GC20-1879-0
File No. 4300-34

Program Product

Fixed Block DASD Installation/Conversion Guide

VSE/Advanced Functions

Program Number 5746-XE8

Release 2

IBM

GC20-1879-0
File No. 4300-34

Program Product

Fixed Block DASD Installation/Conversion Guide

VSE/Advanced Functions

Program Number 5746-XE8

Release 2

IBM

The purpose of this publication is to aid in the conversion from System/360 and System/370 DASD to the new DASD on the IBM 4300 Processors. For the most part, the 4300 Processors are fully compatible with the System/370; therefore, this guide will be concerned primarily with I/O device considerations.

Not all readers will be interested in every chapter of this guide. The following will help in determining the proper sections to read:

- Chapters 1 through 7, Chapters 10 through 12, Appendix D, and Appendix E should be of interest to all readers.
- Chapter 8 should be of interest to 231x DASD users who wish to use the 231x compatibility feature of the IBM 4331.
- Chapter 9 should be read by IBM System/3 users who are converting to the 4331.
- Appendix A is intended for the SAM access method user whose programs are written in Assembler.
- Appendix B is for the DAM, DTFPH/EXCP user.
- Appendix C should be of interest to the VSE/VSAM user.

First Edition (August 1979)

This edition applies to Release 2 of VSE/Advanded Functions, program number 5746-XE8 and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. Changes are continually made to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest IBM System/370 Bibliography, GC20-0001, for the editions that are applicable and current.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Publications are not stocked at the address given below; requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Laboratory, Programming Publications Department, Schoenaicher Strasse 220, D-7030 Boeblingen, Germany. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, or course, continue to use the information you supply.

This Page Intentionally Left Blank

New DASD Devices

Two new DASD devices are introduced with the IBM 4300 Processors: the IBM 3310 and the IBM 3370. These devices provide unique improvements in price/capacity, price/performance, reliability, attachment flexibility, and migration ease, which are essential in optimizing the benefits of migration to the 4300 Processors. To accomplish this, the 3310 and the 3370 incorporate a data organization known as Fixed Block Architecture or FBA. Data is no longer organized in variable blocks with a variable number of interrecord gaps. It is now organized in fixed blocks of 512 bytes on the new DASD. Programming is no longer dependent on device characteristics other than the maximum number of blocks on the DASD. These devices allow programming to be independent of the geometry of the physical device.

Modes of Operation of the IBM 4300 Processors

The IBM 4300 Processors offer IBM users compact central processing units with large storage, fast internal speeds, and a wide range of integrated I/O attachments. The 4300 Processors may operate in two modes:

- ECPS:VSE (Extended Control Program Support:Virtual Storage Extended).
- System/370.

ECPS:VSE Mode

In ECPS:VSE Mode there is a new and simpler way of one-level, dynamic address translation for single virtual storage of up to 16 megabytes. Also, there is no need for channel program address translation. The DOS/VSE supervisor ensures that the I/O areas and CCW strings are fixed in memory, but no translation of virtual addresses to real addresses is required as the channels may now use virtual addresses. DOS/VSE supports the 4300 Processors in ECPS:VSE Mode. In this mode native IBM 3310 and 3370 data sets may be accessed along with emulated 231x on the IBM 3310 on the 4331. IBM 231x DASD may be attached to the 4331 and 4341 via the Block Multiplexor Channel. 3340 DASD may be attached via DASD Adapter to the 4331. 33xx DASD may be attached to the 4341 via the Block Multiplexor Channel.

System/370 Mode

System/370 Mode supports both BC (Basic Control) Mode and EC (Extended Control) Mode. In System/370 mode, the 3310 and 3370 DASD may be attached but are supported only by VM/370 Release 6 with VM/VSE Release 2 and DOS/VSE with VM=YES, specified in the supervisor and running under VM/370. Emulated 231x DASD on the 3310 attached to the 4331 may be accessed by VM/370, DOS/VSE, DOS/VS, or DOS Release 26. IBM 231x DASD may be attached via the Block Multiplexor Channel to the

4331 and the 4341. IBM 3340 DASD may be attached via the DASD Adapter to the 4331. IBM 33xx devices may also be attached via an appropriate control unit to the Block Multiplexor Channel of the 4341.

The following operating systems are supported in System/370 Mode:

- DOS/VSE
- VM/370 Release 6 with VM/BSE Release 2 (ECPS:VM is supported)
- VS/1 Release 7 (ECPS:VS1 on the 4341 is supported)
- DOS/VS Release 34
- DOS Release 26

Hardware Architecture

The principle of fixed block architecture (FBA) is that each storage area or device is a string of equal-length blocks. The blocks are numbered from 0 to (n-1), where n depends on the capacity of the direct access storage medium. All blocks have a length of 512 bytes. In fixed block architecture there are no keys, as with Count-Key-Data DASD. A block is identified only by its Relative Block Number. With FBA you need not be aware of the physical structure of the direct access storage medium, nor concerned with the capacity of tracks, the number of tracks per cylinder, and the number of cylinders per DASD. The only item of concern is the maximum number of blocks per address.

Fixed block architecture makes the application program independent of the geometry of the physical storage device.

Data Format

The fundamental unit of data transfer between the host system and FBA DASD device is a fixed-length block of data. A record, as viewed by the devices, consists of a 512 byte data block plus a block control field or ID.

The block control field contains the address of the block plus optional device-specific control information. The address of the data block is a linear binary number from 0 to (n-1). The address range spans a single unit of physical media (such as a DASD actuator or "arm").

The block control field and the data block are recorded as separate fields on the media. The transfer unit between the host system and the I/O devices is in multiples of 512 byte data blocks. The control fields are not transferred to the host system.

Fixed block architecture allows more data to be placed on a given storage area and facilitates easier transition of data between devices. With all data stored in fixed 512 byte blocks, the storage space normally reserved for the count field and the identification key is eliminated. With both the Count and Key fields eliminated, the number of interblock gaps is reduced when less than full-track blocking is used. Therefore, more storage space is reserved for data because less storage is required to identify the data. FBA DASD capacity is stated as usable space, while the capacity of CKD devices is stated in usable space only if full-track blocking is used.

Fixed block DASD offers performance and data storage advantages over existing CKD DASD.

The IBM 3310

The 3310 is attached to the 4331 and is a single spindle, single actuator unit with a capacity of 64.5 million bytes (megabytes).

The 3310 comes in four models. The A-1 model contains a single spindle but is incapable of attaching additional models. The A-2 model contains two spindles and can attach either a B-1 or B-2 model. The B-1 model attaches to the A-2 model and has one spindle. The B-2 model attaches to the A-2 model and has two spindles. The 4331 maximum configuration capability is four A-2 models (each of which may have a B-1 or B-2 model attached) attached to the DASD Adapter. Thus, the IBM 4331 can support up to 16 3310 spindles. This gives the 4331 a maximum online data base capacity of 1,032 million bytes with the 3310.

The IBM 3310 is a fixed media DASD. Figure 2.01 shows its characteristics.

Spindle capacity (megabytes)	64.5
RPM	3,125
Average latency (milliseconds)	9.6
Data rate (megabytes per second)	1.031
Average access time (milliseconds)	27
Bytes per block	512
Blocks per spindle	126,016
VM/370 support	Yes
DOS/VSE support	Yes

- Figure 2.01: 3310 Characteristics

The IBM 3370

The IBM 3370 attaches to the 4331 via the DASD Adapter; it also attaches to the 4341 via the 3880 Control Unit on the Block Multiplexor Channel. The 3370 offers a large capacity for online data bases. The 3880 Control Unit has two paths, each of which can attach up to four 3370 strings. The 3880 also attaches the 3330, 3340, and 3350 DASD.

The 3370 consists of two actuators on a single DASD spindle. Each actuator has a separate logical volume of 285 megabytes and a separate DASD address on the string; it operates independently of the other actuators on the string. A fully-loaded IBM 3370 string consists of an A-1 model and three B-1 models, with a total capacity of 2,280 million bytes. Each model contains two DASD addresses.

The 3370 is a fixed media DASD. Figure 2.02 shows its characteristics.

Actuator capacity (megabytes)	285
RPM	2,964
Average latency (milliseconds)	10.1
Actuators per spindle	2
Data rate (megabytes per second)	1.859
Average access time (milliseconds)	20
Bytes per block	512
Blocks per actuator	558,000
VM/370 support	Yes
DOS/VSE support	Yes

- Figure 2.02: 3370 Characteristics

This Page Intentionally Left Blank

Software Support

DOS/VSE Support

DOS/VSE fully supports the 4300 Processors and attached DASD, including the 3310 and the 3370. Both ECPS:VSE and System/370 modes are supported. ECPS:VSE mode is recommended on the basis of performance implications. The 3310 and 3370 DASD in native mode are supported in ECPS:VSE Mode. In System/370 Mode, the 3310 and 3370 are supported only while running under VS/370 Release 6 with VM/BSE Release 2. In System/370 Mode (not running under VM/370) the 3310 DASD is only supported in 231x compatibility mode.

DOS/VS Support

DOS/VS Release 34 runs on the 4300 Processors in System/370 Mode but does not support the FBA DASD (except for the 3310 in 231x compatibility mode).

VM/370 Support

VM/370 Release 6, with VM/BSE Release 2, supports the 4300 Processors in System/370 Mode only. The 3310 or 3370 DASD may be attached and supported in native FBA mode by both DOS/VSE and VM/370. The VM Linkage Enhancements of DOS/VSE VSE/Advanced Functions allow the 3310 and the 3370 to be supported in System/370 Mode under VM/370.

VM/370 supports the fixed block DASD for:

- CMS disks (including temporary disks and minidisks)
- System residence and paging

VS1 Support

VS/1 Release 7 supports the 4300 Processors in System/370 Mode only. The fixed block DASDs are not supported. 231x compatibility mode on the 3310 attached to the 4331 is not supported.

DOS Support

DOS Release 26 may be run on the 4300 Processors in System/370 Mode, but it does not support the new 3310 and 3370 DASDs in fixed block mode. 231x compatibility mode on the 3310 attached to the 4331 is supported.

Defining the IBM 3310 and 3370 to DOS/VSE

Under DOS/VSE, both the IBM 3310 and the 3370 are defined to the system as device type FBA. This follows the trend of defining various devices by class rather than by specific type (for example, PRT1 for 3203-4, 3211, etc.).

- ASSGN example: // ASSGN SYS010,FBA,VOL=111111,SHR
- IPL ADD example: ADD 241,FBA

Defining the 231x Compatibility Feature

Emulated 2311 devices are defined to DOS, DOS/VS, and DOS/VSE as device type 2311. Emulated 2314 devices are defined as device type 2314.

Programming Languages Supported

The following programming languages are supported by DOS/VSE:

- Assembler
- DOS/VS RPG-II 5746-RG1
- DOS/VS COBOL/VS 5746-CB1
- PL/I Optimizer 5736-PL3
- FORTRAN IV 360N-FO-479

3310 and 3370 Support

DOS/VSE supports both 3310 and 3370 natively with the following access methods:

- SAM (Sequential Access Method)
- VSE/VSAM (Virtual Storage Access Method)
- PIOCS (Physical Input/Output Control System)
- ISAM, via the ISAM Interface Program (IIP) of VSE/VSAM

In 231x compatibility mode on 3310, all previously supported access methods (DAM, ISAM, etc.) are fully supported.

Sequential Access Method (SAM)

SAM is the acronym for Sequential Access Method. While there are some changes in the DTF specification (to provide new support), most programs using the Sequential Access Method for Count-Key-Data (CKD) files are object compatible with the Sequential Access Method for FBA units. A prime objective of the fixed block architecture is to support the trend away from device-dependent programming. In line with this strategy, the

current application dependence on CKD physical device parameters become meaningless on FBA devices and do not have FBA sequential disk support, such as: Split cylinder extents or Count-Key-Data addressing.

Most programs that run on current Count-Key-Data devices and utilize the Sequential Access Method will run unchanged with FBA devices. It may be desirable to change block sizes to more efficiently utilize the FBA devices. Although most current programs will execute without reprogramming, optional DTF parameters are available for convenience in maximizing device space utilization and performance.

Since data must be transferred in units of fixed size, an additional mapping of SAM physical blocks into FBA physical blocks has been added. Because of this, programs that are sensitive to physical I/O synchronization, such as error exits and logging, must be reevaluated and may require programming changes. For programs that are not sensitive to I/O synchronization, this extra level of blocking should be transparent and can provide a performance advantage.

Data Set Organization

A SAM data set on an FBA device is viewed as an ordered set of data transfer units. This unit of data transfer has the same format as the VSE/VSAM Control Interval (CI) and was chosen to ease data migration.

The following definitions are used to describe the data format and processing of SAM files on FBA devices:

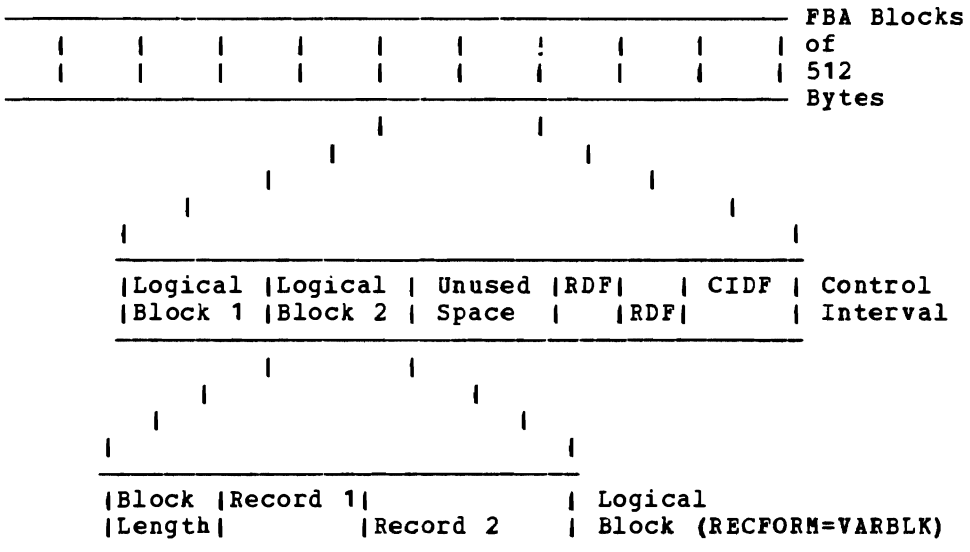
- Logical Record - One record as the application program perceives a record (same as for CKD devices)
- Logical Block - One or more logical records grouped together. A logical block consists of one unit of data as it would be transferred to a CKD device (physical block). This data entity is called a logical block because several of them may be combined to form a data unit for transfer to an FBA device. The blocking structure, as specified by the DTF BLKSIZE parameter, is maintained in logical blocks to provide support for current programs that depend on the CKD physical block concept. When working with unblocked logical records, a logical block contains one logical record.
- FBA Block - A fixed length physical block upon which the fixed block architecture is based. This block size is defined as 512 bytes.
- Control Interval (CI) - This is the unit of data transfer between storage and FBA devices. It has the same format as a VSE/VSAM control interval. In recording the data, LIOCS maps each Control Interval over an integral number of FBA blocks.

For files being opened for output, the size of the Control Interval is determined by OPEN routines (on the basis of user-specified DTF parameters such as CISIZE, BLKSIZE, or RECSIZE, or the user-specified CISIZE parameter on the DLBL statement) and will conform to VSAM CI size restrictions and the FBA blocksize. The CI size will be stored in the Format-1 label on each volume when a new file is created, and will be retrieved from the first volume when a file is opened for input.

The SAM stored record formats do not change in FBA. The mapping of SAM stored records into CI format is handled completely by LIOCS and is irrelevant to problem programs that use LIOCS (non-EXCP) imperative macros to access data.

Each logical block (for example, a fixed-length block if RECFORM=FIXBLK; a variable-length block if RECFORM=VARBLK) is handled as a record in the CI. If CI size determination results in a CI that will hold more than one logical block, the support will automatically place as many logical blocks with associated control information in the CI as possible. Any leftover space in the CI which was insufficient to contain the next logical block and its control information remains unused. Logical blocks that are too large to fit in the maximum CI size (32K) are not supported. The BLKSIZE value will be limited to 32,761 bytes (32,768 minus seven bytes of control information) when DEVICE=FBA is specified, just as it is limited by the track size of Count-Key-Data devices. Logical blocks will not span CIs. For SAM work files (TYPEFLE=WORK) the number of logical blocks per CI is limited to 255 because of compatibility constraints with current CKD support.

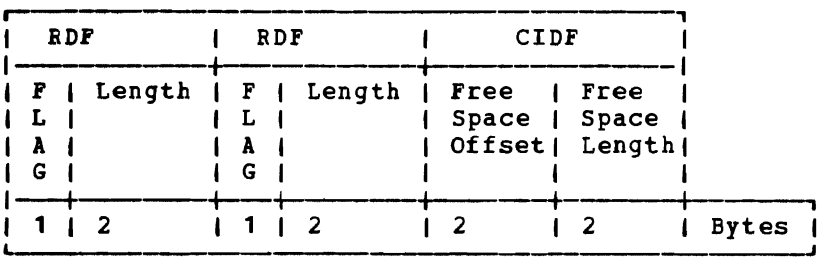
Control information in the CI consists of a Control Interval Definition Field (CIDF) and Record Definition Fields (RDF). A CIDF contains a two byte offset to the beginning of free space in the CI followed by a two byte length of free space. A RDF contains a control flag byte, followed by two bytes which indicate the length of the corresponding logical block. There is a RDF for each logical block, except as indicated on the next page for consecutive logical blocks of equal length. Figure 3.01 shows the basic format.



• Figure 3.01: Logical Block within a CI

Figure 3.01 shows a Control Interval of 1,024 bytes which contains two logical blocks. Logical block two contains two logical variable-length records.

Figure 3.02 is an example of the control information in a Control Interval (CI).



• Figure 3.02: Control Information in a CI

When more than two consecutive logical blocks within the same CI have the same length, the pattern of one RDF for each logical block is broken. Two RDFs are used to describe the set of consecutive logical blocks of equal length. The first RDF (on the right) describes the length for the logical blocks; the second (on the left) tells how many logical blocks the first RDF describes.

As long as TRUNCS is not used, the logical blocks containing RECFORM=FIXUNB|FIXBLK records are all the same length. Thus, the record control information in the CI consists of exactly 2 RDFs instead of one RDF per record, if there is more than one logical block in the CI.

When a SAM user specifies spanned records, the logical records are disassembled/reassembled into logical blocks just as SAM presently maps into physical blocks. The logical blocks are mapped into CIs, as with any other variable-length logical blocks. The SAM logical record can span logical blocks and, therefore, span CIs.

A CI containing all zeros is considered a software end-of-file (SEOF) indicator. A SEOF is used to delimit a file or a portion of a file on the current volume. A file is also considered delimited by end of last extent on input.

VSE/Virtual Storage Access Method (VSE/VSAM)

It is not necessary to change a program that uses VSE/VSAM when changing from CKD to FBA devices. The differences between CKD and FBA VSE/VSAM are handled within VSE/VSAM. By using VSE/VSAM Access Method Services or the ISAM Interface Program (IIP), the new architecture will be transparent to the user.

For FBA devices, the size of a data space or data set can only be specified in terms of "number of records" or "number of blocks" but not in terms of cylinders or tracks.

For some Access Method Services commands, the device type of the device to be used has to be specified as a parameter (for example, ENVIRONMENT parameter). This specification is still necessary for FBA devices; however, the device type will be "FBA" for all FBA devices and no distinction need be made between different FBA devices. Furthermore, there will be a restriction preventing the opening of an index component by itself if the IMBED option has been specified and the sequence set and high-level index are on different device types, one of which is FBA and the other CKD.

However, the current VSAM restriction that data spaces and data set components cannot span devices of different types still applies. This is also true for different FBA device types.

Transparent Support

The following transitions will be transparent to the user:

- SAM CKD to SAM FBA (See SAM section for exceptions)
- VSAM CKD to VSE/VSAM FBA
- ISAM Interface Program (IIP) of VSAM to the IIP of VSE/VSAM FBA

Control Interval

The basic unit of data transfer between the host system and the control unit is a Control Interval (CI). For this reason it is necessary to understand the relationship between CI size and logical block size. A CI may consist of one or a multiple of FBA blocks.

The CI size may be different from file to file. The CI size will be stored in the Format-1 label of each file. CISIZE must be a multiple of 512 and, if greater than 8,192 (8K), a multiple of 2,048 (2K). The following rules apply to the selection of Control Interval size:

- Minimum CISIZE is one FBA block - 512 bytes.
- Maximum CISIZE is 64 FBA blocks - 32,768 bytes (32K).
- Logical block size must be smaller than or equal to the CISIZE minus the control information (up to 32,761 bytes).
- Maximum number of logical blocks per CI is 255 (when TYPEFLE=WORK).

Control Information

Each Control Interval contains the logical block(s) of data plus certain control information. Generally, the length of this control information is four bytes (for the CIDF) plus three bytes (for the RDF) for each logical block within the Control Interval. For example, a Control Interval that contains four variable-length blocks would have 16 bytes of control information (four bytes plus four times three bytes). If three or more consecutive equal-length blocks of data occur within a Control Interval, a special case of control information exists. Only two RDFs are required for three or more consecutive logical blocks of equal-length. For example, a Control Interval that contains four equal-length blocks would have 10 bytes of control information (four bytes plus two times three bytes).

Disk Space Considerations

For non-VSAM suballocated files, the extent information must be changed in the EXTENT statement because, with CKD, you must insert the starting track and number of tracks, and with FBA you must insert the starting block and number of FBA blocks. Therefore, you must know the relationship between these specifications. This relationship is not fixed; it is dependent upon the following:

- Source CKD device characteristics
- Logical block length
- CI length

Appendix E contains a chart and an example to assist you in determining the proper Control Interval (CI) size and the correct number of FBA blocks to allocate for the desired number of records.

Figure 3.03 and Figure 3.04 give examples of typical blocksizes on the IBM 3340 and 3330 as compared to the fixed block DASD. When comparing disk spaces on the 3340/3330 with that in FBA, take into account that in FBA, the number of blocks on a track is meaningless, because there is no "track capacity". Let us assume that there is a full spindle of 3330 or 3340 DASD and compare it to a full spindle of 3310 and 3370 DASD.

Logical Blocksize in Bytes	Number of Logical Blocks Accessible on: 3330 No Key	3340 No Key	3310 with Control Interval Size (bytes) 512	1024	2048	4096	16384	32768
80	468236	283968	756096	756096	787600	803352	803352	805321
200	299364	192096	252032	315040	315040	315040	318978	320947
500	153520	100224	126016	126016	126016	126016	126016	127985
1000	84436	58464	----	63008	63008	63008	63008	63008
2000	46056	25056	----	----	31504	31504	31504	31504
4000	23028	16704	----	----	----	15752	15752	15752
6000	15352	8352	----	----	----	----	7876	7876
8000	7676	8352	----	----	----	----	7876	7876
13000	7676	----	----	----	----	----	3938	3938
16000	----	----	----	----	----	----	3938	3938

- Figure 3.03: IBM 3310 Comparison to 3330/3340

Logical Blocksize in Bytes	Number of Logical Blocks Accessible on: 3330 No Key	3340 No Key	3370 with Control Interval Size (bytes) 512	1024	2048	4096	16384
80	468236	283968	3348000	3348000	3487500	3557250	3557148
200	299364	192096	1116000	1395000	1395000	1395000	1412397
500	153520	100224	558000	558000	558000	558000	558000
1000	84436	58464	-----	279000	279000	279000	279000
2000	46056	25056	-----	-----	139500	139500	139500
4000	23028	16704	-----	-----	-----	69750	69750
6000	15352	8352	-----	-----	-----	-----	34874
8000	7676	8352	-----	-----	-----	-----	34874
13000	7676	----	-----	-----	-----	-----	17437
16000	----	----	-----	-----	-----	-----	17437

- Figure 3.04: IBM 3370 Comparison to 3330/3340

Figure 3.05 makes comparisons between the various libraries of DOS/VSE on CKD DASD and 3310 DASD. The chart assumes a full spindle of library without directory.

Comparison of System Libraries of DOS/VSE			
Library	3340	3330	3310
Core Image	58464	84436	63008
Relocatable	141984	214928	189024
Source Statement	217152	337744	378048
Procedure	283968	468236	756096

- Figure 3.05: IBM 3310 and CKD DASD using DOS/VSE Libraries

Even though the 3310 has a smaller stated capacity than the 3340, the 3310 will contain a larger amount of library blocks in all libraries. The 3310 compares very favorably to the 3330, especially the Source Statement and Procedure Libraries.

CKD DASD capacities are given (assuming that full track blocking is used, although this is rarely the case) and the efficiency of the space utilized decreases dramatically as the the physical block size decreases. FBA DASD capacities are given in usable space and do not vary as much with a logical block size. For example, with a logical block of 500 bytes on the 3340, the usable capacity is only 50,112,000 bytes (or 71 percent of stated capacity) while, with the 3310, the usable capacity is 63,008,000 bytes (or 97 percent of stated capacity).

To utilize the disk space three things have to be considered:

- A large CISIZE will, in general, better utilize the disk space; however, there will be a greater use of the partition GETVIS space and real storage.

- If the problem program blocking factor is changed to one, you may then choose the optimum CISIZE.

  Note: If BLKSIZE is specified in the JCL DLBL statement, the CISIZE cannot also be specified in the DLBL statement.

- The number of FBA blocks per extent should be a multiple of the number of blocks per CI. For example, if the extent = 57 blocks and CISIZE = 5 blocks, the last two blocks of the extent will never be used.

Since you may specify BLKSIZE or CISIZE (but not both) in the DLBL statement, you can choose the optimum BLKSIZE/CISIZE while preparing your JCL, without recompiling your program.

This Page Intentionally Left Blank

This chapter deals with the various alternatives possible in converting to the 4300 Processors and FBA devices. Generally, each conversion will fall into one of the following categories:

- Conversion to native IBM 3310 or 3370
- Conversion to emulated 231x on the 3310 attached to the 4331
- Conversion to the 4300 Processors with 231x/33xx DASD attached
- Conversion to the 4300 Processors with a mixture of old DASD and new 3310 or 3370 DASDs

Native IBM 3310 and 3370

While ISAM files are not supported on the IBM 3310 and the 3370 in fixed block mode, the ISAM Interface Program (IIP) of VSE/VSAM allows a program coded for the ISAM access method to process a VSE/VSAM Key Sequenced Data Set (KSDS). No change is required to the program. The JCL DLBL statement is changed to indicate VSE/VSAM in place of ISAM access.

Native 3310 and 3370 DASDs are not supported by the Direct Access Method (DAM). Any existing programs written for this access method would need to be changed to utilize the 3310 and 3370 natively. DAM programs would have to be converted to VSE/VSAM, to Physical IOCS (PIOCS), or to 2311/14/19 access and use the 231x compatibility on 3310.

On the 3310 DASD, performance is optimized by utilizing native FBA mode.

231x Compatibility on the IBM 3310 Attached to the IBM 4331

The 4331 supports 231x compatibility on the 3310 DASD. Up to seven spindles of 2311 or two spindles of 2314/19 DASD may be emulated on a single 3310 spindle. One type of 2311/2314 compatibility may be used during a single IPL. The 231x compatibility feature on the 4331 allows for "subdisks" to be allocated on the 3310 DASD. This feature allows you to allocate only the required space for the emulated file, and not take up the entire space requirement for a 231x spindle merely to emulate a 10-cylinder file. See Chapter 8 for more information on DASD compatibility.

IBM 4300 Processors with IBM 231x/33xx DASD Attached

The 4331 allows the IBM 3340 to be attached to the DASD Adapter. In addition, the 4331 allows physical attachment of IBM 231x DASD via an appropriate control unit to the Block Multiplexor Channel. The 4341 allows IBM 231x or IBM 33xx DASD to be attached via an appropriate control unit to the Block Multiplexor Channel. This attachment provides for conversion to the 4300 Processors with no data or programming changes required.

IBM 4300 Processors with a Mixture of DASD

The 4300 Processors allow 231x and/or 33xx (231x and 3340 on the 4331) to be physically attached via an appropriate control unit along with the 3310 or 3370. This provides a vehicle for data conversion to the native 3310 or 3370.

System/360/370 to the IBM 4331 using 231x Compatibility

If 231x devices are not to be attached to the 4331, magnetic tapes are necessary for the conversion of data sets. The procedure is as follows:

- Dump 231x volumes to magnetic tapes by using the DOS utility Copy/Restore Disk to Tape, the OS standalone Dump/Restore utility, or your own backup programs.
- On the 4331, initialize the 3310 by running the program INTDK. This program may be run under control of DOS/VSE or in the standalone version.
- On the 4331, run the standalone initialization utility program "Format Emulated Extent" to initialize the 3310 blocks for the use of the compatibility feature. The Compatibility Feature ID, 2311 or 2314, the emulated home address, and the emulated record zero are written onto the disks. The rest of an emulated track is filled with a clear value (X'FF').
- On the 4331, first assign at least one 2311/2314 compatibility feature buffer, then load the 231x initialize disk program and initialize as many 231x packs as required for your installation. If full emulated 231x packs are to be emulated, the operating system (DOS, DOS/VS DOS/VSE, etc.) for 231x may be used. However, if "subdisks" are to be initialized, the standalone initialize disk program must be used.
- Mount the magnetic tape containing the dumped 231x volumes. Restore the 231x volumes onto the physical 3310. The appropriate 231x compatibility feature device address (for example, "290") must be assigned in the program.

Normal Job Processing

At IPL time, you have the option of choosing the first output device (3310 connected via DASD adapter), the number of buffers (one to eight) and the device type to be emulated (2311 or 2314). Selection of the output device is done by specifying the physical device address (in the program load display) of the first drive that the compatibility feature accesses.

You must specify at least one buffer if you want to use the compatibility feature, but you may specify up to eight buffers.

The number of buffers to be used depends on only two criteria: main storage and processing speed. Each 231x buffer assigned in addition to the first increases compatibility feature processing speed if the job is access-intensive or uses sequential files. For jobs that seldom access the files, the increase is not significant.

Each buffer reduces the physical main storage available for user programs by 4K bytes (for 2311) or 8K bytes (for 2314/19).

After buffer assignment normal DOS, DOS/VS, or DOS/VSE processing can begin.

Note: The emulated 231x device can be the IPL device.

This Page Intentionally Left Blank

The following access method options exist when converting to DOS/VSE and the new IBM 3310 and 3370 DASDs:

- SAM to SAM

  Any program currently using Logical IOCS Sequential Access Method (SAM) may continue to use SAM on the 3310 and the 3370. These programs do not need to be changed to specify FBA as the device type. Once the data has been moved to the 3310 or the 3370, programs using SAM may access the data sets. SAM programs that use user-error exits or that are sensitive to I/O synchronization must be reviewed.

  Programs that use SAM work files and the NOTE POINT macros should be reviewed because of the new information format used by these macros.

- VSAM to VSE/VSAM

  Any program currently using VSE/Virtual Storage Access Method (VSE/VSAM) may continue to use VSE/VSAM on the 3310 and 3370. These programs will require no change when data sets are moved to the FBA devices.

- ISAM to VSE/VSAM

  Programs that currently use the Indexed Sequential Access Method (ISAM) must be converted to native VSE/VSAM or use the VSE/VSAM ISAM Interface Program (IIP). Most programs that use ISAM may utilize the VSAM IIP without change. Once the file has been converted to VSE/VSAM using the Access Method Services utility provided by VSE/VSAM, the '// DLBL' statement is changed to indicate that the file is VSE/VSAM and not ISAM. The VSE/VSAM IIP will convert ISAM program requests to VSE/VSAM file access requests.

- DAM to Physical IOCS or VSE/VSAM or 231x Compatibility

  The Direct Access Method (DAM) does not support the 3310 and the 3370. Any program currently using DAM must be changed to VSE/VSAM or to physical IOCS using the correct CCW commands for the IBM 3310 or the 3370. See Appendix B for more information on PIOCS.

  VSE/VSAM offers several data arrangements that should meet the needs of programs using the Direct Access Method:

  - Key Sequenced Data Sets (KSDS) - Files that are accessed via unique keys may be converted to this organization. Files that are variable in length and accessed via relative record number may be converted to this VSE/VSAM organization using the relative record number as the key.

  - Relative Record Data Sets (RRDS) - Files that are fixed in length and accessed via relative record numbers may be converted to this VSE/VSAM organization.

  - Entry Sequences Data Sets (ESDS) - Files that are accessed sequentially but are sometimes accessed via DAM to extend the files may be converted to this VSE/VSAM organization.

  Another alternative is to convert your PIOCS or DAM program to 231x specifications (usually this involves changing a table of number of records per track) and to utilize the 231x Compatibility Feature on the 4331. This would allow the program to be converted to native FBA at a later time.

This Page Intentionally Left Blank

This chapter concentrates on the allocation of your files and the migration of the data to the FBA files. Data Migration to the 4300 Processors with FBA DASD can be via:

- Tape
- 3340 connected to DASD Adapter on the 4331 for System/3 read only
- 3340 connected to DASD Adapter on the 4331
- 33xx connected to the Block Multiplexor Channel on the 4341
- 231x connected to the Block Multiplexor Channel

The steps to be performed to migrate the data from Count-Key-Data to FBA are the same regardless of physical devices used.

The medium most commonly used is tape. The migration steps that should be performed are as follows:

- Generate a system with FBA support.
- Copy all your programs over to the new operating system.
- Allocate file space on the FBA devices.
- Migrate your files to FBA.
- Change JCL.
- Run.

Utilities Used for Data Migration

There are several utility programs that may be used to migrate your data. Some of these are:

- VSE/VSAM Access Method Services REPRO/EXPORT/IMPORT
- VSE/DITTO
- Sort/Merge
- User-written utilities
- IBM System/3 3340 Data Import Program Product (See Chapter 9 for more information on the use of this feature.)

VSE/VSAM Files

The steps that should be performed are:

- Perform LISTCAT ALL for all catalogs containing entries to be migrated
- REPRO or EXPORT the files to tape
- Define new FBA catalogs, clusters, paths, and alternate indexes
- REPRO or IMPORT the files from tape to the FBA devices

Before looking into the migration jobstreams it is necessary to understand how VSE/VSAM supports FBA devices.

Device Type Specification

For some VSE/VSAM Access Method Services commands, the device type of the device used must be specified as a parameter. This specification is still necessary for FBA devices; however, the device type is "FBA" for all FBA devices and no distinction need be made between different FBA device types.

The current VSAM restriction that data spaces and file components cannot span devices of different device types still applies; it also applies to different FBA device types. This restriction is necessary to allow optimization on the basis of actual device geometry (IMBED).

Addressing Scheme

Addressing will be in terms of a numerically contiguous addressing structure. Each storage device attached to the system will have a unique device address and data will be addressed by a linear progression from 0 to (n-1), where n is the number of blocks on a particular device.

Although an FBA device is addressed linearly, it still has the same physical characteristics as a CKD device. Consequently, the optimization features currently offered by VSE/VSAM are as important for FBA as they are for CKD devices. These optimization features are:

- The ability to put the sequence set record of a KSDS on the first track of a Control Area (IMBED option)
- The replication of index records on a track (REPLICATE option)

These features are provided for FBA devices as well as for CKD devices.

With FBA, the basic allocation unit of VSE/VSAM is a track; however, the specification of disk addresses is in terms of blocks and VSE/VSAM will round the number of blocks to a track boundary.

The FBA addressing scheme and its unit of data transfer will be visible to the VSE/VSAM user only with the definition of data spaces and files, and when disk addresses or extent sizes are returned (for example, LISTCAT).

Definition of Data Spaces Including the Definition of Catalogs

The location of the data space has to be specified via the EXTENT statement in terms of the relative block number.

The size of a data space has to be specified via the EXTENT statement in terms of the number of blocks and a VSE/VSAM Access Method Services parameter in terms of the number of blocks or records to be allocated.

Definition of Clusters, Alternate Indexes, and Catalogs

The size of a cluster, alternate index, or catalog, etc. must be specified via a VSE/VSAM Access Method Services parameter in terms of the number of blocks or records to be suballocated.

Since VSE/VSAM will still do space allocation in terms of tracks or cylinders, your specified FBA addresses and block numbers will be translated into CKD type information internally, which may include some rounding according to the algorithms (see Appendix C).

This is the only area in which knowing the true device characteristics will be helpful in giving you a slightly better space utilization; however, knowledge of the cylinder and track layout of the FBA device is not required.

A new allocation parameter, BLOCKS, has been introduced in the Access Method Services DEFINE command. The RECORDS parameter in the DEFINE command is still valid.

Note: TRACKS and CYLINDERS explicit specifications will not be accepted for FBA devices. A file defined in terms of TRACKS|CYLINDERS on CKD DASD and then EXPORTed|IMPORTed to FBA DASD should be IMPORTed into a file defined in terms of BLOCKS or RECORDS.

The new parameter,

```
BLOCKS (primary (secondary))
```

is added to the current grouping CYLINDERS|TRACKS|RECORDS for the following DEFINE command parameters:

```
CLUSTER
AIX
MASTERCATALOG
USERCATALOG
SPACE
DATA
INDEX
```

The abbreviation for BLOCKS is BLOCK or BLK.

Allocation Considerations for VSE/VSAM Files

For each FBA device, VSE/VSAM has a minimum allocation unit value (BLKS/MIN-CA) and a maximum control area size value (BLKS/MAX-CA). For CKD devices these are one track and one cylinder, respectively.

For 3310:

- VSE/VSAM allocation unit value . . . . . = 32 blocks.
- VSE/VSAM maximum control area size value = 352 blocks.

For 3370:

- VSE/VSAM allocation unit value . . . . . = 62 blocks.
- VSE/VSAM maximum control area size value = 744 blocks.

The allocation unit value represents the minimum space that can be allocated (primary or secondary) for index control areas and for NOIMBED data control areas (ESDS and RRDS are always effectively NOIMBED since they have no index). For IMBED data control areas (KSDS, catalog and alternate index option), the minimum space that can be allocated is two allocation units (the imbedded sequence set always occupies one allocation unit within the control area). Control areas can be any integral multiple of the allocation unit value up to the maximum control area size value.

The maximum control area size value represents the maximum space that can be allocated (primary or secondary) for index and data control areas. For IMBED data control areas (KSDS, catalog, and alternate index option), the imbedded sequence set always occupies one allocation unit within the control area. Maximum control area size is always the unit for primary and secondary allocation of Catalog Recovery Areas (CRA).

UNIQUE Files

UNIQUE files are supported in FBA in the same way as in CKD except that UNIQUE files start on an allocation unit boundary on FBA, rather than on a cylinder boundary as required for CKD devices.

If the starting block, as specified on the EXTENT statement, does not coincide with an allocation unit boundary, it will be rounded to the next higher allocation unit boundary. If the ending block, as determined from the EXTENT statement, does not coincide with an allocation unit boundary, it will be rounded to the previous allocation unit boundary.

The control area size of a UNIQUE file in FBA is calculated the same way as for a non-UNIQUE cluster. The size of the data space is adjusted accordingly.

Data Migration

Start the data migration with a LISTCAT of all catalogs. Investigate the output listings to determine the entries that should be:
(1) deleted, (2) defined in FBA but the data should not be migrated, and (3) defined in FBA and the data should be migrated to FBA.

Figure 6.01 can be used to list the master catalog entries.

```
// JOB LIST MASTER CAT
// EXEC IDCAMS,SIZE=AUTO
  LISTCAT-
         CATALOG (AMASTCAT)
/*
/&
```

• Figure 6.01: LISTCAT Example

Figure 6.02 can be used to list the user catalog entries.

```
// JOB LIST USER CAT
// ASSGN SYS005,X'162'
// DLBL IJSYSUC,'YOUR.USER.CAT',,VSAM
// EXTENT SYS005,USER01
// EXEC IDCAMS,SIZE=AUTO
  LISTCAT-
       CATALOG (YOUR.USER.CAT)
/*
/&
```

• Figure 6.02: User Catalog LISTCAT Example

The next step will be the preparation of the REPRO or EXPORT jobstreams.

The recorded data in the VSE/VSAM catalog is expressed in number of cylinders or tracks even if your DEFINE was in number of records. For that reason it is necessary to do all DEFINEs before moving the data to the newly defined FBA files with REPRO or IMPORT and it is not possible to REPRO a catalog to a different device type. Each file and alternate index has to be REPROed or EXPORTed to tape.

Figure 6.03 shows how a VSAM file can be written to tape by using REPRO.

```
// JOB REPRO CLUSTER
// ASSGN SYS005,X'380'
// TLBL TAPEFIL,'PORTABLE.DATFIL1',,TAPE01
// ASSGN SYS006,X'162'
// DLBL IJSYSUC,'YOUR.USER.CAT',,VSAM
// EXTENT SYS006,USER02
// DLBL DATFIL1,'EXAMPLE.KSDS',,VSAM
// EXTENT SYS006,USER02
// EXEC IDCAMS,SIZE=AUTO
  REPRO INFILE (DATFIL1)-
        OUTFILE (TAPEFIL-
                 ENV (PDEV (2400)-
                 RECFM (VARBLK)-
                 BLKSZ (5164)-
                 RECSZ (516) ))
/*
/&
```

• Figure 6.03: REPRO Example

Figure 6.04 shows how a VSAM file can be written to tape by using EXPORT.

```
// JOB EXPORT
// ASSGN SYS006,X'161'
// DLBL IJSYSUC,'YOUR.USER.CAT',,VSAM
// EXTENT SYS006,USER02
// ASSGN SYS005,X'280'
// TLBL RECEIVE,'PORTABLE',,TAPE01
// ASSGN SYS007,X'163'
// DLBL SOURCE,'EXAMPLE.KSDS1',,VSAM
// EXTENT SYS007,USER04
// EXEC IDCAMS,SIZE=AUTO
  EXPORT EXAMPLE.KSDS1-
         INFILE (SOURCE)-
         OUTFILE (RECEIVE-
                  ENV (BLKSZ (6000)-
                  PDEV (2400) ))-
         TEMPORARY
/*
/&
```

• Figure 6.04: EXPORT Example

If your catalog is recoverable you can move all entries and files in one step. In Figure 6.05 all of the files are wholly contained in the volume.

```
// JOB EXPORTRA FOR ONE VOL OWNED BY MASTER
// ASSGN SYS005,X'280'
// TLBL VOLOUT,'OUT.FILE',,TAPE01
// ASSGN SYS012,X'161'
// DLBL VLV0,,,VSAM
// EXTENT SYS012,USER01
// EXEC IDCAMS,SIZE=AUTO
  EXPORTRA OUTFILE (VOLOUT-
                    ENV (PDEV (2400) )-
           CRA ((VLV0 ALL))
/*
/&
```

• Figure 6.05: EXPORTRA Example

Defining a Catalog

The following step is the preparation of the jobstream to define a master catalog, user catalog, data space, clusters, paths and alternate indexes.

In this and all following examples no DLBL statement is shown for the master catalog because it is assumed that the following statements have been placed in the system standard label area.

```
// DLBL IJSYSCT,'AMASTCAT',,VSAM
// EXTENT SYSCAT,USER01,1,0,352,5280
```

Figure 6.06 shows the definition of a master catalog.

```
// JOB DEF MASTCAT
// EXEC IDCAMS,SIZE=AUTO
  DEFINE MCAT-
         (NAME (AMASTCAT)-
          TO (99365)-
          FILE (IJSYSCT)-
          BLOCKS (5280)-
          VOL (USER01) )
/*
/&
```

• Figure 6.06: Define Master Catalog Example

Figure 6.07 shows the definition of a user catalog.

```
// JOB DEFINE USER CATALOG
// ASSGN SYS005,FBA,VOL=USER02,SHR
// DLBL VOL2,'FBA.USER.CAT',,VSAM
// EXTENT SYS005,USER02,1,0,352,5280
// EXEC IDCAMS,SIZE=AUTO
  DEFINE UCAT-
         (NAME (FBA.USER.CAT)-
          TO (99365)-
          FILE (VOL2)-
          BLOCKS (5280)-
          VOL (USER02) )
/*
/&
```

• Figure 6.07: Define User Catalog Example

Note: Considerations are the same for master and user catalogs.

You must define a VSE/VSAM data space from which the catalog (and CRA) is suballocated. This is done on CKD via the MASTERCATALOG|USERCATALOG subparameters: CYLINDERS|TRACKS|RECORDS. The quantity (primary) must correspond to the number of tracks allocated on the JCL EXTENT statement pointed to by the FILE subparameter. FBA requires the same process except that the subparameters BLOCKS|RECORDS must be used because CYLINDERS|TRACKS are not accepted. Therefore, you must convert a CYLINDERS|TRACKS quantity into a BLOCK or RECORDS quantity (no conversion, if you use RECORDS). This can be done by converting the quantity into number of bytes (take PHYREC-SIZE times PHYRECS/TRACK times TRACKS quantities from LISTCAT) and dividing bytes by 512 to arrive at a BLOCK value. Adjust accordingly if you want more or less space allocated. DOS/VSE does not support secondary allocation of data spaces, so this value need not be converted and can be omitted. Then you must specify, in your corresponding EXTENT statement, a beginning block number and number of blocks. The number of blocks will be the same value that you used with the BLOCKS subparameter. If you used the RECORDS subparameter, multiply this value by RECORDSIZE (average), divide by 512, round to the next integer, and use this value for EXTENT statement number of blocks. The beginning block number depends on where you wish the data space to be on the volume. You can use the LVTOC system utility to determine what space is available on your volume. Note that your catalog will be located (suballocated) at the beginning of your defined data space and the CRA will immediately follow it (for recoverable catalog); the CRA size will always be equal to the maximum control area size for the particular FBA device type. You may wish to change other subparameters of MCAT|UCAT (for example, change volume serial number), but there are no special FBA considerations.

