

# z/OS (OS/390) Disk Data Sets At CNS

**CNS Document ID: D0045**

Last Updated: 07/20/2006

This document describes how to create, store, and delete z/OS (OS/390) data sets at CNS.

See also CNS document D0071, *Utilities for z/OS (OS/390) Disk Data Sets*.  
[<http://docweb.cns.ufl.edu/docs/d0071/>]

Related subjects include input/output, JCL, charges, catalog, bulk storage, VSAM, and utilities.

**UF Computing & Networking Services**

**CNS Information Services**

2124 NE Waldo Rd, Suite 2100  
Gainesville Florida 32609-8922  
(352) 392.2061  
<[editor@cns.ufl.edu](mailto:editor@cns.ufl.edu)>

## Table of Contents

Preface .....	3
Introduction .....	3
Tape or Disk? .....	3
Storage Costs .....	4
Data Set Volumes and Names .....	5
How to Name Data Sets .....	5
Cataloged Data Sets .....	6
Creating Data Sets .....	7
Blocking Records .....	8
Allocating Space .....	9
Using the <i>UNIT</i> Parameter .....	11
Sample DD Statements for Data Sets .....	12
Unmovable Data Sets .....	12
Bulk Storage .....	13
Back Up, Delete, and Retrieve Data Sets .....	13
VSAM Data Sets .....	14
Utilities and Procedures .....	15
CNS Utilities .....	15
CNS Procedures .....	15
Utilities to Move and Reblock Data Sets .....	15
IBM Utilities .....	16

## Preface

This manual contains information about storing data on disks at CNS. We assume that you already know how to use CNS facilities and IBM JCL. Many documents in DOCWEB [<http://docweb.cns.ufl.edu/>] describe various utilities to use with OS data sets, but particularly see D0068, *Utilities for Partitioned Data Sets* [<http://docweb.cns.ufl.edu/docs/d0068/>], and D0071, *Utilities for z/OS ( OS/390) Disk Data Sets* [<http://docweb.cns.ufl.edu/docs/d0071/>] for more information on how to create, move, delete, or archive data sets. If you want to store data on magnetic tapes (cartridge or reel), refer to *Using Magnetic Tapes at CNS* (D0017) [<http://docweb.cns.ufl.edu/docs/d0017/>] and *Utilities for Using Magnetic Tapes* (D0069) [<http://docweb.cns.ufl.edu/docs/d0069/>].

This manual includes CNS-specific information such as CNS data set naming conventions and what is and is not supported at CNS. This manual is intended only as a supplement to vendor documentation and provides information about CNS-specific requirements. Refer to the following manuals for more detailed information about data sets, data set manipulation, and data set access methods:

- The latest version of the JCL text by Gary DeWard Brown
- *z/OS MVS JCL Reference by IBM*
- *z/OS MVS JCL User's Guide by IBM*
- *z/OS DFSMSdfp Utilities by IBM*
- *z/OS DFSMS Access Method Services for Catalogs by IBM*

These manuals are available on the Web via the CNS "BOOKSRV" server [<http://booksrv.cns.ufl.edu/>].

## Introduction

CNS uses IBM 3390 disk drives that are connected to an IBM z800 mainframe to store data sets online. We do not support passwords for disk or tape data sets or expiration dates for disk data sets. We actively discourage ISAM data sets since IBM support has been withdrawn. We do support VSAM data sets.

Various CNS documents discuss specific topics relating to data sets. These documents are in DOCWEB, CNS's online documentation system. Enter DOCWEB from the CNS home page on the Web at <http://www.cns.ufl.edu>.

Anyone can read, write, delete, or create data sets that do not belong to them unless those data sets are specifically protected. If you want your disk data sets to be secure from being read or written on by others, contact CNS at [consult@lists.ufl.edu](mailto:consult@lists.ufl.edu) or by telephone at 352.392-2061 to request special security arrangements.

## Tape or Disk?

Before you read the rest of this document for information about managing disk data sets, you might want to evaluate your computing needs to determine if disk storage is, in fact, the most efficient and economical medium on which to store your data. There are several factors to consider when making a choice between disk and tape storage. Convenience is one factor. For example, partitioned data sets can only be processed from disk. Interactive systems cannot access tapes. In situations where both disks and tapes are options, cost might be the deciding factor.

