SYSTEM REFERENCE GUIDE

HCC/MVS 3.0

HOST

COMMUNICATION

CONTROL

11th edition

Document number: 600233-A

IMPRINT

11th Edition

This Manual has been written with utmost care. Textual or formal errors still cannot be excluded!

Protected trademarks are not marked as such in this Manual. The fact that these trademarks are not shown does not imply that the trade names are free for use.

Subject to changes without notice.

Extensions and changes to this Manual are based on PTF with status ZY00113, in as much as software changes were made at the same time.

Publisher: EMASS/GRAU Storage Systems GmbH,
Eschenstraße 3, D-89558 Böhmenkirch

11th Edition in August 1997

CONTENTS

1 SYSTEM REFERENCE GUIDE for HCC/MVS 3.0

1.1 LITERATURE

Manual	Order number	Reference
General Information Manual	600226-A	G
Installation/Customization Guide	600227-A	G
System Reference Guide	600233-A	G
Operators Guide	600231-A	G
Command Reference	600223-A	G
Conversion Notebook	600224-A	G
Installation/Customization Reference	600228-A	G
Messages and Codes	600230-A	G
ISPF User Guide	600229-A	G
Release Guide	600232-A	G

G This Manual is part of the standard HCC documentation. Further copies of the respective Manual can be obtained from the publisher.

2 ABBASEND USER INTERFACE

2.1 OVERVIEW - ABBASEND

ABBASEND is a service function that supports communication with AML over HCC from any MVS address area.

ABBASEND can be used

- in BATCH,
- on-line,
- [as subprogram].

When ABBASEND is used as subprogram, the HACPARM1 definitions must however be allocated as HACCABSE instead of HACCPARM, and the user must release these datasets. There is **no** CLOSE=FREE.

ABBASEND uses Cross-Memory functions and can therefore only be called under APF authorization.

ABBASEND was extended considerably for Release HCC 2.3.1. You can use the new functions directly or continue to use ABBASEND as before.

The command scope depends on the HCC environment (Primary/Secondary system).

2.2 ABBASEND - BATCH (OLD)

2.2.1 CALLING ABBASEND - (OLD)

ABBASEND can be called as a program in batch with the old functional scope in three ways:

```
//stepname  EXEC PGM=ABBASEND,PARM='cmd'
//STEPLIB   DD   DSN=hlq.ZHC....LOAD,DISP=SHR
```

or

```
//stepname  EXEC PGM=ABBASEND
//STEPLIB   DD   DSN=hlq.ZHC....LOAD,DISP=SHR
//PARMIN    DD   *
cmd 1
cmd 2
cmd nnn
/*
```

or

```
//stepname  EXEC PGM=ABBASEND
//STEPLIB   DD   DSN=hlq.ZHC....LOAD,DISP=SHR

//PARMIN    DD   DSN=dsname,DISP=SHR
```

cmd = valid HCC command

The PARMIN file must be created with
`LRECL=80,RECFM=FB,BLKSIZE=n*80`

ABBASEND terminates with USER-ABEND 2201; exception by **EJECT** command, when HCC returns a code > 0.

2.2.2 ABBASEND WAIT OPTION (OLD)

The WAIT operand can be appended to every command called using ABBASEND (for integrity reasons).

Example:

```
EJDSN dsname WAIT
```

The WAIT operand causes ABBASEND to wait until the command is accepted. The acceptance/rejection is signaled by an appropriate return code. ABBASEND does **not** wait for the command to be executed; only PARM=NEW has this effect.

The above example ensures that the desired eject statement is integrated in the HCC Recovery and does not get lost.

The ABSWAIT=YES parameter in the HACCPARM member enables the WAIT option for all commands issued in the old command format.

It is not necessary to call, for example, display commands under TSO with the WAIT option, it can however be useful when insufficient TCAS-VTIOC buffer has been defined.

2.3 ABBASEND - BATCH (NEW)

2.3.1 CALLING ABBASEND (NEW)

```
//stepname    EXEC  PGM=ABBASEND,
//                  PARM='NEW,CMD=cmd,RESULTS=res,WAIT=wait'
//STEPLIB     DD    DSN=hlq.ZHC....LOAD,DISP=SHR
//HACCPARM    DD    DSN=hlq.ZHC....(HACPARM1)
//HACCOUT     DD    SYSOUT=*
```

or

```
//stepname    EXEC  PGM=ABBASEND,PARM=´NEW´
//STEPLIB     DD    DSN=hlq.ZHC....LOAD,DISP=SHR
//HACCPARM    DD    DSN=hlq.ZHC....(HACPARM1)
//HACCOUT     DD    SYSOUT=*
//PARMIN      DD    *
CMD=cmd,RESULTS=res,WAIT=wait
        .
        .
        .
/*
```

or

```
//stepname    EXEC  PGM=ABBASEND,PARM=´NEW´
//STEPLIB     DD    DSN=hlq.ZHC....LOAD,DISP=SHR
//HACCPARM    DD    DSN=hlq.ZHC....(HACPARM1)
//HACCOUT     DD    SYSOUT=*
//PARMIN      DD    DSN=dsname,DISP=SHR
```

cmd = valid HCC command

The PARMIN file must be created with
`LRECL=80,RECFM=FB,BLKSIZE=n*80`

A sequential file with LRECL=132 can be used for HACCOUT instead of a SYSOUT file.

2.3.2 ABBASEND - PARM STATEMENTS

```
PARM='NEW,CMD=command,RESULTS=res,WAIT=wait'
```

NEW		Calling the new functions
CMD =	'command'	Valid HCC command
res =	YES	Output desired
	NO	No output desired
		Default = YES
wait =	YES	Waits for the initiated function to complete
	NO	Does not wait for the initiated function to complete
		Default = YES

The ABSWAIT parameter in HACCPARM has no significance for the new ABBASEND.

The combination RESULTS=YES with WAIT=NO is invalid and is automatically set to RESULTS=NO and WAIT=NO.

2.4 ABBASEND - TSO

2.4.1 CALLING ABBASEND UNDER TSO

```
ABBASEND cmd
```

or

```
ABBASEND MONITOR
HAC109A: ENTER HACC-COMMAND
cmd 1
cmd 2
END
```

2.5 ABBASEND - ISPF INTERFACE (NEW)

2.5.1 HCC/ISPF INTERFACE

The ISPF interface comprises various applications for setting software parameters and controlling the software.

These applications are:

- Generating HACPARM1 using ISPF tables
- Calling the HCC Archive administration (HAA)
- Controlling HCC using commands and displaying results

The ISPF User Guide describes how to install and use the interface.

3 TSO MONITOR SESSION

The following commands can only be used with ABBASEND under TSO. ABBASEND must be called with the MONITOR option.

DCSA is the only function that can be called directly with ABBASEND. Both options are only available to authorized users.

3.1 MONITOR

```
MONITOR
```

MONITOR provides the full functional scope for primary HCC systems and only restricted functions for secondary systems.

Example:

```
ABBASEND MONITOR
```

The following message appears:

```
HAC109A ENTER HACC COMMAND
```

3.2 DISPLAY COMMON SERVICE AREA

```
DCSA
```

DCSA is a function independent from HCC activities that directly displays certain fields of the HACC CVT in the Common Service Area (CSA); the 10 current buffers are displayed.

Exception: no display immediately after IPL and before HCC has been started.

This function is planned to provide support in HCC error situations.

Example:

```
ABBASEND DCSA
```

```
ABS 001                    HACC CSA INFORMATIONS                        V3R00M00
═══════════════════════════════════════════════════════════════════════════════
HACCCVT ENTRY ADDRESS ..: 00B717C0              SSINAME.............: HAC0
CSA-SIZE (BYTES HEX/DEC): E4001840/06208        SUBPOOL NUMBER ....: 228
SSCVT ADDRESS ..........: 00C02508              WTOEXIT EP-ADDRESS : 00B72348
HACC SYSTEM ............:          NOT READY    HACC ASCB-ADDRESS .: 00000000
ERROR INFO COMPCODE: 00000000    PSW-ADDRESS: 00000000     OFFSET: 00000000
-------------------------------------------------------------------------------
WTOEXIT     WAITCOUNT: 00000    TOTAL EVENTS: 0000003      STATUS: NOT READY
ABBASEND    WAITCOUNT: 00000    TOTAL EVENTS: 0000000      STATUS: NOT READY
-------------------------------------------------------------------------------
BUFNO SYNC-FLD JOBNAME  MESSAGE
  1   00000000 RS023001 IEF234E K 48D,011837,PVT,RS023001,ST033
  2   00000000 RS348640 IEC507D REPLY ´U´-USE OR ´M´-UNLOAD
  3   00000000 RZP2     DU
  4   00000000 RS02TEST IEC501A M 48F,PRIVAT,,RS02TEST,HA02.P0551.G0123V00
  5   00000000 ISC033   EJDSN HA33.SOS.DB33(0) WAIT
  6   00000000
  7   00000000
  8   00000000
  9   00000000
 10   00000000
═══════════════════════════════════════════════════════════════════════════════
ENTER:_  <-- E=END(PFK12),C=CONT
```

The CSA recovery area is displayed when "C" is input:

```
ABS 002                    RECOVERY AREA CONTENTS
═══════════════════════════════════════════════════════════════════════════
UNIT HC-SQ MO-SQ KE-SQ JOBNAME  MESSAGE
 480   531   531   477 RZP3S3   IEF433A M 480,PRIVAT,,,,RZP.MVS130.G0033V00
 481   499   484   499 PD173301 IEF433E K 481,010163,SL
 482   528   528   493 PD024002 IEC705I TAPE ON 482,004023,RZP.MVS103,G0019V00
 483   534   530   534 RZP3S4   IEF433A M 483,PRIVAT,,,,RZP.MVS124.G0009V00
 484   ...
 485   ...
 ...   ...
 ...   ...
 48A   ...
 48B   ...
═══════════════════════════════════════════════════════════════════════════
ENTER: _  <-- E=END(PFK12),R=REPEAT
```

Significance:
HC-SQ = HCC recovery sequence number
MO-SQ = Mount recovery sequence number
KE-SQ = Keep recovery sequence number

The recovery sequence number is generated in the SSI Exit.

Every MVS message received is issued a serial number and HCC can then use the UNIT recovery log to reconstruct the current situation within an AML system after a system crash.

3.3 START OPERATOR SESSION

```
SOP
```

This command routes, supplementary to **MONITOR**, all messages normally routed to the (master) console to the TSO monitors as well.

This message routing remains in effect after the ABBASEND session is closed. HCC messages still effect a terminal break.

This function can be used to monitor events without hindering the productivity of the TSO user.

All HCC messages are displayed when TRON is set.

Example:

```
ABBASEND MONITOR
```

The following message appears:

```
HAC109A ENTER HACC COMMAND

SOP
TRON
...
```

3.4 STOP OPERATOR SESSION

```
POP
```

The **POP** command resets the **SOP** command.

The MONITOR session is not affected apart from console message routing.

Example:

```
POP
```

3.5 START PERMANENT DISPLAY

```
SPD
```

Calling **SPD** displays a TSO full screen display that automatically refreshes after each AML action and always shows the current processing status.

When more tape units are assigned to an AML system that can be displayed on the terminal, an entry will be updated when it is displayed. When an entry is not shown on the screen, the oldest entry (bottom line) is deleted and the new entry set in the top line.

Example:

```
ABBASEND MONITOR
SPD
```

The monitor screen is described on the following page.

Only CAR type units are displayed!

```
CUU AB CPU  JOBNAME  VOLSER SQNR MOUNTS CLMO TIME MO  KE  PM CL  ERR. SENDTIME
═══════════════════════════════════════════════════════════════════════════
440 01 SYS1 RE115J02 012345 1760    105   31   39 QSM Q               12.10.12
441 02 SYS1 T02HK11  010123 1768     98   29   39 QS              N304 12.07.45
442 03 PROD V0251002 010105 1763    105   35   40 Q
443 04 PROD V2321604 010102 1762    111   18   39 QSM Q   PM          12.12.14
448 05 PROD V2930001 010133 1765     94   40   39 Q
449 06 PROD V0420002 012101 1761     85   19   39 QSM QS  PM          12.12.47
44A 07 PROD                          66   28   38
44B 08 SYS1 ZBSBE703 013044 1703     96 C 52   39 QSM                 12.22.09
44C 09 PROD                          49    0   39
44D 10 SYS! HACC     010000 1766     77 C 50   38 QSM       CL        12.11.50



000 00      *EJECT*  010115         202        34 QS     K             11.14.04
═══════════════════════════════════════════════════════════════════════════
ENTER PA1-KEY TO TERMINATE _
```

The fields have the following significance:

CUU	MVS address of the tape unit
AB	Assigned AML unit number
CPU	System that reserved the tape unit
SQNR	Send sequence number
MOUNTS	Number of mounts since the last **CC** command (refer to **CC**)
CLMO	Number of mounts since the last tape unit cleaning C = unit marked for cleaning as soon as no more mounts are pending
TIME	Average AML replay time (send time-positive reply)
MO	Mount status: Q = in Robque, S = sent, M = mounted
KE	Keep status: Q = in Robque, S = sent, K = mounted
PM	Mount pending, that is, a further mount request is already pending
CL	Cleaning process active
ERR	Error has occurred (in this case: N304=label not readable) (refer to **DRE**)
SENDTIME	Send time of the individual **MOUNT**/**KEEP** process

Interruption with the PA1 key on local terminals and with the SYS.REQUEST key on remote terminals (Standard Attention Exit).

4 AUTHORIZATION OF COMMANDS

4.1 OVERVIEW - AUTHORIZATION OF COMMANDS

ABBASEND and HAA are TSO commands , which can interrupt production processes under HCC/AML when used without due consideration. Only authorized persons should be able to issue these commands.

The authorization selection can be made using

- simple specifications using the SECURITY statement in HACPARM1
- SAF authorization call.

4.2 INTERNAL AUTHORIZATION

As described under SECURITY, it is easy to establish access control for certain TSO users.

This procedure is independent of other installed security systems, such as RACF for example.

Changes made with the SECURITY statement are however first effective after the next HCC start.

4.3 RACF AUTHORIZATION CALL

RACF can be used to protect the functions HAA, ISPF, ABBASEND-TSO(MONITOR) and ABBASEND Batch.

The following definitions must be made in order to use the RACF interface (SAF interface).

4.3.1 RACF Class Descriptor Table

A resource class for HCC must be defined within RACF. The ICHRRCDE module can be used to define installation-specific resource classes.

The resource class for HCC must be included here when this module already exists. Otherwise the following definitions can be used.

The resource class must be named according to the naming conventions specified in the RACF SPL documentation ($HACC in the following example).
An example can be found in hlq.ZHC...SAMP: ICHRRCDE + JICHRRCD

```
//ASM.SYSIN DD *
$HACC    ICHERCDE CLASS=$HACC,                                  +
               ID=128,                                          +
               MAXLNTH=39,                                      +
               FIRST=ALPHANUM,                                  +
               OTHER=ANY,                                       +
               POSIT=19,                                        +
               OPER=NO,                                         +
               RACLIST=ALLOWED
         ICHERCDE
//LKED.SYSIN DD *
 ORDER $HACC
 ORDER ICHRRCDE       <-- must be the last ORDER statement
 NAME ICHRRCDE(R)
```

4.3.2 SAF Router Table

The SAF Router table must be created or modified (refer to the RACF SPL documentation).

An example can be found in hlq.ZHC...SAMP: ICHRFR01 + JICHRFR.

```
//ASM.SYSIN DD *
ICHRFR01 CSECT
$HACC    ICHRFRTB CLASS=$HACC,ACTION=RACF
ENDTAB   ICHRFRTB TYPE=END
         END   ICHRFR01
//LKED.SYSIN DD *
 NAME ICHRFR01(R)
```

4.3.3 SETROPTS Definitions

The HCC Resource class is activated with the RACF-TSO command **SETROPTS** or with the RACF ISPF panel.

```
SETROPTS CLASSACT($HACC)
```

As from RACF 1.8, the complete HCC Resource class can be loaded into memory with the **SETROPTS** command in order to save I/O accesses during the authorization checks.

```
SETROPTS RACLIST($HACC)
```

4.3.4 RESOURCE Definitions

A resource name prefix must be defined for HCC. The single resource names are then made up of this prefix plus the name of the HCC function. This prefix is set to $HACC.CMD for the following example. The resource name for the **EJECT** command would then be $HACC.CMD.EJ.

This resource is then defined in RACF as follows:

```
RDEFINE $HACC ($HACC.CMD.EJ) UACC(NONE) ...
```

Two exceptions must be considered here. The definition for HAA must be as follows:

```
RDEFINE $HACC ($HACC.CMD.HAA.*) UACC(NONE)
```

An additional definition is needed for ABBASEND to be able to use ABBASEND at all:

```
RDEFINE $HACC ($HACC.CMD.ABS) UACC(NONE)
```

All other commands are protected by the command definitions described above.

4.3.5 AUTHORIZATIONS

The RACF command **PERMIT** allocates authorizations. The access permission READ for commands and ABBASEND is sufficient here because the check only differentiates between. NONE (not allowed) and READ (allowed).

```
PERMIT $HACC.CMD.EJ CLASS($HACC) -
       ACCESS(READ) ID(user-id group ...)
```

ACCESS(UPDATE) must also be authorized for HAA to process the HCC Archive with the Archive administration.

If the resource class is raclisted, **REFRESH** must be made for this class.

```
SETROPTS RACLIST($HACC) REFRESH
```

4.3.6 ACTIVATING THE RACF CHECKS IN HACC

The RACF checks are enabled by specifying:

```
SECURITY TYPE=EXT,NAME=RACF,CLASS=$HACC,
         RESPREF=$HACC.CMD,HLQHAA=uid
```

in HACPARM1 and then restarting HCC.

5 CATALOG INSTALLATION EXIT ZHC026DU

5.1 OVERVIEW INSTALLATION EXIT ZHC026DU

This Section describes how to influence the MVS Allocation using the **CATALOG INSTALLTION EXIT** ZHC026DU. This step is only necessary when the general MVS allocation is to be influenced for several independent tape pools.

This exit **cannot** influence allocations for products which also use the SVC26 exit (for example, DF/HSM)

5.2 INFLUENCING THE MVS ALLOCATION

It is often necessary to influence the MVS allocation. It is then necessary for supporting more than one tape pool with the same unit type (UCBTYP). For example, two completely separate AML systems (refer to the Figure below) or one AML system are operated with a manual cartridge pool run parallel (migration phase).

The MVS allocation is normally influenced with the UNIT parameter in Job Control or by dynamic allocation. The file disposition must be considered. A UNIT or UNITTYP specification is necessary for files to be created, for example,

```
//TAPEOUT  DD  DSN=dsname,DISP=(NEW,CATLG),UNIT=POOL2,...
```

whereas the UNITTYP for existing files is automatically substituted from the system catalog:

```
//TAPEIN   DD  DSN=dsname,DISP=OLD
```

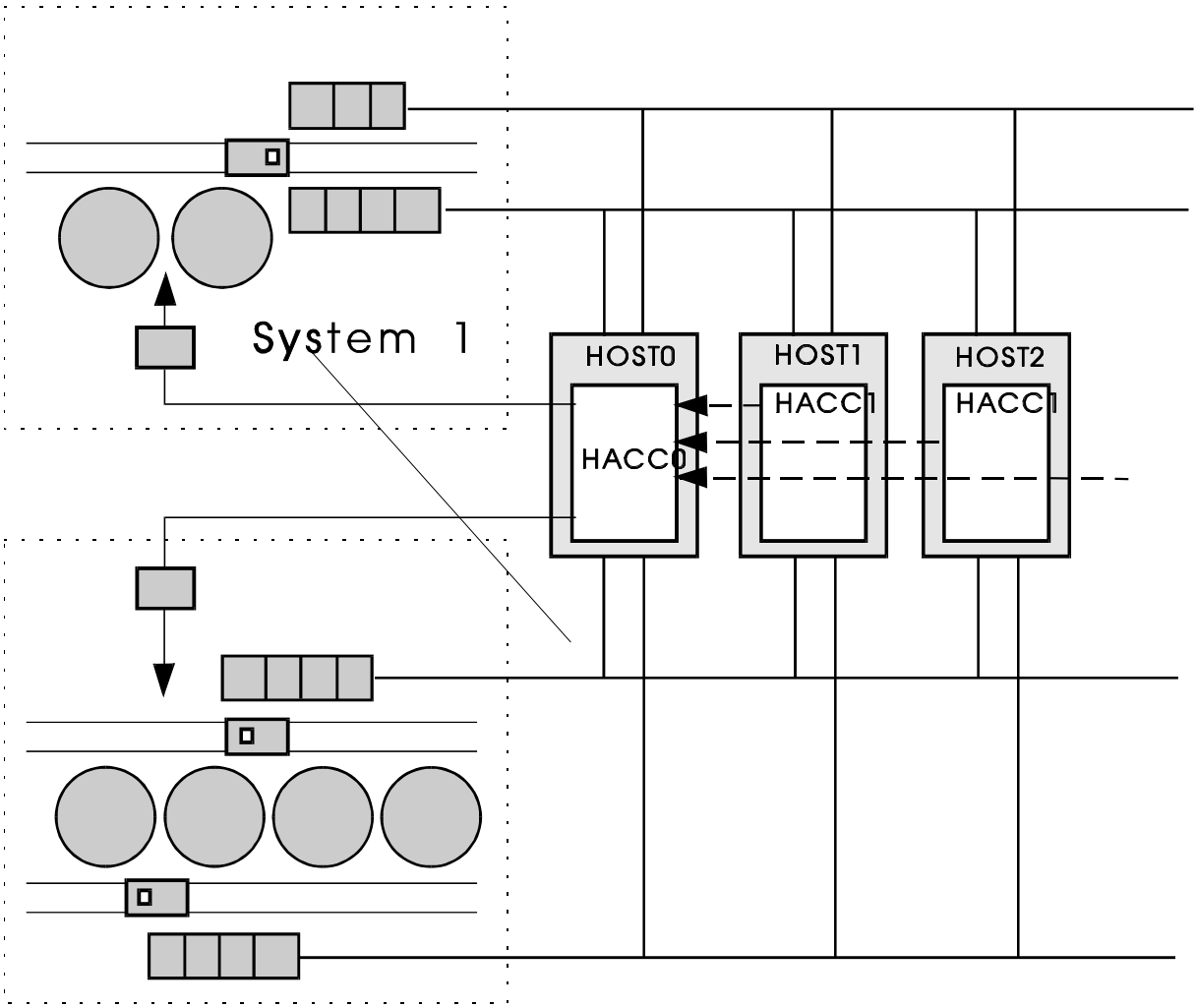


Fig. 1: Two separate tape pools

The following problem then arises: the information in the system catalog is restricted to the general UNITTYP (UCBTYP) and the ESOTERIC NAME in the UNIT parameter of the JCL is not retained when the file is created which means that the pool specification where the cartridge was created is lost.

The lost location information (for example, POOL1 or POOL2) must be supplied to MVS when such a cartridge is requested later, for example in the form

```
//TAPEIN   DD  DSN=dsname,DISP=OLD,UNIT=POOL2
```

This specification leads to a JCL error because VOL=SER= is required when the UNIT parameter refers to an "OLD" file. When the pool is not specified, MVS performs the allocation according to its own rules which means that cartridges from POOL1 (building 1) could be requested from POOL2 (Building 2) and vice versa.

In earlier MVS versions (before SU 60) the ESOTERIC NAME was stored direct as UCB LOOKUP VALUES instead of the UCBTYP (for example, for 3480 UCBTYP=X'78008080'). The location information was thus retained in the catalog. The disadvantage was that all affected files had to be recataloged when appended with ESOTERIC NAMEs because this modified the LOOKUP VALUE.

This location problem is solved automatically when differing unit types are installed in POOL1 and POOL2, for example, standard 3480 units in POOL1 (UCBTYP=X'78008080') and in POOL2 3480 units with compaction feature (UCBTYP=X'78048080' data conversion bit).

The following method has been implemented under HCC for multiple tape pools:

The lost location information is reconstructed in the CATALOG INSTALLATION EXIT IGG026DU using a table (HACEDT), that contains the location of the VOLSERs.