The actual space to be suballocated for your catalog must be specified by using the CYLINDERS|TRACKS|RECORDS (CKD) and BLOCKS|RECORDS (FBA) subparameters OF DATA and INDEX. You should not try to convert directly from a CKD catalog size definition to an FBA definition. Instead, calculate the desired values by using the appropriate VSE/VSAM documentation.

Note: To avoid overly small, inefficient control area size, make your secondary allocation value at least as big as your desired control area size.

Defining a Data Space

The considerations are essentially the same for a data space definition as for defining a catalog. The differences are that a catalog is not suballocated from the space, and a CRA is suballocated only if this is the first space defined on a volume owned by a recoverable catalog.

Figure 6.08 shows the definition of a data space on the volume of the user catalog.

```
// JOB DATA SPACE
// ASSGN SYS007,FBA,VOL=USER02,SHR
// DLBL IJSYSUC,'FBA.USER.CAT',,VSAM
// EXTENT SYS007,USER02
// DLBL VOL4,,,VSAM
// EXTENT SYS007,USER02,1,0,5632,35200
// EXEC IDCAMS,SIZE=AUTO
  DEFINE SPACE-
         (FILE (VOL4)-
          VOL (USER02)-
          BLOCKS (35200) )
/*
/&
```

• Figure 6.08: Define Space Example

Defining a Non-Unique Cluster or Alternate Index

Since these files (or their components) are suballocated from VSE/VSAM data spaces, there are no FBA JCL considerations because the EXTENT parameters, beginning track/block, and number of tracks/blocks are not required and are ignored, if present.

You must convert the TRACKS and CYLINDERS subparameters to BLOCKS or RECORDS (if you use RECORDS, you have no conversion considerations). This conversion is the same as described on page 28 for Defining a Catalog.

Figure 6.09 shows the definition of a cluster.

```
// JOB DEFINE CLUSTER
// ASSGN SYS007,FBA,VOL=USER02,SHR
// DLBL IJSYSUC,'FBA.USER.CAT',,VSAM
// EXTENT SYS007,USER02
// EXEC IDCAMS,SIZE=AUTO
  DEFINE CLUSTER-
         (NAME(EXAMPLE.KSDS)-
          IXD-
          CISZ (1024)-
          RECSZ (100 250)-
          KEYS (12 4)-
          RECORDS (30000 5000)-
          FREESPACE (15 20)-
          VOL (USER02))-
         DATA-
          (NAME (EXAMPLE.KSDS.DATA) )-
         INDEX-
          (NAME (EXAMPLE.KSDS.INDEX) )
/*
/&
```

• Figure 6.09: Define Cluster Example

Defining a Unique Cluster or Alternate Index

Note: If a cluster or AIX contains a unique and a non-unique component, conversion considerations for the non-unique components are as previously described.

For each unique component (i.e., data and, if present, index) you must consider the conversion of EXTENT statement parameters and of the TRACKS and CYLINDERS subparameters. Both conversions are required because a unique component occupies its own VSE/VSAM data space. Thus, for each unique component the conversion requirements are the same as in the section, Defining a Data Space. There are no CRA considerations (after the first space or UNIQUE Cluster has been defined on the volume). If the component is to be on more than one volume, you must convert the EXTENT statement for each volume.

Figure 6.10 defines an alternate index over a previously loaded VSE/VSAM key-sequenced base cluster, defines a path over the alternate index to provide a way of processing the base through the alternate index, and builds the alternate index. The alternate index, path, and base cluster must all be defined in the same catalog, in this case, the user catalog.

```
// JOB ALT INDEX, PATH, BUILD INDEX
// ASSGN SYS010,FBA,VOL=USER02,SHR
// DLBL IJSYSUC,'FBA.USER.CAT',,VSAM
// EXTENT SYS010,USER02
// DLBL BASEDD,'EXAMPLE.KSDS1',,VSAM
// EXTENT SYS010,USER02
// DLBL AIXDD,'EXAMPLE.AIX',,VSAM
// EXTENT SYS010,USER02
// DLBL IDCUT1,'SORT.WORK.ONE',,VSAM
// EXTENT SYS010,USER02
// DLBL IDCUT2,'SORT.WORK.TWO',,VSAM
// EXTENT SYS010,USER02
// DLBL VOL4,'EXAMPLE.PATH',,VSAM
// EXTENT SYS010,USER02
// EXEC IDCAMS,SIZE=AUTO
  DEFINE AIX-
         (NAME (EXAMPLE.AIX)-
          RELATE (EXAMPLE.KSDS)-
          KEYS (3 0)
          RECORDSIZE (40 50)-
          VOL (USER02)-
          BLOCKS (1408 352)-
          FILE (AIXDD)-
          NONUNIQUEKEY-
          UPGRADE)
  DEFINE PATH-
         (NAME (EXAMPLE.PATH)-
          PATHENTRY (EXAMPLE.AIX)-
          FILE (VOL4) )
  BLDINDEX-
          INFILE (BASEDD)-
          OUTFILE (AIXDD)
/*
/&
```

• Figure 6.10: Define Alternate Index, Path Example

The last step is the migration of the data from the tape to the VSE/VSAM files.

Figure 6.11 shows how a tape file can be written to a VSE/VSAM file by using the REPRO command.

```
// JOB REPRO CLUSTER
// ASSGN SYS005,FBA,VOL=USER02,SHR
// DLBL IJSYSUC,'FBA.USER.CAT',,VSAM
// EXTENT SYS005,USER02
// ASSGN SYS004,280
// TLBL TAPEFIL,'PORTABLE.DATFIL1',,TAPE01
// DLBL DATFILE,'EXAMPLE.ESDS',,VSAM
// EXTENT SYS005,USER02
// EXEC IDCAMS,SIZE=AUTO
  REPRO INFILE (TAPEFIL-
                ENV (PDEV (2400)-
                     RECFM (VARBLK)-
                     BLKSZ (5164)-
                     RECSZ(516) ))-
        OUTFILE (DATFILE)
/*
/&
```

• Figure 6.11: REPRO Cluster Example

Figure 6.12 shows how a tape file previously EXPORTed can be written to a VSE/VSAM file by using IMPORT. The name of the cluster is changed by means of the NEWNAME subparameter.

```
// JOB IMPORT
// ASSGN SYS004,280
// MTC REW,SYS004
// TLBL SOURCE,'PORTABLE',,TAPE01
// ASSGN SYS007,FBA,VOL=USER02,SHR
// DLBL IJSYSUC,'FBA.USER.CAT',,VSAM
// EXTENT SYS007,USER02
// DLBL RECDATA,'EXAMNEW.KSDS1',,VSAM
// EXTENT SYS007,USER02
// EXEC IDCAMS,SIZE=AUTO
  IMPORT INFILE (SOURCE-
                 ENV (PDEV (2400)-
                      BLKSZ(6000)-
                      RECSZ(479) ))-
         OUTFILE (RECDATA)-
         OBJECTS ((EXAMPLE.KSDS1 -
                   VOL (USER02)-
                   NEWNAME (EXAMNEW.KSDS1) ))
/*
/&
```

• Figure 6.12: IMPORT Example

Figure 6.13 shows the IMPORTRA of the files previously EXPORTRAed.

```
// JOB IMPORTRA
// ASSGN SYS004,280
// ASSGN SYS007,FBA,VOL=USER02,SHR
// DLBL IJSYSUC,'FBA.USER.CAT',,VSAM
// EXTENT SYS007,USER02
// TLBL TAPEFIL,'IN.FILE'.,TAPE01
// DLBL VSAMIN,'DUMMY.NAME',,VSAM
// EXTENT SYS007,USER02
// EXEC IDCAMS,SIZE=AUTO
  IMPORTRA INFILE (TAPEFIL-
                   ENV (PDEV (2400))-
           OUTFILE (VSAMIN)
/*
/&
```

• Figure 6.13: IMPORTRA Example

The jobstreams in Figure 6.14 shows an example of the migration of a SAM file to a SAM FBA file.

```
// JOB REPRO SAM FILE TO TAPE
// ASSGN SYS005,X'280'
// ASSGN SYS006,X'163'
// TLBL TAPEFIL,'PORTABLE.FILE',,TAPE01
// DLBL DATFIL,'DATA.FILE'
// EXTENT SYS006,USER06
// EXEC IDCAMS,SIZE=AUTO
  REPRO INFILE (DATFIL-
                ENV (BLKSZ (4000)-
                     PDEV (3340) -
                     RECFM (FB)-
                     RECSZ (100) )-
        OUTFILE (TAPEFIL-
                 ENV (BLKSZ (4000)-
                      PDEV (2400)-
                      RECFM (FB)-
                      RECSZ (100) )
/*
/&

// JOB REPRO TAPE FILE TO SAM FBA
// ASSGN SYS004,280
// TLBL TAPE,'PORTABLE.FILE',,TAPE01
// ASSGN SYS005,FBA,VOL=USER05,SHR
// DLBL DATFIL,'DATA.FILE1',,SD,CISIZE=4096
// EXTENT SYS005,USER05,1,0,352,38400
// EXEC IDCAMS,SIZE=AUTO
  REPRO INFILE (TAPE-
                ENV (BLKSZ (4000)-
                     PDEV (2400)-
                     RECFM (FB)-
                     RECSZ (100) )-
         OUTFILE (DATFIL-
                  ENV (BLKSZ (4000)-
                       PDEV (FBA)-
                       RECFM (FB)-
                       RECSZ (100) )
/*
/&
```

• Figure 6.14: REPRO SAM File Example

ISAM Files

ISAM files are not directly supported on FBA devices; however, programs with ISAM DTFs may run without recompilation when the ISAM Interface Program (IIP) of VSE/VSAM is used. You must ensure that your existing ISAM programs comply with the restrictions that are documented in the Data Management Guide.

To convert an ISAM file to a VSE/VSAM file, you must use the Access Method Services DEFINE command to define a key-sequenced VSE/VSAM file. The use of the DEFINE command is fully explained in the appropriate VSE/VSAM documentation (and an example is given earlier in this chapter). After you have defined your VSE/VSAM file, you must load the VSE/VSAM file by copying the ISAM file into it. You may use your ISAM load program by way of the IIP, or you may use the Access Method Services REPRO command. If you have records marked for deletion in the

ISAM file and do not want them copied into the VSE/VSAM file, you should use your ISAM load program because the REPRO command will copy all records from the ISAM file, including those marked for deletion.

Changing ISAM Job Control Statements

All Job Control Statements for ISAM must be replaced by VSE/VSAM Job Control Statements. Figure 6.15 shows an example of VSE/VSAM JCL statements used with an ISAM program.

```
// JOB LOAD VSE/VSAM FILE WITH ISAM LOAD PROGRAM
// ASSGN SYS001,161
// DLBL ISN,'MSTRFILE',,VSAM
// EXTENT SYS001,USER01
// EXEC ISAMLOAD,SIZE=AUTO
        :
(Program containing DTFIS with filename ISN)
        :
/*
/&
```

• Figure 6.15: ISAM using IIP Example

One DLBL statement is required for the file and one EXTENT statement is required for each volume of the file.

Figure 6.16 shows an example of a REPRO jobstream to convert an ISAM file to a VSE/VSAM file.

```
// JOB REPRO ISAM FILE TO TAPE
// ASSGN SYS005,X'280'
// TLBL TAPEFIL,'ISAMFILE',,TAPE01
// ASSGN SYS004,X'162'
// DLBL ISAMFIL,'ISAMFILE',,ISE
// EXTENT SYS004,USERXX,4,1,360,12
// EXTENT SYS004,USERXX,1,2,372,240
// EXTENT SYS004,USERXX,2,3,612,24
// EXEC IDCAMS,SIZE=AUTO
  REPRO INFILE (ISAMFIL-
                ENV (BLKSZ (500)-
                     HDEV (3340)-
                     PDEV (3340)-
                     RECFM (FB)-
                     RECSZ (100) ))-
        OUTFIL (TAPEFIL-
                ENV (BLKSZ (4000)-
                     PDEV (2400)-
                     RECFM (FB)-
                     RECSZ (100) ))
/*
/&

// JOB REPRO ISAM TAPE FILE TO VSE/VSAM FBA
// ASSGN SYS007,FBA,VOL=USER02,SHR
// DLBL IJSYSUC,'FBA.USER.CAT',,VSAM
// EXTENT SYS007,USER02
// ASSGN SYS004,280
// TLBL TAPEFIL,'ISAMFILE',,TAPE01
// DLBL ISAM,'EXAMPLE.KSDS1',,VSAM
// EXTENT SYS007,USER02
// EXEC IDCAMS,SIZE=AUTO
  REPRO INFILE (TAPEFIL-
                ENV (PDEV (2400)-
                     RECFM (FB)-
                     BLKSZ (4000)-
                     RECSZ (100) ))-
        OUTFILE (ISAM)
/*
/&
```

• Figure 6.16: REPRO ISAM File Example

VSE/DITTO is a Program Product utility program which can be used in the conversion of data files.

VSE/VSAM Files

In the following jobstreams, the VSE/VSAM catalog and files as defined above are used. Figure 6.17 shows the migration of a VSE/VSAM file to tape and back to VSE/VSAM.

```
// JOB DITTO VSE/VSAM FILE TO TAPE
// UPSI 1
// DLBL VDSKIN,'EXAMPLE.KSDS1',,VSAM
// EXTENT SYS002,USER01
// ASSGN SYS002,X'161'
// ASSGN SYS003,X'280'
// EXEC DITTO,SIZE=100K
$$DITTO VTP INPUT=SYS002,                                          *
$$DITTO     OUTPUT=SYS003,                                         *
$$DITTO     START=C,                                               *
$$DITTO     POSITION=0,                                            *
$$DITTO     NLRECS=99999,                                          *
$$DITTO     BLKFACTOR=20                                           *
$$DITTO EOJ
/*
/&

// JOB DITTO TAPE TO VSE/VSAM FBA FILE
// ASSGN SYS003,280
// ASSGN SYS002,FBA,VOL=USER02,SHR
// UPSI 1
// DLBL VDSKOUT,'EXAMPLE.KSDS1,,VSAM
// EXTENT SYS002,USER02
// DLBL IJSYSUC,'FBA.USER.CAT',,VSAM
// EXTENT SYS002,USER02
// EXEC DITTO,SIZE=100K
$$DITTO TVS INPUT=SYS003,                                          *
$$DITTO     OUTPUT=SYS002,                                         *
$$DITTO     RECSIZE=100,                                           *
$$DITTO     NLRECS=99999
$$DITTO EOJ
/*
/&
```

• Figure 6.17: VSE/DITTO VSE/VSAM File Example

Figure 6.18 shows the migration of a SAM file to tape and from tape to FBA DASD.

```
// JOB DITTO SAM FILE TO TAPE
// UPSI 1
// ASSGN SYS002,X'280'
// ASSGN SYS003,X'163'
// DLBL SDSKIN,'DATA.FILE1'
// EXTENT SYS003,USER06
// EXEC DITTO,SIZE=100K
$$DITTO STP INPUT=SYS003,                                              *
$$DITTO     OUTPUT=SYS002,                                             *
$$DITTO     RECSIZE=100,                                               *
$$DITTO     BLKFACTOR=40,                                              *
$$DITTO     NLRECS=99999
$$DITTO EOJ
/*
/&

// JOB DITTO TAPE TO SAM FILE FBA
// UPSI 1
// ASSGN SYS002,280
// ASSGN SYS003,FBA,VOL=USER05,SHR
// DLBL SDSKOUT,'DATA.FILE1',,SD,CISIZE=4096
// EXTENT SYS003,USER05,1,0,352,38400
// EXEC DITTO,SIZE=100K
$$DITTO TSQ INPUT=SYS002,                                              *
$$DITTO     OUTPUT=SYS003,                                             *
$$DITTO     RECSIZE=100,                                               *
$$DITTO     BLKFACTOR=1,                                               *
$$DITTO     NLRECS=99999
$$DITTO EOJ
/*
/&
```

• Figure 6.18: VSE/DITTO SAM File Example

ISAM Files

Figure 6.19 shows the migration of an ISAM file to tape and from tape to a VSE/VSAM KSDS file.

```
// JOB DITTO ISAM FILE TO TAPE
// UPSI 1
// ASSGN SYS002,X'162'
// ASSGN SYS003,X'280'
// DLBL IDSKIN,'ISAMFILE',,ISE
// EXTENT SYS002,USERXX,4,1,360,12
// EXTENT SYS002,USERXX,1,2,372,240
// EXTENT SYS002,USERXX,3,3,612,24
// EXEC DITTO,SIZE=100K
$$DITTO ITP INPUT=SYS002,                                          *
$$DITTO     OUTPUT=SYS003,                                         *
$$DITTO     BLKFACTOR=40,                                          *
$$DITTO     START=C,                                               *
$$DITTO     POSITION=0,                                            *
$$DITTO     NLRECS=99999
$$DITTO EOJ
/*
/&

// JOB DITTO TAPE TO VSE/VSAM FILE FBA
// UPSI 1
// ASSGN SYS002,280
// ASSGN SYS003,FBA,VOL=USER02,SHR
// DLBL IJSYSUC,'FBA.USER.CAT',,VSAM
// EXTENT SYS003,USER02
// DLBL VDSKOUT,'EXAMPLE.KSDS1',,VSAM
// EXTENT SYS003,USER02
// EXEC DITTO,SIZE=100K
$$DITTO TVS INPUT=SYS002,                                          *
$$DITTO     OUTPUT=SYS003,                                         *
$$DITTO     RECSIZE=100,                                           *
$$DITTO     NLRECS=99999
$$DITTO EOJ
/*
/&
```

- Figure 6.19: VSE/DITTO ISAM File to VSE/VSAM File Example

Data Migration with Sort/Merge

Sort/Merge can be used if there is an appropriate field within the data records that falls into the proper sequence to satisfy the Sort/Merge requirement for a sort field.

With Sort/Merge you can migrate:

- SAM files
- VSE/VSAM - ESDS files
  - KSDS files

In the following SORT/VS examples, we assume that the necessary Access Method Services DEFINEs have been done. For documentation of the DEFINE Command see Access Method Services Users Guide, or an example shown earlier in this chapter. It is also assumed that the required DLBL and EXTENT statements for the sort work files are in the label cylinder/area.

```
// DLBL SORTWK1,,0,SD
// EXTENT SYS003,SORTWK,,,120,1200
```

Figure 6.20 shows the migration of a SAM file to tape and from tape to FBA DASD:

```
// JOB SORT SAM FILE TO TAPE
// ASSGN SYS001,X'280'
// TLBL SORTOUT,'SAMFILE',,TAPE01
// ASSGN SYS002,X'163'
// DLBL SORTIN1,'DATA.FILE1'
// EXTENT SYS002,USER06
// ASSGN SYS003,X'162'
// EXEC SORT,SIZE=160K
  SORT FIELDS=(5,6,A),FORMAT=BI,WORK=1
  RECORD LENGTH=100,TYPE=F
  INPFIL BLKSIZE=4000
  OUTFIL BLKSIZE=4000,CLOSE=UNLD
  END
/*
/&

// JOB SORT TAPE TO SAM FILE FBA
// ASSGN SYS001,FBA,VOL=USER05,SHR
// ASSGN SYS002,280
// ASSGN SYS003,FBA,VOL=SORTWK,SHR
// TLBL SORTIN1,'SAMFILE',,TAPE01
// DLBL SORTOUT,'DATA.FILE1',,SD,CISIZE=4096
// EXTENT SYS001,USER05,1,0,352,38400
// EXEC SORT,SIZE=160K
  SORT FIELDS=(5,6,A),FORMAT=BI,WORK=1
  RECORD LENGTH=100,TYPE=F
  INPFIL BLKSIZE=4000,CLOSE=UNLD
  OUTFIL BLKSIZE=100
  END
/*
/&
```

- Figure 6.20: Sort/Merge SAM File Example

Figure 6.21 shows the migration of a KSDS VSE/VSAM file to tape and from tape to VSE/VSAM KSDS.

```
// JOB SORT VSE/VSAM KSDS FILE TO TAPE
// ASSGN SYS001,X'280'
// ASSGN SYS002,X'163'
// ASSGN SYS003,X'162'
// TLBL SORTOUT,'VSAMFILE',,TAPE01
// DLBL SORTIN,'EXAMPLE.KSDS1',,VSAM
// EXTENT SYS002,USER01
// EXEC SORT,SIZE=160K
  SORT FIELDS=(5,6,A),FORMAT=BI,WORK=1
  RECORD TYPE=F,LENGTH=100
  INPFIL VSAM
  OUTFIL BLKSIZE=4000,CLOSE=UNLD
  END
/*
/&

// JOB SORT TAPE TO VSE/VSAM KSDS FILE
// ASSGN SYS002,FBA,VOL=USER02,SHR
// DLBL IJSYSUC,'FBA.USER.CAT',,VSAM
// EXTENT SYS002,USER02
// ASSGN SUS003,FBA,VOL=SORTWK,SHR
// TLBL SORTIN1,'VSAMFILE',,TAPE01
// DLBL SORTOUT,'EXAMPLE.KSDS1',,VSAM
// EXTENT SYS002,USER02
// EXEC SORT,SIZE=160K
  SORT FIELD=(5,6,A),FORMAT=BI,WORK=1
  RECORD TYPE=F,LENGTH=100
  INPFIL BLKSIZE=4000,CLOSE=UNLD
  OUTFIL KSDS
  END
/*
/&
```

• Figure 6.21: Sort/Merge VSE/VSAM File Example

Note: Each of the examples using Sort/Merge assumes that there is a field within each record in the file that can satisfy the requirement for a Sort/Merge sort field.

Data Migration with User-Written Utilities

There are two possibilties in migrating data with user-written utilities: writing one program for all files or writing one program for each file.

The first solution is the same solution as was discussed in using IBM utilities. Writing one program to migrate all files is not realistic, because it is much easier to use the existing IBM utilities.

The second solution may be a good one, especially with a simple program which uses a language like RPG-II. Write one program for the first file and modify this program for each of the other files.

If you are migrating DAM files, you should write your own program because there is no utility for DAM files.

Accessing DAM files in RPG-II is possible; see SC33-6031 RPG-II/VS Language Reference Manual for such an example.

Figure 6.22 shows a RPG-II program used to migrate a SAM CKD file to tape.

```
// JOB RPG COMPILE LINK AND GO SAMFILE TO TAPE
// OPTION LINK,LIST
// EXEC RPGII,SIZE=AUTO
01000H
02001FSAMCKDF IPE F4000 100             DISK40 SYS007S
03002FTAPEFIL O   F4000 100             TAPE   SYS008S
04001ISAMCKDF                                  1 100INPUTFLI
05001OTAPEFIL T       LI        INPUTF  100
/*
// LBLTYP TAPE
// EXEC LNKEDT
// ASSGN SYS008,X'280'
// TLBL TAPEFIL,'SAMFILE',,TAPE01
// ASSGN SYS007,X'162'
// DLBL SAMCKDF,'DATA.FILE1'
// EXTENT SYS007,USER06
// EXEC
/&
```

• Figure 6.22: RPG-II SAM CKD File Example

Figure 6.23 shows an Assembler program used to migrate a SAM CKD file to tape.

```
// JOB ASSM COMPILE LINK AND GO SAMFILE TO TAPE
// OPTION LINK,LIST
// EXEC ASSEMBLY,SIZE=AUTO
MIGPRC  CSECT                   ESTABLISH CSECT
        PRINT NOGEN             DON'T PRINT EXPANSION OF MACROS
        BALR  10,0              ESTABLISH BASE REGISTER
        USING *,10              INFORM ASSEMBLER OF BASE REG
        OPEN  SAMCKDF,TAPEFIL   OPEN FILES
B0101   GET   SAMCKDF           GET INPUT FILE
        PUT   TAPEFIL           PUT OUTPUT RECORD
        B     B0101             LOOP UNTIL FINISHED
EOJXX   CLOSE SAMCKDF,TAPEFIL   CLOSE FILES
        EOJ   ,                 END OF JOB
IOAREA1 DS    CL4000            IOAREA FOR INPUT
IOAREA2 DS    CL4000            IOAREA FOR OUTPUT
*
SAMCKDF DTFSD BLKSIZE=4000,EOFADDR=EOJXX,IOAREA1=IOAREA1,              *
              DEVADDR=SYS007,DEVICE=3340,IOREG=(9),                    *
              RECFORM=FIXBLK,RECSIZE=100,TYPEFILE=INPUT
TAPEFIL DTFMT BLKSIZE=4000,FILABL=STD,IOAREA1=IOAREA2,IOREG=(9),       *
              RECFM=FIXBLK,RECSIZE=100,TYPEFLE=OUTPUT,DEVADDR=SYS008
        END
/*
// LBLTYP TAPE
// EXEC LNKEDT
// ASSGN SYS008,X'280'
// TLBL TAPEFIL,'SAMFILE',,TAPE01
// ASSGN SYS007,X'162'
// DLBL SAMCKDF,'DATA.FILE1'
// EXTENT SYS007,USER06
// EXEC
/&
```

• Figure 6.23: Assembler SAM CKD File Example

Figure 6.24 and Figure 6.25 show an RPG-II program and an Assembler program to migrate a tape file to SAM FBA.

```
// JOB RPG COMPILE LINK AND GO TAPE TO SAM FBA FILE
// OPTION LINK,LIST
// EXEC RPGII,SIZE=AUTO
01000H
02001FTAPEFIL IPE F4000 100        TAPE SYS008S
03002FSAMFBAF O   F 100 100        FBA  SYS007S
04001ITAPEFIL                           1 100 INPUTFL1
05001OSAMFBAF        LI      INPUTF 100
/*
// EXEC LNKEDT
// ASSGN SYS008,280
// TLBL TAPEFIL,'SAMFILE',,TAPE01
// ASSGN SYS007,FBA,VOL=USER05,SHR
// DLBL SAMFBAF,'DATA.FILE1',,SD,CISIZE=4096
// EXTENT SYS007,USER05,1,0,352,38400
// EXEC
/&
```

• Figure 6.24: RPG-II SAM FBA File Example

```
// JOB ASSM COMPILE LINK AND GO TAPE TO SAM FBA FILE
// OPTION LINK,LIST
// EXEC ASSEMBLY,SIZE=AUTO
MIGRPRG START 0                ESTABLISH CSECT
        PRINT NOGEN            DON'T PRINT MACRO EXPANSION
        BALR  10,0             ESTABLISH BASE REGISTER
        USING *,10             INFORM ASSEMBLER OF BASE REG
        OPEN  SAMFBAF,TAPEFIL  OPEN FILES
B01011  GET   TAPEFIL          GET INPUT RECORD
        PUT   SAMFBAF          PUT OUTPUT RECORD
        B     B01011           LOOP UNTIL FINISHED
EOJXX   CLOSE SAMFBAF,TAPEFIL  CLOSE FILES
        EOJ   ,                END OF JOB
IOAREA1 DS    CL4000           IOAREA FOR INPUT
*
SAMFBAF DTFSD BLKSIZE=108,EOFADDR=EOJXX,DEVADDR=SYS008,DEVICE=FBA,     *
              IOREG=(9),RECFM=FIXUNB,RECSIZE=100,                       *
              TYPEFLE=OUTPUT
TAPEFIL DTFMT BLKSIZE=4000,FILABL=STD,IOAREA1=IOAREA1,IOREG=(9),       *
              RECFM=FIXBLK,RECSIZE=100,TYPEFLE=INPUT,DEVADDR=SYS007
        END
/*
// EXEC LNKEDT
// ASSGN SYS007,280
// TLBL TAPEFIL,'SAMFILE',,TAPE01
// ASSGN SYS008,FBA,VOL=USER05,SHR
// DLBL SAMFBAF,'DATA.FILE1',,SD,CISIZE=4096
// EXTENT SYS008,USER05,1,0,352,38400
// EXEC
/&
```

• Figure 6.25: Assembler SAM FBA File Example

Migration of Direct Access Method Files and Programs

Files accessed via the Direct Access Method (DAM) generally fall into one of the following categories:

• Records addressed directly via pointers (which are relative record

numbers or physical addresses). Programs accessing this file can be converted to VSE/VSAM without altering the program logic. If the records are fixed in length and addressed by relative record number, then the VSE/VSAM data organization Relative Record Data Set (RRDS) is probably appropriate. If the records are variable in length, the VSE/VSAM data organization Key Sequenced Data Set (KSDS) is appropriate. The relative record number or physical address may be the key for the file.

- Records addressed by unique key. Programs accessing this file can be changed to VSE/VSAM without altering the program logic. The VSE/VSAM data organization Key Sequenced Data Set (KSDS) is appropriate for the file.

- Records having non-unique keys. The location of a record is only approximately known. A starting track is determined by some randomizing routine and the track (or rest of the cylinder) is searched for the appropriate record. This file organization is not directly supported via VSE/VSAM. Programs accessing this file will require logic and file organization changes.

Some programs using DAM must be converted to Physical IOCS or completely redesigned. Information pertaining to Channel Command Words for the FBA DASD is contained in Appendix B.

Library Migration

Library migration from DOS/VS Release 34 to DOS/VSE can be no more difficult than any other normal release-to-release transition in which a change of devices is involved.

In this example, assume that only FBA devices are attached to the 4300 Processors; the procedure is as follows:

- Back up current system and private libraries to tape using BACKUP. System Libraries must be backed up as private libraries as DOS/VSE BACKUP will accept only private libraries in DOS/VS Release 34 format. DOS/VSE BACKUP will not allow DOS/VS Release 34 System Libraries to be restored as private libraries.

- Using PSERV, back up your defined procedures from the Procedure Library.

- Restore DOS/VSE to FBA, allowing room for modules to be merged.

- Restore backup DOS/VS Release 34 Private Libraries to FBA scratch volume.

- With CORGZ, merge the old libraries into the new libraries, using the COPYx NEW function. COPYSERV may also be used.

- With MAINT, catalog your defined procedures into the Procedure Library.

Program Migration without Recompilation

It is not necessary to compile or linkedit your program when you are migrating from DOS/VS Release 34 CKD to DOS/VSE FBA, with the following exceptions:

- Programs using DAM must be rewritten.
- Programs using ISAM must be converted to VSE/VSAM or use the ISAM Interface Program (IIP) of VSE/VSAM.
- Programs using user-written error recovery for SAM must be investigated for compatibility.
- Programs using DTFPH/EXCP must be rewritten.

At OPEN time the program will need GETVIS space from the partition GETVIS area for a DTF extension, a work area for the logic modules and SAM Service Routine (SSR), and an area large enough to read in one control interval of the size specified in the DLBL statement or determined by the access method.

Additionally, OPEN will GETVIS an area in the partition large enough to use for its own workspace.

The old IOCS modules linked with your program are no longer used and a pointer will be set at OPEN time to the FBA I/O module and SSR in the SVA (as is done for RPS). Partition GETVIS space must be provided via default GETVIS or SIZE= parameter on the EXEC statement.

Program Migration with Recompilation

It is not necessary to recompile each of your programs, but if you want to compile your programs for other reasons, the following should be changed:

- ISAM programs should be changed to VSE/VSAM access so that you do not have to use the ISAM Interface Program (IIP) of VSE/VSAM.

In Assembler you should change the following macros:

DTFSD (see Appendix A for more details)

- CONTROL=YES — Should be left out; otherwise, it will be ignored
- DEVICE=FBA — Should be specified
- ERREXT=YES — (Default) always gives an 8 byte parameter list.
- ERROPT=IGNORE
SKIP
NAME — If you are using (name), make sure that your module uses the 8 byte parameter list.
- WLRERR=name — Can be coded, but there could be differences
- MODNAME=name — Should be left out; otherwise, it will be ignored if DEVICE=FBA
- RDONLY=YES — Should be left out; otherwise, it will be ignored
- HOLD=YES — Can be coded
- CISIZE=nnnnn — Default is zero (may be overridden by JCL).
- WORKA=YES — Can be coded
- IOREG=(r) — Points directly into the CI if no IOAREA is used and thus is more efficient.
- IOAREA1=xxxxxxxx — If possible, it should be left out.
- IOAREA2=xxxxxxxx — If possible, it should be left out.
- PWRITE=YES — New parameter
- SEPASMB=YES — Can be used in the same way as for CKD
- VERIFY=YES — Can be used in the same way as for CKD

Note: If you specify DEVICE=FBA, no IOCS module will be linked with your program. This can amount to a considerable saving in Core Image Library space.

In RPG-II you should change the following items:

- File Description Specifications:

  cc 40 - 46 Device = FBA

In COBOL/VS the following changes should be made:

- The ASSIGN Clause should be changed to specify FBA1.
- The Reserve Clause should be left out, since FBA reserves a CI as a buffer. Specification of this clause will cause unnecessary data movement from CI to alternate areas.

In PL/I the following changes should be made:

- The Medium Option of the Environments Attribute should be changed to FBA.
- The Buffers specification should be left out.

Choice of BLKSIZE/CISIZE

On FBA, the blocking is done within a Control Interval. If the only consideration for choice of block size is DASD space utilization, then the best block size is the record size. However, a slightly longer path length is used because the logical block deblock routine must still be entered to determine that a new logical block must be requested.

For each BLKSIZE you can find an optimum CISIZE. See Appendix E for a chart to help you calculate the most efficient CISIZE, based on the desired logical block size.

LBLTYP Statements

The LBLTYP statement defines the amount of storage to be reserved in your partition for processing of tape and nonsequential disk file labels for non-VSAM files.

FBA is not supported by any nonsequential access method, so there is no need for a LBLTYP statement. Even if you have CKD or emulated devices there is no need for a LBLTYP statement in DOS/VSE. If present, it is ignored.

OPEN reads the first DLBL/EXTENT record for a DA or ISAM file type via the Symbolic Label Access routine and obtains the number of extents from the DLBL/EXTENT record. It will then GETVIS sufficient storage from the partition GETVIS area to accomodate the label information. The amount of virtual storage reserved is the same as is currently reserved via the LBLTYP statement. The storage will be FREEVISed by OPEN before returning to the problem program. Partition GETVIS space must be provided via default GETVIS or the SIZE= parameter on the EXEC statement.

Migrating Data Base/Data Communications

Data Bases

If you are currently using VANDL1 you must migrate to DL/I or DL/I Entry before you can use the FBA devices on the 4300 Processors.

Data base specifications are not completely device-independent. In the DBD there is a device parameter. For FBA devices this parameter must be changed and all DBDs regenerated (DEVICE=FBA).

A data base is a normal VSE/VSAM file and for each there is a VSE/VSAM define. These VSE/VSAM defines should be modified according to the rule mentioned above in the section, "Migrating VSE/VSAM Files". The physical data migration can occur via the normal data base reorganization/load utilities.

See DL/I and DL/I Entry documention to prepare the jobstreams.

Data Communications - CICS/VS

Overview

The following CICS/VS functions are potentially affected by changes to DASD support:

- File control
- Temporary storage
- Journaling
- Transient data
- Terminal control
- Dump control
- Auxiliary trace
- Auto-statistics
- Initialization/termination
- Keypointing

Functions Supported

File Control

No change is specified to the File Control Program for support of FBA devices. Files residing on CKD devices will continue to be supported by VSE/VSAM, ISAM, or DAM at your option. FBA resident files are managed only by VSE/VSAM, invoked natively.

Temporary Storage

Temporary Storage already uses VSE/VSAM exclusively when auxiliary storage is requested. FBA support is provided by VSE/VSAM and is transparent to CICS/VS.

Journaling

The Journal Control Table generation macro (DFHJCT) accepts FBA in its JDEVICE=operand.

Transient Data

Extrapartition destinations on FBA devices are supported by the Transient Data Program, using only the GET/PUT interface to SAM. No

modification is specified for extrapartition functions of the Transient Data Program. Generation of the Destination Control Table will accept FBA in the DEVICE= operand of the DFHDCT macro.

Intrapartition destinations will be optionally supported by the VSE/VSAM Control Interval Processing interface. This new option will apply ONLY to the intrapartition facility, and will allow you to specify VSE/VSAM in place of DAM. VSE/VSAM must be specified for FBA; it may be specified for CKD devices. The option will be coded externally through a new suboperand to the INTRA parameter of the DFHSG macro; for example:

```
DFHDG PROGRAM=TDP,INTRA=(TRANSINIT,VSAM)
or...  ,INTRA=TRANSINIT
or...  ,INTRA=(YES,VSAM)
or...  ,INTRA=YES
or...  ,INTRA=VSAM
```

Note: The VSAM suboperand will be available to users of both CICS/DOS/VS and CICS/OS/VS.

When VSE/VSAM is chosen, intrapartition, transient-data PUT requests are satisfied internally by the allocation of space from the current Control Interval for the specified destination and chaining-together of such Control Intervals. This is analogous to the current DAM approach, which allocates space from the current track for the specified destination, and chains tracks together to form queues. Transient data recovery, acquisition, disposition, and initialization functions support this option. When VSE/VSAM is selected for CICS/DOS/VS, FBA support will be transparent to CICS/VS.

Terminal Control

Generation of the Terminal Control Table allows FBA in the DEVICE=operand; for example:

```
DFHTCT TYPE=SDSCI,.....,DEVICE=FBA,...
```

Dump Control

FBA is valid in the DEVICE= operands of the following macro statements:

```
DFHSG PROGRAM=CSU,.....,DEVICE=FBA
DFHSG PROGRAM=DCP,.....,DEVICE=FBA
```

and also in the operator dialogue when executing the Dump Utility Program.

Auxiliary Trace

The offline Trace Utility Program (DFHTUP) only uses the SAM GET interface and will support FBA devices. In the following macro statements:

```
DFHSG PROGRAM=CSU,.....,DEVICE=yyyy
DFHSG PROGRAM=TRP,.....,TRDEV=zzzz
```

FBA is accepted for yyyy, zzzz. FBA is also allowed in the startup parameters when executing the Trace Utility Program.

Statistics

No modification of the online modules is specified, since statistics are written to an extrapartition transient-data destination (see Transient Data Section). Offline printing of the statistics uses only the SAM GET interface; therefore, the only modification specified is that FBA will be accepted in the operating dialogue when executing the Statistics Utility Program.

Initialization/Termination

Additions will be made to support the specified modifications to the Transient Data and Journal Control functions.

Keypointing

Since Keypointing is essentially a sequential operation, which only requests nonsequential access to manage the first record on the Restart file, all Keypoint requests are be changed to use the SAM workfile interface, (READ, WRITE, CHECK, NOTE, POINT-R/W/S macros). Manipulation of NOTEed values will be performed. In this way, use of the work file access method is compatible for FBA and CKD devices. The control record is managed by READ/WRITE sequences.

FBA will be valid in the DEVICE= operand when generating the Keypoint Program - (DFHSG PROGRAM=KPP).

Effects on Application Programmer

Where files are FBA-resident, only VSE/VSAM interfaces are supported. Application programs that use macro-level or command-level requests containing parameters peculiar to ISAM or DAM need to be modified. This is expected to be slight for ISAM but rather more complex for DAM, and depends further on any installation-defined changes to the database.

Effects on System Programmer/Computer Operator

VSAM option for Intra-TDP; see the Transient Data Section on page 49 for new DFHSG parameter with PROGRAM=TDP.

FBA option (where currently 2314, 3330, or 3340 is valid) exists in the following:

```
DFHSG   TYPE=INITIAL,.....,DEVICE=
DFHSG   PROGRAM=CSU,.....,DEVICE=
DFHSG   PROGRAM=CSU,.....,TRDEV=
DFHSG   PROGRAM=DCP,.....,DEVICE=
DFHSG   PROGRAM=TRP,.....,DEVICE=
DFHSG   PROGRAM=KPP,.....,DEVICE=
DFHTCT  TYPE=SDSCI,.....,DEVICE=
DFHDCT  TYPE=SDSCI,.....,DEVICE=
DFHJCT  TYPE=ENTRY,.....,JDEVICE=
```

Job Control statements for all DASD files on FBA devices are affected, depending on the requirements of SAM support of FBA. Changes caused by CICS/DOS/VS support of FBA are as follows:

- Intrapartition transient file, DFHNTRA, may be an optional VSE/VSAM file.
- The Restart file, DFHRSD, must be defined as a sequential file (SD instead of DA).
- Device specifications, when executing the Trace, Dump, and Statistics Utility Programs, will include the name FBA as a valid option where 2314, 3330, 3340 or 3350 are also valid in the file names (JCL), parameter statements, and operator dialogues used by these utilities.

Messages and Codes

MNOTEs confirm action taken in TDP generation, in the event of invalid/incomplete specification, in the form of:

'INTRA=xxxxxxx INVALID,INTRA=yyyyyyy ASSUMED'; for example:

'INTRA=(YES,BSAM) INVALID, INTRA=YES ASSUMED'

No new treatment will be given to errors during utility execution; the name "FBA" is not handled as an error.

Compatibilities/Incompatibilities

The name "FBA" will be considered valid in cases indicated in the foregoing sections; this will facilitate new specifications of the generation parameter. However, users of existing specifications (2314, 3330, 3340, 3350) may migrate from existing CKD to FBA devices without changing these sources, since DOS/VSE SAM will provide this degree of device-independence at OPEN. Because of access method constraints, definitions of block size or buffer size must be consistent with the associated device specification. Thus, 2314, for example, might be specified in connection with a block size of 7,294 (or fewer) bytes; if 7,294 were increased to exploit some FBA capability, then 2314 would be replaced by FBA in the statement that specifies it.

An Intrapartition Data Set created by the DAM implementation of Transient Data Program is not usable under the VSAM option of the Transient Data Program, nor is a VSAM Interpartition Data Set usable under the DAM implementation.

The Asynchronous Transaction Program (ATP) executes only with the DAM option of Transient Data; VSAM may not be specified when generating a TDP, if ATP is to operate with that particular Transient Data Program.

This Page Intentionally Left Blank

There is no general solution to the problem of backup and recovery. Each installation is unique in its configuration, applications, priorities, and machine room procedures. It is possible, however, to set down a general methodology for recovery planning.

It is most important that you develop your own plan, rather than acquire a completed plan from another source. There are several reasons for this. There are differences in priorities, policies, and practices from one installation to another which may be ideal in one environment and almost totally useless in another. It is unlikely that any plan could be developed that would be so comprehensive that it leads directly to each possible course of action in the event of a failure. More likely, there will be several points at which decisions must be made on the spot regarding the most appropriate course of action.

Probably the most important product of planning development is not the plan itself, but, rather, the insight gained into the particular use of DASD, the location of critical data, and the identification and evaluation of alternate courses of action to effect the recovery. It also becomes apparent where additional education is required for operations and systems personnel to enable them to make decisions during a recovery situation. The purpose of this chapter, then, is not to present a plan for recovering from DASD failures; it is to propose a methodology by which you can develop your own plan.

It should be said that fixed block architecture DASD does not materially change the nature of the backup and recovery planning. A plan developed for an IBM 3350 or IBM 3344 installation can easily be updated to accommodate the 3310 or the 3370. However, the introduction of fixed media devices in an installation that has had none previously, does change the nature of the backup and recovery process.

Effective Recovery Action

Before getting into the planning methodology, let us note the activities that are involved in recovery from a DASD failure. There are six steps to an effective recovery action:

- Recognize that a failure has occurred. The first indication that a DASD failure has occurred may be error messages coming from the operating system or some application program. An alert operator may see the message before it is lost, but it is more likely the first indication of trouble to come to the operator's attention will be that jobs of subsystems are ABENDing, or that phone messages arrive with complaints or pleas for help from frustrated users. In these cases, the operator's first task will be to determine that the problems he is experiencing are the result of an earlier DASD failure, rather than a routine programming error or some other unfortunate event. Problem determination can be the most difficult part of the whole recovery action.

- Assess the extent of damage to data. Once a failure has been recognized, it is important to determine which systems, subsystems, or applications are affected by the failure. This requires a knowledge of the data used by each system and application. It is also important to determine whether the data is irrevocably lost and must be recreated, or whether it is just unavailable for a period of time until the device is repaired. At this point, an estimate of the repair's duration is needed, for this will influence the recovery

action to be taken.

- Determine the type of recovery action to be taken. In general, there are several recovery alternatives available, including waiting for the device to be repaired, copying or moving critical data to another volume, restoring data from a backup copy to an available device, or reconstructing the data. A decision must be made to select one of these alternatives.

- Estimate the duration of data outage. Having determined the nature of the problem and the type of recovery action to be taken, you are in a position to estimate how long the data will be unavailable to users; there will most certainly be users and representatives of management who will want to know. If the plan is well conceived and well executed, a reasonable estimate of outage duration can be made.

- Initiate the recovery action. The type of recovery action determined should now be initiated.

- Control the situation. This is one of the most important parts of the whole plan and represents an area that is frequently not covered in recovery plans. It involves establishing a problem manager; a single individual who will be responsible for coordinating the efforts of the various groups who are working to solve the problem, and communicating with users and management to keep them appraised of current progress. It might be desirable to have several problem managers, one for each shift. It could be that access to the machine room should be limited to those working on the problem, in order to reduce the confusion.

Developing the Recovery Plan

When you have a clear idea of the course of action it must follow, you can proceed with developing the recovery plan itself. The steps involved are as follows:

- Prioritize application and system data. The first reaction to many is that, with the possible exception of sort work and compiler scratch space, all of the data on direct access devices is critical, and that a higher priority or value cannot be placed on one data set or class of data sets than on others. There is, however, a reasonably clear way to prioritize the data.

  At the top of the priority list is that data which affects multiple systems and without which none of the systems can operate. Recovery of shared data must be planned for.

  On the second major level is that data which affects only a single system and without which the CPU cannot be IPLed. This would include the VSE/VSAM Master Catalog and those system libraries which are unique to that system.

  A third major priority group includes that data which affects individual applications or groups of applications. Included here are data bases, user catalogs, and batch data sets.