Wise use of disk space saves money and frees space that could be used for other data sets. Sometimes the wisest use of disk space is not to use it at all. Seldom-used sequential data sets should be kept on tape. They can be processed directly from the tape. Other types, such as partitioned data sets and VSAM data sets, can be off-loaded to tape for storage but must be reloaded to disk before they can be used. Appropriate utilities are available to do this, such as **IEBCOPY** for a PDS or **IDCAMS** for a VSAM data set. In this way, you can create an infrequently used disk data set only when you need it and then delete it when you are through with it, thus reducing your disk storage charges and releasing space for others to use. Of course, data sets that are no longer needed should always be deleted.

See the chapter in this manual on creating data sets for a detailed discussion on how to maximize use of disk space.

## Storage Costs

*The CNS Charging Algorithm* document (D0001) [<http://docweb.cns.ufl.edu/docs/d0001/>] lists the charges for storing OS data sets. Please refer to this document for complete information on charges for all CNS services.

## Estimating Storage Costs

To determine whether disk or tape is cheaper, you must compare the cost of storing your data on disk for the length of time your system or project will exist with the cost of having a tape mounted the number of times you will have to access your data.

The formula below may be used to give you a good estimate of the relative costs:

$(D \times S) / U$  is less than 12

where  $D$  is the number of days the system (or project) will exist,  $S$  is the size of the data (in megabytes), and  $U$  is the number of times the data will be accessed. If the result of the calculation is "true", that is, if  $D \times S / U$  is less than 12, disk storage is less expensive. If the result of the calculation is "false", tape storage is less expensive.

For example, suppose a 3-month project accesses the data twice a day and the size of the data is 5 megabytes. Thus,  $D = 90$ ,  $S = 5$ ,  $U = 180$ , and  $D \times S / U = 2.5$ , which is less than 12. Disk will be cheaper to use for this project.

As a second example, consider an ongoing project for which the number of days is indefinite. The formula can still be used by setting  $D$  to 1 and  $U$  to the number of times the data will be used per day. If the data will be accessed once per week, then  $U = 1/7$ . For a 3-megabyte data set to be accessed weekly, the formula yields 21, which is greater than 12. In this case, tape storage would be cheaper.

The value of 12 holds if all runs are made at normal priority and assumes that disk storage is on one of the public (USER9x) packs. If disk storage is to be on one of the bulk (BLK9xx) packs, use 56 instead of 12 in all calculations. To account for priority adjustments, multiply 12 (or 56) by the appropriate priority factor: 3 for urgent priority, 0.5 for low priority, and 0.25 for standby priority. Thus, for a data set on a public pack that will be accessed using only low priority, you should use  $12(0.5) = 6$ .

The value of 12 (or 56) also assumes that a tape is available at no charge for use during the life of a project. This is often not the case, and you might need to purchase a new tape. To incorporate this one-time cost, rewrite the formula as follows:

$(D \times S) / (U + 2C)$  is less than 12 (or 56)

where C is the cost of the tape, in dollars. As the length of the project increases, the cost of the tape will become an insignificant part of the cost of storage. For a 3 megabyte data set that will exist for 2 years and be accessed weekly, the formula calculation is as follows:

$730 \times 3 / (104 + 15) = 18.40$

assuming use of a tape which costs \$7.50. Since 18.40 is greater than 12, tape storage would be cheaper in this case. In addition, when the project ends, the tape may still have residual value.

## Data Set Volumes and Names

There are two classes of CNS-managed volumes: the public disk packs and the bulk-rate disk packs. The available public volumes are those in the USER9x series. The public volumes are, therefore, frequently referred to as the user volumes. The bulk-rate disk packs are the BLKxxx series. They are discussed later.

Every data set stored on CNS-managed disks must have a valid userid or access number associated with it for billing purposes. It is your responsibility to provide an appropriate userid or access number. Each of the CNS-managed disk volumes is searched daily for invalid data sets. Any new data sets that do not have valid userids or access numbers associated with them will be archived to tape and deleted by the Data Set Management System (DSMS) at the end of the day on which they are created.

## How to Name Data Sets

A data set name is made up of one or more simple names. Each simple name is made of alphanumeric or national (@, #, \$) characters and is 1-8 characters long. The first character must not be numeric. A *qualified* data set name is 2 or more simple names connected by single periods to a maximum length of 44 characters, including periods. Each qualifier is used as an index level when cataloging the data set name. A data set with an unqualified name, that is, a single simple name, is not permitted on a CNS-managed volume.