Section HACPARM2 describes the definition and format of this table.

The HAC23202 subsystem initialization module loads this table in the ECS during IPL. The IEFEB4UV standard service function generates appropriate UBC-LOOKUP-VALUES for the specified ESOTERIC NAMEs and adds them to the table.

The UCB-LOOKUP-VALUE is no longer correct when HCD is in use and UNITs are dynamically added or removed. The HACCISSI procedure described in Section HACCISSI procedure must then be used to determine the UCB-LOOKUP-VALUES and correct the table.

The address of the table is stored in the SSCTSUS2 field in HCCs own subsystem control block SSCVT and is thus retrievable from all address areas (SP 228).

IGG026DU uses the catalog exit HAC026DU to perform the following for each catalog access using LOCATE (SVC 26):

- Receive control after SVC 26
- Check Locate Request for CVOL or ICFCAT
- Check DEVICE CLASS "TAPE" in VOLSER entry of the catalog work area
- Localize HACEDT table using SSCVT
- Search for requested VOLSER in the specified VOLSER range
- Substitute DEVICE TYPE with the corresponding LOOKUP-VALUE in the catalog work area

The following diagram shows how ZHC026DU is integrated:

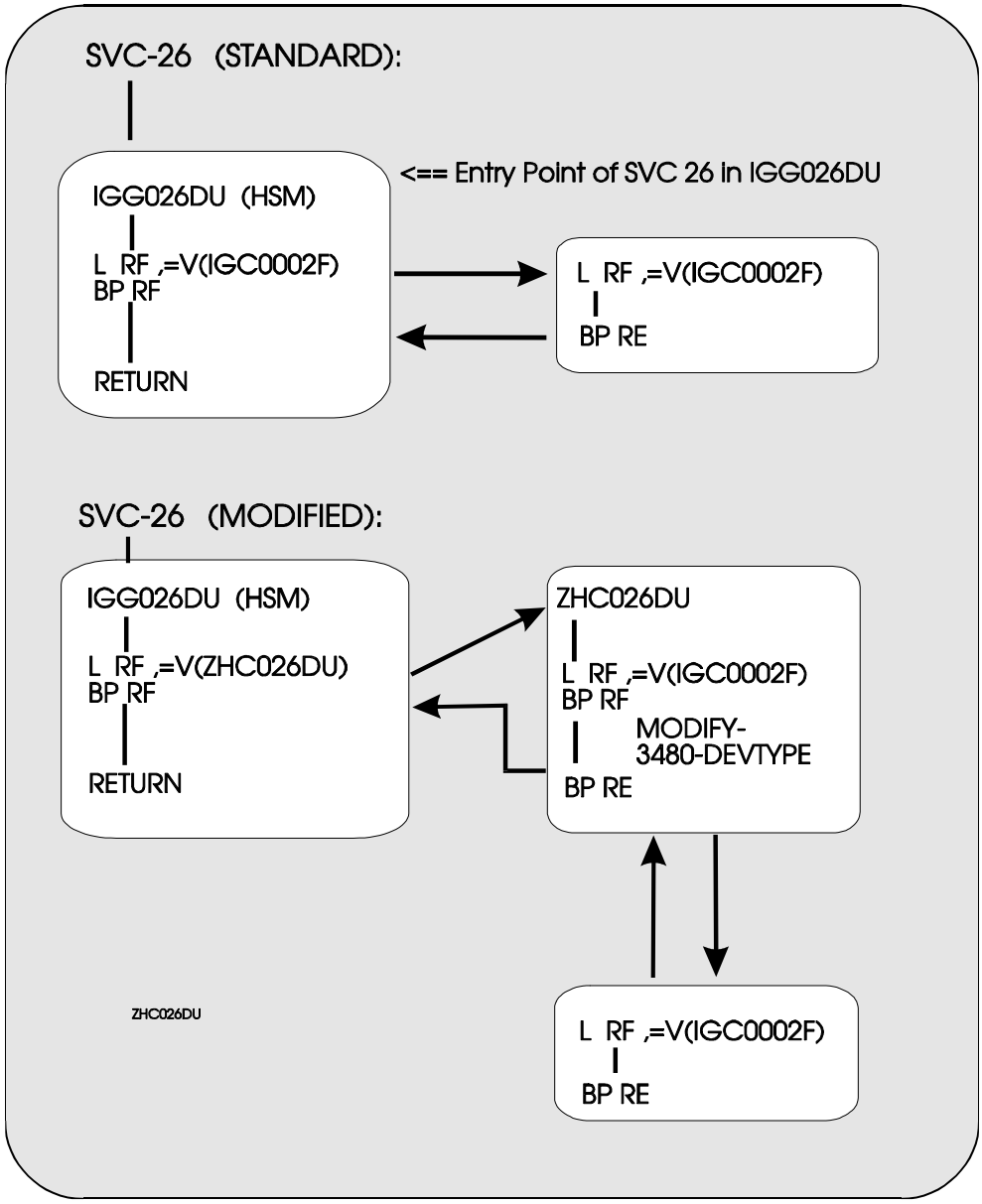


Fig. 2: Implementing the HCC Exit ZHC026DU

This method allows the dynamic reconstruction of the "location" of a cataloged data medium; which means that the JCL for existing installations need not be changed for **DISP=OLD/SHR** requests.

For **DISP=NEW** requests, the UNIT parameters in the JCL must be changed according to the ESOTERIC NAME (EDTGEN) as soon as the second tape pool is implemented.

Literature:
DATA FACILITY PRODUCT
VERSION 2: CUSTOMIZATION
C26-4267 Program Number 6565-XA2

MVS/DFP Version 3
Customization
C26-4560 Program Number 5665-XA3

IBM components that use the exit:
5665-329 DFHSM
5788-DHL THE AUTOMATIC ARCHIVE

The installation description can be found under:
INSTALLATION/CUSTOMIZATION GUIDE HCC/MVS 3.0.0.
Section: INSTALLNG THE CATALOG EXIT

6 COMPAREX MAPS

This Section describes how HCC/MVS functions when run together with the 'MAPS' product.

MAPS (**M**AINFRAME **A**UTOMATED **P**ERIPHERALS **S**UPPORT) is a product from the COMPAREX company and can be used, amongst other purposes, to control the MVS allocation.

The MAPS support is based on an identifier in the HCC/MVS Archive mirror that can be checked by MAPS.

An additional option 'MANPOOL' can be selected when ejecting cartridges from a robot system. When the eject with this option completes correctly, an identifier for the corresponding volser is set to indicate that this cartridge is assigned to a manual pool.

This identifier in the archive mirror is reset after the insert process has completed.

This identifier can also be set during archive mirror creation using HAA (**H**CC **A**rchive **A**dministration).

Volsers in dynamic areas are supported; volsers within a foreign mount area are **not** supported.

Further information on MAPS can be obtained directly from Comparex.

Literature:
COMPAREX
MAINFRAME AUTOMATED PERIPHERALS SUPPORT (MAPS)
GUIDE AND REFERENCE

Document Number: CXAE6082

7 JCL CHANGES FOR OPTIMIZING

No JCL changes are necessary when running AML units under HCC-MVS. However, a few changes should be made within the data migration framework. Apart from the item "ESOTERIC NAMEC" - all changes described here serve to improve system performance.

7.1 ESOTERIC NAMES

In order to attain operational separation of tape units under robot control and those without AML support (manual cartridge drive pool), an **ESOTERIC NAME** should be created for all cartridge drives within an AML system. Two unique esoteric names must be used for an installation with two AML systems.

Example: ROBA for the work archive
ROBS for the backup archive

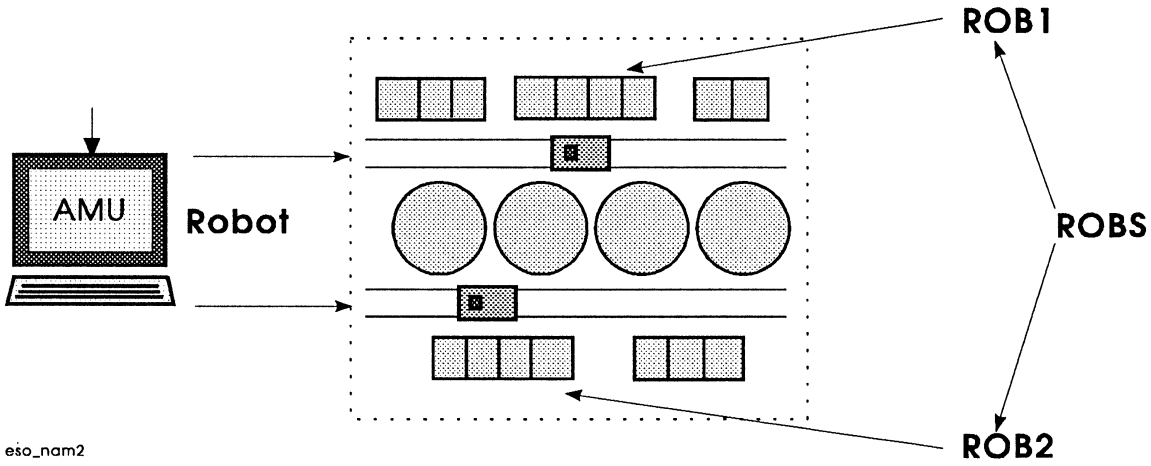


Fig. 3: Esoteric names for two AML systems

The system programmer allocates the two names and informs the job schedulers.

It is recommended to use the same length for new esoteric names as for already existing names (four characters in our example: ROBA).

It is then easier to convert complete JCL libraries without encroaching the protected area in columns 72-80 or, by shorter names, avoid removing the continuation character in column 72.

It is of course only necessary to use this method in installations with both robot cartridge drives and manual drives.

For twin-robot systems, it has proven to be useful to provide the cartridge station of both robots with **additional** esoteric names.

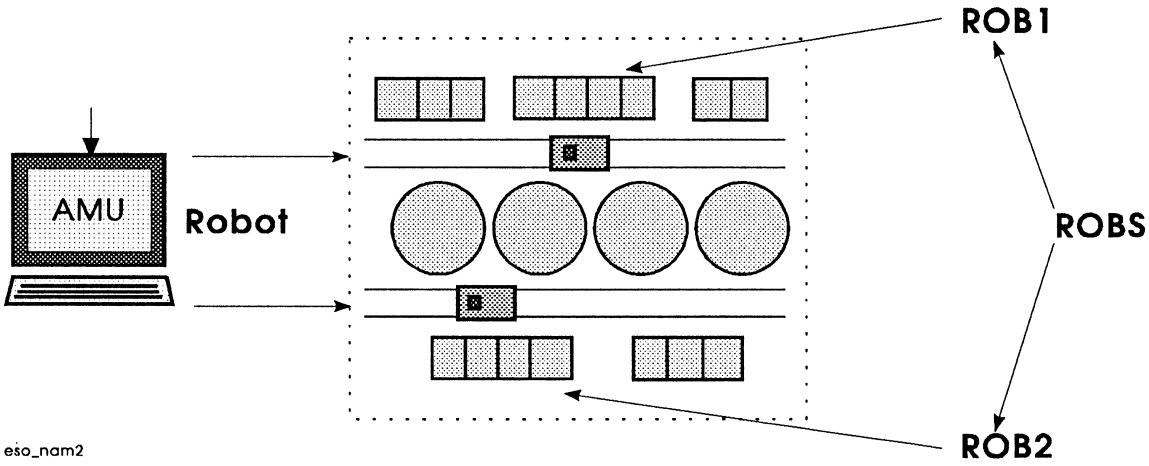


Fig. 4: Esoteric names for cartridge stations (twin-robot)

The esoteric names are then used in the "UNIT" parameter in the JCL.

7.2 DEFERRED MOUNTING

Normally, cartridges are requested at the start of a JOB (ALLOCATION) and the required cartridges are provided by the operator. The operating system generates these requests regardless of whether the program to be executed actually needs the data.

A cartridge is requested (from MVS) but the job ends within a few seconds before the cartridge is used. Such jobs are often canceled before the data is required. To minimize these unnecessary **MOUNT**/**KEEP** processes,

DEFERRED MOUNTING

should be used wherever possible. The JCL should be changed as follows:

```
//PROGR01 JOB...
//STEP1   EXEC PGM=PROG1

//INPUT   DD   DSN=OLDFILE,DISP=(OLD,KEEP),
//         UNIT=(,,DEFER)
//*
//OUTPUT  DD   DSN=NEWFILE,DISP=(NEW,CATLG),
//         UNIT=(ROBA,,DEFER)
```

The JCL parameter holds back the corresponding **MOUNT** activity until the:

FILE-OPEN

executes, that is, when the data is actually required. This method avoids almost all unnecessary **MOUNT**/**KEEP** processes.

7.3 FREE=CLOSE

A further measure which improves performance is introducing (or increasing the use of) the JCL parameter "FREE=CLOSE".

The following change is recommended when, for example, PROG1 (Section DEFERRED MOUNTING) should first sort the data read in using the DD-Statement INPUT:

```
//PROGR01 JOB...
//STEP1   EXEC PGM=PROG1

//INPUT   DD   DSN=OLDFILE,DISP=(OLD,KEEP),
//         UNIT=(,,DEFER),FREE=CLOSE
//*
//OUTPUT  DD   DSN=NEWFILE,DISP=(NEW,CATLG),
//         UNIT=(ROBA,,DEFER)
```

The parameter "FREE=CLOSE" releases the cartridge and the drive before the JOB completes (after the CLOSE). The cartridge unit can then be used for other jobs.

7.4 WORK FILES ON CARTRIDGES

Occasionally, larger TEMPORARY DATASETs must be written to cartridges. Such files can be recognized by their name and expiration date.

```
//WORK    DD   DSN=&TEMP,DISP=(NEW,DELETE),
//         UNIT=(ROBA,,DEFER),FREE=CLOSE,
//         LABEL=(1,SL,RETPD=0)
```

MVS recognizes such files and no **KEEP** is made at the end of the job. MVS assumes that a following job will need a SCRATCH file and can avoid a **KEEP** and a **MOUNT** process in such cases.

This MVS reaction is not desired when various cartridge pools are available for selection (refer to VOLGR and DSNGR). HCC can only intervene when **MOUNT** requests appear. The JCL parameter

VOLUME=(PRIVATE)

must be added to force MVS to keep temporary files. This ensures that HCC assigns following jobs SCRATCH cartridges from the correct pool.

8 SMF RECORDING

8.1 OVERVIEW

SMF recording is performed for forward recovery of the archive mirror and to support various evaluation possibilities.

HCC can record certain processes using SMF logging. The SMFWTM macro is then used and the HCC parameter settings control the writing of SMF records.

The HACPARM1 statements parameter

SMFRECORD=nnn

must be enabled when SMF records are to be written (refer to the HCC 3.0.0 Installation/Customization Reference Manual for a description of the HACPARM1 statements).

SMF records are written automatically for all processes described in Item 2 of this Section when this parameter is enabled in the HACPARM.

8.2 SMF Records

The following HCC events are recorded using SMF when the SMFRECORD parameter is enabled:

- ACTION messages occurring during processing.
- AML messages (requests to the robot, the archive or the AMU)
- Replies to AML messages
- Commands issued in HCC (including erroneous commands)
- MVS messages with routing codes 3 and 5 checked by HCC
- Intercommunication between HOST complexes
- Primary/Secondary HCC intercommunication using shared DASD (HACC EXCHANGE-LOG)
- Changes to the Archive mirror
- HACPARM1 statements
- HCC start parameter
- Mount/Keep/Load/Unload details, after incoming messages were accepted by HCC.

8.2.1 EVALUATION

The corresponding SMF records for the desired time frame must be filtered from the SMF file SYS1.MAN* to a file to be named.

The program ZHC15000 outputs the subtypes to a file. The record types of the subtypes are described in the Appendix (Section SMF-REPORTS), JCL samples follow in the next Section and can also be found in the hlq.ZHC....SAMP library.

8.2.2 SMF-DUMP

The SYS1.MAN* file from which the required SMF records are to be filtered must not be active. The SMF record subtypes can be copied with the following JCL:

```
//Jobname   JOB
//*-------------------------------------------------------------------*
//*     This job copies an extract from an SMF file to DUMPOUT        *
//*     according to the following criteria                           *
//*                                                                   *
//*       - DATE: Date from, Date to                                  *
//*       - TYPE: possible SMF record types for USER WRITTEN RECORDS  *
//*                                                                   *
//*                                                                   *
//*     Please change:                                                *
//*                                                                   *
//*       *:       A-Z OR 0-9                                         *
//*                (according to the active SMF DATA SET)             *
//*       DSNAME:  DATA SET NAME                                      *
//*       YYDDD:   Time interval for which the SMF records are        *
//*                to be dumped, whereby DDD must not                 *
//*                exceed 366.                                        *
//*       NNN:     SMF RECORD TYPE (between 128 and 255)              *
//*       n:       according to requirements                          *
//*                                                                   *
//*                                                                   *
//*     The DCB KEYWORD is not specified, which assigns               *
//*     DCB ATTRIBUTE (BLKSIZE=4096, LRECL=32760, RECFM=VBS) by the   *
//*     SMF DUMP program.                                             *
//*                                                                   *
//*                                                                   *
//*     Further information on SMF-Dump:                              *
//*      'THE SMF DUMP PROGRAM' IN: SYSTEM MANAGEMENT FACILITIES      *
//*                                                                   *
//*-------------------------------------------------------------------*
//STEP     EXEC PGM=IFASMFDP
//DUMPIN   DD  DSN=SYS1.MAN*,DISP=SHR  <-- or backup file
//DUMPOUT  DD  DSN=dsname,DISP=(NEW,CATLG,DELETE),
//             UNIT=SYSDA,SPACE=(TRK,(n),RLSE)
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
    DATE(YYDDD,YYDDD)
    OUTDD(DUMPOUT,TYPE(NNN))
```

Example 1 Job SMF-Dump refer to hlq.ZHC...SAMP (SMFDUMP)

EVALUATING SMF SUBTYPES

The ZHC15000 program outputs the SMF records in 'readable' form. The SMF records are written in sequential, chronological sequence, not sorted according to internal subtypes, to a list.

An example of the list output and its description can be found in the Appendix (Section SMF-REPORTS).

The following is a JCL example:

```
//Jobname  JOB
//*--------------------------------------------------------------------*
//*      PRINT WITH JOB SMFDUMP SELECTED SMF RECORDS                   *
//*--------------------------------------------------------------------*
//*--------------------------------------------------------------------*
//*      REPLACE THE FOLLOWING:                                        *
//*                                                                    *
//*      hlq         TO YOUR INSTALLATION HLQ                          *
//*      yyyyddd     START AND END DATE FOR THE PERIODE FOR WHICH      *
//*                  RECORDS ARE WRITTEN. ddd CAN NOT EXCEED 366.      *
//*      hlq.smfdat  DATASET WRITTEN BY JOB SMFDUMP                    *
//*      hlq.smflist LISTFILE OUTPUT DATASET OR SYSOUT=*               *
//*                                                                    *
//*      THE SPACE PARAMETER OF DDNAME LISTE SHOULD BE CHANGED         *
//*      ACCORDING OF THE SPACE YOU NEED                               *
//*--------------------------------------------------------------------*
//STEP1    EXEC PGM=ZHC15000,PARM='yyyyddd-yyyyddd'
//STEPLIB  DD  DSN=hlq.ZHC....LOAD,DISP=SHR
//SMFDAT   DD  DSN=hlq.smfdat,DISP=SHR
//LISTE    DD  DSN=hlq.smflist,DISP=(,CATLG,DELETE),
//             UNIT=SYSDA,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330),
//             SPACE=(TRK,(5),RLSE)
//*LISTE    DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//
```

Example SMF List editing refer to. hlq.ZHC....SAMP(SMFLIST)

8.3 ARCHIVE MIRROR FORWARD RECOVERY

Recovery is possible by comparing the reloaded Archive mirror file with the corresponding SMF file subtypes (59F/59S).

The control record of the Archive mirror contains the timestamp of the backup version. SMF records with subtype 59F and 59S are compared with the reloaded Archive mirror records as from this point in time. All Archive mirror changes recorded in the SMF records are repeated in the reloaded Archive mirror.

Preconditions:

- The customer must backup the Archive mirror at regular intervals
- SMF records must be present in unbroken chronological sequence
- HCC must not be active

8.3.1 Archive Mirror Backup

Example for Archive mirror backup:

```
//Jobname  JOB
//*-------------------------------------------------------------------*
//*      SAVE HACC ARCHIVE                                            *
//*-------------------------------------------------------------------*
//*-------------------------------------------------------------------*
//*      REPLACE THE FOLLOWING:                                       *
//*                                                                   *
//*      hlq         TO YOUR INSTALLATION HLQ                         *
//*      volser      TO VOLSER HACC ARCHIVE SAVE DATASET              *
//*-------------------------------------------------------------------*
//STEP1    EXEC PGM=IDCAMS
//DD1      DD  DSN=hlq.PDSNARCH,DISP=SHR
//DD2      DD  DSN=hlq.HACCARCH.SAVE,DISP=(NEW,CATLG),
//             UNIT=TAPE,LABLE=(,SL),VOL=SER=volser,
//             DCB=(RECFM=VB,LRECL=104,BLKSIZE=23440)
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
     REPRO -
        INFILE(DD1) -
        OUTFILE(DD2)
/*
```

Example Archive backup refer to hlq.ZHC....SAMP(ARCSAVE)

8.3.2 JCL for Recovery

Batch program ZHC16000 can be used to recreate the Archive. The Archive mirror records are filtered from the SMF records and updated in the Archive mirror.

The following sample job can be used for this purpose:

```
//Jobname  JOB .......
//*-------------------------------------------------------------------*
//*      ARCHIVE RECOVERY                                             *
//*                                                                   *
//*      STEP1:  DELETE/DEFINE CLUSTER                                *
//*      STEP2:  RELOAD ARCHIVE SAVE DATASET                          *
//*      STEP3:  RECOVERY                                             *
//*                                                                   *
//*-------------------------------------------------------------------*
//*-------------------------------------------------------------------*
//*      REPLACE THE FOLLOWING:                                       *
//*                                                                   *
//*      hlq          TO YOUR INSTALLATION HLQ                        *
//*      volser       TO VOLSER HACC ARCHIVE CLUSTER RESIDES ON       *
//*      size         TO SPACE OF HACC ARCHIVE CLUSTER                *
//*      hlq.smfdat   DATASET WITH HACC WRITTEN SMF-RECORDS FOR       *
//*                   RECOVERY PERIODE                                *
//*-------------------------------------------------------------------*
//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  DEL   hlq.PDSNARCH
  DEF   CL(NAME(hlq.PDSNARCH) VOL(volser) CYL(size) -
        SPEED RUS RECSZ(84,84) KEYS(6,0) SHR(3,3) FREESPACE(0,0)) -
        INDEX(NAME(hlq.PDSNARCH.INDEX) CISZ(512)) -
        DATA(NAME(hlq.PDSNARCH.DATA) CISZ(4096))
/*
//STEP2    EXEC PGM=IDCAMS
//DD1      DD   DSN=hlq.HACCARCH.SAVE,DISP=SHR
//DD2      DD   DSN=hlq.PDSNARCH,DISP=OLD
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
     REPRO -
        INFILE(DD1) -
        OUTFILE(DD2)
/*
//STEP3    EXEC PGM=ZHC16000
//STEPLIB  DD DSN=hlq.ZHC300.LOAD,DISP=SHR
//SMFDAT   DD DSN=hlq.smfdat,DISP=SHR
//ARCHIVES DD DSN=hlq.PDSNARCH,DISP=SHR
//SYSPRINT DD   SYSOUT=*
//*
```

Example for Recovery refer to hlq.ZHC....SAMP(ARCRECOV)

9 ARCHIVE GEOMETRY

All components of an AML system can be defined with linear coordinates using archive geometry. The LOCDESC statement assigns these coordinates to the corresponding AML components. They describe the distance from its defined base point.