  It may be possible to lose access to a data base or portions of a data base and still retain the ability to process some application systems as transaction types. It can be particularly difficult to try to assign priorities to different data bases or collections of batch data sets. While it is not absolutely necessary to do so, prioritizing this data can be helpful because the priority, or relative value, may influence the recovery action to be taken and the amount of resource to be applied to the recovery.

Finally, it is important to document the location of all critical data and to determine which critical data sets are on any volume, so that everyone who needs to know can immediately discover the location of each critical data set.

- Categorize DASD hardware failures. Each general type of DASD hardware failure should be categorized and its consequences noted. Head/Drive Assembly (HDA) or media failures will usually require reconstruction or restoration of data from some backup source. Drive failures will render the data termporarily unaccessible but do not threaten integrity of the data. Head-of-string failures affect access to multiple drives for the duration of the outage. Control unit and channel failures, like head-of-string failures, will affect accessibility to multiple devices and volumes.

- Categorize available hardware repair alternatives. The Customer Engineer (CE) has several alternative courses of action when responding to a drive failure. The normal course - and the one that is generally most satisfactory - is to proceed through the diagnostic and problem determination process to identify the single field replaceable unit (FRU) responsible for the failure and replace the failing component.

  In a critical situation the isolation process may be shortened by replacing multiple parts within the failing device and rescheduling maintenance time for the final isolation steps.

  The IBM 3370 device is serviceable at the same time that other 3370 devices on the same string are in use. If the defect in the failing unit is not the HDA and repair time is critical, the CE may move the HDA to another available drive to give access to its data.

- Categorize the recovery alternatives. A list should be compiled of all of the reasonable procedures available to regain access to the data. These might include copying or moving data from a drive that is experiencing intermittent errors to another volume, restoring data from a backup tape to an available spindle, and reconstructing data from a previous backup by applying logged transactions or manual changes to bring the data up to the level it was at the time of the failure.

- Perform a failure impact analysis. This is the heart of the recovery planning process. The analysis is performed by taking a combination of a possible type of hardware failure (for examples, device, head-of-string, or control unit) and a data set, and asking the following questions:

  - What are the symptoms or indications of the failure; that is, just what will the operator see when the selected failure occurs?

  - What is the impact of the failure? Will the entire system or just a single application or group of applications be affected? Will they be totally disabled or will they run with diminished capability?

  - What configuration alternatives are available that might lessen the impact of the outage? For example, string switches can greatly reduce the consequences of channel or control unit failures.

  - What data recovery alternatives are available? Is there a backup copy of the data that can be restored? Can the data be copied from some other source or must it be reconstructed? If it must be reconstructed, what tools are available to accomplish the reconstruction?

  - What is the best recovery technique to apply? Having identified in the preceding step the alternatives available, now select the one that is most effective.

- How long will the recovery action take? When the most appropriate recovery actions have been selected, it is possible to make a reasonable estimate of how long the recovery will take.

- What immediate actions will the system operator take? These may include running diagnostic programs, notifying individuals responsible for the custody of data, notifying IBM Field Engineering, etc.

This set of questions should be applied to each combination of data sets (or group of data sets having like attributes) and category of hardware failure. Clearly, this phase of the planning process involves several different disciplines and groups within the organization (operations, systems programming, and applications programming) and should also include participation by IBM Field Engineering, Systems Engineering, and marketing people. It is likely that, during this phase many questions will be raised that cannot be answered. In the case of some software systems and subsystems, it may not be apparent what error to expect and what recovery alternatives are offered by different program products.

As the different error situations are analyzed, it should become clear where additional backup procedures are required and the trade offs between the critical nature of the data and the resources required to produce additional backup copies, duplicate copies, or journaling.

- Produce operator instructions and documentation. This step involves collecting the information gathered in the inpact analysis and translating it into a concise set of instructions for operating personnel. These instructions will identify the problem determination steps to be taken, list persons to be notified, specify documentation to be gathered in order to identify the problem more accurately, and include any other actions that are appropriate. It may also be desirable at this time to produce guidelines for the system data base administrator and others who will be involved in the recovery.

- Define educational requirements. It is likely that, during the failure impact analysis, some educational requirements will be identified. For example, it may have been noted that operators need to be able to read and interpret the DASD error messages produced by the operating system or application subsystems. At this stage, it is appropriate to identify that education which will be required for operators, data administrators, systems programmers, and problem managers to enable them to swiftly and accurately execute their tasks in the recovery.

- Produce situation management guidelines and procedures. This is a very important step and is frequently omitted when recovery plans are created. This is the time to identify the individual, by name or position, who will be responsible for effecting the overall coordination of the recovery action. Procedures should be defined to ensure that those who should be aware of the problem are notified, whether the problem occurs during the first shift on a weekend or a weekday, or the third shift on a weekend. Wherever possible, the procedure should restrict access to the problem area to those who are needed to execute the recovery plan or to repair the equipment.

- Test the plan. The only way to determine the quality of your planning effort is to actually follow the plan to effect a recovery from hardware failure. It is possible to simply wait for a failure to occur, observe the success of the recovery attempt, and then evaluate the plan. However, if there are any shortcomings in the plan, it would be far better to discover them under the more controlled conditions of a simulated failure, rather than while in the midst of a genuine emergency. It is advisable to schedule several simulated failures involving various types of data to initially test the plan, then to arrange periodic "fire drills" to ensure that all parties involved in the recovery process are familiar with their rules and

responsibilities. This is particularly important after personnel changes have been made in key areas of responsibility.

To go through all of the steps outlined so far and produce a comprehensive recovery plan is not a particularly easy task. It will require a significant expenditure of time and effort on the part of a number of people. But the benefits derived will justify the time spent. Even though the plan itself may not always be executed exactly as written, the experience, knowledge, and education that is gained by going through this planning process will yield a much better awareness of the way data sets are used and the dependence upon certain data sets and devices. A better estimate of the duration and severity of outage resulting from a DASD device failure will certainly be gained. This, in turn, will minimize the element of surprise if a failure reoccurs.

This Page Intentionally Left Blank

The IBM 4331 Processor offers a disk compatibility feature to ease migration of files. This feature is called the 2311/2314/2319/3310 Direct Access Storage Compatibility Feature.

This feature is a standalone microprogram compatibility feature on the 4331 which performs the emulation of 231x data sets on the 3310 DASD device. The microprogram compatibility feature is loaded at IML time. The storage space it requires for buffers and tables is taken away from processor storage and is not accessible by user programs. The standalone program, Format Emulated Extent, is required to initialize the emulated 231x areas on the 3310.

Either 2311 or 2314/2319 compatibility (but not both) may be active. The device type to be emulated is selected by the operator at IPL time.

Function

The compatibility feature accepts the original 231x Channel Command Words (CCWs) and produces the same logical results as the 231x formerly used. The following access methods are supported:

- Sequential Access Method (SAM)
- Index Sequential Access Method (ISAM)
- Direct Access Method (DAM)
- Virtual Storage Access Method (VSAM)
- Physical IOCS (PIOCS)

Migration

The disk compatibility feature allows you to stay with the old DOS or DOS/VS releases during the migration period and change to DOS/VSE at your own pace because it is used to:

- Transfer existing System/360/370 DASD datasets to the 3310 DASD device without change
- Allow for coexistence of System/360/370 DASD files and FBA DASD files on the same volume

This mode of operation allows for the conversion of 231x files into FBA files on a gradual basis.

Method of Operation

The DASD Adapter of the 4331 is only capable of handling FBA files of the natively attached 3310 and the 3340 DASD. All emulated 231x I/O instructions or CCW commands, therefore, are routed to the Compatibility Feature microcode. In addition, at least one emulation buffer must be assigned to make the feature operational.

The size of the buffer is determined by the track length of the emulated disk rounded to a 2K boundary (4K for the 2311; 8K for the 2314).

Up to eight emulation buffers may be optionally assigned at IPL time by manual operation. The processor storage available to the user is reduced in accordance with the size and number of the assigned buffers.

The file mapping algorithm is chosen in such a way that Home Address, Record Zero, and the maximum capacity of the data records (without any gaps) of the emulated disk are mapped onto as many 512 byte data blocks of the native disk as are needed.

A fixed relationship between emulated disk addresses and native disk addresses is established.

IML/IPL Procedure

At IML time, the 231x compatibility feature is loaded into processor storage.

At IPL time, you may choose:

- Emulated device type (2311 or 2314)
- Address of native device where the emulated space starts
- Number of emulation buffers

Mapping of 231x Volumes on the IBM 3310

Mapping is the method by which 231x volumes are recorded on a 3310. The block organization of the 3310 is not changed in any way; instead, each 231x track is spread over several of its blocks.

Mapping of 2311 Volumes

- Each 2311 track (including home address, count, key, and data areas but excluding gaps) becomes eight 512 byte blocks on the 3310.
- 203 cylinders of the 2311 (at 10 tracks per cylinder) build one volume.
- Up to seven 2311 volumes may be mapped on one 3310 spindle.
- You may initialize subdisks if you do not need the total 2311 space. The size of a subdisk is specified by the number of emulated cylinders.

Mapping of 2314 Volumes

- Each 2314 track (including home address, count, key, and data areas but excluding gaps) becomes fifteen 512 byte blocks on the 3310.
- 203 cylinders of the 2314 (at 20 tracks per cylinder) build one volume.

- Up to two 2314 volumes may be mapped on one 3310 spindle.
- You may initialize subdisks if you do not need the total 2314 space. The size of a subdisk is specified by the number of emulated cylinders.

Association of Emulated Device Address with Native Device Address

To identify each 2311 or 2314 volume, an emulated device is associated with each group of 3310 blocks that holds the emulated volume. The 3310 may be attached via the DASD Adapter with channel address 2 or 3. A maximum of two 3310 strings at a time may be used for the compatibility feature. The standard addressing for the 3310 is "240-243", "250-253", "260-263", and "270-273". The compatibility addresses map to the address specified at IPL time, and are mapped over the remaining devices on that string and the devices on the next string (if attached).

Addressing an emulated 231x device that was not selected for the compatibility feature at IPL time results in condition code 3 (device not operational).

The user specifies the native device address (at IPL time), which contains the first emulated pack. For each native device a range of eight addresses is reserved for the emulated devices addresses, of which two are used for the 2314 and seven for the 2311.

This means, for example, that when two 2314 are mapped on one 3310, the addresses "x90" and "x91" are valid, but "x92" through "x97" are not.

The scheme in the chart in Figure 8.01 leads to the following association between 3310 device addresses and the emulated device addresses for the respective emulated devices.

To use the figures below:

- Find the correct table section for the starting FBA device address.
- From the table section, find the supported compatibility addresses (which are based on compatibility device type and to which 3310 spindle each address maps).

Starting FBA Device Address	Compatibility Address 2311	Compatibility Address 2314	Maps to FBA Device Address
xy0	x90—x96	x90—x91	xy0
	x98—x9E	x98—x99	xy1
	xA0—xA6	xA0—xA1	xy2
	xA8—xAE	xA8—xA9	xy3
	xB0—xB6	xB0—xB1	xz0
	xB8—xBE	xB8—xB9	xz1
	xC0—xC6	xC0—xC1	xz2
	xC8—xCE	xC8—xC9	xz3
xy1	x90—x96	x90—x91	xy1
	x98—x9E	x98—x99	xy2
	xA0—xA6	xA0—xA1	xy3
	xA8—xAE	xA8—xA9	xz0
	xB0—xB6	xB0—xB1	xz1
	xB8—xBE	xB8—xB9	xz2
	xC0—xC6	xC0—xC1	xz3
xy2	x90—x96	x90—x91	xy2
	x98—x9E	x98—x99	xy3
	xA0—xA6	xA0—xA1	xz0
	xA8—xAE	xA8—xA9	xz1
	xB0—xB6	xB0—xB1	xz2
	xB8—xBE	xB8—xB9	xz3
xy3	x90—x96	x90—x91	xy3
	x98—x9E	x98—x99	xz0
	xA0—xA6	xA0—xA1	xz1
	xA8—xAE	xA8—xA9	xz2
	xB0—xB6	xB0—xB1	xz3

Note: x = Channel Number (2 or 3)
y = 0 through 7 (string number)
z = y + 1 (next string number)

- Figure 8.01: 2311/2314 Compatibility Addresses on the 3310

Figure 8.01 is used in the three examples following:

- If you specify (at IPL time via the program load display) the native device address "240" for the starting native address for emulation, the emulated 2314 devices with the addresses "290" and "291" must have been mapped on the native device with the address "240". If during subsequent processing, you refer to the address "2A8" or "2A9", the emulator automatically accesses the native device with the address "243".

- If you specify (at IPL time) the native device address "351" as the starting native address for emulation, the emulated 2314 volumes with the addresses "390" and "391" must have been mapped on the device with the address "351". If you refer to "3A8", the emulator automatically accesses the native device with address "360" if this string is attached; otherwise, condition code 3 (device not operational) is presented.

- If you specify (at IPL time) the native device address "273" as the starting native address for emulation, the emulated 2314 volumes "290" and "291" must have been mapped on native device address "273". In this case, a reference to any other emulated 2314 volume results in condition code 3 (device not operational).

The formulas and tables in Figure 8.02 show the mapping of emulated devices on a native device.

Alternate tracks (cylinder numbers 200-202) are mapped in front of the prime cylinders in order to allow for subdisks. The conversion is done by the compatibility feature.

	2311	2314/2319
NOBTRK (No. of blocks per emulated track)	8	15
No. of emulated alternate cylinders	3	3
No. of tracks per emulated cylinder	10	20
NOBCYL (No. of blocks per emulated cylinder)	80	300
No. of prime cylinders per emulated pack	200	200
NOEB (No. of blocks per emulated device)	16240	60900
No. of emulated packs per native device	7	2
Starting PBN for emulated pack No. 1	126015	126015
(PBN=physical block number) No. 2	109775	65115
No. 3	93535	
No. 4	77295	
No. 5	61055	
No. 6	44815	
No. 7	28575	
Remainder of FBA space (512 byte blocks)	12336	4216

• Figure 8.02: Formula for Mapping 231x on the 3310

Note: Emulated cylinders are mapped in the following order:

- Cylinders 200, 201, 202, 0, 1, ... 199 (or number of emulated cylinders)
- The above cylinders are mapped in descending FBA address (See figures that follow.) Conversion is handled by the emulation microcode.

To compute the available free space after mapping of 231x emulated volumes, use the following formula:

126015 - (X-1) * NOEB - (3+Y) * NOBCYL = ending free block number

where X = No. of emulated pack on active device (1 through 7)
Y = No. of emulated cylinders (not including alternates)
NOEB = No. of blocks per emulated device
NOBCYL = No. of blocks per emulated cylinder

Note: Three alternate cylinders are assigned by the microcode, regardless of the number of emulated cylinders. This means that a five cylinder emulated subdisk would use eight emulated cylinders.

Example: Assume you want to define a single 2314 subdisk of 50 cylinders. The formula for the ending free block would then be:

126015 - 0 * 60900 - (3+50) * 300 = 126015 - 0 - 15900 = 110115

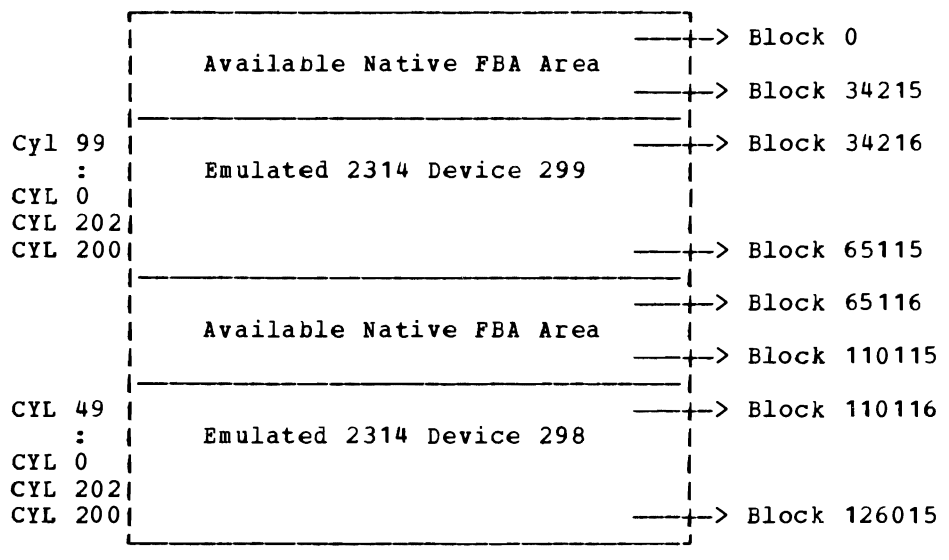
This means that the highest block number available is 110115. If no other 2314 space is to be defined on the same 3310 spindle, blocks 0 through 110115 would be available for use in native FBA mode.

Now assume that you wish to define two 2314 subdisk areas: 50 cylinders for the space addressed as "298" and 100 cylinders for the space addressed as "299" (a maximum of two 2314 areas are supported on each 3310). The formula for the ending free blocks would then be:

126015 - 0 * 60900 - (3+50) * 300 = 126015 - 0 - 15900 = 110115

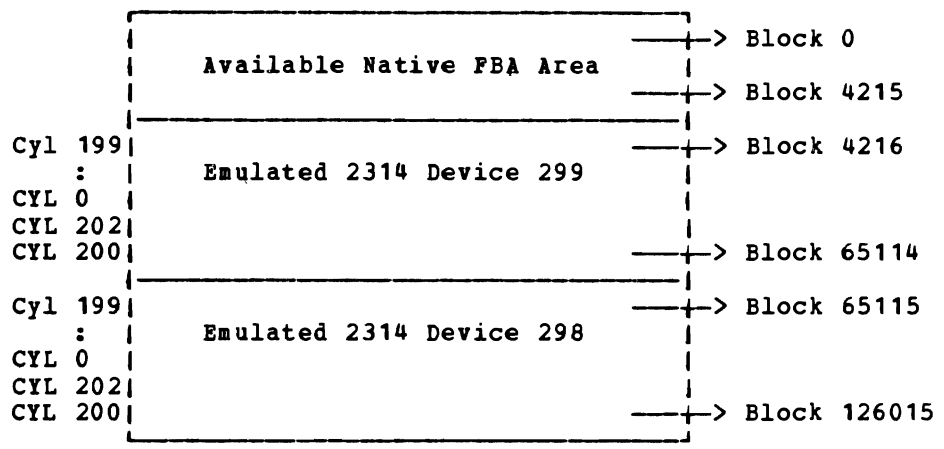
126015 - 1 * 60900 - (3+100) * 300 = 126015 - 60900 - 30900 = 34215

This means that the area from 65116 (the start of pack 2 from the table in Figure 8.02 plus one) through 110115 is available for native FBA files. Also, the area from block 0 through 34215 is available for native FBA files. Figure 8.03 would represent the 3310.



- Figure 8.03: 2314 Subdisks Mapping on the 3310

Assume that full 203 cylinder emulated 2314 packs were to be defined on the 3310. Figure 8.04 would represent the 3310:



• Figure 8.04: 2314 Mapping on the the 3310

Performance Considerations

The performance of the 231x compatibility feature is highly dependent upon the number of emulation buffers that are assigned at IPL time and the type of access to the data.

For each read operation of an emulated 231x DASD record, a full track of emulated data is read into an emulation buffer and the physical record requested is passed to the program. If the data requested is already in an emulation buffer, no FBA read occurs. For this reason, sequential processing of emulated 231x can gain a significant performance improvement.

For each write operation of an emulated 231x DASD record, a full track of emulated data must be read into an emulation buffer (provided that it is not already in a buffer). The emulated track is updated in the buffer and is written back to the 3310.

Note: The specific operation of the emulator does not yield any performance improvements when using seek separation or split cylinder extents; therefore, they should not be used.

The IBM System/3 Data Import Utility (a program product) reads data files from an IBM 3348 data module - written by an IBM System/3 Model 12 or 15 - and transfers those files to DOS/VSE SAM or VSE/VSAM files. No intermediate media for data transport (for example, magnetic tapes) are required; instead the IBM System/3 Data Import feature available on the IBM 4331 Processor is used.

Any number of files can be converted during one run of the program. One set of control statements is required for every file to be converted. Files to be converted to VSE/VSAM data sets must have been defined previously in the VSE/VSAM catalog. Only the data area of the 3348 is supported. Files in one of the IBM 5444 simulation areas must have been previously copied to the data area by the IBM System/3 $SCOPY Utility.

Figure 9.01 shows conversions supported by this utility.

S/3 File Type	DOS/VSE Data Set
Sequential	SAM
Direct (1)	VSE/VSAM ESDS or RRDS (2)
Indexed Ordered or Unordered	VSE/VSAM KSDS (2)
Indexed Unordered	VSE/VSAM ESDS (3)

(1) Handled as sequential file.
(2) Other VSE/VSAM file types are possible, if warranted by record format and application.
(3) To get indexed capability, a secondary index must be built by using the BLDINDEX function of VSE/VSAM.

• Figure 9.01: S/3 Data Import Utility Conversions

All SAM record formats are supported. Because of fixed length data format on IBM System/3 it is advisable, however, to convert to fixed blocked format.

Multiversion input files must be distinguished by either different DATE parameters or different LOCATION parameters in the FILE control statements. Since multiversion files are not supported by DOS/VSE data management, the target files must be distinguished by different file-ids. If more than one version of a multiversion file is converted in the same run, the DLBL file name parameters must also be different.

Input files may be spread over multiple volumes (up to 40). Indexed files must be mounted in proper sequence.

Input files must have been created with standard IBM System/3 data management. Their record size must not be larger than 32,760 bytes.

Job Control

The IBM 3340 attached in System/3 Data Import Mode must be assigned to SYS004. No other JCL statements are required to further define input.

For the output files created by the import utility, DLBL and EXTENT statements are required for any single file, consistent with the access method selected for output. If multiversion files are to be converted, different versions of the same file must have different file-ids in DOS/VSE. File names can be freely chosen but no two file names may be the same within the same run of the program.

The default DASD output device is SYS007, but this device number can be overridden by the EXTENT statement. This program is called by:

```
// EXEC S3DATA,SIZE=AUTO
```

The size parameter is required because the program uses VSE/VSAM during execution.

Utility Control Statements

Two control statements are required per System/3 file to be converted: FILE and OUTPUT. They must appear in that order. The use of continuation statements is allowed. The syntax of both statements follows the syntax rules of IBM System/3 OCL statements. (See IBM System/3 Model 15, System Control Programming, Reference Manual, GC21-5077).

The FILE control statement is syntactically identical with the FILE control statement of IBM System/3 for disk (single or multivolume file version).

The table in Figure 9.02 shows the required, used, and ignored parameters of the FILE statement.

Required	Used	Ignored
NAME	LABEL	UNIT
PACK		RECORDS
DATE (1)		TRACKS
LOCATION (1)		RETAIN
		HIKEY

(1) DATE or LOCATION required for multiversion files. If used for single version files, the parameters are checked for correctness.

• Figure 9.02: S/3 FILE Statement Parameters

The OUTP control statement defines the output file of the conversion of a System/3 file. It must immediately follow the FILE control statement and must be present even if none of its parameters are specified.

SYSGEN

No special PUB type is required for the 3340 that contains an IBM System/3 format data module. DVCTYP=3340 or DVCTYP=3340R should be used. When the System/3 mode is not used, the DVCDN command should be issued by the operator to avoid a generic assignment for this device.

Hardware Required/Supported

The 4331 Processor with the optional System/3 Data Import feature is required.

Output device can be IBM 3340 or 3310 (native or in 2314 emulation mode). SYSLST and SYSLOG must be assigned in the partition in which the program is to run.

Software Required

DOS/VSE with VSE/Advanced Functions is required to run the program. In addition, VSE/VSAM must be installed on the system.

When a severe error has been detected, file conversion processing on the file to be converted is terminated. If the output file was already OPENed it will be CLOSEd. The program proceeds with reading control statements for the next file, if any.

Examples of Jobstreams

Assume that the three IBM System/3 files in Figure 9.03 have to be converted to DOS/VSE files:

Name	File 1	File 2	File 3
IBM S/3 Pack(s)	VOL001	VOL002	VOL001 VOL002 (Multivolume file)
IBM S/3 Organization	SEQ	SEQ	Indexed
Multiversion	No	Yes (2nd vers)	No
Target Organization	SAM	VSE/VSAM	VSE/VSAM
Record length	64	96	128

• Figure 9.03: S/3 Conversion Example

Figure 9.04 is a typical job for defining the new VSE/VSAM files.

```
// JOB DEFINE
// DLBL IJSYSCT,'AMASTCAT',,VSAM                    (1)
// EXTENT SYSCAT,CATVOL
// DLBL IJSYSUC,'FBA.USER.CAT',,VSAM                (1)
// EXTENT SYS001,VSAM01
// ASSGN SYS001,FBA,VOL=VSAM01,SHR
// EXEC IDCAMS,SIZE=AUTO
  DEFINE CLUSTER-
            (NAME (FILE2.OLD)-                      (2)
             CISZ (1024)-
             NONINDEXED-
             RECORDS (2000)-
             RECSZ (96 96)-
             VOL (VSAM01 VSAM02))                   (3)
 DEFINE CLUSTER-
            (NAME (FILE2.NEW)-
             MODEL (FILE2.OLD))                     (4)
 DEFINE CLUSTER-
            (NAME (FILE3)-
             CISZ(1024)-
             FSPC(20 30)-
             IMBED-
             KEYRANGES ((A M) (N Z))-               (5)
             KEYS (10 0)-
             ORDERED-                               (6)
             RECORDS (1000  200)-
             RECSZ (128 128)-
             VOL (VSAM01 VSAM02))-
             INDEX (CISZ(512))
/*
/&
```

Notes:

1- Only DLBL/EXTENT for catalog is required at DEFINE time, if files are suballocated.
2- Name of FILE2 is extended to distinguish it from the file version that is by name only in DOS/VSE.
3- Even though the original files were not multivolume, two possible target volumes have been chosen for more flexibility.
4- Since characteristics between multiversion files are similar, a model parameter can be used to shorten parameter list.
5- In Access Method Services, low keys are automatically extended with binary zeros and high keys with binary ones on the right end.
6- ORDERED forces the first key range on the first volume (VSAM01) and the second key range on the second volume (VSAM02).

• Figure 9.04: S/3 Data Import Utility VSE/VSAM Define Example

Figure 9.05 shows a typical job to convert the files defined in Figure 9.04 from IBM System/3 files.

```
// JOB CONVERT
// ASSGN SYS007,DISK,VOL=VSAM01,SHR
// ASSGN SYS008,DISK,VOL=VSAM02,SHR
// DLBL IJSYSCT,'AMASTCAT',,VSAM
// EXTENT SYSCAT,CATVOL
// DLBL IJSYSUC,'FBA.USER.CAT',,VSAM
// EXTENT SYS007,VSAM01
// DLBL FILE21,'FILE2.OLD',,VSAM                (1)
// EXTENT SYS007,VSAM01
// EXTENT SYS008,VSAM02
// DLBL FILE22,'FILE2.NEW',,VSAM                (1)
// EXTENT SYS007,VSAM01
// EXTENT SYS008,VSAM02
// DLBL FILE3,'FILE3',,VSAM
// EXTENT SYS007,VSAM01
// EXTENT SYS008,,VSAM02
// ASSGN SYS009,FBA,VOL=SAM002,SHR              (2)
// DLBL FILE1,'CONVRTED.SAM.FILE',,SD
// EXTENT SYS009,SAM002,,,2000,140
// ASSGN SYS004,X'160'                          (3)
// EXEC S3DATA,SIZE=AUTO
// FILE NAME-FILE1,UNIT-D1,                     (4)
//      PACK-VOL001,RECORDS-1000
// OUTP AM-SAM,BLKSIZE-960                      (5)
// FILE NAME-FILE2,UNIT-D2,
//    PACK-VOL002,DATE-04/02/77                 (6)
// OUTP DLBL-FILE21                             (7)
// FILE NAME-FILE2,UNIT-D2,
//     PACK-VOL002,DATE-04/16/77                (6)
// OUTP DLBL-FILE22
// FILE NAME-FILE3,UNIT-'D1,D2',
//     PACK-'VOL001,VOL002',
//     TRACKS-'13,13',
//     HIKEY-'MUTZELMANN,ZZZZZZZZZZ'
// OUTP                                         (8)
/*
/&
```

Notes:

1- File names (and fileids) for multiversion files must differ since both versions of FILE2 are converted in the same run.
2- Start job control for SAM file.
3- Only ASSGN is required to define input device.
4- UNIT parameter is ignored on DOS/VSE.
5- Defaults applied are DLBL-FILE1 (from FILE) and RECFM-FB.
6- DATE parameter indicates version of file.
7- DLBL parameter is required since file name in DLBL statement is different from IBM System/3 file name (see note 1).
8- Defaults applied are DLBL-FILE3 (from FILE) and VSE/VSAM.

• Figure 9.05: S/3 Data Import Utility Conversion Example

This Page Intentionally Left Blank

The IBM 3310

Figure 10.01 provides hardware planning information needed when installing the 3310.

Attachment:	IBM 4331 via the DASD Adapter.
Power:	A-1 .15 KVA. A-2 .23 KVA. B-1 .08 KVA. B-2 .16 KVA.
Space:	Width: 24 inches (609 mm). Depth: 32 inches (813 mm). Height: 40 inches (1000 mm).
Heat Output:	A-1 1030 BTU per hour. A-2 1975 BTU per hour. B-1 940 BTU per hour. B-2 1880 BTU per hour.
Operating Range:	10°-40.6° C. 50°-105° F.

• Figure 10.01: 3310 Planning Information

The IBM 3370

Figure 10.02 provides hardware planning information needed when installing the 3370.

Attachment:	IBM 4331 via the DASD Adapter and 4341 via the 3880 Control Unit on the Block Multiplexor Channel.
Power:	A-1 .9 KVA. B-1 .5 KVA.
Space:	Width: 30.5 inches (770 mm) - A-1. Width: 21.0 inches (542 mm) - B-1. Depth: 32.0 inches (814 mm). Height: 40.0 inches (1000 mm).
Heat Output:	A-1 3080 BTU per hour. B-1 1710 BTU per hour.
Operating Range:	16°-32° C. 60°-90° F.

• Figure 10.02: 3370 Planning Information

This Page Intentionally Left Blank

Program performance considerations for files on FBA DASD are much the same as those for files on CKD DASD. Generally, the same considerations for block size or blocking factor on CKD DASD apply to Control Interval size.

Sequentially Processed Data

If the data is to be processed sequentially, such as a SAM data set or a Entry Sequenced Data Set (ESDS) in VSE/VSAM, then the CI size should be large. This allows the IOCS module to bring in large amounts of data with one physical I/O operation and then pass the various logical records to the program via storage to storage transfers. Of course, the larger CI sizes will require a larger GETVIS space and will, therefore, increase the working set of the program. In this case you are trading the guaranteed requirement for I/O operations with small CI sizes for the possibility of I/O operations via paging.

Randomly Processed Data

If the data is to be processed randomly,such as a Relative Record Data Set (RRDS) or Key Sequenced Data Set (KSDS) in VSE/VSAM, then the CI size should be the smallest that will accomodate the logical block. This will allow the IOCS modules to bring in the smallest amount of data when only the single logical block is desired.

Both Random and Sequential

When the access to a file is both random and sequential, a happy medium must be picked. Large CI sizes during random processing are costly in the sense of storage used, working set, and overhead for the I/O. Small CI sizes during sequential processing are costly in the sense of more physical I/O operations required to process the file. You must choose a CI size that will provide a reasonable amount of logical records without having high data transfer times for the I/O operation.

Using the IBM 3340 on the DASD Adapter

The 3340, when attached to the DASD adapter of the 4331, does not operate as a normal CKD DASD device. Each track of the 3340 must be read into processor storage for a read operation. For a write operation, the track must be read into storage and then rewritten.

RPS should not be specified for the 3340 devices attached to the DASD adapter on the 4331. The operation of the adapter does not yield improved performance of the 3340, whereas the RPS overhead in the I/O supervisor is still executed.

This Page Intentionally Left Blank

This chapter is a summary of the procedures to convert to the FBA DASD devices on the IBM 4300 Processors.

Organize Your Department for the Conversion

- Establish objectives
- Appoint conversion coordinator
- Set up review procedures, such as:
  - Periodic system reviews within department
  - Checkpoint meetings with IBM personnel
- Order a new library of hardware and software publications
- Determine the manpower requirements for conversion and choose personnel

Establish Education Plan

- Review IBM curriculum for such topics as:
  - System generation
  - Job Control Language
  - Operating procedures
  - Languages
  - Data management
  - Supervisor and I/O macros
  - Sort/Merge
  - System utilities
- Schedule classes

Develop Detailed Conversion Plan

- Establish an approach to program and file conversion
- Develop a plan for parallel testing to minimize production slowdown
- Assign tasks to personnel
- Establish program testing standards
- Take inventory of programs and files and state interdependencies

Review Data Center Facilities

- Check availability of programming facilities well in advance, especially:
  - Supervisor requirements
  - Sort/Merge and utilities availability
- Check hardware configuration
- Check device characteristics, such as:
  - Tape densities
  - DASD types
  - Model numbers
- Verify device addresses and system main storage size
- Determine whether alternate solutions or device substitutions are required
- Determine backup procedures
- Investigate the number and location of test sites you require, for example:
  - Find one location for dumping CKD files to tape.
  - Schedule time at data center.
  - Inquire about availability of scratch tapes for backup of FBA volumes at the end of each test session.

Contact IBM Services

- FE hardware specialist
- FE software specialist
- Systems Engineers
- Sales personnel

Take Inventory of Tape Library

- Check whether or not you have enough tapes
- Order more, if necessary
- Check delivery schedules

Implement Conversion Plan

- Convert programs, if necessary
- Convert existing sorts and utilities
- Convert files
- Investigate the possibility of changing record or block sizes to take advantage of new FBA DASD
- Check I/O area requirements
- Determine data set placement on each volume
- Review file organization techniques
- Standardize terminology (label procedures, file names, etc.)

Review Status of Equipment on Order

- Check for correctness and completeness
- Confirm delivery schedule

Review Status of Programming System

- Order DOS/VSE release and check delivery schedule
- Order program products
- Order optional material (for example, source listings of components) if program logic descriptions are needed

Lay Out Physical Plan with IBM

- Check site and schedule improvements, if necessary, for:
  - Floors
  - Air conditioning
  - Electrical power
  - Cable lengths
  - Lighting
  - FE service clearance
- Review requirements for parallel testing and concurrent operations
- Check physical requirements for delivery, such as:
  - Door height and width
  - Corridor width
  - Elevator capacity
  - Rigging requirements
  - Tape storage cabinets
  - Tape library requirements

Plan for System Generation

- Hold review with staff
- Hold review with IBM Systems Engineers and Customer Engineers
- Determine supervisor parameters and size
- Begin planning for future expansion
- Prepare system generation parameters
- Determine standard assignments and labels
- Apply PTFs, if necessary
- Generate system
- Test system with sample programs or with samples of your own programs
- Generate system again, if necessary

Complete Conversion to New System

- Have all programs running on the new system and modify the JCL for your configuration
- Complete file conversion
- Run parallel tests and verify results
- Balance the results with control procedures
- Run new work on the new system

Release Old System

Hold Periodic Reviews

- Check system performance
- Determine where incremental improvements can be made
- Determine where performance and efficiency can be improved

Plan Major Redesigns and Expansion

Processing Considerations

Since the unit of data transfer for FBA devices is a CI instead of a physical block employed with CKD devices, error notification and recovery must be viewed differently. With CKD devices, error detection and notification was done on the basis of a physical block. With FBA devices, the error detection and notification will be done on the basis of a Control Interval. Use of the SKIP, IGNORE or RETRY options with the ERET macro or with the ERROPT parameter will be considered as referring to the CI in error.

User-Written Error Exits

User-written error exits (ERROPT=xxxxxxxx and WLRERR=xxxxxxxx) that are sensitive to data format may require modification to correctly process the CI, instead of the physical block, in error. No LIOCS imperative macro other than ERET can be issued to the logic modules from a written error routine for the file that encountered the error.

For spanned records the logic modules will skip or ignore CIs until a valid first segment of a record is accessed; whereas, with CKD devices they bypass physical blocks until a valid first segment is found. Therefore, even though the logical action is the same for CKD and FBA devices, different records may result from the search for the beginning of next record. Any dependence of user-written error routines on the use of physical device addressing and the mapping of consecutive logical blocks to consecutive physical device addresses will require evaluation to determine the necessity for modifications.

Since blocking within a CI buffer is done by the SSR (SAM Service Routine), there are times when error detection is not synchronized with the user's current request. For example, output errors may be detected during update processing of a GET. This situation arises with current CKD support in certain cases.

Errors for non-update data files are also affected by blocking within the CI buffer. Errors are not detected on output until the user has PUT subsequent records.

When the user provides two IOAREAs with a DTF (input or output without update), an overlapping I/O with processing of the next logical block results. During utilization of I/O overlap there are times when error detection is not sychronized with the user's request.

Error status indicators are maintained in the CCB by the logic modules and the SSR. The CCB will not be maintained by the supervisor since the logic modules and SSR use an I/O Request Block (IORB) which does not reside at the beginning of the DTF, in order to take performance advantage of a user-specified fix list. This means that problem programs must not depend on the CCB being maintained synchronously with actual I/O. Once error status indicators are copied to the CCB from the IORB, they will remain in the CCB until a subsequent CI I/O operation has been completed. If an input error occurs, the residual count in the CCB will be set to the CI size, indicating that no data was transferred. If the user's response to this error is IGNORE, useful data will not be

provided because the contents of the work area or I/O area are unreliable. Other fields of the DTF that normally contain information unique to CKD devices will contain invalid information with the exception of the following:

- Filename.S will be maintained with a physical FBA device address denoting the current position in the file analogous to CKD support. The value in filename.S has zeros in the first two bytes - a physical block number of the first FBA block used to contain the current CI. The physical block number is relative to the beginning of the device.

- The upper extent limit will be maintained in the DTF. This value is in terms of physical block number in place of CCHH, as is used for filename.S.

Because of the requirement of certain applications, such as logging, that the physical output be triggered by the transfer of a logical block instead of the filling of a CI, a way is provided (via the DTF) for the user to force the logic modules to request that the physical output be done for each logical block, whether or not the CI buffer is full. This will result in the complete CI being written each time a logical block is transferred to the CI as a result of a formatting write request; see the PWRITE parameter of the DTFSD Macro.

As each CI is written for a formatting output request (PUT or WRITE SQ) a software end-of-file (SEOF) is written in the CI following the data CI. Note that in work file processing WRITE SQ causes the insertion of this file delimiter where, in CKD processing, it erased one track of the file. If such a file is later read sequentially, it will be terminated at the SEOF.

ERREXT Changes

Since ERREXT=YES is forced when accessing SAM files on FBA DASD, register one contains the address of a two-part parameter list. This list contains a four byte DTFSD address and a four byte address of the Control Interval in error. The unit of data transfer is a Control Interval instead of a physical block.

Declarative Macros

DTFSD Changes

- CONTROL=YES - This operand may be specified or left out. The CNTRL macro will always be ignored as there is no seek in FBA.

- ERREXT=YES is assumed. This means that user error routines can use the ERET macro to return to LIOCS and, on entry to user error routines, register 1 points to a two-part parameter list containing a four byte address of the DTFSD followed by the four byte address of the Control Interval in error. Assumption of ERREXT=YES is necessary because there is no indication in the DTF of whether the user's error routine expects a one-part parameter list or a two-part parameter list. The FBA superset module will handle the two-part parameter list just as the RPS superset module does.

- ERROPT=IGNORE - The error that will be ignored is an error in reading or writing a Control Interval.

- ERROPT=SKIP - All records in logical blocks in the Control Interval in error will be skipped. The first record in the next error-free Control Interval will be the next record returned to the user.

- ERROPT=xxxxxxxx may be used to specify a user-error handling routine. The use of ERET=(IGNORE or SKIP or RETRY) options will result in actions similar to ERROPT=SKIP and ERROPT=IGNORE in that the Control Interval in error is the target for the error handling instead of the logical block. No IOCS imperative macro other than ERET can be issued to the logic modules from a user-written error routine for the file that encountered the error.

- WLRERR=xxxxxxxx - This routine is no longer called for physical wrong-length errors since physical I/O is now concerned with Control Intervals. It will, however, be called for "logical" wrong-length errors detected while retrieving a logical block from within a control interval. The CCB information (residual count and the CSW status flag for incorrect length) will be set up by the logic module when the error is detected. No LIOCS imperative macro other than ERET can be issued to the logic modules from a user-written error routine for the file that encountered the error.

- BLKSIZE=nnnnn - This parameter is no longer used to determine physical block size. It will, however, be used to determine the "logical" block size and Control Interval size. Thus, it is still a required parameter, even if IOAREA1 is omitted. BLKSIZE will be restricted to 32,761 bytes, 7 bytes less than the maximum CISIZE.

- MODNAME=xxxxxxxx - This parameter will be ignored by FBA OPEN if coded. It will also result in an MNOTE with the DTFSD macro if coded in conjunction with DEVICE=FBA.

- RDONLY=YES - This parameter is ignored. All FBA logic modules are coded as reentrant (also read-only) modules. This allows a single copy of the logic module to serve all users of that logic module in the system. These modules do not depend on a save area being provided by the problem program at the address in register 13 but depend on a save area per DTF in the DTF extension provided by OPEN. This means that issuing LIOCS macros other than ERET in user-written error routines is not supported.

- HOLD=YES - Use of this parameter provides the write integrity protection function as it does for CKD devices, except that the unit held is a Control Interval instead of a track.

- DEVICE=FBA - This parameter is optional for files on FBA devices. Coding this parameter will cause an expanded DTF to be generated to contain an optional CISIZE. Coding this new parameter will also avoid the generation of a External Address Constant (VCON) for a logic module and will allow the specification of a block of any size up to 32,761 bytes. The logic module to be used resides in the SVA and its address supplied to the DTF at OPEN time. Specifying DEVICE=FBA will force MODNAME to be ignored.

- CISIZE=nnnnn - nnnnn is the CISIZE and may be as large as 32,768 bytes. If DEVICE=FBA is specified and CISIZE is not, zeros will be generated in CISIZE field of DTFSD. The CISIZE must be a multiple of the FBA block size and, further, must be a multiple of 2K if it is to be greater than 8K. This will not be checked by the macro generation but is checked at at OPEN time when the CISIZE is not being overridden by a CISIZE parameter on the DLBL statement.

  Note: If the CISIZE is coded in the DTF and the logical block size is increased to be larger than the CISIZE, an OPEN failure will occur.

  If DTFSD is opened for output and the CISIZE is overridden by a CISIZE specification on the DLBL statement, the CISIZE will be

moved to the CISIZE field in DTFSD.

If DTFSD is opened for input, and DEVICE=FBA is either specified or defaulted to in the DTF, the CISIZE found in the Format-1 label will be moved to the CISIZE field in the DTF.

- WORKA=YES - If this parameter is specified, logical records will be moved directly to or from the work area without using any IOAREAs. If IOAREAs are also specified, it is ignored. This applies to all input or output files.

- IOREG=(r) - The IOREG register will now point into the CI buffer if no IOAREA is used. See IOAREA1 below.

- IOAREA1=xxxxxxxx - This parameter is no longer always required. It is not needed and will be ignored if specified if WORKA=YES is specified. This is true for any input or output file. If IOREG is specified for any input file or for an output file with fixed-length records without truncation, IOAREA1 is not required. If IOAREA1 is not specified and IOREG is specified, the I/O module will point IOREG directly to data in the CI buffer. If IOAREAs are specified, logical blocks will be moved to and from the IOAREAs, and IOREG will point into the IOAREA. IOAREA1 is still required for all other output files.

- IOAREA2=xxxxxxxx - This parameter will be ignored if IOAREA1 is not specified.

- PWRITE=YES - This is a new parameter applicable to output files and work files. This operand is specified if the formatting of output operations (PUT for data files and WRITE SQ for work files) is to cause an actual physical write operation for each logical block. If this operand is not specified, the physical write will only occur when the CI buffer is full. Because the complete Control Interval is written each time a logical block is placed into the Control Interval, performance may be degraded.

SDMODxx

The logic modules are assembled and linkedited before system distribution. They will be loaded into the SVA at IPL time. A determination will be made at OPEN time about which one to use and it will be connected to the DTF. Because of this, problem programs no longer need to use SDMODxx to produce a logic module. A considerable saving of space in the Relocatable Library can be realized. Since no IOCS module is linkedited with your programs when DEVICE=FBA is specified, a considerable amount of Core-Image Library space can be saved.

DTFPH Changes

- DEVICE=FBA - Specifying this parameter on DTFPH causes the DTF device type code for FBA devices to be generated in the DTF. This parameter is mandatory for DTFPH on FBA devices. It also causes DTFPH to generate an expanded DTF to contain an optional CISIZE.

- CISIZE=nnnnn - nnnnn is the CISIZE in bytes and may be as great as 32,768. If this parameter is specified, DEVICE=FBA will be assumed. If DEVICE=FBA is specified and CISIZE is not specified, zero will be generated by the DTFPH macro in the CISIZE field. The CISIZE must be a multiple of the FBA block size and, further, must be a multiple of 2K if it is to be greater than 8K. This will not be checked by

the macro generation but is checked at at OPEN time when the CISIZE is not being overridden by a CISIZE parameter on the DLBL statement.

Note: This parameter is supported whenever a file is created using DTFPH for later processing with DTFSD. If no CISIZE is specified at file creation time, the file may not be processed by DTFSD. It will be assumed by SD processing that the file is in CI format if a CISIZE is specified.

