Miklos.Szigetvari@isis-papyrus.com

MVSDSSN

An ADRDSSU extension, to manage the dumps in the network.

Miklos Szigetvari
28.11.2012

Table of Contents

Introduction

The MVSDSSN is an extension of the standard ADRDSSU utility, to manage the dumps in the network. We have here (ISIS Information System) two very old tape units, but "nearly" unlimited storage capacity on the local network. We extended the ADRDSSU utility via UIM (User Interaction Module) exits to write dumps to HFS (Hierarchical File System) files, and store the dump information in a DB2 database. This modification is running here for several years, maybe smaller shops can use this. A large part of the modification was written in C/C++, we provide the complete source code, and the necessary definitions for DB2. The code uses the open source "zlib" general purpose compress library, Copyright (C) 1995-2002 Jean-loup Gailly and Mark Adler.

Miklos Szigetvari November 2012

<miklos.szigetvari@isis-papyrus.com>

Quick Start

1. For a quick start you need only the LOADLIB, SAMPLIB and DBRMLIB. So from the received XMIT dataset EDIT and SUBMIT the QUICKSTR job to restore these three datasets.
2. The LOADLIB should be APF authorized so authorize it or copy the MVSDSSN load module into an APF authorized library.
3. The SAMPLIB IVP10DUM job creates a DUMP of the SAMPLIB dataset to the "/tmp" temporary root directory without DB2 support. The created HFS file name will contain the JOBNAME, JOBNUMBER and a time stamp.
4. EDIT the IVP10RES job to insert the created HFS file name after the DIRECT DD statement and SUBMIT this job.

Installation

System Requirements

1. z/OS 1.8 or later (we are using this module for several years we didn't noticed any release dependency).
2. Disk space about 1000 tracks for the complete restore.
3. IBM C/C++ compiler, if you want to recompile the C/C++ source code.
4. IBM DB2 C/C++ preprocessor if you would like to recompile the DB2 access module.
5. DB2 subsystem (we are using this since DB2 5.1) the current was compiled with DB2 9.1.
6. IBM communication server NFS client, if you want to send the dump to the NFS network.
7. z9 machine or higher(no any hardware dependency, but the C/C++ code complied with ARCH(7), target architecture z9 or higher, compiler option).

Of course the ADRDSSU dump/restore utility.

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/C++ we need to generate the C structures for a number of system control blocks (CVT PSA TCB UIM control blocks etc.). We are using for this, the DSECT2C utility from the C/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/C++ modules to system functions.(Logstream access , name/token services etc.)
- SSQLASM C/C++ to DB2 routine, to dynamically load the DSNALI DSNHLI2 DB2 interface routines.
- UIMASM the UIM exit routine, its function is only to call the proper UIM C code.

3. COMPLIST library
The compiler lists from the last compilation.

4. CPP library
The C/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.
A part of the code is the "zlib" general compress library, Copyright (C) 1995-2002 Jean-loup Gailly and Mark Adler.
The dump/restore extension modules:

- DSSMAIN the main program to process the parameters and call the ADRDSSU. It also contains the different UIM exit entry point code sections.
- DSSCLAS for the dump/restore objects.
- DSSDB the SQL interface module.
- DSSFILE HFS file access and calls to the “zlib” library inflate deflate routines.

The third part is a collection of general C/C++ routines for message processing, file allocation, date/time conversions etc.

5. DBRMLIB
The DB2 DBRM module.

6. HPP header files
The headers file definitions for the different C/C++ external functions.
This HPP library contains the different generated structures from the system control blocks.
EIREC* members for the UIM exits or the IHA* members for CVT, ASCB, RB etc (see ADATA above).

7. JOBS
Library we used here to compile/link the project.

8. LOADLIB
The MVSDSSN module, it was linked with AC(1), it should be in an APF authorized library.

9. OBJECT
The object library

10. PROC
The compile/link JCL procedures and compiler options.

11. PPONLY
DB2 C/C++ preprocessor output library.

12. SAMPLIB
Sample library with IVP jobs, DB2 definitions etc.

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

Compile or Not

The current version was compiled with the IBM C/C++ compiler with the following compiler options:

```
SEARCH(CEE.SCEEH.+,CBC.SCLBH.+,DD:USERLIB)
GONUMBER
OPT(2)
TUNE(7)
ARCH(7)
SPILL(3600)
DLL(CBA)
LOC(DE_DE.IBM-273)
DEFINE(__OS390__)
DEFINE(_POSIX_SOURCE)
DEFINE(_XOPEN_SOURCE_EXTENDED)
DEFINE(MU_STANDALONE)
DEFINE(ISIS_PF_MU)
LANGLVL(EXTENDED)
```

The JOBS dataset contains the compile/link job.
The output is a PDSE dataset

DB2 or Not

If the DB2 support active, you can store the dump information in a DB2 database, and during restore you can select the proper input via an SQL query.