Path lengths which in turn serve for priority decrement calculations can be calculated from these coordinates. The specifications in the LOCDESC statement are converted to a table. The **D LOC** command displays the contents of this table (refer to the Operator Guide)

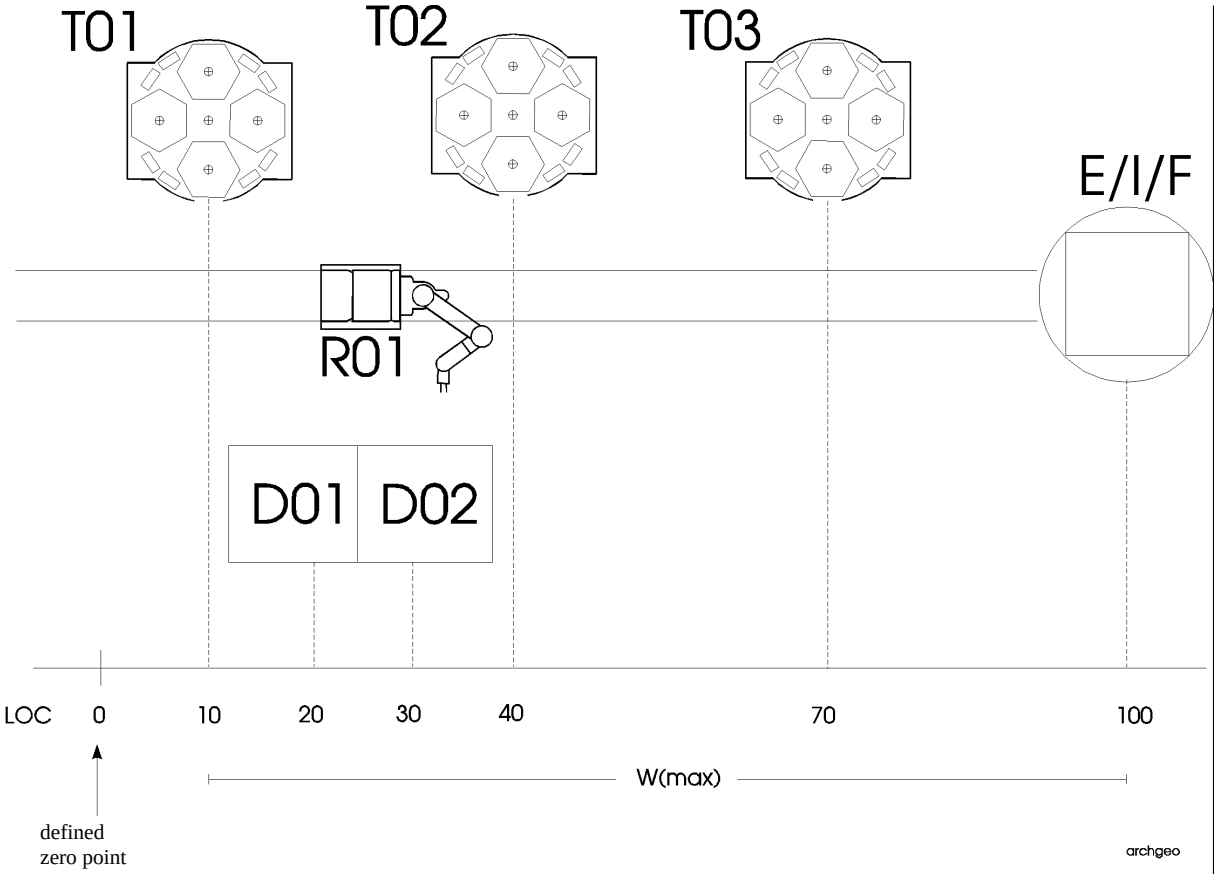


Fig. 5: Archive geometry

Example for priority decrement: (eject from T01)

```
W(max) = LOC(E/I/F) - LOC(T01) = 90
W1     = LOC(R01) - LOC(T01)   = 20
W2     = LOC(T01) - LOC(E/I/F) = 90

               (W1+W2)*10     (20 + 90) * 10   1100
PRIOABZUG = ------------ = -------------- = ---- = 6
               2*(W(max))         2 * 90       180
```

9.1 Priority Calculation

The maximum priority decrement (DECR) is set to 10 for the longest possible path W(max). The following formula results for a command with a partial path:

$$DECR = \frac{Path(1)*10}{W(max)}$$

The following formula results for a command with two partial paths:

$$DECR = \frac{Path(1)+ Path(2)*10}{2*W(max)}$$

9.2 Path Length Calculation

LOC(rob) represents the current robot position for all formulas. Only the absolute (positive) values are used for the calculation. Some values used for path length calculations vary depending on the command:

KE **KEC** **KECL**	W1=abs(LOC(rob) - LOC("ABBAUNIT")) W2=abs(LOC("ABBAUNIT") - LOC("VOLSER"))
MO **MOCL**	W1=abs(LOC(rob) - LOC("VOLSER")) W2=abs(LOC("VOLSER") - LOC("ABBAUNIT"))
MOVEH **MOVET**	W1=abs(LOC(rob) - LOC("FROMCOOR")) W2=abs(LOC("FROMCOOR") - LOC("TOCOOR"))
EJ	W1=abs(LOC(rob) - LOC("FROMCOOR")) W2=abs(LOC("FROMCOOR") - LOC("ABBAUNIT"))
VIC **VICC**	W1=abs(LOC(rob) - LOC("ABBAUNIT")) W2=abs(LOC("ABBAUNIT") - LOC("TOCOOR"))
SCH **VI DIR**	W1=abs(LOC(rob) - LOC("ABBAUNIT"))
INC	W1=abs(LOC(rob) - LOC("TOCOOR"))
INV	W1=abs(LOC(rob) - LOC("VOLSER"))

10 SCRATCH TAPE PROCESSING

10.1 OVERVIEW SCRATCH TAPE PROCESSING

HCC substitutes the volser of an available cartridge for non-specific **MOUNT** requests (volser=SCRTCH or PRIVAT).

HCC must receive information on available cartridges at regular intervals.

10.2 ALTERNATIVES FOR SCRATCH TAPE DETERMINATION

HCC has two alternatives to gain this information:

1. Standard exits (refer to the TMS statement)
2. Individual exit (refer to the TMSEXIT statement)

No further installation measures are necessary when the Standard exits are implemented with the TMS statement.

An individual exit can be used when the Standard exits are not applicable. The conventions for an individual exit are described in the Appendix. The Standard exits included in the scope of delivery function on the same principle.

Examples for exits for various tape management systems can also be found in the Appendix and in hlq.ZHC....SAMP (X01....).

Diagrams 6 and 7 on the following pages show the SCRATCH TAPE procedure flow, whereby, for performance reasons on twin-robot systems, the Scratch volser to be used from the associated pool is first determined when the Scratch mount command is actually generated. The decision is based on the following rules as well as other criteria:

- Towers 01/03/... are assigned to robot 01
- Towers 02/04/... are assigned to robot 02

and whether one of the robots is active is also considered. Sequential selection is used for all other systems.

10.3 SCRATCH TAPE PROCEDURE FLOW

The following diagram shows the Scratch tape procedure flow in HCC.

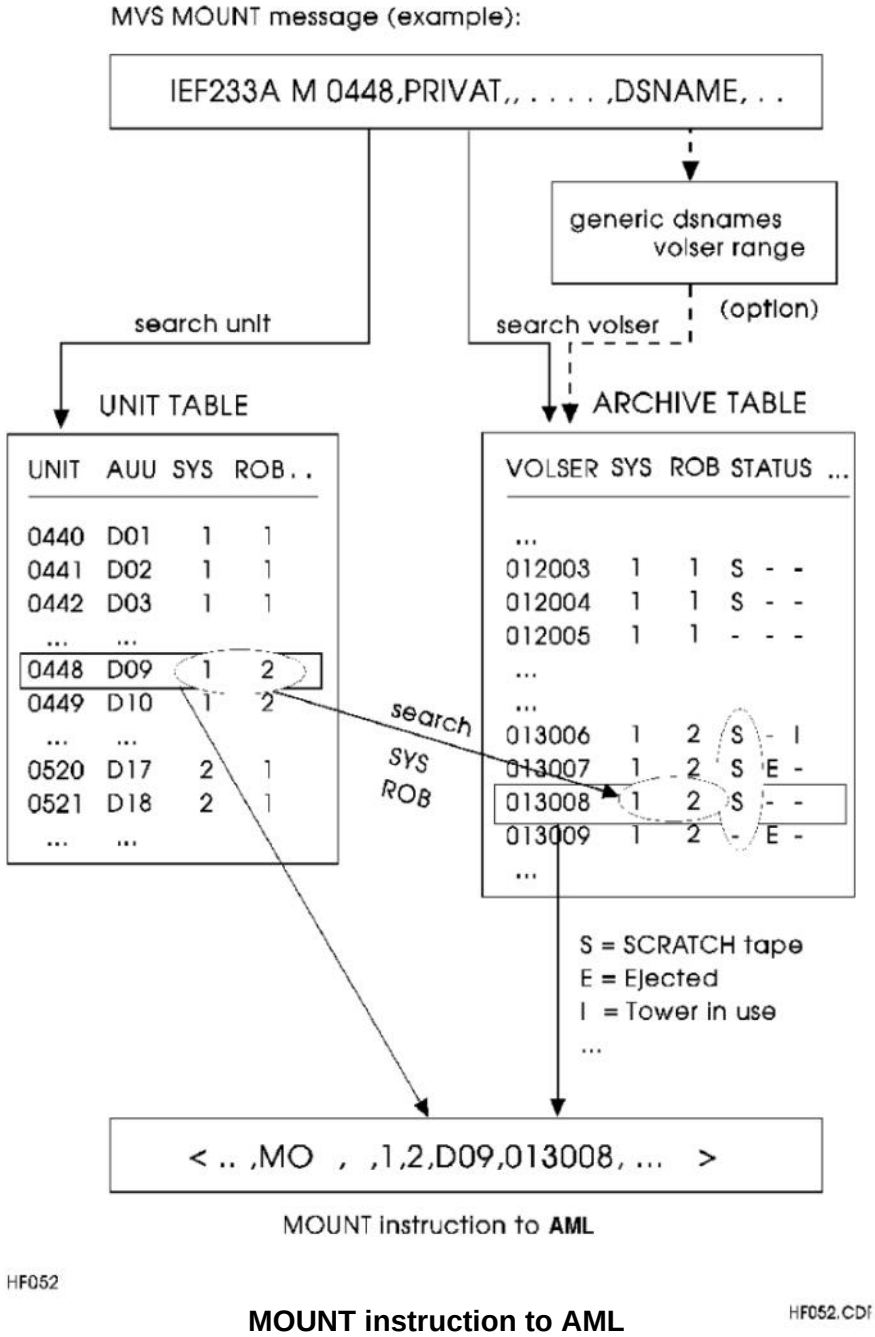


Fig. 6: SCRATCH TAPE substitution

The "SCRATCH" status in the HCC Archive mirror is reset to "NONE SCRATCH".

The status can only be set to "SCRATCH" again with a **FREEVOL** command.

10.4 FREEVOL PROCEDURE FLOW

The FREEVOL process runs asynchronous and does not hinder active scratch tape substitution.

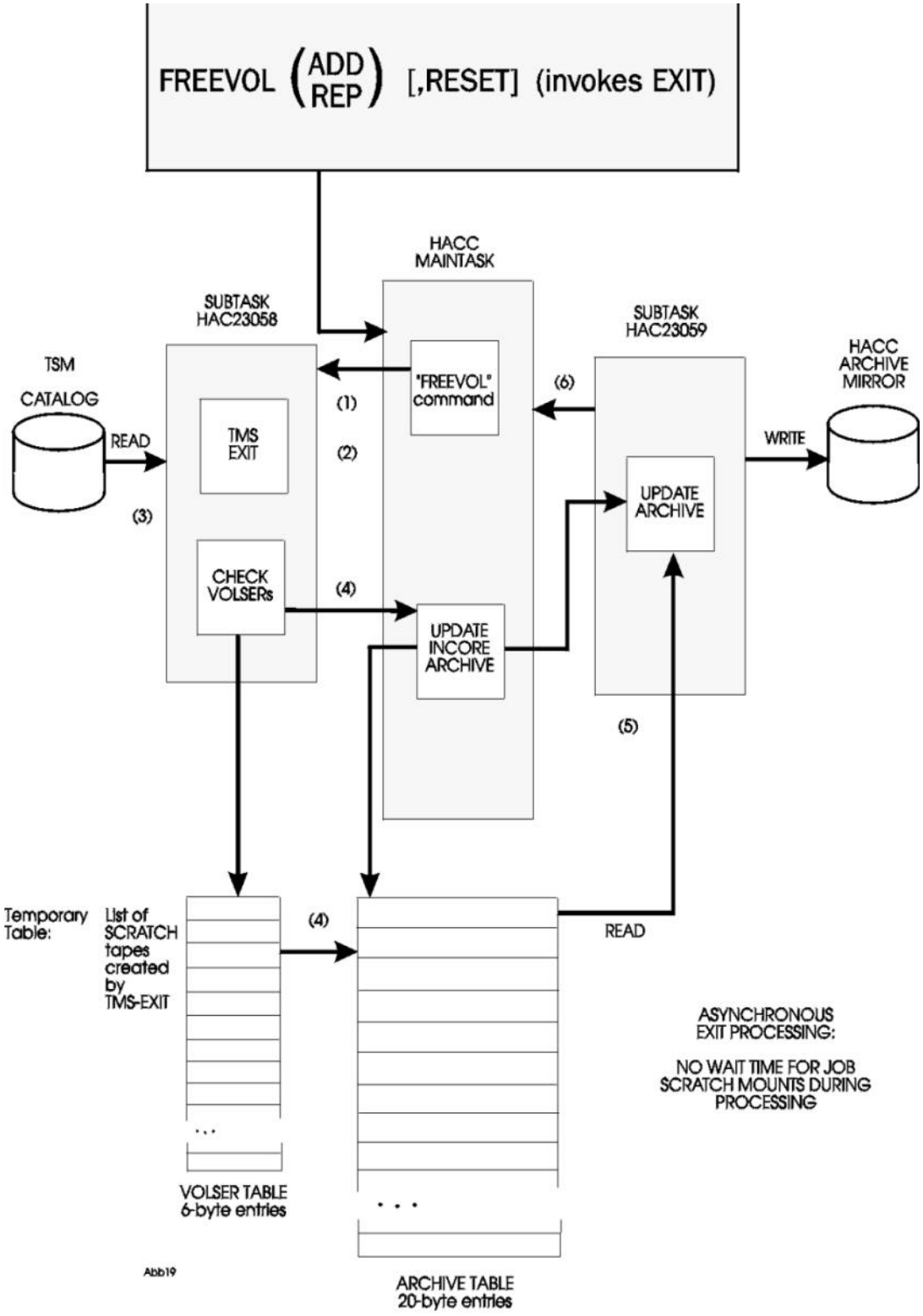


Fig. 7: SCRATCH TAPE selection using FREEVOL

(1) **FREEVOL** command calls subtask

(2) Subtask calls TMSEXIT, function: DYNALLOC, OPEN, READ TMC, selection SCRATCH and storage in table, CLOSE FREE

(3) Subtask checks table against the Archive mirror

(4) Return; main task updates the Archive mirror in storage

(5) Subtask call; update Archive mirror file

(6) Return to main task. HCC continues to operate up to the scratch tape warning limit during this asynchronous processing.

11 TAPE UNIT CLEANING

11.1 OVERVIEW - AUTOMATIC CLEANING

The current operating system software has no facilities to directly recognize that a tape unit needs to be cleaned. The only control is through the control unit using microcode without information exchange with MVS.

The unit cleaning period depends on the megabyte volume read from or written to the tape as well as the tape quality itself determined by the number of erase and noise gaps etc.

Average values are:

1 cleaning per 50 write/read transactions on a full cartridge.
The cleaning action itself takes approx. 45 seconds.

The CLEAN statement supports installation-specific adjustment for automatic tape cleaning depending on the average tape quality as well as on the maintenance status of the tape station.

11.2 CLEANING PERIOD

IBM 3480 needs to be cleaned after approx. 50 MOUNT/KEEP actions and an IBM 3490 after approx. 35. This also applies to comparable units from other manufacturers.

The cleaning period value for an IBM 3480 is not critical even when a cleaning process is prompted for in the tape unit DISPLAY (microcode controlled), there is no danger that a further MOUNT will be rejected.

This danger does however exist on an IBM 3490. The service technician can disable this forced cleaning when installing IBM 3490-E devices. Please point this out to your IBM technician.

HCC assigns an individual cleaning cartridge to each installed robot and maintains a unit-specific counter for MOUNTs in the UNIT LOG. Before each MOUNT, this value is compared to the value *nn* in the CLEAN statement to determine whether the cleaning limit has been reached. A cleaning MOUNT is triggered immediately when this threshold is reached. A simultaneous pending MOUNT is held back until cleaning is finished.

When the situation arises that two or more UNITs need to be cleaned at the same time, one UNIT is cleaned and the others continue to accept MOUNTs. This means that a particular cleaning threshold *nn* can be exceeded by one or two cycles.

11.3 WAIT TIME AT THE UNIT

Cleaning itself takes approx. 45 seconds. Cleaning can still be carried out successfully even when time=0 was specified for the CLEAN statement because the robot waits up to 60 seconds internally, for example when the flap on a 3480 is closed. The robot is then not available for any parallel **MOUNT**/**KEEP**/**EJECT** actions arising during such a situation.

Between 40 and 50 seconds is a representative value for the time.

11.4 REMOVING A CLEANING CARTRIDGE

A cartridge is automatically ejected when the maximum usage value specified in the CLEAN statement is reached. HCC then switches to the next cleaning cartridge specified for the robot involved.

The cleaning cartridges are subjected to a high mechanical load and should not be used until the very last possible cycle. It is recommended to limit cartridges with 500 cleaning cycles to 450 actual cleaning processes. Cartridges with a limit of 100 should be limited as well (95).

HCC sends the warning message HAC129A every two minutes when no further cleaning cartridges are available in the AML Archive after the cleaning cartridge has reached its limit. The current process is continued (for example at night-time) despite pending cleaning requirements to avoid interrupting productive processing. The pending cleaning is triggered automatically as soon a new cleaning cartridge is inserted.

11.5 COMPUTER CENTRE ORGANIZATION

1. During installation, at least two cleaning cartridges should be made available for each robot in the AML Archive (refer to the HCC commands for the definition).

2. It is recommended to identify the cleaning cartridges optically so that they are immediately recognized after eject and not unintentionally inserted again.

3. Cleaning cartridges are normally allocated to fixed locations (coordinates) in the AML Archive. Sufficient BARCODE labels should be available for unused cleaning cartridges inserted with the same VOLSER as the automatically ejected, used cleaning cartridges.

Sticking copies of the original labels on the cartridges is normally sufficient.

4. Cleaning cartridges can always be identified by the recess on the bottom transport surface. Inserting a normal cartridge for the cleaning process interrupts processing.

12 APPENDIX

12.1 HCC SAMPLIB CONTENTS

Contents of hlq.ZHC...SAMP

12.1.1 HCC MAPPING MACROS

ZHCASR	Mapping macro for HCC Archive mirror
ZHCEX1	Mapping macro for TMS-Userexit Parmlist
ZHCLDV	Mapping macro for HCC Library Device Description
ZHCLGD	Mapping macro for HCC Message Log Record
ZHCMOX	Mapping macro for HCC Mount Exit Parmlist
ZHCRMF	Mapping macro for HCC AML Robot Communication
ZHCSMF	Mapping macro for HCC SMF Recording

other Mapping macros

ARCHIV	Volume MASTER FILE BASE CA-DYNAM/TLMS R.5

12.1.2 RACF DEFINITION EXAMPLES

ICHRFR01	Example for SAF Router Table
ICHRRCDE	Example for Resource Class Descriptor Table

12.1.3 VTAM DEFINITION EXAMPLES

VTAMDEF	LOGMODE 3270 session (OS/2)
	LOGMODE LU 6.2 single session (OS/2)
	LOGMODE LU 6.2 parallel session (OS/2)
	Resource Def. HOST LU 6.2 (APPL statement)
	Resource Def. PC TOKEN RING/3745 (SWNET statement)
	Resource Def. PC TOKEN RING/3174-OS/2 1.3EE
	HACNET Statements for single session and parallel sessions
CM2DEF	Networking Definition File for Single Session CM/2 1.1

12.1.4 USER EXIT EXAMPLES

X01CA101	FREEVOL-Exit CA1 V.4.9
X01CA102	FREEVOL-Exit CA1 V.5.x
X01CA103	FREEVOL-Exit CA1 V.5.x with more than one physical TMC
X01CA104	FREEVOL-Exit CA1 with several [alpha] volser ranges
X01CNTT1	FREEVOL-Exit CONTROL-T V.1.2.2
X01EPIC1	FREEVOL-Exit EPIC
X01IBMA1	FREEVOL-Exit Automatic Archive
X01SEQF1	FREEVOL-Exit Sequential dataset
X01SEQF2	FREEVOL-Exit Seq. dataset with several volser ranges
X01TLMS1	FREEVOL-Exit CA/Dynam TLMS Release 5
X01TLMS2	FREEVOL-Exit CA/Dynam TLMS Release 5.4
X01ZARA1	FREEVOL-Exit ZARA V.1.1
X02MO001	Mount exit
X03VI001	Insert exit
X04EJ001	Eject exit
X04EJ002	Generate eject statements for CA1 V.4.9 and V.5.0
X04EJ031	Generate eject statements for CA1 V.5.1 from Vault Management Report
X04EJ032	Example on setting the Vault Descriptions for CA1-VMS Reports
X04EJ033	Example JCL for a CA1-EARL Report

12.1.5 ABBASEND EXAMPLES

ABSSAMP	Call to subprogram whereby the allocation described in Section 2.1 must be considered
ABSTEST	Batch test after installation
ABSTESTN	Batch test after installation (PARM='NEW')

12.1.6 OPTICAL DISK EXAMPLES

ODSC	Describes possible methods to gain optical disks with FREESPACE
ODSCJOB	Example JCL to create a sequential dataset as input for HCC
ODSCREXX	Example REXX procedure to select optical disks with FREESPACE

12.1.7 TEST AID

COMTEST	Utility to test the EXCP communication path

12.1.8 OTHER EXAMPLES AND PROCEDURES

ARCRECOV	Example JCL for Archive recovery
ARCSAVE	Example JCL for Archive backup
ASPALLOC	Allocate Archive mirror
CLEARSSI	Restore HCC SSI after Apply (PTF-Status ZY00069)
DFSMSRMM	Example procedure to select SCRATCH or EJECT VOLSERs from DFSMSRMM catalog
HACCSTC1	HCC Started Task / Single Host Complex
HACCSTC2	HCC Started Task / Multi Host Complex LU 6.2
HACCSTC3	HCC Started Task / Multi Host Complex Token Ring
HACCISSI	INITIALIZE HACPARM2 for HAC026DU
HACPAPPC	Example on operating multi-complex with APPC
HACPARM1	Example on operating 2 AML systems
HACPARM2	Describes the correlations between Volser-ranges and esoteric-unit names
X00026DU	SVC-26 exit to modify 3480-devtype catalog installation exit (MVS XA 2.2.0)
JICHRFR	Example JCL for RACF SAF ROUTER TABLE
JICHRRCD	Example JCL for RACF CLASS DESCRIPTOR TABLE
LCK	Started task for tape label display --> S LCK,V=012345,UNIT=SYSKS
LKSVC26	Example JCL for catalog installation exit
LNKCTTEX	Example JCL for LINK CONTROL-T FREEVOL exit
LNK26LPA	Example JCL for LINK catalog installation exit from LPALIB
DLNK26LP	Deinstallation ZHC026DU when linked from LPALIB
DLNK26DU	Deinstallation ZHC026DU
MIGARC24	Example JCL to migrate HCC Archive 2.3.1 to 2.4.0
SMFDUMP	Example JCL to extract the SMF dataset to a user file
SMFLIST	Example JCL to print the SMF user file
TLI	Started task for tape label initialization

