

Miklos.Szigetvari@isis-papyrus.com

# MVSHSM

Extension of HSM, to manage ML2 in the network.

## Inhalt

Introduction .....	2
Installation .....	3
System Requirements .....	3
XMIT, Datasets .....	4
Usage .....	5
Exit parameters .....	5
HSM startup procedure.....	6
HSM parameters .....	7
IVP.....	8
Miscellaneous .....	9
Performance.....	9
Diagnosis .....	9
Practical Usage .....	10
ML2SYNC.....	10

---

## Introduction

The MVSHSM is an extension of the Hierarchical Storage Manager (HSM), to manage the secondary migration level data (ML2) on the network.

Here, in the ISIS Papyrus Software, we have 2 very old tape drives, no operators, we are 5 kilometers from the machine location, and we have 500 Gigabyte online storage, but practically “unlimited” storage on the network. We extended the HSM, via standard exits, to store the ML2 data on the network. We use this modification since 2005; maybe some other small installations can take advantage of this.

We define an ML2 disk volume, and via the HSM migration exits, we move everything from this volume to a Network File System (NFS) directory. Before a recall, the proper HSM exit moves the requested dataset back to our ML2 disk volume, and the HSM can finish the restore.

The exit routines are called by the HSM before the proper HSM action. It is good for the “recall”, but for the “migrate” we can just move everything from the ML2 disk drive to the NFS directory, and the last migrated dataset will reside on the ML2 disk volume.

We are using:

- MM Second Level Migration Dataset exit
- MD Dataset Migration exit
- RD Recall exit.

The exit routines were written in C++, we provide the complete source code and a load library. As sample, we modified the HSM installation STARTUP, and we provide also our current, running HSM configuration by ISIS, as it is.

## Warning

**This is a limited HSM setup, with some simple functionality; it is working here in this way for several years, but has never tested with a complex HSM configuration. We can handle securely only one ML2 recall or one ML2 migration request. It works only with non SMS volumes.**

Miklos Szigetvari  
December 2012

<miklos.szigetvari@isis-papyrus.com>

## Installation

### System Requirements

1. z/OS 1.10 or later (we are using this module for several years we didn't noticed any release dependency).
2. Disk space about 500 tracks for the complete restore.
3. IBM C++ compiler, if you want to recompile the C++ source code.
4. IBM communication server NFS client.
5. z9 machine or higher(no any hardware dependency, but the C++ code complied with ARCH(7), target architecture z9 or higher, compiler option).
6. DFSMSHSM.

## XMIT, Datasets

The XMIT partitioned dataset contains several installation datasets in XMIT format:

### 1. ADATA assembler compiler ADATA output.

As the largest part of the code is C++ we need to generate the C structures for a number of system and HSM control blocks, MCDS records etc. We are using for this, the DSECT2C utility from the C++ compiler library. The input for this generation is the assembler compiler ADATA dataset.

As the size of the ADATA is very big, we left this dataset intentionally empty.

### 2. ASM assembler library.

Beside some macros, it contains:

- `$$ASMIF` macro / module, a general interface for C++ modules to system functions.(Logstream access , VTOC access etc.).

### 3. COMPLIST library

The compiler lists from the last compilation.

### 4. CPP library

The C++ module library. As we are using here German code page, the routines are in IBM273 code page, and the compiler options also refer to this code page.

The main program is the MVSHSMX module; it represents the used exit functions. As the exits are using a number of common functions, we select the exit type according the calling programs name.

The rest is a collection of some general function; we are calling from the exit.

### 5. HPP header files

The headers file definitions for the different C++ external functions.

This HPP library contains the different generated structures from the system control blocks.

The `ARCMC*` members for the HSM exits, the `IHA*` members for CVT, ASCB, RB etc (see ADATA above).

### 7. JOBS

Library we used here to compile/link the project.

### 8. LOADLIB

- `ARCMDEXT` Dataset Migration exit, alias of `MVSHSM`.
- `ARMMEXT` Second Level Migration exit, alias of `MVSHSM`.
- `ARMRDEXT` Recall exit, alias of `MVSHSM`.
- `ML2SYNC` ML2 synchronization module, alias of `MVSHSM`.
- `MVSHSM` main module.