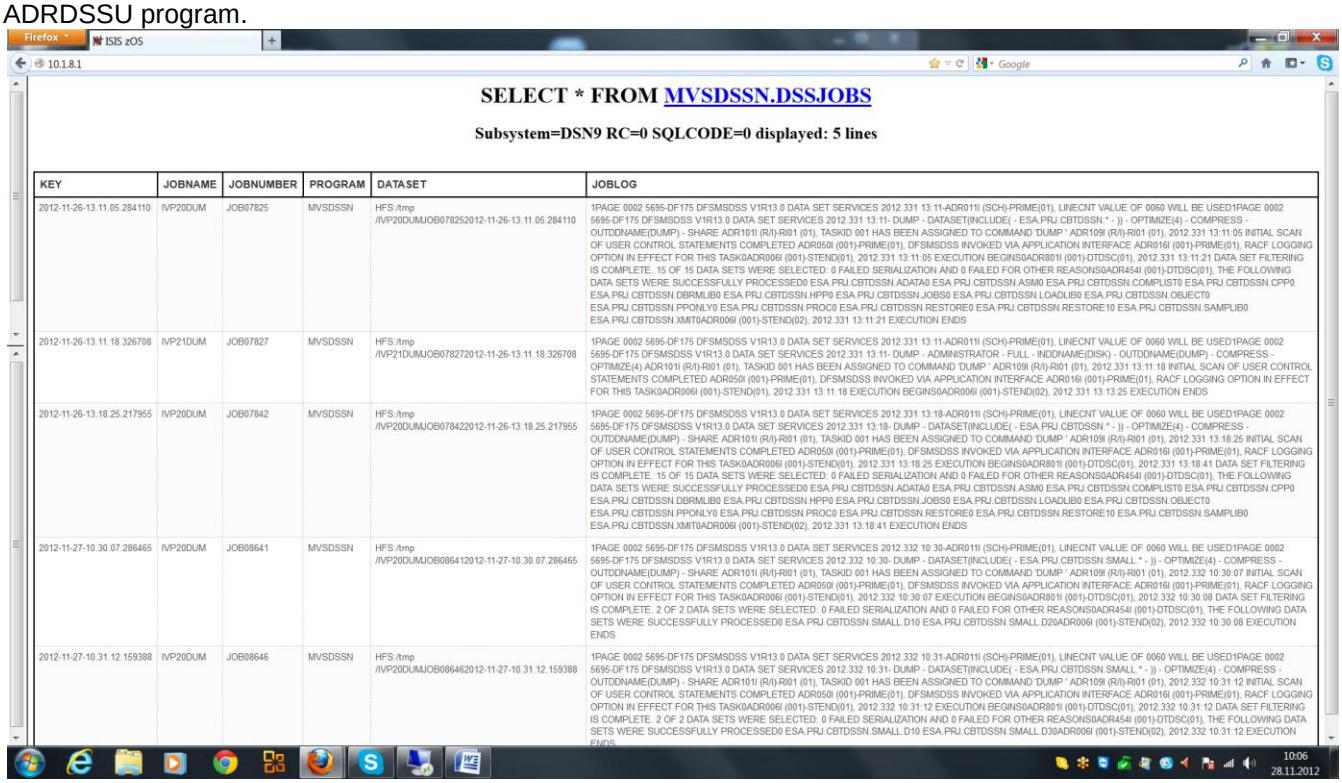
If the DB2 support not active (set the environment variable USEDDB2=NO), you have to specify in the DIRECT DD input, the file name for restore.

