AR-001-455

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

ELECTRONICS RESEARCH LABORATORY

TECHNICAL REPORT

ERL-0055-TR

MANAGEMENT PROCEDURES FOR CONTROLLING DATA STORAGE ON THE IBM SYSTEM 370 COMPUTER AT DRCS

J.C. Gwatking
R.W. Collier

S U M M A R Y

This document is one of a series describing various aspects of data management on the DRCS Central Computer. It includes methods for controlling user datasets on both disk and magnetic tape. Where appropriate the operating procedures and supporting software are also outlined.

Approved for Public Release

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

TABLE OF CONTENTS

1. INTRODUCTION

Automated data management techniques are essential in an installation such as the DRCS Computing Centre. Without them the problems of controlling the activities and providing service to the many hundreds of users would require the full time attention of several system programmers.

The DRCS data migration and dataset backup and recovery schemes are described in detail in references 1, 2 and 3. This document discusses the DRCS approach to the remaining aspects of dataset management. Sections 2 to 5 are concerned with disk dataset control, Section 6 with tape datasets, Section 7 with dataset access reporting and finally Section 8 describes the special case of the recording and storage of the System Management Facilities (SMF) data. Details of software routines written to implement these procedures are contained in Appendix I.

Table 1 presents a housekeeping schedule showing the frequency with which each maintenance task is performed and their dependence on other tasks to provide input data.

2. THE CLEANUP PROCEDURE

With the high use of user disk volumes a large number of superfluous datasets are created. A program, which is run daily, removes as much of this material as possible and generally ensures that the standards for the management and naming of permanent datasets, as described in the Users Guide(ref.15), are adhered to. The CLEANUP procedure is as follows.

(i) Any uncatalogued datasets are deleted. This follows from the basic rule of the Computing Centre that all permanent datasets must be catalogued.

(ii) If a catalogue entry exists for a non-existent dataset, the catalogue entry is deleted. This situation is normally the result of a user error and the removal of the catalogue entry is essential to ensure that the catalogues are not filled with superfluous information.

(iii) Any dataset that does not have a valid userid as its first qualifying name is deleted. This again follows from a basic rule of the Computing Centre.

(iv) All datasets with a last qualifying name of OBJ, LIST, OUTLIST, TESTLIST, LINKLIST or LOADLIST are deleted. TSO often creates datasets of this form unbeknown to the user and they are classed as temporary by the installation. Users are warned not to create permanent datasets with any of these last qualifying names.

(v) Datasets of the form

userid.xxxxxxxx.D7xxxx.Txxxxxxx.xxxxxxxx
and
userid.MERGE.D7xxxx.Txxxxxxx

are deleted. These are temporary datasets whose names are generated by the Operating System and they are usually deleted when a job terminates. However, under some abnormal circumstances they may remain on disk.

A version of the utility program IEHPROGM(ref.5) with authorization to bypass Operating System password checks is used to delete protected datasets.

The CLEANUP procedure also endeavours to keep the work areas on disk clean. These are areas on designated non-user volumes intended as scratch space only,

available to users for the duration of a job. Any residual datasets found in these areas are deleted.

An added feature of the CLEANUP procedure is a summary of the space each user occupies on each disk volume. This information is useful in investigating the day to day fluctuations in available free space on user packs.

It is essential that the BACKUP procedure(ref.3) is run prior to the CLEANUP procedure so that any of the datasets deleted can be recovered if required. It is also essential that the procedures WARNLIST and ARCHLIST(ref.1) and COMPRESS are run after the CLEANUP procedure because they are dependent on all user datasets being catalogued.

3. THE COMPRESS PROCEDURE

When a member of a partitioned dataset is replaced the new version is written following the existing information. The space occupied by the older version becomes unusable, imbedded within the dataset. Unless this is reclaimed the size of the dataset will continually grow. The IBM Utility program IEBCOPY(ref.5) provides a reclaiming function. It "compresses" the dataset, by a process of reorganizing the members.

The DRCS Data Migration Scheme monitors the use of partitioned datasets, registering when they were last updated(ref.1). From this information a software procedure, run weekly, determines which ones have been updated during the week. It generates several batch jobs to compress these datasets and reclaim the free space. There is one job per user disk volume. Optional input to the procedure identifies datasets that should not be compressed. A version of IEBCOPY which has authorization to bypass Operating System password checking is used to compress protected datasets(ref.6).

This automatic compression of partitioned datasets relieves users of the task, except in unusual cases where compression is required more often than once a week.

In addition, if the dataset has provision for secondary extents, all unused space is released, down to the unit of allocation (track or cylinder). This process is essential in helping to achieve maximum use of available disk space.

An extra benefit of regular dataset compression is the detection of irregularities in the organization of some datasets. The main cause is the use of a partitioned dataset for sequential output processing, for instance by omitting a member name in the JCL statement. This causes the directory of members, stored at the beginning of the dataset, to be overwritten, rendering it unusable. Unfortunately there is no warning or indication when this happens, and the error might otherwise go undetected for some time, decreasing the chance of recovering the data.

4. DISK VOLUME REORGANIZATION

The contents of user disk volumes are quite volatile, with many creations and deletions occurring each day. Under these conditions the available free space soon becomes very fragmented, unsuitable for datasets with large primary allocations or for temporary work files.

The DUMPRSTR program(ref.7) was acquired by DRCS to help relieve the problem. It is a data transfer program, operating on whole volumes or just selected datasets, and has three modes of operation : disk-to-disk, disk-to-tape and tape-to-disk. When copying a complete disk volume or restoring one from magnetic tape DUMPRSTR will optionally reorganize the data to create a volume with minimal fragmentation.

This ability to reorganize datasets when copying them forms the basis of the DRCS procedures to combat fragmentation. The existing extents of each input dataset are consolidated by DUMPRSTR to form a single-extent output

dataset, which still has 15 secondary extents available if required. In addition, two of the DUMPRSTR modules, COPYVOL (disk-to-disk) and DISKTAPE (disk-to-tape) have been modified at DRCS to release all unused units of space allocation (tracks or cylinders) from the new primary extent before it is allocated on the output volume. Therefore, besides greatly reducing fragmentation, DUMPRSTR also has the ability to reclaim waste space for subsequent reuse. The exceptions to the reorganization are ISAM, VSAM and unmovable datasets, all of which retain the identical locations and extents that they occupied on the original disk volume.

