CA EPIC™ for z/VSE

User Guide

r5.2

CA Product References

This document references the following CA products:

- CA EPIC™ for z/VSE
- CA CIS for z/VSE
- CA SORT® for VSE

Contact CA Technologies

Contact Technical Support

For your convenience, CA Technologies provides one site where you can access the information you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Provide Feedback

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

If you would like to provide feedback about CA Technologies product documentation, complete our short customer survey, which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 11: Job Activity Reports 329

Chapter 12: Managing Tape Volumes 347

Chapter 1: Introduction

This section contains the following topics:

CA EPIC for z/VSE Fundamentals

This chapter contains an overview of features and functions provided by CA EPIC for z/VSE™. This manual is written to include the novice mainframe programmer. If you have more experience in the field, you already are familiar with some of the subjects discussed and you may want to skim those sections. However, most of the chapter presents information unique to CA EPIC for z/VSE, so we strongly encourage everyone to read the chapter for a full introduction.

Configuration Options

CA EPIC for z/VSE provides configuration options which allow you to customize many aspects of your processing.

Many configuration options are discussed in this manual. Please consult your systems programmer or the *Installation and System Guide* if you have a question about a specific configuration option.

Controlled Datasets

CA EPIC for z/VSE controls datasets best when they are known to its *Dataset Name (DSN) Catalog*. The information contained in the records that make up a DSN Catalog entry determines how a dataset is processed.

With DSN Catalog control of a dataset, CA EPIC for z/VSE specifies the following:

- where each existing version is written and where new versions can be written
- how each version is written (allocation size, block size, density, multi-dataset, and so on)
- how long each version is retained and how many versions are kept at one time
- processing statistics for each version (creation date, last access date, block count)
- each tape version's vault location

Start Track 1 Datasets

A *Start Track 1 dataset* is a disk dataset that is defined by the use of the number "1" in the relative track field of the VSE EXTENT statement. For example:

```
// DLBL MYFILE,'ANY.FILE'
// EXTENT SYS005,SYSWK3,1,0,1,20
```

The use of the number "1" in the relative track field above indicates that CA EPIC for z/VSE will allocate the dataset. The use of any other number in this field indicates that CA EPIC for z/VSE does not handle the allocation.

Start Track 1 datasets are only partially controlled. CA EPIC for z/VSE allocates disk space for the dataset on both input and output. However, you supply the name of disk pool to be used and the number of tracks or blocks to allocate for the dataset. All information is supplied by JCL.

Start Track 1 datasets provide one benefit: the dataset does not have to be entered in the DSN Catalog. However, they are less efficient than DSN Catalog datasets and they provide limited control over retention.

Start Track 1 datasets also require more I/O operations when they are opened. Because a Start Track 1 dataset's extents are not recorded in the catalog, CA EPIC for z/VSE must find the dataset's extents from the VTOC. In addition, all controlled datasets are written in pools, and pools can contain more than one volume. CA EPIC for z/VSE must read the VTOC of every volume in the pool until it finds the dataset's extents. This is true for both input and output. CA EPIC for z/VSE deletes any previous version of a Start Track 1 dataset before creating a new one. CA EPIC for z/VSE must search on output to locate and delete the old version.

In contrast, CA EPIC for z/VSE only needs to read a few records in the DSN Catalog to locate each extent of a cataloged dataset, no matter where it is written.

Start Track 1 datasets are retained and scratched like uncontrolled datasets. CA EPIC for z/VSE does not protect them from being scratched, and only one version can exist at a time. To retain the dataset using JCL, specify an expiration date or the number of days. If you do not specify an expiration date or Days Retention, CA EPIC for z/VSE provides a default retention of seven days.

Because of the limitations on retention, Start Track 1 datasets are useful only for work datasets. Because of the overhead required for these datasets, even a work dataset in a frequently run job should be cataloged to improve performance.

Configuration Options

Start Track 1 datasets are affected by the AUTOCAT, CATRT1, MINRET, NCATDSK, and STRTRK configuration options. In addition, they can be affected by other configuration options that apply to disk. See the *Installation and System Guide* for further information.

Controlling Different Types of Datasets

CA EPIC for z/VSE can control SAM, ISAM, and DA datasets.

If the VSAM configuration option is set to *YES*, CA EPIC for z/VSE logs VSAM extent information in the DSN Catalog. Logging occurs each time a VSAM dataset is opened. CA EPIC for z/VSE always reflects the currently used extents, even if the dataset contains multiple secondary allocations.

CA EPIC for z/VSE can only control TYPEFLE=WORK files on disk. It cannot control them on tape.

CA EPIC for z/VSE does not control datasets processed with the DTF names IJSYSNR, IJSYSPC, or xJSYSRS, even if the dataset name is cataloged. The system programmer can exclude more DTF names from CA EPIC for z/VSE control by adding the DTF name to the ILIST parameter of the TSIDMAP macro.

Configuration Options

The CNTRL, UNCDSK, UNCTAP, and UNCTL configuration options affect the way uncontrolled datasets are processed. See the *Installation and System Guide* for details about the TSIDMAP macro.