On the public volumes, all data set names must begin with a high-level index (qualifier) of U or UF, unless you have special permission to use another index. A data set with an improper high-level index cannot be cataloged, regardless of the volume it is on. The second qualifier of a data set name must specify either the userid or access number to which charges for the data set are billed.

## Specifying the Userid

The first option is to use your userid as the second qualifier in the data set name. Your userid implicitly specifies both an access number and a sequence number. The following examples show qualified names for the userid "STARTRK".

```
STARTRK.other.simple.names  
UF.STARTRK.other.simple.names
```

## Specifying An Access and Sequence Number

The second option is to explicitly code your 8-digit access number as the second qualifier in the data set name. Since the first character cannot be numeric, you must change the first digit of the access number to the corresponding letter of the alphabet, that is, you must change

- 1 to A
- 2 to B
- 3 to C

and so on. An explicit access number implies sequence number 1. You need do nothing further if sequence number 1 is intended. If your data set charges should be billed to a sequence number other than 1, include that sequence number, prefaced by the letter S, as the third qualifier in the data set name.

For example, a data set named UF.STARTRK.DATA would be charged to the access number and sequence number associated with userid STARTRK. A data set named UF.A0069401.PROGLIB would be charged to access number 10069401, sequence number 1. A data set named U.G0057314.S3.PLOT would be charged to access number 70057314, sequence number 3.

## Bulk-Rate Volumes

If you contract to use the bulk-rate volumes, you provide a list of data set name qualifiers which you intend to use and a single userid or access/sequence number to which the data sets will be charged. No other data sets are allowed on the bulk-rate volumes. Specifically, the naming conventions for data sets given above for the public volumes do not apply to the bulk-rate volumes.

## Cataloged Data Sets

Any data set that resides on CNS-managed disk volumes must have a valid name and must be cataloged. See the chapter on "Data Set Volumes and Names" for detailed data set naming conventions. If a data set name meets these rules and if the first level of qualification is one of the high-level indexes established by CNS, the data set can be cataloged.

A cataloged data set name is placed in a table of information, the catalog, so that the system can automatically locate the data set without a specific reference to volume or device type in

your JCL. In addition to simplifying your JCL, this also provides easier portability. The data set's location can be changed from tape to disk or vice versa without modification (except for **SETUP** statements). JCL can be device-independent since standard labeled tapes provide the same facilities for DCB specifications in the label that direct-access data sets get from the DSCB (Data Set Control Block).

You can create and catalog a data set using ISPF, JCL, IEHPROGM, or TSO commands. For example, to allocate and catalog a small load module library on disk, you could use the following JCL:

### Figure 1. Example of Allocating a Small Load Library

```
//jobname JOB (, ,time,lines), 'your name', CLASS=A
/*ROUTE PRINT node.location
//ALLOC EXEC PGM=IEFBR14
//MYLIB DD DSN=U.userid.LOADLIB, UNIT=SYSDA,
// SPACE=(6000, (12, 2, 3)), DISP=(NEW, CATLG),
// DCB=(DSORG=PO, LRECL=0, RECFM=U, BLKSIZE=13680)
```

You do not need to specify an explicit volume serial. The system will pick one with available space and then record it in the catalog. To refer to the data set thereafter, you will need to specify the data set name (**DSN=**) and disposition (**DISP=**). The catalog will contain information about the device and volume serial, and the DSCB will contain the data set attributes.

One example would be to execute a program that has been stored in the load library created above.

```
//RUN EXEC PGM=MYPROG
//STEPLIB DD DSN=U.userid.LOADLIB, DISP=SHR
```

With cataloged, qualified data sets, you can run most jobs without knowing exactly where the data set resides. (Certain IBM utilities require that explicit volume serial references be made, but these utilities usually have a CNS replacement that does not.)

Data sets on CNS-managed volumes that are not cataloged will be archived to tape and deleted from the disk pack, regardless of their name. Catalog entries for nonexistent data sets will be periodically removed from the catalog.

If you need help qualifying and cataloging your data sets, contact CNS via e-mail at <consult@lists.ufl.edu> or by telephone at 352.392-2061.