An operational procedure using DUMPRSTR was first developed for the IBM 3330-1 disks that were in use from late 1975 till late 1977. This was later replaced by a new procedure for handling the IBM 3350 drives that were installed during 1977 and 1978.

The two procedures are described in the following subsections.

4.1 IBM 3330 volumes

Six 3330 drives were allocated to general user datasets during the first two years of operation and fragmentation on them soon became a problem, prompting the acquisition of DUMPRSTR.

Interchangeable volumes, like those used on the 3330 drives, are well suited to the disk-to-disk function of DUMPRSTR. The procedure used at DRCS to reorganize a 3330 volume is therefore based on this feature and is summarized below.

(a) If the volume contains paging datasets, then an IPL must be performed to deactivate them(ref.8).

(b) Analyze a spare 3330 volume on a spare drive. The Volume Table of Contents(VTOC) may be placed in a different location and be of a different size from that on the volume about to be copied. The only criterion is that it must not occupy a location required by an unmovable dataset on the input volume.

(c) Copy the input volume to the spare 3330, using the disk-to-disk function of DUMPRSTR.

(d) Relabel the spare volume with the serial number of the original, and remove the latter from the system. This then provides an easily recovered backup.

(e) Swap the new volume to the drive previously occupied by the original.

Using this technique several 3330 volumes can be reorganized in parallel. The whole process generally takes about 20 min (excluding IPL time) and is initiated and controlled by software.

4.2 IBM 3350 volumes

With the introduction of IBM 3350 drives for storing user data the reorganization procedure had to be changed. The technique of copying disk-to-disk is unsuitable for a fixed device. Firstly there would not normally be a spare 3350 drive available to receive the copy. Even if there were, the reorganized volume could not be moved back to the original drive, so that volumes would tend to migrate around the available addresses. This is undesirable for operational and performance reasons.

Therefore, disk-to-disk copying was abandoned in favour of a disk-to-tape dump, followed by a tape-to-disk restore to the same device, incorporating the same reorganization and space-saving benefits.

However, there are two disadvantages inherent in this technique. Firstly, the elapsed time for the process is obviously greater, since the data is copied twice. Secondly, the only complete backup of the data is on tape, since the second stage of the procedure involves restoring to the original device. This is only a problem if permanent tape I/O errors occur during the restore. One might then be left with a partially restored volume and a partially unreadable tape.

Under these circumstances the only method available to recover the data is to use the backup system(ref.3) to restore the volume to its state at backup time that evening. Normally, there will have been few changes since then.

The steps required to reorganize a 3350 volume are outlined below.

(a) If the volume contains paging datasets they must be deactivated, by performing an IPL.

(b) Use DUMPRSTR to dump the contents of the disk volume to tape. The dump will contain all control information that is necessary to restore the volume, in either reorganized format or as a direct copy.

(c) Use DUMPRSTR to restore the tape contents to the same disk unit, selecting the reorganization option.

As is the case with 3330 volumes, multiple 3350's can be reorganized in parallel, under software control. Excluding IPL's the elapsed time is about 30 min. Although there has been little experience with 3350's so far, it is expected that they will have to be reorganized with about the same frequency as 3330's, every three or four weeks.

5. VOLUME STATUS REPORT

Every evening while the disk backups are being performed, and before CLEANUP is run, the Volume Status Report is produced. This gives a list of all datasets residing on each disk volume that is in normal use. For each system pack, the dataset names are printed in alphabetical order. For the user packs, one composite list is printed giving the name of each dataset in alphabetical order, with the pack on which it resides.

The status reports are retained for at least a week and are useful for investigating the disappearance of a dataset, for example.

6. MAGNETIC TAPE MANAGEMENT

The Operations Section of the Computing Centre maintains a pool of 9-track standard labelled magnetic tapes that are available for general use. Only in very special circumstances, and by prior arrangement, can a user retain data on any other magnetic tape. Even then, such tapes are not stored in the computer room but are solely the responsibility of their owners. These tapes must be presented to the Operations Section before being required by a batch job. They are returned after the job completes, unless arrangements have been made to leave them in the computer room for a short period of heavy usage.

Conversely, tapes from the general pool cannot be removed from the computer room. Special tapes must be used if this is a requirement.

To obtain the use of a tape from the general pool for output purposes, the user simply omits the VOLUME parameter from the JCL statement. Operators will automatically assign the next available spool from a rack of scratch tapes. The user's job might use the tape only for temporary storage, or might write one or more permanent datasets to it. In the latter case the datasets must be catalogued. When the job completes the tape will be unloaded and cannot be

written to again until it has been erased (see Section 6.3). The aim of this restriction is to protect tape datasets from being overwritten by other users. The resultant usage inefficiency is the penalty for data integrity. However, a procedure does exist to lessen this inefficiency in some cases (see Section 6.1).

To retrieve permanent data from a tape, the user only needs to supply the dataset name. Since it must be catalogued, the Operating System determines the volume serial number and issues the mount request. Users should never need to know which volumes contain their tape datasets. When the dataset is no longer required, it should be uncatalogued. Tape volumes that no longer contain any catalogued datasets are periodically selected for erasure and reuse (see Section 6.3).

The data stored on the labelled tapes needs regular review in order to keep the pool at a manageable size. The review procedures carried out by the Operations Section are described in the following subsections.

6.1 Transcription

Some users require long term retention of a large number of moderately sized datasets. For maximum efficiency, several of these should be stored on each tape. However, the rules for tape usage outlined above preclude writing to a volume that already contains permanent datasets created by another job. This means that each dataset must be written to a new volume.

To reclaim some of the volumes occupied by these long term datasets, the Operations Section uses a transcription procedure that transcribes selected groups of datasets to a single tape volume. Selection is made with the aid of a list of all catalogued tape datasets, sorted in both dataset name and tape volume sequence. However, the owners are always consulted before proceeding, in case they intend to use the datasets in a way which precludes them from being on the same tape volume.