The DB2 database has the following structure:

```
CREATE TABLE MVSDSSN.DSSJOBS
(
    KEY             TIMESTAMP NOT NULL,
    JOBNAME          CHAR(8),
    JOBNUMBER        CHAR(8),
    PROGRAM          CHAR(8),
    DATASET         CHAR(128),
    JOBLOG          VARCHAR(32000),
    PRIMARY KEY(KEY)
) IN MVSDSSN.DSSNTBLS;
CREATE TABLE MVSDSSN.DSSDATASETS
(
    KEY             TIMESTAMP NOT NULL,
    JOB             TIMESTAMP NOT NULL,
    DATASET          CHAR(44),
    VOLUMES          CHAR(36),
    TYPE            INTEGER,
    SIZE            INTEGER,
    PRIMARY KEY(KEY),
    FOREIGN KEY   FJOB (JOB)
    REFERENCES    MVSDSSN.DSSJOBS
) IN MVSDSSN.DSSNTBLS;
```

The DSSJOBS table contains a row for every dump request.

It contains the JOBNAME JOBID the output HFS file name and the first 32K of the DUMP messages from the ADRDSSU program.

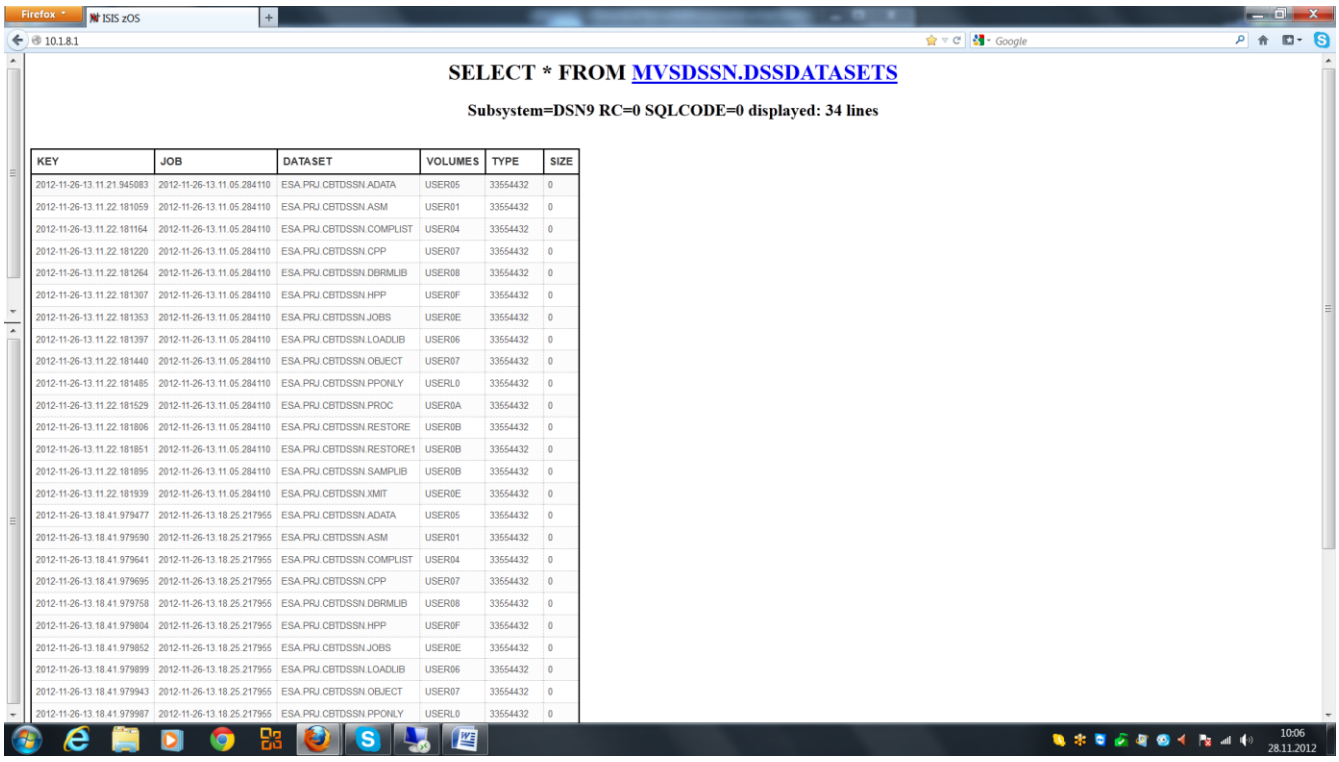


SELECT * FROM MVSDSSN.DSSJOBS

Subsystem=DSN9 RC=0 SQLCODE=0 displayed: 5 lines

KEY	JOBNAME	JOBNUMBER	PROGRAM	DATASET	JOBLOG
2012-11-26-13.11.05.284110	IVP20DUM	JOB07825	MVSDSSN	HFS:/tmp /IVP20DUMJOB078252012-11-26-13.11.05.284110	1PAGE 0002 5695-DF175 DFSMSDSS V1R13.0 DATA SET SERVICES 2012.331 13:11-ADR011I (SCH)-PRIME(01), LINECNT VALUE OF 0060 WILL BE USED1PAGE 0002 5695-DF175 DFSMSDSS V1R13.0 DATA SET SERVICES 2012.331 13:11- DUMP - DATASET(INCLUDE(- ESA.PRJ.CBTDSSN.* -)) - OPTIMIZE(4) - COMPRESS - OUTDDNAME(DUMP) - SHARE ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'DUMP ' ADR109I (R/I)-RI01 (01), 2012.331 13:11:05 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED ADR050I (001)-PRIME(01), DFSMSDSS INVOKED VIA APPLICATION INTERFACE ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK0ADR006I (001)-STEND(01), 2012.331 13:11:05 EXECUTION BEGINS0ADR801I (001)-DTDSC(01), 2012.331 13:11:21 DATA SET FILTERING IS COMPLETE. 15 OF 15 DATA SETS WERE SELECTED: 0 FAILED SERIALIZATION AND 0 FAILED FOR OTHER REASONS0ADR454I (001)-DTDSC(01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED0 ESA.PRJ.CBTDSSN.ADATA0 ESA.PRJ.CBTDSSN.ASM0 ESA.PRJ.CBTDSSN.COMPLIST0 ESA.PRJ.CBTDSSN.CPP0 ESA.PRJ.CBTDSSN.DBRMLIB0 ESA.PRJ.CBTDSSN.HPP0 ESA.PRJ.CBTDSSN.JOBS0 ESA.PRJ.CBTDSSN.LOADLIB0 ESA.PRJ.CBTDSSN.OBJECT0 ESA.PRJ.CBTDSSN.PPONLY0 ESA.PRJ.CBTDSSN.PROC0 ESA.PRJ.CBTDSSN.RESTORE0 ESA.PRJ.CBTDSSN.RESTORE10 ESA.PRJ.CBTDSSN.SAMPLIB0 ESA.PRJ.CBTDSSN.XMIT0ADR006I (001)-STEND(02), 2012.331 13:11:21 EXECUTION ENDS
