



Legac-E Education

Calculate SPACE requirement

A recent question related to how to find the size of a cartridge data set in Megabytes. Following the thread it seemed that the actual requirement was how to determine the amount of space a data set of a given size would occupy on disk. It transpired that the size in Megabytes had already been calculated as information relating to BLKSIZE, Block Count and LRECL was already available via a query against the data set in the Cartridge Management software.

This information is all that is needed to calculate the structure of the SPACE operand provided that AVGREC is used rather than the traditional TRK and CYL notation. This REXX routine, which has been tested with a range of values, performs the SPACE calculation.

```
/* REXX routine to calculate SPACE requirement, from known LRECL,
   BLKSIZE and Block Count */
/* Invoke with 3 parameters, LRECL value, BLKSIZE value, Block
Count*/
Parse Arg lrecl blksize blkcnt .          /* get the knowns */
blkfac = blksize / lrecl                 /* no. records/block */
/*-----*/
/* Calculate the number of records. This will only be approximate */
/* as the last block is likely to be short. */
/*-----*/
reccnt = blkcnt * blkfac                 /* no. of records */
Say 'Total records' reccnt
/*-----*/
/* Calculate the number of records per megabyte of space and */
/* the required Primary SPACE operand value. */
/*-----*/
Select
  When reccnt > 1000000 Then Do
    avgrec = 'M'
    spcpr = reccnt % 1048576
    If spcpr = 0 Then spcpr = 1
  End
  When reccnt > 1000 Then Do
    avgrec = 'K'
    spcpr = reccnt % 1024
    If spcpr = 0 Then spcpr = 1
  End
  Otherwise avgrec = 'U'
    spcpr = reccnt
End
Say 'Suggested SPACE is: SPACE=('lrecl',('spcpr',1)),AVGREC='avgrec'
Exit 0
```



Legac-E Education

Calculate SPACE requirement

Testing was performed with the following job stream, changing the SPACE operand and quantity values in S0020 to ensure results across the range.

```
//S0010 EXEC PGM=IEFBR14
//OUTDD DD DSN=&SYSUID..CALSPACE.TEST,LRECL=80,RECFM=FB,
//          SPACE=(TRK,0),DISP=(MOD,DELETE)
//S0020 EXEC PGM=IEBDG
//SYSPRINT DD SYSOUT=*
//OUTDD DD DSN=&SYSUID..CALSPACE.TEST,LRECL=80,RECFM=FB,
//          BLKSIZE=27920,SPACE=(80,(34,1)),AVGREC=K,
//          DISP=(,CATLG)
//SYSIN DD *
DSD OUTPUT=(OUTDD)
FD NAME=FIELD1,LENGTH=80,FORMAT=AL,ACTION=RP,STARTLOC=1
CREATE QUANTITY=34900,NAME=FIELD1
END
```

The testing was performed with fixed length records but the routine will work with variable length records, although the resultant SPACE operand will not be as accurate.

When about to add a new record to a block, the system calculates whether another maximum length record could be added to the block after the current one is inserted. If that scenario is not true, the current block is written as a short block, and a new block started. This means that more short blocks are likely with RECFM=VB. (RECFM=VBS, was introduced to circumvent this.)

To get a more accurate result when using the REXX routine, use the average record length, rather than the maximum record length. The maximum record length as observed in the JCL or File Label may result is too small a primary request in the SPACE operand which can be compensated for by the inclusion of a secondary request and the RLSE operand to avoid wasting space..