6.2 Tape usage report

Because of the operational overheads associated with tape storage and handling, users are encouraged to use tapes only for large datasets, or ones of moderate size and infrequent use, and to release them when no longer required. This is the purpose of a report that each owner of tape datasets receives approximately every three months(ref.4). The information contained in the report includes the names of the user's tape datasets and an indication of their size and frequency of use.

The second function of the report generation program is to identify small tape-resident datasets which will be subsequently transferred to disk, thereby freeing more volumes. The report also lists any frequently used datasets of moderate size that would be better on disk.

The procedure uses SMF data (see Section 8) to determine dataset size, creation date and frequency of access.

6.3 The erase process

Each labelled magnetic tape in the general pool is serially numbered and has a corresponding slot in a bank of tape racks. The tapes are always stored here with their write-enable rings removed, and are available for input processing when the data they contain is required.

When the operators need more scratch tapes, they have a procedure to identify which volumes in the general pool are available. The procedure searches the Operating System catalogues to determine which volumes contain catalogued datasets. The remainder either

(a) contain unwanted temporary or permanent datasets,
(b) were in the scratch rack when the procedure was started, or
(c) were mounted on a unit for output processing when the procedure was

started.

All but those in the last category are therefore available for re-use, although the procedure itself cannot determine which ones to exclude. Instead it prompts the operator for the information, as well as for the serial numbers of those volumes already in the scratch pool. The procedure then generates batch jobs to erase the rest.

Next, the operators remove the selected volumes from the general rack, insert the write-enable rings and mount the tapes when required for erasure. Those that contain password protected datasets are erased fully, to the end-of-volume indicator. This is required for security reasons but is often wasteful as datasets may occupy only a small portion of the tape. For economy, tapes containing unprotected datasets (the majority) are merely overwritten with 100 end-of-file marks directly after the volume label. This is considered sufficient protection to prevent accidental reading of the previous contents.

Finally the erased tapes are placed in a separate scratch rack where they are available for re-use as output tapes.

The operators are responsible for noting which tapes were mounted on units when the selection procedure was initiated and for ensuring that these are not erased (since the data on them may have been subsequently catalogued). There have been several instances when the loss of data on magnetic tape has been directly attributable to this being overlooked, so the selection procedure is now run out of normal operating hours whenever possible.

7. DATASET ACCESS REPORTING

Because of the strict security requirements at DRCS, user and system datasets must be adequately protected. Password protection provides a means of limiting dataset access to authorized users. However, it is often cumbersome to use and has not been widely accepted. The result is that most datasets on the system are unprotected and can be accessed, updated and even deleted by any user.

To partially protect users from the dangers of their natural reluctance to use passwords, the DRCS Computing Centre monitors all dataset accesses by other than the registered owner. Approximately once a fortnight each user receives a report listing all foreign accesses to his datasets. The report includes the names of the datasets that were accessed, the type of access, the perpetrator, and the frequency and dates of access. Users may elect to exempt certain authorized co-workers from their reports(ref.9). If they do, then the exception list will be printed at the top of the report for validation. For this reason the report is issued even if all accesses are suppressed from the list. However, if there were no accesses by any other user (exempt or not) during the period, the report is not issued.

Each individual is responsible for studying the contents of his access report and ascertaining the reasons for each entry. Security personnel should be informed of any violations that cannot be satisfactorily explained. The Computing Centre may be able to help by reconstructing the circumstances surrounding the access, using SMF information. The dataset backup scheme(ref.3) might also be needed to recover information accidentally or maliciously destroyed by another user.

Besides providing important after-the-event notification, the access reporting procedure will also discourage deliberate security violations.

Another aspect of access reporting is to trap the "browser" - the person who may read many harmless looking datasets, but only one or two from each user. Each may disregard the access as unimportant or accept the person's reason for it. However, this is a dangerous practice that must be discouraged. For this reason the monthly computer usage report produced for Computing Services Group management includes a list of possible offenders for subsequent evaluation.

8. SMF DATA STORAGE

System Management Facilities (SMF) is a component of the Operating System that gathers and records information on system usage, primarily for accounting purposes(ref.10). However, at DRCS SMF data is used extensively for other purposes, particularly dataset management. The data migration scheme(ref.1,2), backup scheme(ref.3) and access reporting scheme (Section 7) are important examples. There have also been many occasions when SMF data has been invaluable in ascertaining the exact sequence of events relating to a problem being investigated. Sometimes the data required may be weeks or even months old.

In view of the importance of SMF data, DRCS retains it indefinitely on magnetic tape. The Operating System records the data alternately on two disk datasets, SYS1.MANX or SYS1.MANY. When the current dataset becomes full the system automatically switches to the other and requests the operator to empty the full one. The procedure used at DRCS is to dump the dataset contents to tape and then re-initialize it.

A three level generation data group, OPS.SMFDATA, is used by the dump procedure. The latest generation contains all available data for the current month. The procedure first copies all records from the current generation to a new tape and then adds the information from the disk dataset (SYS1.MANX or SYS1.MANY) to the end. This tape then becomes the latest generation. A simpler approach would be to just add records to the end of a tape. Although this would reduce the overheads of the procedure considerably, there are two disadvantages. Firstly, the DRCS tape management standards prohibit writing to a volume that already contains permanent data. The SMF dump tape would therefore have to be treated as an exception, which is a situation that should be avoided. Secondly, the data would not be adequately protected. Should the tape be accidentally overwritten or become unreadable, up to one month's information could be lost. However, with the generation approach, the most that can be lost is the data from the last disk dataset dumped, assuming that the previous generation is still readable.

There are also two by-products from the SMF dump procedure. Firstly, the program extracts dataset update records and transfers them to a further dataset (SYS1.BACKUP.SMF) for use by the DRCS backup scheme(ref.3). Secondly, costing information is written on the dataset OPS.COST.DATA for processing by ADP Section.

During the first week of each month, the previous month's data is removed from the generation data group OPS.SMFDATA. A program reads the last generation and transfers those records for the previous month to a new generation of OPS.SMFMONTH and those for the current month to the next generation of OPS.SMFDATA. OPS.SMFMONTH is an open-ended generation data group. The current generation always contains the data for the previous month (after the split has taken place), the next to last generation contains the data for the month before that, and so on. This SMF splitting process is part of the data migration procedure WARNLIST(ref.1).