During the latest tests, we have noticed that `DFSMSHSM` ignores the `STEPLIB`, loads the exit modules from `LNKLST`, so you have to put the exit routines to the `LNKLST` list into a `PDSE` dataset.

### 9. OBJECT

The object library

### 10. PROC

The compile/link JCL procedures and compiler options.

### 12. SAMPLIB

Sample library with IVP jobs, HSM definitions etc.

The `#DOC` member is the short introduction, `#DOCDX` the current document in "Word" format and `'DOCPDF` in PDF format.

### 13. CNTL

A sample library for the HSM STARTUP system installation.

Here we used, as project name "ESA.PRJ.CBTHSM", you have to change this, maybe with the help of the `CHANGEAL` job from the `SAMPLIB`.

## Usage

### Exit parameters

The exit parameters are defined in the HSM startup procedure via the EXITPARM DD statement. The parameters have a “keyword = value” syntax:

- unit the dataset unit type, used in DYNALLOC (SVC 99), default is 3390.
- ml2dir the ML2 migration directory, it will contain the ML2 migrated datasets.
- delta the acceptable file size difference between the MCDS and the NFS file size, the default is 1. The program checks the file size in the MCDS (Migration Control Dataset) and the actual file size. If the difference is not lower as the “delta” value, it will be not restored.
- lrecl the logical record length of the migrated datasets, the default is 2048.
- blksize the block-size of the migrated dataset, the default is 18432.
- logger the logstream name. The MVSHSM exit routines are using a logstream to record the migration/recall messages. For the logstream allocation see the SAMPLIB LOGSTRM job, to list the logstream see the LISTLOG sample.
- devtype the migration device type we intercept. The default is x'20' (disk)
- ml2vol the VOLSER of the migration ML2 disk volume.
- ml2mask the migration dataset high level qualifier(s). Default” DFHSM.HMIG.\*\*”.
- remove a flag to indicate if the program will have to remove the NFS file at recall, or the ML2 dataset at migrate. Default is 1.
- trace exit trace requested or not. Default is 0.
- mcds the DD name for the MCDS allocation in the ML2SYNC function (see later).

This is the EXITPARM dataset, we used for the IVP process:

```
EDIT      ESA.PRJ.CBTHSM.SAMPLIB(EXITPARM) - 01.05
***** Top of Data *****
000001 ml2dir=/tmp/MVSHSM
000002 delta=1
000003 trace=1
000004 remove=1
000005 logger=DFHSM.MVSHSM.LOGSTRM
000006 ml2mask=MVSHSM.HMIG.**
000007 ml2vol=RESML2
***** Bottom of Data *****
```

## HSM startup procedure

You have to change the HSM STARTUP procedure:

