

# Data Set File System (aka DSFS)- Simplified Administration

By Lionel B. Dyck

October 20, 2023

IBM introduced the Data Set File System (DSFS) with z/OS 2.5 as an option for shell users of z/OS to easily access z/OS datasets using traditional shell services. By shell I'm referring to the OMVS environment using the Unix System Services (USS) filesystem. Included with DSFS are a number of z/OS operator commands, as well as TSO and shell commands, all intended to help manage the DSFS environment. As with nearly all IBM services, the commands are well documented with the expectation that the admins (aka System Programmers) and users will look them up when they are needed. This further assumes that these individuals will even know these exist and where to look for the information.

Enter the z/OS open-source tool **DSFSADM** that was inspired by a presentation given by Kershaw Mehta of IBM at the summer SHARE of 2021. I was however saddened by the fact that it wasn't available for z/OS 2.4 as I worked in an environment with both 2.4 and 2.5 systems and thus was not able to implement it at the time. At the New Orleans SHARE of 2023, I was once again inspired when I attempted to attend a session on DSFS and was unable to get in the door as it was standing-room-only. Once I had a full z/OS 2.5 environment I implemented it and started to experiment with it. I found it worked well but I could never remember all the commands and related parameters.

For those who have not heard about DSFS, or may have heard of it but don't know how it works, once configured it will provide the z/OS shell users (via OMVS or ssh) with the ability to access sequential and partitioned data sets (both PDS and PDSE) using standard shell commands such as cp, grep, vim, awk, etc. without having to specify the z/OS dataset using the arcane // prefix. That is assuming the shell command even supports that nomenclature to access a z/OS dataset. Note that PDSE member generations are **not** supported.

For example, to see the contents of a member of my JCL library while in the OMVS shell, I used the cat command thus:

```

/u/lbdyck>cat /dsfs/txt/lbdyck/jcl.cntl/tmp
//TESTXX JOB LBD, 'LIONEL', REGION=0M, NOTIFY=&SYSUID, CLASS=A
//OUT OUTPUT DEFAULT=YES, JESDS=ALL, OUTDISP=(HOLD,HOLD)
/*
//TSO EXEC PGM=IKJEFT1A
//STEPLIB DD DISP=SHR, DSN=LBDYCK.LIONEL.LOAD
//SYSXSPRT DD SYSOUT=*
//SYSTSIN DD *
PDS TEST.PDS HI :
/*
/u/lbdyck>

```

You will notice that the path `/dsfs/txt/lbdyck/jcl.cntl/tmp` is divided thus:

<code>/dsfs</code>	The root for the DSFS filesystem
<code>/txt</code>	Indicated to DSFS that the files are text files
<code>/lbdyck</code>	The userid for the datasets to be accessed (like the users home directory)
<code>/jcl.cntl</code>	The dataset name
<code>/tmp</code>	The member name as this is a PDS

The **DSFSADM** tool is an ISPF dialog that utilizes the SDSF REXX interface for the z/OS operator commands and the bpxwunix REXX interface for the shell commands.

It is invoked using the DSFSADM REXX command with one of three options.

**A** for admin which is used for the z/OS Operator commands and provides an option for accessing the other two menus.

**D** for the dsadm shell commands, both admin and user.

**U** for the dsadm shell commands typically used by a user. This is the default.

This is the Admin Menu:

```
----- DSFS Admin Menu 1.5 -----
Option ==>

* 1 abort          Causes DSFS to Dump and Restart
* 2 dump           Dumps DSFS including the z/OS Unix address space
                   and associated address spaces
* 3 fsinfo         Display detailed information about the DSFS Utility
                   file system
* 4 query          Displays All DSFS counters or values
* 5 reset          Resets all DSFS counters to zero
  6 dsadm commands dsadm commands (admin and user)
  7 user commands  dsadm commands (user only)

      * These actions require SDSF Operator Command capability.
```

Be careful of the **abort** (option 1) and **dump** (option 2) options as they will do what they say. Option 5, **reset**, should be used sparingly (imho).

And a sample report after entering **3** for the **fsinfo** command:

```
Menu Utilities Compilers Help

BROWSE    SYS23280.T110432.RA000.LBDYCK.R0101381          Line 0000000000 Col 001 103
Command ==> _____ Scroll ==> CSR

***** Top of Data *****
CBT      2023280  11:04:31.55      ISF031I  CONSOLE LBDYCK ACTIVATED
CBT      2023280  11:04:31.55      -F DSFS,fsinfo
CBT      2023280  11:04:31.55      IDFS00248I DSFS kernel: MODIFY command - FSINFO accepted.
CBT      2023280  11:04:31.55      IDFS00169I Starting FSINFO command.
CBT      2023280  11:04:31.55      IDFS00261I File System Status:
                                     File System Name:      CBT.DSFS.UTILITY.FS.CBT

                                     System:      CBT      Devno:      19
                                     Size:      1440000K      Free 8K Blocks:  174183
                                     Free 1K Fragments:  7      Log File Size:  14400K
                                     Bitmap Size:      208K      Anode Table Size: 25016K
                                     File System Objects: 99845      Version:      1
                                     Overflow Pages:  0      Overflow HighWater: 0
                                     Space Monitoring: 0,0
                                     ENOSPC Errors:  0      Disk IO Errors:  0
                                     Status:      NE,NC

                                     File System Creation Time: Aug 18 22:32:20 2023
                                     Mount Time:      Oct  2 17:42:16 2023
                                     Last Grow Time:      n/a
```