2012-11-26-13.11.18.326708	IVP21DUM	JOB07827	MVSDSSN	HFS:/tmp /IVP21DUMJOB078272012-11-26-13.11.18.326708	1PAGE 0002 5695-DF175 DFSMSDSS V1R13.0 DATA SET SERVICES 2012.331 13:11-ADR011I (SCH)-PRIME(01), LINECNT VALUE OF 0060 WILL BE USED1PAGE 0002 5695-DF175 DFSMSDSS V1R13.0 DATA SET SERVICES 2012.331 13:11- DUMP - ADMINISTRATOR - FULL - INDDNAME(DISK) - OUTDDNAME(DUMP) - COMPRESS - OPTIMIZE(4) ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'DUMP ' ADR109I (R/I)-RI01 (01), 2012.331 13:11:18 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED ADR050I (001)-PRIME(01), DFSMSDSS INVOKED VIA APPLICATION INTERFACE ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK0ADR006I (001)-STEND(01), 2012.331 13:11:18 EXECUTION BEGINS0ADR006I (001)-STEND(02), 2012.331 13:13:25 EXECUTION ENDS
2012-11-26-13.18.25.217955	IVP20DUM	JOB07842	MVSDSSN	HFS:/tmp /IVP20DUMJOB078422012-11-26-13.18.25.217955	1PAGE 0002 5695-DF175 DFSMSDSS V1R13.0 DATA SET SERVICES 2012.331 13:18-ADR011I (SCH)-PRIME(01), LINECNT VALUE OF 0060 WILL BE USED1PAGE 0002 5695-DF175 DFSMSDSS V1R13.0 DATA SET SERVICES 2012.331 13:18- DUMP - DATASET(INCLUDE(- ESA.PRJ.CBTDSSN.* -)) - OPTIMIZE(4) - COMPRESS - OUTDDNAME(DUMP) - SHARE ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'DUMP ' ADR109I (R/I)-RI01 (01), 2012.331 13:18:25 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED ADR050I (001)-PRIME(01), DFSMSDSS INVOKED VIA APPLICATION INTERFACE ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK0ADR006I (001)-STEND(01), 2012.331 13:18:25 EXECUTION BEGINS0ADR801I (001)-DTDSC(01), 2012.331 13:18:41 DATA SET FILTERING IS COMPLETE. 15 OF 15 DATA SETS WERE SELECTED: 0 FAILED SERIALIZATION AND 0 FAILED FOR OTHER REASONS0ADR454I (001)-DTDSC(01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED0 ESA.PRJ.CBTDSSN.ADATA0 ESA.PRJ.CBTDSSN.ASM0 ESA.PRJ.CBTDSSN.COMPLIST0 ESA.PRJ.CBTDSSN.CPP0 ESA.PRJ.CBTDSSN.DBRMLIB0 ESA.PRJ.CBTDSSN.HPP0 ESA.PRJ.CBTDSSN.JOBS0 ESA.PRJ.CBTDSSN.LOADLIB0 ESA.PRJ.CBTDSSN.OBJECT0 ESA.PRJ.CBTDSSN.PPONLY0 ESA.PRJ.CBTDSSN.PROC0 ESA.PRJ.CBTDSSN.RESTORE0 ESA.PRJ.CBTDSSN.RESTORE10 ESA.PRJ.CBTDSSN.SAMPLIB0 ESA.PRJ.CBTDSSN.XMIT0ADR006I (001)-STEND(02), 2012.331 13:18:41 EXECUTION ENDS