DSN Catalog Entries

CA EPIC for z/VSE's control of datasets is based on information contained in the DSN Catalog. All information pertaining to one cataloged dataset name is called a DSN Catalog entry. The format and contents of an entry are described below.

DSN Catalog information is divided into two levels: the CA EPIC for z/VSE dataset definition (EDD) level and the version level.

Each EDD contains all the attributes associated with a unique dataset name. When you process a controlled dataset, CA EPIC for z/VSE uses these attributes to control processing. The EDD is discussed later in this chapter.

CA EPIC for z/VSE enters version information automatically as each version is created. At least one catalog record is created for each version of the dataset. If the version spans more than one volume or extent, a catalog record is created for each additional volume or extent.

Version information include the version's location, its expiration date, creation date, and so on. CA EPIC for z/VSE uses this data to control the use and retention of each version individually.

The following sections give a detailed explanation of the data contained in each DSN Catalog entry, and how the various fields operate in conjunction with each other.

DSN Catalog Management

Most updates to the DSN Catalog are made automatically as datasets are processed. However, sometimes you want to add a special dataset or change the defaults you have supplied for an existing EDD. This *manual cataloging* can be done using either the Online Manager or the TSIDMNT utility.

The CA EPIC for z/VSE Online Manager (EOM) is a panel-driven catalog management program. It leads you step-by-step through the catalog management process and allows you to view the modifications you have made to a dataset entry. The TSIDMNT program provides the same functions in batch mode.

Please refer to Chapters 2 and 3 for more information about what these programs can do.

DSN Catalog Maintenance

The DSN Catalog must be maintained regularly. Maintenance includes regular backups and scratch runs, as well as periodic re-indexing.

Catalog maintenance is done using the DSN Catalog maintenance utility (TSIDUTL). See the *Installation and System Guide* for information about required maintenance.

CA EPIC for z/VSE Dataset Definitions (EDDs)

CA EPIC for z/VSE uses the dataset name in the DSN Catalog to identify a dataset as controlled. Each time a controlled dataset is opened, CA EPIC for z/VSE participates in processing.

Dataset Name

The rules for dataset naming are:

- Dataset names can be up to 44 characters long, except for generation datasets (see below), whose names are limited to 35 characters.

Special considerations apply to tape datasets that belong to a multi-dataset group. See "Prime Dataset Name" under "EDD Tape Information" in this chapter.

Comment

You can record one 30-character comment with any CA EPIC for z/VSE dataset definition. Comments provide additional documentation for the dataset and are printed in CA EPIC for z/VSE reports.

Passwords

Passwords provide protection against unauthorized use of RELEASE and UNLOCK on this dataset and its versions. RELEASE and UNLOCK are described in Chapters 2 and 3. Entering a password does not lock, release, or unlock the dataset.

Encryption Key

Software data encryption encodes tape datasets as they are written. They are decoded only when they are read under the control of the same DSN Catalog that was used to write them. Data encryption can be used to prevent tape datasets from being read at external locations. It provides no protection against unauthorized internal use.

Entering an encryption key for a dataset invokes data encryption for each tape version created. The encryption key is defined with the ‘DEC=’ parameter of the TSIDMNT ‘CATALOG’ or ‘UPDATE’ commands, or through the EPIC Online Manager. You do not have to remember this as you would a password.

The encryption key and the dataset password (above) function independently of each other. You can use both if you wish. The software encryption key is not related to the Key Encryption Key Label (KEKL) that is used for Hardware data encryption.

Important! Encrypted dataset versions must not be deleted from the DSN Catalog. Once a version is deleted, it cannot be decoded and is unreadable. Furthermore the encryption key cannot be added, changed, or deleted to a dataset that has existing versions.

KMODEL

Hardware data encryption is supported by defining a KEKL Model Definition (KMODEL) that will supply encryption key labels for the DSN should it be written to a 3592 tape drive with an encrypted density. The KMODEL name must be 8 characters or less and must already have been defined in the CA EPIC for z/VSE catalog (with the TSIDMNT 'KMODEL' command), or this parameter will be rejected.

User ID and System ID

System ID and User ID assist in grouping tape datasets for reporting and scratching purposes. Each can contain any two-character code that is significant to you and your installation. There is no difference in function between them. You can use both together to group datasets at two levels. Either one or both can be used for any dataset.

SYS Number

You can cause a specific SYS number to be used for a dataset by entering that SYS number in the EDD. This inhibits LUB allocation (described in Chapter 7) for the dataset each time it is processed. We recommend that you avoid assigning a SYS number here unless it is absolutely necessary. It is better to allow LUB allocation to assign a SYS number for you, or to inhibit LUB allocation through JCL. If you do assign a SYS number in the EDD, ensure that you do not use the same number in your JCL for a different dataset in the same job step.

Block Size

Block size reblocks datasets. It overrides any block size coded in the program's DTF. If the block size specified is not a multiple of the record length, CA EPIC for z/VSE automatically rounds down to the closest multiple.