As you can see, the report is presented using ISPF Browse and as this is an operator command presents the system log as returned by the SDSF REXX interface. On the second line of the display you can see the command that was generated F **DSFDS,fsinfo** followed by the commands response.

The **dsadm** menu, option **6** on the admin menu or option **D** with the **DSFSADM** command, presents the set of supported **dsadm** commands, some of which are not included for various reasons but which you'll probably not miss.

```
----- DSFS dsadm Menu 1.5 -----
Option ==> _

1 dsadm configquery  query configuration settings for the DSFS kernel
2 dsadm createparm   Set Creation Parameters
3 dsadm fileinfo     Obtain information on a file or directory
                    txt _          bin _          rec _
                    hlq _____
4 dsadm fsinfo       Obtain Utility File System Information
5 dsadm help         Get Help on Commands
                    config _      configquery _   createparm _
                    fileinfo _    fsinfo _        query _      salvage _
6 dsadm query        Query information for the DSFS kernel
                    compression _  dscache _    filecache _
                    iobydasd _    knpfs _      locking _
                    metacache _   storage _    vnodecache _ reset _
7 dsadm salvage      Salvage Utility File System * Requires uid=0
                    cancel _      verifyonly _

Notes: All options allow only a single selection except query.
       Confirm for query reset and salvage when prompted
       Enter RESET to clear all selections.
```

The **user** menu, option **7** on the admin menu or option **U** with the **DSFSADM** command, includes those commands that I, as the author of this tool, felt are most useful for users (and safest).

```
----- DSFS User Menu 1.5 -----
Option ==> _

1 dsadm createparm Set Creation Parameters
2 dsadm fileinfo Obtain information on a file or directory:
    txt _          bin _          rec _
    hlq _____
3 dsadm fsinfo Obtain utility file system information
4 dsadm help Get Help on Commands:
    config _      configquery _  createparm _
    fileinfo _    fsinfo _      query _
5 dsadm query Query information for the DSFS kernel:
    compression _ dscache _      filecache _
    iobydasd _    knpfs _        locking _
    metacache _   storage _      vnodecache _

Notes: All options allow only a single selection except query.
Enter RESET to clear all selections.
```

Note that the **dsadm fileinfo** will only report on the **createparm** path models that match the provided **hlq**. Thus, if the **hlq** is *userid* and there is a model under *userid.test*, that model will not be reported on.

The **dsadm createparm** option presents a panel to specify those options:

```
----- DSFS dsadm Createparm 1.5 -----
Option ==> _

Using HLQ of LBDYCK

Type:      _ bin or _ rec or _ txt

DSORG:     _ Partitioned or _ Sequential
RECFM:     _ FB or _ VB
LRECL:     _____
BLKSIZE:   _____
Space:     _ Trks or _ Cyl
Primary:   _____ and Secondary _____

Notes:  1. The HLQ must start with the active users User ID and
         there must be at least one dataset under the HLQ
        2. For BLKSIZE a value of 0 will use the system determined blksize
        3. If RECFM is FB and BLKSIZE is not 0, then it must be a multiple
         of the LRECL
```

The HLQ, or high-level-qualifier, is pre-filled in with the active users **userid** as is required by DSFS at this time. And another requirement is that at least one dataset already be allocated under the provided HLQ for this to work.

### Where to get DSFSADM

To get this open-source tool there are two options:

1. <https://github.com/lbdyck/dsfsadm>
2. File 312 on <https://cbttape.org> (check <https://cbttape.org/updates.htm> first)

### IBM Ideas

The IBM Ideas portal is where you can help to influence the direction of any IBM product. In this case, as of the time this article was written, there are 11 Ideas for DSFS.

The link that I use to access the Ideas portal is: <https://ibm-z-hardware-and-operating-systems.ideas.ibm.com/ideas>. Then in the 'Search all ideas' field enter **dsfs** and then enter to do the search.

You can vote for any idea as well as add your own comments if you are so inclined. Both of these will require that you register a userid with IBM.

Among the ideas are:

[Enhance DSFS CreateParm to support masking for the path](#)

[Add DSFS DeleteParm to remove obsolete HLQ's](#)

[DSFS CreateParm should NOT require a previous allocation under secondary qualifiers](#)

[DSFS Fileinfo should report All models under HLQ](#)

[DSFS CreateParm limitations](#)

[DSFS support for GDGs](#)