## Creating Data Sets

When creating a disk data set there are several factors which you need to consider. Each is controlled by an appropriate JCL parameter on the DD statement. Among these factors are:

- good blocking of the records in the data set (the *BLKSIZE* parameter),
- efficient allocation of space for the data set (the *SPACE* parameter),

- and proper specification of a device on which the data set is to reside (the *UNIT* parameter).

## Blocking Records

Programs deal with logical records, which are frequently quite short. For example, object modules, macro libraries, and most source programs have records which are 80 bytes long. Blocking is a technique where records are held in memory and not written to disk until several records can be transmitted together as a single physical record or block. This means that fewer input/output (I/O) operations are needed to process the data set. Since you pay for I/O counts, this can result in lower charges. Blocking records also allows the program to run faster since less time is spent waiting for the chance to transmit information to or from the disk.

Furthermore, blocking conserves storage space on the disk because having a smaller number of larger records reduces the number of interrecord gaps (IRGs) in the data set. The IRG has a fixed length regardless of the record length. With very small records, the gaps can account for 80% or more of the available space.

## Choosing a Block Size

You use the *BLKSIZE* keyword parameter on the **DD** statement to specify how large a block you wish to create. Some guidelines are needed for choosing a block size. The basic consideration is making efficient use of the track capacity of the disk. For example, if you block 80-byte card images at 27920 on a 3390 disk, you will get 2 blocks per track with an efficiency of around 99%. If you add one more card image to the block (increasing the block size to 28000), you will get only one block per track and less than 50% efficiency.

While track capacity needs to be considered, you should avoid block sizes which are tailored to a specific device. Even if you specifically allocate your data set on a particular device, it could be moved to a different device in the future. We recommend that you choose a block size from one of two ranges of block sizes that will give reasonable efficiency with either device:

between 9077 and 10796

or

between 11477 and 13682.

For fixed-length records (*RECFM=FB* or *FBS*), the block size must be an exact multiple of the logical record length. This means selecting the largest multiple of your logical record length that falls within one of the ranges. For example, 80-byte card images should be blocked at 10720 or 13680. For variable-length records (*RECFM=VB* or *VBS*) or unspecified-length records (*RECFM=U*), use the upper bound of one of the ranges, 10796 or 13682.

You need to be aware that main storage is used for buffers in which to process blocked records, usually five buffers per data set. This can affect the size of the region in which your job runs. The block sizes suggested above are intended to provide good track efficiency without excessive main storage use. They will result in from three to five blocks per track, depending on which range you select and on which device type the data set resides.

## Blocking Partitioned Data Sets

The preceding discussion, which applies specifically to sequential data sets, also applies to partitioned data sets with some additional considerations. A partitioned data set (PDS) contains many sequential data sets, called members, placed one after another in the data set.

Like any sequential data set, the last block of a member may vary in length, containing anywhere from one logical record to the maximum number of logical records the block can hold. If there is insufficient space on a track for a full block following the short block, the next member will be started on the next track. This wastes the remainder of the previous track, an amount equal, on the average, to half the block size.

If the PDS contains many small members, this waste can occur on almost every track. In this case, choose a smaller block size to reduce the amount of wasted space at the ends of the tracks. Use a block size that is approximately one-half the average member size but still within one of the two recommended ranges.

With a PDS that contains mostly large members that span several tracks each, the potential for waste is reduced and you can choose a block size as if it were a sequential data set.

For load module libraries (RECFM=U), use the upper bound of one of the ranges. The linkage editor will attempt to make the most efficient use it can of the space.

## Program Dependencies

Some programs may impose limitations on block size while others may make default assumptions. The linkage editor allows a maximum block size of 3200 for its primary input data set and for data sets containing object modules, although 3120 is a more efficient block size to use in these cases.

If a block size is not specified for a data set created by a Fortran program, the default value is 800. Since a block size of 800 has an efficiency of around 55%, you should avoid the default by providing a larger block size.

There is, in all cases, an upper limit of 32760 for block sizes, imposed not by hardware limitations but by the design of the operating system.

## Allocating Space

The *SPACE* keyword parameter on the **DD** statement is used to request amounts of space when allocating a disk data set:

```
SPACE=(unit,(primary,secondary,directory),RLSE)
```

## Unit of Allocation

Space may be requested in units of average block length, tracks, or cylinders. To be more device-independent you should specify an average block length, in bytes, rather than **TRK** or **CYL** in the "*unit*" parameter above.

## Primary Quantity

The primary quantity requests the initial number of units of allocation needed. This number

should be as close to the total requirement as possible.