- OPEN will take the following actions:
  - If DTFPH is opened for output and a CISIZE is specified on the DLBL statement, this CISIZE will be moved to the CISIZE field in DTFPH.
  - If DTFPH is opened for input, the CISIZE with which the file was created with will be moved to the CISIZE filed in DTFPH from the Format-1 label.
  - The CISIZE will be checked to be sure it is a multiple of a FBA block size, and is a valid VSAM CI size.

Note: For MOUNTED=SINGLE, the "codes" parameter of the DLBL statement should be "SD". For MOUNTED=ALL, the "codes" parameter should be "DA".

DTFDI Changes

- FBA=YES - Specifying this parameter on DTFDI causes DTFDI to generate an expanded DTF to contain an optional CISIZE.
- CISIZE=nnnnn - nnnnn is the CISIZE in bytes and may be as large as 32,768. If DEVICE=FBA is specified and CISIZE is not, a zero CISIZE will be generated by the DTF macro. The CISIZE must be a multiple of 512 and, further, must be a multiple of 2K if it is to be greater than 8K. This will not be checked by the macro generation but will be checked at OPEN time when the CISIZE is not being overridden by a CISIZE parameter on the DLBL statement.

If DTFDI is opened for output and the CISIZE is overridden by a CISIZE specification on the DLBL statement, the CISIZE will be moved to the CISIZE field in DTFDI.

If DTFDI is opened for input and DEVICE=FBA or CISIZE=nnn is specified in the DTF, the CISIZE found in the Format-1 label will be moved to the CISIZE field in DTFDI.

DIMOD Changes

- A functional superset logic module is assembled and linkedited before system distribution. It will be loaded into the SVA at IPL time and connected to the DTF at OPEN time.

General Information

Interfaces that are maintained without change are not discussed (for example, POINTS).

OPEN Changes

With DASD Output, only Type 1 extents are valid. Split cylinder extents are not valid on FBA devices because the addressing scheme does not use cylinders.

- At least one Control Interval (the first) on the first extent of each volume of a file will be reserved for user labels when they are specified in the DTF.

  Note: A programming change is necessary to programs that read and write their own user labels.

- After processing the last extent of an output file, the operator will be given the option of entering another extent and number of FBA blocks.

- OPEN for output will determine Control Interval size as follows:

  - If CISIZE has been specified on the DLBL statement, the CI size will be taken from there.

  - If CISIZE was specified in the DTF, the CI size will be taken from there.

  - For SYSLNK open, the CI size will be 1K.

  - If (BLKSIZE+7) is less than or equal to 8K, choose the smallest integral number of FBA blocks that will contain the (BLKSIZE+7). (If two IOAREAs were specified, the BLKSIZE will be doubled before making the calculation).

  - If (BLKSIZE+7) is greater than 8K, round up to a 2K integral. (If two IOAREAs were specified, the BLKSIZE will be doubled before making the calculation unless doing so would cause a CISIZE greater than 32K to appear).

  - The CISIZE will be stored in the Format-1 label at file creation time and retrieved directly from there when opening an input file.

- From the partition GETVIS area, OPEN will GETVIS a DTF extension, a work area for the logic modules and SSR, and an area large enough to read in one Control Interval of the size determined above for all files except system files. For system files, only the DTF extension and a work area for the SSR and logic module is GETVISed.

- Additionally, OPEN will GETVIS an area in the partition large enough to use for its own work space. This area will be FREEVISed before exiting OPEN.

GET Macro

Because of the Control Interval concept of managing data records on FBA devices, a GET does not necessarily result in a physical read in all of the same cases as in sequential disk processing for Count-Key-Data devices.

PUT Macro

PUT does not necessarily result in a physical write in all of the same cases as in sequential disk processing for Count-Key-Data devices. For applications that require physical writes for each logical block, the force write parameter (PWRITE=YES) on the DTF must be used.

CNTRL Macro

The CNTRL macro is treated as a NOP because there is no seek on FBA devices.

ERET Macro

All options on this macro refer to Control Intervals only.

TRUNC Macro

The TRUNC macro causes IOCS to discontinue the addition of logical records to the current logical block. It will not necessarily result in a physical write to the FBA device unless the force write parameter (PWRITE=YES) of the DTF is specified.

RELSE Macro

The RELSE macro discontinues deblocking of the current logical block. It will not necessarily result in a physical read from the FBA device.

READ Macro

READ results in the moving of a logical record to the input area.

Because of the Control Interval concept of managing data records on FBA devices, a READ will not necessarily result in a physical read in all of the same cases as in sequential disk processing for Count-Key-Data devices.

WRITE Macro

WRITE results in transferring the logical record from the output area to the CI buffer. The record length will come from either the length parameter on the WRITE macro or the BLKSIZE parameter in the DTF. Physical writing to the device does not necessarily occur for each WRITE macro unless the force write parameter (PWRITE=YES) of the DTF is specified.

CHECK Macro

The CHECK is still required but will sometimes provide an immediate return (when IOCS is providing an extra level of buffering and only a move is done). The CHECK macro must be issued following each READ or WRITE request; if it is not, the results will be unpredictable.

NOTE Macro

The NOTE macro returns the following information:

Register 0 will contain the size of the space that would be available in the CI less three, if the NOTEd record were the last record in the CI. The NOTEd record will, in fact, be the last record in the CI whenever the NOTE follows a WRITE SQ. The value returned is the length of largest logical block (record) that can be guaranteed to fit in the CI following the NOTEd logical block (record). A logical block (record) three bytes longer than the returned value will fit in the CI, provided it is of the same length as the NOTEd record and the record preceding the NOTEd record in the same CI. If the NOTE was preceded by a POINTR or POINTW macro, the register 0 passback will be the value provided in the fifth and sixth bytes of the address operand of the POINT macro.

Register 1 contains an address relative to the beginning of the file. The first three bytes contain the relative CI number of the current CI within the file (with origin 0). The fourth byte contains a relative record number of the logical block within the current CI (with origin one).

The format of the NOTE ID passed in register 1 imposes the following restrictions on work files:

- The size of a work file is limited to the number of CIs that may be expressed in a three byte number. If you wish to further enlarge your file after having used 16,777,216 CIs, you must use larger CIs.
- No more than 255 logical blocks per CI are allowed. This is consistent with the one byte record number in CCHHR for CKD devices.

FEOVD Macro

FEOVD causes IOCS to discontinue processing all extents of the data set on the present volume. FEOVD does not write end-of-volume indicators as is done in CKD support, since a SEOF would have already been written following the last data CI for output files if there were room.

CLOSE Macro

CLOSE effects a call to the logic module to write the last Control Interval, if necessary. CLOSE also writes the software end-of-file (SEOF).

POINTR Macro

The POINTR macro provides the same function as was provided for CKD devices but the data passed via the address operand must be of the form obtained from the NOTE macro. The first three bytes must be a relative CI number within the file and the fourth byte must be a logical block number within the CI.

POINTW Macro

The POINTW macro provides the same function as was provided for CKD devices but the data passed via the address operand must be of the form obtained from the NOTE macro. The first three bytes must be a relative CI number within the file and the fourth byte must be a logical block number within the CI.

FREE Macro

Because the HOLD=YES support is done on a CI basis rather than for each logical block, this support is handled implicitly for work files as has been done for data files. Therefore, there is no need for the work file user to issue an explicit FREE and the FREE macro, if issued, will be treated as a NOP.

This Page Intentionally Left Blank

Programs using DTFPH/EXCP will not run on the 4300 Processors with native FBA DASD without change. Programs using DAM files must be converted to native VSE/VSAM or to DTFPH/EXCP. Since VSE/VSAM provides both an indexed structure and a relative record structure, program logic should remain relatively similar for those programs which use unique keys for accessing records, or physical addresses or relative record numbers. However, the conversion of some programs using DAM requires a considerable amount of work and the type of addressing of records must be examined carefully.

The programs using DTFPH have to change in accordance with the new architecture of the FBA DASD devices. Below is a discussion of the channel commands. For more detailed information pertaining to channel commands for a specific FBA DASD device, see the appropriate reference manual for that DASD device.

Fixed Block Channel Commands

The channel commands can be divided in the following command types: Control, Read, Write, Sense, and Diagnostic.

Control Commands

The Control commands are:

- No-Operation Hex Code (03)
- Define Extent Hex Code (63)
- Locate Hex Code (43)

No-Operation (03)

The No-Operation command causes no action to be performed.

Define Extent (63)

The Define Extent defines the location and size of a data extent. Within this extent subsequent chained commands are permitted to operate.

Locate (43)

The Locate command specifies the location of the data and the number of blocks to be processed. The Locate command may be issued only if a Define Extent or a Read IPL command had been previously issued in the same command chain.

Read Commands

The Read commands are Read and Read IPL.

Read Command (42)

The Read command causes data to be transferred from the control unit to storage. The Read command must be chained from a Locate command. Although a Read command may not be command-chained from another Read command, data chaining of Read commands is permitted. The count in the Locate command gives the number of blocks to be read by one Read command.

If the CCW count is less than the block count of the Locate command times the bytes per block, data transfer will be stopped as soon as the CCW count equals zero. If the CCW count is greater than the block count of the Locate command times the bytes per block, data transfer will be stopped as soon as the block count equals zero.

Read IPL (02)

The Read Initial Program Load command must be the first command in a chain, or it must be chained from another Read IPL. This command executes an implicit Define Extent and a Read of block zero.

Write Command

The Write command is Write.

Write (41)

The Write command causes data to be transferred from storage to the control unit. The Write command must be chained from a Locate command. Although a Write command may not be command-chained to another Write command, data chaining of Write commands is possible.

The count in the Locate command gives the number of blocks to be written by one Write command.

If the preceding Locate command specified Write Data, the control unit will present Channel End and Device End at the completion of data transfer. If the preceding Locate command specified Write and Check Data, then the Channel End will be presented after the data transfer and an access back to the first block to be read. The Device End will be presented after the last block has been read by the control unit, but no data is transferred to the channel.

If the CCW count is less than the block count of the Locate command times bytes per block, then data transfer will be stopped as soon as the CCW count equals zero. If the CCW count is greater than the block count of the Locate command times bytes per block, then data transfer will be stopped as soon as the block count equals zero.

Example of Channel Program

A channel program to read data contains the following commands:

- Define Extent.
- Locate.
- Read.

A channel program to write data contains the following commands:

- Define Extent.
- Locate.
- Write.

Programs Using DTFPH/EXCP

As you can see, the new channel program is rather different from the old CKD channel program.

A typical CKD channel program consists of the following:

- SEEK to Cylinder/Head.
- Set Sector (if using RPS).
- Search Key.
- TIC to the Search.
- Read.

When a CKD channel program is compared to a FBA channel program, the following differences can be seen:

- Define Extent has no equivalent in CKD. This command must be the first command of each channel program; it should cause no problems because the information needed - starting block number and ending block number - is available to you via the EXTENT statement information in the DTF.
- The Locate command has the same function as the Seek command on CKD. The differences are that the Seek points to a track (up to the capacity of the track) and the Locate points to an FBA block (512 bytes). This command requires that you be much more specific in defining the location of the data.
- With FBA the Locate also performs the function of the CKD Set Sector command.
- With FBA there is no equivalent for the CKD Search command. You cannot search for a key; the Locate will point to an FBA block that is to be read or written. Finding the desired record within an FBA block is the responsibility of the programmer.
- With FBA you can only read or write one or more blocks of 512 bytes, which may require additional blocking/deblocking via programming. Additional blocking may cause changes in your programs.

This Page Intentionally Left Blank

Define Space/UNIQUE Cluster

On the EXTENT statement a relative block number indicates where the data extent is to begin. The actual beginning block is calculated by rounding to the beginning of the next higher track:

$$RTRKNO = \text{ceiling of } \left(\frac{RBLKNO}{BLKPTRK}\right)$$

Note: RTRKNO = relative track number of extent beginning.
RBLKNO = relative block number specified.
BLKPTRK = blocks per track, a device characteristic.

In addition, the extent size is specified in terms of blocks, which will be converted to tracks:

$$TRKS = \text{floor of } \left(\frac{RBLKNO + BLKS}{BLKPTRK}\right) - RTRKNO$$

Note: TRKS = number of tracks to be allocated.
BLKS = number of blocks specified for extent size.

With this conversion algorithm, the allocated space will never be larger than the space specified and will always be within the specified limits.

This scheme avoids overlaps with files not in VSAM space and VSAM data spaces on the same pack at the expense of providing, at most, less space than requested. However, by specifying the relative block starting number and number of blocks each as a multiple of the blocks per track, no rounding is necessary.

Define Suballocated Cluster, Alternate Index

The space allocated for a cluster will be converted from blocks to tracks as follows:

$$TRKS = \text{ceiling of } \left(\frac{BLKS}{BLKPTRK}\right)$$

The same conversion algorithm applies to DATA and INDEX component allocations. Once the number of tracks is calculated, allocation follows the same rules as with CKD DASD. Specifically:

If there is less than one cylinder, the allocation is TRKS. If TRKS is greater than or equal to one cylinder, then the allocation is a number of cylinders equal to:

$$CYLS = \text{ceiling of } \left(\frac{TRKS}{TRKPCYL}\right)$$

Note: TRKPCYL = number of tracks per cylinder.

Define Catalog

The data space defined for a catalog will be allocated in accordance with the same rules as DEFINE SPACE, using the information in the EXTENT statement.

If an allocation parameter has been specified on the catalog level only, the allocation is as follows:

$$TRKS = \text{ceiling of } \left(\frac{BLKS}{BLKPTRK}\right)$$

If this result should be larger than the first extent of the data space, TRKS is decreased by one.

If allocation parameters have been specified on the catalog cluster and the data levels, the catalog size will be as follows, using the allocation specified at the data level:

$$TRKS = \text{ceiling of } \left(\frac{BLKS}{BLKPTRK}\right) + HLI + HKR$$

If TRKS is larger than the first extent specified on the EXTENT statement, it will be be calculated in the following way:

$$TRKS = \text{floor of } \left(\frac{BLKS}{BLKPTRK}\right) + HLI + HKR$$

Note: HLI = Calculated High Level Index size.
HKR = Calculated High Key Range size.

If allocation parameters have been specified on the catalog, data, and index levels, the values on the data and index levels will be added and the catalog size will be calculated as follows:

$$TRKS = \text{ceiling of } \left(\frac{BLKS}{BLKPTRK}\right)$$

If this result should be larger than the first extent of the data space, TRKS is decreased by one.

After those rounding algorithms have been applied, the current algorithm will be used to round TRKS to the next smaller multiple of a data control area size and the result is checked against the data space size. If it should be larger than the data space size, an error code will be returned to the caller.

If RECORDS is specified on the Access Method Services DEFINE command, current rounding algorithms will be applied.

This Page Intentionally Left Blank

SAM Space Management

This appendix is provided for planning for the use of VSE/VSAM Release 2, which includes VSE/VSAM Space Management for SAM.

If the VSE/VSAM Space Management for SAM feature is not installed, the starting block number and the number of blocks for each SAM file must be specified. You would then have to keep track of the used disk space per file for each disk, as you did in the past for CKD; however, there is a new automatic space management for SAM files.

The space management for SAM files is performed by VSE/VSAM and is described in the next section. This function requires VSE/VSAM Release 2 and VSE/VSAM Space Management for SAM feature, both of which are program products.

VSE/VSAM Space Management for SAM

Introduction

Currently, you must be concerned with planning, organization, and maintenance of space for DASD files. Consideration must be given to specific location of files, DASD device characteristics, blocksizes, and blocking factors and involves considerable manual control. This is especially a problem in a multiprogramming environment. This burden will be relieved by providing a DASD space management function. DASD space can be assigned to space management so that you can request amounts of DASD space rather than absolute locations.

This chapter is a description of the space management support using VSE/VSAM. The support provides the capabilility for the SAM user to define and process a SAM ESDS within VSE/VSAM data space providing automatic space management of the file. The support includes VSE/VSAM data spaces on all DASDs (FBA and CKD) supported by VSE/VSAM. It supports users of DTFSD (data and work files), and DTFPH for DASD with MOUNTED=SINGLE. This support is for programmer logical units (and SYSLNK) only, and is especially useful for work files.

Note: For purposes of this discussion, "SAM ESDS" will be used to indicate a SAM file in managed VSE/VSAM space.

The following functions are provided by VSE/VSAM Space Management Program Product:

- Definition of a SAM ESDS in VSE/VSAM space via Access Method Services
- Dynamic extension of a SAM ESDS according to the secondary allocation specified at the definition of the file
- Fixed Head Support (CLASS) by Access Method Services and Catalog Management for a SAM ESDS (as for VSE/VSAM files)
- Password protection for a SAM ESDS in VSE/VSAM data space (if explicitly defined) provided via operator response or Access Method Services commands (for example, REPRO).
- SHAREOPTION support for a SAM ESDS in VSE/VSAM data space (if explicitly defined)
- REPRO and PRINT for a SAM ESDS in VSE/VSAM data space by Access Method Services (except for files that are not in SAM ESDS format; in other words, DTFPH non-CI format)
- Listing of allocation and other attributes of SAM files in VSE/VSAM data space via LISTCAT by Access Method Services
- EXPORT(RA) and IMPORT(RA) of a SAM ESDS (except non-CI format)
- The ability to alter or delete catalog entries, by Access Method Services in the ALTER and DELETE commands
- Listing of catalog recovery areas or comparing catalog recovery areas with catalogs via LISTCRA command by Access Method Services
- Catalog recovery support via RESETCAT
- Full DASD independence by loading the proper IOMOD during OPEN. The DEVICE parameter of the DTF is ignored by OPEN.
- Automatic CLOSE of files that have not yet been CLOSEd by end of job step
- Implicit DEFINE of a reusable ESDS during OPEN if the file has not yet been explicitly DEFINEd via Access Method Services
- Implicit DELETE of a file that has been implicitly DEFINEd under control of the DTF (DELETFL parameter) disposition parameter (at CLOSE) or that is using the DISP parameter on the DLBL statement
- VSE/VSAM access of a SAM ESDS for read, update, and create access (except for non-CI format)

Restrictions

The following restrictions apply to a SAM ESDS in VSE/VSAM space:

- SAM spanned records are not supported.
- FEOVD is not supported.
- CNTRL is not supported.
- User labels are not supported.
- Support is provided for programmer logical units and SYSLNK only.
- Files are limited to 16 extents per volume.
- CISIZE or BLKSIZE override of the DTF in the DLBL statement is not supported.
- Data Secured File (DSF) is not supported in the DLBL statement. VSE/VSAM files are always Data Secured Files. VSE/VSAM password protection may be used.
- Files to be SAM-accessible must be in CI format on both CKD and FBA, and must be indicated as CI format in the VSE/VSAM catalog.
- Split cylinders are not supported.
- Subsetting of the input file via EXTENT statement specifications is not supported.
- Special system open (DTFSD (data) + X'10', bit 1 (X'40') on) is not supported.
- Neither an AIX nor a PATH may be defined over the file.
- Explicit define of the file may not specify ERASE or SPANNED.
- SAM files in VSE/VSAM space are not supported by OS VSAM.

Benefits of Support

Through the use of VSE/VSAM space management, you can place many details of space management responsibility on the system. By introducing space (ideally, entire volumes) to VSE/VSAM, you may leave the management of that space (volume) to VSE/VSAM. If necessary, you can still subdivide the managed space through the use of VSE/VSAM space classes.

Better space utilization can occur by freeing space at CLOSE under control of the DTF disposition parameter, or at end of job or jobstep using the DISP parameter of the DLBL statement.

Usability is improved, since the use of VSE/VSAM allows a single view of space management and a similar data format for all managed files created by SAM or VSE/VSAM.

Increased integrity is provided via the SHAREOPTION function of VSE/VSAM. Automatic close allows the last block of a blocked file to be written if a program failure occurs. Because VSE/VSAM is managing the the space, file overlaps do not occur.

Increased security is provided through the use of the password protection facility of VSE/VSAM.

Migration to VSE/VSAM is enhanced through the ability to access the SAM ESDS (CI format only) with VSE/VSAM for read, update, or create access.

Use of VSE/VSAM for space management allows functional extendability and allows the SAM user to benefit from enhancements to VSE/VSAM space management.

Most Access Method Service utility commands (for example, PRINT, EXPORT, IMPORT, REPRO, LISTCAT) apply to SAM in VSE/VSAM files, providing a single view of utilities for the managed SAM and the VSE/VSAM files.

Some SAM restrictions are removed for managed SAM. For example, DTFSD TYPEFLE=WORK may now be a multivolume file. The DEVICE specification of the DTF is ignored so that the program may run on any DASD.

Operator communications are simplified because there is no possibility of overlap on another file. Also, the operator is not requested to specify additional extents for the file.

Using the Support

Before a SAM ESDS can be created (written as a new file), the file must have been defined. This may be accomplished in several ways:

- You may explicitly define the SAM ESDS via the Access Method Services DEFINE command.
- You may provide the required information to allow implicit definition of the SAM ESDS at OPEN time.
- The file may already be defined (via explicit or implicit DEFINE) because of some previous use of the file (for example, a work file).

After the SAM ESDS has been DEFINEd, you may access the file via DTF and SAM imperative macros or via ACB and the VSE/VSAM imperative macros. In addition, most Access Method Services commands support a SAM ESDS. When you are finished with the file, the Access Method Services DELETE command may be used, the file may be implicitly deleted, or the file reset to "empty".

Functional Description

Define Space

The space managment function offered with this support, even for files that do not use VSE/VSAM as an access method, is based on the current VSE/VSAM space management. This means that space for SAM data spaces and the extents of the suballocated space are communicated between VSE/VSAM and the respective SAM access methods at OPEN of extent time. Sufficient VSE/VSAM data space has to be defined, even for SAM ESDS, before using the new space management.

The VSE/VSAM data spaces can be defined with the current VSE/VSAM DEFINE SPACE, DEFINE MASTERCATALOG or DEFINE USERCATALOG commands.

Defining a SAM ESDS

A SAM ESDS may be defined explicitly by using the current Access Method Services DEFINE CLUSTER command or implicitly, through SAM and VSAM OPEN functions. When explicit define is used to establish a SAM ESDS, the RECORDFORMAT parameter is required (described below) and signals that the ESDS is for a SAM ESDS in VSE/VSAM space.

In addition to the Access Method Services DEFINE command, you may cause a reusable ESDS to be implicitly defined by OPEN if the file has not been explicitly defined. If the file has been implicitly defined, the file may be implicitly deleted from the catalog by CLOSE, according to the disposition parameter on the DTF or the JCL statement. This function is only available with the SAM ESDS Feature of the VSE/VSAM Program Product. An error will result if the function is called without the required feature installed.

Implicit Define Cluster

Under certain conditions, the operations of the DEFINE CLUSTER command for a SAM ESDS are performed by VSE/VSAM OPEN, at the request of managed SAM OPEN for an output or work file, with the Access Method Services DEFINE CLUSTER command. This function is not supported for native VSE/VSAM files.

Restrictions

It is not possible to relate an alternate index to, or define a path over, a SAM ESDS.

The SPANNED parameter must not be specified with the RECORDFORMAT parameter. The SPEED parameter is required when RECORDFORMAT is specified. SPEED is the default for SAM ESDS. If RECOVERY is specified explicitly for a SAM ESDS, an error will occur.

The subparameter NOCIFORMAT is not permitted with the CONTROLINTERVALSIZE, ERASE, BUFFERSPACE, EXCEPTIONEXIT or WRITECHECK parameters of DEFINE.

Implicit DEFINE CLUSTER is requested under the following conditions: (1) If the SAM ESDS is not currently defined in the VSAM catalog (either via explicit or implicit define) and (2) You have specified sufficient information for the implicit DEFINE to occur on the VSAM JCL statements for the file.

The following information is required in addition to what is needed for access:

- Allocation information (either number of tracks/blocks on the first EXTENT statement or RECORDS and AVERECLN on the DLBL statement
- Volume information (either volume serial number on EXTENT statement or a default model for a SAM ESDS that has been previously defined

  Note: If you leave the volume serials unspecified (that is, use default volumes) and the default volume list contains CDK and FBA devices, VSE/VSAM will choose whether to use a CDK or a FBA device. If the allocation information is specified via tracks/blocks on the EXTENT statement, the amount of actual space varies greatly, depending on the device type used.

Therefore, if default volumes are used, it is recommended that allocation information be specified via RECORDS and AVERECLN.

Other JCL parameters that will be used for implicit define, if present, are:

- Retention period or expiration date (the JCL default of seven days is taken if there is no specification on the DLBL statement)
- Catalog specification via CAT=filename parameter

In addition, an implicit delete followed by an implicit define will occur under the following conditions:

- If a SAM ESDS of the same name (that is, file-id) had been implicitly defined and still exists in the VSE/VSAM catalog, and the file characteristics of the existing file do not match the access characteristics specified in the DTF. For example, the DTF BLKSIZE is greater than the maximum VSE/VSAM record size contained in the catalog entry for the file. (If the file characteristics do match the access characteristics, then the file may simply be reset or extended, as with DISP=OLD specified on the DLBL statement).
- If the existing file is either expired, or the operator has been informed of the unexpired duplicate file-id and has responded 'DELETE'
- If sufficient information has been supplied on the VSE/VSAM JCL statements

Note: An implicitly defined file cannot be password-protected; however, the catalog that the file is being implicitly defined into may be password-protected. In this case, the operator will be asked for the update or higher-level password of the catalog during implicit define and the master password of the catalog during implicit delete.

Single Extent Allocation

The use of a unique file-id prefix of the form DOS.WORKFILE.SYSxxx, where xxx is alphanumeric (for examples, 001 or LNK), will cause a single extent to be allocated for the primary allocation. An example of a file that requires a single extent is SYSLNK.

The file-id, 'DOS.WORKFILE.SYSLNK.FOR.BG', will cause a single extent to be allocated.

Considerations for Defining System Work Files

System work files should be reusable. In order to avoid DASD space usage when the work file is not in use, they should also be dynamic (that is, specify NOALLOCATION in addition to REUSE). It is also desirable to indicate that the work file is expired (this is the default) in order to avoid the operator message on each usage.

The file may be implicitly defined at OPEN and implicitly deleted at CLOSE. However, explicitly defined work files will give better performance. The catalog should not be password-protected since the operator will be requested to enter the password at implicit define and implicit delete.

The partition/CPU independent function may be used during the explicit define of a work file in order to make this file unique to a particular partition and CPU.

You should be careful to specify the maximum RECORDSIZE that the system program (for example, COBOL/VS) will use. If multiple system programs will use the same system work files, the maximum RECORDSIZE should be the greatest that any of the system programs will use.

VSE/VSAM for a SAM ESDS

Support Provided

VSE/VSAM will read, update and create SAM ESDS files. The formats supported are F, FB, V, VB, and U. VSE/VSAM Record Management provides the same capabilities for processing a SAM ESDS as for a VSE/VSAM ESDS.

The following restrictions apply:

- VSE/VSAM access of a non-CI format SAM ESDS (created via DTFPH) is not supported.
- No RECOVERY mode (VSE/VSAM will create a SAM ESDS in SPEED mode) exists.
- No extension of a SAM ESDS will be performed by VSE/VSAM access of a SAM ESDS.

  Note: When a VSE/VSAM ESDS has been CLOSEd (or TCLOSEd), the SPEED mode option is changed to RECOVERY mode. In addition, the remainder of the last used CA is preformatted with software end-of-files (SEOFs). VSE/VSAM extension of a VSAM ESDS is always done in RECOVERY mode, which assumes the preformatting. The CA format concept is not implemented for a SAM ESDS.
- No AIX support is provided.
- No SPANNED support is provided (SAM SPANNED records or VSE/VSAM SPANNED records).

RECORDFORMAT Parameter

RECORDFORMAT, a new subparameter, can be specified in the Access Method Services DEFINE command of a NONINDEXED CLUSTER. This parameter establishes the ESDS as a SAM ESDS. RECORDFORMAT is abbreviated 'RECFM'.

Its format is as follows:

```
                U
                F
  RECORDFORMAT (FB,logicalrecordsize)
                V
                VB
```

RECORDFORMAT can be specified at the CLUSTER or DATA level of DEFINE CLUSTER.

The ERASE parameter must not be specified in DEFINE or ALTER when RECORDFORMAT is specified. THE SPANNED parameter must not be specified with the RECORDFORMAT parameter. The RECOVERY parameter may not be specified when RECORDFORMAT is specified.

NOALLOCATION Parameter

This parameter in the DEFINE causes a file to be defined, but no space is allocated until the file is actually opened for output. This will allow you to explicitly define your system work files, to gain the performance advantage of explicitly defined files, and avoid the utilization of space when these files are not being used.

NOCIFORMAT Parameter

A new parameter, NOCIFORMAT, will be added to the DEFINE CLUSTER Access Method Services command with the NONINDEXED and RECORDFORMAT parameters. This new parameter is mutually exclusive with the CONTROLINTERVALSIZE and RECORDSIZE (DEFINE and ALTER) parameters and can be specified at the CLUSTER or DATA level of DEFINE CLUSTER. It is abbreviated 'NCIF'.

This parameter is intended for files in VSE/VSAM space that are processed by EXCP only and are not written in control interval format. Consequently, those files cannot be opened except with DTFPH.

PRINT

A file that has been defined in managed space can be printed by using the PRINT command and specifying the file name. Currently, ENVIRONMENT parameters are specified and SAM support is used to print the file via DTF. Current PRINT options for a SAM ESDS are supported. Note that ENVIRONMENT parameters will override catalog attributes.

This function is not supported for files that are not in SAM ESDS (that is, DTFPH non-CI format).

REPRO

It is possible to REPRO:

- An unmanaged SAM file into a SAM ESDS. This allows you to access the SAM ESDS as a managed file.
- A managed SAM ESDS into another managed SAM ESDS. This could be used to either create a backup copy of the file or move the file from managed space under one catalog to managed space under another catalog.
- A managed SAM ESDS into a VSE/VSAM file
- A VSE/VSAM file into a managed SAM ESDS
- A managed SAM ESDS into an unmanaged SAM file

The ENVIRONMENT parameter is necessary to access a managed SAM ESDS. Current REPRO options for a SAM ESDS are supported.

This function is not supported for files that are not in valid SAM format (that is, DTFPH non-CI format).

LISTCAT

Since space for managed SAM ESDS is ordinary VSE/VSAM data space, it may be displayed by using LISTCAT SPACE. All files that have been defined for a particular catalog can be displayed by using LISTCAT, including all SAM ESDS defined either explicitly or implicitly in managed space.

The ATTRIBUTES portion of LISTCAT output will modified as follows:

- CISIZE will indicate 'NOCIFORMAT' if NOCIFORMAT was specified on either an explicit or implicit define.
- CI/CA will indicate 'NOCIFORMAT' if NOCIFORMAT was specified on either an explicit or implicit define.
- SAMLRECL is a new attribute that will contain the SAM logical record length.
- RECFORM (xx) is a new attribute that will contain the SAM record format as specified in the new field (AMDRCFRM) of the AMDSB.
- If the file was implicitly defined, the new attribute of 'IMPLICITLY-DEFINED' will be listed; otherwise, 'EXPLICITLY-DEFINED' will be listed.
- If the file is a SAM ESDS, the new 'SAM-DATA-SET' attribute will be listed; otherwise, the 'VSAM-DATA-SET' attribute will be listed.

Statistics are not maintained for a SAM ESDS in VSE/VSAM space; therefore, they are not printed.

ALTER

The ALTER command will work on a SAM ESDS defined in managed space in exactly the same manner as a VSE/VSAM file with no additional support.

EXPORT/EXPORTRA

EXPORT(RA) provides full support for CI-format SAM ESDS files. Non-CI format files are not supported by EXPORT(RA) and will be bypassed with an appropriate message.

IMPORT/IMPORTRA

IMPORT(RA) provides full support for those files which may be EXPORTed or EXPORTRAed.

DELETE

The DELETE command can delete a managed SAM file. ERASE is not supported.

BLDINDEX

An Alternate Index cannot be defined against a SAM ESDS.

VERIFY

VERIFY does not apply to a managed SAM ESDS.

LISTCRA/RESETCAT

LISTCRA and RESETCAT can be used to validate and recover a catalog with SAM ESDS entries.

OPEN/CLOSE/END OF VOLUME

SAM ESDS OPEN/CLOSE/End of Volume

In order to OPEN a SAM ESDS you must specify 'VSAM' on the DLBL/EXTENT statement associated with the file.

The device type specification and any symbolic unit specified in the DTF are always ignored.

The general technique used to open a SAM ESDS is to build an ACB and a SAM parameter list from the information in the DTF/JCL, OPEN the ACB, retrieve extent information from the VSE/VSAM blocks, and store it into the DTF/DTF extension. The proper IOMOD is located in the SVA and a pointer is stored in the DTF.

SAM access of a file created by VSE/VSAM is not possible because SAM cannot handle files in CA format.

DTFPH MOUNTED=SINGLE

For DTFPH, EXCP is used to read and write data in the managed files. If you intend to create a managed file via EXCP and then read it with SAM, the data must be written in SAM ESDS (CI) format. To indicate that the data is to be written in CI format, CISIZE=n must be specified in the DTF, where n is a non-zero value that will be rounded to the next valid CI size. If CISIZE is not specified, the file is assumed to be in non-CI format and is not accessible by SAM. Note that this is true for a managed file on both FBA and CKD. If the data is not written in CI format, the CISIZE parameter of the DTF should not be specified.

If you want a CKD DTFPH-created file to be accessible by SAM, you should be aware that the SAM data format is different, depending on whether or not the file is managed; therefore, you should always run either in managed or unmanaged space. If you want to support both, you should determine whether or not you are in managed space, so that you may write the correct format. When you want to support managed space, a bit in the version III extension of the DTF, will be set. This bit may be used to determine whether of not you are in managed space. This situation does not occur on FBA since managed and unmanaged data formats are the same.

At CLOSE time, if a Version III extension exists, the bit indicating managed space will be turned off.

DTFPH MOUNTED=SINGLE OPEN Processing

To ensure that correct OPEN processing occurs, SAM OPEN will take the following steps for DTFPH output when CISIZE is specified and 'VSAM' is encountered on the DLBL statement:

- An ACB will be created from the DTFPH, specifying a MACRF=(ADR,SEQ,RST).
- A parameter list will be created indicating the CISIZE; the maximum RECORDSIZE will be set to CISIZE-7; and the RECORDSIZE FORMAT, unknown, will be set to 'U'.
- 'Implicit DEFINE allowed' will be set.
- 'Reset by expiration date' will be set.
- 'Non-CA format' will be set.
- VSE/VSAM OPEN is issued.
- If OPEN is successful, the high CI number in storage will be set to one (the second CI of the file).
- An empty CI followed by a SEOF will be written into the file.

These steps will alow a subsequent SAM OPEN output to properly fail (because of a nonzero high used RBA) if the file is unexpired. Note that it is the EXCP user's responsibility to ensure that there is a SEOF at the end of the file.

If CISIZE is not specified in DTFPH, the first six steps will be executed, except that CISIZE is specified as zero and no maximum RECORDSIZE will be specified.

Access to VSE/VSAM Managed Space

Data Format

All data written by SAM logic modules in VSE/VSAM managed space will be recorded in Control Interval (CI) format. Any data to be read from VSE/VSAM managed space by SAM must have been recorded in the CI Format. If SAM logic modules encounter invalid CIs, the results will be unpredictable. The CI size and other file attributes will be determined at DEFINE time and maintained in the VSE/VSAM catalog. OPEN ensures that any conflicts between the DTF and file attributes in the catalog are resolved before the file can be accessed.

The SAM logic modules do all that is necessary to map logical records, as seen by the problem program, to the CIs. The SAM stored record format will be maintained (using the SAM Service Routine), even though the data is recorded in CI format. SAM blocking structure is maintained as it existed in unmanaged space. The term "logical block" is used here to describe that set of logical records which makes up a SAM physical block for unmanaged CKD files.

Each SAM logical block (for example, a fixed-length block if RECFORM=FIXBLK; a variable-length block if RECFORM=VARBLK) is handled as a record in the CI. If the CI is a size that will hold more than one logical block, the support will automatically place as many logical blocks with associated control information in the CI as possible. Any leftover space in the CI which was insufficient to contain the next logical block and its control information remains unused. Logical blocks that are too large to fit in the maximum CI size (32K) are not supported. Logical blocks will not span CIs.

The mapping of SAM logical blocks to CIs in VSE/VSAM managed space is the same as the mapping used to provide SAM support for FBA devices. Refer to Chapter 3 for a graphic representation of the mapping. Note that all data recorded in VSE/VSAM managed space by SAM (FBA devices or CKD devices) have logical blocks mapped into CIs.

In mapping CIs over the DASD device, CIs are allowed to cross cylinder boundaries. The SAM logic modules depend on OPEN to ensure that each extent is at least large enough to contain one CI and to round the extent upper boundary to the next lower integral CI boundary for FBA.

SAM logic modules are not sensitive to Control Area (CA) boundaries so that files created via the SAM I/O modules will not contain gaps to align CIs to begin on CA boundaries. Files to be read by the logic modules must not contain gaps, to provide for CA boundary alignment; therefore, SAM will not be able to access VSE/VSAM created files.

The data format differences described so far apply only to CKD unmanaged files as compared to managed files (FBA and CKD). The following differences in format exist between all managed and unmanaged files:

- User labels are not supported. They are never present in a SAM ESDS file.
- SAM spanned record format is not supported. If RECFORM=SPNUNB or SPNBLK, the job is cancelled.

SAM Access

SAM access to VSE/VSAM managed space is achieved by using DTFSD and its corresponding imperative macros. Since the data is recorded in CI format, the DTF expansion and the DTF interface to the user is very similar to that provided for FBA device support. The logic modules to support these DTFs with VSE/VSAM managed space will be functional superset modules assembled and linkedited before system distribution. They will be loaded into SVA at IPL time. MODNAME in the DTF will be ignored. At OPEN time, catalog information related to the file is checked against the DTF characteristics for validity.

OPEN ignores the DEVICE= parameter of the DTF and retrieves device characteristics from the catalog. This provides for DASD independence to the problem program.

The CI data format is applicable to CKD devices as well as FBA devices when the space is VSE/VSAM-managed. This means that DTF interface characteristics that changed for FBA device support now apply to CKD devices as well, if the space on those devices is managed by VSE/VSAM. Examples of such changes are:

- Actual I/O not necessarily related to processing of a physical block
- Error detection on a CI basis instead of a physical block basis
- I/O overlap possible only for output or strict input files with two IOAREAs and IOREG specified
- Non-synchronous maintenance of CCB in DTF

Use of CI format on CKD also implies a continual ignoring of CNTRL=YES since CIs may cross cylinders on CKD devices and problem programs would be unable to know when CNTRL will provide performance advantage. The RPS feature available on certain CKD devices will be used by the SAM logic modules. Position identifiers, as provided by NOTE and required by POINT macros, will consist of relative CI number and relative record number as in SAM FBA device support, even on CKD devices.

In processing data in VSE/VSAM managed space, the SAM logic modules are not sensitive to Control Areas (CA) as is VSE/VSAM. In writing data, all the space in an extent is treated as a linear continuous data area into which CIs may be mapped. The SAM logic modules depend on a dense mapping of CIs over any extent that is successfully OPENed for input. Attempts to retrieve files that contain invalid CIs and a valid CISIZE in the catalog using the SAM logic modules will produce unpredictable results.

The SAM logic modules maintains an internal high CI number for each format-type write operation. This high CI number is converted to a high used RBA and is stored in the catalog at CLOSE to keep track of the logical end-of-file.

FEOVD will ignored in VSE/VSAM managed space to be consistent with the volume independence objective and no SEOFs will be embedded in a file.

RECFORM=SPNUNB|SPNBLK is not supported in VSE/VSAM managed space.

The SAM logic modules support empty CIs (CIDF points to the front of the CI as the beginning of free space) on input to facilitate data interchangeability.

Files created by VSE/VSAM are not accessible by SAM because VSE/VSAM files are in CA format.

Figure D.01 gives the storage and working set requirements for SAM access.

SD	DOS/VS Unmanaged CKD Non RPS	DOS/VS Unmanaged CKD RPS	DOS/VSE Unmanaged CKD	DOS/VSE Unmanaged FBA	DOS/VSE Managed CKD	DOS/VSE Managed FBA
VIRTUAL						
CODE	2-4K	2-4K	S	6-8K	8-10K	S
NON-CODE	2K	4K	A	4K	4K	A
TOTAL	4-6K	6-8K	M	10-12K	12-14K	M
			E			E
						AS
WORKING SET			A			UN
CODE	2K	2K	S	4K	6K	M
NON-CODE	2K	4K		4K	4K	A
TOTAL	4K	6K	D	8K	10K	N
			O			A
			S			G
			/			E
			V			D
			S			FBA

These figures include the logic module, DTF, DTF extension, plus the SSR used by logic module for FBA and managed CKD. IOAREAs and work areas are not included. For FBA and managed CKD, the CI buffer should also be added (the CISIZE can be from 512 to 32K bytes).

• Figure D.01: VSE/VSAM SAM Space Management Storage Requirements

Data Transfer and I/O Overlap

The movement of data in the implementation of CI blocking has been minimized whenever possible. For example, when SD data file users specify WORKA=YES in the DTF, no extra moving of data is required since the data is moved from/to the users' work area to/from the CI buffer. There are situations, however, where the extra movement of data beyond what is required for unmanaged CKD support cannot be avoided. For example, with SD work files the unmanaged CKD support initializes channel programs so that the channel places the logical blocks directly in the user-specified work area. For managed CKD and FBA support, the channel places the CI into the CI buffer from which the logical blocks are deblocked as READ/CHECK requests are processed. This extra move of the data will show up as a throughput degradation if other performance factors are equal.

The CISIZE parameter in the DTF may be used to control interaction of working set size, use of EXCP, and (consequently) average path lengths for accessing logical records. If the CISIZE is such that only one logical block fits into the CI, the number of EXCPs required to access a file is essentially the same as required for unmanaged CKD devices, and the increased working set and path lengths will probably result in some performance degradation. If the CISIZE is large enough to contain two or more logical blocks per CI, the decreased use of EXCP will compensate, to some extent, for the increased working set and path

lengths.

Increasing CISIZE can also overcome the degradation caused by the extra data movement. If the SD work files example cited in this section uses logical records of 80 bytes, the minimum CISIZE of 512 bytes would accommodate six logical blocks per CI and the extra data movement would be compensated by the reduced usage of EXCP.

Indiscriminately increasing ClSIZE in a heavy multiprogramming environment must be avoided or system throughput may be adversely affected by excessive paging activity.

The specification of two IOAREAs for SAM data files will not result in I/O overlap in all cases in which it ocurrs for unmanaged CKD devices. If WORKA=YES was specified, all user-provided IOAREAs are ignored in order to minimize movement of data. If UPDATE=YES is specified, no overlap is attempted because the SSR routine works with only one CI buffer. In other cases in which two IOAREAs were specified, I/O overlap will be achieved via the asynchronous interface between the logic module and the SSR in use.

Access of a SAM ESDS

The main difference between the access of a managed SAM ESDS and an unmanaged SAM file is the JCL to be specified. Figure D.02 is a description of the JCL parameters that apply to accessing a managed SAM ESDS.

DLBL

Filename	- The symbolic name of the program DTF.
'File-id'	- The unique name associated with the file in the VSE/VSAM catalog.
Date	- If the file has been previously defined (explicitly or implicitly), this parameter is ignored.
	- If the file will be implicitly defined at OPEN time, this parameter specifies the retention period that will be recorded in the VSE/VSAM catalog.
Codes	- VSAM must be specified to indicate that the SAM ESDS is in VSE/VSAM space.
DSF	- This operand is ignored. VSE/VSAM files are always Data Secured Files.
BUFSP=n	- This operand is not meaningful for SAM access. Bufferspace is determined by the number of IOAREAs specified in the DTF.
CAT=Filename	- This operand has the same meaning as the current VSE/VSAM access of a VSE/VSAM file.
BLKSIZE=n	- This operand is invalid for a VSE/VSAM DLBL.
CISIZE=n	- This operand is invalid for a VSE/VSAM DLBL.
DISP=	- This operand allows you to specify the actions that OPEN and CLOSE are to take pertaining to the file. A file may be deleted at CLOSE time even though the DTF TYPEFLE=WORK did not specify DELETFL=YES.
RECORDS=	- This operand is used in place of the tracks/blocks parameter of the EXTENT statement for space allocation.
AVERECLN=	- This operand also is used for space allocation and requires the RECORDS= parameter.

• Figure D.02: DLBL Parameters for SAM ESDS

EXTENT (Can be omitted entirely if using explicitly defined files)

Symbolic Unit	- This operand is required and overrides any DTF DEVADDR specification. Only programmer logical units (or SYSLNK) are supported for managed SAM ESDS.
Serial Number	- This operand is required. If the file has been previously defined, an EXTENT statement must be provided for each volume specified in the VSE/VSAM catalog entry for this file. If the file will be implicitly defined at OPEN time, each serial number in the EXTENT statments will be entered into the VSE/VSAM catalog entry created for this file.
Type	- Type 1 (data area-no split cylinder) is the only valid parameter (may be omitted).
Sequence Number	- This operand is ignored.
Relative Track or Block	- This operand is ignored. If the file has been previously defined, VSE/VSAM will locate the file. If the file is to be implicitly defined at OPEN time, VSE/VSAM will choose the location of the file.
Number of Tracks or Blocks	- If the file has been previously defined the allocation size of the file is contained in the VSE/VSAM catalog and this parameter is ignored. If the file will be implicitly defined at OPEN time, this parameter specifies the primary allocation size for the file. Note that the primary allocation is not necessarily a single extent. A single extent may be forced by the file-id specification. A secondary allocation size of 20 percent (rounded to the next integral number of tracks or blocks) will be assumed.
Split Cyl. Track	- This parameter is invalid for a VSE/VSAM DLBL.
Bins	- This parameter is invalid for a VSE/VSAM DLBL.