2012-11-27-10.30.07.286465	IVP20DUM	JOB08641	MVSDSSN	HFS:/tmp /IVP20DUMJOB086412012-11-27-10.30.07.286465	1PAGE 0002 5695-DF175 DFSMSDSS V1R13.0 DATA SET SERVICES 2012.332 10:30-ADR011I (SCH)-PRIME(01), LINECNT VALUE OF 0060 WILL BE USED1PAGE 0002 5695-DF175 DFSMSDSS V1R13.0 DATA SET SERVICES 2012.332 10:30- DUMP - DATASET(INCLUDE(- ESA.PRJ.CBTDSSN.SMALL.* -)) - OPTIMIZE(4) - COMPRESS - OUTDDNAME(DUMP) - SHARE ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'DUMP ' ADR109I (R/I)-RI01 (01), 2012.332 10:30:07 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED ADR050I (001)-PRIME(01), DFSMSDSS INVOKED VIA APPLICATION INTERFACE ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK0ADR006I (001)-STEND(01), 2012.332 10:30:07 EXECUTION BEGINS0ADR801I (001)-DTDSC(01), 2012.332 10:30:08 DATA SET FILTERING IS COMPLETE. 2 OF 2 DATA SETS WERE SELECTED: 0 FAILED SERIALIZATION AND 0 FAILED FOR OTHER REASONS0ADR454I (001)-DTDSC(01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED0 ESA.PRJ.CBTDSSN.SMALL.D10 ESA.PRJ.CBTDSSN.SMALL.D20ADR006I (001)-STEND(02), 2012.332 10:30:08 EXECUTION ENDS
2012-11-27-10.31.12.159388	IVP20DUM	JOB08646	MVSDSSN	HFS:/tmp /IVP20DUMJOB086462012-11-27-10.31.12.159388	1PAGE 0002 5695-DF175 DFSMSDSS V1R13.0 DATA SET SERVICES 2012.332 10:31-ADR011I (SCH)-PRIME(01), LINECNT VALUE OF 0060 WILL BE USED1PAGE 0002 5695-DF175 DFSMSDSS V1R13.0 DATA SET SERVICES 2012.332 10:31- DUMP - DATASET(INCLUDE(- ESA.PRJ.CBTDSSN.SMALL.* -)) - OPTIMIZE(4) - COMPRESS - OUTDDNAME(DUMP) - SHARE ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'DUMP ' ADR109I (R/I)-RI01 (01), 2012.332 10:31:12 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED ADR050I (001)-PRIME(01), DFSMSDSS INVOKED VIA APPLICATION INTERFACE ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK0ADR006I (001)-STEND(01), 2012.332 10:31:12 EXECUTION BEGINS0ADR801I (001)-DTDSC(01), 2012.332 10:31:12 DATA SET FILTERING IS COMPLETE. 2 OF 2 DATA SETS WERE SELECTED: 0 FAILED SERIALIZATION AND 0 FAILED FOR OTHER REASONS0ADR454I (001)-DTDSC(01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED0 ESA.PRJ.CBTDSSN.SMALL.D10 ESA.PRJ.CBTDSSN.SMALL.D30ADR006I (001)-STEND(02), 2012.332 10:31:12 EXECUTION ENDS