UNASSIGN	Sample ISPF edit commands to UNASSIGN ejected volumes 1. HAA SAVE 2. Copy this member to dataset in SYSPROC concentration 3. Edit SEQ. Backup-DS 4. %UNASSIGN
UM000001	USERMOD to modify HACPARM1 DSNAME in program ZHC01300
UM000002	USERMOD to modify HACPARM1 DSNAME in program ZHC20200
ZAPPRM1	Setup HACPARM1 DSNAME and/or member Example for SYS1.PARMLIB/HACPARM1
ZAPPRM2	Setup HACPARM2 DSNAME and/or member Example for SYS1.PARMLIB/HACPARM2

12.2 RECORD FORMATS

The following HCC RECORDS are shown in detail:

A.2.1. ARCHIVE MIRROR
A.2.2. MESSAGE LOG
A.2.3. TMS-EXIT PARAMETER LIST
A.2.4. SMF REPORTS
A.2.5. SMF-RECORD
A.2.6. MSG-EXIT PARAMETER LIST

12.2.1 ARCHIVE MIRROR

COMMON NAME: ARCHIVE RECORD
MACRO ID: ZHCASR
RECORD SIZE: 8-84 BYTES,VARIABLE LENGTH
DATA FORMAT: VSAM-KSDS,KEYLEN=6,RKP=0

DEC	HEX	TYPE	LENGTH	NAME	DESCRIPTION
0	0	STRUCTURE	84	ARENTRY	ARCHIVE VOLSER RECORD
0	0	STRUCTURE	20	ARINCORE	INCORE VOLSER ENTRY
0	0	CHARACTER	6	ARVOL	VOLSER (KEY)
6	6	BITSTRING	1	ARSTA	VOLUME STATUS
		1...		ARXCLT	- CLEANING TAPE else DATATAPE
		.1..		ARXINA	- IN ARCHIVE else EJECTED
		..1.		ARXEJD	- EJECT TO BE DONE
		...1		ARXSCR	- SCRATCH TAPE
		.... 1...		ARXMPMA	- IN MPMA
		.... .1..		ARXIDEV	- IN VOLUME INSERT DEVICE
		.... ..1.		ARXFPMA	- IN FPMA
		1		ARXHOME	- ON HOME POSITION
7	7	BITSTRING	1	ARSTA1	RECOVERY STATUS
		1...		ARXVIR	- IS VIRTUAL TAPE
		.1..		ARXMVD	- MOVE TO BE DONE
		..1.		ARXUPD	- UPDATE STATUS TO BE DONE
		...1		ARXMAN	- WAS PROCESSED DURING MANUEL MODE
		.... 1...		ARXMP	- MANUAL POOL-MAPS
		.... .1..		ARXHWS	- in HWS (HACC WORK STORAGE)
		.... ..1.		ARXVIEX	- EJECTED (VOLUME INSERT EXIT)
		1		ARXINVC	- INVALID COORDINATE
8	8	BITSTRING	1	ARFLG	STATUS FLAG USED BY MAINTASK
		1...		ARXINQ	- VOLUME WAS PLACED INTO ROBQUE
		.1..		ARXIUS	- VOLUME IN USE(ON UNIT,IN EJECT..)
		..1.		ARXART	- ALLOCATED FOR ROBTEST FUNCTIONS
		...1		ARXFKES	- FORCE KES DUE TO ROB UNACCESSABILITY
		.... 1...		ARXCLC	- CURRENT CLEANING TAPE
		.... .1..		ARXPML	- PMA LOAD PENDING
		.... ..1.		ARXPMU	- PMA UNLOAD PENDING
9	9	BITSTRING	1	ARSYS	ABBA SYSTEM TO WHICH TAPE BELONGS
		1		ARXSYS1	- SYSTEM 1
		.... ..1.		ARXSYS2	- SYSTEM 2
10	A	BITSTRING	1	ARROB	ROBOT WHICH CAN ACCESS THE TAPE
		1		ARXROB1	- ROBOT 1
		.... ..1.		ARXROB2	- ROBOT 2
11	B	BINARY	1	AREJDEV	EJECT-DEVICE-NO (RECOVERY ONLY)
12	C	UNSIGNED	4	ARHCO	HOME-COORDINATE
12	C	UNSIGNED	1	ARHTOW	- TOWER NO./SIDE NO.
13	D	UNSIGNED	1	ARHSEG	- SEGMENT NO.
14	E	UNSIGNED	1	ARHROW	- ROW IN SEGMENT
15	F	UNSIGNED	1	ARHPOS	- POSITION IN ROW
16	10	UNSIGNED	4	ARTCO	TEMPORARY COORDINATE
16	10	UNSIGNED	1	ARTTOW	- TOWER NO./SIDE NO.
17	11	UNSIGNED	1	ARTSEG	- SEGMENT NO.
18	12	UNSIGNED	1	ARTROW	- ROW IN SEGMENT
19	13	UNSIGNED	1	ARTPOS	- POSITION IN ROW
		...1 .1..		ARLI	*-ARENTRY,LENGTH OF INCORE-RECORD

This structure is used as INCORE-RECORD in MAINTASK.

DEC	HEX	TYPE	LENGTH	NAME	DESCRIPTION
20	14	STRUCTURE	52	ARDATA	STATISTIC DATA
20	14	BINARY	4	ARTMK	BINARY-TH OF LAST MOUNT/KEEP
24	18	PACKED	4	ARDMK	00YYDDDF OF LAST MOUNT/KEEP
28	1C	BINARY	4	ARTEJ	BINARY-TH OF LAST EJECT
32	20	PACKED	4	ARDEJ	00YYDDDF OF LAST EJECT
36	24	BINARY	4	ARTIN	BINARY-TH OF LAST INSERT
40	28	PACKED	4	ARDIN	00YYDDDF OF LAST INSERT
44	2C	BINARY	2	ARTUSE	TOTAL USECOUNT OF VOLSER
46	2E	BINARY	2	AROUSE	USECOUNT OF VOLSER DISP=OLD
48	30	BINARY	2	ARSWAPS	NO.OF SWAPS FOR THIS TAPE
50	32	PACKED	3	ARCRDTE	CREATION DATE DISP=NEW (YYDDDF)
53	35	PACKED	3	AREXDTE	EXPIRATION DATE (YYDDDF)
56	38	CHARACTER	4	ARUSER	USER FIELD
60	3C	BINARY	1	ARRESPT	MOUNT RESPONSE TIME (SEC'S)
61	3D	CHARACTER	3	ARUNIT	LAST MVS-UNIT FOR MOUNT/KEEP
64	40	CHARACTER	8	ARJOB	JOBNAME OF LAST MOUNT
72	48	CHARACTER	4	ARUNITXA	LAST MVS-UNIT FOR MO/KE MVS/ESA
		..11 .1..		ARLD	*-ARDATA,LENGTH OF STATISTIC DATA

This entry describes statistic data for user evaluations.

DEC	HEX	TYPE	LENGTH	NAME	DESCRIPTION
76	4C	STRUCTURE	8	ARRECOV	RECOVERY DATA
76	4C	CHARACTER	1	ARREJST	EJECT STATUS
77	4D	CHARACTER	2	ARREJDV	EJECT DEVICE
79	4F	CHARACTER	1	ARRUPST	UPDATE STATUS
80	50	CHARACTER	4	ARRMVCO2	MOVE TO DO TARGET COORDINATE
		.... 1...		ARRECOL	*-ARRECOV,LENGTH OF RECOVERY DATA
		.1.1		ARL	*-ARENTRY,TOTAL RECORD LENGTH

This entry is used by archive recovery.

HCC ARCHIVE comprises different record types. The first byte of NON-VOLSER-keys is hexadecimal between X'00'- X'40', default X'FF'.

12.2.2 MESSAGE LOG

COMMON NAME: MESSAGE LOG RECORD
MACRO ID: HACLGD
RECORD SIZE: 132 BYTES (LIST-OUTPUT FORMAT)
DATA ACCESS: BSAM

DEC	HEX	TYPE	LENGTH	NAME	DESCRIPTION
0	0	STRUCTURE	132	LRECORD	LOG RECORD
0	0	CHARACTER	3	LRTYP	MESSAGE TYPE
					- 'ROB' ROBOT COMMUNICATION
					- 'CMD' HACC-COMMAND
					- 'MVS' MVS-MESSAGE
					- 'MSG' HACC-MESSAGE
					- 'SWI' LOG-SWITCH INDICATOR
					- 'TRC' Module/Subroutine TRACE
					- 'SUP' Message Suppressed by HACC
					- 'SSI' Message Suppressed by SSI
					- 'WTO' Message Suppressed by WTO
					- 'MPF' Message Suppressed by MPF
3	3	CHARACTER	1	*	BLANK-CHARACTER
4	4	CHARACTER	5	LRDATE	DATE 'YYDDD'
9	9	CHARACTER	1	*	' '-CHARACTER
10	A	CHARACTER	8	LRTIME	TIME 'HH/MM/SS'
18	12	CHARACTER	1	*	BLANK-CHARACTER
19	13	CHARACTER	4	LRCPU	CPUID ORIGINATING THE LOG ENTRY
23	17	CHARACTER	1	*	BLANK-CHARACTER
24	18	CHARACTER	8	LRJOB	JOBNAME RELATED TO THE LOG ENTRY
32	20	CHARACTER	1	*	BLANK-CHARACTER
33	21	CHARACTER	87	LRMSG	MESSAGE TEXT
120	78	CHARACTER	1	*	BLANK-CHARACTER
121	79	CHARACTER	2	LREPID	WQE-REPLYID (OR BLANK IF ABSENT)
123	7B	CHARACTER	1	*	BLANK-CHARACTER
124	7C	CHARACTER	8	LRECSQ	RECOVERY-SEQ.NO (OR BLANK IF ABSENT)
		1... .1..	8	LRECL	*-LRECORD,LENGTH OF LOG-RECORD

12.2.3 TMS EXIT PARAMETER LIST

```
          ZHCEX1 INDEX=any prefix
*================================================*   ************
*         PARMLIST EXIT1 ZHC02500                *   * ZHCEX1   *
*================================================*   ************
&I.E1PARM DS    0F
*-----------------------------------------------------------------*
*         INCOMING VALUES FROM ZHC05800 TO USER-EXIT              *
*-----------------------------------------------------------------*
&I.E1DSNA DC    A(0)              ADDRESS OF TMS DSNAME (HACPARM1)
&I.E1NSCR DC    F'0'              NO.OF SCRATCHTAPES EXPECTED
&I.E1FTMS DC    CL6'000000'       - FIRST VOLSER IN RANGE
&1.E1LTMS DC    CL6'000000'       - LAST VOLSER IN RANGE
&1.E1FREE DC    3F'0'             UNUSED
*-----------------------------------------------------------------*
*         VALUES TO BE STORED DURING EXIT1-PROCESSING             *
*-----------------------------------------------------------------*
&I.E1CORE DC    F'0'           GETMAIN SIZE OF FREEVOL-TAB
&I.E1TABA DC    A(0)           ENTRY-POINT OF FREEVOL-TAB
&I.E1TABE DC    A(0)           LOGICAL END OF FREEVOL-TAB (BEHIND)
&I.E1EL   EQU   6              FREEVOL-TAB ENTRY-LENGTH
```

12.2.4 SMF REPORTS

Example SMF output listing ZHC15000

```
                                                                                                                                   PAGE    1
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----1----+----1----+----2----+-
94014 15:54:08 S500 002 ZZZ129A KEINE REINIGUNGS-KASSETTE IM ARCHIV SYS=1,ROB=2, BITTE UMGEHEND EINLAGERN
94014 15:54:08 S500 015 *HACC0*       CMS AUTO LOGSTART
94014 15:54:08 S500 015 *HACC0*       CMS AUTO SYSLOG
94014 15:54:17 S500 015 *HACC0*       CON .... ROSA
94014 15:54:17 S500 U59 .... H1 01 01 <A1H1A00,0001,ROSA, ,    ,1,1,   ,      ,  ,        ,        ,14/155417,H1,0001>
94014 15:54:22 S500 042      *HACC0*  <H1A1A00,0001,ROSA, ,    ,1,1,   ,      ,  ,        ,        ,14/155417,H1,0001>
94014 15:54:22 S500 007 .... *HACC0*  <H1A1A00,0001,ROSA,P,    ,1,1,   ,      ,  ,        ,        ,14/155417,H1,0001>
94014 15:54:31 S500 015 *HACC0*       CON .... ROSA 2,1
94014 15:54:31 S500 U59 .... H1 02 01 <A2H1A00,0002,ROSA, ,    ,2,1,   ,      ,  ,        ,        ,14/155431,H1,0002>
94014 15:54:38 S500 042      *HACC0*  <H1A2A00,0002,ROSA, ,    ,2,1,   ,      ,  ,        ,        ,14/155431,H1,0002>
94014 15:54:38 S500 007 .... *HACC0*  <H1A2A00,0002,ROSA,P,    ,2,1,   ,      ,  ,        ,        ,14/155431,H1,0002>
94014 15:54:44 S500 015 *HACC0*       CON .... ROSA 2,2
94014 15:54:44 S500 U59 .... H1 02 02 <A2H1A00,0003,ROSA, ,    ,2,2,   ,      ,  ,        ,        ,14/155444,H1,0003>
94014 15:54:49 S500 042      *HACC0*  <H1A2A00,0003,ROSA, ,    ,2,2,   ,      ,  ,        ,        ,14/155444,H1,0003>
94014 15:54:49 S500 007 .... *HACC0*  <H1A2A00,0003,ROSA,P,    ,2,2,   ,      ,  ,        ,        ,14/155444,H1,0003>
94014 15:54:54 S500 015 *HACC0*       CON .... DU
94014 15:55:17 S500 015 *HACC0*       CON .... M 92A,000100
94014 15:55:17 S500 018 092A H1 *HACC0*  S500 IEF233A M 92A,000100,
94014 15:55:17 S500 005 092A H1 01 01 <A1H1A00,0004,MO  , ,    ,1,1,D01,000100, 1,        ,        ,14/155517,H1,0004>
94014 15:55:26 S500 042      *HACC0*  <H1A1A00,0004,MO  , ,    ,1,1,D01,000100, 1,        ,        ,14/155517,H1,0004>
94014 15:55:26 S500 007 092A *HACC0*  <H1A1A00,0004,MO  ,P,    ,1,1,D01,000100, 1,        ,        ,14/155517,H1,0004>
                            .
                            .
                            .                                                                                                       PAGE   35
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----1----+----1----+----2----+-
```

DESCRIPTION OF LISTING

HEADER (1-23)

POSITION 1-23 SAME FORMAT FOR ALL RECORDS

POS	LEN	DESCRIPTION
1- 5	5	DATE RECORD WAS WRITTEN (YYDDD)
7- 14	8	TIME RECORD WAS WRITTEN (HH:MM:SS)
16- 19	4	CPUID (SMFCPUID)
21- 23	3	SHORT PROGRAM NUMBER FOR EXAMPLE: 002: ZHC00200 EXCEPT 59F: ZHC05900 F FOR FULL UPDATE 59S: ZHC05900 S FOR STATUS UPDATE

DIFFERENT FORMATS: (25-132)

002: MESSAGE ISSUER

POS	LEN	DESCRIPTION
25-132	108	ALL ACTION MESSAGES AND THOSE MESSAGES WHICH ARE CHOSEN BY 'SMF=Y' ARE PRINTED

005: AMU COMMUNICATION MESSAGE SENT TO AMU

POS	LEN	DESCRIPTION
25- 28	4	MVS UNIT
30- 31	2	HACCID
33- 34	2	LIBRARY NO (REQUESTER)
36- 37	2	ROB NO
39-132	94	MESSAGE TEXT (<> DATAQUE)

007: AMU COMMUNICATION MESSAGE RETURN (RESPONSE FROM AMU)

POS	LEN	DESCRIPTION
25- 28	4	MVS UNIT
30- 37	2	JOBNAME
39-132	94	MESSAGE TEXT (<> DATAQUE)

015: ISSUED HCC COMMANDS

POS	LEN	DESCRIPTION
25- 33	8	JOBNAME ORIGINATING THE LOG ENTRY
34- 38	4	SYSTEM SENDS MESSAGE (CPUId)
39- 42	3	FUNCTION ORIGIN
		CON CONSOLE
		ABS ABBASEND
		EXL EXCHANGE LOG
		COX COMPLEX
		CMS INTERNAL CMS
		COM COMMUNICATION
43- 46	4	FLAG
		MSGR MESSAGE RETURN
		CPWY CRX-POST ON WAIT=YES
		CPCO CRX-POST AFTER FUNCTION COMPLETE
		NEWM MUST BE DONE BY NEW METHODE
		ABSI ABS UNDERISPF
		AUTO COMMAND ENTERED VIA AUTOCMD=
		DEL1 DELETE ENTRY BY ZHC00100
48-127	80	MESSAGE TEXT

018: MVS MESSAGES VERIFIED BY HCC

POS	LEN	DESCRIPTION
25- 26	2	ABBA LIBRARY NO.
27- 28	2	ABBA UNIT NO.
30- 31	2	HACCID
33- 40	8	JOBNAME ORIGINATING THE LOG ENTRY
42- 45	4	SYSTEM SENDS MESSAGE
47-132	86	MVS MESSAGE

041: HCC TO HCC INTERCOMMUNICATION (CX COMMANDS)

POS	LEN	DESCRIPTION
30- 37	8	JOBNAME
39-132	94	COMMAND

042: INCOMING COMMUNICATION MESSAGES

POS	LEN	DESCRIPTION
30- 37	8	JOBNAME
39-132	94	COMMAND

044: CY COMMAND RETURN

POS	LEN	DESCRIPTION
25- 32	8	JOBNAME
34- 37	4	SYSTEM SENDS MESSAGE (CPUId)
(39-132	94	EXCHANGE LOG RECORD)
39- 42	4	CPUID OR ZERO (FREE RECORD)
44- 48	5	DATE STAMP
50- 57	8	TIME STAMP
59- 60	2	RECORD TYPE
		80 - MVS MOUNT/KEEP MESSAGE
		40 - ABBASEND COMMAND
		20 - INTERNAL FUNCTION
62- 63	2	FUNCTION BRANCH CODE
		04 - SWAP UPDATE (CMD(3)=CUU)
		08 - CY-COMMAND RETURN
65- 66	2	WRITE CONTROL
		04 - ERASE PROCESSED RECORD
		08 - WRITE RECORD
		08 - HIGHEST BRANCH ENTRY
68- 69	2	RECTYPE CY-COMMAND BRANCH ENTRY
		04 - ECHO
		08 - COMMAND
		0C - MESSAGE
		10 - SHUTDOWN
		14 - UNLOAD UNIT (MVS-COMMAND)
		14 - MAX.BRANCH ENTRY VALUE
71- 78	8	JOBNAME
80-132	53	MVS MESSAGES (124)
		(ABBASEND/SECSYS INFORMATION)

U28: HCC EXCHANGE LOG

POS	LEN	DESCRIPTION
25- 32	8	JOBNAME
34- 37	4	SYSTEM SENDS MESSAGE (CPUId)
(39-132	94	EXCHANGE LOG RECORD)
39- 42	4	CPUID OR ZERO (FREE RECORD)
44- 48	5	DATE STAMP (YYDDD)
50- 57	8	TIME STAMP (HH:MM:SS)
59- 60	2	RECORD TYPE
		80 - MVS MOUNT/KEEP MESSAGE
		40 - ABBASEND COMMAND
		20 - INTERNAL FUNCTION
62- 63	2	FUNCTION BRANCH CODE
		04 - SWAP UPDATE (CMD(3)=CUU)
		08 - CY-COMMAND RETURN
65- 66	2	WRITE CONTROL
		04 - ERASE PROCESSED RECORD
		08 - WRITE RECORD
		08 - HIGHEST BRANCH ENTRY
68-69	2	RECTYPE CY-COMMAND BRANCH ENTRY
		04 - ECHO
		08 - COMMAND
		0C - MESSAGE
		10 - SHUTDOWN
		14 - UNLOAD UNIT (MVS-COMMAND)
		14 - MAX.BRANCH ENTRY VALUE
71- 78	8	JOBNAME
80-132	53	MVS MESSAGES (124)
		(ABBASEND/SECSYS INFORMATION)

59F: FULL UPDATE ARCHIVES CONTROL RECORD

POS	LEN	DESCRIPTION
25- 26	2	CONTROL RECORD IDENTIFIER (00)
32- 43	12	'HACC-ARCHIVE' IDENTIFIER
44- 47	4	'S ' TEMP. TO HACC 2.1
49- 53	5	HAA LOAD/UNLOAD MODIFY COUNTER
55- 61	7	HAA USERID
63- 64	2	FLAG
		80 -> NORMAL CLOSE AFTER HACC STOP
66- 75	10	HAA TOTAL NO.OF RECORDS IN ARCHIVES
77- 81	5	DATE OF LAST CREATION (YYDDD)
83- 90	8	TIME OF LAST CREATION (HH:MM:SS)
92- 99	8	VERSION OF HACICC
101-105	5	DATE OF LAST TIME STAMP (YYDDD)
107-114	8	TIME OF LAST TIME STAMP (HH:MM:SS)

59F: FULL UPDATE ARCHIVES LDEV RECORD

POS	LEN	DESCRIPTION
25- 26	2	RECORD TYPE (01 LDEV)
		- EQU '01' IDENTIFIER
28- 29	2	SYSTEM
31- 33	3	DEVICE
		31-31 1 - LDEV TYPE (E,I,F,W,R,T..)
		32-33 2 - LDEV DEVICE NO.
35- 37	3	SPLIT NUMBER (0-255)
39- 42	4	RELATIVE POSITION OF DEVICE
44- 45	2	TYPE FLAGS
		80 FPMA
		40 MPMA (ONLY RACK)
		20 HWS (ONLY TOWER)
		10 DYNAMIC AREA
		08 NO ASR-POINTER VALID
47- 47	1	DEVICE TYPE (E,F,I,R,T,D,W)
49- 50	2	DEVICE NO.
52- 53	2	SYSTEM NO.
55- 56	2	ROBOT NO.
58- 59	2	TOWER/RACK
60- 61	2	FIRST-SEGMENT
62- 63	2	LAST-SEGMENT
64- 65	2	FIRST-ROW
66- 67	2	LAST-ROW
69- 70	2	FIRST-POSITION OF FIRST ROW
71- 72	2	LAST-POSITION OF LAST ROW
72- 74	2	NO.OF SLOTS PER ROW (10/25)
76- 78	3	NO.OF REDEFINITION
80- 80	1	ORGANIZATION (V=VERT,H=HORZ)
82- 89	8	START-COORD
91- 98	8	END-COORD
100-109	10	CAPACITY (NO.OF SLOTS)
111-120	10	NO.OF ASSIGNED SLOTS
122-125	4	OFFSET OF NEXT FREE SLOT IN DEVICE

59F: FULL UPDATE ARCHIVES UNIT RECORD

POS	LEN	DESCRIPTION
25- 26	2	RECORD TYPE (20 UNIT)
		- EQU '20' IDENTIFIER
28- 29	2	SYSTEM
31- 38	8	COORD
		31-32 2 - TOWER/RACK (ALWAYS=00)
		33-34 2 - SEGMENT (ALWAYS=00)
		35-36 2 - ROW=UNIT (RANGE 01-99)
		37-38 2 - POSITION (ALWAYS 01)
40- 41	2	ROBOT
42- 42	1	'D' ID
44- 45	2	ABBA-UNIT NO (01-FF)
46- 48	3	MVS-UNIT ADDRESS
50- 52	3	ALTERNATE MVS-UNIT ADDRESS
54- 58	5	DATE YYDDD OF FIRST INIT
60- 64	5	DATE YYDDD OF LAST REFERENCE
66- 75	10	TOTAL MOUNT COUNTER
77- 86	10	SUM OF ABBA RESPONSE TIME (SEC'S)
88- 97	10	TOTAL CLEANING COUNTER
99-108	10	TOTAL SWAP COUNTER
110-119	10	TOTAL NO.OF ABBA ERRORS
121-124	4	MVS-UNIT ADDRESS MVS/ESA
126-129	4	ALTERN. MVS-UNIT ADDRESS MVS/ESA