Reblocking is valid only for sequential datasets with fixed or variable record formats, and the program DTF must already specify a blocked format. It is supported for all datasets processed by logical IOCS, but may be ignored by programs that use physical IOCS to process the dataset. It is not valid for tape datasets which are read backward.

Block size and CISIZE are mutually exclusive; you can only specify one of these in an CA EPIC for z/VSE dataset definition.

Generation or Work Dataset

CA EPIC for z/VSE provides four retention criteria: *generation or work* (described here), and *retention days*, *cycles retention*, and *DLA retention* (described below). All of them work together to determine retention for active versions.

The 'generation or work' field specifies whether to keep more than one version of the dataset at a time. When you designate a dataset as a generation dataset, CA EPIC for z/VSE maintains *at least* the number of versions specified in the cycles retention field. Because each version is also retained as long as the retention days and DLA retention fields dictate, CA EPIC for z/VSE can retain more versions than the number specified with cycles retention.

When you designate a dataset as a non-generation (or work) dataset, CA EPIC for z/VSE retains only one version of the dataset at a time. As long as no new version is created, that version is retained as long as the retention days and DLA retention dictate. So, for disk versions, when you create a new version of a non-generation dataset, the old version is scratched automatically, whether or not the old version has expired. In addition, entries are automatically scratched from the VTOC to avoid the "OVERLAP ON UNEXPIRED FILE" error generated by VSE. For tape versions, the old version is scratched when you execute the SCRATCH function of the TSIDUTL program.

When you specify a cycles retention of one (1) for a non-generation dataset, CA EPIC for z/VSE always retains one version, even if its other retention criteria have expired.

Retention Days

The retention days field specifies the number of days each version should be retained after its creation date.

You can use the retention days field to specify permanent retention (PERM). When a disk version is created with permanent retention, an expiration date of 2099366 is recorded in the VTOC entry. When CA EPIC for z/VSE sees the 2099366 expiration date, it does not allow the operator to display or delete the dataset in response to an INSUFFICIENT SPACE condition.

Cycles Retention

The cycles retention field specifies the minimum number of versions (cycles) to maintain at all times. To maintain more than one version, you must also designate the dataset as a generation dataset (see above).

Days after Last Access (DLA) Retention

DLA retention relates expiration to dataset use. Even if other retention criteria have expired, the dataset cannot expire if it has been processed as input during the last *n* days.

EDD Tape Information

Standard Label

The label field specifies whether an output tape dataset is to be created on a labeled or unlabeled tape.

Tape Density

The tape density specifies which density to use when creating a tape dataset. When a density is specified, CA EPIC for z/VSE only scans drives that support the specified density when it looks for a valid scratch tape. For example, you can use this field to assign a dataset to cartridge tapes only.

A default density for your site can be specified using the MODE configuration option. Specifying a tape density in the EDD overrides the MODE default. Specifying a density in the JCL overrides both the MODE default and the EDD tape density. If no density is specified in any of these places, CA EPIC for z/VSE uses the last density that was assigned to the selected drive.

Prime Dataset Name

The prime dataset name links all datasets in a multi-dataset group. The link is accomplished in the DSN Catalog records, and it allows CA EPIC for z/VSE to group the datasets automatically. The *prime dataset* is the first dataset on the tape. The second and subsequent datasets on the tape are called *subdatasets*.

Multi-dataset processing can be complex, and there are pre-processing tasks required before a multi-dataset group can be created. Multi-dataset processing is described in detail in Chapter 7.

Tape Pool ID

The tape pool ID field associates a dataset with a particular tape pool. Once a dataset is assigned to a tape pool, it can only be written on scratch tapes from that pool. Datasets not assigned to a tape pool can only be written on scratch tapes that are not assigned to any tape pool. If a default tape pool is defined for an installation, that pool ID will be associated with any dataset that does not have a tape pool defined.

Owner ID

The owner ID is a secondary pooling mechanism for tape datasets. An owner ID is a two-character code that matches the first two characters of a tape's volume serial number. Owner ID is provided primarily for compatibility with older tape management systems. If an owner ID is specified for a dataset, all new tape versions must be created on tape volumes which begin with those two characters.

Revolving Tape Datasets

Specifying Revolving Tapes=YES for a dataset establishes an exclusive relationship between the dataset and the tapes it uses. A revolving tape volume can only contain a version of a specific dataset, and a revolving tape dataset can only be written on specific tape volumes. A revolving tape volume cannot be used in an Automated Cartridge Library (ACL).

Vault Method Number

A vault method establishes a pattern of movement through tape vaults for each tape volume in the dataset. The vault method number field assigns a vault method and activates vaulting for the dataset.

EDD Disk Information

CISIZE

The CISIZE field specifies a CISIZE for FBA devices and forces the dataset to be written to maximize the number of records written to a control interval. Use CISIZE instead of block size for datasets to be written on FBA devices.

Block size and CISIZE are mutually exclusive; you can only specify one of these in an CA EPIC for z/VSE dataset definition.