• Figure D.03: EXTENT Parameters for SAM ESDS

A file that is implicitly defined is not allocated space until an OPEN occurs. At CLOSE time, if the DTF is a DTFSD TYPEFLE=WORK with DELETFL=YES, the file is implicitly deleted and the allocated space is returned to VSE/VSAM. No other DTF-type CLOSE causes implicit delete because current support allows the possibility of OPENing the file at a later time for INPUT. The file will not be implicitly deleted until a subsequent OPEN for OUTPUT occurs for the same file-id. At this time, the file is deleted (if expired or through operator response).

A file may be deleted via the Access Method Services DELETE command. This should be done periodically against expired SAM ESDSs to prevent depletion of VSE/VSAM space.

The DTF DEVICE and DEVADDR parameters are ignored for a managed SAM ESDS.

The DTF CISIZE parameter is rounded to the next higher valid CI size in a manner similar to the CONTROLINTERVALSIZE parameter of the explicit DEFINE CLUSTER command of Access Method Services.

The SIZE parameter of the EXEC statement must be used, as in VSE/VSAM access, or the default GETVIS area for the partition must be large enough for VSE/VSAM use.

Transferring Unmanaged SAM Data into a SAM ESDS

In order to get data into VSE/VSAM managed space, the data must be moved. This is accomplished by using REPRO (with ENVIRONMENT parameters) with the output parameters indicating a file that may be either explicitly or implicitly defined.

Data Interchange Considerations

Managed and Unmanaged SAM

There is no facility to "surround" a SAM file with VSE/VSAM space; therefore, steps must be taken to move SAM data into VSE/VSAM managed space. A managed SAM ESDS may not be processed under a prior release of DOS/VS without first REPROing the data into an unmanaged space.

Managed SAM and VSE/VSAM

All files created by SAM in managed space will be written in non-CA format. SAM logic modules will not handle files created in CA format (that is, files created by VSE/VSAM). VSE/VSAM will, however, process a SAM ESDS.

Integrity and Security

Improved SAM Integrity

A SAM ESDS has increased integrity over native SAM in the following ways:

- VSE/VSAM controls the space that is available for allocation. A file cannot be overlapped because extent limits are never specified.
- In general, when a SAM ESDS is opened for output, it is reset. VSE/VSAM will not reset a file when it is in use for either input or output. It is not possible, therefore, to delete a file while another user has the same file open for access, as can occur with native SAM. When the file is not reset (for examples, a work file, input) VSE/VSAM controls access to the file according to the share option specified or defaulted. The default is SHROPT 1 which allows only one output user at a time to have the file OPEN.
- If a SAM ESDS has not been CLOSEd by the end of the job step, VSE/VSAM will attempt to CLOSE it. This will ensure that any SAM write requests will be written to the device (for example, in the case in which the data is in SAM blocked format).

A specification of SHAREOPTION 4 on an explicit define of a SAM ESDS is ignored, as it is for a VSE/VSAM ESDS. Track/block hold is controlled by the HOLD=YES specification in the DTF (but does not require a supervisor with TRACK HOLD feature since LOCK/UNLOCK service is used).

Security

VSE/VSAM password protection is provided to the SAM ESDS user via operator response. Explicit DEFINE is required to password-protect a file.

This Page Intentionally Left Blank

The following chart is provided to assist you in choosing the proper CI size on the basis of the desired record size (for a new file) or the existing CKD physical block size (logical block).

To use the charts:

- Your exact logical block size may not appear in the chart. Locate the column under the smallest logical block size number that is equal to or greater than your logical block size.
- This column of figures is the number of logical blocks that fits into a single FBA block of 512 bytes.
- The largest number in the column indicates the CI size that is most efficient in utilizing the FBA DASD.
- To determine the number of FBA blocks to be allocated to the file, simply divide the desired number of records by the number found in the column. This number of blocks should be rounded to the next higher CI boundary and to that number you should add one additional CI for the SEOF (software end-of-file).

For example, assume that your current logical block size is 1000 bytes. By looking at the table on page 161, you find the column that is headed by the logical block size of 1014. There are several CI sizes from which to choose. You must now decide how much processor storage you wish to devote to the I/O area for processing this file. Assume you pick a CI size of 8192. Assume also that space is needed for 2500 logical blocks. The calculations would be as follows:

```
FBA blocks required = 2500 / 0.50 = 5000.
FBA blocks per CI = 8192 / 512 = 16.
5000 rounded to next higher CI boundary = 5008.
One additional CI must be added for SEOF.
Total FBA blocks to be allocated = 5008 + 16 = 5024.
```

Note: The following chart assumes fixed-length records. To use the chart for variable-length records, search for the smallest logical block size which is greater than or equal to your average logical block plus three bytes.

CKD Calculation Formulas

The formulas in Figure E.01 through Figure E.05 are given to assist you in calculating the number of records within a given extent when all that is known are the extent and the logical block size.

IBM 2311 Track Capacity Formula

The following information is needed to determine the number of logical blocks per track:

- Bytes per Record (BPR) = (537 * (KL + DL) / 512) + C + 61.
- Bytes in Last Record (BLR) = KL + DL + C.
- Records per Track (RPT) = ((3625 - BLR) / BPR) + 1.

Where: KL = Key Length.
DL = Data Length.
C = 0 when KL = 0.
C = 20 when KL ≠ 0.

- Figure E.01: 2311 Track Capacity Formula

Truncate any fractions in BPR and RPT. Multiply RPT by the number of tracks in the extent to obtain the number of logical blocks in the extent.

Example: Assume that the current logical block is 256 bytes without keys and the current file occupies 100 tracks. The calculations would then be:

BPR = (537 * (0 + 256) / 512) + 0 + 61 = 329.
BLR = 0 + 256 + 0 = 256
RPT = ((3625 - 256) / 329) + 1 = 3369 / 329 + 1 = 10 + 1 = 11.

The number of logical blocks in the extent would be: 100 * 11 = 1100.

IBM 2314 Track Capacity Formula

The following information is needed to determine the number of logical blocks per track:

- Bytes per Record (BPR) = (2137 * (KL + DL) / 2048) + C + 101.
- Bytes in Last Record (BLR) = KL + DL + C.
- Records per Track (RPT) = ((7294 - BLR) / BPR) + 1.

Where: KL = Key Length.
DL = Data Length.
C = 0 when KL = 0.
C = 45 when KL ≠ 0.

- Figure E.02: 2314 Track Capacity Formula

Truncate any fractions in BPR and RPT. Multiply RPT by the number of tracks in the extent to obtain the number of logical blocks in the extent.

Example: Assume that the current logical block is 256 bytes without keys and the current file occupies 200 tracks. The calculations would then be:

BPR = (2137 * (0 + 256) / 2048) + 0 + 101 = 368.
BLR = 0 + 256 + 0 = 256.
RPT = ((7294 - 256) / 368) + 1 = 7038 / 368 + 1 = 19 + 1 = 20.

The number of logical blocks in the extent would be: 200 * 20 = 4000.

IBM 3330 Track Capacity Formula

The following information is needed to determine the number of logical blocks per track:

- Bytes per Record (BPR) = KL + DL + C.
- Records per Track (RPT) = 13165 / BPR.

Where: KL = Key Length.
DL = Data Length.
C = 135 when KL = 0.
C = 191 when KL ≠ 0.

- Figure E.03: 3330 Track Capacity Formula

Truncate any fraction in RPT. Multiply RPT by the number of tracks in the extent to obtain the number of logical blocks in the extent.

Example: Assume that the current logical block is 1024 bytes without keys and the current file occupies 190 tracks. The calculations would then be:

BPR = 0 + 1024 + 135 = 1159.
RPT = 13164 / 1159 = 11.

The number of logical blocks in the extent would be: 190 * 11 = 2090.

IBM 3340/3344 Track Capacity Formula

The following information is needed to determine the number of logical blocks per track:

- Bytes per Record (BPR) = KL + DL + C.
- Records per Track (RPT) = 8535 / BPR.

Where: KL = Key Length.
DL = Data Length.
C = 167 when KL = 0.
C = 242 when KL ≠ 0.

- Figure E.04: 3340/3344 Track Capacity Formula

Truncate any fraction in RPT. Multiply RPT by the number of tracks in the extent to obtain the number of logical blocks in the extent.

Example: Assume that the current logical block is 1024 bytes without keys and the current file occupies 120 tracks. The calculations would then be:

BPR = 0 + 1024 + 167 = 1191.
RPT = 8535 / 1191 = 7.

The number of logical blocks in the extent would be: 120 * 7 = 840.

IBM 3350 Track Capacity Formula

The following information is needed to determine the number of logical blocks per track:

- Bytes per Record (BPR) = KL + DL + C.
- Records per Track (RPT) = 19254 / BPR.

Where: KL = Key Length.
DL = Data Length.
C = 185 when KL = 0.
C = 267 when KL ≠ 0.

- Figure E.05: 3350 Track Capacity Formula

Truncate any fraction in RPT. Multiply RPT by the number of tracks in the extent to obtain the number of logical blocks in the extent.

Example: Assume that the current logical block is 1024 bytes without keys and the current file occupies 300 tracks. The calculations would then be:

BPR = 0 + 1024 + 185 = 1209.
RPT = 19254 / 1209 = 15.

The number of logical blocks in the extent would be: 300 * 15 = 4500.

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	50	51	52	53	54	55	56	57	58
512	10.00	9.00	9.00	9.00	9.00	9.00	8.00	8.00	8.00
1024	10.00	9.50	9.50	9.50	9.00	9.00	9.00	8.50	8.50
1536	10.00	9.66	9.66	9.33	9.33	9.00	9.00	8.66	8.66
2048	10.00	9.75	9.75	9.50	9.25	9.25	9.00	8.75	8.75
2560	10.20	10.00	9.80	9.60	9.40	9.20	9.00	8.80	8.60
3072	10.16	10.00	9.66	9.50	9.33	9.16	9.00	8.83	8.66
3584	10.14	10.00	9.71	9.57	9.42	9.14	9.00	8.85	8.71
4096	10.12	10.00	9.75	9.62	9.37	9.25	9.00	8.87	8.75
4608	10.11	10.00	9.77	9.55	9.44	9.22	9.11	8.88	8.77
5120	10.20	10.00	9.80	9.60	9.40	9.20	9.10	8.90	8.80
5632	10.18	10.00	9.81	9.63	9.45	9.27	9.09	8.90	8.72
6144	10.16	10.00	9.75	9.58	9.41	9.25	9.08	8.91	8.75
6656	10.15	10.00	9.76	9.61	9.46	9.23	9.07	8.92	8.76
7168	10.21	10.00	9.78	9.64	9.42	9.28	9.07	8.92	8.78
7680	10.20	10.00	9.80	9.60	9.46	9.26	9.06	8.93	8.80
8192	10.18	10.00	9.81	9.62	9.43	9.25	9.12	8.93	8.81
10240	10.20	10.00	9.80	9.65	9.45	9.30	9.10	8.95	8.80
12288	10.20	10.00	9.83	9.62	9.45	9.29	9.12	8.95	8.79
14336	10.21	10.00	9.82	9.64	9.46	9.28	9.10	8.96	8.82
16384	10.21	10.03	9.81	9.62	9.46	9.28	9.12	8.96	8.81
18432	10.22	10.02	9.83	9.63	9.47	9.27	9.11	8.97	8.80
20480	10.22	10.02	9.82	9.65	9.47	9.30	9.12	8.97	8.80
22528	10.22	10.02	9.84	9.63	9.47	9.29	9.13	8.97	8.81
24576	10.22	10.02	9.83	9.64	9.45	9.29	9.12	8.95	8.81
26624	10.23	10.01	9.82	9.65	9.46	9.28	9.13	8.96	8.80
28672	10.23	10.03	9.83	9.64	9.46	9.30	9.12	8.96	8.82
30720	10.23	10.03	9.83	9.65	9.46	9.30	9.13	8.96	8.81
32768	10.23	10.03	9.82	9.65	9.46	9.29	9.12	8.96	8.81

CI Size	Logical Block Size								
	59	60	61	62	63	64	65	66	67
512	8.00	8.00	8.00	8.00	7.00	7.00	7.00	7.00	7.00
1024	8.50	8.00	8.00	8.00	8.00	7.50	7.50	7.50	7.50
1536	8.33	8.33	8.33	8.00	8.00	7.66	7.66	7.66	7.33
2048	8.50	8.25	8.25	8.00	8.00	7.75	7.75	7.50	7.50
2560	8.60	8.40	8.20	8.20	8.00	7.80	7.80	7.60	7.60
3072	8.50	8.50	8.33	8.16	8.00	7.83	7.83	7.66	7.50
3584	8.57	8.42	8.28	8.14	8.00	7.85	7.71	7.71	7.57
4096	8.62	8.50	8.25	8.12	8.00	7.87	7.75	7.62	7.50
4608	8.55	8.44	8.33	8.22	8.00	7.88	7.77	7.66	7.55
5120	8.60	8.50	8.30	8.20	8.10	7.90	7.80	7.70	7.60
5632	8.63	8.45	8.36	8.18	8.09	7.90	7.81	7.72	7.54
6144	8.58	8.50	8.33	8.16	8.08	7.91	7.83	7.66	7.58
6656	8.61	8.46	8.30	8.23	8.07	7.92	7.84	7.69	7.61
7168	8.64	8.50	8.35	8.21	8.07	7.92	7.85	7.71	7.57
7680	8.66	8.46	8.33	8.20	8.06	7.93	7.86	7.73	7.60
8192	8.62	8.50	8.37	8.18	8.06	7.93	7.81	7.68	7.62
10240	8.65	8.50	8.35	8.25	8.10	7.95	7.85	7.75	7.60
12288	8.66	8.50	8.37	8.25	8.08	7.95	7.83	7.75	7.62
14336	8.64	8.50	8.35	8.25	8.10	7.96	7.85	7.75	7.60
16384	8.65	8.50	8.37	8.25	8.09	7.96	7.84	7.75	7.62
18432	8.66	8.52	8.38	8.25	8.11	7.97	7.86	7.75	7.61
20480	8.65	8.52	8.37	8.25	8.10	7.97	7.85	7.75	7.62
22528	8.65	8.52	8.38	8.25	8.11	7.97	7.86	7.75	7.63
24576	8.66	8.52	8.37	8.25	8.10	7.97	7.85	7.75	7.62
26624	8.67	8.51	8.38	8.25	8.11	7.98	7.86	7.75	7.63
28672	8.66	8.51	8.37	8.25	8.10	7.98	7.85	7.75	7.62
30720	8.66	8.51	8.38	8.25	8.11	7.98	7.86	7.75	7.63
32768	8.67	8.51	8.39	8.25	8.10	7.98	7.85	7.75	7.62

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	68	69	70	71	72	73	74	75	76
512	7.00	7.00	7.00	7.00	6.00	6.00	6.00	6.00	6.00
1024	7.00	7.00	7.00	7.00	7.00	6.50	6.50	6.50	6.50
1536	7.33	7.33	7.00	7.00	7.00	6.66	6.66	6.66	6.66
2048	7.25	7.25	7.25	7.00	7.00	6.75	6.75	6.75	6.50
2560	7.40	7.20	7.20	7.00	7.00	6.80	6.80	6.80	6.60
3072	7.50	7.33	7.16	7.16	7.00	6.83	6.83	6.66	6.66
3584	7.42	7.28	7.28	7.14	7.00	6.85	6.85	6.71	6.71
4096	7.50	7.37	7.25	7.12	7.00	6.87	6.87	6.75	6.62
4608	7.44	7.33	7.22	7.11	7.00	6.88	6.88	6.77	6.66
5120	7.50	7.40	7.30	7.10	7.00	7.00	6.90	6.80	6.70
5632	7.45	7.36	7.27	7.18	7.09	7.00	6.81	6.72	6.63
6144	7.50	7.33	7.25	7.16	7.08	7.00	6.83	6.75	6.66
6656	7.46	7.38	7.23	7.15	7.07	7.00	6.84	6.76	6.69
7168	7.50	7.35	7.28	7.14	7.07	7.00	6.85	6.78	6.71
7680	7.46	7.40	7.26	7.20	7.06	7.00	6.86	6.80	6.66
8192	7.50	7.37	7.25	7.18	7.06	7.00	6.87	6.81	6.68
10240	7.50	7.40	7.30	7.20	7.10	7.00	6.90	6.80	6.70
12288	7.50	7.37	7.29	7.16	7.08	7.00	6.87	6.79	6.70
14336	7.50	7.39	7.28	7.17	7.07	7.00	6.89	6.82	6.71
16384	7.50	7.40	7.28	7.18	7.09	7.00	6.90	6.81	6.71
18432	7.50	7.38	7.30	7.19	7.08	7.00	6.88	6.80	6.72
20480	7.52	7.40	7.30	7.20	7.10	7.00	6.90	6.80	6.72
22528	7.52	7.40	7.29	7.20	7.09	7.00	6.90	6.81	6.72
24576	7.52	7.41	7.29	7.20	7.10	7.00	6.89	6.81	6.72
26624	7.51	7.40	7.30	7.19	7.09	7.00	6.90	6.80	6.73
28672	7.51	7.41	7.30	7.19	7.10	7.00	6.91	6.82	6.73
30720	7.51	7.41	7.30	7.20	7.10	7.00	6.91	6.81	6.73
32768	7.51	7.40	7.29	7.20	7.09	7.00	6.90	6.81	6.73

CI Size	Logical Block Size								
	77	78	79	80	81	82	83	84	85
512	6.00	6.00	6.00	6.00	6.00	6.00	6.00	5.00	5.00
1024	6.50	6.50	6.00	6.00	6.00	6.00	6.00	6.00	5.50
1536	6.33	6.33	6.33	6.33	6.00	6.00	6.00	6.00	5.66
2048	6.50	6.50	6.25	6.25	6.25	6.00	6.00	6.00	5.75
2560	6.60	6.40	6.40	6.20	6.20	6.20	6.00	6.00	6.00
3072	6.50	6.50	6.33	6.33	6.16	6.16	6.00	6.00	6.00
3584	6.57	6.42	6.42	6.28	6.28	6.14	6.14	6.00	6.00
4096	6.62	6.50	6.37	6.37	6.25	6.12	6.12	6.00	6.00
4608	6.55	6.44	6.44	6.33	6.22	6.22	6.11	6.00	6.00
5120	6.60	6.50	6.40	6.30	6.30	6.20	6.10	6.00	6.00
5632	6.63	6.54	6.45	6.36	6.27	6.18	6.09	6.00	6.00
6144	6.58	6.50	6.41	6.33	6.25	6.16	6.08	6.08	6.00
6656	6.61	6.53	6.46	6.38	6.30	6.23	6.15	6.07	6.00
7168	6.57	6.50	6.42	6.35	6.28	6.21	6.14	6.07	6.00
7680	6.60	6.53	6.46	6.33	6.26	6.20	6.13	6.06	6.00
8192	6.62	6.50	6.43	6.37	6.31	6.18	6.12	6.06	6.00
10240	6.60	6.55	6.45	6.35	6.30	6.20	6.15	6.05	6.00
12288	6.62	6.54	6.45	6.37	6.29	6.20	6.12	6.08	6.00
14336	6.64	6.53	6.46	6.39	6.28	6.21	6.14	6.07	6.00
16384	6.62	6.53	6.46	6.37	6.31	6.21	6.15	6.06	6.00
18432	6.63	6.55	6.47	6.38	6.30	6.22	6.13	6.08	6.00
20480	6.62	6.55	6.47	6.37	6.30	6.22	6.15	6.07	6.00
22528	6.63	6.54	6.47	6.38	6.31	6.22	6.15	6.09	6.00
24576	6.64	6.54	6.45	6.39	6.31	6.22	6.14	6.08	6.02
26624	6.63	6.55	6.46	6.38	6.30	6.23	6.15	6.07	6.01
28672	6.64	6.55	6.46	6.39	6.30	6.23	6.16	6.08	6.01
30720	6.63	6.55	6.46	6.38	6.31	6.23	6.16	6.08	6.01
32768	6.64	6.54	6.46	6.39	6.31	6.23	6.15	6.07	6.01

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	86	87	88	89	90	91	92	93	94
512	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00
1024	5.50	5.50	5.50	5.50	5.50	5.50	5.50	5.00	5.00
1536	5.66	5.66	5.66	5.66	5.33	5.33	5.33	5.33	5.33
2048	5.75	5.75	5.75	5.50	5.50	5.50	5.50	5.25	5.25
2560	5.80	5.80	5.60	5.60	5.60	5.60	5.40	5.40	5.40
3072	5.83	5.83	5.66	5.66	5.66	5.50	5.50	5.33	5.33
3584	5.85	5.85	5.71	5.71	5.57	5.57	5.42	5.42	5.42
4096	5.87	5.75	5.75	5.62	5.62	5.50	5.50	5.37	5.37
4608	5.88	5.77	5.77	5.66	5.66	5.55	5.44	5.44	5.33
5120	5.90	5.80	5.80	5.70	5.60	5.60	5.50	5.40	5.40
5632	5.90	5.81	5.72	5.72	5.63	5.54	5.54	5.45	5.36
6144	5.91	5.83	5.75	5.66	5.66	5.58	5.50	5.41	5.41
6656	5.92	5.84	5.76	5.69	5.61	5.61	5.53	5.46	5.38
7168	5.92	5.85	5.78	5.71	5.64	5.57	5.50	5.42	5.42
7680	5.93	5.86	5.80	5.73	5.66	5.60	5.53	5.46	5.40
8192	5.93	5.87	5.75	5.68	5.62	5.56	5.50	5.43	5.43
10240	5.90	5.85	5.80	5.70	5.65	5.60	5.55	5.50	5.40
12288	5.91	5.87	5.79	5.70	5.66	5.58	5.54	5.50	5.41
14336	5.92	5.85	5.78	5.71	5.67	5.60	5.53	5.50	5.42
16384	5.93	5.87	5.81	5.71	5.65	5.59	5.53	5.50	5.43
18432	5.94	5.86	5.80	5.72	5.66	5.61	5.55	5.50	5.41
20480	5.95	5.87	5.80	5.75	5.67	5.60	5.55	5.50	5.42
22528	5.93	5.86	5.79	5.75	5.68	5.61	5.54	5.50	5.43
24576	5.93	5.87	5.81	5.75	5.66	5.60	5.56	5.50	5.43
26624	5.94	5.86	5.80	5.75	5.67	5.61	5.55	5.50	5.44
28672	5.94	5.87	5.80	5.75	5.67	5.60	5.55	5.50	5.42
30720	5.95	5.86	5.80	5.75	5.68	5.61	5.55	5.50	5.43
32768	5.93	5.87	5.81	5.75	5.67	5.60	5.56	5.50	5.43

CI Size	Logical Block Size								
	95	96	97	98	99	100	101	102	103
512	5.00	5.00	5.00	5.00	5.00	5.00	4.00	4.00	4.00
1024	5.00	5.00	5.00	5.00	5.00	5.00	5.00	4.50	4.50
1536	5.33	5.00	5.00	5.00	5.00	5.00	5.00	4.66	4.66
2048	5.25	5.25	5.25	5.00	5.00	5.00	5.00	4.75	4.75
2560	5.20	5.20	5.20	5.20	5.00	5.00	5.00	5.00	4.80
3072	5.33	5.16	5.16	5.16	5.00	5.00	5.00	5.00	4.83
3584	5.28	5.28	5.14	5.14	5.14	5.00	5.00	5.00	4.85
4096	5.37	5.25	5.25	5.12	5.12	5.00	5.00	5.00	4.87
4608	5.33	5.22	5.22	5.11	5.11	5.00	5.00	5.00	4.88
5120	5.30	5.30	5.20	5.20	5.10	5.10	5.00	5.00	4.90
5632	5.36	5.27	5.18	5.18	5.09	5.09	5.00	5.00	4.90
6144	5.33	5.25	5.25	5.16	5.08	5.08	5.00	5.00	4.91
6656	5.30	5.30	5.23	5.15	5.15	5.07	5.00	5.00	4.92
7168	5.35	5.28	5.21	5.21	5.14	5.07	5.00	5.00	4.92
7680	5.33	5.26	5.26	5.20	5.13	5.06	5.00	5.00	4.93
8192	5.37	5.31	5.25	5.18	5.12	5.06	5.06	5.00	4.93
10240	5.35	5.30	5.25	5.20	5.15	5.10	5.05	5.00	4.95
12288	5.37	5.29	5.25	5.20	5.16	5.08	5.04	5.00	4.95
14336	5.35	5.32	5.25	5.21	5.14	5.10	5.03	5.00	4.96
16384	5.37	5.31	5.25	5.21	5.15	5.09	5.06	5.00	4.93
18432	5.36	5.30	5.25	5.19	5.16	5.11	5.05	5.00	4.94
20480	5.37	5.32	5.27	5.20	5.15	5.10	5.05	5.00	4.95
22528	5.38	5.31	5.27	5.20	5.15	5.11	5.04	5.00	4.95
24576	5.37	5.31	5.27	5.20	5.16	5.10	5.06	5.00	4.95
26624	5.38	5.32	5.26	5.21	5.15	5.11	5.05	5.00	4.96
28672	5.37	5.32	5.26	5.21	5.16	5.10	5.05	5.01	4.96
30720	5.38	5.31	5.26	5.21	5.16	5.11	5.06	5.01	4.96
32768	5.37	5.32	5.26	5.21	5.15	5.10	5.06	5.01	4.96

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	104	105	106	107	108	109	110	111	112
512	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
1024	4.50	4.50	4.50	4.50	4.50	4.50	4.50	4.50	4.50
1536	4.66	4.66	4.66	4.66	4.66	4.66	4.33	4.33	4.33
2048	4.75	4.75	4.75	4.75	4.50	4.50	4.50	4.50	4.50
2560	4.80	4.80	4.80	4.60	4.60	4.60	4.60	4.40	4.40
3072	4.83	4.83	4.66	4.66	4.66	4.66	4.50	4.50	4.50
3584	4.85	4.85	4.71	4.71	4.71	4.57	4.57	4.57	4.42
4096	4.87	4.75	4.75	4.75	4.62	4.62	4.62	4.50	4.50
4608	4.88	4.77	4.77	4.66	4.66	4.66	4.55	4.55	4.55
5120	4.90	4.80	4.80	4.70	4.70	4.60	4.60	4.60	4.50
5632	4.90	4.81	4.81	4.72	4.72	4.63	4.63	4.54	4.54
6144	4.83	4.83	4.75	4.75	4.66	4.66	4.58	4.58	4.50
6656	4.84	4.84	4.76	4.76	4.69	4.61	4.61	4.53	4.53
7168	4.85	4.85	4.78	4.71	4.71	4.64	4.64	4.57	4.50
7680	4.86	4.86	4.80	4.73	4.73	4.66	4.60	4.60	4.53
8192	4.87	4.81	4.81	4.75	4.68	4.68	4.62	4.56	4.56
10240	4.90	4.85	4.80	4.75	4.70	4.65	4.65	4.60	4.55
12288	4.91	4.83	4.79	4.75	4.70	4.66	4.62	4.58	4.54
14336	4.89	4.85	4.82	4.75	4.71	4.67	4.64	4.60	4.53
16384	4.90	4.84	4.81	4.78	4.71	4.68	4.62	4.59	4.56
18432	4.91	4.86	4.80	4.77	4.72	4.69	4.63	4.58	4.55
20480	4.90	4.85	4.82	4.77	4.72	4.67	4.65	4.60	4.55
22528	4.90	4.86	4.81	4.77	4.72	4.68	4.63	4.59	4.56
24576	4.91	4.85	4.81	4.77	4.72	4.68	4.64	4.60	4.56
26624	4.90	4.86	4.82	4.76	4.73	4.69	4.63	4.59	4.55
28672	4.91	4.85	4.82	4.76	4.73	4.67	4.64	4.60	4.55
30720	4.91	4.86	4.81	4.78	4.73	4.68	4.65	4.60	4.56
32768	4.90	4.85	4.82	4.78	4.73	4.68	4.64	4.60	4.56

CI Size	Logical Block Size								
	113	114	115	116	117	118	119	120	121
512	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
1024	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
1536	4.33	4.33	4.33	4.33	4.33	4.00	4.00	4.00	4.00
2048	4.50	4.25	4.25	4.25	4.25	4.25	4.25	4.00	4.00
2560	4.40	4.40	4.40	4.20	4.20	4.20	4.20	4.20	4.20
3072	4.50	4.33	4.33	4.33	4.33	4.16	4.16	4.16	4.16
3584	4.42	4.42	4.42	4.28	4.28	4.28	4.28	4.14	4.14
4096	4.50	4.37	4.37	4.37	4.25	4.25	4.25	4.25	4.12
4608	4.44	4.44	4.33	4.33	4.33	4.22	4.22	4.22	4.22
5120	4.50	4.40	4.40	4.40	4.30	4.30	4.20	4.20	4.20
5632	4.45	4.45	4.36	4.36	4.36	4.27	4.27	4.18	4.18
6144	4.50	4.41	4.41	4.33	4.33	4.25	4.25	4.25	4.16
6656	4.46	4.46	4.38	4.38	4.30	4.30	4.23	4.23	4.15
7168	4.50	4.42	4.42	4.35	4.35	4.28	4.28	4.21	4.21
7680	4.46	4.46	4.40	4.40	4.33	4.33	4.26	4.20	4.20
8192	4.50	4.43	4.43	4.37	4.31	4.31	4.25	4.25	4.18
10240	4.50	4.45	4.40	4.40	4.35	4.30	4.25	4.25	4.20
12288	4.50	4.45	4.41	4.37	4.33	4.33	4.29	4.25	4.20
14336	4.50	4.46	4.42	4.39	4.35	4.32	4.28	4.25	4.21
16384	4.50	4.46	4.43	4.40	4.34	4.31	4.28	4.25	4.21
18432	4.52	4.47	4.44	4.38	4.36	4.33	4.27	4.25	4.22
20480	4.52	4.47	4.45	4.40	4.35	4.32	4.30	4.25	4.22
22528	4.52	4.47	4.43	4.40	4.36	4.31	4.29	4.25	4.22
24576	4.52	4.47	4.43	4.39	4.35	4.33	4.29	4.25	4.22
26624	4.51	4.48	4.44	4.40	4.36	4.32	4.28	4.25	4.21
28672	4.51	4.48	4.44	4.41	4.35	4.32	4.28	4.25	4.21
30720	4.51	4.48	4.45	4.40	4.36	4.33	4.30	4.25	4.21
32768	4.51	4.48	4.43	4.40	4.35	4.32	4.29	4.25	4.21

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	122	123	124	125	126	127	128	129	130
512	4.00	4.00	4.00	4.00	3.00	3.00	3.00	3.00	3.00
1024	4.00	4.00	4.00	4.00	4.00	3.50	3.50	3.50	3.50
1536	4.00	4.00	4.00	4.00	4.00	4.00	3.66	3.66	3.66
2048	4.00	4.00	4.00	4.00	4.00	4.00	3.75	3.75	3.75
2560	4.00	4.00	4.00	4.00	4.00	4.00	3.80	3.80	3.80
3072	4.16	4.00	4.00	4.00	4.00	4.00	3.83	3.83	3.83
3584	4.14	4.14	4.00	4.00	4.00	4.00	3.85	3.85	3.85
4096	4.12	4.12	4.00	4.00	4.00	4.00	3.87	3.87	3.87
4608	4.11	4.11	4.11	4.00	4.00	4.00	3.88	3.88	3.88
5120	4.10	4.10	4.10	4.00	4.00	4.00	3.90	3.90	3.90
5632	4.18	4.09	4.09	4.00	4.00	4.00	3.90	3.90	3.90
6144	4.16	4.08	4.08	4.08	4.00	4.00	3.91	3.91	3.91
6656	4.15	4.15	4.07	4.07	4.00	4.00	3.92	3.92	3.92
7168	4.14	4.14	4.07	4.07	4.00	4.00	3.92	3.92	3.92
7680	4.13	4.13	4.06	4.06	4.00	4.00	3.93	3.93	3.93
8192	4.18	4.12	4.06	4.06	4.00	4.00	3.93	3.93	3.87
10240	4.15	4.15	4.10	4.05	4.05	4.00	3.95	3.95	3.90
12288	4.16	4.12	4.12	4.08	4.04	4.00	3.95	3.95	3.91
14336	4.17	4.14	4.10	4.07	4.03	4.00	3.96	3.96	3.92
16384	4.18	4.15	4.12	4.06	4.03	4.00	3.96	3.93	3.90
18432	4.19	4.13	4.11	4.08	4.05	4.02	3.97	3.94	3.91
20480	4.17	4.15	4.12	4.07	4.05	4.02	3.97	3.95	3.92
22528	4.18	4.15	4.11	4.09	4.04	4.02	3.97	3.95	3.93
24576	4.18	4.14	4.12	4.08	4.04	4.02	3.97	3.95	3.91
26624	4.19	4.15	4.11	4.07	4.05	4.01	3.98	3.96	3.92
28672	4.17	4.16	4.12	4.08	4.05	4.01	3.98	3.96	3.92
30720	4.18	4.15	4.11	4.08	4.05	4.01	3.98	3.96	3.93
32768	4.18	4.15	4.12	4.09	4.04	4.01	3.98	3.95	3.92

CI Size	Logical Block Size								
	131	132	133	134	135	136	137	138	139
512	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
1024	3.50	3.50	3.50	3.50	3.50	3.50	3.50	3.50	3.50
1536	3.66	3.66	3.66	3.66	3.66	3.66	3.66	3.66	3.33
2048	3.75	3.75	3.75	3.75	3.75	3.50	3.50	3.50	3.50
2560	3.80	3.80	3.80	3.80	3.60	3.60	3.60	3.60	3.60
3072	3.83	3.83	3.83	3.66	3.66	3.66	3.66	3.66	3.66
3584	3.85	3.85	3.71	3.71	3.71	3.71	3.71	3.57	3.57
4096	3.87	3.75	3.75	3.75	3.75	3.75	3.62	3.62	3.62
4608	3.88	3.77	3.77	3.77	3.77	3.66	3.66	3.66	3.66
5120	3.90	3.80	3.80	3.80	3.70	3.70	3.70	3.70	3.60
5632	3.81	3.81	3.81	3.72	3.72	3.72	3.72	3.63	3.63
6144	3.83	3.83	3.83	3.75	3.75	3.75	3.66	3.66	3.66
6656	3.84	3.84	3.76	3.76	3.76	3.69	3.69	3.69	3.61
7168	3.85	3.85	3.78	3.78	3.78	3.71	3.71	3.64	3.64
7680	3.86	3.86	3.80	3.80	3.73	3.73	3.66	3.66	3.66
8192	3.87	3.81	3.81	3.81	3.75	3.75	3.68	3.68	3.62
10240	3.90	3.85	3.80	3.80	3.75	3.75	3.70	3.70	3.65
12288	3.87	3.87	3.83	3.79	3.75	3.75	3.70	3.66	3.66
14336	3.89	3.85	3.82	3.78	3.78	3.75	3.71	3.67	3.67
16384	3.87	3.87	3.84	3.81	3.78	3.75	3.71	3.68	3.65
18432	3.88	3.86	3.83	3.80	3.77	3.75	3.72	3.69	3.66
20480	3.90	3.87	3.82	3.80	3.77	3.75	3.72	3.70	3.67
22528	3.88	3.86	3.84	3.81	3.77	3.75	3.72	3.70	3.68
24576	3.89	3.87	3.83	3.81	3.77	3.75	3.72	3.70	3.66
26624	3.90	3.86	3.84	3.80	3.78	3.75	3.73	3.69	3.67
28672	3.89	3.87	3.83	3.80	3.78	3.75	3.73	3.69	3.67
30720	3.90	3.86	3.83	3.81	3.78	3.75	3.73	3.70	3.66
32768	3.90	3.87	3.84	3.81	3.78	3.75	3.73	3.70	3.67

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	140	141	142	143	144	145	146	147	148
512	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
1024	3.50	3.50	3.50	3.50	3.50	3.00	3.00	3.00	3.00
1536	3.33	3.33	3.33	3.33	3.33	3.33	3.33	3.33	3.33
2048	3.50	3.50	3.50	3.50	3.50	3.50	3.25	3.25	3.25
2560	3.60	3.60	3.40	3.40	3.40	3.40	3.40	3.40	3.40
3072	3.50	3.50	3.50	3.50	3.50	3.50	3.33	3.33	3.33
3584	3.57	3.57	3.57	3.42	3.42	3.42	3.42	3.42	3.42
4096	3.62	3.50	3.50	3.50	3.50	3.50	3.37	3.37	3.37
4608	3.55	3.55	3.55	3.55	3.44	3.44	3.44	3.44	3.44
5120	3.60	3.60	3.50	3.50	3.50	3.50	3.50	3.40	3.40
5632	3.63	3.54	3.54	3.54	3.54	3.45	3.45	3.45	3.36
6144	3.58	3.58	3.58	3.50	3.50	3.50	3.50	3.41	3.41
6656	3.61	3.61	3.53	3.53	3.53	3.46	3.46	3.46	3.38
7168	3.64	3.57	3.57	3.57	3.50	3.50	3.50	3.42	3.42
7680	3.60	3.60	3.60	3.53	3.53	3.46	3.46	3.46	3.40
8192	3.62	3.62	3.56	3.56	3.50	3.50	3.50	3.43	3.43
10240	3.65	3.60	3.60	3.55	3.55	3.50	3.50	3.45	3.45
12288	3.62	3.62	3.58	3.54	3.54	3.50	3.50	3.45	3.41
14336	3.64	3.60	3.57	3.57	3.53	3.50	3.50	3.46	3.42
16384	3.62	3.62	3.59	3.56	3.53	3.50	3.50	3.46	3.43
18432	3.63	3.61	3.58	3.55	3.52	3.52	3.50	3.47	3.44
20480	3.65	3.62	3.60	3.57	3.55	3.52	3.50	3.47	3.45
22528	3.63	3.61	3.59	3.56	3.54	3.52	3.50	3.47	3.45
24576	3.64	3.62	3.60	3.56	3.54	3.52	3.50	3.47	3.43
26624	3.65	3.61	3.59	3.57	3.53	3.51	3.50	3.48	3.44
28672	3.64	3.62	3.58	3.57	3.55	3.51	3.50	3.46	3.44
30720	3.65	3.61	3.60	3.56	3.55	3.51	3.50	3.46	3.45
32768	3.64	3.62	3.59	3.57	3.54	3.51	3.50	3.46	3.45

CI Size	Logical Block Size								
	149	150	151	152	153	154	155	156	157
512	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
1024	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
1536	3.33	3.33	3.33	3.33	3.00	3.00	3.00	3.00	3.00
2048	3.25	3.25	3.25	3.25	3.25	3.25	3.25	3.25	3.00
2560	3.40	3.40	3.20	3.20	3.20	3.20	3.20	3.20	3.20
3072	3.33	3.33	3.33	3.33	3.33	3.16	3.16	3.16	3.16
3584	3.28	3.28	3.28	3.28	3.28	3.28	3.28	3.14	3.14
4096	3.37	3.37	3.37	3.25	3.25	3.25	3.25	3.25	3.25
4608	3.33	3.33	3.33	3.33	3.33	3.22	3.22	3.22	3.22
5120	3.40	3.40	3.30	3.30	3.30	3.30	3.20	3.20	3.20
5632	3.36	3.36	3.36	3.27	3.27	3.27	3.27	3.27	3.18
6144	3.41	3.33	3.33	3.33	3.33	3.25	3.25	3.25	3.25
6656	3.38	3.38	3.38	3.30	3.30	3.30	3.23	3.23	3.23
7168	3.42	3.35	3.35	3.35	3.28	3.28	3.28	3.21	3.21
7680	3.40	3.40	3.33	3.33	3.33	3.26	3.26	3.26	3.20
8192	3.37	3.37	3.37	3.31	3.31	3.31	3.25	3.25	3.25
10240	3.40	3.40	3.35	3.35	3.30	3.30	3.30	3.25	3.25
12288	3.41	3.37	3.37	3.33	3.33	3.29	3.29	3.25	3.25
14336	3.42	3.39	3.35	3.35	3.32	3.32	3.28	3.25	3.25
16384	3.40	3.40	3.37	3.34	3.34	3.31	3.28	3.25	3.25
18432	3.41	3.38	3.38	3.36	3.33	3.30	3.27	3.27	3.25
20480	3.42	3.40	3.37	3.35	3.32	3.30	3.30	3.27	3.25
22528	3.43	3.40	3.38	3.36	3.34	3.31	3.29	3.27	3.25
24576	3.41	3.39	3.37	3.35	3.33	3.31	3.29	3.27	3.25
26624	3.42	3.40	3.38	3.36	3.32	3.30	3.28	3.26	3.25
28672	3.42	3.41	3.37	3.35	3.33	3.32	3.28	3.26	3.25
30720	3.43	3.40	3.38	3.36	3.33	3.31	3.30	3.26	3.25
32768	3.42	3.40	3.37	3.35	3.34	3.31	3.29	3.26	3.25

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	158	159	160	161	162	163	164	165	166
512	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
1024	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
1536	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
2048	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
2560	3.20	3.20	3.00	3.00	3.00	3.00	3.00	3.00	3.00
3072	3.16	3.16	3.16	3.16	3.00	3.00	3.00	3.00	3.00
3584	3.14	3.14	3.14	3.14	3.14	3.00	3.00	3.00	3.00
4096	3.12	3.12	3.12	3.12	3.12	3.12	3.00	3.00	3.00
4608	3.22	3.11	3.11	3.11	3.11	3.11	3.11	3.00	3.00
5120	3.20	3.20	3.10	3.10	3.10	3.10	3.10	3.00	3.00
5632	3.18	3.18	3.18	3.09	3.09	3.09	3.09	3.09	3.00
6144	3.16	3.16	3.16	3.16	3.08	3.08	3.08	3.08	3.00
6656	3.23	3.15	3.15	3.15	3.15	3.07	3.07	3.07	3.07
7168	3.21	3.21	3.14	3.14	3.14	3.07	3.07	3.07	3.07
7680	3.20	3.20	3.13	3.13	3.13	3.13	3.06	3.06	3.06
8192	3.18	3.18	3.18	3.12	3.12	3.12	3.06	3.06	3.06
10240	3.20	3.20	3.15	3.15	3.15	3.10	3.10	3.10	3.05
12288	3.20	3.20	3.16	3.16	3.12	3.12	3.08	3.08	3.04
14336	3.21	3.21	3.17	3.14	3.14	3.10	3.10	3.07	3.07
16384	3.21	3.18	3.18	3.15	3.15	3.12	3.09	3.09	3.06
18432	3.22	3.19	3.19	3.16	3.13	3.13	3.11	3.08	3.05
20480	3.22	3.20	3.17	3.17	3.15	3.12	3.10	3.10	3.07
22528	3.22	3.20	3.18	3.15	3.15	3.13	3.11	3.09	3.06
24576	3.22	3.20	3.18	3.16	3.14	3.12	3.10	3.08	3.06
26624	3.23	3.21	3.19	3.17	3.15	3.13	3.11	3.09	3.07
28672	3.23	3.21	3.19	3.17	3.14	3.12	3.10	3.08	3.07
30720	3.23	3.21	3.18	3.16	3.15	3.13	3.11	3.10	3.08
32768	3.23	3.21	3.18	3.17	3.15	3.12	3.10	3.09	3.07

CI Size	Logical Block Size								
	167	168	169	170	171	172	173	174	175
512	3.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
1024	3.00	3.00	3.00	2.50	2.50	2.50	2.50	2.50	2.50
1536	3.00	3.00	3.00	2.66	2.66	2.66	2.66	2.66	2.66
2048	3.00	3.00	3.00	2.75	2.75	2.75	2.75	2.75	2.75
2560	3.00	3.00	3.00	3.00	2.80	2.80	2.80	2.80	2.80
3072	3.00	3.00	3.00	3.00	2.83	2.83	2.83	2.83	2.83
3584	3.00	3.00	3.00	3.00	2.85	2.85	2.85	2.85	2.85
4096	3.00	3.00	3.00	3.00	2.87	2.87	2.87	2.87	2.87
4608	3.00	3.00	3.00	3.00	2.88	2.88	2.88	2.88	2.88
5120	3.00	3.00	3.00	3.00	2.90	2.90	2.90	2.90	2.90
5632	3.00	3.00	3.00	3.00	2.90	2.90	2.90	2.90	2.90
6144	3.00	3.00	3.00	3.00	2.91	2.91	2.91	2.91	2.91
6656	3.00	3.00	3.00	3.00	2.92	2.92	2.92	2.92	2.84
7168	3.00	3.00	3.00	3.00	2.92	2.92	2.92	2.92	2.85
7680	3.00	3.00	3.00	3.00	2.93	2.93	2.93	2.93	2.86
8192	3.00	3.00	3.00	3.00	2.93	2.93	2.93	2.93	2.87
10240	3.05	3.00	3.00	3.00	2.95	2.95	2.95	2.90	2.90
12288	3.04	3.04	3.00	3.00	2.95	2.95	2.91	2.91	2.91
14336	3.03	3.03	3.00	3.00	2.96	2.96	2.92	2.92	2.89
16384	3.06	3.03	3.00	3.00	2.96	2.96	2.93	2.93	2.90
18432	3.05	3.02	3.02	3.00	2.97	2.97	2.94	2.91	2.91
20480	3.05	3.02	3.02	3.00	2.97	2.97	2.95	2.92	2.90
22528	3.04	3.04	3.02	3.00	2.97	2.95	2.95	2.93	2.90
24576	3.06	3.04	3.02	3.00	2.97	2.95	2.95	2.93	2.91
26624	3.05	3.03	3.01	3.00	2.98	2.96	2.94	2.92	2.92
28672	3.05	3.03	3.01	3.00	2.98	2.96	2.94	2.92	2.91
30720	3.05	3.03	3.01	3.00	2.98	2.96	2.95	2.93	2.91
32768	3.06	3.03	3.01	3.00	2.98	2.96	2.95	2.93	2.92