For example, you can specify the following:

```
SPACE=( 12000 , 380 )
```

to request enough space to store 380 blocks of 12000 bytes each. The system will use this information and the physical characteristics of the device selected to compute an appropriate number of tracks or cylinders to allocate.

If you are unsure exactly how much space you need, it is better to slightly overestimate your allocation than to underestimate it and cause secondary allocation. Releasing any overallocated space is discussed below (see "Releasing Unneeded Space").

## Secondary Quantity

The secondary quantity is a safeguard that will be used only if your data set exceeds the space that you allocated with the primary quantity. The system does not guarantee that the space for a secondary allocation will be available.

Generally, you should specify a secondary allocation quantity to avoid wasteful ABENDs in the event that the initial size estimate was insufficient. For example,

```
SPACE=( 12000 , ( 380 , 38 ) )
```

allows up to 15 additional 38-block extents of space to be added to the original allocation if it is needed and if space is available.

## Directory Blocks

When you create a partitioned data set (PDS), you must specify the number of directory blocks you will need. The directory holds the names of the members of the data set, along with other needed information. If you do not allocate enough directory blocks, you might restrict the growth of the PDS and have to re-create it later to enlarge the directory. Therefore, it might be wise to anticipate growth by allocating 10-15 percent more directory blocks than required.

The operating system uses the directory of a PDS to locate its members. Each member has a unique entry in the directory. The size of the entry varies depending on what kind of data set is involved and how the data set has been maintained.

Directory entries for members of a library containing source code, macros, object modules or data can require the least space because only the name of the member and its relative location in the data set need to be stored. However, if one of these data sets has been maintained by the ISPF editor with the STATS option turned on, additional space is needed to hold the requested statistics.

Directory entries for load modules created by the linkage editor also require more space because the system must include additional information for use at execution time (such as the size of the module and its entry point address.)

As a guideline for determining the number of directory blocks you will need for a PDS, figure on 6 entries per directory block if the data set is a load module library or if it will be edited

with the ISPF editor. Otherwise, allow for 16 entries per directory block.

To specify the number of directory blocks, use the directory allocation on the space parameter. For example, to specify five directory blocks, you could use:

```
SPACE=( 12000 , ( 380 , 38 , 5 ) )
```

To check the status of partitioned data sets, use the **PDSL** utility program described in CNS document D0068, *Utilities for Partitioned Data Sets* [<http://docweb.cns.ufl.edu/docs/d0068/>].

## Releasing Unneeded Space

Since data sets which have unneeded space allocated to them cost you unnecessary storage charges--in addition to tying up space which could be used productively by others--you should release unused space back to the system. One common situation where you want to release unused space is when you have intentionally overallocated a new data set because you were unsure exactly how much space was needed. A second common problem is a PDS which has waste space in it due to the manner in which a PDS grows.

### The RLSE Subparameter

If you use the *RLSE* subparameter of the *SPACE* parameter in the job step in which you create a data set, the system will release any allocated space which is still unused at the end of the job step. Thus, in the example used earlier, if you included the *RLSE* subparameter as follows:

```
SPACE=( 12000 , ( 380 , 38 ) , RLSE )
```

and the data only required 340 blocks rather than the 380 blocks requested, the unused 40 blocks would be released when the job step ended.

### Reclaiming Space in a PDS

Whenever a member of a partitioned data set (PDS) is replaced, the old space it occupied is not reused or released. In time, a PDS can become fragmented by blocks of wasted space and the data set will require additional extents as new members are added or old members replaced. Besides increasing storage costs, this burdens the system with extra operating overhead. Utility programs and procedures and TSO commands are available that can compress partitioned data sets to recover wasted space.

The TSO **RLSE** command allows you to release unused storage space after the data set has been compressed. For more information on the **RLSE** command, sign on to TSO and enter

```
HELP RLSE
```

It is your responsibility to maintain your data sets.

## Using the *UNIT* Parameter

Specify *UNIT=SYSDA* for all new disk data sets. This will make the data sets device-independent.

When allocating new non-VSAM data sets, do not specify the volume serial. The system will automatically place the data set on the correct volume. A specific device type should only be requested when that particular device type is required by a program. For instance, when using the **IEHPROGM** utility to catalog a data set, the *UNIT* parameter on the **CATLG** statement must be specified as `3390` rather than `SYSDA`.