9. SUMMARY

This document has presented an outline of some of the operational procedures used at DRCS for dataset management. In general the procedures described here, and in references 1, 2 and 3, work well and require little supervision, using only the part-time services of two system programmers. Without such automated procedures the drain on system programming staff would be far greater.

The most time consuming human involvement remaining in dataset management is in balancing free space levels on the user disks. Imbalances often occur as the result of heavy creation activity by some users. A program is being developed to correct these imbalances automatically, by selecting and moving suitable datasets between volumes. When it becomes operational, programmer

effort will be even further reduced.

10. ACKNOWLEDGEMENTS

Several Computing Services Group personnel other than the authors contributed to the programming and, in some cases, design of procedures described in this report. They include J.L. Roughan, T.J. Galvin, P.J. Willcocks and P.A. Rugless. The authors wish to thank them for their efforts.

REFERENCES

No.	Author	Title
1	Gwatking, J.C.	'A Data Migration Scheme for the 370/168 Computer at W.R.E.' W.R.E. Technical Report 1870(A)
2	Gwatking, J.C.	'An Efficient Application of Disk Storage to the W.R.E. Data Migration Scheme'. Technical Report ERL-0009-TR
3	Gwatking, J.C.	'A Selective Dataset Backup Scheme for the W.R.E. Central Computer'. Technical Report ERL-0021-TR
4	Rugless, P.	'Management of Catalogued Tape Datasets on the IBM 370/168 Computer at DRCS' ERL Technical Report (Draft)
5	IBM Corporation	'OS/VS Utilities' GC35-0005
6	IBM Corporation	'OS/VS2 System Programming Library: Job Management' GC28-0627-1
7	North Carolina State University Systems Services	'DUMPRSTR, Dumping and Restoring Direct Access Volumes'. Library Services Series, Document No. LS-290-1
8	IBM Corporation	'OS/VS2 System Programming Library: Initialization and Tuning Guide'. GC28-0755-0

9	Computing Services Group	'Dataset Access Reports' Computer Bulletin No. 86, April 1977.
10	IBM Corporation	'OS/VS2 System Programming Library: System Management Facilities (SMF)'. GC28-0754-0
11	IBM Corporation	'OS/VS Virtual Storage Access Method(VSAM) Options for Advanced Applications'. GC26-3819-2
12	IBM Corporation	'OS/VS2 Independent Component: Catalog Management Logic'. SY26-3847-0
13	IBM Corporation	'OS/VS2 Access Method Services'. GC26-3841-1
14	IBM Corporation	'OS/VS Data Management Macro Instructions'. GC26-3793-4
15	Roughan, J.L.	'W.R.E. Computing Centre Users Guide'. W.R.E. Manual 1492(AP)

APPENDIX I

PROGRAM MODULES

This appendix outlines the type, purpose and external data requirements of program modules associated with the procedures described in the main body of the report. The tape usage report programs (see Section 2.3) are exceptions. They are fully documented in reference 4.

Several routines from the data migration software(ref.1) and backup software(ref.3) will also be mentioned below. These will be identified by the letters DM or BU in brackets after the module name.

The PL/I attributes of arguments are indicated for subprograms.

I.1 ACCESS

type - PL/I, main program

This program performs the first stage of the dataset access reporting procedure. It examines all SMF records that represent a dataset access. These are types 14, 15, 17, 18, 62, 63, 67, 68, 128, 129, 130, 132, 133 and 140(ref.1,10). When it detects an access to a dataset by someone other than the owner or userid OPS (the Operations Section userid, which is exempted) the program generates a record to file name ACCREC for processing by program ACCREP (see below).

The SMF data scanned may be limited to a specific day range if required. Similarly the program can be made to report on accesses to selected users' datasets only.

Since read accesses to SYS1 datasets are performed by most users they are only reported for datasets which are password protected. All write accesses are still recorded, however.

Input formats

(a) File SMF
The program reads the SMF data for the period of interest from this file.

(b) File SYSPROT
The names of password protected SYS1 datasets are obtained from this file, one per 80-byte record, in positions 1 to 44.

(c) File SELECT
Three character dataset name prefixes may be supplied in this file to limit the report to accesses to these users' datasets. There is one per 80-byte record, in the first 3 positions.

(d) File SYSIN
The day range of interest may be specified in this file. The program reads the 80-byte record using a PL/I GET DATA statement. The variables TO and FROM define dates in Julian format, identifying the range. If not specified all SMF data from file SMF will be used to generate the report.

Output formats

(a) File ACCREC
The program writes access records to this file for processing by program ACCREP. The records are each 80-bytes long. The first indicates the day range covered by the report. The begin and end dates, in Julian form, occupy positions 2 to 6 and 8 to 12 respectively. The format of the remaining records is as follows.

Offset	Size	Field
0	54	Dataset name plus member name or generation member, in brackets, if relevant.
54	2	Access type - numeric display format (see below).
56	8	Accessing job name.
64	4	Time of day, in hundredths of a second, of the access.
68	4	Date of the access - Julian format, packed decimal.
72	8	Unused (blank).

The access type is a number from 1 to 14, indicating the type of activity performed on the dataset. The meanings are given below, together with the SMF record types and special conditions used to identify each.

(1) The dataset was archived - SMF record type 128.

(2) The dataset was backed up to the archives - SMF record type 132.

(3) The dataset was retrieved from the archives - SMF record type 133.

(4) The dataset was reloaded from the archives - SMF record type 129.

(5) The dataset was deleted from the archives - SMF record type 130.

(6) The dataset was read - SMF record types 14 and 140.

(7) The dataset is VSAM and may have been read or written - SMF record type 62.

(8) The dataset was created - SMF record type 15 for non-VSAM and 63 for VSAM, both with a disposition of NEW.

(9) The dataset was created by renaming another dataset - SMF record type 18 for non-VSAM, 68 for VSAM, 135 for an archived dataset.

(10) The dataset was opened for update - SMF record type 15 with other than NEW disposition.

(11) The dataset was deleted - SMF record type 17 for non-VSAM, and type 67 for VSAM, with both the delete and uncatalogue indicators set.

(12) The dataset was deleted by changing its name - SMF record type 18 for non-VSAM, 68 for VSAM, 136 for an archived dataset.