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	176	177	178	179	180	181	182	183	184
512	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
1024	2.50	2.50	2.50	2.50	2.50	2.50	2.50	2.50	2.50
1536	2.66	2.66	2.66	2.66	2.66	2.66	2.66	2.66	2.66
2048	2.75	2.75	2.75	2.75	2.75	2.75	2.75	2.75	2.75
2560	2.80	2.80	2.80	2.80	2.80	2.80	2.80	2.60	2.60
3072	2.83	2.83	2.83	2.83	2.83	2.66	2.66	2.66	2.66
3584	2.85	2.85	2.85	2.71	2.71	2.71	2.71	2.71	2.71
4096	2.87	2.87	2.75	2.75	2.75	2.75	2.75	2.75	2.75
4608	2.88	2.77	2.77	2.77	2.77	2.77	2.77	2.77	2.66
5120	2.90	2.80	2.80	2.80	2.80	2.80	2.80	2.70	2.70
5632	2.81	2.81	2.81	2.81	2.81	2.81	2.72	2.72	2.72
6144	2.83	2.83	2.83	2.83	2.83	2.75	2.75	2.75	2.75
6656	2.84	2.84	2.84	2.84	2.76	2.76	2.76	2.76	2.76
7168	2.85	2.85	2.85	2.78	2.78	2.78	2.78	2.78	2.71
7680	2.86	2.86	2.86	2.80	2.80	2.80	2.80	2.73	2.73
8192	2.87	2.87	2.81	2.81	2.81	2.81	2.75	2.75	2.75
10240	2.90	2.85	2.85	2.85	2.80	2.80	2.80	2.75	2.75
12288	2.87	2.87	2.83	2.83	2.83	2.79	2.79	2.79	2.75
14336	2.89	2.85	2.85	2.85	2.82	2.82	2.78	2.78	2.75
16384	2.90	2.87	2.84	2.84	2.81	2.81	2.78	2.78	2.75
18432	2.88	2.88	2.86	2.83	2.83	2.80	2.80	2.77	2.77
20480	2.90	2.87	2.87	2.85	2.82	2.82	2.80	2.77	2.77
22528	2.88	2.88	2.86	2.84	2.84	2.81	2.79	2.79	2.77
24576	2.89	2.87	2.87	2.85	2.83	2.81	2.79	2.79	2.77
26624	2.90	2.88	2.86	2.84	2.82	2.82	2.80	2.78	2.76
28672	2.89	2.87	2.87	2.85	2.83	2.82	2.80	2.78	2.76
30720	2.90	2.88	2.86	2.85	2.83	2.81	2.80	2.78	2.76
32768	2.90	2.89	2.87	2.85	2.82	2.81	2.79	2.79	2.78

CI Size	Logical Block Size								
	185	186	187	188	189	190	191	192	193
512	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
1024	2.50	2.50	2.50	2.50	2.50	2.50	2.50	2.50	2.50
1536	2.66	2.66	2.66	2.66	2.66	2.66	2.33	2.33	2.33
2048	2.75	2.50	2.50	2.50	2.50	2.50	2.50	2.50	2.50
2560	2.60	2.60	2.60	2.60	2.60	2.60	2.60	2.60	2.60
3072	2.66	2.66	2.66	2.66	2.66	2.66	2.66	2.50	2.50
3584	2.71	2.71	2.71	2.71	2.57	2.57	2.57	2.57	2.57
4096	2.75	2.62	2.62	2.62	2.62	2.62	2.62	2.62	2.62
4608	2.66	2.66	2.66	2.66	2.66	2.66	2.66	2.55	2.55
5120	2.70	2.70	2.70	2.70	2.70	2.60	2.60	2.60	2.60
5632	2.72	2.72	2.72	2.63	2.63	2.63	2.63	2.63	2.63
6144	2.75	2.66	2.66	2.66	2.66	2.66	2.66	2.58	2.58
6656	2.69	2.69	2.69	2.69	2.69	2.61	2.61	2.61	2.61
7168	2.71	2.71	2.71	2.71	2.64	2.64	2.64	2.64	2.64
7680	2.73	2.73	2.73	2.66	2.66	2.66	2.66	2.60	2.60
8192	2.75	2.68	2.68	2.68	2.68	2.68	2.62	2.62	2.62
10240	2.75	2.75	2.70	2.70	2.70	2.65	2.65	2.65	2.65
12288	2.75	2.75	2.70	2.70	2.66	2.66	2.66	2.62	2.62
14336	2.75	2.75	2.71	2.71	2.67	2.67	2.67	2.64	2.64
16384	2.75	2.75	2.71	2.71	2.68	2.68	2.65	2.65	2.62
18432	2.75	2.75	2.72	2.69	2.69	2.66	2.66	2.63	2.63
20480	2.75	2.75	2.72	2.70	2.70	2.67	2.67	2.65	2.65
22528	2.75	2.75	2.72	2.70	2.70	2.68	2.65	2.65	2.63
24576	2.75	2.75	2.72	2.70	2.68	2.68	2.66	2.64	2.64
26624	2.75	2.75	2.73	2.71	2.69	2.69	2.67	2.65	2.63
28672	2.75	2.75	2.73	2.71	2.69	2.67	2.67	2.66	2.64
30720	2.76	2.75	2.73	2.71	2.70	2.68	2.66	2.65	2.65
32768	2.76	2.75	2.73	2.71	2.70	2.68	2.67	2.65	2.64

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	194	195	196	197	198	199	200	201	202
512	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
1024	2.50	2.50	2.50	2.50	2.50	2.50	2.50	2.50	2.50
1536	2.33	2.33	2.33	2.33	2.33	2.33	2.33	2.33	2.33
2048	2.50	2.50	2.50	2.50	2.50	2.50	2.50	2.50	2.50
2560	2.60	2.60	2.60	2.40	2.40	2.40	2.40	2.40	2.40
3072	2.50	2.50	2.50	2.50	2.50	2.50	2.50	2.50	2.50
3584	2.57	2.57	2.57	2.57	2.57	2.42	2.42	2.42	2.42
4096	2.62	2.50	2.50	2.50	2.50	2.50	2.50	2.50	2.50
4608	2.55	2.55	2.55	2.55	2.55	2.55	2.44	2.44	2.44
5120	2.60	2.60	2.60	2.50	2.50	2.50	2.50	2.50	2.50
5632	2.54	2.54	2.54	2.54	2.54	2.54	2.54	2.45	2.45
6144	2.58	2.58	2.58	2.58	2.50	2.50	2.50	2.50	2.50
6656	2.61	2.61	2.53	2.53	2.53	2.53	2.53	2.53	2.46
7168	2.57	2.57	2.57	2.57	2.57	2.50	2.50	2.50	2.50
7680	2.60	2.60	2.60	2.53	2.53	2.53	2.53	2.53	2.46
8192	2.62	2.56	2.56	2.56	2.56	2.56	2.50	2.50	2.50
10240	2.60	2.60	2.60	2.55	2.55	2.55	2.55	2.50	2.50
12288	2.62	2.58	2.58	2.58	2.58	2.54	2.54	2.54	2.50
14336	2.60	2.60	2.60	2.57	2.57	2.53	2.53	2.53	2.50
16384	2.62	2.59	2.59	2.59	2.56	2.56	2.53	2.53	2.53
18432	2.61	2.61	2.58	2.58	2.58	2.55	2.55	2.52	2.52
20480	2.62	2.60	2.60	2.57	2.57	2.55	2.55	2.52	2.52
22528	2.63	2.61	2.59	2.59	2.56	2.56	2.54	2.54	2.52
24576	2.62	2.60	2.60	2.58	2.58	2.56	2.54	2.54	2.52
26624	2.63	2.61	2.59	2.59	2.57	2.55	2.55	2.53	2.51
28672	2.62	2.60	2.60	2.58	2.57	2.57	2.55	2.53	2.51
30720	2.63	2.61	2.60	2.58	2.58	2.56	2.55	2.53	2.53
32768	2.62	2.60	2.60	2.59	2.57	2.56	2.54	2.53	2.53

CI Size	Logical Block Size								
	203	204	206	207	208	209	210	211	212
512	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
1024	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
1536	2.33	2.33	2.33	2.33	2.33	2.33	2.33	2.33	2.33
2048	2.50	2.25	2.25	2.25	2.25	2.25	2.25	2.25	2.25
2560	2.40	2.40	2.40	2.40	2.40	2.40	2.40	2.40	2.40
3072	2.50	2.50	2.33	2.33	2.33	2.33	2.33	2.33	2.33
3584	2.42	2.42	2.42	2.42	2.42	2.42	2.42	2.28	2.28
4096	2.50	2.50	2.37	2.37	2.37	2.37	2.37	2.37	2.37
4608	2.44	2.44	2.44	2.44	2.44	2.44	2.33	2.33	2.33
5120	2.50	2.50	2.40	2.40	2.40	2.40	2.40	2.40	2.40
5632	2.45	2.45	2.45	2.45	2.45	2.36	2.36	2.36	2.36
6144	2.50	2.50	2.41	2.41	2.41	2.41	2.41	2.41	2.33
6656	2.46	2.46	2.46	2.46	2.38	2.38	2.38	2.38	2.38
7168	2.50	2.50	2.42	2.42	2.42	2.42	2.42	2.35	2.35
7680	2.46	2.46	2.46	2.46	2.40	2.40	2.40	2.40	2.40
8192	2.50	2.50	2.43	2.43	2.43	2.43	2.37	2.37	2.37
10240	2.50	2.50	2.45	2.45	2.45	2.40	2.40	2.40	2.40
12288	2.50	2.50	2.45	2.45	2.45	2.41	2.41	2.41	2.37
14336	2.50	2.50	2.46	2.46	2.42	2.42	2.42	2.39	2.39
16384	2.50	2.50	2.46	2.46	2.43	2.43	2.40	2.40	2.40
18432	2.50	2.50	2.47	2.44	2.44	2.44	2.41	2.41	2.38
20480	2.50	2.50	2.47	2.45	2.45	2.42	2.42	2.42	2.40
22528	2.50	2.50	2.47	2.45	2.45	2.43	2.43	2.40	2.40
24576	2.52	2.50	2.47	2.45	2.45	2.43	2.41	2.41	2.39
26624	2.51	2.50	2.48	2.46	2.44	2.44	2.42	2.42	2.40
28672	2.51	2.50	2.48	2.46	2.44	2.44	2.42	2.41	2.41
30720	2.51	2.50	2.48	2.46	2.45	2.43	2.43	2.41	2.40
32768	2.51	2.50	2.48	2.46	2.45	2.43	2.42	2.42	2.40

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	213	214	215	216	217	218	219	220	221
512	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
1024	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
1536	2.33	2.33	2.33	2.33	2.33	2.33	2.00	2.00	2.00
2048	2.25	2.25	2.25	2.25	2.25	2.25	2.25	2.25	2.25
2560	2.20	2.20	2.20	2.20	2.20	2.20	2.20	2.20	2.20
3072	2.33	2.33	2.33	2.33	2.33	2.33	2.16	2.16	2.16
3584	2.28	2.28	2.28	2.28	2.28	2.28	2.28	2.28	2.28
4096	2.37	2.37	2.37	2.25	2.25	2.25	2.25	2.25	2.25
4608	2.33	2.33	2.33	2.33	2.33	2.33	2.22	2.22	2.22
5120	2.30	2.30	2.30	2.30	2.30	2.30	2.30	2.30	2.30
5632	2.36	2.36	2.36	2.36	2.27	2.27	2.27	2.27	2.27
6144	2.33	2.33	2.33	2.33	2.33	2.33	2.33	2.25	2.25
6656	2.38	2.38	2.30	2.30	2.30	2.30	2.30	2.30	2.30
7168	2.35	2.35	2.35	2.35	2.28	2.28	2.28	2.28	2.28
7680	2.40	2.33	2.33	2.33	2.33	2.33	2.33	2.26	2.26
8192	2.37	2.37	2.37	2.31	2.31	2.31	2.31	2.31	2.31
10240	2.40	2.35	2.35	2.35	2.35	2.30	2.30	2.30	2.30
12288	2.37	2.37	2.37	2.33	2.33	2.33	2.33	2.29	2.29
14336	2.39	2.35	2.35	2.35	2.35	2.32	2.32	2.32	2.28
16384	2.37	2.37	2.37	2.34	2.34	2.34	2.31	2.31	2.31
18432	2.38	2.38	2.36	2.36	2.33	2.33	2.33	2.30	2.30
20480	2.40	2.37	2.37	2.35	2.35	2.32	2.32	2.32	2.30
22528	2.38	2.38	2.36	2.36	2.34	2.34	2.31	2.31	2.29
24576	2.39	2.37	2.37	2.35	2.35	2.33	2.33	2.31	2.31
26624	2.38	2.38	2.36	2.36	2.34	2.34	2.32	2.30	2.30
28672	2.39	2.37	2.37	2.35	2.35	2.33	2.32	2.32	2.30
30720	2.40	2.38	2.36	2.36	2.35	2.33	2.33	2.31	2.30
32768	2.39	2.39	2.37	2.35	2.34	2.34	2.32	2.31	2.31

CI Size	Logical Block Size								
	222	223	224	225	226	227	229	230	231
512	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
1024	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
1536	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
2048	2.25	2.25	2.25	2.25	2.25	2.00	2.00	2.00	2.00
2560	2.20	2.20	2.20	2.20	2.20	2.20	2.20	2.20	2.20
3072	2.16	2.16	2.16	2.16	2.16	2.16	2.16	2.16	2.16
3584	2.28	2.28	2.14	2.14	2.14	2.14	2.14	2.14	2.14
4096	2.25	2.25	2.25	2.25	2.25	2.25	2.12	2.12	2.12
4608	2.22	2.22	2.22	2.22	2.22	2.22	2.22	2.11	2.11
5120	2.30	2.20	2.20	2.20	2.20	2.20	2.20	2.20	2.20
5632	2.27	2.27	2.27	2.18	2.18	2.18	2.18	2.18	2.18
6144	2.25	2.25	2.25	2.25	2.25	2.25	2.16	2.16	2.16
6656	2.23	2.23	2.23	2.23	2.23	2.23	2.23	2.15	2.15
7168	2.28	2.28	2.21	2.21	2.21	2.21	2.21	2.21	2.14
7680	2.26	2.26	2.26	2.26	2.20	2.20	2.20	2.20	2.20
8192	2.25	2.25	2.25	2.25	2.25	2.25	2.18	2.18	2.18
10240	2.30	2.25	2.25	2.25	2.25	2.25	2.20	2.20	2.20
12288	2.29	2.29	2.25	2.25	2.25	2.25	2.20	2.20	2.20
14336	2.28	2.28	2.25	2.25	2.25	2.25	2.21	2.21	2.21
16384	2.28	2.28	2.28	2.25	2.25	2.25	2.21	2.21	2.18
18432	2.27	2.27	2.27	2.25	2.25	2.25	2.22	2.22	2.19
20480	2.30	2.27	2.27	2.25	2.25	2.25	2.22	2.22	2.20
22528	2.29	2.27	2.27	2.27	2.25	2.25	2.22	2.20	2.20
24576	2.29	2.29	2.27	2.27	2.25	2.25	2.22	2.20	2.20
26624	2.28	2.28	2.26	2.26	2.25	2.25	2.23	2.21	2.21
28672	2.30	2.28	2.26	2.26	2.25	2.25	2.23	2.21	2.21
30720	2.30	2.28	2.28	2.26	2.25	2.25	2.23	2.21	2.20
32768	2.29	2.28	2.28	2.26	2.25	2.25	2.23	2.21	2.20

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	232	233	234	235	236	237	238	239	240
512	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
1024	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
1536	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
2048	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
2560	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
3072	2.16	2.16	2.16	2.16	2.00	2.00	2.00	2.00	2.00
3584	2.14	2.14	2.14	2.14	2.14	2.14	2.14	2.00	2.00
4096	2.12	2.12	2.12	2.12	2.12	2.12	2.12	2.12	2.12
4608	2.11	2.11	2.11	2.11	2.11	2.11	2.11	2.11	2.11
5120	2.20	2.10	2.10	2.10	2.10	2.10	2.10	2.10	2.10
5632	2.18	2.18	2.18	2.09	2.09	2.09	2.09	2.09	2.09
6144	2.16	2.16	2.16	2.16	2.08	2.08	2.08	2.08	2.08
6656	2.15	2.15	2.15	2.15	2.15	2.15	2.07	2.07	2.07
7168	2.14	2.14	2.14	2.14	2.14	2.14	2.14	2.07	2.07
7680	2.20	2.13	2.13	2.13	2.13	2.13	2.13	2.13	2.06
8192	2.18	2.18	2.12	2.12	2.12	2.12	2.12	2.12	2.12
10240	2.20	2.15	2.15	2.15	2.15	2.15	2.10	2.10	2.10
12288	2.16	2.16	2.16	2.16	2.16	2.12	2.12	2.12	2.12
14336	2.17	2.17	2.17	2.14	2.14	2.14	2.14	2.10	2.10
16384	2.18	2.18	2.15	2.15	2.15	2.15	2.12	2.12	2.12
18432	2.19	2.19	2.16	2.16	2.16	2.13	2.13	2.13	2.11
20480	2.20	2.17	2.17	2.17	2.15	2.15	2.15	2.12	2.12
22528	2.20	2.18	2.18	2.15	2.15	2.15	2.13	2.13	2.11
24576	2.18	2.18	2.16	2.16	2.16	2.14	2.14	2.12	2.12
26624	2.19	2.19	2.17	2.17	2.15	2.15	2.13	2.13	2.11
28672	2.19	2.19	2.17	2.16	2.16	2.14	2.14	2.12	2.12
30720	2.20	2.18	2.18	2.16	2.16	2.15	2.15	2.13	2.11
32768	2.20	2.18	2.17	2.17	2.15	2.15	2.14	2.14	2.12

CI Size	Logical Block Size								
	241	242	243	244	245	246	247	248	249
512	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
1024	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
1536	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
2048	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
2560	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
3072	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
3584	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
4096	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
4608	2.11	2.11	2.00	2.00	2.00	2.00	2.00	2.00	2.00
5120	2.10	2.10	2.10	2.00	2.00	2.00	2.00	2.00	2.00
5632	2.09	2.09	2.09	2.09	2.00	2.00	2.00	2.00	2.00
6144	2.08	2.08	2.08	2.08	2.08	2.00	2.00	2.00	2.00
6656	2.07	2.07	2.07	2.07	2.07	2.07	2.00	2.00	2.00
7168	2.07	2.07	2.07	2.07	2.07	2.07	2.00	2.00	2.00
7680	2.06	2.06	2.06	2.06	2.06	2.06	2.06	2.00	2.00
8192	2.06	2.06	2.06	2.06	2.06	2.06	2.06	2.00	2.00
10240	2.10	2.10	2.10	2.05	2.05	2.05	2.05	2.05	2.05
12288	2.08	2.08	2.08	2.08	2.08	2.04	2.04	2.04	2.04
14336	2.10	2.10	2.07	2.07	2.07	2.07	2.07	2.03	2.03
16384	2.09	2.09	2.09	2.09	2.06	2.06	2.06	2.06	2.03
18432	2.11	2.11	2.08	2.08	2.08	2.05	2.05	2.05	2.02
20480	2.10	2.10	2.10	2.07	2.07	2.07	2.05	2.05	2.05
22528	2.11	2.11	2.09	2.09	2.06	2.06	2.06	2.04	2.04
24576	2.10	2.10	2.10	2.08	2.08	2.06	2.06	2.06	2.04
26624	2.11	2.09	2.09	2.09	2.07	2.07	2.05	2.05	2.03
28672	2.10	2.10	2.08	2.08	2.07	2.07	2.07	2.05	2.05
30720	2.11	2.10	2.10	2.08	2.08	2.06	2.06	2.05	2.05
32768	2.10	2.10	2.09	2.09	2.07	2.07	2.06	2.06	2.04

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	250	251	252	253	254	255	257	258	259
512	2.00	2.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1024	2.00	2.00	2.00	2.00	1.50	1.50	1.50	1.50	1.50
1536	2.00	2.00	2.00	2.00	2.00	1.66	1.66	1.66	1.66
2048	2.00	2.00	2.00	2.00	2.00	1.75	1.75	1.75	1.75
2560	2.00	2.00	2.00	2.00	2.00	2.00	1.80	1.80	1.80
3072	2.00	2.00	2.00	2.00	2.00	2.00	1.83	1.83	1.83
3584	2.00	2.00	2.00	2.00	2.00	2.00	1.85	1.85	1.85
4096	2.00	2.00	2.00	2.00	2.00	2.00	1.87	1.87	1.87
4608	2.00	2.00	2.00	2.00	2.00	2.00	1.88	1.88	1.88
5120	2.00	2.00	2.00	2.00	2.00	2.00	1.90	1.90	1.90
5632	2.00	2.00	2.00	2.00	2.00	2.00	1.90	1.90	1.90
6144	2.00	2.00	2.00	2.00	2.00	2.00	1.91	1.91	1.91
6656	2.00	2.00	2.00	2.00	2.00	2.00	1.92	1.92	1.92
7168	2.00	2.00	2.00	2.00	2.00	2.00	1.92	1.92	1.92
7680	2.00	2.00	2.00	2.00	2.00	2.00	1.93	1.93	1.93
8192	2.00	2.00	2.00	2.00	2.00	2.00	1.93	1.93	1.93
10240	2.00	2.00	2.00	2.00	2.00	2.00	1.95	1.95	1.95
12288	2.04	2.00	2.00	2.00	2.00	2.00	1.95	1.95	1.95
14336	2.03	2.03	2.00	2.00	2.00	2.00	1.96	1.96	1.96
16384	2.03	2.03	2.00	2.00	2.00	2.00	1.96	1.96	1.96
18432	2.02	2.02	2.02	2.00	2.00	2.00	1.97	1.97	1.97
20480	2.02	2.02	2.02	2.00	2.00	2.00	1.97	1.97	1.97
22528	2.04	2.02	2.02	2.02	2.00	2.00	1.97	1.97	1.95
24576	2.04	2.02	2.02	2.02	2.00	2.00	1.97	1.97	1.95
26624	2.03	2.03	2.01	2.01	2.00	2.00	1.98	1.98	1.96
28672	2.03	2.03	2.01	2.01	2.00	2.00	1.98	1.98	1.96
30720	2.03	2.03	2.01	2.01	2.00	2.00	1.98	1.98	1.96
32768	2.04	2.03	2.01	2.01	2.00	2.00	1.98	1.96	1.96

CI Size	Logical Block Size								
	260	261	262	263	264	265	266	267	268
512	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1024	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
1536	1.66	1.66	1.66	1.66	1.66	1.66	1.66	1.66	1.66
2048	1.75	1.75	1.75	1.75	1.75	1.75	1.75	1.75	1.75
2560	1.80	1.80	1.80	1.80	1.80	1.80	1.80	1.80	1.80
3072	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83
3584	1.85	1.85	1.85	1.85	1.85	1.85	1.85	1.85	1.85
4096	1.87	1.87	1.87	1.87	1.87	1.87	1.87	1.87	1.87
4608	1.88	1.88	1.88	1.88	1.88	1.88	1.88	1.88	1.88
5120	1.90	1.90	1.90	1.90	1.90	1.90	1.90	1.90	1.90
5632	1.90	1.90	1.90	1.90	1.90	1.90	1.90	1.90	1.81
6144	1.91	1.91	1.91	1.91	1.91	1.91	1.91	1.83	1.83
6656	1.92	1.92	1.92	1.92	1.92	1.92	1.84	1.84	1.84
7168	1.92	1.92	1.92	1.92	1.92	1.92	1.85	1.85	1.85
7680	1.93	1.93	1.93	1.93	1.93	1.86	1.86	1.86	1.86
8192	1.93	1.93	1.93	1.93	1.87	1.87	1.87	1.87	1.87
10240	1.95	1.95	1.95	1.90	1.90	1.90	1.90	1.90	1.90
12288	1.95	1.95	1.91	1.91	1.91	1.91	1.91	1.87	1.87
14336	1.96	1.92	1.92	1.92	1.92	1.92	1.89	1.89	1.89
16384	1.93	1.93	1.93	1.93	1.93	1.90	1.90	1.90	1.90
18432	1.94	1.94	1.94	1.94	1.91	1.91	1.91	1.88	1.88
20480	1.95	1.95	1.95	1.92	1.92	1.92	1.90	1.90	1.90
22528	1.95	1.95	1.93	1.93	1.93	1.90	1.90	1.90	1.90
24576	1.95	1.95	1.93	1.93	1.93	1.91	1.91	1.91	1.89
26624	1.96	1.94	1.94	1.94	1.92	1.92	1.92	1.90	1.90
28672	1.96	1.94	1.94	1.92	1.92	1.92	1.91	1.91	1.89
30720	1.96	1.95	1.95	1.93	1.93	1.91	1.91	1.91	1.90
32768	1.95	1.95	1.95	1.93	1.93	1.92	1.92	1.90	1.90

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	269	270	271	272	273	274	275	276	277
512	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1024	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
1536	1.66	1.66	1.66	1.66	1.66	1.66	1.66	1.66	1.66
2048	1.75	1.75	1.75	1.75	1.75	1.75	1.75	1.75	1.75
2560	1.80	1.80	1.80	1.80	1.80	1.80	1.80	1.80	1.80
3072	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83
3584	1.85	1.85	1.85	1.85	1.85	1.85	1.71	1.71	1.71
4096	1.87	1.87	1.87	1.87	1.75	1.75	1.75	1.75	1.75
4608	1.88	1.88	1.77	1.77	1.77	1.77	1.77	1.77	1.77
5120	1.80	1.80	1.80	1.80	1.80	1.80	1.80	1.80	1.80
5632	1.81	1.81	1.81	1.81	1.81	1.81	1.81	1.81	1.81
6144	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83	1.83
6656	1.84	1.84	1.84	1.84	1.84	1.84	1.84	1.84	1.76
7168	1.85	1.85	1.85	1.85	1.85	1.85	1.85	1.78	1.78
7680	1.86	1.86	1.86	1.86	1.86	1.80	1.80	1.80	1.80
8192	1.87	1.87	1.87	1.87	1.81	1.81	1.81	1.81	1.81
10240	1.90	1.85	1.85	1.85	1.85	1.85	1.85	1.85	1.80
12288	1.87	1.87	1.87	1.87	1.83	1.83	1.83	1.83	1.83
14336	1.89	1.89	1.85	1.85	1.85	1.85	1.85	1.82	1.82
16384	1.87	1.87	1.87	1.87	1.84	1.84	1.84	1.84	1.84
18432	1.88	1.88	1.86	1.86	1.86	1.86	1.83	1.83	1.83
20480	1.90	1.87	1.87	1.87	1.85	1.85	1.85	1.85	1.82
22528	1.88	1.88	1.88	1.86	1.86	1.86	1.84	1.84	1.84
24576	1.89	1.87	1.87	1.87	1.85	1.85	1.85	1.85	1.83
26624	1.88	1.88	1.88	1.86	1.86	1.86	1.84	1.84	1.84
28672	1.89	1.89	1.87	1.87	1.85	1.85	1.85	1.83	1.83
30720	1.90	1.88	1.88	1.86	1.86	1.86	1.85	1.85	1.83
32768	1.89	1.89	1.87	1.87	1.85	1.85	1.85	1.84	1.84

CI Size	Logical Block Size								
	278	279	280	281	282	283	284	285	286
512	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1024	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
1536	1.66	1.66	1.66	1.66	1.66	1.66	1.66	1.66	1.66
2048	1.75	1.75	1.75	1.75	1.75	1.75	1.75	1.75	1.75
2560	1.80	1.80	1.80	1.80	1.80	1.80	1.60	1.60	1.60
3072	1.83	1.66	1.66	1.66	1.66	1.66	1.66	1.66	1.66
3584	1.71	1.71	1.71	1.71	1.71	1.71	1.71	1.71	1.71
4096	1.75	1.75	1.75	1.75	1.75	1.75	1.75	1.75	1.75
4608	1.77	1.77	1.77	1.77	1.77	1.77	1.77	1.77	1.77
5120	1.80	1.80	1.80	1.80	1.80	1.80	1.70	1.70	1.70
5632	1.81	1.81	1.81	1.81	1.72	1.72	1.72	1.72	1.72
6144	1.83	1.75	1.75	1.75	1.75	1.75	1.75	1.75	1.75
6656	1.76	1.76	1.76	1.76	1.76	1.76	1.76	1.76	1.76
7168	1.78	1.78	1.78	1.78	1.78	1.78	1.78	1.78	1.78
7680	1.80	1.80	1.80	1.80	1.80	1.80	1.80	1.73	1.73
8192	1.81	1.81	1.81	1.81	1.81	1.75	1.75	1.75	1.75
10240	1.80	1.80	1.80	1.80	1.80	1.80	1.80	1.75	1.75
12288	1.83	1.83	1.79	1.79	1.79	1.79	1.79	1.79	1.75
14336	1.82	1.82	1.82	1.78	1.78	1.78	1.78	1.78	1.78
16384	1.81	1.81	1.81	1.81	1.81	1.78	1.78	1.78	1.78
18432	1.83	1.83	1.80	1.80	1.80	1.80	1.77	1.77	1.77
20480	1.82	1.82	1.82	1.80	1.80	1.80	1.80	1.77	1.77
22528	1.84	1.81	1.81	1.81	1.79	1.79	1.79	1.79	1.77
24576	1.83	1.83	1.81	1.81	1.81	1.79	1.79	1.79	1.77
26624	1.82	1.82	1.82	1.80	1.80	1.80	1.78	1.78	1.78
28672	1.83	1.82	1.82	1.82	1.80	1.80	1.78	1.78	1.78
30720	1.83	1.83	1.81	1.81	1.80	1.80	1.80	1.78	1.78
32768	1.82	1.82	1.81	1.81	1.81	1.79	1.79	1.78	1.78

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	287	288	289	291	292	295	296	297	298
512	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1024	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
1536	1.66	1.66	1.66	1.66	1.66	1.66	1.66	1.66	1.66
2048	1.75	1.75	1.75	1.75	1.50	1.50	1.50	1.50	1.50
2560	1.60	1.60	1.60	1.60	1.60	1.60	1.60	1.60	1.60
3072	1.66	1.66	1.66	1.66	1.66	1.66	1.66	1.66	1.66
3584	1.71	1.71	1.71	1.71	1.71	1.71	1.71	1.71	1.57
4096	1.75	1.75	1.75	1.75	1.62	1.62	1.62	1.62	1.62
4608	1.77	1.66	1.66	1.66	1.66	1.66	1.66	1.66	1.66
5120	1.70	1.70	1.70	1.70	1.70	1.70	1.70	1.70	1.70
5632	1.72	1.72	1.72	1.72	1.72	1.72	1.63	1.63	1.63
6144	1.75	1.75	1.75	1.75	1.75	1.66	1.66	1.66	1.66
6656	1.76	1.76	1.69	1.69	1.69	1.69	1.69	1.69	1.69
7168	1.71	1.71	1.71	1.71	1.71	1.71	1.71	1.71	1.71
7680	1.73	1.73	1.73	1.73	1.73	1.73	1.66	1.66	1.66
8192	1.75	1.75	1.75	1.75	1.75	1.68	1.68	1.68	1.68
10240	1.75	1.75	1.75	1.75	1.75	1.70	1.70	1.70	1.70
12288	1.75	1.75	1.75	1.75	1.75	1.70	1.70	1.70	1.70
14336	1.75	1.75	1.75	1.75	1.75	1.71	1.71	1.71	1.71
16384	1.78	1.75	1.75	1.75	1.75	1.71	1.71	1.71	1.68
18432	1.77	1.75	1.75	1.75	1.75	1.72	1.72	1.72	1.69
20480	1.77	1.77	1.75	1.75	1.75	1.72	1.72	1.70	1.70
22528	1.77	1.77	1.75	1.75	1.75	1.72	1.72	1.70	1.70
24576	1.77	1.77	1.77	1.75	1.75	1.72	1.70	1.70	1.70
26624	1.76	1.76	1.76	1.75	1.75	1.73	1.71	1.71	1.71
28672	1.76	1.76	1.76	1.75	1.75	1.73	1.71	1.71	1.71
30720	1.78	1.76	1.76	1.75	1.75	1.73	1.71	1.71	1.71
32768	1.78	1.76	1.76	1.75	1.75	1.73	1.71	1.71	1.70

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	299	300	301	302	303	304	305	306	307
512	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1024	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
1536	1.66	1.66	1.66	1.66	1.66	1.66	1.66	1.33	1.33
2048	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
2560	1.60	1.60	1.60	1.60	1.60	1.60	1.60	1.60	1.60
3072	1.66	1.66	1.66	1.66	1.66	1.66	1.66	1.66	1.50
3584	1.57	1.57	1.57	1.57	1.57	1.57	1.57	1.57	1.57
4096	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62
4608	1.66	1.66	1.66	1.66	1.66	1.66	1.66	1.66	1.55
5120	1.70	1.70	1.60	1.60	1.60	1.60	1.60	1.60	1.60
5632	1.63	1.63	1.63	1.63	1.63	1.63	1.63	1.63	1.63
6144	1.66	1.66	1.66	1.66	1.66	1.66	1.66	1.66	1.58
6656	1.69	1.69	1.69	1.69	1.61	1.61	1.61	1.61	1.61
7168	1.64	1.64	1.64	1.64	1.64	1.64	1.64	1.64	1.64
7680	1.66	1.66	1.66	1.66	1.66	1.66	1.66	1.66	1.60
8192	1.68	1.68	1.68	1.68	1.68	1.62	1.62	1.62	1.62
10240	1.70	1.70	1.65	1.65	1.65	1.65	1.65	1.65	1.65
12288	1.70	1.66	1.66	1.66	1.66	1.66	1.66	1.66	1.62
14336	1.67	1.67	1.67	1.67	1.67	1.67	1.64	1.64	1.64
16384	1.68	1.68	1.68	1.68	1.68	1.65	1.65	1.65	1.65
18432	1.69	1.69	1.69	1.69	1.66	1.66	1.66	1.66	1.66
20480	1.70	1.70	1.70	1.67	1.67	1.67	1.67	1.65	1.65
22528	1.70	1.70	1.68	1.68	1.68	1.68	1.65	1.65	1.65
24576	1.70	1.68	1.68	1.68	1.68	1.66	1.66	1.66	1.66
26624	1.71	1.69	1.69	1.69	1.67	1.67	1.67	1.65	1.65
28672	1.69	1.69	1.69	1.67	1.67	1.67	1.66	1.66	1.66
30720	1.70	1.70	1.70	1.68	1.68	1.68	1.66	1.66	1.66
32768	1.70	1.70	1.68	1.68	1.68	1.67	1.67	1.67	1.65

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	308	309	310	311	312	313	314	316	317
512	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1024	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
1536	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33
2048	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
2560	1.60	1.60	1.60	1.60	1.60	1.60	1.60	1.60	1.60
3072	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
3584	1.57	1.57	1.57	1.57	1.57	1.57	1.57	1.57	1.57
4096	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.50	1.50
4608	1.55	1.55	1.55	1.55	1.55	1.55	1.55	1.55	1.55
5120	1.60	1.60	1.60	1.60	1.60	1.60	1.60	1.60	1.60
5632	1.63	1.63	1.63	1.63	1.63	1.54	1.54	1.54	1.54
6144	1.58	1.58	1.58	1.58	1.58	1.58	1.58	1.58	1.58
6656	1.61	1.61	1.61	1.61	1.61	1.61	1.61	1.61	1.53
7168	1.64	1.64	1.64	1.64	1.57	1.57	1.57	1.57	1.57
7680	1.60	1.60	1.60	1.60	1.60	1.60	1.60	1.60	1.60
8192	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.56	1.56
10240	1.65	1.65	1.65	1.60	1.60	1.60	1.60	1.60	1.60
12288	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.58	1.58
14336	1.64	1.64	1.64	1.64	1.60	1.60	1.60	1.60	1.60
16384	1.65	1.62	1.62	1.62	1.62	1.62	1.62	1.59	1.59
18432	1.63	1.63	1.63	1.63	1.63	1.61	1.61	1.61	1.61
20480	1.65	1.65	1.65	1.62	1.62	1.62	1.62	1.60	1.60
22528	1.65	1.63	1.63	1.63	1.63	1.61	1.61	1.61	1.61
24576	1.64	1.64	1.64	1.62	1.62	1.62	1.62	1.60	1.60
26624	1.65	1.65	1.63	1.63	1.63	1.63	1.61	1.61	1.59
28672	1.66	1.64	1.64	1.64	1.62	1.62	1.62	1.60	1.60
30720	1.65	1.65	1.65	1.63	1.63	1.63	1.61	1.61	1.60
32768	1.65	1.65	1.64	1.64	1.62	1.62	1.62	1.60	1.60

CI Size	Logical Block Size								
	318	319	320	321	322	323	324	325	326
512	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1024	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
1536	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33
2048	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
2560	1.60	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40
3072	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
3584	1.57	1.57	1.57	1.57	1.57	1.57	1.57	1.42	1.42
4096	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
4608	1.55	1.55	1.55	1.55	1.55	1.55	1.55	1.55	1.55
5120	1.60	1.60	1.50	1.50	1.50	1.50	1.50	1.50	1.50
5632	1.54	1.54	1.54	1.54	1.54	1.54	1.54	1.54	1.54
6144	1.58	1.58	1.58	1.58	1.58	1.50	1.50	1.50	1.50
6656	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.53
7168	1.57	1.57	1.57	1.57	1.57	1.57	1.57	1.57	1.50
7680	1.60	1.60	1.53	1.53	1.53	1.53	1.53	1.53	1.53
8192	1.56	1.56	1.56	1.56	1.56	1.56	1.56	1.56	1.56
10240	1.60	1.60	1.55	1.55	1.55	1.55	1.55	1.55	1.55
12288	1.58	1.58	1.58	1.58	1.58	1.58	1.54	1.54	1.54
14336	1.60	1.57	1.57	1.57	1.57	1.57	1.57	1.57	1.53
16384	1.59	1.59	1.59	1.59	1.56	1.56	1.56	1.56	1.56
18432	1.58	1.58	1.58	1.58	1.58	1.58	1.55	1.55	1.55
20480	1.60	1.60	1.57	1.57	1.57	1.57	1.57	1.55	1.55
22528	1.59	1.59	1.59	1.59	1.56	1.56	1.56	1.56	1.56
24576	1.60	1.60	1.58	1.58	1.58	1.58	1.56	1.56	1.56
26624	1.59	1.59	1.59	1.57	1.57	1.57	1.57	1.55	1.55
28672	1.60	1.58	1.58	1.58	1.58	1.57	1.57	1.57	1.55
30720	1.60	1.60	1.58	1.58	1.58	1.58	1.56	1.56	1.56
32768	1.60	1.59	1.59	1.59	1.57	1.57	1.57	1.56	1.56

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	327	328	329	330	331	332	333	334	335
512	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1024	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
1536	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33
2048	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
2560	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40
3072	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
3584	1.42	1.42	1.42	1.42	1.42	1.42	1.42	1.42	1.42
4096	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
4608	1.55	1.55	1.44	1.44	1.44	1.44	1.44	1.44	1.44
5120	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
5632	1.54	1.54	1.54	1.54	1.45	1.45	1.45	1.45	1.45
6144	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
6656	1.53	1.53	1.53	1.53	1.53	1.53	1.46	1.46	1.46
7168	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
7680	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.46	1.46
8192	1.56	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
10240	1.55	1.55	1.55	1.55	1.50	1.50	1.50	1.50	1.50
12288	1.54	1.54	1.54	1.54	1.54	1.50	1.50	1.50	1.50
14336	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.50	1.50
16384	1.56	1.53	1.53	1.53	1.53	1.53	1.53	1.53	1.50
18432	1.55	1.55	1.52	1.52	1.52	1.52	1.52	1.52	1.50
20480	1.55	1.55	1.55	1.55	1.52	1.52	1.52	1.52	1.52
22528	1.54	1.54	1.54	1.54	1.54	1.52	1.52	1.52	1.52
24576	1.56	1.54	1.54	1.54	1.54	1.52	1.52	1.52	1.52
26624	1.55	1.55	1.53	1.53	1.53	1.53	1.51	1.51	1.51
28672	1.55	1.55	1.55	1.53	1.53	1.53	1.53	1.51	1.51
30720	1.55	1.55	1.55	1.55	1.53	1.53	1.53	1.51	1.51
32768	1.56	1.54	1.54	1.54	1.53	1.53	1.53	1.53	1.51

CI Size	Logical Block Size								
	336	337	338	339	340	341	344	345	346
512	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1024	1.50	1.50	1.50	1.00	1.00	1.00	1.00	1.00	1.00
1536	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33
2048	1.50	1.50	1.50	1.50	1.25	1.25	1.25	1.25	1.25
2560	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40
3072	1.50	1.50	1.50	1.50	1.50	1.33	1.33	1.33	1.33
3584	1.42	1.42	1.42	1.42	1.42	1.42	1.42	1.42	1.42
4096	1.50	1.50	1.50	1.50	1.50	1.37	1.37	1.37	1.37
4608	1.44	1.44	1.44	1.44	1.44	1.44	1.44	1.44	1.44
5120	1.50	1.50	1.50	1.50	1.50	1.40	1.40	1.40	1.40
5632	1.45	1.45	1.45	1.45	1.45	1.45	1.45	1.45	1.45
6144	1.50	1.50	1.50	1.50	1.50	1.41	1.41	1.41	1.41
6656	1.46	1.46	1.46	1.46	1.46	1.46	1.46	1.46	1.46
7168	1.50	1.50	1.50	1.50	1.50	1.42	1.42	1.42	1.42
7680	1.46	1.46	1.46	1.46	1.46	1.46	1.46	1.46	1.46
8192	1.50	1.50	1.50	1.50	1.50	1.43	1.43	1.43	1.43
10240	1.50	1.50	1.50	1.50	1.50	1.50	1.45	1.45	1.45
12288	1.50	1.50	1.50	1.50	1.50	1.50	1.45	1.45	1.45
14336	1.50	1.50	1.50	1.50	1.50	1.50	1.46	1.46	1.46
16384	1.50	1.50	1.50	1.50	1.50	1.50	1.46	1.46	1.46
18432	1.50	1.50	1.50	1.50	1.50	1.50	1.47	1.47	1.47
20480	1.50	1.50	1.50	1.50	1.50	1.50	1.47	1.47	1.47
22528	1.52	1.50	1.50	1.50	1.50	1.50	1.47	1.47	1.47
24576	1.52	1.50	1.50	1.50	1.50	1.50	1.47	1.47	1.47
26624	1.51	1.50	1.50	1.50	1.50	1.50	1.48	1.48	1.46
28672	1.51	1.51	1.50	1.50	1.50	1.50	1.48	1.48	1.46
30720	1.51	1.51	1.50	1.50	1.50	1.50	1.48	1.48	1.46
32768	1.51	1.51	1.50	1.50	1.50	1.50	1.48	1.46	1.46

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	347	348	349	350	351	352	353	354	355
512	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1024	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1536	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33
2048	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25
2560	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40
3072	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33
3584	1.42	1.42	1.42	1.42	1.42	1.42	1.42	1.42	1.42
4096	1.37	1.37	1.37	1.37	1.37	1.37	1.37	1.37	1.37
4608	1.44	1.44	1.44	1.44	1.44	1.44	1.44	1.33	1.33
5120	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40
5632	1.45	1.45	1.45	1.45	1.45	1.36	1.36	1.36	1.36
6144	1.41	1.41	1.41	1.41	1.41	1.41	1.41	1.41	1.41
6656	1.46	1.46	1.46	1.38	1.38	1.38	1.38	1.38	1.38
7168	1.42	1.42	1.42	1.42	1.42	1.42	1.42	1.42	1.42
7680	1.46	1.46	1.40	1.40	1.40	1.40	1.40	1.40	1.40
8192	1.43	1.43	1.43	1.43	1.43	1.43	1.43	1.43	1.43
10240	1.45	1.45	1.45	1.45	1.45	1.45	1.40	1.40	1.40
12288	1.45	1.45	1.45	1.45	1.41	1.41	1.41	1.41	1.41
14336	1.46	1.46	1.46	1.42	1.42	1.42	1.42	1.42	1.42
16384	1.46	1.46	1.43	1.43	1.43	1.43	1.43	1.43	1.43
18432	1.47	1.44	1.44	1.44	1.44	1.44	1.44	1.44	1.41
20480	1.45	1.45	1.45	1.45	1.45	1.45	1.42	1.42	1.42
22528	1.45	1.45	1.45	1.45	1.45	1.43	1.43	1.43	1.43
24576	1.45	1.45	1.45	1.45	1.43	1.43	1.43	1.43	1.43
26624	1.46	1.46	1.46	1.46	1.44	1.44	1.44	1.44	1.42
28672	1.46	1.46	1.46	1.44	1.44	1.44	1.44	1.42	1.42
30720	1.46	1.46	1.45	1.45	1.45	1.45	1.43	1.43	1.43
32768	1.46	1.46	1.45	1.45	1.45	1.45	1.43	1.43	1.43

