IBM Backup, Archive and Retrieval System/VM Program Description and Operations Manual Release 1.3

Document Number SB11-8974-03

October 6, 1997 - 5:26 p.m.

Steve Garraway
Gerhard Widmayer

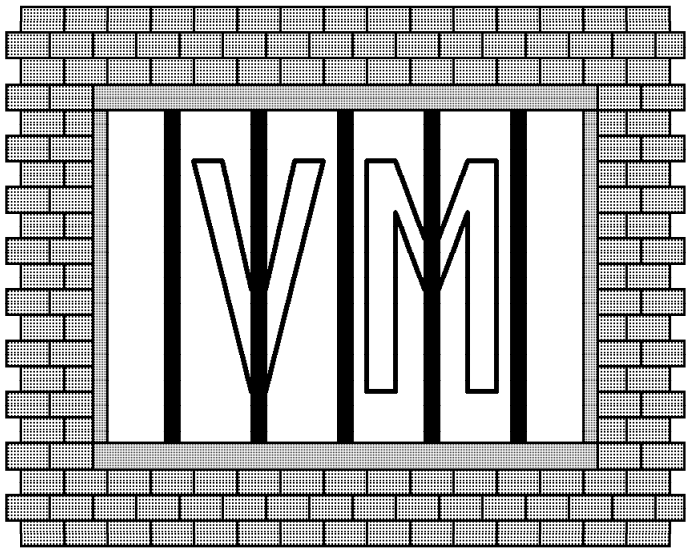
GARRAWAY at WINVMB
WIDMAYER at STUTVM1

SB11-8974-03

IBM Backup, Archive and Retrieval System/VM

Program Description and Operations Manual

Release 1.3



SB11-8974-03

IBM Backup, Archive and Retrieval System/VM

Program Description and Operations Manual

Release 1.3

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page xiii.

Fourth Edition (July 1995)

This edition, SB11-8974-03, applies to Release 1.3 of IBM Backup, Archive and Retrieval System/VM (BARS/VM), Program Number 5785-DJD, and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

A form for readers’ comments appears at the back of this publication. If the form has been removed, address your comments to:

GARRAWAY at WINVMB
WIDMAYER at STUTVM1

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Notices

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

Trademarks and Service Marks

The following terms, denoted by an asterisk (*) in this publication, are trademarks or service marks of IBM Corporation in the United States and/or other countries:

IBM
RACF
VM/ESA

Summary of Changes

6 December 1995 - Modification level 03

Changes made since the modification level 02 of BARS/VM in October 1994.

- Add HDFXGV user exit to allow the installation to do its own authorization checking.
- New installation options EAVFULL and EBVFULL allow more control over reorganizations.
- New CATMAINT command allows an administrator to perform maintenance on the catalogs and control files.
- New VOLMAINT command allows an administrator to perform maintenance on the control files that define minidisks and tapes used for backup and archive.
- DISPLAY NEXTIVOL shows the internal volume serial number which will be given to the next unused minidisk or tape selected for backup or archive use.
- DISPLAY VOLUME REAL display backup and archive minidisks on a specified real volume.
- A new operand, MAXRET, on AALTER allows an administrator to change the maximum archive retention period for an individual user.
- New report messages list the total number of blocks used for backup and archive.
- New FORCE operand on START TAPEDUMP.
- New NOALIAS operand on RESTORE.
- Various minor documentation and code fixes.

31 October 1994 - Modification level 02 refresh

Changes made since the modification level 02 of BARS/VM in January 1994. The changes made are mainly error corrections or preventive maintenance.

- Save and restore SFS file attributes RECOVER and INPLACE.
- Add support for DEVNO full-pack minidisk (if "linkable").
- Add DBR options MAXUNITS, MAXDRIVE, EEXPIRE, EEXPBASE.
- Add LPUT command to enable for user managed "vital records" application.
- All BARS/VM virtual machines must run in XA mode.
- Various minor documentation and code fixes.

21 January 1994 - Modification level 02

Changes made since the modification level 01 of BARS/VM in February 1993.

- New ESECDATA installation option to exploit external security manager in addition to OWNED logic.
- New ECYCLER installation option to define how many different tape volumes to be tried.

- New EBZODAYS installation option to define after how many days backup levels > 0 should be eliminated.
- New EBRSBLK installation option to improve the performance of full volume restore by reducing the number of backup volumes on which backup data is spread.
- New ERASENOW and INCNEW options on BACKUP command.
- Add ARCHIVE option on RENAME command.
- Add backup tape pool support to tape spill.
- Add HDFXLC user exit to validate possible HDR1 record.
- Enhance KILLSUB for interface userid to allow for subsystem restart.
- Add RESTORE of SFS storage group.
- New RESTART option for minidisk RESTORE command.
- Add 19C minidisk for RESTORE checkpoint data.
- Add full volume and storage group RESTORE checkpointing.
- Enhance RETAIN option from three to four digits.
- Add DBR policy Exit HDFXDP.
- Add TAKEALL and FORCEEND options on START DBR command.
- Add DBR possibility for minidisks with no backup data.
- Add ENUMDBRT installation option to define the maximum number of tape units to be used for DBR processes.
- Add 19F minidisk for panel interface code (end user disk).
- Add BEFORE option on RESTORE command.
- Add SELECT option on QUERY command.
- Add DASD volser option on REMOVE command.
- Add archive directly to tape.
- New EDBRCDUP installation option to allow for duplicate DBR control tape.
- New EARFORCE installation option to allow for archive REORG to complete.
- Allow up to 30 backup levels for CMS via line command interface system.
- Obtain installation options EMACHID and ENETID directly from the system.
- Report on minidisks excluded via PRIMARY command.
- Access user minidisks higher than filemode S to avoid overlay problems.
- Add administration automation code to the manager userid.
- Various minor documentation and code fixes.

1 February 1993 - Modification level 01

Changes made since the initial delivery of BARS/VM in January 1992.

- Lots of internal processing corrections and improvements
- Improved progress messages together with reason indicators
- SYSAFFIN logic added to source directory scan to better support CSE coupled VM clusters

- Improved backup restart processing by checkpointing to disk
- Allow general user to RESTORE his own minidisks
- Improved tape RESTORE speed
- Improved full-pack RESTORE

  Pre-links all user minidisks in M mode to minimize interlocks with users logging on during an active RESTORE for their minidisks. This implies that user minidisks are defined in the source directory with a mode other than MW
- Improved tape handling (speed and function)
- Improved DBR ONLINE dumps
  - If ETAPCOMP NO specified, data is dumped much faster to tape, especially if hardware compaction is available
  - RECOVERY directly loads data to the target minidisks if previously dumped directly to tape
- FBA support for devices larger than 999,999 blocks
- New TO option on RETRIEVE command
- New ONEDAY option on BPUT command
- New DAYS keyword on AALTER and APUT commands
- New NOACK option on APUT command
- New USAGE option on AAUDIT command
- New options (NOBACKUP, NODBR, DBR, DATE) on privileged QUERY command
- New authorized DISPLAY SESSION command
- New authorized DISPLAY STATUS RESTART command
- Additional information from DISPLAY STATUS SYSTEM command
- Additional parameters on START REORG and START TAPEDUMP command
- New ETPFULL installation option to avoid unnecessary tape reorganizations
- Limit of backup tapes raised to 37800
- New HDFXCM and HDFXLB user exits
- Usage of storage above 16MB for data
- Various minor documentation fixes

Migration considerations from modification level 02

There are no migration considerations when moving from level 02 to level 03, other than reviewing the new features mentioned above.

Migration considerations from modification level 01

Due to the need for checkpointing data for the RESTART function of RESTORE you must add a **19C** minidisk to the Interface userid.

Due to the provision of a more automated administration via PROP in the Manager userid you should check your local automation already applied. Please refer to Chapter 11, “Manager Commands Description” on page 147 for more details on the functions provided.

The new end user command interface allows to move the panel interface code onto a not accessed minidisk. Please make sure you place the file "***VMBARS USRIFACE***" onto the end user disk of your choice if you decide not to follow the suggested placement.

Migration considerations from modification level 00

Due to restrictions in the control files it was not possible to support more than 999,999 blocks per object. This modification level includes 9336 support with restructured control files. The following migration steps have to be performed once per system.

Migration is required only once to replace cylinder character values with full word values. No data is destroyed if it is run more than once.

1. Shutdown the BARS/VM subsystem (e.g. via ***VMBARS END***).
2. Logon to the Interface userid and refresh the material as described in the installation section using ***BARSINST EXEC***.
3. Logon to any userid authorized to link to the Interface userid minidisks in R/W mode, or to the Interface userid itself.
4. Run ***HDFCNTRL MIGRAT "interface_userid"***.
   This step requires access to the HDFUCM, HDFDUM and HDFMIM migration modules, which are only used here. This step takes some time ... be patient. Note that ARC files do not need conversion, so the migration process may say that 0 files were converted for the 197 minidisk if you do not have any MIG files (used by the MDISK command).
5. Restart the BARS/VM subsystem.

Preface

The objectives of IBM Backup, Archive and Retrieval System/VM are to improve the backup and retrieval of CMS and non-CMS user data for both the installation and users, provide an archive and retrieval facility for CMS files, an archive facility for minidisks, and a disaster backup and recovery facility for minidisks.

The BARS/VM product functions are implemented using certain specific resources (userID, EXEC, MODULE) named 'VMBARS'. In this document, the name VMBARS will be used to refer to these product functions.

BARS/VM is designed to run on VM/ESA* Release 1.1 and subsequent releases. Support for Access Verification Routines (ACI: Access Control Interface) is required unless RACF* (5740-XXH) is installed which provides an Access Control Interface by itself.

Part 1. 'General Description'

Chapter 1. Introduction

The term “backup” is defined as a process of making copies of data by the installation. IBM Backup, Archive and Retrieval System/VM (BARS/VM) provides two types of backup which are complementary - incremental backup and disaster backup.

BARS/VM is designed to automatically back up new and changed CMS files. These files may reside in minidisks and the Shared File System (SFS). BARS/VM also backs up non-CMS minidisks when the user or installation requests it. These backup copies may reside on DASD or tape. The installation may choose one of three methods of backup - to DASD only, to DASD with spill to tape, or directly to tape with no DASD.

BARS/VM provides the user with a QUERY facility which permits the displaying of the current backup data. The user is also furnished with a retrieval capability which obtains the backup copies of data. BARS/VM also provides authorized users with the capability of restoring minidisks, SFS directories, SFS file spaces, SFS storage groups, and complete real volumes from the last backup.

The term “disaster backup” is defined as a process of making copies of data by the installation to be stored outside the computing center, so that data can be recovered after a disaster where the normal backup data is not useable. BARS/VM is designed to dump, on operator request, data to a disaster backup tape pool. This data may be from the original minidisks, or may be the compacted and inactive BARS/VM backup data. It can back up either the complete installation's minidisks (CMS and non-CMS data) or only just a single minidisk or a logical set of items (defined via the LSET command). There is no BARS/VM function to recover disaster backup data. Recovery is achieved by running a RECOVERY EXEC which does not rely on BARS/VM, and which is automatically dumped along with the data.

The archive facilities of BARS/VM provide a means for users to save seldom used CMS files and free the space they occupy. Commands are provided to allow users to send files to the archive, retrieve from the archive and query archived files. Archived files reside on DASD or tape. The installation may choose one of three methods of storing archived files. They may be placed on DASD only, on DASD with spill to tape, or they may go directly to tape with no DASD.

HELP is provided for all commands and messages.

The incremental backup, disaster backup, retrieval and archive processes operate under the control of parameters specified by the installation.

Chapter 2. Requirements and Objectives

2.1 Requirement for a Backup and Retrieval System

Backup is defined as making duplicate copies of CMS files and non-CMS minidisks by the installation.

Backup data is normally stored on magnetic tape and is retrievable if the primary copy is damaged. Because the installation does not know which CMS files or non-CMS minidisks have been changed, all data is usually copied on a regular basis. This involves needless duplication of data that is already backed up and the entire process can require hours. Further, should it be necessary to restore the damaged data, extracting the specific data from tape dumps can take a long time.

If the installation does not assume responsibility to back up user data, it will be done whenever the user deems it necessary. In addition, the user will probably do it at the worst possible time - during prime shift, when the installation already has to handle the peak processor load. Few users will come to work during off-shift hours just to back up their data. To compound the problem, users will make multiple online copies of the same data using different names, or will copy the data onto private tapes. The first approach wastes online storage; the second increases operator tape handling. On the other hand, the user may elect not to back up vital files, and thus risk their irretrievable loss.

An additional, very important consideration may be the ability to store backup copies of data at a remote host computer. This enables an installation to create external backup copies of e.g. vital record data. Such data then can be stored at a location far away from the central main frame and will not be destroyed in case of a disaster at the main location.
Furthermore, remote backup may also be used just to save DASD space for backup at the distributed locations.

BARS/VM Disaster Backup and Recovery (DBR) satisfies the requirements for disaster backup. This allows an installation to create BASE and DELTA data backup (which is a different view from incremental backup) on tapes to be stored outside the computing center. Recovery is independent of BARS/VM and is done on a minidisk basis.

2.2 Objectives of BARS/VM Backup

User Expectations of a Backup System

- Ease of use
- User-initiated backup after critical changes
- User-initiated retrieval whenever there is a need to retrieve data
- The provision of a QUERY and an online HELP facility
- The ability to maintain backed up data easily.

Installation Expectation of a Backup System

- Automatic incremental backup in a reasonable time

- Use of DASD or tape or a mixture of the two as backup media.
- The ability to backup/retrieve data to/from a remote host, e.g. to make external backup copies of vital record data
- The ability to create disaster backup data on tapes, in preparation for moving to an external site as vital record data
- The ability to restore at the CMS file level
- The ability to restore a real volume
- The ability to restore a Shared File System storage group
- User access only to backup of his own data: data security
- The provision of installation controls and reports
- Ease of installation
- Ease of use
- Use of standard CP and CMS commands and interfaces whenever possible.

The general objective of BARS/VM backup is to create an end user oriented backup and retrieval system that satisfies the general user's and the installation's expectations.

2.3 Requirements for an Archive System

Archive is defined as the making of duplicate copies of CMS files for users outside their minidisks or SFS directories and at their request. The requirements for such a system lie in two main areas.

The first requirement is for a long term storage of files which are no longer in active use. This may be needed, for example, on completion of a particular project. Normally a retention period would be specified for the files and this would guarantee retention up to a specified date.

The second requirement is for an extension of the users minidisk or SFS space. Because of increasing demands for DASD space it is becoming more important to control the use of this resource. By providing an archive system, users can remove files which are not required for the particular job they are doing at the time with the knowledge that these files may be retrieved within a reasonable time. Thus user minidisks and SFS filespaces may be smaller and many of their files will be held in compacted form in the archive system.

In addition to archiving individual files, the ability to archive whole minidisks can enhance savings in DASD space by allowing entire minidisks to be removed from online storage to archive storage, once a predetermined period of inactivity had passed.

An additional, very important consideration may be the ability to store data to be archived at a remote host computer. This is an alternative way to create external copies of e.g. vital record data. Such data may be stored at a location far away from the central main frame; they will not be destroyed in case of a disaster at the main location. Furthermore, remote archive may also be used just to save DASD space for archive at the distributed locations.

2.4 Objectives of BARS/VM Archive

User Expectations of an Archive System

- User-initiated archive of CMS files either immediately or overnight for large amounts of data
- User-initiated retrieval of data from archive within a reasonable amount of time
- Ease of use
- The provision of a QUERY and an online HELP facility
- Provision for the backup of archived data to protect against storage media failures
- Provision for the recovery of accidentally deleted files.
- The ability to maintain archived data easily
- The ability to archive data at a remote location.

Installation Expectations of an Archive System

- Minimization of processor resource, especially during prime shift
- Use of DASD or tape or a mixture of the two as archive media.
- The provision of installation controls and reports
- The provision of a means of classifying and identifying the owners of archived files
- The provision of a means of easily identifying and erasing redundant archived data
- Ease of installation
- Ease of use
- Use of standard CP and CMS commands and interfaces whenever possible.

The general objective of BARS/VM archive is to create an end user oriented archive and retrieval system that satisfies the general user's and the installation's expectations.

2.5 Advantages and Disadvantages of BARS/VM

Advantages:

- Reduced manpower costs:
  - Automatic, time controlled backup process
  - User-controlled retrieval
  - No tape handling if DASD is used
  - No operator action necessary for backup, archive and retrieve, if exclusive DASD use is chosen
  - If tape spill is used then unattended spill to DASD may be performed with spill to tape when operators are available
  - If backup is directly to tape, a tape pool may be defined and any tape from this pool will be accepted.

  - Automatic tape pool and full OS label support for disaster backup.
- Backup of new and changed CMS files (incremental backup)
- User requested retrieval of CMS files
- Non-CMS backup on demand or at a user specified interval
- User-requested retrieval of non-CMS minidisks
- Remote backup and archive facilities
- disaster backup facility
- Full minidisk archive facility for authorized users
- Retrieval of user-owned data only - data security
- MUCH improved backup cost effectiveness
- MUCH improved backup effectiveness
- MUCH improved retrieval effectiveness.

Disadvantages:

- The time to restore a real DASD volume is longer than DDR restore from tape.

2.6 Recommendations and Limitations

Recommendations:

Completely automatic operation can be achieved by using DASD as the backup and archive medium. With tapes, manual intervention can be eliminated by use of a suitable tape robot. If DASD is select as the medium, **large capacity DASD**, e.g. 3390, should be used for backup and archive minidisks. If it is desired to use the tape medium, 3480 or 3490 tapes are recommended.

Limitations:

- Only files on permanently mounted volumes can be backed up or archived.
- Only data on disks defined by MDISK statements can be backed up or archived. Data on ATTACHed or DEDICATEd volumes **cannot** be backed up or archived.
- If backup or archive is to DASD, a CMS file cannot be backed up if it is larger than the size of a backup or archive minidisk (defined in ENOBLKBV installation parameter - “General Options” on page 226).

Chapter 3. BARS/VM Overview

3.1 Summary

- Automatic backup of CMS data in CMS copy format.
- Backup of all CMS data (in minidisks and Shared File System) unless otherwise requested.
- Backup of only new or changed CMS files (**incremental backup**).
- Backup of non-CMS minidisks using DDR to copy to CMS files.
- Backup of non-CMS minidisks only if user requests it.
- Multiple backup levels.
- Erasure of backup data "n" days after user erases his data ("n" being an installation-defined number).
- Remote backup and retrieval.
- User and installation initiated retrieval of lost or erased data.
- User commands to create and retrieve archive copies of CMS files.
- Multiple levels of archived files.
- Retention dates and space limitations for archive.
- Backup for archived data.
- CLASS and TAG facilities for archived files.
- Remote archive and retrieval.
- Multiple virtual machines.
- Automatic restart after abnormal termination.
- Disaster backup (system-wide) (**BASE** and **DELTA** dumps).
- Disaster backup for logical sets of minidisks (**only BASE** dumps).

3.2 Terms and Definitions

Term	Definition
Primary Volume	All real volumes defined in MDISK statements of the VM user directory ***minus*** archive minidisks, backup minidisks and volumes excluded by the PRIMARY command. All minidisks on primary volumes are backed up **if** • The device type is supported **and** • Not excluded by an authorized user or the owner of the minidisk. Supported device types are: • **CMS** - all device types supported by the CMS COPY command; • **non-CMS** - all device types supported by the DDR module.
BACKUP minidisks	Contain the backup copies of CMS and non-CMS data on DASD.
BACKUP tapes	Contain the backup copies of CMS and non-CMS data on tape.
BACKUP volumes	Used as a synonym for backup minidisks or tapes
ARCHIVE minidisks	Contain the archived CMS files and minidisks.
ARCHIVE tapes	Contain archived CMS files on tape.
ARCHIVE volumes	Used as a synonym for archive minidisks and tapes. **Note:** Backup and archive minidisks must have virtual addresses in the range 001 to 5FF and must be defined by the VM User Directory in the BARS/VM NOLOG user ID (see ENUSERID in “General Options” on page 226). See 20.1, “Calculating the Size of BARS/VM Minidisks” on page 211 for the calculation of backup and archive minidisk sizes.
REORG volumes	Used to reorganize the data on backup and archive minidisks and tapes.
REMOVAL volumes	Used to eliminate a minidisk or tape from BARS/VM use.
REUSE volumes	Used to return a 'REMOVAL' minidisk or tape to BARS/VM use.
BACKUP cycle	The period when automatic backup is performed. The backup cycle is defined with the BARS/VM TIME command described in 10.2.39, “TIME” on page 140.
DBR	A method of backing up files or minidisks for disaster recovery. In comparison to backup, no restore capability is provided in BARS/VM itself, but a RECOVERY tool is placed on the tapes, so that recovery can be done on a system which does not have BARS/VM installed.

DBR tapes Contain the disaster backup copies of CMS files and non-CMS data on tape.

LSET Contains the definition of minidisks for a logical set. This LSET will be used to group data together on system-wide DBR dumps and provide the capability to run a DBR dump just for this data.

3.3 Virtual Machines Overview

Following is a brief description of all user IDs required to run BARS/VM. For a detailed description to plan and define these user IDs, refer to chapter 20.2, “Planning BARS/VM Virtual Machines” on page 217.

General Users Allowed to issue non-privileged commands only.

Authorized user(s) Additionally allowed to issue authorized commands.

MANAGER user ID The user ID for the person technically responsible for BARS/VM. It is provided for installation and maintenance of the BARS/VM product, but may also be used to perform BARS/VM administration.
Status and error messages are sent to this user ID by BARS/VM.

INTERFACE user ID Controls the work and communication with and within BARS/VM.

Backup Controller (B.C.) decides what backup and archive work has to be done. It is also called primary subuser.

Query Controller (Q.C.) is responsible for handling commands of a query nature, e.g. QUERY, AQUERY, DQUERY. It is also called secondary subuser.

Subusers Do the work assigned by B.C. with the resources allocated by resource control in the Interface machine. The number of subusers is installation defined.

Note: A minimum of **three** subusers (including the BACKUP and QUERY controller) must be defined (see 20.2.1, “BARS/VM Manager User ID - EMUSERID” on page 218).

Figure 1 on page 13 shows the relationships between the various virtual machines used by BARS/VM. Communication between the BARS/VM virtual machines is accomplished by using the Virtual Machine Communication Facility of CP. BARS/VM always sends 2K blocks of storage with codes within them defining the required function (VHCODEs and VHECODEs). A summary of the VHCODEs is given below.

- 00 - command
- 04 - resource
- 10 - backup
- 14 - restore
- 18 - reorganization
- 1C - request for backup work

- 20 - logoff
- 24 - subuser available for work
- 28 - request use of subusers
- 2C - request for restore work
- 34 - deferred RETRIEVE
- 38 - deferred archive
- 3C - tape spill
- 40 - DBR backup

3.4 Issuing BARS/VM Commands

Commands are sent to the BARS/VM Interface user ID using the VMBARS EXEC which uses the VMBARS2 EXEC and VMBARS MODULE. To issue a command to the local BARS/VM the EXEC name VMBARS is used followed by the function desired, e.g. VMBARS QUERY 191. Commands may also be sent to a remote node by specifying VMBARS TONODE followed by the node name and function, e.g. VMBARS TONODE VMSYS1 QUERY 191. In this case the command is executed at the remote node 'VMSYS1' and resulting messages and files are sent back to the issuing node. Commands are sent to remote nodes as CMS files using the virtual machine defined in the ENETID installation parameter (see “General Options” on page 226). Note that general user commands sent to a node are only accepted at that node if the sending node has been defined at the receiving node through the NODENAME command. See the description of the NODENAME command in 10.2.23, “NODENAME” on page 121. Authorized commands are accepted only if the user id and node name of the sender have been defined through the AUTHORIZ command.

3.5 QUERY and HELP Facilities

QUERY and AQUERY: A QUERY command (see 8.2.20, “QUERY” on page 76) is available that allows a user to display information about his backed up minidisks and SFS directories. For CMS data, information may be displayed for each file.

The AQUERY command (see 8.2.6, “AQUERY” on page 61) allows a user to display information about his archived files.

DBR QUERY: The DBR QUERY command allows a user to display disaster backup information for his minidisks and any other data he is authorized to display.

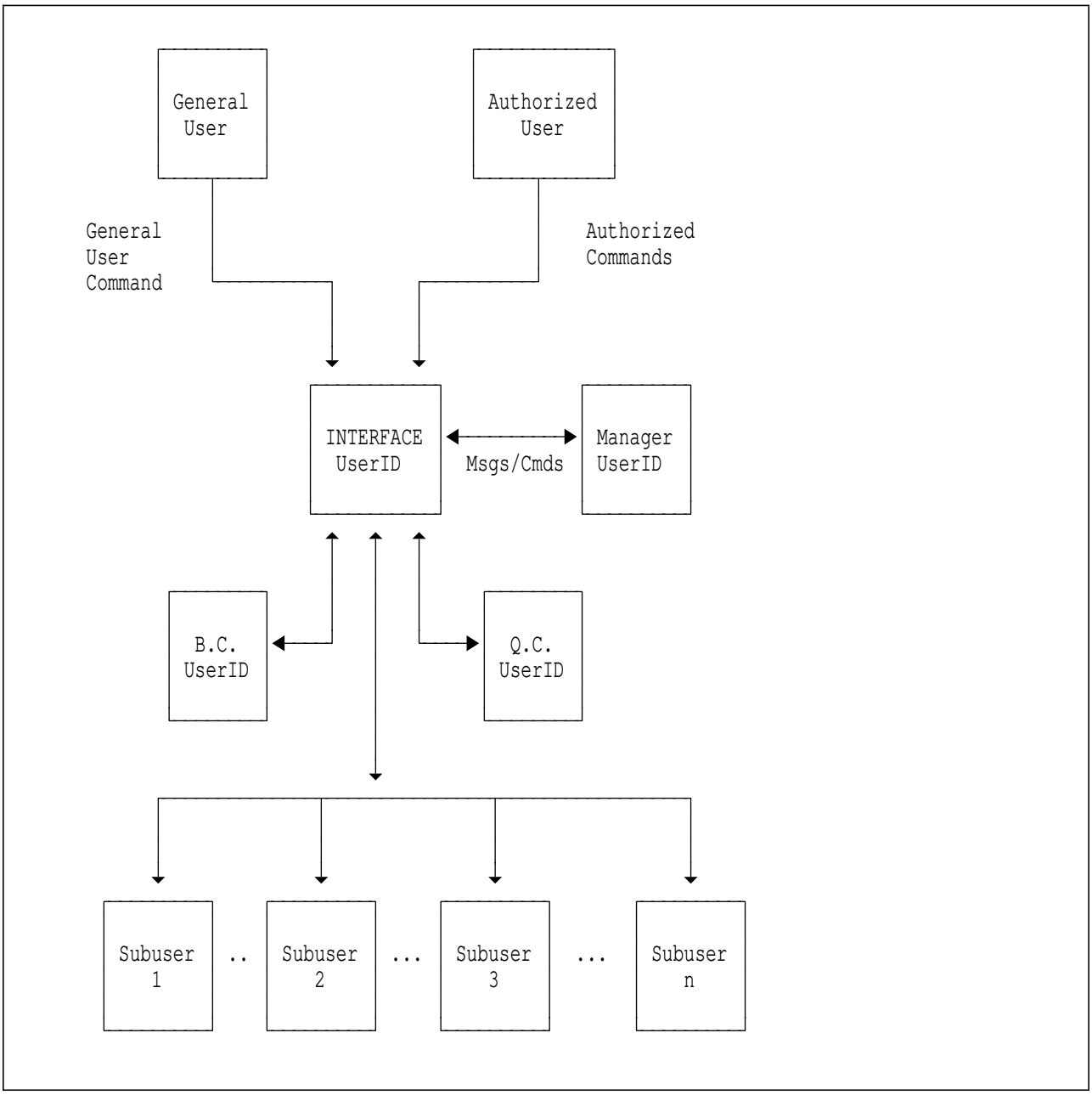


Figure 1. BARS/VM Virtual Machine Overview

DQUERY and DRECOVER: When archived files are deleted, either explicitly through ADELETE or implicitly by an APUT which replaces a file, the information about the deleted file is retained in a special file (DEL file) which is kept for a number of days equal to the ERASDAYS parameter. The DQUERY command (see 8.2.15, “DQUERY” on page 72) allows a user to display this information about archived files which he has deleted and which are still available for recovery and the DRECOVER command (see 8.2.16, “DRECOVER” on page 73) may be used to recover them to his virtual card reader. See 4.16.4, “Archive Catalog: Archive DEL Files” on page 34 for further discussion of archive DEL files.

HELP: A HELP command for the general user (see 8.2.18, “HELP” on page 74) and the administrator (see 10.2.19, “HELP” on page 118) is available to explain commands and messages.

Chapter 4. BARS/VM Processing Overview

4.1 Backup Process

4.1.1 Overview

The control of the number of backup copies is an installation option (see ECMSBACK and EOSBACK in “General Options” on page 226) The frequency of the backup process for a user's minidisk or SFS directory is limited to a minimum number of days set by an installation (see EMINCMS and EMINOS “General Options” on page 226). But, between this minimum and 99, the user may control his own backup interval via the BACKUP command (see Chapter 8, “User Commands Description” on page 51). The backup frequency for CMS data is defaulted to an installation value (see ECMSDAYS in “General Options” on page 226). The user also has other backup controls such as the STOP, ERASE, and TONIGHT options of the BACKUP command (see Chapter 8, “User Commands Description” on page 51).
There are also some occasions when immediate deletion of backup data is necessary. The BDELETE command (see 8.2.9, “BDELETE” on page 64) may be used to do this.

The installation has control over when the user's backup data is erased via the options EBACKE, ERASDAYS, and EMINIDAY (see “General Options” on page 226).

Backup takes place to DASD unless the option EBTONLY is set to YES in the installation options. In this case backup is directly to tape. Control over the compaction of data when backup is directly to tape is provided by the options ETAPCOMP and EBTXF. EBTPOOL may be used to name a pool of backup tapes so that tapes are not requested specifically but any tape will be accepted.

Backup takes place once a day according to the times set by the TIME command. This sets start and stop times for the backup cycles. One or two cycles may be defined. If two are specified the second cycle completes any backup left outstanding at the end of the first cycle. If backup was completed in the first cycle then no more is done during the second. The backup controller virtual machine is logged on at the start of the backup cycle. This machine determines which minidisks and directories are to be backed up and passes this information, via VMCF, to the subuser virtual machines. The unit of work that each subuser is directed to perform is the backup of either a real DASD volume or a Shared File System storage group. Each subuser obtains a backup volume, copies data to this volume from the minidisks on the real volume or the directories in the storage group, and updates the control files. When a backup volume becomes full, it is flagged as such in the control file BACKUP VOLUMES (for DASD) or TAPE VOLUMES (for tapes) and the subuser obtains another one. When a backup volume is marked as full the current date is also set for use by the reorganization process. See also 12.1, “Reorganization” on page 151 for a more detailed description of the REORG process.

The backup cycle defined by the TIME command normally takes place every day. However it can be restricted to specific days by means of the EBDAY option.

If backup is to DASD and no backup minidisk is available for a subuser because they are all full, tape spill will take place if tapes have been defined for backup. Files are dumped from a selected backup minidisk to tape to empty the minidisk so that it may then be used for backup. See also the section 4.8, “Backup Spill to Tape” on page 28 for a more detailed discussion of the tape spill process.

The STOP command may be used to stop the backup process. It will stop when all subusers have completed the requests they are processing.

The BPUT command makes an immediate backup of a minidisk, file or set of files without waiting for the normal backup process. However it is only expected to be used in special circumstances when crucial changes have been made which should be backed up without delay, or when a backup is to be made at a specific time.

Figure 2 on page 17 shows the relationship between user data and backup minidisks and tapes during the backup process to DASD.

Figure 3 on page 18 shows the relationship between user data and backup tapes when backup is directly to tape.

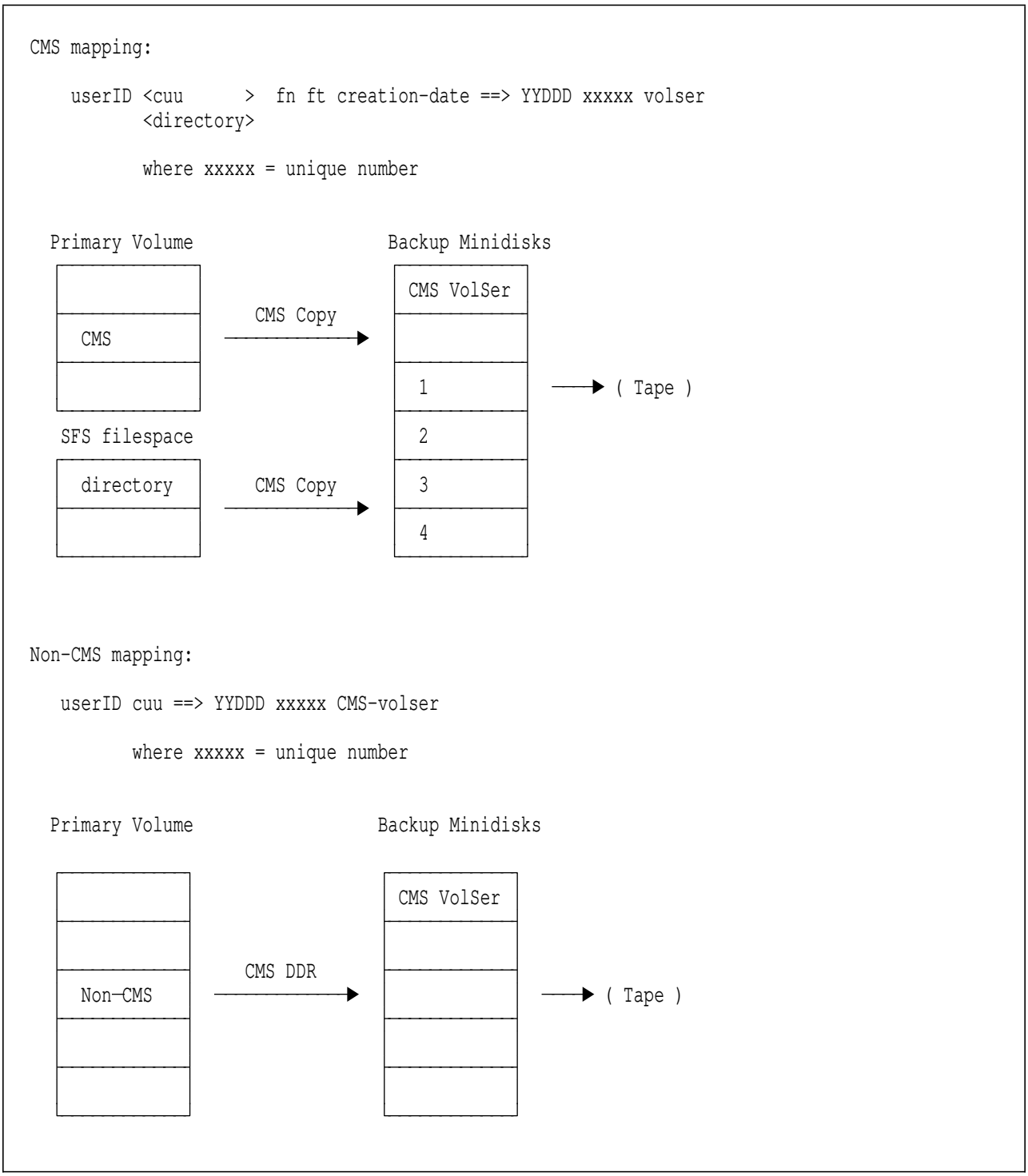


Figure 2. Backup Process Flow. (All Backup Minidisks Are CMS Formatted).

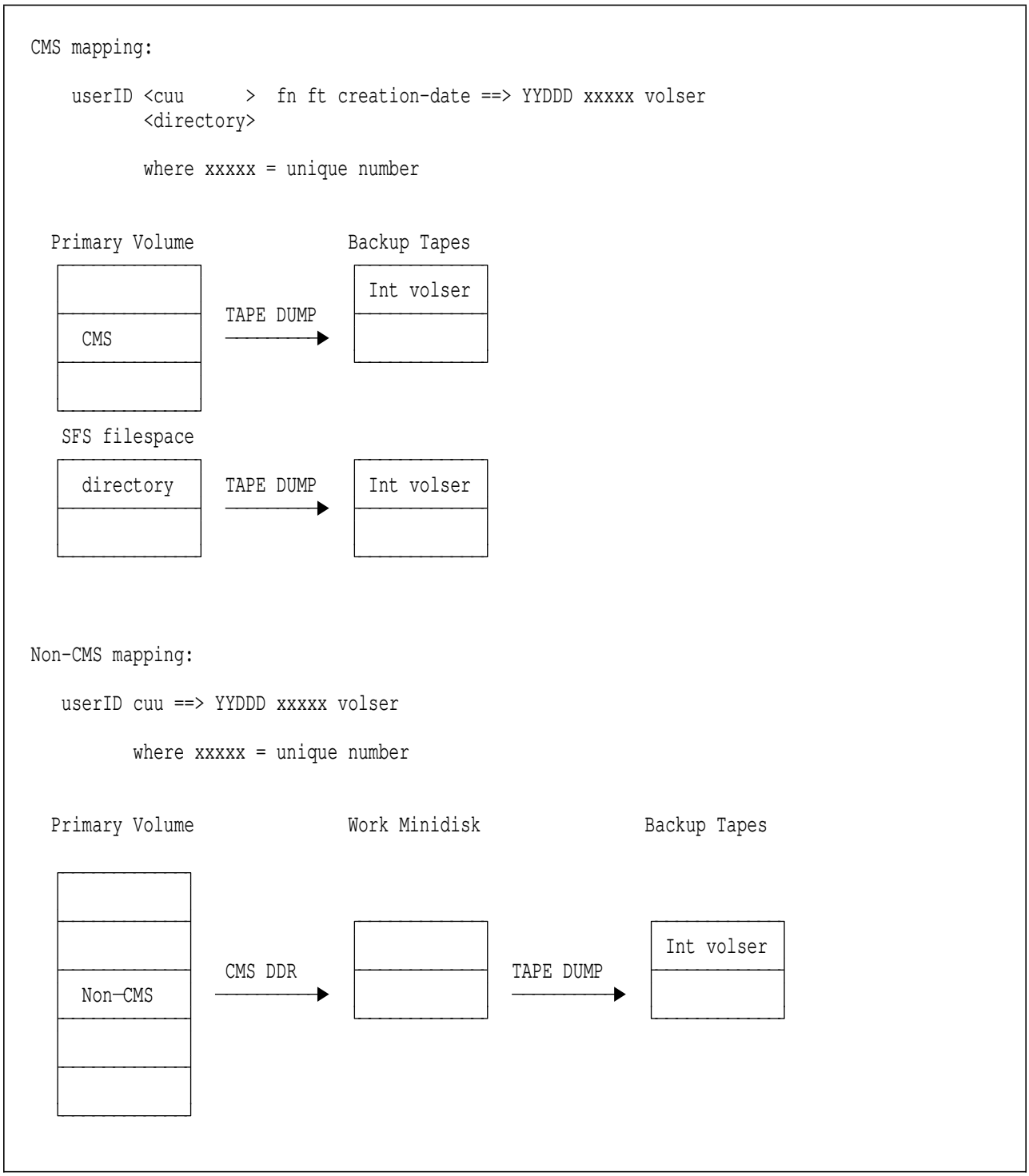


Figure 3. Backup Process Directly to Tape

4.1.2 SFS Backup Considerations

Shared File System data is backed up incrementally in the same way as CMS minidisk data. All the information in the previous section applies both to minidisks and SFS. However there are some additional considerations which apply only to SFS.

Data from a file pool is only backed up if the file pool is defined to BARS/VM by means of the FILEPOOL command. At the start of backup BARS/VM determines which users are enrolled in each defined file pool and schedules incremental backup for all the file spaces. The directory structure in each file space is also backed up, as is enrollment information. This allows the restoration of files, directories, subtrees of directories, and complete file spaces. Storage group information is also maintained so that a complete storage group can be restored.

The backup of alias and authorization information is controlled by the EALIAS and EAUTH installation options. Such information is backed up to a special SFS file space, called BARFS. If alias and authorization data is to be backed up, the file pool in which BARFS is enrolled must be defined in the EBFPOOL option.
| Alias and authorization information for a directory is stored in a directory called
| BARFS.BACKUP.NOLQNAME.NOGLOBALRES.Auuuuuuuu.Axxxxxx where
| uuuuuuuu is the file space name and xxxxxx is the internal directory
| abbreviation. See 4.16.1, “The Backup Catalogs: User ID Cuu and User ID
| Filepool Files” on page 33 for a discussion of these abbreviations. The alias
| backup is stored in a file called DIR ALIASES. The file authorization data is
| stored in a file called DIR AUTHS and file authorization data in FILE AUTHS.

| Alias and authorization backup does not have levels like file backup. At each
| backup cycle the alias and authorization data is saved in its entirety and this
| replaces the data saved from the previous cycle. So it is impossible to go back
| to a previous level of data.

4.2 Retrieval Process

General users may retrieve data from backup using the RETRIEVE command (see 8.2.22, “RETRIEVE” on page 79).

The installation controls the amount of data retrieved by a single command via the values of EMAXFILE and EMAXCYL (see “General Options” on page 226). If the user retrieves a CMS file, the file is sent to his virtual card reader. If the user retrieves a non-CMS minidisk, the retrieved data is DDRed to his minidisk.

If the user defers the retrieval of his data it is processed at the start of the backup cycle or when an authorized START DEFERRED command is received.

Figure 4 on page 20 shows the relationship between BARS/VM backup minidisks and the retrieving user ID during the RETRIEVE process.

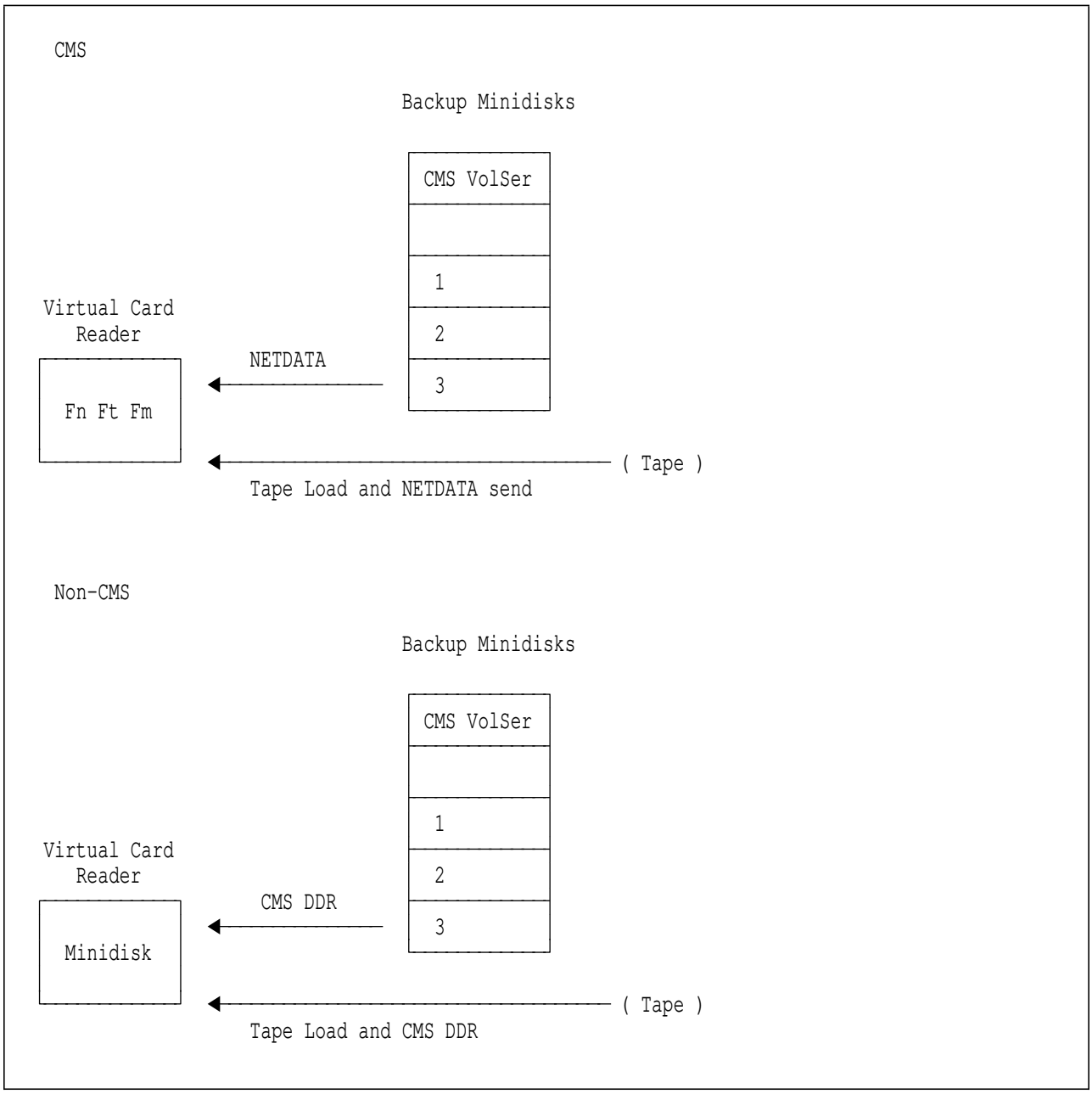


Figure 4. Retrieve Process Flow

4.3 General User Restore Process

A general user may restore a complete minidisk provided it belongs to his user id or if he has been authorized to the user id who owns the minidisk. The RESTORE command is used to do this.

A general user may also restore SFS directories and objects into SFS directories. The RESTORE command can restore a specified Shared File System (SFS) directory, file or alias with the last backup copy.

A restore of an SFS directory includes creation of the directory, the restoration of all files within the directory and the creation of any aliases that existed at the

time of the last backup cycle. Optionally, any sub-directories may also be restored. Directory and file authorizations are regranted unless specified otherwise. A directory to be restored must not already exist.

Files and aliases may be restored directly into an SFS directory by specifying file name and file type. This does not create any directories but simply restores the specified objects to the directory, which must already exist. Subdirectories are not processed in this case.

4.4 Authorized Restore Process

An authorized user may restore a minidisk, SFS directories, an SFS file space, a complete primary volume or an SFS storage group.

4.4.1 Minidisk Restore

When a minidisk is restored it is set to the state it had at the time of the last backup. Note that the RESTORE of a CMS minidisk is different to a RETRIEVE because the disk is formatted, and backed up files are copied directly to it.

The performance of minidisk restores may be improved by periodic use of the BACKUP command with the COLLECT option to make a new complete backup of the minidisk. See the description of the authorized version of the BACKUP command in 10.2.6, “BACKUP” on page 102. The EMAXBD installation option also allows control over the performance of restore and retrieve operations by limiting the number of backup minidisks or tape volumes used for individual minidisks.

4.4.2 File Space Restore

For an SFS restore, the object to be restored must not exist when the command is received otherwise it is rejected. A restore of an SFS directory includes the restoration of all files within the directory and the creation of any aliases which existed at the time of the last backup cycle. Directory and file authorizations are regranted unless otherwise specified. If a top directory is specified (a file space restore), the user is enrolled into the appropriate SFS file pool.

4.4.3 Primary Volume Restore

For a total primary volume restore, all minidisks for which backup exists are restored. Non-CMS minidisks are restored with DDR; CMS disks are formatted and files copied to them from backup. The restore process is organized to minimize the number of backup minidisk LINKs and ACCESSes and tape mounts required. Minidisks are not restored one at a time but rather all data on one backup minidisk or tape is restored to the relevant minidisks and then another backup minidisk or tape is selected. This reduces the time required for the restore process but even allowing for this the total time could be considerable if there are many backup minidisks. One way of reducing this time is if there is a complete backup of the volume available. This may be used to restore the volume as it was at the time the complete backup was taken. The RESTORE command with the SINCE option may then be used to bring the data up to date. With this option, CMS minidisks are not formatted and only files backed up since the specified date are restored.

The EBRSBLK installation option allows control over the performance of primary volume restores by limiting the number of backup minidisks or tape volumes used.

At the start of the restore process, the controller id obtains a write link to all minidisks on the volume in an attempt to prevent users from linking their minidisks read-write. The subusers then use MW links to enable them to write data to the minidisks. Data integrity is maintained via internal resource queueing and because the controller id does not ACCESS the minidisks.

4.4.4 Storage Group Restore

An SFS file pool is made up of two parts, the control data and the user data. The control data consists of the file pool control minidisk and the catalog minidisks (storage group 1). The control minidisk contains a map of all the blocks in the file pool and shows which blocks are in use and which are not. The catalog minidisks contain information about files, aliases, directories, authorizations and so on. The user data consists of storage groups 2 and above.

BARS/VM backs up user data, and catalog information but does not back up the control minidisk. If you do not do control data backups you will be forced to re-generate the file pool and restore all the file spaces individually. For this reason, the control data should be backed up as described in the SFS administration manual using FILEPOOL CONTROL BACKUP. This backs up the catalog storage group and the control minidisk.

In the event of the loss of a minidisk belonging to a user storage group, the storage group may be restored by means of the BARS/VM RESTORE command. When the RESTORE command is issued for a storage group it restores only user data. Since the control data is still valid, all files, directories, etc still exist. BARS/VM therefore restores only the file data from backup.

In summary, the recommended method for SFS storage group recovery is

- Use FILEPOOL CONTROL BACKUP to back up the control data. If it is required to restore this data, use the method described in the SFS manuals.
- Use BARS/VM to back up the user data. In the event of the loss of a minidisk in a user storage group, restore the complete storage group using BARS/VM RESTORE, once the minidisk has been correctly formatted for SFS use. To format a minidisk use the following commands

  ```
  FORMAT vdev access-letter (BLKSIZE 4K
  RESERVE filename filetype filemode
  ```

  The filename should be the name of the file pool and the filetype the DDNAME for the minidisk as defined in the POOLDEF file for the file pool.

The method used to restore the data is very similar to that described above for a primary volume restore. The EBRSBLK installation option may be used to improve the performance of the restore process in the same way as for primary volumes.

4.4.5 Restore Checkpointing

The checkpoint feature of primary volume restore, storage group restore and minidisk restore allows the restore process to be restarted without losing the benefit of any work successfully completed by the initial command. Use the DISPLAY CKPT command to display checkpoint information for restore processes which have not completed. This command is also useful while a restore process is running to illustrate the progress of the restore. The following example shows this.

```
vmbars restore filepool wintom group 2
Command issued
Ready;
HDFRES0301I There are 319 directories to be restored
HDFCKP0510I Checkpoint file WINTOM SG00002 K2 has been created
vmbars dis ckpt all
Command issued
Ready;
HDFDIS0004I DISPLAY command started
Checkpoint ALL      (WINTOM   SG00002 )
Initial command issued by TOM      at WINVMSXA at 15:01:45 on 1995/02/28
HDFDIS0001I DISPLAY command complete
vmbars dis ckpt all sum
Command issued
Ready;
HDFDIS0004I DISPLAY command started
Checkpoint ALL      (WINTOM   SG00002 )
Initial command issued by TOM      at WINVMSXA at 15:01:45 on 1995/02/28
--------------------------------------------------------------------------------
Total number of steps                    977
Number of steps completed                334
Number of steps completed with errors      0
Number of steps currently running          2
HDFDIS0001I DISPLAY command complete
vmbars dis ckpt all sum
Command issued
Ready;
HDFDIS0004I DISPLAY command started
Checkpoint ALL      (WINTOM   SG00002 )
Initial command issued by TOM      at WINVMSXA at 15:01:45 on 1995/02/28
--------------------------------------------------------------------------------
Total number of steps                    977
Number of steps completed                345
Number of steps completed with errors      0
Number of steps currently running          2
HDFDIS0001I DISPLAY command complete
```

Restart processing is not invoked automatically, so restarting BARS/VM while checkpoint information exists will not cause the restore process to restart. To restart a restore use the RESTART option of the command. The following example illustrates the use of RESTART.

```
vmbars restore filepool wintom group 2
Command issued
Ready;
HDFRES0301I There are 319 directories to be restored
HDFCKP0510I Checkpoint file WINTOM SG00002 K2 has been created
  .
  (At this point the system goes down and is restarted)
  .
vmbars dis ckpt all sum
Command issued
Ready;
HDFDIS0004I DISPLAY command started
Checkpoint ALL      (WINTOM   SG00002 )
Initial command issued by TOM      at WINVMSXA at 14:51:40 on 1995/03/01
------------------------------------------------------------------------------
Total number of steps                    977
Number of steps completed                 42
Number of steps completed with errors      0
Number of steps currently running          1
HDFDIS0001I DISPLAY command complete
vmbars restore filepool wintom group 2 ( restart
Command issued
Ready;
HDFRES0301I There are 319 directories to be restored
vmbars dis ckpt all sum
Command issued
Ready;
HDFDIS0004I DISPLAY command started
Checkpoint ALL      (WINTOM   SG00002 )
Initial command issued by TOM      at WINVMSXA at 14:51:40 on 1995/03/01
Restart command issued by TOM      at WINVMSXA at 15:05:03 on 1995/03/01
------------------------------------------------------------------------------
Total number of steps                    977
Number of steps completed                 71
Number of steps completed with errors      0
Number of steps currently running          2
HDFDIS0001I DISPLAY command complete
HDFRES0318I Restore of filepool WINTOM storage group 2 complete
HDFCKP0511I Checkpoint file WINTOM SG00002 K2 has been deleted due to successful
```

To remove checkpoint information, use the `RESET` option of the command.

CAUTION:
The backup, tape spill and reorg processes will not run while primary volume or storage group restore checkpoint information exists. This restriction is necessary to prevent the checkpoint information being invalidated by changes to the backup catalogues and/or backup repository contents. In addition, any checkpoint information for minidisk restores is deleted at the start of backup, since backup would invalidate the saved checkpoint information.

4.5 Archive Process

The user sends files to archive with the APUT command (see 8.2.5, “APUT” on page 59) and retrieves them by using AGET (see 8.2.4, “AGET” on page 58).

When the user retrieves an archived CMS file, the file is sent to his virtual card reader in NETDATA format.

Before being permitted to use the archive system the user must define his department and account identification using the ADEFINE command (see 8.2.2, “ADEFINE” on page 56) so that archived data can always be associated with an owner.

Files are not erased from archive when an AGET is issued, but only when explicitly deleted using ADELETE (see 8.2.3, “ADELETE” on page 57) or implicitly deleted using APUT. The AALTER command (see 8.2.1, “AALTER” on page 55) is used to alter information associated with archived data and the ARENAME command (see 8.2.7, “ARENAME” on page 62) can be used to rename archived files.

See Figure 5 on page 26 for the relationship between user data and the archive volumes during APUT processing and Figure 6 on page 27 for AGET processing.

Archive data resides on DASD or tape as defined by the installation. Data on DASD is automatically backed up in the same way as user minidisks. If an archive minidisk is damaged, it may be restored from backup using the RESTORE command. Archive data on tape may be protected using twin tapes. See the section 4.12, “Archive Twin Tapes” on page 30 for a discussion of twin tapes.

4.6 Installation Controls of Archive Data

An important consideration for any archive system is the ability to control the quantity of archived data and the provision for the regular elimination of data which is no longer required.

BARS/VM provides two methods of controlling the growth of archived data.

The first method of control is to limit the amount of archive space for each user. When a user first issues an ADEFINE command he is set a maximum amount of archive space, which is defined with **EMAXSPAC** in the installation options (see “General Options” on page 226). Once a user reaches his limit, he is unable to archive any more files unless he deletes data by using the ADELETE command or has his limit changed by the BARS/VM administrator. The authorized **AALTER** command described in 10.2, “Command Syntax Description” on page 98 is provided to make such a change.

The second method of control is through retention periods and deactivation. Each installation chooses a default retention period for archived files (ERETAIN). A maximum is also specified (EMAXRET) and users may give retention periods to their data up to this value. Both installation retention values may be listed using the DISPLAY OPTIONS command. Rather than simply erasing data when its retention period has expired, the data will be set as "deactivated". This provides an intermediate stage for data between normal archived state and being deleted. The deactivated state gives an installation the opportunity to warn users that their data has exceeded its retention date and will be deleted unless reactivated within a certain time. Users should be informed by data administration personnel of their policy toward the erasure of expired data.

Files which have been deactivated may be listed using the AQUERY command with the DEACT operand. They may be reactivated by assigning new retention dates with AALTER. Administrators can reactivate complete sets of files with the REACT operand of the AAUDIT command.

See page “General Options” on page 226 for a definition of the installation parameters for archive.

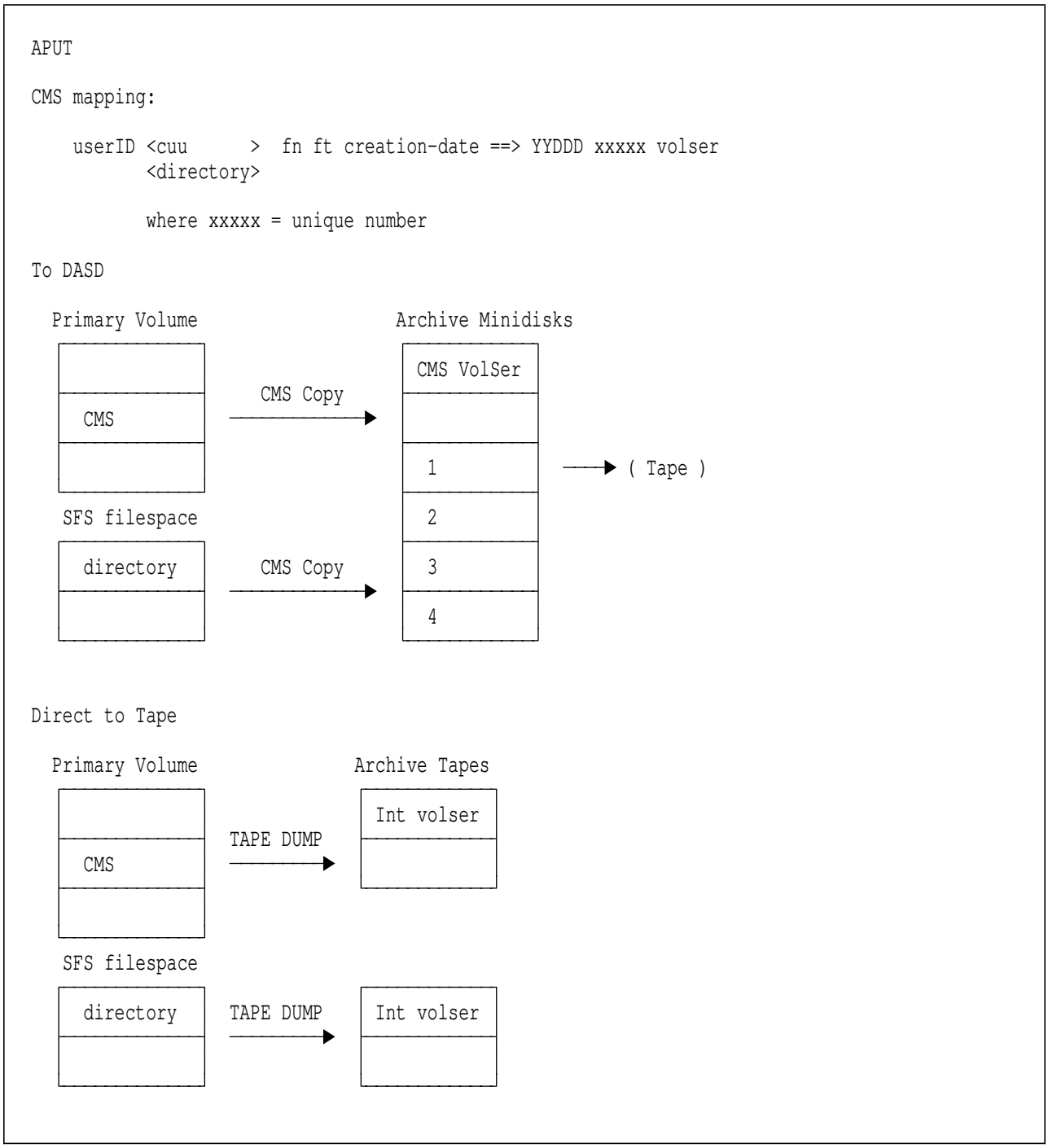


Figure 5. APUT Process Flow

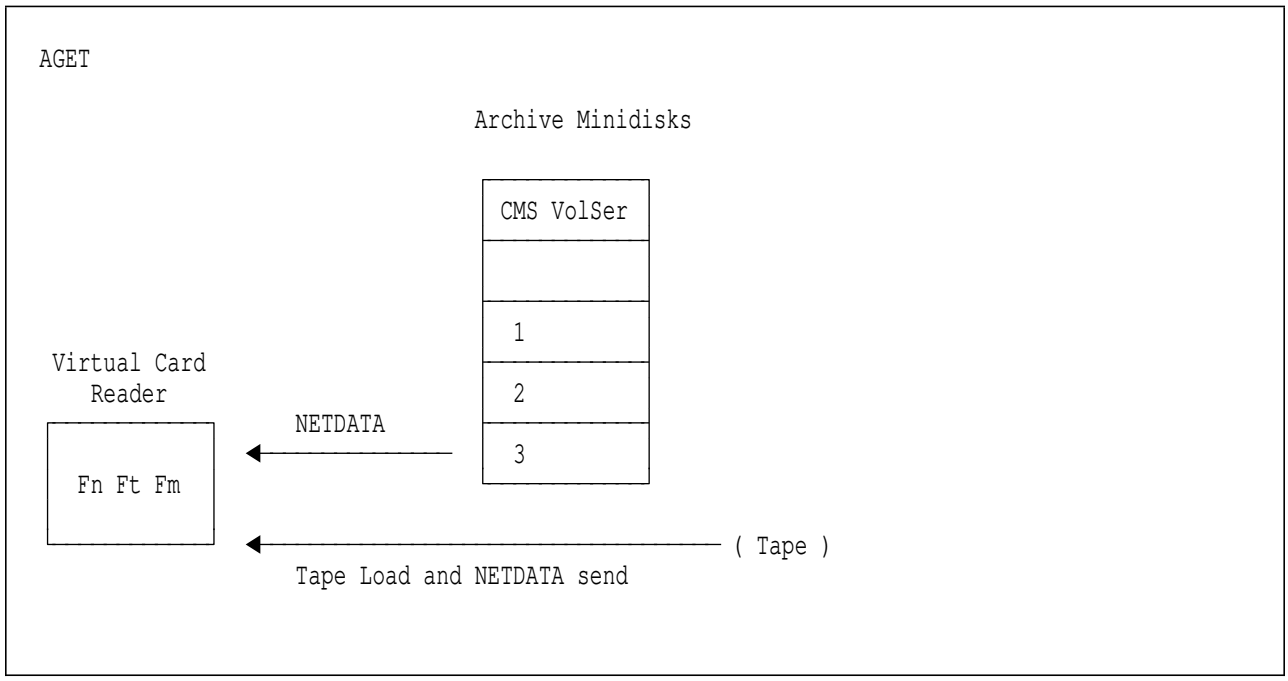


Figure 6. AGET Process Flow

4.7 Disaster Backup Process

Disaster backup takes place when an authorized person issues a DBR START command.

If the option BASE is used, BARS/VM scans its backup information and dumps all needed files from its inactive and highly compacted backup data. If a file does not exist there, maybe because remote or tape backup is used, it is dumped from the users minidisk. This default backup policy may be modified by means of a user exit. This makes it possible to dump data which has no existing backup and to choose how the data is dumped. Only tapes are used to store disaster backup data. They are taken out of a disaster backup tape pool which have been created using the DBRTAPE command. All tapes receive a full OS standard label, including an expiration date to protect disaster backup tapes against misuse.

If the option DELTA is used, BARS/VM only dumps those files which are new or have changed since the last BASE dump. This means that any DELTA dump includes all data from the previous DELTA dumps. So there is no need to do any reorganization as is done for BARS/VM backup tapes (backup data is created via an incremental method).

BARS/VM also provides a restart capability.

At the end of a disaster backup dump, a control file tape is created containing all data which is needed to recover the data after a disaster occurred. Recovery is achieved through a RECOVERY EXEC at any location running VM with a CMS Release 4 or higher without the need to have BARS/VM installed.

A special disaster backup dump can be requested in addition to the system wide BASE or DELTA dumps for either a user ID/minidisk or a logical set of minidisks

(herein called LSET, see LSET command for further information). This special run will have no effect on the system wide disaster backup dumps. It may be used to provide the data for applications with need a shorter frequency than the normal BARS/VM disaster backup.

Note: The user ID or LSET disaster backup dumps are always full dumps, no BASE/DELTA logic is provided.

4.8 Backup Spill to Tape

As well as backup to DASD minidisks, BARS/VM supports backup to tape. Whether the tapes used are mounted by hand or by a system providing automatic mounting, sometimes the best way to use the tape drives is to write large amounts of data to a tape at a time and not have tapes mounted and either idle or being written to intermittently. For this reason, BARS/VM can be set up to use DASD backup minidisks which are emptied to tape when there is no space available. The installation can choose the amount of DASD backup space it provides and hence the quantity of data which can be retrieved from that medium rather than the slower tapes.

Spilling of backup data to tape may take place if any tapes are defined. Automatic spill will occur if the EAUTOSPL installation parameter is set to YES and it can be made to tape place at any time through the START TAPEDUMP command. When tape backup is used, backup minidisks are not reorganized but the data copied to tape when a DASD backup minidisk becomes full. Then the DASD minidisk is reused. When files are copied to tape, the corresponding UC files are updated to show the new volume and the tape file numbers. Tapes are reorganized in a similar way to backup minidisks in a BARS/VM system with no tapes. The USERID CUU files are updated to show the new positions of the files on tape.

Tapes are defined to BARS/VM through the TAPE command. The DASD minidisks used to contain the data initially before copying to tape are defined in the directory of the NOLOG user ID in the normal way. If no tapes are defined, BARS/VM carries out backup to DASD minidisks only. The installation parameter, ENUMTAPE, defines how many tape units are available. BARS/VM will then organize its use of tapes according to this number. It is especially important to reduce the number of tape mounts required for data retrieval. For this reason it is recommended that use is made of the EMAXBD installation parameter. See the description in the section on installation parameters in “General Options” on page 226.

Data is normally copied to tape when there is no DASD minidisk space available. This is sometimes called spilling. The START TAPEDUMP command allows the spill process to be started at any time, allowing an installation to empty backup minidisks to tape at a specific time if it so desires. This is useful in the situation where tape operators are only available at certain times. For example, the DASD minidisks may be emptied by the operators and then backup performed overnight to DASD while the system runs unattended.

4.9 Backup Directly to Tape

The installation may choose to do backup directly to tape. In this case no DASD backup minidisks are defined.

The options ETAPCOMP and EBTXF control whether compression takes place when backing up CMS files. Compression may be done using the IDRC feature of tape drives or by compression to a work disk before being dumped to tape. See Figure 7 for a table of the actions taken for various settings of ETAPCOMP and EBTXF and whether the tape drive has IDRC capability or not. Non-CMS data is always compressed to the work disk before being dumped to tape

EBTXF	ETAPCOMP	Device has IDRC	Action
No	No	No	Dump file direct to tape
No	No	Yes	Dump file direct to tape
Yes	No	No	Dump file direct to tape
Yes	No	Yes	Dump file direct to tape with XF
No	Yes	No	Pack file to work disk and dump
No	Yes	Yes	Pack file to work disk and dump
Yes	Yes	No	Pack file to work disk and dump
Yes	Yes	Yes	Dump file direct to tape with XF

Figure 7. Compression when Backup is Directly to Tape

When compression is performed using a work disk, each subuser obtains a work disk and keeps it until the end of backup. This is to avoid the overhead of repeatedly accessing and releasing the disk. For this reason it is necessary to have several work disks so that each subuser can allocate one. See 20.2, “Planning BARS/VM Virtual Machines” on page 217 for information on allocating additional work minidisks.

The tapes produced by direct backup are compatible with those produced by the spill process and vice versa. So the installation can change from one method to the other. Data retrieval and the reorganization process are the same for both methods.

4.10 Archive Spill to Tape

As well as backup minidisks, archive minidisks may also be spilled to tape. Tape spill is initiated by the START TAPEDUMP command. This may be used to spill all or selected minidisks. After spilling, the DASD minidisks are reused for further archive data. When files are copied to tape, the corresponding catalog files are updated to show the new volume and the tape file numbers. Archive tapes are reorganized in a similar way to backup tapes, except fast reorganization is never used - data is always copied from an archive tape to a tape reserved for reorganization. The USERID ARC catalog files are updated to show the new positions of the files on tape. Archive spill is only performed if the installation parameter EARSPILL is set to YES.

By the specification of a date on the START TAPEDUMP command it is possible to spill only files that were archived on or before that date. This allows older files to be moved to tape while keeping more recent ones on DASD.

