INFORMATION.

New functions in the data set allocation area.

Table of contents.

System-Determined Blocksize.

Description and Benefits.

System-Determined Blocksize is a new feature for sequential and partitioned DASD data sets. Provided that certain criteria described in this section are met, the system will choose the optimum blocksize for a data set.

The benefits are as follows:

- o More efficient use of DASD
- o Better people productivity
- o Device independence
- o Improved performance

Getting a System-Determined Block Size.

- o During data set allocation:

  1. The data set organisation (DSORG) must be physical sequential (PS) or partitioned (PO).

  2. the record format (RECFM) must be fixed blocked, or variable blocked (FB, FBS, FBA, FBM, FBT, VB, VBS, VBA, or VBM).

  3. The logical record length (LRECL) must be specified.

  4. Block size (BLKSIZE) must be omitted or specified as zero.

  If all the above conditions are satisfied, the system will calculate an optimal block size for the data set.

  If block size is not specified, or is zero and DSORG, RECFM, or LRECL is not specified, the block size is set to zero - the optimum block size has not been selected.

- o During data set open:

  If an "optimal block size selected" has been selected, OPEN will consider the data set to be "reblockable".

  If during open, one or both of the following conditions are true:

  - The block size is zero (set by the program for example) or

  - LRECL or RECFM are changed

  then the DASD Calculation Services (DASDCALC) is called to compute the block size.

Usage considerations:

Allocation via ISPF/PDF panel 3.2 option "A" does not accept a block size of zero or blank and allocation via panel 3.2 option "M" does not accept a block size of zero. This will be changed soon. The panels will accept a block size of zero or blank which generates a system-determined blocksize.

SYSLIN block size: Note that the block size for SYSLIN in the linkage editor is still 3200 bytes. System-determined block size should not be applied to this data set.

Example 1.

In this example, a data set (USER.EXTRACT) is allocated with system-determined block size. A program (MOVEDATA) then writes to this data set, overriding the LRECL in the program. Because the LRECL has changed, the system also changes the block size.

JOB1 allocates a new dataset; BLKSIZE is omitted.

```
//JOB1    JOB
//ALLOC   EXEC PGM=IEFBR14
//DD1      DD  DSN=USER.EXTRACT,DISP=(NEW,CATLG),
//             SPACE=(80,(5,1)),AVGREC=K,
//             LRECL=80,RECFM=FB,DSORG=BLKSIZE=0
```

JOB2 lists the DSCB for the newly created data set.

```
//JOB2    JOB
//LIST    EXEC PGM=IEHLIST
```

The listed DSCB containes:

```
SMS/IND  B        optimal block size selected
LRECL    80       record length is 80
RECFM    FB       fixed blocked
BLKSIZE  23440    block size is 23440
```

JOB3 moves data to the data set. The program 'MOVEDATA' copies the input data set to the output data set. In the DCB for the output data set 'USER.EXTRACT', the program 'MOVEDATA' sets the LRECL to 100 bytes.

```
//JOB3    JOB
//GEN     EXEC PGM=MOVEDATA
//SYSIN    DD  DUMMY
//INPUT    DD  DSN=USER.MASTER,DISP=OLD
//OUTPUT   DD  DSN=USER.EXTRACT,DISP=OLD
```

JOB4 lists the DSCB for the data set 'USER.EXTRACT' after the move. Because the LRECL changed, the optimal block size was calculated. The block size has been changed from 23440 to 23400.

```
//JOB4    JOB
//LIST    EXEC PGM=IEHLIST
```

The listed DSCB containes:

```
SMS/IND  B        optimal block size selected
LRECL    100      record length is 100
RECFM    FB       fixed blocked
BLKSIZE  23400    selected block size is 23400
```

Recall Reblocking.

DFHSM recall and recover will be changed to reblock physical sequential and partitioned data sets with record format FB, FBA, VB and VBA regardless if the data set was marked as reblockable due to system-determined blocksize at migration or backup. System-determined block size will be used.

Tape Concatenation Enhancements.

Tape data sets may be concatenated in any order of block size. DASD and tape data sets may also be concatenated.

GDG Management Enhancements.