59F: FULL UPDATE ARCHIVES JUKEBOX RECORD

POS	LEN	DESCRIPTION
25- 26	2	RECORD TYPE (21 JUKEBOX)
		- '21' IDENTIFIER
28- 30	2	SYSTEM
31- 32	2	ROBOTER
34- 35	2	ABBA UNIT
37- 39	3	SUBRECORD NUMBER (000 CONTROL REC.
		001-256 FOR VOLSER RECORDS)
41- 46	8	NAME OF JUKEBOX
50- 54	5	MAX. CAPACITY OF JUKEBOX
56- 60	5	USED SLOTS IN JUKEBOX
62- 63	2	FLAGBYTE FOR CONTROL RECORD
		(NOT YET USED)
65- 66	2	FLAGBYTE FOR VOLSER RECORD
		80 - IF ON: VOLSER RECORD IS FULL

		IF VOLSER RECORD (> 000)
25- 26	2	RECORD TYPE (21 JUKEBOX)
		- '21' IDENTIFIER
28- 30	2	SYSTEM
31- 32	2	ROBOTER
34- 35	2	ABBA UNIT
37- 39	3	SUBRECORD NUMBER (000 CONTROL REC.
		001-256 FOR VOLSER RECORDS)
50- 51	2	FLAGBYTE FOR VOLSER RECORD
		80 - IF ON: VOLSER RECORD IS FULL
53- 58	6	VOLSER1 (A-SIDE)
60- 65	6	VOLSER2 (B-SIDE)
67- 72	6	VOLSER3 (A-SIDE)
74- 79	6	VOLSER4 (B-SIDE)
81- 86	6	VOLSER5 (A-SIDE)
88- 93	6	VOLSER6 (B-SIDE)
95-100	6	VOLSER7 (A-SIDE)
102-107	6	VOLSER8 (B-SIDE)

59F: FULL UPDATE ARCHIVE VOLSER RECORD

PART 1:

POS	LEN	DESCRIPTION
25- 30	6	VOLSER (>40)
32- 33	2	STATUS FLAG 1
		VOLUME STATUS
	80	- CLEANING TAPE ELSE DATATAPE
	40	- IN ARCHIVE ELSE EJECTED
	20	- EJECT TO BE DONE
	10	- SCRATCH TAPE
	08	- OD BIT MUST BE OFF
	04	- VOLUME IN INSERT DEVICE
	02	- IN FPMA
	01	- ON HOME POSITION
32- 33	2	STATUS FLAG 1
		REDEFINITION FOR OPTICALS

	80	- MOUNTED ON OAD ELSE HOME OR IN JB
	40	- IN ARCHIVE ELSE EJECTED
	20	- EJECT TO BE DONE
	10	- SCRATCH TAPE
	08	- OD BIT MUST BE ON
	04	- VOLUME IN INSERT DEVICE
	02	- VOLUME IN JUKEBOX
	01	- ON HOME POSITION
35- 36	2	STATUS FLAG 2
		RECOVERY STATUS
	80	- IS VIRTUAL-TAPE
	40	- MOVE TO BE DONE
	20	- MOVE HOME
	10	- IF ON: B-SIDE OFF: A-SIDE
	08	- MANUAL POOL - MAPS
	04	- IN HWS
	02	- INITIAL VOLSER ENTRY
	01	- INVALID COORDINATE
38- 39	2	TEMP.VOLUME FLAG USED BY HACC/HAA
	80	- VOLUME WAS PLACED INTO ROBQUE
	40	- VOLUME IN USE (ON UNIT)
	20	- VOLUME ALLOCATED FOR ROBTEST
	10	- FORCE KES DUE TO EJECT-DEVICE
	08	- IS USED AS CL-TAPE BY ANY ROBOT
	04	- PMA LOAD
	02	- PMA UNLOAD
	80	- HAA: HOME COORD UPDATE
	80	- HAC23059: RECOVER COORDINATE
41- 42	2	SYSTEM-NO TO WICH TAPE BELONGS
	01	- SYSTEM 1
	02	- SYSTEM 2
44- 45	2	ROBOTS WICH CAN MOUNT THE TAPE
	01	- ROBOT 1
	02	- ROBOT 2
47- 48	2	TEMP.EJECT-DEVICE NO. (BINARY 01-99)
50- 57	8	HOME-COORDINATE
	49-50 2	- TOWER/SIDE
	51-52 2	- SEGMENT
	53-54 2	- ROW
	55-56 2	- POSITION
50- 57	6	VOLSER A-SIDE (FOR VOLSER OF B-SIDE)
	2	FREE
59- 66	8	TEMP-COORDINATE
	58-59 2	- TOWER/SIDE
	60-61 2	- SEGMENT
	62-63 2	- ROW
	64-65 2	- POSITION
59- 66	8	RESEVED (INDEX FOR OD A-SIDE)

68- 72	5	YYDDD OF LAST MO/KE
74- 81	8	HH:MM:SS OF LAST MO/KE
83- 87	5	YYDDD OF EJ
89- 96	8	HH:MM:SS OF EJ
98-102	5	YYDDD OF IN
104-111	8	HH:MM:SS OF IN
113-117	5	TOTAL USECOUNT OF VOLSER
119-123	5	USECOUNT OF VOLSER DISP=OLD

PART 2:

25- 29	5	NO.OF SWAPS FOR THIS TAPE
31- 35	5	CREATION DATE OF TAPE (YYDDD)
37- 41	5	EXPIRATION DATE OF TAPE (YYDDD)
43- 46	4	USER-FIELD
48- 50	3	MOUNT RESPONSE TIME (SECS)
52- 54	3	LAST MVS-UNIT FOR MOUNT/KEEP
56- 63	8	JOBNAME OF LAST MO/KE
65- 68	4	LAST MVS-UNIT FOR MOUNT/KEEP MVS/ESA
70- 71	2	EJECT STATUS
73- 74	2	EJECT DEVICE
76- 77	2	UPDATE STATUS
79- 86	8	MOVE TARGET COORDINATE

59S: STATUS UPDATE IN ARCHIVES VOLSER RECORD

POS	LEN	DESCRIPTION
25- 30	6	VOLSER (>40)
32- 33	2	STATUS FLAG 1
	80	- CLEANING-TAPE ELSE DATA-TAPE
	40	- IN ARCHIVE ELSE EJECTED
	20	- EJECT TO BE DONE
	10	- IS SCRATCHTAPE
	08	- OD (MUST BE OFF FOR NON-OPTICAL)
	04	- VOLUME IN INSERT DEVICE (T-COORD)
	02	- IN FPMA
	01	- ON HOME-POSITION

U59: AMU COMMUNICATION (FOR COMMANDS THAT BYPASS THE QUEUE)

POS	LEN	DESCRIPTION
30- 31	2	HACCID
33- 34	2	SYSTEM NO (REQUESTER)
36- 37	2	ROB NO
39-132	94	MESSAGE TEXT (<> DATAQUE)

U13: HANDLE MOUNT MESSAGE
(AFTER SYNTAX CHECK, WHEN IT IS ACCEPTED BY HCC TO WORK WITH)

POS	LEN	DESCRIPTION
25- 29	4	MVS UNIT
31- 32	2	HACC ID
34- 41	8	JOBNAME
43- 46	4	CPU ID (PRIMARY)
52- 59	8	FUNCTION (MOUNT)
61- 68	8	UNIT NAME
70- 72	3	ABBA UNIT
75- 80	6	VOLSER

U14: HANDLE KEEP MESSAGE
(AFTER SYNTAX CHECK, WHEN IT IS ACCEPTED BY HACC TO WORK WITH)

POS	LEN	DESCRIPTION
25- 29	4	MVS UNIT
31- 32	2	HACC ID
34- 41	8	JOBNAME
43- 46	4	CPU ID (PRIMARY)
52- 59	8	FUNCTION (KEEP)
61- 68	8	UNIT NAME
70- 72	3	ABBA UNIT
75- 80	6	VOLSER

491: HANDLE OAD MOUNT MESSAGE
(AFTER SYNTAX CHECK, WHEN IT IS ACCEPTED BY HCC TO WORK WITH)

POS	LEN	DESCRIPTION
25- 29	4	MVS UNIT
31- 32	2	HACC ID
34- 41	8	JOBNAME
43- 46	4	CPU ID (PRIMARY)
52- 59	8	FUNCTION (MOUNT)
61- 68	8	UNIT NAME
70- 72	3	ABBA UNIT
75- 80	6	VOLSER

492: HANDLE OAD KEEP MESSAGE
(AFTER SYNTAX CHECK, WHEN IT IS ACCEPTED BY HCC TO WORK WITH)

POS	LEN	DESCRIPTION
25- 29	4	MVS UNIT
31- 32	2	HACC ID
34- 41	8	JOBNAME
43- 46	4	CPU ID (PRIMARY)
52- 59	8	FUNCTION (KEEP)
61- 68	8	UNIT NAME
70- 72	3	ABBA UNIT
75- 80	6	VOLSER

493: HANDLE OAD FLIP MESSAGE
(AFTER SYNTAX CHECK, WHEN IT IS ACCEPTED BY HCC TO WORK WITH)

POS	LEN	DESCRIPTION
25- 29	4	MVS UNIT
31- 32	2	HACC ID
34- 41	8	JOBNAME
43- 46	4	CPU ID (PRIMARY)
52- 59	8	FUNCTION (FLIP)
61- 68	8	UNIT NAME
70- 72	3	ABBA UNIT
75- 80	6	VOLSER

494: HANDLE LOAD JUKEBOX MESSAGE
(AFTER SYNTAX CHECK, WHEN IT IS ACCEPTED BY HCC TO WORK WITH)

POS	LEN	DESCRIPTION
25- 29	4	MVS UNIT
31- 32	2	HACC ID
34- 41	8	JOBNAME
43- 46	4	CPU ID (PRIMARY)
52- 59	8	FUNCTION (LOAD)
61- 68	8	UNIT NAME
70- 72	3	ABBA UNIT
75- 80	6	VOLSER

495: HANDLE UNLOAD JB MESSAGE
(AFTER SYNTAX CHECK, WHEN IT IS ACCEPTED BY HCC TO WORK WITH)

POS	LEN	DESCRIPTION
25- 29	4	MVS UNIT
31- 32	2	HACC ID
34- 41	8	JOBNAME
43- 46	4	CPU ID (PRIMARY)
52- 59	8	FUNCTION (UNLOAD)
61- 68	8	UNIT NAME
70- 72	3	ABBA UNIT
75- 80	6	VOLSER

013: READ AND PROCESS HCCPARM1

POS	LEN	DESCRIPTION
25- 26	2	HACC ID
28-131	103	HACCPARM1 STATEMENT

101: HCC START PARAMETER

POS	LEN	DESCRIPTION
25- 26	2	HACC ID
28-131	103	HACC START PARAMETERS

12.2.5 SMF RECORDS

```
         MACRO
         ZHCSMF &INDEX=
.*-------------------------------------------------------------------*
.*       HACC - ABBA ROBOT COMMUNICATION            HACC 3.0.0       *
.*-------------------------------------------------------------------*
         LCLC  &I
         AIF   (T'&INDEX NE 'O').GO1
&I       SETC  'S'
         AGO   .GO2
.GO1     ANOP
&I       SETC  '&INDEX'
.GO2     ANOP
*=========================================================*  ***********
*           SMF RECORD                                    *  * ZHCSMF   *
*=========================================================*  ***********
&I.MFENT   DS    0F
*-------------------------------------------------------------------*
*          SMF STANDARD RECORD HEADER
*-------------------------------------------------------------------*
&I.MFHDR   DS    0CL18
&I.MFHLEN  DC    BL2'0'       REC LENGTH
&I.MFHSEG  DC    BL2'0'       =X'00' FOR SPANNED RECORDS
&I.MFHRET  DS    0CL202       RETURN FROM SMFDUMP
&I.MFHFLG  DC    BL1'0'       SYSTEM INDICATOR
&I.MFHRTY  DC    BL1'0'       RECORD TYPE
&I.MFHTME  DC    BL4'0'       TIME
&I.MFHDTE  DC    PL4'0'       DATE
&I.MFHSID  DC    CL4' '       CPUID (SMFCPUID)
*-------------------------------------------------------------------*
*          HACC HEADER
*-------------------------------------------------------------------*
&I.MFREC   DS    0CL10
&I.MFRID   DC    CL4'HACC'    RECORD ID
&I.MFRECT  DC    X'00'        INTERNAL RECORD NUMBER
           DC    CL1' '
&I.MFUNIT  DC    CL4' '       MVS UNIT
&I.MFHRL   EQU   *-&I.MFENT
*-------------------------------------------------------------------*
* INTERNAL RECID                                                   *
* ==                                                               *
* 05       RECORD 005             94                               *
*          AMU COMMUNICATION MESSAGES SENT TO AMU                   *
*-------------------------------------------------------------------*
&I.MFR005  DS    0CL94
&I.MFHI005 DC    CL2' '       HACCID
           DC    CL4' '
&I.MFS005  DC    X'00'        SYSTEM NO (REQUESTER)
&I.MFRB05  DC    X'00'        ROB NO
           DC    CL6' '
&I.MFDATA  DS    0CL(&I.ABAL) DATA ENTRY (80 BYTES)
            ZHCRMF INDEX=&I   ROB MESSAGE
&I.R005L   EQU   *-&I.MFR005
&I.RH05L   EQU   *-&I.MFENT
*-------------------------------------------------------------------*
* 10       RECORD U59             94                               *
*          AMU COMMUNICATION (FOR COMMANDS THAT BYPASS THE QUEUE)   *
*-------------------------------------------------------------------*
           ORG   &I.MFR005
&I.MFR010  DS    0CL94
&I.MFHI010 DC    CL2' '       HACCID
           DC    CL4' '
&I.MFS010  DC    X'00'        SYSTEM NO (REQUESTER)
&I.MFRB10  DC    X'00'        ROB NO
           DC    CL6' '
&I.MFDATAR DS    0CL(RABAL) DATA ENTRY (80 BYTES)
            ZHCRMF INDEX=R    ROB MESSAGE
&I.R010L   EQU   *-&I.MFR010
&I.RH10L   EQU   *-&I.MFENT
```

```
*-------------------------------------------------------------------*
* 13       RECORDS U13/U14 (MOUNT/KEEP) ZHC06900                    *
*          491/492/493/494/495 (MOUNT/KEEP/FLIP/LOAD/UNLOAD) ZHC049*
*          MESSAGES AFTER SYNTAX CHECK, WHEN THEY ARE ACCEPTED BY   *
*          HACC TO WORK WITH THEM                                   *
*-------------------------------------------------------------------*
           ORG   &I.MFR005
&I.MFR013  DS    0CL80
&I.MFHI013 DC    CL2' '       HACCID
&I.MFJ013  DC    CL8' '       JOBNAME
&I.013CPU  DC    CL4' '       CPUID SYSTEM SENDS MESSAGE
           DC    CL4' '       FREE
&I.MFMU013 DC    CL8' '       MVS UNIT
&I.MFAU013 DC    CL3' '       ABBA UNIT
           DC    CL1' '       FREE
&I.MFV013  DC    CL6' '       VOLSER
&I.MFDN013 DC    CL44' '      DSNAME
&I.R013L   EQU   *-&I.MFR013
&I.RH13L   EQU   *-&I.MFENT
*-------------------------------------------------------------------*
* 15       RECORD 015            138         (COMMAND PROCESSOR)  *
*          ALL COMMANDS SENT TO HACC                                *
*-------------------------------------------------------------------*
           ORG   &I.MFR005
&I.MFR015  DS    0CL138
&I.MFJ015  DC    CL8' '       JOBNAME
&I.015CPU  DC    CL4' '       SYSTEM SENDS MESSAGE (CPUID)
&I.MFORIG  DC    X'00'        COMMAND FROM
&I.MFFLG   DC    X'00'        ABBASEND FLAG
&I.MFC015  DC    CL124' '     COMMAND
&I.R015L   EQU   *-&I.MFR015
&I.RH15L   EQU   *-&I.MFENT
*-------------------------------------------------------------------*
* 20       RECORD 059            84                                 *
*          FULL UPDATE OF ARCHIVE RECORD                            *
*-------------------------------------------------------------------*
           ORG   &I.MFR005
&I.MFR020  DS    0CL84
&I.MFASR   DC    CL84' '      ARCHIVES RECORD
&I.R020L   EQU   *-&I.MFR020
&I.RH20L   EQU   *-&I.MFENT
*-------------------------------------------------------------------*
* 25       RECORD 059            08                                 *
*          STATUS UPDATE OF ARCHIVE RECORD                          *
*-------------------------------------------------------------------*
           ORG   &I.MFR005
&I.MFR025  DS    0CL8
&I.MFAVOL  DC    CL6'000000'  VOLSER
&I.MFASTA  DC    X'00'        STATUS
           DC    X'00'
&I.R025L   EQU   *-&I.MFR025
&I.RH25L   EQU   *-&I.MFENT
*-------------------------------------------------------------------*
* 30       RECORD 007            138                                *
*          AMU COMMUNICATION (RESPONSE FROM AMU)                    *
*-------------------------------------------------------------------*
           ORG   &I.MFR005
&I.MFR030  DS    0CL138
&I.MFJ030  DC    CL8' '       JOBNAME
           DC    CL6' '
&I.MFC030  DC    CL124' '     COMMAND
&I.R030L   EQU   *-&I.MFR030
&I.RH30L   EQU   *-&I.MFENT
```

```
*-------------------------------------------------------------------*
* 40       RECORD 041            138                                *
*          COMMAND PROCESSOR FOR HOST COMPLEX CX-COMMANDS             *
*-------------------------------------------------------------------*
           ORG   &I.MFR005
&I.MFR040  DS    0CL138
&I.MFJ040  DC    CL8' '       JOBNAME
           DC    CL6' '
&I.MFC040  DC    CL124' '     COMMAND
&I.R040L   EQU   *-&I.MFR040
&I.RH40L   EQU   *-&I.MFENT
*-------------------------------------------------------------------*
* 45       RECORD 042            138                                *
*          FIRST LEVEL COMMAND RETURN HANDLER IN HOST COMPLEX         *
*-------------------------------------------------------------------*
           ORG   &I.MFR005
&I.MFR045  DS    0CL138
&I.MFJ045  DC    CL8' '       JOBNAME
           DC    CL6' '
&I.MFC045  DC    CL124' '     COMMAND
&I.R045L   EQU   *-&I.MFR045
&I.RH45L   EQU   *-&I.MFENT
*-------------------------------------------------------------------*
* 50       RECORD 018            154             (MVS MSG SCANN)  *
*          MVS MESSAGES WITH ROUTING CODES 3 AND 5 VERIFIED           *
*          BY HACC                                                    *
*-------------------------------------------------------------------*
           ORG   &I.MFR005
&I.MFR050  DS    0CL154
&I.MFHI050 DC    CL2' '       HACCID
&I.MFJ050  DC    CL8' '       JOBNAME
&I.050CPU  DC    CL4' '       SYSTEM SENDS MESSAGE (CPUId)
&I.MFMSG   DC    CL140' '
&I.R050L   EQU   *-&I.MFR050
&I.RH50L   EQU   *-&I.MFENT
*-------------------------------------------------------------------*
* 65       RECORD 101            132           (START PARAMETER)  *
*          PARAMETERS USED WITH HACC START                            *
*-------------------------------------------------------------------*
           ORG   &I.MFR005
&I.MFR065  DS    0CL132
&I.MFHI065 DC    CL2' '       HACCID
&I.MFSP065 DC    CL130' '     START PARAMETER
&I.R065L   EQU   *-&I.MFR065
&I.RH65L   EQU   *-&I.MFENT
*-------------------------------------------------------------------*
* 70       RECORD 002            108             (MESSAGE ISSUER) *
*          HACC ACTION MESSAGES ISSUED DURING PROCESS                 *
*-------------------------------------------------------------------*
           ORG   &I.MFR005
&I.MFR070  DS    0CL108
&I.MFDMSG  DC    CL108' '     DISPLAY MESSAGE
&I.R070L   EQU   *-&I.MFR070
&I.RH70L   EQU   *-&I.MFENT
*-------------------------------------------------------------------*
* 80       RECORD 044            178               (CY CMD RET)  *
*          SHARED DASD INTERCOMMUNICATION (CY REQUESTS)*
*-------------------------------------------------------------------*
           ORG   &I.MFR005
&I.MFR080  DS    0CL178
&I.MFJ080  DC    CL8' '       JOBNAME
&I.080CPU  DC    CL4' '       SYSTEM SENDS MESSAGE (CPUId)
           DC    CL2' '
&I.MFC080  DC    CL164' '     COMMAND
&I.R080L   EQU   *-&I.MFR080
&I.RH80L   EQU   *-&I.MFENT
```

```
*-------------------------------------------------------------------*
* 85       RECORD U28            178                    (EXCLOG)  *
*          RECORD WRITTEN TO EXCLOG                                 *
*-------------------------------------------------------------------*
           ORG   &I.MFR005
&I.MFR085  DS    0CL178
&I.MFJ085  DC    CL8' '       JOBNAME
&I.085CPU  DC    CL4' '       SYSTEM SENDS MESSAGE
           DC    CL2' '
&I.MFC085  DC    CL164' '     COMMAND
&I.R085L   EQU   *-&I.MFR085
&I.RH85L   EQU   *-&I.MFENT
*-------------------------------------------------------------------*
* 55       RECORD 013            200              (HACCPARM REC.)  *
*          HACPARM1 STATEMENTS                                      *
*-------------------------------------------------------------------*
           ORG   &I.MFR005
&I.MFR055  DS    0CL200
&I.MFHI055 DC    CL2' '       HACCID
&I.055PAR  DC    CL198' '     HACCPARM RECORD
&I.R055L   EQU   *-&I.MFR055
&I.RH55L   EQU   *-&I.MFENT
*-------------------------------------------------------------------*
&I.MFL     EQU   *-&I.MFENT   TOTAL LENGTH OF SMF RECORD
         MEND
```

12.2.6 MSG EXIT PARAMETER LIST

MVS STANDARD:

POS	LEN	DESCRIPTION
0- 6	7	MVS MESSAGE e.g. „IEF233A“
7- 7	1	-
8- 8	1	Function „M“
9- 9	1	-
10-13	4	MVS UNIT „0440“
14-14	1	,
15-20	6	VOLSER „012345“
21-21	1	,
22-216	194	-

JES3:

POS	LEN	DESCRIPTION
0- 6	7	MVS MESSAGE e.g. „IAT5210“
7- 7	1	-
8-15	8	JOBNAME
	.	
	.	
	.	
0- 6	7	MVS MESSAGE e.g. „CBR....“
	.	
	.	

Refer to "MVS System Messages" (IBM) for further possible message structures.

12.3 USER TMS EXIT

12.3.1 FUNCTION

The exit receives control when the **FREEVOL** command is executed. The exit is loaded during the HCC start with LOAD and is activated from the subtask ZHC05800 using the CALL macro:

```
         LA       R1,E1PARM   PARMLIST ZHCEX1
         L        RF,EP025    EP-ADDRESS
         BALR     RE,RF       CALL EXIT
         LTR      RF,RF       EXIT OK ?
         BNZ      error       NO
```

The parameter list to be passed is described in the ZHCEX1/TMS macro parameter list.

Task of the exit is to create a 6-byte table described by the E1CORE, E1TABA and E1TABE fields. This 6-byte table contains the volsers for SCRATCH cartridges to be used by HCC during substitution of non-specific MOUNTs.

After the exit has returned to subtask ZHC05800, the passed table is checked against the Archive mirror (example: for ejected cartridges) and then passed onto the subtask ZHC05900 (UPDATE ARCHIVE).

An entry is made in the VSAM Archive mirror depending on the option ADD (= add new SCRATCH TAPES) or REP (= replace all existing SCRATCH TAPE information). The HCC main task receives control when the UPDATE is finished and the INCORE Archive mirror is updated to the current status in the same manner as the VSAM Archive.

MOUNT requests pending due to a previous lack of scratch tapes are continued.

HCC releases the table created by the user exit (FREEMAIN).