Disk Pool Name

The disk pool name field is used to assign disk datasets to a disk pool. Disk pools are created by the systems programmer. They are referenced by a six-character pool name which can be a real volume serial number or an assigned name.

The systems programmer can establish CPU or partition-independent disk pools. To specify CPU and partition-independent pools, use three dashes at the end of the pool name (for example, POL - - -). For partition-independent pools, use two dashes at the end of the name (for example, POOL - -).

If you do not assign a pool, and a default pool has been specified by the DEFPOL configuration option, the dataset will be assigned to the default pool.

Primary Extent Size

The primary extent size field defines the number of tracks or blocks to be allocated to the first extent. The minimum size is 1 CKD track or your CISIZE for FBA.

Secondary Extent Size

Secondary extents are allocated automatically to a version of a sequential disk dataset when the primary extent is filled. Secondary extents are only supported for programs that use logical IOCS and programs that use physical IOCS programs if they support secondary allocations.

The secondary extent size field defines the number of tracks or blocks to be allocated to every extent after the first. The default secondary extent size is 50% of the size of the primary extent.

Number of Secondary Allocations

You can specify the maximum number of secondary extents allowable for a single dataset version. The maximum is 255.

Allocation by Logical Records

Allocation by logical records (ALR) is used to calculate the size of the primary extent automatically. It overrides the primary extent size field. CA EPIC for z/VSE calculates the extent size based on:

- record length
- estimated number of records
- block size
- track/block capacity of the selected volume

ALR allows you to migrate from one device type to another without changing the allocation size.

ALR is valid only for sequential disk datasets created with a DTFSD.

Version Information

As each version of a dataset is created, CA EPIC for z/VSE automatically records attributes specific to that version in the DSN Catalog. The attributes recorded at this level are described in this section.

Version Number

The current (most recently created) version is always version 1. Whenever a new version of a dataset is created, previous versions are incremented automatically. Versions in scratch, open, or conditionally cataloged status (see below) do not have version numbers.

Generation Number

A generation number is assigned to each generation disk version. It is a nine-character suffix that is appended to the dataset name (in the VTOC only) as ".G=nnnnnn", beginning in position 36 of the dataset name.

Since several versions of a generation dataset can be written on the same disk volume, CA EPIC for z/VSE appends a unique generation number to the dataset name to ensure that the dataset name in the VTOC is unique.

Although the VSE VTOC name of the dataset includes the generation number, CA EPIC for z/VSE does not consider the generation number to be part of the dataset name. Therefore, do not include it in your JCL when you want CA EPIC for z/VSE to process the dataset under DSN Catalog control. Specify the dataset name only as it appears in the CA EPIC for z/VSE dataset definition.

Version Status

CA EPIC for z/VSE tracks the status of each version. CA EPIC for z/VSE status types are:

Active (A) is the default status of an unlocked active version.

Open (O) indicates that close processing did not complete successfully. An open version cannot be used or scratched until you declare it closed using the Online Manager or TSIDMNT program.

Conditional Catalog (C) indicates that the version was created with the TESTJOB feature or a conditional catalog option in the JCL. It is accessible as input only if it is the latest in the dataset's chain of versions. In addition, it can only be accessed by a job with the same name as the job that created it. To make it an active version, declare it closed it using Online Manager or the TSIDMNT utility.

Open and conditionally cataloged versions have different rules for retention from active versions. See "Rules for Dataset Retention" in this chapter for details.

Scratch (S) denotes a tape version in scratch status. The tape can be overwritten at any time. Disk version records are deleted, not marked as scratch.

Lock (L) denotes a version in locked status. It cannot be accessed as input until it is unlocked or released using Online Manager or the TSIDMNT utility.

Subdataset (N) denotes a dataset in a multi-dataset group with file sequence number 2 or greater.

Work (W) indicates a non-generation dataset.

Unlabeled (U) indicates that the version was created on an unlabeled tape.

MCAT (M) indicates that the version has been MCATALOGed. It can be an MCAT master or alias version.

Offload status indicates that a disk version has been moved to tape with CA EPIC for z/VSE's offload system. The offload system is described in detail in Chapter 13.

Volume Serial Number

The volume serial number field indicates the volume serial number of the tape or disk volume on which the version is written. For multi-volume datasets, there is one volume serial number recorded for each volume.

Reel Number

The reel number indicates the tape volume sequence number for multi-volume datasets.

File Sequence Number

The file sequence number field indicates the position of the dataset in a multi-dataset group.

Extent Number and Address

CA EPIC for z/VSE records the extent addresses used for each disk dataset. For each extent it records:

- the extent sequence (1 for primary, 2 for the first secondary, and so on)
- the beginning relative track or block number
- the size of the initial allocation
- the amount of allocation actually used

The amount of allocation actually used is smaller than the initial allocation if the dataset has been truncated. CA EPIC for z/VSE can allocate an extra extent (called an EOF extent) if the extent being processed is nearly full when the dataset is closed.

Note: For more information, see Close Processing (see page 51).