CI Size	Logical Block Size								
	356	357	358	359	360	361	362	363	364
512	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1024	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1536	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33
2048	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25
2560	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40
3072	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33
3584	1.42	1.42	1.28	1.28	1.28	1.28	1.28	1.28	1.28
4096	1.37	1.37	1.37	1.37	1.37	1.37	1.37	1.37	1.37
4608	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33
5120	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40
5632	1.36	1.36	1.36	1.36	1.36	1.36	1.36	1.36	1.36
6144	1.41	1.41	1.41	1.41	1.41	1.33	1.33	1.33	1.33
6656	1.38	1.38	1.38	1.38	1.38	1.38	1.38	1.38	1.38
7168	1.42	1.42	1.35	1.35	1.35	1.35	1.35	1.35	1.35
7680	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40
8192	1.37	1.37	1.37	1.37	1.37	1.37	1.37	1.37	1.37
10240	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40
12288	1.41	1.41	1.41	1.41	1.41	1.41	1.37	1.37	1.37
14336	1.42	1.42	1.42	1.39	1.39	1.39	1.39	1.39	1.39
16384	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.37
18432	1.41	1.41	1.41	1.41	1.41	1.41	1.38	1.38	1.38
20480	1.42	1.42	1.42	1.42	1.40	1.40	1.40	1.40	1.40
22528	1.43	1.43	1.40	1.40	1.40	1.40	1.40	1.40	1.38
24576	1.43	1.41	1.41	1.41	1.41	1.41	1.39	1.39	1.39
26624	1.42	1.42	1.42	1.42	1.40	1.40	1.40	1.40	1.40
28672	1.42	1.42	1.42	1.41	1.41	1.41	1.41	1.39	1.39
30720	1.43	1.43	1.41	1.41	1.41	1.41	1.40	1.40	1.40
32768	1.43	1.42	1.42	1.42	1.40	1.40	1.40	1.40	1.39

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	365	366	367	368	369	370	371	372	374
512	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1024	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1536	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33
2048	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25
2560	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20
3072	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33
3584	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28
4096	1.37	1.37	1.37	1.37	1.37	1.37	1.37	1.25	1.25
4608	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33
5120	1.40	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30
5632	1.36	1.36	1.36	1.36	1.36	1.36	1.36	1.36	1.36
6144	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33
6656	1.38	1.38	1.38	1.38	1.38	1.30	1.30	1.30	1.30
7168	1.35	1.35	1.35	1.35	1.35	1.35	1.35	1.35	1.35
7680	1.40	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33
8192	1.37	1.37	1.37	1.37	1.37	1.37	1.37	1.31	1.31
10240	1.40	1.35	1.35	1.35	1.35	1.35	1.35	1.35	1.35
12288	1.37	1.37	1.37	1.37	1.37	1.37	1.37	1.37	1.33
14336	1.39	1.39	1.39	1.35	1.35	1.35	1.35	1.35	1.35
16384	1.37	1.37	1.37	1.37	1.37	1.37	1.37	1.37	1.34
18432	1.38	1.38	1.38	1.38	1.36	1.36	1.36	1.36	1.36
20480	1.40	1.37	1.37	1.37	1.37	1.37	1.37	1.37	1.35
22528	1.38	1.38	1.38	1.38	1.38	1.36	1.36	1.36	1.36
24576	1.39	1.39	1.37	1.37	1.37	1.37	1.37	1.37	1.35
26624	1.38	1.38	1.38	1.38	1.38	1.36	1.36	1.36	1.36
28672	1.39	1.39	1.39	1.37	1.37	1.37	1.37	1.37	1.35
30720	1.40	1.38	1.38	1.38	1.38	1.38	1.36	1.36	1.36
32768	1.39	1.39	1.39	1.39	1.37	1.37	1.37	1.37	1.35

CI Size	Logical Block Size								
	375	376	377	378	379	380	381	382	383
512	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1024	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1536	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.00	1.00
2048	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25
2560	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20
3072	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.16
3584	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28
4096	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25
4608	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33
5120	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30
5632	1.27	1.27	1.27	1.27	1.27	1.27	1.27	1.27	1.27
6144	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33
6656	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30
7168	1.35	1.35	1.28	1.28	1.28	1.28	1.28	1.28	1.28
7680	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33
8192	1.31	1.31	1.31	1.31	1.31	1.31	1.31	1.31	1.31
10240	1.35	1.35	1.35	1.35	1.30	1.30	1.30	1.30	1.30
12288	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33
14336	1.35	1.35	1.35	1.32	1.32	1.32	1.32	1.32	1.32
16384	1.34	1.34	1.34	1.34	1.34	1.34	1.31	1.31	1.31
18432	1.36	1.33	1.33	1.33	1.33	1.33	1.33	1.33	1.33
20480	1.35	1.35	1.35	1.35	1.35	1.32	1.32	1.32	1.32
22528	1.36	1.34	1.34	1.34	1.34	1.34	1.34	1.31	1.31
24576	1.35	1.35	1.35	1.33	1.33	1.33	1.33	1.33	1.33
26624	1.34	1.34	1.34	1.34	1.34	1.34	1.32	1.32	1.32
28672	1.35	1.35	1.35	1.33	1.33	1.33	1.33	1.33	1.32
30720	1.35	1.35	1.35	1.35	1.35	1.33	1.33	1.33	1.33
32768	1.35	1.35	1.34	1.34	1.34	1.34	1.32	1.32	1.32

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	385	386	387	388	389	390	391	392	393
512	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1024	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1536	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2048	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25
2560	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20
3072	1.16	1.16	1.16	1.16	1.16	1.16	1.16	1.16	1.16
3584	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28
4096	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25
4608	1.22	1.22	1.22	1.22	1.22	1.22	1.22	1.22	1.22
5120	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30
5632	1.27	1.27	1.27	1.27	1.27	1.27	1.27	1.27	1.27
6144	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25
6656	1.30	1.30	1.30	1.30	1.30	1.30	1.23	1.23	1.23
7168	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28	1.28
7680	1.26	1.26	1.26	1.26	1.26	1.26	1.26	1.26	1.26
8192	1.31	1.31	1.31	1.31	1.31	1.25	1.25	1.25	1.25
10240	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30
12288	1.29	1.29	1.29	1.29	1.29	1.29	1.29	1.29	1.29
14336	1.32	1.32	1.32	1.28	1.28	1.28	1.28	1.28	1.28
16384	1.31	1.31	1.31	1.31	1.31	1.28	1.28	1.28	1.28
18432	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.27	1.27
20480	1.32	1.32	1.30	1.30	1.30	1.30	1.30	1.30	1.30
22528	1.31	1.31	1.31	1.31	1.29	1.29	1.29	1.29	1.29
24576	1.31	1.31	1.31	1.31	1.31	1.29	1.29	1.29	1.29
26624	1.32	1.30	1.30	1.30	1.30	1.30	1.30	1.28	1.28
28672	1.32	1.32	1.32	1.30	1.30	1.30	1.30	1.30	1.28
30720	1.31	1.31	1.31	1.31	1.30	1.30	1.30	1.30	1.30
32768	1.32	1.31	1.31	1.31	1.31	1.29	1.29	1.29	1.29

CI Size	Logical Block Size								
	394	395	396	397	398	399	400	401	402
512	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1024	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1536	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2048	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25
2560	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20
3072	1.16	1.16	1.16	1.16	1.16	1.16	1.16	1.16	1.16
3584	1.28	1.28	1.28	1.28	1.14	1.14	1.14	1.14	1.14
4096	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25
4608	1.22	1.22	1.22	1.22	1.22	1.22	1.22	1.22	1.22
5120	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20
5632	1.27	1.27	1.27	1.27	1.27	1.27	1.27	1.27	1.18
6144	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25
6656	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23
7168	1.28	1.28	1.28	1.28	1.21	1.21	1.21	1.21	1.21
7680	1.26	1.26	1.26	1.26	1.26	1.26	1.26	1.26	1.26
8192	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25
10240	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25
12288	1.29	1.29	1.29	1.25	1.25	1.25	1.25	1.25	1.25
14336	1.28	1.28	1.28	1.28	1.25	1.25	1.25	1.25	1.25
16384	1.28	1.28	1.28	1.28	1.28	1.28	1.25	1.25	1.25
18432	1.27	1.27	1.27	1.27	1.27	1.27	1.27	1.25	1.25
20480	1.27	1.27	1.27	1.27	1.27	1.27	1.27	1.27	1.25
22528	1.29	1.29	1.27	1.27	1.27	1.27	1.27	1.27	1.27
24576	1.29	1.29	1.29	1.27	1.27	1.27	1.27	1.27	1.27
26624	1.28	1.28	1.28	1.28	1.26	1.26	1.26	1.26	1.26
28672	1.28	1.28	1.28	1.28	1.28	1.26	1.26	1.26	1.26
30720	1.28	1.28	1.28	1.28	1.28	1.26	1.26	1.26	1.26
32768	1.29	1.28	1.28	1.28	1.28	1.28	1.26	1.26	1.26

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	403	404	407	408	409	414	415	416	417
512	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1024	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1536	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2048	1.25	1.25	1.25	1.00	1.00	1.00	1.00	1.00	1.00
2560	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20
3072	1.16	1.16	1.16	1.16	1.16	1.16	1.16	1.16	1.16
3584	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14
4096	1.25	1.25	1.25	1.25	1.12	1.12	1.12	1.12	1.12
4608	1.22	1.22	1.22	1.22	1.22	1.22	1.22	1.22	1.22
5120	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20
5632	1.18	1.18	1.18	1.18	1.18	1.18	1.18	1.18	1.18
6144	1.25	1.25	1.25	1.25	1.16	1.16	1.16	1.16	1.16
6656	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.15	1.15
7168	1.21	1.21	1.21	1.21	1.21	1.21	1.21	1.21	1.21
7680	1.26	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20
8192	1.25	1.25	1.25	1.25	1.25	1.18	1.18	1.18	1.18
10240	1.25	1.25	1.25	1.25	1.25	1.20	1.20	1.20	1.20
12288	1.25	1.25	1.25	1.25	1.25	1.20	1.20	1.20	1.20
14336	1.25	1.25	1.25	1.25	1.25	1.21	1.21	1.21	1.21
16384	1.25	1.25	1.25	1.25	1.25	1.21	1.21	1.21	1.21
18432	1.25	1.25	1.25	1.25	1.25	1.22	1.22	1.22	1.22
20480	1.25	1.25	1.25	1.25	1.25	1.22	1.22	1.22	1.22
22528	1.25	1.25	1.25	1.25	1.25	1.22	1.22	1.22	1.22
24576	1.25	1.25	1.25	1.25	1.25	1.22	1.22	1.22	1.20
26624	1.26	1.25	1.25	1.25	1.25	1.23	1.23	1.21	1.21
28672	1.26	1.25	1.25	1.25	1.25	1.23	1.23	1.21	1.21
30720	1.26	1.26	1.25	1.25	1.25	1.23	1.23	1.21	1.21
32768	1.26	1.26	1.25	1.25	1.25	1.23	1.21	1.21	1.21

CI Size	Logical Block Size								
	418	419	420	421	422	423	424	425	426
512	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1024	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1536	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2048	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2560	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.00
3072	1.16	1.16	1.16	1.16	1.16	1.16	1.16	1.16	1.16
3584	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14
4096	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.12
4608	1.22	1.11	1.11	1.11	1.11	1.11	1.11	1.11	1.11
5120	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.10
5632	1.18	1.18	1.18	1.18	1.18	1.18	1.18	1.18	1.18
6144	1.16	1.16	1.16	1.16	1.16	1.16	1.16	1.16	1.16
6656	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15
7168	1.21	1.21	1.21	1.21	1.14	1.14	1.14	1.14	1.14
7680	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20
8192	1.18	1.18	1.18	1.18	1.18	1.18	1.18	1.18	1.18
10240	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20
12288	1.20	1.20	1.20	1.20	1.20	1.20	1.16	1.16	1.16
14336	1.21	1.21	1.21	1.21	1.17	1.17	1.17	1.17	1.17
16384	1.21	1.21	1.18	1.18	1.18	1.18	1.18	1.18	1.18
18432	1.22	1.19	1.19	1.19	1.19	1.19	1.19	1.19	1.19
20480	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20
22528	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.18	1.18
24576	1.20	1.20	1.20	1.20	1.20	1.20	1.18	1.18	1.18
26624	1.21	1.21	1.21	1.21	1.21	1.19	1.19	1.19	1.19
28672	1.21	1.21	1.21	1.21	1.19	1.19	1.19	1.19	1.19
30720	1.21	1.21	1.21	1.20	1.20	1.20	1.20	1.20	1.20
32768	1.21	1.21	1.20	1.20	1.20	1.20	1.20	1.20	1.18

CI Size	Number of Logical Blocks per FBA Block — Logical Block Size 427	428	429	430	431	432	433	434	435
512	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1024	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1536	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2048	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2560	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
3072	1.16	1.16	1.16	1.16	1.16	1.16	1.16	1.16	1.16
3584	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14
4096	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.12
4608	1.11	1.11	1.11	1.11	1.11	1.11	1.11	1.11	1.11
5120	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10
5632	1.18	1.18	1.18	1.18	1.18	1.18	1.09	1.09	1.09
6144	1.16	1.16	1.16	1.16	1.16	1.16	1.16	1.16	1.16
6656	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15
7168	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14
7680	1.13	1.13	1.13	1.13	1.13	1.13	1.13	1.13	1.13
8192	1.18	1.18	1.18	1.18	1.12	1.12	1.12	1.12	1.12
10240	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15
12288	1.16	1.16	1.16	1.16	1.16	1.16	1.16	1.16	1.16
14336	1.17	1.17	1.17	1.17	1.17	1.17	1.17	1.17	1.14
16384	1.18	1.18	1.18	1.18	1.15	1.15	1.15	1.15	1.15
18432	1.19	1.19	1.16	1.16	1.16	1.16	1.16	1.16	1.16
20480	1.17	1.17	1.17	1.17	1.17	1.17	1.17	1.17	1.17
22528	1.18	1.18	1.18	1.18	1.18	1.18	1.18	1.15	1.15
24576	1.18	1.18	1.18	1.18	1.16	1.16	1.16	1.16	1.16
26624	1.19	1.19	1.19	1.17	1.17	1.17	1.17	1.17	1.17
28672	1.19	1.17	1.17	1.17	1.17	1.17	1.17	1.17	1.16
30720	1.18	1.18	1.18	1.18	1.18	1.18	1.16	1.16	1.16
32768	1.18	1.18	1.18	1.18	1.18	1.17	1.17	1.17	1.17

CI Size	Number of Logical Blocks per FBA Block — Logical Block Size 436	437	438	440	441	442	443	444	445
512	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1024	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1536	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2048	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2560	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
3072	1.16	1.16	1.00	1.00	1.00	1.00	1.00	1.00	1.00
3584	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14
4096	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.12
4608	1.11	1.11	1.11	1.11	1.11	1.11	1.11	1.11	1.11
5120	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10
5632	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09
6144	1.16	1.16	1.16	1.08	1.08	1.08	1.08	1.08	1.08
6656	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.07	1.07
7168	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14
7680	1.13	1.13	1.13	1.13	1.13	1.13	1.13	1.13	1.13
8192	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.12
10240	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.10
12288	1.16	1.16	1.16	1.12	1.12	1.12	1.12	1.12	1.12
14336	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14
16384	1.15	1.15	1.15	1.15	1.15	1.15	1.12	1.12	1.12
18432	1.16	1.16	1.16	1.13	1.13	1.13	1.13	1.13	1.13
20480	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15	1.15
22528	1.15	1.15	1.15	1.15	1.15	1.13	1.13	1.13	1.13
24576	1.16	1.16	1.16	1.14	1.14	1.14	1.14	1.14	1.14
26624	1.17	1.15	1.15	1.15	1.15	1.15	1.15	1.13	1.13
28672	1.16	1.16	1.16	1.16	1.14	1.14	1.14	1.14	1.14
30720	1.16	1.16	1.16	1.15	1.15	1.15	1.15	1.15	1.15
32768	1.17	1.15	1.15	1.15	1.15	1.15	1.14	1.14	1.14

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	446	447	448	449	450	451	454	458	459
512	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1024	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1536	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2048	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2560	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
3072	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
3584	1.14	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
4096	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.00	1.00
4608	1.11	1.11	1.11	1.11	1.11	1.11	1.11	1.11	1.11
5120	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10
5632	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09
6144	1.08	1.08	1.08	1.08	1.08	1.08	1.08	1.08	1.08
6656	1.07	1.07	1.07	1.07	1.07	1.07	1.07	1.07	1.07
7168	1.14	1.14	1.07	1.07	1.07	1.07	1.07	1.07	1.07
7680	1.13	1.13	1.13	1.13	1.13	1.13	1.06	1.06	1.06
8192	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.06	1.06
10240	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10
12288	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.08	1.08
14336	1.14	1.14	1.10	1.10	1.10	1.10	1.10	1.10	1.10
16384	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.09	1.09
18432	1.13	1.13	1.13	1.13	1.11	1.11	1.11	1.11	1.11
20480	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.10	1.10
22528	1.13	1.13	1.13	1.13	1.13	1.11	1.11	1.11	1.11
24576	1.14	1.12	1.12	1.12	1.12	1.12	1.12	1.10	1.10
26624	1.13	1.13	1.13	1.13	1.13	1.13	1.11	1.11	1.09
28672	1.14	1.14	1.12	1.12	1.12	1.12	1.12	1.10	1.10
30720	1.13	1.13	1.13	1.13	1.13	1.13	1.11	1.11	1.10
32768	1.14	1.14	1.14	1.12	1.12	1.12	1.12	1.10	1.10

CI Size	Logical Block Size								
	460	461	462	463	464	465	466	467	468
512	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1024	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1536	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2048	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2560	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
3072	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
3584	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
4096	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
4608	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
5120	1.10	1.10	1.10	1.10	1.10	1.00	1.00	1.00	1.00
5632	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09
6144	1.08	1.08	1.08	1.08	1.08	1.08	1.08	1.08	1.08
6656	1.07	1.07	1.07	1.07	1.07	1.07	1.07	1.07	1.07
7168	1.07	1.07	1.07	1.07	1.07	1.07	1.07	1.07	1.07
7680	1.06	1.06	1.06	1.06	1.06	1.06	1.06	1.06	1.06
8192	1.06	1.06	1.06	1.06	1.06	1.06	1.06	1.06	1.06
10240	1.10	1.10	1.10	1.10	1.10	1.10	1.05	1.05	1.05
12288	1.08	1.08	1.08	1.08	1.08	1.08	1.08	1.08	1.08
14336	1.10	1.10	1.10	1.07	1.07	1.07	1.07	1.07	1.07
16384	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.06
18432	1.11	1.08	1.08	1.08	1.08	1.08	1.08	1.08	1.08
20480	1.10	1.10	1.10	1.10	1.10	1.10	1.07	1.07	1.07
22528	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09
24576	1.10	1.10	1.10	1.10	1.08	1.08	1.08	1.08	1.08
26624	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.07	1.07
28672	1.10	1.10	1.10	1.08	1.08	1.08	1.08	1.08	1.08
30720	1.10	1.10	1.10	1.10	1.10	1.10	1.08	1.08	1.08
32768	1.10	1.10	1.09	1.09	1.09	1.09	1.09	1.09	1.07

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	469	471	472	474	475	476	477	479	481
512	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1024	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1536	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2048	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2560	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
3072	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
3584	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
4096	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
4608	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
5120	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
5632	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
6144	1.08	1.08	1.00	1.00	1.00	1.00	1.00	1.00	1.00
6656	1.07	1.07	1.07	1.07	1.00	1.00	1.00	1.00	1.00
7168	1.07	1.07	1.07	1.07	1.07	1.07	1.07	1.00	1.00
7680	1.06	1.06	1.06	1.06	1.06	1.06	1.06	1.06	1.00
8192	1.06	1.06	1.06	1.06	1.06	1.06	1.06	1.06	1.06
10240	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
12288	1.08	1.08	1.08	1.04	1.04	1.04	1.04	1.04	1.04
14336	1.07	1.07	1.07	1.07	1.07	1.07	1.07	1.03	1.03
16384	1.06	1.06	1.06	1.06	1.06	1.06	1.06	1.06	1.06
18432	1.08	1.08	1.08	1.05	1.05	1.05	1.05	1.05	1.05
20480	1.07	1.07	1.07	1.07	1.07	1.07	1.05	1.05	1.05
22528	1.09	1.06	1.06	1.06	1.06	1.06	1.06	1.06	1.04
24576	1.08	1.08	1.08	1.06	1.06	1.06	1.06	1.06	1.06
26624	1.07	1.07	1.07	1.07	1.07	1.05	1.05	1.05	1.05
28672	1.08	1.07	1.07	1.07	1.07	1.07	1.07	1.05	1.05
30720	1.08	1.08	1.08	1.06	1.06	1.06	1.06	1.06	1.05
32768	1.07	1.07	1.07	1.07	1.06	1.06	1.06	1.06	1.06

CI Size	Logical Block Size								
	483	484	485	487	488	489	491	492	494
512	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1024	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1536	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2048	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2560	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
3072	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
3584	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
4096	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
4608	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
5120	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
5632	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
6144	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
6656	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
7168	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
7680	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
8192	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
10240	1.05	1.05	1.05	1.05	1.00	1.00	1.00	1.00	1.00
12288	1.04	1.04	1.04	1.04	1.04	1.04	1.04	1.00	1.00
14336	1.03	1.03	1.03	1.03	1.03	1.03	1.03	1.03	1.03
16384	1.03	1.03	1.03	1.03	1.03	1.03	1.03	1.03	1.03
18432	1.05	1.05	1.02	1.02	1.02	1.02	1.02	1.02	1.02
20480	1.05	1.05	1.05	1.05	1.02	1.02	1.02	1.02	1.02
22528	1.04	1.04	1.04	1.04	1.04	1.04	1.02	1.02	1.02
24576	1.04	1.04	1.04	1.04	1.04	1.04	1.04	1.02	1.02
26624	1.05	1.03	1.03	1.03	1.03	1.03	1.03	1.03	1.01
28672	1.05	1.05	1.05	1.03	1.03	1.03	1.03	1.03	1.03
30720	1.05	1.05	1.05	1.05	1.03	1.03	1.03	1.03	1.03
32768	1.04	1.04	1.04	1.04	1.04	1.03	1.03	1.03	1.03

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	495	496	497	499	500	501	502	503	507
512	1.00	1.00	1.00	1.00	1.00	1.00	1.00	———	———
1024	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1536	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2048	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2560	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
3072	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
3584	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
4096	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
4608	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
5120	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
5632	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
6144	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
6656	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
7168	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
7680	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
8192	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
10240	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
12288	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
14336	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
16384	1.03	1.03	1.00	1.00	1.00	1.00	1.00	1.00	1.00
18432	1.02	1.02	1.02	1.00	1.00	1.00	1.00	1.00	1.00
20480	1.02	1.02	1.02	1.02	1.00	1.00	1.00	1.00	1.00
22528	1.02	1.02	1.02	1.02	1.02	1.00	1.00	1.00	1.00
24576	1.02	1.02	1.02	1.02	1.02	1.02	1.00	1.00	1.00
26624	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.00	1.00
28672	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.00	1.00
30720	1.03	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.00
32768	1.03	1.03	1.01	1.01	1.01	1.01	1.01	1.01	1.00

CI Size	Logical Block Size								
	508	509	510	511	519	520	521	522	523
512	———	———	———	———	———	———	———	———	———
1024	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
1536	1.00	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
2048	1.00	1.00	0.75	0.75	0.75	0.75	0.75	0.75	0.75
2560	1.00	1.00	1.00	0.80	0.80	0.80	0.80	0.80	0.80
3072	1.00	1.00	1.00	0.83	0.83	0.83	0.83	0.83	0.83
3584	1.00	1.00	1.00	0.85	0.85	0.85	0.85	0.85	0.85
4096	1.00	1.00	1.00	0.87	0.87	0.87	0.87	0.87	0.87
4608	1.00	1.00	1.00	0.88	0.88	0.88	0.88	0.88	0.88
5120	1.00	1.00	1.00	1.00	0.90	0.90	0.90	0.90	0.90
5632	1.00	1.00	1.00	1.00	0.90	0.90	0.90	0.90	0.90
6144	1.00	1.00	1.00	1.00	0.91	0.91	0.91	0.91	0.91
6656	1.00	1.00	1.00	1.00	0.92	0.92	0.92	0.92	0.92
7168	1.00	1.00	1.00	1.00	0.92	0.92	0.92	0.92	0.92
7680	1.00	1.00	1.00	1.00	0.93	0.93	0.93	0.93	0.93
8192	1.00	1.00	1.00	1.00	0.93	0.93	0.93	0.93	0.93
10240	1.00	1.00	1.00	1.00	0.95	0.95	0.95	0.95	0.95
12288	1.00	1.00	1.00	1.00	0.95	0.95	0.95	0.95	0.95
14336	1.00	1.00	1.00	1.00	0.96	0.96	0.96	0.96	0.96
16384	1.00	1.00	1.00	1.00	0.96	0.96	0.96	0.96	0.96
18432	1.00	1.00	1.00	1.00	0.97	0.97	0.97	0.97	0.97
20480	1.00	1.00	1.00	1.00	0.97	0.97	0.97	0.97	0.97
22528	1.00	1.00	1.00	1.00	0.97	0.97	0.97	0.97	0.97
24576	1.00	1.00	1.00	1.00	0.97	0.97	0.97	0.97	0.95
26624	1.00	1.00	1.00	1.00	0.98	0.98	0.98	0.96	0.96
28672	1.00	1.00	1.00	1.00	0.98	0.98	0.98	0.96	0.96
30720	1.00	1.00	1.00	1.00	0.98	0.98	0.96	0.96	0.96
32768	1.00	1.00	1.00	1.00	0.98	0.96	0.96	0.96	0.96

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	524	526	528	529	530	532	533	534	536
512	———	———	———	———	———	———	———	———	———
1024	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
1536	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
2048	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
2560	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80
3072	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83
3584	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85
4096	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87
4608	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88
5120	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90
5632	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90
6144	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91
6656	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92
7168	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92
7680	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93
8192	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93
10240	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
12288	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.91	0.91
14336	0.96	0.96	0.96	0.96	0.96	0.92	0.92	0.92	0.92
16384	0.96	0.96	0.96	0.93	0.93	0.93	0.93	0.93	0.93
18432	0.97	0.97	0.94	0.94	0.94	0.94	0.94	0.94	0.94
20480	0.97	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
22528	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
24576	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.93
26624	0.96	0.96	0.96	0.96	0.96	0.96	0.94	0.94	0.94
28672	0.96	0.96	0.96	0.96	0.96	0.94	0.94	0.94	0.94
30720	0.96	0.96	0.96	0.96	0.95	0.95	0.95	0.95	0.95
32768	0.96	0.96	0.96	0.95	0.95	0.95	0.95	0.95	0.95

CI Size	Logical Block Size								
	537	538	540	541	543	545	547	548	549
512	———	———	———	———	———	———	———	———	———
1024	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
1536	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
2048	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
2560	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80
3072	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83
3584	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85
4096	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87
4608	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88
5120	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90
5632	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90
6144	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91
6656	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92
7168	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92
7680	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.86	0.86
8192	0.93	0.93	0.93	0.93	0.93	0.93	0.87	0.87	0.87
10240	0.95	0.95	0.90	0.90	0.90	0.90	0.90	0.90	0.90
12288	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91
14336	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92
16384	0.93	0.93	0.93	0.93	0.93	0.93	0.90	0.90	0.90
18432	0.94	0.94	0.94	0.94	0.91	0.91	0.91	0.91	0.91
20480	0.95	0.95	0.92	0.92	0.92	0.92	0.92	0.92	0.92
22528	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93
24576	0.93	0.93	0.93	0.93	0.93	0.93	0.91	0.91	0.91
26624	0.94	0.94	0.94	0.94	0.94	0.92	0.92	0.92	0.92
28672	0.94	0.94	0.94	0.92	0.92	0.92	0.92	0.92	0.92
30720	0.95	0.95	0.93	0.93	0.93	0.93	0.93	0.93	0.91
32768	0.95	0.93	0.93	0.93	0.93	0.93	0.92	0.92	0.92

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	550	551	553	554	555	557	558	562	564
512	——	——	——	——	——	——	——	——	——
1024	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
1536	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
2048	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
2560	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80
3072	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83
3584	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85
4096	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87
4608	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88
5120	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90
5632	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.81
6144	0.91	0.91	0.91	0.91	0.91	0.91	0.83	0.83	0.83
6656	0.92	0.92	0.92	0.84	0.84	0.84	0.84	0.84	0.84
7168	0.92	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85
7680	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86
8192	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87
10240	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90
12288	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.87	0.87
14336	0.92	0.92	0.89	0.89	0.89	0.89	0.89	0.89	0.89
16384	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90
18432	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.88	0.88
20480	0.92	0.92	0.92	0.90	0.90	0.90	0.90	0.90	0.90
22528	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.88
24576	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.89	0.89
26624	0.92	0.92	0.92	0.92	0.90	0.90	0.90	0.90	0.90
28672	0.92	0.92	0.91	0.91	0.91	0.91	0.91	0.91	0.89
30720	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.90	0.90
32768	0.92	0.92	0.92	0.92	0.92	0.90	0.90	0.90	0.90

CI Size	Logical Block Size								
	566	567	568	571	573	574	575	577	578
512	——	——	——	——	——	——	——	——	——
1024	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
1536	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
2048	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
2560	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80
3072	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83
3584	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85
4096	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87
4608	0.88	0.88	0.88	0.88	0.88	0.88	0.77	0.77	0.77
5120	0.90	0.90	0.80	0.80	0.80	0.80	0.80	0.80	0.80
5632	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81
6144	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83
6656	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84
7168	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85
7680	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86
8192	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87
10240	0.90	0.90	0.90	0.85	0.85	0.85	0.85	0.85	0.85
12288	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87
14336	0.89	0.89	0.89	0.89	0.89	0.85	0.85	0.85	0.85
16384	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87
18432	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.86	0.86
20480	0.90	0.90	0.90	0.87	0.87	0.87	0.87	0.87	0.87
22528	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.86
24576	0.89	0.89	0.89	0.89	0.87	0.87	0.87	0.87	0.87
26624	0.90	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88
28672	0.89	0.89	0.89	0.89	0.89	0.87	0.87	0.87	0.87
30720	0.90	0.90	0.90	0.88	0.88	0.88	0.88	0.88	0.88
32768	0.89	0.89	0.89	0.89	0.89	0.89	0.87	0.87	0.87

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	579	583	584	590	591	592	594	595	596
512	——	——	——	——	——	——	——	——	——
1024	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
1536	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
2048	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
2560	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80
3072	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83
3584	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.71
4096	0.87	0.87	0.75	0.75	0.75	0.75	0.75	0.75	0.75
4608	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77
5120	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80
5632	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81
6144	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83
6656	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84
7168	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85
7680	0.86	0.86	0.86	0.86	0.80	0.80	0.80	0.80	0.80
8192	0.87	0.87	0.87	0.81	0.81	0.81	0.81	0.81	0.81
10240	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85
12288	0.87	0.87	0.87	0.83	0.83	0.83	0.83	0.83	0.83
14336	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85
16384	0.87	0.87	0.87	0.84	0.84	0.84	0.84	0.84	0.84
18432	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.83	0.83
20480	0.87	0.87	0.87	0.85	0.85	0.85	0.85	0.85	0.85
22528	0.86	0.86	0.86	0.86	0.86	0.86	0.84	0.84	0.84
24576	0.87	0.87	0.87	0.85	0.85	0.85	0.85	0.85	0.85
26624	0.86	0.86	0.86	0.86	0.86	0.84	0.84	0.84	0.84
28672	0.87	0.87	0.87	0.85	0.85	0.85	0.85	0.85	0.85
30720	0.88	0.86	0.86	0.86	0.85	0.85	0.85	0.85	0.85
32768	0.87	0.87	0.87	0.85	0.85	0.85	0.85	0.85	0.84

CI Size	Logical Block Size								
	597	599	601	602	604	606	608	609	612
512	——	——	——	——	——	——	——	——	——
1024	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
1536	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
2048	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
2560	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80
3072	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83
3584	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71
4096	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
4608	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77
5120	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80
5632	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81
6144	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83
6656	0.84	0.84	0.84	0.84	0.84	0.76	0.76	0.76	0.76
7168	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78
7680	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80
8192	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81
10240	0.85	0.85	0.85	0.80	0.80	0.80	0.80	0.80	0.80
12288	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83
14336	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82
16384	0.84	0.84	0.84	0.84	0.84	0.84	0.81	0.81	0.81
18432	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83
20480	0.85	0.85	0.85	0.85	0.82	0.82	0.82	0.82	0.82
22528	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.81	0.81
24576	0.85	0.85	0.83	0.83	0.83	0.83	0.83	0.83	0.83
26624	0.84	0.84	0.84	0.84	0.84	0.82	0.82	0.82	0.82
28672	0.85	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.82
30720	0.85	0.85	0.85	0.85	0.83	0.83	0.83	0.83	0.83
32768	0.84	0.84	0.84	0.84	0.84	0.84	0.82	0.82	0.82

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	613	614	618	620	622	623	624	625	626
512	———	———	———	———	———	———	———	———	———
1024	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
1536	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
2048	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
2560	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80
3072	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
3584	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71
4096	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
4608	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77
5120	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80
5632	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.72	0.72
6144	0.83	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
6656	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76
7168	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78
7680	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80
8192	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81
10240	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80
12288	0.83	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79
14336	0.82	0.82	0.82	0.82	0.82	0.78	0.78	0.78	0.78
16384	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81
18432	0.83	0.83	0.80	0.80	0.80	0.80	0.80	0.80	0.80
20480	0.82	0.82	0.82	0.82	0.80	0.80	0.80	0.80	0.80
22528	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.79
24576	0.83	0.83	0.81	0.81	0.81	0.81	0.81	0.81	0.81
26624	0.82	0.82	0.82	0.80	0.80	0.80	0.80	0.80	0.80
28672	0.82	0.82	0.82	0.82	0.82	0.82	0.80	0.80	0.80
30720	0.83	0.83	0.81	0.81	0.81	0.81	0.81	0.81	0.81
32768	0.82	0.82	0.82	0.81	0.81	0.81	0.81	0.81	0.81

CI Size	Logical Block Size								
	629	633	635	636	637	638	639	642	643
512	———	———	———	———	———	———	———	———	———
1024	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
1536	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
2048	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
2560	0.80	0.80	0.80	0.80	0.80	0.60	0.60	0.60	0.60
3072	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
3584	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71
4096	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
4608	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77
5120	0.80	0.80	0.80	0.80	0.80	0.80	0.70	0.70	0.70
5632	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72
6144	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
6656	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76
7168	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78
7680	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.73	0.73
8192	0.81	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
10240	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.75	0.75
12288	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79
14336	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78
16384	0.81	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78
18432	0.80	0.80	0.80	0.77	0.77	0.77	0.77	0.77	0.77
20480	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.77	0.77
22528	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79
24576	0.81	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79
26624	0.80	0.80	0.78	0.78	0.78	0.78	0.78	0.78	0.78
28672	0.80	0.80	0.80	0.80	0.78	0.78	0.78	0.78	0.78
30720	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.78	0.78
32768	0.81	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.78

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	646	649	650	651	653	654	655	656	657
512	———	———	———	———	———	———	———	———	———
1024	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
1536	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
2048	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
2560	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
3072	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
3584	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71
4096	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
4608	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.66
5120	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70
5632	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72
6144	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
6656	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76
7168	0.78	0.78	0.78	0.71	0.71	0.71	0.71	0.71	0.71
7680	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73
8192	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
10240	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
12288	0.79	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
14336	0.78	0.78	0.78	0.78	0.75	0.75	0.75	0.75	0.75
16384	0.78	0.78	0.78	0.78	0.78	0.78	0.75	0.75	0.75
18432	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77
20480	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77
22528	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77
24576	0.79	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77
26624	0.78	0.78	0.76	0.76	0.76	0.76	0.76	0.76	0.76
28672	0.78	0.78	0.78	0.78	0.76	0.76	0.76	0.76	0.76
30720	0.78	0.78	0.78	0.78	0.78	0.76	0.76	0.76	0.76
32768	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.76	0.76

CI Size	Logical Block Size								
	660	662	663	664	665	666	667	668	679
512	———	———	———	———	———	———	———	———	———
1024	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
1536	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
2048	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
2560	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
3072	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
3584	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71
4096	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
4608	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
5120	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70
5632	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72
6144	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
6656	0.76	0.76	0.76	0.76	0.69	0.69	0.69	0.69	0.69
7168	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71
7680	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73
8192	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
10240	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
12288	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
14336	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
16384	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
18432	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
20480	0.77	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
22528	0.77	0.77	0.75	0.75	0.75	0.75	0.75	0.75	0.75
24576	0.77	0.77	0.77	0.75	0.75	0.75	0.75	0.75	0.75
26624	0.76	0.76	0.76	0.76	0.76	0.75	0.75	0.75	0.75
28672	0.76	0.76	0.76	0.76	0.76	0.76	0.75	0.75	0.75
30720	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.75	0.75
32768	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.75

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	681	682	696	697	699	700	701	702	703
512	———	———	———	———	———	———	———	———	———
1024	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
1536	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
2048	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
2560	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
3072	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
3584	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71
4096	0.75	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62
4608	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
5120	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70
5632	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.63
6144	0.75	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
6656	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69
7168	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71
7680	0.73	0.73	0.73	0.73	0.66	0.66	0.66	0.66	0.66
8192	0.75	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68
10240	0.75	0.75	0.70	0.70	0.70	0.70	0.70	0.70	0.70
12288	0.75	0.75	0.70	0.70	0.70	0.70	0.70	0.70	0.70
14336	0.75	0.75	0.71	0.71	0.71	0.71	0.71	0.71	0.71
16384	0.75	0.75	0.71	0.71	0.71	0.71	0.71	0.71	0.71
18432	0.75	0.75	0.72	0.72	0.72	0.72	0.72	0.72	0.72
20480	0.75	0.75	0.72	0.72	0.72	0.72	0.72	0.72	0.72
22528	0.75	0.75	0.72	0.72	0.72	0.72	0.72	0.72	0.72
24576	0.75	0.75	0.72	0.72	0.72	0.72	0.72	0.70	0.70
26624	0.75	0.75	0.73	0.73	0.73	0.73	0.71	0.71	0.71
28672	0.75	0.75	0.73	0.73	0.73	0.71	0.71	0.71	0.71
30720	0.75	0.75	0.73	0.73	0.71	0.71	0.71	0.71	0.71
32768	0.75	0.75	0.73	0.71	0.71	0.71	0.71	0.71	0.71

CI Size	Logical Block Size								
	705	708	711	712	714	715	716	719	722
512	———	———	———	———	———	———	———	———	———
1024	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
1536	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
2048	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
2560	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
3072	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
3584	0.71	0.71	0.71	0.71	0.71	0.57	0.57	0.57	0.57
4096	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62
4608	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
5120	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70
5632	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.63
6144	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
6656	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69
7168	0.71	0.71	0.71	0.71	0.71	0.71	0.64	0.64	0.64
7680	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
8192	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68
10240	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70
12288	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70
14336	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.67	0.67
16384	0.71	0.71	0.71	0.68	0.68	0.68	0.68	0.68	0.68
18432	0.72	0.72	0.69	0.69	0.69	0.69	0.69	0.69	0.69
20480	0.72	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70
22528	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70
24576	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70
26624	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.69
28672	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.69	0.69
30720	0.71	0.71	0.71	0.71	0.71	0.70	0.70	0.70	0.70
32768	0.71	0.71	0.71	0.71	0.70	0.70	0.70	0.70	0.70

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	726	727	730	731	734	736	738	739	743
512	———	———	———	———	———	———	———	———	———
1024	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
1536	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
2048	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
2560	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
3072	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
3584	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57
4096	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62
4608	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
5120	0.70	0.70	0.70	0.60	0.60	0.60	0.60	0.60	0.60
5632	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.63
6144	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
6656	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.61	0.61
7168	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64
7680	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
8192	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68
10240	0.70	0.70	0.70	0.65	0.65	0.65	0.65	0.65	0.65
12288	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
14336	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67
16384	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68
18432	0.69	0.69	0.69	0.69	0.69	0.69	0.66	0.66	0.66
20480	0.70	0.70	0.70	0.70	0.67	0.67	0.67	0.67	0.67
22528	0.70	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68
24576	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68
26624	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.67
28672	0.69	0.69	0.69	0.69	0.69	0.67	0.67	0.67	0.67
30720	0.70	0.70	0.70	0.70	0.68	0.68	0.68	0.68	0.68
32768	0.70	0.70	0.68	0.68	0.68	0.68	0.68	0.68	0.68

CI Size	Logical Block Size								
	744	749	750	754	758	760	761	763	765
512	———	———	———	———	———	———	———	———	———
1024	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
1536	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.33
2048	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
2560	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
3072	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
3584	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57
4096	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62
4608	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
5120	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
5632	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.63
6144	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
6656	0.61	0.61	0.61	0.61	0.61	0.61	0.61	0.61	0.61
7168	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64
7680	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
8192	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62
10240	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65
12288	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
14336	0.67	0.67	0.67	0.67	0.64	0.64	0.64	0.64	0.64
16384	0.68	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65
18432	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
20480	0.67	0.67	0.67	0.67	0.67	0.65	0.65	0.65	0.65
22528	0.68	0.68	0.68	0.65	0.65	0.65	0.65	0.65	0.65
24576	0.68	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
26624	0.67	0.67	0.67	0.67	0.67	0.67	0.65	0.65	0.65
28672	0.67	0.67	0.67	0.67	0.66	0.66	0.66	0.66	0.66
30720	0.68	0.68	0.66	0.66	0.66	0.66	0.66	0.66	0.66
32768	0.68	0.67	0.67	0.67	0.67	0.67	0.67	0.65	0.65

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	766	767	774	776	779	782	786	787	792
512	----	----	----	----	----	----	----	----	----
1024	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
1536	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
2048	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
2560	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
3072	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
3584	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57
4096	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62
4608	0.66	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.55
5120	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
5632	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.63
6144	0.66	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58
6656	0.61	0.61	0.61	0.61	0.61	0.61	0.61	0.61	0.61
7168	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64
7680	0.66	0.66	0.60	0.60	0.60	0.60	0.60	0.60	0.60
8192	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62
10240	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.60	0.60
12288	0.66	0.66	0.62	0.62	0.62	0.62	0.62	0.62	0.62
14336	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64
16384	0.65	0.65	0.65	0.65	0.65	0.62	0.62	0.62	0.62
18432	0.66	0.66	0.63	0.63	0.63	0.63	0.63	0.63	0.63
20480	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.62
22528	0.65	0.65	0.65	0.65	0.63	0.63	0.63	0.63	0.63
24576	0.66	0.66	0.64	0.64	0.64	0.64	0.64	0.64	0.64
26624	0.65	0.65	0.65	0.65	0.65	0.65	0.63	0.63	0.63
28672	0.66	0.66	0.66	0.64	0.64	0.64	0.64	0.64	0.64
30720	0.66	0.66	0.65	0.65	0.65	0.65	0.65	0.65	0.63
32768	0.65	0.65	0.65	0.65	0.65	0.64	0.64	0.64	0.64

CI Size	Logical Block Size								
	795	796	798	800	803	804	806	808	817
512	----	----	----	----	----	----	----	----	----
1024	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
1536	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
2048	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
2560	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
3072	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
3584	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57
4096	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62
4608	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.55
5120	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
5632	0.63	0.63	0.63	0.63	0.63	0.54	0.54	0.54	0.54
6144	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58
6656	0.61	0.61	0.61	0.61	0.61	0.61	0.61	0.61	0.61
7168	0.64	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57
7680	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
8192	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62
10240	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
12288	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62
14336	0.64	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
16384	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62
18432	0.63	0.63	0.63	0.63	0.61	0.61	0.61	0.61	0.61
20480	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62
22528	0.63	0.63	0.63	0.63	0.63	0.63	0.61	0.61	0.61
24576	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62
26624	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.61	0.61
28672	0.64	0.64	0.62	0.62	0.62	0.62	0.62	0.62	0.62
30720	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.61
32768	0.64	0.64	0.64	0.62	0.62	0.62	0.62	0.62	0.62

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	818	830	831	834	837	839	842	843	847
512	———	———	———	———	———	———	———	———	———
1024	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
1536	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
2048	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
2560	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
3072	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
3584	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57
4096	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
4608	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.55
5120	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
5632	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54
6144	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58
6656	0.61	0.61	0.53	0.53	0.53	0.53	0.53	0.53	0.53
7168	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57
7680	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
8192	0.62	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56
10240	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
12288	0.62	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58
14336	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.57	0.57
16384	0.62	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59
18432	0.61	0.61	0.61	0.61	0.61	0.58	0.58	0.58	0.58
20480	0.62	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
22528	0.61	0.61	0.61	0.61	0.59	0.59	0.59	0.59	0.59
24576	0.62	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
26624	0.61	0.61	0.61	0.59	0.59	0.59	0.59	0.59	0.59
28672	0.62	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.58
30720	0.61	0.61	0.60	0.60	0.60	0.60	0.60	0.60	0.60
32768	0.62	0.60	0.60	0.60	0.60	0.60	0.59	0.59	0.59

CI Size	Logical Block Size								
	850	851	852	853	858	861	862	866	868
512	———	———	———	———	———	———	———	———	———
1024	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
1536	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
2048	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
2560	0.60	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40
3072	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
3584	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57
4096	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
4608	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.55
5120	0.60	0.60	0.50	0.50	0.50	0.50	0.50	0.50	0.50
5632	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54
6144	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58
6656	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53
7168	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57
7680	0.60	0.60	0.60	0.53	0.53	0.53	0.53	0.53	0.53
8192	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56
10240	0.60	0.60	0.60	0.55	0.55	0.55	0.55	0.55	0.55
12288	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58
14336	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57
16384	0.59	0.59	0.59	0.59	0.59	0.59	0.56	0.56	0.56
18432	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58
20480	0.60	0.60	0.60	0.57	0.57	0.57	0.57	0.57	0.57
22528	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.56
24576	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58
26624	0.59	0.59	0.59	0.59	0.59	0.57	0.57	0.57	0.57
28672	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58
30720	0.60	0.60	0.60	0.60	0.58	0.58	0.58	0.58	0.58
32768	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.57	0.57