## Sample DD Statements for Data Sets

The following examples show how to use the **DD** statement in JCL to specify a data set.

### Creating a New Sequential Data Set

The following sample DD statement shows how to create a sequential data set on one of the user public volumes. It will be billed to userid "STARTRK."

```
//MYDATA DD DSN=UF.STARTRK.PHASER.DATA,  
//          UNIT=SYSDA,  
//          SPACE=(13680,(50,5),RLSE),  
//          DCB=(LRECL=80,RECFM=FB,BLKSIZE=13680),  
//          DISP=(NEW,CATLG)
```

### Creating a Load Module Library

This example shows how to create a load module library. It will be stored on a bulk disk volume for which the user has indicated that "STRFLEET" is the high-level index to be used on the bulk volume.

```
//MYLOAD DD DSN=STRFLEET.KLINGON.LOAD,  
//          UNIT=SYSDA,  
//          SPACE=(13682,(300,30,10),RLSE),  
//          DCB=(RECFM=U,DSORG=PO,BLKSIZE=13682,LRECL=0),  
//          DISP=(NEW,CATLG)
```

## Unmovable Data Sets

---

### Note

Unmovable data sets are strongly discouraged at CNS.

---

An unmovable data set is one that contains location-dependent information, such as records chained together by absolute track addresses. It might be impossible to recover data from such a data set if there is a disk input/output (I/O) error or if the data set is archived by DSMS. Such unmovable data sets cannot be moved from one volume to another unless they occupy the same relative location on the new volume.

Since movable data sets can be moved from one volume to another for systems maintenance purposes, it is imperative that anyone with unmovable data sets ensure that they are properly marked by specifying a `U` in the data set-organization parameters (*DSORG*) when the data sets are created. If this is not done, the data sets can be copied to different relative locations and become unusable.

For example, code `DSORG=PSU` for an unmoveable physical sequential data set. Code

*DSORG=POU* for an unmoveable partitioned data set.

If you need help converting your data sets from unmovable to movable, contact CNS via e-mail at <[consult@lists.ufl.edu](mailto:consult@lists.ufl.edu)> or by telephone at 352.392-2061.

## Bulk Storage

Bulk storage is available to anyone willing to contract with CNS for a minimum of 300 megabytes (300 million bytes) per month of disk space. Bulk storage rates are considerably less expensive than regular disk storage rates. You must indicate in advance the number of megabytes of storage you want to use, the high-level indexes of data sets you want to allocate on the bulk volumes, and the access number and sequence number to bill for bulk-storage use. If you use less than the contracted amount, you will be billed for the contracted minimum at the current bulk storage rate. See the CNS "*Charging Algorithm*" document (D0001) [<http://docweb.cns.ufl.edu/docs/d0001/>] for the current bulk storage charge. You will be allowed to exceed your contracted amount by up to 10% without penalty; you will simply be billed for the overrun at the bulk storage rate. If you exceed 110% of your contracted amount (on bulk volumes only) you will be notified to either reduce your storage to within your contracted amount or to change your contracted amount. Bulk storage is administered independently from whatever regular disk storage you may have.

Data set placement can be done automatically for non-specific allocations or manually by using specific allocations. Non-VSAM data sets using an index identified by a bulk storage contract can be allocated automatically on a bulk volume by specifying UNIT=SYSDA in your JCL, with no VOLSER specified. You must still specify a VOLSER when allocating VSAM data sets because of IDCAMS (Access Method Services) requirements. (Bulk storage volumes are designated BLKxxx.) If you are currently using USER9x volumes, you should consider bulk storage as an alternative. Include your own data set backups in your planning because the bulk volumes are not backed up by CNS as are USER9x volumes.

Anyone who wants to contract with CNS for bulk storage must make requests in writing by the fifteenth of the month (earlier for massive amounts) for the following month. For more details and bulk storage request forms, contact CNS Systems at (352) 392-2061.

## Back Up, Delete, and Retrieve Data Sets

You are responsible for maintaining an up-to-date backup of your data. CNS can restore data (copy data from a backup source), but cannot recreate or reconstruct data if no backup tape exists, even if the loss is caused by faulty equipment or mishandling. Disk drives malfunction occasionally. Therefore, backup tapes should be maintained for any data that are hard to replace, and are essential for data that result from involved research or experiments, for data from outdated or very old files, or for data that cannot be recreated.

