UNCLASSIFIED

AR-001-186

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

ELECTRONICS RESEARCH LABORATORY

TECHNICAL REPORT

ERL-0009-TR

AN EFFICIENT APPLICATION OF DISK STORAGE TO THE DRCS DATA MIGRATION SCHEME

J.C. Gwatking

S U M M A R Y

A data migration scheme has been in operation on the DRCS Central Computer for approximately two years. During this period tens of thousands of datasets have been migrated to magnetic tape to make the best use of disk storage space. The number of dataset retrievals, although still well within the processing capabilities of the IBM 370 computer system, has predictably grown. This report describes an innovation designed to lessen the impact of retrieval requests on the system resources, especially in the operator intensive area of tape handling.

Approved for Public Release

POSTAL ADDRESS: Chief Superintendent, Electronics Research Laboratory,
Box 2151, G.P.O., Adelaide, South Australia, 5001.

UNCLASSIFIED

TABLE OF CONTENTS

LIST OF APPENDICES

1. INTRODUCTION

The data migration scheme developed at DRCS has been the subject of two previous papers(ref.1,2). Since it first came into operation, approximately two years ago, the scheme has been used to transfer tens of thousands of datasets to magnetic tape to minimize the shortage of disk storage space.

With such large numbers of datasets involved, it is reasonable to expect a fairly high retrieval rate, and this has been observed. In fact over the last several months there has been an average of 1000 retrievals per month.

This report is not concerned with analyzing the reasons for the high migration and retrieval rates. It is sufficient to say that the migration scheme has not escaped the effects of thrashing(ref.14,15), which occurs in any virtual system when the real resource (in this case disk storage space) is overtaxed. In practice some relief is being given in this area with the progressive replacement of the IBM 3330-1 disk drives by superior IBM 3350 drives.

The report instead describes an innovation made to the migration scheme to lessen the impact of retrievals on the system resources, particularly with a view to reducing the tape handling necessarily associated with the original approach.

2. AN ANALYSIS OF DATASET CHARACTERISTICS

It had long been suspected that most user datasets were sequential and less than one track in size, although no statistics were available. To confirm this view an analysis was made of all the datasets currently stored on disk and the results obtained were as follows -

(a) The total number of user datasets was 5400.

(b) The total number of datasets that were sequential and used less than one track of 3330 space (regardless of how much they had allocated) was 3000. Thus 55% of disk datasets were of this form.

(c) The average size of these 3000 datasets was 0.28 3330 tracks, or approximately 3700 bytes.

When datasets are shifted to 3350 volumes, the percentage occupying less than one track will increase even further.

To confirm that roughly the same statistic applied to the 11000 datasets in the archives a further analysis was undertaken. It was found that just over 50% of these datasets were sequential and used less than one track. The slight decrease in percentage is explained by the fact that larger datasets are more likely to be migrated than small ones. The same figure (50%) also applies to retrieved datasets. This was verified by scanning the SMF records produced by the migration scheme for retrieval operations(ref.1).

A further relevant statistic is that one third of all retrievals are made for datasets that have been in the archives less than a week.

3. OVERVIEW OF THE NEW APPROACH

The observations presented in Section 2 suggested that some further investigation of better ways to handle small sequential datasets would be profitable.

The primary reason for migration is the reclamation of disk space to accomodate new user datasets. For the larger datasets there is no alternative to tape storage, given the available hardware configuration. However it is apparent that small sequential datasets could be left on disk and still release about 75% of the space they occupy, provided they could be compressed in some way to achieve economy of storage. The easiest way to accomplish this objective is to create a special dataset with direct access capabilities and to copy all small datasets into it, maintaining appropriate control information to enable them to be located and extracted at a later time.

This idea forms the basis of the amendment to the migration scheme. Both a VSAM key-sequenced dataset(ref.13) and a partitioned dataset (PDS)(ref.6) were considered for the storage organization to contain the many small datasets; the PDS was finally selected on the grounds of its simpler backup, recovery and maintenance procedures. A lesser consideration was the amount of space the two types of dataset would require, with the PDS again, on our previous experience, being superior.

The approach, then, is for the weekly migration run to distinguish between small sequential datasets and others. The latter are still copied to magnetic tape, while the small sequential ones, at least half of the total number, are each copied to a different member of a special partitioned dataset. The same distinction is made by the user-initiated archive and backup procedures. The archive catalogue records identify the location of each dataset and contain the member name for those in the PDS.

