

---

## Appendix C. Linux-z/OS DFSMSHsm Dump or Restore HOW-TO

This appendix describes how to back up your Linux partitions and volumes. z/OS is the IBM operating system that can run on either the IBM S/390™ processor or the IBM eServer zSeries™ processor. The term mainframe refers to either of these two machines. This appendix applies only to Linux for S/390 and Linux for zSeries.

You can include your Linux volumes in an existing z/OS backup solution that uses DFSMSHsm. This backup solution works in two environments: (1) a z/OS focused environment with z/OS running on several LPARs and a few dozen Linux servers running under z/VM in an LPAR; (2) a Linux-focused environment with z/VM running in BASIC mode, with hundreds to thousands of Linux guests, and one or more z/OS guests that will perform the DFSMSHsm processing.

LINUX is a registered trademark of Linus Torvalds. You can direct your comments to [linux390@de.ibm.com](mailto:linux390@de.ibm.com).

To get the latest version of this HOW TO and related utilities, check with your Linux provider.

---

### How DFSMSHsm Supports Linux

This appendix provides a discussion of only the supported DFSMSHsm functions for Linux volumes.

For Linux volumes, DFSMSHsm supports only the dump and recover functions. DFSMSHsm dump is a physical copy of DASD tracks to tape. An automatic dump process initiates dump processing on a repeating schedule. After reading this HOW TO, you will know what to do to have your Linux volumes dumped by DFSMSHsm.

**Note:** Linux volumes should not be managed by SMS.

---

### Requirements

This section presents a discussion of several requirements related to using Linux.

#### Hardware Environment

Linux can run on a mainframe under the following three modes:

- |                |  |
|----------------|--|
| <b>BASIC</b>   | A single operating system that owns the entire mainframe.  |
| <b>LPAR</b>    | A mainframe divided into logical partitions, with each partition running its own operating system image. There can be up to 15 LPARs.                                      |
| <b>VIRTUAL</b> | VM supports hundreds to thousands of guests. Each guest runs its own operating system—for example, one guest using Linux and another using z/OS, all on the same hardware. |

Tests were run with Linux kernel 2.4 and OS/390 R10 under the same z/VM system.

## DASD Background

The mainframe environment can store data on Direct Access Storage Device (DASD) by utilizing an extended count-key-data (ECKD) protocol. DASD is divided into logical units known as volumes. Volumes divide into fixed-size tracks, and the track's geometry determines its size. Linux supports both the 3380 and 3390 track geometries. You can reference a volume by using either a 16-bit device number or a six-character volume serial number (referred to as the *volser*). More than one operating system can access a volume at the same time. A volume can have a different device number for each operating system that has access to it, but only one *volser*. Linux allows multiple volumes to have the same *volser*, while z/OS allows only one volume per *volser* to be accessible at a time.

## Linux Requirements

In order for z/OS to process the Linux volumes, they must be formatted in the compatible disk layout (CDL) using **dasdfmt** version 1.0 or later and partitioned using **fdasd** version 1.0 or later.

## Linux Disk Utilities

### dasdfmt

The default disk layout for **dasdfmt** is CDL. Volumes in the original format, Linux disk layout (LDL), are not compatible with the z/OS system and cannot be accessed and backed up by z/OS. Below is an example of how to format a disk with **dasdfmt** at address 0198, having a byte block size of 4096, and a *volser* of LNX200:

```
dasdfmt -n 198 -b 4096 -l lnx200
```

A screen appears like the one below:

```
Drive Geometry: 3339 Cylinders * 15 Heads = 50085 Tracks
I am going to format the device 198 in the following way:
  Device number of device : 0x198
  Major number of device : 94
  Minor number of device : 8
  Labelling device       : yes
  Disk label             : VOL1
  Disk identifier        : LNX200
  Extent start (trk no)  : 0
  Extent end (trk no)    : 50084
  Compatible Disk Layout : yes
  Blocksize              : 4096

--->> ATTENTION! <<--- All data in the specified range of that device
will be lost. Type "yes" to continue, no will leave the disk untouched:
```

You can see a 3339 cylinder volume that is attached to Linux at address 198 in the screen capture above. The volume serial number (*volser*) is LNX200, and its block size is 4096 bytes—the optimal and recommended block size. The *volser* must be six characters in length. The disk label, VOL1, indicates that a z/OS system can process this volume.

An LDL volume has a disk label of LNX1. z/OS cannot process any volume with a disk label of LNX1.

## fdasd

After you have formatted a volume with **dasdfmt**, you must partition it before Linux can use it. Use the **fdasd** program to partition the volume. This program is similar to the **fdisk** program that comes with the Linux version that runs on personal computers. One difference is that it creates partitions on ECKD DASD instead of hard drives. With **fdasd**, you can create up to three partitions on a volume. Each partition appears to z/OS as a data set. **fdasd** asks you to choose the beginning track and ending track or size of the partition. The naming conventions for the data sets and partitions are listed in “Data Set and Partition Naming Conventions” on page 638.