(13) The dataset was uncatalogued - SMF record type 67 with only the uncatalogue indicator set.

(14) The dataset is VSAM and its catalogue entry was altered - SMF record type 63 with the alter indicator set.

(b) Since the program invokes SORT via entry point PLISRTA the standard DD cards should be present. These include SYSOUT, SORTLIB and SORTWKOn. In addition the SORTIN and SORTOUT statements should reference the same dataset as file ACCREC.

I.2 ACCREP

type - PL/I, main program
calls - GETXEPT, GETUSER (DM), JULIAN (DM), ADDRESS (DM)

This program reads the information generated by ACCESS and produces a report for each user whose datasets have been accessed. The report lists the dataset names, the userids that accessed them, the type of accesses, their frequency and the day or day range on which they occurred.

Any user may elect to suppress from his report either all accesses or just "read" type accesses by certain other users(ref.9). The program prints such specifications at the head of the user's report, for verification. The user will still receive a null report if all accesses to his datasets were by exempt users.

Accesses to datasets with invalid names are listed on a special report for system programmer information.

Input formats

(a) File ACCREC
This is the dataset generated by program ACCESS - see I.1 above.

(b) File MESSAGE
Up to ten 80-byte records may be supplied in this file. It will be interpreted as general textual information and printed near the top of each user report.

(c) File EXCEPT - dataset SYS1.EXCEPT.DATA
This partitioned dataset contains one member for each user who has specified access exceptions(ref.9). The member name is the userid. The program interprets the information in these members to decide which accesses to suppress from the reports.

(d) File SYSIN
The program uses a PL/I GET DATA statement to read a single variable from this 80-byte record. The variable, XEPT, is a single character with a value of either 'Y' (the default) or 'N', indicating whether the exceptions in file EXCEPT are to be used this run or not, respectively. The exceptions apply to all users or none.

Output formats

(a) File USEREPT
The individual user reports are written to this file.

(b) File SYSPRINT

The special system programmer report is written to this file.

I.3 CATCOPY

type - PL/I, main program, must be authorized

Program CATCOPY opens and reads the low key range of a VSAM catalogue as a normal VSAM dataset(ref.11). It suppresses some record types and writes the remainder to an output dataset. The types selected for inclusion are non-VSAM (A), generation data group base (B), cluster (C), data component of a cluster (D), index component of a cluster (I), user catalogue (U), alternate index (G), path (R) and alias (X).

Input formats

(a) File CAT
The VSAM catalogue is read through this file name. See reference 12 for full details of formats of the various record types.

(b) File STEPCAT
If the catalogue defined by file CAT is a user catalogue it must also be specified on a STEPCAT DD statement.

Output formats

(a) File CATOUT
The catalogue records selected by CATCOPY are copied to this file as variable length records, with a maximum data size of 505 bytes. The information will later be read and interpreted by subroutine CATLGRD.

(b) File SYSPRINT
In each input record there is a field that contains the number of bytes of information in the record (this defines the length of the output record). Program CATCOPY checks the value, and if it is less than 55 (the minimum allowed) or greater than 505 (the maximum) writes an error message to file SYSPRINT and terminates. The same action is taken should an I/O error occur.

I.4 CATLGRD

type - PL/I, subprogram
called from - TAPEVOL, CLEANUP, SHIFTID(DM), DSUPDTE(BU)
additional entry point - CATLGIN

Subroutine CATLGRD reads a sequential dataset produced by program CATCOPY containing a copy of one or more VSAM catalogues. The file used to perform the accesses is CATLG and any program that invokes CATLGRD must include a DD statement of this name in its JCL. At each call the details of the next catalogue entry are returned. The fixed information includes entry name (44 bytes) and its type (A,B,C,D,I,G,R,U or X - see I.3). In addition certain extra variable information is provided, whose length and meaning depend on the entry type (see below).

Entry point CATLGIN returns the same information as CATLGRD. However, the catalogue records are passed to CATLGIN as an argument, not through a file.

arguments

catalogue record - CHARACTER(512) - entry point CATLGIN only
control interval number of record - CHARACTER(3)

record type - CHARACTER(1)
entry name - CHARACTER(44)
variable information - CHARACTER(500)
length of variable information - FIXED BINARY(15)
end-of-file indicator - CHARACTER(1) - entry point CATLGRD only. This will be blank except when the end-of-file is reached on file CATLG.

The format of the variable information is given below for each valid record type.

(1) Type A (non-VSAM)
length = 16 x n, where n is the number of volumes the dataset occupies. Each 16-byte volume field has the following format.

(a) device type - 4 bytes
(b) volume serial number - 6 bytes
(c) file sequence number - 2 bytes, binary
(d) volume flag - 1 byte
(e) TTR of format-1 DSCB if a DASD device - 3 bytes.

(2) Type B (GDG base)
length = 2. The format is

(a) number of concurrent generations - 1 byte, binary
(b) GDG attributes - 1 byte.

(3) Type C (cluster)
length = 4 x n, where n is the number of entries having an association with this one. The format of each field is

(a) association type code - 1 byte (I for index, D for data, G for alternate index, R for a path).
(b) control interval number of the association - 3 bytes, binary.

(4) Type D (data)
length = 4 x m + 13 x n, where m is the number of entries having an association with the component and n is the number of volumes the dataset occupies. The association set of fields begin at byte 1 of the variable information area and have the same format as those for the cluster entry type described above. The possible association type codes are C for cluster, G for alternate index and Y if the component's cluster has an alternate index that is part of the upgrade set(ref.12). The offset of the volume set of fields from the beginning of the 500 byte area is contained in bytes 499 and 500, as a binary number. The format of each volume field is

(a) device type - 4 bytes
(b) volume serial number - 6 bytes
(c) file sequence number - 2 bytes, binary
(d) volume flag - 1 byte

(5) Type I (index)
The format and meaning of the fields in index entries are the same as for data entries(type D). The possible association type codes are C for cluster and G for alternate index.

(6) Type G (alternate index)
The format and meaning of the fields in alternate indices are identical to those of cluster entries(type C). The possible association type codes are I for index, D for data, C for cluster and R for path.