A backup copy of disk data to a tape is preferred. Two tape backups are recommended for data that are difficult or impossible to recreate. A computer printout can also serve as a backup, but if the data on disk are destroyed, both time and money will be required to have the information reentered. The **DISKTAPE** cataloged procedure described in CNS document D0069, *Utilities For Using Magnetic Tapes* [<http://docweb.cns.ufl.edu/docs/d0069/>] can be used to copy a disk data set onto tape.

Any stored files, data sets, workspaces, or documents that are stored under an access number that you ask CNS Accounting to delete will automatically be deleted with the access number.

If you get a new access number, all files associated with the original access number will be deleted from the system if the old access number is deleted. It is your responsibility to ensure that all files are transferred to the new access number or that copies are made of those files that you want to keep. To retrieve a data set that has been archived by the Data Set Management System, contact the CNS Support Desk at 352/392-2061. There is a \$25 charge for retrieving a data set. Archived data sets should be retrieved within 60 days or the data may be lost. See the *CNS "Charging Algorithm"* document (D0001) [<http://docweb.cns.ufl.edu/docs/d0001/>] for current CNS charges.

## VSAM Data Sets

The Virtual Storage Access Method (VSAM) allows you to organize data in several ways:

- as an entry-sequenced data set (similar to a sequential data set)
- as a relative record data set (like a direct access data set)
- as a key-sequenced data set (like an indexed-sequential data set).

All VSAM data sets must be stored on disk and must be named according to standard CNS data set naming conventions. VSAM data sets must be allocated with fully specified data set names. This means that the cluster, as well as the data and index component names, must be specified. We suggest that the data and index component names be derived from the cluster name by adding `.DATA` or `.INDEX` to the end of the cluster name as shown in the following example:

```
DEFINE                                     -
  CLUSTER ( NAME(UF.A2345678.S5.EXAMPLE)   -
            cluster attributes             ) -
  DATA   ( NAME(UF.A2345678.S5.EXAMPLE.DATA) -
            data attributes                 ) -
  INDEX   ( NAME(UF.A2345678.S5.EXAMPLE.INDEX) -
            index attributes                )
```

The utility program **IDCAMS** (Access Method Services), which is used to define and support VSAM data sets, is more than just a VSAM utility. It can also be very useful in working with Physical Sequential (PS) and Indexed Sequential (ISAM) data sets. Two useful **IDCAMS** commands are **PRINT** and **REPRO**. For more information on access method services, see the IBM publication *z/OS DFSMS Access Method Services for Catalogs*.

To access **IDCAMS**, use the following JCL:

```
//jobname JOB (, ,time,line), 'your name', CLASS=A
/*ROUTE PRINT node.location
// EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
```

If you use the `INFILE` or `OUTFILE` parameters, include **DD** statements pointing to your data sets. If you use `INDATASET` or `OUTDATASET`, **IDCAMS** will try to allocate the data sets dynamically.

# Utilities and Procedures

## CNS Utilities

The following CNS utility programs and procedures can be used to manipulate data sets. These are documented in various DOCWEB documents, which you can find by searching on the term for the utility you need.

<b>DELETE</b> in TSO or IDCAMS	delete and uncatalog a data set.
<b>DSAT</b> in TSO	list information about datasets for a selected high-level index.
<b>IEHPROGM</b>	allocate, scratch, create aliases, rename, catalog, uncatalog.
<b>LISTCTLG</b>	prints a list of the data sets that have been cataloged under a specific data set index, for example, UF.
<b>PDSLIST</b>	print the contents of a PDS and generate a report that gives the number of directory blocks allocated, number of blocks used, total main members, total alias members, and the average number of entries per active directory block.
<b>REPRO</b> Command of IDCAMS	copy sequential and VSAM data sets. The output file can differ from the input file in record format, block size, and data set organization. Optional record selection is provided.
<b>SRCHFOR</b> Command of <b>SUPERC</b> in TSO <b>ISPF</b>	search data sets on disk for occurrences of specific text strings.
<b>SUPERC</b> in TSO <b>ISPF</b>	compare the directories of two partitioned data sets (PDSs).