Record Format, Length, Block Count, Block Size

The version's actual record format, record length, block count and block size are recorded. This information is recorded when the dataset is created by a program that uses logical IOCS; it can be zeros if the dataset is created by a program that uses physical IOCS.

Creation Date/Time

CA EPIC for z/VSE records the date and time the version was written.

Creating Job/Device Address

CA EPIC for z/VSE records the name of the job that created the version, the partition in which it was created, and the address (cuu) of the device on which it was created.

Last Access: Job and Date

At each open, CA EPIC for z/VSE records the date the version was accessed and the name of the job that accessed it.

Expiration Date

CA EPIC for z/VSE calculates an expiration date for each version based on retention days and DLA retention. A version remains active past its expiration date if it is still needed to satisfy cycles retention.

Tape Use and Error Count

The tape use count indicates the number of times the tape has been opened. The tape error count indicates the number of read/write errors encountered.

Both counts are kept for each tape while it has belonged to this version and for total tape use since the last date the tape was cleaned.

Current Vault and Slot Location

The current vault and slot location are recorded for each tape if the dataset is subject to a vault rotation method.

Automatic Cataloging

Each new dataset and version must have an entry in the DSN Catalog to be controlled. These entries can be created automatically for uncontrolled datasets using the AUTOCAT and AUTOLOG features or the parameter DISP=(x,CATLG) parameter on a DD statement.

AUTOCAT and AUTOLOG are configuration options described in the *Installation and System Guide*.

AUTOCAT and AUTOLOG operate globally on all qualifying datasets. DISP=(x,CATLG) operates locally on the single dataset named in the DD statement that contains this parameter. For more information on DISP=(x,CATLG), see Chapter 6, "CA EPIC for z/VSE DD Statements".

AUTOCAT and AUTOLOG are independent but parallel functions. They operate under the same rules. However there is an important functional difference between them: AUTOCAT catalogs uncontrolled *output* tape datasets and versions. AUTOLOG catalogs uncontrolled *input* tape datasets only (no versions).

AUTOCAT

AUTOCAT creates a dataset definition for an output dataset if that dataset is not already entered in the DSN Catalog. All subsequent versions created under the dataset name will also be cataloged, including the one that triggers the AUTOCAT operation.

When autocataloging multi-dataset groups that are read out of sequence, ensure that the prime dataset is entered in the DSN Catalog before any subdataset in the group is read. This requirement is automatically satisfied when the datasets are read in sequence beginning with the prime dataset.

The following fields are entered in each autocataloged dataset definition:

- Retention days
- Cycles retention
- Default disk pool
- Default primary extent

On reports and displays, automatically cataloged datasets are identified by a User ID of "AC" and given the comment "**** AUTO CATALOGUE ****". All automatically cataloged datasets are generation datasets. These dataset definitions can be changed later using Online Manager or the TSIDMNT utility.

AUTOLOG

AUTOLOG creates an CA EPIC for z/VSE dataset definition (EDD) for an input tape dataset that is not already in the DSN Catalog. However, AUTOLOG does not also autocatalog any versions to the EDD, as AUTOCAT does. AUTOLOG is primarily used to force standard EP001 MOUNT messages for previously uncontrolled input tape datasets.

When autologging multi-dataset groups that are read out of sequence, ensure that the prime dataset is entered in the DSN Catalog before any subdataset in the group is read. This requirement is satisfied automatically when datasets are read in sequence beginning with the prime dataset.

Configuration Options

Automatic cataloging is affected by the AUTOCAT, AUTOLOG, CATRT1, NCATDSK, CYCL, RETEN, DEFPOL, and DEFEXT configuration options. AUTOCAT functions can be specified differently for tape and disk.

There are three choices for automatic cataloging functions: Global, SYSPARM, and None.

Global AUTOCAT

Global automatic cataloging catalogs all datasets.

SYSPARM AUTOCAT

Only selected datasets are cataloged under SYSPARM automatic cataloging. They are cataloged automatically if the current SYSPARM matches the SYSPARM in the AUTOCAT or AUTOLOG configuration option.

No Automatic Cataloging

When you turn off automatic cataloging, new datasets can only be cataloged using Online Manager or the TSIDMNT utility.

Retention and Scratching

All catalog-controlled datasets must be scratched before tapes and disk space can be reused. However, CA EPIC for z/VSE does not scratch any datasets unless specifically told to do so. This can be done in four ways.

- Using JCL options (work/non-generation datasets only)
- Defining a dataset as "work" or as a common work dataset (CWDS)
- Using the Online Manager or TSIDMNT
- Using the SCRATCH function of the TSIDUTL utility
  The JCL scratch functions can be used to delete work (non-generation) dataset versions when they are closed as input or at the end of the job or job step. These versions are scratched regardless of whether or not the retention criteria have been satisfied.

By design, CA EPIC for z/VSE maintains only one version of a work (non-generation) dataset. As soon as a new version is opened, work dataset scratch processing immediately deletes any previous disk versions. Any previous tape versions become eligible for scratch.