12.3.2 FLOW

- Standard link conventions with own SAVEAREA
- Reusable code (reentrant not required)
- GETMAIN for 6-byte table, number of entries = max. number of scratch tapes expected
- DYNALLOC on the scratch tape file
- OPEN INPUT
- Read the log. record of the master file and determine available tapes using the structure of the tape management system.
- Enter available tapes in the 6-byte table
- CLOSE FREE
- Update parameter list HACEX1
- Standard RETURN

Examples can be found under X01.... in the hlq.ZHC....SAMP library.

The created EXIT must be defined under TMSEXIT in HACPARM1. The utility command **TMSCHECK** can be used to test the exit under TSO.

12.4 HEADER INFORMATION

The independent utility ZHC08000 can be used to display header information from HDR1 and HDR2 of the IBM standard label, as well as to initialize the headers.

12.4.1 JCL START PROCEDURE

The following JCL statements must then be added to the SYS1.PROCLIB or to the linked user.PROCLIB, whereby any member name can be chosen (LCK is used here for header display and TLI for initialization):

MEMBER NAME LCK:

```
//LCK       PROC V=ANYTAPE,U=unitname,FILENO=1
//ST010     EXEC PGM=ZHC08000,PARM='VOL=&V,UNIT=&U,FILENO=&FILENO'
//STEPLIB   DD   DSN=HLQ.ZHC....LOAD,DISP=SHR
//SYSUDUMP  DD   SYSOUT=A
```

MEMBER NAME TLI:

```
//TLI       PROC V=ANYTAPE,U=unitname
//ST010     EXEC PGM=ZHC08000,PARM='VOL=&V,UNIT=&U,TLI'
//STEPLIB   DD   DSN=HLQ.ZHC....LOAD,DISP=SHR
//SYSUDUMP  DD   SYSOUT=A
```

The PARM entries are different for both functions for consistency reasons: FILENO and TLI are mutually exclusive.

12.4.2 START COMMANDS

```
S LCK,V=volser[,U=unit,FILENO=n]      Display header

S TLI[,U=unit]                        Initialization
```

volser	VOLSER
unit	MVS unit address, esoteric or generic name (ROB0, 3420, 3480, 3490 ...)
n	FILE Number of the file on the tape

12.5 BATCH EXAMPLES

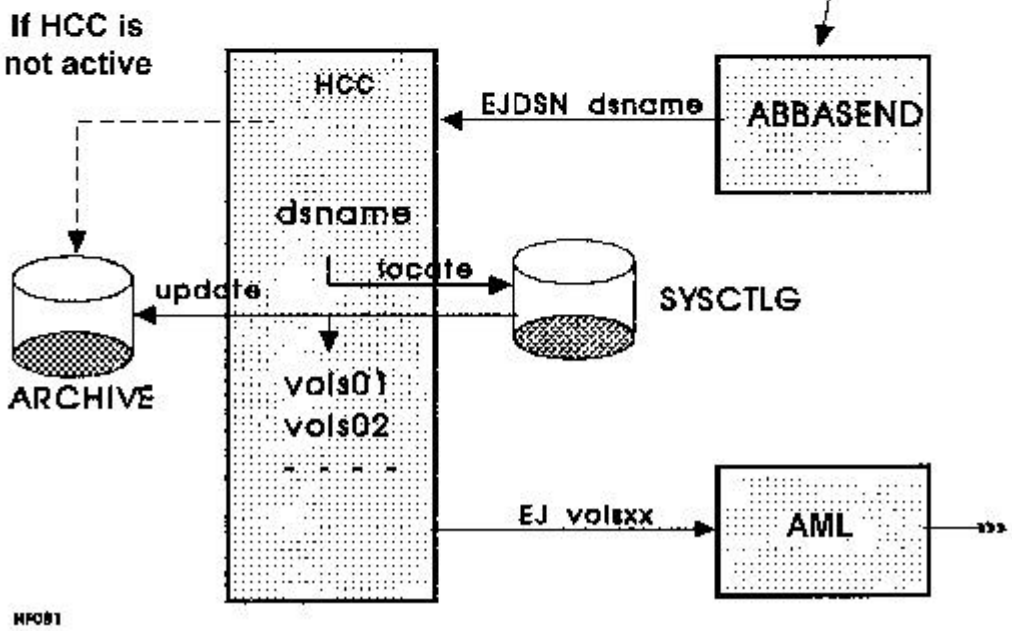
12.5.1 EJECTING CARTRIDGES USING FILE NAMES

Inserting the ABBASEND step causes immediate eject of all cartridges created and cataloged under the "dsname" file without any further organizational measures.

When HCC is not active, the information from ABBASEND is directly marked as "set for ejection" in the HCC Archive mirror.

The information is automatically moved to the HCC wait queue when HCC is restarted.

```
//*  ------------------------------------------------------------------
//*          DAILY DATASAVING
//*  ------------------------------------------------------------------
//ST020    EXEC PGM=utility
//DISK     DD   UNIT=disk
//TAPE     DD   DISP=(,CATLG,DELETE),DSN=dsname(+1),
//         UNIT=tape,VOL=(,,10)
//*  ------------------------------------------------------------------
//*          AUTOMATIC EJECT FROM CURRENT JOB
//*  ------------------------------------------------------------------
//ST030    EXEC PGM=ABBASEND,PARM='EJDSN dsname(0)'
//.....
```



Remark: The "dsname" file is cataloged after the execution of ST020 and must be requested with "dsname(0)" (GDG-specific).

12.5.2 EJECTING CARTRIDGES USING TMS INFORMATION

A daily administration job is normally run when a TAPE MANAGEMENT SYSTEM is implemented. ABBASEND can be used to activate the HCC task to transfer the new scratch information (FREEVOL ADD / REP) when this job is finished.

It may be necessary to simultaneously transform eject criteria of the tape management system directly into an eject command for the AML system.

This can be achieved by including a "Selection program" in the daily administration job; identified with program name X04EJ002 in the following.

```
//* ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
//*          TMS  MAINTENANCE
//* ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
//ST010     EXEC  PGM=tmsutil
//SYSPRINT  DD    DSN=&&LIST,DISP=(,PASS),...
//---
//* ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
//*          CONVERT  TO  HACC  COMMAND
//* ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
//ST020     EXEC  PGM=X04EJ002
//EJIN      DD    DSN=&&LIST,DISP=(OLD,PASS)
//EJOUT     DD    DSN=&&EJECT,DISP=(,PASS),DCB=(LRECL=80,...
//* ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
//*          SEND  TO  ABBA
//* ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
//ST030     EXEC  PGM=ABBASEND
//PARMIN    DD    DSN=&&EJECT,DISP=(OLD,DELETE)
//---
```

Fig. 9: Example for X04EJ002

Such information can be created under DFSMSrmm (TM) Version 1 as follows:

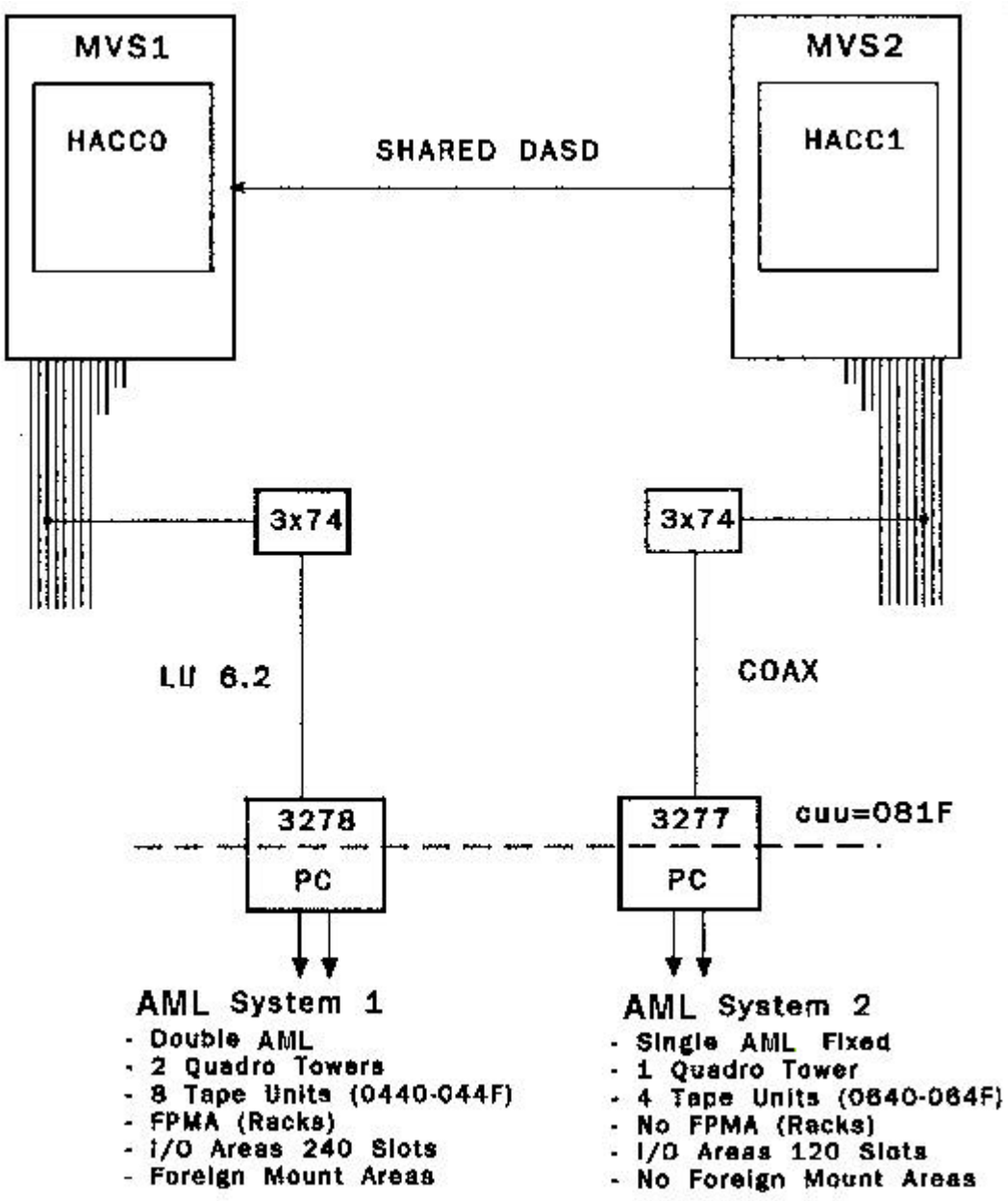
```
SEARCHVOLUME VOLUME(*) STATUS(SCRATCH) CLIST
                    or
SERCHVOLUME VOLUME(*) ACTION(NOTIFY CLIST('EJ','Exx')
```

Step ST020 can be omitted in this case.

Also refer to the Section on TMS= in the Installation/Customization-Manual or Section 9.28 in the IBM document DFSMS/MVS Version 1 Release 1 using DFSMSrmm (SC26-Y931-00).

12.6 HACCPARM EXAMPLES

12.6.1 HACCPARM EXAMPLE FOR AML/2



```
*=========================================================================*
* HACC VERSION 3.0.0 SYSTEM DEFINITIONS                                   *
*=========================================================================*
* THIS MEMBER IS AN EXAMPLE FOR OPERATION OF 2 ABBA SYSTEMS.              *
* THE CONFIGURATION IS ASSUMED AS                                         *
*-------------------------------------------------------------------------*
*-------------------------------------------------------------------------*
* JOB ENTRY SUBSYSTEM                                                     *
*-------------------------------------------------------------------------*
JES=JES2                                JOB ENTRY SUBSYSTEM
*
*-------------------------------------------------------------------------*
* PHYSICAL LIBRARY DEFINITION                                             *
*-------------------------------------------------------------------------*
* DEVICES SYSTEM 1, ROBOT 1,2                                             *
*-------------------------------------------------------------------------*
LDEV=T01,01,01,32,01,18,01,10,10,S=1,R=1,2         TOWER 1 - 5760 SLOTS
LDEV=T02,02,01,32,01,15,01,10,10,S=1,R=1,2         TOWER 2 - 4800 SLOTS
LDEV=T03,03,01,32,01,12,01,10,10,S=1,R=1,2         TOWER 3 - 3840 SLOTS
LDEV=R04,04,01,01,01,12,01,10,10,S=1,R=1           LINEAR RACK ROBOT 1
LDEV=R05,05,01,01,01,12,01,10,10,S=1,R=1           - 360 SLOTS
LDEV=R06,06,01,01,01,12,01,10,10,S=1,R=1
LDEV=R07,07,01,01,01,12,01,10,10,S=1,R=2           LINEAR RACK ROBOT 2
LDEV=R08,08,01,01,01,12,01,10,10,S=1,R=2           - 360 SLOTS
LDEV=R09,09,01,01,01,12,01,10,10,S=1,R=2
*
```

```
*-------------------------------------------------------------------------*
* DEVICES SYSTEM 1, ROBOT 1,2                                             *
*-------------------------------------------------------------------------*
LDEV=T01,01,01,32,01,15,01,10,10,S=2,R=1           TOWER 1 - 4800 SLOTS
*
*-------------------------------------------------------------------------*
* I/O-AREAS SYSTEM 1, ROBOT 1                                             *
*-------------------------------------------------------------------------*
LDEV=E01,00,02,02,13,23,01,10,10,S=1,R=1           EJ 110 SLOTS
LDEV=F01,00,03,03,24,24,01,10,10,S=1,R=1           FM  10 SLOTS
LDEV=I01,00,01,01,01,01,01,01,10,S=1,R=1           VI   1 SLOT
LDEV=I10,00,01,01,01,01,01,03,10,S=1,R=1           VI   3 SLOTS
LDEV=I11,00,01,01,01,01,01,05,10,S=1,R=1           VI   5 SLOTS
LDEV=I12,00,01,01,01,01,01,10,10,S=1,R=1           VI  10 SLOTS
LDEV=I13,00,01,01,01,03,01,10,10,S=1,R=1           VI  30 SLOTS
LDEV=I14,00,01,01,01,06,01,10,10,S=1,R=1           VI  60 SLOTS
LDEV=I15,00,01,01,01,09,01,10,10,S=1,R=1           VI  90 SLOTS
LDEV=I16,00,01,01,01,12,01,10,10,S=1,R=1           VI 120 SLOTS
LDEV=I19,00,01,01,01,24,01,10,10,S=1,R=1           VI 240 SLOTS
LDEV=W01,00,00,00,00,00,01,01,99,S=1,R=1           CARTRDGE POCKET
*
*-------------------------------------------------------------------------*
* I/O-AREAS SYSTEM 1, ROBOT 2                                             *
*-------------------------------------------------------------------------*
LDEV=E02,00,02,02,13,23,01,10,10,S=1,R=2           EJ 110 SLOTS
LDEV=F02,00,03,03,24,24,01,10,10,S=1,R=2           FM  10 SLOTS
LDEV=I02,00,01,01,01,01,01,01,10,S=1,R=2           VI   1 SLOT
LDEV=I20,00,01,01,01,01,01,03,10,S=1,R=2           VI   3 SLOTS
LDEV=I21,00,01,01,01,01,01,05,10,S=1,R=2           VI   5 SLOTS
LDEV=I22,00,01,01,01,01,01,10,10,S=1,R=2           VI  10 SLOTS
LDEV=I23,00,01,01,01,03,01,10,10,S=1,R=2           VI  30 SLOTS
LDEV=I24,00,01,01,01,06,01,10,10,S=1,R=2           VI  60 SLOTS
LDEV=I25,00,01,01,01,09,01,10,10,S=1,R=2           VI  90 SLOTS
LDEV=I26,00,01,01,01,12,01,10,10,S=1,R=2           VI 120 SLOTS
LDEV=I29,00,01,01,01,24,01,10,10,S=1,R=2           VI 240 SLOTS
LDEV=W02,00,00,00,00,00,01,01,99,S=1,R=2           CARTRDGE POCKET
*
*-------------------------------------------------------------------------*
* I/O-AREAS SYSTEM 2, ROBOT 1                                             *
*-------------------------------------------------------------------------*
LDEV=E03,00,02,02,05,09,01,25,25,S=2,R=1           EJ 120 SLOTS
LDEV=I03,00,01,01,01,04,01,25,25,S=2,R=1           VI 120 SLOTS
LDEV=W03,00,00,00,00,00,03,03,99,S=2,R=1           PROBLEMBOX
*
*-------------------------------------------------------------------------*
* LOGICAL REDEFINITIONS SYSTEM 1                                          *
*-------------------------------------------------------------------------*
LDEV=T02,02,01,32,15,15,01,10,10,S=1,R=1,2,DYN          DYNAMIC AREA
LDEV=R04,04,01,01,01,12,01,10,10,S=1,R=1,FPMA           FPMA ROBOT 1
LDEV=R05,05,01,01,01,12,01,10,10,S=1,R=1,FPMA
LDEV=R06,06,01,01,01,12,01,10,10,S=1,R=1,FPMA
LDEV=R07,07,01,01,01,12,01,10,10,S=1,R=2,FPMA           FPMA ROBOT 2
LDEV=R08,08,01,01,01,12,01,10,10,S=1,R=2,FPMA
LDEV=R09,09,01,01,01,12,01,10,10,S=1,R=2,FPMA
*
*-------------------------------------------------------------------------*
* HACC DATASET NAMES (PRODUCTION)                                         *
*-------------------------------------------------------------------------*
PDSNARCH=OBISD.ZHC300.HACCARCH
PDSNEXC1=OBISD.ZHC300.HACCEXC1
PDSNMLG1=OBISD.ZHC300.HACCMLG1
PDSNMLG2=OBISD.ZHC300.HACCMLG2
PDSNMLG3=OBISD.ZHC300.HACCMLG3
PDSNMLG4=OBISD.ZHC300.HACCMLG4
PDSNSTMS=OBISD.TAPE.MANAGE.CATALOG
PDSNULOG=OBISD.ZHC300.HACCULOG
*
```

```
*-------------------------------------------------------------------*
* ABBA CONFIGURATION                                                *
*-------------------------------------------------------------------*
SYSDEF1=1,2                          SYSTEM 1 WITH ROB 1 & 2
SYSDEF2=1                            SYSTEM 2 WITH ROB 1
*
*-------------------------------------------------------------------*
* TSO SECURITY (IF NO RACF AUTHORISATION CALL)                      *
*-------------------------------------------------------------------*
TSOSEC=SYSOPR1,ABS=Y,HAA=R,HAAIND=ZHC300
TSOSEC=SYSOPR2,ABS=Y,HAA=N,HAAIND=ZHC300
TSOSEC=SYSPRG1,ABS=Y,HAA=U,HAAIND=ZHC300
TSOSEC=SYSPRG2,ABS=Y,HAA=U,HAAIND=ZHC300
*
*-------------------------------------------------------------------*
* 3480/3490-ABBA-UNITS SYSTEM 1                                     *
*-------------------------------------------------------------------*
UNIT=0440,01,1,1,FPMALOC=04010101,06011210
UNIT=0441,02,1,1,FPMALOC=04010101,06011210
UNIT=0442,03,1,1,FPMALOC=04010101,06011210
UNIT=0443,04,1,1,FPMALOC=04010101,06011210
UNIT=0444,05,1,2,FPMALOC=07010101,09011210
UNIT=0445,06,1,2,FPMALOC=07010101,09011210
UNIT=0448,07,1,2,FPMALOC=07010101,09011210
UNIT=0449,08,1,2,FPMALOC=07010101,09011210
*
*-------------------------------------------------------------------*
* 3480/3490-ABBA-UNITS SYSTEM 2                                     *
*-------------------------------------------------------------------*
UNIT=0640,09,2,1
UNIT=0641,10,2,1
UNIT=0642,11,2,1
UNIT=0643,12,2,1
*
*-------------------------------------------------------------------*
* SCRATCH TAPE ACCESS BY SCRATCHGROUP-ID'S                          *
*-------------------------------------------------------------------*
VOLGR=EX0000,EX0499,010,S=1,EXTERN
VOLGR=100001,115000,050,S=1,PRIVAT
VOLGR=100001,115000,050,S=1,SCRTCH
*
*-------------------------------------------------------------------*
* HACC-ABBA COMMUNICATION PATH EXCP                                 *
*-------------------------------------------------------------------*
COMDEF2=0033,EXCP,SYS=MVS2             COMMUNICATION DEVICES SYS2,MVS2
COMDEF2=081F,EXCP,SYS=MVS1             COMMUNICATION DEVICES SYS2,MVS1
*
*-------------------------------------------------------------------*
* DEFINITION OF APPC-CONECTIONS                                     *
*-------------------------------------------------------------------*
HACNET HID=A1,LU=OBIAMUR,LU2=OBISAMUR,LM=OBISL62S,TPN=H01RTP
HACNET HID=H1,LU=OBISACB1,LM=OBISL62S
HACNET HID=H2,LU=OBISACB2,LM=OBISL62S
*
*-------------------------------------------------------------------*
* WAIT ON COMPLETION OF A HACC COMMAND                              *
*-------------------------------------------------------------------*
ABSWAIT=YES
*
```

```
*-------------------------------------------------------------------*
* INITIATE HACC COMMANDS AFTER START AND/OR REPEAT PERIODICALLY      *
*-------------------------------------------------------------------*
AUTOCMD='ACOM 1',INTV=15
AUTOCMD='ACOM 2',INTV=15
AUTOCMD='DU',INTV=6
AUTOCMD='LOGSTART'
AUTOCMD='OS MN DSNAME'
AUTOCMD='SYSLOG ON'
*
*-------------------------------------------------------------------*
* INITIATE FREEVOL-COMMAND AT SCRATCH TAPE SHORTAGE                  *
*-------------------------------------------------------------------*
AUTOFREEVOL=YES
*
*-------------------------------------------------------------------*
* AUTOREPEAT AFTER COMMUNICATION TIMEOUT - REPEAT AFTER TWO MINUTES  *
*-------------------------------------------------------------------*
AUTOREPEAT=YES
*
*-------------------------------------------------------------------*
* INTERNAL WORK BUFFER SIZE                                          *
*-------------------------------------------------------------------*
BUFNO ARC=0500
BUFNO CUU=0150
BUFNO EXC=0200
BUFNO MSG=0150
BUFNO SSI=0200
*
*-------------------------------------------------------------------*
* TAPE CLEANING ACTIVITY                                             *
*-------------------------------------------------------------------*
CLEAN=40,500,80
*
*-------------------------------------------------------------------*
* SCRATCH TAPE ACCESS BY GENERIC DSNAME                              *
*-------------------------------------------------------------------*
DSNGR=EX0000,EX0099,005,EXT.DSN1
DSNGR=EX0100,EX0499,010,EXT.DSN2
*
*-------------------------------------------------------------------*
* EJECT SELECTION BY GENERIC DSNAME                                  *
*-------------------------------------------------------------------*
EJDSN=02,1,EXT.DSN1
EJDSN=02,1,EXT.DSN2
*
*-------------------------------------------------------------------*
* EJECT SELECTION BY VOLSER-RANGE                                    *
*-------------------------------------------------------------------*
EJVOL=01,1,000001,010000
EJVOL=01,1,100001,110000
*
*-------------------------------------------------------------------*
* EXTERNAL SECURITY DEFINITION (RACF)                                *
*-------------------------------------------------------------------*
EXTSEC=RACF,CLASS=$HACC,RESPREF=$HACC.CMD
*
*-------------------------------------------------------------------*
* DEFINE SCRATCH GROUPS TO FPMA AREA                                 *
*-------------------------------------------------------------------*
FPMADEF=04010101,06011210,S=1,PRIVAT (=SCRTCH)           FOR ROBOT 1
FPMADEF=07010101,09011210,S=1,PRIVAT (=SCRTCH)           FOR ROBOT 2
*
```

```
*-------------------------------------------------------------------*
* UNLOAD CONTROL OF FPMA WHILE KEEP                                 *
*-------------------------------------------------------------------*
FPMAKEC=YES                          MOVE TO HOME AFTER KEEP
*
*-------------------------------------------------------------------*
* FPMA LOAD/UNLOAD PROCESSING                                       *
*-------------------------------------------------------------------*
FPMALOAD=URO                         FPMA LOAD BY UNIT ROTATION
*-------------------------------------------------------------------*
* CONTROL OF GENERATION OF MOVE COMMANDS TO FPMA AREA               *
*-------------------------------------------------------------------*
FPMAMVMAX=10                         MAX.NO.OF INTERNAL GENERATED MOVE'S
*
*-------------------------------------------------------------------*
* WAITTIME FOR ISSUEING KEEP-MESSAGE, DEFAULT=00                    *
*-------------------------------------------------------------------*
* KEEPWT=SS(,RR,DDD)         DEFAULT=00                             *
*-------------------------------------------------------------------*
KEEPWT=10,03,045
*
*-------------------------------------------------------------------*
* LANGUAGE SELECTION                                                *
*-------------------------------------------------------------------*
LANG=E                                  MESSAGES IN ENGLISH LANGUAGE
*
*-------------------------------------------------------------------*
* DEFINE ABBASEND OUTPUT BUFFER (ISPF)                              *
*-------------------------------------------------------------------*
MAXTSO USERS=5,BUFFN=200,LOC=ABOVE
*
*-------------------------------------------------------------------*
* MSG AUTOREPLY                                                     *
*-------------------------------------------------------------------*
MSG=ARC*,AUTOREPLY
MSG=IEC507D,REPLY=M
MSG=IEC510D,REPLY=F
MSG=IEC534D,REPLY=U
MSG=IEC701D,REPLY=M
MSG=IEF238D,AUTOREPLY
MSG=IEF433D,REPLY=NOHOLD
*
*-------------------------------------------------------------------*
* PREFIX FOR HACC MESSAGES                                          *
*-------------------------------------------------------------------*
MSGPREFIX=OBI
*
*-------------------------------------------------------------------*
* BASE PRIORITY FOR ESSENTIAL ABBA COMMANDS                         *
*-------------------------------------------------------------------*
*-------------------------------------------------------------------*
PRTY EJ=25
PRTY JOB=A,INCR=+7
PRTY JOB=HSM,INCR=07,G
PRTY JOB=IMSA1,INCR=+11
PRTY JOB=J0000001,INCR=1
PRTY JOB=J0000002,INCR=2
PRTY JOB=J0000003,INCR=3
PRTY JOB=PRODA,INCR=11
PRTY JOB=TESTA,INCR=-10
PRTY KE=30
PRTY MO=35
PRTY MV=20
PRTY SCH=30
PRTY VI=26
*
```

```
*--------------------------------------------------------------------*
* ROUTINGCODES FOR CONSOLE MESSAGES                                  *
*--------------------------------------------------------------------*
ROUTCD=1,3,5,11
*
*--------------------------------------------------------------------*
* NUMBER OF USER WRITTEN SMF-RECORDS                                 *
*--------------------------------------------------------------------*
SMFRECORD=250
*
*--------------------------------------------------------------------*
* SUBSYSTEM NAME                                                     *
*--------------------------------------------------------------------*
SSINAME=HAC0
*
*--------------------------------------------------------------------*
* HACC STATISTICS                                                    *
*--------------------------------------------------------------------*
STATISTICS=YES
*
*--------------------------------------------------------------------*
* INTERVALL TIMER INTERUPT PERIOD TRANFER LOG                        *
*--------------------------------------------------------------------*
STIMER=02                              CPU EXCHANGE LOG INTERVALL TIMER
*
*--------------------------------------------------------------------*
* MAX.NO.OF SWAPS PER UNITS (0-2)                                    *
*--------------------------------------------------------------------*
SWAPLIM=1                       1 SWAP ALLOWED
*
*--------------------------------------------------------------------*
* PASSWORD FOR TAPE LABEL INITIALIZATION                             *
*--------------------------------------------------------------------*
TLIPW=NOPW
*
*--------------------------------------------------------------------*
* STANDARD TMS EXIT                                                  *
* ATTENTION: TMS & TMSEXIT ARE MUTUALLY EXCLUSIVE                    *
*--------------------------------------------------------------------*
TMS=UCV5,EX0000,115000,5000
*
*--------------------------------------------------------------------*
* TSO SECURITY (IF NO RACF AUTHORISATION CALL)                       *
* ATTENTION: AT LEAST ONE TSOSEC STATEMENT IS REQUIRED               *
*--------------------------------------------------------------------*
TSOSEC=OBI1GA1,HAA=R,ABS=Y
TSOSEC=OBI1KF1,HAA=U,ABS=Y,HAAIND1=OBISD
TSOSEC=OBI1KF2,HAA=U,ABS=Y,HAAIND1=OBISD
*
*--------------------------------------------------------------------*
* UCB ATTENTION INDEX                                                *
*--------------------------------------------------------------------*
UCBATI=08
```