Archive tapes are defined to BARS/VM through the TAPE command in the same way as backup tapes.

4.11 Archive Directly to Tape

The installation may choose to do archive directly to tape and to define no DASD archive minidisks. This is controlled by the EATONLY installation option. Control over the compaction of data when archive is directly to tape is provided by the options EATCOMP and EBTXF. See Figure 8 for a table of the actions taken for various settings of EATCOMP and EBTXF and whether the tape drive has IDRC capability or not.

EBTXF	EATCOMP	Device has IDRC	Action
No	No	No	Dump file direct to tape
No	No	Yes	Dump file direct to tape
Yes	No	No	Dump file direct to tape
Yes	No	Yes	Dump file direct to tape with XF
No	Yes	No	Pack file to work disk and dump
No	Yes	Yes	Pack file to work disk and dump
Yes	Yes	No	Pack file to work disk and dump
Yes	Yes	Yes	Dump file direct to tape with XF

Figure 8. Compression when Archive is Directly to Tape

The tapes produced by direct backup are compatible with those produced by the spill process and vice versa. So the installation can change from one method to the other. Data retrieval and the reorganization process are the same for both methods.

4.12 Archive Twin Tapes

Archive data on DASD is backed up automatically by BARS/VM in the same way as any other minidisk. To provide optional backup for archive data on tape BARS/VM uses twin tapes. If the value of EARTWIN is set to YES in the installation options, a twin tape is allocated for every archive tape. This twin is maintained as a duplicate of the archive tape and every file written to the archive tape is also written to the twin. During the archive or spill process, an archive tape and a twin are both assigned. The same data is written to both tapes and they both have the same internal label. During reorganization, data is copied from the original archive tape to a reorganization tape and to another reorganization tape which then becomes a new twin. The original archive tape and the old twin tape, if any, are then marked as reserved for reorganization.

In the event of loss or damage to an archive tape, it can be replaced by the twin. The original tape may be removed by using the REMOVE command. A TWERP command issued against the twin will turn it from a twin into an ordinary archive tape which then replaces the original. A new twin will be created when this is subsequently reorganized. If a twin tape is damaged, it can be removed. A new twin will then be created when the corresponding archive tape is reorganized.

If archive tapes are created when EARTWIN is set to NO and then the parameter is set to YES to turn on twinning, twins are created when the archive tapes are reorganized and are maintained normally after that. If EARTWIN is changed from YES to NO, the twin tapes are left as they are and no longer used. They may be removed by the REMOVE command if necessary.

4.13 Use of Specific Tapes

During a normal (not fast) tape reorganization, when a RETRIEVE, RESTORE, APUT or AGET command is being processed, or during backup or tape spill when pool tapes are not being used, a subuser requests a specific tape through a resource request to the Interface virtual machine and then calling the user exit, HDFXMO, to issue a mount request. The subuser waits for the tape to be mounted and then verifies it is the correct tape. When the subuser is finished with the tape it either gives it to Interface or demounts it and then requests Interface to free the tape resource.

Note that the tape is only demounted when either the ETAPEREL option is set to YES or when the GIVE command is not implemented and the LEAVE option of DETACH is not available. The discussion below is for the situation when ETAPEREL is set to NO. If ERTAPEREL is set to YES, no attempt is made to use GIVE or DETACH with LEAVE.

The way BARS/VM handles tapes can be improved if the system allows the LEAVE option on DETACH. If this is not available a GIVE command may be added to CP. GIVE and DETACH with LEAVE are only used if ETAPEREL is set to NO, which is the default. In this case, to determine the processing required, BARS/VM first attempts a GIVE command. If this fails it tries DETACH with LEAVE and if this fails it knows neither are available. The way BARS/VM processes tapes for the different cases is described below.

4.13.1 Without the GIVE Command and the LEAVE Option on DETACH

When a tape resource request is issued by a subuser, Interface checks its tables to see if the requested resource is already allocated. If it is allocated the request is queued. Note that here the resource can only be already allocated to a subuser and not to Interface itself (as with the GIVE command). If the resource is not allocated, Interface updates its tables to show that the requesting subuser has the tape resource. If all the available tape addresses are in use by subusers then Interface queues the request.

When the subuser obtains the tape resource it calls the tape mount user exit module, HDFXMO, to request mounting of the tape. An installation may place any desired code in this module according to local arrangements for tape mounting. The default module supplied simply issues a message of the following format to the operator user ID:

```
MOUNT label address mode FOR USER subuser

Where  label   = External label of tape or tape pool name
       address = Address at which tape is to be attached
       mode    = R/O or R/W
       subuser = Subuser that tape shall be attached to
```

An address at which to give the tape back is set by Interface in the VMCF block passed back to the subuser on completion of the tape resource request. This address is saved by the requesting subuser. The subuser waits for the tape to be received and verifies the tape label.

If no error occurs while mounting the tape then, on finishing with the tape, the subuser will call the demount module, HDFTPD, which attempts to give the tape back to the interface user ID. This fails and so the tape unit is detached.

A VMCF is then issued by the subuser to free the resource. Interface checks to see if it has been given a tape at the expected address. It will not find it, and so the entry in its tables for this tape is deleted. Also the address at which the tape was expected back is made available to interface again.

4.13.2 With the LEAVE Option of DETACH

Getting a new tape mounted is identical to the situation when we do not have the LEAVE option.

If all tape addresses are in use, Interface detaches the first one in its tables belonging to the Interface user ID. If no tape is found belonging to Interface then the request is queued.

Note that the maximum number of tape addresses available to Interface is defined by ENUMTAPE.

When the subuser has finished with the tape it calls HDFTPD, which gives the tape back to the Interface user ID. The address at which to give the tape back has been saved by the calling program, immediately after issuing the VMCF when requesting the tape resource. A tape is given to Interface by issuing a DETACH command with the LEAVE option and then an ATTACH command to attach the tape to the Interface user ID.

A VMCF is then issued by the subuser to free the resource. Interface checks to see if it has been given a tape at the expected address. It finds the tape, and so it updates its tables to show that it has the tape.

If the tape is requested again by a subuser Interface recognizes it has the tape and gives it to the subuser, updating its tables and allocating the resource. The tape is given to the subuser by a DETACH command with the LEAVE option and then an ATTACH command to attach the tape to the subuser.

4.13.3 With the GIVE Command

The processing with the GIVE command is the same as for DETACH with the LEAVE option except that the GIVE command is used to give tapes instead of DETACH with LEAVE followed by an ATTACH.

4.14 Use of Pool Tapes

If the EBTPOOL installation option is not specified, tapes used for backup spill or backup directly to tape are requested specifically by their volume serial numbers as described above. If EBTPOOL is used to define a name for a pool of tapes, the mount message specifies the backup tape pool name. The operator may mount any backup tape from the pool. Alternatively, tape units may be premounted and attached to the subusers using any of the virtual addresses in the normal CMS tape range (180-187 and 288-28F). The Automatic Cartridge Loader is supported so that several tapes can be loaded into a tape drive and these tapes are used in turn.

The tape processing for data retrieval, tape reorganization and archive spill does not use the tape pool name and always requests specific tapes.

4.15 Use of Tapes by the Disaster Backup Process

Disaster backup tapes are defined to BARS/VM through the DBRTAPE command. All tapes defined for disaster backup are controlled by BARS/VM in an automatic way. The operator receives the same mount message as for BARS/VM backup tapes, with the tape pool name as defined in the installation parameter, EDBRPOOL.

As with the tape pool processing for backup, tape units may be premounted and attached to the subusers using any of the virtual addresses in the normal CMS tape range (180-187 and 288-28F). For disaster backup tapes BARS/VM also supports the Automatic Cartridge loader (ACL feature) of 3480/3490 tape units. This can be done through the introduced concept of a logical scratch tape pool and is the recommended way to use it if ACL units are available.

4.16 Catalog Disks and Files

4.16.1 The Backup Catalogs: User ID Cuu and User ID Filepool Files

When BARS/VM backs up a minidisk it must save information about the backed up data and its relationship with the source data. The backup information for each minidisk is maintained in a separate file whose file name is the user ID of the minidisk owner and whose file type is the minidisk address or cuu. These files are therefore known generically as User ID Cuu (UC) files. For example the backup control data for the 191 minidisk of user ID EDUCATN would be in a CMS file called EDUCATN 191. The UC files reside on the 191 minidisk of the BARS/VM Interface user ID. Because of their vital importance to BARS/VM they are copied for backup to the 192 and 193 minidisks of the Interface user ID. This 191 is copied to the 192 at the start of a backup cycle, and the 191 is copied to the 193 at the end. It is **strongly** recommended that the 191, 192 and 193 minidisks be placed on separate **real** volumes to avoid complete loss of the catalogs in the event of DASD problems. This would, of course, mean complete LOSS of the ability to restore or retrieve any data via BARS/VM

UC files consist of a header record containing general information about the minidisk or directory, for example for a minidisk, its size and format. Then, for CMS minidisks and SFS directories, there is a record for each backed up file giving its name and the location of each backup copy. For non-CMS minidisks there is one record for each backup copy with the information about the location of the backup data and statistical data.

UC files which have been created by the backup of a remote system have a one-character node name abbreviation following the cuu in the file type. See chapter 5.3, “Remote Backup and Retrieval” on page 38 for a discussion of remote backup.

The SFS implementation makes the backup of a directory correspond to the backup of a minidisk. A UC file exists for every backed up directory containing backup information for each file in the directory. The file type of the UC file is _dddddd, followed possibly by a one character node name abbreviation. The value dddddd is a 'directory name abbreviation', internally generated, with

characters in the ranges 0-9 and A-Z which therefore allows each user ID up to 2,176,782,340 backed up SFS directories.

To equate the directory name abbreviation to a real directory name an additional type of file is required. There is one such file for each user ID who has backup in each file pool. These files are known generically as User ID Filepool (UF) files. The file name of the file is the user ID. The file type is +fff where fff is a 'file pool name abbreviation', again internally generated by BARS/VM. Each file pool which is specified for backup by the FILEPOOL command is assigned such an abbreviation.

4.16.2 Archive Catalog: USERID ARC Files

Corresponding to the USERID CUU files for backup there are a set of files which serve much the same purpose for archive. These are known as USERID ARC files. In this case the relationship between an archived file and the original minidisk or SFS directory on which it resided is not maintained and so all files have a file type of ARC. The USERID ARC files reside on the 197 minidisk of the BARS/VM Interface user ID. Because of their vital importance to BARS/VM they are copied for backup to the 198 and 199 minidisks of the Interface user ID. This 197 is copied to the 198 at the start of a backup cycle, and the 197 is copied to the 199 at the end. Again, it is **highly** recommended that the 197,198 and 199 minidisks be placed on separate **real** volumes.

USERID ARC files consist of a header record containing general information about the user such as his department name, account number and how much data he is allowed to archive. Also there is a record for each archived file giving its name and the location of each archive copy.

4.16.3 Migration Catalog: USERID MIG Files

Whereas USERID ARC files record archived files, USERID MIG files record archived minidisks. Each USERID MIG file records information about all the archived minidisks belonging to a particular user ID. These files, and the disks they record, are processed by the MDISK commands.

Like the USERID ARC files, the USERID MIG files are stored on the 197 minidisk of the BARS/VM Interface user ID. They are also backed up to the 198 and 199 minidisks of the Interface user ID at the beginning of each backup cycle.

The files consist of a header record containing general information such as the number of PUTs and GETs, the dates of the last PUT and GET, and the total number of 2K blocks of minidisk archive storage used by this user ID. There is also at least one record for each archived minidisk, giving details such as where the archived data is stored and when it was put there.

4.16.4 Archive Catalog: Archive DEL Files

A user may delete an archived file either explicitly by using the ADELETE command or implicitly by issuing an APUT which causes a file level to be replaced. To protect users against accidental erasure of archived files, each time an archived file is deleted the relevant information from the USERID ARC file is saved in a file which has the file type of DEL and a file name of the current date. Thus the information about the deleted file is still available so that it may be retrieved from the archive volume, provided a reorganization has not taken place, or from the backup of an archive minidisk. A DEL file is erased after it has been in existence for a number of days equal to the ERASDAYS parameter.

The DQUERY command displays information in the DEL files for a user and the DRECOVER command may be used to recover files whose descriptions are in those files.

4.16.5 Disaster Backup Catalog: USERID CUU Files

When BARS/VM backs up a minidisk it must save information about the dumped data and its relationship with the source data. The disaster backup information for each minidisk is maintained in a separate file whose file name is the user ID of the minidisk owner and whose file type is the minidisk address or cuu. These files are therefore known generically as USERID CUU (UC) files. For example the disaster backup control data for the 191 minidisk of user ID EDUCATN would be in a CMS file called EDUCATN 191. These UC files reside on the 2A1 minidisk of the BARS/VM Interface user ID. Because they are needed to provide a BASE/DELTA dump logic, the 2A1 minidisk is not excluded from the BARS/VM backup process like the other Interface user ID minidisks are.

USERID CUU files consist of a header record containing general information about the minidisk such as its size and format. Then, for CMS minidisks, there is a record for each backed up file giving its name and the file number on the disaster backup tape. For non-CMS minidisks there is one record for each backup copy with the information about where the backup data is, and statistical data. In comparison to the backup USERID CUU files disaster backup USERID CUU files do not contain many logical records into one physical record and contain a maximum of 3 levels which are BASE, DELTA and LSET.

4.17 BARS/VM Recovery

4.17.1 Recovery from Program Checks and Abends

When a BARS/VM subuser suffers a program check or abend, it may be automatically restarted, depending on the value of the ERESTART installation parameter. If this parameter is non-zero and the failing user ID is not the backup controller then any resources held by the subuser are freed and it is restarted. To prevent recurring program checks, the number of such restarts is limited to the value of ERESTART. If a controller user ID, for example the backup controller, or Interface abends, an attempt is made to completely restart BARS/VM. The Interface machine does this by forcing all subusers and issuing an IPL command for CMS to restart the BARS/VM system. To avoid a possible loop, this complete restart is only performed once. On the second restart, Interface is placed in a disabled wait state.

4.17.2 Recovery from Subuser LOGOFF

When a BARS/VM subuser is forced off, for any reason, the Interface virtual machine detects the fact and attempts to recover. If the subuser is not a controller then any resources it owns are freed and it is logged on again. If the subuser is a controller, a complete restart of the system is attempted. As for the case of a controller abend, this restart is only attempted once.

4.17.3 Recovery from System Failures

When the BARS/VM Interface machine receives a command and a CMS formatted 196 minidisk exists, it writes the command as a CMS file to that disk. The file is erased after the command is sent for processing to a subuser machine. In the event of BARS/VM being terminated and restarted, for example after a system failure, the commands which are saved on the 196 minidisk are processed when BARS/VM initialization completes. Commands which were in progress when the failure occurred are lost.

If no 196 minidisk exists then, if a failure occurs, all unprocessed commands are lost.

If a timer-driven process, such as incremental backup, was in progress when the system failed, it is restarted when BARS/VM restarts provided the time is within the time interval defined for that process.

Chapter 5. BARS/VM Special Considerations

5.1 BARS/VM in a Shared Data Environment

In some environments, data is shared between two or more processors. In this situation, backup and archive should only be performed from one of the processors but commands requiring only read-only access to the BARS/VM minidisks, such as RETRIEVE and AGET, can be processed on any system. To set up such an environment there will be a BARS/VM system with normal read-write access to its minidisks and this system does backup and archive. This is called the host system.

The other processors run BARS/VM but should have the ESLAVE option set to YES in the installation options (see “General Options” on page 226). When ESLAVE is set to YES, the system assumes it has only read-only access to BARS/VM minidisks, and that a host BARS/VM is installed as defined in EHOSTID and EHOSTIU. Since the minidisk definitions are the same on all processors, BARS/VM has to avoid multiwrite links over more than one processor.

When a command is received on a shared processor a test is made to see if it may be executed on this processor. All allowed commands are executed normally, except that Interface 191 and 197 are linked in read-only mode instead of read-write and instead of 195 a 295 minidisk is linked when necessary. Interface uses for command queueing purposes a 296 instead of the normal 196 minidisk.

All other commands are constructed into a command file via HDFSND and sent to the host BARS/VM as a remote command. This removes the burden from the user of identifying which command is processed locally or remotely.

APUT and BPUT with a file are supported in this way. A message is sent to the user telling him that the command has been routed to the host BARS/VM.

So, to run a shared BARS/VM system, the host is set up as normal and the other BARS/VM systems are set up with ESLAVE set on, EHOSTID and EHOSTIU defined and a 295 and 296 minidisk defined in the Interface machine's directory to be used instead of 195 and 196.

Note: If you run more than 2 processors shared, that means, you have more than one SLAVE BARS/VM system, all SLAVE BARS/VM Interface user IDs must have different 295 and 296 minidisks defined to avoid multiple write on those minidisks.

Note: Never specify EHOSTID for the MASTER BARS/VM system. It will then assume that it is a remote (DRONE) BARS/VM system.

5.2 BARS/VM in a Single System Image Environment

BARS/VM SSI (Single System Image) is another way to do backup processing when data is shared between two or more processors. In this situation, backup and archive will be performed from any of the processors defined to be a member of the BARS/VM SSI complex.

To define processors to become a member of the SSI complex, they should have the value ESSI set to YES in the installation options (see “General Options” on page 226).
Also all Interface user IDs in the complex must be made known to all systems by the NODENAME command with option SSI.

Backup is done on real volume base, the backup node is defined using the PRIMARY command. This command is only to be entered on one of the BARS/VM SSI processors.

Backup for SFS may be divided on a file pool basis by means of the FILEPOOL command which defines a file pool to be backed up.

When the ESSI is set to YES, the system assumes it shares the backup and archive work among some processors. No backup volume will be linked from another system, but information is stolen from Interface 191 (PRIMARY VOLUMES) and 197 (Archive information) minidisk.

From now on, every user in the BARS/VM SSI complex can access his backup or archive data in a transparent way without the need to know where his data resides.
Most privileged commands (like queries, SHOWDISK and even disaster backup) do not know about SSI and act exactly like they have done before.

5.3 Remote Backup and Retrieval

5.3.1 Definitions

BARS/VM supports distributed data processing by allowing backed up files to be sent from one node in a network to another for storage. The type of network envisaged consists of a central system, called the host, to which are attached the distributed systems, the drones[1] . The host is a VM system running BARS/VM with the capacity to back up a large amount of data. A drone is a system large enough to run VM and BARS/VM with locally attached DASD which can be used by BARS/VM to store backup data before transmission to the host. Tape backup at the drone is not supported.

A drone may have no skilled DP personnel and could either be operated by a user or have no operator. However, a drone may also be a large VM system with a substantial backup requirement where it is desired to back up or archive files remotely for disaster recovery or because the host site has access to the necessary backup storage media.

[1] drone - one who lives on the labour of others.

The unit of data transmission between the drone and the host is the file as this has been proved to be superior to the minidisk or logical or physical records.

5.3.2 Backup at the Drone

The drone is running BARS/VM with enough online DASD to store at least one day's backup data. Normal BARS/VM backup is carried out at the desired time of day to backup data on local DASD. During backup minidisk reorganization, files which have been backed up for at least an installation-defined number of days (EHDAYS in (see “General Options” on page 226) are sent to the host. Later reorganizations will then make available the space previously occupied by those files. If a reorganization is started by using the START REORG command, the value of EHDAYS can be ignored by using the SENDHOST option. This provides a way to force the sending of files to the host.

There are two situations where files are not sent to the host during reorganization. For CMS minidisks a backup file is not sent unless all higher levels have already been sent. For example a level 1 backup may be because of be sent but will only be sent if the level 0 is already backed up remotely.

The other situation concerns non-CMS minidisks. Backed up data for a non-CMS minidisk is not sent to the host if the backup frequency of the minidisk is not greater than the value of EHDAYS divided by EOSDAYS. This is because there is little point is sending a backup version to the host if the frequency of backup is such that the version will very soon become redundant.

To ensure the regular reorganization of backup minidisks and thus the sending of files to the host, a backup minidisk at a drone is marked full if at least EHDAYS have elapsed since it was last reorganized if it has not become full before that time. The frequency specified in EHDAYS will depend on the availability of drone DASD and how much backup data the installation wishes to maintain at the drone. When a file has been successfully received at the host, an acknowledgment command is sent back to the drone. When this command is processed, the UC file at the drone is updated to show the file's new location.

When a file is sent to the host, header records are added to it containing a BPUT command which contains all the information required to process the file. This BPUT is processed in the normal way by the host BARS/VM system Subuser machines. Successful completion of the BPUT results in a BLAH command being sent to the drone. This command, when executed at the drone, marks the file as backed up at the host.

When data is backed up at the host, a UC file is maintained for the drone minidisk or SFS directory to which the files belong. Thus information for files backed up at the host is maintained both in the drone and host UC files. To ensure that minidisk information is set in the UC file at the host, BUMPA commands are sent by the drone to set size, block size, label and device type data for CMS minidisks. Since BPUT commands are used to send data, the host must have DASD backup space available to receive the files. Data can be spilled to tape if required.

Since the transmission of files to the host is part of the backup minidisk reorganization process, the specification of the frequency of this process (EREORG) requires careful consideration. A small value is essential to ensure that data is removed from the drone as soon as possible. See also chapter 12.1,

“Reorganization” on page 151 for a discussion covering the REORG process of BARS/VM.

5.3.3 Purging Files

Since the host has no access to the data at the drone it must be informed directly when backup data is no longer required. When doing backup, the drone is able to determine that a particular file is backed up at the host but that the backup data is now redundant. In such a case, a BDELETE command is sent by BARS/VM at the drone to BARS/VM at the host.

5.3.4 Retrieving Files

If backup data is available at the drone, i.e. it has not yet been transmitted to the host and erased, then a RETRIEVE or RESTORE command for that data is processed locally. If the required data is at the host it is obtained from there by transmitting the appropriate RETRIEVE commands to the host.

5.3.5 BARS/VM at the Host

One important consideration when planning backup to a remote host is the updating of the UC files on the Interface 191 disk. A conflict arises because the Backup Controller normally has the 191 minidisk of the BARS/VM INTERFACE user ID allocated to it during the backup process and therefore the other subusers are unable to update it. For this reason, remote files are not processed while host local backup is in progress. It is recommended that the host system have a separate BARS/VM system running to deal with remote backup. This BARS/VM will carry out only a small amount of local backup and so the conflict problem with the 191 minidisk will not occur; the Subusers are always available, regardless of the work done in the local BARS/VM Also, this technique results in a separate set of backup minidisks from the local backup system, allowing much faster retrieval than if the backup was mixed within a single set of backup minidisks. Another benefit is the ease with which the backup data may be moved to a different host system if the present one were to become overloaded. It may be advisable to have more than one BARS/VM system handling remote backup if there are many drones to service.

Typically then, the host VM system would be running a BARS/VM system to receive files from each drone supported and a further BARS/VM to perform backup of the local host data. These two BARS/VM systems must be completely separate except they may share the Interface 194 minidisk which contains the BARS/VM code and which is never modified by BARS/VM There would be no need for Interface 197, 198 and 199 minidisks if archive functions would not be performed.

Each BARS/VM system which supports a drone must do local backup for at least one minidisk. This is so to ensure that the normal mechanism of backup minidisk reorganization takes place, ensuring that space occupied by redundant backup data is reclaimed. The directory file which is defined to BARS/VM by the DIRECT command should contain at least the definitions for all the BARS/VM virtual machines, i.e. Interface, Subusers and the NOLOG user identification.

If a Host BARS/VM system is to receive files from a drone for backup, the node name of the drone and the name of the BARS/VM Interface virtual machine running at the drone must first be defined to the host through the NODENAME command. See the description of the command in 10.2.23, “NODENAME” on page 121.

Part 2. 'User's Information'

Chapter 6. Using the BACKUP Function

6.1 Incremental Backup and Retrieval

This chapter is intended to provide information about the backup facilities of BARS/VM for the general user.

Backup is defined as the making of duplicate copies of CMS files, on CMS minidisks and Shared File System (SFS) directories, and non-CMS minidisks by the installation. The purpose of BARS/VM backup is to protect users against loss of data because of hardware failure or accidental erasure. The backup is managed automatically by BARS/VM. Retrieval of backed up data is normally the responsibility of the user who owns that data and commands and a full screen menu facility are provided to allow this.

The BARS/VM incremental backup function operates automatically each night. The frequency of backup of a particular minidisk or SFS directory may be set by the owner through the BACKUP command. For CMS data, any frequency up to 99 days may be set. However, all CMS data is backed up by default and normally the user need take no action. The default frequency of backup for CMS data is set by the BARS/VM administrator.

For non-CMS minidisks, the system overhead in carrying out backup is much higher. For this reason the minimum frequency of backup allowed is normally set to a larger number of days than for CMS data. Also non-CMS data is not backed up by default, so a BACKUP command must be issued if backup is required. If an important change has been made to a non-CMS minidisk, the user may specify that backup is to be performed that night by issuing the BACKUP command with the TONIGHT option. Users should ensure that the non-CMS minidisks for which they are responsible are backed up at an interval which satisfies their requirements but minimizes the system resources used. The BACKUP command may be used to change the backup parameters for a minidisk.

BARS/VM maintains a number of backup versions which is set by the administrator. Available backup data may be displayed by using the QUERY command and backup data may be recovered using the RETRIEVE command. If the FILE option of the QUERY command is used, the resulting information is sent as a file called REPORT LIST, otherwise it is displayed as messages on the screen. Retrieval of backup data may be requested either overnight or immediately. A RETRIEVE command for a non-CMS minidisk restores the data directly to the minidisk and an acknowledgement file is sent to the command issuer to confirm the status of the retrieval. A RETRIEVE for CMS files results in the files being sent to the virtual card reader. If one or more files fail to retrieve, a file, REPORT LIST, is sent giving a complete list of the files that were to be retrieved and the status of each one. An example is given below.

```
RETRIEVE OF TOM      DIRECTORY
WINSFSS:TOM.
BARQUERY EXEC      RETRIEVE SUCCESSFUL
BARSTART EXEC      RETRIEVE SUCCESSFUL
BARSTAT  EXEC      RETRIEVE SUCCESSFUL
BARTRANS EXEC      RETRIEVE SUCCESSFUL
BGET     EXEC      RETRIEVE FAILED  RC = 0028
BRAND    EXEC      RETRIEVE SUCCESSFUL
```

This file is not sent if all files are successfully retrieved. The return codes given in REPORT LIST result from errors either loading or copying files, for example,

```
Error loading file from tape                    RC = 3
The file was not found                          RC = 28
The file is too large to process                RC = 255
```

Any other return codes suggest problems with BARS/VM or CMS and the administrator should be notified.

When a file is erased from a minidisk, or when a minidisk is deleted, BARS/VM retains the backup data for an installation-defined period. After this time the backup is erased and retrieval is no longer possible.

6.2 BPUT and BDELETE

CMS files and non-CMS minidisks may be backed up at any time using the BPUT command. Since data is backed up regularly through the incremental backup process there should not be a need to use BPUT very often. However it may be useful after critical changes have been made. Care must be take however since BPUT will create a new level zero backup which will cause one of the old levels to be lost. So there is a danger of losing backup versions created by incremental backup which might be required for recovery.

The BDELETE command is used to delete backup data. Again this is only needed in special circumstances since backup will be automatically erased when it is not required. However there are occasions when immediate deletion is required such as when a minidisk is being transferred to another user and some files on the minidisk and the backup of those files are not to be given to the new user.

6.3 RESTORE

The user can restore a specified Shared File System (SFS) directory, file or alias with the last backup copy through the RESTORE command. When an SFS directory is restored, the directory is created and all files and aliases are restored that existed at the time of the last backup cycle. The sub-directories may also be restored. Directory and file authorizations are regranted unless specified otherwise.

Files and aliases may be restored directly into an existing SFS directory by giving a file specification.

The RESTORE command may also be used to restore a complete minidisk to the state it was in at the last backup.

Chapter 7. Using the ARCHIVE Function

This chapter is intended to provide information about the archive facilities of BARS/VM for the general user.

The purpose of BARS/VM archive is to minimize the use of online storage for CMS files which are not required for the work currently being performed. It provides a way for users to make copies of files on volumes managed by BARS/VM and to be able to retrieve those copies when required. While files are held on archive storage the original versions may be erased, thus making space available for new files.

Files are archived by the APUT command. In this command is specified the cuu of the minidisk or the SFS dirid where the file or files reside (e.g. 191) and the name of the file or files to be archived. More than one file may be archived with a single command by using the * and % notation as described in Chapter 8, “User Commands Description” on page 51. BARS/VM copies each file to a suitable archive minidisk giving it a unique name. This consists of a file name which is the current date and a file type of a sequence number unique to that date. To copy data BARS/VM links and accesses the user's minidisk or accesses the user's SFS directory in read-only mode and copies the requested files in packed format.

Files are retrieved from archive with the AGET command. BARS/VM locates the required file, unpacks it, and sends it to the requester's card reader in NETDATA format. The requester can then load the files using RECEIVE. As for APUT, more than one file may be retrieved with a single command by using the * and % notation as described in Chapter 8, “User Commands Description” on page 51. If the installation uses tape to hold archived files there may be delays during AGET processing while tapes are mounted.

An alternative way of archiving files is available for the situation where BARS/VM would be unable to copy the file, for example when it resides on temporary disk or if the archive is to be performed on a different system using the TONODE parameter on the BARS/VM command. For such cases a single file mode letter may be specified instead of a minidisk cuu. The file is then sent to BARS/VM for archive through DISK DUMP to its card reader.

A full screen interface to all of the archive commands is provided as an alternative to the line commands. An initial panel guides the user to the appropriate command where the possible operands are displayed.

7.1 Deferred Processing and APUT/AGET Completion

The process of archiving files or retrieving them from archive may be carried out immediately on demand or may be deferred to the next normal BACKUP cycle, usually overnight. It is preferable, whenever possible, to defer the process and thus minimize the impact on the VM system during prime shift and so limits are imposed on how many files may be sent to and retrieved from archive immediately. The precise values of these limits are decided by each installation and may be displayed using the DISPLAY OPTIONS command. Deferred archive and retrieval takes place at the beginning of BARS/VM BACKUP cycle.

Deferred commands which have been entered but not yet processed may be displayed by using the DISPLAY DEFERRED command. Should it be required to cancel a deferred AGET or APUT then the CANCEL option of the appropriate command is used.

On completion of an APUT, whether deferred or immediate, the requester is sent a file which describes the result of the APUT. This file has a file type of EXEC and a file name of the user ID under which the files were archived. The file is in CMS EXEC format and may be used to erase the files which were successfully archived. If some files fail to archive they are included in the EXEC but have an asterisk in column 1 and are marked ARCHIVE FAILED. An example of such a file which might be received after the successful completion of an APUT for all EXEC files is given below. This is followed by an example where one of the files failed to archive.

```
&1 FREE     EXEC     &2
&1 PROFILE  EXEC     &2
&1 SCRIP    EXEC     &2
&1 VRDRDEF  EXEC     &2
&1 VUTP     EXEC     &2


&1 PROFILE  EXEC     &2
&1 PROFILE  XEDIT    &2
*  LASTING  GLOBALV    ARCHIVE FAILED  RC =     0013
```

When the EXEC is called, the &1 and &2 are replaced by the first and second arguments. For example, if I receive a file called TOM EXEC, I can erase the files that have been archived by typing

```
EXEC TOM ERASE
```

or if they were on, say, my B disk

```
EXEC TOM ERASE B
```

It is very useful if many files have been archived and is similar to the function provided by the CMS LISTFILE command with the EXEC option. Lines that start with * are ignored and so files that failed to archive would not be erased in the above example. The return codes given in the EXEC file result from errors either receiving or copying the file.

```
   The file has an unknown record format (not F or V)    RC = 1
   Error receiving the file                              RC = 2
   Error copying file to tape                            RC = 3
   No space is available                                 RC = 13
   The file was not found                                RC = 28
   The file has an incorrect format                      RC = 255
   The file is too large to process                      RC = 255
```

Any other return codes suggest problems with BARS/VM or CMS and the administrator should be notified.

If an error occurs within BARS/VM during the execution of a deferred APUT or AGET command the requester is sent a file containing the command which was received and an indication of how to determine the cause of error that has occurred. These explanatory files are called ARCHIVE ERROR and AGET ERROR respectively. Below is an example of an ARCHIVE ERROR file. In this case the file NEW FILE did not exist in the specified Shared File System directory.

```
You issued the following BARS/VM command for deferred execution -
APUT     .    NEW      FILE
( LEVEL    CLASS  A RETAIN 001 TAG            FROM           FOR
This command failed. To obtain further information issue following command -
HELP MESSAGE HDF0338I
```

7.2 User IDs and Defining

Archived CMS files are associated with a particular **user ID** which is a one to 8 character name which identifies the owner of the archived files. This will normally be the same as the VM logon user ID of the person who archived the data. However this need not be so and, in fact, the user ID need not be an actual VM user ID. It is recognized that frequently files from different minidisks of different VM users may be required to be archived and associated with each other and under a single name thus creating what is sometimes called a **pool** of data. So the name under which the data is held is sometimes called a pool user ID or name.

Users may always archive data under their own VM user IDs but to archive data to such a pool they must be authorized to do so by someone who has special BARS/VM authority, normally the person who administers the users minidisks.

Because user IDs or pool names may be retained when persons move to different groups or departments, BARS/VM requires extra definition of the owners of archived data. Before data can be archived under a user ID or pool name, the ADEFINE command must be issued to give BARS/VM the name and also a department and account identifier to associate with that name. The account identifier is included to allow an installation to charge for archive space. BARS/VM saves the information from the ADEFINE command but does not verify it in any way. It is left to the installation to decide what to do with data associated with invalid departments or account identifiers.

The is a special consideration in a shared system (SSI) environment. In such a BARS/VM environment there may be the same user ID on several systems with the same owner or with different owners or different work. The ADEFINE command, through the SHARED and NONSHARED options allows the user to define whether there is to be a single archive or a separate archive on each system. If there is a single archive, all archive commands are routed to the node where the first ADEFINE was processed. All other ADEFINEs affect only that single archive in a BARS/VM coupled system. That should be the normal way of working for most users who don't know where they sign on and who have access to their minidisks on all systems. However ADEFINE also allows different sets of archive in such a complex. In this case only the local user ID gains access to the archive data.

7.3 Special ARCHIVE Commands

The ADEFINE, AGET and APUT commands have already been introduced in the discussions above. Once a user ID has been defined to BARS/VM, files may be archived under that user ID name and retrieved when required. Note that when an archive copy is made of a file it is not erased from the source minidisk, nor is the archive copy erased when an AGET is performed. To erase archived files the ADELETE command may be used. All file levels or only specified levels may be deleted. If an ADELETE command is used to delete all archived files for a user

ID, then an ADEFINE must be issued before files can be archived again for that user ID.

The files archived for a particular user ID or pool name may be listed by the AQUERY command. Since there may be many files it is normal to limit this display in some way. A single file may be displayed or the * and % notation may be used. Additional keywords on the AQUERY command also limit the display. For example, only files with a particular tag may be listed. See 7.4, “Archive Levels, Classes, Tags and Retention Dates” for a discussion of tags.

The AALTER command may be used to alter information associated with already archived files. The retention period may be changed as may the class and tag. See 7.4, “Archive Levels, Classes, Tags and Retention Dates” for a discussion of classes and tags. The ARENAME command can be used to rename archived files.

If a command has been sent to BARS/VM, the status of the BARS/VM system may be displayed to determine whether the command is being processed or is currently queued awaiting processing. The command DISPLAY STATUS QUEUES will display the queue of commands waiting for processing and DISPLAY STATUS USERS will display the status of each of the BARS/VM virtual machines and the work they are performing.

7.4 Archive Levels, Classes, Tags and Retention Dates

A facility which is important for the user of the archive system to understand is that of **levels** of archived files.

A user may create several levels of archive for the same file name, up to a maximum defined by the installation. These levels are defined as 0, 1, etc.. The user is allowed initially a number of levels corresponding to the default for the installation. The permitted number of levels may then be changed for a specific user by an authorized command issued by the BARS/VM administrator. If a user does an APUT without specifying LEVEL, the archived file is assigned a level of 0. Another APUT without LEVEL would make the previously archived file level 1 and the new one level 0. If LEVEL is specified on the APUT command, the newly archived file is assigned the specified level, unless doing so would create a gap in the levels. Here the lowest available level number smaller than the one specified is assigned.

As an example, APUT X EXEC would create an archived file with a level number of 0 (call it X0). Another APUT would make this archived file level 1, i.e. X0 is now X1, and create a new X0. If the command is now repeated but specifying LEVEL 1 then the level 1 file, X1, is deleted and a new X1 created. The level 0 file, X0, is untouched. If LEVEL 2 is specified, a new level 2 file, X2, is created. If, however, LEVEL 3 were specified instead of LEVEL 2 and since there is no level 2 the newly archived file would be assigned level 2 to avoid gaps in the levels.

An archived file is always assigned a security **class**. This is a single character whose precise meaning is defined by the installation, one particular character meaning UNCLASSIFIED, and so on. Users may specify a class on the APUT or AALTER commands and there is an installation default if it is omitted. The DISPLAY OPTIONS command will display this default value.

Tags may be assigned to archive files to provide additional information or to define logical group of files within the ARCHIVE. The AQUERY command may be instructed to list only those archived files which have a particular tag. The tag is a character string with a length of up to 8 characters. Since files may be archived for long periods of time it may be useful to use the tag mechanism as a aid to recalling the reason for archiving the file. Alternatively some checkpoint may be reached and a set of files archived with the same tag so they are thus related to each other within the archive system.

A **retention date** is always assigned to an archived file. This may be done explicitly when the file is archived. If none is explicitly given an installation default is given. When a file expires there are two alternative possibilities. If the DELEXP option was specified on ADEFINE, a file is erased when it expires. If not, the file is not erased and it is left to the administrators of the archive system to erase archived files. Various tools are available to them to identify and process expired files. If a user chooses not to use DELEXP he should be aware of the installation's policy in this matter.

7.5 DQUERY and DRECOVER

Archived files may be deleted by the user either explicitly by means of ADELETE or implicitly through an APUT which replaces a file. To protect users against accidental erasure of archived files BARS/VM saves information about any deleted file for an installation-defined number of days. During this period an erased file may be recovered by using the DRECOVER command. The DQUERY command may be used to list files which have been deleted from archive but for which information has been retained by BARS/VM to allow recovery. Note that the information is saved at the time the ADELETE or APUT command is processed causing the erasure of the file and so the file may be recovered immediately. The setting of the various installation options may be displayed by using the command DISPLAY OPTIONS.

Chapter 8. User Commands Description

8.1 Issuing BARS/VM Commands

Commands are sent to BARS/VM by the VMBARS EXEC. To issue a command to the local BARS/VM the EXEC name VMBARS is used followed by the command, e.g. VMBARS QUERY 191. A command may also be sent to a remote node by specifying VMBARS TONODE followed by the node name and command, e.g. VMBARS TONODE VMSYS1 QUERY 191. In this case the command is executed at the remote node (VMSYS1) and resulting messages and files are sent back to the issuing node.

You may issue all of the BARS/VM functions described in this chapter by using the BARS/VM full screen interface. This interface provides a selection panel and some input menus to enter the parameters required for a specific function, so there is no need to know the several command formats for the end-user. An ONLINE HELP display assists you in determining the appropriate input parameters.

The interface is called by issuing the VMBARS EXEC without operands.

BARS/VM will only allocate one subuser at a time to process commands for a general user. This prevents one person from using up all the subusers, and thus locking out all other users.

8.1.1 Generic Specification

In some commands a generic name may be given for such things as file names or tags. "*", or "%" may be specified in the name to indicate a set of items, usually files, rather than a single one. The special character "*" indicates that any number of any characters will be accepted as a correct match. The special character "%" indicates that only a single character in the position represented by the "%" will be accepted as a correct match. The two special characters may be combined in any order. Two simple examples may serve to clarify the use of the special characters. B* represents a set of character strings consisting of any number of characters starting with the character B. B% represents a set of character strings consisting of two characters, the first of which is B and the second of which may be any character.

8.1.2 SFS Directory Specification

Some commands allow the specification of a Shared File System directory name or "dirid". This can have the general form:

```
<filepoolid:><user ID>.<n1<.n2<.n3<.n4<.n5<.n6<.n7<.n8>>>>>>>>
```

where filepoolid gives the ID of the file pool in which the directory resides and n1 to n8 are directory names. At least a . must be specified. If the file pool ID is not specified, the default that you set with the CMS SET FILEPOOL command is used. The user ID is defaulted to the virtual machine issuing the command. The directory can also be specified in the form:

```
+fm.n1<.ni+1...ni+7>
or
-fm<.ni.ni+1...ni+7>
```

where these have the same meanings as described in the CMS command reference.

8.1.3 How to Read a Syntax Diagram

Follow the path of the line from left to right, from top to bottom.

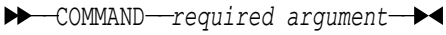
The definition of a command begins with two arrowheads pointing to the right.

The definition of a command ends with two arrowheads pointing to each other.

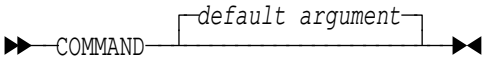
An arrowhead pointing to the right at the end of a line means that the definition is continued below.

An arrowhead pointing to the right at the beginning of a line means that the definition is continued from above.

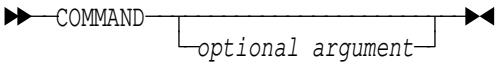
A required item is on the main path along the horizontal line.



A default item is above the main path.

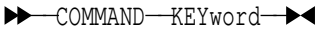


An optional item is below the main path.

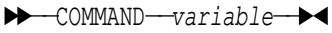


An item is a keyword, a variable, or a reference to a fragment in a syntax definition.

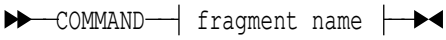
Keywords are shown in a Gothic font with the minimum abbreviation in uppercase.



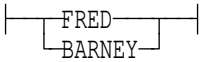
Variables are shown in lowercase italics.



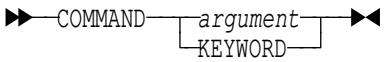
A reference to a fragment of a syntax definition breaks the main path with vertical bars. The fragment is defined later in the diagram.



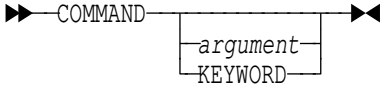
fragment name:



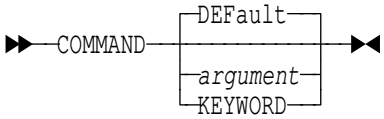
When you must choose between two or more items, they are stacked with the first one on the main path.



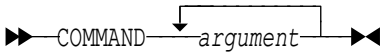
When you can select an item or take none, the choices are stacked below the main path.



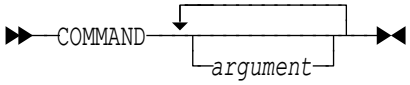
A default is shown above the main path.



An item may be repeated when an arrow returns to the left in front of it. The item is on the main path when you must write it at least once.



The item is below the main path when you may omit it altogether.



8.1.4 User Commands Summary

Following is a complete list of BARS/VM functions available to the general user:

Communication with Backup/Retrieve Processor

BACKUP	Controls backup parameters for minidisks and directories
BDELETE	Deletes files from backup
BLAH	Marks files backed up at a remote host
BPUT	Backs up files and minidisks immediately
QUERY	Requests information about backed up files and minidisks
RESTORE	Restores minidisks, SFS files and directories
RETRIEVE	Retrieves files and non-CMS minidisks from backup

Communication with Archive Processor

AALTER	Changes information for archived files
ADEFINE	Initializes archive processing for a user
ADELETE	Deletes files from archive
AGET	Retrieves files from archive
APUT	Submits files into archive

AQUERY	Requests information about archived files
ARENAME	Changes the name of an archived file
DQUERY	Requests information about deleted files in archive
DRECOVER	Makes deleted files available from archive

Communication with DBR Processor

DBR QUERY	Requests information about files in disaster backup
LPUT	Sends an LSET file to BARS/VM

Miscellaneous Functions

CQUEUE	Cancels, holds or releases commands which are queued but have not yet started executing.
DISPLAY	Displays general BARS/VM information
FILELIST	An alternative way of accessing the FILELIST screens for backup and archive.
HELP	Online HELP facility of BARS/VM

8.1.5 How to Authorize Users

Users may be authorized to issue BARS/VM commands on behalf of other users via the `OWNED` command. This can only be done by your BARS/VM administrator. Alternatively, if enabled by the BARS/VM administrator, authorization for some commands may be controlled via an External Security Manager. See 8.4, “Authorizing other users” on page 85 for more details.

8.1.6 Return and Reason Codes

Commands may give back return codes and reason codes. These may appear in acknowledgement files that are sent by some commands when something goes wrong. These files are called BACKUP ACK. The possible return code values are listed below.

```
 0 - successful completion
 4 - command not executed but not an error situation
 8 - syntax error
12 - bad value specified
16 - authorization error
20 - error obtaining or releasing storage
24 - backup or restore is in progress
28 - general error - see reason code and messages
32 - Callable Services Library error
```

Reason codes correspond to message numbers, and the explanation for a particular reason code may be found in the help for the corresponding message. For example, reason code 345 from APUT means the user is over his space allocation and the help for message HDF0345I explains this.

8.1.7 Dates

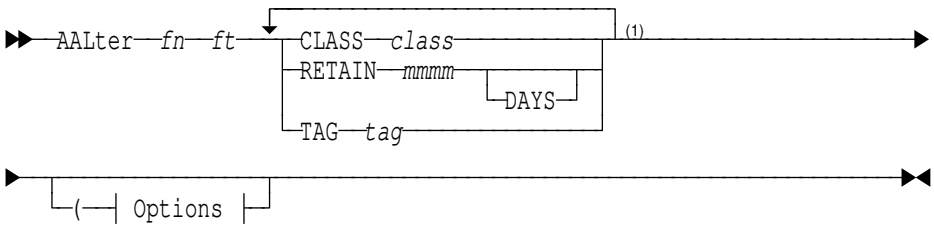
When dates are specified on commands, they are usually required to be of the form ddmmmyy, where dd is the day number, mmm is the mnemonic month and yy the year number, e.g. 10JAN96. Dates from 1JAN60 to 31DEC99 are taken to have years of 1960 to 1999 and dates from 1JAN00 to 31DEC59 are taken to run from years 2000 to 2059.

8.2 Command Syntax Description

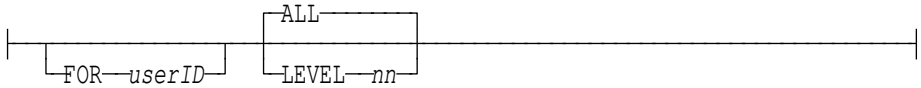
8.2.1 AALTER

The AALTER command allows the user to alter information associated with archived files.

General User Command



Options:



Note:
1 Each choice may only be specified once.

Operands

fn Specifies the file name of a specific file or set of files whose archive information is to be altered.

ft Specifies the file type of a specific file or set of files whose archive information is to be altered.

CLASS Specifies that the class of archived files is to be changed to that specified by *class*, which must be a single character.

RETAIN Specifies that the retention period is to be changed to *mmmm* months from now. If a file to be altered has been previously deactivated it is reactivated. *mmmm* may be any value from 1 to 999, but the actual maximum value is determined by the installation.

The optional **DAYS** keyword specifies that the RETAIN value be treated as days, rather than months. Again, it is limited by the installation maximum, but may be up to four digits.

TAG Specifies that the tags of archived files are to be changed to that specified by *tag*. The tag may not be more than 8 characters. The characters * and % are not allowed in the tag specification.

Options

FOR Specifies that the user ID whose data is to be changed belongs to the specified user ID rather than the user ID of the command issuer. The command issuer must have been authorized to do this.

LEVEL Specifies that only a particular archived file level (*nn*) is to be altered.

ALL Specifies that all levels of the specified files are to be altered.

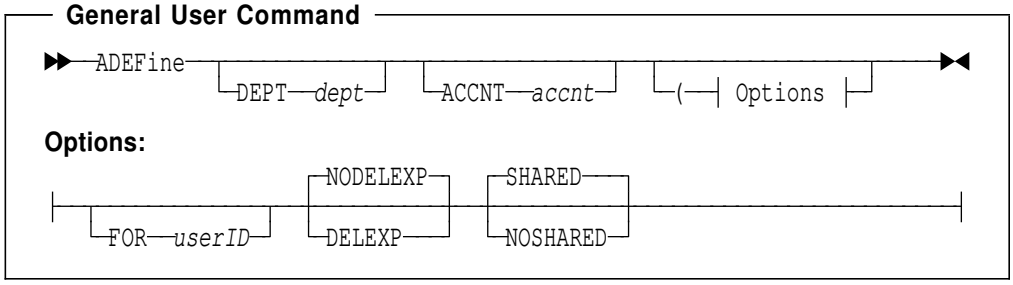
Usage Notes:

1. For *fn* and *ft*, the ***** and **%** notation as described in 8.1.3, “How to Read a Syntax Diagram” on page 52 is **not allowed** when using the command with the **RETAIN** operand.

See 8.3, “Sample BARS/VM Commands” on page 81. for some examples of the command.

8.2.2 ADEFINE

The ADEFINE command defines the potential owner of archived data via the department number or description and the account number. Each user ID under which data is to be archived must be first defined to BARS/VM so that data may be associated with a department or account number. When defining a new user ID, both DEPT and ACCNT must be specified. Data associated with an already existing user ID may be changed by specifying either or both of these operands. If only the options are to be changed, both DEPT and ACCNT may be omitted. The SHARED and NOSHARED operands apply to a shared system (SSI) environment only.



Operands

DEPT Specifies a one to 8 character department name or number.

ACCNT Specifies a one to 8 character account identification.

Options

FOR Specifies that the user ID to be defined is the specified user ID rather than the user ID of the command issuer. The command issuer must have been authorized to do this.

DELEXP Specifies that archived files may be erased by the installation after they have reached their expiry date.

NODELEXP Specifies that expired archived files may only be erased after they have been deactivated. This is the default.

SHARED NOSHARED

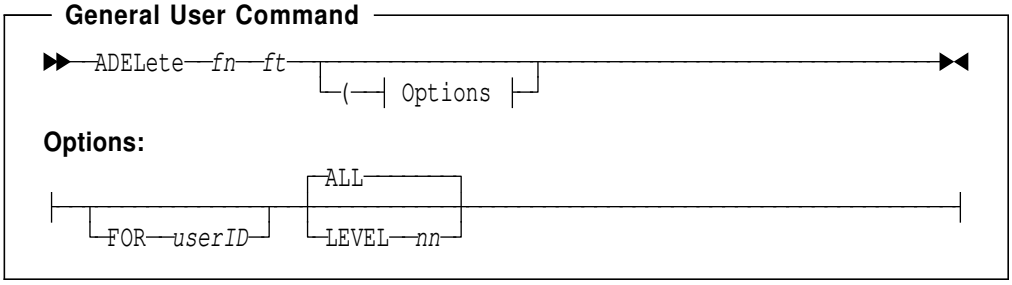
Apply to a shared data environment or SSI only. In such a BARS/VM environment there may be the same user ID on several systems with the same owner or with different owners or different work. The SHARED and NOSHARED options allow the user to define whether there is to be a single archive or a separate archive on each system. The default option in ADEFINE is SHARED, which means that the user ID has a single archive and all archive commands are routed to the node where the first ADEFINE was processed. All other ADEFINEs affect only that single archive in a BARS/VM coupled system. That is the way it should work for most

users who don't know where they sign on and who have access to their minidisks on all systems. NOSHARED allows different sets of archive in such a complex. Only the local user ID gains access to the archive data. When NOSHARED is specified on one system, all following ADEFINEs on other BARS/VM systems are reset to NOSHARED.

See 8.3, “Sample BARS/VM Commands” on page 81. for some examples of the command.

8.2.3 ADELETE

The ADELETE command is used to delete files from the archive system. Note that if the command is used to delete all files for a user ID, then an ADEFINE must be issued before files can be archived again for that user ID.



Operands

fn Specifies the file name of a specific file or set of files which are to be deleted from the archive subsystem.

ft Specifies the file type of a specific file or set of files which are to be deleted from the archive subsystem.

Options

FOR Specifies that the user ID whose data is to be deleted belongs to the specified user ID rather than the user ID of the command issuer. The command issuer must have been authorized to do this.

LEVEL Specifies that only a particular level of an archived file or files is to be deleted.

ALL Specifies that all levels of the specified files are to be deleted.

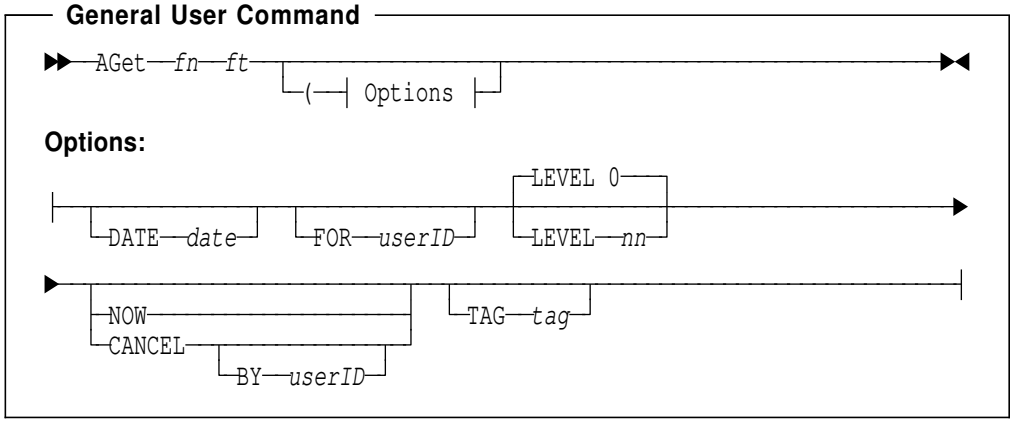
Usage Notes:

1. File name and file type may be specified using the * and % notation as described in 8.1.3, “How to Read a Syntax Diagram” on page 52.

See 8.3, “Sample BARS/VM Commands” on page 81. for some examples of the command.

8.2.4 AGET

The AGET command retrieves specified files from BARS/VM archive volumes. Individual archived files may be retrieved and these are placed in the command issuer's virtual card reader. The retrieves may be requested immediately or the operation deferred until a time determined by the installation. There are installation limits on how many files a user is allowed to get from archive during the day. Note that retrieved files are **NOT** erased from archive. Archived files are only erased explicitly with the ADELETE command. See Chapter 7, “Using the ARCHIVE Function” on page 45 for general information on the use of the AGET command.



Operands

fn Specifies the file name of a specific file or set of files to be retrieved from archive.

ft Specifies the file type of a specific file or set of files to be retrieved from archive.

Options

BY Allows administrators to cancel deferred requests submitted by the specified user ID.

CANCEL Cancels a deferred request.

Only the first matching request is cancelled. The *fn*, *ft* and **FOR** *userID* fields are compared. Other fields, including **LEVEL**, are ignored.

DATE Specifies that only archived files which were archived on the specified date are to be retrieved. The date must be of the form ddmmmyy, where dd is the day number, mmm is the mnemonic month and yy the year number, e.g. 10JAN96.

FOR Specifies that the archived data is associated with the specified user ID rather than the user ID of the command issuer. The command issuer must have been authorized to do this.

LEVEL Specifies the level number for the archived file to be retrieved. *nn* specifies the particular level. The default is level 0.

NOW Specifies that the command is to be processed immediately and not deferred. Deferred commands may be displayed by the DISPLAY command and cancelled by using the **CANCEL** option of AGET.

TAG Specifies that only archived files which have the specified tag are to be retrieved. *tag* may be up to 8 characters in length. A generic tag may be specified using the * and % characters.

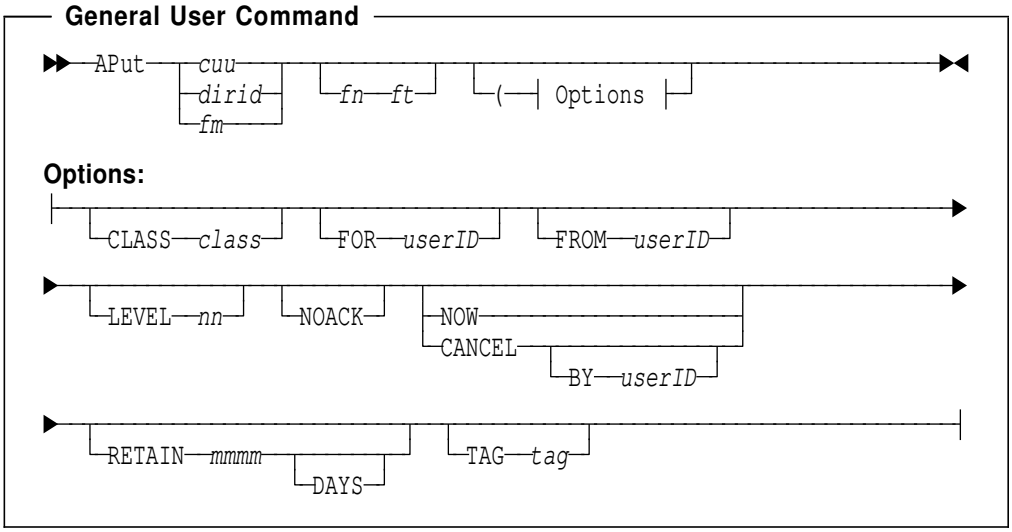
Usage Notes:

1. File name and file type may be specified using the * and % notation as described in 8.1.3, “How to Read a Syntax Diagram” on page 52.

See 8.3, “Sample BARS/VM Commands” on page 81. for some examples of the command.

8.2.5 APUT

The APUT command makes archive copies of specified CMS files. The copies may be made immediately or the operation deferred to an installation defined time of day. There are installation limits on how many files a user is allowed to archive during the day. Note that when a file is archived the original file on the user's minidisk or SFS directory is **NOT** erased. See Chapter 7, “Using the ARCHIVE Function” on page 45 for general information on the use of the APUT command.



Operands

cuu Specifies a minidisk on which the files to be archived reside. It must be specified as 2-4 characters.

dirid Specifies an SFS directory in which the files to be archived reside.

fm Specifies a single file mode letter defining the minidisk or SFS directory on which the file or files to be archived reside. If a file mode letter is specified for an accessed minidisk, the requested file or files are sent to BARS/VM using DISK DUMP instead of being copied from the minidisk on which they reside. This allows files to be sent to another node for archive. If archive is being done to the local BARS/VM system it is more efficient to use the cuu operand. If a file mode letter is specified for an accessed directory, the files are copied in the same way as if a dirid was specified. If the files reside on a temporary minidisk, then specifying a file mode letter is the only way of archiving them.

fn Specifies the file name of a specific file or set of files for which an archive copy is to be made.

ft Specifies the file type of a specific file or set of files for which an archive copy is to be made.

Options

BY Allows administrators to cancel deferred requests submitted by the specified user ID.

CANCEL Cancels a deferred request.

Only the first matching request is cancelled. The *fn*, *ft*, *cuu* or *dirid*, **FROM** *userID* and **FOR** *userID* fields are compared. Other fields, including **LEVEL**, are ignored.

CLASS Specifies a one character class for the archived files. The meanings of these characters are defined by the installation. For example, `CLASS C` could have the meaning **"CONFIDENTIAL"**. The class definitions may be displayed by using the DISPLAY CLASS command.

FOR Specifies that the archived files are to be associated with the specified user ID rather than the user ID of the command issuer. The command issuer must have been authorized to do this.

FROM Specifies that the files to be archived reside on a minidisk belonging to the specified user ID rather than that of the command issuer. The command issuer must have been authorized to do this. This option is not allowed if "dirid" is specified since the user ID to whom the files belong is taken from the dirid.

LEVEL Specifies a level number for the archived file. nn is the Level number - 0 is the newest, then 1, 2 and so on. If LEVEL nn is not specified, the files being archived become level 0 and the older versions increase by 1 - 1 becomes 2, 2 becomes 3, 3 becomes 4, and so on. If LEVEL nn is specified, the existing nnth level is erased and a new nnth Level created, and other levels are unaffected. If no level exists, the lowest available level number smaller than the specified one is used. *nn* may be a number from 0 to 99 but the number of levels which may be created are limited by the installation.

NOACK Specifies that BARS/VM should not return a spool file acknowledgment when the command has completed.
Warning: This option should be used with care, since the archive operation may fail, and this file is normally used to provide confirmation of success.

NOW Specifies that the command is to be processed immediately and not deferred. Deferred commands may be displayed with the DISPLAY command and cancelled by using the **CANCEL** option of APUT.

RETAIN Specifies a retention period for the archived files. mmmm is the number of months the files should be retained by BARS/VM. After this time the files will be deactivated if an AAUDIT command with the DEACT operand is issued by the administrator. Some time later the deactivated data may be erased by the administrator via the AAUDIT command specifying DELDEACT. Expired data will also be

deleted if the user specified DELEXP when issuing the ADEFINE command.

mmmm may be any value from 1 to 999 but the actual maximum value is determined by the installation. If RETAIN is not specified than the default value as defined by the installation is used. The optional **DAYS** keyword specifies that the RETAIN value be treated as days, rather than months. In this case, the installation maximum is converted from months to days (assuming 30 days per month) before the limit is checked. Four digits may be specified with the DAYS option.

TAG Specifies that a "tag" value is to be associated with the archived files to aid later identification. If specified, the tag can be up to 8 characters in length. (e.g. TAG docrel32 could be used to group together all documentation from REL.32). The characters * and % are not allowed in the tag.

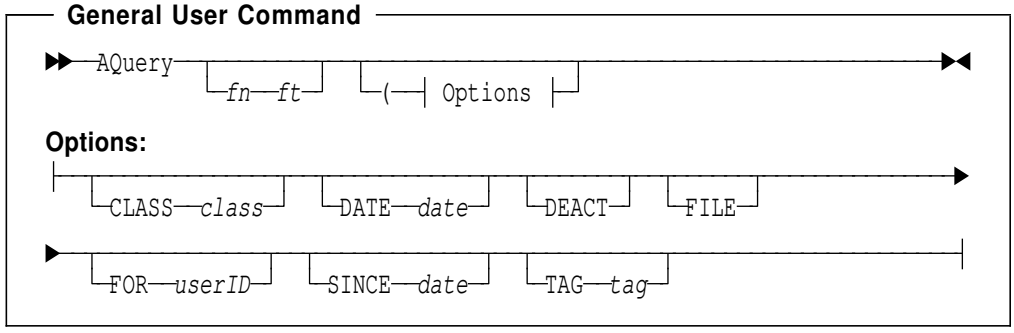
Usage Notes:

1. If the file name and type are omitted, then all files on the specified minidisk will be archived (i.e. equivalent to * *). However, only a one line acknowledgement file will be sent.
2. See 7.1, “Deferred Processing and APUT/AGET Completion” on page 45 for details of the acknowledgement file formats and return codes.
3. File name and file type may be specified using the * and % notation as described in 8.1.3, “How to Read a Syntax Diagram” on page 52.

See 8.3, “Sample BARS/VM Commands” on page 81. for some examples of the command.

8.2.6 AQUERY

The AQUERY command displays information describing archived files. The display may be of all archived files or may be limited in various ways.



Operands

fn ft Specify particular files.

If these are specified then BARS/VM displays additional information for the particular file or files specified.

Options

CLASS Specifies that only archived files which have the specified class are to be displayed. *class* must be a single character.

DATE Restricts the display of files to those files which were archived on the specified date. The date must be of the form ddmmmyy, where dd is the day number, mmm is the mnemonic month and yy the year number, e.g. 10JAN96. The DATE operand must not be used if SINCE is specified.

DEACT Specifies that only archived files which have been deactivated are to be displayed.

FILE Specifies that the result of the command is to be sent to the command issuer as a file to his card reader rather than being displayed at his terminal. The file is sent in NETDATA format.

FOR Specifies that the user ID to be queried is the specified user ID rather than the user ID of the command issuer. The command issuer must have been authorized to do this.

SINCE Restricts the display of files to those files which were archived on or after the specified date. The date must be of the form ddmmmyy, where dd is the day number, mmm is the mnemonic month and yy the year number, e.g. 10JAN96. The SINCE operand must not be used if DATE is specified.

TAG Specifies that only archived files which have the specified tag are to be displayed. *tag* may be up to 8 characters in length.

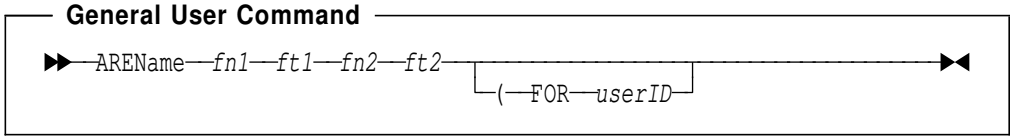
Usage Notes:

1. File name, file type and tag may be specified using the ***** and **%** notation as described in 8.1.3, “How to Read a Syntax Diagram” on page 52.
2. SFS aliases are not displayed if the SINCE or DATE option is specified.

See 8.3, “Sample BARS/VM Commands” on page 81. for some examples of the command.

8.2.7 ARENAME

The ARENAME command allows a user to rename archived files.



Operands

fn1 Specifies the file name of the archived file to be renamed. A single asterisk may be used to specify all file names but in this case fn2 must be specified as = (see below).

ft1 Specifies the file type of the archived file to be renamed. A single asterisk may be used to specify all file types but in this case fn2 must be specified as = (see below).

fn2 Specifies a new file name. It may be specified as = if the file name is to remain unchanged.

ft2 Specifies a new file type. It may be specified as = if the file type is to remain unchanged.

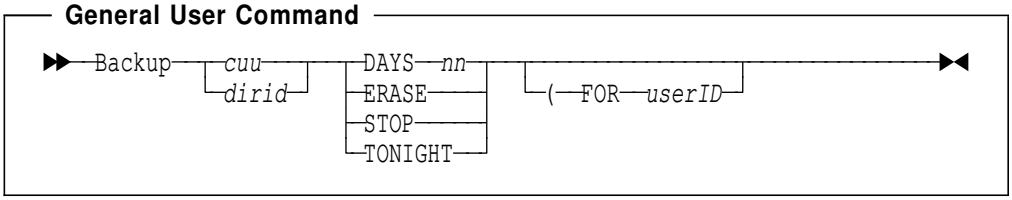
Options

FOR Specifies that the user ID whose data is to be renamed is the specified user ID or pool name rather than the user ID of the command issuer. The command issuer must have been authorized to do this.

See 8.3, “Sample BARS/VM Commands” on page 81. for some examples of the command.

8.2.8 BACKUP

The BACKUP command changes the backup parameters for a specified minidisk or SFS directory.



Operands

cuu Defines a minidisk whose backup parameters are to be changed.

dirid Specifies an SFS directory whose backup parameters are to be changed.

DAYS nn Requests an automatic backup every nn days. The number specified may be between an installation defined minimum and 99.

ERASE Ends automatic backup and requests that all backup data be erased after an installation defined period. A DISPLAY OPTIONS command displays the value of the period.

STOP Ends automatic backup of the minidisk or directory.

TONIGHT Requests that a backup should occur at the next backup cycle. The scheduling of cycles is the responsibility of the BARS/VM administrator.

Options

FOR Specifies that a minidisk whose backup parameters are be changed belongs to the specified user ID rather than the user ID of the command issuer.

This is only used in the case where one user ID owns another, i.e. is authorized to issue BARS/VM commands on behalf of that user ID. This option is not allowed if an SFS "dirid" was specified since the user ID is taken from the dirid.

Usage Notes:

1. Files on CMS minidisks and SFS files in file pools selected for backup by the administrator are backed up unless the user or installation requests otherwise.

   The last **'v'** versions of CMS files are backed up. **'v'** is an installation option and may be shown with the DISPLAY OPTIONS command.

2. A non-CMS minidisk (i.e. a minidisk not CMS formatted or a BLOCKIO minidisk) is **only** backed up if the user or administrator issues a BACKUP command for it.

   Note: Minidisks reserved for ***BLOCKIO** (e.g. SQL minidisks) are treated as **NON-CMS minidisks**. Therefore you have to issue a BACKUP command to get backup for those minidisks.

   There are two ways to request backup:

   a. `VMBARS BACKUP cuu DAYS nn`

   b. `VMBARS BACKUP cuu TONIGHT`

   DAYS nn. - request that backup occurs every *nn* days.

   Advantage: Automatic backup every *nn* days.

   Disadvantage: Your 'critical' changes will not be backed up until *nn* days.

   TONIGHT - request a 'deferred BACKUP', which means that backup occurs in the **next backup cycle**.

   Advantage: You can enter BACKUP after 'critical' changes.

   Disadvantage: You must remember to enter the command.

   TONIGHT - may be used for **one time only** backup.

   Note: Users should use the BACKUP command only if they require backup. If they will never RETRIEVE data, they should issue a BACKUP STOP or ERASE command.