- Specify in the STEPLIB the load library, contains the exit routines. In the current z/OS 1.13, seems that the exit routines will be called from the LNKLIST, but not from the STEPLIB. We still investigate here.
- EXITPARM DD to specify the exit parameters.

```
EDIT      ESA.PRJ.CBTHSM.SAMPLIB(DFSMSHSM) - 01.01
Command ==>
000039 //*****
000040 //DFSMSHSM  PROC CMD=99,      USE PARMLIB MEMBER ARCCMD00 FOR CMDS
000041 //          STR=00,          PARMLIB MEMBER FOR STARTUP PARMS
000042 //          EMERG=NO,        SETS HSM INTO NON-EMERGENCY MODE
000043 //          CDSQ=YES,        CDSs SERIALIZED WITH ENQUEUES
000044 //          PDA=YES,         PROBLEM DETERMINATION AID
000045 //          SIZE=0M,         REGION SIZE FOR DFSMSHSM
000046 //          DDD=50,          MAX DYNAMICALLY ALLOCATED DATASETS
000047 //          HOST=C,          PROC.UNIT ID AND LEVEL FUNCTIONS
000048 //          PRIMARY=YES LEVEL FUNCTIONS
000049 //*****
000050 //DFSMSHSM  EXEC PGM=ARCCTL,DYNAMBR=&DDD,REGION=&SIZE,TIME=1440,
000051 //          PARM=( 'EMERG=&EMERG', 'CMD=&CMD', 'CDSQ=&CDSQ',
000052 //          'UID=MVSHSM', 'PDA=&PDA', 'HOST=&HOST', 'STR=&STR',
000053 //          'PRIMARY=&PRIMARY')
000054 //STEPLIB   DD DSN=ESA.PRJ.CBTHSM.LOADLIB,DISP=SHR
000055 //*****
000056 //* HSM Parm DD must be deleted from the JCL or made into a      */
000057 //* a comment to use Concatenated Parmlib Support                */
000058 //*****
000059 //HSM Parm   DD DSN=ESA.PRJ.CBTHSM.SAMPLIB,DISP=SHR
000060 //MSYSOUT    DD SYSOUT=A
000061 //MSYSIN     DD DUMMY
000062 //SYSPRINT   DD SYSOUT=A,FREE=CLOSE
000063 //SYSUDUMP   DD SYSOUT=A
000064 //*
000065 //*****
000066 //* THIS PROCEDURE ASSUMES A SINGLE CLUSTER MCDS.  IF MORE THAN */
000067 //* ONE VOLUME IS DESIRED, FOLLOW THE RULES FOR A MULTICLUSTER  */
000068 //* CDS.                                                         */
000069 //*****
000070 //*
000071 //MIGCAT     DD DSN=MVSHSM.MCDS,DISP=SHR
000072 //JOURNAL    DD DSN=MVSHSM.JRNL,DISP=SHR
000073 //ARCLOGX    DD DSN=MVSHSM.HSMLOGX1,DISP=OLD
000074 //ARCLOGY    DD DSN=MVSHSM.HSMLOGY1,DISP=OLD
000075 //ARCPDOX    DD DSN=MVSHSM.HSMPODX,DISP=OLD
000076 //ARCPDOY    DD DSN=MVSHSM.HSMPODY,DISP=OLD
000077 //*
000078 //*****
000079 //* REMOVE THE NEXT DD STATEMENT IF YOU DO NOT INTEND TO USE   */
000080 //* BACKUP AND DUMP.                                           */
000081 //*
000082 //* THIS PROCEDURE ASSUMES A SINGLE CLUSTER BCDS.  IF MORE THAN */
000083 //* ONE VOLUME IS DESIRED, FOLLOW THE RULES FOR A MULTICLUSTER  */
000084 //* CDS.                                                         */
000085 //*****
000086 //*
000087 //BAKCAT     DD DSN=MVSHSM.BCDS,DISP=SHR
000088 //*
000089 //* HSM EXITS
000090 //EXITPARM   DD DISP=SHR,DSN=ESA.PRJ.CBTHSM.SAMPLIB(EXITPARM)
000091 //* HSM EXITS
000092 //*
***** ***** Bottom of Data *****
```

## HSM parameters

The following HSM (ARCCMDxx) parameters changed, added:

- Specify the exit routines:  
SETSYS EXITON(MM)  
SETSYS EXITON(MD)  
SETSYS EXITON(RD)
- Define the ML2 volume:  
DEFINE MIGRATIONLEVEL2 ( VOLUMES(xxxxxx)) xxxxxx is the ML2 disk volume
- Add the xxxxxx volume to the ML2 volume list via:  
ADDVOL xxxxxx               /\* ADD A VOLUME (PROVIDE SERIAL) \*/ -  
    UNIT(3390)               /\* WITH UNIT TYPE (PROVIDE TYPE) \*/ -  
    MIGRATION               /\* AS A MIGRATION LEVEL 1 VOLUME \*/ -  
    (MIGRATIONLEVEL2       /\* WITH NO SMALL DATA SET       \*/ -  
      NOSMALLDATASETPACKING) /\* PACKING AVAILABLE.       \*/ -  
    THRESHOLD(100)       /\* NO THRESHOLD PROCESSING.       \*/
- We can handle securely only one ML2 recall, so specify:  
SETSYS TAPEMAXRECALLTASKS(1)
- Specify, that we would like to get a DUMP, if an exit abend occurs:  
PATCH .MCVT.+2D BITS(.....1) /\* DUMP AFTER INSTALLATION EXIT \*/

The SAMPLIB's SAMPARCC member is the current, running HSM parameter set.

The ARCCMD99 member we used for the IVP procedure.