12.6.2 HACCPARM EXAMPLE FOR ABBA/1

```
*=====================================================================*
*        HACC Version 3.0.0 System Definitions for ABBA/1             *
*=====================================================================*
SSINAME=HAC0                          Subsystem name in SYS1.PARMLIB
JES=JES2                              JES2 installed
ROUTCD=1,3,5,11                       Routing codes for HACC messages
LANG=E                                HACC messages in English
STAM=NO                               STAM not installed
*
*---------------------------------------------------------------------*
*----File names Production- ------------------------------------------*
*---------------------------------------------------------------------*
PDSNARCH=OBI1D.HACCDOKU.ARCHIVE
PDSNSTMS=OBI1D.HACCDOKU.TMSCAT
PDSNULOG=OBI1D.HACCDOKU.UNITLOG
*
PDSNMLG1=OBI1D.HACCDOKU.MSGLOG1
PDSNMLG2=OBI1D.HACCDOKU.MSGLOG2
PDSNMLG3=OBI1D.HACCDOKU.MSGLOG3
PDSNMLG4=OBI1D.HACCDOKU.MSGLOG4
*
PDSNEXC1=OBI1D.HACCDOKU.EXCHNG1,RECNO=10
PDSNEXC2=OBI1D.HACCDOKU.EXCHNG2,RECNO=15
*
*---------------------------------------------------------------------*
*-----Hardware / Redefinitions----------------------------------------*
*---------------------------------------------------------------------*
SYSDEF1=1,2                           1 twin-robot system
*
*---------------------------------------------------------------------*
*----Towers / Access with robots 1 + 2--------------------------------*
*---------------------------------------------------------------------*
LDEV=T01,01,01,32,01,15,01,10,10,S=1,R=1,2
LDEV=T02,02,01,32,01,15,01,10,10,S=1,R=1,2
LDEV=T03,03,01,32,01,15,01,10,10,S=1,R=1,2
LDEV=T04,04,01,32,01,15,01,10,10,S=1,R=1,2
*
*---------------------------------------------------------------------*
*----Redefinition for HWS in tower 2----------------------------------*
*---------------------------------------------------------------------*
LDEV=T02,02,32,32,01,01,01,10,10,S=1,R=1,2,HWS
*
*---------------------------------------------------------------------*
*----I/O rack robot 1-------------------------------------------------*
*---------------------------------------------------------------------*
LDEV=I01,00,01,01,01,08,01,20,25,S=1,R=1    00010101-00010820 - 190
LDEV=I11,00,01,01,01,04,01,25,25,S=1,R=1    00010101-00010425 - 100
LDEV=I91,00,01,01,01,01,01,01,01,S=1,R=1    00010101-00010101 - 1
*
LDEV=E01,00,02,02,01,08,01,20,25,S=1,R=1    00020101-00020820 - 190
LDEV=E11,00,02,02,05,08,01,20,25,S=1,R=1    00020501-00020820 - 90
LDEV=E91,00,02,02,08,08,01,20,25,S=1,R=1    00020801-00020820 - 20
*
LDEV=F01,00,03,03,08,08,20,25,25,S=1,R=1    00030820-00030825 - 5
LDEV=W01,00,00,00,00,00,01,01,99,S=1,R=1    00000001
*
```

```
*------------------------------------------------------------------------*
*----Control-------------------------------------------------------------*
*------------------------------------------------------------------------*
PRTY EJ=25
PRTY KE=30
PRTY MO=35
PRTY MV=20
PRTY SCH=20
PRTY VI=10
PRTY JOB=A,INCR=+7
PRTY JOB=OBISA,INCR=+22
PRTY JOB=RZP2B,INCR=-55
PRTY JOB=J0000001,INCR=1
*
STIMER=02
KEEPWT=05,03,030
CLEAN=50,300,40
*
*------------------------------------------------------------------------*
*----Organizational functions--------------------------------------------*
*------------------------------------------------------------------------*
TLIPW=NOPW
*
EJDSN=01,1,CFIMSB
EJDSN=02,1,CF99B.SOS
EJVOL=01,1,000000,010999
EJVOL=03,1,012000,012999
*
*------------------------------------------------------------------------*
*----Automatic commands--------------------------------------------------*
*------------------------------------------------------------------------*
AUTOREPEAT=YES
AUTOCMD=LOGSTART
AUTOCMD=SYSLOG ON
AUTOCMD='DU',INTV=6
*
*------------------------------------------------------------------------*
*----Auto-Reply----------------------------------------------------------*
*------------------------------------------------------------------------*
MSG=ARC*,AUTOREPLY
MSG=IEF238D,AUTOREPLY
MSG=IEF433D,REPLY=NOHOLD
MSG=IEC507D,REPLY=M
MSG=IEC510D,REPLY=F
MSG=IEC534D,REPLY=U
MSG=IEC701D,REPLY=M
SWAPLIM=1                                  1 SWAP allowed
*
*-------------- END -----------------------------------------------------*
```

12.6.3 HACCPARM EXAMPLE FOR FPMA

FPMA (FIXED PREFERRED MOUNT AREA) is usually linear storage installed as close as possible to the tape station. FPMA only serves to optimize scratch tape processing.

The FPMALOC parameter can assign each tape drive to a certain area within the FPMA.

12.6.3.1 FPMA WITH 3490 (ABBA/1)

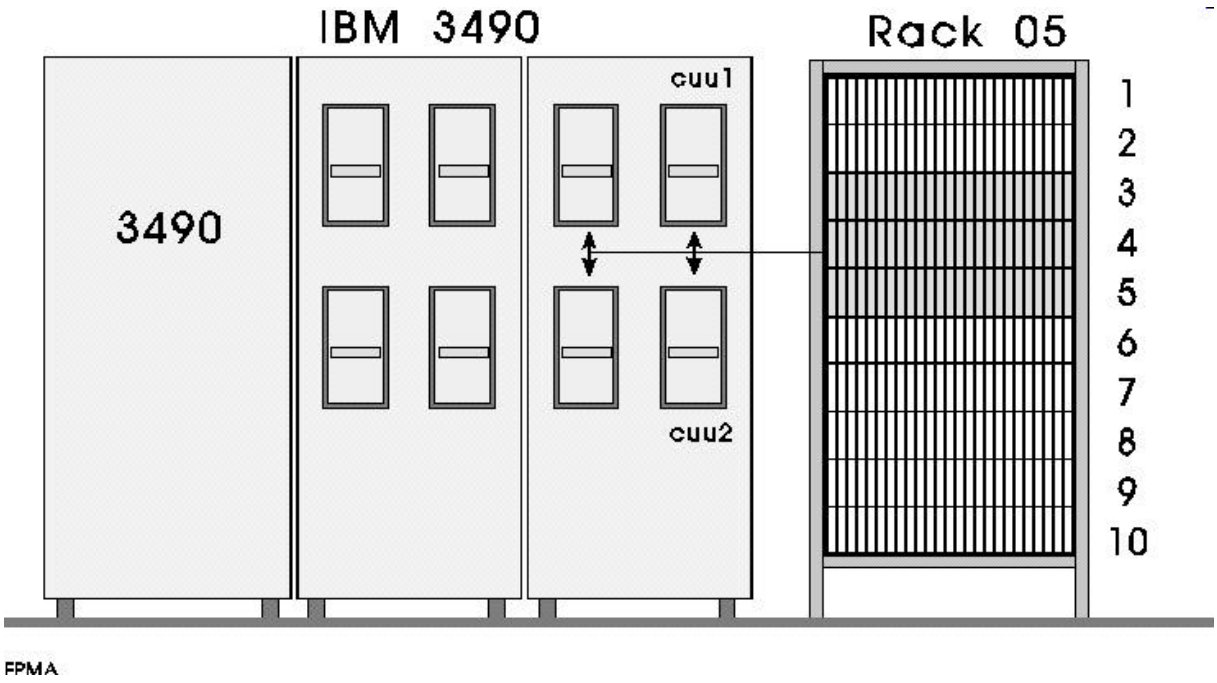


Fig. 10: FPMA with IBM 3490 for ABBA/1

The FPMA for ABBA/1 is defined as a rack as follows:

```
LDEV=R05,05,01,01,01,10,01,25,25,S=1
LDEV=R05,05,01,01,01,10,01,25,25,S=1,FPMA
...
FPMADEF=05010101,05011025,S=1,PRIVAT
...
UNIT=0462,03,1,1,FPMALOC=05010101,05011025
UNIT=0463,04,1,1,FPMALOC=05010101,05011025
```

12.6.3.2 FPMA WITH 3490 (AML/2)

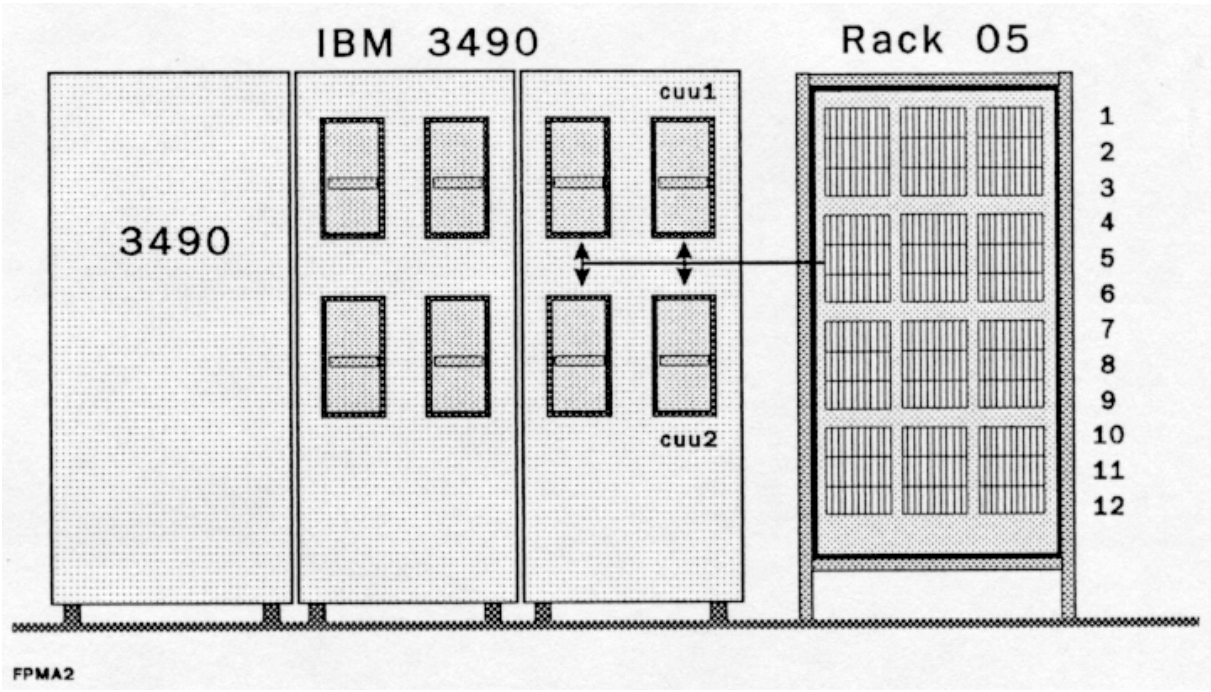


Fig. 11: FPMA with IBM 3490 for AML/2

The FPMA for AML/2 is defined as a rack as follows:

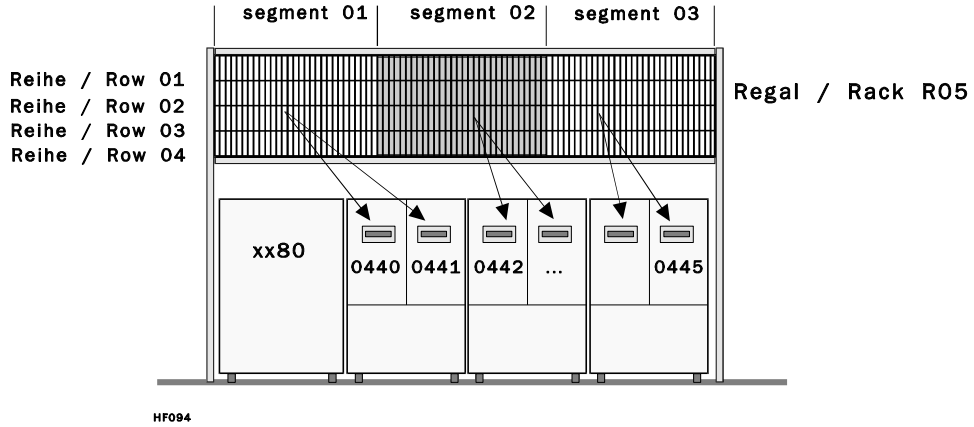
```
LDEV=R05,05,01,01,01,12,01,10,10,S=1
LDEV=R06,06,01,01,01,12,01,10,10,S=1
LDEV=R07,07,01,01,01,12,01,10,10,S=1
LDEV=R05,05,01,01,01,12,01,10,10,S=1,FPMA
LDEV=R06,06,01,01,01,12,01,10,10,S=1,FPMA
LDEV=R07,07,01,01,01,12,01,10,10,S=1,FPMA
...
FPMADEF=05010101,07011210,S=1,PRIVAT
...
 UNIT=0462,03,1,1,FPMALOC=05010101,07011210
UNIT=0463,04,1,1,FPMALOC=05010101,07011210
```

12.6.3.3 FPMA WITH 3480

All FPMALOCs overlap in this example because path optimization is hardly possible for the units.

FPMA control

FPMA [s,r,]	SLU	Start LOAD/UNLOAD
	PLU	STOP
	UNL	UNLOAD all
	SMK	Start MOUNT/KEEP
	PMK	Stop



HACPARM1 definitions (example)

```
LDEV=R05,05,01-03,01-04,01-25,S=1,FPMA          (phys.coordinates)

FPMADEF=05010101-05030425,S=1,PRIVAT            (subareas & groups)

UNIT=0440,01,1,1,FPMALOC=05010101-05010425 (unit assignments)
UNIT=0441,02,1,1,FPMALOC=05010101-05010425
UNIT=0442,03,1,1,FPMALOC=05020101-05020425
 ...
```

HF094.CDR

Fig. 12: FPMA with IBM 3480

Definition examples:

```
LDEV=R05,05,01,04,01,05,01,12,12,s=1
LDEV=R05,05,01,04,01,05,01,12,12,S=1,FPMA
...
FPMADEF=05020101,05050512,S=1,PRIVAT
...
UNIT=440,01,1,1,FPMALOC=05010101,05010412..
UNIT=441,02,1,1,FPMALOC=05010101,05010412
UNIT=442,03,1,1,FPMALOC=05020101,05020412
UNIT=443,04,1,1,FPMALOC=05020101,05020412
UNIT=444,05,1,1,FPMALOC=05030101,05030412
UNIT=445,06,1,1,FPMALOC=05030101,05030412
...
FPMALOAD=URO
```

Load by rotation starting at FPMALOC 440, then 441 and so on.

Units 440/441: common area in Segment 1.
Units 442/443: common area in Segment 2.
Units 444/445: common area in Segment 3.

AML/2 no longer supports FPMA for 3480.

12.7 MVS MESSAGES

The following messages from the MVS operating system, including HSM and non-IBM software products, must be available to HCC through SSI broadcasting. No message suppression mechanism affecting the required messages may be active, for example, IEECVXIT or JES3exits.

The exact message formats can be found in the corresponding literature.

12.7.1 MOUNT MESSAGES

```
IEC400A M cuu,volser,...
IEC501A M cuu,volser,...
IEC705I TAPE ON cuu,volser...
IEF233A M cuu,volser,...
IEF233D M cuu,volser,...
IAT5210 ...
```

12.7.2 KEEP MESSAGES

```
IEC502E (K¦R¦D¦RK¦RD),cuu,volser,...
IEF234E (K¦R¦D) cuu,volser,...
IAT5410 ...
```

12.7.3 EXCEPTION MESSAGES

```
IEC135A U cuu,volser      (not file protected RACF)
IEC507D E cuu,volser,...  (expiration date not occurred)
IEC509A F cuu,volser      (file protect ring required)
IEC512I LBL ERR cuu ....
IEC701D M cuu,..volser    (IEHINIT tape labeling)
IEC703I cuu,VOLUME IS FILE PROTECTED
IGF500I  SWAP cuu to cuu ...
IGF502E  PROCEED WITH SWAP OF cuu TO cuu
IGF503I  ERROR ON cuu,SELECT NEW DEVICE
IGF505I  SWAP FROM cuu TO cuu COMPLETE
IGF509I  SWAP cuu - I/O ERROR
IEF244I jobname,step - UNABLE TO ALLOCATE...
IEF289I jobname,step,ddname NEEDS...
IEF247I jobname - cuu,...
IEF290E jobname,step - UNABLE TO ALLOCATE...
IEF703E jobname NEEDS 1 UNIT(S) nnn
IAT5918 ...
IAT5310 ...
```

12.7.4 WTOR MESSAGES

```
IEC507D REPLY 'U'-USE or 'M'-UNLOAD...
IEC510D F cuu,volser      (file protect ring required)
IEC534D A ddd,volser,... REPLY 'U'=USE OR 'M'=UNLOAD
IEF433D jobname - WAIT REQUESTED - REPLY HOLD OR NOHOLD
IEF238D jobname - n UNIT(S) NEEDED ...
IGF500D  REPLY 'YES','DEVICE','NO'...
IGF509D  REPLY DEVICE OR 'NO'
```

12.7.5 HSM WTOR MESSAGES

```
ARC0310A CAN TAPE ...
ARC0311A SYSTEM STIMER ...
ARC0312A TAPES ... NEEDED
ARC0314A CAN THE ...
ARC0366A REPLY ...
ARC0380A RECALL WAITING ...
ARC0381A ALLOCATION REQUEST ...
```

12.7.6 DFSMSrmm MESSAGES

```
EDG6620I VOLUME volser(oldvolser) INITIALISATION AND VARYFICATION
         SACCSESFUL-RETURN TO RACK NUMBER rack_number

EDG6622I VOLUME volser(oldvolser) INITIALISATION SACCSESFUL-RETURN TO RACK
         NUMBER rack_number

EDG6623I VOLUME volser(oldvolser) ERASE, INITIALISATION AND VARYFICATION
         SACCSESFUL-RETURN TO RACK NUMBER rack_number

EDG6625I VOLUME volser(oldvolser) ERASE AND INITIALISATION SACCSESFUL-
         RETURN TO RACK NUMBER rack_num

EDG6627A M drive_nr V(volser) R(rack_nr) TO BE indicated_action, label_type

EDG6642I VOLUME volser LABELLED SUCCESSFULLY
```

HCC Release 3.0.0 only supports the **MNTMSG** command within the Parmlib member EDGRMMxx inasmuch as the specified sample cannot be modified and the existing, fixed MVS/JES(3) messages, and therefore interceptible, cannot be changed.