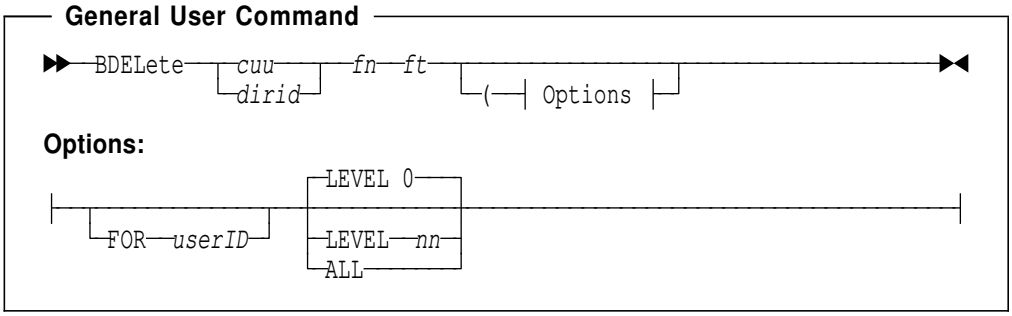
3. An erasure date set by the ERASE option will be removed by a BPUT command specifying that all files are to be backed up. See the usage notes for the BPUT command.

8.2.9 BDELETE

The BDELETE command deletes specified files from the backup catalogs.

It is not normally necessary to erase any backup data since BARS/VM automatically deletes backup copies of CMS files following their erasure from the user's minidisk or SFS directory. The backup copies of any erased files are retained by BARS/VM for a number of days as defined by the installation.

This command is useful in circumstances where sensitive data may have been backed up without proper protection. (e.g. highly confidential files which have been accidentally backed up unciphered).



Operands

cuu Defines a minidisk from which the files were backed up.

dirid Specifies an SFS directory from which the files were backed up.

fn Specifies the file name of a specific file or set of files to be deleted. "*" on its own is not allowed.

ft Specifies the file type of a specific file or set of files to be deleted. "*" on its own is not allowed.

Options

LEVEL nn Requests deletion of a particular backup level (0 is the latest level, 1 the next to latest level and so on).

The QUERY command may be used to determine which backup levels exist.

ALL Specifies that all existing backup levels are to be deleted.

FOR Specifies that the files to be deleted are owned to the specified user ID rather than the user ID of the command issuer. This is only used in the case where one user ID owns another, i.e. is authorized to issue commands on behalf of that user ID. This option is not allowed if an SFS "dirid" was specified since the user ID is taken from the dirid.

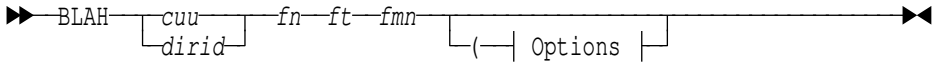
Usage Notes:

1. Data erased with the BDELETE command is irretrievable.
2. File name and file type may be specified using the * and % notation as described in 8.1.3, “How to Read a Syntax Diagram” on page 52.
3. Authorized users may specify * * for the file name and file type.

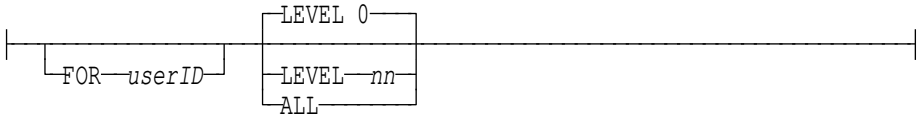
8.2.10 BLAH

The BLAH command marks the specified files as backed up at a remote host system. This informs the local BARS/VM system that it may erase the backup on this system and reuse the space that it occupied.

General User Command



Options:



Operands

cuu Defines a minidisk from which the files were backed up.

dirid Specifies an SFS directory from which the files were backed up.

fn Specifies the file name of a specific file or set of files to be processed.

ft Specifies the file type of a specific file or set of files to be processed.

fmn Specifies the filemode number of the file to be processed.

The filemode number may be specified either with a number only or with a letter combined with a number.

```
e.g. 1 and C1 means both filemode number 1.
```

Options

LEVEL nn Specifies a particular backup level (0-latest level, 1-next to latest level).

The QUERY command may be used to determine which backup levels exist.

ALL Specifies that all existing backup levels are processed.

FOR Specifies that the files belongs to the specified user ID rather than the user ID of the command issuer.

This is only used in the case where one user ID owns another, i.e. is authorized to issue commands on behalf of that user ID. This option is not allowed if an SFS "dirid" is specified since the user ID is taken from the dirid.

Usage Notes:

1. This command is normally sent by a remote host BARS/VM system to another BARS/VM system; it is **not** provided to be issued by a general user.
2. Data processed by the BLAH command cannot be retrieved from the local BARS/VM system.
3. File name and file type may be specified using the ***** and **%** notation as described in 8.1.3, “How to Read a Syntax Diagram” on page 52.

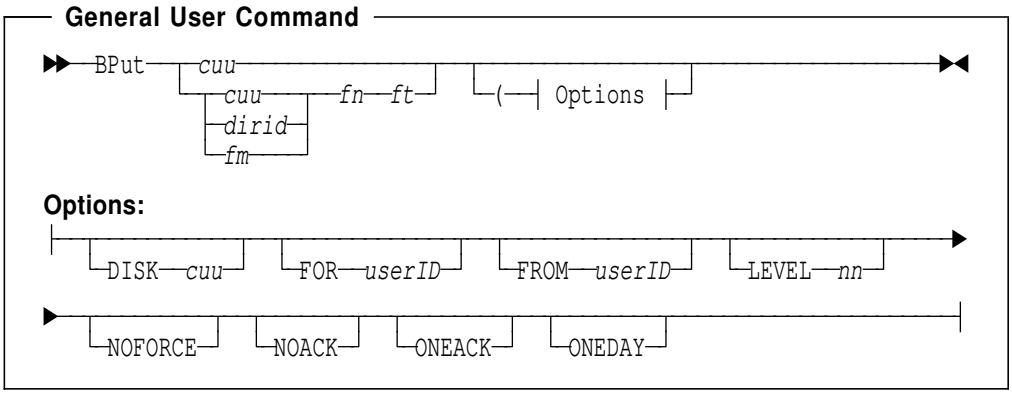
8.2.11 BPUT

The BPUT command makes backup copies of specified CMS files, or it may be used to backup a complete non-CMS minidisk (see usage note 2 on page 69). Users should be aware that any file backed up by this command has a new level 0 backup created and the oldest level is then lost, so care must be taken to avoid destroying a good backup of a file which might be needed in the event of its loss.

CMS files are backed up even if they are unchanged since the last backup cycle unless the NOFORCE option is specified.

SFS aliases are not backed up by the BPUT command. However, entries are maintained in the backup catalogs.

When the file name and file type are specified (i.e. not a non-CMS BPUT), a multi-line acknowledgement file is returned on command completion which contains one record for each file processed. The record indicates whether the file was processed successfully, ignored (if NOFORCE was specified, and flagged by an * in column 1), or failed to be processed (a return code is shown). Production of this file can be overridden by use of the ONEACK or NOACK options.



Operands

cuu Specifies a minidisk to be backed up. It must be specified as 2-4 characters.

dirid Specifies an SFS directory on which files to be backed up reside.

fm Specifies a single filemode letter defining a minidisk on which a file or files to be backed up reside. If a filemode letter is specified for an accessed minidisk, the requested file or files are sent to BARS/VM using DISK DUMP instead of being copied from the minidisk on which they reside. This allows files to be sent to another node for backup. If fm is specified then DISK (see below) should also be given to define the minidisk cuu.

If a filemode letter is specified for an accessed directory, the files are copied in the same way as if a dirid was specified.

fn Specifies the file name of a specific file or set of files for which backup copies are to be made.

ft Specifies the file type of a specific file or set of files for which backup copies are to be made.

Options

LEVEL nn Specifies a level number for the backed up file. nn is the Level number - 0 is the newest, then 1,2.. If LEVEL nn is not specified, a new one is built as level 0 and the older one increases by 1, i.e. 1 becomes 2, 2 becomes 3, 3 becomes 4, and so on. If LEVEL nn is specified, the existing nnth level is erased and a new nnth Level created. Levels before are not changed. If no level exists, the lowest available level number smaller than the specified one is used. *nn* may be a number from 0 to 99 but the number of levels which may be created are limited by the installation.

DISK Specifies the disk on which the files to be backed up reside. This option is used when a filemode is specified as the first operand of the command to give the cuu of the minidisk with which the backed up file is to be associated. This option is not allowed if an SFS "dirid" is specified since the user ID is taken from the dirid.

NOFORCE Specifies that each CMS file to be backed up is only to be backed up if it has changed since the last backup time. This command option is useful for automating database backup at a time when the database is consistent. Setting such a minidisk to NOBACKUP will enable a user to have complete control over his backup times.

NOACK Specifies that the command issuer does not wish to receive the acknowledgement file normally created by the BPUT command. NOACK will override the ONEACK option.

ONEACK Specifies that the command issuer wishes to receive a one-line acknowledgement file rather than the normal multi-line acknowledgement file. This option is overridden by the NOACK option.

ONEDAY Specifies that the backup is to proceed *only* if no backup has already taken place today for the specified minidisk, i.e. the response to a QUERY command shows todays date. This option is useful if you have a requirement of ensuring that a BPUT is only executed once a day for a minidisk. If no backup is done because the last backup date is today, then a normal one-line acknowledgement file is sent indicating success.

FROM Specifies that the files to be backed up reside on a minidisk belonging to the specified user ID rather than that of the command issuer. The command issuer must have been authorized to do this. This option is not allowed if an SFS "dirid" is specified since the user ID to whom the files belong is taken from the dirid.

FOR Specifies that the backed up files are to be associated with the specified user ID rather than the user ID of the command issuer. The command issuer must have been authorized to do this. This option is not allowed if an SFS "dirid" is specified since the user ID with whom the backed up files will be associated is taken from the dirid.

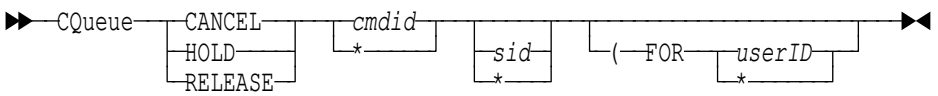
Usage Notes:

1. File name and file type may be specified using the * and % notation as described in 8.1.3, “How to Read a Syntax Diagram” on page 52.
2. File name and file type must be specified with a dirid or fm. If a cuu is specified and file name and file type are omitted, this requests that the minidisk be backed up as **non-CMS**.
3. The date last processed (which is displayed by the QUERY command) is only set to the current date by a BPUT of CMS files when * is specified for the file name and file type (i.e. when all files are to be processed). For a non-CMS minidisk the date is always set since the entire minidisk is backed up.
4. When * is specified for the file name and file type (i.e. when all files are to be processed), then any existing date which is set for the deletion of all backup data for the minidisk or directory is removed. Such a date may be set, for example, by the BACKUP command with the ERASE option.
5. If the NOFORCE option is specified, then any erased files are detected (and marked as such) in the catalog file.
6. Automatic volume separation is not enforced when the BPUT command is used to backup a file or minidisk. Thus, there can be no guarantee that the level 0 and 1 copies will reside on different backup volumes.
7. If the installation has specified that backup is directly to tape, then BPUT of CMS files and non-CMS minidisks will be done to tape.
8. For a non-CMS BPUT, the EMAXPCE setting is honored. For EMAXPCE values which equate to less than one cylinder, the value is effectively rounded up to that required for a single cylinder.

8.2.12 CQUEUE

The CQUEUE command alters the status of commands which are queued for execution but not yet started. These commands are shown by DISPLAY STATUS QUEUES.

General User Command



Operands

CANCEL Cancels the specified queued commands.

HOLD Specifies that commands are to be held in the command queue and not executed.

RELEASE Specifies that commands are to be released from HOLD status.

cmdid Is the command identification as shown by the DISPLAY STATUS QUEUES command. An * specifies all command identifications.

sid Is the session identification as shown by the DISPLAY STATUS QUEUES command. An * specifies all session identifications. If the operand is not specified it defaults to the current session.

Options

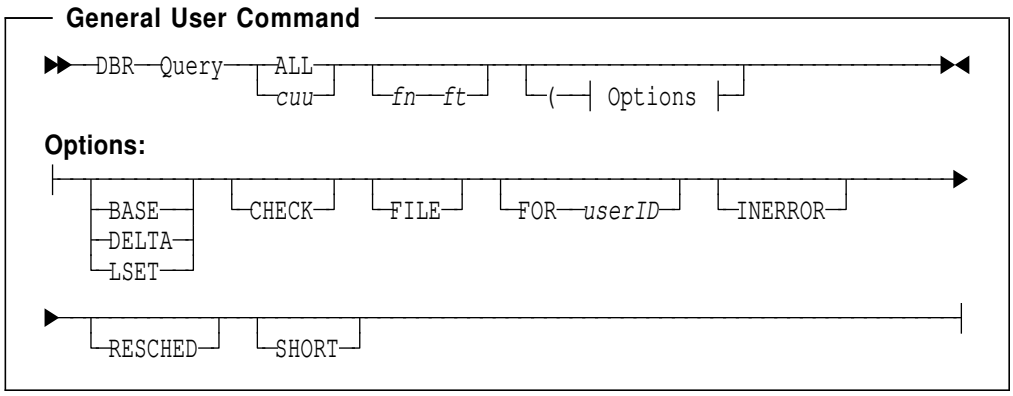
FOR This may only be used by authorized users. It specifies that commands initiated by the specified user id are to be processed and not those of the command issuer. If * is specified, commands from all user ids are processed.

Usage Notes:

1. This command does not process deferred commands. Such commands may only be cancelled through the CANCEL option on the relevant command.

8.2.13 DBR

The DBR QUERY command displays disaster backup information for the minidisk defined by the parameter *cuu* or for all minidisks where disaster backup data exists if **ALL** is specified.



Operands

ALL Requests disaster backup information for all minidisks owned by the specified user ID.

cuu Defines the user minidisk address for which disaster backup information should be displayed.

fn ft Specify particular CMS files. If these are specified then BARS/VM displays additional information for the particular file or files specified.

Options

BASE Restricts the display of files to those files which have been backed up for the last BASE DBR run.

DELTA Restricts the display of files to those files which have been backed up for the last DELTA DBR run.

LSET Restricts the display of files to those files which have been backed up for the last LSET or User ID DBR run.

CHECK Specifies that the result of the QUERY command is stripped down to those which have **not completed the last DBR run**.

FILE Specifies that the result of the DBR QUERY command is to be sent as a file to the command issuer's card reader in NETDATA format.

FOR Specifies that the user ID to be queried is the specified user ID rather than the user ID of the command issuer. The command issuer must have been authorized to do this.

INERROR Restricts the display of files to those files which could not be dumped during the specified DBR run.

RESCHED Restricts the display of files to those files which could not be dumped from a backup volume during the specified DBR run, but got rescheduled from the users minidisk. The RESCHED option is useful to verify the consistency of the BARS/VM backup data.

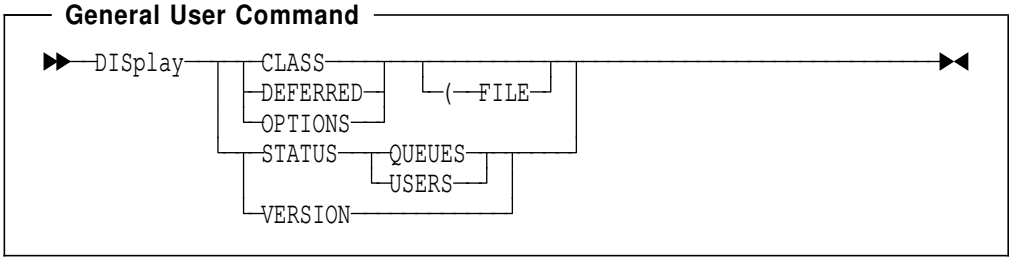
SHORT Specifies that the result of the DBR QUERY is to be a single line message for each minidisk requested giving the format of the minidisk and when disaster backup occurred. The SHORT option is useful when specifying **ALL** to reduce the number of messages received.

Usage Notes:

1. The cuu, file name and file type may be specified using the * and % notation as described in 8.1.3, "How to Read a Syntax Diagram" on page 52.

8.2.14 DISPLAY

The DISPLAY command displays general BARS/VM information.



Operands

CLASS Displays the archive classes and their meanings. These classes are used on the APUT command.

DEFERRED Requests a display of outstanding deferred commands for the user issuing the command, i.e. RETRIEVE, APUT or AGET commands which are to be processed at the time determined by the installation for deferred command processing.

OPTIONS Displays the current settings of the BARS/VM installation options.

STATUS Displays the current status of BARS/VM.

QUEUES Lists all pending commands which are queued but not yet started execution. These commands may be cancelled, held, or released using the CQUEUE command.

USERS Lists the current status of the various subuser virtual machines.

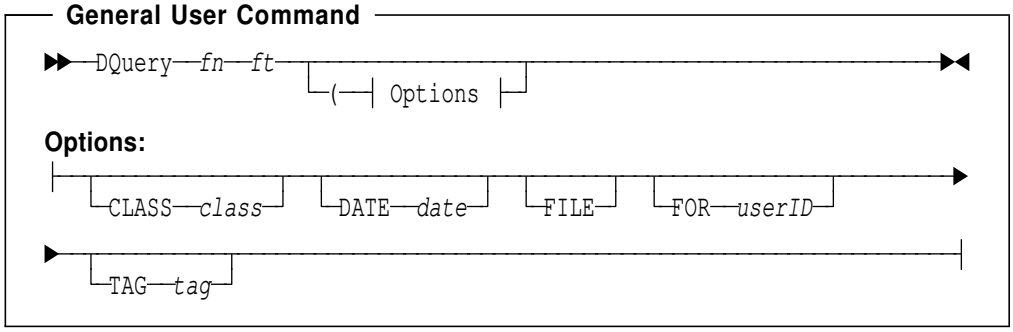
VERSION Displays the version, release and modification level of the BARS/VM system.

Options

FILE Specifies that the result of the command is to be sent to the command issuer as a file to his card reader rather than being displayed at his terminal. The file is sent in NETDATA format.

8.2.15 DQUERY

The DQUERY command displays information describing archived files which have been deleted but are still available for recovery via the DRECOVER command. See 7.5, “DQUERY and DRECOVER” on page 49 for a discussion on recovering deleted archive files.



Operands

fn Specifies the file name of a specific file or set of files whose information is to be displayed.

ft Specifies the file type of a specific file or set of files whose information is to be displayed.

Options

CLASS Specifies that only deleted files which have the specified class are to be displayed. *class* must be a single character.

DATE Restricts the display of files to those files which were archived on the specified date. The date must be of the form ddmmmyy, where dd is the day number, mmm is the mnemonic month and yy the year number, e.g. 10JAN96.

FILE Specifies that the result of the command is to be sent to the command issuer as a file to his card reader rather than being displayed at his terminal. The file is sent in NETDATA format.

FOR Specifies that the user ID to be queried is the specified user ID rather than the user ID of the command issuer. The command issuer must have been authorized to do this.

TAG Specifies that only deleted files which have the specified tag are to be displayed. *tag* may be up to 8 characters in length.

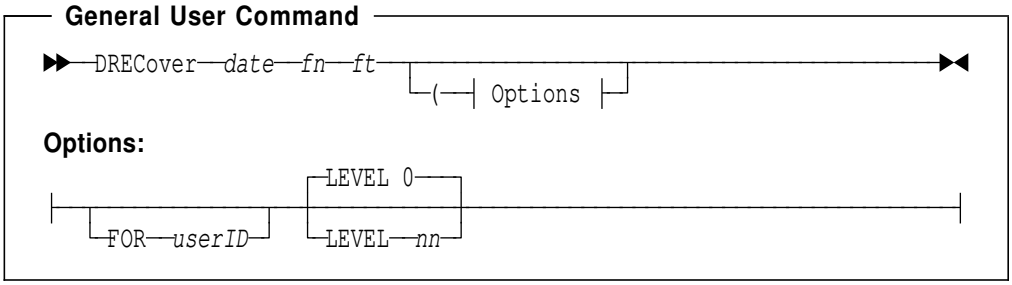
Usage Notes:

1. File name, file type and tag may be specified using the ***** and **%** notation as described in 8.1.3, “How to Read a Syntax Diagram” on page 52.

See 8.3, “Sample BARS/VM Commands” on page 81. for some examples of the command.

8.2.16 DRECOVER

The DRECOVER command recovers specified files which have been deleted from archive by the user but whose archive information is still available, thus allowing access to the archived file. The recovered files are placed in the command issuers virtual card reader. See 7.5, “DQUERY and DRECOVER” on page 49 for a discussion on recovering deleted archive files.



Operands

date Defines the date on which the file was deleted from archive. This is the date shown in the “Deletion Date” column of the DQUERY response. It must be of the form ddmmmyy, where dd is the day number, mmm is the mnemonic month and yy the year number, e.g. 10JAN96.

fn Specifies the file name of a specific file or set of files to be recovered.

ft Specifies the file type of a specific file or set of files to be recovered.

Options

FOR Specifies that the user ID whose data is to be recovered is the specified user ID rather than the user ID of the command issuer. This is only used in the case where one user ID owns another, i.e. is authorized to issue commands on behalf of that user ID.

LEVEL Requests a particular archive level (0-latest level, 1-next to latest level). The DQUERY command may be used to determine which archive levels exist. If **LEVEL** is unspecified, then level 0 is assumed.

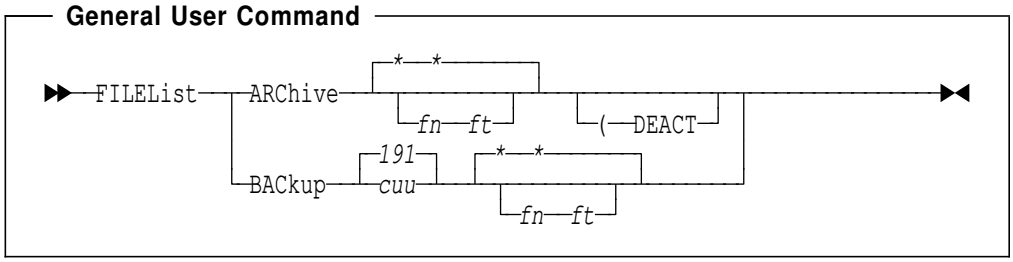
Usage Notes:

1. File name and file type may be specified using the * and % notation as described in 8.1.3, “How to Read a Syntax Diagram” on page 52.

See 8.3, “Sample BARS/VM Commands” on page 81. for some examples of the command.

8.2.17 FILELIST

The FILELIST command is an alternative method of accessing the FILELIST panel results screens for archive and backup, without having to go via the intermediate panels. On exit from the display, you are returned to where you were when you invoked the command, and hence do not need to step back out through the panels.



Operands

cuu Defines a user minidisk address for which the backup files should be displayed. ALL may be specified to show all minidisks. The default for cuu is 0191.

fn Specifies the file name of a specific file or set of files to be displayed.

ft Specifies the file type of a specific file or set of files to be displayed.

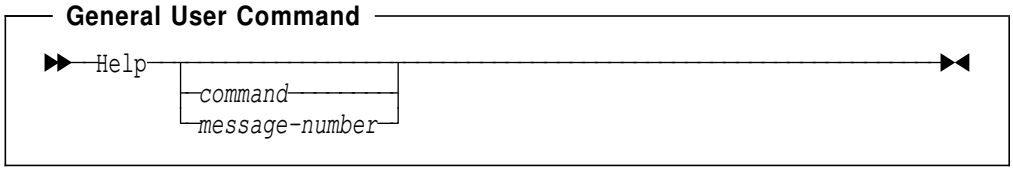
Options

DEACT Specifies that only archived files which have been deactivated are to be displayed. If this is not specified, then only active files are shown.

Usage Notes:

1. File name and file type may be specified using the ***** and **%** notation as described in 8.1.3, “How to Read a Syntax Diagram” on page 52.

8.2.18 HELP



The HELP command explains a specified command or message or gives a list of available commands.

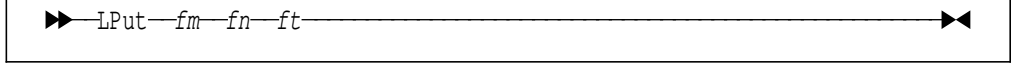
- If `VMBARS HELP command-name` is specified then a description of the specified command/function will be displayed.
- If `VMBARS HELP message-number` is specified then a description of the specified message will be displayed. The format of message-number is **HDFnnnns** where **nnnn** is the numeric part of the message and **s** is the severity. For example, for message `HDFRET0303E` the corresponding message-number for HELP would be `HDF0303E`.

- If no command or message number is specified, a list of commands is provided with the option of obtaining further help for any of those.

8.2.19 LPUT

The LPUT command sends an LSET file to the backup subsystem to be applied to the disaster backup definitions. The LSET file must exist. Only the owning or defined authorized users are allowed to manipulate data.
LSET is an abbreviation of "Logical SET of minidisks", which is a description of related minidisk objects for an LSET based disaster dump.

General User Command



Operands

fm Specifies a single file mode letter defining the minidisk or SFS directory on which the LSET file to be updated resides. The specified file is sent together with the command.

fn Specifies the file name of a specific LSET file for which disaster backup information is to be updated.

ft Specifies the file type of a specific LSET file. The only file type allowed is LSET.

Usage Notes:

1. The use of the LPUT command requires CMS PIPELINE.
2. The following is a possible layout of an LSET file. Note that only *_OWNER: and *_AUTHORIZED: are used for authorization verification.

```
*_OWNER: MANAGER NODEID * owner name, department, location
*_Authorized: TOM HARRY
*_LSET description: This LSET is a critical application.
*_LSET description: It is required to ...........
*_Critical LSET: N
*_Corequisite LSET: CMSR10 PL1
*_Corequisite LSET: PASCAL
*_Disaster Backup Recovery Plan: RECOVER PLAN USERREC 191
*_Cycle: 12 days
APPLID1 *
APPLID  0191
HARRY 0192
```

Sample possible LSET file layout

The interpretation is as follows:

***_OWNER:** Owner userid is MANAGER at NODEID. This userid is seen as the person who owns the LSET definition.

***_AUTHORIZED:** A maximum of two parameters can be defined. These local userids are seen as authorized by the owner to change and adjust the LSET definitions.

***_Corequisite LSETs:** Any number of statements can appear. Each word is checked at dump time for an existing LSET file with such a filename and, if found, the minidisk definitions are appended to the list of minidisks to be dumped.

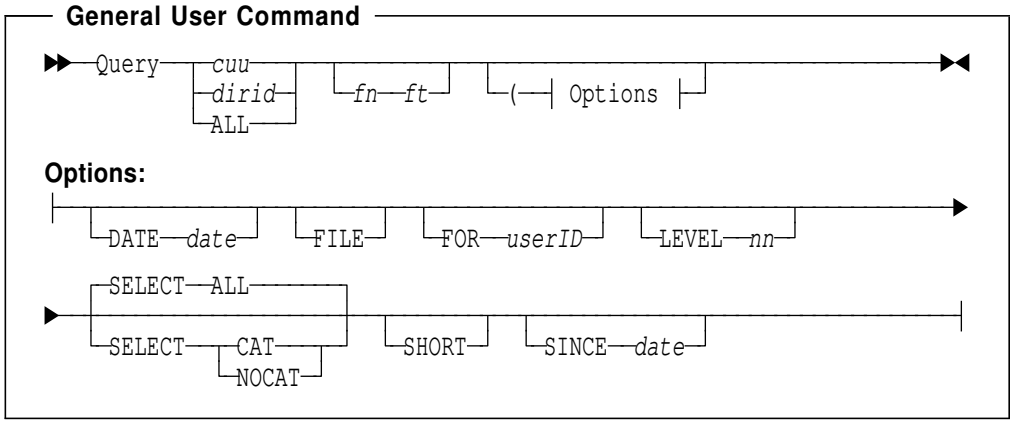
***_Disaster Backup Recovery plan:** The third and fourth word define the userid and minidisk address where the recovery information resides. This minidisk is automatically added to the minidisk definitions to be dumped.

***_xxxxx:** All other comment cards are treated as "comments".

Userid CCUU defines a Userid/Minidisk pair, one on each line. Userid and minidisk may be specified using the * and % notation as described in 8.1.3, “How to Read a Syntax Diagram” on page 52.

8.2.20 QUERY

The QUERY command displays backup information for minidisks and SFS directories.



Operands

cuu Defines a user minidisk address or set of addresses for which backup information should be displayed.

ALL Requests backup information for all minidisks and SFS directories for the specified user ID.

dirid Specifies an SFS directory for which backup information is to be displayed.

fn ft Specify particular CMS files.

If these are specified then BARS/VM displays additional information for the particular file or files specified.

Options

DATE Restricts the display of files to those files which were backed on the specified date. The date must be of the form ddmmmyy, where dd is the day number, mmm is the mnemonic month and yy the year number, e.g. 10JAN96. The DATE operand must not be used if SINCE is specified.

FILE Specifies that the result of the QUERY command is to be sent as a file to the command issuers card reader in NETDATA format.

FOR Specifies that the user ID whose backup data is to be queried is the specified user ID rather than the user ID of the command issuer.

This is only used in the case where one user ID owns another, i.e. is authorized to issue commands on behalf of that user ID. This option is not allowed if an SFS "dirid" is specified since the user ID is taken from the dirid.

LEVEL Specifies that only a particular backup level (*nn*) is to be displayed.

SELECT Determines which information is selected for display. SELECT CAT limits the display to objects for which there exists a catalog entry. SELECT NOCAT limits it to objects with no catalog. The default, SELECT ALL, displays all objects.

SHORT Specifies that the result of the QUERY is to be a single line message for each minidisk requested giving the format of the minidisk and whether backup will take place and at what interval. The SHORT option is useful when specifying **ALL** to reduce the number of messages received.

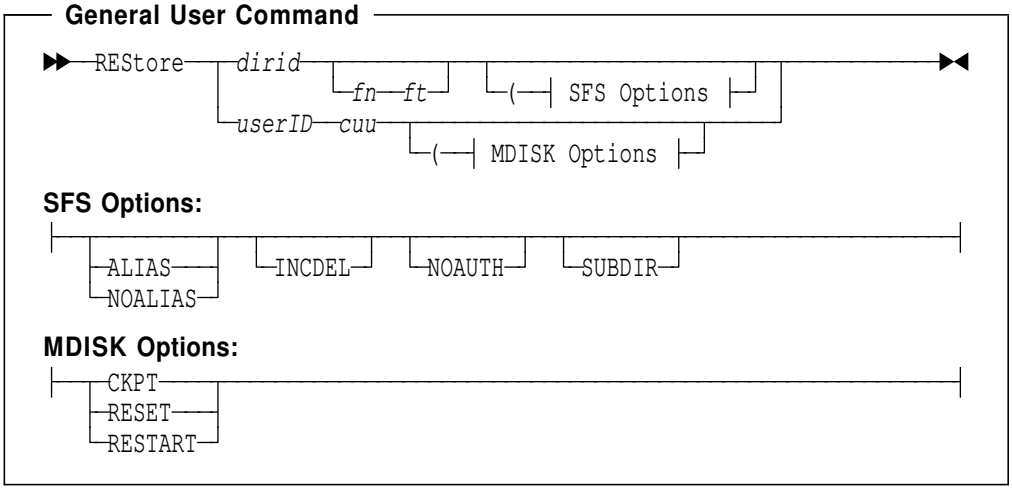
SINCE Restricts the display of files to those files which were backed up on or after the specified date. The date must be of the form ddmmmyy, where dd is the day number, mmm is the mnemonic month and yy the year number, e.g. 10JAN96. The SINCE operand must not be used if DATE is specified.

Usage notes:

1. The cuu, file name and file type may be specified using the ***** and **%** notation as described in 8.1.3, “How to Read a Syntax Diagram” on page 52.

8.2.21 RESTORE

The RESTORE command restores a specified SFS directory, file or alias with the latest backup copy or it restores a specified minidisk.



A restore of a minidisk restores it to the state it had at the last backup.

A restore of an SFS directory includes creation of the directory, the restoration of all files within the directory and the creation of any aliases which existed at the time of the last backup cycle. If the SUBDIR option is specified, then any

sub-directories are also restored. Directory and file authorizations are regranted unless the NOAUTH option is specified. The specified directory must not already exist or the command will be rejected.

If fn and ft are specified then this specifies a file or alias restore. The specified objects are restored to the directory, which must already exist. Subdirectories are not processed in this case.

SFS Operands

dirid Specifies an SFS directory to be restored, or, if fn and ft are specified, the SFS directory into which the specified files are to be restored.

fn Identifies the file name of a file or alias to be restored.

ft Identifies the file type of a file or alias to be restored.

SFS Options

ALIAS Specifies that only aliases are to be restored. This option is useful when backup exists for a file and an alias of the same name. It allows only the alias to be restored. It is also useful for restoring only aliases after all files have been restored. This ensures that the aliases can be recreated successfully.

NOALIAS Specifies that aliases are not to be restored. This option is most useful when restoring directories which may have had aliases to files in other directories which have not yet been restored. After the files have been restored, the aliases can be separately restored using the ALIAS option.

INCDEL Allows files and aliases to be restored which are marked as deleted, i.e. did not exist in the directory when it was last backed up. If this operand is not specified, such files are not restored.

NOAUTH Specifies that SFS directory and file authorizations are not to be regranted as part of the restore process.

SUBDIR Requests that all SFS sub-directories of the specified "dirid" also be restored. This option may not be used if *fn* and *ft* are specified.

Minidisk Operands

userID Specifies the user ID that owns the minidisk to be restored. The command issuer must be authorized to act for the user ID if it is not his own.

cuu Specifies the minidisk to be restored.

Minidisk Options

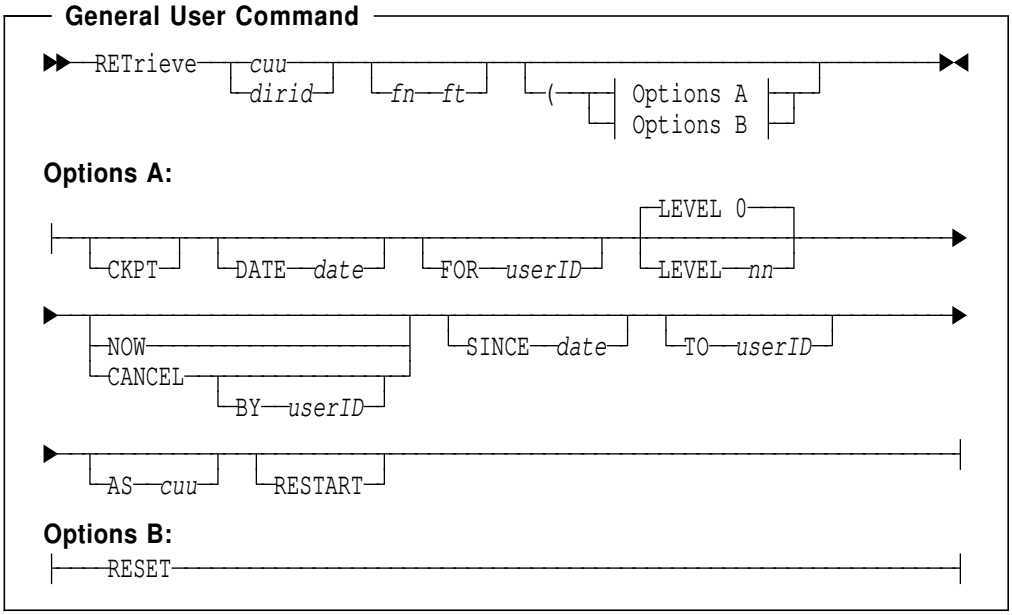
CKPT Specifies that the minidisk restore is to be checkpointed.

RESET Specifies that checkpoint information associated with this minidisk is to be erased.

RESTART Specifies that the minidisk restore is to restart from a checkpoint created by a previous restore of the same minidisk.

8.2.22 RETRIEVE

The RETRIEVE command retrieves specified data from BARS/VM backup minidisks and tapes. For CMS minidisks and SFS directories, individual files may be retrieved and these are placed in the virtual card reader. For non-CMS minidisks the data is retrieved by a DDR process directly to the specified minidisk.



Operands

cuu Defines a minidisk whose backup data is to be retrieved.

dirid Specifies an SFS directory whose backup data is to be retrieved.

fn Specifies the file name of a specific file or set of files to be retrieved from the backup of CMS data.

ft Specifies the file type of a specific file or set of files to be retrieved from the backup of CMS data.

Options A

BY Allows administrators to cancel deferred requests submitted by the specified user ID.

CANCEL Eliminates a previous deferred retrieval request.

Only the first matching request is cancelled. The *fn*, *ft*, *cuu* or *dirid* and **FOR** *userID* fields are compared. Other fields, including **LEVEL**, are ignored.

CKPT Specifies that the retrieve is to be checkpointed.

DATE date Causes the command to retrieve only files which were backed up on the specified date. The date must be of the form ddmmmyy, where dd is the day number, mmm is the mnemonic month and yy the year number, e.g. 10JAN96. The DATE operand must not be used if SINCE is specified.

FOR Specifies that the user ID whose backup data is to be retrieved is the specified user ID rather than the user ID of the command issuer.

This is only used in the case where one user ID owns another, i.e. is authorized to issue commands on behalf of that user ID. This option is not allowed if an SFS "dirid" is specified since the user ID is taken from the dirid.

LEVEL n Requests a particular backup version (0 - latest level, 1 - next to latest level).

The QUERY command may be used to determine which backup levels exist. If **LEVEL** is left unspecified, the latest level (level 0) is retrieved.

NOW Requests immediate retrieval. If NOW is not specified the retrieval is deferred to the next BACKUP cycle.

SINCE date Causes the command to retrieve only files which were backed up on or after the specified date. The date must be of the form ddmmmyy, where dd is the day number, mmm is the mnemonic month and yy the year number, e.g. 10JAN96. The date is specified as it appears in the QUERY command output. The SINCE operand must not be used if DATE is specified.

TO Specifies, for a CMS RETRIEVE, that retrieved files are to be sent to the specified user ID instead of the command issuer. For a non-CMS RETRIEVE the target minidisk is that of the specified userid instead of the command issuer.

This is only allowed for an authorized user ID, the operator user ID and when the command issuer owns the target user ID, i.e. is authorized to issue commands on behalf of that user ID.

AS May only be used for a non-CMS RETRIEVE and it specifies the cuu of the target minidisk when it is to be different from the one whose backup is being retrieved.

RESTART Specifies that the RETRIEVE is to restart from a checkpoint created by a previous RETRIEVE of the same minidisk.

Options B

RESET Specifies that checkpoint information associated with this minidisk is to be erased.

Usage Notes:

1. File name and file type may be specified using the ***** and **%** notation as described in 8.1.3, “How to Read a Syntax Diagram” on page 52.
2. File name and file type must be specified with a dirid. If a cuu is specified and file name and file type are omitted, this requests that a complete non-CMS minidisk be retrieved.
3. SFS aliases cannot be retrieved. They may, however, be restored through the RESTORE command.

8.3 Sample BARS/VM Commands

This chapter shows and explains examples of frequently needed BARS/VM commands.

ADEFINE

```
VMBARS ADEFINE DEPT EDUCATN ACCNT H919722
```

This command defines the user ID of the command issuer as belonging to department EDUCATN and having an account identification of H919722.

```
VMBARS ADEFINE ACCNT H919822
```

This command changes the account identification of the command issuer to H919822. The command issuer's user ID must be already defined to BARS/VM.

```
VMBARS ADEFINE DEPT USUPPORT ACCNT D231400 (FOR USUPP1
```

This command defines the user ID USUPP1 as belonging to department USUPPORT and having an account identification of D231400. The user ID of the command issuer must have been authorized to issue this command on behalf of USUPP1.

APUT

```
VMBARS APUT 191 TEST DATA (NOW CLASS U
```

This command requests the creation of a level 0 archived copy of the file TEST DATA and that it be assigned a class of U. The file resides on the 191 minidisk of the command issuer. The operation is to be carried out immediately.

```
VMBARS APUT .CODE62 R62* ASSEMBLE (TAG REL6.2
```

This command requests archive for all files starting with the characters R62 and with a file type of ASSEMBLE which reside in the SFS directory CODE62 of the command issuer and that the archived files have the tag REL6.2 associated with them. The copying of the files is to be deferred until the time determined by the installation for deferred command processing.

```
VMBARS APUT 291 DOCREL DATA (LEVEL 1 RETAIN 6
```

This command requests archive for the file DOCREL DATA on the command issuer's 291 minidisk. The archived file is to be retained for 6 months and assigned to level 1. Any existing level 1 copy of the file will be erased. The archiving of the file is to be deferred until the time determined by the installation for deferred command processing.

```
VMBARS APUT 191 * ASSEMBLE (FROM PUSER1 FOR APOOL
```

This command requests archive for all files with a file type of ASSEMBLE which reside on the 191 disk of user PUSER1. The archived files are to be associated with user ID APOOL. The archiving of the file is to be deferred until the time determined by the installation for deferred command processing.

```
VMBARS APUT 191 * ASSEMBLE (FROM PUSER1 FOR APOOL CANCEL
```

This command cancels the deferred APUT request which was illustrated in the previous example.

AGET

```
VMBARS AGET TEST DATA (NOW
```

This command requests immediate retrieval from archive of level 0 of the file TEST DATA.

```
VMBARS AGET R62* ASSEMBLE (LEVEL 1
```

This command requests retrieval from archive of the level 1 of all archived files with file name starting with the characters R62 and with a file type of ASSEMBLE. The retrieval is to be deferred until the time determined by the installation for deferred command processing.

```
VMBARS AGET R62* ASSEMBLE (CANCEL
```

This command cancels the deferred AGET request which was illustrated in the previous example.

```
VMBARS AGET * * (TAG REL6.2
```

This command requests deferred retrieval from archive of all level 0 files which have a tag of REL6.2.

AQUERY

```
VMBARS AQUERY TEST DATA
```

This command requests information about archived levels of the file TEST DATA.

```
VMBARS AQUERY R62* ASSEMBLE (CLASS U
```

This command requests data for all files with file name starting with the characters R62 and with a file type of ASSEMBLE and which were archived with a class of U.

```
VMBARS AQUERY * * (TAG REL6.2
```

This command requests information for all files which are archived with a tag of REL6.2

```
VMBARS AQUERY * * (DEACT
```

This command requests information for all deactivated files.

AALTER

```
VMBARS AALTER TEST DATA CLASS U
```

This command requests that all archived levels of the file TEST DATA should be altered to have class U.

```
VMBARS AALTER R62CNV ASSEMBLE RETAIN 12 (LEVEL 1
```

This command requests that the file R62CNV ASSEMBLE have the retention period for its level 1 archive changed to 12 months from now. If the file level is deactivated it will be reactivated.

ADELETE

```
VMBARS ADELETE TEST DATA
```

This command requests that all archived levels of the file TEST DATA be deleted.

```
VMBARS ADELETE R62* ASSEMBLE (LEVEL 1
```

This command requests that all files with file name starting with the characters R62 and with a file type of ASSEMBLE have their level 1 archived files deleted.

ARENAME

```
VMBARS ARENAME TEST DATA TEST16 DATAOLD
```

This command renames the archived file TEST DATA to the new name TEST16 DATAOLD.

```
VMBARS ARENAME * LEVEL1 =  LEVEL2
```

This command requests a rename of all archived files with a file type of LEVEL1 to a file type of LEVEL2. The file names are to remain unchanged.

```
VMBARS ARENAME TEST DATA =  DAT2 (FOR APOOL
```

This command renames the file TEST DATA to TEST DAT2 in the archive belonging to user APOOL. You must be authorized for the user ID APOOL by BARS/VM.

DQUERY

```
VMBARS DQUERY TEST DATA
```

This command requests information about previously deleted versions of the file TEST DATA.

```
VMBARS DQUERY R62* ASSEMBLE (CLASS U
```

This command requests data for all deleted files with file name starting with the characters R62 and with a file type of ASSEMBLE and which were archived with a class of U.

```
VMBARS DQUERY * * (TAG REL6.2
```

This command requests information for all deleted files which were archived with a tag of REL6.2

```
VMBARS DQUERY * * (FILE FOR APOOL
```

This command requests information for all deleted files for user ID APOOL and requests the result of the command to be send as a file.

DRECOVER

```
VMBARS DRECOVER 10OCT94 TEST DATA
```

This command requests recovery of the level zero archived file TEST DATA which was deleted on the 10th October 1994.

```
VMBARS DRECOVER 31JUL94 R62* ASSEMBLE (LEVEL 1
```

This command requests recovery for all deleted files with file name starting with the characters R62 and with a file type of ASSEMBLE and which were deleted on 31st July 1994. The level 1 versions of these files are to be recovered.

```
VMBARS DRECOVER 1JAN95 * * (FOR APOOL
```

This command requests recovery for all deleted files which were archived under the pool name of APOOL and deleted on 1st January 1995.

BDELETE

```
VMBARS BDELETE 191 TEST DATA
```

This command deletes backup data for the file TEST DATA on the command issuers minidisk 191 with backup level 0.

```
VMBARS BDELETE 191 T* ASSEMBLE (LEVEL 1
```

This command requests deletion for all files with file name starting with the character T and with a file type of ASSEMBLE. The level 1 versions of these files are to be deleted.

```
VMBARS BDELETE 192 TEST DATA (FOR APOOL
```

This command deletes backup information from the backup of the 192 minidisk of user ID APOOL for the file TEST DATA with backup level 0. You must be authorized for the user ID APOOL.

QUERY

```
VMBARS QUERY 191 * *
```

This command queries the backup of the command issuer's 191 minidisk and lists information for all files.

```
VMBARS QUERY .NEW.FILES * EXEC
```

This command queries the backup of the command issuer's directory NEW.FILES and lists information for files with a file type of EXEC.

RESTORE

```
VMBARS RESTORE PUSER1 191
```

This command requests a restore of the 191 minidisk of user PUSER1. If the command issuer is not PUSER1, previous authorization must have been given.

```
VMBARS RESTORE .DATA (SUBDIR
```

This command requests a restore of the directory DATA and all its subdirectories. The directory must not exist. The directory tree will be created and all files, aliases and authorizations restored.

```
VMBARS RESTORE MYPOOL:ARNOLD.MY.LIFE * EXEC
```

This command restores all files with a file type of EXEC to the directory ARNOLD.MY.LIFE in filepool MYPOOL, which must already exist.

RETRIEVE

```
VMBARS RETRIEVE 191 * EXEC
```

This command requests a retrieve of all files with a file type of EXEC that were backed up from the 191 minidisk of the command issuer. The retrieve is to be deferred.

```
VMBARS RETRIEVE .REL001 * ASSEMBLE (NOW LEVEL 1
```

This command requests a retrieve of all files with a file type of ASSEMBLE that were backed up from the directory REL001 of the command issuer. The retrieve is to be done now and to return the level 1 backed up data.

8.4 Authorizing other users

8.4.1 OWNED authorization

A BARS/VM administrator may use the OWNED command to grant one user authority to issue any BARS/VM general user command on behalf of another user. For example, if user BOB is granted OWNED authority for user TIM, then BOB may use the FOR operand of any BARS/VM general user command to perform actions on TIM's backup or archive data.

Note: This mechanism gives BOB authority for all commands —i.e. it does not allow authority to be restricted to only archive or backup, or just one particular minidisk.

8.4.2 ESM authorization

An installation may choose to also allow control of access to BARS/VM data via the use of an External Security Manager (ESM), such as RACF/VM*. This option is controlled via an installation option set by the administrator. If the ESM is used, then general users can control who has authority to access their BARS/VM data themselves, without involving the administrator. Any such access is ***in addition*** to access allowed via the OWNED command.

Use of an ESM allows TIM access to BOB's BARS/VM data in a way equivalent to his access to BOB's minidisks. For example, if TIM has READ access to BOB's 191 minidisk, then TIM can QUERY or RETRIEVE files backed up from BOB's 191 minidisk. However, he cannot use BDELETE or RESTORE, which require a higher level of access authority.

Note: The ESM only controls access to the DBR QUERY command and the minidisk (not SFS) functions of the backup commands.

The following tables define the ESM authority required to issue the relevant BARS/VM commands. When no options are used, the command operates on the command issuer's own data, and hence explicit authority is not required.

frame=box rules=both
split=yes width=column.

Figure 9. Backup Command ESM authority

Operands/Options	Target authority
cuu	—
cuu (FOR uid	Control uid.cuu

frame=box rules=both
split=yes width=column.

Figure 10. BDelete Command ESM authority

Operands/Options	Target authority
cuu	—
cuu (FOR uid	Update uid.cuu

frame=box rules=both
split=yes width=column.

Figure 11. Blah Command ESM authority

Operands/Options	Target authority
cuu	—
cuu (FOR uid	Update uid.cuu

frame=box rules=both
split=yes width=column.

Figure 12. BPut Command ESM authority

Operands/Options	Source authority	Target authority
cuu	—	—
cuu (FOR uid1	—	Update uid1.cuu
cuu (FROM uid2	Read uid2.cuu	—
cuu (FOR uid1 FROM uid2	Read uid2.cuu	Update uid1.cuu

frame=box rules=both
split=yes width=column.

Figure 13. DBR Query Command ESM authority

Operands/Options	Target authority
cuu	—
cuu (FOR uid	Read uid.cuu

frame=box rules=both
split=yes width=column.

Figure 14. Query Command ESM authority

Operands/Options	Target authority
cuu	—
cuu (FOR uid	Read uid.cuu

frame=box rules=both
split=yes width=column.

Figure 15. Restore Command ESM authority

Operands/Options	Target authority
uid cuu	Update uid.cuu

scale='0.8'
frame=box rules=both
split=yes width=column.

Figure 16. Retrieve Command ESM authority

Operands/Options	Source authority	Target authority
CMS:		
cuu1	—	—
cuu1 (FOR uid1	Read uid1.cuu1	—
cuu1 (TO uid2	—	—
cuu1 (FOR uid1 TO uid2	Read uid1.cuu1	—
Non-CMS:		
cuu1	—	—
cuu1 (FOR uid1	Read uid1.cuu1	Update uid1.cuu1
cuu1 (TO uid2	—	Update uid2.cuu1
cuu1 (AS	—	—
cuu1 (FOR uid1 TO uid2	Read uid1.cuu1	Update uid2.cuu1
cuu1 (FOR uid1 AS cuu2	Read uid1.cuu1	Update uid1.cuu2
cuu1 (FOR uid1 TO uid2 AS cuu2	Read uid1.cuu1	Update uid2.cuu2
cuu1 (TO uid2 AS cuu2	—	Update uid2.cuu2

Chapter 9. A Sample Communication with BARS/VM

The following is a console log of a sample communication with BARS/VM showing the results of some of the functions:

```
vmbars query 191
R;

HDFQUE004I QUERY command started
VMBARS   0191
                CMS Format, Backup Interval 1 Day   , Latest    18Nov94
             Disk label HJF191, Number of cylinders        2, Block size 2048
HDFQUE001I QUERY command complete



vmbars query 191 * script
R;

HDFQUE004I QUERY command started
VMBARS   0191
                CMS Format, Backup Interval 1 Day   , Latest    18Nov94
             Disk label HJF191, Number of cylinders        2, Block size 2048
-CMS File Backed Up-   Version  Creation   Backup  Backup  Erasure
Filename Filetype Fm#   Number    Date    Sequence Volume    Date
ACCTIN   EXEC      1       0       21Mar94  9W974   AAAAWJ
BARTEXT  EXEC      1       0       13Jun94  AGR3H   AAAATT
BARTEXT  EXEC      1       1        8Nov92  7KRUM   AAAAS2
CHAY     EXEC      1       0       26Sep94  B28IY   AAAARJ
CHECKAUX EXEC      1       0       13Jan94  9ECAG   AAAAX8
CHECKAUX EXEC      1       1       22Dec93  9AYPQ   AAAAXB
COLOFF   EXEC      1       0       17Sep91  5HGHY   AAAAQ9    23Nov94
ECCY     EXEC      1       0       15Nov94  BCL6M   AAAAXQ
FRAGI    EXEC      1       0       17Sep91  5HGNQ   AAAAQ9
NN       EXEC      1       0       26Apr94  A54QQ   AAAAXF    22Nov94
PROFILE  EXEC      0       0       10Dec93  97U39   AAAAQ1
PROFILE  EXEC      0       1        8Oct93  8UU0S   AAAARL
TOOLDSM  EXEC      1       0       12Aug93  8LAHP   AAAAQJ
TOOLDSM  EXEC      1       1       31Jan93  7WCXJ   AAAAVM
TOOLR8   EXEC      1       0        6Aug94  ASVE3   AAAAT6
UPDTDATH EXEC      5       0       17Sep91  5HH1V   AAAAQ9
HDFQUE001I QUERY command complete



vmbars retrieve 191 profile exec (level 0
R;

HDFDCA309I Request has been deferred



vmbars display deferred
R;

VMBARS  : RETRIEVE 191 PROFILE  EXEC     For:          From:
HDFDIS004I DISPLAY command complete
```

```
vmbars retrieve 191 profile exec (level 0  cancel
R;

HDFDCA311I RETRIEVE request cancelled


vmbars retrieve 191 profile exec (level 0  now
R;

HDFRET004I RETRIEVE command started


vmbars display status users

Userid   Status           Origin    Command being processed
-------------------------------------------------------------------------
BARSD    working          VMBARS   RETRIEVE191     PROFILE EXEC    A1
BARSE    waiting for work
BARSF    logged off       BARSA
BARSG    logged off       BARSA
BARSH    logged off       BARSA
R;


vmbars display options

Backup Installation Options:
----------------------------
Maximum number of files allowed to be retrieved per day:             20
Maximum number of non-CMS cylinders allowed to be retrieved per day:404
Number of days between file erasure and backup erasure:              21
Number of days between BACKUP ERASE command and backup erasure:      14
Number of days between erasure from directory and backup erasure:    14
Number of non-CMS backup copies requested:                            2
Number of CMS file backup copies requested:                           2
Minimum number of days allowed in the BACKUP command (NON-CMS):       7
Minimum number of days allowed in the BACKUP command (CMS):           1
Default backup frequency for CMS minidisks:                           1
Archive Installation Options:
-----------------------------
Default retention period:         012
Default archive class:              A
Maximum retention period:         024

Maximum AGETs allowed per day:     40
Maximum APUTs allowed per day:     40
R;
```

```
vmbars display status queues

No commands queued
R;

PUN FILE 5567 FROM BARSD    COPY 001 NOHOLD
RDR FILE 0700 SENT FROM BARSD    PUN WAS 0107 RECS 0005 CPY  001 A NOHOLD NOKEEP
HDFRET313I RETRIEVE of VMBARS   191 complete

q r all
ORIGINID FILE CLASS RECORDS  CPY HOLD DATE  TIME     NAME      TYPE
BARSD    0700 A PUN 00000005 001 NONE 10/27 12:13:10 PROFILE   EXEC
R;
```

Part 3. 'Administrator's Information'

Chapter 10. Authorized Commands Description

10.1 Issuing BARS/VM Commands

BARS/VM does not restrict the allocation of subusers for processing commands from an authorized user, as it does for general users.

10.1.1 Generic Specification

In some commands a generic name may be given for such things as file names or tags. "*", or "%" may be specified in the name to indicate a set of items, usually files, rather than a single one. The special character "*" indicates that any number of any characters will be accepted as a correct match. The special character "%" indicates that only a single character in the position represented by the "%" will be accepted as a correct match. The two special characters may be combined in any order. Two simple examples may serve to clarify the use of the special characters. B* represents a set of character strings consisting of any number of characters starting with the character B. B% represents a set of character strings consisting of two characters, the first of which is B and the second of which may be any character.

10.1.2 SFS Directory Specification

Some commands allow the specification of a Shared File System directory name or "dirid". This can have the general form:

```
<filepoolid:><user ID>.<n1<.n2<.n3<.n4<.n5<.n6<.n7<.n8>>>>>>>>
```

where filepoolid gives the ID of the file pool in which the directory resides and n1 to n8 are directory names. At least a . must be specified. If the file pool ID is not specified, the default that you set with the CMS SET FILEPOOL command is used. The user ID is defaulted to the virtual machine issuing the command. The directory can also be specified in the form:

```
+fm.n1<.ni+1...ni+7>
or
-fm<.ni.ni+1...ni+7>
```

where these have the same meanings as described in the CMS command reference.

10.1.3 How to Read a Syntax Diagram

Follow the path of the line from left to right, from top to bottom.

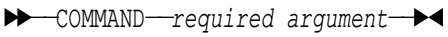
The definition of a command begins with two arrowheads pointing to the right.

The definition of a command ends with two arrowheads pointing to each other.

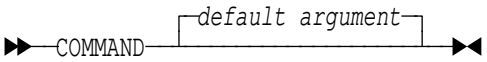
An arrowhead pointing to the right at the end of a line means that the definition is continued below.

An arrowhead pointing to the right at the beginning of a line means that the definition is continued from above.

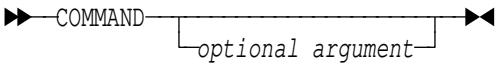
A required item is on the main path along the horizontal line.



A default item is above the main path.

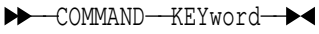


An optional item is below the main path.

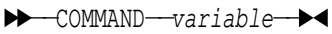


An item is a keyword, a variable, or a reference to a fragment in a syntax definition.

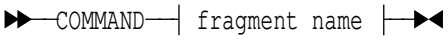
Keywords are shown in a Gothic font with the minimum abbreviation in uppercase.



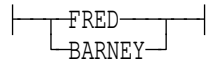
Variables are shown in lowercase italics.



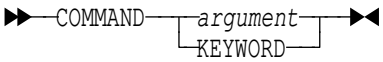
A reference to a fragment of a syntax definition breaks the main path with vertical bars. The fragment is defined later in the diagram.



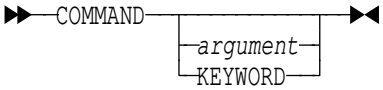
fragment name:



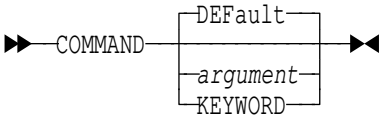
When you must choose between two or more items, they are stacked with the first one on the main path.



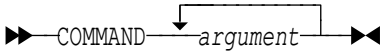
When you can select an item or take none, the choices are stacked below the main path.



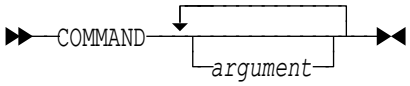
A default is shown above the main path.



An item may be repeated when an arrow returns to the left in front of it. The item is on the main path when you must write it at least once.



The item is below the main path when you may omit it altogether.



10.1.4 Authorized Commands Summary

User IDs are authorized to issue these commands via the AUTHORIZ command (see 10.2.5, “AUTHORIZ” on page 101). The OPTIONS file defines the BARS/VM manager user ID - EMUSERID (see “General Options” on page 226). This is the only user ID that can issue the AUTHORIZ command.

Directory management

- DIRECT
- PRIMARY
- QUERY
- RENAME

Operations

- BACKUP
- DISPLAY
- QUERY
- REPLY (only for OPERATOR and BARS/VM MANAGER user IDs)
- RESTORE
- SHUTDOWN
- START
- STOP

BARS/VM administrator

- AALTER
- AAUDIT
- AMERGE
- ATRANS
- AUTHORIZ (only for MANAGER user ID; requires BARS/VM restart)
- BACKUP
- BUMPA
- CATMAINT
- COLLECT
- DBR
- DBRTAPE
- DELETE
- DISKINIT
- DISPLAY
- EXTERNAL
- FILEPOOL
- HALT (only for MANAGER user ID)
- KILLSUB

- LSET
- MDISK
- NODENAME (requires BARS/VM restart)
- OWNED
- PASSWORD (requires BARS/VM restart)
- QUERY
- REMOVE
- RESTORE
- REUSE
- SHOWDISK
- SUBUSERS (requires BARS/VM restart)
- TIME
- TAPE
- TRACE
- TWERP
- UNCOMPL
- VMBUTIL
- VOLMAINT

10.1.5 How to Authorize Users

User IDs are authorized to issue these commands via the AUTHORIZ command (see 10.2.5, “AUTHORIZ” on page 101) which can only be issued from the BARS/VM Manager User ID.

The following authorized commands are also allowed to be issued from the **OPERATOR** user ID (EOUSERID):

- DISPLAY
- SHUTDOWN
- START
- STOP
- TIME
- RESTORE

10.1.6 Dates

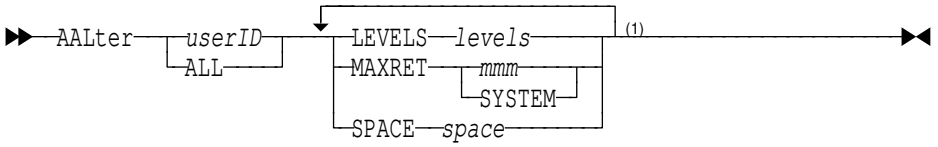
Where commands state that dates must be of the form ddmmmyy, dates from 1JAN60 to 31DEC99 are taken to have years of 1960 to 1999 and dates from 1JAN00 to 31DEC59 are taken to run from years 2000 to 2059.

10.2 Command Syntax Description

10.2.1 AALTER

The AALTER command may be used to alter the number of levels of archived files a particular user is allowed to maintain or change the total amount of archive space a user is allowed to use.

Authorized User Command



Note:
[1] Each choice may only be specified once.

Operands

userID Defines the user ID whose archive parameters are to be altered.

ALL Specifies that parameters for all user IDs are to be altered.

LEVELS Changes the maximum number of levels of archived files to the specified number, up to a maximum of 42. If the number of levels is reduced then archived files with levels higher than the new maximum are erased.

MAXRET Changes the maximum retention period which the specified user ID is allowed to use. The value *mmm* is the number of months to be allowed in the range 1 to 999. If the keyword **SYSTEM** is specified, then the maximum retention period allowed is reset to the system default as specified by EMAXRET.

SPACE Changes the amount of space the user is allowed to use for archived files to the specified number of 2K blocks, up to a maximum of 99,999,999. Files are not allowed to be archived for the user ID once this amount of space is used up.

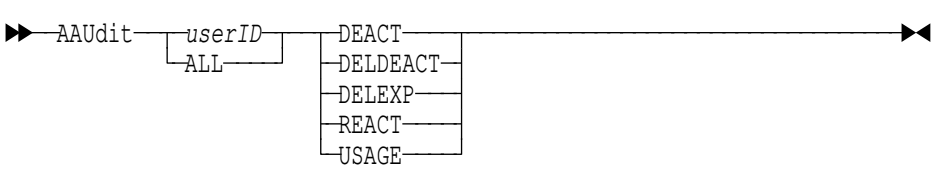
Usage Notes:

1. `AALTER ... MAXRET SYSTEM` is not the same as `AALTER ... MAXRET mmm`, where `mmm` is the current value of EMAXRET. Specifying **SYSTEM** means that if the value of EMAXRET changes at some time in the future, then the maximum retention value for the userID will also change. This is as if the **MAXRET** option had never been used for this archive ID.

10.2.2 AAUDIT

The AAUDIT command may be used to erase expired archived files, deactivate archived files or erase archived files which have already been deactivated.

Authorized User Command



Operands

userID Defines a user ID whose archived files are to be processed.

ALL Specifies that archived files belonging to all user IDs are to be processed.

DEACT Specifies that all files which have retention periods assigned to them and whose retention periods have expired are to be deactivated.

DELDEACT Specifies that all deactivated files are to be erased.

DELEXP Specifies that all files whose retention periods have expired are to be deleted immediately provided the user has issued the ADEFINE command with the DELEXP parameter.

REACT Specifies that all files which are currently deactivated are to be made active again. This can be used to undo the results of the DEACT option.

USAGE Specifies that a report be sent to the reader of the command issuer containing one record per defined archive user. Each record contains the archive ID, department, account number, DELEXP and SHARED flags from the ADEFINE command, the number of 2K blocks held on DASD, the number of 2K blocks held on tape, the maximum allowed number of 2K blocks and the date that an APUT or AGET was last issued. This report is useful for charging for BARS/VM archive usage, and for helping to identify inactive or unused archive definitions.

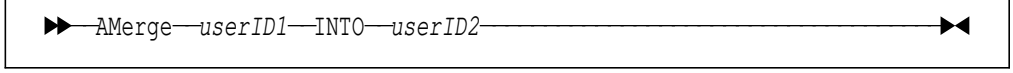
Usage Notes:

1. The AAUDIT ALL DELEXP processing is automatically run at the start of the backup window.

10.2.3 AMERGE

The AMERGE command may be used to merge the archive catalog (USERID ARC) files of two users.

Authorized User Command



Operands

userID1 Identifies the source user ID from which to merge.

userID2 Identifies the target user ID which is to receive the archive files.

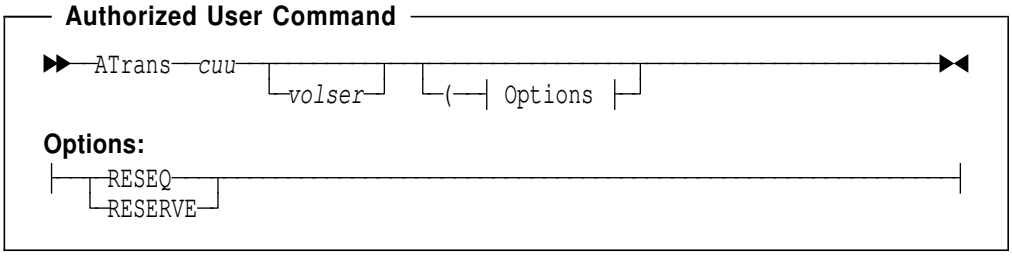
Usage Notes:

1. It is the first user ID specified whose archive data is merged into the catalog of the second user ID. The archive control file of the first user ID remains unchanged.
2. Conflicts with duplicate file names and file types do not cause the overwriting of archive data. Instead, an error message is sent to the command issuer, identifying the duplicate file ids. Those conflicts must be resolved manually, possibly by liaison with the users involved.
3. Once the merge has been completed successfully, the archive catalog for userID1 may be deleted.

4. If userID1 has more archive levels than userID2, then the archive catalog of userID2 will be altered to contain the same number of levels as userID1.

10.2.4 ATRANS

The ATRANS command is used during the process of transferring data on archive minidisks from one system to another.



Operands

cuu Defines the minidisk of the BARS/VM NOLOG user ID to be used for the ATRANSFER process.

volser This is the internal BARS/VM volid as generated by an ATRANS *cuu* (RESERVE command on the receiving system.

Options

RESEQ Merge all archive data on the specified minidisk *cuu* into an existing archive system.

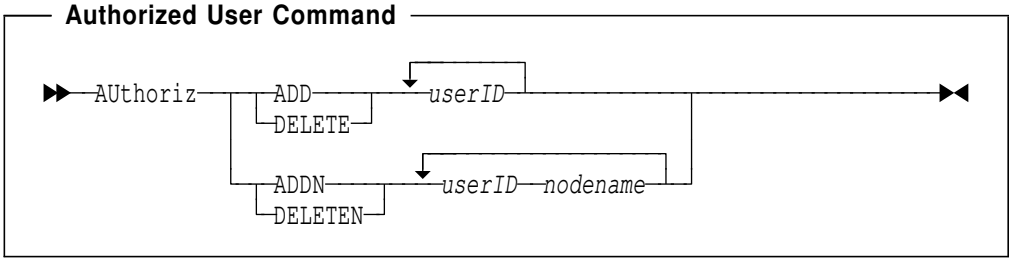
RESERVE Reserve an unused minidisk (*cuu*) to be used as an ATRANSFER minidisk. The minidisk is marked as formatted, archive and external. BARS/VM generates and displays a CMS volser for the minidisk.

Usage Notes:

1. If **no option** is specified, the archive catalog files are prepared for the transfer process. This involves changing the label of the minidisk specified as *cuu* to the specified *volser* and changing all "userID ARC" files affected by the transfer (previously collected to the specified *cuu* by a COLLECT command) to point to the new *volser*.
2. See 12.8, “Transfer of Archive Data to Another System” on page 161 for a discussion of the ATRANS process

10.2.5 AUTHORIZ

The AUTHORIZ command adds or deletes specified user IDs to or from the BARS/VM authorization list.



Operands

ADD Specifies that a local user ID is to be given authority.

DELETE Specifies that authority is to be removed from a local user ID.

ADDN Specifies that a remote user ID is to be given authority.

DELETEN Specifies that authority is to be removed from a remote user ID.

userID The user ID for which authority is to be granted or removed.

nodename The node name of the system where a remote user ID resides.