## IVP

For the Installation Verification Procedure, we defined a “mini” HFS system in a separate test LPAR, to verify the installation:

- The migration “ml2dir” is the /tmp/MVSHSM temporary directory, so you have to define this directory with the proper access rights, and after the IVP you have to change this.
- We defined one 3390-3 primary volume called RESML0.
- We defined one 3390-3 first level migration volume RESML1.
- The ML2 volume is RESML2.
- The LOGSTRM job defines a logstream.
- The DFSMSHSM job starts the HSM.

After the startup we tested the HSM functionality with some test datasets on the RESML0 volume. With the TSO HSEND command:”HSEND MIGRATE DATASET (dsn1) ML2” we migrate dsn1 to ML2, but this dataset will remain in the RESML2 volume, as the migration exit gets control before the migration. You can recall this dataset. If you migrate dsn1 and dsn2 to ML2, you will find the migrated dsn1 in the /tmp/MVSHSM directory.

## Miscellaneous

### Performance

Could be better, but we are currently satisfied with the recall performance. The exits are using very small amount of CPU, the most critical is the NFS performance. The TAPMAXRECALLTASKS value is 1 currently. Most of the ML2 migration happens at evenings, during the secondary space management process.

We have tested the performance with a library set as bellow. The restore elapsed time was about 6 min 20 seconds, the ML2 migration time was about 3 min 45 seconds here in a z9 S07 machine.

DSLIST - Data Sets Matching DPR.V080801.**					Row 1 of 22	
Command ==>					Scroll ==> CSR	
Total Tracks:	13176 non-x:	13176 Data Sets:	22 non-x:	22		
Command - Enter "/" to select action					Tracks	%Used
					XT	
DPR.V080801.BR080801.PDEMF620.MESSAGE					2	54
DPR.V080801.BR080801.PDEMI620.EXP					105	90
DPR.V080801.BR080801.PDEMI620.LOADLIB					3135	99
DPR.V080801.BR080801.PDEMU620.DBRMLIB					1	58
DPR.V080801.BR080801.PDEMU620.EXP					120	93
DPR.V080801.BR080801.PDEMU620.LOADLIB					3744	99
DPR.V080801.BR080801.PSEMF620.ARC.MESSAGE					2	79
DPR.V080801.BR080801.PSEMF620.MET.MESSAGE					1	75
DPR.V080801.BR080801.PSEMF620.PCL.MESSAGE					1	75
DPR.V080801.BR080801.PSEMF620.PDF.MESSAGE					1	75
DPR.V080801.BR080801.PSEMF620.PE.MESSAGE					1	83
DPR.V080801.BR080801.PSEMU620.ARC.DBRMPASB					1	100
DPR.V080801.BR080801.PSEMU620.ARC.EXP					75	91
DPR.V080801.BR080801.PSEMU620.ARC.LOADLIB					1627	99
DPR.V080801.BR080801.PSEMU620.MET.EXP					60	79
DPR.V080801.BR080801.PSEMU620.MET.LOADLIB					1138	99
DPR.V080801.BR080801.PSEMU620.PCL.EXP					60	80
DPR.V080801.BR080801.PSEMU620.PCL.LOADLIB					1153	99
DPR.V080801.BR080801.PSEMU620.PDF.EXP					60	80
DPR.V080801.BR080801.PSEMU620.PDF.LOADLIB					1182	99
DPR.V080801.BR080801.PSEMU620.PE.EXP					30	88
DPR.V080801.BR080801.PSEMU620.PE.LOADLIB					677	99
***** End of Data Set List *****						

Some options for the future development:

- Use the CEEPIPI and the initialization termination exits, to avoid the LE initialization at every call.
- Use the FTP API to access the ML2 disk.
- Put the ML2 datasets parallel to several locations.

