ABSTRACT

The MVS Data Library has been completely redesigned to support the I/O access features of the Version 6 SAS System and to improve the Host interface to MVS Data Management facilities. I/O device independence, multivolume Library support, and improved DASD space utilization and I/O performance are all provided by this architecture. The new Library structure is designed to be consistent with IBM’s direction in System Managed Storage as well as providing a platform for future MVS SAS data management features.

This paper will discuss the new architecture and provide examples of how it overcomes limitations in the existing MVS SAS Data Library implementation.

NOTE

This document covers the Version 6 MVS SAS Data Library design. Some of the capabilities presented may not be delivered in the first release of Version 6 SAS Software for MVS. Such capabilities will be noted as they are discussed.

INTRODUCTION

The MVS SAS Data Library is an MVS physical data set with an internal format defined by SAS Software. The internal format subdivides the DASD space allocated to the MVS data set into space for each SAS Data Set stored in the Library. SAS Data Sets are members of the Library.

Some of the outstanding features of the Library design include: the reuse of free Library DASD tracks without requiring reorganization, direct access capabilities for each member of the Library, and the fact that Library DASD space requires no preformatting or special initialization.

The Version 5 MVS SAS Data Library format was first introduced in SAS76. Although it has supported SAS Software well for nearly 15 years, the evolution of MVS data management has exposed several limitations in the Version 5 Library format. The problems of I/O device dependence, wasted DASD space, and inflexibility for improving I/O performance or adding functionality provided the basis for a complete redesign of the Data Library in Version 6 SAS Software for MVS.
VERSION 6 DATA LIBRARY DESIGN GOALS

• I/O Device Independence
• Optimal Utilization of DASD Space
• Optimization of Physical DASD I/O Operations
• Optimization of Library Directory Operations
• Future Expandability

I/O DEVICE INDEPENDENCE

To achieve I/O device independence, the MVS physical data set which contains the Version 6 Data Library is comprised of fixed-length physical DASD blocks. The unit of internal space allocation in the Library is a single DASD block and the internal format of the Library does not depend on DASD Geometry.

The DCB attributes of the Version 6 Data Library are:

• DSORG = PS
• RECFM = FS
• BLKSIZE = 4096 to 23040 by 512
• LRECL = BLKSIZE

These characteristics allow standard utilities like IEBGENER and ISPF 3.3 to transfer a Version 6 Data Library between unlike I/O device types. Automatic migration between unlike devices in software such as IBM's DFHSM should not be a problem either. The Version 6 access method establishes the DASD I/O environment for a Data Library when the MVS data set that contains the Library is physically OPENed. Only the access method will require modification to support future I/O devices and/or changes in the MVS DASD I/O interface. I/O device independence should position the Version 6 MVS SAS Data Library for System Managed Storage.

OPTIMAL UTILIZATION OF DASD SPACE

In the Version 5 Data Library, Members occupy DASD space in track units and an average of 50% of the last track of each member is wasted DASD space. Statistically, as many times as 75% of the last track of Member is used only 25% of the track is used; as many times as 90% of the track is used only 10% is used,
etc.. Many SAS Data Libraries, especially those used by large Computer Performance Evaluation applications can easily have 800 to 1500 Members. This translates to several hundred wasted DASD tracks on these Libraries.

There is potential for wasted space with any unit of storage allocation. However, optimization can be achieved by tuning the unit of DASD space allocation to the application that uses the space. In the Version 6 Data Library, Members occupy DASD space in fixed-length blocks and DCB BLKSIZE can be set when the Data Library is initially allocated. MVS SAS Software does not allow BLKSIZE to exceed half-track on a 3380, so track blocks can be divided between different Library Members without wasting DASD space.

The "rules of thumb" for choosing a BLKSIZE when allocating a Version 6 Data Library are fairly simple. If your Library will primarily contain members with a large number of large observations, choose the largest BLKSIZE that MVS SAS Software supports for the device (23040 on a 3380). This results in the minimum amount of overhead required to map and transfer Library DASD blocks as there are fewer total blocks to handle. If your Library is more likely to contain members with a small number of small observations or members whose size and number of observations varies widely, choose a smaller BLKSIZE like 6144. This increases the opportunity for apportioning DASD space between Library Members and generally results in excellent utilization of DASD space. If BLKSIZE is not specified when a Data Library is physically OPENed for the first time, 6144 is chosen as the default.