The advantages of this technique are obvious. Firstly the weekly migration run requires less execution time. Fewer datasets are transferred to tape and the two operations (disk to tape and disk to PDS) can be overlapped. However, the second and major benefit is the halving of the number of retrieval jobs that require a tape mount, with the associated reduction in the average elapsed time of these jobs.

The partitioned dataset needs to be compressed from time to time to reclaim waste space created by retrieval operations, but this is a fairly trivial task. However, of more concern is the tendency for the PDS to grow in size as members are added. The technique used to keep the size under control is to periodically transfer those members that contain datasets migrated some time ago (say six months) onto an archive tape. The probability of them being retrieved after this length of time is quite small, so it is best to remove them from the PDS to make room for more active datasets. In this context the PDS can be considered as a staging area between disk and archive tape, a concept employed by the TSO/MSS Archiver program(ref.3).

A major design constraint was that the new technique should be completely transparent to users. No problems were encountered in meeting this objective.

The following paragraphs describe the characteristics of the partitioned dataset and its integration into the migration scheme in more detail.

4. DETAILS OF THE PARTITIONED DATASET

4.1 Protection status

The name of the special partitioned dataset is SYS1.ARCHIVE.PDS, and it is password protected for both read and write access. There are two reasons for this. The first is so that the dataset can be used to store sensitive information and the second to protect it from accidental or malicious damage.

As a consequence, any program that accesses the PDS must either know its password or be exempt. The former is impractical for the programs of the migration scheme since they are used by many people. The password would therefore rapidly lose its secrecy. The only alternative is to include the relevant migration program names in an Operating System module called the Program Properties Table(ref.4). This gives the programs the privileged status needed to access a protected dataset without knowing its password. Each of these programs includes code of its own to ensure that users cannot access other users' protected datasets without knowing their passwords.

4.2 Dataset characteristics

The partitioned dataset SYS1.ARCHIVE.PDS has very general data control block parameters, namely RECFM=U and BLKSIZE=19069 (the maximum 3350 track size)(ref.5). This means that it can accomodate blocks from almost any type of sequential dataset. The only exceptions are datasets that have keyed records and those that use the track overflow feature. These cannot be stored in the partitioned dataset and must still be migrated to tape. However there are very few such datasets, primarily because the DRCS Computing Centre has prohibited use of track overflow.

The initial space allocation for the PDS was 30 3350 cylinders, with extensions of 5 cylinders. The 500 directory blocks can accomodate 6500 member names.

4.3 The directory

A directory entry of any partitioned dataset, besides identifying the location of a particular member, has provision for storing a maximum of 62 bytes of user-defined data(ref.6).

When a dataset is stored in the PDS it adopts the characteristics of the latter. The retrieval operation therefore requires the original DCB parameters so that it can rebuild the dataset to be identical with its state before being migrated. This restriction does not apply to datasets migrated to tape, since they retain their own characteristics while on tape. Therefore the migration scheme needs to retain the three DCB parametres LRECL, BLKSIZE and RECFM for datasets in SYS1.ARCHIVE.PDS.

The PDS directory entries were chosen to store this extra information. Various other possibilities were evaluated, including an expansion of the archive catalogue records to incorporate the three new entries. However this was abandoned because of the widespread changes that would be required to the migration software. The only disadvantage with the directory approach is that the entries can only be manipulated in assembler language, straightforward though it is.

The format of each 18-byte directory entry in SYS1.ARCHIVE.PDS is as follows. The entries are always in ascending alphabetic sequence of the member names.

Offset	Size	Description
0	8	the member name
8	3	the relative location of the first block of the member (TTR)
11	1	the number of user data halfwords (always 3)
12	2	the blocksize of the dataset (BLKSIZE)
14	2	the record length of the dataset (LRECL)
16	1	the record format of the dataset (RECFM) - see OS/VS1 System Data Areas(ref.7) for the bit representation
17	1	unused (zeros)

4.4 The member names

As each dataset is copied into SYS1.ARCHIVE.PDS a unique member name must be generated for it. This could be done in several ways, such as by using a random generator based on the dataset name or date and time. However the method adopted was to assign 8-byte member names in ascending alphabetic sequence, starting with AAAAAAAA, then AAAAAAAB, AAAAAAAC etc.