12.7.7 OAM MESSAGES

```
CBR1000I OAM LABEL COMMAND EXECUTION SCHEDULED

CBR2102I LABEL FUNCTION COMPLETE FOR VOLUMES vvvvvv AND vvvvvv.

CBR2211E ENTER AN OPTICAL DISK INTO LIBRARY xxxxxxxx TO RELIEVE
         THE OUT OF SPACE CONDITION IN STORAGE GROUP XXXXXXXX.

CBR2600A SPECIFY SHELF LOCATION FOR VOLUMES vvvvvv AND vvvvvv.

CBR3122I VOLUMES vvvvvv AND vvvvvv WERE EJECTED FROM LIBRARY
         lib1. PLACE IN SHELF LOCATION xxxxxxxx.

CBR4400A MOUNT VOLUME XXXXXX ON DRIVE DRV5.SHELF LOCATION IS xxxxxxxx

CBR4401I VOLUME vvvvvv MOUNTED ON DRIVE dddddddd.

CBR4402I DEMOUNT VOLUME XXXXXX ON DRIVE DRV5.SHELF LOCATION IS xxxxxxxx

CBR4405D ENTER VOLSER FOR VOLUME ON DRIVE dddddddd.

CBR4406D ENTER OWNER INFORMATION FOR VOLUME vvvvvv ON DRIVE dddddddd.

CBR4418I INVALID LABEL OPERATION ON DRIVE dddddddd VOLUME SERIAL
         NUMBER vvvvvv.

CBR4423D ENTER SHELF INFORMATION FOR VOLUME vvvvvv ON DRIVE dddddddd.

CBR4424D VOLSER ENTERED FOR UNLABELED VOLUME IN DRIVE dddddddd
         IS vvvvvv. REPLY 'U' TO USE THIS VOLSER OR, 'R' TO RETRY.

CBR4426D MOUNT PENDING FOR VOLUME vvvvvv ON DRIVE dddddddd.
         REPLY 'R' TO RETRY OR 'C' TO CANCEL THIS REQUEST.

CBR4430A REMOVE AND FLIP CARTRIDGE ON DRIVE XXXXXXXX.

CBR4432D ENTER STORAGE GROUP NAME FOR VOLUMES A-VOLS AND B-VOLS, OR
         REPLY 'U' TO ASSIGN TO SCRATCH.

CBR4438D VOLUME IN DRIVE dddddddd HAS UNRECOGNIZED MEDIA
         FORMAT. REPLY 'F' TO FORMAT OR 'C' TO CANCEL.

CBR4439D ENTER VOLSER FOR OPPOSITE SIDE OF VOLUME vvvvvv IN
         DRIVE dddddddd.
```

12.7.8 NON-IBM MESSAGES

12.7.8.1 CA1

```
TMSxxxx                                    (Max. 6 bytes for volser or pool)
CA$F810A  M...                             (Init as from Version 5.1)
CA$F813A  K...
```

12.7.8.2 TLSE

```
TLSO178A  MOUNT VOLSER VOLUME. UNIT=CUU
```

12.7.8.3 EPIC/MVS

```
EPT001A   M cuu,...                        (Max. 6 bytes for volser or pool)
EPT166A   MOUNT VOLUME nnnnnn ON DRIVE cuu
EPT167A   VOLUME nnnnnn INITIALIZED ON DRIVE cuu
```

12.7.8.4 STAM

```
STM202I
```

12.7.8.5 CONTROL-T

```
CTT100A   M cuu,volser,...
CTT101A   M cuu,poolid,...                 (Max. 6 bytes for volser or pool)
CTT301I   VOLUME volume BECAME SCRATCH
```

12.8 CLEARSSI UTILITY

12.8.1 DESCRIPTION

PTF ZY00069 includes a new utility that serves to deactivate a subsystem used for HCC and to release the CSA storage (ZHCCVT, ZHCRCA, the SSI buffer queue and the storage for ZHC01200).

12.8.2 CALL

```
//JOB1    valid job card
//*
//STEP1   EXEC PGM=ZHC17000,PARM='ssss'
//STEPLIB DD   DSN=hlq.ZHC300.LOAD,DISP=SHR
```

ssss = Subsystem name

This job is stored in the SAMPLIB under CLEARSSI.

12.8.3 MESSAGES

`HAC900I PARM INPUT FOR ZHC17000 has to be 4 Bytes in length`
- The SSINAME must be entered as 4 bytes.

`HAC901I SSINAME IS ssss`
- This message displays the SSINAME entered ssss.

`HAC902I ENTER "Y" TO CONFIRM SSI-NAME OR "N" TO REJECT`
- Confirm the entered SSINAME with "Y", or reject with "N".

`HAC903I SSINAME FROM INPUT REJECTED`
- This message appears when HAC802I is replied to with "N".

`HAC904I INVALID REPLY. ONLY "Y" OR "N" ARE ALLOWED.`
- The reply to HAC802I or HAC808I was not "Y" or "N".

`HAC905I SSINAME .... NOT DEFINED`
- The entered SSINAME ssss is invalid.

`HAC906I SSINAME .... DOES NOT POINT TO HACCVT`
- This message can have two meanings:
    1.) The SSCT user field (SSCTSUSE) has NO address (X'0') --> SSI is free
    2.) The address in SSCTSUSE does NOT point to a ZHCCVT control block (for example, another application is using this SSI).

HAC907I SSINAME .... POINTS TO A HACCVT
- Confirmation that SSCTSUSE points to a HACCVT control block.

HAC908I REPLY "Y" TO FREE ALL STORAGE FOR THIS SSI, OR "N" TO CANCEL THE REQUEST.
- Enter whether the structures belonging to SSINAME should be released "Y" or not "N".

HAC909I REQUEST CANCELLED
- This message appears when HAC808I is answered with "N".

HAC910I SSI FUNCTIONS FOR .... DEACIVATED
- This message appears when the SSI function in SSCT has been deactivated.

HAC911I STORAGE FOR xxxxxxxx has been freed
- This message appears when the following storage has been released:
  - SSI-BUFFER-QUEUE
  - SFM-MODULE ZHC01200
  - HACCVT

12.9 CONFIGURATION EXAMPLES

12.9.1 SINGLE HOST COMPLEX

This configuration shows 1-36 MVS systems active on one or more CPUs and with access to one or more tape pools.

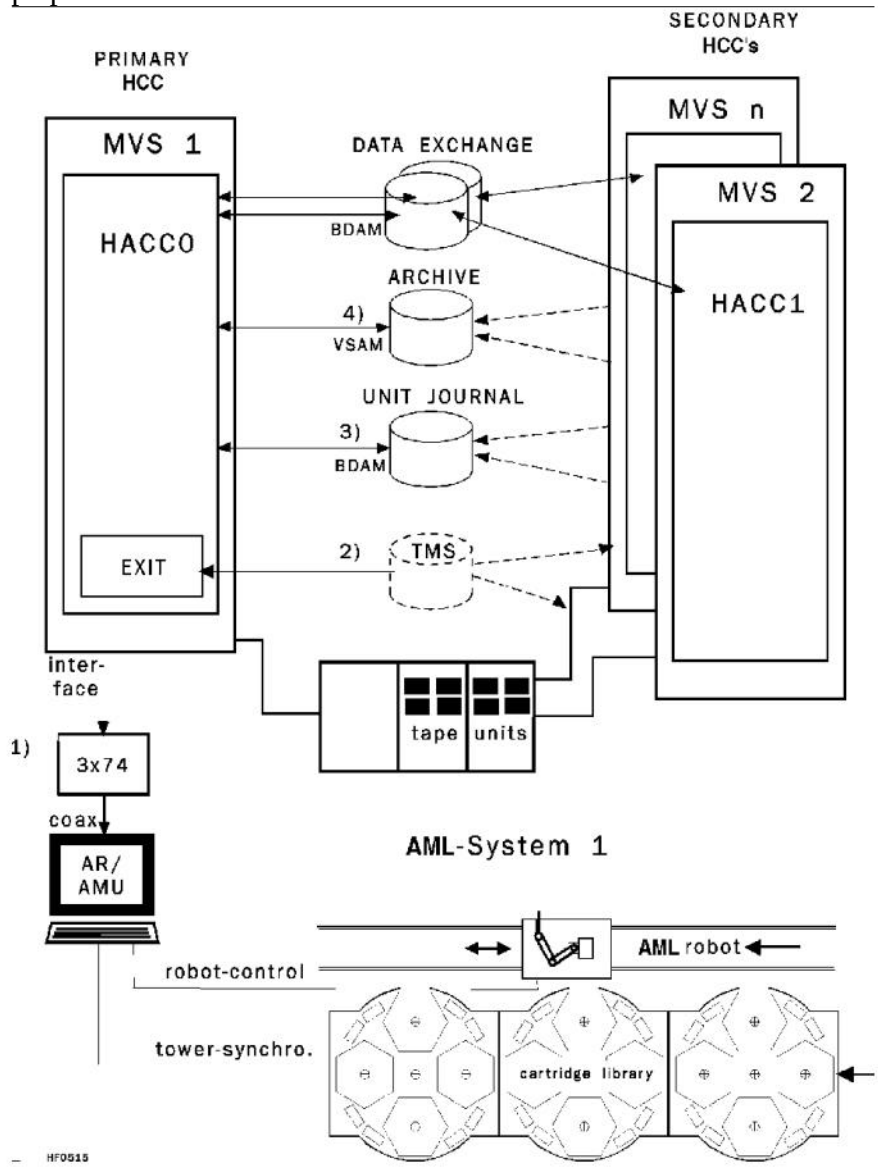


Fig. 13: Single host complex

A HCC system task is active on each of the CPUs involved. Only one system task assumes control of the AML system: the PRIMARY TASK HACC0. Any CPU can be selected for HACC0. A SECONDARY TASK HACCx (x=1-Z) is then active on each of the other CPUs.

12.9.2 EXAMPLE OF AN INSTALLATION CONFIGURATION

This example shows how HCC is integrated within a HOST complex from a technical viewpoint. The MVS-MVS intercommunication runs over shared DASD: HACC EXCHANGE LOG.

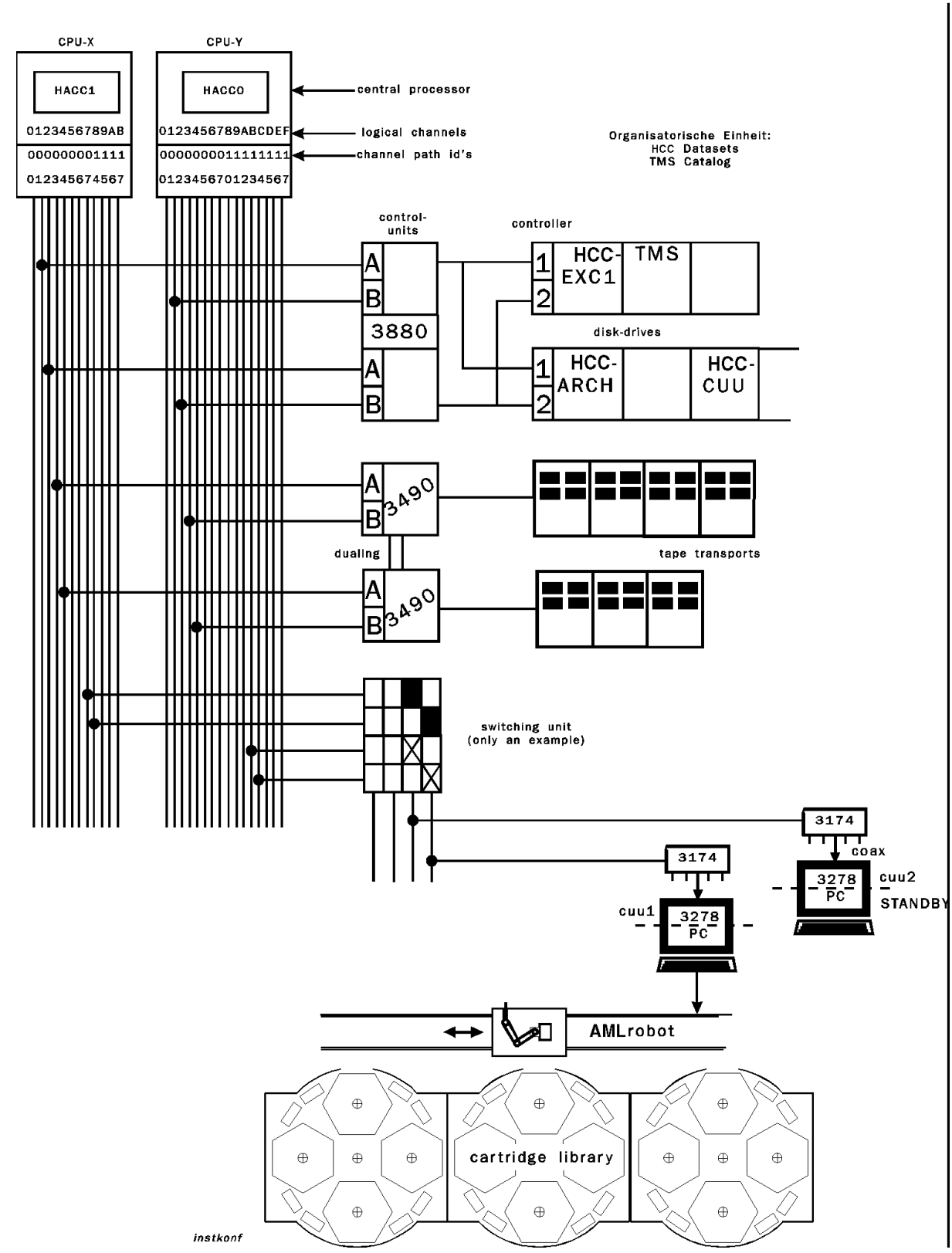


Fig. 14: Example of an installation configuration

12.9.3 MULTI-HOST COMPLEX

This configuration shows 1-9 independent computer centers which can each comprise 1-8 loosely coupled MVS systems, whereby all the MVS systems can access both common and dedicated tape stations connected to the AML system. Each of the independent HOST complexes can still retain its own TMS catalog of a tape management system for organizational reasons.

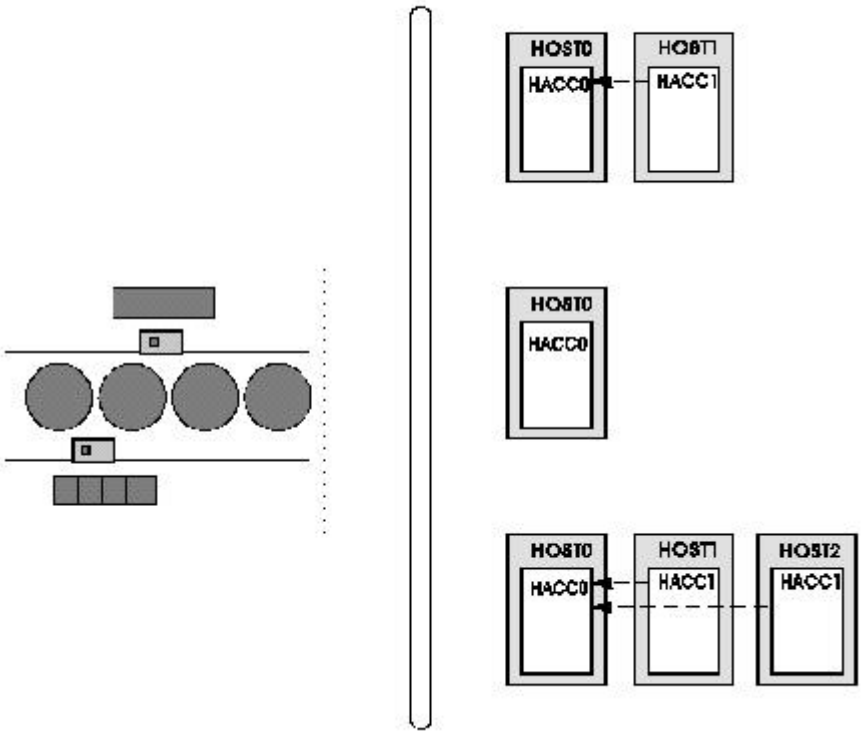


Fig. 15: Multi-host complex

Each HOST complex has its own TMS catalog.

One PRIMARY HCC system and optional SECONDARY systems must be active within a complex.

Any PRIMARY HCC system can be selected to run the central control of the AML; the MAJOR HCC system. Only the MAJOR HCC system communicates directly with the AML system, all other MINOR HCC systems indirectly through the MAJOR HCC (central control of the AML system).

MAJOR and MINOR HCC can be switched per command (refer to the **CX** commands).

12.9.4 SINGLE HOST COMPLEX WITH AMU

This configuration shows 1-36 MVS systems active on one or more CPUs with access to one or more tape pools.

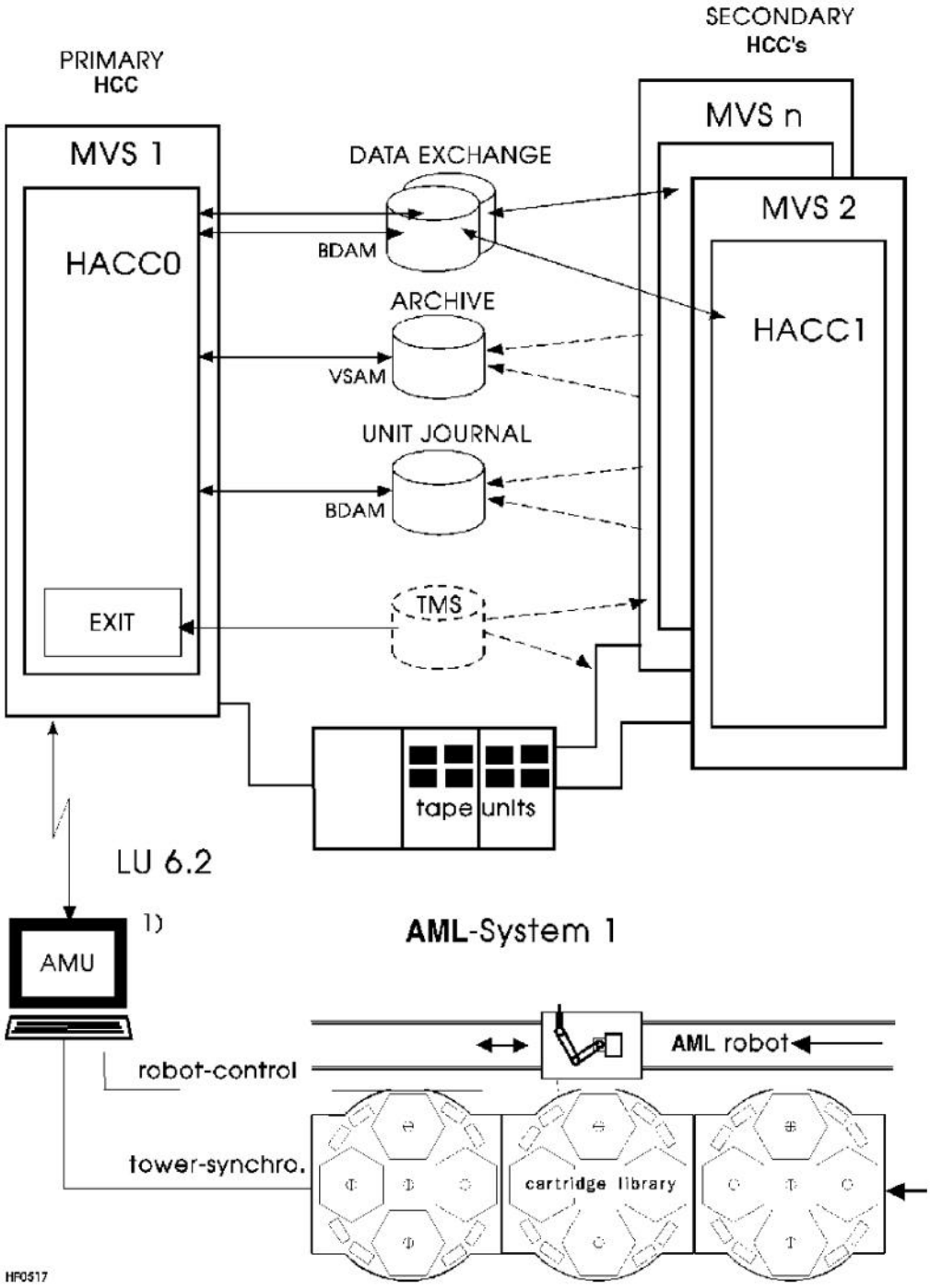


Fig. 16: Single host complex with AMU

A HCC system task is active on each of the CPUs involved. Only one system task assumes control of the AML system: the PRIMARY TASK HACC0. Any CPU can be selected for HACC0. A SECONDARY TASK HACCx (x=1-Z) is then active on each of the other CPUs.

12.9.5 MULTI-HOST COMPLEX (LU 6.2)

This configuration shows 1-n independent computer centers which can each comprise 1-16 loosely coupled MVS systems, whereby all the MVS systems can access both common and dedicated tape stations connected to the AML system. Each of the independent HOST complexes can still retain its own TMS catalog of a tape management system for organizational reasons.

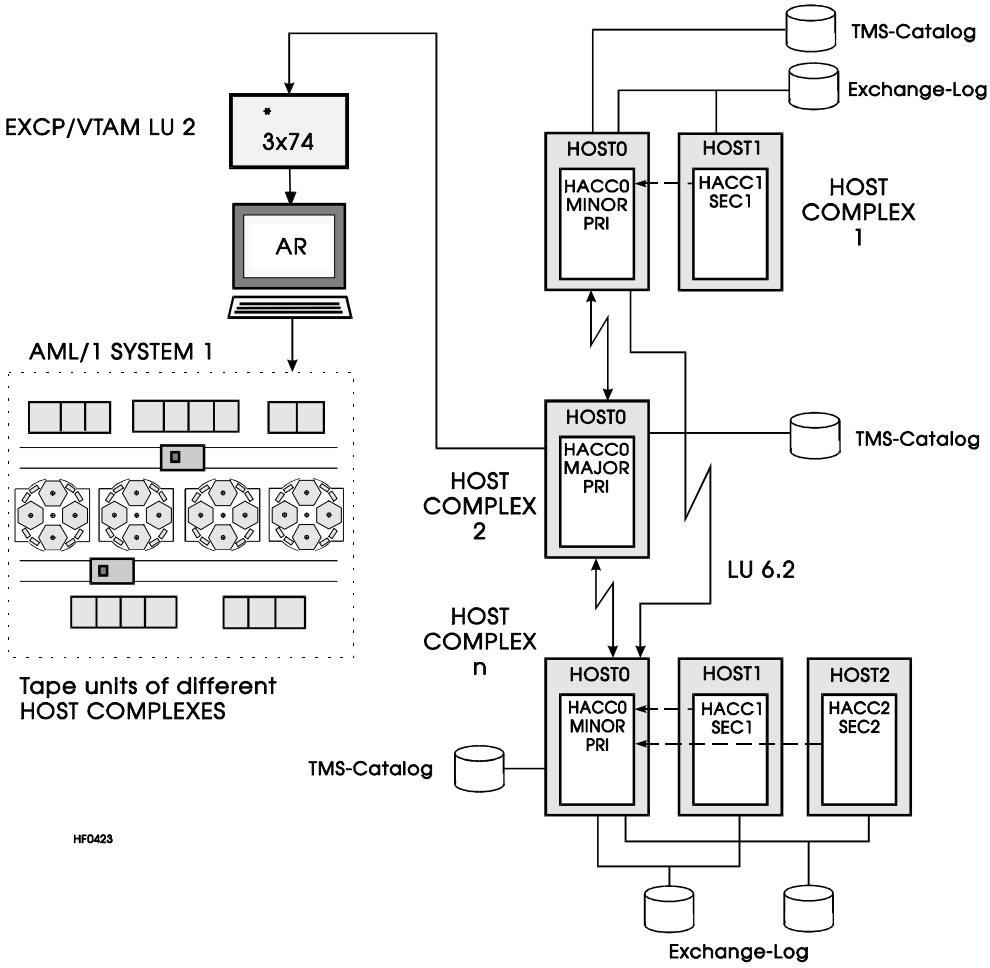


Fig. 17: Multi-host complex (LU 6.2)

One PRIMARY HCC system and optional SECONDARY systems must be active within a complex.

Any PRIMARY HCC system can be selected to run the central control of the AML; the MAJOR HCC system. Only the MAJOR HCC system communicates directly with the AML system, all other MINOR HCC systems indirectly through the MAJOR HCC (central control of the AML system).

MAJOR and MINOR HCC can be switched per command (refer to the CX commands).

The Archive can be connected to the HOST through EXCP/VTAM (Fig. 17) or LU 6.2 (AMU).

Each HOST complex has its own TMS catalog.

INDEX