In earlier releases, in order to change the number of entries in a GDG, you must:

- Uncatalog all the generation data sets (GDS'es)
- Delete the GDG
- Redefine the GDG
- Recatalog the GDS'es

The ALTER command has a new LIMIT parameter. The format of the command is:

```
ALTER gdgname LIMIT(nnn)
```

where nnn specifies the new maximum number of GDS'es in the range 1-255.

WARNING:
If the current number of active generations exceeds the new limit, the oldest generations will be "rolled off" until the limit is reached. The rolled-off generation data sets will be uncataloged and deleted if SCRATCH was specified for the GDG.

New AMS command ALLOCATE.

The ALLOCATE command can now be issued through Access Method Services (AMS) as well as in TSO.

Example:

```
//IDCAMS  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//DDIN     DD  DSN=USER.VSAM.DS,DISP=SHR
//SYSIN    DD  *
                              /* TRY TO ALLOCATE NEW OUTPUT */
   ALLOC DSN(USER.TEMP.REPRO) -
         NEW LRECL(80) RECFM(V B) AVGREC(K) DSORG(PS) -
         SPACE(1 1) MGMTCLAS(DEL2)
                              /* ELSE REUSE THE OLD ONE */
   SET MAXCC = 0              /* RESET RETURN CODE */

                              /* RUN REPRO */
   REPRO INFILE(DDIN) OUTDATASET(USER.TEMP.REPRO) -
         COUNT(5)
```

New AMS KILOBYTES and MEGABYTES keywords.

For the DEFINE command the new keywords KILOBYTE and MEGABYTE specifies the amount of space needed. The system allocates the minimum number of tracks or cylinders required to contain the specified number of kilobytes or megabytes. A 4K block size is used in the computation of the required space.

Example:

```
//IDCAMS  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *

   DEFINE CLUSTER   (NAME(USER.VSAM.KSDS) -
          KILOBYTES (200 200) -
          ...
```

Changed DD Statement Parameters or Subparameters.

The following parameters for defining new data sets have been changed: (For more information see JCL User's Guide.)

- SPACE=(reclen...) specifies the average record length when SMS is active and AVGREC is specified.
- BLKSIZE determines the block size for a new data set, but if not specified, the system determines an optimal block size.
- It's allowed to specify EXPDT and RETPD without coding "LABEL=".
- It's allowed to specify all DCB keyword subparameters (RECFM, LRECL, for example) without coding "DCB=".

Allocating VSAM Data Sets Using JCL.

VSAM data sets can be allocated using JCL DD statements together with other DD statements within a job step. A separate job step previously used to call Access Method Services to define VSAM cluster may not be necessary when JCL DD statements are used.

JCL example.

```
//MYPGM1  EXEC PGM=MYPGM1
//VSAM1    DD  DSN=USER.VSAM.DBASE,DISP=(NEW,CATLG),
//             RECORG=KS,KEYOFF=9,KEYLEN=6,LRECL=250,
//             AVGREC=K,SPACE=(250,(15,5)),
//             MGMTCLAS=NOBACKUP
```

The JCL DSNAME parameter defines the name of the cluster. However, there is no way to specify data and index component name. The system generates these names when needed as follows:

Example 1. ("CLUSTER" replaced by "DATA"/"INDEX")

```
DSN=USER.VSAM1.CLUSTER  -->   CLUSTER --- USER.VSAM1.CLUSTER
                              DATA    --- USER.VSAM1.DATA
                              INDEX   --- USER.VSAM1.INDEX
```

Example 2. ("DATA"/"INDEX" appended)

```
DSN=USER.VSAM2          -->   CLUSTER --- USER.VSAM2
                              DATA    --- USER.VSAM2.DATA
                              INDEX   --- USER.VSAM2.INDEX
```

Some limitations exist in allocating VSAM data sets via JCL. Not all AMS DEFINE CLUSTER attributes can be specified via JCL DD keywords. See JCL User's Guide.

Temporary VSAM data is allocated like non-VSAM datasets if the data set name is omitted or preceded with & or &&.

JCL example.

```
//MYPGM2  EXEC PGM=MYPGM2
//VSAM2    DD  DSN=&&TEMPTEST,DISP=(NEW,PASS),
//             RECORG=KS,KEYOFF=4,KEYLEN=7,LRECL=130,
//             AVGREC=K,SPACE=(250,(15,5))
```

Allocating Space in Average Records.