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	876	877	885	887	890	893	894	895	900
512	----	----	----	----	----	----	----	----	----
1024	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
1536	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
2048	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
2560	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40
3072	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
3584	0.57	0.57	0.57	0.57	0.57	0.57	0.42	0.42	0.42
4096	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
4608	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.55
5120	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
5632	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54
6144	0.58	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
6656	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53
7168	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.50	0.50
7680	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53
8192	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56
10240	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.55
12288	0.58	0.58	0.54	0.54	0.54	0.54	0.54	0.54	0.54
14336	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.53
16384	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56
18432	0.58	0.58	0.55	0.55	0.55	0.55	0.55	0.55	0.55
20480	0.57	0.57	0.57	0.57	0.57	0.55	0.55	0.55	0.55
22528	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56
24576	0.58	0.58	0.56	0.56	0.56	0.56	0.56	0.56	0.56
26624	0.57	0.57	0.57	0.57	0.55	0.55	0.55	0.55	0.55
28672	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.55
30720	0.58	0.58	0.56	0.56	0.56	0.56	0.56	0.56	0.56
32768	0.57	0.57	0.57	0.56	0.56	0.56	0.56	0.56	0.56

CI Size	Logical Block Size								
	903	909	917	919	921	924	930	935	937
512	----	----	----	----	----	----	----	----	----
1024	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
1536	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
2048	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
2560	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40
3072	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
3584	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42
4096	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
4608	0.55	0.55	0.55	0.55	0.44	0.44	0.44	0.44	0.44
5120	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
5632	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54
6144	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
6656	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53
7168	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
7680	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53
8192	0.56	0.56	0.50	0.50	0.50	0.50	0.50	0.50	0.50
10240	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.50	0.50
12288	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54
14336	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53
16384	0.56	0.56	0.53	0.53	0.53	0.53	0.53	0.53	0.53
18432	0.55	0.55	0.55	0.55	0.55	0.52	0.52	0.52	0.52
20480	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.52	0.52
22528	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54
24576	0.56	0.56	0.54	0.54	0.54	0.54	0.54	0.54	0.54
26624	0.55	0.55	0.55	0.53	0.53	0.53	0.53	0.53	0.53
28672	0.55	0.55	0.55	0.55	0.55	0.55	0.53	0.53	0.53
30720	0.56	0.55	0.55	0.55	0.55	0.55	0.55	0.53	0.53
32768	0.56	0.56	0.54	0.54	0.54	0.54	0.54	0.54	0.53

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	938	944	949	950	955	958	959	963	969
512	———	———	———	———	———	———	———	———	———
1024	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
1536	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
2048	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
2560	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40
3072	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
3584	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42
4096	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
4608	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44
5120	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
5632	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
6144	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
6656	0.53	0.53	0.53	0.46	0.46	0.46	0.46	0.46	0.46
7168	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
7680	0.53	0.53	0.53	0.53	0.53	0.53	0.46	0.46	0.46
8192	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
10240	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
12288	0.54	0.54	0.50	0.50	0.50	0.50	0.50	0.50	0.50
14336	0.53	0.53	0.53	0.53	0.53	0.50	0.50	0.50	0.50
16384	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.50
18432	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52
20480	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52
22528	0.54	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52
24576	0.54	0.54	0.52	0.52	0.52	0.52	0.52	0.52	0.52
26624	0.53	0.53	0.53	0.53	0.51	0.51	0.51	0.51	0.51
28672	0.53	0.53	0.53	0.53	0.53	0.51	0.51	0.51	0.51
30720	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.51	0.51
32768	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.51

CI Size	Logical Block Size								
	974	979	982	985	988	990	992	1014	1019
512	———	———	———	———	———	———	———	———	———
1024	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	———
1536	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
2048	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
2560	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40
3072	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
3584	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42
4096	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
4608	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44
5120	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
5632	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
6144	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
6656	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.46
7168	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
7680	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.46
8192	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
10240	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
12288	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
14336	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
16384	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
18432	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
20480	0.52	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
22528	0.52	0.52	0.50	0.50	0.50	0.50	0.50	0.50	0.50
24576	0.52	0.52	0.52	0.50	0.50	0.50	0.50	0.50	0.50
26624	0.51	0.51	0.51	0.51	0.50	0.50	0.50	0.50	0.50
28672	0.51	0.51	0.51	0.51	0.51	0.50	0.50	0.50	0.50
30720	0.51	0.51	0.51	0.51	0.51	0.51	0.50	0.50	0.50
32768	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.50	0.50

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	1020	1021	1022	1023	1056	1058	1061	1064	1068
512	———	———	———	———	———	———	———	———	———
1024	———	———	———	———	———	———	———	———	———
1536	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
2048	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
2560	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40
3072	0.50	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
3584	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42
4096	0.50	0.50	0.37	0.37	0.37	0.37	0.37	0.37	0.37
4608	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44
5120	0.50	0.50	0.50	0.40	0.40	0.40	0.40	0.40	0.40
5632	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
6144	0.50	0.50	0.50	0.41	0.41	0.41	0.41	0.41	0.41
6656	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.46
7168	0.50	0.50	0.50	0.42	0.42	0.42	0.42	0.42	0.42
7680	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.46
8192	0.50	0.50	0.50	0.43	0.43	0.43	0.43	0.43	0.43
10240	0.50	0.50	0.50	0.50	0.45	0.45	0.45	0.45	0.45
12288	0.50	0.50	0.50	0.50	0.45	0.45	0.45	0.45	0.45
14336	0.50	0.50	0.50	0.50	0.46	0.46	0.46	0.46	0.46
16384	0.50	0.50	0.50	0.50	0.46	0.46	0.46	0.46	0.46
18432	0.50	0.50	0.50	0.50	0.47	0.47	0.47	0.47	0.47
20480	0.50	0.50	0.50	0.50	0.47	0.47	0.47	0.47	0.47
22528	0.50	0.50	0.50	0.50	0.47	0.47	0.47	0.47	0.47
24576	0.50	0.50	0.50	0.50	0.47	0.47	0.47	0.47	0.47
26624	0.50	0.50	0.50	0.50	0.48	0.48	0.48	0.48	0.46
28672	0.50	0.50	0.50	0.50	0.48	0.48	0.48	0.46	0.46
30720	0.50	0.50	0.50	0.50	0.48	0.48	0.46	0.46	0.46
32768	0.50	0.50	0.50	0.50	0.48	0.46	0.46	0.46	0.46

CI Size	Logical Block Size								
	1072	1077	1083	1091	1095	1096	1102	1107	1108
512	———	———	———	———	———	———	———	———	———
1024	———	———	———	———	———	———	———	———	———
1536	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
2048	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
2560	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40
3072	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
3584	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42
4096	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37
4608	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44
5120	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40
5632	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
6144	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41
6656	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.38
7168	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42
7680	0.46	0.46	0.46	0.46	0.46	0.40	0.40	0.40	0.40
8192	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43
10240	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
12288	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
14336	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.42	0.42
16384	0.46	0.46	0.46	0.46	0.43	0.43	0.43	0.43	0.43
18432	0.47	0.47	0.47	0.44	0.44	0.44	0.44	0.44	0.44
20480	0.47	0.47	0.45	0.45	0.45	0.45	0.45	0.45	0.45
22528	0.47	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
24576	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
26624	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.46
28672	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.44	0.44
30720	0.46	0.46	0.46	0.46	0.46	0.46	0.45	0.45	0.45
32768	0.46	0.46	0.46	0.46	0.45	0.45	0.45	0.45	0.45

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	1116	1124	1125	1129	1136	1137	1146	1149	1151
512	———	———	———	———	———	———	———	———	———
1024	———	———	———	———	———	———	———	———	———
1536	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
2048	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
2560	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40
3072	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
3584	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42
4096	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37
4608	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.33
5120	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40
5632	0.45	0.45	0.36	0.36	0.36	0.36	0.36	0.36	0.36
6144	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41
6656	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38
7168	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42
7680	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40
8192	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43
10240	0.45	0.45	0.45	0.45	0.45	0.40	0.40	0.40	0.40
12288	0.45	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41
14336	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42
16384	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43
18432	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44
20480	0.45	0.45	0.45	0.45	0.45	0.45	0.42	0.42	0.42
22528	0.45	0.45	0.45	0.43	0.43	0.43	0.43	0.43	0.43
24576	0.45	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43
26624	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44
28672	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.42	0.42
30720	0.45	0.45	0.45	0.45	0.45	0.45	0.43	0.43	0.43
32768	0.45	0.45	0.45	0.45	0.43	0.43	0.43	0.43	0.43

CI Size	Logical Block Size								
	1157	1168	1169	1181	1185	1191	1193	1194	1204
512	———	———	———	———	———	———	———	———	———
1024	———	———	———	———	———	———	———	———	———
1536	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
2048	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
2560	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40
3072	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
3584	0.42	0.42	0.42	0.42	0.42	0.42	0.28	0.28	0.28
4096	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37
4608	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
5120	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40
5632	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36
6144	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41
6656	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38
7168	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.35	0.35
7680	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40
8192	0.43	0.43	0.37	0.37	0.37	0.37	0.37	0.37	0.37
10240	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40
12288	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41
14336	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.39	0.39
16384	0.43	0.43	0.43	0.40	0.40	0.40	0.40	0.40	0.40
18432	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41
20480	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42
22528	0.43	0.43	0.43	0.43	0.43	0.40	0.40	0.40	0.40
24576	0.43	0.43	0.43	0.41	0.41	0.41	0.41	0.41	0.41
26624	0.44	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42
28672	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.41
30720	0.43	0.43	0.43	0.43	0.41	0.41	0.41	0.41	0.41
32768	0.43	0.43	0.43	0.42	0.42	0.42	0.42	0.42	0.42

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	1209	1213	1226	1227	1228	1246	1251	1259	1267
512	----	----	----	----	----	----	----	----	----
1024	----	----	----	----	----	----	----	----	----
1536	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
2048	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
2560	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40
3072	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
3584	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
4096	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37
4608	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
5120	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40
5632	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36
6144	0.41	0.41	0.41	0.33	0.33	0.33	0.33	0.33	0.33
6656	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38
7168	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35
7680	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40
8192	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37
10240	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40
12288	0.41	0.41	0.41	0.41	0.37	0.37	0.37	0.37	0.37
14336	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39
16384	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.37
18432	0.41	0.41	0.41	0.41	0.41	0.38	0.38	0.38	0.38
20480	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40
22528	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.38	0.38
24576	0.41	0.41	0.41	0.41	0.41	0.39	0.39	0.39	0.39
26624	0.42	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40
28672	0.41	0.41	0.41	0.41	0.41	0.41	0.39	0.39	0.39
30720	0.41	0.41	0.41	0.41	0.41	0.40	0.40	0.40	0.40
32768	0.42	0.42	0.40	0.40	0.40	0.40	0.40	0.40	0.39

CI Size	Logical Block Size								
	1275	1277	1278	1279	1292	1302	1310	1315	1324
512	----	----	----	----	----	----	----	----	----
1024	----	----	----	----	----	----	----	----	----
1536	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
2048	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
2560	0.40	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
3072	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
3584	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
4096	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37
4608	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
5120	0.40	0.40	0.30	0.30	0.30	0.30	0.30	0.30	0.30
5632	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36
6144	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
6656	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38
7168	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35
7680	0.40	0.40	0.40	0.33	0.33	0.33	0.33	0.33	0.33
8192	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37
10240	0.40	0.40	0.40	0.35	0.35	0.35	0.35	0.35	0.35
12288	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37
14336	0.39	0.39	0.39	0.39	0.39	0.39	0.35	0.35	0.35
16384	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37
18432	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.36
20480	0.40	0.40	0.40	0.40	0.37	0.37	0.37	0.37	0.37
22528	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38
24576	0.39	0.39	0.39	0.39	0.39	0.37	0.37	0.37	0.37
26624	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38
28672	0.39	0.39	0.39	0.39	0.39	0.39	0.37	0.37	0.37
30720	0.40	0.40	0.40	0.40	0.38	0.38	0.38	0.38	0.38
32768	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.37	0.37

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	1329	1330	1335	1362	1363	1364	1395	1400	1405
512	———	———	———	———	———	———	———	———	———
1024	———	———	———	———	———	———	———	———	———
1536	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
2048	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
2560	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
3072	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
3584	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
4096	0.37	0.37	0.37	0.37	0.25	0.25	0.25	0.25	0.25
4608	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
5120	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30
5632	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36
6144	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
6656	0.38	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30
7168	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35
7680	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
8192	0.37	0.37	0.37	0.37	0.37	0.31	0.31	0.31	0.31
10240	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35
12288	0.37	0.37	0.37	0.37	0.37	0.37	0.33	0.33	0.33
14336	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35
16384	0.37	0.37	0.37	0.37	0.37	0.37	0.34	0.34	0.34
18432	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36
20480	0.37	0.37	0.37	0.37	0.37	0.37	0.35	0.35	0.35
22528	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36
24576	0.37	0.37	0.37	0.37	0.37	0.37	0.35	0.35	0.35
26624	0.38	0.38	0.36	0.36	0.36	0.36	0.36	0.36	0.34
28672	0.37	0.37	0.37	0.37	0.37	0.37	0.35	0.35	0.35
30720	0.38	0.38	0.38	0.36	0.36	0.36	0.36	0.35	0.35
32768	0.37	0.37	0.37	0.37	0.37	0.37	0.35	0.35	0.35

CI Size	Logical Block Size								
	1407	1417	1424	1431	1432	1433	1445	1461	1462
512	———	———	———	———	———	———	———	———	———
1024	———	———	———	———	———	———	———	———	———
1536	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
2048	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
2560	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
3072	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
3584	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
4096	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
4608	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
5120	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30
5632	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27
6144	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
6656	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30
7168	0.35	0.35	0.35	0.35	0.28	0.28	0.28	0.28	0.28
7680	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
8192	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31
10240	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.30
12288	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
14336	0.35	0.35	0.35	0.35	0.35	0.32	0.32	0.32	0.32
16384	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
18432	0.36	0.36	0.33	0.33	0.33	0.33	0.33	0.33	0.33
20480	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35
22528	0.36	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
24576	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.33	0.33
26624	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
28672	0.35	0.35	0.35	0.35	0.35	0.35	0.33	0.33	0.33
30720	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35
32768	0.35	0.35	0.35	0.34	0.34	0.34	0.34	0.34	0.34

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	1478	1488	1489	1501	1508	1526	1531	1532	1533
512	——	——	——	——	——	——	——	——	——
1024	——	——	——	——	——	——	——	——	——
1536	0.33	0.33	0.33	0.33	0.33	0.33	——	——	——
2048	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
2560	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
3072	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.16	0.16
3584	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
4096	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
4608	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.22
5120	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30
5632	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27
6144	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
6656	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30
7168	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
7680	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
8192	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31
10240	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30
12288	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
14336	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32
16384	0.34	0.34	0.31	0.31	0.31	0.31	0.31	0.31	0.31
18432	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
20480	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32
22528	0.34	0.34	0.34	0.34	0.31	0.31	0.31	0.31	0.31
24576	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
26624	0.34	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32
28672	0.33	0.33	0.33	0.33	0.33	0.32	0.32	0.32	0.32
30720	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
32768	0.34	0.34	0.34	0.32	0.32	0.32	0.32	0.32	0.32

CI Size	Logical Block Size								
	1534	1535	1559	1565	1574	1591	1592	1608	1616
512	——	——	——	——	——	——	——	——	——
1024	——	——	——	——	——	——	——	——	——
1536	——	——	——	——	——	——	——	——	——
2048	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
2560	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
3072	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16
3584	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
4096	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
4608	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22
5120	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30
5632	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27
6144	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
6656	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30
7168	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
7680	0.33	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26
8192	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31
10240	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30
12288	0.33	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29
14336	0.32	0.32	0.32	0.32	0.32	0.32	0.28	0.28	0.28
16384	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31
18432	0.33	0.33	0.30	0.30	0.30	0.30	0.30	0.30	0.30
20480	0.32	0.32	0.32	0.32	0.32	0.30	0.30	0.30	0.30
22528	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.29
24576	0.33	0.33	0.31	0.31	0.31	0.31	0.31	0.31	0.31
26624	0.32	0.32	0.32	0.32	0.30	0.30	0.30	0.30	0.30
28672	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.30	0.30
30720	0.33	0.33	0.31	0.31	0.31	0.31	0.31	0.31	0.31
32768	0.32	0.32	0.32	0.31	0.31	0.31	0.31	0.31	0.31

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	1636	1637	1661	1663	1674	1686	1703	1705	1706
512	———	———	———	———	———	———	———	———	———
1024	———	———	———	———	———	———	———	———	———
1536	———	———	———	———	———	———	———	———	———
2048	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
2560	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
3072	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16
3584	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
4096	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
4608	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22
5120	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.20	0.20
5632	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27
6144	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
6656	0.30	0.30	0.30	0.23	0.23	0.23	0.23	0.23	0.23
7168	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
7680	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26
8192	0.31	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
10240	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.25
12288	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29
14336	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
16384	0.31	0.31	0.28	0.28	0.28	0.28	0.28	0.28	0.28
18432	0.30	0.30	0.30	0.30	0.30	0.27	0.27	0.27	0.27
20480	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.27
22528	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29
24576	0.31	0.31	0.29	0.29	0.29	0.29	0.29	0.29	0.29
26624	0.30	0.30	0.30	0.30	0.28	0.28	0.28	0.28	0.28
28672	0.30	0.30	0.30	0.30	0.30	0.30	0.28	0.28	0.28
30720	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30
32768	0.31	0.31	0.29	0.29	0.29	0.29	0.29	0.29	0.29

CI Size	Logical Block Size								
	1724	1732	1754	1774	1787	1789	1790	1791	1806
512	———	———	———	———	———	———	———	———	———
1024	———	———	———	———	———	———	———	———	———
1536	———	———	———	———	———	———	———	———	———
2048	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
2560	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
3072	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16
3584	0.28	0.28	0.28	0.28	0.28	0.14	0.14	0.14	0.14
4096	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
4608	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22
5120	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
5632	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27
6144	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
6656	0.23	0.23	0.23	0.23	0.23	0.23	0.23	0.23	0.23
7168	0.28	0.28	0.28	0.28	0.28	0.28	0.21	0.21	0.21
7680	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26
8192	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
10240	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
12288	0.29	0.29	0.29	0.25	0.25	0.25	0.25	0.25	0.25
14336	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.25	0.25
16384	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
18432	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27
20480	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27
22528	0.29	0.29	0.27	0.27	0.27	0.27	0.27	0.27	0.27
24576	0.29	0.29	0.29	0.27	0.27	0.27	0.27	0.27	0.27
26624	0.28	0.28	0.28	0.28	0.26	0.26	0.26	0.26	0.26
28672	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.26
30720	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
32768	0.29	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	1819	1842	1860	1874	1876	1889	1901	1910	1917
512	——	——	——	——	——	——	——	——	——
1024	——	——	——	——	——	——	——	——	——
1536	——	——	——	——	——	——	——	——	——
2048	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
2560	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
3072	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16
3584	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
4096	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
4608	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22
5120	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
5632	0.27	0.27	0.27	0.27	0.18	0.18	0.18	0.18	0.18
6144	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
6656	0.23	0.23	0.23	0.23	0.23	0.23	0.23	0.23	0.23
7168	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
7680	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26
8192	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
10240	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
12288	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
14336	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
16384	0.28	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
18432	0.27	0.27	0.25	0.25	0.25	0.25	0.25	0.25	0.25
20480	0.27	0.27	0.27	0.25	0.25	0.25	0.25	0.25	0.25
22528	0.27	0.27	0.27	0.27	0.27	0.25	0.25	0.25	0.25
24576	0.27	0.27	0.27	0.27	0.27	0.27	0.25	0.25	0.25
26624	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.25	0.25
28672	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.25
30720	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26
32768	0.28	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26

CI Size	Logical Block Size								
	1919	1926	2038	2043	2044	2045	2046	2047	2183
512	——	——	——	——	——	——	——	——	——
1024	——	——	——	——	——	——	——	——	——
1536	——	——	——	——	——	——	——	——	——
2048	0.25	0.25	0.25	——	——	——	——	——	——
2560	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
3072	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16
3584	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
4096	0.25	0.25	0.25	0.25	0.12	0.12	0.12	0.12	0.12
4608	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22
5120	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
5632	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18
6144	0.25	0.25	0.25	0.25	0.25	0.16	0.16	0.16	0.16
6656	0.23	0.23	0.23	0.23	0.23	0.23	0.23	0.23	0.23
7168	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
7680	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
8192	0.25	0.25	0.25	0.25	0.25	0.25	0.18	0.18	0.18
10240	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.20	0.20
12288	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.20	0.20
14336	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.21	0.21
16384	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.21	0.21
18432	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.22	0.22
20480	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.22
22528	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.22
24576	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.22
26624	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.23
28672	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.23
30720	0.26	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.23
32768	0.26	0.26	0.25	0.25	0.25	0.25	0.25	0.25	0.23

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	2193	2204	2215	2217	2233	2251	2274	2299	2302
512	----	----	----	----	----	----	----	----	----
1024	----	----	----	----	----	----	----	----	----
1536	----	----	----	----	----	----	----	----	----
2048	----	----	----	----	----	----	----	----	----
2560	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
3072	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16
3584	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
4096	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12
4608	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.11
5120	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
5632	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18
6144	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16
6656	0.23	0.23	0.23	0.15	0.15	0.15	0.15	0.15	0.15
7168	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
7680	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
8192	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18
10240	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
12288	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
14336	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
16384	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
18432	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22
20480	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.20	0.20
22528	0.22	0.22	0.22	0.22	0.22	0.22	0.20	0.20	0.20
24576	0.22	0.22	0.22	0.22	0.22	0.20	0.20	0.20	0.20
26624	0.23	0.23	0.23	0.23	0.21	0.21	0.21	0.21	0.21
28672	0.23	0.23	0.21	0.21	0.21	0.21	0.21	0.21	0.21
30720	0.23	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
32768	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21

CI Size	Logical Block Size								
	2339	2362	2386	2387	2388	2419	2455	2456	2502
512	----	----	----	----	----	----	----	----	----
1024	----	----	----	----	----	----	----	----	----
1536	----	----	----	----	----	----	----	----	----
2048	----	----	----	----	----	----	----	----	----
2560	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
3072	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16
3584	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
4096	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12
4608	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11
5120	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
5632	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18
6144	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16
6656	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15
7168	0.21	0.21	0.21	0.14	0.14	0.14	0.14	0.14	0.14
7680	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
8192	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18
10240	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
12288	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.16	0.16
14336	0.21	0.21	0.21	0.21	0.17	0.17	0.17	0.17	0.17
16384	0.21	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18
18432	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19
20480	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
22528	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
24576	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.18
26624	0.21	0.21	0.21	0.21	0.21	0.21	0.19	0.19	0.19
28672	0.21	0.21	0.21	0.21	0.21	0.19	0.19	0.19	0.19
30720	0.21	0.21	0.20	0.20	0.20	0.20	0.20	0.20	0.20
32768	0.21	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	2519	2550	2555	2556	2557	2558	2559	2605	2631
512	——	——	——	——	——	——	——	——	——
1024	——	——	——	——	——	——	——	——	——
1536	——	——	——	——	——	——	——	——	——
2048	——	——	——	——	——	——	——	——	——
2560	0.20	0.20	——	——	——	——	——	——	——
3072	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16
3584	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
4096	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12
4608	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11
5120	0.20	0.20	0.20	0.10	0.10	0.10	0.10	0.10	0.10
5632	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18
6144	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16
6656	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15
7168	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
7680	0.20	0.20	0.20	0.20	0.13	0.13	0.13	0.13	0.13
8192	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18
10240	0.20	0.20	0.20	0.20	0.20	0.15	0.15	0.15	0.15
12288	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16
14336	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17
16384	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18
18432	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19
20480	0.20	0.20	0.20	0.20	0.20	0.20	0.17	0.17	0.17
22528	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18
24576	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18
26624	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19
28672	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.17
30720	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.18	0.18
32768	0.20	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18

CI Size	Logical Block Size								
	2661	2727	2729	2791	2811	2814	2865	2866	2924
512	——	——	——	——	——	——	——	——	——
1024	——	——	——	——	——	——	——	——	——
1536	——	——	——	——	——	——	——	——	——
2048	——	——	——	——	——	——	——	——	——
2560	——	——	——	——	——	——	——	——	——
3072	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16
3584	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
4096	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12
4608	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11
5120	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
5632	0.18	0.18	0.18	0.18	0.18	0.09	0.09	0.09	0.09
6144	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16
6656	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15
7168	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
7680	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13
8192	0.18	0.18	0.12	0.12	0.12	0.12	0.12	0.12	0.12
10240	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15
12288	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16
14336	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.14	0.14
16384	0.18	0.18	0.18	0.15	0.15	0.15	0.15	0.15	0.15
18432	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16
20480	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17
22528	0.18	0.18	0.18	0.18	0.18	0.18	0.15	0.15	0.15
24576	0.18	0.18	0.18	0.16	0.16	0.16	0.16	0.16	0.16
26624	0.19	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17
28672	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.16
30720	0.18	0.18	0.18	0.18	0.16	0.16	0.16	0.16	0.16
32768	0.18	0.18	0.18	0.17	0.17	0.17	0.17	0.17	0.17

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	2957	2978	3062	3067	3069	3070	3071	3184	3216
512	----	----	----	----	----	----	----	----	----
1024	----	----	----	----	----	----	----	----	----
1536	----	----	----	----	----	----	----	----	----
2048	----	----	----	----	----	----	----	----	----
2560	----	----	----	----	----	----	----	----	----
3072	.166	.166	.166	----	----	----	----	----	----
3584	.142	.142	.142	.142	.142	.142	.142	.142	.142
4096	.125	.125	.125	.125	.125	.125	.125	.125	.125
4608	.111	.111	.111	.111	.111	.111	.111	.111	.111
5120	.100	.100	.100	.100	.100	.100	.100	.100	.100
5632	.090	.090	.090	.090	.090	.090	.090	.090	.090
6144	.166	.166	.166	.166	.083	.083	.083	.083	.083
6656	.153	.153	.153	.153	.153	.153	.153	.153	.153
7168	.142	.142	.142	.142	.142	.142	.142	.142	.142
7680	.133	.133	.133	.133	.133	.133	.133	.133	.133
8192	.125	.125	.125	.125	.125	.125	.125	.125	.125
10240	.150	.150	.150	.150	.150	.150	.150	.150	.150
12288	.166	.166	.166	.166	.166	.125	.125	.125	.125
14336	.142	.142	.142	.142	.142	.142	.142	.142	.142
16384	.156	.156	.156	.156	.156	.156	.156	.156	.156
18432	.166	.166	.166	.166	.166	.166	.138	.138	.138
20480	.150	.150	.150	.150	.150	.150	.150	.150	.150
22528	.159	.159	.159	.159	.159	.159	.159	.159	.159
24576	.166	.166	.166	.166	.166	.166	.145	.145	.145
26624	.173	.153	.153	.153	.153	.153	.153	.153	.153
28672	.160	.160	.160	.160	.160	.160	.160	.160	.142
30720	.166	.166	.166	.166	.166	.166	.166	.150	.150
32768	.171	.171	.156	.156	.156	.156	.156	.156	.156

CI Size	Logical Block Size								
	3274	3275	3323	3326	3410	3411	3412	3509	3574
512	----	----	----	----	----	----	----	----	----
1024	----	----	----	----	----	----	----	----	----
1536	----	----	----	----	----	----	----	----	----
2048	----	----	----	----	----	----	----	----	----
2560	----	----	----	----	----	----	----	----	----
3072	----	----	----	----	----	----	----	----	----
3584	.142	.142	.142	.142	.142	.142	.142	.142	.142
4096	.125	.125	.125	.125	.125	.125	.125	.125	.125
4608	.111	.111	.111	.111	.111	.111	.111	.111	.111
5120	.100	.100	.100	.100	.100	.100	.100	.100	.100
5632	.090	.090	.090	.090	.090	.090	.090	.090	.090
6144	.083	.083	.083	.083	.083	.083	.083	.083	.083
6656	.153	.153	.153	.076	.076	.076	.076	.076	.076
7168	.142	.142	.142	.142	.142	.142	.142	.142	.142
7680	.133	.133	.133	.133	.133	.133	.133	.133	.133
8192	.125	.125	.125	.125	.125	.125	.125	.125	.125
10240	.150	.150	.150	.150	.150	.100	.100	.100	.100
12288	.125	.125	.125	.125	.125	.125	.125	.125	.125
14336	.142	.142	.142	.142	.142	.142	.142	.142	.142
16384	.156	.125	.125	.125	.125	.125	.125	.125	.125
18432	.138	.138	.138	.138	.138	.138	.138	.138	.138
20480	.150	.150	.150	.150	.150	.150	.125	.125	.125
22528	.136	.136	.136	.136	.136	.136	.136	.136	.136
24576	.145	.145	.145	.145	.145	.145	.145	.145	.125
26624	.153	.153	.153	.153	.134	.134	.134	.134	.134
28672	.142	.142	.142	.142	.142	.142	.142	.142	.142
30720	.150	.150	.150	.150	.150	.150	.150	.133	.133
32768	.156	.156	.140	.140	.140	.140	.140	.140	.140

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	3579	3581	3582	3639	3684	3753	3802	3835	3838
512	———	———	———	———	———	———	———	———	———
1024	———	———	———	———	———	———	———	———	———
1536	———	———	———	———	———	———	———	———	———
2048	———	———	———	———	———	———	———	———	———
2560	———	———	———	———	———	———	———	———	———
3072	———	———	———	———	———	———	———	———	———
3584	———	———	———	———	———	———	———	———	———
4096	.125	.125	.125	.125	.125	.125	.125	.125	.125
4608	.111	.111	.111	.111	.111	.111	.111	.111	.111
5120	.100	.100	.100	.100	.100	.100	.100	.100	.100
5632	.090	.090	.090	.090	.090	.090	.090	.090	.090
6144	.083	.083	.083	.083	.083	.083	.083	.083	.083
6656	.076	.076	.076	.076	.076	.076	.076	.076	.076
7168	.142	.071	.071	.071	.071	.071	.071	.071	.071
7680	.133	.133	.133	.133	.133	.133	.133	.133	.066
8192	.125	.125	.125	.125	.125	.125	.125	.125	.125
10240	.100	.100	.100	.100	.100	.100	.100	.100	.100
12288	.125	.125	.125	.125	.125	.125	.125	.125	.125
14336	.142	.142	.107	.107	.107	.107	.107	.107	.107
16384	.125	.125	.125	.125	.125	.125	.125	.125	.125
18432	.138	.138	.138	.138	.138	.111	.111	.111	.111
20480	.125	.125	.125	.125	.125	.125	.125	.125	.125
22528	.136	.136	.136	.136	.136	.136	.113	.113	.113
24576	.125	.125	.125	.125	.125	.125	.125	.125	.125
26624	.134	.134	.134	.134	.134	.134	.134	.115	.115
28672	.142	.142	.142	.125	.125	.125	.125	.125	.125
30720	.133	.133	.133	.133	.133	.133	.133	.133	.133
32768	.140	.140	.140	.140	.125	.125	.125	.125	.125

CI Size	Logical Block Size								
	4086	4091	4092	4093	4094	4387	4435	4503	4598
512	———	———	———	———	———	———	———	———	———
1024	———	———	———	———	———	———	———	———	———
1536	———	———	———	———	———	———	———	———	———
2048	———	———	———	———	———	———	———	———	———
2560	———	———	———	———	———	———	———	———	———
3072	———	———	———	———	———	———	———	———	———
3584	———	———	———	———	———	———	———	———	———
4096	.125	———	———	———	———	———	———	———	———
4608	.111	.111	.111	.111	.111	.111	.111	.111	.111
5120	.100	.100	.100	.100	.100	.100	.100	.100	.100
5632	.090	.090	.090	.090	.090	.090	.090	.090	.090
6144	.083	.083	.083	.083	.083	.083	.083	.083	.083
6656	.076	.076	.076	.076	.076	.076	.076	.076	.076
7168	.071	.071	.071	.071	.071	.071	.071	.071	.071
7680	.066	.066	.066	.066	.066	.066	.066	.066	.066
8192	.125	.125	.062	.062	.062	.062	.062	.062	.062
10240	.100	.100	.100	.100	.100	.100	.100	.100	.100
12288	.125	.125	.125	.083	.083	.083	.083	.083	.083
14336	.107	.107	.107	.107	.107	.107	.107	.107	.107
16384	.125	.125	.125	.125	.093	.093	.093	.093	.093
18432	.111	.111	.111	.111	.111	.111	.111	.111	.111
20480	.125	.125	.125	.125	.125	.100	.100	.100	.100
22528	.113	.113	.113	.113	.113	.113	.113	.113	.090
24576	.125	.125	.125	.125	.125	.104	.104	.104	.104
26624	.115	.115	.115	.115	.115	.115	.115	.096	.096
28672	.125	.125	.125	.125	.125	.107	.107	.107	.107
30720	.116	.116	.116	.116	.116	.116	.100	.100	.100
32768	.125	.125	.125	.125	.125	.109	.109	.109	.109

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	4605	4679	4775	4777	4913	5110	5115	5117	5118
512	———	———	———	———	———	———	———	———	———
1024	———	———	———	———	———	———	———	———	———
1536	———	———	———	———	———	———	———	———	———
2048	———	———	———	———	———	———	———	———	———
2560	———	———	———	———	———	———	———	———	———
3072	———	———	———	———	———	———	———	———	———
3584	———	———	———	———	———	———	———	———	———
4096	———	———	———	———	———	———	———	———	———
4608	———	———	———	———	———	———	———	———	———
5120	.100	.100	.100	.100	.100	.100	———	———	———
5632	.090	.090	.090	.090	.090	.090	.090	.090	.090
6144	.083	.083	.083	.083	.083	.083	.083	.083	.083
6656	.076	.076	.076	.076	.076	.076	.076	.076	.076
7168	.071	.071	.071	.071	.071	.071	.071	.071	.071
7680	.066	.066	.066	.066	.066	.066	.066	.066	.066
8192	.062	.062	.062	.062	.062	.062	.062	.062	.062
10240	.100	.100	.100	.100	.100	.100	.100	.050	.050
12288	.083	.083	.083	.083	.083	.083	.083	.083	.083
14336	.107	.107	.107	.071	.071	.071	.071	.071	.071
16384	.093	.093	.093	.093	.093	.093	.093	.093	.093
18432	.111	.083	.083	.083	.083	.083	.083	.083	.083
20480	.100	.100	.100	.100	.100	.100	.100	.100	.075
22528	.090	.090	.090	.090	.090	.090	.090	.090	.090
24576	.104	.104	.104	.104	.104	.083	.083	.083	.083
26624	.096	.096	.096	.096	.096	.096	.096	.096	.096
28672	.107	.107	.107	.107	.089	.089	.089	.089	.089
30720	.100	.100	.100	.100	.100	.100	.100	.100	.100
32768	.109	.109	.093	.093	.093	.093	.093	.093	.093

CI Size	Logical Block Size								
	5322	5458	5459	5622	5629	5732	6134	6139	6140
512	———	———	———	———	———	———	———	———	———
1024	———	———	———	———	———	———	———	———	———
1536	———	———	———	———	———	———	———	———	———
2048	———	———	———	———	———	———	———	———	———
2560	———	———	———	———	———	———	———	———	———
3072	———	———	———	———	———	———	———	———	———
3584	———	———	———	———	———	———	———	———	———
4096	———	———	———	———	———	———	———	———	———
4608	———	———	———	———	———	———	———	———	———
5120	———	———	———	———	———	———	———	———	———
5632	.090	.090	.090	.090	———	———	———	———	———
6144	.083	.083	.083	.083	.083	.083	.083	———	———
6656	.076	.076	.076	.076	.076	.076	.076	.076	.076
7168	.071	.071	.071	.071	.071	.071	.071	.071	.071
7680	.066	.066	.066	.066	.066	.066	.066	.066	.066
8192	.062	.062	.062	.062	.062	.062	.062	.062	.062
10240	.050	.050	.050	.050	.050	.050	.050	.050	.050
12288	.083	.083	.083	.083	.083	.083	.083	.083	.041
14336	.071	.071	.071	.071	.071	.071	.071	.071	.071
16384	.093	.093	.062	.062	.062	.062	.062	.062	.062
18432	.083	.083	.083	.083	.083	.083	.083	.083	.083
20480	.075	.075	.075	.075	.075	.075	.075	.075	.075
22528	.090	.090	.090	.090	.090	.068	.068	.068	.068
24576	.083	.083	.083	.083	.083	.083	.083	.083	.083
26624	.096	.076	.076	.076	.076	.076	.076	.076	.076
28672	.089	.089	.089	.089	.089	.089	.071	.071	.071
30720	.083	.083	.083	.083	.083	.083	.083	.083	.083
32768	.093	.093	.093	.078	.078	.078	.078	.078	.078

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	6141	6142	6551	6646	6653	6823	7158	7163	7165
512	----	----	----	----	----	----	----	----	----
1024	----	----	----	----	----	----	----	----	----
1536	----	----	----	----	----	----	----	----	----
2048	----	----	----	----	----	----	----	----	----
2560	----	----	----	----	----	----	----	----	----
3072	----	----	----	----	----	----	----	----	----
3584	----	----	----	----	----	----	----	----	----
4096	----	----	----	----	----	----	----	----	----
4608	----	----	----	----	----	----	----	----	----
5120	----	----	----	----	----	----	----	----	----
5632	----	----	----	----	----	----	----	----	----
6144	----	----	----	----	----	----	----	----	----
6656	.076	.076	.076	.076	----	----	----	----	----
7168	.071	.071	.071	.071	.071	.071	.071	----	----
7680	.066	.066	.066	.066	.066	.066	.066	.066	.066
8192	.062	.062	.062	.062	.062	.062	.062	.062	.062
10240	.050	.050	.050	.050	.050	.050	.050	.050	.050
12288	.041	.041	.041	.041	.041	.041	.041	.041	.041
14336	.071	.071	.071	.071	.071	.071	.071	.071	.035
16384	.062	.062	.062	.062	.062	.062	.062	.062	.062
18432	.055	.055	.055	.055	.055	.055	.055	.055	.055
20480	.075	.075	.075	.075	.075	.075	.050	.050	.050
22528	.068	.068	.068	.068	.068	.068	.068	.068	.068
24576	.083	.062	.062	.062	.062	.062	.062	.062	.062
26624	.076	.076	.076	.076	.076	.057	.057	.057	.057
28672	.071	.071	.071	.071	.071	.071	.071	.071	.071
30720	.083	.083	.066	.066	.066	.066	.066	.066	.066
32768	.078	.078	.078	.062	.062	.062	.062	.062	.062

CI Size	Logical Block Size								
	7506	7670	7677	8182	8187	8188	8189	8871	9211
512	----	----	----	----	----	----	----	----	----
1024	----	----	----	----	----	----	----	----	----
1536	----	----	----	----	----	----	----	----	----
2048	----	----	----	----	----	----	----	----	----
2560	----	----	----	----	----	----	----	----	----
3072	----	----	----	----	----	----	----	----	----
3584	----	----	----	----	----	----	----	----	----
4096	----	----	----	----	----	----	----	----	----
4608	----	----	----	----	----	----	----	----	----
5120	----	----	----	----	----	----	----	----	----
5632	----	----	----	----	----	----	----	----	----
6144	----	----	----	----	----	----	----	----	----
6656	----	----	----	----	----	----	----	----	----
7168	----	----	----	----	----	----	----	----	----
7680	.066	.066	----	----	----	----	----	----	----
8192	.062	.062	.062	.062	----	----	----	----	----
10240	.050	.050	.050	.050	.050	.050	.050	.050	.050
12288	.041	.041	.041	.041	.041	.041	.041	.041	.041
14336	.035	.035	.035	.035	.035	.035	.035	.035	.035
16384	.062	.062	.062	.062	.062	.031	.031	.031	.031
18432	.055	.055	.055	.055	.055	.055	.055	.055	.055
20480	.050	.050	.050	.050	.050	.050	.050	.050	.050
22528	.068	.045	.045	.045	.045	.045	.045	.045	.045
24576	.062	.062	.062	.062	.062	.062	.041	.041	.041
26624	.057	.057	.057	.057	.057	.057	.057	.057	.038
28672	.053	.053	.053	.053	.053	.053	.053	.053	.053
30720	.066	.066	.066	.050	.050	.050	.050	.050	.050
32768	.062	.062	.062	.062	.062	.062	.062	.046	.046

CI Size	Number of Logical Blocks per FBA Block								
	Logical Block Size								
	9554	10230	10235	10236	10919	11259	12278	12283	13307
512	——	——	——	——	——	——	——	——	——
1024	——	——	——	——	——	——	——	——	——
1536	——	——	——	——	——	——	——	——	——
2048	——	——	——	——	——	——	——	——	——
2560	——	——	——	——	——	——	——	——	——
3072	——	——	——	——	——	——	——	——	——
3584	——	——	——	——	——	——	——	——	——
4096	——	——	——	——	——	——	——	——	——
4608	——	——	——	——	——	——	——	——	——
5120	——	——	——	——	——	——	——	——	——
5632	——	——	——	——	——	——	——	——	——
6144	——	——	——	——	——	——	——	——	——
6656	——	——	——	——	——	——	——	——	——
7168	——	——	——	——	——	——	——	——	——
7680	——	——	——	——	——	——	——	——	——
8192	——	——	——	——	——	——	——	——	——
10240	.050	.050	——	——	——	——	——	——	——
12288	.041	.041	.041	.041	.041	.041	.041	——	——
14336	.035	.035	.035	.035	.035	.035	.035	.035	.035
16384	.031	.031	.031	.031	.031	.031	.031	.031	.031
18432	.027	.027	.027	.027	.027	.027	.027	.027	.027
20480	.050	.050	.050	.025	.025	.025	.025	.025	.025
22528	.045	.045	.045	.045	.045	.045	.022	.022	.022
24576	.041	.041	.041	.041	.041	.041	.041	.041	.020
26624	.038	.038	.038	.038	.038	.038	.038	.038	.038
28672	.053	.035	.035	.035	.035	.035	.035	.035	.035
30720	.050	.050	.050	.050	.033	.033	.033	.033	.033
32768	.046	.046	.046	.046	.046	.031	.031	.031	.031

CI Size	Logical Block Size								
	14326	14331	15355	16374	16379	18422	20470	22518	24566
512	——	——	——	——	——	——	——	——	——
1024	——	——	——	——	——	——	——	——	——
1536	——	——	——	——	——	——	——	——	——
2048	——	——	——	——	——	——	——	——	——
2560	——	——	——	——	——	——	——	——	——
3072	——	——	——	——	——	——	——	——	——
3584	——	——	——	——	——	——	——	——	——
4096	——	——	——	——	——	——	——	——	——
4608	——	——	——	——	——	——	——	——	——
5120	——	——	——	——	——	——	——	——	——
5632	——	——	——	——	——	——	——	——	——
6144	——	——	——	——	——	——	——	——	——
6656	——	——	——	——	——	——	——	——	——
7168	——	——	——	——	——	——	——	——	——
7680	——	——	——	——	——	——	——	——	——
8192	——	——	——	——	——	——	——	——	——
10240	——	——	——	——	——	——	——	——	——
12288	——	——	——	——	——	——	——	——	——
14336	.035	——	——	——	——	——	——	——	——
16384	.031	.031	.031	.031	——	——	——	——	——
18432	.027	.027	.027	.027	.027	.027	——	——	——
20480	.025	.025	.025	.025	.025	.025	.025	——	——
22528	.022	.022	.022	.022	.022	.022	.022	.022	——
24576	.020	.020	.020	.020	.020	.020	.020	.020	.020
26624	.019	.019	.019	.019	.019	.019	.019	.019	.019
28672	.035	.035	.017	.017	.017	.017	.017	.017	.017
30720	.033	.033	.033	.016	.016	.016	.016	.016	.016
32768	.031	.031	.031	.031	.031	.015	.015	.015	.015

CI Size	Number of Logical Blocks per FBA Block			
	Logical Block Size			
	26614	28662	30710	32758
512	———	———	———	———
1024	———	———	———	———
1536	———	———	———	———
2048	———	———	———	———
2560	———	———	———	———
3072	———	———	———	———
3584	———	———	———	———
4096	———	———	———	———
4608	———	———	———	———
5120	———	———	———	———
5632	———	———	———	———
6144	———	———	———	———
6656	———	———	———	———
7168	———	———	———	———
7680	———	———	———	———
8192	———	———	———	———
10240	———	———	———	———
12288	———	———	———	———
14336	———	———	———	———
16384	———	———	———	———
18432	———	———	———	———
20480	———	———	———	———
22528	———	———	———	———
24576	———	———	———	———
26624	.019	———	———	———
28672	.017	.017	———	———
30720	.016	.016	.016	———
32768	.015	.015	.015	.015

A

B

C

D

E

[W]

Fixed Block DASD
Installation/Conversion
Guide
GC20-1879-0

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name and mailing address:

_____

_____

_____

Note: Staples can cause problems with automated mail sorting equipment. Please use pressure sensitive or other gummed tape to seal this form.

What is your occupation? _____

Number of latest Newsletter associated with this publication: _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

GC20-1879-0

Reader's Comment Form

Cut or Fold Along Line

Fold and tape | Please Do Not Staple | Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department 6R1BP
180 Kost Road
Mechanicsburg, PA 17055

Fold and tape | Please Do Not Staple | Fold and tape

IBM

Fixed Block DASD Installation/Conversion Guide File No. 4300-34 Printed in U.S.A. GC20-1879-0

GC20-1879-0

Fixed Block DASD Installation/Conversion Guide File No. 4300-34 Printed in U.S.A. GC20-1879-0

GC20-1879-0