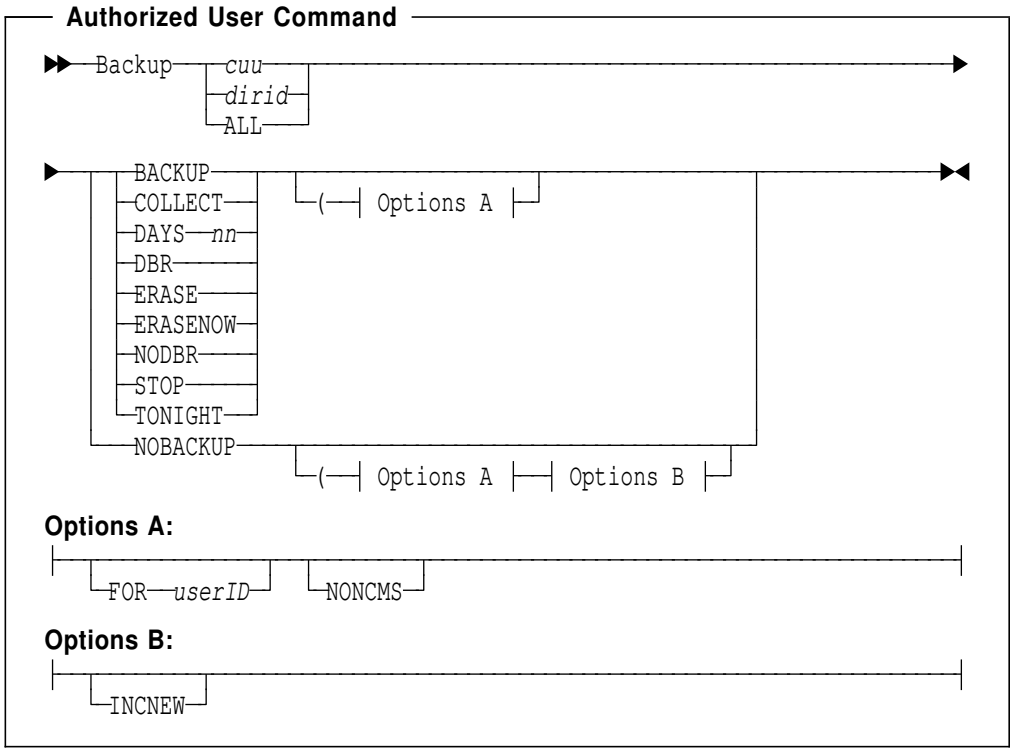
Options: None

Usage Notes:

1. **The AUTHORIZ command can only be issued from the BARS/VM Manager user ID.**
2. If ADDN or DELETEN is specified a list of user ID and nodename pairs may be given and the authorization applies only to commands received from the user IDs at the specified nodes.
3. The new list of node names becomes effective on the **next initialization** of BARS/VM.

10.2.6 BACKUP

The BACKUP command changes the backup parameters for a specified minidisk, set of minidisks or for a specified SFS directory.



Operands

cuu Defines the minidisk or minidisks whose backup parameters are to be changed.

dirid Identifies the SFS directory for which the backup parameters are to be changed.

ALL Specifies that all minidisks and SFS directories belonging to the user ID specified in the FOR option are to be changed.

BACKUP Cancels a NOBACKUP request.

COLLECT Requests that a backup occurs at the next backup cycle for all files, whether they have been changed or not. This allows a faster restore capability since fewer backup volumes will be needed for a RESTORE. The backup replaces the level 0 and level 1 files are not changed. This operand should be used with care since it will increase the amount of backup processing required at the next backup cycle.

DAYS nn Requests an automatic backup every 'nn' days. The number specified may be between 1 and 99. The installation defined minimum is not enforced for authorized users.

DBR Requests that a minidisk or SFS directory be eligible for dumping by the DBR (Disaster Backup and Recovery) component of BARS/VM. This is effective only if EDBRREQ has been set to YES in the installation options. If set to NO, all minidisks are candidates for DBR backup, and the use of this operand is superfluous. This operand is not allowed if an SFS "dirid" was specified.

ERASE Ends automatic backup of a minidisk or SFS directory and requests that all backup data be erased after a period of days equal to the EBACKE installation option. If the NONCMS option is specified and a minidisk is being backed up as both CMS and non-CMS then the erasure is only for the non-CMS backup; otherwise all backup for the minidisk is stopped and all backup data is erased.

ERASENOW Completely removes any backup for the specified minidisk or SFS directory by immediately erasing the relevant catalogue file. At the next backup cycle the minidisk will be treated like a new minidisk.

Warning: Once this command has completed, the backup data is totally lost. The only way to recover is to replace the relevant catalogue file on the interface 191 minidisk before the next backup cycle runs, possibly by using the backup copy from the interface 193 or 192 minidisks.

NOBACKUP Requests no backup for this minidisk, set of minidisks, or SFS directory and requests erasure of all existing backup data at the next backup cycle. If the NONCMS option is specified and a minidisk is being backed up as both CMS and non-CMS then the erasure is only for the non-CMS backup; otherwise all backup for the minidisk is stopped and all backup data is erased. If ALL and NOBACKUP are specified, the command excludes **all** objects for the user id from backup, both currently existing and those created in the future. The NOBACKUP option may also be used to prevent backup of specific sets of minidisks created in the future. See the INCNEW option below.

NODBR Cancels a DBR request. This operand is not allowed if an SFS "dirid" was specified.

STOP Ends automatic backup of a minidisk or SFS directory. Effectively the existing backup is "frozen". If the NONCMS option is specified and a minidisk is being backed up as both CMS and non-CMS then only the non-CMS backup is stopped; otherwise all backup for the minidisk is stopped.

TONIGHT Requests that a backup should occur at the next backup cycle. The timing of these cycles is determined by the administrator.

Options

FOR Identifies the user ID owning the minidisks for which the backup parameters are to be changed. This option is not allowed if an SFS "dirid" is specified since the user ID is derived from the dirid.

NONCMS Is for minidisks which are being backed up both as CMS and non-CMS minidisks and specifies that the command is to apply to the non-CMS backup only. This option is not allowed if an SFS "dirid" is specified.

INCNEW Applies only if NOBACKUP is specified. If INCNEW is not specified, the command applies only to objects that BARS/VM currently knows about (see the usage note below) except if ALL was specified. However, if INCNEW is specified, the command also applies to any new objects created. As an example, to prevent any backup for minidisks for user SQLDBA with virtual addresses starting with 2, either currently existing or created in the future, issue the command:

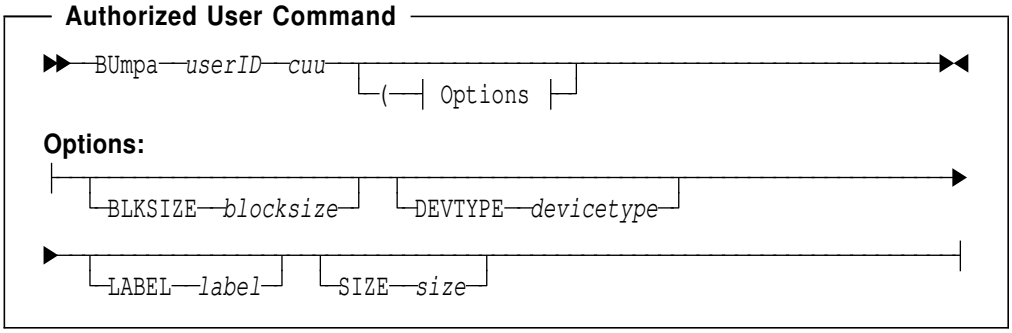
```
BACKUP 2* NOBACKUP (FOR SQLDBA INCNEW
```

Usage Notes:

1. The NONCMS option is not supported by ERASENOW.
2. cuu and userID may be specified using the * and % notation as described in 8.1.3, “How to Read a Syntax Diagram” on page 52. but the command will only be applied to objects that BARS/VM currently knows. For example, a BACKUP * NOBACKUP sets all minidisks of the command issuer to no backup for which BARS/VM currently has catalogs. Minidisks subsequently created are backed up in the normal way.

10.2.7 BUMPA

The BUMPA command sets particular attributes for a minidisk to be associated with the backup of the minidisk. The information is stored into the catalog (UC) file for the specified minidisk. This command is used in situations where the information is not available to the backup process, for example when remote backup is performed. For remote backup, this command is generated automatically by BARS/VM and sent to the host system to record the minidisk attributes.



Operands

userID Is the user ID owning the minidisk for which attributes are to be updated.

cuu Is the minidisk for which attributes are to be set.

Options

BLKSIZE Specifies the block size of the minidisk.

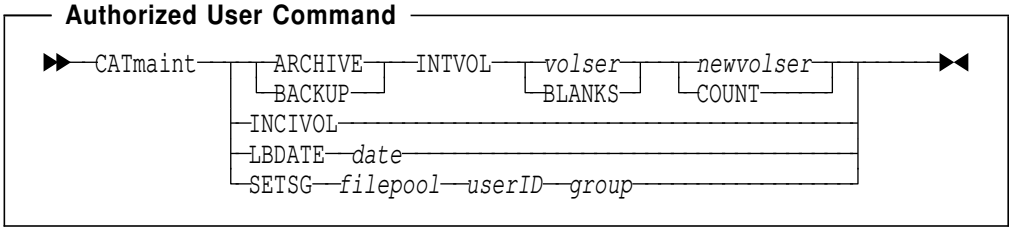
DEVTYPE Specifies the device type.

LABEL Specifies a CMS label for the minidisk.

SIZE Specifies the size of the minidisk in cylinders or blocks.

10.2.8 CATMAINT

The CATMAINT command allows maintenance to be performed on the catalogs and control files.



Operands

ARCHIVE Changes or reports on the archive catalog files.

BACKUP Changes or reports on the backup catalog files.

INTVOL Specifies that all occurrences of a specified internal volume serial number are to be changed or counted.

volser Gives the internal volume serial number to be changed or counted.

BLANKS Specifies that the internal volume serial number to be changed or counted is blank.

newvolser Specifies that the internal volume serial number to be changed to this value.

COUNT Specifies that the number of occurrences of the internal volume serial number are to be counted.

INCIVOL Increments the value of the next available internal volume serial number in the control file. This volume serial is used whenever a new minidisk or tape is brought into use for backup or archive. The current setting of the volume serial number can be displayed by the command DISPLAY NEXTIVOL.

LBDATE Sets the date of last backup to a specified date. This is the date which is displayed by the command DISPLAY CURRENT. Since only one backup cycle will be run per day, this command can be used to reset the date and therefore allow backup to be run again.

date Is the date to be set. It must be of the form ddmmmyy, where dd is the day number, mmm is the mnemonic month and yy the year number, e.g. 10JAN96.

SETSG Changes the Shared File System storage group number stored in the catalog for the specified filepool and userID.

filepool Identifies the filepool.

userID Specifies the userid whose catalog is to be changed.

group Specifies the new storage group.

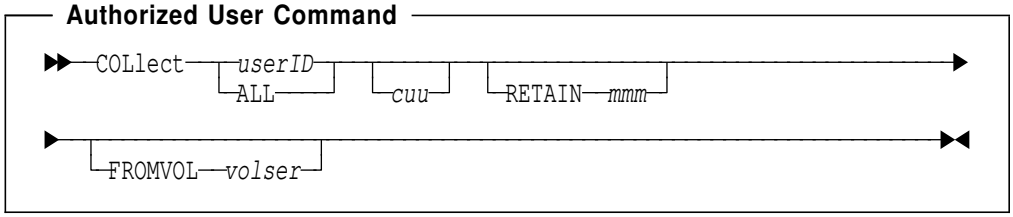
Usage Notes:

1. CATMAINT SETSG can be used to transfer a user to a different storage group. After setting the new storage group in the catalog, the user's filespace may be deleted. A RESTORE of the filespace will then restore him to the new storage group.

   Warning: If the user's filespace is deleted, then all their authorities to files in other filespaces will be lost, and any aliases which other users have to the deleted filespace will also be lost. These are not reinstated during a filespace restore.

10.2.9 COLLECT

The BARS/VM COLLECT command moves archived files from their current archive minidisks to new minidisks. The command may be used to collect files on a volume which may be removed from the system. COLLECT only works with archive on minidisks, not tapes.



Operands

userID Specifies that only files archived under a specific user ID or pool name are to be collected.

ALL Specifies that files for all user IDs are to be collected. This is normally used together with the RETAIN operand.

cuu Specifies, optionally, the cuu of the first archive minidisk to which the files are to be moved. If *cuu* is not specified, an unformatted minidisk will be selected, formatted and used to collect the files.

When the first minidisk is full subsequent unformatted minidisks are formatted and used. This happens whether or not *cuu* is given.

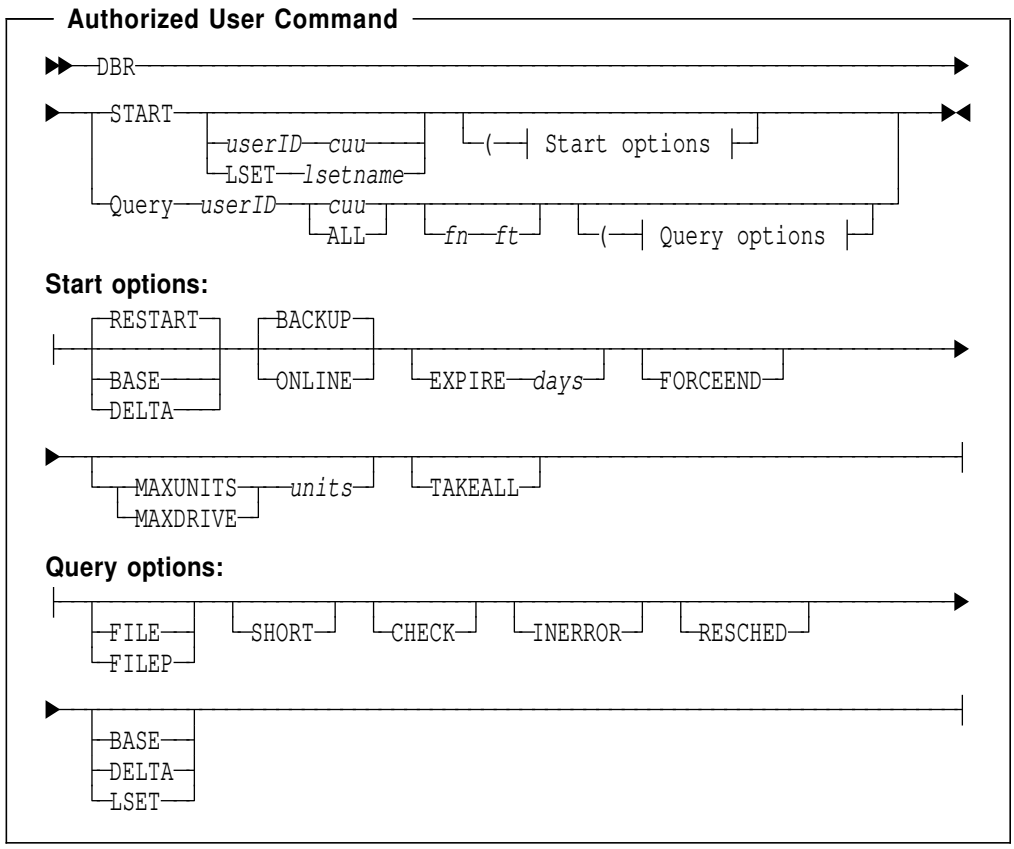
RETAIN Specifies that only files whose retention period expires at or after *mmm* months from now are to be collected. This operand is optional, and may be any value from 1 to 999.

FROMVOL Specifies that only files on the minidisk whose internal label (CMS volume serial) is *volser* are to be collected. This operand is useful when it is desired to move all files from an archive minidisk.

Usage Notes:

1. COLLECT only processes ARC catalogues, and not MIG catalogues. Hence you cannot use it to collect minidisk archive files (see the `MDISK` command).

10.2.10 DBR



Start Operands

START Starts a disaster backup process either for the minidisk(s) defined by the parameter **userID cuu** or for a logical set defined by the parameter **LSET lsetname** or, if nothing is specified, for the entire system.

userID Specifies the user ID or user IDs for which disaster backup is to be started.

cuu Defines the user minidisk address for which disaster backup is to be started.

LSET lsetname
Requests disaster backup for all minidisks defined in the LSET control file with the name lsetname. This control file is maintained via the LSET command.

Start Options

BASE Indicates that a new disaster backup dump should be started which has to back up all files for the minidisks selected (BASE run).

DELTA Indicates that a new disaster backup dump should be started which has to back up only those files which were not included in the last BASE run or have changed since then (DELTA run).

RESTART Indicates that the latest uncompleted disaster backup dump should be restarted at the point where it stopped (restart capability).

BACKUP Specifies that the data is to be dumped from the backup data whenever possible.

ONLINE Specifies that the data is to be dumped from the original minidisks.

EXPIRE days
Specifies the expiration date in days to be used for the DBR dump. The value passed overrides the installation option EEXPBASE or EEXPIRE for this run only. It needs to be reissued in case of RESTART processing.

FORCEEND Specifies that a control file tape should be created regardless of errors experienced during a DBR run. This option is most useful for DBR RESTART processes.

MAXUNITS units
Specifies the maximum number of tape drives to be used for the DBR dump. The value passed overrides the installation option ENUMTDBR for this run only. It needs to be reissued in case of RESTART processing.

MAXDRIVE units
This is an alternative form of the option MAXUNITS.

TAKEALL Specifies that the *default* DBR policy should be overridden in a way to make all accessible minidisks eligible for this DBR run. See 13.1.3, “The Disaster Backup Policy and How to Change it” on page 174 before using this option.

Usage Notes:

1. UserID and cuu may be specified using the * and **%** notation as described in 8.1.3, “How to Read a Syntax Diagram” on page 52.
2. To start a user ID/minidisk or an LSET dump you **must** specify the BASE or DELTA option. BARS/VM will always dump **all** data for the selected items.
3. The RESTART feature will only work if the restart is attempted on the same day that the dump was started.

Query Operands

QUERY Displays disaster backup information for the minidisk defined by the parameter *cuu*, or for all minidisks where disaster backup data exists if **ALL** is specified.

userID Specifies the user ID or user IDs for which information is to be displayed.

cuu Defines the user minidisk address for which disaster backup information should be displayed.

ALL Requests disaster backup information for all minidisks defined in the specified user ID.

fn ft Specify particular CMS files.

If these are specified then BARS/VM displays additional information for the particular file or files specified.

Query Options

FILE Specifies that the result of the DBR QUERY command is to be sent as a file to the command issuer's card reader in **DISK DUMP** format.

FILEP Specifies that the result of the DBR QUERY command is to be sent as a file to the command issuer's card reader in **CMS PUNCH** format.

CHECK Specifies that the result of the QUERY command is stripped down to those items which were **not completed during the last DBR run**.

SHORT Specifies that the result of the DBR QUERY is to be a single line message for each minidisk requested giving the format of the minidisk and when disaster backup occurred. The SHORT option is useful when specifying **ALL** to reduce the number of messages received.

INERROR Restricts the display of files to those files which could not be dumped during the specified DBR run.

RESCHED Restricts the display of files to those files which could not be dumped from a backup volume during the specified DBR run, but got rescheduled from the users minidisk. The RESCHED option is useful to verify the consistency of the BARS/VM backup data.

BASE Restricts the display of files to those files which have been backed up for the last BASE DBR run.

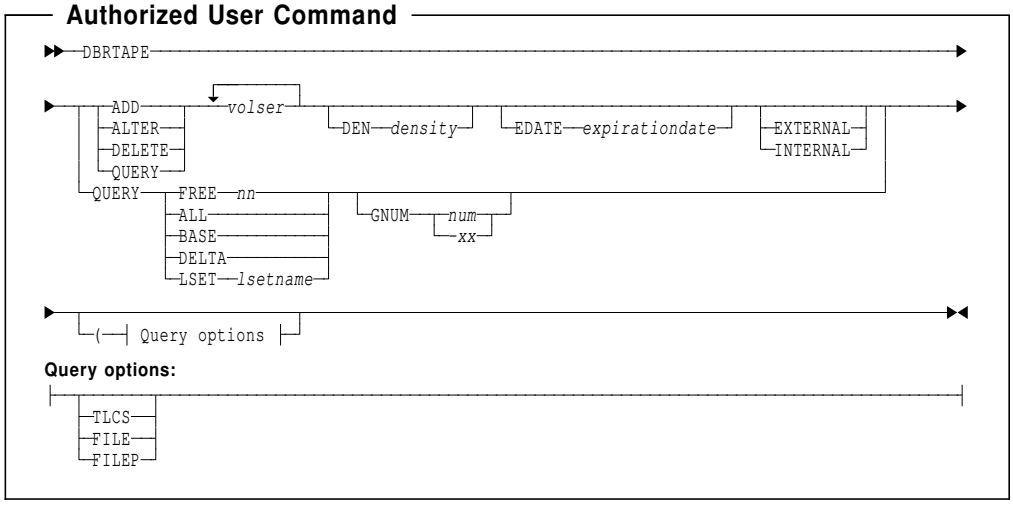
DELTA Restricts the display of files to those files which have been backed up for the last DELTA DBR run.

LSET Restricts the display of files to those files which have been backed up for the last LSET or user ID DBR run.

Usage Notes:

1. UserID, file name and file type may be specified using the ***** and **%** notation as described in 8.1.3, “How to Read a Syntax Diagram” on page 52.

10.2.11 DBRTAPE



The DBRTAPE command maintains and displays the tape resources which can be used to store DBR information.

Operands

ADD Specifies that the following volser or volsers should be added to the DBR tape list. Tapes to be added must be standard labelled.

ALTER Specifies that the following volser or volsers should be altered in the DBR tape list.

DELETE Specifies that the following volser or volsers should be deleted from the DBR tape list.

QUERY Specifies that the following volser or volsers should be queried. This is the first form of the DBRTAPE QUERY command.

volser Specifies the external label of the tape volume. This is the visual label as recognized by the Operator.

DEN density Defines the tape density to be used to dump the data to the given volser or volsers.

EDATE expirationdate
Defines the expiration date when the given volser or volsers can be reused. This date will not be written to the OS standard label, so it does not protect the misuse of a DBR tape. If this parameter is not specified, a default expiration date of 1JAN87 is given.

EXTERNAL Defines the given volser or volsers to be marked as external, which means not accessible.

INTERNAL Defines the given volser or volsers to be marked as internal, which means accessible.

Note: If DBR is given a volser during a DBR run, then it will automatically reset an existing EXTERNAL indication.

QUERY Specifies that volsers should be queried matching the following parameters. This is the second form of the DBRTAPE QUERY command.

FREE nn Request 'nn' volsers to be displayed which are candidates to be used or reused (which means free from the DBR point of view).

ALL Requests all volsers to be displayed which are defined as to be used by DBR.

BASE Requests all volsers to be displayed which are used by the latest DBR BASE run or the generation asked for by GNUM parameter.

DELTA Requests all volsers to be displayed which are used by the latest DBR DELTA run or the generation asked for by GNUM parameter.

LSET lsetname Requests all volsers to be displayed which are used by the LSET dump identified via 'lsetname' or the generation asked for by GNUM parameter. For a User ID dump display, 'lsetname' specifies the user ID.

GNUM num Specifies the generation number to be displayed. If not specified, the newest generation will be displayed. 'num' may be either an absolute number or given as '-num', which means newest generation number - 'num' generations.

Query Options

FILE Specifies that the result of the DBR QUERY command is to be sent as a file to the command issuer's card reader in NETDATA format.

FILEP Specifies that the result of the DBR QUERY command is to be sent as a file to the command issuer's card reader in **CMS PUNCH** format.

TLCS Specifies that the result of the DBR QUERY command is to be formatted in a way so that TLCS input cards are produced. TLCS is a TAPE Library Control System, running as an MVS application. Option TLCS implies option FILE.

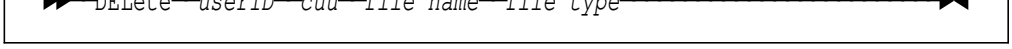
Usage Notes:

1. DBR tapes are independent of tapes defined for normal backup usage.

10.2.12 DELETE

The DELETE command may be used to remove a specified backup file name and file type from the specified catalog (UC) file. The file name and type specified on this command is the internal file identifier used by BARS/VM.

Authorized User Command



Operands

userID Identifies the owner of the minidisk for which the backup file is to be removed from the catalog.

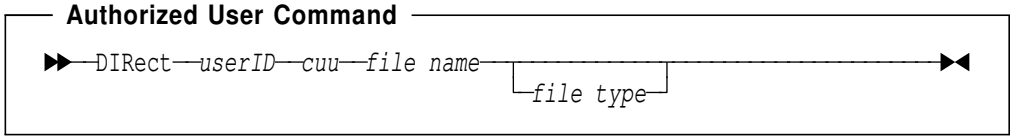
cuu Identifies the virtual address of the minidisk for which the backup file is to be removed from the catalog.

file name Specifies the name of the file which is to be deleted from the catalog (UC) file. This is the creation date returned by a QUERY command in Julian date format.

file type Specifies the type of the file which is to be deleted from the catalog (UC) file. This is the backup sequence returned by a QUERY command.

10.2.13 DIRECT

The DIRECT command tells BARS/VM the location and fileID of the VM source directory.



Operands

userID Identifies the owner of the minidisk which contains the source directory.

cuu Identifies the virtual address of the minidisk which contains the source directory.

file name Identifies the file name of the source directory.

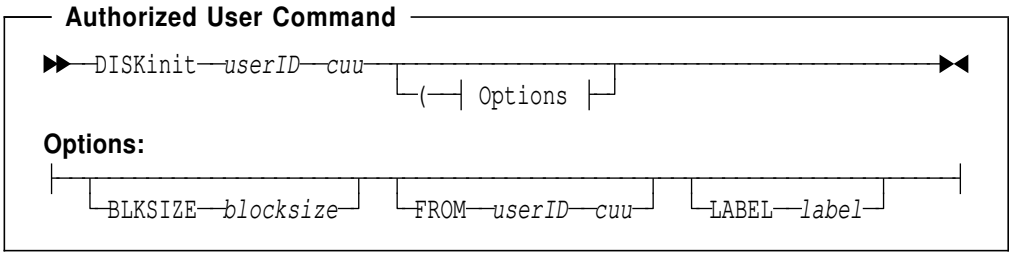
file type Identifies the file type of the source directory. If this is not specified, then it defaults to DIRECT.

Usage Notes:

1. For VM/DIRMAINT Release 3.0, the location of the source directory will be the DIRMAINT 195 minidisk and the fileID will be USER DIRECT.
2. For VM/DIRMAINT Release 4.0, the source directory may be in “clustered” format. This means that BARS/VM will not be able to read the files that comprise the directory. Therefore it is suggested that the backup copy (in normal format) be used instead. This is found on the DIRMAINT 193 minidisk and the fileID will be USER BACKUP. It is recommended that this backup is performed just before the commencement of each backup cycle (see the BACKUP= parameter in the DIRMAINT DATA file).
3. For VM/DIRMAINT Release 5.0, the backup copy is found on the DIRMAINT 1DB minidisk and the fileID will be USER BACKUP.
4. The referenced file does not need to be sorted as BARS/VM will sort it as required.

10.2.14 DISKINIT

The BARS/VM DISKINIT command allows an authorized user to initialize a minidisk. The minidisk may be CMS formatted and, optionally, files copied to it from another minidisk. A minidisk may also be initialized from a non-CMS minidisk.



Operands

userID Is the user ID whose minidisk is to be initialized.

cuu Is the minidisk to be initialized.

Options

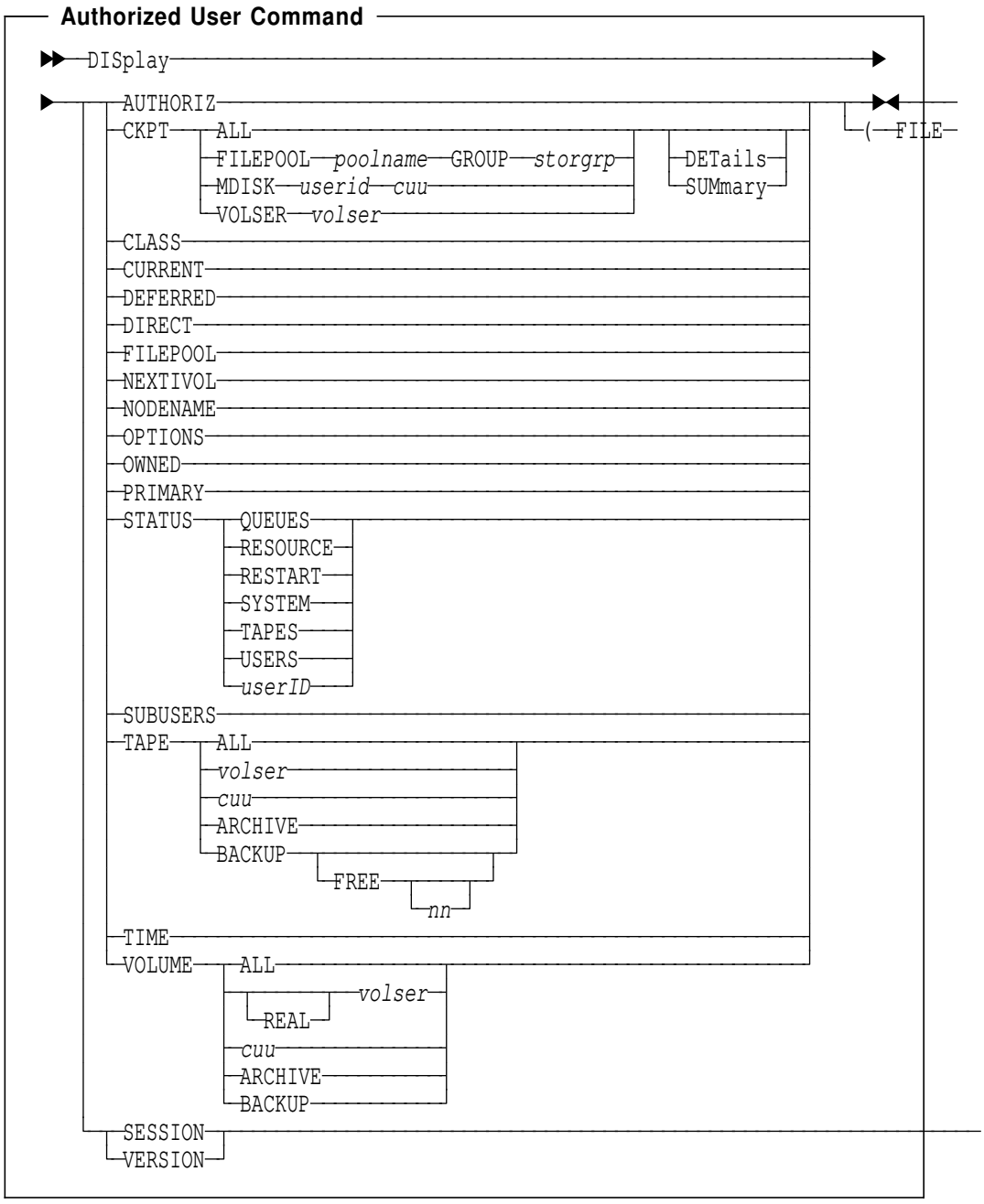
BLKSIZE Optionally specifies a block size for the case where the minidisk is to be initialized as a CMS minidisk. If a block size is not specified, the minidisk is given the same one as the source minidisk if one is specified, otherwise it is formatted with a block size of 2K. The blocksize may be specified in either format supported by the CMS FORMAT command - i.e. 4K and 4096 are both allowed.

FROM Specifies that the minidisk is to be initialized (i.e. copied) from another minidisk. The user ID and cuu of this source minidisk must follow. If the source minidisk is CMS, the target minidisk is formatted and files (if any) copied from the source minidisk. If the source minidisk is non-CMS, data is copied to the target with DDR, and hence the source and target minidisks must be of the same device type and have the same number of cylinders.

LABEL Optionally specifies a label for the case where the minidisk is to be initialized as a CMS minidisk. The label name must follow. If a label is not specified, the minidisk is given the same label as the source minidisk if one is specified, otherwise it is labelled MDcuu where cuu is the minidisk address.

10.2.15 DISPLAY

The DISPLAY command displays general BARS/VM information.



Operands

AUTHORIZ Displays the current user IDs defined by the AUTHORIZ command.

CKPT Displays checkpoint/restart information for any pending RESTORE or RETRIEVE operations.

- **ALL** All checkpoint information is to be displayed
- **FILEPOOL** Only checkpoint information for a RESTORE of the the specified storage group is to be displayed

MDISK Only checkpoint information for a RESTORE or RETRIEVE of the specified minidisk is to be displayed

VOLSER Only checkpoint information for a RESTORE of the specified real volume is to be displayed

DETails Requests a list of all restore steps

SUMmary Requests a summary of the restore steps

CLASS Displays the archive classes and their meanings. These classes are used on the APUT command.

CURRENT Displays the current backup cycle date and time.

DEFERRED Displays all outstanding deferred commands.

DIRECT Displays the current values defined by the DIRECT command.

FILEPOOL Displays the Shared File System file pools which are specified for backup.

NEXTIVOL Displays the internal volume serial number which will be given to the next unused minidisk or tape selected for archive or backup use. The value is incremented each time a new tape or minidisk is used.

NODENAME Displays the current node names defined by the NODENAME command.

OPTIONS Displays the current settings of the installation options.

OWNED Displays the current owned user IDs together with their owner user ID as defined by the OWNED command.

PRIMARY Displays the current volumes and device types defined by the PRIMARY command.

SESSION Displays the current session number. This is a number from 0 to 255 which is incremented by one each time BARS/VM is restarted. After 255 it starts again at 0.

STATUS Display various status information, depending on the following qualifiers:

QUEUES Lists all pending commands which are queued but not yet started execution. These commands may be cancelled, held, or released using the CQUEUE command.

RESOURCE Lists the SUBUSERS using or waiting to use BARS/VM resources.

RESTART Shows the total number of subuser restarts for this BARS/VM session, together with the maximum number allowed as defined by ERESTART.

SYSTEM Gives information about whether backup, volume restore or DBR is taking place and, if so, the user ID of the controller virtual machine. An additional message will be issued if reorg is active during the backup cycle.

TAPES Shows mounted tape volumes.

USERS Lists the current status of all subusers. See 3.3, “Virtual Machines Overview” on page 11 for a description of the VMCF codes which are shown by DISPLAY STATUS USERS.

userID Lists the current status of the specified subuser user ID.

SUBUSERS Displays the current user IDs defined by the SUBUSERS command.

TAPE Displays the current status of backup or archive tapes, depending on the following qualifiers:

ALL All tapes will be shown.

volser Only the specified tape will be shown.

cuu Only the specified tape will be shown.

ARCHIVE All ARCHIVE tapes will be shown.

BACKUP All BACKUP tapes will be shown. If FREE is specified, only backup tapes which are not full will be shown.

TIME Displays current times defined by the TIME command.

VERSION Displays the version, release and modification level of the BARS/VM system.

VOLUME Displays the current status of backup or archive minidisks, depending on the following qualifiers:

ALL All minidisks will be shown.

volser Only the specified minidisk will be shown. The *volser* is the CMS label of the minidisk.

REAL volser All qualifying minidisks will be shown. The *volser* is the real volume label of the volume containing the minidisks.

cuu Only the specified minidisk will be shown.

ARCHIVE All ARCHIVE minidisks will be shown.

BACKUP All BACKUP minidisks will be shown.

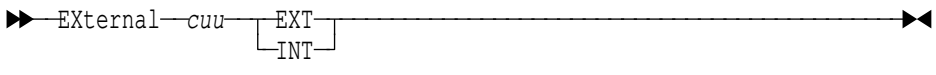
Options

FILE Specifies that the result of the command is to be sent as a file to the command issuer's card reader in **NETDATA** format.

10.2.16 EXTERNAL

The EXTERNAL command marks an archive minidisk as external, i.e. the volume on which it resides is not available for immediate mounting. It may also be used to mark the minidisk as internal again.

Authorized User Command



Operands

cuu Specifies the cuu of the archive minidisk.

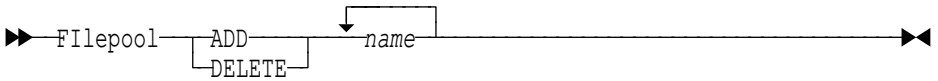
EXT Specifies that the minidisk is to be marked as an external minidisk.

INT Specifies that the minidisk is to be marked as an internal minidisk.

10.2.17 FILEPOOL

The FILEPOOL command adds or deletes a file pool to a list of file pools eligible for backup.

Authorized User Command



Operands

ADD Adds one or more file pool IDs to the list of those eligible for backup.

DELETE Deletes one or more file pool IDs from the list.

name Is the file pool identifier(s) to be added or deleted.

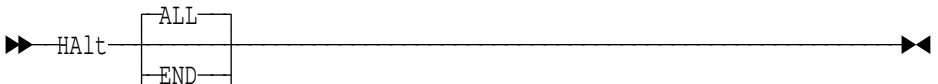
Usage Notes:

1. If the DELETE operand is specified then **ALL** backup data for files residing in the specified file pool(s) will be made **irretrievable**.
2. The filepool name must be specified without a trailing colon (":").

10.2.18 HALT

Use the HALT command to control whether BARS/VM accepts communications from general and authorized users. Its normal use is to prevent the execution of commands while maintenance is being carried out.

Authorized User Command



Operands

ALL Suspends communications with all users except the BARS/VM manager user ID (EMUSERID). This is the default.

END Allows communications to resume.

USER Suspends communications with general users.

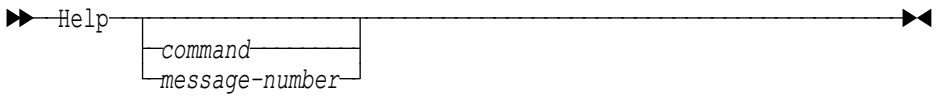
Usage Notes:

1. The HALT command may only be issued from the BARS/VM manager user ID (EMUSERID).

10.2.19 HELP

The HELP command describes a specified command or message. If no command or message number is specified, a list of commands is provided with the option of obtaining further help for any of those.

Authorized User Command



Operands

command The name of a command for which help is required.

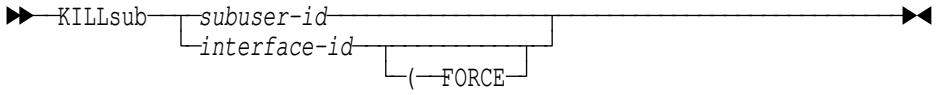
message-number

A message number for which help is required. The format of 'message-number' is **HDFnnnns** where **nnnn** is the numeric part of the message and **s** is the severity. For example for message **HDFRET0303E** the corresponding message-number for HELP would be **HDF0303E**.

10.2.20 KILLSUB

The KILLSUB command is used for two purposes. The first is only for use in error situations, where a **CP FORCE** command is issued for the specified subuser and all resources allocated to it are freed. The second is for restarting the entire BARS/VM subsystem, and is normally used after a code update or change in one of the installation options. The default action is to reject this command with message 1068 if any of the subusers are busy.

Authorized User Command



Operands

subuser-id Specifies user ID of the subuser to be logged off.

interface-id Specifies user ID of the interface virtual machine.

Options

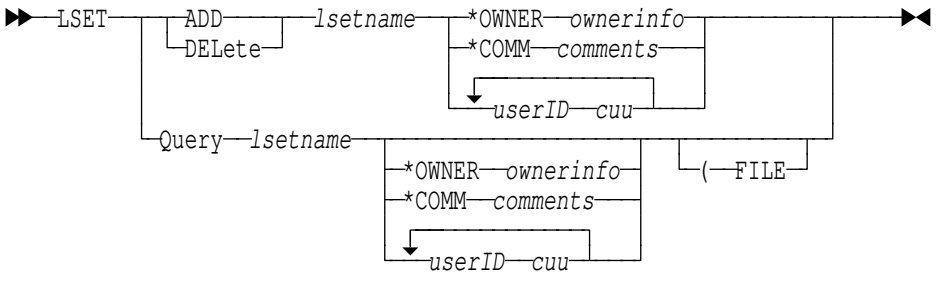
FORCE Specifies that the subsystem be restarted even if one or more subusers are busy.

10.2.21 LSET

The LSET command defines a list of user IDs and/or minidisks that form a logical set (herein referred to as LSET). An LSET is used to group together data on DBR tapes as closely as possible. The definition of LSETs also allows for the creation of application based DBR tapes in addition to system-wide DBR dumps.

Note: An LSET dump will always dump all data associated with the minidisks defined by the LSET - i.e. there is no base/delta logic applied to LSETs.

Authorized User Command



Operands

ADD Specifies that an entry is to be added to the LSET definition identified via the *lsetname*.

Note: To create an LSET you **must** first issue an LSET ADD command with a `*OWNER` parameter.

DELETE Specifies that matching entries should be deleted from the LSET definition identified via the *lsetname*.

QUERY Specifies that matching entries should be displayed from the LSET identified via the *lsetname*. `*` or `ALL` may be specified for the *lsetname* to display a list of all known LSETs. If no further operands are specified, only the first record is shown for each LSET found. If further operands are specified, then any matching entries within the LSET will be shown. Specify `* *` to match all entries within the LSET.

lsetname The name by which this LSET is known. The allowable character set is the same as for normal CMS file names.

***OWNER ownerinfo**
Adds some meaningful information to identify an owner of an LSET by the installation.

***COMM comments**
Adds some meaningful comments to an LSET by the installation.

userID cuu Specifies pairs of user IDs and minidisks.

Options

FILE Specifies that the result of the command is to be sent as a file to the command issuer's card reader in **NETDATA** format.

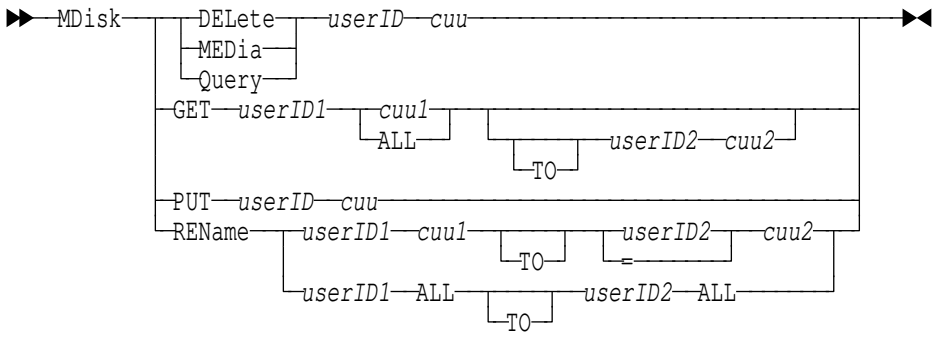
Usage Notes:

1. *lsetname*, *userID* and *cuu* may be specified using the ***** and **%** notation as described in 8.1.3, “How to Read a Syntax Diagram” on page 52.
2. The `*OWNER` and `*COMM` records are tokenized (i.e. translated to upper case and truncated to 8 characters).

10.2.22 MDISK

The MDISK command is used to perform minidisk archive functions. These allow a specified user's minidisks to be copied to or from archive storage. There is no relationship between minidisk archive and file archive.

Authorized User Command



Operands

DELETE Specifies that the following user ID's minidisk(s) should be deleted from archive storage.

userID Defines the user ID which owns the disk.

cuu Defines the cuu of the disk belonging to 'userID'

ALL Specifies that the requested operation is to be performed on all minidisks owned by the user ID(s).

MEDIA Displays messages to show whether the data for the specified minidisks resides on DASD, TAPE or both. The acknowledgement file will contain return code 0 (SUCCESSFUL) for DASD, 1 for TAPE or 2 for both.

QUERY Displays information about archived minidisks belonging to the specified user ID.

GET Specifies that the following user ID's minidisk(s) should be restored from archive storage. The optional destination userID/cuu parameters may be used to restore to a different minidisk than the one archived (provided the number of cylinders and device type are the same). Wildcards may not be specified in the userID fields.

PUT Specifies that the following user ID's minidisk should be copied to archive storage. A unique user ID must be specified, and 'ALL' may not be specified for the *cuu* field.

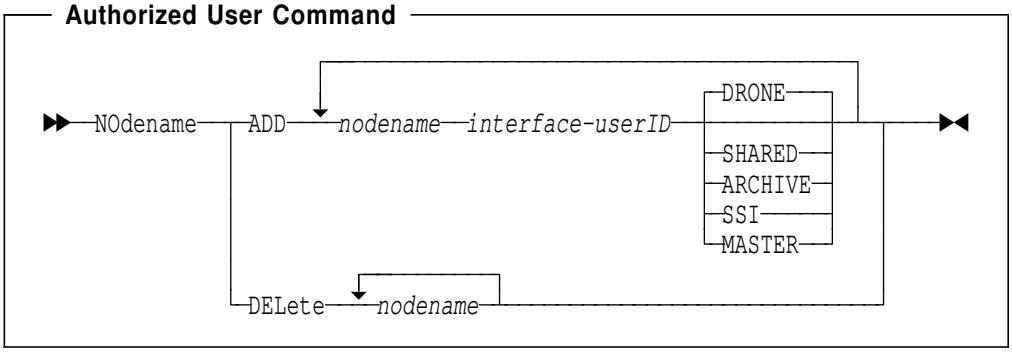
RENAME Specifies that the specified minidisk should be renamed. If **=** is specified for the second user ID, then just the cuu for that minidisk is changed. If the user ID is different, then the archived minidisks are effectively transferred to the new owner. In this situation, a cuu of ALL may specified to transfer all disks. Wildcards may not be specified in the userID fields.

Usage Notes:

1. An acknowledgement file will always be returned, unless the command was QUERY or there was a syntax error.

10.2.23 NODENAME

The NODENAME command defines which remote nodes are allowed to send commands for execution, or files for backup, to the current node BARS/VM. Several types of node are recognized: a DRONE is a node running BARS/VM which sends files to the current Host BARS/VM for backup (see the section on remote backup and retrieval); users at a SHARED node are allowed to send any general user commands to the current BARS/VM for execution; users at an ARCHIVE node are allowed to send archive commands. MASTER and SSI nodes are defined for a shared data or Single System Image (SSI) environment and define which node has read-write access to BARS/VM minidisks (the MASTER) and which have read-only access (SSI).



Operands

ADD Specifies that the following node name or list of node names is to be added.

DELETE Specifies that the following node name or list of node names is to be deleted.

nodename Specifies the name of the node.

interface-userID
Specifies the user ID of the interface virtual machine of the BARS/VM system at the specified node.

DRONE Defines the node as one which is allowed to send files to the current BARS/VM for remote backup. Files sent from nodes not thus defined are rejected.

SHARED Defines the node as one from which users are allowed to issue any general user commands to the current BARS/VM system for execution. Commands from undefined nodes are rejected.

ARCHIVE Defines the node as one from which users are allowed to send general user archive commands to the current BARS/VM execution. Note the similarity with SHARED but here only archive commands are accepted.

SSI Defines the node as one in a shared data environment which has read/write access to its BARS/VM minidisks and read/only access to other participating BARS/VM systems to correctly route all user commands to the processing BARS/VM system transparently.

MASTER Defines the node as one in a shared data environment which has read-write access to BARS/VM minidisks.

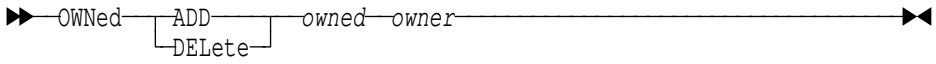
Usage Notes:

1. If specifying a list of node names on an ADD command, and DRONE is the desired node type, then it must be specified and not left to default.
2. The new list of node names becomes effective on the **next initialization** of BARS/VM.

10.2.24 OWNED

Use the OWNED command to grant one user ID (the owner) authority to issue certain BARS/VM commands for another (the owned) user ID. Once authorized, the owner may use the FOR option on general user BARS/VM commands to issue a command for the owned user ID. See 8.4, “Authorizing other users” on page 85 for a possible alternative to the OWNED command.

Authorized User Command



Operands

ADD Specifies that an entry is to be added to the owned list.

DELETE Specifies that an entry is to be deleted to the owned list.

owned The user ID for which authority is to be granted.

owner The user ID which is authorized to act for the "owned" user ID.

Usage Notes:

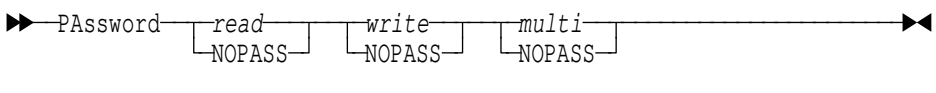
1. Note that an owner user ID has access to ***all*** data belonging to an owned user ID.
2. The authority granted via the OWNED command applies to both the backup and archive components of BARS/VM.
3. When adding an entry, the two user IDs must be different.
4. Both user IDs must be in full, wildcards are not allowed.
5. When the last entry is deleted, the OWNED NAMES file is also deleted. This is indicated by message 1074I.

10.2.25 PASSWORD

The PASSWORD command may be used to define read, write, and multiple-write passwords for all minidisks LINKed by BARS/VM user IDs.

If specified, BARS/VM uses these passwords for all LINKs to be done for minidisk access. The passwords are only required when a master password modification to CP is used as the method of allowing BARS/VM to LINK to all minidisks. See “Requirements to CP” on page 209 for a description of the CP modifications needed to allow BARS/VM to LINK to all minidisks.

Authorized User Command



Operands

read Defines the read password

write Defines the write password

multi Defines the multi write password

NOPASS Specifies that no password is to be used. This operand may be specified instead of any of the above passwords. NOPASS may be used to remove a previously defined password.

Usage Notes:

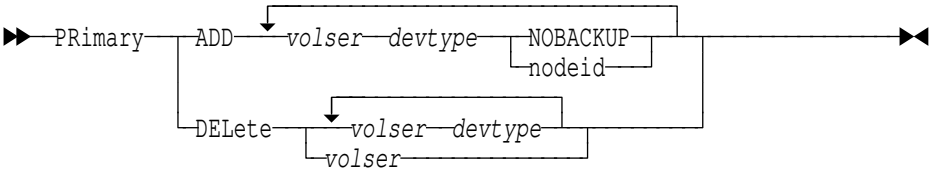
1. The new passwords become effective on the next initialization of BARS/VM.

10.2.26 PRIMARY

The PRIMARY command defines the DASD volumes that are **excluded** from the backup process. All minidisks defined on volumes defined in the VM source directory are eligible for backup unless specifically excluded.

Within an SSI (Single System Image) environment, the PRIMARY command has to be used to define the nodeid from which the backup of each DASD volume should be performed.

Authorized User Command



Operands

ADD Specifies volume serial(s) are to be added to the primary list.

DELETE Specifies volume serial(s) are to be deleted from the primary list.

volser Defines the DASD volume(s).

devtype Defines the DASD device type. Valid device types are 2305, 2311, 2314, 2319, 3330, 3340, 3350, 3370, 3375, 3380, 3390 or FB-512.

NOBACKUP Specifies that the DASD volume(s) are to be excluded from the backup process.

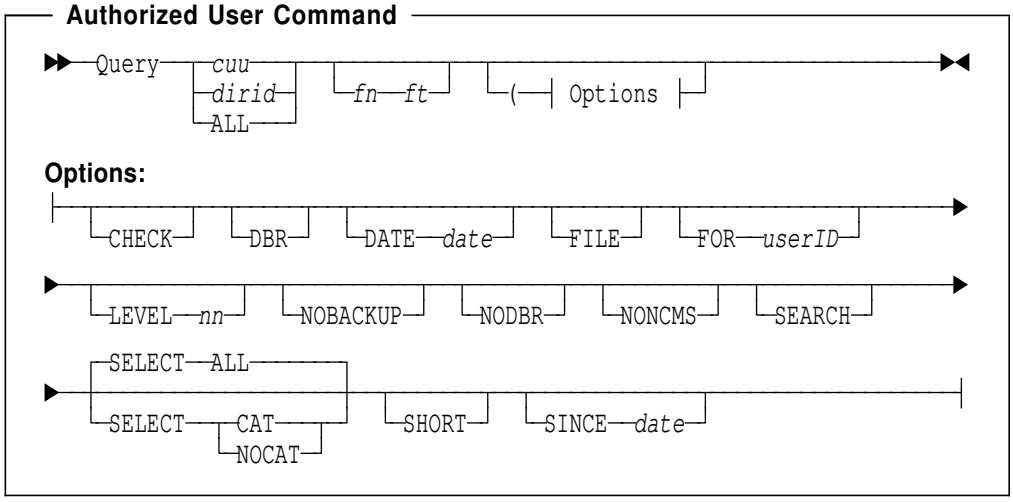
nodeid Specifies the system (as specified in the 'VMBARS NETID' and the OPTIONS files) where minidisks on those DASD volume(s) are eligible for backup. This only applies in an SSI environment.

Usage Notes:

1. volser and devtype may be specified using the * and % notation as described in 8.1.3, “How to Read a Syntax Diagram” on page 52.
2. All minidisks on the DASD volume(s) excluded by a PRIMARY command are excluded from the backup process. Therefore any minidisks which were previously being backed up are treated as if they had been deleted, and the backup data will be erased after a installation defined time period.

10.2.27 QUERY

The QUERY command displays backup information for minidisks and SFS directories.



Operands

cuu Defines the minidisk address for which backup information should be displayed.

dirid Identifies the SFS directory for which backup information is to be displayed.

ALL Specifies that all minidisks and SFS directories belonging to the user ID specified in the FOR option are to be displayed.

fn ft Specify particular CMS files. If these are specified then BARS/VM displays additional information for the particular file or files specified.

Options

CHECK Specifies that the result of the QUERY command is stripped down to those which have received **no backup since the last backup cycle date** or the date specified on the SINCE option.

DATE Restricts the display of files to those files which were backed on the specified date. The date must be of the form ddmmmyy, where dd is the day number, mmm is the mnemonic month and yy the year number, e.g. 10JAN96. The DATE operand must not be used if SINCE is specified.

DBR Specifies that information only be returned for minidisks for which a BACKUP command with the DBR option has been issued.

FILE Specifies that the result of the QUERY command is to be sent as a file to the command issuer's card reader in **NETDATA** format.

FOR Identifies the user ID owning the minidisks and/or SFS directories for which backup information is to be displayed. This option is not allowed if an SFS "dirid" is specified since the user ID is derived from the dirid.

LEVEL Specifies that only a particular backup level (*nn*) is to be displayed.

NOBACKUP Specifies that information only be returned for minidisks or SFS directories for which a BACKUP ... NOBACKUP, ERASE or STOP has been issued.

NODBR Specifies that information only be returned for minidisks for which a BACKUP command with the DBR option has not been issued (or a BACKUP command with the NODBR option has been issued).

NONCMS Is for minidisks which are being backed up both as CMS and non-CMS minidisks and specifies that the command is to display the non-CMS backup only. For such a CMS minidisk, a QUERY command without the NONCMS option displays only the CMS backup. A QUERY with NONCMS displays only the non-CMS backup. In both cases the display shows the minidisk as CMS formatted.

SEARCH Specifies that the result of the QUERY command is stripped down to EXEC processable lines which can be used to identify and manage possible migration files. For example, if a new product level needs parameter changes in user settable profiles this command can identify where a change is needed.

The output is in the format: User ID Cuu : Fn Ft Fmn Backup-level backup-date backup-sequence backup-volume

SELECT Determines which information is selected for display. SELECT CAT limits the display to objects for which there exists a catalog entry. SELECT NOCAT limits it to objects with no catalog. The default, SELECT ALL, displays all objects.

SHORT Specifies that the result of the QUERY is to be a single message for each minidisk or SFS directory requested giving the format of the item and whether backup will take place and at what interval. The SHORT option is useful when specifying **ALL** to reduce the number of messages received.

SINCE Restricts the display of files to those files which have been backed up on or after the specified date. The date must be of the form ddmmmyy, where dd is the day number, mmm is the mnemonic month and yy the year number, e.g. 10JAN96. The SINCE operand must not be used if DATE is specified.

Usage Notes:

1. UserID, file name and file type may be specified using the ***** and **%** notation as described in 8.1.3, “How to Read a Syntax Diagram” on page 52.
2. Specifying `DBR` and `NODBR`, while allowed, will result in no information being returned, since a minidisk cannot be both eligible and ineligible at the same time.

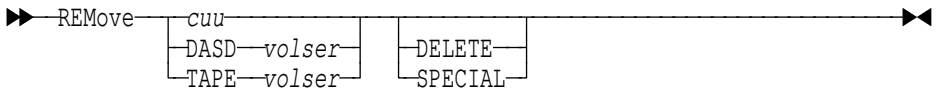
10.2.28 REMOVE

The REMOVE command either eliminates the specified backup or archive minidisk or tape from use by BARS/VM or checks the validity of the data on a backup minidisk. When a backup volume is to be eliminated, the command issuer may request that the data be moved to other backup minidisks or tapes. See chapter 12.2, “Maintain BACKUP Volumes: REMOVE and REUSE” on page 154 for a more detailed description of the removal/reuse process.

The behavior of the command depends on the type of minidisk or tape being removed:

- For a backup minidisk or tape, when neither DELETE nor SPECIAL is specified, the valid data on the specified minidisk or tape is moved to other backup minidisks or tapes if space is available and the data can be moved. The catalog (UC) files are then updated to point to the new location of the data.
- For an archive minidisk no data is moved and no catalog files are updated. The archive minidisk is simply marked as archive and reserved for reuse. When the minidisk is restored from backup a REUSE command will set it to the state is was in before the REMOVE.
- For an archive tape, no data is moved and no catalog files are updated. The tape is marked as unused, reserved for reuse, and, if it was a twin, it is marked no longer a twin. The tape should normally be deleted after such a remove.

Authorized User Command



Operands

cuu Defines the minidisk or tape to be removed.

DASD volser Defines the minidisk to be removed by its BARS/VM generated internal CMS label.

TAPE volser Defines the tape to be removed by its BARS/VM generated internal label.

DELETE Is valid only for backup minidisks and tapes. It specifies that no data is to be moved but that the catalog (UC) files are to be updated to erase entries for files backed up to this backup volume so that they will be backed up during the next backup cycle. The DELETE operand is used to remove an unusable backup volume.

SPECIAL Is valid only for backup minidisks. It requests that the command verify that all files that, according to the catalogs, should exist on the minidisk do actually exist. If not, the file entry in the catalog (UC) for each missing file is deleted and the next backup cycle creates new backup files for these.

Usage Notes:

1. Because a REMOVE command may need lengthy read/write access to the 191 catalog minidisk, other commands, including other REMOVE commands, may be held up. This does not occur, however, during tape operations and while waiting for tape mounts.
2. DELETE may be used to remove catalogue entries which reference backup minidisks or tapes which do not exist in either BACKUP or TAPE VOLUMES. In this situation, the *volser* form must be used, since no *cuu* exists.

10.2.29 RENAME

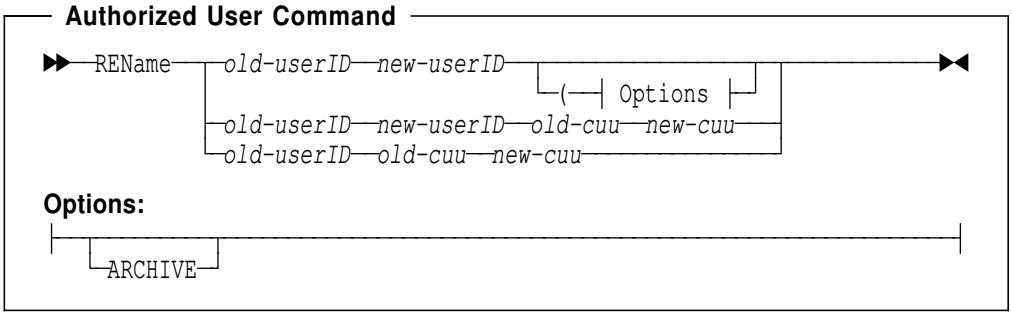
The RENAME command allows BARS/VM to reflect directory and SFS enrollment changes.

The first form of the command allows associated backup and archive data to be renamed following the rename of a user ID. It also allows a just a user's archive to be renamed to a new userID or pool name.

The second form of the command renames the backup data for a minidisk which is transferred to another user ID.

The third form of the command is used to reflect a change in the virtual address (cuu) of a minidisk.

The RENAME command should be used before the DIRECT(XA) updates the online VM directory with the change.



Operands

old-userID Specifies the current user ID, the one to be renamed.

new-userID Specifies the new user ID.

old-cuu Specifies the cuu of a minidisk whose backup is to be renamed.

new-cuu Specifies the new cuu for the minidisk.

Options

ARCHIVE Specifies that only the archive is to be renamed. The backup data remains unchanged.

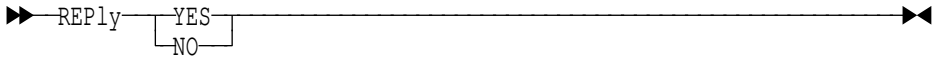
Usage Notes:

1. Minidisk archive is not affected by this command. You must issue a `MDISK RENAME` to reflect a user ID rename.
2. For a user ID rename, entries in the OWNED NAMES file are updated. If the new user ID already exists in OWNED NAMES, then message 0401 is issued, the OWNED NAMES file is not updated, and RENAME processing continues.

10.2.30 REPLY

The REPLY command allows the Operator or the BARS/VM Manager user ID to answer questions from the BARS/VM system when prompted.

Authorized User Command



Operands

YES Answers YES to the question.

NO Answers NO to the question.

Usage Notes:

1. The REPLY command will only be accepted if BARS/VM has prompted the manager or Operator user ID with a message in the form of a question.

10.2.31 RESTORE

The RESTORE command restores the specified primary DASD volume, SFS storage group, minidisk or SFS directories with the latest backup copy.

Please read the associated "Usage Notes" before using this command.

Primary volume restore

For a total primary volume restore all minidisks for which backup exists are restored. CMS minidisks are formatted as they were at the time they were last backed up unless the SINCE option is specified (see below) in which case they are assumed to be already formatted. See note 1 on page 132 before starting a volume restore. Also see 4.4.3, "Primary Volume Restore" on page 21 for information on primary volume restore.

Storage group restore

For a storage group restore, the data in all file spaces in the group for which backup exists are restored. See 4.4.4, "Storage Group Restore" on page 22 for information on SFS backup and restore.

Minidisk restore

For a single minidisk restore, a non-CMS minidisk is restored with DDR to its state at the last backup and, unless SINCE or RESTART is specified, a CMS minidisk is formatted as it was when last backed up and all files which existed on the minidisk when last backed up are copied to the disk. If SINCE is specified, the minidisk is not formatted and only files backed up since the specified date are restored. The RESTART option also suppresses the formatting of the minidisk. In this case only files which do not already exist on the minidisk are restored.

SFS restore

A restore of an SFS directory includes the restoration of all files within the directory and the creation of any aliases (unless the NOALIAS option is specified) which existed at the time of the last backup cycle. Directory and file authorizations are granted unless the NOAUTH option is specified. If the SUBDIR option is specified, then any sub-directories will also be created and restored. The specified directory must not already exist or the command will be rejected (unless the NOCREATE option is specified).

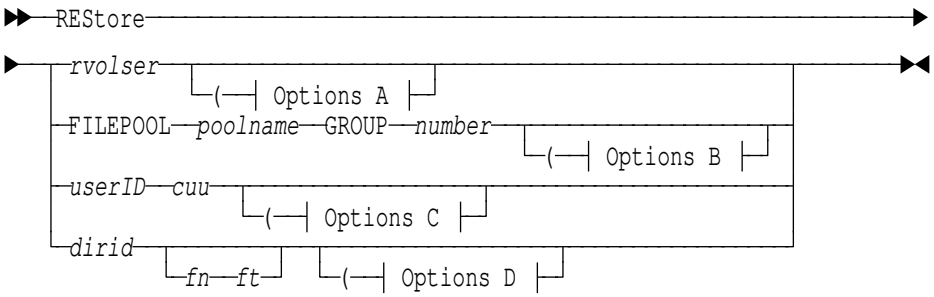
If a top directory is specified, then the user will be enrolled into the appropriate SFS storage group within the file pool. To restore the entire filespace specify the SUBDIR option as well. BARS/VM will obtain an exclusive disable lock on the filespace during the restore to prevent access by other users. If the user is already enrolled in the file pool the command will be rejected (unless the NOCREATE option is specified).

If *fn* and *ft* are specified then this specifies a file or alias restore. The specified objects are restored to the directory, which must already exist. Subdirectories are not processed in this case. If the file already exists in the directory the command will be rejected (unless the NOCREATE option is specified).

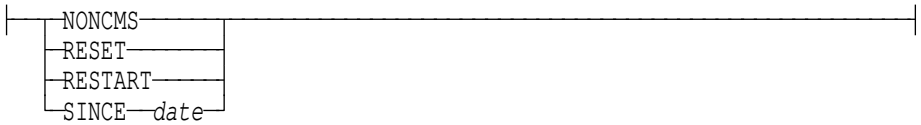
Checkpointing

The use of checkpointing during a restore of a primary volume, storage group or minidisk allows the command to be restarted in the event of a failure without having to repeat steps that have already been processed. Checkpointing always takes place for volume and storage group restores, and is optional for minidisk restores (there is a slight performance penalty associated with minidisk checkpointing). See 4.4.5, “Restore Checkpointing” on page 23 for a description of checkpointing.

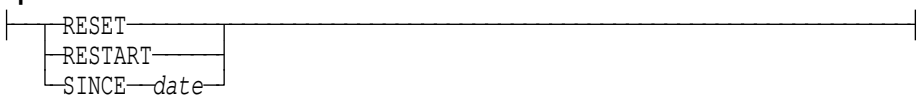
Authorized User Command



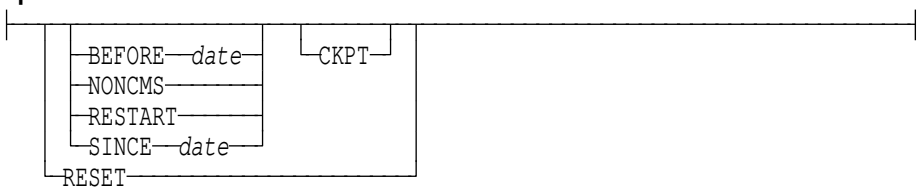
Options A:



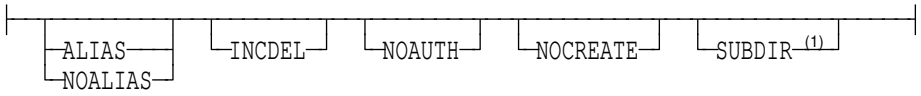
Options B:



Options C:



Options D:



Note:
[1] This option may not be used if *fn* and *ft* are specified.

Operands

poolname Specifies the name of the file pool whose storage group is to be restored.

number Specifies the number of the storage group to be restored.

rvolser Specifies the real volume serial number of a primary volume to be restored.

userID Specifies the user ID for the restore of a single minidisk.

cuu Specifies the cuu for the restore of a single minidisk.

dirid Specifies an SFS directory to be restored, or, if fn and ft are specified, the SFS directory into which the specified files are to be restored.

fn Identifies the file name of a file or alias to be restored. Omit *fn* and *ft* for a directory or filespace level restore.

ft Identifies the file type of a file or alias to be restored.

Options

BEFORE date This may be specified for a minidisk restore. It causes a restore of the latest backup copy of each file which was taken before the specified date. This allows the minidisk to be restored as near as possible to the state is was in before that date. The restore process runs as a normal restore except that

- Files that do not qualify, because all backup levels were created on or after the specified date, are excluded. It is possible that a file existed on the minidisk before the specified date but, because it changes frequently, all backup copies were taken after the date. Such a file is not restored.
- It is not always the level 0 backup of a file that is restored. If the level 0 backup was created on or after the specified date but level 1 was created before, then the level 1 is restored. Higher levels are used in the same way if necessary.
- Something similar to the following sequence of events may occur

  1. A user creates a file on day 1
  2. The file is backed up that night
  3. The user deletes the file on day 4
  4. That night BARS/VM marks the backup for deletion after ERASDAYS days
  5. The user reinstates file on day 7
  6. That night BARS/VM no longer marks the backup for deletion
  7. A RESTORE ... BEFORE day 6 is issued

  The file is restored because at the time the RESTORE command is issued, the backup copy is active and was created before day 6. However on day 5 the file did not exist on the minidisk.

CKPT Specifies that the minidisk restore is to be checkpointed.

ALIAS Specifies that only aliases are to be restored. This option is useful when backup exists for a file and an alias of the same name. It allows only the alias to be restored. It is also useful for restoring only aliases after all files have been restored. This ensures that the aliases can be recreated successfully.

NOALIAS Specifies that aliases are not to be restored. This option is most useful when restoring directories which may have had aliases to files in other directories which have not yet been restored. After the files have been restored, the aliases can be separately restored using the ALIAS option.

INCDEL Allows files and aliases to be restored which are marked as deleted, i.e. did not exist in the directory when it was last backed up. If this operand is not specified, such files are not restored. This option should be used with care, since it will restore all known deleted files, regardless of when they were deleted.