The dataset SYS1.ARCHIVE.NEXTMEM always contains the next available member name. Whenever either the weekly migration run or a user-initiated archive or backup operation decides to migrate a dataset to SYS1.ARCHIVE.PDS, the member name in SYS1.ARCHIVE.NEXTMEM is used and incremented for the next request. Once assigned, a member name will not be reused, even though it may subsequently be deleted from the PDS. At the current rate of migration this simple approach would yield enough distinct member names for several thousand years of operation!

Apart from simplicity and the assurance of always assigning a unique member name, this technique has one other major advantage. It minimizes the amount of directory maintenance required during the weekly migration run. Whenever a member is added to or deleted from any partitioned dataset, that part of the directory containing names of higher value is rewritten to ensure that all entries are in ascending alphabetic sequence, with no imbedded free space. The migration scheme will typically add several hundred members to the PDS each run. As each member is written its name will always have a higher alphabetic value than any other already in the partitioned dataset. This means that the entry will always be stored at the end of the directory; consequently no rewriting is necessary, which results in considerable saving of time.

Unfortunately, because of their random nature, deletions from the partitioned dataset each incur the overhead of rewriting part of the directory.

4.5 Maintenance, backup and recovery

As with all partitioned datasets, SYS1.ARCHIVE.PDS will gradually accumulate imbedded unuseable free space as members are deleted. This space is easily recovered by performing a compress operation, using the IEBCOPY utility program(ref.8). This is done once a week, during the automatic migration run.

Because it resides on disk SYS1.ARCHIVE.PDS is subject to continual risk of damage or loss. Although the actual probability is very low there must always be recent backup copies available to enable speedy recovery. Since the disk volume on which it is stored also contains user datasets, SYS1.ARCHIVE.PDS is automatically included in the DRCS selective backup scheme(ref.9). This guarantees that there will always be at least 13 different versions of the dataset on backup tapes, including a copy taken each evening in the past fortnight. Recovery to the version that existed at backup time on the previous night is therefore always possible. Members added to or deleted from the PDS are easily determined by scanning the SMF records produced by the migration scheme since this time(ref.1). Datasets that have been migrated will also need to be recovered from the backup tapes to repeat the operation.

One advantage of migrating datasets to tape is that the actual data is available for a considerable time (even up to a year) after it has been deleted from the archives. On occasions users have made use of this to recover datasets mistakenly thought to have been obsolete at the time of their deletion. So far a 100% success rate has been achieved in meeting these requests. The same service for members of SYS1.ARCHIVE.PDS can be obtained by restoring it from one of the backup copies to a temporary disk dataset. However the maximum recovery period under these circumstances is only about two months, limiting the level of success possible. This could be overcome to a large extent by periodically backing up the PDS to an archive tape using the backup feature of the migration scheme itself. It is considered the overhead incurred would far outweigh the small benefits that might be obtained.

5. CHANGES TO THE ARCHIVE CATALOGUE

The format of the archive catalogue records for tape resident datasets has not changed. However the records for datasets in SYS1.ARCHIVE.PDS do have a slightly different composition.

A previously unused bit in the flag byte (bit 3), now indicates the location of the dataset. If the bit is off, the dataset is on tape, if on it is in the PDS. Naturally the archive tape serial number and dataset sequence number fields (a total of nine consecutive bytes) have no meaning for PDS datasets. Instead the PDS member name occupies the first eight bytes and the last is blank. The complete structure of an archive catalogue record is now -

Offset	Size	Description
0	1	flag byte - bit 0 : 0 if dataset is partitioned : 1 if dataset is sequential - bits 1-2 : dataset protection bits from DSCB - bit 3 : 0 if dataset resides on tape : 1 if dataset resides in SYS1.ARCHIVE.PDS - bits 4-7 : unused (zero)
1	44	dataset name
45	5	last access date (Julian, numeric, display format)
50	5	date migrated (Julian, numeric, display format)
55	5	expiry date (Julian, numeric, display format)
60	5	dataset size in tracks (numeric, display format)
		Tape resident datasets only
65	6	archive tape volume
71	3	dataset sequence number (numeric, display format)
		PDS resident datasets only
65	8	PDS member name

73	1	unused (blank)
74	6	disk volume serial no.