The DSSDATASETS table contains a row for every DUMP request.
It contains the DATASET name, VOLUMES list and the KEY in the DSSJOBS table.



SELECT * FROM MVSDSSN.DSSDATASETS

Subsystem=DSN9 RC=0 SQLCODE=0 displayed: 34 lines

KEY	JOB	DATASET	VOLUMES	TYPE	SIZE
2012-11-26-13.11.21.945083	2012-11-26-13.11.05.284110	ESA.PRJ.CBTDSSN.ADATA	USER05	33554432	0
2012-11-26-13.11.22.181059	2012-11-26-13.11.05.284110	ESA.PRJ.CBTDSSN.ASM	USER01	33554432	0
2012-11-26-13.11.22.181164	2012-11-26-13.11.05.284110	ESA.PRJ.CBTDSSN.COMPLIST	USER04	33554432	0
2012-11-26-13.11.22.181220	2012-11-26-13.11.05.284110	ESA.PRJ.CBTDSSN.CPP	USER07	33554432	0
2012-11-26-13.11.22.181264	2012-11-26-13.11.05.284110	ESA.PRJ.CBTDSSN.DBRMLIB	USER08	33554432	0
2012-11-26-13.11.22.181307	2012-11-26-13.11.05.284110	ESA.PRJ.CBTDSSN.HPP	USER0F	33554432	0
2012-11-26-13.11.22.181353	2012-11-26-13.11.05.284110	ESA.PRJ.CBTDSSN.JOBS	USER0E	33554432	0
2012-11-26-13.11.22.181397	2012-11-26-13.11.05.284110	ESA.PRJ.CBTDSSN.LOADLIB	USER06	33554432	0
2012-11-26-13.11.22.181440	2012-11-26-13.11.05.284110	ESA.PRJ.CBTDSSN.OBJECT	USER07	33554432	0
2012-11-26-13.11.22.181485	2012-11-26-13.11.05.284110	ESA.PRJ.CBTDSSN.PPONLY	USERL0	33554432	0
2012-11-26-13.11.22.181529	2012-11-26-13.11.05.284110	ESA.PRJ.CBTDSSN.PROC	USER0A	33554432	0
2012-11-26-13.11.22.181806	2012-11-26-13.11.05.284110	ESA.PRJ.CBTDSSN.RESTORE	USER0B	33554432	0
2012-11-26-13.11.22.181851	2012-11-26-13.11.05.284110	ESA.PRJ.CBTDSSN.RESTORE1	USER0B	33554432	0
2012-11-26-13.11.22.181895	2012-11-26-13.11.05.284110	ESA.PRJ.CBTDSSN.SAMPLIB	USER0B	33554432	0
2012-11-26-13.11.22.181939	2012-11-26-13.11.05.284110	ESA.PRJ.CBTDSSN.XMIT	USER0E	33554432	0
2012-11-26-13.18.41.979477	2012-11-26-13.18.25.217955	ESA.PRJ.CBTDSSN.ADATA	USER05	33554432	0
2012-11-26-13.18.41.979590	2012-11-26-13.18.25.217955	ESA.PRJ.CBTDSSN.ASM	USER01	33554432	0
2012-11-26-13.18.41.979641	2012-11-26-13.18.25.217955	ESA.PRJ.CBTDSSN.COMPLIST	USER04	33554432	0
2012-11-26-13.18.41.979695	2012-11-26-13.18.25.217955	ESA.PRJ.CBTDSSN.CPP	USER07	33554432	0
2012-11-26-13.18.41.979758	2012-11-26-13.18.25.217955	ESA.PRJ.CBTDSSN.DBRMLIB	USER08	33554432	0
2012-11-26-13.18.41.979804	2012-11-26-13.18.25.217955	ESA.PRJ.CBTDSSN.HPP	USER0F	33554432	0
2012-11-26-13.18.41.979852	2012-11-26-13.18.25.217955	ESA.PRJ.CBTDSSN.JOBS	USER0E	33554432	0
2012-11-26-13.18.41.979899	2012-11-26-13.18.25.217955	ESA.PRJ.CBTDSSN.LOADLIB	USER06	33554432	0
2012-11-26-13.18.41.979943	2012-11-26-13.18.25.217955	ESA.PRJ.CBTDSSN.OBJECT	USER07	33554432	0
2012-11-26-13.18.41.979987	2012-11-26-13.18.25.217955	ESA.PRJ.CBTDSSN.PPONLY	USERL0	33554432	0