OPTIMIZATION OF PHYSICAL DASD I/O OPERATIONS

Before explaining the details of the DASD I/O optimizations in the Version 6 Data Library, it is necessary to discuss the basic layering scheme in the Version 6 SAS BASE I/O Engine. The majority of the code required to access Version 6 SAS Data Sets is written portably in C Language and is used on all Host platforms that support Version 6. This portable code invokes the Host I/O code which is responsible for physically accessing the SAS Data Library on MVS.
The Page I/O Concept

The Portable Component requests I/O from the Host in Page units. A Page in the Version 6 Data Library implementation is simply an integral number of Library DASD blocks and is not to be confused with a virtual memory page in the IBM 370 architecture. Page size is an attribute of each Library Member and can be explicitly set via the BUFSIZE option. See (Clifford 1989) for more details on the BUFSIZE option. The Page concept allows SAS observations to span DASD blocks and provides an elegant way for each SAS Data Set in a Library to have an optimal unit of storage allocation backed by the advantages of a fixed-length DASD block architecture. Consider the following diagram:

```
TRACK | BLOCK 1 | 2 | 3 | 4 | 5 | 6 | 7
0
  | LBLK 1 | LBLK 2 | LBLK 3 | LBLK 4 | LBLK 5 | LBLK 6 | LBLK 7
1
  | LBLK 8 | LBLK 9 | LBLK 10 | LBLK 11 | LBLK 12 | LBLK 13 | LBLK 14
  | DATA01 | DATA01 | PAGE 1 | PAGE 1 | PBLK 1 | PBLK 2 |
2
  | LBLK 15 | LBLK 16 | LBLK 17 | LBLK 18 | LBLK 19 | LBLK 20 | LBLK 21
  | DATA01 | DATA01 | PAGE 2 | PAGE 1 | PBLK 2 | PBLK 1 | PBLK 2
3
  | LBLK 22 | LBLK 23 | LBLK 24 | LBLK 25 | LBLK 26 | LBLK 27 | LBLK 28
  | PAGE 2 | PAGE 1 | PAGE 1 | PAGE 1 | PAGE 1 | PBLK 3 |
4
  | LBLK 29 | LBLK 30 | LBLK 31 | LBLK 32 | LBLK 33 | LBLK 34 | LBLK 35
  | PAGE 1 | PBLK 4 |
```

DEVICE=3380
BLKSIZE=6144
BLOCKS/TRACK=7

The previous diagram represents the first five tracks of a Version 6 MVS SAS Data Library. The LBLK numbers are DASD blocks relative to the beginning of the
Library. DATA01 and DATA02 are Library Members used for this example. PAGE is the respective Page number of each Member and PBLK is the underlying DASD block for a particular segment of a Page. Notice that the DASD blocks for PAGE 1 of Member DATA02 are in discontiguous DASD locations. The Host BASE I/O Engine component is responsible for materializing this Page into contiguous I/O buffer memory. The following diagram illustrates:

```
<table>
<thead>
<tr>
<th>LBLK 18</th>
<th>LBLK 19</th>
<th>LBLK 25</th>
<th>LBLK 34</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA02</td>
<td>DATA02</td>
<td>DATA02</td>
<td>DATA02</td>
</tr>
<tr>
<td>PAGE 1</td>
<td>PAGE 1</td>
<td>PAGE 1</td>
<td>PAGE 1</td>
</tr>
<tr>
<td>PBLK 1</td>
<td>PBLK 2</td>
<td>PBLK 3</td>
<td>PBLK 4</td>
</tr>
</tbody>
</table>
```

Host Component Layering

The Host BASE I/O Engine Component is layered in a scheme that parallels the functional subdivisions of the Version 6 Data Library design. The Library Member Services layer provides basic services to open, close, read, write, delete, rename, etc. Library Members. The Library DASD Space Management Services layer handles requests to allocate or free Library DASD blocks. This layer also controls all Library Directory processing. The Physical DASD I/O Services layer is the DASD access method for the Version 6 Data Library. It uses proprietary EXCP-level technology to dynamically generate optimal channel programs for each DASD I/O operation. The generated channel programs are optimized for the characteristics of the I/O device being accessed.

```
<table>
<thead>
<tr>
<th>Library Member Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Library DASD Space Management Services</td>
</tr>
<tr>
<td>Physical DASD I/O Services</td>
</tr>
<tr>
<td>- EXCP -</td>
</tr>
</tbody>
</table>
```

The Physical DASD I/O Services layer can be interchanged with a device independent block level access method should IBM migrate away from EXCP in System Managed Storage. The Version 6 Data Library architecture is not bound to EXCP-level I/O device support and a properly designed System Managed Storage access method from IBM could simplify ongoing development and support of the Data Library. However, EXCP is currently the only technology that meets the functional and performance requirements of the Data Library.

08APR89 700
DASD I/O Optimizations

Many large sequential batch SAS applications suffer from being I/O bound and require disproportional amounts of elapsed job time because they constantly WAIT for DASD I/O. In Version 5, this is due to the fact that only one DASD block is transferred between the device and the buffer in a single DASD I/O operation. This also results in excessive channel/device connect time because a DASD Queue/Seek/Search sequence is repeated for each I/O yet only results in the transfer of single DASD block. The Version 6 BASE I/O Engine is capable of transferring multiple DASD blocks in a single I/O operation. There are two ways this can occur.