Follow these rules when you create partitions with **fdasd** that you want to back up with z/OS tools.

- Create partitions by starting from the lowest possible track.
- Do not leave gaps between partitions.
- If you want to restore a partition, *do not* delete it with **fdasd** first. When you delete the partition with **fdasd**, the naming convention between the partition names and data set names is no longer valid. If you then recover a partition on that volume, you could recover over the wrong partition, losing data. When you restore the partition, you replace the existing partition.
- If you follow the previous rules of thumb above, then you can add partitions to a volume at any time up to a maximum of three partitions or until the entire volume is used. If you need to reorganize the partitions, you should back up the data on the partitions and then delete all partitions and create new ones in the desired configuration.

## z/OS Requirements

The tools that IBM supports for backing up Linux volumes are DFSMSdss (Data Facilities Storage Management Subsystem data set services) and DFSMShsm. DFSMSdss and DFSMShsm are elements of z/OS and OS/390 for backing up data and managing disk space. The z/OS image must also be able to attach and vary online the volumes that Linux is using that you want backed up.

## Authorization Requirements

You need to have root authority for Linux in order to mount and unmount the partitions, as well as to format and partition the volumes using **dasdfmt** and **fdasd**. For z/OS, you need authority to submit DFSMShsm commands and, optionally, to run ADRDSSU, which is the program that is invoked when using DFSMSdss. z/OS sees the Linux partitions as data sets. You can prevent unauthorized access of the Linux partitions by z/OS applications and users by using the IBM RACF, a component of the Security Server for z/OS.

---

## Setting Up Your Dump Environment for Linux Volumes

A Linux volume is formatted with **dasdfmt** in the CDL format and partitions put on it with **fdasd**. It has a valid VTOC on it and a data set for each Linux partition. A Linux volume holds only Linux partitions. No MVS or VM data sets or any non-Linux data should be allowed on a Linux volume. A Linux volume can hold one to three partitions and can be a 3380 or 3390.

The data sets created by **fdasd** are not added to or removed from the catalog by **fdasd** as they are allocated and deleted. If you manually catalog the data sets, you must uncatalog them if you delete them with **fdasd**. It is recommended that you

leave the data sets uncataloged. The partition and data set naming conventions are listed in “Data Set and Partition Naming Conventions” on page 638.

## Performing Only Dumps

DFSMSHsm can only dump and recover from dump those Linux volumes that are formatted with **dasdfmt** and partitioned with **fdasd**. All other DFSMSHsm functions are unsupported for Linux volumes. No data sets should be migrated off or recalled to Linux volumes. Backup is not supported for Linux volumes or partitions. Use Linux volumes only as PRIMARY volumes and use them with the NOAUTOBACKUP, NOAUTOMIGRATION, and NOAUTORECALL options. For the dump class that you use with Linux, use the option VTOCCOPIES(0). VTOCCOPIES is not supported for Linux volumes.

## When Do You Want Your Dumps Performed?

If you already use automatic dump, then a time of day and dump cycle is already in place. If not, then typically the time and day of the week is when the system is not very busy. The daily dump cycle consists of a repeating pattern of up to 31 days where dumps are made or not made. Most DFSMSHsm administrators use a weekly or daily dump cycle. You can choose any days where dumps are made to have your volumes dumped; however, some dump days might be busier than others, limiting tape hardware availability on those days. You can also specify how frequently you perform dumps by the minimum number of days that must pass from one dump to the next. If you do not already have automatic dump set up, then tape hardware availability and DASD write activity are the only factors limiting when you can dump your Linux volumes. You can also perform a manual dump of your volumes for special situations.

When scheduling your dumps, you must ensure that no writes to the volumes that you are dumping occur while the dumps are being made. The best way to ensure the data integrity of your partitions and their dumps is to unmount them or to have them mounted as read-only. By unmounting a partition, you flush all buffered data that has not yet been written to the partition.

## How Long Do You Want to Keep Your Dumps?

You can specify how many days you want to retain your dumps. You may want to specify how many days you retain your dumps in multiples of how frequently in days that you dump. Then you will always have the same number of dump generations at your disposal.

## How Many Dumps Are Stacked on Each Dump Tape?

The capacity of current tape technology allows several DASD volumes to be dumped to one dump tape. Only one dump of a specific volume can exist on a dump tape. Once a dump tape has dumps on it, no new dumps can be added to it on any subsequent dump day. Also, a dump tape holds only dumps from one dump class.

Consideration: If you have data that belongs to customers, you may not want to have data from more than one customer on the same tape. If your customer has a right to request all copies of data, and if the data co-exists on a dump tape with another customer’s data, you will have to copy the data to another tape. Assigning different dump classes to different customers ensures that their data is on separate tapes.