6. ACCESS TO SYS1.ARCHIVE.PDS

All accesses to SYS1.ARCHIVE.PDS are performed by an assembler routine using BPAM macro instructions (OPEN, CLOSE, READ, WRITE, CHECK) and directory maintenance macros (STOW, FIND, BLDL)(ref.10). The name of the module is PDSUTIL and it has the following entry points, which are described in more detail in Appendix I.

(1) PDSOPEN

This entry point opens SYS1.ARCHIVE.PDS for either input or output processing, as required.

(2) PDSSHUT

Entry point PDSSHUT closes the PDS and releases any areas of storage that have been dynamically obtained.

(3) PDSDEL

This entry point deletes a member from the partitioned dataset.

(4) PDSDCB

PDSDCB obtains the three DCB parameters, LRECL, RECFM and BLKSIZE (see Section 4.3) from the directory entry of a particular member.

(5) PDSADD

This entry point copies the contents of a dataset into a member of SYS1.ARCHIVE.PDS and stores the DCB parameters in the directory entry (i.e. the migration operation).

(6) PDSREAD

Entry point PDSREAD extracts the data from a member of SYS1.ARCHIVE.PDS and writes it to a sequential output dataset (i.e. the retrieval operation).

Any other module in the migration scheme software that requires access to SYS1.ARCHIVE.PDS must do so by using a combination of these entry points.

Of particular note is the manner in which entry points PDSADD and PDSREAD overcome the DCB conflicts between the PDS and migrated dataset. As mentioned in Section 4.2, SYS1.ARCHIVE.PDS has undefined length blocks with a maximum size equal to the track size of a 3350. However the migrated dataset may have almost any combination of DCB parameters. Despite this PDSADD allocates the dataset with the same attributes as the PDS. Therefore each block is transferred to the PDS in full as if it had an undefined format and length.

In the retrieval process the same technique is used. PDSREAD allocates the new output dataset with the DCB parameters of SYS1.ARCHIVE.PDS. This allows it to copy each block from the PDS member to the dataset intact. At the conclusion the correct DCB attributes are assigned to the new dataset and the data in it will conform to these. This must be so because all data handling is done at block and not record level.

7. CONCLUDING REMARKS

This report has presented an approach to data archival storage for small sequential datasets, namely by storing them as members of a large partitioned dataset SYS1.ARCHIVE.PDS. The increased storage efficiency of this method saves at least 75% of the space occupied by such datasets. In addition, tape handling overheads are reduced and as more members are added to the PDS at least half of all retrieval operations will be able to obtain their data from disk, without the need for a time-consuming tape mount and search.

There is no real reason for restricting the scheme to single track sequential datasets. Those that use more space (say two or three tracks) could also be included and the tape handling reduced even further. However the larger the dataset the less will be the space saving advantages.

The scheme does not attempt to handle small partitioned datasets at this stage. The extra complexity of programming required was not justified by the anticipated benefits, bearing in mind the comparatively small number of such datasets. However if their numbers grow, there may be some justification for renewed efforts to find a simple and satisfactory solution.

Section 4.4 addressed the general question of directory maintenance and in particular the overhead associated with deleting a member of a partitioned dataset. The actual magnitude of this overhead when the directory contains thousands of members is not known at this stage. However, as the dataset grows, the overhead will be monitored and, if it becomes unacceptable, an alternative method will be implemented. Rather than have several programs that delete members when and as required, as happens now, the members will only be logically deleted by the removal of their associated archive catalogue record. Then, say once a week, a maintenance program could use the IEBCOPY utility to rebuild SYS1.ARCHIVE.PDS, specifically excluding the unused members. This method would only incur the overhead of building a new directory from scratch, in ascending member name sequence, and then only once a week. The reasons for this method not being used from the outset are that it is more difficult to implement and exposes the scheme to a greater likelihood of the archive catalogue and the PDS being unsynchronized and therefore inconsistent.

