fulfill its part of the contract implied by the profile specification.

### Example of a part of a simple profile:

- **Profile Number** = XXXXh (Registered with CFA)
- **Guaranteed Video Stream Speed** = 20MB/sec
- **Minimum Host Buffer Size** = 60MB. No more than 60MB will be required to be cached on the host during the Streaming Performance session.
- **File System**= FAT32
- **File System Update Frequency** = 5 times per second
- **Random sector writes allowed** = 20 consecutive single sector commands once a minute, starting one minute after the stream starts to be written, in addition to the stream data and file system required operations.

When the host supports the XXXXh profile and recognizes a CompactFlash card that supports the XXXXh profile, it will use the Streaming Performance Assign and Streaming Performance Release commands at the beginning and end of the recording session. It will follow the above mentioned rules regarding allocation units, recording units, and offset. It will issue Performance Management commands specifying the LBAs used by the stream and the file system areas to be modified. It will also follow any specific rules that are specific to the XXXXh profile.



# Whitepaper

---

As a result, it will be guaranteed to record the video at 20MByte per second, and, unless there is a major card failure, the host buffer will never overflow and no frames will be lost.

Specific Profiles will be registered and available through CFA. Specific Profiles will also be accompanied by a compatibility logo so that the proper host/card combination can be identified. Use of the CFA trademarks such as specific profile compatibility logos requires the execution of a license agreement with the CFA.

If you are interested in finding out more about CF5.0 and the new Streaming Performance Control Feature Set, please visit the CompactFlash Associations website: [www.compactflash.org](http://www.compactflash.org). At the CFA website you will find published specifications for sale and other notices regarding new CFA specifications. We encourage you to join the CFA to participate in the definition of future specification.

CompactFlash is a registered trademark of SanDisk Corporation and licensed to the CFA (CompactFlash Association) and by the CFA to members through the CFA Licensee Agreement.