## How Many Tape Drives Do You Have?

Your mainframe environment probably has many tape drives, but you probably will not be able to use all of them. To limit the number of tape drives that DFSMShsm uses for dump, use the SETSYS command with the MAXDUMPTASKS option. Once you set the maximum number of dump tasks, you prevent DFSMShsm from taking up too many tape drives and delaying other functions that also require tape drives.

---

## Recovering Your Data

When recovering data, unmount any partitions that are being recovered during the recovery process to ensure data integrity. There are several ways you can recover your data from a DFSMShsm dump. You can recover an entire volume with DFSMShsm or DFSMSdss. You can also recover a partition from a dump with DFSMShsm or DFSMSdss. To use DFSMSdss to process Linux volumes, see the HOW TO appendix in the *z/OS DFSMSdss Storage Administration Guide*. When a volume or partition is recovered, the dump overwrites it; the partition or volume then returns to the state at which the dump was made. All changes since the dump was made are lost. See “Data Set and Partition Naming Conventions” on page 638 on the data set and partition naming conventions to make sure that the right partition is recovered.

There are some limitations in how you can recover to a Linux volume. For a volume of more than 65536 tracks, you can recover only the full volume from a dump; no individual data sets can be recovered to it with DFSMShsm. However, you can recover a data set to a volume over 65536 tracks from a DFSMShsm dump with DFSMSdss. To use DFSMSdss on Linux volumes and partitions, see the HOW-TO appendix in the *z/OS DFSMSdss Storage Administration Guide*. Either a full volume or an individual data set can be recovered to a volume of less than 65536 tracks by DFSMShsm. When you recover a data set or volume with DFSMShsm, you must specify the FROMDUMP(DUMPVOLUME(volser)) option. You can list the dump volumes with the LIST DUMPVOLUME command.

## Recovering a Volume with DFSMShsm

To recover a volume with DFSMShsm, you need to know only the volser of the volume you want recovered and from how far back you want your volume recovered. Remember, you can go only as far back as the number of days you have been keeping your dumps. You can go back by date or by the generation of dump where generation 0 is the most recent dump, generation 1 is the second most recent dump, and so on.

The following command recovers volume LNX400 from the third most recent dump.

```
RECOVER * TOVOLUME(LNX400) UNIT(3390) -  
FROMDUMP(DUMPGENERATION(2))
```

You can also recover a dump to a volume other than the one that was used to make the dump with DFSMShsm. For instance, you can dump volume LNX400 and recover the dump to volume SPARE5; but in the process, volume SPARE5 is renamed with volume LNX400's volser, which is not a problem for Linux as it is for z/OS. The z/OS operating system cannot have two or more volumes with the same volser online at the same time. The original volume must be varied offline before the recovery is started. You need to know the volsers of both the volume you want recovered and the volume you want the dump recovered to.

The following command recovers the most recent dump of volume LNX400 to volume SPARE5 and changes the volser of volume SPARE5 to LNX400. Volume LNX400 must be varied offline in MVS before this recovery command is issued.

```
RECOVER * TOVOLUME(LNX400) UNIT(3390) TARGETVOLUME(SPARE5) -
FROMDUMP
```

If you do not have an extra volume attached to your Linux environment, you need to shut it down and reconfigure your environment with the extra volume and reboot your Linux environment. You do not have to do this in MVS.

## Recovering a Volume with DFSMSdss

To recover a volume from a DFSMSShsm dump with DFSMSdss, use the LIST command to find the volser of the dump tape volume that contains the dump you want to recover. See “The DFSMSShsm LIST Command” on page 637 on how to use the LIST command. You can search the output data set for the volser of the volume you want to recover. Once you find the dump that you want to recover in the output data set, you will find the volser of the dump volume and the data set name of your dump. If you recover the dump to an alternate volume, DFSMSdss does not change the volser to that of the volume that the dump was created from. You need to submit JCL to use DFSMSdss. Once you have the information that you need from the output data set, you can submit a job like the following to recover your volume.

The following JCL recovers the dump data set DFHSM.DMP.DAILY.VLNX300.D01244.T072510 to volume LNX300 from dump tape A02001.

```
//LNXJOB JOB , 'IBMUSER',MSGLEVEL=(1,1),TIME=(5,0),REGION=4096K,
//      MSGCLASS=A,CLASS=A
//STEPDSS EXEC PGM=ADRDSU
//SYSPRINT DD SYSOUT=*
//SOURCE  DD UNIT=3480,VOL=SER=A02001,DISP=OLD,
//          DSN=DFHSM.DMP.DAILY.VLNX300.D01244.T072510
//TARGET  DD UNIT=3390,VOL=SER=LNX300,DISP=OLD
//SYSIN   DD *
//        RESTORE FULL INDDNAME(SOURCE) OUTDDNAME(TARGET) PURGE
/*
```

## Recovering a Partition with DFSMSHsm

When you recover a partition, leave the original damaged partition in place. If you do not still have the original partition or you want to recover the partition to another volume, then you first must preallocate a partition of the exact same number of tracks with **fdasd**. You must know the volser of the volume that contains the partition you want to recover, which partition to recover, and how far back you want to recover from. If you had to preallocate a partition, then you also need to know the volser of the preallocated partition and which partition you preallocated. The preallocated partition does not need a file system.