CA EPIC for z/VSE common work datasets (CWDS) are retained only for the duration of a VSE job. Scratch processing always deletes a CWDS at end of job, whether the job completes normally or not.

Online Manager and TSIDMNT scratch functions delete only the dataset versions you specify. Those versions are scratched regardless of whether or not the retention criteria have been satisfied.

The SCRATCH function of the TSIDUTL program reviews all the datasets in the DSN Catalog and deletes all versions that are eligible for scratch. A dataset is considered eligible for scratch when ALL the retention criteria have been satisfied.

You can also preview a scratch run with TSIDUTL's projected scratch function (PSCRATCH) which gives you a listing of all tape datasets that are eligible for scratch on a particular date. Use this report to identify any expired datasets you might want to save, and to obtain an estimate of the number of scratch tapes you will have available at any point.

Rules for Dataset Retention

The following are the CA EPIC for z/VSE dataset retention rules.

Cataloged Datasets

CA EPIC for z/VSE's retention rules for cataloged datasets are:

Tapes are not made available for reuse and disk space is not released until dataset versions are scratched.

Non-temporary versions are not scratched until the SCRATCH function of the TSIDUTL utility is executed. TSIDUTL does not scratch any versions until they are eligible. A version is not eligible until ALL the retention criteria have been satisfied.

Any version can be scratched at any time using the SCRATCH function of Online Manager or the TSIDMNT utility.

A previous work dataset version is scratched immediately when a new version is opened. A previous tape work dataset version becomes eligible for scratch when a new version is opened.

Work dataset versions can be scratched at close (on input) or at end of job or job step using JCL options. Disk space is freed and tapes become available for reuse immediately.

Multi-dataset tape volumes are not made available for reuse until all dataset versions on the tape have been scratched.

Conditionally cataloged versions become eligible for scratch the day after their creation.

Common work dataset versions are always scratched at end of job.

Open versions become eligible for scratch according to the THLDOPEN and DHLDOPEN parameters of the TSIDUTL utility's SCRATCH function. The default is to retain open tape versions for one day and open disk versions for two days after their creation dates.

Important! Tapes in open status can be reused before they are actually scratched. They are accepted as scratch tapes if they are accessed in the same partition with a different job name. While this means that the tape is vulnerable to accidental reuse, it also means that these tapes are immediately available as scratch. Please ensure that your operators are informed of this. Use of tapes in open status can be modified by CA EPIC for z/VSE special options.

Examples

This section contains examples of retention under multiple criteria.

Retain x Versions for y Days

You need to retain each version of an inventory dataset for 100 days. You also need to retain the last 10 versions, regardless of when they were created. The dataset definition specifies:

```
Generation Dataset......YES
Retention Days.......100
Retention Cycles.....10
Retention DLA........0
```

Case 1

During the first 100 days after creating the first version, you create a total of nine versions. On day 101, no versions are scratched because you still have not completed 10 cycles.

Case 2

If you create Version 11 after only 60 days, no version is eligible for scratch because the first version you created is only 60 days old. It is retained until day 100 regardless of the number of versions created up to that time.

Retain a Work Dataset x Days after its Last Access

An accounting job produces results from work datasets in multiple stages. If you decide to save the work datasets for 15 days after the last time you use the data, catalog the dataset with:

```
Generation Dataset......NO
Retention Days.......0
Retention Cycles.....0
Retention DLA........15
```

Each version is saved for 15 days after its last access.

Temporary Datasets

If you want a temporary dataset, you specify:

```
Generation Dataset......NO
Retention Days.......0
Retention Cycles.....0
Retention DLA........0
```

All versions are eligible for scratch at all times.

Retention for Start Track 1 Datasets

By default, CA EPIC for z/VSE's retains Start Track 1 datasets with period retention only. Expiration is determined by the JCL or by the default retention for Start Track 1 datasets specified in the DSKRET configuration option. The existing version is deleted as a new version is created.

The space occupied by a Start Track 1 dataset becomes available when the expiration date in the VTOC has been reached.

Overriding Standard Retention

Since the default retention criteria may not be suitable for every version of every dataset, you may want to modify retention criteria occasionally. You can change retention for new versions, for existing versions, or for the dataset as a whole.

Retention days or an expiration date specified on a TLBL, DLBL, or DD statement for an output dataset overrides the catalog default.

You can change the retention of a cataloged version with the RETAIN and CYCLE commands of Online Manager or the TSIDMNT program. To update the retention for all new versions of a dataset, you must update the retention of previous versions explicitly with RETAIN or CYCLE.

You can update existing Start Track 1 datasets to permanent retention with the RETAIN, PERM, and PACKSCAN functions of TSIDVUT. You can delete them before their expiration date is reached with the DELETE or PACKSCAN functions.

Automatic Generation Purge

When *Automatic Generation Purge (AGP)* is enabled, CA EPIC for z/VSE automatically deletes DSN Catalog information about any disk dataset whose version number exceeds the cycles retention in the CA EPIC for z/VSE dataset definition (EDD).

AGP is enabled by setting the AGP configuration option to YES when creating or updating an EDD in the DSN Catalog.