## CNS Procedures

<b>CARDDISK</b>	create a disk data set from card images or add card images to the end of an existing disk data set.
<b>CREATLIB</b>	allocate an empty partitioned data set.
<b>DISKTAPE</b>	copy a disk data set to an output tape
<b>PANMOVE</b>	move cataloged Panvalet libraries.
<b>SQUEEZE</b>	compress PDSs. Recopies the data set so that space left by deleted members can be reused.
<b>TAPEDISK</b>	copy a tape data set to a disk data set or append a tape data set to the end of an existing disk data set.

## Utilities to Move and Reblock Data Sets

Before you can move or reblock data sets, you must know the data set organization. To do this, use the TSO **DSAT** command (type **HELP DSAT** under TSO). Data set organization will be listed in the output under DSORG. If you need help determining your data set's organization, or if you need assistance in moving your data sets, contact the CNS Support Desk (392-2061).

To display information about your datasets, use the following JCL:

```
//jobname JOB (,,time,line),'yourname',CLASS=A
/*ROUTE PRINT node.location
// EXEC TSO
//SYSIN DD *
DSAT 'your.index' LASTREF SECONDARY
```

The table below shows utilities to move your data set.

## Utilities to Move Data Sets

---

### Note

There are no utilities for moving unmovable or device-dependent data sets. You will have to create your own programs to move these data sets. CNS does not support or recommend unmovable data sets.

---

Organization	Utility
AM (VSAM)	IDCAMS
DA (Direct Access -- BDAM )	IEBGENER (IEBGENER should only be used for non-keyed BDAM data sets)
MARKIV	MARKIV programs
Panvalet	PANMOVE
PO (Partitioned or BPAM)	IEBCOPY
PS (Physical Sequential -- BSAM or QSAM)	IEBGENER IEHMOVE
SAS (Statistical Analysis System)	PROC COPY of SAS

## IBM Utilities

In addition to the CNS utilities listed in a previous section, the following IBM system utilities are available for working with data sets. These utilities are documented in the IBM manual *z/OS DFSMSdfp Utilities*.

**Table 1. IBM Utilities**

Utility	Data Sets	Members or Records	Other
	PDS=Partitioned PS=Physical Sequential		Volume, Test Data, Generation Data Group (GDG), Index
<b>IEHLIST</b>	List PDS Directory		List VTOC List Catalog

<b>Utility</b>	<b>Data Sets</b>	<b>Members or Records</b>	<b>Other</b>
<b>IEHMOVE</b>	Copy PDS, PS Load PDS, PS Merge PDS	Copy selected members; Exclude members from a copy operation; Rename moved or copied members; Replace selected members in a move or copy operation	Copy volume; Copy or move data set group
<b>IEHPROGM</b>	Uncatalog Scratch Catalog; Rename PDS, PS	Rename Scratch	Catalog a GDG; Build a GDG; Delete an index; Build index
<b>IEBCOMPR</b>	Compare PDS, PS		
<b>IEBCOPY</b>	Compress PDS in place; Copy PDS; Expand PDS; List PDS number of unused directory blocks; Merge PDS; Reblock PDS; Unload, Restore PDS	Copy or replace selected members; Exclude members from copy operation; Replace members; Replace selected members in move or copy; Rename members when copied	
<b>IEBDG</b>	Create PS output	Copy members; Create members	Generate test data
<b>IEBGENER</b>	Convert PS to PDS; Copy PS; Edit PS; Expand PS; Print PS; Reblock PS	Copy members Change record length	
<b>IEBPTPCH</b>	Print PDS, PS; Edit PS; Print records	Punch members or records	
<b>IEBUPDTE</b>	Alter organization; Create PDS; Modify PDS, PS; Update PDS in place; Convert PDS to PS or PS to PDS; Copy PS; Edit PS; Update PS; Print PS; Reblock PS	Copy member; Update member; Replace member; Renumber record; Replace record; Number records; Print members; Print records	Enter a procedure into a procedure library
<b>IDCAMS</b>	Physical Sequential (PS) and VSAM	Print all or selected records; Copy all or selected records	Define, support VSAM data sets

## Your Comments are Welcome

We welcome your comments and suggestions on this and all CNS documentation. Please send your comments to:

**CNS Information Services**

2124 NE Waldo Rd, Suite 2100  
Gainesville Florida 32609-8922  
(352) 392.2061  
<editor@cns.ufl.edu>

## **UF Computing & Networking Services**

### **CNS Information Services**

2124 NE Waldo Rd, Suite 2100  
Gainesville Florida 32609-8922  
(352) 392.2061  
<editor@cns.ufl.edu>