REFERENCES

No.	Author	Title
1	J.C. Gwatking	'A Data Migration Scheme for the 370/168 computer at W.R.E.' WRE Technical Report 1870(A)
2	R.H. Evans, D.A.B. Fogg	'A Mathematical Model to Simulate the Archiving Scheme at W.R.E.' WRE-TN-1580 (A), March 1976
3	IBM	'TSO/MSS Archiver IUP', 5796-AJK
4	IBM	'OS/VS2 System Programming Library : Job Management', GC28-0627-1
5	IBM	'OS/VS2 JCL', GC28-0692-1
6	IBM	'OS/VS2 MVS Data Management Services Guide', GC26-3875-0
7	IBM	'OS/VS1 System Data Areas', SY28-0605-3
8	IBM	'OS/VS Utilities', GC35-0005-4
9	J.C. Gwatking	'A Selective Dataset Backup Scheme for the WRE Central Computer', WRE Technical Report (draft)
10	IBM	'OS/VS Data Management Macro Instructions', GC26-3793-4
11	A.J. Legg	'DYNPARM - An Assembler Macro for Dynamic allocation of Data Sets in the IBM OS/VS2 Operating System', WRE Technical Memorandum - (draft version)
12	IBM	'OS/VS2 Supervisor Service and Macro Instructions', GC28-0683-0

13	IBM	'OS/VS Virtual Storage Access Method(VSAM) Programmer's Guide', GC26-3838-0
14	J.G. Rogers	'Structured Programming for Virtual Storage Systems', IBM Systems Journal, No.4, 1975
15	D.J. Hatfield, J. Gerald	'Program Restructuring for Virtual Memory', IBM Systems Journal, No.3, 1971

GLOSSARY

Several terms and abbreviations used in this report are unique to IBM computer systems. Although most are satisfactorily explained in the appropriate IBM publication, brief descriptions are included here for quick reference.

BLKSIZE(block size).
: This is the maximum physical record size of a dataset and is a DCB parameter.

BPAM(Basic Partitioned Access Method).
: That part of the IBM Operating System that interfaces between application programs and partitioned datasets.

Compression.
: The process of copying a partitioned dataset to reorganize it and thereby reclaim imbedded free space.

DCB(Data Control Block).
: This is an Operating System control block that contains all particulars of an associated dataset's type, organization and internal format.

DD(data definition) statement.
: A job control statement that associates a dataset with an application program defined file name.

Directory.
: A series of 256-byte records at the beginning of a partitioned dataset that contains an entry for each member.

LRECL(logical record length).
: This is either the maximum or exact length of all logical records in a dataset and is a DCB parameter. The meaning depends on the record type(RECFM). A logical record is defined by a problem program based on data content, as distinct from physical storage characteristics. There may be several per physical block or even several physical blocks per logical record.

PDS(Partitioned dataset).
: This is a dataset in direct access(disk) storage divided into partitions, called members, each of which can contain a separate set of data. Each PDS has a directory to enable the members to be located.

RECFM(record format).
This defines the type of logical records in a dataset and their relationship to the physical blocks and is a DCB parameter. For instance, logical records may be fixed, variable or undefinable in length, blocked or unblocked.

SMF(System Management Facilities)
This is a component of the IBM Operating System that gathers and records details of system events. The data can be used for a variety of purposes, including accounting, performance monitoring and activity reporting.

VSAM(Virtual Storage Access Method).
A type of dataset organization that can provide direct as well as sequential access capabilities, depending on application program requirements.

APPENDIX I

SUMMARY OF ADDITIONS TO THE DATA MIGRATION SCHEME

This Appendix, like Appendix II, assumes prior knowledge of the function and characteristics of the datasets, program modules and catalogued procedures of the data migration scheme. Reference 1 contains full details.

I.1 Datasets

(1) SYS.ARCHIVE.PDS

This dataset is the focal point of the new storage technique for small sequential datasets. It is partitioned, with each member containing the contents of one user dataset. See Section 4 for full details.

(2) SYS1.ARCHIVE.NEXTMEM

This dataset contains a single 80-byte record identifying the name of the next member to be added to SYS1.ARCHIVE.PDS, in the first 8 bytes. It is used and updated by the weekly migration run and by the user-initiated archive and backup operations; it is also password protected.

(3) SYS1.ARCHIVE.PDSUPDT

This dataset is used exclusively by the weekly migration run. It contains a copy of control statements generated by programs ARCHIVE and LSTARCH for processing by program PDSUPDT. They indicate the additions and deletions to be made to SYS1.ARCHIVE.PDS. The format of each 80-byte record is -

Offset	Size	Description
0	1	record type - 'A' for addition, 'D' for deletion
1	8	PDS member name
9	44	dataset name associated with the member
	Deletion records only	
53	27	unused (blank)

Addition records only		
53	2	number of characters in the dataset name (binary)
55	1	bit representation from the format 1 DSCB of the dataset's RECFM
56	2	dataset's LRECL (binary)
58	2	dataset's BLKSIZE (binary)
60	6	disk volume containing dataset
66	4	device type of disk volume ('3330' or '3350')
70	10	unused (blank)

I.2 Program Modules

The PL/I attributes of subprogram arguments are indicated where appropriate.