What Happens if the Job Cancels?

If the configuration option PURBYP is set to 'YES', the automatic purge is bypassed if the job cancels. If PURBYP is set to NO, AGP remains in effect and the version is deleted.

AGP Example

The following TSIDMNT command adds an EDD to the DSN Catalog. CA EPIC for z/VSE retains 3 versions of the dataset *DSNA* at all times. When a new version is created, the oldest version is automatically deleted at end-of-job.

```
CAT  'DSNA',CYC=3,AGP=YES,...
```

The Recorder File

The Recorder logs controlled dataset activity. This includes:

- all DSN Catalog maintenance transactions, excluding TSIDVLT (vaulting) maintenance
- all opens and closes of controlled datasets
- machine use and job accounting statistics

The Recorder File is vital to emergency recovery of the DSN Catalog. It contains a record of all changes made to the DSN Catalog since the last catalog backup. CA EPIC for z/VSE uses the data in the Recorder File to bring your recovered DSN Catalog completely up to date.

Recorder data can also be used to produce job accounting reports.

You can also log your own comments in the Recorder File for documentation using the Recorder maintenance utility (TSIDRFS) which is documented in the *Installation and System Guide*.

Important! We strongly recommend keeping the Recorder active at all times.

Configuration Options

Configuration options RECORD, RECWARN, RECWMSG, and RECSHR affect the functioning of the Recorder feature.

Additional CA EPIC for z/VSE Features and Functions

LUB Allocation

VSE's Logical Unit Block (LUB) connects the logical unit (SYS number) in a program's file definition (DTF) with the physical device (cuu) that controls the dataset.

The LUB allocation feature of CA EPIC for z/VSE assigns logical units for tape and disk devices if they are opened when the SYS number specified in the program is already in use. This eliminates potential conflicts in SYS number assignments when:

- a program or job stream uses the same SYS number for two different datasets, and the datasets are stored on two different DASDs
- the requested SYS number is permanently assigned to another device

The LUB allocation feature reassigns new SYS numbers (LUBs) to both datasets, and continues processing without interruption. In addition, LUB allocation allows CA EPIC for z/VSE to write multiple extents for a single dataset on separate disk volumes to make full use of its disk pooling feature.

When a dataset is opened, if its SYS number is already in use, CA EPIC for z/VSE selects an unused SYS number. For disk datasets, it begins assigning SYS numbers with the number specified in the DSKLUB configuration option. For tape datasets, it begins with the number specified in the TAPLUB configuration option.

LUB allocation is automatic in most circumstances. In some cases, it is inhibited automatically. You can also inhibit it using JCL parameters.

Managing Tape Devices

Drive Assignments

CA EPIC for z/VSE makes and releases drive assignments automatically with the Automatic Volume Recognition (AVR), Autoattach, and Early Drive Release features. With these functions, tape drives are assigned to a partition or virtual machine only between open and close. They aren't tied up between job start and open time or between close time and EOJ, so tape drive availability is increased.

Tape drive assignments are affected by the SHARE and AUTOATT configuration options. Early Drive Release is activated for uncontrolled tapes if the configuration option UNCTAP=YES is specified.

Automatic Volume Recognition

CA EPIC for z/VSE does *Automatic Volume Recognition (AVR)* processing for both tape and disk. When necessary, it searches available drives for the correct volume and assign that drive to the partition running the job.

A tape drive is considered available for output if it:

- contains a valid tape. That is, the tape volume serial number is in the DSN Catalog in scratch status, or it has been initialized by DITTO.
- is ready and at load point
- supports the mode setting selected for the dataset
- is not in "device down" status
- is not assigned to any partition
- is not attached to any VM machine if Autoattach is on, or is attached to your machine if Autoattach is off

When AVR finds the tape, CA EPIC for z/VSE assigns the drive to the partition running the job. To function at its best, AVR requires you to delete your ASSGN statements from your JCL. For more information about assignments and JCL, please see "ASSGN Statements and LUB Allocation".

Auto Stop

If AVR cannot find an appropriate tape, CA EPIC for z/VSE pauses the partition and waits for the interval specified by the AUTOSTP configuration option, then AVR scans the drives again. This process repeats at the interval specified by the AUTOSTP configuration option. If AVR still cannot find the appropriate tape after this process has completed, then it sends a mount message to the operator and puts the partition in a wait state until the operator responds.

Autoattach

Autoattach attaches tape drives to virtual machines. Autoattach works with AVR to attach a tape drive to the virtual machine while AVR assigns it to the partition. Autoattach adds two new commands to VM: VATTACH and VDETACH. VATTACH corresponds to the IBM ATTACH command. VDETACH corresponds to the IBM DETACH command.

CA EPIC for z/VSE issues these commands during AVR to search for the desired tape on a drive that is owned by VM.

These commands differ from their IBM counterparts in the following ways:

- VDETACH detaches the tape drive without unloading the tape. This leaves the tape accessible for another AVR search.
- Both the VATTACH and VDETACH commands operate in "silent" mode. That is, they do not issue messages.