The default is to use the DB2 interface. You have to BIND the DBRMLIB module according the BIND sample, and create the database according the CREATEDB sample job.

Usage

JCL and SYSIN

Here is a sample JOB to create a dump

```
//MVSDSSN     EXEC PGM=MVSDSSN,REGION=0M,
// PARM=('LINECNT=60')
//STEPLIB   DD DISP=SHR,DSN=ESA.PRJ.CBTDSSN.LOADLIB
//DUMP        DD DUMMY
//RESTORE   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTERM   DD SYSOUT=*
//ENVFILE   DD DSN=ESA.PRJ.CBTDSSN.SAMPLIB(IVP2ENV),DISP=SHR
//DISK        DD UNIT=3390,VOL=SER=Z1DT11,DISP=OLD
//CEEOPTS   DD *
POSIX(ON)
ENVAR("_CEE_ENVFILE_S=DD:ENVFILE")
TRAP(OFF,NOSPIE)
HEAP(8M,1M,ANY,FREE)
//SYSIN       DD *
  DUMP -
  ADMINISTRATOR -
  FULL -
  INDDNAME(DISK) -
  OUTDDNAME(DUMP) -
  COMPRESS   -
  OPTIMIZE(4)
```

- In the EXEC statement we call the MVSDSSN program via STEPLIB and it will call the ADRDSSU utility
- The PARM's are the normal ADRDSSU parameters, we use often the TYPRUN(NORUN) option to check the dataset selection etc.
- We use here 0M region size
- The APF authorized STEPLIB contains the MVSDSSN program
- The //DUMP DD DUMMY DD statement necessary to prevent ADRDSSU to complain about the missing output
- The main program is a Language Environment C program, under the CEEOPTS DD statement we specify the necessary LE runtime options. Here is a definition of the ENVFILE, the dataset contains the environment variables for the dump/restore (see below)
- Under the SYSIN DD, the normal parameters for the ADRDSSU utility. Here is one exception, during the restore process with DB2 support, the MVSDSSN module should know the dataset mask for the SQL query. We mark this via "/*include*/ comment marker. If the programs see this comment line in the input, it take the dataset mask from this line to generate the SQL query:

```
//SYSIN       DD *
 RESTORE                                  -
      DATASET (INCLUDE(                   -
           ESA.PRJ.CBTDSSN.* -    /*include*/
                                 ))       -
      INDDNAME(DUMP)                      -
      RENAME(**.SAMPLIB,**.RESTORE1)    -
      CATALOG -
      TOL(ENQF) WAIT(0,0)
```

- If no DB2 support , under the //DIRECT DD statement we specify the input HFS file for the restore:

```
//DIRECT DD *
hfs:/tmp/ESAIVP1JOB064842012-11-23-12.05.59
```

Here the "hfs:" prefix defines the dataset as HFS file, and after the prefix the full or relative HFS path name.