(1) PDSUTIL

entry points - PDSOPEN, PDSSHUT, PDSDEL, PDSDCB, PDSADD, PDSREAD
type - assembler, subroutine
called from - ARCHIVE, FORCE2, RETRVE, SCRATCH, PDSUPDT
This module performs all maintenance tasks for SYS1.ARCHIVE.PDS.

(a) Entry point PDSOPEN opens SYS1.ARCHIVE.PDS either for input or output.
Arguments -
open indicator - CHARACTER(1)
- 'I' for input processing,
'O' for output

(b) Entry point PDSSHUT closes SYS1.ARCHIVE.PDS and releases any storage that has been dynamically allocated by either PDSREAD or PDSADD.
Arguments - none

(c) Entry point PDSDEL deletes a member from SYS1.ARCHIVE.PDS. If the operation is not successful the non-zero return code from the failing macro is passed back to the calling program.
Arguments -
member name - CHARACTER(8)
return code - FIXED BINARY(31)

(d) Entry point PDSDCB extracts and returns the DCB parameters from the directory entry of a particular member. A non-zero return code indicates unsuccessful completion.
Arguments -
member name - CHARACTER(8)
RECFM - BIT(8)
LRECL - FIXED BINARY(15)
BLKSIZE - FIXED BINARY(15)
return code - FIXED BINARY(31)

(e) Entry point PDSADD uses the DYNPARM macro instruction to dynamically allocate an input dataset (ref.11). If a buffer area is not already available, the routine will obtain sufficient space using the GETMAIN macro instruction(ref.12). It then opens the dataset with the same DCB attributes as SYS1.ARCHIVE.PDS and transfers its contents to a new member of the latter. At the conclusion the correct DCB attributes of the dataset are stored in the new directory entry and the dataset is closed and dynamically deallocated. The non-zero return code of any failing macro is passed back to the calling program for analysis.
Arguments -
dataset name - CHARACTER(44)
characters in dataset name - FIXED BINARY(15)
RECFM - BIT(8)
LRECL - FIXED BINARY(15)
BLKSIZE - FIXED BINARY(15)
member name - CHARACTER(8)
return code - FIXED BINARY(31)

(f) Entry point PDSREAD first dynamically obtains a buffer area if one is not already available. Then it transfers the contents of a member of SYS1.ARCHIVE.PDS to a dataset already allocated to filename PREALL by the calling program. The non-zero return code of any failing macro is passed back to the calling program for further analysis.
Arguments -
member name - CHARACTER(8)
return code - FIXED BINARY(31)

(2) INCRMEM

type - PL/I, subroutine
called from - ARCHIVE, FORCE2
This routine accepts an 8-byte alphabetic member name and returns the next name in the alphabetic sequence. For example, member AAAAAACD is incremented to AAAAAACE, while AAAAARZZ becomes AAAAASAA.
Arguments -
old member name - CHARACTER(8)
new member name - CHARACTER(8)

(3) PDSUPDT

type - PL/I, main program
calls - PDSUTIL, ENQDEQ, DELDSN
load module - ARCHPDSU (must have bypass-password-protection status)
Program PDSUPDT is a general purpose update program for SYS1.ARCHIVE.PDS. It is used during the weekly migration run to delete members from the PDS (using PDSDEL) and add members (PDSADD). After successful add operations, the sequential dataset used will be uncatalogued and deleted. Control statements read from file PDSCNTL define the actions to be performed and provide the necessary parameters for routines PDSDEL and PDSADD.
Input format -
(a) File PDSCNTL - dataset SYS1.ARCHIVE.PDSUPDT

Output formats -
(a) File ARCHPDS - dataset SYS1.ARCHIVE.PDS

(b) File SYSPRINT
The program writes messages indicating the success or failure of each operation to this file.

(4) PDSTAPE

type - PL/I, main program
The purpose of this program is to periodically remove old datasets from SYS1.ARCHIVE.PDS and transfer them to an archive tape. This keeps the size of the PDS at a reasonable level. At the time this document was prepared, investigation on the best approach to the problem of moving a large number of members to separate tape datasets was still being carried out.