If a Linux user does not follow the rules of thumb in “fdasd” on page 627, then the data set name of one or more of the partitions could be changed. The **fdasd** utility initially creates the data sets in order where PART0001 is always the data set that has the lowest starting track, PART0002 has the next lowest starting track, and PART0003 has the highest starting track. Once a partition is removed with **fdasd**,

any other changes made after the first removal may cause the order and naming not to be valid. See the naming conventions in “Naming Conventions” on page 638.

You also need to know the volser of the dump tape that contains the dump from which you want to recover. To find the volser of the dump tape, use the LIST command. See “The DFSMSShm LIST Command” on page 637 on how to use the LIST command.

The following command recovers data set LINUX.VLNX300.PART0001.NATIVE on volume LNX300 from the dump of LNX300 on dump tape LX0001. If the dump of volume LNX300 was the only dump on the dump tape, omit the SOURCEVOLUME parameter, but if dumps from two or more volumes are stacked on the dump tape, then you must specify SOURCEVOLUME.

```
RECOVER LINUX.VLNX300.PART0001.NATIVE TOVOLUME(LNX300) -  
UNIT(3390) FROMDUMP(DUMPVOLUME(LX0001) SOURCEVOLUME(LNX300)) -  
REPLACE
```

To recover a data set to an alternate location, use the RENAME and TOVOLUME options to rename the data set (using the naming convention listed in “Data Set and Partition Naming Conventions” on page 638) and to specify the volume that you want the data set recovered to.

The following command recovers data set LINUX.VLNX300.PART0002.NATIVE from the dump on dump tape LX0002 to volume LNX100 and renames it LINUX.VLNX100.PART0001.NATIVE, replacing the preallocated data set on LNX100.

```
RECOVER LINUX.VLNX300.PART0002.NATIVE -  
FROMDUMP(DUMPVOLUME(LX0002)) -  
NEWNAME(LINUX.VLNX100.PART0001.NATIVE) REPLACE -  
TOVOLUME(LNX100) UNIT(3390)
```

## Recovering a Partition with DFSMSdss

To recover a partition from a DFSMSShm dump with DFSMSdss, use the LIST command to get the same information needed to recover a volume with DFSMSdss. See “The DFSMSShm LIST Command” on page 637 on how to use the LIST command. You also need the data set name for the data set that holds the partition you want recovered and the data set name for the partition that you want the dump recovered to.

The following JCL recovers LINUX.VLNX300.PART0001.NATIVE from dump DFHSM.DMP.DAILY.VLNX300.D01244.T072510 on dump tape A02001 to volume LNX200 and renames it LINUX.VLNX200.PART0001.NATIVE, replacing the data set of the same name on LNX200. If you are restoring the data set to its original volume, then you can omit the RENAME portion.

```

//LNXJOB JOB , 'IBMUSER',MSGLEVEL=(1,1),TIME=(5,0),REGION=0M,
//      MSGCLASS=A,CLASS=A
//STEPDSS EXEC PGM=ADRDSU
//SYSPRINT DD SYSOUT=*
//SOURCE  DD UNIT=3480,VOL=SER=A02001,DISP=OLD,
           DSN=DFHSM.DMP.DAILY.VLNX300.D01244.T072510
//TARGET  DD UNIT=3390,VOL=SER=LNX200,DISP=OLD
//SYSIN   DD *
RESTORE INDDNAME(SOURCE) OUTDDNAME(TARGET) -
        DATASET(INCLUDE(LINUX.VLNX300.PART0001.NATIVE)) -
        RENAME(LINUX.VLNX200.PART0001.NATIVE) REPLACE
/*