First, the Host component always attempts to transfer the DASD blocks that comprise a Page in a single I/O request. Increasing Page size with the BUFSIZE option will reduce the number of DASD I/O operations required when sequentially processing a Library Member. However, a larger BUFSIZE will increase the amount of potentially wasted DASD space in the last Page of a Member. Increasing BUFSIZE also requires more buffer memory. When running Version 6 under MVS/XA or MVS/ESA, I/O buffers are obtained above the 16 Meg line and channel programs transfer data directly between the I/O buffer and the DASD.

Second, if data is being processed sequentially, the BUFNO option determines the number of consecutive Pages the Host component attempts to chain the transfer of in a single I/O request. This type of processing is commonly referred to as look ahead read/deferred write. It can be used to tune applications to reduce elapsed job time for sequential processing during a batch window while still using a Page size optimal for interactive processing during prime shift. The BUFNO Page DASD I/O chaining capability may not arrive in the initial release of Version 6 SAS Software for MVS.

The only time multiple DASD I/O operations may be required for an access method I/O request is when the request spans certain device boundaries. In this case the EXCP-level access method generates the channel program segment up to the device boundary, issues an EXCP for that segment and continues channel program generation asynchronously while the previous channel program segment(s) are running. Only when all channel program segments for an I/O request are complete does the access method WAIT on the DASD I/O operation(s). This provides the potential for overlapping of DASD I/O and channel program generation.

Multivolume Support

Multivolume Support for the Version 6 Data Library is implemented in the Physical DASD I/O services layer. The internal format of the Library is completely independent of the Multivolume capabilities. Volumes can be added dynamically as additional Library DASD space is required. If System Managed Storage functions compress a Library DASD allocation to fewer volumes or expand the allocation to
additional volumes, Data Library integrity will not be impacted. Multivolume Data Libraries may exist across DASD volumes connected to heterogeneous Storage Director types and models. The access method will dynamically generate the correct channel program for each volume accessed. Basic Multivolume support has been developed but may not appear in the initial release of Version 6 SAS Software for MVS.

OPTIMIZATION OF LIBRARY DIRECTORY OPERATIONS

The Version 6 Data Library Directory is implemented with a proprietary internal format that controls access to Library Members and manages Library DASD space. The internal format is comprised of related sets of Directory segments linked together to provide the functions required to define, access, and update a Library. Directory segments are subdivisions of Library DASD Blocks and the segment addressing scheme is independent of DASD type. The maximum size for a Directory segment is 512 bytes thus the requirement that Library BLKSIZE be an integral multiple of 512.

The number of members a Library may contain is limited only by the amount of available DASD space. Library DASD blocks containing directory segments can be scattered throughout the Library. A contiguous allocation of DASD blocks is assigned from the beginning of the Library to the Directory when the Library is initialized. This helps optimize Directory I/O performance. If Library growth consumes this allocation, Directory blocks are obtained from the lowest free DASD block location in the Library. Because of the unique structure of the Directory, minimal DASD I/O is required for Directory Processing no matter how many members a Library contains. (Squillace 1989) provides an excellent analysis of Version 6 Library Directory I/O performance.

FUTURE EXPANDABILITY

The internal format of the Version 6 Data Library was designed to provide expandability in support of future data management capabilities and requirements in MVS SAS Software. The goal of this design is to be able to support the data structures required to implement new capabilities within the existing internal format of the Library. Since the Directory is composed of discrete functional sets, new sets can be added to support the required capability.

Extensive research was conducted to develop the architecture of the Version 6 Data Library. During this research two prototypes were built before the final Library internal format was determined. Much of the research centered on providing high performance; especially for Libraries with large numbers of large Members. As a result, a future release of Version 6 SAS Software for MVS will include an advanced Library Directory Naming Tree structure. This will provide even greater Directory processing performance for Libraries with large numbers of Members. The ability to add this function in a future release is a direct result of Future Expandability in the Library Directory design.

08APR89 702
SUMMARY

The Version 6 MVS SAS® Data Library has been completely redesigned. This design overcomes the limitations of the Version 5 Library by providing Device Independence, Optimal Utilization of DASD space, Optimization of DASD I/O operations, Optimization of Directory Operations, and Future Expandability. The Version 6 Library architecture maximizes tradeoffs in DASD space utilization and DASD I/O optimization to achieve a synergism that is congruent with IBM directions in System Managed Storage.

References:


Acknowledgments:

The author is indebted to David Jefferys of the Graphics Development Division at SAS Institute Inc. Cary, NC for producing the Annotate Graphics used in the SUGI presentation.