NOAUTH Specifies that SFS directory and file authorizations are not to be granted as part of the restore process.

NOCREATE This option should be used with care since it assumes that the directory tree structure is unchanged since the last backup.

- For a filespace restore it specifies that the user is not to be enrolled in the filepool. This permits a filespace restore to be run for a user who is still enrolled
- For a filespace or directory level restore it specifies that no directories should be created
- For a file level restore it specifies that the restored files may replace existing files

NONCMS Is for CMS minidisks which have been backed up both as CMS and non-CMS minidisks and specifies that the minidisk is to be restored from the non-CMS backup.

RESET Causes removal of any checkpoint information associated with the specified real volume, storage group or minidisk. For example, if you wish to remove the checkpoint information for a real volume restore, re-issue the real volume restore command and add the RESET option.

RESTART For a real volume or storage group restore it will cause the restore process to use the checkpoint information (which must exist) to avoid repeating work already successfully completed. For a minidisk restore for which checkpoint information exists, it will cause the restore process to use the checkpoint information to avoid repeating work already successfully completed. Any options which were specified on the original restore (e.g. SINCE) **must** be specified as well.

For a minidisk restore for which no checkpoint informations exists, causes a restore of only those objects which do not already exist on the minidisk.

SINCE date Causes a restore of only those objects which were backed up on or since the specified date. This allows an **incremental restore** after a total volume restore. The date may be specified in the "OS" format, yyddd, or may be of the form ddmmmyy, where dd is the day number, mmm is the mnemonic month and yy the year number, e.g. 10JAN96.

SUBDIR Requests that all SFS sub-directories of the specified "dirid" also be restored. This option may not be used if *fn* and *ft* are specified.

Usage Notes:

1. For a total volume restore, BARS/VM processes the file defined by the `DIRECT` command before starting to determine the list of minidisks which should be restored. You should ensure that this file accurately represents the minidisks present on the volume at the time of the volume failure.
2. For a storage group restore, BARS/VM determines which users are enrolled in the group before starting to determine the list of file spaces to be restored. The minidisks that comprise the storage group must be in the correct format before the restore is done. Therefore any of the minidisks that may have been damaged must be formatted. See 4.4.4, “Storage Group

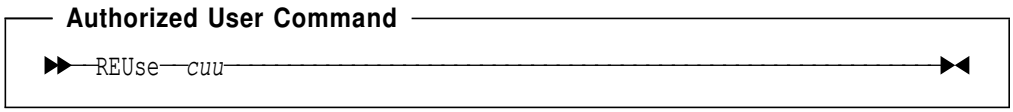
Restore" on page 22 for information on how to format and reserve the minidisks. Minidisks in the group which have not been damaged will already have the correct format.

3. On completion of a total volume or storage group restore the identities of some of the subusers will have been changed. In particular, the subuser which is normally the backup controller becomes a normal subuser and another subuser (the one which controlled the restore process) becomes the backup controller. This should cause no problem at backup time but it is recommended that BARS/VM be shut down and restarted so that the normal subuser assignment is restored. This makes things easier when, for example, the log messages are being read or processed.
4. When the SINCE option is specified, **no** files are erased from the users minidisk for which entries do not exist in the BARS/VM catalogs. This means there is a possibility that some files may be unable to be restored should a minidisk become full as a result.
5. See also the ERASEDAT option of the VMBUTIL command for restoring expired files.
6. A normal restore will not start if any associated checkpoint information exists.
7. Use DISPLAY CKPT to monitor progress of a total volume or storage group restore. See 4.4.5, "Restore Checkpointing" on page 23 for an example.
8. You cannot cancel individual steps of a RESTORE - i.e. you cannot use the RESET option of the minidisk RESTORE to cancel restoration of a minidisk which forms part of a full volume RESTORE.
9. At the start of the restore process, the controller id obtains a write link to all minidisks on the volume in an attempt to prevent users from linking their minidisks read-write. The subusers then use MW links to enable them to write data to the minidisks. Data integrity is maintained via internal resource queueing and because the controller id does not ACCESS the minidisks.
10. When restoring a filespace or directory sub-tree using the SUBDIR option, it is possible to receive message HDF303I for directories in the tree which do not contain any files.
11. See the SETSG option of the CATMAINT command for a possible use of filespace restore to move users between storage groups within a filepool (taking careful note of the associated usage notes).
12. The filepool name must be specified without a trailing colon (":").

10.2.32 REUSE

The REUSE command specifies that a backup or archive minidisk or tape which has previously been removed by use of the REMOVE command may now be reused.

In the case of a backup minidisk, it is now considered as a new unformatted backup minidisk and for a backup or archive tape, a new unlabeled tape. REUSE for an archive minidisk sets its status to that which existed before the REMOVE was done, i.e. to the status formatted and archive. See chapter 12.2, "Maintain BACKUP Volumes: REMOVE and REUSE" on page 154 for a more detailed description of the removal/reuse process.

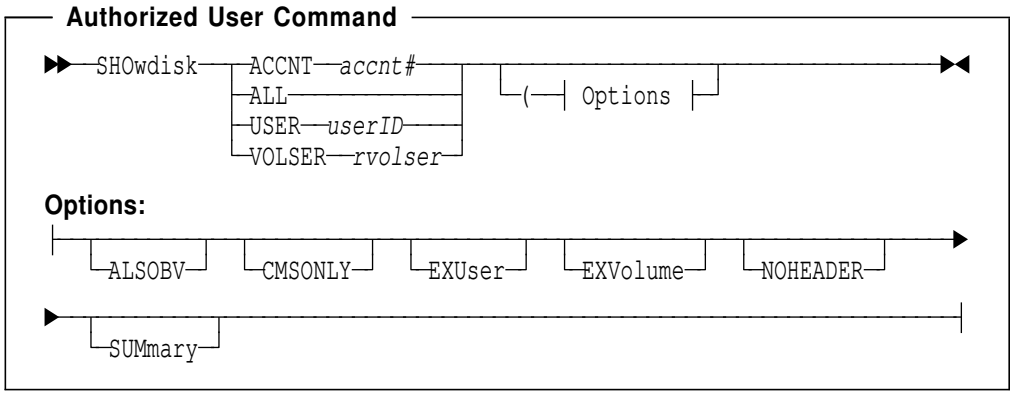


Operands

cuu Identifies the backup or archive volume which is to be made available for use.

10.2.33 SHOWDISK

The SHOWDISK command produces a user minidisk utilization report file from data collected by BARS/VM.



Operands

ACCNT Produces a report of all minidisks by the specified account number.

accnt# The account number (as specified in the VM source directory) to be used with ACCNT operand.

ALL Produces a report for all real DASD volumes. This is treated like **SHOWDISK VOLSER ***

USER Produces a report for the minidisks belonging to the specified user IDs.

VOLSER Produces a report for one or more real DASD volume(s). All minidisks defined in the directory for this volume will be included in the report. However, it will not include any **excluded volumes**.

Options

ALSOBV Produces a report information including BARS/VM ENUSERID's minidisks which are excluded without this option specified.

CMSONLY Produces report information only for CMS minidisk which have backup information in BARS/VM.

EXUser Produces a file with all minidisks defined in control file VMHDFS EXCLUDE excluded (overlapped minidisks). See Figure 17 on page 135 for the format of this file.

EXVolume Produces a file with all real volumes defined in control file VMHDFS EXCLUDE excluded (shared volumes). See Figure 17 on page 135 for the format of this file.

NOHEADER Produces report information without any header or sum lines. This is useful if the output file is to be sorted afterwards with specific search arguments.

SUMmary Produces a file with only summary lines. This implies the option ALSOBV and excludes the option CMSONLY. SUMMARY should only be specified with the **VOLSER** operand to obtain reasonable output.

Usage Notes:

1. The accnt#, userID and rvolser operands may be specified using the * and % notation as described in 8.1.3, “How to Read a Syntax Diagram” on page 52.
2. The report file is sent back to the caller in NETDATA format.

VMHDFS EXCLUDE file: This file is only used by the SHOWDISK command, and must reside on the INTERFACE 191 disk with mode 2.

```
EXCLUDE_ALWAYS
SYSVOLS *


EXCLUDE_USER

BARSDDR ALL READMVT ALL
MSSVMS *
RUMVM 530 RUNVM 531 RUNVM 532  RUNVM 534


EXCLUDE_VOLSER

CBVM* * CXVM* * CZVM* * CEVM* *
```

Figure 17. Layout of the Control File VMHDFS EXCLUDE

The EXCLUDE_xxxxxx tags are used to define the use of the data lines which follow them. The same tags may appear more than once in the file. The meanings of the tags are as follows:

EXCLUDE_ALWAYS All minidisks which match the patterns listed in this section will be excluded from any SHOWDISK report regardless of the SHOWDISK options. The patterns are made up of *userID cuu* pairs.

EXCLUDE_USER All minidisks which match the patterns listed in this section will be excluded from any SHOWDISK report only if the EXUSER option is specified. The patterns are made up of *userID cuu* pairs.

EXCLUDE_VOLSER All minidisks which reside on real volumes whose volsers match the patterns listed in this section will be excluded from any SHOWDISK report only if the EXVOLUME option is specified. The patterns are made up of *volser* * pairs (the “*” is a place-holder).

Notes:

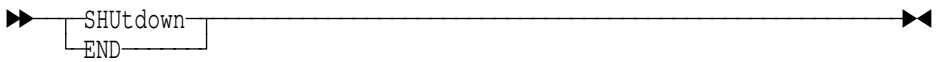
1. There can be no comments in the file.
2. Blank lines may be included.
3. There may be more than one pattern pair per data line.
4. `ALL` or `*` both match any cuu.
5. The userID and cuu pattern values may be specified using the * and % notation as described in 8.1.3, “How to Read a Syntax Diagram” on page 52.
6. Every minidisk eligible for inclusion in the SHOWDISK report is matched against the entries in this file. The patterns are checked in the order they appear in the file. The order has no bearing on the results, only on the efficiency of the matching operation.

10.2.34 SHUTDOWN

The SHUTDOWN command stops BARS/VM and causes all the BARS/VM virtual machines to log off. If backup is in progress, backup stops when each subuser has completed the task it is currently performing. At that time all subusers are logged off and the Interface virtual machine logs off after sending a 'SHUTDOWN command complete' message. Note that the stopping of backup could take some time if a minidisk or tape reorganization is in progress.

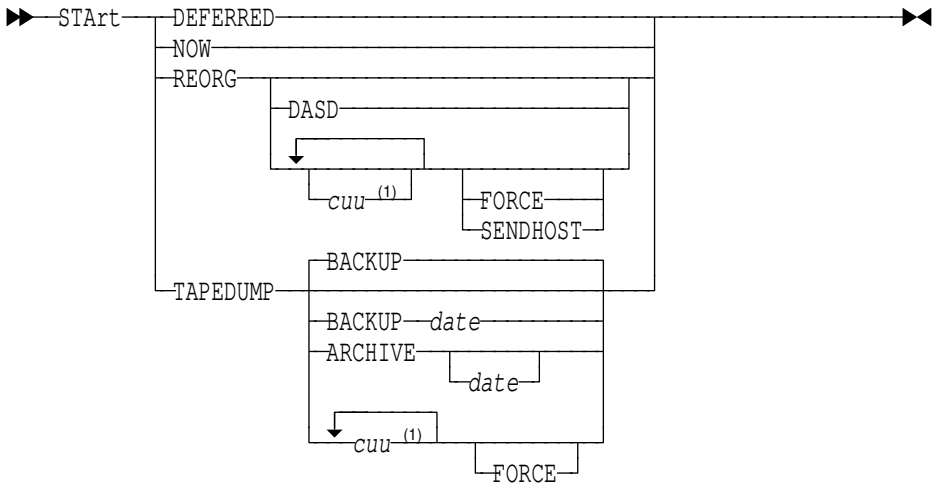
An alternative name for this command is END. This synonym is provided to avoid the possibility of closing down VM by mistake.

Authorized User Command



10.2.35 START

Authorized User Command



Note:
1 The cuu must be 3 digits (i.e. A1B, not 0A1B)

Operands

DEFERRED Is used to cause BARS/VM to process all deferred commands. These are normally processed by default at the start of a backup cycle, but the DEFERRED operand allows such commands to be executed on demand.

NOW This may be used to complete the backup process outside the normal times as set by the TIME command. START NOW can only be used to complete an unfinished backup. If backup has already completed that day, no further backup is performed but reorganizations may take place if any are due.

REORG Causes BARS/VM to reorganize backup and archive minidisks and tapes. This is the same process that occurs normally at the end of a backup cycle. If only the keyword REORG is specified, all minidisks or tapes which are full and due for reorganization are reorganized.

DASD If DASD is specified, all full minidisks which are due for reorganization are reorganized.

cuu If one or more cuus are specified then only the minidisks or tapes with the specified cuus are considered for reorganization. In this case reorganization takes place for those minidisks or tapes which are full and have been so for at least one day. Note that if tapes are used for backup then only tapes are reorganized and not the backup minidisks.

FORCE If FORCE is specified, completion of each reorganization is forced even if errors result in backup files being lost. Without the FORCE option, a reorganization will terminate after a certain number of errors have occurred. If backup files are lost, the original files are flagged so that they are backed up again at the next backup cycle.

SENDHOST The SENDHOST keyword is for those systems which use remote backup. It specifies that during the reorganization process, files are to be sent to the host without taking account of the value of the EHDAYS installation parameter. See 5.3, “Remote Backup and Retrieval” on page 38 for a discussion of remote backup.

TAPEDUMP If tape backup or archive minidisks are defined, START TAPEDUMP initiates a spill to tape. For backup, this is the same process that occurs when minidisk space is to be made available during normal backup and it allows an installation to empty backup minidisks to tape at a specific time if so desired. This is useful in the situation where tape operators are only available at certain times. For archive, this command is the only way that data is spilled to tape.

ARCHIVE If ARCHIVE is specified all archive minidisks are spilled. If a date is given, only files which were archived on or before this date are spilled. If no date is given, the minidisks are emptied to tape.

BACKUP If BACKUP is specified, all backup minidisks are spilled. This is the default. If a date date is given, only files which were backed up on or before this date are spilled. If no date is given, the minidisks are emptied to tape.

cuu If one or more cuu values are specified then only these minidisks are spilled. No date is allowed in this case.

FORCE If FORCE is specified, completion of a tape spill from a minidisk is forced even if some files to be spilled cannot be found on the specified minidisk. For a backup spill, if files are lost, the original files are flagged so that they are backed up again at the next backup cycle. Files which cannot be found during an archive spill are deleted from the archive catalogs.

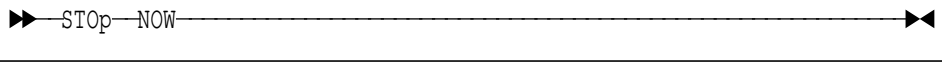
Usage Notes:

1. Archive minidisks are only spilled to tape if the installation option EARSPILL is specified as YES.

10.2.36 STOP

The STOP command cause the backup process to complete once each subuser has completed the unit of work it is currently performing. A work unit during backup is either a real DASD volume or an SFS storage group. It could also be a minidisk or tape reorganization. Since these units of work can take some time to complete, it is possible that the backup process will not terminate for a considerable time after a STOP command is received.

Authorized User Command



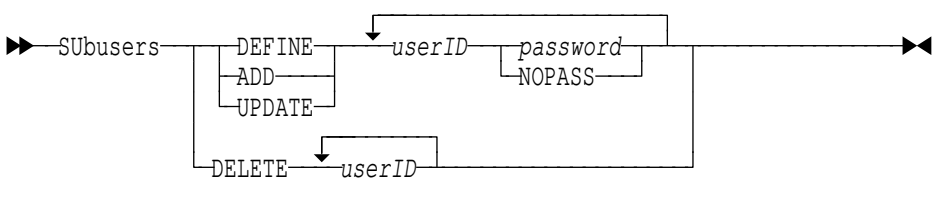
Usage Notes:

1. The keyword **NOW** must always be specified.

10.2.37 SUBUSERS

Use the SUBUSERS command to define, update or delete entries in the list of user IDs which comprise the BARS/VM subusers and, for those systems which require them, their passwords.

Authorized User Command



Operands

DEFINE Defines the specified user IDs and their passwords as BARS/VM subusers. This function is valid **only when the system is first initialized**.

You must specify a minimum of three subusers. The first will be used as the Backup Controller, the second as the Query Controller, and the rest as general subusers (see usage note 2).

ADD Adds the specified user IDs and their passwords as new BARS/VM subusers (see usage note 2).

UPDATE Causes the specified user IDs passwords to be updated in the BARS/VM catalogs. This should be used when a subuser logon password is changed **and** the VM system requires that passwords are specified on the CP (X)AUTOLOG command. If this is not so, then the subuser passwords are not required by BARS/VM, and the NOPASS operand should be used.

DELETE Removes the specified user IDs.

password Specifies the logon password for the subuser. This will be used by BARS/VM when the subuser needs to be autologged by the interface virtual machine. On some releases of VM, a password needs to be specified on the CP (X)AUTOLOG command.

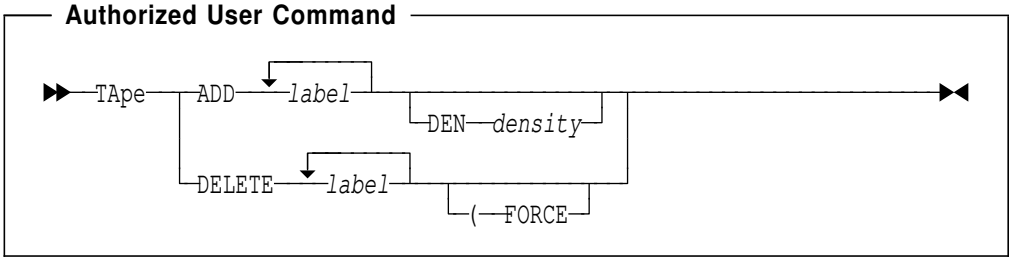
NOPASS Specifies that no password is to be used on the CP (X)AUTOLOG command for that subuser.

Usage Notes:

1. Any updates to the subuser information becomes effective on the **next initialization** of BARS/VM.
2. The list of subusers is maintained in alphabetical order. Remember that the first two in the list are used as the Backup and Query Controllers.
3. A `DEFINE` command always creates the subuser list from scratch. Any previously defined list will be totally replaced.
4. The minimum number of subusers is **three**. See chapter 20.2, “Planning BARS/VM Virtual Machines” on page 217 for a description of subusers and their functions.

10.2.38 TAPE

Use the TAPE command to add or delete tapes to or from the BARS/VM tape inventory to be used for backup and archive. Each tape to be added must be standard labelled with a unique volume serial number which is the same as the visual label as recognized by the Operator.



Operands

ADD Specifies that the stated tape volume is to be added to the tape inventory.

DELETE Specifies that the stated tape volume is to be removed from the tape inventory. Unless the FORCE option is specified, a tape is not deleted if it contains valid files.

label Specifies the external label of the tape volume. This is the visual label as recognized by the Operator and also the volume serial number in the standard label.

DEN States that a specific tape density is to be used when reading or writing from or to the tape volume.

density Is the density to be used for the tape. If it is not specified the installation default of ETPDEN is used. See “General Options” on page 226 for a description of the installation options.

Options

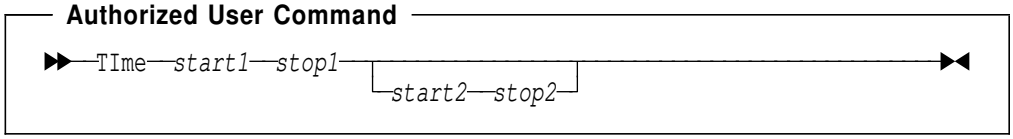
FORCE Forces deletion of tapes which may contain valid files. Care should be taken when using this option as it may leave catalog entries referring to the deleted tapes. Normally a REMOVE command should be used before tape deletion to remove catalog references.

Usage Notes:

1. A maximum of 37800 tapes may be defined.
2. Backup tapes are independent of tapes defined for DBR usage.

10.2.39 TIME

The TIME command defines the time "window" during which the automatic backup process (or cycle) will run.



Operands

start1 Defines the time at which the first window starts. This should be specified in the format "hhmm".

stop1 Defines the time at which the first window stops. This should be specified in the format "hhmm".

start1 Defines the time at which the second window starts. This should be specified in the format "hhmm".

stop1 Defines the time at which the second window stops. This should be specified in the format "hhmm".

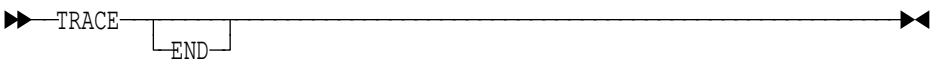
Usage Notes:

1. If two windows are specified, and backup completes during the first, then BARS/VM will **not start a fresh backup during the second**. If backup did not complete during the first window, then BARS/VM will try to complete it during the second.
2. A window may span midnight.
3. Once a window has been defined, then BARS/VM will trigger the backup process at the selected start times every day. It is only necessary to issue the TIME command to change the window (or stop it altogether).
4. No backup will take place if the first window is specified as 0000 0000.
5. To prevent the automatic backup process from running, issue a TIME 0000 0000 0000 0000 command.
6. When the stop time is reached, the backup process completes once each subuser has finished the unit of work it is currently performing. Since a unit of work can take some time to complete, it is possible that the backup process will not terminate for a some time after the stop time.
7. BARS/VM normally only performs one complete backup cycle per day. If a cycle completes and the TIME command is used to start backup again on the same day, no backup will be done. If a cycle fails to complete, the command START NOW should be used to complete the cycle.

10.2.40 TRACE

Use the TRACE command to activate or deactivate a temporary VMCF transaction trace of user ID EIUSERID (Interface userID).

Authorized User Command



Operands

END Stop the VMCF transaction trace. If this operand was not specified, then the trace is started.

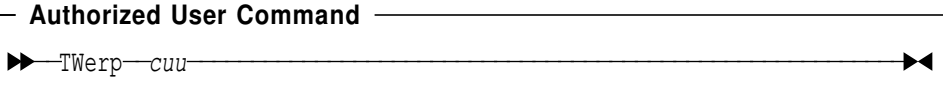
Usage Notes:

1. The TRACE command may only be issued from the BARS/VM manager user ID.

10.2.41 TWERP

The TWERP command marks an archive twin tape as no longer a twin. This allows the tape to be used in place of the archive tape of which it is the twin when the original archive tape is lost or damaged. A new twin tape will then be created when the tape is reorganized.

Authorized User Command



Operands

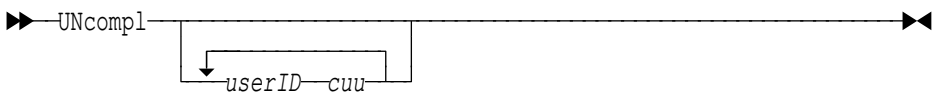
cuu Specifies the cuu of the archive twin tape.

10.2.42 UNCOMPL

The UNCOMPL command may be used to request specific backup of certain minidisks following a successful backup cycle.

Following the completion of a backup cycle, or when a START NOW command is issued after a completed cycle, this file is used to process again some specific minidisks for backup. The specified minidisks are recorded in a file called UNCOMPL FILES.

Authorized User Command



Operands

userID Specifies the user ID owning the minidisk which is to be processed again.

cuu Specifies the virtual address of the minidisk.

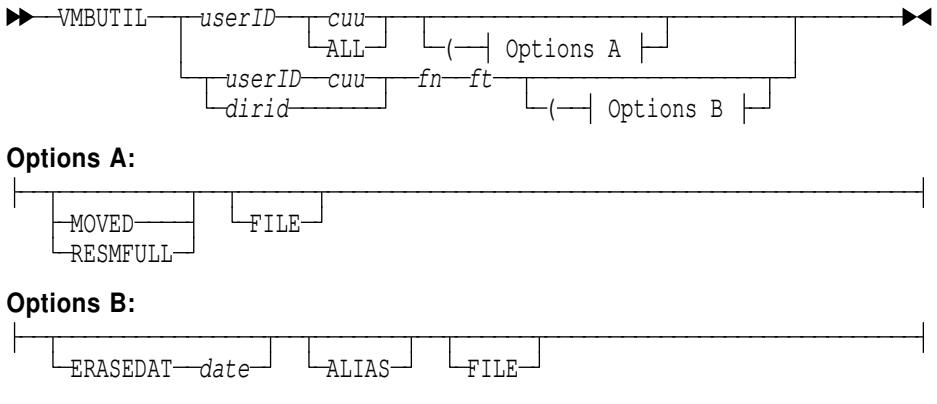
Usage Notes:

1. If the command is specified without any operands, then the contents of the file UNCOMPL FILES are displayed.
2. No checks are made for duplicate minidisks being specified.

10.2.43 VMBUTIL

The VMBUTIL command allows an authorized user ID to change information for the minidisk defined by the parameter *cuu*, for a SFS directory, or for all minidisks and SFS directories where backup data exists if ALL is specified.

Authorized User Command



Operands

userID Is the user ID for which information is to be changed.

cuu Specifies the minidisk for which information is to be changed.

dirid Identifies the SFS directory for which information is to be changed.

fn ft Specify particular CMS files. If these are specified then BARS/VM checks only for the particular file or files specified. File name and file type may be specified using the * and % notation as described in 8.1.3, “How to Read a Syntax Diagram” on page 52.

Options A

MOVED Specifies that the ADT information is to be changed for a specified minidisk, e.g.device type and block size. VMBUTIL links to the minidisk(s) and updates the ADT information kept in the catalog (UC) file. This is useful when moving a minidisk to another device type to fill the time gap between the move and the next backup.

RESMFULL Specifies that the kept value "maximum number of blocks used" is to be reset to 0. This value is displayed by the SHOWDISK command.

FILE Specifies that the result of the VMBUTIL command is to be sent as a file to the command issuer's card reader. This file is in NETDATA format.

Options B

ERASEDAT Searches for files which have been flagged as erased on or after the specified date and removes the erasure date. This allows a RESTORE command to restore these files. The date must be of the form ddmmmyy, where dd is the day number, mmm is the mnemonic month and yy the year number, e.g. 10JAN96.

Note that the erasure date shown in the BARS/VM QUERY command is the erasure date (when BARS/VM detects that a file no longer exists on a user's minidisk) plus the value of ERASDAYS. Hence the date specified with the ERASEDAT option is that shown in the QUERY response *minus* the value of ERASDAYS.

ALIAS Specifies that only aliases which match the file name and file type are to be processed. File data is not affected.

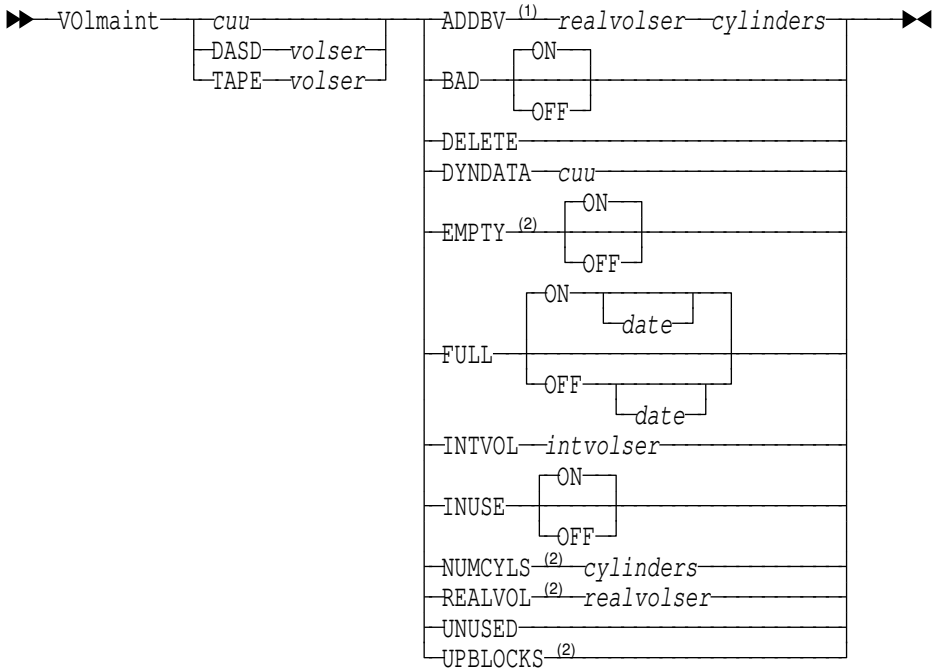
FILE Specifies that the result of the VMBUTIL command is to be sent as a file to the command issuer's card reader. This file is in NETDATA format.

10.2.44 VOLMAINT

The VOLMAINT command allows maintenance to be performed on the two control files which contain information about the status of backup and archive volumes. The BACKUP VOLUMES file contains information about minidisks and TAPE VOLUMES is the control file for tapes. The command allows an authorized user to change the status of minidisks and tapes.

CAUTION:
This command should be used with care and it is recommended that copies of the affected files are taken from the Interface 191 minidisk before the command is used.

Authorized User Command



Notes:

1 This operand is only valid with the *cuu* form of the command.

2 This operand is not valid with the **TAPE** form of the command.

Operands

cuu Defines a minidisk or tape whose status is to be changed or a new minidisk to be added.

DASD volser Defines an existing minidisk to be processed by its BARS/VM generated internal label.

TAPE volser Defines an existing tape to be processed by its BARS/VM generated internal label.

ADDBV Specifies that a new minidisk is to be added as an UNFORMATTED minidisk. The parameter *realvolser* gives the volume serial number of the volume on which the minidisk resides and this is followed by the size of the minidisk in cylinders. If ADDBV is used, the TAPE and DASD forms of the first operand are not allowed.

BAD Specifies that a tape is either to be marked as bad (BAD ON) or not bad (BAD OFF).

DELETE Specifies that information about the minidisk or tape is to be deleted from the relevant control file.

Note: This command will not remove references to the deleted volume from backup or archive catalogues. Before using this option with a backup volume read the description of the REMOVE command. For archive volumes read the description of the COLLECT command. See Chapter 12, “BARS/VM Administrator's Tasks and Controls” on page 151 for more information.

DYNDATA Specifies that information about the minidisk or tape is to be set in the relevant control file from another minidisk or tape in the same control file. Only dynamic data is set. Static information is unchanged. For minidisks the static data includes cuu, real label, number of cylinders, and number of formatted blocks. For tapes it includes cuu, external tape label, date added, density, and tracks. The cuu following the keyword DYNDATA gives the source minidisk or tape from which data is taken.

EMPTY Specifies that the minidisk is to be marked as empty (EMPTY ON) or not empty (EMPTY OFF).

FULL Specifies that the minidisk or tape is either to be marked full (FULL ON) or not full (FULL OFF). The default, if neither ON nor OFF is specified, is ON. Optionally a date may also be given. This sets the date of last reorganization to the value given. If specified, the date must be of the form *ddmmmyy*, where *dd* is the day number, *mmm* is the mnemonic month and *yy* the year number, e.g. 10JAN96.

INTVOL Changes the internal volume label for an existing minidisk or tape to the specified value.

INUSE Specifies that the minidisk or tape is either to be marked as in use (INUSE ON) or not in use (INUSE OFF).

NUMCYLS Changes the number of cylinders for an existing backup or archive minidisk to the specified number. This option is not valid with tape volumes.

REALVOL Changes the real volume label for an existing backup or archive minidisk to the specified value. This option is not valid with tape volumes.

UNUSED Specifies that the minidisk or tape is to be assigned unused status. For a minidisk, DISPLAY VOLUME will show it as UNFORMATTED. For a tape, DISPLAY TAPE will show the tape as UNUSED.

UPBLOCKS Updates the number of blocks on a backup or archive minidisk and the number of free blocks from the minidisk itself. The minidisk is accessed and the block counts determined. This option is not valid with tape volumes.

Usage Notes:

1. VOLMAINT can be useful when moving backup or archive data to new minidisks or tapes. After the data has been copied, VOLMAINT can be used to set the correct control information. See 12.9, “Transfer of Backup and Archive Data to New Volumes” on page 165 for a discussion of how to move backup and archive data to new devices.
2. VOLMAINT is the only way an archive minidisk can be changed to UNFORMATTED in the BACKUP VOLUMES file. A REMOVE command followed by REUSE returns an archive minidisk to its original state.
3. Remember that minidisk volumes must have an address in the range 001-5FF. Addresses from 600 onwards are for tape volumes.

Chapter 11. Manager Commands Description

11.1 Issuing Manager Commands

The authority to issue the following commands is controlled by entries in the MANAGER AUTHLIST and VMBARS RTABLE files held on the manager 194 minidisk. See 20.3.5, “Manager Installation Options” on page 247 for details of these files.

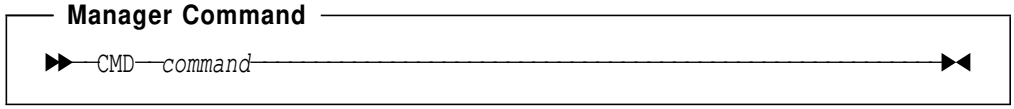
Commands are invoked by sending messages to the manager user ID while it is running disconnected — e.g. `MSG BARMANGR SENDREP SUMMARY` would cause the SENDREP command to execute with the argument SUMMARY.

11.2 Command Syntax Description

11.2.1 CMD

The CMD command lets the command issuer execute any CP or CMS command in the manager user ID.

Manager Command



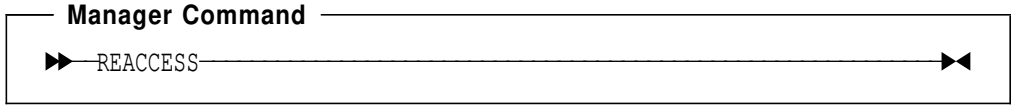
Usage Notes:

1. This can be used to execute BARS/VM commands requiring manager authority. Issuing `MSG BARMANGR CMD VMBARS AUTHORIZ ADD WILMA` would define local user ID WILMA as an authorized BARS/VM user.

11.2.2 REACCESS

The REACCESS command causes the manager to re-access its 194 minidisk, re-read the authorisation tables and re-load the PROP table.

Manager Command



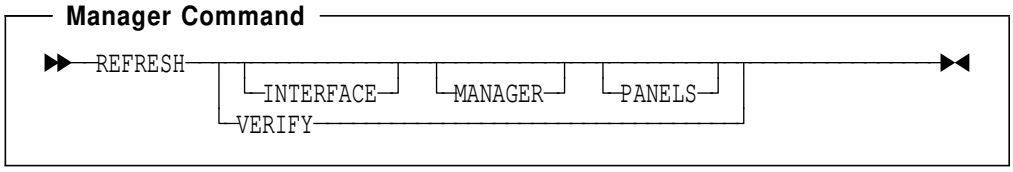
11.2.3 REFRESH

The REFRESH command will refresh the specified code repositories from the distribution disk (either the Manager 291 or a SFS directory).

Before starting, the contents of the distributed code disk is validated against the VMBARS CHKLIST file. If any files are found to be missing or back-level, then the refresh operation is aborted without any code updates taking place.

On completion of the refresh, a report file is sent listing those files which were updated. To activate the changes issue the appropriate restart command

(KILLSUB Interface for the BARS/VM subsystem, or REIPL for the manager machine).



Operands

INTERFACE Specifies that the Interface 194 minidisk be refreshed.

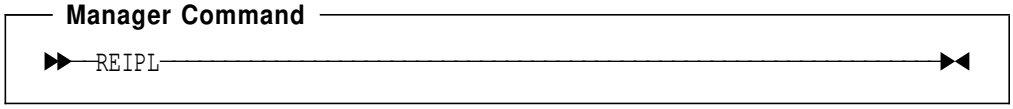
MANAGER Specifies that the Manager 194 minidisk be refreshed.

PANELS Specifies that the Interface 19F minidisk be refreshed.

VERIFY Specifies that only the validation step be performed.

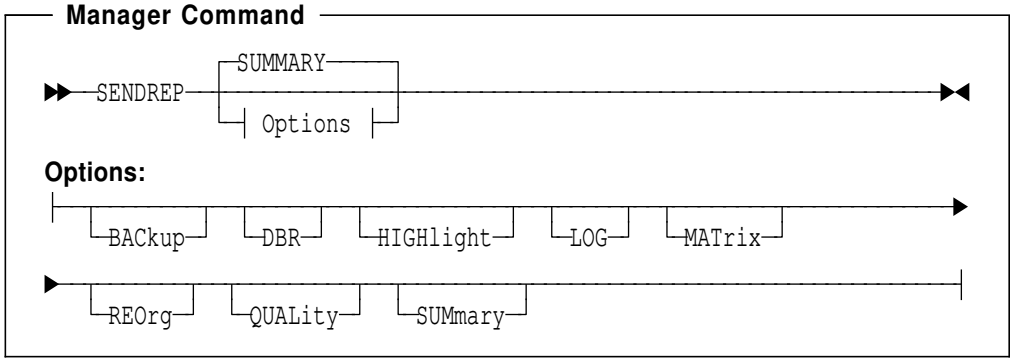
11.2.4 REIPL

The REIPL command causes the manager to re-IPL.



11.2.5 SENDREP

The SENDREP command causes the manager to send the specified report files to the command issuer.



Operands

BACKUP Specifies that the backup report file be sent.

DBR Specifies that the DBR report file be sent.

HIGHLIGHT Specifies that the highlighted messages file be sent.

LOG Specifies that the entire log file be sent.

MATRIX Specifies that the matrix report file be sent.

REORG Specifies that the reorg report file be sent.

QUALITY Specifies that the quality report file be sent. This file is only produced if the HDFXCA, HDFXCB, HDFXCE and HDFXCM user exits have been coded, and collects messages with a message prefix starting HDFXCA, HDFXCB, HDFXCE or HDFXCM.

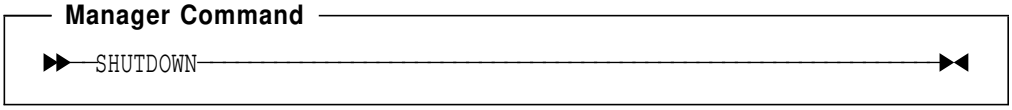
SUMMARY Specifies that the backup cycle summary report file be sent. This file contains an analysis of the last backup cycle, and can be used to determine from a remote system whether the backup cycle was successful. It also indicates when it might be necessary to read some of the other log files.

Usage Notes:

1. The creation of each of these reports (other than the LOG or SUMMARY reports) is controlled by the MANAGER OPTIONS file, as described in 20.3.5, “Manager Installation Options” on page 247.

11.2.6 SHUTDOWN

The SHUTDOWN command causes the manager to logoff.



Chapter 12. BARS/VM Administrator's Tasks and Controls

This section explains the necessity for some of the special processes which may be used by the BARS/VM administrator. These include backup minidisk, archive minidisk and tape reorganization and removal; archived data collection and the erasure of redundant archived data.

12.1 Reorganization

The reorganization process is used to make space on full backup and archive minidisks and tapes, which contain files no longer required, available for use again. In the discussion below the term “volume” refers to either a tape volume or a backup or archive minidisk. A volume is only eligible for reorganization once it has been marked as full.

To improve performance during the backup and archive processes, the backup and archive files are written but redundant files are never erased until reorganization time. There are two types of reorganization: “normal” (also known as “slow”) and “fast”. At a normal reorganization, all the files that could be retrieved are copied to a new volume called a reorganization volume. This frees all the space no longer needed on the volume while keeping the files written in sequential blocks. Then, the volumes are logically switched in BARS/VM. During this process, any data on this backup or archive volume may be retrieved from BARS/VM.

The normal reorganization method of erasing redundant files is best for MSS performance for those systems which have backup and archive on that medium. The problem with MSS performance is because of the normal size of backup/archive minidisks and the manner in which CMS writes a minidisk.

Normal use of a minidisk results in files being written, erased, and rewritten. The CMS system allocates 1K, 2K or 4K blocks as required to satisfy write requests. This allocation is done sequentially throughout the blocks of the minidisk. When a file is erased, the corresponding blocks can be used for another file. But if the next file is larger than the one erased, then the blocks are not contiguous. Figure 18 on page 152 shows this effect.

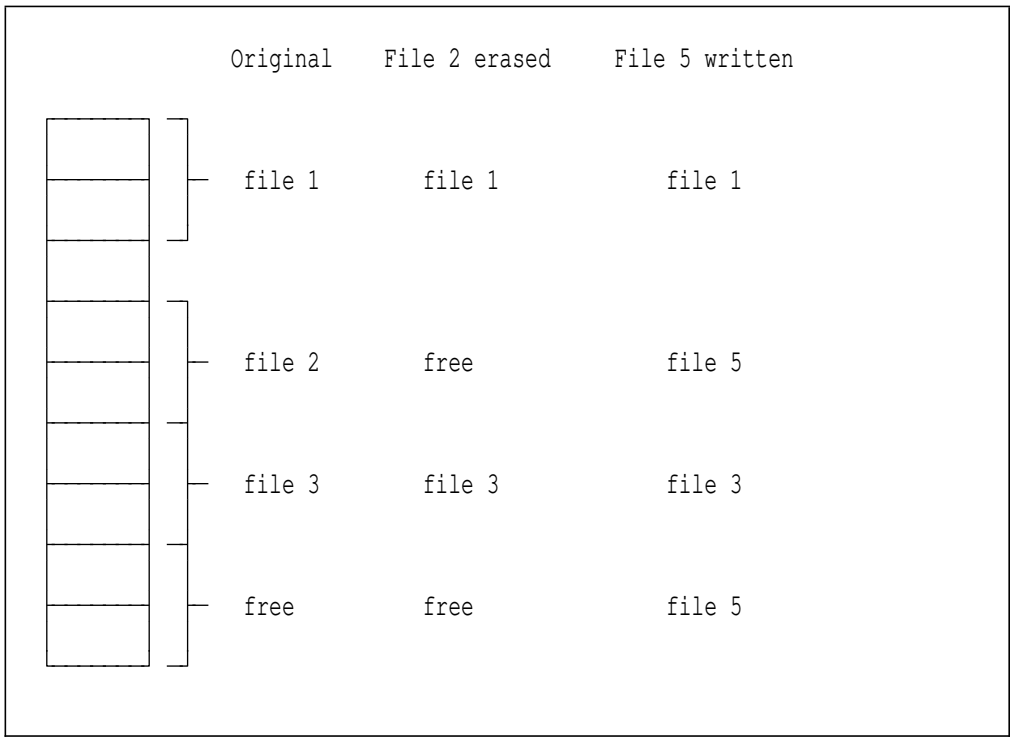


Figure 18. CMS Minidisk Organization

The second part of the problem is that the amount of time required for the MSS to stage (make available to VM) cylinders is dependent on the number of cylinders it must stage. Thus, if file 5 and 3 were the same number of blocks, it would take the MSS longer to stage file 5 than 3 since file 5 is contained on more cylinders.

Because of the nature of the backup process, it is expected that the writing and erasing of files would occur frequently. Thus the number of cylinders used for any one file would increase over time.

A “fast” reorganization of a backup or archive minidisk simply erases the redundant files from the minidisk and does not use a reorganization volume. This gives a much faster reorganization but is only suitable for real DASD devices. Fast reorganizations for backup and archive minidisks are done if the installation option EFASTR is set to YES.

Since tape is a sequential medium, normal reorganization is the only method for that device type which will ensure all valid data is kept and redundant data erased. This method is therefore always used for the reorganization of archive tapes, regardless of the EFASTRT setting.

In the case of backup, however, it is recognized that this process is very slow, and that many installations would rather create new backup copies from the original files than perform a tape to tape reorganization. This is the way BARS/VM works if fast reorganization for tapes is specified. Fast reorganizations for tapes is selected if the installation option EFASTRT is set to YES. In effect, there is no reorganization at all. When a tape is due for reorganization it is treated as empty and all files that were backed up to that tape are scheduled for backup at the next cycle.

Warning: In this situation, all the backups which are higher than level zero are lost forever, since the original data is no longer available to be backed up again.

Most installations can accept that levels greater than zero are only kept for a limited time, and for these systems fast reorganization saves a large amount of time.

CAUTION:
Until the original data is backed up again, the level 0 backup data is not available for retrieval.

This exposure may not be acceptable, and will depend on local requirements.

Reorganization takes place at one of three possible times: first when a subuser is doing overnight backup or deferred archive and there is no backup or archive volume available; second on completion of backup and deferred archive; and third is after a START REORG command is issued. In the first two cases no volume is reorganized unless it has been marked full for at least a specified number of days: EREORG for backup volumes; EAREORG for archive minidisks. This ensures reorganizations do not take place too frequently. The installation should set these parameters carefully so that reorganizations take place on completion of backup to keep a reasonable number of volumes available - not too few so that backup or archive is interrupted to perform reorganization and not too many so that maximum benefit is obtained from the reorganization process. A START REORG command which specifies no volumes causes a general reorganization as described above. If the command specifies particular volumes then these are reorganized provided they have been full for at least one day.

If it is desired to limit the days on which automatic reorganizations take place, the ERDAY installation option may be used. This allows such reorganizations to be restricted to specified days. It does not, however, apply to the START REORG command.

For BARS/VM systems that perform remote backup (see 5.3.1, “Definitions” on page 38), files are sent to the remote host during backup minidisk reorganization. A file is sent provided it has been backed up for at least the number of days specified in the EHDAYS installation option (see “General Options” on page 226). Each file is sent to the remote host together with a BPUT command which describes the processing required. A file sent in this way is still copied to the reorganization volume and so is still available for retrieval by the local BARS/VM system until the host sends a BLAH command acknowledging successful backup.

A backup volume is normally marked full if it becomes full during the normal process of copying files. However, to ensure regular reorganization and thus sending of files to the host, a backup minidisk at a drone is marked full if at least EHDAYS have elapsed since it was last reorganized.

If no tapes are defined for backup, then backup minidisks are reorganized, otherwise the tapes are reorganized. Similarly, if archive spill is not allowed, the archive minidisks are reorganized otherwise it is the tapes. At the start of backup or reorganization a certain number of volumes are reserved for the reorganization process unless fast reorganization is to be used. One volume is reserved for each subuser except when twin archive tapes are used in which case two tapes are reserved for each subuser.

Refer to chapter 20.1.2, “Planning Considerations for Backup and Archive Minidisks” on page 213 for detailed information about DASD capacity planning regarding the reorganization process.

12.2 Maintain BACKUP Volumes: REMOVE and REUSE

If errors are encountered with a backup minidisk or tape, it may be eliminated or removed. The REMOVE command is used to perform this function. When an authorized user ID issues the REMOVE command with the DELETE option for a backup minidisk or tape, BARS/VM simply erases the entries in the control (USERID CUU) files which refer to the data backed up on the volume being removed. The source data will then be backed up to other backup volumes during the next backup cycle. If a backup REMOVE is requested without the DELETE option, BARS/VM attempts to move all the data on the backup minidisk or tape to other volumes. If there are errors moving the data, messages are written to the manager user ID identifying the CMS file or non-CMS minidisk in error and the process continues. The corresponding backup entry is also eliminated so a RETRIEVE is no longer able to access this data. The source data is scheduled for backup again at the next backup cycle.

If there is no space on other minidisks or tapes, a message is send to the manager and the REMOVE ends (at this point the minidisk is marked 'reserved for remove'). In the case of backup minidisks, more space should be made available at the next backup cycle. BARS/VM will automatically reorganize other minidisks to make space available at that time and the REMOVE command should then be reissued. For backup tapes, either new tapes can be added immediately, or the REMOVE reissued when reorganization makes space available.

When a backup REMOVE is completed, the installation can remove a backup minidisk by eliminating the corresponding entry in the BARS/VM NOLOG (ENUSERID) VM directory entry. For a backup tape, the TAPE command may be used to delete the tape definition. Alternatively, the installation can replace the volume and then, by using the REUSE command, have BARS/VM use it again. While a volume is marked as 'reserved for remove' only a REMOVE or REUSE command will use that volume.

Figure 19 on page 155 helps to explain the logic of REMOVE and REUSE for backup minidisks and tapes.

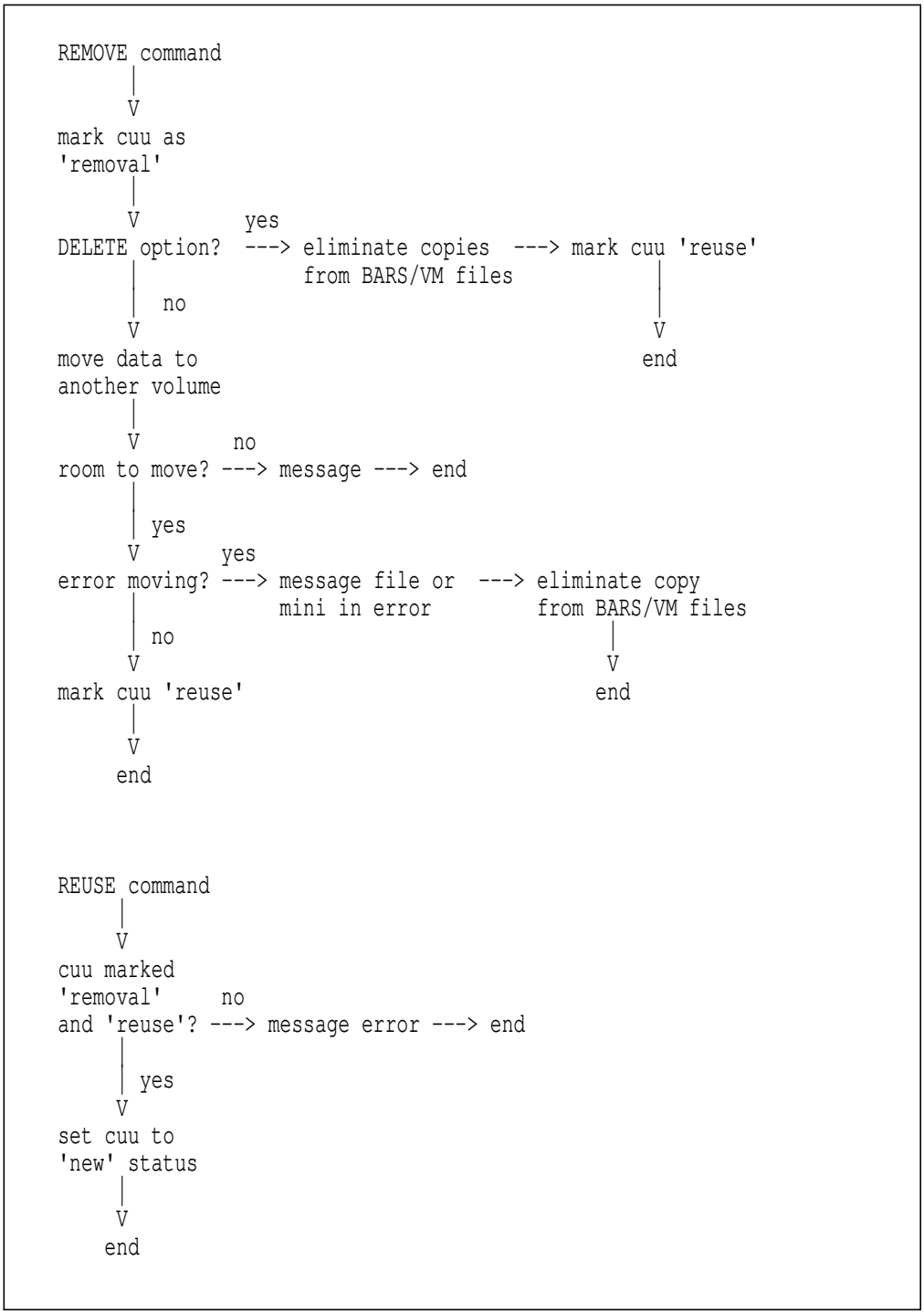


Figure 19. REMOVE/REUSE Process Flow

REMOVE for archive volumes is treated differently because backup normally exists either on DASD or through twin tapes and so the data can be restored to its original state. So, for an archive minidisk no data is moved and no catalog files are updated. The archive minidisk is simply marked as archive and reserved for reuse. When the minidisk is restored from backup a REUSE command will set it to the state is was in before the REMOVE. For an archive tape, no data is moved and no catalog files are updated. The tape is marked as unused, reserved for reuse, and, if it was a twin, it is marked no longer a twin.

The tape should normally be deleted after such a remove. The TWERP command may be used to activate the twin tape.

12.3 Maintain ARCHIVE Volumes

12.3.1 Selection of an ARCHIVE Volume

When an APUT command is to be processed BARS/VM first checks for an available archive volume. If none is found, and archive is to DASD, an unformatted (free) minidisk is selected. This is formatted and marked for archive usage. Once a minidisk is flagged as archive it can **not** be redefined to be used for backup. If archive is directly to tape and no existing useable archive tape is found, an unlabelled tape is selected and defined as an archive tape.

12.3.2 COLLECT

The COLLECT command, when used with the EXTERNAL command, is designed to enable the BARS/VM manager to move archived files which have a long retention period to archive minidisks on one (or more) volumes which may then be removed from the system for long term storage.

The command may be used to COLLECT archived files for all archive pools, for one archive pool or for a subset of archive pools by using the % and * notation. The retention period need not be specified so all archived files belonging to one pool can be collected to a specific minidisk. If a COLLECT minidisk is not specified an unused one is selected from Backup Volumes and formatted for use. If a minidisk is specified it will be used. If, during the collection process there is insufficient room on a COLLECT minidisk, an additional unused minidisk will be selected and formatted for use. The reason for specifying a specific COLLECT minidisk may be to fill up a minidisk which is to be made external using the EXTERNAL command and to which some files have already been collected.

12.3.3 EXTERNAL

The EXTERNAL command is for use by the BARS/VM manager when making an archive minidisk external to BARS/VM by removing the volume on which it resides from the system for long term storage and when reintroducing the volume into the system. The Backup Volumes file is read and the entry for the specified minidisk located and marked external or internal according to the parameter specified.

12.3.4 AAUDIT

The AAUDIT command is provided as an aid to the identification and elimination of redundant archived data. When used with the DEACT operand any file processed whose expiry date has been attained is marked as “deactivated”. The deactivated state provides an intermediate stage between the normal archived state and being deleted. When AAUDIT is specified with DELDEACT, any file processed which is marked as deactivated is erased from archive.

The intermediate deactivated state provides an installation with the opportunity to inform users of the fact that their data has expired and may be erased.

See also Chapter 4.6, “Installation Controls of Archive Data” on page 25 for a more detailed description of how an installation controls the amount of archive space.

12.3.5 REMOVE and REUSE

The removal process is not necessary for archive minidisks used for archived files since these are backed up by BARS/VM. If data on such an archive minidisk is lost it may be retrieved from the backup of the archive minidisk. Archive minidisks used for minidisk archive operations through the MDISK command are not backed up since the normal backup for the minidisks is still available. A REMOVE command issued for an archive minidisk simply marks it as removed but performs no other action. A REUSE for such a minidisk marks the minidisk as a formatted archive minidisk again, i.e. it sets it to the same status it had before the REMOVE was done. So the recovery procedure when an archived minidisk is lost is as follows:

- Issue a REMOVE command for the minidisk. This will prevent it from being used until it has been restored.
- Restore the minidisk using the RESTORE command.
- Issue a REUSE command for the minidisk.

A REMOVE of an archive tape simply marks the tape as reserved for reuse and no data movement is done. A damaged archive tape can be removed and replaced by a twin tape. See 4.12, “Archive Twin Tapes” on page 30 for a discussion of twin tapes.

The recovery procedure when an archived tape is lost and a twin exists is as follows:

- Issue a REMOVE command for the tape. This will prevent it from being used again.
- Delete the tape using TAPE DELETE.
- Issue a TWERP command for the twin to create a new archive tape.
- Either reorganize the new archive tape immediately or ensure it is reorganized as soon as possible to create a new twin.

If an archive tape is lost which has no twin, a START REORG with the FORCE option will erase all references to the tape in the catalog files.

12.4 BARS/VM HALT Processing

The HALT command is used for testing, maintenance, or debugging purposes by the BARS/VM manager.

After the HALT ALL command is processed, no other work is accepted by BARS/VM except from the BARS/VM manager. Then, the BARS/VM manager may use the DISPLAY command to determine when all currently running work has completed, at which point he may run any work required. The HALT command guarantees to the BARS/VM manager that no other work will occur until he issues the HALT END command. **This remains true even if VM or BARS/VM restarts**.

An alternative to issuing a HALT ALL command is the HALT USER command. This prevents general users from using BARS/VM, but allows BARS/VM authorized users to continue working.

12.5 BARS/VM Report Messages

BARS/VM produces report messages of its activity during the backup cycle. A long or short report can be specified. The messages are sent to the manager user ID.

Example:

```
-BARS/VM- Report from    mm/dd/yy hh:mm:ss
Backup process start:    mm/dd/yy hh:mm:ss
Number of minidisks   to be processed:              6283
Backup cycle start:      mm/dd/yy hh:mm:ss
Number of directories to be processed:               184
Number of filespaces  to be processed:                30
$CKPT    001  minidisk 000009 Cyl not backed up (NOBACKUP)
BELLTG   191  CMS      000010 Cyl   files copied:       2
 ........
IX7MDB   390  minidisk 000009 Cyl not backed up (NON-CMS)
 ........
JACKIE   _AAAAAB FILEPOOL  TEST     files copied:      21
 ........
PETERS   191  minidisk 000002 Cyl not backed up (not due)
 ........
RACFVM   200  NON-CMS  000010 Cyl   parts copied:       1
 ........
SQLDBA   205       reformatted to NON-CMS
 ........
WILLIAMS 191  CMS      000020 Cyl   files copied:       1
*** ALL  OBJECTS HAVE BEEN PROCESSED ***
Number of minidisks excluded (PRIMARY command)         9
Number of NON-CMS minidisks processed:                10
Number of NON-CMS minidisk  megabytes processed:      86
Number of CMS minidisks processed:                  5428
Number of CMS minidisk  megabytes processed:       85380
Number of CMS files processed:                     10151
Number of minidisks processed as CMS and NON-CMS:      0
Number of minidisks not requiring backup:            845
Number of minidisks   not processed:                   0
Number of SFS directories processed:                 184
Number of SFS files processed:                      5229
Number of SFS directories not requiring backup:        0
Number of directories not processed:                   0
Backup cycle end:        mm/dd/yy hh:mm:ss
```

Figure 20 (Part 1 of 2). Sample Backup Report Messages

```
Number of tape backup files for level = 0:             0
Number of tape backup files for level > 0:             0
Number of tape backup parts for level = 0:             0
Number of tape backup parts for level > 0:             0
Number of tape backup blocks for level = 0:            0
Number of tape backup blocks for level > 0:            0
Number of dasd backup files for level = 0:       1892102
Number of dasd backup files for level > 0:        466469
Number of dasd backup parts for level = 0:            37
Number of dasd backup parts for level > 0:            35
Number of dasd backup blocks for level = 0:     24916877
Number of dasd backup blocks for level > 0:      4921984
Number of tape archive files for level = 0:            0
Number of tape archive blocks for level = 0:           0
Number of tape archive files for level > 0:            0
Number of tape archive blocks for level > 0:           0
Number of dasd archive files for level = 0:        43752
Number of dasd archive blocks for level = 0:     1646320
Number of dasd archive files for level > 0:         3628
Number of dasd archive blocks for level > 0:      134949
Number of dasd MDISK parts for level = 0:           1248
Number of dasd MDISK blocks for level = 0:       1261482
Backup process end:      mm/dd/yy hh:mm:ss
```

Figure 20 (Part 2 of 2). Sample Backup Report Messages

12.6 DBR Report Messages

BARS/VM also produces report messages of its activity during the DBR backup cycle. A long or short report can be specified. The messages are sent to the manager user ID.

Example:

```
-BARS/VM- Disaster Backup Report from   mm/dd/yy hh:mm:ss
DBR PROCESS START ( DELTA DUMP )    :   mm/dd/yy hh:mm:ss
ABCDEF   194       CMS 000010 cyl files copied:  13
CDEFGT   222       CMS 000009 cyl files copied:   2
USERX1   191       CMS 000010 cyl files copied:   8
USERX1   291       CMS 000040 cyl files copied:   2
USERX3   191       CMS 000020 cyl files copied:  38
MVSXA    600   NON-CMS 000400 cyl parts copied:   1
WIDUSER  191       CMS 000010 cyl files copied: 880
***-----------------------------------***
*** ALL MINIDISKS HAVE BEEN PROCESSED ***
***-----------------------------------***
Number of NON-CMS minidisks processed:                1
Number of NON-CMS minidisk  megabytes processed:     14
Number of CMS minidisks processed:                  151
Number of CMS minidisk  megabytes processed:       5806
Number of CMS files processed:                    63349
Number of CMS files missed (or in error):             0
DBR Process End   ( Delta Dump )    :   mm/dd/yy hh:mm:ss
```

Figure 21. Sample DBR Report Messages

12.7 Synchronization between BARS/VM and VM User Directory

- All backup information for a users minidisk is saved in a CMS file on the INTERFACE minidisk 191 with the file name of the user ID and the file type of the cuu (USERID CUU files).
- Thus, user minidisks may be moved, increased or decreased in size independently from the backup data.
- The MDISK statements for the user minidisks should be defined with link mode 'MR'. If the minidisks are specified with WR mode and BARS/VM has the minidisks in read mode when the user logs on (or is autologged), he will receive his disk in read only mode.

DIRECTORY CHANGES	Command required	Function Performed
add new user	none	none
delete user	none	none
change user ID name	RENAME	rename USERID CUU file and/or USERID ARC file
add new minidisk	DIRECT (See note below)	CMS mini - Automatic backup Non-CMS mini - none
delete minidisk	DIRECT (see note below)	eventual data erasure see EMINIDAY (in "General Options" on page 226)
change minidisk name	RENAME	rename USERID CUU file
exclude real volume from backup	PRIMARY	eliminates a real volume from the backup process

Figure 22. Synchronization between BARS/VM and VM User Directory

Notes:

1. Synchronization between BARS/VM and VM user directory merely effects minimization of DASD space usage on backup minidisks. It prevents the duplicate saving of data for minidisks whose addresses and/or owner user ID have been changed. It does not affect correct BARS/VM operation. Therefore, if you have enough DASD space for backup there is no necessity to synchronize BARS/VM and VM user directory.
2. However, if you like to ensure the synchronization, the directory maintenance person should be educated and authorized to issue the commands listed above correctly!
3. The DIRECT command need not be issued when adding or deleting a minidisk if the installation parameter EDIRPROC is set to YES.

12.8 Transfer of Archive Data to Another System

The following process describes how archive data held on one BARS/VM system may be moved to another BARS/VM system. This is not a difficult process, but it can take several hours to complete.

CAUTION:

1. **Read through the relevant section before starting, to familiarize yourself with the process.**
2. **Ensure that the process will not be interrupted by any other major BARS/VM process such as automatic backup, tape spills or reorganization, or users, as these may compromise the integrity of the archive data. Once started, you must either complete the process or back out any changes before letting these other processes run.**

12.8.1 If the Data is on DASD

Make a copy of the checklist on page 165 and use it to keep track of your progress. If multiple minidisks are being moved, then each step from step 3 to step 8 on page 163 should be repeated for each minidisk.

1. Issue `COLLECT` command(s)

   If a subset of the data is to be moved it should be collected to a set of archive minidisks which may then be copied to the target system. If all the archive data is to be moved, it is still often a good idea to collect all the data to be moved onto the minimum number of archive minidisks. This may reduce the total amount of work.

2. Make a backup copy of all files on the Interface 197 minidisk in case of mistakes. Also take a copy of the BACKUP VOLUMES file from the Interface 191 minidisk.

3. Issue `ATRANS cuu (RESERVE` command

   To reserve an unused minidisk as an ATRANSFER minidisk for each archive minidisk to be moved from the source system. The internal label used is shown and should be kept for later processing. The minidisk is marked as formatted, archive and external.

   This command must be issued on the receiving system.

   Example:

   ```
   vmbars atrans 009 (reserve
   Ready;
   HDFTRA0004I ATRANS command started
   HDFTRA0473I Cuu 009 marked as EXTERNAL ARCHIVE volume with internal label AAAADM
   HDFTRA0001I ATRANS command complete
   ```

4. Issue `ATRANS cuu volser` command

   To prepare all affected "userID ARC" files for the transfer process and change the internal label to 'volser'. This also changes the label of the specified archive minidisk, but does not update the BACKUP VOLUMES file on Interface 191.

   This command must be issued on the sending system. The cuu is that of the source minidisk and the volser the one returned by the previous ATRANS (RESERVE.

   Example:

   ```
   vmbars atrans 00a aaaadm
   Ready;
   HDFTRA0004I ATRANS command started
   HDFTRA0001I ATRANS command complete
   ```

5. Issue `EXTERNAL EXT` command

   To prevent the source archive minidisk from being used again.

   Example:

   ```
   vmbars ext 00a ext
   Ready;
   HDFCEX0001I EXTERNAL command complete
   ```

6. Copy the data from the source minidisk to the reserved minidisk.

   Then copy all "userID ARC" files affected to the receiving machine's Interface 197 disk. If naming conflicts arise, rename the new ARC files on the target

system, and when the transfer process is complete use AMERGE to consolidate the user's archive into one piece (or advise them of the new name, and update OWNED NAMES to allow them access).

7. Issue `ATRANS cuu volser (RESEQ` command

   Now merge in all transferred archive data into the existing archive system with correct file names. Depending on the amount of archive data involved, this command may take some time to run.

   Example:

   ```
   vmbars atrans 009 aaaadm (reseq
   Ready;
   HDFTRA0004I ATRANS command started
   HDFTRA0001I ATRANS command complete
   ```

8. Issue `EXTERNAL INT` command

   To make the archive minidisk on the target system available for use.

   Example:

   ```
   vmbars ext 009 int
   Ready;
   HDFCEX0001I EXTERNAL command complete
   ```

9. If authorizations have been given on the source system through the OWNED command, the file OWNED NAMES must be merged with the file of the same name on the target system so that authorizations are not lost.

10. If required, BARS/VM on the source system can be setup to automatically route all archive requests to the target system. This is done by defining the target node identification and interface user identification in the OPTIONS file on the Interface 194 minidisk, and restarting the BARS/VM system. See the `ERARCID` and `ERARCIU` options in 20.3.2, “Nodename OPTIONS - Define the Installation Options” on page 226.

To back off the changes on the source system if anything goes wrong, restore the files to the 197 minidisk that were backed up at the start of the process (see above), restore the saved BACKUP VOLUMES file to the 191 minidisk and re-label any changed archive minidisks.

Finally, remember that archive minidisks are backed up just like any other CMS formatted minidisk, and hence lost files can always be retrieved from BARS/VM backup of the source system.

12.8.2 If the Data is on Tape

The ATRANS command will work for archive held on DASD only. It is possible to merge tape archive data from one system to another using a relatively simple manual method. However, there is a possibility that some files transferred may have the same names as files on the target system. This is the situation which the ATRANS command with the RESEQ option avoids. If duplicate names do exist it is possible that subsequent tape reorganizations may result in files with the same name ending up on the same tape, in which case an AGET may return the wrong file.

If it is desired to continue despite this, the following method can be used. Take backup copies of all the affected files before attempting this so that you can back out at any time.

1. Ensure all the data on the source system is on tape.

   If there is archive on DASD, empty the minidisks to tape by using `START TAPEDUMP`.

2. Make a backup copy of the TAPE VOLUMES and CONTROL DATA files on both systems in case of mistakes.

3. Resolve conflicts between the two systems.

   If there is archive on both systems belonging to the same userID, then one of the ARC files must be renamed before the merge can be done. When the merge is complete they can be combined using the AMERGE command.

   The most difficult conflict to resolve is that of the internal volume serial numbers (labels) of the tapes. Issue a `DISPLAY TAPE ALL` on both systems. This shows an internal volume serial for each tape. The archive tapes from the source system are to be merged with those of the target system so there must be no conflict of internal serial numbers. If there are, the labels must be changed on one of the systems to remove the conflict. The system on which this should be done is the one with the higher value for the 'next internal volume serial' as displayed by the command `DISPLAY NEXTIVOL`. Then to change the labels you should do the following for each tape whose label is to be changed. Ensure all the catalog files from the Interface 197 minidisk are backed up before doing this.

   a. Get the 'next internal volume serial' value using the command `DISPLAY NEXTIVOL`. Call this vol2.

   b. Get the cuu of the tape and its internal volume serial number (label) by using the `DISPLAY TAPE` command. Call these cuu and vol1.

   c. Issue `VOLMAINT cuu INTVOL vol2`. This will change the internal volser for the tape to the new value.

   d. If the tape has a twin, issue the same VOLMAINT command, but specifying the cuu of the twin this time.

   e. Issue `CATMAINT ARCHIVE INTVOL vol1 vol2`. This will change all occurrences of vol1 in the archive catalogs to vol2.

   f. Update the 'next internal volume serial' value to the next higher value by using the command `CATMAINT INCIVOL`.