```

## Example Dump Environment

This example consists of one 3390-3 volume of about 2.9 gigabytes and one 3390-9 volume of about 8.7 gigabytes and a 65 cylinder 3390 of about 45 megabytes. The volumes will be dumped at 10:00 am to 3480 tapes on two 3480 tape drives. The 3390-3 and 3390-9 will be dumped daily and the 65 cylinder will be dumped weekly. The dump cycle will be a seven day, weekly cycle. The **fdasd** utility will be used to partition each volume as follows.

```

Disk /dev/dasd/0199/disc:
  65 cylinders,
  15 tracks per cylinder,
  12 blocks per track
  4096 bytes per block
volume label: VOL1, volume identifier: LNX100
maximum partition number: 3

-----tracks-----
   Device      start   end   length  Id System
 /dev/dasd/0199/part1      2    974    973    1 Linux native

Disk /dev/dasd/0198/disc:
  3339 cylinders,
  15 tracks per cylinder,
  12 blocks per track
  4096 bytes per block
volume label: VOL1, volume identifier: LNX200
maximum partition number: 3

-----tracks-----
   Device      start   end   length  Id System
 /dev/dasd/0198/part1      2    3201   3200    1 Linux native
 /dev/dasd/0198/part2   3202   50084   46883    2 Linux native

Disk /dev/dasd/019a/disc:
  10017 cylinders,
  15 tracks per cylinder,
  12 blocks per track
  4096 bytes per block
volume label: VOL1, volume identifier: LNX300
maximum partition number: 3

-----tracks-----
   Device      start   end   length  Id System
 /dev/dasd/019a/part1      2    3201   3200    1 Linux native
 /dev/dasd/019a/part2   3202    4267   1066    2 Linux native
 /dev/dasd/019a/part3   4268  150254  145987    3 Linux native

```

The partitions will be mounted as follows.

```

/dev/dasd/0198/part1 on /example type ext2 (rw)
/dev/dasd/0199/part1 on /example/boot type ext2 (rw)
/dev/dasd/0198/part2 on /example/usr type ext2 (rw)
/dev/dasd/019a/part1 on /example/var type ext2 (rw)
/dev/dasd/019a/part2 on /example/tmp type ext2 (rw)
/dev/dasd/019a/part3 on /example/home type ext2 (rw)

```

This example is designed to demonstrate how one might implement a simple DFSShsm dump environment for a Linux guest on z/VM. It is not intended to demonstrate how to best partition volumes for Linux.

## How to Define Your Dump Cycle

Before you start automatic dump, you must define a dump cycle. This environment uses a daily dump cycle. The start date must be today's date or a date that has already passed. You need to use a past date if you are using a dump cycle of more than one day and today is not the first day in the dump cycle.

The command below has the first Y day coincide with the day of 2001/08/26, which is a Sunday. Of the dump classes listed in the next section, the WEEKLY dump class has a DAY value of 7, so it is dumped on Saturdays. The MONTHLY dump class has a DAY value of 1, so it is dumped on Sundays. Because the DAILY dump class has no DAY value specified, it dumps on all Y days. If an N were in the dump cycle, automatic dump would not dump any volumes on that day.

```
DEFINE DUMPCYCLE(YYYYYYY CYCLESTARTDATE 2001/08/26)
```

## How to Define Your Dump Classes

LNx100 must be backed up only once a week because it will not change very often. LNx200 and LNx300 change more often and must be backed up daily. Because this environment uses a weekly dump cycle, the DAY and FREQUENCY options determine which days each dump class is dumped. The daily and weekly dumps are short-term dumps that are not be kept for long. Another dump class creates dumps monthly and keeps the dumps for three months. All of the dump classes have a STACK value of three to allow dumps from up to three volumes to be put on one dump tape. One is the default STACK value, meaning that no stacking will occur. The following commands define the dump classes.

```
DEFINE DUMPCLASS(DAILY AUTOREUSE FREQUENCY(0) -
RETENTIONPERIOD(3) STACK(3) UNIT(3480) VTOCCOPIES(0))
```

```
DEFINE DUMPCLASS(WEEKLY AUTOREUSE FREQUENCY(0) DAY(7) -
RETENTIONPERIOD(28) STACK(3) UNIT(3480) VTOCCOPIES(0))
```

```
DEFINE DUMPCLASS(MONTHLY AUTOREUSE FREQUENCY(28) DAY(1) -
RETENTIONPERIOD(90) STACK(3) UNIT(3480) VTOCCOPIES(0))
```

If, for example, you have only two tape drives at your disposal, use the SETSYS command to set the maximum number of dumps that may be taken at any one time.

```
SETSYS MAXDUMPTASKS(2)
```

## How to ADDVOL Your Volumes

Because the volumes will only be dumped, automatic migration and backup features are not allowed. When the dumps expire, the dump tapes are automatically reused for later dumps.

```
ADDVOL LNX100 UNIT(3390) PRIMARY(AUTODUMP(WEEKLY,MONTHLY) -  
NOAUTOBACKUP NOAUTOMIGRATION NOAUTORECALL)
```

```
ADDVOL LNX200 UNIT(3390) PRIMARY(AUTODUMP(DAILY,MONTHLY) -  
NOAUTOBACKUP NOAUTOMIGRATION NOAUTORECALL)
```

```
ADDVOL LNX300 UNIT(3390) PRIMARY(AUTODUMP(DAILY,MONTHLY) -  
NOAUTOBACKUP NOAUTOMIGRATION NOAUTORECALL)
```

## How to Start Automatic Dumping

To automatically dump the volumes, issue the SETSYS AUTODUMPSTART command with the start and finish times.

```
SETSYS AUTODUMPSTART(1100 1200 0000)
```

The times are in a 24-hour format. The first time is when automatic can start, the second time is the latest that an automatic dump can start, and the third time is the latest that a dump of a volume can start. Because the third time is 0000, all volumes are dumped if automatic dump starts. To turn off automatic dump, resubmit the command with all three times equal to 0000.