(7) Type R (path)
Similarly the fields of a path are identical to those of a cluster(type C). The association type codes that may appear are C, G, I and D.

(8) Type X (alias)
Again the fields in the variable data for an alias entry have the same meaning and format as for a cluster(type C). However, the types of possible associated entries are A for non-VSAM and U for a user catalogue.

(9) Type U (user catalogue)
No variable information is returned for this type of record.

I.5 CATTAPE

type - PL/I, main program
calls - TAPEVOL

This program determines all catalogued tape datasets and their volume serial numbers, from catalogue copies created by program CATCOPY. The information is sorted and printed in both dataset name and volume serial number sequence.

Input format

(a) File CATLG
The catalogue copies are read from this file.

Output formats

(a) File SYSPRINT
The sorted reports are written to this file.

(b) Since the program invokes SORT via entry point PLISRTD, standard DD cards for SYSOUT, SORTLIB and SORTWKOn are required.

I.6 CLEANUP

type - PL/I, main program
calls - DEVSIZE (BU), DEVFREE (BU), SECSPCE (DM), CATLGRD

This program performs the major task of creating control statements to delete invalid user disk datasets and catalogue entries from the system. In addition it reports on various other unusual conditions for system programmers to investigate. CLEANUP also displays the disk volume free space levels that will exist after the dataset deletions have taken place. Note that tape datasets are not considered at all by the program.

The primary input to CLEANUP is a copy of all VSAM catalogues, as produced by program CATCOPY, plus a dataset containing the format-1 DSCB's of all datasets on the participating disk volumes, in dataset name sequence. The latter is produced by program VTOCDSN (DM).

Input formats

(a) PARM field

LIST
NOLIST

LIST indicates that the program is to produce a list of all

datasets it considers to be exceptions. These include dummy generation data group labels, plus exempt datasets (see below). The default is NOLIST.

(b) File CATLG
The catalogue copies produced by program CATCOPY are read from this file.

(c) File VTOC
The sorted format-1 DSCB's are contained in this file.

(d) File DISKS
The serial numbers of participating disk volumes are specified in the first 6 bytes of each 80-byte record. They include all user volumes but not system ones. No maintenance will be performed on any volume not in the list.

Output formats

(a) File SCRSET
The program writes IEHPROGM control statements to delete and/or uncatalogue invalid or temporary data sets to this file. A subsequent step in the CLEANUP catalogued procedure will process these control statements.

(b) File CNTRLS
Similarly IDCAMS control statements to uncatalogue datasets in the wrong catalogue are written to this file, to be processed by a later step in the procedure.

(c) File TEMP
When a list of dataset exceptions is requested in the PARM field this file is used for temporary storage until the information is required for printing.

(d) File SYSPRINT
All reports and statistics are written to file SYSPRINT, including the contents of files CNTRLS, TEMP and SCRSET.

(e) The program invokes SORT via entry point PLISRTA to sort the information it has acquired about each user catalogue into ascending name sequence. The standard DD cards for SYSOUT, SORTLIB and SORTWK0n are therefore required. In addition files SORTIN and SORTOUT should identify temporary datasets with fixed-length 57-byte records.

I.7 CMPRESS

type - PL/I, main program
calls - DSCB1 (DM), ENQDEQ (DM), JULIAN (DM), LASTACC (DM)

This program builds a job stream to compress all partitioned datasets that have been updated during the past week. The data migration scheme dataset SYS1.DATASET.LASTACC(ref.1) has a record for each user disk dataset, which includes a field indicating how recently it has been updated. CMPRESS uses these records to identify datasets that need compression.

The program reads the VTOC of each participating disk volume in turn and builds IEBCOPY jobsteps to compress the relevant datasets. There is one job for each disk volume, with jobnames OPSCMP01, OPSCMP02 etc. The JCL is also coded to release unused space from datasets that have

provision for secondary allocations. Password protected datasets are compressed by a special privileged version of IEBCOPY which can bypass the Operating System password protection checks.

The program also accepts a list of exception datasets which should not be compressed, regardless of their update activity.

Another special function is that all partitioned datasets on volumes VSAMXA and VSAMXB are compressed, unless they appear in the list mentioned above.

Input formats

(a) File CATLG
This file contains the catalogue copies produced by program CATCOPY.

(b) File DISKS
The serial numbers of participating disk volumes are specified in the first 6 bytes of each 80-byte record.

(c) File SYSIN
The names of datasets to be excluded from compression are supplied through file SYSIN, one per card image.

Output formats

(a) File CMPRS
The jobstream to perform the compress operations is written to this file. There will be one job for each disk volume. The file is directed to the internal reader.

(b) File COPY
This file will contain the same as file CMPRS, but is directed to a permanent dataset, to be used for restart purposes.

I.8 COPY

type - PL/I, main program

Program COPY generates a job to perform the tape transcription function described in Section 2.2. It accepts one or more tape dataset names as input and constructs IEBGENER jobsteps to copy each one to consecutive files on a single output tape volume. The last step in the generated job executes only if all previous steps return a zero condition code. It uses the IDCAMS utility program(ref.13) to uncatalogue the transcribed datasets and recatalogue them in their new locations.

Input format

(a) File SYSIN
The first card image contains the volume serial number of the output tape in positions 1 to 6. Remaining records each contain a dataset name beginning in byte 7. These are the datasets to be transcribed.

Output formats

(a) File CPYCNTL
The job to perform the transcription is written to this file, which is directed to the internal reader.

(b) File SYSPRINT

If the program detects anomalies in the input records from file SYSIN it writes diagnostic messages to SYSPRINT.

I.9 ERASE

type - assembler, main program

This program performs the actual tape erasure. It reads and checks the tape volume label, then the dataset label. From this it determines whether the first dataset currently on the tape is password protected. If so the entire volume is erased using a special channel command available on model 3420 tape drives. Otherwise only the first 300 inches or so are overwritten by 100 file marks. In both cases the tape is first backspaced to a point just after the volume label, where a file mark is written. Note that if a tape contains more than one dataset they must all have the same password protection status. Therefore the status of the first dataset on the volume defines that of any others.

If the program detects an error of any kind, such as an incorrect tape volume or an I/O error, it aborts, displaying an appropriate message on the operator console.