   Repeat these steps, if necessary, for further tapes.

4. Merge the TAPE VOLUMES records.

   Take the relevant records from the TAPE VOLUMES file on the source system and add them to the end of the file on the target system. Change the cuu values (the three characters in the second to fourth bytes) of the new records so that they follow the sequence of the records which already exist. In this sequence, the letters follow each other alphabetically and the digits in numerical order. '0' follows 'Z'. No tape cuu is allowed to end with 'W' and the numbers '0' to '5' are not allowed as the first character.

5. Merge the catalog files.

   Copy all the ARC files from the 197 minidisk of the source system to the target system.

6. Check the CONTROL DATA file

CONTROL DATA contains the next value that will be used as an internal volume serial. Ensure this is higher than any existing one in the new TAPE VOLUMES file.

12.8.3 Checklist for DASD Archive Transfer

The following can be used as a check list for the archive transfer process:

__ 1. On source machine collect the data to minimize the number of archive minidisks involved

__ 2. On source machine take backup copies of all ARC files and the BACKUP VOLUMES file

__ 3. On target machine issue `ATRANS cuu (RESERVE` commands and note new labels

__ 4. On source machine issue `ATRANS cuu volser` commands

__ 5. On source machine issue `EXTERNAL cuu EXT` commands

__ 6. Copy the archive minidisks from the source to the target machine. Ensure the labels of the disks on the new machine are correct

Copy all ARC files from source 197 disk to target 197 disk. Rename duplicate archive IDs according to a convention (e.g. prefix all duplicates from source machine with a 5)

__ 7. On target machine issue `ATRANS cuu volser (RESEQ` commands

__ 8. On target machine issue `EXTERNAL cuu INT` commands

__ 9. Merge entries from source machine OWNED NAMES file with target machine

__ 10. Either merge duplicate archive IDs (if there are no naming conflicts), or notify the owner of their new archive and authorize them to access the new archive

__ 11. Change the `ERARCID` and `ERARCIU` options on the source system to automatically route commands to the new system

__ 12. Perform some test AQuery and AGet commands to check all is working.

12.9 Transfer of Backup and Archive Data to New Volumes

12.9.1 If the Data is on DASD

The following process describes how backup and archive data held on one set of minidisks may be moved to another set. This operation may be necessary when replacing old devices with new ones. The process described here is for guidance only and may need to be modified for local conditions.

1. Make a copy of the BACKUP VOLUMES file from the Interface 191 minidisk for use in the situation where something goes wrong.
2. Add a set of new minidisks to the nolog user ID's directory. These minidisks are the ones which will replace the old ones. If there are a lot of minidisks to convert, a small subset should be done at a time.
3. Use the VOLMAINT command to add the new minidisks to the BACKUP VOLUMES file as unformatted minidisks.

   Example:

```
vmbars volmaint 200 addbv sbb010 158
Ready;
HDFVCP0001I VOLMAINT command complete
```

4. Mark the source minidisks as full so they will not be used during the conversion process.

   Example:

   ```
   vmbars volmaint 001 full on 14feb95
   Ready;
   HDFVCP0001I VOLMAINT command complete
   ```

5. Copy the data from the source minidisks to the new minidisks.
6. Update BACKUP VOLUMES to set the data for the new minidisks.

   Example:

   ```
   vmbars volmaint 200 dyndata 001
   Ready;
   HDFVCP0001I VOLMAINT command complete
   ```

7. Delete the old minidisks from BACKUP VOLUMES.

   Example:

   ```
   vmbars volmaint 001 delete
   Ready;
   HDFVCP0001I VOLMAINT command complete
   ```

8. Delete the old minidisks from the directory of the nolog user ID.
9. Mark the new minidisks full with a date that will cause them to be reorganized at the next backup cycle.

   Example:

   ```
   vmbars volmaint 200 full on 1jan96
   Ready;
   HDFVCP0001I VOLMAINT command complete
   ```

10. If archive data has been moved, rename the backup to correspond with the new minidisks.

    Example:

    ```
    vmbars rename barnolog 001 200
    Ready;
    HDFREN0001I RENAME command complete
    ```

Until the source minidisks are deleted from BACKUP VOLUMES and the directory entries erased, the process can be backed out by restoring the saved version of BACKUP VOLUMES to Interface 191.

12.9.2 If the Data is on Tape

This is probably the simplest way of moving from one set of tapes to another, though not necessarily the fastest.

1. Use the VOLMAINT command to mark the source tapes as full so they will not be used for new backup or archive during the conversion process.

   Example:

   ```
   vmbars volmaint ad6 full on 14feb95
   Ready;
   HDFVCP0001I VOLMAINT command complete
   ```

2. Add the new tapes using the TAPE ADD command.
3. Ensure there are none of the old tapes that are 'unused' or 'reserved for reorganization'. If there are, delete them using TAPE DELETE.
4. Use the START REORG command to copy data from the source to target tapes. Do not specify more tapes than there are subusers available otherwise the old tape might get reused.
5. Issue TAPE DELETE for those tapes that have now been emptied so that they will not be used again,
6. Repeat the previous two steps until all the tapes have been converted.

The important thing to remember is that once a tape has been reorganized it might be used for a future reorganization and it must be deleted before that can happen. That is why you should not specify too many tapes on the START REORG command (not more than one per subuser).

12.10 Dealing with Bad Tapes

When certain errors have occurred writing to a tape it may be flagged as bad. As long as it is in this state it will not be used for writing new backup or archive data, but will continue to be used for the retrieval of existing data. The command DISPLAY TAPE shows whether a tape is bad or not.

A tape remains flagged as bad until either it is successfully reorganized, or REMOVE and REUSE commands are issued for the tape, or the VOLMAINT command is used to reset the bad status. The fact that a tape is marked as bad does not prevent it from being reorganized in the normal way.

12.11 Improving RESTORE Performance

BARS/VM allows the restoration of files, minidisks, directories, complete primary volumes and storage groups. A characteristic of incremental backup is that it spreads the backup files over a large number of backup volumes. This is because a small fraction of data is backed up each time and, since there are normally a large number of backup volumes, that data has a high probability of going on different volumes from the last time. However, it is possible to keep the backup files on a smaller number of backup volumes by backing up data that has not changed. This, of course, has an adverse effect on the performance of the backup process since files are being backed up when a good backup copy already exists. Installations must therefore balance the requirement to complete backup in a certain period of time with the requirement for reasonable restoration times. Since backup is normally carried out overnight it is often possible to allow extra time for it. It may also result in an increase in the amount of backup space used and it is recommended that the reorganization period is reduced to offset this.

The two installation options EMAXBD and EBRSBLK may be used to improve RESTORE performance at the expense of backup time. EMAXBD specifies the maximum number of backup volumes *per minidisk or directory*. If this option is non-zero, the backup process attempts to keep to this limit. A RESTORE of a minidisk or directory should therefore need no more than this number of backup volumes.

More important is the number of backup volumes containing level zero backup for each *primary volume or storage group* since this is the number of volumes required to restore that volume or group from backup. If the backup is on tape it is very important to keep this number low and it is advantageous even when backup is to DASD, though not quite so critical.

If EBRSBLK is set to a non-zero value, at the start of backup the total number of blocks used on each backup volume for level zero backup is computed for each primary volume and storage group. The number of backup volumes for which this value is not zero represents the number required to restore the real volume or storage group. This number is sent in a message to the manager user id together with the total number of blocks backed up so that the administrator may assess the time which would be needed to restore each primary volume and storage group. Then each file which is backed up to a volume which contains, for that primary volume or storage group, a number of blocks less than the number of blocks calculated from EBRSBLK is scheduled for backup again, even if unchanged. This number is calculated as

```
(number of blocks) * EBRSBLK/1000
```

In order to limit the impact on the backup process when EBRSBLK is first used on a system which has been running backup for some time, BARS/VM does not reduce the number of backup volumes by more than 20 at each backup cycle. The messages issued at the next backup cycle indicate the results of the volume reduction and allow gradual tuning of the EBRSBLK specification.

The larger the backup volume, the bigger the impact of this setting. For example, a 3490E tape may hold up to a million blocks, whereas a 158 cylinder 3390 minidisk will only hold about 50,000 blocks. Hence the tape volume would require a smaller value of EBRSBLK than if minidisks were being used.

It is strongly recommended to start with a value of 1 and increase it over time, observing the effects on the backup time and the number of backup volumes per primary volume.

Below is an example for a system which had been running for several months with EBRSBLK set to zero. It shows the messages issued for a particular primary volume. The value of EBRSBLK in this case was 10 and so the block count value in message 0502 is 10/1000 times the total block count in message 500. The backup process will attempt to backup again all files backed up on these 10 volumes in order to reduce the total number of backup volumes.

```
HDFMAT0500I Volume CORDM4 is currently backed up on 20 volumes (8899 blocks)
HDFMAT0502I Volume CORDM4 has 10 backup volumes with a block count of 88 or less
```

Below are the same messages issued the following day showing that the number of backup volumes was successfully reduced.

```
HDFMAT0500I Volume CORDM4 is currently backed up on 11 volumes (8948 blocks)
HDFMAT0502I Volume CORDM4 has 0 backup volumes with a block count of 89 or less
```

12.12 Applying Code Updates

See Chapter 22, “Applying Maintenance” on page 263 for details of how to apply code updates. This also explains how to modify the various control files.

12.13 Authorizing Users

12.13.1 OWNED authorization

An administrator may use the `OWNED` command to grant one user authority to issue any BARS/VM general user command on behalf of another user.

12.13.2 ESM authorization

An installation may choose to also allow control of access to BARS/VM data via the use of an External Security Manager (ESM), such as RACF/VM*. This option is controlled via the ESECDATA option, the value of which may be displayed via the `DISPLAY OPTIONS` command. If this option is set to `OWNED`, then the ESM is not used. If the ESM is used, then general users can control who has authority to access their BARS/VM data themselves, without involving the administrator. Any such access is ***in addition*** to access allowed via the OWNED command.

See 8.4, “Authorizing other users” on page 85 for a complete description of the ESM authorisations for each command.

12.13.3 HDFXGV authorization

Authorization may be controlled by the installation through a user exit, HDFXGV. This exit is called whenever the ESECDATA option is set so that an ESM would be used, e.g. ESM1. The exit can, by setting the appropriate return code, authorize the command, cause the command to be rejected, or defer authorization to the ESM. The default is to pass the decision to the ESM.

See 20.3.4, “User Exits” on page 239 for a description of HDFXGV.

12.14 Daily Checks

It is recommended that the BARS/VM administrator carry out the following checks on a daily basis. This will alert them to problems which may have occurred during the last backup cycle and allow for their resolution as soon as possible. It may also give them early warning of some potential future problems and enable them to take preventative action.

The log file generated on the manager user ID's A-disk (assuming the supplied manager PROP routines are being used) should be checked after the backup cycle has completed. This can either be done through linking and accessing the manager A-disk directly, or by using the manager `SENDREP` command. If SENDREP is used, then it is best to have the **Highlight_Report** and **Append_HM** manager options enabled (see 20.3.5, “Manager Installation Options” on page 247).

The `DISPLAY CURRENT` command gives an indication of whether the last backup cycle completed successfully. If the **current userid cuu** and **last userid cuu** fields are both blank, then the backup cycle completed. If they are non-blank, then backup did not complete within the backup window.

The `DISPLAY STATUS SYSTEM` command can be used to check that BARS/VM considers the backup cycle to have finished. The following response should be seen:-

```
Backup/Restore/DBR inactive
```

A QUERY * ALL (CHECK SHORT FILE command will return a list of all minidisks not successfully processed during the last backup run.

An EXEC to CP QUERY all the BARS/VM user IDs is useful for checking that the correct subuser plus interface user IDs are logged on, and that the other subusers only used during processes such as backup are logged off.

```
/* * * * * * * * * * * * * * * * * ** * * * * * * * * * * * * * * * * * */
/*                                                                       */
/*  EXEC to query the CP status of BARS/VM subusers.                     */
/*                                                                       */
/* * * * * * * * * * * * * * * * * ** * * * * * * * * * * * * * * * * * */

Address COMMAND

'CP QUERY USERS BARMANGR'
'CP QUERY USERS BARINTFC'
'CP QUERY USERS BARUSER1'
'CP QUERY USERS BARUSER2'
'CP QUERY USERS BARUSER3'
'CP QUERY USERS BARUSER4'
'CP QUERY USERS BARUSER5'
'CP QUERY USERS BARUSER6'

Exit
```

Figure 23. Example BARS/VM user ID query exec

In addition to running the above EXEC, a DISPLAY STATUS USERS command can be used to show what state BARS/VM considers the subusers to be in. The following response should be seen:-

```
Userid   Status           VH VHE Origin   Command
------------------------------------------------------------------------
BARUSER1 logged off       20  00 BARINTFC
BARUSER2 waiting for work 24  00
BARUSER3 waiting for work 24  00
BARUSER4 waiting for work 24  00
BARUSER5 logged off       20  00 BARINTFC
BARUSER6 logged off       20  00 BARINTFC
```

Part 4. 'Disaster Backup and Recovery Information'

Chapter 13. DBR Overview

13.1 Definitions

- "Backup" is defined as the set of copies of all data, programs, instructions and plans which is necessary to recover data for **all users** after data loss because of hardware, software or human error while normal computing center resources are available.
- "Disaster backup" is defined as the set of copies of all data, programs, instructions and plans which are necessary for resumption of normal computing center operations for **all users** while normal computing center resources are not available.
- "Vital Records" are defined as the set of copies of all data, programs, instructions and plans which are necessary for resumption of computing center operations for all **critical applications**, in the same or an alternate computing center. (Local corporate guidelines may have defined that process.)

The copies created for backup stay inside the computer center ("glass house"), the ones for disaster backup or Vital Records have to be stored at an outside location and therefore require removable backup media.

Note: BARS/VM is only able to provide data/program backup, that's why it is called "disaster backup" and not "Vital Records backup". One can use the disaster backup function to create data backup for a Vital Records backup, but remember that this can never be the complete process.

13.1.1 Reasons for BARS/VM Disaster Backup

The rapid increase in on-line DASD space is causing backup and recovery procedures to become more and more time consuming and administration intensive. In particular, the growth of the number of tapes has become a serious problem regarding space for archiving, administration effort and the objective of "unattended operation" and the elapsed time necessary for one backup cycle.

With an expected CGR (compound growth rate) of 40% a small system which currently needs 50 tapes for a DDR based disaster backup dump will need in 5 years about 200 tapes to do the same job. If the CGR is 60% the disaster backup dump will need about 335 tapes by then.

Administrator or user driven home grown procedures often result in incomplete data required for a possible recovery. So a process to create disaster backup data in the most automatic and reliable way is needed.

There is a trade-off decision to be made between speed and function. For CMS minidisks BARS/VM uses a device independent approach, which means dumping files for CMS minidisks. This gives improved function, but reduced speed. Dumping in non-CMS format will improve the speed, but reduce the function because individual files can no longer be recovered.

13.1.2 Which Data Needs Disaster Backup

In general only data which is needed to resume work after a disaster in the computer center is worth copying as disaster backup data and sending to a recovery location. Since the end-user should not have the burden of deciding which files he will need and which he will not, the smallest selectable unit is a minidisk.

An installation will normally decide to go with the default disaster backup policy, which is that all minidisks which have backup are selected for disaster backup automatically.

But there might be a point when the data to be dumped must be reduced by the computing center. Then the installation may choose the administration or user controlled way, where each minidisk has to be defined to BARS/VM to receive disaster backup.

Also there may be situations where it is desired to dump minidisks which have no backup, for example minidisks covering a complete volume when full pack dumps are required.

13.1.3 The Disaster Backup Policy and How to Change it

The installation may choose to reduce the amount of data being dumped by defining that each minidisk to be dumped must have been previously specified through the BACKUP command with the DBR option. This is achieved through the EDBRREQ installation parameter.

A second way of taking a dump of a limited set of minidisks is to define a logical group of minidisks, called an LSET. Facilities are provided to allow the dumping of a specified LSET.

The default DBR backup policy depends on whether an LSET is to be dumped and on the setting of EDBRREQ. It also depends on whether a minidisk has backup or not. In addition, this policy may be overridden by an option, TAKEALL, on the command to start DBR. The basic policy is as follows:

- EDBRREQ=NO
  1. If it is not an LSET dump, all minidisks are eligible which have backup.
  2. If it is an LSET dump, all minidisks in the LSET are eligible provided they have backup.
- EDBRREQ=YES
  1. If it is not an LSET dump, all minidisks are eligible which have backup and which have been specified for DBR by previous use of the DBR option of the BACKUP command.
  2. If it is an LSET dump, all minidisks in the LSET are eligible if they have backup and have been specified for DBR by previous use of the DBR option of the BACKUP command.

If TAKEALL is specified on the DBR START command, minidisks are eligible irrespective of the setting of EDBRREQ and whether they have backup.

For each eligible minidisk, a call is made to the user exit HDFXDP. This allows the installation to further modify the policy on a minidisk basis. A return code of zero from HDFXDP means continue using the eligibility already determined, as

above. Other return codes may be used to force DBR backup and specify which way the backup is to be done. See “HDFXDP” on page 244 for a detailed description.

13.2 Requirements

13.2.1 Requirements for Disaster Backup

1. Use of tape as medium for backup.

   Reason: The case of disaster requires “remote” backup data. This can be done by using removable media or by using the network. Using the network is too expensive because of the large amount of data in countries where network costs relate to the amount of data transported. Also using the network requires a computer center at the “remote” node.

   Decision made: Use tapes.

2. Minimize number of tape mounts.
3. Backup is to be done on a file level (not dump). The files are to be ordered corresponding to user IDs.

   Reason: Even if the data of a single user ID may not be vital as defined by the security guidelines, it could be necessary to have access to it before resumption of the normal computer center operation.

4. Frequency of backup must be able to be customized.

   Reason: The computer center must be able to adapt the frequency to its service level agreement.

5. The on-line user interface for the administrator must provide at least support for

   - A “quiesce” function
   - A “resume” function
   - Definition of (un-)available disaster backup tape volumes

   Output should be in a form as required by the system administrator responsible to run the disaster backup process.

6. In case of Abends restart of the procedure must be possible.

13.2.2 Requirements for Disaster Recovery

1. Restore must be possible for single user IDs in alternate computer centers.

   Reason: Even if the data of a single user ID may not be vital as defined by the security guidelines, it could be necessary to have access to it before resumption of the normal computer center operation. It must be possible to restore data of single user IDs in alternate computer centers which do not have BARS/VM installed.

2. The on-line user interface for the administrator must provide at least support for

   - Specification of user ID(s) to be restored
   - Specification of user ID minidisk(s) to be restored
   - Specification of ALL minidisks to be restored

In case of errors restart of the procedure must be possible. Provide a QUERY type function to determine the resources needed for the recovery process.

13.2.3 Minimization of Tape Operations

One of the most important objectives is the minimization of tape mounts. There are the following possibilities:

1. Reduction of saved data

   The discussion of possible criteria for data reduction for backup has not shown a reasonable way. Therefore no requirement for such a method has been formulated. However such a way may be desirable.

2. Reduction of backup frequency

   This is limited by the service level agreement.

3. Incremental instead of full backup

   This point resulted in BASE and DELTA dump types, where incremental means dumping all changed or new files (delta) since the latest base dumped. The BASE and the newest DELTA files will provide full restore capability.

4. Use of enhanced compaction

5. Automatic Mount System

   Use 3480/3490 tape systems with ACL feature (Automatic Cartridge Loader)

13.2.4 Source Data to be Dumped From

As mentioned above, only data in the local backup is considered to be relevant for disaster backup. This data is available on the user's minidisks and on the backup volumes of BARS/VM.

Using BARS/VM backup data has the following advantages:

- It can be guaranteed that during the disaster backup dump no files are open.
- Much better performance because of fewer LINKs and ACCESSes and the avoidance of compaction (BARS/VM data is already compacted).
- Consistency of data as good as in the local backup.

Backup from the BARS/VM data has the following disadvantages:

- If BARS/VM tape backup is used, backup files residing on tape will be obtained from the users minidisk.
- A lot of LINK/ACCESS overhead to BARS/VM backup volumes if backup data for a single minidisk is spread over many backup volumes.

Notes:

1. You may choose at DBR START time using the `DBR START ( BACKUP or ONLINE` command options from which source you wish to take the data.
2. Since in most cases only files described as being backed up in the BARS/VM backup control files are dumped to disaster backup media, you should verify that the last BARS/VM backup cycle has been completed **before** starting the disaster backup dump.

Chapter 14. DBR Data Reduction

14.1 How to Reduce Disaster Backup Data

Minidisk based
First eliminate the unallocated space on a real volume. This saves a lot of cylinders to be dumped.

Note: All incremental backup systems are based on minidisks.

File based On a CMS formatted minidisk it is easy to select data on a CMS file base. This eliminates dumping of unallocated space on each user minidisk. While copying at the file level, a lot of error handling can be done, so the resulting data is much more reliable than the output produced by DDR.

Note: If the files are copied from the incremental backup system they are never active, so there is significantly reduced error handling required to achieve the same data reliability as the backup system.

Compaction Compaction reduces the data by up to a factor of 3. But, compaction needs a lot of processor time. Here again the advantage of using the backup data is great, since files are already packed with sophisticated compaction routines.

Note: This implies that during a restore of disaster backup data a step is needed to unpack restored data to its original format.

Only data that is backed up should get disaster backup
It is not always necessary to backup data from guest systems or program products which can be easily reproduced - e.g. from PID tapes. So it is a good rule to take the data defined to have backup as the data which needs disaster backup.

Note: A command interface should be provided to exclude minidisks which have backup from being treated as disaster backup data.

Conclusion The data can be reduced to 1/7th of the originally allocated minidisk space, like the following example shows. It is based on a typical CMS intensive VM system with office communication.

14.1.1 Data Reduction

The following figure is a “picture” of a live system.

Data Reduction (Gigabytes)	
Datatype	Gigabyte
Total Dasd storage	50
System Minidisks (Paging, Nucleus, Operating Systems)	20
Allocated CMS Minidisks	21
VMBARS backup storage allocated	9
VMBARS backup level 0	3
Note: Only data in the local backup are considered to be relevant for disaster backup which is represented by BARS/VM backup level 0.	

Figure 24. Data Snapshot

Approximately 200 megabytes can be written on a standard 3480 tape. Using BARS/VM level 0 backup data as a base for disaster backup, the following amount of data must be expected for a “BASE dump” run. The assumption is 60% data growth rate per year.

Number of Tapes for Full Dump DBR				
Year	Total Minidisks		VMBARS Level 0	
1986	21	110	3.2	16
1987	33	170	5.1	26
1988	54	270	8.2	41
1989	86	430	13.1	66
1990	138	680	21.0	105
Note: The above figures relate to the average 3480 tape capacity.				

Figure 25. Estimated Number of Tapes for a Full Dump (BASE) Backup Cycle

By using a Disaster backup DELTA method, the following figure will result to the actual data of this live system:

Number of Tapes for Delta DBR			
Month	Gigabytes		Tapes
	GB Level 0	GB Delta	Tapes
Base	3.2		16
+ 1	3.36	.652	4
+ 2	3.52	.926	5
+ 3	3.68	1.188	6
+ 4	3.84	1.386	7
+ 5	4.00	1.630	8
+ 6	4.16	1.849	10
+ 7	4.32	2.096	11
+ 8	4.48	2.321	12
+ 9	4.64	2.553	13
+ 10	4.80	2.767	14
+ 11	4.96	2.997	15
+ 12	5.12	3.187	16
Base	5.12		26
Note: The above figures relate to serial processing. Therefore the total execution time can be minimized by using multiple subusers for the dump (means multiple tape units in parallel) which will result in not filling all tapes totally as serial processing would.			

Figure 26. Estimated Gigabytes and Tapes for a DELTA Backup Cycle

Chapter 15. Planning and Installation for DBR

15.1.1 DBR Command

The DBR command is used to start and query the status of the DBR part of BARS/VM. If the installation chooses to keep the control files of the DBR system online, the general user may also query his DBR status.

After the operator starts the DBR process, BARS/VM selects what needs to be copied to DBR storage minidisk by minidisk, keeping LSETs together, minimizing LINK/ACCESSes to backup volumes etc. Tapes are requested directly by the subusers. A set of standard label formatted tape volumes must have been defined by the DBRTAPE command before the DBR run is started. This set is called a **"logical SCRTCH tape pool"**. The operator receives a mount message asking for **any** tape out of this tape pool by replacing the standard label volid with the "logical SCRTCH tape pool" name as it has been defined by the installation in the EDBRPOOL installation parameter. This logic has been chosen to provide the ability to use the 3480/3490 ACL feature (Automated Cartridge Loader) to be able to process as automatically as possible.

Using the VMBARS DBRTAPE QUERY FREE *nn* command the operator can always ask which tapes can be used by disaster backup from a program logic point of view.

A "restart" function is provided via the VMBARS DBR (RESTART command option which is the default option. The DBR controller user ID checks if the restart is valid and informs the issuer if it is not. If it is valid, BARS/VM continues the current disaster backup dump with those minidisks which have not yet been completed.

Note: No tape is reused since it is expiration date protected.

15.1.2 DBRTAPE Command

The DBRTAPE command is used to create, update and query the tape volume list to be used for BARS/VM DBR runs. This list is contained in the file "RECOVERY VOLUMES" on the BARS/VM Interface user ID 2A1 disk and forms the "logical SCRTCH tape pool" which is maintained by BARS/VM itself.

All tapes which are added **MUST** be standard labeled (e.g. via TAPE WVOL1).

All tapes are referred to by BARS/VM DBR via the logical SCRTCH tape pool name which is defined in the installation parameter EDBRPOOL.

Tapes defined for disaster backup will be kept separate from the BARS/VM backup tape volumes.

15.1.3 LSET Command

The LSET command is provided to enable administrators to define, modify and query logical sets of minidisks (called "LSETs") which will be grouped together at disaster backup time on as few tapes as possible.

This allows recovery for an application defined as vital with much fewer tapes regardless of the names of the user IDs contained in the LSET.

The definition of LSETs is restricted to BARS/VM administrators since vital record procedures have to be followed.

Note: Please note that DBR is **not a vital record system**, but may be used to do the job of the Computing Center, which is copying the data and moving it out of the area. Additional tasks may be needed for Vital record management which is outside the scope of a Disaster backup system.

15.1.4 LPUT Command

The `LPUT` command is provided to enable administrators as well as users to define, modify and query logical sets of minidisks (called "LSETs"). The enhancement to the `LSET` command is the more user friendly definition of an LSET file, as well as the recognition of the tag **"*_COREQUISITE LSET:"**.

This tag allows the definition of other LSET file names which should be treated as a single entity in case of a disaster backup dump. The result will be a hierarchical structure of defined LSETs. Missing corequisite LSET files are ignored as well as minidisk objects which do not exist.

The `LPUT` command also forms the base for the sample VITAL application as prototyped in the on-line user interface.

The non-truncated LSET files are placed onto an optional 2A2 disk of the Interface user ID. This disk will also receive LSET based report files at the end of an LSET dump. This optional 2A2 disk is also the base for the sample VITAL application as prototyped in the on-line user interface.

Note that this does not mean that BARS/VM now provides a means of a vital record system, it is still only a method to dump data as part of any vital record process. Additional tasks may be needed for vital record management which is outside the scope of a Disaster backup system. Therefore the whole processing logic has been implemented in an EXEC called "`HDFDL1`". You may adjust the function therein carefully to match your needs. You should inform the product owner of your changes as they may contain general needs for the befit of other BARS/VM users.

15.1.5 Disaster Backup to Migrate Services

If you have to migrate applications from one system to another or to migrate entire services to another location, then the disaster backup function may help you to do the data transfer job. Especially when services are merged onto a single large system and renaming of user IDs on the target system is required, the RECOVERY function of BARS/VM disaster backup makes that job easier.

Since CMS file dumps are device independent, you may at the same time move to a different device type and/or a different CMS block size.

Because the recovery can take a lot of time, a "DELTAONLY" option has been added to RECOVERY EXEC. It allows just the recovery of new or changed files since the latest base dump, regardless how those base files have been recovered.

So the following steps may be used to migrate data for an application or system:

1. Run BARS/VM backup at the source system
2. Run a disaster backup BASE dump at the source system
3. Define the minidisks to be recovered on the target system

4. Recover BASE disaster backup on the target system during the next two weeks (or whatever period chosen)
   Notice: Do not let the new users work on the target system now.
5. Stop service on the source system at predefined time
   (This might be two weeks after the BASE dump).
6. Run BARS/VM backup at the source system
7. Run a disaster backup DELTA dump at the source system
8. Recover DELTA dump on the target system for the minidisks to be migrated by specifying “DELTAONLY” when running the RECOVERY EXEC to
   - Erase erased files from minidisks with changed data
   - Replace changed and add new files based on latest DELTA dump
9. Start service on target system

Note: If you do not plan to change anything while migrating a service (device type, minidisk size, CMS block size etc.), you may take a DDR copy of all the involved minidisks **in addition to** the BASE dump to enable a faster recovery of the BASE dump data.

15.1.6 What About the Risk of Damaged Tapes?

There is always a chance that a tape gets damaged. If normal DDR is used to dump data the complete minidisk or real volume may be lost because of a single read error on the tape. Disaster backup tapes are CMS file based, so the maximum loss is the loss of a single CMS file. If this file had been changed since the latest Base dump, which is true for active files, there is more than one version available on other Delta or the Base dump tape.

If a whole Base dump tape is lost there is no chance to recover the data which hasn't changed. For this it might be a good thing to copy the Base dump tapes to other tapes with the same volume serial numbers, e.g. via DITTO. The problem there is that BARS/VM fills the tapes until EOT is reached and the length of the tape media might be different.

It is not a good idea to only do Base dumps. A good frequency is to do a Base dump every six months, with monthly or weekly Delta dumps.

15.2 Disaster Backup Recovery Testing

Testing of a recovery is required for a vital business process and should be defined by your corporation. If there is no definition or no vital data contained in the disaster dump data, simply do it for some minidisks from time to time to verify your local recovery procedure.

Testing for a real recovery can be performed at different locations as long as there is no local dependency like data sharing between MVS and VM (recovery tests should also verify those dependencies).

It is the responsibility of the owner of the data to verify the correctness and completeness of the data after the restore process has been completed.

15.2.1 VM Disaster Data Restore Test

The disaster backup dump/restore process is a very simple process that ensures that the data dumped can be restored at another location. The dependencies on the host operating system are minimal, and any level of VM should be satisfactory (regular tests of the restore process will verify this).

A possible vital application process should be tested at least every year and should involve the following:

1. Taking the VM disaster backup dump tapes and a local recovery document, along with the recovery document of one of the user groups, to another location.
2. Restoring the data specified in the user groups document to verify that the restore process works and is adequately documented.
3. Checking of the restored data by the owner of the data.

15.2.2 Security of Data for the Test

Any VM minidisks must be set up in such a way that by default no one has access to the data unless authorized.

The test site should not backup any of the data restored during the test. If that data does get backed up, then once the test is complete the backup copy(s) of the data must be deleted.

Once the test is complete the disks (or parts of disks) that held the data during the test must be formatted to ensure that no residual data is left. Remove any user IDs and RACF/VM profiles setup for the test.

15.3 Installation

15.3.1 2A1 Disk

BARS/VM uses a special control file minidisk to store DBR information based on file level. Since the control files are EXEC readable to ease recovery, the size is about double the size of the Interface 191 minidisk if all data is to be dumped. If the data to be dumped is reduced by any of the methods discussed previously, the size of the 2A1 minidisk may be reduced accordingly.

Before starting any DBR preparation, ensure this minidisk is defined to the Interface user ID and formatted with a CMS block size of 4K.

15.3.2 2A2 Disk

BARS/VM uses a special optional minidisk to store DBR LSET information in a non-truncated format. This minidisk is also used to store LSET reports in a readable format at the end of an DBR LSET dump. The size is just a few cylinders, the worst case is to lose some reports. The disk is optional.

If LSET processing is to be exploited, ensure this minidisk is defined to the Interface user ID and formatted with a CMS block size of 4K. The minidisk should be defined in a way that all users needing access to it can read the data on there.

15.3.3 195 Work Disks

If you will be dumping data from the ONLINE storage, rather than from the BACKUP storage (see `DBR START` command), then you should ensure that you have sufficient work minidisks defined to allow for one per active subuser (see 6 on page 220). This will avoid contention if software compression is being used.

15.3.4 DBR Tapes

BARS/VM then needs a pool of tape volumes. Define and maintain this pool with the `DBRTAPE` command.

Notes:

1. Only tapes which contain a valid OS standard label will be selected for disaster backup work. The easiest way to write an OS standard label in VM is to use the **TAPE WVOL1** command. Be sure that the external label matches the internal label written by the TAPE WVOL1 command. BARS/VM can only verify the readable internal tape label.
2. Be sure that you add enough tape volumes. More can be added later if the dump does not complete.

15.3.5 Installation Options

Set the required parameters in the BARS/VM installation options file. Special care should be taken for the values of:

EMAXBD Number of backup volumes to be used for a single minidisk. Reduction of the number decreases the number of LINK/ACCESS commands needed to dump the information by increasing the LOAD to dump some more files on the BARS/VM backup system side. A good number to start with is 10, which is the supplied default.

EDBRMID Specifies the machine identification, which is used to build a simulated OS dataset name which is written on the disaster backup tapes.

EDBRPOOL Defines the name of the “logical SCRTCH tape pool” which is used in the mount command instead of the real volume label.

EDBRREQ This controls whether minidisk eligibility is controlled via the `BACKUP ... DBR` command, or not.

EDBRUSE This controls how many subusers are used to dumps minidisks belonging to a given userID.

EEXPBASE Specifies the number of days after which a DBR Base dump expires. The calculated expiration date will be written on the disaster backup tapes to protect the volumes. A good value to start with is 180 days, which is the supplied default.

EEXPIRE Specifies the number of days after which a DBR Delta or DBR LSET dump expires. The calculated expiration date will be written on the disaster backup tapes to protect the volumes. A good value to start with is 14 days, which is the supplied default.

15.3.6 Verify DBR Installation

Run a small DBR test. Just dump your own user ID via the BARS/VM command

```
VMBARS DBR START userID * (BASE
```

and see what happens. The EOUSERID should receive a calculation message and mount requests, BARS/VM will check the label of the tapes mounted and carry out the disaster backup dump and the control file dump.

Take that tape and try a simple recovery of one of your “user ID” minidisks to another one able to be linked by the user ID in which the RECOVERY EXEC will be run.

15.3.7 Disaster Backup System Report

BARS/VM produces report messages of its DBR activity. These are sent to the manager user ID in the same way as backup report messages.

A long or short report can be specified, again based on the value requested for the backup report.

Example:

```
V M B A R S  D B R  Report from                 09/25/87 14:56:47
  DBR started ( Delta dump )                  : 09/25/87 14:56:48
    ABCDEF   194       CMS 000010 cyl.  00013 files copied
    CDEFGT   222       CMS 000009 cyl.  00002 files copied
    USERX1   191       CMS 000010 cyl.  00008 files copied
    USERX1   291       CMS 000040 cyl.  00002 files copied
    USERX3   191       CMS 000020 cyl.  00038 files copied
    MVSXA    600   NON-CMS 000400 cyl.  00001 parts DDR'ed
    WIDUSER  191       CMS 000010 cyl.  00880 files copied
   *--------------------------------------*
   *    ALL minidisks have been DBR'ed    *
   *--------------------------------------*
   Number of NON-CMS minidisks          processed: 0000001
   Number of NON-CMS minidisk megabytes processed: 0000284
   Number of CMS minidisks              processed: 0000006
   Number of CMS minidisk     megabytes processed: 0000056
   Number of CMS files                  processed: 0000943
   Number of CMS files                  in error : 0000000
  DBR ended   ( Delta dump )                  : 09/25/87 15:09:40
```

Figure 27. A Sample DBR REPORT

15.3.8 Disaster Backup Tape Volume Report

BARS/VM keeps all DBR tape volume related information in the file “RECOVERY VOLUMES”. You may request the volume serial numbers used and other information, based on the various `DBRTAPE QUERY` command flavors, to be displayed at the terminal or to be sent to you as a spool file by specifying the option `FILE`. The tapes which contain control files are clearly indicated.

15.3.9 Disaster Backup Minidisk Reports

BARS/VM also keeps all disaster backup information in the “userID cuu” DBR control files which reside on Interface 2A1 disk.

With the `VMBARS DBR QUERY` command an authorized user ID is able to produce all kind of reports based on the options specified from an overall view down to one on the file level.

Note: It is not useful to print such reports because the control files are needed for recovery and so all information can be found in the “userID cuu” DBR control files at recovery time.

15.3.10 Disaster Backup Minidisk Report for General Users

Each general user can query his disaster backup status on a file, minidisk or LSET level. The results may be requested to be either displayed via messages at the terminal or as a file to be sent to the virtual reader.

15.3.11 Disaster Backup System LSET Report

BARS/VM produces a report for each LSET in case of a DBR LSET dump. These reports are generated on the optional Interface user ID 2A2 disk. The filename of the report is the same as the LSET filename.

Note that at the next DBR LSET run, which includes an LSET, an existing report will be replaced by a newer one.

Example:

```
*_OWNER: MANAGER NODEID * owner name, department, location
*_Authorized: TOM HARRY
*_LSET description: This LSET is a critical application.
*_LSET description: It is required to ...........
*_Critical LSET: N
*_Corequisite LSET: CMSR10 PL1
*_Corequisite LSET: PASCAL
*_Disaster Backup Recovery Plan: RECOVER PLAN USERREC 191
*_Cycle: 12 days
*_Cntrl tapes: H53132
*_Data  tapes: H53143 H53146 H53147 H53149
APPLID   0191     CMS    OK 3380         13.58 Megabytes
APPLID1  0301     CMS    OK 3390          0.158 Megabytes
APPLID1  049A     CMS    OK 3380          0.011 Megabytes
APPLID1  0491     CMS    OK 3380          0.101 Megabytes
APPLID1  0492     CMS    OK 3380          0.066 Megabytes
APPLID1  0497     CMS    OK 3380          0.093 Megabytes
APPLID1  0592     CMS    OK 3380          0.099 Megabytes
HARRY    0192     CMS    OK 3380         16.30 Megabytes
USERREC  0191     CMS    OK 3380         54.33 Megabytes
```

Figure 28. A Sample DBR LSET report

15.4 What You Can Expect from DBR

The following are samples experienced on real life systems.

Capacity:	20 Gigabytes defined minidisks
Files:	400,000 on 2400 minidisks
Backup level 0:	3.3 Gigabytes packed data
DBR Base Dump:	6.5 hours / 24 tapes / 3.3 GB data
DBR Delta Dump:	1.5 hours / 3 tapes / 320 MB data
Subusers defined for backup:	5

Note: Performance improvements have been added to the BARS/VM DBR special tape module HDFTAP.

15.5 What You Cannot Expect from DBR

Normally all dumps are carried out of a file base for CMS files, and so performance never matches that of DDR, for example. DDR uses full track read and dump methods without any consistency checking, BARS/VM uses normal CMS file operations which are much slower. CMS data can be dumped as non-CMS - see the section on changing the default policy for details.

To speed up the performance you should use multiple subusers for the unlikely event of a recovery being necessary. There is no stand alone loader function for disaster backup. Although you do not need a functional BARS/VM for recovery, you do need a functional CMS to work with. Minidisk without normal backup will not automatically receive disaster backup because of the data reduction policy.

Chapter 16. DBR Dump Process

The process provides a means of dumping to tape all VM data which has been defined to receive disaster backup, either by user or administrator driven commands or by the default selection policy. In all cases definitions are made on a minidisk basis.

The process consists of a **BASE** dump and several **DELTA** dumps between BASE dumps. The minidisks may be restored on any VM system with the **RECOVERY EXEC** (see 18.3, “Load the Recovery Service Programs” on page 202 for how to obtain the RECOVERY EXEC). ***Important:*** Each system running BARS/VM has its own independent disaster backup dump and therefore has to be recovered separately.

Each dump uses its own set of tapes from the “logical SCRTCH tape pool”. The used tapes are identified by the “VMBARS DBRTAPE QUERY BASE or DELTA (FILE” command, which resulting file can be used as supplementary data together with the disaster backup dump tapes.

BARS/VM will control its tape pool, so mounting unneeded or unusable tapes will not be reflected in the tape pool control file.

16.1 Types of DBR Dumps

16.1.1 System Wide DBR Dump

There are two ways of providing a system wide disaster backup dump. Both use the same method to determine the set of minidisks to be dumped. If the EDBRREQ option is set to NO, then all minidisks which are backed up by BARS/VM are eligible for dumping. If the EDBRREQ option is set to YES, then only those minidisks for which a `BACKUP ... DBR` command has been issued are eligible. Also see the section entitled 13.1.3, “The Disaster Backup Policy and How to Change it” on page 174.

Base Dumps Only

The first one is a series of Base dumps which always dumps all files from all minidisks selected to be dumped. This is achieved by a `DBR START (BASE` command.

The advantage is that only one set of tapes is required for recovery, and hence the time taken to recover will be less. The disadvantage is that every dump will process all selected data, which can take a long time.

Base and Delta Dumps

The second is via the Base/Delta methodology, where one Base dump is followed by a sequence of Delta dumps. A Delta dump dumps all new or changed files from all minidisks selected to be dumped. This is achieved by a `DBR START (DELTA` command.

Each Delta dump is cumulative, which means that a Delta dump contains ***all*** changes done since the latest Base dump, and hence will obsolete the previous Delta dump. For recovery, you will only need the newest control file tape which is the one from the newest Delta dump.

The advantage is that a Delta dump should take significantly less time than a Base dump, because only the changed subset of data is dumped. The disadvantage is that two sets of tapes are required for recovery. However, most installations perform many more dumps than recoveries, and hence this second approach gives an overall saving.

Guarding Against Data Loss

Regardless of which method is chosen, the latest version of any file is only available from one tape. If that tape, or part of it, is unreadable for any reason, then the file cannot be recovered.

To reduce this risk, an installation may wish to make copies of the Base dump tapes immediately after the dump. This is a better idea than reducing the interval between Base dumps, since you still gain the benefits of the Base/Delta system, whilst guarding against tape damage occurring before the next Base dump. Of course, if Delta dumps are not being used, then the following paragraph referring to Delta tapes will apply to the Base tapes instead.

For Delta tapes, having access to a previous set of Delta dump tapes allows for the previous copy of a file to be recovered if the latest version is unreadable.

Studies have shown that it is a good starting point to do a Base dump every 6 months and a Delta dump once a week. The break-even point for a new Base dump strongly depends on the data change rate on the system.

16.1.2 Base Dump

The Base dump is done on an periodic basis, and is a complete dump of ***all*** the selected minidisks. Two sets of tapes are required for this process to prevent an exposure during the actual dump process.

At the end of the dump process, a separate control file tape is created which contains all data required for the recovery process. This control tape contains the control files which describe which tapes hold which minidisks for this Base dump. No disaster backup data is held on the control tape. The installation may request that a duplicate control tape is created for added security. See the information for the EDBRCDUP installation option.

Warning: It is not possible to recover data without the files contained on the control file tape.

16.1.3 Delta Dump

The Delta dump is done much more frequently, and is a dump of ***all*** files which have been created or changed ***since the last Base dump***. This means that each successive Delta dump will contain all the data contained in any previous Delta dump, and which is still considered to be vital.

Again, keeping two versions of Delta dumps at the vital record site prevents an exposure during the actual dump process.

At the end of the dump process, a separate control file tape is created which contains all data required for the recovery process. This control tape contains the control files which describe which tapes hold which minidisks for this Delta dump. No disaster backup data is held on the control tape.

Again, without the files on the control file tape, data cannot be recovered from the dump. A duplicate control tape may be requested through the EDBRCDUP installation option.

16.1.4 LSET or User ID

An LSET dump is way to provide the data dumps for “critical applications” which need a special backup beside the normal Base or Delta dump logic. Mainly this is needed because there is a need for a higher dump frequency than the normal disaster backup provides. Because there is also a need for fast recovery, each LSET dump contains all files from all selected minidisks, just like a Base dump.

The minidisks to be selected for a given LSET dump are defined through the authorized BARS/VM LSET command. This dump is started via a `DBR START LSET` *`lsetname`* `(BASE` command. Note that the option BASE must be specified.

A special case is an LSET dump for a user ID or just some minidisks belonging to a single user ID or a group of user IDs defined by a LISTFILE pattern. This special case needs no LSET definitions via the LSET command, it is just started by `DBR START` *`userID cuu`* `(BASE`. This is called a user ID dump. The selection logic is the same as for an LSET with one “userID cuu” defined.

Note: Base and Delta dumps always group together as close as possible minidisks defined in LSETs or minidisks from the same user ID, so if there is no requirement for a shorter disaster dump frequency, the normal disaster dump data can also be used to recover LSETs. From a recovery view there are just a few more tapes to be mounted due to the need to mount both Base and Delta tapes.

16.1.5 The VITAL Sample Application

BARS/VM provides a sample vital records management application which is based on the LSET dump method. As the vital records process is different most everywhere, it has been implemented using CMS PIPELINE (EXEC HDFDL1). The EXEC can be carefully adjusted to your local needs.

The enhancement above the usual LSET command manipulation of DBR LSET files is the delegation of the LSET maintenance to the responsible user groups. The BARS/VM on-line user interface provides panels assisting the users in manipulation their LSETs, as well as allowing them to view other LSET definitions. Any user can choose any LSET as `COREQUISITE` to his definition. This avoids redundant definitions and is fully transparent to the user.

To provide the information to the user, Interface user ID 2A2 disk will be accessed in read/only mode. The data can be manipulated through the use of the sample panels. An alternative is the use of an editor combined with an `LPUT` command. The LSET file will be placed on the 2A2 disk after authorization check as it is, and on the 2A1 disk in a truncated format as the DBR dump process expects it to be. Intermediate `LSET` commands, which may modify an LSET file, will pass through their information changes to the file on the 2A2 disk.

16.2 Restart Considerations

BARS/VM can be setup to process all minidisks defined to BARS/VM or a specified Logical Set of minidisks (LSET) or a specified user ID/minidisk pair. DBR will react to all errors with messages sent to the BARS/VM manager user ID and the defined operator id. Since DBR can be run during prime shift it expects operators to be there to correct detected problems.

To restart after any BARS/VM or system related interruption of a DBR run just reissue the last DBR START command with the RESTART option, which is the default. If you re-enter the option BASE or DELTA then the whole process starts from the beginning.

Note: No tape volume which has already been used will be reused.

16.2.1 Verification that DBR is Completed

To verify if BARS/VM disaster backup dump has been completed, just issue a `VMBARS DBR (RESTART` command again and the system will tell you if the process has been completed. If it is not, no control file tape has been created and without that disaster backup data tapes are not useable.

16.3 Operation Tasks

16.3.1 DBR “logical SCRTCH tape pool”

DBR uses a new term to access tapes which we call **“logical SCRTCH tape pool”**. A “logical SCRTCH tape pool” is a set of standard labelled tapes which are being referenced with a unique “pool name”. In case of BARS/VM disaster backup this tape pool is maintained through the BARS/VM DBRTAPE command. The tape pool name is defined in the BARS/VM installation options file via the field EDBRPOOL.

The concept of the “logical SCRTCH tape pool” allows BARS/VM disaster backup to fully use the 3480/3490 ACL (Automated Cartridge Loader) feature to automate the tape mounts as far as possible, whilst providing full security checks against OS standard labels. If there are enough ACLs available to pre-mount the necessary tapes, then the whole disaster backup will require no manual intervention.

The tapes may be mounted in any order, since BARS/VM will verify the labels. BARS/VM checks in the following sequence:

1. Is the tape OS standard labelled?
2. Is the tape volid defined in the “logical SCRTCH tape pool” list (which is the file RECOVERY VOLUMES maintained by the VMBARS DBRTAPE command)?
3. Is the tape (still) expiration date protected based on the internal file “RECOVERY VOLUMES”?
4. Is the tape protected by the real tape expiration date in the OS HDR1 label?
5. If all above points show that the volume is useable, it will be used.

By providing this technique, any series of pre-mounted volumes can be used, therefore the 3480/3490 ACL feature (Automated Cartridge Loader) can be fully used. Operations are able to pre-mount tapes for DBR and let it work “nearly” automatically.

BARS/VM subusers will check all CMS defined tape addresses for pre-mounted tape units before issuing a MOUNT command. Before detaching a tape unit a check loop is run to detect ACL features. The CMS defined tape addresses are checked in the order: 181, 180, 182 to 187, 288 to 28F.

Note: If there are too many tapes mounted in a 3480/3490 stacker and BARS/VM disaster backup ends, all tapes in the unit are unloaded automatically.

16.3.2 DBRTAPE QUERY Operator Interface

Use the `DBRTAPE QUERY FREE nn` command to display a list of tape volumes useable for BARS/VM DBR runs. After BARS/VM completes its calculation run, it will ask for a number of given tape volumes, followed by a simulated `DBRTAPE QUERY FREE nn` command. There is no need to mount the indicated volumes, any suitable free tapes will be accepted.

It is a good idea to mount more tapes than BARS/VM asks for. This is because BARS/VM cannot always put 170 megabytes on each tape volume (the basis for the calculation) since the capacity is related to the size of the files. Also, when a subuser has to be restarted, the active tape will **not** be reused.

Another form of the DBRTAPE QUERY command is used to figure out which tapes belong to which generation of a Base, Delta or LSET dump. For a more detailed description see the BARS/VM DBRTAPE command as described in this manual or use the BARS/VM HELP facility.

16.3.3 DBR START

Use the `VMBARS DBR START` command to start a disaster backup run.

For a Base or Delta dump do **not** specify a “userID cuu”. Specifying `DBR START * * (BASE` is **not** the same as specifying `DBR START (BASE`, since specifying any “userID cuu” is treated as a special LSET dump with no relationship to the normal Base/Delta logic.

You may issue the `DBR START (RESTART` command to restart an uncompleted disaster backup dump of any type.

If you do not mount any tapes for BARS/VM, it will wait some time and then the DBR activity stops. Later you may use the DBR START command to restart the started run. This allows you to get the calculation of the needed tape volumes during prime shift, and then prepare and pre-mount the tapes needed. Then the DBR START command can be used to start the actual dump processing out of prime shift (when the system performance allows).

Note: Since it is nearly automated, starting DBR outside prime shift is usually a good idea.

16.3.4 DISPLAY STATUS RESOURCE

Use the `DISPLAY STATUS RESOURCE` command to get information about the currently processing minidisks and tape volumes. This may also help with identifying resource problems.

A message will be sent to the BARS/VM manager user ID for each minidisk when work is scheduled to a subuser.

16.3.5 STOP NOW or SHUTDOWN

Use the STOP NOW or SHUTDOWN command to stop DBR activity. Since there is no time cycle defined for disaster backup, it will continue processing as long as there is any work left to be done. Work should not be done during prime shift since other users may be affected if BARS/VM backup volumes are behind user or paging areas. Additionally, BARS/VM response will be degraded for users since the subusers will be busy dumping files.

16.4 Supplements to Tapes Stored Outside

To be able to restore data produced by BARS/VM DBR you have to do the following:

1. Determine on which tape volume the control file information resides. This can be done by means of the `VMBARS DBRTAPE QUERY BASE or DELTA (FILE` command. All control file tapes are identified by the word **CNTRL**. These volids are needed for recovery and **MUST** be identifiable at the restoring location.
2. Ship the tapes to the vital record location to deposit the tapes. You should sent along with the tapes:
   - Recovery documentation
   - Volids of the control files tapes
   - Possible additional vital records descriptions

Note: The recovery syntax can be displayed on-line via `RECOVERY ?`.

Chapter 17. Suggested Disaster Recovery Process

The recovery process is designed so that there are no dependencies on the software available on the VM system used to recover the data. Although the BARS/VM Disaster Backup and Recovery (DBR) feature is used to dump vital data, there is no requirement to install BARS/VM to restore data. All necessary programs and files reside on the dump tapes themselves.

The person responsible for data recovery must have access to the latest set of base and delta dump tapes, and to any hard-copy tape lists which may accompany those tapes.

Boxes holding the tapes and the hard-copy tape lists should be externally labeled in a dated form, see sample below. Hard-copy tape lists should be kept in these boxes. They give details of the tapes used for the dump.

```
********************** VM Disaster Backup Dump **********************
*                                                                   *
*                                                                   *
*                  Corporate Confidential Restricted                *
*                                                                   *
*                        Dump Cycle: Delta 2                        *
*                                                                   *
*                        Dump Date:  7 Dec 1991                     *
*                                                                   *
*                  Location xyz, Address information                *
*                                                                   *
*                                                                   *
********************** VM Disaster Backup Dump **********************
```

Figure 29. Example of Tape Box Label

17.1 Recovery Requirements

17.1.1 People

The recovery process should be performed by someone with a basic knowledge of CP, CMS, data management and directory administration. Also, the assistance of an operator for mounting tapes and solving tape unit problems would be helpful.

The setting up of the user IDs (along with the necessary authorizations (e.g. RACF/VM)) will require the cooperation of the relevant people at the recovery site.

17.1.2 Hardware

- At least one 3420 tape drive and one 3480/3490 tape drive, depending on what types have been used for the disaster dump.
- At least one 3270 type terminal.
- The restore of disaster backup data requires real dasd capacity. You may calculate the actual number at recovery time after you have decided what to

recover and by adding the corresponding values of the minidisks defined in the recovery control file “RECOVER VARS” and the appropriate minidisk values as defined in the informational file “RECOVER DIRECT”.

17.1.3 Software

- VM/ESA 1.1 or a later release.
- CMS 8, or higher, is required to run the RECOVERY EXEC.
- BARS/VM disaster backup data can be recovered on systems without the need for a running BARS/VM backup system, either on a newly generated system (“starter system”) or on a runable system providing service for other users. The recovery procedure allows for renaming of user IDs and/or minidisks so that an existing environment has not to be changed when recovery for other systems is performed on an active system.
- The user IDs performing the recovery must be able to link to all minidisks without or with the same password. You should provide one of the following 3 methods:
  1. RACF/VM OPERATIONS attribute (if RACF/VM is active)
  2. LNKNOPAS directory option (VM/ESA 1.1 or higher)
  3. Same Multiwrite password on all minidisks involved in the process (minidisks of recovery user IDs and minidisks to be recovered).

17.1.4 VM Classified Disaster Backup Data

Please note that some of the disaster backup data may be classified vital data (e.g. “Corporate Confidential”). It should be treated as such.

17.2 RECOVERY EXEC Syntax

RECOVERY Syntax

```
RECOVERY userID cuu      < AS userID <cuu>> < ( options >
         LSET   lsetname
         VOLSER volid
```

Function: Recover DBR (Disaster Backup & Recovery) data created by BARS/VM DBR function.

Where:

userID Cuu specifies the minidisk to be recovered.
Both can be specified using LISTFILE syntax.
e.g. To recover all minidisks from user ID MAINT
issue **RECOVERY MAINT ***

Note: The default for Cuu is '*'

LSET lsetname specifies the LSET (logical set of minidisks) to be recovered. The LSET file must exist on the control file disk in a format like it is created by the BARS/VM LSET command.

VOLSER volid specifies the real volid of a volume to be recovered. 'VOLSER volid' is replaced by the 'userid Cuu' definitions which did exist on the volume with the label volid.

AS userID Cuu specifies the user ID minidisk value to be used to recover if the user ID/minidisk needs to be renamed. You may specify '=' to use the same value as userID/cuu were originally.
e.g. To recover all minidisks from user ID MAINT as user ID MAINT2 (since MAINT exists..)
issue **RECOVERY MAINT * AS MAINT2 =**

Note: The default for Cuu is '='

Options:

NOAPPend specifies that the old RECOVER VARS * file should be erased before adding new entries.

GETTapenr specifies that only the number of volids to be used for the Recovery should be passed to the calling REXX EXEC preceded by a return code.

FORMAT specifies that CMS minidisks should be reformatted if the CMS block size does not match the original one. If not specified, only the CMS label will be reset to the original CMS label.

NOFORMAT Only switch the CMS label to the original one. This allows recovery with a different CMS block size that the original one.

Note: If a CMS minidisk is found which is not formatted at all, it will be formatted regardless of FORMAT/NOFORMAT option.

RECover Do recover work now.

VALIdate Validate all minidisks defined in RECOVER VARS * control file. A message will be issued for each LINK error and the user will be prompted for SKIP, RENAME or QUIT.

Note: Do not run this step in DISConnect mode.

DELTAonly Used to recover only files which have been changed since the last BASE DBR run. It does assume that a BASE DBR or an equivalent DDR dump is already recovered. DELTAonly will also erase all files from the target minidisks processed which are no longer found in the minidisk control files.

READData Read the tape containing the control file.

Note: You must know on which tape the control files exist and mount it on virtual 181 address after the prompt or on any valid CMS tape address before starting the EXEC.
The range checked is: 181 180 182 183 184 185 186 187 288 289 28A 28B 28C 28D 28E 28F

FILEMode fm specifies the filemode where the control files exist for processing.

Note: Control files are always loaded to A-disk.

MULTIPass pw specifies the password to be used to link to all minidisks in MR mode. If you have VM/RACF or equivalent installed, specify NONE to omit a password on the LINK command.

Note: For VM/RACF the recovering user ID(s) need the operations attribute, other security packages an equivalent authorization.

OPERatorid id specifies the user ID to receive the MOUNT message if nothing is specified, 'OP' will be used.
Current default is: OP

INFOrmid id specifies an additional user ID to receive the messages issued to the console of the recovering user ID. This is especially useful if you recover with more than one 'worker' user IDs.

17.3 Format of DBR Tapes

All Disaster backup tapes have the same format. In the control files loaded from the control file tapes there is no differentiation between BASE, DELTA or LSET/user ID dump types.

BASE and DELTA dumps do not have control file information of the same tape volume which contains data dumped, the control information is kept on different “Cntrl” file tapes.

LSET or user ID dumps combine both, data and control files, on the last tape volume dumped, which may also be the only one. This have been done since very often an LSET/user ID dump contains just a few megabyte of data, not filling a complete tape.

17.3.1 General Tape Layout

1. All tapes receive an OS-type standard label and an OS-type HDR1 record containing a generated OS GDG-name and an expiration date to protect the tape. After that service modules needed to load the tapes follow. A tape mark (TM) closes this first logical file on the tape.
2. The second tape file contains dumped files, which are not separated by tape marks. They therefore simulate a multivolume tape file.
3. The third tape file contains control files needed for recovery, other service modules for recovery and supplementary files.
4. The last control file or data tape contains about three tape marks to indicate end of tape.

```
VOL1                Standard label

HDR1                OS-style HDR1 label (with expiration date)

HDFTAP MODULE   Program required by RECOVERY EXEC
DMSTAP MODULE     "      "    "    "     "
HDFSLC MODULE     "      "    "    "     "
RECOVERY EXEC   The EXEC which performs recovery

<tape mark>         Tape mark

YYDDD seqnr         backup file for a minidisk
YYDDD seqnr           "    "   " "   "

<tape mark>          "   "

userID cuu          Control file for a minidisk
userID cuu            "    "   " "   "
  :    :              "    "   " "   "
  :    :              "    "   " "   "
userID cuu            "    "   " "   "

<tape mark>         Tape mark (end of tape)
<tape mark>          "   "    " "  "
<tape mark>          "   "    " "  "
```

Figure 30. Format of DBR Control Tape

Chapter 18. Disaster Recovery Steps / How to Recover Data

18.1 Setup User ID(s) to Perform Data Recovery

At least one user ID should be defined on the system for use by the person(s) responsible for data recovery (to reduce the overall time taken for the recovery of a large amount of data, several recovery user IDs should be used). The special requirements of this virtual machine are as follows:

Virtual Storage The minimum storage size for normal operations is 2M, the suggested size is 6MB.

Minidisks The following minidisks are required:

191 Work disk: Ten cylinders of 3390 should be sufficient.

192 Must be large enough to contain all the BARS/VM DBR control files and recovery programs/utilities loaded from the control file tapes.

Authorization The user ID must be authorized to link read/write to all minidisks which will be recovered (obviously this must be done after the user IDs have been setup).

Sample Directory Entry

```
USER VITAL xxxxxxxx 6M 16M G
ACCOUNT nnnnnnnn
OPTION LNKNOPAS      ( XA/SP 2 and later)
MACHINE 370          ( XA/SP 2 and later)
IPL CMS
CONSOLE 009 3215
SPOOL 00C 2540 READER A
SPOOL 00D 2540 PUNCH A
SPOOL 00E 1403 A
LINK MAINT 19E 19E RR
LINK MAINT 190 190 RR
MDISK 191 3390 cyl 010 volser MR
MDISK 192 3390 cyl 050 volser MR
*MDISK 195 3390 cyl 020 volser MR
```

Figure 31. Sample Directory Entry for Recovery User ID VITAL

If extra recovery user IDs are required then it is suggested they are setup in the following way:

```
USER VITALn xxxxxxxx 6M 16M G
ACCOUNT nnnnnnnn
OPTION LNKNOPAS      ( XA/SP 2 and later)
MACHINE 370          ( XA/SP 2 and later)
IPL CMS
CONSOLE 009 3215
SPOOL 00C 2540 READER A
SPOOL 00D 2540 PUNCH A
SPOOL 00E 1403 A
LINK MAINT 19E 19E RR
LINK MAINT 190 190 RR
LINK VITAL 192 192 RR
MDISK 191 3390 cyl 010 volser MR
```

Figure 32. Sample Directory Entry for Recovery Subuser VITAL(n)

18.2 Logon to the 'VITAL' User ID

It is recommended that the console be spooled during the entire restore process, and any messages noted in case action is needed.

Logon to the 'VITAL' user ID and issue the following commands:

1. FORMAT 191 A (BLKSIZE 2K

   Reply “YES” to the first prompt
   And “VIT191” to the second prompt

2. FORMAT 192 A (BLKSIZE 2K

   Reply “YES” to the first prompt
   And “VIT192” to the second prompt

18.3 Load the Recovery Service Programs

Arrange for any current disaster backup tape to be attached as 181. Enter the following commands:

1. TAPE FSR 2
2. TAPE LOAD * * A (EOF 2
3. TAPE RUN

If problems occur, try a different tape or tape drive.

The required recovery programs/utilities are now on the 192 minidisk which may be shared with additional recovery user IDs.

The following is a list of the four programs:

- HDFTAP MODULE
- DMSTAP MODULE
- HDFSLC MODULE
- RECOVERY EXEC

18.4 Load the Latest Control Files

Select the newest control file tapes for this system.

When delta dumps exist, be sure to use the DELTA control tapes, and **not** the BASE control tapes.

Note: You should never load control tapes from several systems if you have to recover more than one system. Recover systems one by one.

Figure 33 shows an example of the hard-copy tape list for one system, and which will be found in one of the boxes. The first line may contain the cycle name, system name and date of dump. The next line may indicate the used tape unit device type (3420, 3480, 3490). These lines have to be generated locally. The remaining lines give details about each tape, derived from "VMBARS DBRTAPE QUERY BASE and DELTA" commands.

```
   VM Disaster Dumps Delta 2      Nodeid            Classification         15 Aug 1990


      * 3480 ** 3480 ** 3480 ** 3480 ** 3480 ** 3480 ** 3480 ** 3480 ** 3480 *


Volid  Density  Created Expire  Run id.  Gnum Seq    Status information
010020 38K      14MAR90 10SEP90 RZ       0002 001    BASE               Cntrl
010040 38K      14MAR90 10SEP90 RZ       0002 001    BASE      full
010041 38K      14MAR90 10SEP90 RZ       0002 004    BASE
010042 38K      14MAR90 10SEP90 RZ       0002 002    BASE      full
010043 38K      14MAR90 10SEP90 RZ       0002 003    BASE
010046 38K      15AUG90 29AUG90 RZ       0004 002    DELTA              Cntrl
010047 38K      15AUG90 29AUG90 RZ       0004 001    DELTA              Cntrl
010048 38K      15AUG90 29AUG90 RZ       0004 002    DELTA
010049 38K      15AUG90 29AUG90 RZ       0004 001    DELTA


                                                                        Page 1
```

Figure 33. Example of Hard-Copy Tape List

Arrange for the latest control file tape to be attached as 181. The control file tape can be identified by referring to the hard copy document accompanying the tapes, and is flagged with "Cntrl" on the right-hand side of the line. If there is more than one such tape, then you have to mount them in ascending order, based on the sequence number they received. (see field "Seq" in the hard-copy tape list). In the previous sample hard-copy list you would have to mount the control file tapes "010047" and "010046" in order.

Note: All tape mount requests will be sent as messages to the operator id (default OP) or whatever you specify when being asked by the RECOVERY EXEC. This may be overridden always with the OPERATORID option of the RECOVERY EXEC - however, this must be included on all calls to the EXEC. The full syntax for the RECOVERY EXEC may be found in "RECOVERY Syntax" on page 196.

Enter the following commands to load the control files:

1. ACCESS 192 A
2. RECOVERY (READDATA

Answer the questions to create the RECOVERY PROFILE (first RECOVERY invocation only).
Remember to answer “D” if you are getting asked for the FILEMODE containing the recovery control files.

The 192 disk will now contain the RECOVERY DATA, RECOVERY PROFILE and all the service modules needed for recovery plus all the control files (identified by filemode number 5).

18.5 Recover the Minidisk Containing the VM Source Directory

To be able to define the user IDs to be recovered you need the original directory definitions. These are contained in the file “RECOVER DIRECT” which is now on VITAL 192 disk. In that file all passwords have been replaced by “XXXXXXXX”.

If this is sufficient for you, you may skip this step. If not, you might recover now the minidisk which contained the original directory (e.g. DIRMAINT 195).

To do so, you have to define a 195 disk in user ID VITAL which is about the same size than the original DIRMAINT 195 (see commented minidisk sample for 195 in VITAL directory).

Then enter the following commands:

1. ACC 191 A
2. ACC 192 D
3. CP DETACH 195
4. RECOVERY DIRMAINT 195 AS VITAL 195
5. RECOVERY (GETTAPENR
6. RECOVERY (RECOVER FORMAT
7. CP LINK * 195 195
8. ACCESS 195 C

Mount the tape(s) listed via the GETTAPENR command in any order when needed.

The original source directory is now on the 195 C-disk.

18.6 Setup the User ID(s) which Have Minidisks to Be Recovered

Add the required user IDs to the existing system directory. These can be extracted directly from the file “USER DIRECT” which is now on the VITAL 195 minidisk if step 5 was executed or from “RECOVER DIRECT” on the VITAL 192 minidisk. Remember to allocate the minidisks afresh.

If any user ID names conflict with existing ones, they will have to be given new names when added to the directory. In addition, the recovery process must be informed of the change. This is performed as part of the next step.

Note: Ensure that the VITAL(n) user ID(s) have read/write authority to the relevant disks.

18.7 Create the RECOVER VARS Work File

LOGON to the VITAL user ID

If only selected user IDs and minidisks are to be recovered, enter a separate RECOVERY command for each user ID and/or minidisk.

To recover all minidisks, enter the command 'RECOVERY * *'. This updates a work file (RECOVER VARS) used later in the recovery process

The command syntax is:

RECOVERY userID cuu AS new-userID new-cuu

18.8 Verify All Minidisks to Be Recovered

Again, this verification is performed by the RECOVERY EXEC, and is done by issuing the following command:

RECOVERY (VALIDATE

It performs the functions:

1. Checks that each minidisk to be restored can be linked R/W. If a LINK fails, check that the VITAL(n) id is permitted to LINK the minidisk R/W.
2. Ensures non-CMS minidisks are the same size as the original. If a disk is not the correct size, check its definition in the directory against the original directory entry (VITAL 192 or VITAL 195 minidisk).
3. If a minidisk is not defined, you will be prompted for RENAME (e.g. rename to another minidisk) or SKIP (do not recover this minidisk).

Copy the resulting RECOVER VARS file to the VITAL 192 minidisk to allow for multiple user ID recovery:

```
COPY RECOVER VARS A = = D (OLDDATE REPLACE
```

18.9 Logon to the Additional 'VITALn' User IDs

This step is optional, and should only be performed if multiple recovery user IDs are being used. If you do so, make sure that user ID **VITAL is logged off**.

Logon to each additional user ID and issue the following commands:

1. FORMAT 191 A (BLKSIZE 2K
2. ACCESS 192 D

It is necessary that VITAL 192 minidisk is not linked in R/W mode by any user ID to be able to update the RECOVER VARS file there after each completed tape volume by any user ID doing recovery work.

All subusers therefore notice after each update all processed minidisks and tapes and will no longer request already processed tape volumes.

18.10 Initiate the Recovery Process

Use the following command to display the tape serial numbers which will be required to restore the data belonging to the user IDs specified in the section 18.7, “Create the RECOVER VARS Work File”:

```
RECOVERY ( GETTAPENR
```

This command gives the same result in all VITAL(n) user IDs and may therefore just be issued once. The tapes can be mounted to any recovering user ID in any order.