To manually dump the volumes, issue the BACKVOL command.

```
BACKVOL VOLUMES(LNX100,LNX300,LNX400) -  
DUMP(DUMPCLASS(WEEKLY,MONTHLY))
```

If you omit DUMPCLASS(WEEKLY,MONTHLY), the volumes are dumped to the dump classes listed for each volume with the AUTODUMP option when it was added with ADDVOL, but no stacking occurs. You can list up to five dump classes. Dumps made manually are taken into account when recovering from a dump by generation or date. If you are only going to be dumping manually, you can omit the AUTODUMP option when you ADDVOL your volumes. You can use any valid dump class, regardless of the AUTODUMP option.

If you decide to create your dumps manually, you still need to run automatic dump to have old dumps expire. To have automatic dump run without actually taking dumps, you can define a dump cycle of one N day.

## Recovering from a Dump

Before recovering any data, you must leave any partitions that you want to recover in place or reallocate them with **fdasd**; otherwise, Linux cannot see the partitions that have been recovered. Listed here are only a few basic examples of recovering data. For a full description on how to recover data, see the *z/OS DFSMSHsm Storage Administration Reference* and the *z/OS DFSMSdss Storage Administration Reference*.

### Recovering with DFSMSHsm

To recover a volume with DFSMSHsm, issue the RECOVER command.

The following RECOVER command recovers LNX200 from the most recent dump.

```
RECOVER * TOVOLUME(LNX200) UNIT(3390) FROMDUMP
```

The following RECOVER command recovers LNX200 from the most recent dump of the MONTHLY dump class.

```
RECOVER * TOVOLUME(LNX200) UNIT(3390) -  
FROMDUMP(DUMPCLASS(MONTHLY))
```

The following RECOVER command recovers LNX200 from the third most recent dump. Generation 0 is the most recent dump, 1 is the second most recent, and so on.

```
RECOVER * TOVOLUME(LNX200) UNIT(3390) -  
FROMDUMP(DUMPGENERATION(2))
```

To recover a partition from a DFSMSHsm dump with DFSMSHsm, you again use the RECOVER command and the LIST command. See “The DFSMSHsm LIST Command” on page 637 on how to use the LIST command. You must use the LIST command to find the dump tape that has the dump from which you want to recover your partition. You also need to know the data set and partition naming conventions so that you recover the correct data set for your partition. The naming conventions are listed in “Naming Conventions” on page 638.

The following RECOVER command recovers data set LINUX.VLNX200.PART0002.NATIVE that holds partition part2 on LNX200. You must specify the SOURCEVOLUME parameter if the dump class has a STACK value greater than one because the dump tape may hold more than one dump data set.

```
RECOVER LINUX.VLNX200.PART0002.NATIVE -  
FROMDUMP(DUMPVOLUME(LX0001) SOURCEVOLUME(LNX200)) REPLACE
```

### Recovering Data from DFSMSHsm Dump with DFSMSdss

Before you recover a volume from a DFSMSHsm dump with DFSMSdss, you must use the LIST command to see what dumps you have. See “The DFSMSHsm LIST Command” on page 637 on how to use the LIST command. Once you find the dump you want to use, you can write and submit JCL to have DFSMSdss recover the volume.

The following JCL recovers dump data set DFHSM.DMP.MONTHLY.VLNX300.D01245.T234610 to volume LNX300. If there is more than one dump stacked on the tape, you must specify the file sequence number of the label keyword. You can get the file sequence number from the information provided by the LIST command in the next section.

```
//LNXJOB JOB ,USER=IBMUSER,PASSWORD=IBM,MSGLEVEL=(1,1),TIME=(5,0),REGION=4096K,  
//      MSGCLASS=A,CLASS=A  
//STEPDSS EXEC PGM=ADRDSU  
//SYSPRINT DD SYSOUT=*  
//SOURCE  DD UNIT=3480,VOL=SER=(A02005,A02008,A02010),DISP=OLD,  
//        LABEL=(2),DSN=DFHSM.DMP.MONTHLY.VLNX300.D01245.T234610  
//TARGET  DD UNIT=3390,VOL=SER=LNX300,DISP=OLD  
//SYSIN   DD *  
RESTORE INDDNAME(SOURCE) OUTDDNAME(TARGET) PURGE  
/*
```

To recover a partition from a DFSMSShsm dump with DFSMSDss, use the LIST command like before to find the dump tape and dump data set from which you want to recover your partition. You also need the data set name of your partition. See “Data Set and Partition Naming Conventions” on page 638 for the partition and data set naming conventions.