Input format

(a) PARM field
The 6-byte volume serial no. is passed to the program in the PARM field.

Output formats

(a) File TAPE
The tape volume is allocated to the program via this DD statement. Since ERASE reads and processes the volume label as normal data the BLP(Bypass Label Processing) option must be specified in the LABEL parameter.

(b) Completion messages, both successful or otherwise, are displayed on the operator console.

I.10 GETXEPT

type - assembler, subprogram
called from - ACCREP, BACKOPT (BU)

Subroutine GETXEPT reads information from a member of a partitioned dataset and returns it to the calling program. The data must consist of 80-byte records, which may be blocked. Only the first 30 logical records will be read and returned.

GETXEPT uses the BPAM macros OPEN, DCB, FIND, READ and CHECK(ref.14) to locate and access the members in the dataset associated with file EXCEPT (see program ACCREP in Appendix I.2).

arguments

member name - CHARACTER(8)
number of records returned - FIXED BINARY(15) - the maximum value is 30.
If it is zero, then either the member could not be found or it was empty.
storage area containing the records - (30) CHARACTER(80).

I.11 NEWMTH

type - PL/I, main program

This program is used by the data migration procedure WARNLIST(ref.1) to determine if the current day is in the first week of a month or not. If it is, WARNLIST splits the SMF data (see Section 8). The program returns a condition code of 1 if this is the first week of a month, 0 otherwise. This code can be interrogated by subsequent job steps to determine which processing should be performed.

I.12 SETCODE

type - PL/I, main program

This is a simple program that returns a condition code equal to a value passed as a parameter. It is useful for checking complex combinations of condition codes and is used in many procedures.

Input format

(a) PARM field
The required condition code value is specified in this field. It must be numeric.

I.13 SMFCOPY

type - PL/I, main program

This is the principal program in the SMF dump procedure (Section 8). It copies all data from the current to a new generation of OPS.SMFDATA and then adds new records from SYS1.MANX or SYS1.MANY to the end of the new dataset. In addition SMFCOPY generates costing data from the new type 5 (job completion) and type 35 (TSO session completion) records. It also copies any dataset update records (types 15,17,18,63 and 64) from SYS1.MANX or SYS1.MANY to a separate dataset for use by the backup scheme(ref.2).

Input formats

(a) File SMFIN
This is the current generation of OPS.SMFDATA

(b) File SMFNEW
This file is connected to either SYS1.MANX or SYS1.MANY. The program obtains the new SMF data from here.

Output formats

(a) File SMFOUT
The next generation of OPS.SMFDATA is written to this file. It is a combination of the data from files SMFIN and SMFNEW.

(b) File SYSPRINT
The number of records on file SMFOUT, and the timestamps of the first and last, are displayed on file SYSPRINT.

(c) File SMFBACK
This file is assigned to the dataset SYS1.BACKUP.SMF. The program adds the new SMF update records to the end of this dataset.

(d) File COST

Similarly SMFCOPY appends new costing records to the dataset associated with this file. The records are each 80 bytes long and have the following format.

Offset	Length	Field
0	6	works authorization no. (WA) - numeric, display format
6	3	cost centre - numeric, display format
9	20	programmer name
29	4	CPU time used by job or TSO session, in hundredths of a second (binary)
33	4	TSO session time, in hundredths of a second (binary)
37	4	Service units used (binary)
41	8	timestamp from SMF record
49	31	unused (blank)

The first record written by SMFCOPY to the file is a special dump header. The characters DUMP appear in the last 4 bytes and the timestamp in the first 8 bytes. These records delimit the information added to file COST each time the program is run. In this way reruns of the program, where duplicate data may be written to the file, can be detected and disregarded by the ADP Costing Scheme.

I.14 SMFMTHS

type - PL/I, main program
calls - ELAPSED (DM), JULIAN (DM)

This is the program that splits the SMF data to form the next generation of both OPS.SMFDATA and OPS.SMFMONTH (see Section 8). The program examines the Julian data found in each input record and determines whether it lies in the previous or current month. The output file is selected accordingly.

Input format

(a) File SMFIN
This is the current generation of OPS.SMFDATA, containing all data from the beginning of the previous month.

Output formats

(a) File LASTMTH
This file is associated with the next generation of OPS.SMFMONTH and will contain all data for the previous month.

(b) File NEWMTH.
This file is associated with the next generation of OPS.SMFDATA and will contain all data for the current month to the present

time.

I.15 STATUS

type - PL/I, main program

This program generates the alphabetic lists of datasets that form the disk volume status report obtained each night (Section 5.) It reads a dataset created by the data migration program VTOCDSN containing the sorted format-1 DSCB's of the datasets. It interprets the information and generates a line of the report for each. Finally the program states the current free space levels for each participating disk volume.

Input formats

(a) File DISKS
This file contains several 80-byte records, each containing the serial number of one participating disk volume in the first 6 bytes.

(b) File VTOC
The program reads the sorted format-1 DSCB's from this file.

Output format

(a) File SYSPRINT
The dataset report is written to file SYSPRINT.

I.16 SQUEEZE

type - assembler, main program

This program generates and submits a batch job to reorganize a disk volume (see Section 4). It reads a set of JCL and replaces occurrences of '$VOLID' with the serial number of the volume and '$DEV' with its device type and then submits the job to the internal reader.

Input formats

(a) PARM field
This field can take one of two forms.

(i) volume(device type)
Volume is the serial number of the disk and its device type must either be 3330 or 3350.

(ii) volume$vol1,vol2,.....
Again volume is the serial number of the disk to be reorganized and it must have one of the values vol1, vol2 etc. The device type is assumed to be 3350.

The program terminates if it detects errors in the PARM field (for instance, if volume is not one of the permitted serial numbers specified in a PARM of form (ii)).

(b) File SYSIN
The input JCL stream is read from this file.

Output formats

(a) File SYSOUT

The program writes the modified JCL to this file, which is directed to the internal reader.

(b) If the PARM field has an illegal format a message is displayed on the operator console.