APPENDIX II

SUMMARY OF ALTERATIONS TO THE DATA MIGRATION SCHEME

II.1 Datasets

(1) SYS1.ARCHIVE.CATLG

The format of archive catalogue records migrated to tape has not changed. However, a bit in the flag byte now indicates which datasets reside in SYS1.ARCHIVE.PDS and the member name is stored in the same location as the tape serial number and file sequence number fields of tape dataset records. See Section 5 for full details.

(2) SYS1.ARCHIVE.ARCHLIST

The only changes to SYS1.ARCHIVE.ARCHLIST occur in the records identifying datasets migrated to SYS1.ARCHIVE.PDS. The nine bytes that would otherwise contain the tape serial number and file sequence number are used instead for the PDS member name plus the character 'Y' in the ninth position.

II.2 Program Modules

The functional changes to existing migration scheme modules imposed by the introduction of SYS1.ARCHIVE.PDS are briefly outlined in this section.

(1) ARCHIVE

This program now performs the extra task of deciding which datasets should be migrated to tape and which to SYS1.ARCHIVE.PDS. For those falling into the latter category, it generates an addition record to dataset SYS1.ARCHIVE.PDSUPDT through filename PDSMEM. These will be processed by program PDSUPDT later to transfer the datasets into SYS1.ARCHIVE.PDS. The fully linked load module of ARCHIVE, named ARCHIV, must now have bypass-password-protection status in the Program Properties Table (see Section 4.1).

(2) ARCHSET

The only addition to this routine is code to delete SYS1.ARCHIVE.PDS members for datasets that are being replaced in the archives by a migration or backup operation.

(3) ENQDEQ

Two entry points, ENQPDS and DEQPDS have been added to the ENQDEQ module. The former obtains a system-wide exclusive ENQ with major name (qname) QPDSARCH and minor name (rname) RPDSARCH, while DEQPDS frees the resource. The routines are called by all modules accessing SYS1.ARCHIVE.PDS to ensure that they have exclusive use of it. This technique is used instead of OLD disposition on the JCL statement to minimize contention.

(4) FORCE2

Like ARCHIVE, module FORCE2 decides whether a dataset should go to tape or the PDS. In the latter case it invokes the routine PDSADD to perform the addition.

(5) GETVTOC

GETVTOC now returns the complete format1 DSCB of each dataset to the calling program, so that it can extract the information it needs. Previously only selected fields were returned, but these proved to be insufficient for the extra functions of program ARCHIVE.

(6) LSTARCH

The main changes to LSTARCH occur in the formats of the reports it produces, since provision must be made for the two distinct types of archive catalogue records. In addition the program now generates control statements for program PDSUPDT to delete PDS members associated with datasets that have expired during the week. The filename used is MEMDEL.

(7) PREALLC

This routine dynamically allocates space for sequential datasets that are being retrieved. It previously ignored the DCB parameters, since they are automatically set by the IEHMOVE utility when the dataset is copied from tape. However PREALLC must now allocate sequential datasets that are being retrieved from SYS1.ARCHIVE.PDS with the correct DCB parameters. They are obtained from the member's directory entry (see Section 4.3).

(8) RETRVE

This module has undergone the greatest change. It now must determine whether the dataset being retrieved resides on tape or in SYS1.ARCHIVE.PDS. In the latter case it invokes entry points in module PDSUTIL to perform the operation. In particular it calls PDSDCB to obtain the dataset's DCB attributes from the directory entry of its PDS member. These are used by subroutine PREALLC. Next PDSREAD copies the contents of the member into the preallocated space. Finally, for a retrieval operation only (as distinct from a reload) it uses PDSDEL to delete the member.

Since RETRVE accesses SYS1.ARCHIVE.PDS its load module (ARCHRET) must now have bypass-password-protection status in the Program Properties Table (see Section 4.1). This means that it can now dispense with creating and deleting artificial passwords to satisfy the Operating System's password checking mechanism. In particular entry points PROTAS and PROTDS of module PROTADD are no longer used and LNKRET has been replaced by LNKMOVE for retrieving tape resident datasets.

However RETRVE still maintains its own checks on the protection status of user datasets.

(9) SCRATCH

The only alteration to program SCRATCH is the provision for deleting members of SYS1.ARCHIVE.PDS associated with datasets being deleted from the archives. The load module (ARCHSCR) must now have bypass-password-protection status.