The following JCL recovers data set LINUX.VLNX300.PART0001.NATIVE from dump data set DFHSM.DMP.MONTHLY.VLNX300.D01245.T234610 to volume LNX300.

```
//LNXJOB JOB ,USER=IBMUSER,PASSWORD=IBM,MSGLEVEL=(1,1),TIME=(5,0),REGION=4096K,
//      MSGCLASS=A,CLASS=A
//STEPDSS EXEC PGM=ADRSSU
//SYSPRINT DD SYSOUT=*
//SOURCE  DD UNIT=3480,VOL=SER=(A02005,A02008,A02010),DISP=OLD,
//          LABEL=(2),DSN=DFHSM,DMP.MONTHLY.VLNX300.D01245.T234610
//TARGET  DD UNIT=3390,VOL=SER=LNX300,DISP=OLD
//SYSIN   DD *
RESTORE INDDNAME(SOURCE) OUTDDNAME(TARGET) -
        DATASET(INCLUDE(LINUX.VLNX300.PART0001.NATIVE)) REPLACE
/*
```

## Submitting DFSMSShsm Commands Via JCL and FTP

You can submit DFSMSShsm commands by sending JCL files through ftp to z/OS and have output data sets sent back to your Linux system. You must have TCP/IP and ftp running and a userid and password authorized to use ftp. Once you log into the z/OS ftp server you must enter the following command.

**site filetype=jes**

```
//LNXDUMP JOB ,CLASS=A,USER=IBMUSER,PASSWORD=IBM,
//      MSGCLASS=H,REGION=4096K,MSGLEVEL=(1,1)
//LISTDUMP EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=A
//SYSTSIN  DD *
HSEND LIST DUMPVOLUME ODS(LIST.DUMP.VOLUMES)
//FTPDUPE EXEC PGM=FTP,COND=(0,NE,LISTDUMP),
//          PARM='1.2.3.4'
//SYSPRINT DD SYSOUT=*
//OUTPUT   DD SYSOUT=*
//SAVE     DD *
/*
//INPUT    DD *
userid
password
ascii
put 'LIST.DUMP.VOLUMES' dump.list
quit
/*
```

The IKJEFT01 program in the above JCL allows you to submit TSO line commands. The HSEND command allows DFSMSShsm commands to be submitted from TSO. After HSEND you can use any DFSMSShsm command. In the second step, the data set LIST.DUMP.VOLUMES is sent through ftp to IP address 1.2.3.4 and given a file name of dump.list .

## The DFSMSHsm LIST Command

The LIST command can be used to list several types of information about a DFSMSHsm environment. For Linux volumes, we need to list only information for dump tapes. To create a list of dump tapes, use the following command.

```
LIST DUMPVOLUME OUTPUTDATASET(LIST.DUMP.VOLUMES)
```

The above command creates data set LIST.DUMP.VOLUMES. You can give the data set your own data set name. You can then browse the data set and search it for the information you need. The list is sorted by the volser of the dump tapes. A sample list is below.

```
-- DFSMSHSM CONTROL DATASET -DUMP VOLUME-BCDS-- LISTING      --- AT 11:18:56 ON 01/09/02 FOR SYSTEM=3090
```

DUMP VOLSER	VOL STATUS	UNIT TYPE	FILE SEQ	SOURCE VOLSER	SMS	CLASS	DUMPED DATE	DUMPED TIME	EXP DATE	IDRC	LIBRARY	PCT FULL	SET OF DUMP VOLSERS
A02001	UNEXP	3480	01	LNx200	N	DAILY	2001/09/01	10:00:01	2001/09/04	N	*NO LIB*	100	A02001
			02	LNx300	N		2001/09/01	10:25:07					A02001 A02002
													A02003
A02002	UNEXP	3480	**	LNx300	N	DAILY	2001/09/01	10:25:07	2001/09/04	N	*NO LIB*	100	A02001 A02002
													A02003
A02003	UNEXP	3480	**	LNx300	N	DAILY	2001/09/01	10:25:07	2001/09/04	N	*NO LIB*	255	A02001 A02002
													A02003
A02004	UNEXP	3480	01	LNx200	N	DAILY	2001/09/02	10:00:01	2001/09/05	N	*NO LIB*	100	A02004
			02	LNx300	N		2001/09/02	10:46:23					A02004 A02007
													A02009
A02005	UNEXP	3480	01	LNx200	N	MONTHLY	2001/09/02	10:00:01	2001/11/25	N	*NO LIB*	100	A02005
			02	LNx300	N		2001/09/02	10:46:23					A02005 A02008
													A02010
A02006	UNEXP	3480	01	LNx100	N	MONTHLY	2001/09/02	10:00:02	2001/11/25	N	*NO LIB*	25	A02006
A02007	UNEXP	3480	**	LNx300	N	DAILY	2001/09/02	10:46:23	2001/09/05	N	*NO LIB*	100	A02004 A02007
													A02009
A02008	UNEXP	3480	**	LNx300	N	MONTHLY	2001/09/02	10:46:23	2001/11/25	N	*NO LIB*	100	A02005 A02008
													A02010
A02009	UNEXP	3480	**	LNx300	N	DAILY	2001/09/02	10:46:23	2001/09/05	N	*NO LIB*	255	A02004 A02007
													A02009
A02010	UNEXP	3480	**	LNx300	N	MONTHLY	2001/09/02	10:46:23	2001/11/25	N	*NO LIB*	255	A02005 A02008
													A02010
A02101	UNEXP	3480	01	LNx100	N	WEEKLY	2001/09/01	10:00:02	2001/09/29	N	*NO LIB*	25	A02101

```
----- END OF - DUMP VOLUME - LISTING -----
```