I.17 TAPECAT

type - PL/I, subroutine
called from - TERASE
calls - TAPEVOL

This routine calls TAPEVOL to determine all catalogued tape datasets and their volumes. It then returns the serial numbers of all tapes that contain at least one catalogued dataset to the calling program, in an array. The current maximum size is 2000 entries.

arguments

number of volumes containing catalogued datasets - FIXED BINARY(15)
the volume serial numbers - (2000) CHARACTER(6)

I.18 TAPEVOL

type - PL/I, subroutine
called from - TAPECAT, CATTAPE
calls - CATLGRD

Subroutine TAPEVOL uses the information returned by CATLGRD to determine the name and volume serial numbers of each tape resident dataset defined in the Operating System's VSAM catalogues. At each call it passes one dataset name and its volume back to the calling program. For multi-volume datasets, the same name is returned on successive calls, each time with the next volume serial number it occupies.

arguments

dataset name - CHARACTER(44)
tape serial number - CHARACTER(6)
end-of-file indicator - CHARACTER(1) - when this is non-blank it indicates that no further tape resident datasets could be found.

I.19 TERASE

type - PL/I, main program
calls - TAPECAT

Program TERASE calls TAPECAT to determine all tape volume serial numbers that contain permanent data. Other input includes the current range of serial numbers of tapes currently in the general pool (see Section 2). From these sources of information the program can deduce which volumes are available for reuse. It presents the serial numbers of each of these volumes in turn to the operator, asking whether it should be erased or not (in case some are already in the scratch pool).

This question and answer sequence can be terminated at any time by the operator, or it will automatically end when the upper limit of serial numbers is reached.

For each volume that the operator consents to erase, the program generates and submits a batch job to perform the operation, using program ERASE (see I.9). All the jobs have unique names and are automatically held, to be released at an appropriate time.

Input formats

(a) PARM field
serial1-serial2
The serial number range of tapes in the general pool is specified in this field.

(b) File CATLG
This file contains a copy of each Operating System VSAM catalogue, as generated by program CATCOPY.

Output formats

(a) File TAPEJCL
The jobstream of 80-byte JCL records is written to file TAPEJCL, which is directed to the internal reader. Each job in the stream will erase one tape.

(b) The system console is used to communicate with the operator.

(c) File SYSPRINT
This file will contain a list of the jobnames and their associated tape serial numbers. This information is used to fetch all the volumes before releasing the erase jobs.

I.20 WARNCMP

type - PL/I, main program

This program is used during the data migration procedure WARNLIST(ref.1) to compress all partitioned datasets selected for migration. Using VTOC information it determines which of the selected datasets are partitioned and generates one IEBCOPY compress job step for each. Finally WARNCMP submits the batch job to the internal reader. A privileged version of IEBCOPY, which can bypass Operating System password checks, is used to compress protected datasets.

The main objective of the program is to identify which of the datasets have an invalid organization, so that the problem can be corrected before the actual migration takes place. This is necessary because the utility used to transfer the datasets to tape, IEHMOVE, will abort if it detects a partitioned dataset with an invalid format.

Input formats

(a) File VTOC
This file contains the sorted format-1 DSCB's of all datasets on disk volumes participating in the migration scheme, and is generated by the data migration routine VTOCDSN(ref.1).

(b) File WARNLS
This file identifies the data migration dataset SYS1.ARCHIVE.WARNLIST(ref.1). The only information from each 80-byte record used by program WARNCMP is the dataset name, in the first 44 positions. These are the names of the datasets which have been selected for migration.

Output format

(a) File COMPRS
The compress job is written to file COMPRS, which is directed to the internal reader.

TABLE 1. MAINTENANCE SCHEDULE

No	Frequ.	Name	Function	Dependencies
1	daily	STATUS (section 5)	list contents of each disk volume	
2		CLEANUP (section 2)	enforce disk dataset naming and management conventions	
3		user disk backups(ref.3)	backup datasets on user disk volumes	11
4	weekly	WARNLIST (ref.1)	select datasets for migration and warn users. Special SMF data processing at end-of-month	2,11
5		ARCHLIST (ref.1,2)	migrate datasets to archives	2,4
6		COMPRESS (section 3)	reorganize partitioned datasets and reclaim space from them	4
7		disk volume reorganization (section 4)	reorganize disk volumes to reduce fragmentation of free space and reclaim space from datasets	
8		system disk backups (ref.3)	backup system disk volumes	
9	fortnightly	dataset access reporting (section 7)	report accesses to datasets by unauthorized users	4,11
10	3 monthly	tape usage reporting (section 6.2)	report names, size and usage of tape datasets	4,11
11	as required	SMF data storage (section 8)	save contents of system SMF datasets - once or twice daily	
12		tape erasing (section 6.3)	erase tapes for re-use as scratches - about 3 times a week	
13		tape transcription (section 6.1)	transcribe tape datasets to a single volume - several times a year	

DOCUMENT CONTROL DATA SHEET

Security classification of this page: UNCLASSIFIED

1 DOCUMENT NUMBERS	
AR Number:	AR-001-455
Report Number:	ERL-0055-TR
Other Numbers:	

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

3 TITLE

MANAGEMENT PROCEDURES FOR CONTROLLING DATA STORAGE ON THE IBM SYSTEM 370 COMPUTER AT DRCS

4 PERSONAL AUTHOR(S):

J.C. Gwatking and
R.W. Collier

5 DOCUMENT DATE:

January 1979

6	
6.1 TOTAL NUMBER OF PAGES	33
6.2 NUMBER OF REFERENCES:	15

7 7.1 CORPORATE AUTHOR(S):

Electronics Research Laboratory

7.2 DOCUMENT SERIES AND NUMBER

Electronics Research Laboratory
0055-TR

8 REFERENCE NUMBERS	
a. Task:	DST 20/014
b. Sponsoring Agency:	DEFENCE

9 COST CODE:

228753/135

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.	1	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

Computers
Data storage
Magnetic disks
Magnetic tapes
Data storage devices
Magnetic storage
Data recording
Information systems
Data processing
Digital computers

b. Non-Thesaurus Terms

Data management
Dataset management

15 COSATI CODES:

0902

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)

This document is one of a series describing various aspects of data management on the DRCS Central Computer. It includes methods for controlling user datasets on both disk and magnetic tape. Where appropriate the operating procedures and supporting software are also outlined.

Security classification of this page: UNCLASSIFIED