(10) TAPEMAP

Module TAPEMAP produces a listing of the current archive contents by tape serial number. It now also reports on datasets resident in SYS1.ARCHIVE.PDS.

(11) LNKRET

This subroutine, previously called by program RETRVE to access datasets on an archive tape, is no longer used by the migration scheme software.

II.3 Catalogued Procedures

(1) General

All procedures that invoke programs which use SYS1.ARCHIVE.PDS must have a DD card named ARCHPDS defining this dataset in the appropriate step. These include RETRIEVE, RELOAD, ARCHIVE and BACKUP(step ARCHIV), and ARCHLIST(step ARCH). In addition the same steps in procedures ARCHIVE, BACKUP and ARCHLIST require a NEXTMEM DD statement defining dataset SYS1.BACKUP.NEXTMEM.

Procedure ASCRATCH(step SCR) must now include a SYSPRIN DD statement for error messages directed to system programmers (output class S).

(2) ARCHLIST

The ARCHLIST catalogued procedure has several other changes specific to it.

There must be DD cards named PDSMEM in step ARCH and MEMDEL in step PRINT to receive control statements for action by program PDSUPDT (load module ARCHPDSU). In addition there are two new steps after PRINT. The first (ADDEL) adds the records from file MEMDEL to the dataset SYS1.ARCHIVE.PDSUPDT, while the second (DEL) executes ARCHPDSU to effect the member deletions. Finally there is a new step (SUBMIT) after step ARCH. It submits a batch job to the internal reader that compresses SYS1.ARCHIVE.PDS and invokes ARCHPDSU to add the new members.

The procedures ARCHMOVE and ARCHCONT, which are used to recover from error situations in ARCHLIST, also contain those additions mentioned above that are relevant to their subset of steps.

DOCUMENT CONTROL DATA SHEET

Security classification of this page: UNCLASSIFIED

1 DOCUMENT NUMBERS	
AR Number:	AR-001-186
Report Number:	ERL-0009-TR
Other Numbers:	

2 SECURITY CLASSIFICATION	
a. Complete Document:	UNCLASSIFIED
b. Title in Isolation:	UNCLASSIFIED
c. Summary in Isolation:	UNCLASSIFIED

3 TITLE

AN EFFICIENT APPLICATION OF DISK STORAGE TO THE DRCS DATA MIGRATION SCHEME

4 PERSONAL AUTHOR(S):

J.C. Gwatking

5 DOCUMENT DATE:

April 1978

6		
6.1	TOTAL NUMBER OF PAGES	23
6.2	NUMBER OF REFERENCES:	15

7 7.1 CORPORATE AUTHOR(S):

Electronics Research Laboratory

7.2 DOCUMENT SERIES AND NUMBER

Electronics Research Laboratory 0009-TR

8 REFERENCE NUMBERS	
a. Task:	DST 20/14
b. Sponsoring Agency:	

9 COST CODE:

228753

10 IMPRINT (Publishing organisation)

Defence Research Centre Salisbury

11 COMPUTER PROGRAM(S)
(Title(s) and language(s))

12 RELEASE LIMITATIONS (of the document):

Approved for Public Release

12.0	OVERSEAS	NO		P.R.		A		B		C		D		E	

Security classification of this page: UNCLASSIFIED

Security classification of this page: UNCLASSIFIED

13 ANNOUNCEMENT LIMITATIONS (of the information on these pages):

No limitation.

14 DESCRIPTORS:

a. EJC Thesaurus Terms

Data recording
Data processing
Data links
Data storage
Magnetic disks
Magnetic tapes
Computer storage devices
Magnetic storage

b. Non-Thesaurus Terms

15 COSATI CODES:

0902

1403

16 LIBRARY LOCATION CODES (for libraries listed in the distribution):

SW SR SD AACA

17 SUMMARY OR ABSTRACT:
(if this is security classified, the announcement of this report will be similarly classified)

A data migration scheme has been in operation on the DRCS Central Computer for approximately two years. During this period tens of thousands of datasets have been migrated to magnetic tape to make the best use of disk storage space. The number of dataset retrievals, although still well within the processing capabilities of the IBM 370 computer system, has predictably grown. This report describes an innovation designed to lessen the impact of retrieval requests on the system resources, especially in the operator intensive area of tape handling.

Security classification of this page: UNCLASSIFIED