To start the recovery process issue the following command:

```
RECOVERY ( RECOVER FORMAT
```

Note: It is recommended that the dump tapes be pre-mounted and attached before the RECOVERY command is issued. If this is not done, then a delay may be encountered before the mount requests are issued.
Each user ID will check all basic CMS tape addresses in the following order for pre-mounted tapes:

```
181, 180, 182 to 187, 288 to 28F
```

If the tape unit is found to be of 3480/3490 unit-type, a check loop will be run of about 2 minutes to check for the ACL (Automated Cartridge Loader) feature to automatically mount the next tape without issuing a mount command.

18.11 Verification

Following completion of the restore process, the CMS files “RECOVER VARS” on VITAL 192 disk and the “RECOVERY MSGLOG A” file (on all 191 A-disks for all VITALn user IDs if used) should be scanned for error messages. Restores can be retried for any minidisks which were flagged as incomplete (incomplete means “not C” (completed) and “not S” (to be skipped)).

XEDIT the RECOVER VARS file and use the following to display any errors:

1. ZONE 95 95
2. ALL /C/ & /S/
3. VERIFY 1 16 81 137

The RECOVERY EXEC supplies on-line help on some of the options which have not been described here (enter RECOVERY ?). This help is also shown in “RECOVERY Syntax” on page 196.

18.12 End of Recovery

You have now finished the recovery of the selected minidisks for one system. If there are more systems to recover continue from 18.2, “Logon to the 'VITAL' User ID” on page 202 after printing or saving of needed recovery reference information for this system.

Part 5. 'Planning, Installing and Maintenance'

Chapter 19. Requirements for Installation

19.1 Software Requirements

Requirements to CP

BARS/VM requires VM/ESA Release 1.1 or a later release with support of Access Verification Routines ('ACI') included, especially support of the user-exit to the CP LINK command, module DMKRPI or HCPRPI.

To copy user data from minidisks and to be able to restore data BARS/VM needs the ability to LINK to all minidisks.

There are two methods to make available this LINK authority to BARS/VM:

- First, you may install the RACF/VM Program Product (5767-002 and 5740-XXH), which supplies a version of the DMKRPI/HCPRPI link exit. In this case, the following BARS/VM virtual machines should be given the RACF OPERATIONS user attribute, and they will have the CP LINK authority they require:
  - Interface
  - Backup Controller
  - Query Controller
  - All other subusers.

  Note: If you have BARS/VM running in a RACF environment, take care about synchronization between the VM user directory and the RACF database because of the following reasons:
  - A CP LINK to a MDISK which is not defined in the RACF database will be counted as an invalid LINK by VM.
  - If BARS/VM reaches the LINK limit defined in the SYSJRL macro of DMKSYS or HCPSYS (LNKLMT), it will no longer be able to LINK to (and back up) any other minidisk.
  - Therefore it must be ensured that all minidisks BARS/VM has to back up are defined in the RACF database.
- Second, if RACF/VM is not installed, the BARS/VM virtual machines should have the OPTION LNKNOPAS statement coded in their directory entries.

If tapes are used, manual mounting of tapes for data retrieval can be reduced if the system supports DETACH with the LEAVE option or the CP GIVE command.

DETACH with LEAVE leaves a tape mounted thus allowing BARS/VM to reuse it without it having to be remounted. The GIVE command allows a user to 'give' a dedicated device to another user. It is essentially a general user attach and detach. When the device to be given is a tape, the tape is not unloaded, but left exactly where it is at the time of the GIVE command. BARS/VM tape support does not depend on DETACH LEAVE or GIVE but without either there will be a greater number of tape mounts. This applies not so much to the backup process but more to RETRIEVE, RESTORE and AGET operations. See 4.13, “Use of

Specific Tapes" on page 31 for details of tape usage with and without DETACH LEAVE and GIVE.

Requirements to CMS

The following modifications have been made to CMS modules. They are supplied with the BARS/VM distribution tape.

- The ACCESS module is modified to permit access of mode 0 files in read only mode. This modification is provided via a modified CMS ACCESS module available only to BARS/VM user IDs.
- The DDR module is modified to permit dumping to a CMS file and restoring from such a file.
- The TAPE module is modified to permit the specification of an option to avoid the writing of a tape mark after each file.

19.2 Hardware Requirements

BARS/VM supports any hardware supported by the software required to run BARS/VM.

All existing DASD device types are supported.

19.3 DASD Space Requirements

Unless backup and archive are directly to tape, BARS/VM requires DASD space to contain copies of user data. You should plan and calculate the amount of space required for your needs carefully, corresponding to the rules described in chapter 20.1, "Calculating the Size of BARS/VM Minidisks" on page 211 in this manual.

As a rule of thumb, you will require about half of your currently active ONLINE space to be made available as backup DASD space if all your backup is to DASD. If the tape spill process is used, the installation may choose how much backup will be on DASD and the remaining data will be placed on tape. If there is a small amount of DASD, files will be copied to tape frequently. The larger the amount of DASD defined to BARS/VM, the less frequent will be the demand for tapes mounts.

19.4 SFS Space Requirements

If Shared File System alias and authorization data is to be backed up, space is required in an SFS filepool. It is recommended that a special filepool be set up for this purpose.

See 20.2.5, "SFS Alias and Authorization Backup" on page 225 for details of how to plan this space.

Chapter 20. Planning BARS/VM Installation

20.1 Calculating the Size of BARS/VM Minidisks

20.1.1 Size of Backup and Archive Minidisks

The following discussion is to help you calculate the amount of DASD backup space you will need if you use DASD only. If you use also use tapes, the requirement for DASD is reduced; you may choose the amount of DASD you provide and the remaining data will be placed on tape. If you back up and archive directly to tape, the DASD requirement is zero. The total amount of backup space required in a tape system is the same as in a DASD-only system. Therefore you can calculate the number of tapes required by subtracting the amount of DASD backup space from the total requirement to give the amount of space required on tape. This figure can then be converted to a number of tapes according to the tape devices and densities used.

Backup and archive minidisks are defined by the VM Directory in the BARS/VM NOLOG user ID (see ENUSERID in “General Options” on page 226). These minidisks must have virtual addresses in the range **001 to 5FF**.

The total number of backup cylinders is approximately equal to CMS backup cylinders plus non-CMS backup cylinders plus optional space for CMS archive cylinders.

The formula below evaluates the number of DASD cylinders required to back up CMS files:

$$\text{CMS backup cylinders} = (a \times p + s) \times v \times r$$

```
 where:
 a =  number of CMS primary cylinders to be backed up
 p =  For minidisks, the average ratio of the amount of used space
      to the total amount of space (e.g. 0.6 means minidisks are on
      average 60% full)
 s =  For SFS, the equivalent in cylinders to the amount of file
      data to be backed up
 v =  number of backup versions
      (see ECMSBACK in "General Options" on page 226).
 r =  compression ratio            (about 0.5 usually)
```

Figure 34. Formula for dasd space required to back up CMS minidisks

The formula below evaluates the number of CMS backup cylinders required to save non-CMS minidisks:

```
Non-CMS backup cylinders = a × v × r

 where:
 a =  number of primary cylinders to be backed up
 v =  number of backup versions
      (see EOSBACK in "General Options" on page 226).
 r =  compression ratio             (about 0.3 usually)
```

Figure 35. Formula for dasd space required to back up non-CMS minidisks

Note: The compression ratio of non-CMS minidisk greatly depends on the content of the respective minidisk. It varies from 0.2 for a usual OS/VSE volume up to 0.9 for highly compacted dataset volumes, e.g. SQL database minidisk to be backed up as non-CMS minidisks.

The formula below evaluates the number of 2K CMS blocks required to archive CMS user files:

$$\text{Number of 2K blocks for ARCHIVE} = \frac{a \times u \times v \times r \times p}{100}$$

```
where:
a =  number of 2K blocks allowed for each user
(see EMAXSPAC in "General Options" on page 226)
u =  number of user IDs using ARCHIVE function
v =  number of ARCHIVE versions per file
(see EARLEVEL in "General Options" on page 226)
r =  compression ratio            (about 0.5 usually)
p =  percentage utilization of the allowed EMAXSPAC (= a) per user
```

Figure 36. Formula for dasd space required to archive CMS files

The table below shows the number of 2K blocks per cylinder for various device types.

Device type	2k blocks per cylinder
3390	315
3380	270
3375	204
3350	240
3340	48
3330	114
2314	60
2311	30
2305	168
FB-512	blocks / 4

20.1.2 Planning Considerations for Backup and Archive Minidisks

1. If your online space exists on different types of DASD devices, you should calculate in megabytes or CMS minidisk blocks instead of cylinders because of the difference in cylinder capacities between primary volumes and backup/archive minidisks.
2. The sum of CMS and non-CMS backup space plus the space required for archive evaluated above yields the total amount of space you have to provide.

   Divide this total space into several parts, which should be equal in size, representing the sizes of the CMS minidisks to be defined as backup minidisks.
3. The minimum number of backup minidisks you have to define depends on the number of BARS/VM subusers which will perform backup activities concurrently. The BACKUP and QUERY controller subusers need **no** backup minidisk. So, this number is derived from:

   ```
   rsv = n - 2
   ```

   where **'n'** is the total number of subusers.

   If backup is performed to DASD only and if the fast reorg (EFASTR) installation option is set to NO, then BARS/VM reserves **'rsv'** backup minidisks for REORG purposes. These REORG minidisks are never used in the backup process itself, but are required for the reorganization of backup minidisks. Therefore you have to add **'rsv'** minidisks to the number of backup minidisks you plan to be used for normal backup. In this situation all backup minidisks must be equal in size so, if you plan large contiguous DASD space (e.g. whole real volumes) to reserve as backup and archive volumes, you should split them into some smaller minidisks. This avoids wasting large DASD capacity just for reorganization.
   On the other hand, of course, the size of a backup minidisk limits the size of the largest CMS file a user may want to backup or archive.
4. It is recommended that more backup minidisks are provided than the basic minimum, at least twice the number of subusers. BARS/VM separates the first and second level backup copies of data onto different minidisks. This may not be possible if there are insufficient backup minidisks defined and so a complete backup may not be possible. Also, unless ENOSEPRV is set to YES, the first two backup copies are separated onto different real volumes and this must be taken into account when allocating backup minidisks.
5. Archive minidisks are taken out of the total set of minidisks defined by the NOLOG user ID. No special considerations are necessary for defining archive space.
6. The total number of backup and archive minidisks you have determined in the way described above are to be defined in the directory entry of the NOLOG user ID with addresses of 001-5FF. Chapter 20.2, “Planning BARS/VM Virtual Machines” on page 217 describes the definition of the BARS/VM user IDs in detail.
7. The minimum size of a backup or archive minidisk in 2K blocks, has to be defined in the **ENOBLKBV** option of the installation options (see 20.3.2, “Nodename OPTIONS - Define the Installation Options” on page 226). A minidisk with fewer than this number of blocks will not be used. See the table above for the number of 2K blocks per cylinder.

8. It is strongly recommended that backup minidisks be located on dedicated real volumes to avoid the possibility of backup versions of user data being saved on the same real volume as the original minidisk. If you share backup minidisks with user minidisks on the same real volume or actuator, then loss of the device may mean both the original data and the backup copies may be destroyed and be unrecoverable. Backup minidisks are placed by some installations on highly used DASD such as paging DASD where the space might not otherwise be used.
9. If the tape spill process is to be used for backup, it is recommended that the backup minidisks be large enough to make the process of copying the files to tape as efficient as possible. Each time a backup minidisk becomes full it must be spilled to tape. If the amount of data to be copied is small in relation to the capacity of the tape, it will result in the tape being requested again and again until it becomes full itself. This will result in many more tape mounts if there is no CP GIVE command installed. Much extra time will also be spent scanning the tape to find the resume point. Overall this will result in bottlenecks within the backup controller when it is updating the catalog and control files.

   The optimum size for a backup minidisk where tapes are being used depends on the number of files to be dumped and the capacity of the tape. Experience has shown that a backup minidisk of around 250 to 300 cylinders of 3380 will nearly fill a 3480 cartridge.

20.1.3 Size of BARS/VM Catalog Disks

BARS/VM catalog minidisks are defined in the VM Directory of the BARS/VM Interface user ID (see EIUSERID in “General Options” on page 226). These minidisks must have virtual addresses 191, 192 and 193 for the backup catalog, 197, 198 and 199 for the archive catalog and 2A1 for the DBR catalog. It is **strongly recommended** that these minidisks be formatted with a 4K block size in order that they be eligible for minidisk caching in expanded storage. This will give considerable performance improvements for some of the BARS/VM functions.

Their sizes depend on the number of CMS minidisks to be backed up (for ARCHIVE the number of user IDs using the archive function), the total number of CMS files to be backed up/archived and the number of versions to be held in backup/archive.

In your calculation, **non-CMS** minidisks are to be considered as one CMS file.

The following formulas result in **minimum sizes** of backup and archive catalog minidisks. Reserve some additional space for these minidisks. This is advisable because of the fact that catalog entries pointing to files which have been deleted on the original minidisks, will remain in backup for the number of days specified in the ERASDAYS installation option.

Calculate the minimum number of blocks for the backup catalog minidisks as follows:

$$n + (m + d)\left(\frac{(1 + f)(33 + 27v)}{b}\right) + \frac{208(d + e)}{b}$$

where:
n = total number of non-CMS minidisks to be backed up
m = total number of CMS minidisks to be backed up
d = total number of SFS directories to be backed up
e = total number of enrolled user IDs in SFS filepools to be backed up
f = average number of files to be backed up per CMS minidisk or
    SFS directory
v = number of backup versions per CMS file (ECMSBACK) unless
    this value is 1 in which case set v = 2
b = block size of the catalog minidisk - must be 2048 or 4096 (4096
    is recommended for the best performance)

Figure 37. Calculating the size of the backup catalog

Calculate the minimum number of blocks for the archive catalog minidisks as follows:

$$m\left(\frac{(1 + f)(24 + 48b)}{b}\right)$$

where:
m = total number of user IDs using archive function
f = average number of files to be archived by each user
v = number of versions per file in the archive (EARLEVEL)
b = block size of the catalog minidisk - must be 2048 or 4096 (4096
    is recommended for the best performance)

Figure 38. Calculating the size of the archive catalog

Notes:

1. The results of the formulas above are **minimum** values. They will increase because catalog entries for files which have been deleted from the primary volume will be kept until the number of **ERASDAYS** has been reached.
2. At each backup cycle BARS/VM sends a message to the MANAGER user ID containing information about percentage usage of the backup catalog minidisk. This helps the BARS/VM administrator to control the size of these minidisks and to enlarge them if it becomes necessary.

20.1.4 A Sample DASD Space Calculation

Assume the following sample environment:

```
Total number of users in the directory is  300
Total number of CMS users              is  250
Total number of CMS minidisks          is  400
Total number of non-CMS minidisks      is  100
Average files per CMS minidisk         is  200
Number of backup versions              is    2
CMS user data equivalent of                 10  3390 model 3
Non-CMS user data equivalent of             20  3390 model 3
The average CMS minidisk is 60% full.
No SFS data.
```

Calculate the DASD space required for backup volumes as follows:

```
The number of CMS backup cylinders
 = ((10 × 3339) × 0.6 + 0) × 2 × 0.5

 = 20034 cyl. 3390 (equivalent to 6 volumes of 3390 model 3)


The number of non-CMS backup cylinders
 = (20 × 3339) × 2 × 0.3

 = 40068 cyl. 3390 (equivalent to 12 volumes of 3390 model 3)
```

Figure 39. Sample calculation of backup space

You have to reserve an equivalent of 17 * 3390 model 3 as DASD space for backup.

Calculate the DASD space required for archive volumes as follows, assuming

- The value for EMAXSPAC is 3150 2K blocks (approx. 10 Cyl. 3390)
- The number of version per file to be held in archive is 2
- The average utilization of archive space is 60 percent.

```
Number of 2K blocks
 = 3150 × 250 × 2 × 0.5 × 60/100

 = 472500 2K blocks

 = 1500 cylinders of 3390
```

Figure 40. Sample calculation of archive minidisk size

You have to reserve an equivalent of 1500 cylinders of 3390 for archive.

Calculate the DASD space required for the backup catalog as follows:

The number of 4K blocks for the backup catalog:

$$100 + (400 + 0)\left(\frac{(1 + 200)(33 + 27 \times 2)}{4096}\right) + \frac{208(0 + 0)}{4096}$$
$$-100 + 400 \times \left(\frac{201 \times 87}{4096}\right)$$
$$-1808$$

which is equivalent to 10 cylinders of 3390

Figure 41. Sample calculation of backup catalog minidisk size

You have to define minidisks 191 - 193 of BARS/VM Interface user ID with 14 cylinders of 3390 each.

Calculate the DASD space required for the archive catalog as follows, assuming

- The average number of files each user will archive: 50
- The number of versions per file to be hold in archive: 2

The number of 4K blocks for the archive catalog:

$$300 \times \left(\frac{(1 + 50) \times (24 + 48 \times 2)}{4096}\right)$$
$$-300 \times (51 \times \frac{120}{4096})$$
$$-600$$

which is equivalent to 4 cylinders of 3390

Figure 42. Sample calculation of archive catalog minidisk size

You have to define minidisks 197 - 199 of BARS/VM Interface user ID with 4 cylinders 3390 each.

20.2 Planning BARS/VM Virtual Machines

All values for minidisk cylinders are based on device type 3390. Recompute them if you have different DASD device types.

The sample directory entries on the distribution tape, which are shown later in this section, specify user ID names for ease of reference. If the installation wishes to choose different names for any of the user IDs comprising the BARS/VM virtual machines, they may do so.

Caution:

It is important that you do not define or link any extra minidisks in the directory, SYSPROF or PROFILE EXECs of the BARS/VM user IDs. Doing so could cause conflict with virtual addresses and filemodes which BARS/VM expects to be free and available for its own use. This will also avoid the chance of the virtual machine environment being modified before the BARS/VM code is started.

20.2.1 BARS/VM Manager User ID - EMUSERID

This user ID receives and logs all messages from the other BARS/VM user IDs. It also has the highest level of BARS/VM privilege, and should be used to authorize new BARS/VM administrators.

It is strongly recommended that a dedicated virtual machine be set up to act as the manager user ID, as this will ensure that all messages are logged. However, this is not essential, and any user ID on the same system may be specified as the manager ID.

If a specific virtual machine is created to act as the manager, then there should be a dedicated manager user ID for each BARS/VM subsystem —i.e. don't specify the same manager user ID for more than one BARS/VM subsystem on any given VM system.

The code supplied to run in the manager user ID uses the Programmable Operator (PROP) facility of VM to automate the logging of all messages. It also provides action routines to perform various tasks useful to BARS/VM administrators. The REFRESH routine requires authority to link the Interface 194 and 19F minidisks R/W (LINK mode M). See 11.2, “Command Syntax Description” on page 147 for details of these routines. This code may require modification by the installation to meet local requirements, and if desired can be completely replaced by locally written routines, since the operation of the other BARS/VM service machines has no reliance on the manager user ID.

The following table shows the minidisks required by the manager user ID. The distributed code disk is used to hold all the BARS/VM files as shipped. It is only used by the manager userID.

Figure 43. Minidisks of BARS/VM Manager user ID

CUU	Purpose	Size
191	Log and work disk	25 cyls
194	Manager code disk	2 cyls
291	BARS/VM distributed code disk	30 cyls
Note: The 191 minidisk may be replaced by a SFS filespace as long as the filepool server is always available before the manager user ID starts.		
Note: The 291 minidisk may be replaced by a SFS filespace. Use of a remote filepool server would avoid the need for duplicate 291 minidisks on each system, together with the work involved in updating them.		

Figure 44 shows a sample directory entry for the manager user ID.

```
USER BARMANGR password 4M 6M G
ACCOUNT 1234 VMBARS
IPL CMS
CONSOLE 009 3215
SPOOL 00C 2540 READER A
SPOOL 00D 2540 PUNCH A
SPOOL 00E 3211 A
LINK MAINT 190 190 RR
LINK MAINT 19E 19E RR
* Log files disk
MDISK 191 3390 cyl 025 volser MR
* Manager code disk
MDISK 194 3390 cyl 002 volser RR
* BARS/VM distributed code disk
MDISK 291 3390 cyl 030 volser MR
```

Figure 44. Sample Directory Entry for the BARS/VM Manager User ID

20.2.2 BARS/VM Interface User ID - EIUSERID

Controls the work and communication within BARS/VM.

The INTERFACE user ID must be defined with privilege classes A, B and G. It requires these privileges to issue the CP FORCE, MSGNOH, AUTOLOG, and XAUTOLOG commands and, if tapes are used, the GIVE and ATTACH commands. You may wish to use UCR to provide these authorities via a new class. This virtual machine must be run in XA mode.

The virtual storage requirements will largely depend on your local environment. The major factors involved will be first the size of your backup minidisks (and hence the numbers of files they can contain) and secondly any user minidisks with a large number of CMS files. The values suggested in the sample directory entries should be regarded as minimum values.

The following minidisks must be defined to the INTERFACE user ID:

Figure 45. Minidisks of BARS/VM Interface user ID

CUU	Purpose	Size
191	Backup catalogues and control files	n cyls
192	Backup of 191	n cyls
193	Backup of 191	n cyls
194	Code disk - subuser's 191 minidisk	8 cyls
195	Unpack and work minidisk	see notes
1A5-1FF	Optional additional unpack and work minidisks	see notes
196	Command queueing minidisk	2 cyls
197	Archive catalogues	m cyls
198	Backup of 197	m cyls
199	Backup of 197	m cyls
19C	Checkpoint information	10 cyls
19F	Panel interface code minidisk	2 cyls
2A1	DBR catalogues and control files	2n cyls
2A2	Optional minidisk for DBR LSET reports	5 cyls

Notes:

1. Minidisks 191, 192 and 193 must be equal in size. Chapter 20.1.3, “Size of BARS/VM Catalog Disks” on page 214 assists you in determining the amount of DASD cylinders 'n' for these minidisks.
2. It is **strongly recommended** that the 191, 192 and 193 minidisks are placed on separate real volumes to avoid the possibility of data loss in the event of DASD problems. This includes placing the 191 on a separate HDA from the 192 and 193.
3. Minidisks 197, 198 and 199 must be equal in size. Chapter 20.1.3, “Size of BARS/VM Catalog Disks” on page 214 assists you in determining the number of DASD cylinders 'm' for these minidisks.
4. It is **strongly recommended** that the 197, 198 and 199 minidisks are placed on separate real volumes to avoid the possibility of data loss in the event of DASD problems. This includes placing the 191 on a separate HDA from the 192 and 193.
5. Minidisk 195 is used to unpack (opposite of the pack or compression technique during backup process) the data when a CMS RETRIEVE is processed. Its size limits the size of the largest CMS file a user can RETRIEVE, so it must be large enough to hold the largest CMS file in your backup and archive system. This minidisk must be formatted with a block size of 2K.

   Note: This limitation does not apply to the RESTORE process.
6. One or more of the minidisks 1A5 through 1FF may optionally be defined. If this is done, a subuser requiring the 195 minidisk and finding it unavailable will use one of these disks instead. This therefore can give performance benefits in certain situations where the 195 is heavily used. The comments above about the 195 minidisk also apply to these minidisks. Note that these minidisks, if defined, should be the same size as 195. If this is not possible

because of different device types, they should be larger than the 195 but as close as possible in the number of blocks they contain. These minidisks must be formatted with a block size of 2K.

7. Minidisk 19C is used to hold checkpoint restart information used by Primary Volume and Storage Group restore processing.
8. Minidisk 19F is used to hold the code for the ISPF panel interface. It must be able to be linked by all users - i.e. have a UACC of READ.
9. Minidisk 2A1 is needed only if the BARS/VM disaster backup function is to be used. The size of this minidisk should be double the size of minidisk 191, since the control information is stored in unpacked format.
10. Minidisk 2A2 is needed only if the BARS/VM disaster backup function is to be used. It is optional. Non-truncated LSET files are placed onto this minidisk, which also receives LSET based report files at the end of an LSET dump.
11. It is **strongly recommended** that all catalogue minidisks (191, 192, 193, 197, 198, 199 and 2A1), the checkpoint minidisk (19C) and the panel interface minidisk (19F) be formatted with a 4K block size so that they are eligible for minidisk caching in expanded storage. This is not essential on VM/ESA 1.2.2 and later systems due to the extension of MDC to non 4K blocked minidisks.
12. The 193 and 199 minidisks are only written to as backups, and hence may be excluded from caching (i.e. MINIOPT NOCACHE NOMDC).

Figure 46 on page 222 shows a sample directory entry for the interface user ID.

```
USER BARINTFC password 6M 6M ABG
ACCOUNT 1234 VMBARS
IPL CMS
MACHINE XA
CONSOLE 009 3215
SPOOL 00C 2540 READER A
SPOOL 00D 2540 PUNCH A
SPOOL 00E 3211 A
LINK MAINT 190 190 RR
LINK MAINT 19E 19E RR
* Backup catalogs and control files
MDISK 191 3390 cyl 100 volsr1 MR
MDISK 192 3390 cyl 100 volsr2 MR
MDISK 193 3390 cyl 100 volsr3 MR
MINIOPT NOCACHE NOMDC
* TEXT decks
MDISK 194 3390 cyl 008 volser RR
* Pack/Unpack disk
MDISK 195 3390 cyl 100 volser MR
* Command queueing disk
MDISK 196 3390 cyl 002 volser MR
* Archive catalogs
MDISK 197 3390 cyl 025 volsr1 MR
MDISK 198 3390 cyl 025 volsr2 MR
MDISK 199 3390 cyl 025 volsr3 MR
MINIOPT NOCACHE NOMDC
* Checkpoint restart disk
MDISK 19C 3390 cyl 010 volser MR
* Panel interface code disk
MDISK 19F 3390 cyl 002 volser RR
* Disaster catalogs and control files
MDISK 2A1 3390 cyl 200 volser MR
* Disaster backup report files
MDISK 2A2 3390 cyl 005 volser MR
```

Figure 46. Sample Directory Entry for the BARS/VM Interface User ID

20.2.3 Subusers

A minimum of **three** subusers must be defined for BARS/VM:

- One working as the BACKUP controller
- One working as the QUERY controller
- And a minimum of one additional subuser to do the real work.

All of these virtual machines must run in XA mode.

The user IDs must be authorised to link any minidisk on the system, and to access any SFS data. If an ESM is not being used, then the LNKNOPAS directory option and SFS ADMIN authority are required (if SFS backup is enabled). If an ESM is being used, then the appropriate equivalent ESM authority must be granted.

Backup Controller - B.C.

This decides what backup and archive work must be done. It is the first subuser ID you define with the SUBUSERS command as described in “Step 8: Initialize BARS/VM the First Time” on page 258.

The virtual storage requirements will largely depend on your local environment. The major factors involved will be first the size of your backup minidisks (and hence the numbers of files they can contain) and secondly any user minidisks with a large number of CMS files. The values suggested in the sample directory entries should be regarded as minimum values.

See below for a sample directory entry of this user ID.

Query Controller - Q.C.

This is responsible for handling commands of a query nature, e.g. QUERY, AQUERY, DQUERY. It is the second subuser ID you define with the SUBUSERS command as described in “Step 8: Initialize BARS/VM the First Time” on page 258.

This user ID will use the MSGNOH command to respond to user queries if available, and hence provide a more readable display. Therefore you may wish to either give it class B, or use UCR to redefine MSGNOH to be available as part of another privilege class which this ID can have.

If the value of ESECDATA is such that an ESM is being used, then this user ID will need any CP privilege class of A-F besides G to allow it to use the RACROUTE[2] macro.

See below for a sample directory entry of this user ID.

General

These user IDs do the work assigned by Interface and B.C. with the resources allocated by resource control in the Interface machine. The number of subusers is installation defined. The user IDs of the subusers are defined with the SUBUSERS command described in 10.2.37, “SUBUSERS” on page 138 and in chapter “Step 8: Initialize BARS/VM the First Time” on page 258.

In addition to privilege class G, the subuser machines need the appropriate privilege class if tapes are used and the commands GIVE, ATTACH, and DETACH with the LEAVE option are to be used as described in 4.13, “Use of Specific Tapes” on page 31. These are normally class B. Class H is also required if the update to DMKRPI/HCPRPI is used to allow user IDs with this class to link to any user minidisk.

In the VM directory, each subuser's 191 is defined as INTERFACE's 194 (LINK 'EIUSERID' 194 191 RR). All other minidisks are LINKed and ACCESSed.

These user IDs will use the MSGNOH command to send messages to users if available, and hence provide a more readable display. Therefore you may wish to either give them class B, or use UCR to redefine MSGNOH to be available as part of another privilege class which these IDs can have.

---

[2] ESM checking is performed via the limited-function third-party REQUEST=AUTH function of the RACROUTE macro. See the section entitled “RACROUTE: VM Router Support” in “Appendix A. Independent RACF System Macros” of the *External Security Interface (RACROUTE) Macro Reference for MVS and VM - GC28-1366* manual for further details.

If the value of ESECDATA is such that an ESM is being used, then the subusers will need any CP privilege class of A-F in addition to G to allow it to use the RACROUTE[2] macro.

The virtual storage requirements will largely depend on your local environment. The major factors involved will be first the size of your backup minidisks (and hence the numbers of files they can contain) and secondly any user minidisks with a large number of CMS files. The values suggested in the sample directory entries should be regarded as minimum values. Figure 47 shows a sample directory entry for the subuser user ID.

```
USER BARSUSER password 8M 8M BG
ACCOUNT 1234 VMBARS
IPL CMS
MACHINE XA
OPTION LNKNOPAS
CONSOLE 009 3215
SPOOL 00C 2540 READER A
SPOOL 00D 2540 PUNCH A
SPOOL 00E 3211 A
LINK MAINT 190 190 RR
LINK MAINT 19E 19E RR
LINK BARINTFC 194 191 RR
```

Figure 47. Sample Directory Entry for the BARS/VM Subuser User ID

20.2.4 BARS/VM NOLOG User ID - ENUSERID

This user ID exists merely to define the BARS/VM backup and archive minidisks as described in 20.1, “Calculating the Size of BARS/VM Minidisks” on page 211. All minidisks defined in its directory are allocated as backup or archive minidisks. No other devices need be defined as this user ID should never be logged on. If backup is directly to tape and archive is either directly to tape or not used, there is no need to define this user ID.

Minidisks **001-5FF** identify the backup and archive minidisks (maximum 1535). Figure 48 shows a sample directory entry for the NOLOG user ID.

```
USER BARNOLOG NOLOG
ACCOUNT 1234 VMBARS
* Backup minidisks
MDISK 001 3390 cyl 158 volser RR
MDISK 002 3390 cyl 158 volser RR
MDISK 003 3390 cyl 158 volser RR
MDISK 004 3390 cyl 158 volser RR
MDISK 005 3390 cyl 158 volser RR
 ...  ..   ... ... ... ...... ..
 ...  ..   ... ... ... ...... ..
MDISK 009 3390 cyl 158 volser RR
MDISK 00A 3390 cyl 158 volser RR
 ...  ..   ... ... ... ...... ..
 ...  ..   ... ... ... ...... ..
MDISK 5FF 3390 cyl 158 volser RR
```

Figure 48. Sample Directory Entry for the BARS/VM NOLOG User ID

To add new backup and/or archive minidisks it is only necessary to add the new minidisk definitions to the directory entry of the NOLOG userID. These new minidisk definitions are picked up at the start of operations such as incremental backup and deferred command processing. The new minidisks are added initially as 'unformatted' and subsequently assigned as archive and backup minidisks as needed and according to the EMAXCMSV and EMAXARC installation options.

If it has been specified that BARS/VM get its directory from the USER BACKUP file of DIRMAINT, a DIRMAINT USER BACKUP command should be issued after defining the new minidisks to ensure that the file is updated with the new information.

20.2.5 SFS Alias and Authorization Backup

SFS aliases and authorizations are backed up to an SFS file space. If SFS backup is required, the name of the file pool must be defined by the EBFPOOL installation parameter. The name of the file space used is BARFS. This user ID must be enrolled in the file pool with sufficient space for alias and authorization backup. It is not necessary to define a matching BARFS VM userID, but it is recommended in order to prevent such a userID being allocated for other purposes in the future, and to aid in identifying the owner of the BARFS filespace. All BARS/VM subusers must be given administrator authority to this filespace.

It is recommended that this file space be in a file pool separate from any which will contain data to be backed up. This will avoid the situation where the backup data is held in the same file pool as the data which is being backed up, resulting in a “catch-22” position if the file pool should fail and need restoring. It will also help to reduce possible contention in the backup file pool.

For each alias backed up, a number of bytes is required in the file pool equal to the size of the record returned by the Get Directory CSL routine for ALIAS. This length is defined by the value DIRALEN in the CMS DIRBUFF macro and has a value of 207 for ESA Release 1.1 and 2.

For each authorization backed up, a number of bytes is required in the file pool equal to the size of the record returned by the Get Directory CSL routine for AUTH. This length is defined by the value DIRULEN in the CMS DIRBUFF macro and has a value of 40 for ESA Release 1.1 and 2.

20.3 Planning BARS/VM Installation Options

20.3.1 BARS/VM PROFILE - Define the Interface User IDs to the BARS/VM Module

The module which is used to communicate with BARS/VM must know the names of the Interface virtual machines at nodes to which commands can be sent. This is done through a file with a file type of PROFILE and a file name equal to the file name of the BARS/VM module, normally VMBARS. This file must be initialized with a record for each BARS/VM system the current system will communicate with. The file must be fixed format with a record length of 16 and each record must consist of an 8 byte node name followed by an 8 byte Interface user ID. It must reside on a public minidisk with the BARS/VM module.

20.3.2 Nodename OPTIONS - Define the Installation Options

The installation may set the following variables from defaults provided in the module HDFEQU. The installation options are specified in a file whose file type is OPTIONS and whose file name is the node name of the system on which BARS/VM is running. This file must be on a minidisk accessible to all BARS/VM virtual machines, e.g.. the 194 minidisk of the Interface virtual machine. Below are listed the options which may be specified.

Options to Define User IDs

EIUSERID The INTERFACE user ID, from 1 to 8 characters.

EMUSERID The BARS/VM manager user ID, from 1 to 8 characters. This user ID is authorized to issue all BARS/VM commands and also to authorize other user IDs.

ENUSERID The NOLOG user ID, from 1 to 8 characters.

EOUSERID The operator user ID, from 1 to 8 characters. This user ID is authorized to issue the BARS/VM REPLY command to reply to date verification requests at startup time.

General Options

EARONLY This must be specified as YES or NO. If YES is specified, the automatic backup function will only be used to back up archive minidisks. Backup of other minidisks and SFS directories will not be performed. The default is NO.

EAUTOSPL This must be specified as YES or NO. If YES is specified, then automatic spill to tape will occur during the backup process. This means that when backup minidisks become full during a backup cycle, a mount request will be generated to empty the minidisk. If NO is specified, then no mount requests will be issued during backup, and the spill to tape may only be performed via the START TAPEDUMP command. If the option is not specified, then YES is the default.

EBAONLY This must be specified as YES or NO. If YES is specified, the file and minidisk archive functions of BARS/VM will not be used on this system. If the parameters ERARCID and ERARCIU are set archive commands will be routed to the specified user id and system. The default is NO.

EDATEC This must be specified as a single character. It is used as a delimiter for dates to separate the day, month and year characters. The default is a / character.

EDATEF This specifies the date format for reports and messages. It is a single character which may be one of the following:

```
  E = European      dd/mm/yy      e.g.   22/08/95
  U = USA           mm/dd/yy      e.g.   08/22/95
  I = ISO date    yyyy/mm/dd      e.g. 1995/08/22
  O = Ordered       yy/mm/dd      e.g.   95/08/22
 '' = Normal         ddmmmyy      e.g.    22Aug95
```

EFASTR This must be specified as YES or NO. If YES is specified, this BARS/VM system is to use the "fast reorganization" code for backup and archive minidisks. In this case

redundant files are erased from the backup or archive minidisk instead of the required files being copied to a new minidisk. The default is NO.

EFASTRT This must be specified as YES or NO. If YES is specified, this BARS/VM system is to use the "fast reorganization" code for backup tape reorganizations. The default is NO.

It is very important that you read the reorganization section of Chapter 12, "BARS/VM Administrator's Tasks and Controls" on page 151 before setting this option to YES.

EMACHID Specifies the node identification of the system, from 1 to 8 characters. This is used when sending commands to other nodes so that responses can be routed back to the issuing node. It is also used on some BARS/VM output to identify the system on which it was generated.

ENAME Specifies the name by which BARS/VM is to be invoked. The default is VMBARS. This name appears in messages such as message 0122 requesting a command or a reply. The option allows those installations which may have more than one VMBARS on the same system to identify the one requiring the action.

ENETID Specifies the user ID of a virtual machine to be used to send messages and files to remote systems. It must be from 1 to 8 characters.

ENOBLKBV The number of 2K blocks per backup and archive minidisk. It must be a number between 1 and 99999999.

See 20.1, "Calculating the Size of BARS/VM Minidisks" on page 211 for a discussion of how to calculate this value.

ENOFARC This must be specified as YES or NO. If YES is specified, the file archive functions of BARS/VM will be not be used but minidisk archive is allowed. This option need not be set if EBAONLY YES is specified. If file archive is disallowed by this parameter but ERARCID ERARCIU are set, archive commands are routed to the specified user id and system. The default is NO.

ERESTART Specifies the maximum number of subuser restarts after program checks. It must be a number between 0 and 32767. This parameter prevents repetitive program checks when there is a serious problem with BARS/VM or the system. Each time a subuser ends with a program check a counter is incremented and if the value of the counter is not greater than ERESTART the subuser will be autologged again. The count is a total for all subusers - i.e. a separate count is not kept for each subuser. See the `DISPLAY STATUS RESTART` command for information on displaying this count.

ERFORCE This must be specified as YES or NO. If YES is specified, backup reorganizations are forced to complete even if errors result in files being lost. If backup files are lost, the original files are flagged in the catalogs so that they are backed up again at the next cycle. The option is ignored for archive reorganizations. The FORCE option on the START REORG command may be used to force the completion of specified reorganizations if ERFORCE is set to NO. The default is NO.

ESECDATA This parameter defines how authorization checking is done when one general user attempts to act on behalf of another. The default value is OWNED which means the authorization is based solely on the use of the OWNED command. The following table list the possible values for this option.

Value	Meaning
OWNED	Authorization checking is based solely on the OWNED command
ESM0	Equivalent to `OWNED`
ESM1	Call an External Security Manager (such as RACF) to check authority for all minidisk backup commands plus the DBR QUERY command (see 8.4, “Authorizing other users” on page 85 for more details). This checking is ***in addition*** to the OWNED check, and is only done if authority was not obtained via OWNED checking. It is also done after a call to HDFXGV, and then only if the return code from that module indicates that the authorization check is to be done by the External Security Manager.

ESLAVE This must be specified as YES or NO. If YES is specified, this BARS/VM system only has a read-only access to its minidisks. Commands which can be processed using read-only access are processed by this system and other commands are sent to the host system defined in EHOSTID. The default is NO.

See 5.1, “BARS/VM in a Shared Data Environment” for a discussion of sharing BARS/VM minidisks.

ESSI This must be specified as YES or NO. If YES is specified, this BARS/VM system is a shared system in an SSI environment. Commands are automatically routed to the correct system where they will be processed. The default is NO.

Backup Options

EALIAS Controls the backup of aliases. It must be specified as YES or NO. If YES is specified alias information is backed up. The file pool used for the backup must be specified in EBFPOOL. The default is YES.

EAUTH Controls the backup of authorizations. It must be specified as YES or NO. If YES is specified authorization information is backed up. The file pool used for the backup must be specified in EBFPOOL. The default is YES.

EBACKE The number of days to retain the backup data after a BACKUP ... ERASE command has been issued. It must be a number between 0 and 32767.

EBACKMIG Specifies whether migrated SFS files are backed up or not. It must be either YES or NO. The default is NO, in which case migrated files are ignored for backup purposes. If YES is specified, the HDFXMG user exit must be set up correctly to initialize the Callable Services Library interface for handling the backup and restore of migrated files.

EBACKML2 Specifies whether SFS files which are migrated to migration level 2 are backed up or not. It must be either YES or NO. The default is NO. If EBACKMIG is set to YES and EBACKML2 to NO, only migration level 1 is backed up. **Warning:** The backup of migration level 2 has not been subjected to any formal test and results cannot therefore be guaranteed.

EBDAY Allows the specification of which days automatic incremental backup will take place. The parameter is optional. If specified, it must consist of seven characters with the values Y or N. Each character corresponds to a day of the week, the first representing Monday. A value of Y causes backup to take place that day, and N suppresses backup. The default is YYYYYYY. This parameter does not affect backup run as a result of the START command.

As an example, a value of YYYYYNN will prevent backup from running on Saturdays and Sundays. The values of EBDAY do not affect the running of reorganizations; these are controlled by a separate parameter, ERDAY.

EBFPOOL Specifies the name of the file pool to be used for the backup of SFS alias and authorization data.

EBRSBLK Is a number in tenths of one percent. During backup the total number of blocks used on each backup volume is computed for each real volume and storage group (entity). The number of backup volumes where this value is not zero equals the number of backup volumes required to restore the entity. N = (number of blocks) * EBRSBLK/1000 is calculated for each such entity. Then, to reduce the number of backup volumes, all files backed up to volumes containing N blocks or less are scheduled for backup again. **Because of the effect the option may have on the backup process it is important to read the discussion in 12.11, “Improving RESTORE Performance” on page 167 on reducing restore time and on how to use the option.**

EBVFULL Is a number of days between 0 and 32767. If it is non-zero, a backup minidisk or tape which is not yet full, will be marked full after this period has elapsed since it was last reorganized. If it is zero, the default, backup volumes are only marked full when they really do become full. This allows an installation to ensure that

| | minidisks and tapes are reorganized regularly, since they are only eligible for reorganization when full. It is recommended that the value is set to zero initially. It can then be changed, if necessary, at a later date when the pattern of use of backup volumes becomes clear.

EBZODAYS Allows the erasure of levels of backup higher than level zero after a specified number of days. It must be a number between 0 and 32767. If the parameter is non-zero, any CMS file whose level zero backup was created more than the specified number of days ago, has any levels higher than zero erased. This allows an installation to keep at least two levels of backup for files being actively changed (or just created and perhaps to be changed soon), but not keep more than one backup level of files where the last backup date is very old, indicating that the source file hasn't changed in a long time. This allows the installation to decrease DASD or tape usage while maintaining an acceptable capability to restore lost files.

ECMSBACK The number of backup copies to be kept for each CMS file. This must be a number between 1 and 9.

ECMSDAYS The default backup interval for a CMS minidisk or SFS directory. This may be changed via the DAYS parameter of the BACKUP command. It must be a number between 1 and 99.

EDIRPROC Specifies whether BARS/VM is to process the CP directory each night before backup. If the value is YES the directory will always be processed. If the value is NO the directory is not processed unless a DIRECT command has been issued since the last time backup was done. The default is NO.

EMAXBD Is used to limit the number of backup volumes (DASD minidisks or tapes) used to back up a particular minidisk. It must be a number between 0 and 32767. If the parameter is zero, no limit is applied. If nonzero, the system will attempt to keep the number of backup volumes at or below this number. It is envisaged that this parameter will be used when the backup medium involves operator intervention, e.g. tapes. In this situation, it is unacceptable for operators to have to mount a large number of volumes when a restore is to be done. It should be noted however that the use of this parameter gives improved restore and retrieve performance at the expense of backup performance and therefore care must be taken in choosing a value. For a backup medium involving no operator intervention a large value, or zero, should be specified. To limit the number of backup volumes, some files are backed up even when unchanged. If this happens, only the level 0 is updated and higher levels remain unaltered. For DBR, practical experience has shown that a value of 10 is a good trade off between DBR dump performance and

the additional backup that has to be taken to limit the number of backup minidisks per minidisk backed up.

EMAXCMSV Maximum number of CMS backup minidisks which will be used. It must be a number between 0 and 32767.

EMAXCYL The maximum number of cylinders that may be retrieved in a single RETRIEVE command for a non-CMS minidisk that is not deferred. It must be a number between 0 and 32767.

EMAXFILE The maximum number of CMS files that may be retrieved in a single RETRIEVE command that is not deferred. It must be a number between 0 and 32767.

EMAXPCE Is used to limit the size of files produced by the backup of non-CMS minidisks. The parameter is normally used with tape backup to avoid problems where large files might not fit on the tapes. It is also used for remote backup where the files are to be sent to a host system when it is undesirable to have very large files. It must be a number between 0 and 99999999. If the parameter is zero, no limit is applied. If nonzero it specifies the maximum size, in 2K blocks, for a CMS file produced by non-CMS backup.

EMINCMS The minimum number of days (1-99) that may be specified in the BACKUP command for a CMS minidisk or SFS directory. i.e. the minimum backup frequency.

EMINIDAY The number of days to retain the backup data for a minidisk after the minidisk has been deleted from the VM directory or to retain for an SFS directory after the directory has been erased. It must be a number between 0 and 32767.

EMINOS The minimum number of days (1-99) that may be specified in the BACKUP command for a non-CMS minidisk, i.e the minimum backup frequency.

ENOSEPRV This must be specified as YES or NO. If YES is specified, this BARS/VM system is not to perform real-volume separation of backup data. If it is set to NO, BARS/VM ensures that archive and backup data reside on separate real volumes and also that the level 0 and level 1 backup for each file are on separate real volumes. The default is NO.

ENSUBUSE The number of subusers, apart from the Query Controller, to be kept logged on when not carrying out the backup or restore process. It must be a number between 1 and 32767.

EOSBACK The number of backup copies of non-CMS minidisks to be kept. It must be a number between 1 and 9.

ERASDAYS The number of days to retain a CMS file after the owner of file has erased it and the number of days after the erasure of an archived file that it will be available for recovery through the DRECOVER command. It must be a number between 0 and 32767.

ERDAY Allows the specification of which days automatic reorganizations will take place. The parameter is optional. If specified, it must consist of seven characters with the values Y or N. Each character corresponds to a day of the week, the first representing Monday. A value of Y allows reorganizations to take place that day, and N suppresses them. The default is YYYYYYY. This parameter does not affect reorganizations run as a result of the START REORG command.

The day is taken from the day at the start of the backup cycle. So if a backup cycle crosses midnight, reorganizations running after midnight are controlled by the setting for the previous day. As an example, suppose the backup window is 2000 to 0600 and backup starts on Tuesday for which the value of ERDAY is Y. Reorganizations will run in that cycle on Wednesday morning even if the value of ERDAY for Wednesday is N because the value at the start of the cycle was Y.

EREORG Number of days between backup reorganizations. It must be a number between 1 and 32767.

See 12.1, “Reorganization” on page 151 for a discussion and some special considerations useful to determine this value.

EXDAYS Allows the erasure of backup data after a specified number of days. It must be a number between 0 and 32767.

If this parameter is non-zero, CMS files with a time-stamp older than this value will not receive backup. Also, existing backup data for such files is erased from backup, even if the original files still exist. This allows an installation that takes regular full dumps to minimize the amount of BARS/VM backup data.

Archive Options

EARCLASS Specifies the default class for archived files. It must be a single character.

EAREORG Specifies the number of days between archive reorganizations. It must be a number between 1 and 32767.

EARFORCE This must be specified as YES or NO. If YES is specified, archive reorganizations are forced to complete even if errors result in files being lost. The option is ignored for backup reorganizations. The FORCE option on the START REORG command may be used to force the completion of specified reorganizations if EARFORCE is set to NO. The default is NO.

EARLEVEL Specifies the default number of levels allowed for archived files. It must be a number from 1 to 40.

EARLINK Determines how an APUT command from a remote system is processed. If NO is specified and a cuu is specified on the command, a normal LINK command is

used to link to the specified minidisk. If YES is specified a 'remote LINK' is used to attempt to link to the minidisk on the system from which the command was sent. This 'remote LINK' is done by issuing a CP command of the form 'REMOTE LINK system userID cuu1 cuu2'. The operands are as for a normal LINK command except for the addition of the keyword REMOTE and the remote system name. The default is NO.

EARSPILL Must be specified as YES or NO. If YES, archive data is spilled to tape when a START TAPEDUMP command is received. The default is NO.

EARSUB Specifies the number of subusers to be used for deferred archive processing. It must be a number between 0 and 32767.

EARTWIN Must be specified as YES or NO. If YES, each archive tape is assigned a twin which may be used for recovery should the archive tape be damaged. See 4.12, “Archive Twin Tapes” on page 30 for a discussion of twin tapes. The default is NO.

EATCOMP Must be specified as YES or NO. YES specifies that compression is to be used when archiving directly to tape. If EBTXF is set to YES hardware compression is used, otherwise software is used. The default is NO.

See 4.11, “Archive Directly to Tape” on page 30 for information on the use of this option.

EATONLY Must be specified as YES or NO. YES specifies that archive is to go directly to tape. The default is NO.

EAVFULL Is a number of days between 0 and 32767. If it is non-zero, an archive minidisk or tape which is not yet full, will be marked full after this period has elapsed since it was last reorganized. If it is zero, the default, archive volumes are only marked full when they really do become full. This allows an installation to ensure that minidisks and tapes are reorganized regularly, since they are only eligible for reorganization when full. It is recommended that the value is set to zero initially. It can then be changed, if necessary, at a later date when the pattern of use of archive volumes becomes clear.

EDCENT Specifies the current century - two characters, e.g. 19.

EMAXARC Specifies the maximum number of archive minidisks that BARS/VM is to use. It must be a number between 0 and 32767.

EMAXGET Specifies the maximum number of files any user is allowed to retrieve from archive immediately, i.e. using the NOW option of the AGET-command. It must be a number between 0 and 32767.

EMAXPUT Specifies the maximum number of files any user is allowed to send to archive immediately, i.e. using the NOW option of the APUT-command. It must be a number between 0 and 32767.

EMAXRET Specifies, in months, the system default maximum retention period allowed to be specified for archived files. It must be a number from 1 to 999.

EMAXSPAC Specifies, in 2K blocks, the default maximum amount of archive space per user ID. It must be a number between 1 and 999999.

ERARCID Defines the node identification of a system to which archive commands are to be sent if archive is disallowed on the current system. If this parameter is not set and archive commands have been disallowed (by EBAONLY or ENOFARC) any such command will be rejected.

ERARCIU Defines the user identification of a BARS/VM Interface virtual machine to which archive commands are to be sent if archive commands are not allowed on the current system and ERARCID is set.

ERETAIN Specifies, in months, the default retention period for archived data. It must be a number between 1 and 999.

Remote Backup Specific Options

EHDAYS Specifies a number of days after which a file is eligible for sending to a remote host for backup. It must be a number between 0 and 32767. A file is not sent until it has been backed up for this number of days. It also specifies the number of days after reorganization that a backup minidisk should be set as full, even if not so, and thus made available for reorganization again. If remote backup is not to be performed this field should be set to zero. See 5.3.1, “Definitions” on page 38 for a discussion of remote backup.

EHOSTID Defines the node identification of a host system to which backed up files are to be sent if remote backup is to be performed or, in an environment where the BARS/VM minidisks are shared, it defines the host system which has read-write access to these minidisks. If these facilities are not to be used, this options should be set to blanks. See 5.3.1, “Definitions” on page 38 for a discussion of remote backup and 5.1, “BARS/VM in a Shared Data Environment” on page 37 for a shared environment.

EHOSTIU Defines the user identification of the BARS/VM Interface virtual machine on the remote host. If remote backup and shared minidisks are not used this option should be set to blanks.

ESENDMAX Specifies, for remote backup, the number of files that each subuser may send to the host before receiving acknowledgment that remote backup has been

successful. It must be a number between 0 and 32767. This is to prevent an excessive number of files being sent over a network and to avoid filling spool space at the host or intermediate nodes when there is a problem with the host BARS/VM If remote backup is not to be performed this value should be set to zero.

Tape Handling Options

EBTONLY Must be specified as YES or NO. YES specifies that backup is to go directly to tape. The default is NO.

EBTPOOL Defines the name of the backup tape pool if backup is directly to tape or is tape spill is being used. This name is used in messages to the operator requesting a tape mount, and may be up to six characters long. If EBTPOOL is not used, mount messages request specific tapes.

EBTXF Must be specified as YES or NO. YES specifies that when backup and/or archive is to go directly to tape, use is to be made of the IDRC feature of the tape units for compression. The default is NO.

See 4.9, “Backup Directly to Tape” on page 29 and 4.11, “Archive Directly to Tape” on page 30 for information on the use of this option.

ECYCLEN Is described below in the description of ECYCLET. It specifies a number of seconds and must be between 1 and 99.

ECYCLER If this parameter is specified, it defines a retry count. When a specific tape is requested during the backup process and the tape is not mounted in the time defined by ECYCLEN and ECYCLET, a different tape is requested. This is repeated until the number of tapes requested reaches the value specified. It must be between 0 and 99. A value of zero requests no retry, which is the default.

ECYCLET ECYCLET and ECYCLEN control the issue of tape mount messages. ECYCLET specifies a count for the number of times a message is issued via the HDFXMO exit requesting a tape mount and must be a number between 0 and 99999999. Each time the message is issued, BARS/VM waits for the number of seconds defined in ECYCLEN and then tests to see if the tape is available. If it is not, the message is issued again.

When the message/wait cycle has been executed for the number of times in ECYCLET, message 0880 is issued and the tape is not used.

ENUMTAPE Specifies the maximum number of tape drives available to BARS/VM for tape operations such as backup, reorg and restore. It does not apply to disaster backup, which uses ENUMTDBR. It must be a number between 0 and 32767.

If backup tapes are used and EFASTRT is set to NO, ENUMTAPE must be at least two to allow tape reorganization to take place. If archive tapes are used, reorganization is always tape to tape and so there must be at least two tape units. If twin tapes are used, there must be at least three, since reorganization copies from the source tape to two others.

ETAPCAP Use this parameter to specify a threshold number of 2K blocks for tapes. At the end of a dump or spill to tape, the tape will be marked as full if the number of blocks it contains is greater than or equal to this value. This option can be useful in reducing the number of tape mounts by avoiding repeated mounts for the same tape when BARS/VM is trying to use the ever decreasing remaining available space on the tape.

For tape spill, this option can be used to limit the capacity of a tape to match the capacity of the minidisks being spilled.

A value of zero means the tape will only be marked full when no more space is available on it. This is the default action if the parameter is not specified. It must be a number between 1 and 99999999.

ETAPCOMP Must be specified as YES or NO. YES specifies that compression is to be used when backing up directly to tape. If EBTXF is set to YES hardware compression is used, otherwise software is used. The default is NO.

See 4.9, “Backup Directly to Tape” on page 29 for information on the use of this option.

ETAPEREL Must be specified as YES or NO. YES specifies that when a subuser is finished with a tape it is not to attempt to give it to the Interface user id but just DETACH it. The default is NO.

See 4.13, “Use of Specific Tapes” on page 31 for information on the use of the GIVE command and DETACH with the LEAVE option.

ETAPLIFE Specifies, in years, the expected life of a tape. If this option is specified and is not zero, warning messages will be generated for tapes that have been added to BARS/VM for longer than this period, indicating that the tapes should be replaced. The value specified must be a number between 0 and 32767.

ETPDEN Defines, for BARS/VM systems using tape backup, the default tape density to be used if none is specified when a tape is added using the TAPE command. To use 3480 or 3490 tapes, specify the value as 38000 which gives a density of 38K.

ETPFULL Defines, for BARS/VM systems using tape, a threshold for tape reorganization. The number is a value from 1 to 99 which is taken as a percentage, and applies to both backup and archive tapes. If a tape is more than this

percentage occupied by valid files, the reorganization process marks the tape as successfully reorganized but does not attempt to reclaim any space. This is to avoid the overhead of tape reorganization when the amount of space that would be reclaimed does not justify it. The default is 70. Specify a value of 99 to avoid this threshold checking and always allow reorganization to take place.

ETRETRY Specifies the number of times to retry a tape mount when an invalid label has been found on the mounted tape.

Disaster Backup and Recovery (DBR) Options

EDBRCDUP If specified, it must be YES or NO. YES will cause duplicate control file tapes to be created for BASE and DELTA backup runs. This is for sites who require the extra protection that two sets of tapes provide. If NO is specified, or if the default is taken, only one set of control file tapes is created. This parameter does not apply to LSET dumps. The default is NO.

EDBRMID Is a machine identification, used to build an OS dataset name for DBR tapes. The length is limited to three characters because of OS DSNAME length limitation in the OS HDR1 record.

EDBRPOOL Specifies a DBR tape pool name which is placed in the mount request message. DBR then checks the OS label and compares it with the possible ones in the file RECOVERY VOLUMES. The value must be from 1 to 6 characters.

EDBRREQ This must be specified as YES or NO. If YES is specified, this specifies that DBR backup will only be done for minidisks for which a BACKUP DBR command has been issued. If the setting is NO, all minidisks are candidates for backup. The default is NO.

EDBRUSE This must be specified as YES or NO. The option controls how subusers are used to back up minidisks for DBR. If YES is specified then any subuser may process any minidisk. This may give better performance if there are some user IDs with many minidisks. If NO is specified, all minidisks for a particular user ID are processed by the same subuser. This results in the backup data for each user ID being on the smallest number of tapes. The default is NO.

EEXPBASE Specifies the number of days after which a DBR BASE dump expires. It must be a number between 0 and 32767.
This information is not only used for internal calculations during the DBR process, but it is also written into the OS HDR1 record on the disaster backup tapes.

EEXPIRE Specifies the number of days after which a DBR DELTA or LSET dump expires. It must be a number between 0 and 32767.

This information is not only used for internal calculations during the DBR process, but it is also written into the OS HDR1 record on the disaster backup tapes.

ENUMTDBR Specifies the maximum number of tape drives available for the disaster backup process. It must be a number between 0 and 32767.

Report Message Selection Options

ERFLCMS0 Specifies that a long report is to include report messages for CMS minidisks and directories for which no files were copied for backup because there were no new or changed files. The parameter must be specified as YES or NO. It is ignored if either ERFLREP or ERFLONG is specified as NO. The default is NO.

ERFLONG Specifies that a long report is to be generated. It must be specified as YES or NO. It is ignored if ERFLREP is specified as NO. The long report generates informational messages for minidisks and directories during the backup process detailing the amount of data backed up. The default is NO.

ERFLREP Specifies that a report is to be generated. It must be specified as YES or NO. If YES is specified, messages summarizing the results of the backup process are generated. Additional information is produced if a long report is requested (see above). No information is generated if NO is specified. The default is NO.

Trace Options

ETRACCMD Specifies whether the remote commands trace is required or not. It must be specified as YES or NO. The default is NO.

ETRACCMS Specifies whether the CMS command trace is required or not. It must be specified as YES or NO. The default is NO.

ETRACCP Specifies whether the CP command trace is required or not. It must be specified as YES or NO. The default is NO.

ETRACINT Specifies whether the VMCF INTERFACE trace is required or not. It must be specified as YES or NO. The default is NO.

ETRACVMC Specifies whether the VMCF contents trace is required or not. It must be specified as YES or NO. The default is NO.

20.3.3 HDFCLS ASSEMBLE - Define the Permitted ARCHIVE Classes

The installation must set those characters which are to be permitted in the CLASS parameter of the APUT command into a list in module HDFCLS. The list must terminate with a byte containing X'FF'. Below is an example of a possible list.

```
HDFCLS   CSECT
         DC    C'U'              UNCLASSIFIED
         DC    C'I'              INTERNAL USE ONLY
         DC    C'C'              CONFIDENTIAL
         DC    C'X'              CONFIDENTIAL RESTRICTED
         DC    C'R'              REGISTERED CONFIDENTIAL
         DC    X'FF'             END OF LIST MARKER
         END
```

20.3.4 User Exits

Special user exit modules are provided so that the installation may provide its own code to handle certain situations. All of these exits, with the exception of HDFXMG, are assembler modules and default versions are distributed as ASSEMBLER and TEXT files. In some cases an EXEC file with the same name is available. In this case, the assembler routine usually tests for the existence of the EXEC file and invokes it if it does exist. The main function of the exit can then be coded in the EXEC. Check the distributed files to determine the functions they perform and modify them as necessary. The user exit modules are as follows:

HDFXAC

HDFXAC is called is called from the ADEFINE command processor to verify and optionally change the account number specified on the command. On entry, general register 1 contains the address of a parameter list. The parameter list consists of:

```
A(account number) - the account number is 8 characters
```

The module sets a return code in register 15 before returning. The return code must be as follows:

```
       0 - proceed with the command and use the account number, which
           may have been modified
 not 0 - terminate the command with a message to say the account
           number is invalid
```

HDFXBP

The function of HDFXBP is to allow an installation to perform additional processing before the application of the default BARS/VM backup philosophy. The default philosophy is to achieve automatic backup for any CMS formatted object (CMS minidisk or SFS directory) and to skip others. The exit gets called for each object which is not excluded from backup and for which no control file exists. So after the first time this exit was passed telling BARS/VM to backup data it will not be called again for the same object.

The exit must set a return code before returning which must be as follows:

```
0 = successful completion; proceed to backup
    the object with default backup policy.
1-999 = successful completion; override backup
        interval from return code.
1000 = do not backup the object.
```

The easiest way to implement local backup philosophy is to place an HDFXBP EXEC onto the BARS/VM 194 minidisk containing the run time code. The supplied assembler exit will call an HDFXBP EXEC if one exists on the Interface 194 minidisk. The first time the exit gets called the existence of the EXEC is checked and if so loaded using EXEC LOAD. This is done for performance

reasons and implies a subsystem restart each time a logic change has been applied to the EXEC.

On entry to HDFXBP the EXEC gets passed the following parameters:

```
format            (CMS or non-CMS)
backup interval   number of days of default intervals
                  for this type of object
User ID            User ID owning the object
Cuu               Minidisk address or X'00'
Dirid             Directory name or X'00'
```

Note: You should question any non-CMS backup for reliability and consistency of the data as long as the application is running during the backup time. A better way to backup non-CMS data might be to create a backup once and to set the minidisk to STOP using the BACKUP userID cuu STOP command. Subsequent backup should be taken under control of the administration personal of the non-CMS application using a BPUT command.

HDFXB1

HDFXB1 is called is called when backup is about to take place for a minidisk or an SFS directory. On entry general register 1 contains the address of a parameter list. The parameter list consists of:

```
A(user ID) - the user ID is 8 characters
A(minidisk cuu or zero)   - zero for an SFS directory
                          - the cuu is 4 characters
A(directory name or zero) - zero for a minidisk
                          - the directory name is 153 characters
```

The module sets a return code in register 15 before returning. The return code must be as follows:

```
      0 - proceed to backup the minidisk or directory
not 0 - do not backup the minidisk or directory
```

HDFXB2

HDFXB2 is called is called when backup is complete for a minidisk or SFS directory. On entry, general register 1 contains the address of a parameter list. The parameter list consists of:

```
A(user ID) - the user ID is 8 characters
A(minidisk cuu or zero)   - zero for an SFS directory
                          - the cuu is 4 characters
A(directory name or zero) - zero for a minidisk
                          - the directory name is 153 characters
```

HDFXCA

HDFXCA is called when a command is received by BARS/VM. It is called in the Interface virtual machine. The parameters are:

```
Register 1 contains the address of the following command
           information:
                    tokenized command - 160 bytes
                    reserved - 24 bytes
                    originating node or zero if local - 8 bytes
                    originating user id - 8 bytes
                    reserved - 9 bytes
                    reason code - 2 bytes hexadecimal
                    reserved - 161 bytes
                    session id - 1 byte hexadecimal
                    reserved - 2 bytes
                    command id - 4 bytes hexadecimal
```

The module sets a return code in register 15 before returning. The return code must be as follows:

```
      0 - proceed to queue the command for execution
not 0 - ignore the command
```

If this exit sends a message to the manager user ID with a message prefix of HDFXCA (such as HDFXCA0001I), and the “Quality_Report” option in the MANAGER OPTIONS file (see 20.3.5, “Manager Installation Options” on page 247) is set to YES, then this message will be written to the quality log file.

HDFXCB

HDFXCB is called immediately before a command is to be executed. It is called in the subuser virtual machine. The parameters are:

```
Register 1 contains the address of the following command
           information:
                    tokenized command - 160 bytes
                    reserved - 24 bytes
                    originating node or zero if local - 8 bytes
                    originating user id - 8 bytes
                    reserved - 9 bytes
                    reason code - 2 bytes hexadecimal
                    reserved - 161 bytes
                    session id - 1 byte hexadecimal
                    reserved - 2 bytes
                    command id - 4 bytes hexadecimal
```

The module sets a return code in register 15 before returning. The return code must be as follows:

```
      0 - proceed to execute the command
not 0 - do not execute the command
```

If this exit sends a message to the manager user ID with a message prefix of HDFXCB (such as HDFXCB0001I), and the “Quality_Report” option in the MANAGER OPTIONS file (see 20.3.5, “Manager Installation Options” on page 247) is set to YES, then this message will be written to the quality log file.

HDFXCE

HDFXCE is called immediately after a command is executed. It is called in the subuser virtual machine. The parameters are:

```
Register 0 contains the return code from the command
Register 1 contains the address of the following command
           information:
                    tokenized command - 160 bytes
                    reserved - 24 bytes
                    originating node or zero if local - 8 bytes
                    originating user id - 8 bytes
                    reserved - 9 bytes
                    reason code - 2 bytes hexadecimal
                    reserved - 161 bytes
                    session id - 1 byte hexadecimal
                    reserved - 2 bytes
                    command id - 4 bytes hexadecimal
```

The module sets a return code in register 15 before returning. The return code must be as follows:

```
0 - normal completion
```

If this exit sends a message to the manager user ID with a message prefix of HDFXCE (such as HDFXCE0001I), and the “Quality_Report” option in the MANAGER OPTIONS file (see 20.3.5, “Manager Installation Options” on page 247) is set to YES, then this message will be written to the quality log file.

HDFXCM

HDFXCM is called after a command has been merged with another command. This merging is done to improve performance. It is called in the Interface virtual machine. The parameters are:

```
Register 0 contains the address of the following parameters for the
          command that is being merged into:
         Parameters are the same as for Register 1 below
Register 1 contains the address of the following information for the
           command that is being merged:
                    tokenized command - 160 bytes
                    reserved - 24 bytes
                    originating node or zero if local - 8 bytes
                    originating user id - 8 bytes
                    reserved - 9 bytes
                    reason code - 2 bytes hexadecimal
                    reserved - 161 bytes
                    session id - 1 byte hexadecimal
                    reserved - 2 bytes
                    command id - 4 bytes hexadecimal
```

The module sets a return code in register 15 before returning. The return code must be as follows:

```
0 - normal completion
```

If this exit sends a message to the manager user ID with a message prefix of HDFXCM (such as HDFXCM0001I), and the “Quality_Report” option in the MANAGER OPTIONS file (see 20.3.5, “Manager Installation Options” on page 247) is set to YES, then this message will be written to the quality log file.

HDFXCO

The function of HDFXCO is to perform a disk file copy function for BARS/VM. It is called for each disk file copy operation. The module may be coded by the installation to suit its local requirements for file copying. However care must be taken to ensure the correct return codes are set.

On entry to HDFXCO register 1 contains the address of a CMS format parameter list. The format of the parameter list is the same as that for the CMS COPYFILE command except for the first 8-byte token which is set as follows to specify the type of copy:

```
'COPYFILE' this is for a normal file copy
'COPYOUT ' this is for a copy to backup or archive
             of data in the user's format
'COPYIN  ' this is for a copy from backup or archive
             to data in the user's format, for example for
             RETRIEVE or AGET commands.
'COPYBACK' this is for a copy from backup or archive
             to backup or archive, for example for
             reorganizations.
'COPYFAIL' this indicates that a previous file copy failed. It
             gives the installation a chance to retry the copy
             in a different way.
```

The module sets a return code in register 15 before returning. The return code must be as follows:

```
'COPYFILE'
     0 - files copied successfully
     28 - file not found
     All others - files not copied successfully


'COPYOUT '
     0 - files copied as specified
     4 - files copied unchanged (PACK was specified
           but the files would not pack)
     28 - file not found
     All others - files not copied successfully


 'COPYIN  '
     0 - files copied as specified
     4 - files copied unchanged (UNPACK was specified
           but the files would not unpack)
     28 - file not found
     All others - files not copied successfully


 'COPYBACK'
     0 - files copied successfully
     28 - file not found
     All others - files not copied successfully


'COPYFAIL'
     0 - files copied successfully
     All others - files not copied successfully
```

HDFXDP

The function of HDFXDP is to allow an installation to perform additional processing before the application of the default DBR backup policy. The default policy is described in 13.1.3, “The Disaster Backup Policy and How to Change it” on page 174. The exit gets called for each object which is not excluded from DBR by means of the EDBRREQ installation option.