### Diagnosis

We set via PATH command the proper MCVT bit to get DUMP if an exit terminates abnormally.

```
PATCH .MCVT.+2D BITS(.....1) /* DUMP AFTER INSTALLATION EXIT */
/* ABENDS */
```

The EXITPARM “trace” option can give some details:

```
<Program> program:??????? programRb:ARCMDSH ARCMDSH CompileDate:Dec 11 2012 Version:1.1.0
<Logger>DFHSM.MVSHSM.LOGSTRM
<makespace> volume RESML2
<makespace> ENQ RESML2
<After dynalloc> rc:0 0 0, dsn:SYS1.VTOCIX.RESML2 ddn:SYS00033
<makespace> selected MVSHSM.HMIG.T533109.ESA.PRJ.A2347
<After dynalloc> rc:0 0 0, dsn:MYSHSM.HMIG.T533109.ESA.PRJ.A2347 ddn:SYS00034
<makespace> input allocated to //DD:SYS00034 and opened as SYS00034
<makespace> flag:81 mode:1ff
<makespace> file:
2012:12:12 09:32:16 ARC6025I copied MVSHSM.HMIG.T533109.ESA.PRJ.A2347 to /tmp/MVSHSM/MVSHSM.HMIG.T533109.ESA.PRJ.A2347 , 29343
ocks
<makespace> remove //DD:SYS00034
2012:12:12 09:32:16 ARC6027I removed MVSHSM.HMIG.T533109.ESA.PRJ.A2347
<makespace> CloseVtoc SYS1.VTOCIX.RESML2
<makespace> DEQ RESML2
<ARCMxEXT> ESA.PRJ.CBTHSM.T2MB 1
2012:12:12 09:32:16 ARC6029I migrating ESA.PRJ.CBTHSM.T2MB from RESML0 volume
***** BOTTOM OF DATA *****
```

## Practical Usage

We are a software development company, so maybe our disk usage is not typical. We produce during the weekly delivery a new version of our complete z/OS software, it is 17 000 track. This version is tested with a number of test jobs, and if we have a customer problem (we have some), we take an older version.

In our ML2 we have about 48 000 dataset, the primary (PVOL) space is 150 Gigabyte, the ML1 space is 55 Gigabyte. In an NFS network drive called "stoneage" we have an ML2 directory, it is currently about 650 Gigabyte. The system administrators are making backups from the ML2 directory. Checked the HSM ML2 activity for about two months, we have in this period , about 2200 ML2 migrations during the space management, and about 800 restore requests from ML2.

We are using the NFS client in z/OS and the "stoneage" was mounted via the following automount member:

```
BROWSE /HVID/etc/stoneagebin.map Line 00000000 Col 001 127
Command ==> Scroll ==> CSR
***** Top of Data *****
name
type NFS
filesystem stoneagebin.<asis_name>
parm 10.1.4.63:/data/<asis_name>,hard,vers(3),DataCaching(y),ReadAhead(16),usize(8192),acregmin(0),acregmax(1),vers(
mode rdur
duration nolimit
delay 10
***** Bottom of Data *****
```

## ML2SYNC

We didn't find a proper way to get control during the "HDELETE" operation, so if you delete in this way from the MCDS, the migrated dataset remains in the NFS directory. To synchronize time to time the MCDS with the ML2 directory, the MVSHSM program has the ML2SYNC entry, it reads the MCDS and the ML2 directory, and deletes everything from the directory which has no MCDS entry:

```
EDIT ESA.PRJ.CBTHSM.SAMPLIB(ML2SYNC) - 01.02
Command ==>
***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
==MSG> your edit profile using the command RECOVERY ON.
000001 //ML2SYNC JOB ,&SYSUID,NOTIFY=&SYSUID,
000002 // CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
000003 // REGION=0M
000004 //ML2SYNC EXEC PGM=ML2SYNC,REGION=0M
000005 //STEPLIB DD DISP=SHR,DSN=ESA.PRJ.CBTHSM.LOADLIB
000006 //HSM Parm DD DSN=SYSPLEX.PARM.PARMLIB,DISP=SHR
000007 //MSYSOUT DD SYSOUT=A
000008 //MSYSIN DD DUMMY
000009 //SYSPRINT DD SYSOUT=A,FREE=CLOSE
000010 //MIGCAT DD DSN=MVSHSM.MCDS,DISP=SHR,
000011 // AMP='BUFHI=128,BUFHD=128'
000012 //* HSM EXITS
000013 //EXIT Parm DD *
000014 delta=1
000015 logger=DFHSM.MVSHSM.LOGSTRM
000016 ml2mask=MVSHSM.HMIG.**
000017 ml2vol=RESML2
000018 remove=1
000019 ml2dir=/tmp/MVSHSM
000020 trace=1
000021 //* HSM EXITS
***** Bottom of Data *****
```