- For the restore SQL queries we have a default dynamic SQL statement :

```
" SELECT * FROM DSSDATASETS A "
" WHERE ( %m% ) AND "
" ( DATE(KEY) <= DATE('%d%') ) AND "
" ( "
" A.KEY =   "
" (SELECT MAX(B.KEY) FROM DSSDATASETS B WHERE A.DATASET = B.DATASET   "
" AND DATE(B.KEY) <= DATE('%d%') ) "
" ) " ;
```

Where: %m% will contain the actual restore mask
%d% will contain the actual date or the date limit from the REFDT expression

Under the //SQL DD statement you can define another dynamic SQL query, using the %m% and %d% variables:

```
//SQL DD     *
 SELECT * FROM DSSDATASETS A
 WHERE ( %m% )
 AND DATE(KEY) BETWEEN DATE('10.02.2004') AND DATE('%d%')
 AND VOLUMES = 'OS21D1'
 AND A.KEY =
 (SELECT MAX(B.KEY) FROM DSSDATASETS B
 WHERE A.DATASET = B.DATASET
 AND DATE(B.KEY) BETWEEN DATE('10.02.2004') AND DATE('%d%')
 AND VOLUMES = 'OS21D1'
 )
```

ENVARS

The environment variables define the options for the MVSDSSN program:
The environment variables have the syntax: variable-name = value

- PLAN the DB2 plan name
- SUBSYS the DB2 subsystem name, (as default the program uses the CAF facility to connect to DB2)
- USEDB2 "YES" or "NO"
- PREFIX the HFS path prefix. The program dynamically generates the HFS dataset path names and creates this files in the directory defined via PREFIX.
  So if I define:
  PREFIX=/mnt/mydumps/change
  The dumps will be written into this subdirectory and the restore will search also in this subdirectory
- COMPR compression level. During the dump the generated dump image will be compressed by the "zlib" general purpose compression library. The compression level parameter defines the compression value. This can be between 0 and 9.
  0 - no compression
  9 -maximal compression.
  We have tested the effect of different compression levels, higher compression lead to increased CPU usage, but the dataset size reduction was minimal.
- DEBUG the debug level 0-3. 0 NO debug, default. During the dump or restore process the program can write diagnostic , debug information
- The actual ENVFILE:

```
PREFIX=HFS:/tmp
SUBSYS=DSN9
PLAN=MVSDSSN
COMPR=1
```

IVP

The IVP (Installation Verification Procedure) contains jobs test the basic functionality. We are using here as dump target the "/tmp" temporary USS file system:

- IVP10DUM dump all the MVSDSSN datasets without DB2 support.
- IVP10RES restore the SAMPLIB with RENAME from the dump created above.
- IVP20DUM dump all the MVSDSSN datasets with DB2 support.
- IVP20RES restore the SAMPLIB with RENAME from the latest dump.
- IVP21DUM a volume DUMP (Z1DT11).

Miscellaneous

Performance

The current DUMP elapsed and CPU times for some 3390-27 (30 Gigabyte) volumes:

Elapsed time in minutes	CPU time in minutes	Disk usage about
9	1.92	35%
13	2.94	48%
19	4.23	60%

:

We got the best performance/size ratio with the default compression level.

Diagnosis

For diagnosis we are using the TYPRUN(NORUN) parameter, and the DEBUG environment variable.

Practical Usage

Our online 3390 disk space about 400 Gigabyte. We have an NFS drive called "stoneage" mounted on our mainframe via NFS client. This has a capacity of 3.5 Terabytes.
We write the daily backups and the weekly full dumps to this drive, and the network administrators keep online this for about a half year. If we need an older backup, we have to wait till the network administrators restore the backup from a tape. Here is the input for our daily backup:

```
DUMP -
      COMPRESS                                                    -
      DATASET(                                                    -
            INCLUDE(AKLEIN.**,                                    -
                  AND*.**,                                        -
                 )                                                -
            EXCLUDE( -
                  *.D0*.T*.P*,   /* ZA DUMP DATASETS */         -
                  *.D1*.T*.P*,                                    -
                  *.D2*.T*.P*,                                    -
                  *.D3*.T*.P*,                                    -
                  *.SPFTEMP*.CNTL,                                -
                  *.ISPF.ISPPROF,                                 -
                  *.SPFLOG*.LIST,                                 -
                 ) -
            BY((DSORG EQ (SAM,PDS,PDSE,VSAM)) (DSCHA EQ 1))       -
           )          -
      RESET OPTIMIZE(4)    OUTDDNAME(DUMP) TOL(ENQF)
```

Practically we are using very seldom the dataset group restore option, we restore individual datasets or complete volumes.

For future development, maybe it would be useful to use the FTP API to communicate directly with an FTP servers, and maybe create 2 or more DUMPs in different locations.