The exit must set a return code before returning which must be as follows:

```
 0 = successful completion; proceed to backup
    the object with default policy.
 4 = successful completion; proceed to backup
    the object in CMS format.
 8 = successful completion; proceed to backup
    the object in NON-CMS format.
>8 = do not backup the object.
```

The easiest way to implement a local DBR policy is to place an HDFXDP EXEC onto the BARS/VM 194 minidisk containing the run time code. The supplied assembler exit will call an HDFXDP EXEC if one exists on the Interface 194 minidisk. The first time the exit gets called the existence of the EXEC is checked and if so loaded using EXEC LOAD. This is done for performance reasons and implies a subsystem restart each time a logic change has been applied to the EXEC.

On entry to HDFXDP the EXEC gets passed the following parameters:

```
format            ('CMS', 'NON-CMS' or 'BLOCKIO')
DBR dump type     ('BASE', 'DELTA' or 'LSET')
LSET dump name    (lsetname or '*' if none)
eligibility       'P' = eligible for EDBRREQ NO policy
                  'Y' = eligible for EDBRREQ YES
                  'N' = not eligible for EDBRREQ YES
nobackup set      'Y' = Backup NOBACKUP set
                  'N' = Backup NOBACKUP not set
backup data       'B' = Both, CMS and NONCMS backup data exists
                  'N' = NONCMS backup data exists
                  'C' = CMS backup data exists
                  '-' = No backup data exists
recomp cyls       'Y' = CMS minidisk with RECOMP cyls
                  'N' = No RECOMP cyls detected
User ID           User ID owning the object
Cuu               Minidisk address or X'00'
DIRID             Directory name or X'00'
                  Note that there is no DBR SFS support
                        at present.
```

HDFXGV

HDFXGV is called before External Security Manager authorization checking when the installation parameter ESECDATA is set to ESM1. It allows an installation to provide its own authorization checking instead of, or in addition to, the ESM. See 8.4, “Authorizing other users” on page 85 for a discussion of ESECDATA. The parameters are:

```
Register 0 contains the address of the following command
           information:
                    tokenized command - 160 bytes
                    reserved - 24 bytes
                    originating node or zero if local - 8 bytes
                    originating user id - 8 bytes
                    reserved - 11 bytes
                    SFS directory name or zero - 153 bytes
Register 1 contains the address of the following authorization
           information:
                    Userid requiring access (8 characters)
                    Access required (8 characters)
                    Class (8 characters)
                    Userid whose resource it is (8 characters)
                    Name of resource for class VMMDISK (4 characters)
                      or
                    Address of 153 character directory name for
                      class DIRECTRY (4 bytes)
```

The module sets a return code in register 15 before returning. The return code must be as follows:

```
 0 - defer the decision to the ESM
 1 - allow the command and do not ask the ESM
-1 - disallow the command and do not ask the ESM
```

HDFXLB

HDFXLB is called after a backup or archive tape was labelled or re-labelled. It is to allow installations to add values to the HDR1 area of IBM standard labelled tapes if required by simply issuing a LABELDEF command. On entry, general register 1 contains the address of a parameter list. The parameter list consists of:

```
LABELDEF name  - 8 characters
Nolog userid   - 8 characters
external label - 6 characters
```

On exit, the return code will determine processing:

RC=0 No HDR1 needed. This is the default function.
RC¬=0 LABELDEF command issued. BARS/VM will try to write HDR1 information using the CMS TAPESL macro. See LABELDEF command description for possible values to set.

The supplied assembler exit will call an HDFXLB EXEC if one exists on the Interface 194 minidisk, which allows the coding of the exit in REXX. Such an EXEC will be EXECLOADed for performance reasons, and hence a subsystem restart is required to pick up any changes to this EXEC.

HDFXLC

HDFXLC is called during tape mount processing to allow the installation to verify HDR1 labels created via the HDFXLB exit. The exit is called for all tape mounts except DBR, and is only called if the VOL1 label has already been successfully validated against the list of known BARS/VM tapes.

On entry, general register 1 contains the address of a tokenized parameter list. The parameter list consists of:

```
Virtual tape address - 3 characters
Nolog userid         - 8 characters (used by HDFXLB - "." if none)
External label       - 6 characters (as specified on mount request)
Read or write flag   - 3 characters (either R/O or R/W)
```

On exit, the return code will determine processing:

RC=0 Use the tape.
RC¬=0 Do not use the tape.

The supplied assembler exit will call an HDFXLC EXEC if one exists on the Interface 194 minidisk, which allows the coding of the exit in REXX. Such an EXEC will be EXECLOADed for performance reasons, and hence a subsystem restart is required to pick up any changes to this EXEC.

HDFXMG

The HDFXMG EXEC is called during subuser initialization in order to establish the Callable Services Library interface for handling the backup and restoring of migrated files. This exit is only called if the installation option EBACKMIG is set to YES. The distributed EXEC may be modified to suit the local environment. There are no parameters.

The EXEC must set a return code as follows:

```
     0 - successful completion
 not 0 - unsuccessful completion - do not attempt to back up
         migrated files
```

HDFXMO

The user exit, HDFXMO, requests mounting of a tape. An installation may place any desired code in this exit according to local arrangements for tape mounting. The default exit supplied simply issues a message of the following format to the operator user ID:

```
Mount label address mode for user subuser

Where  label   = External label of tape
       address = Address at which tape is to be attached
       mode    = R/O or R/W, XF if IDRC required
       subuser = Subuser to which the tape is to be attached
```

On entry, general register 1 contains the address of a parameter list consisting of:

```
A(6 character external tape label)
A(3 character address for ATTACH)
A(8 character mode)
    CL3 R/O or R/W
    CL1' '
    CL2'XF' or CL2'  '
    CL2'  '
A(8 character user id for ATTACH)
A(8 character task name)
A(8 character operator user ID)
A(8 character manager user ID)
```

The taskname will be VMDBR for a DBR pool tape request, VMBACK for a backup pool tape request or BACKUP for all other tape requests.

The supplied assembler exit will call an HDFXMO EXEC if one exists on the Interface 194 minidisk, which allows the coding of the exit in REXX. Such an EXEC will be EXECLOADed for performance reasons, and hence a subsystem restart is required to pick up any changes to this EXEC.

HDFXTP

The function of HDFXTP is to perform tape functions for BARS/VM. It is called for each tape operation. The module may be coded by the installation to suit its local requirements for tape operations. However care must be taken to ensure the correct return codes are set. The following functions must be supported: FSF, REW, WTM, WVOL1, DUMP, LOAD, SCAN.

On entry to HDFXTP register 1 contains the address of a CMS format parameter list. The format of the parameter list is the same as that for the CMS TAPE command except for the first 8-byte token which is set to either 'HDFCOPY ' or 'HDFCPY '.

The module sets a return code in register 15 before returning. The return code must be as follows:

```
          0 - successful completion
         28 - file not found
         32 - invalid record format
         40 - end-of-file or end-of-tape
        100 - error reading or writing tape
all others - unsuccessful completion
```

20.3.5 Manager Installation Options

MANAGER OPTIONS - Define the Manager Options

Subsystem_Name Define the name of the BARS/VM subsystem. This is the same as the filename of the VMBARS EXEC, MODULE and PROFILE files (i.e. VMBARS) and the value of the ENAME installation option.

Dist_Code_Disk Define the location of the distributed code disk. Set this to 291 to use the Manager 291 minidisk, or a fully qualified directory name (e.g. filepool:userid.) to use SFS.

Backup_Report Controls whether a separate backup report file is generated. It must be specified as YES or NO.

DBR_Report Controls whether a separate DBR report file is generated. It must be specified as YES or NO.

DBR_Keep Defines how old a DBR report file has to be before being eligible for erasure. It must be a number of days greater than 0.

Matrix_Report Controls whether a separate matrix report file is generated. It must be specified as YES or NO.

Reorg_Report Controls whether a separate reorg report file is generated. It must be specified as YES or NO.

Quality_Report Controls whether a separate quality log file is generated. It must be specified as YES or NO. See the HDFXCA, HDFXCB, HDFXCE and HDFXCM descriptions in 20.3.4, “User Exits” on page 239 for how these messages are created.

Highlight_Report Controls whether a separate highlighted messages file is generated. It must be specified as YES or NO.

Archive_Logs Controls whether the main log file is automatically archived. It must be specified as YES or NO.

Archive_Node Defines the node on which to archive the log file. An initial ADEFINE command must be done manually to register the manager user ID for archive on this system. The DELEXP option is recommended, and sufficient space must be allocated.

Archive_Months Defines the retention period for the archived logs in months. It must be a number between 1 and the value of EMAXRET.

Date_File_Size Defines the number of records to keep in the manager new day log. It must be a number greater than 0.

Interface_High Defines the value at which to start highlighting Interface minidisk full messages. It must be a number between 1 and 100.

Append_HM Defines whether the highlighted messages log is automatically appended to the summary report when a SENDREP command is executed. It must be specified as YES or NO.

MANAGER AUTHLIST - Define the Manager Authorities

The following is an example of the MANAGER AUTHLIST file which resides on the manager 194 minidisk:

```
BEDROCK1 BACKMON  SENDREP
BEDROCK1 BARNEY   REACCESS SENDREP
BEDROCK1 FRED     *
-local-  BARNEY   REACCESS REIPL SENDREP
-local-  FRED     *
-local-  OPERATOR SHUTDOWN
```

The first two columns form a system/userID pair which identifies specific user IDs authorized to issue certain manager commands. A value of `-local-` is a special symbol representing the host system.

Columns 3 onwards indicate which commands the user ID is allowed to issue. The possible values are REACCESS, REIPL, SENDREP and SHUTDOWN. A value of * represents all possible commands.

VMBARS RTABLE - Manager PROP Table

The start of the VMBARS RTABLE file needs to be modified if the manager userID is running a level of CMS prior to release 9. For earlier releases, the keyword LOWCASE needs to be removed from the LOGGING statement. See the following examples:

```
*--------------------------------------------------------------------*
*                                                                    *
* VMBARS RTABLE: BARS/VM Manager userID PROP table.                  *
*                                                                    *
*--------------------------------------------------------------------*
LGLOPR AUTOLOG1
TEXTSYM / ¢ ¬
LOGGING ALL LOWCASE
ROUTE
```

Figure 49. Start of VMBARS RTABLE for CMS 9 and later

```
*--------------------------------------------------------------------*
*                                                                    *
* VMBARS RTABLE: BARS/VM Manager userID PROP table.                  *
*                                                                    *
*--------------------------------------------------------------------*
LGLOPR AUTOLOG1
TEXTSYM / ¢ ¬
LOGGING ALL
ROUTE
```

Figure 50. Start of VMBARS RTABLE for CMS 8 and earlier

The end of the VMBARS RTABLE file needs to be modified in two places.

The first is to define who is authorized to use the `CMD` command to execute any CP or CMS command. The following example shows BARNEY and FRED at BEDROCK1 and BEDROCK6 are authorized:

```
*--------------------------------------------------------------------*
*                                                                    *
* Authorize BARS/VM support personnel to issue CP/CMS commands       *
*                                                                    *
*--------------------------------------------------------------------*
/CMD/                       1   3    BARNEY   BEDROCK1 DMSPOR   TOVM
/CMD/                       1   3    FRED     BEDROCK1 DMSPOR   TOVM
/CMD/                       1   3    BARNEY   BEDROCK6 DMSPOR   TOVM
/CMD/                       1   3    FRED     BEDROCK6 DMSPOR   TOVM
```

The second is to define the manager user ID itself to give it authority to execute the PROP routines DMSPOL and DMSPOR. This should have already been set by the BARSINST EXEC, but should be checked for correctness. The following example shows BARMANGR at BEDROCK6 is authorized:

```
*-------------------------------------------------------------------------*
*                                                                         *
* Authorize BARS/VM manager userID to re-load PROP table.                 *
*                                                                         *
*-------------------------------------------------------------------------*
/LOADTBL/                  1   7    BARMANGR BEDROCK6 DMSPOL   VMBARS
*-------------------------------------------------------------------------*
*                                                                         *
* Authorise BARS/VM manager userID to re-load PROP table.                 *
*                                                                         *
*-------------------------------------------------------------------------*
/STOP/                     1   4    BARMANGR BEDROCK6 DMSPOR   STOP
```

20.4 ISPF Panel Interface

20.4.1 How the ISPF Panel Interface is Made Up

The ISPF Panel Interface is made up of a series of panels which prompt the user to select a function and enter the necessary parameters to execute a BARS/VM command. These panels are organized in a hierarchical manner, and are split into several main sections, some of which are split into further subsections. These panels are held in the VMBPANEL MACLIB file.

Associated with each panel is a help panel, which is accessed by pressing F1 (or typing help), and which explains the function of the panel and how to fill in the fields - this help may consist of more than one screen. These help panels are held in the VMBHELP MACLIB file.

Also, each panel has associated with it a set of messages which contain two levels of error message relating to the various input fields on the panel. These messages are held in the VMBMSGS MACLIB file.

The bottom-level panels (i.e. not the selection panels) also have an EXEC file which takes the data entered and calls the VMBARS MODULE to perform the required operation. In the case of the two FILELIST operations, these are performed by the EXECs themselves.

The following figure illustrates this relationship:

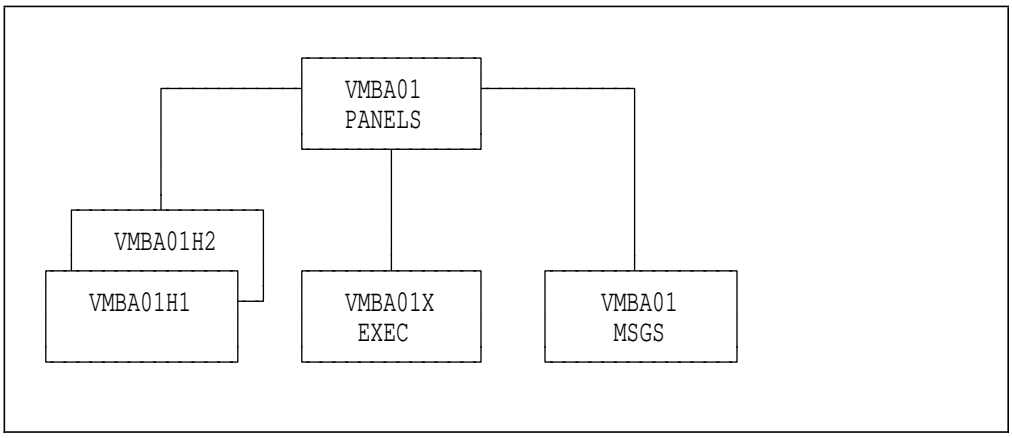


Figure 51. Relationship between Files for BARS/VM Option A.1

20.4.2 How to Install the ISPF Panel Interface

The VMBARS EXEC is written to always call the VMBXISPF exit to obtain the ISPF code disk. This sample exec is coded to do the following:

1. Do nothing and exit if the ISPF code disk is already accessed (indicated by the presence of the ISPSTART EXEC)
2. Try a VMLINK ISPF if VMLINK is available
3. Try a RZLINK ISPF if RZLINK is available
4. Try linking and accessing the ISPVM 192 disk

If your installation has installed the ISPF product on a different disk, and VMLINK or RZLINK are not available, then you should modify the exec in order to LINK/ACCESS and RELEASE/DETACH the correct minidisk. BARS/VM does not require the ISPF disk to be accessed at any specific filemode.

For line-mode commands, only the VMBARS EXEC, VMBARS MODULE and VMBARS PROFILE files are used, and since these all reside on a public minidisk (normally the Y-disk), no extra minidisks are linked. When the panel interface is invoked, as well as requiring the ISPF code disk, the VMBARS EXEC will link and access the Interface 19F minidisk which contains the panel code (if you wish to use a different minidisk to hold this code, or share a common minidisk between multiple BARS/VM systems on one VM system, then replace the MDISK card in the Interface directory with a LINK to the appropriate minidisk). This minidisk is released and detached on exit from the panels. To determine whether the Interface 19F minidisk needs to be linked, the VMBARS EXEC looks for the VMBARS USRIFACE file. If this file is not found, then the Interface 19F minidisk is linked and accessed. If it is found, then the panel code is assumed to be already available. Hence if all files (including the VMBARS USRIFACE file) are installed on the Y-disk then the Interface 19F minidisk is not needed.

The VMBXDEFS EXEC contains all the installation definable variables required by the panel interface. Some of these variables are duplicates of those found in the OPTIONS file on the interface 194 minidisk, and should be set to the same values. Others are specific to the panel interface, and in some cases may be overridden by individual users of the interface.

Panel Specific Options

Variable	*Purpose*
acat[3]	Enter Y to enable use of the archive related panels, else enter N.
bcat[3]	Enter Y to enable use of the backup related panels, else enter N.
dcat[3]	Enter Y to enable use of the DBR related panels, else enter N.
cmdmsg[4]	Allows choice of confirmation message following commands issued by the panels. The valid choices are Y for the standard message, E to echo the command issued and N for no message
timeout[4]	To prevent accidental re-issue of the same command, there is a time-out before identical commands can be re-issued
adefacnt	Default account number for ADEFINE panel Leave blank for no default.
adefdept	Default department name for ADEFINE panel Leave blank for no default
adefdlxp	Enter Y to set DELEXP as the default for ADEFINE panel, else enter N
adefshrd	Enter Y to set SHARED as the default for ADEFINE panel, else enter N
retnow	Enter Y to set the default for the NOW field on the RETRIEVE panel to Y, otherwise enter N.
retnowf	Enter Y to set the RETRIEVE default for the NOW field on the Filelist Backup panel use the NOW option. Enter N for deferred RETRIEVEs.

Options Relating to Those Found in OPTIONS File

Variable	*Purpose*
class1	Archive class - Unclassified
class2	Archive class - Internal Use Only
class3	Archive class - Confidential
class4	Archive class - Confidential Restricted
class5	Archive class - Registered Confidential
earclass	Default archive class
ibackmc[5]	The minimum permissible interval between backups for CMS disks (in days)
ibackmn[5]	The minimum permissible interval between backups for non-CMS disks (in days)
iarcml[5]	The highest archive level allowed. Remember that the number of levels can be altered for individual users via the AALTER command
ibackml[5]	The highest backup level allowed
iretain[5]	The maximum number of months which may be specified for archive retention
earcret[6]	Number of days to retain information after an archive file is deleted. This is used by the DQUERY and DRECOVER commands. Set this to the value of ERASDAYS in your OPTIONS file
edatef[6]	Allows choice of date format to be used in the filelist displays
edatec[6]	Allows choice of date field separator in filelist displays.

The following is a list of the files which make up the ISPF Panel Interface:-

- VMBARS2 EXEC
- VMBA01X EXEC
- VMBA02X EXEC

---

3 Specifying N for all three variables is not supported, and will be treated the same as specifying Y for all variables.

4 Users may override this default using the Profile panel.

5 These are used to help validate some of the input fields of the panels

6 The FILELIST panel code obtains its data in a format unaffected by the date settings in the OPTIONS file on the interface 194 minidisk. These settings define how the panel will show dates

- VMBA03X EXEC
- VMBA04X EXEC
- VMBA05X EXEC
- VMBA06X EXEC
- VMBA07X EXEC
- VMBB01X EXEC
- VMBB02X EXEC
- VMBB03X EXEC
- VMBB04X EXEC
- VMBB05X EXEC
- VMBB06X EXEC
- VMBB07X EXEC
- VMBC00X EXEC
- VMBD01X EXEC
- VMBD02X EXEC
- VMBE00X EXEC
- VMBFATBL EXEC
- VMBFBTBL EXEC
- VMBF01X EXEC
- VMBF02X EXEC
- VMBISSUE EXEC
- VMBNOPX EXEC
- VMBTBSAV EXEC
- VMBXDEFS EXEC
- VMBXISPF EXEC
- VMBHELP MACLIB
- VMBMSGS MACLIB
- VMBPANEL MACLIB
- VMBARS USRIFACE
- VMBFRECV XEDIT

The VMBXISPF EXEC must be placed on a public minidisk (normally the Y-disk) along with the VMBARS EXEC, VMBARS MODULE and VMBARS PROFILE files. They are required for BARS/VM line commands and to start the ISPF session.

The other files should reside on the Interface 19F minidisk, or may be placed on the Y-disk or the ISPF code disk (i.e. the disk defined in the VMBXISPF EXEC) if preferred. These files are only required once ISPF is started.

Chapter 21. Installation Steps

The installation of the BARS/VM system is done via the supplied BARSINST EXEC. Before you can run this EXEC you have to perform the following steps:

Step 1: Load the BARS/VM Distribution Files

1. Logon to the user ID which owns the minidisk or SFS directory where you wish to place the distribution materials. It is suggested in the section 20.2, "Planning BARS/VM Virtual Machines" on page 217 that this should be the manager user ID.
2. If you are using a minidisk, and have not already done so, then link the minidisk R/W and format it with the CMS command:

   ```
   FORMAT vaddr B ( BLKSIZE 2K
   ```

3. Ensure that you have accessed the minidisk or directory as mode B.
4. Load the BARS/VM materials onto the B-disk. If the files have been received on a tape use the CMS command:

   ```
   VMFPLC2 LOAD * * B ( EOF 2
   ```

Step 2: Define the BARS/VM Virtual Machines

The virtual machines which comprise the BARS/VM subsystem should be defined in the VM source directory.

Please refer to the section 20.2, "Planning BARS/VM Virtual Machines" on page 217 in this manual for details on the directory entries and the DASD requirements for the virtual machines. There are samples of the directory entries supplied on the distribution tape.

Step 3: Ensure BARS/VM Can LINK All Minidisks and has SFS Authority

BARS/VM must be able to CP LINK all minidisks on the system to perform backup of those minidisks. If the RACF/VM program product is installed, then this is achieved by the granting of the OPERATIONS attribute via the RACF ALTUSER privileged command.

If RACF/VM is not available, then CP modifications may be required (as previously discussed) in order for the BARS/VM virtual machines to CP LINK any minidisk. The authority to use this modification may be via CP privilege, directory statement or the code itself. On VM/ESA there is a directory statement OPTION LNKNOPAS which allows the user ID with this option to CP LINK any minidisk without having to specify a password. Whichever method is used, ensure that authority is granted to the following BARS/VM virtual machines:

- Interface
- Backup Controller
- Query Controller
- All General Subusers.

If Shared File System (SFS) backup is to be performed, the virtual machines listed above must be given administration authority to all file pools requiring backup, plus the file pool containing the BARFS filespace if alias and authorization backup is enabled. This allows BARS/VM read and write authority

to all SFS objects and allows it to create and delete objects in other users' file spaces. Consult the SFS administration manual for details of how to give administration authority.

Grant the manager user ID authority to link the Interface 194 and 19F minidisks R/W (LINK mode M) if you intend to use the manager REFRESH command to apply code updates.

Step 4: Modify HDFCLS ASSEMBLE if Required

Warning:

If you tailor the HDFCLS or user exit files in the following two steps, then remember to save copies of the modified versions under a different filetype so that they will not be lost when the code is refreshed.

If the default BARS/VM archive classes in the supplied version of HDFCLS TEXT do not meet your requirements, then you may modify them accordingly. To do this, update the file HDFCLS ASSEMBLE, re-assemble, and replace the supplied version of HDFCLS TEXT with your local version. This is described further in the section 20.3.3, “HDFCLS ASSEMBLE - Define the Permitted ARCHIVE Classes” on page 238.

Note: If HDFCLS is modified, then ensure that the default class as defined by the EARCLASS installation option (see “General Options” on page 226) is specified in the HDFCLS ASSEMBLE. Also ensure that matching values are coded in the VMBXDEFS EXEC exit as described in “Step 7: Run the BARSINST EXEC” on page 257.

Step 5: Prepare User Exit Modules and EXECs

Check the files HDFXAC ASSEMBLE, HDFXBP ASSEMBLE, HDFXBP EXEC, HDFXB1 ASSEMBLE, HDFXB2 ASSEMBLE, HDFXCA ASSEMBLE, HDFXCB ASSEMBLE, HDFXCE ASSEMBLE, HDFXCM ASSEMBLE, HDFXCO ASSEMBLE, HDFXDP ASSEMBLE, HDFXDP EXEC, HDFXGV ASSEMBLE, HDFXLB ASSEMBLE, HDFXLB EXEC, HDFXLC ASSEMBLE, HDFXLC EXEC, HDFXMG EXEC, HDFXMO ASSEMBLE, and HDFXTP ASSEMBLE to ensure the required functions are performed in them. The exits are described in chapter 20.3.4, “User Exits” on page 239.

If you wish to modify any of the user exits, then update the appropriate ASSEMBLE file, re-assemble, and replace the supplied version of the TEXT file with your local version. The user exit EXEC files may be modified directly.

Step 6: Check for Correct ACCESS Module

The following ACCESS modules are distributed:

- For CMS Rel 5.6 - ACCESS55 MODULE
- For CMS Rel 6 - ACCESS6 MODULE
- For CMS Rel 7 - ACCESS7 MODULE.

The correct ACCESS MODULE for the release of CMS which is running is automatically NUCXLOADed.

For CMS Release 8 and later, no special ACCESS module is needed. Access to mode zero files is controlled by ACCESSM0. A copy of ACCESSM0 MODULE should therefore be placed on the 194 minidisk of Interface.

Step 7: Run the BARSINST EXEC

The BARSINST EXEC was written to simplify the installation of the BARS/VM system by using prompts where appropriate. To perform installation, do the following:

1. Logoff the manager virtual machine (to detach its minidisks).
2. Logon to the interface virtual machine.
3. If using a minidisk, then:
   a. CP LINK the minidisk containing the BARS/VM distribution materials set up previously, as virtual address 100.
   b. Issue the CMS command `ACCESS 100 A`.

   Otherwise issue the CMS command `ACCESS dirid A`.
4. Enter BARSINST and reply to the prompts as apply to your installation.

The BARSINST EXEC carries out the following steps:

1. Format the minidisks belonging to the interface virtual machine.

   Note: Only the 195 work minidisk is formatted automatically. If you have allocated extra work minidisks in the range 1A5-1FF, then they will need to be formatted manually with a block size of 2K.
2. Load files onto the interface 191 minidisk. You have the option to modify the default PROFILE EXEC at this stage.
3. Load files onto the interface 194 minidisk. You have the option to update the file containing the installation options (see 20.3.2, “Nodename OPTIONS - Define the Installation Options” on page 226 for details of these options). Also, you are given the opportunity to modify the default PROFILE EXEC at this stage.
4. Load the user interface files onto the interface 19F minidisk and a public minidisk of your choice. You have the option to update the VMBARS PROFILE file containing the system and interface user ID, along with the VMBXDEFS and VMBXISPF EXECs, at this stage. See 20.4, “ISPF Panel Interface” on page 250 for details on tailoring these two EXECs.
5. Load the help files onto a public minidisk of your choice.
6. Format the minidisks belonging to the manager virtual machine.
7. Load files onto the manager A-disk and 194 minidisk. The A-disk can be a 191 minidisk or a SFS filespace. You have the option to modify the default PROFILE EXEC at this stage. In addition, you are given the opportunity to update the VMBARS RTABLE, MANAGER AUTHLIST and MANAGER OPTIONS files (see 20.3.5, “Manager Installation Options” on page 247 for details of these files).

Note: You may have to re-save CMS if you chose to place files on the system extension minidisk (Y-disk).

Once the BARSINST installation exec has completed you should LOGOFF the BARS/VM interface virtual machine.

Step 8: Initialize BARS/VM the First Time

You should logon to the user ID defined as the manager (EMUSERID in the installation options file), as you will need to issue certain restricted commands to start BARS/VM for the first time. BARS/VM is initialized simply by autologging the Interface user ID. It may be useful to spool the consoles of the various virtual machines initially to help in diagnosing any problems which may occur when first starting up the subsystem. Once it is running OK, this can be removed.

Once Interface has been logged on, the manager user ID should receive message HDFIN30011R, requesting a SUBUSERS DEFINE command. Issue this command and then BARS/VM should continue to initialize. Once initialization is complete and message HDFIN00100I has been received, issue the following commands to setup BARS/VM initially:

```
DIRECT
PASSWORD (only necessary where master passwords are to be used)
```

The scenario below illustrates the first time initialization process from the console of the manager user ID:

```
 HDFIN30011R SUBUSERS DEFINE command must be issued
vmbars subusers define vmbars1 nopass vmbars2 nopass vmbars3 nopass vmbars4 nopa
ss vmbars5 nopass vmbars6 nopass
Command issued
Ready;
 HDFIN30141I Subusers are being checked for validity
 HDFIN00113I Subusers defined and valid
 HDFIN00100I VMBARS initialization complete, session 1
vmbars direct dirmaint 193 user backup
Command issued
Ready;
 HDFDIR0012I Command accepted
```

Note: If DIRMAINT 1.4, or later, is in use (i.e. source directory is in cluster file format), then specify the USER BACKUP file on DIRMAINT 193 on the DIRECT command.

It is necessary to restart BARS/VM to activate some of these initial settings. Do this by issuing a BARS/VM SHUTDOWN command and then autolog the interface virtual machine again once it has shut down.

Step 9: Preparing for the Initial Backup Cycle

If SFS backup is required enroll user BARFS in the file pool defined by the EBFPOOL installation parameter. This file space is used for alias and authorization backup. Remember to grant all the subusers administration authority to this file pool.

Before the initial backup cycle is run, you should consider what volumes, file pools and minidisks are to be backed up. BARS/VM allows this to be controlled in the following ways:

The FILEPOOL command is used to specify which file pools are to be backed up.

The PRIMARY command is used to exclude minidisks on particular volumes from the backup process. You might use this to ensure that shared DASD is only backed up from one system, for example.

The BACKUP command is used to alter the backup parameters for minidisks. When issued with the NOBACKUP operand, BARS/VM will stop taking backups of that minidisk until a BACKUP command is reissued for it. When you issue a `BACKUP userID ALL ( NOBACKUP`, a dummy catalog file gets created to stop backup for all minidisks for that user ID. This would be used for "dummy" full pack extent minidisks which are used for dumping or charging purposes, for example.

Now issue the following commands if required:

```
AUTHORIZ command
PRIMARY commands
BACKUP ..... NOBACKUP commands
FILEPOOL commands
TAPE commands (if tape backup or archive is to be used)
TIME command
```

The scenario below illustrates the first time preparation process from the console of the manager user ID:

```
enroll user barfs barpool: (blocks 10000
Ready;
vmbars authoriz add maint
Ready;
vmbars display authoriz
Ready;
HDFDIS0004I DISPLAY command started
Current authorized users are:
User ID: MAINT    Node: local
HDFDIS0001I DISPLAY command complete
vmbars primary add cors* 3390 nobackup
Ready;
 Volser: CORS*  Devicetype: 3390   No backup added
vmbars backup $dumps all nobackup
Ready;
 HDFBAC0001I BACKUP command complete
vmbars tape add c12345 den 38000
Ready;
 HDFTCP0001I TAPE command complete
vmbars time 0030 0300
Ready;
First start time: 0030
First stop time: 0300
Second start time: 0000
Second stop time: 0000
vmbars display time
Ready;
HDFDIS0004I DISPLAY command started
The current START and STOP times are:
 first  cycle: 0030 to 0300
 second cycle: 0000 to 0000
HDFDIS0001I DISPLAY command complete
```

Step 10: Check for Correct Working of BARS/VM

The first backup cycle after BARS/VM has been installed begins at the time specified in the BARS/VM TIME command. Issuing a BARS/VM START NOW command will have no effect, as this will only restart an incomplete backup cycle. In addition, this initial backup cycle must be run before commands such as BPUT can be used.

It is recommended that a small backup window be selected with the BARS/VM TIME command, to run a "test" cycle. The messages sent to the manager user ID should be monitored to ensure that BARS/VM has been setup correctly. During the first cycle, if backup is to DASD, minidisks must be formatted, so this can take some time. Also note that the unit of work is either a real volume or a storage group. When the backup window ends, backup will not terminate until all existing units of work complete. So the first cycle will probably run past the backup stop time by a considerable amount. This also applies if a STOP NOW command is used to end the backup cycle.

The initial backup cycle has to backup every CMS file and each selected non-CMS minidisk on the system. This cycle will therefore take some time to complete. If the tape spill process is to be used, then it is strongly recommended that as much DASD space as possible be allocated as backup minidisks for the first cycle. This will greatly increase throughput and minimize tape mounts. If the duration of the cycle is not long enough to allow backup to complete, then you should consider emptying some (or all) backup minidisks to tape by using the START TAPEDUMP command. Do this before the commencement of each backup cycle. This will ensure the optimum performance during the cycle.

It is also advisable to define extra subusers just for the initial backup cycle, as this will also increase throughput. Once this cycle is complete, then these additional subusers can be deleted. If backup is to DASD, backup minidisks are allocated to subusers to try to ensure each subuser is writing to a different real volume whenever possible. This improves the performance of the initial backup run.

If backup is directly to tape, use can be made of Automatic Cartridge Loaders (ACLs) if available. These may be loaded with tapes and each tape drive attached to a subuser. The system may then be left to perform backup. A new tape will be automatically loaded when one becomes full until either the ACL is emptied or backup is complete.

During the initial backup cycle it is not advisable to incur the additional overhead of the process of reorganization. This is achieved by setting the EREORG installation option to a value greater than the number of days it will take to complete the initial cycle. This will obviously depend on the amount of data involved and the length of the cycle. Setting EREORG to a large value will ensure no reorganizations are performed. Once the initial backup is completed, then EREORG can be changed to whatever value is appropriate for your installation.

Note that BARS/VM will normally only perform one complete backup cycle per day. If a cycle completes and the TIME command is used to start backup again on the same day, no backup will be done. If a cycle fails to complete, the command START NOW will request BARS/VM to complete the cycle. The CATMAINT command with the LBDATE option may be used to set the date of last backup to a previous day and so allow a new backup cycle. However this may not result in a complete backup since individual minidisks and SFS directories will have been marked as backed up during the current day in the catalogs.

Step 11: Initialize ARCHIVE Function

The selection of the first archive minidisk is performed on receipt of the first APUT command (described in 8.2.5, “APUT” on page 59). If you are activating the archive component, then it is suggested that an initial APUT command is issued to verify successful installation.

See also chapter 12.3.1, “Selection of an ARCHIVE Volume” on page 156 for a discussion how archive minidisks and tapes will be selected.

Step 12: Operational Initialization of BARS/VM

To start BARS/VM in your operational environment, just include an 'AUTOLOG' command for the BARS/VM Interface user ID into your common start procedure for disconnected service machines. This usually will be done in the PROFILE EXEC of user ID 'AUTOLOG1', which is autologged automatically by CP at IPL time.

The Interface userID will attempt to autolog the manager userID during startup. If this does not work, you will also have to arrange for the manager virtual machine to be autologged by the same method as for Interface. This will ensure collection of BARS/VM status and error messages.

During the initialization process, the interface virtual machine will autolog the subusers as part of the initial verification.

Step 13: Modification of BARS/VM

BARS/VM is designed to run without any user modifications necessary. All installation dependent options may be specified in the OPTIONS file or by issuing the appropriate BARS/VM commands.

Chapter 22. Applying Maintenance

22.1 Updating Distributed Code

The distributed code is stored on either the Manager 291 minidisk or in a SFS directory. This is referred to as the distributed code disk.

How the files are placed on this disk is decided by the BARS/VM administrator at each installation. A suggested method is to manually place all the files in a SFS filespace and then define this in the MANAGER OPTIONS files so that it is accessed remotely by the Manager user ID.

22.2 Updating Executable Code

The manager REFRESH command can be used to replace executable files (EXEC, MACLIB, MODULE, TEXT, XEDIT, XMOD) on the Interface 194 and 19F and Manager 194 minidisks. This process will only proceed if the files on the distribution disk match the details recorded in the VMBARS CHKLIST file supplied. This check is intended to avoid the problem of an incomplete update being performed because of missing or back-level files.

Note: Remember that the manager user ID needs authority to link the Interface 194 and 19F minidisks R/W (LINK mode M) for this to work.

After updating the code, the BARS/VM subsystem or manager user ID needs to be restarted to pick up the changes. Use the KILLSUB command to restart the BARS/VM subsystem, and the REIPL command to restart the manager machine.

Executable files on any other minidisks (such as the VMBARS EXEC on the public minidisk) must be maintained by hand or with the BARSINST EXEC. However there are very few of these, and they seldom change.

22.3 Updating Control and Options Files

The remaining files (such as the Interface OPTIONS, MANAGER OPTIONS, MANAGER AUTHLIST and VMBARS PROFILE) are maintained manually. Link the appropriate minidisk R/W and edit the relevant files using XEDIT.

Warning: Take care not to change the record format or record length when editing these files.

After making changes, the BARS/VM subsystem or manager user ID needs to be restarted to pick up the changes. Use the KILLSUB command to restart the BARS/VM subsystem, and the REIPL command to restart the manager machine.

Part 6. Appendixes

Appendix A. Machine-Readable Material

Following is a complete list of files distributed with the BARS/VM tape:

- Sample directory entries, option file, etc

```
BARINTFC DIRECT
BARMANGR DIRECT
BARNOLOG DIRECT
BARSUSER DIRECT
PROF191  PROFILE
PROF194  PROFILE
SAMPLE   OPTIONS
SAMPLE   PROFILE
```

- Installation and operational EXEC files

```
BARSINST EXEC
CMSDDR   EXEC
CMSDDRR  EXEC
CMSDDRT  EXEC
HDFCNTRL EXEC
HDFDL1   EXEC
HDFFRM   EXEC
NONDDR   EXEC
PROF191  EXEC
PROF194  EXEC
RECOVERY EXEC
VMBARS   EXEC
VMBXISPF EXEC
```

- ISPF files for full screen user interface

```
VMBARS2  EXEC
VMBA01X  EXEC
VMBA02X  EXEC
VMBA03X  EXEC
VMBA04X  EXEC
VMBA05X  EXEC
VMBA06X  EXEC
VMBA07X  EXEC
VMBB01X  EXEC
VMBB02X  EXEC
VMBB03X  EXEC
VMBB04X  EXEC
VMBB05X  EXEC
VMBB06X  EXEC
VMBB07X  EXEC
VMBC00X  EXEC
VMBD01X  EXEC
VMBD02X  EXEC
VMBE00X  EXEC
VMBF01X  EXEC
VMBF02X  EXEC
VMBFATBL EXEC
VMBFBTBL EXEC
VMBISSUE EXEC
VMBNOPX  EXEC
VMBXDEFS EXEC
VMBHELP  MACLIB
VMBMSGS  MACLIB
VMBPANEL MACLIB
VMBARS   USRIFACE
VMBFRECV XEDIT
```

- Installation options and user exits

```
HDFCLS   ASSEMBLE
HDFXAC   ASSEMBLE
HDFXBP   ASSEMBLE
HDFXBP   EXEC
HDFXB1   ASSEMBLE
HDFXB2   ASSEMBLE
HDFXCA   ASSEMBLE
HDFXCB   ASSEMBLE
HDFXCE   ASSEMBLE
HDFXCM   ASSEMBLE
HDFXCO   ASSEMBLE
HDFXDP   ASSEMBLE
HDFXDP   EXEC
HDFXGV   ASSEMBLE
HDFXLB   ASSEMBLE
HDFXLB   SAMPEXEC
HDFXLC   ASSEMBLE
HDFXLC   SAMPEXEC
HDFXMG   EXEC
HDFXMO   ASSEMBLE
HDFXMO   SAMPEXEC
HDFXTP   ASSEMBLE
```

- NLS repository source and text

```
$HDFUME  REPOS
HDFUME   TEXT
```

- Dummy CSL module for CMS releases prior to release 6.

```
DMSCSL   TEXT
```

- HELP files

```
ADMINIST HELPMENU
VMBARS   HELPMENU
AALTER   HELPADMI
AAUDIT   HELPADMI
AMERGE   HELPADMI
ATRANS   HELPADMI
AUTHORIZ HELPADMI
BACKUP   HELPADMI
BUMPA    HELPADMI
CATMAINT HELPADMI
COLLECT  HELPADMI
DBR      HELPADMI
DBRTAPE  HELPADMI
DELETE   HELPADMI
DIRECT   HELPADMI
DISKINIT HELPADMI
```

```
DISPLAY  HELPADMI
EXTERNAL HELPADMI
FILEPOOL HELPADMI
HALT     HELPADMI
KILLSUB  HELPADMI
LSET     HELPADMI
MDISK    HELPADMI
NODENAME HELPADMI
OWNED    HELPADMI
PASSWORD HELPADMI
PRIMARY  HELPADMI
QUERY    HELPADMI
REMOVE   HELPADMI
RENAME   HELPADMI
REPLY    HELPADMI
RESTORE  HELPADMI
REUSE    HELPADMI
SHOWDISK HELPADMI
SHUTDOWN HELPADMI
START    HELPADMI
STOP     HELPADMI
SUBUSERS HELPADMI
TAPE     HELPADMI
TIME     HELPADMI
TRACE    HELPADMI
TWERP    HELPADMI
UNCOMPL  HELPADMI
VMBUTIL  HELPADMI
VOLMAINT HELPADMI
AALTER   HELPVMBA
ADEFINE  HELPVMBA
ADELETE  HELPVMBA
AGET     HELPVMBA
APUT     HELPVMBA
AQUERY   HELPVMBA
ARENAME  HELPVMBA
BACKUP   HELPVMBA
BDELETE  HELPVMBA
BLAH     HELPVMBA
BPUT     HELPVMBA
CQUEUE   HELPVMBA
DBR      HELPVMBA
DISPLAY  HELPVMBA
DQUERY   HELPVMBA
DRECOVER HELPVMBA
FILELIST HELPVMBA
LPUT     HELPVMBA
QUERY    HELPVMBA
RESTORE  HELPVMBA
RETRIEVE HELPVMBA
HDF0000S HELPMSG
HDF0001I HELPMSG
HDF0002S HELPMSG
HDF0003E HELPMSG
HDF0004I HELPMSG
HDF0006S HELPMSG
HDF0007S HELPMSG
HDF0008E HELPMSG
HDF0009S HELPMSG
HDF0010S HELPMSG
HDF0011R HELPMSG
HDF0012I HELPMSG
HDF0013S HELPMSG
HDF0014I HELPMSG
HDF0015E HELPMSG
HDF0016E HELPMSG
HDF0017E HELPMSG
HDF0018E HELPMSG
HDF0019S HELPMSG
HDF0100I HELPMSG
HDF0101E HELPMSG
HDF0102E HELPMSG
HDF0103I HELPMSG
HDF0104E HELPMSG
HDF0105E HELPMSG
HDF0106E HELPMSG
HDF0107R HELPMSG
HDF0108E HELPMSG
HDF0109E HELPMSG
HDF0110S HELPMSG
HDF0111R HELPMSG
HDF0113I HELPMSG
HDF0114S HELPMSG
HDF0120E HELPMSG
HDF0121R HELPMSG
HDF0122R HELPMSG
HDF0123R HELPMSG
HDF0124R HELPMSG
HDF0125R HELPMSG
HDF0126R HELPMSG
HDF0128S HELPMSG
HDF0130E HELPMSG
HDF0131E HELPMSG
HDF0132I HELPMSG
HDF0141I HELPMSG
HDF0151E HELPMSG
HDF0153E HELPMSG
HDF0154E HELPMSG
HDF0155E HELPMSG
HDF0156I HELPMSG
HDF0157I HELPMSG
HDF0158I HELPMSG
HDF0159E HELPMSG
HDF0160E HELPMSG
HDF0161I HELPMSG
HDF0162I HELPMSG
HDF0171I HELPMSG
HDF0172I HELPMSG
HDF0173S HELPMSG
HDF0174E HELPMSG
HDF0175I HELPMSG
HDF0176I HELPMSG
HDF0190E HELPMSG
HDF0201S HELPMSG
HDF0202S HELPMSG
HDF0230I HELPMSG
HDF0231I HELPMSG
HDF0240E HELPMSG
```

```
HDF0242E HELPMSG
HDF0243E HELPMSG
HDF0244E HELPMSG
HDF0250S HELPMSG
HDF0260E HELPMSG
HDF0270I HELPMSG
HDF0271I HELPMSG
HDF0272E HELPMSG
HDF0280E HELPMSG
HDF0281E HELPMSG
HDF0282E HELPMSG
HDF0283E HELPMSG
HDF0284E HELPMSG
HDF0290E HELPMSG
HDF0291E HELPMSG
HDF0293E HELPMSG
HDF0294E HELPMSG
HDF0295E HELPMSG
HDF0296I HELPMSG
HDF0297E HELPMSG
HDF0298E HELPMSG
HDF0299I HELPMSG
HDF0300E HELPMSG
HDF0301I HELPMSG
HDF0302I HELPMSG
HDF0303I HELPMSG
HDF0304E HELPMSG
HDF0305I HELPMSG
HDF0306I HELPMSG
HDF0307E HELPMSG
HDF0308I HELPMSG
HDF0309I HELPMSG
HDF0310E HELPMSG
HDF0311I HELPMSG
HDF0312I HELPMSG
HDF0313I HELPMSG
HDF0314I HELPMSG
HDF0315E HELPMSG
HDF0316E HELPMSG
HDF0317E HELPMSG
HDF0318I HELPMSG
HDF0319I HELPMSG
HDF0320E HELPMSG
HDF0321E HELPMSG
HDF0322E HELPMSG
HDF0323E HELPMSG
HDF0324E HELPMSG
HDF0325I HELPMSG
HDF0326I HELPMSG
HDF0327E HELPMSG
HDF0328I HELPMSG
HDF0329E HELPMSG
HDF0330E HELPMSG
HDF0331E HELPMSG
HDF0332E HELPMSG
HDF0333E HELPMSG
HDF0334E HELPMSG
HDF0335I HELPMSG
HDF0336I HELPMSG
HDF0337E HELPMSG
HDF0338I HELPMSG
HDF0339E HELPMSG
HDF0340E HELPMSG
HDF0341S HELPMSG
HDF0342E HELPMSG
HDF0343E HELPMSG
HDF0344E HELPMSG
HDF0345I HELPMSG
HDF0346I HELPMSG
HDF0347E HELPMSG
HDF0348E HELPMSG
HDF0349I HELPMSG
HDF0350I HELPMSG
HDF0351I HELPMSG
HDF0352I HELPMSG
HDF0353E HELPMSG
HDF0354I HELPMSG
HDF0355E HELPMSG
HDF0356E HELPMSG
HDF0357I HELPMSG
HDF0358I HELPMSG
HDF0359I HELPMSG
HDF0360I HELPMSG
HDF0362I HELPMSG
HDF0363E HELPMSG
HDF0364I HELPMSG
HDF0365E HELPMSG
HDF0366E HELPMSG
HDF0367I HELPMSG
HDF0368E HELPMSG
HDF0369I HELPMSG
HDF0370E HELPMSG
HDF0371I HELPMSG
HDF0372E HELPMSG
HDF0373E HELPMSG
HDF0374I HELPMSG
HDF0375I HELPMSG
HDF0376E HELPMSG
HDF0377E HELPMSG
HDF0378I HELPMSG
HDF0379E HELPMSG
HDF0380I HELPMSG
HDF0381E HELPMSG
HDF0382E HELPMSG
HDF0383E HELPMSG
HDF0384I HELPMSG
HDF0385E HELPMSG
HDF0386I HELPMSG
HDF0387I HELPMSG
HDF0388I HELPMSG
HDF0390E HELPMSG
HDF0391E HELPMSG
HDF0392E HELPMSG
HDF0393E HELPMSG
HDF0394E HELPMSG
HDF0395E HELPMSG
HDF0396E HELPMSG
HDF0397E HELPMSG
```

HDF0398E HELPMSG
HDF0399I HELPMSG
HDF0400E HELPMSG
HDF0401E HELPMSG
HDF0402E HELPMSG
HDF0403E HELPMSG
HDF0404E HELPMSG
HDF0405E HELPMSG
HDF0406I HELPMSG
HDF0408S HELPMSG
HDF0409I HELPMSG
HDF0410E HELPMSG
HDF0411I HELPMSG
HDF0412E HELPMSG
HDF0413E HELPMSG
HDF0414E HELPMSG
HDF0415I HELPMSG
HDF0416I HELPMSG
HDF0417E HELPMSG
HDF0418E HELPMSG
HDF0421I HELPMSG
HDF0422E HELPMSG
HDF0423E HELPMSG
HDF0424E HELPMSG
HDF0426W HELPMSG
HDF0427W HELPMSG
HDF0428E HELPMSG
HDF0429E HELPMSG
HDF0430E HELPMSG
HDF0431E HELPMSG
HDF0432I HELPMSG
HDF0433I HELPMSG
HDF0434E HELPMSG
HDF0435I HELPMSG
HDF0436I HELPMSG
HDF0437I HELPMSG
HDF0438E HELPMSG
HDF0439E HELPMSG
HDF0440I HELPMSG
HDF0441I HELPMSG
| HDF0442I HELPMSG
HDF0443I HELPMSG
HDF0444E HELPMSG
HDF0445I HELPMSG
HDF0446E HELPMSG
HDF0447I HELPMSG
HDF0448I HELPMSG
HDF0449I HELPMSG
HDF0450E HELPMSG
HDF0451E HELPMSG
HDF0452E HELPMSG
HDF0453E HELPMSG
HDF0454E HELPMSG
HDF0455E HELPMSG
HDF0456I HELPMSG
HDF0457S HELPMSG
HDF0458I HELPMSG
HDF0459E HELPMSG
HDF0460I HELPMSG
HDF0461I HELPMSG
HDF0462S HELPMSG
HDF0463S HELPMSG
HDF0464E HELPMSG
HDF0465E HELPMSG
HDF0466E HELPMSG
HDF0467E HELPMSG
HDF0468E HELPMSG
HDF0469I HELPMSG
HDF0470E HELPMSG
HDF0471I HELPMSG
HDF0472E HELPMSG
HDF0473I HELPMSG
HDF0474E HELPMSG
HDF0475S HELPMSG
HDF0476E HELPMSG
HDF0477I HELPMSG
HDF0478I HELPMSG
HDF0479E HELPMSG
HDF0480I HELPMSG
HDF0481I HELPMSG
HDF0483E HELPMSG
HDF0484E HELPMSG
HDF0485E HELPMSG
HDF0486E HELPMSG
HDF0487A HELPMSG
HDF0488S HELPMSG
HDF0489E HELPMSG
HDF0490E HELPMSG
HDF0491E HELPMSG
HDF0492E HELPMSG
HDF0493I HELPMSG
HDF0494E HELPMSG
HDF0495E HELPMSG
HDF0496E HELPMSG
HDF0497E HELPMSG
HDF0498E HELPMSG
HDF0499E HELPMSG
HDF0500I HELPMSG
HDF0501I HELPMSG
HDF0502I HELPMSG
HDF0503I HELPMSG
| HDF0504E HELPMSG
| HDF0505E HELPMSG
| HDF0506I HELPMSG
| HDF0507I HELPMSG
| HDF0508E HELPMSG
| HDF0509E HELPMSG
HDF0510I HELPMSG
HDF0511I HELPMSG
HDF0512W HELPMSG
HDF0513E HELPMSG
HDF0514I HELPMSG
HDF0515I HELPMSG
HDF0516E HELPMSG
HDF0517I HELPMSG
HDF0518E HELPMSG
HDF0519W HELPMSG
HDF0520I HELPMSG

```
| HDF0521E HELPMSG
  HDF0600I HELPMSG
  HDF0601I HELPMSG
  HDF0602I HELPMSG
  HDF0603E HELPMSG
  HDF0604I HELPMSG
  HDF0605E HELPMSG
  HDF0606E HELPMSG
  HDF0607E HELPMSG
  HDF0608S HELPMSG
  HDF0609E HELPMSG
  HDF0610S HELPMSG
  HDF0611S HELPMSG
  HDF0612E HELPMSG
  HDF0613I HELPMSG
  HDF0614I HELPMSG
  HDF0615I HELPMSG
  HDF0616I HELPMSG
  HDF0617I HELPMSG
  HDF0620E HELPMSG
  HDF0630E HELPMSG
  HDF0631I HELPMSG
  HDF0632E HELPMSG
  HDF0633E HELPMSG
  HDF0634I HELPMSG
  HDF0650S HELPMSG
  HDF0666T HELPMSG
  HDF0701E HELPMSG
  HDF0702E HELPMSG
  HDF0703E HELPMSG
  HDF0704E HELPMSG
  HDF0705E HELPMSG
  HDF0706I HELPMSG
  HDF0708I HELPMSG
  HDF0709I HELPMSG
  HDF0710I HELPMSG
  HDF0713S HELPMSG
  HDF0714E HELPMSG
  HDF0715I HELPMSG
  HDF0716E HELPMSG
  HDF0717E HELPMSG
  HDF0718E HELPMSG
  HDF0719E HELPMSG
  HDF0720E HELPMSG
  HDF0721E HELPMSG
  HDF0723S HELPMSG
  HDF0724S HELPMSG
  HDF0725I HELPMSG
  HDF0727I HELPMSG
  HDF0728E HELPMSG
  HDF0730I HELPMSG
  HDF0731I HELPMSG
  HDF0732I HELPMSG
  HDF0733E HELPMSG
  HDF0740I HELPMSG
  HDF0750I HELPMSG
  HDF0751E HELPMSG
  HDF0777E HELPMSG
  HDF0800E HELPMSG
  HDF0802E HELPMSG
  HDF0803E HELPMSG
  HDF0805I HELPMSG
  HDF0806I HELPMSG
| HDF0807I HELPMSG
  HDF0808S HELPMSG
  HDF0809S HELPMSG
  HDF0810E HELPMSG
  HDF0811I HELPMSG
  HDF0812E HELPMSG
  HDF0813E HELPMSG
  HDF0814I HELPMSG
  HDF0815E HELPMSG
  HDF0816E HELPMSG
  HDF0817I HELPMSG
  HDF0818E HELPMSG
  HDF0819S HELPMSG
  HDF0878E HELPMSG
  HDF0879E HELPMSG
  HDF0880E HELPMSG
  HDF0881E HELPMSG
  HDF0882E HELPMSG
  HDF0883E HELPMSG
  HDF0888E HELPMSG
  HDF0990S HELPMSG
  HDF0991I HELPMSG
  HDF0993I HELPMSG
  HDF0994S HELPMSG
  HDF0995S HELPMSG
| HDF0996T HELPMSG
  HDF0999E HELPMSG
  HDF1000E HELPMSG
  HDF1001E HELPMSG
  HDF1002E HELPMSG
  HDF1003E HELPMSG
  HDF1004E HELPMSG
  HDF1005E HELPMSG
  HDF1006E HELPMSG
  HDF1007E HELPMSG
  HDF1008S HELPMSG
  HDF1009E HELPMSG
  HDF1010E HELPMSG
  HDF1011E HELPMSG
  HDF1012E HELPMSG
  HDF1013E HELPMSG
  HDF1014E HELPMSG
  HDF1015I HELPMSG
  HDF1016I HELPMSG
  HDF1017I HELPMSG
  HDF1018I HELPMSG
  HDF1019I HELPMSG
  HDF1020I HELPMSG
  HDF1021I HELPMSG
  HDF1022I HELPMSG
  HDF1023I HELPMSG
  HDF1024E HELPMSG
  HDF1025E HELPMSG
  HDF1026E HELPMSG
  HDF1027I HELPMSG
```

```
HDF1028S HELPMSG
HDF1029E HELPMSG
HDF1030E HELPMSG
HDF1031I HELPMSG
HDF1032I HELPMSG
HDF1033I HELPMSG
HDF1034E HELPMSG
HDF1035E HELPMSG
HDF1036I HELPMSG
HDF1037I HELPMSG
HDF1038I HELPMSG
HDF1039I HELPMSG
HDF1040I HELPMSG
HDF1041I HELPMSG
HDF1042I HELPMSG
HDF1043I HELPMSG
HDF1044I HELPMSG
HDF1045E HELPMSG
HDF1046E HELPMSG
HDF1047E HELPMSG
HDF1048I HELPMSG
HDF1049I HELPMSG
HDF1050I HELPMSG
HDF1051I HELPMSG
HDF1052E HELPMSG
HDF1053E HELPMSG
HDF1054S HELPMSG
HDF1055I HELPMSG
HDF1060T HELPMSG
HDF1062T HELPMSG
HDF1063T HELPMSG
HDF1064T HELPMSG
HDF1065T HELPMSG
HDF1066T HELPMSG
HDF1067T HELPMSG
HDF1068W HELPMSG
HDF1069I HELPMSG
HDF1070I HELPMSG
HDF1071E HELPMSG
HDF1072E HELPMSG
HDF1073I HELPMSG
HDF1074I HELPMSG
HDF1075I HELPMSG
HDF1076I HELPMSG
HDF1077E HELPMSG
HDF1078I HELPMSG
HDF1079I HELPMSG
HDF1080I HELPMSG
HDF1081I HELPMSG
HDF1082I HELPMSG
HDF1084E HELPMSG
HDF1085I HELPMSG
HDF1086I HELPMSG
HDF1087E HELPMSG
HDF1088E HELPMSG
HDF1089E HELPMSG
HDF1090S HELPMSG
HDF1091I HELPMSG
HDF1092E HELPMSG
HDF1093I HELPMSG
HDF1110S HELPMSG
HDF1111S HELPMSG
HDF1112E HELPMSG
HDF1113E HELPMSG
HDF1114E HELPMSG
HDF1115E HELPMSG
HDF1154I HELPMSG
HDF1156E HELPMSG
HDF1157I HELPMSG
HDF1158I HELPMSG
HDF1159I HELPMSG
HDF1160E HELPMSG
HDF1161E HELPMSG
HDF1162I HELPMSG
HDF1163I HELPMSG
HDF1500E HELPMSG
HDF1501E HELPMSG
HDF1502E HELPMSG
HDF1503E HELPMSG
HDF1504I HELPMSG
HDF1505E HELPMSG
HDF1506E HELPMSG
HDF1507E HELPMSG
HDF1508E HELPMSG
HDF1509E HELPMSG
HDF1510E HELPMSG
HDF1511I HELPMSG
HDF1512I HELPMSG
HDF1513I HELPMSG
HDF1514I HELPMSG
HDF1515I HELPMSG
HDF1516E HELPMSG
HDF1516I HELPMSG
HDF1517I HELPMSG
HDF1518E HELPMSG
HDF1519E HELPMSG
HDF1520E HELPMSG
HDF1521E HELPMSG
HDF1522E HELPMSG
HDF1523E HELPMSG
HDF1524E HELPMSG
HDF1525E HELPMSG
HDF1526E HELPMSG
HDF1527E HELPMSG
HDF1528I HELPMSG
HDF1529I HELPMSG
HDF1530E HELPMSG
HDF1531I HELPMSG
HDF1532I HELPMSG
HDF4000I HELPMSG
HDF4001I HELPMSG
HDF4002I HELPMSG
HDF4003I HELPMSG
HDF4004I HELPMSG
HDF4005I HELPMSG
HDF4006I HELPMSG
HDF4007I HELPMSG
HDF4008I HELPMSG
```

```
HDF4009I HELPMSG
HDF4010I HELPMSG
HDF4011I HELPMSG
HDF4012I HELPMSG
HDF4013I HELPMSG
HDF4014I HELPMSG
HDF4015I HELPMSG
HDF4016I HELPMSG
HDF4017I HELPMSG
HDF4018I HELPMSG
HDF4019I HELPMSG
HDF4020I HELPMSG
HDF4021I HELPMSG
HDF4022I HELPMSG
HDF4023E HELPMSG
HDF4024I HELPMSG
HDF4025I HELPMSG
HDF4026I HELPMSG
HDF4027I HELPMSG
HDF4028I HELPMSG
HDF4029I HELPMSG
HDF4032I HELPMSG
HDF4040E HELPMSG
HDF4041I HELPMSG
HDF4042I HELPMSG
HDF4043I HELPMSG
HDF4044I HELPMSG
HDF4045I HELPMSG
HDF4046I HELPMSG
HDF4047I HELPMSG
HDF4048I HELPMSG
HDF4049I HELPMSG
HDF4050I HELPMSG
HDF4051I HELPMSG
HDF4052I HELPMSG
```

- Modules

```
ACCESS5  MODULE
ACCESS55 MODULE
ACCESS6  MODULE
ACCESS7  MODULE
DDR2CMS  MODULE
DDR2CMSX MODULE
DMSTAP   ESAMOD
DMSTAP   XAMOD
HDFCPY   ESAMOD
HDFCPY   MODULE
HDFCPY   XMOD
HDFDUM   MODULE
HDFMIM   MODULE
HDFSLC   MODULE
HDFTAP   ESAMOD
HDFTAP   XAMOD
HDFUCM   MODULE
VMBARS   MODULE
```

- Operational TEXT files

```
HDFABN   TEXT
HDFACK   TEXT
HDFACL   TEXT
HDFADE   TEXT
HDFADI   TEXT
HDFALT   TEXT
HDFAMG   TEXT
HDFAQE   TEXT
HDFARE   TEXT
HDFARP   TEXT
HDFAR1   TEXT
HDFAUD   TEXT
HDFAUT   TEXT
HDFAXE   TEXT
HDFBAA   TEXT
HDFBAC   TEXT
HDFBCC   TEXT
HDFBCE   TEXT
HDFBCK   TEXT
HDFBCT   TEXT
HDFBCV   TEXT
HDFBCX   TEXT
HDFBCY   TEXT
HDFBC0   TEXT
HDFBC1   TEXT
HDFBC2   TEXT
HDFBC5   TEXT
HDFBC6   TEXT
HDFBDE   TEXT
HDFBLA   TEXT
HDFBOO   TEXT
HDFBPD   TEXT
HDFBPN   TEXT
HDFBPT   TEXT
HDFBP1   TEXT
HDFBP2   TEXT
HDFBP3   TEXT
HDFBP4   TEXT
HDFBP5   TEXT
|HDFBP7   TEXT
HDFBUF   TEXT
HDFBUM   TEXT
|HDFCCA   TEXT
|HDFCCB   TEXT
|HDFCCP   TEXT
HDFCEX   TEXT
HDFCHK   TEXT
HDFCKP   TEXT
HDFCKR   TEXT
HDFCLS   TEXT
HDFCMD   TEXT
HDFCOL   TEXT
HDFCO1   TEXT
HDFCSF   TEXT
HDFCTW   TEXT
HDFCVD   TEXT
HDFCVI   TEXT
HDFDAB   TEXT
HDFDAT   TEXT
```

```
HDFDBL   TEXT
HDFDBP   TEXT
HDFDBQ   TEXT
HDFDBR   TEXT
HDFDCA   TEXT
HDFDCC   TEXT
HDFDCP   TEXT
HDFDCR   TEXT
HDFDC0   TEXT
HDFDC1   TEXT
HDFDC2   TEXT
HDFDC3   TEXT
HDFDC4   TEXT
HDFDEF   TEXT
HDFDEL   TEXT
HDFDIR   TEXT
HDFDIS   TEXT
HDFDI2   TEXT
HDFDI3   TEXT
HDFDI4   TEXT
HDFDL1   TEXT
HDFDQU   TEXT
HDFDRE   TEXT
HDFDR0   TEXT
HDFDR1   TEXT
HDFDR2   TEXT
HDFDR3   TEXT
HDFDT0   TEXT
HDFDU0   TEXT
HDFDU1   TEXT
HDFDU2   TEXT
HDFDU3   TEXT
HDFDU4   TEXT
HDFEQU   TEXT
HDFEXT   TEXT
HDFFCP   TEXT
HDFFMD   TEXT
HDFFOR   TEXT
HDFFPR   TEXT
HDFGET   TEXT
HDFGE2   TEXT
HDFGE3   TEXT
HDFHDR   TEXT
HDFHLT   TEXT
HDFINA   TEXT
HDFINB   TEXT
HDFINC   TEXT
HDFIND   TEXT
HDFINE   TEXT
HDFINM   TEXT
HDFINR   TEXT
HDFINS   TEXT
HDFINT   TEXT
HDFINZ   TEXT
HDFIN0   TEXT
HDFIN1   TEXT
HDFIN2   TEXT
HDFIN3   TEXT
HDFIN4   TEXT
HDFLID   TEXT
HDFLIS   TEXT
HDFLRW   TEXT
HDFMAT   TEXT
HDFMDD   TEXT
HDFMDG   TEXT
HDFMDK   TEXT
HDFMDM   TEXT
HDFMDP   TEXT
HDFMDQ   TEXT
HDFMDR   TEXT
HDFMSG   TEXT
HDFNAM   TEXT
HDFNCP   TEXT
HDFNKO   TEXT
HDFNOD   TEXT
HDFOWN   TEXT
HDFOWT   TEXT
HDFPRM   TEXT
HDFPRN   TEXT
HDFPRO   TEXT
HDFPSS   TEXT
HDFPUT   TEXT
HDFPU1   TEXT
HDFPU2   TEXT
HDFPU3   TEXT
HDFPU4   TEXT
HDFPU5   TEXT
HDFPU6   TEXT
HDFPU7   TEXT
HDFPWU   TEXT
HDFQDT   TEXT
HDFQUD   TEXT
HDFQUE   TEXT
HDFQU1   TEXT
HDFRDI   TEXT
HDFREM   TEXT
HDFREN   TEXT
HDFREP   TEXT
HDFRES   TEXT
HDFRET   TEXT
HDFREU   TEXT
HDFRE1   TEXT
HDFRE2   TEXT
HDFRE4   TEXT
HDFRE5   TEXT
HDFRE6   TEXT
HDFRE7   TEXT
HDFRM1   TEXT
HDFRM2   TEXT
HDFRRC   TEXT
HDFRRW   TEXT
HDFRSA   TEXT
HDFRSC   TEXT
HDFRSD   TEXT
HDFRSF   TEXT
HDFRSO   TEXT
HDFRST   TEXT
HDFRTN   TEXT
```

```
HDFSAV   TEXT
HDFSA6   TEXT
HDFSA7   TEXT
HDFSCN   TEXT
HDFSD6   TEXT
HDFSHD   TEXT
HDFSHO   TEXT
HDFSH1   TEXT
HDFSH2   TEXT
HDFSH3   TEXT
HDFSLC   TEXT
HDFSMB   TEXT
HDFSMC   TEXT
HDFSMR   TEXT
HDFSND   TEXT
HDFSPC   TEXT
HDFSPM   TEXT
HDFSPX   TEXT
HDFSP2   TEXT
HDFSP3   TEXT
HDFSP4   TEXT
HDFSP5   TEXT
HDFSP6   TEXT
HDFSP8   TEXT
HDFSTO   TEXT
HDFST0   TEXT
HDFST6   TEXT
HDFSUA   TEXT
HDFSUD   TEXT
HDFSUE   TEXT
HDFSUL   TEXT
HDFSUR   TEXT
HDFSUS   TEXT
HDFSUT   TEXT
HDFSUU   TEXT
HDFSU0   TEXT
HDFSU2   TEXT
HDFSU4   TEXT
HDFSU5   TEXT
HDFSU6   TEXT
HDFSU7   TEXT
HDFSU8   TEXT
HDFSU9   TEXT
HDFSWU   TEXT
HDFTCP   TEXT
HDFTFC   TEXT
HDFTIM   TEXT
HDFTOT   TEXT
HDFTPM   TEXT
HDFTRA   TEXT
HDFTTT   TEXT
HDFUCB   TEXT
HDFUCO   TEXT
HDFUCS   TEXT
HDFUNC   TEXT
HDFUTM   TEXT
HDFUT1   TEXT
HDFUT2   TEXT
HDFVCP   TEXT
HDFVMC   TEXT
HDFVMN   TEXT
HDFVM1   TEXT
HDFVM2   TEXT
HDFWKD   TEXT
HDFWRU   TEXT
HDFWTO   TEXT
HDFWUE   TEXT
HDFXAC   TEXT
HDFXBP   TEXT
HDFXB1   TEXT
HDFXB2   TEXT
HDFXCA   TEXT
HDFXCB   TEXT
HDFXCE   TEXT
HDFXCM   TEXT
HDFXCO   TEXT
HDFXDP   TEXT
HDFXGV   TEXT
HDFXLB   TEXT
HDFXLC   TEXT
HDFXMO   TEXT
HDFXTP   TEXT
HDF91W   TEXT
VMHDFS   TEXT
```

- Manager execs and control files

```
MANAGER  AUTHLIST
VMBARS   CHKLIST
AIDUSER  EXEC
ARCLOG   EXEC
CHKAUTH  EXEC
HDFDISC  EXEC
LOADOPTS EXEC
MIDNIGHT EXEC
NEWDAY   EXEC
PROCVACK EXEC
PROFMGR  EXEC
PROPLGER EXEC
PROPSTOP EXEC
RDRINT   EXEC
REACCESS EXEC
REFRESH  EXEC
REIPL    EXEC
SENDREP  EXEC
SHUTDOWN EXEC
WRITELOG EXEC
MANAGER  OPTIONS
VMBARS   RTABLE
```

Rename the PROFMGR EXEC file to PROFILE EXEC on the manager A-disk.

Index

Numerics

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

Program Number: 5785-DJD

Printed in Hursley Park

SB11-8974-03