When the AVGREC parameter is used with the SPACE parameter, allocations can be requested in number of records. The user does not need to know the physical characteristics of the device.

To allocate space in average records, the user specifies:

- the AVGREC parameter (the scale factor) and
- the average number of bytes per record as the average record length (first parameter on the SPACE keyword)

The average record length on the SPACE parameter should be the same as the record length (LRECL) unless you are dealing with variable length records. With variable length records:

- the LRECL is the maximum number of bytes in a record and
- the average record length is the average number of bytes per record.

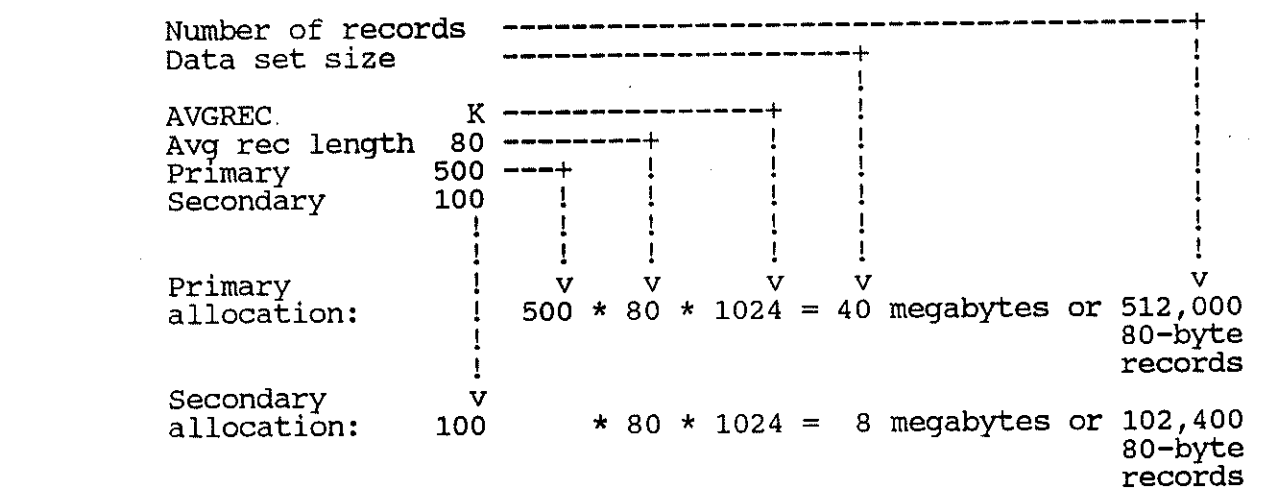
By allocating space using average records, the system will calculate the least number of tracks necessary to contain the amount of space specified. Note that if ROUND is specified, the space will be allocated on a cylinder boundary.

The following example shows the relationship between AVGREC and SPACE.

Example:

Consider the following allocation:

```
//DD1       DD SPACE=(80,(500,100)),AVGREC=K
```



AVGREC is the scale factor:
  where U = multiplier of 1 (U = Unit)
  where K = multiplier of 1024 (K = Kilobyte)
  where M = multiplier of 1048576 (M = Megabyte)

JCL enhancements due to SMS.

Several JCL DD parameters and subparameters have been added or changed.

The system ignores these parameters and subparameters when SMS is not active. This could lead to unexpected results. Therefore, it is advisable not to use any of these new JCL parameters unless SMS is active. This section identifies these JCL parameters that USER SHOULD AVOID USING WHEN SMS IS INACTIVE. For more information see JCL User's Guide.

- AVGREC
- DATACLAS
- KEYOFF
- LIKE
- MGMTCLAS
- RECORG
- REFDD
- STORCLAS

Information.

Fore more informnation and help, please contact dept. 2120, Operations Technical Support, phone 031-667481, memoid VD.SPM or dept. 2510, Large Systems, phone 031-667479, memoid VD.2510LD.

Gothenburg 31 January 1991

VOLVO DATA CORP.
EDP-Techniques

Tomas Svensson

FORSBERG INGE
02510
DA2