Once you find the dump tape made on the date and of the same dump class as the volume you want to recover from, you must submit another LIST command to list the dump data sets on that dump tape.

```
LIST DUMPVOLUME(A02001) ODS(LIST.DUMP.A02001)
```

The above command creates the following list of the dump information about dump tape A02001.

```

-- DFSMSHSM CONTROL DATASET -DUMP VOLUME-BCDS-- LISTING      --- AT 11:21:09 ON 01/09/02 FOR SYSTEM=3090
DUMP VOL UNIT FILE SOURCE DUMPED DUMPED PCT SET OF DUMP
VOLSER STATUS TYPE SEQ VOLSER SMS CLASS DATE TIME EXP DATE IDRC LIBRARY FULL VOLSERS
A02001 UNEXP 3480 01 LNX200 N DAILY 2001/09/01 10:00:01 2001/09/04 N *NO LIB* 100 A02001
DUMP COPY DATA SET NAME = DFHSM.DMP.DAILY.VLNX200.D01244.T010010
02 LNX300 N 2001/09/01 10:25:07 A02001 A02002
A02003
DUMP COPY DATA SET NAME = DFHSM.DMP.DAILY.VLNX300.D01244.T072510
----- END OF - DUMP VOLUME - LISTING -----

```

From the sample listing above, you can see that dump tape A02001 contains two dump data sets, one of volume LNX200 and the first part of the dump data set for volume LNX300. The name for the dump data set of volume LNX200 is DFHSM.DMP.DAILY.VLNX200.D01244.T010010 and the file sequence number is 1. The dump data set for volume LNX300, DFHSM.DMP.DAILY.VLNX300.D01244.T072510, starts on dump tape A02001 and spans onto dump tapes A02002 and A02003. The file sequence number is 2. You need only the volser of the first dump tape when using the RECOVER command to recover this dump.

## Naming Conventions

The following naming conventions apply to data sets, partitions, and volsers.

### Data Set and Partition Naming Conventions

The data sets and the partitions they hold have the following naming convention.

Dataset Names	Partition Names
LINUX.Vvolser.PART0001.type	/dev/dasd/yyyy/part1
LINUX.Vvolser.PART0002.type	/dev/dasd/yyyy/part2
LINUX.Vvolser.PART0003.type	/dev/dasd/yyyy/part3

Where *volser* is the volume serial number of the volume that the data set is on. *yyyy* is the device number of the volume in the Linux environment. Type can be NATIVE or SWAP. Only NATIVE partitions need to be backed up. SWAP partitions are the Linux equivalent to page packs.

### Volser Naming Conventions

The volume serial number of a volume must be 6 characters of upper case letters or numbers. The volser can also have @ , # , and \$ .

Valid Volsers

LINUXA	LNX100	\$BILLS	VOL#11
123456	A@FB2#	\$@#32A	5#2\$XA

---

## References

You can search and view mainframe documentation online or download it in the pdf format from the following web pages.

- For os/390 at [http://www-1.ibm.com/servers/s390/os390/bkserv/find\\_books.html](http://www-1.ibm.com/servers/s390/os390/bkserv/find_books.html)
- For z/OS at [http://www-1.ibm.com/servers/eserver/zseries/zos/bkserv/find\\_books.html](http://www-1.ibm.com/servers/eserver/zseries/zos/bkserv/find_books.html)
- <http://www.redbooks.ibm.com/>

To find DFSMShsm or DFSMSdss documentation, search either of the first two sites for any of the following manuals by title or serial number.

- *z/OS DFSMShsm Storage Administration Guide*
- *z/OS DFSMShsm Storage Administration Reference*
- *z/OS DFSMShsm Implementation and Customization Guide*
- *z/OS DFSMSdss Storage Administration Guide*
- *z/OS DFSMSdss Storage Administration Reference*
- *z/OS MVS JCL Reference*

At the third link, you can search for more DFSMS documentation, and you can find the Linux for the S/390 Redbook, SG24-4987.