Early Drive Release

At close time, CA EPIC for z/VSE releases the tape drive assignment with *Early Drive Release* and Autoattach.

Early Drive Release and Autoattach are automatic, but they are inhibited if you use alternate assignments.

Autoattach cannot detach a drive if the job cancels or when the assignment is held past close with the TLBL option 8 or DD parameter FREE=END. However, another job on that virtual machine can be assigned to the drive, and if that job releases the assignment at close, Autoattach detaches the drive automatically.

Managing Your Tape Volumes

CA EPIC for z/VSE's tape initialization program (TSIDINIT) prevents initializing tapes with duplicate volume serial numbers and adds new tape volumes to the DSN Catalog.

TSIDLBL creates gummed volume labels for your tapes.

Vault management facilities control tape vaulting.

Tapes are added to the DSN Catalog automatically as you process datasets. In most circumstances, they do not have to be added manually. If you wish to add tapes to the DSN Catalog manually, use the ADD command (in Online Manager or the TSIDMNT program) or reinitialize the tapes with TSIDINIT.

You can migrate to cartridge tapes automatically by specifying a cartridge density as the global default for all output tapes. To do so, use the MODE configuration option.

Managing Disk Devices

Disk Pooling

CA EPIC for z/VSE uses disk pooling to make disk allocation more flexible and more efficient. The systems programmer defines disk pools for use in your installation. Disk pools can include one or more disk volumes. Part of a volume can also be allocated to a pool.

Each disk dataset must be assigned to a disk pool before allocation. Each time the dataset is written, its allocation is made in that pool. Disk pools are referenced with a six-character pool name. The same pool can also be referenced by alias pool names. In addition, you can have CPU and partition-independent pool names. If these are defined, you can reference a partition independent pool by ending its name with two dashes (for example, P O O L - -). For CPU and partition independent pools, end the name with three dashes (for example, 'P O L – – –').

Pool names are assigned to datasets through JCL or the DSN Catalog.

Leveling

Leveling reduces head contention by distributing allocations over the entire disk pool, rather than concentrating them on a single volume. Each time a new extent is written, CA EPIC for z/VSE makes the allocation on the next volume in the pool sequence.

Leveling is activated for each disk pool when the pool is defined.

Offload Utilities

CA EPIC for z/VSE's offload utility moves datasets from disk to tape and back again. Processing is done logically, so you can use the offload utility for backup, or to reorganize your disk space, or just to offload infrequently used datasets (which remain known to CA EPIC for z/VSE and automatically reload to a temporary disk dataset whenever needed). The only restriction is that the offload utility only processes controlled, sequential datasets.

VTOC Utilities

CA EPIC for z/VSE provides three utilities for displaying and updating your VTOC entries. TSIDLVT lists VTOC entries in batch mode. TVTO displays them online. TSIDVUT allows you to modify existing VTOC entries.

Starting and Stopping Partitions

You can stop a partition by responding STOP to most CA EPIC for z/VSE messages. To restart, submit the following command to the attention routine (AR):

```
MSG xx
```

where *xx* specifies the partition identifier (for example, BG, F1, F2).

Partitions can be stopped and restarted at any time using the VSE STOP and START commands.

Creating Reports

CA EPIC for z/VSE programs provide a variety of reports about tape and disk datasets, storage use, and even job accounting. For report examples and an explanation of report fields, see "Report Fields and Examples".

You can create your own reports containing CA EPIC for z/VSE data from the DSN Catalog and the Recorder File.

More information:

Customized Reports (see page 396)

The CA EPIC for z/VSE Environment

When is CA EPIC for z/VSE On?

CA EPIC for z/VSE is activated by partition. If you have just IPLed your system, or you have deactivated and reactivated CA EPIC for z/VSE, CA EPIC for z/VSE becomes functional when the first OPEN is issued in the partition.

CA EPIC for z/VSE does no processing until it is activated. This means that JCL enhancements (such as DD statements) cannot be processed until then.

There is one exception to this rule: QDAS can always be active, whether CA EPIC for z/VSE is active or not. QDAS is described below.

What Happens When CA EPIC for z/VSE Scratches or Deletes a Dataset?

CA EPIC for z/VSE provides several methods for scratching datasets. Datasets can be scratched at your request using JCL, Online Manager, the TSIDMNT catalog management utility, or the TSIDUTL catalog maintenance utility. Regardless of which method is used, CA EPIC for z/VSE scratch processing is the same.

When CA EPIC for z/VSE scratches a disk dataset, it deletes the VTOC entry, and that disk space is made available for reuse. The DSN Catalog entry for that version is deleted.

For tapes containing only one dataset, the tape is made eligible for scratch immediately. For multi-dataset tapes, the tape will not be scratched until all its datasets have been scratched. The dataset's version entry in the DSN Catalog is changed to scratch status; it is not deleted. If the dataset's definition (EDD) still exists, the scratched version record remains in the DSN Catalog. If the dataset definition is deleted, the tape version record is moved to the definition name

'** DSN DELETED **'.

For both tape and disk, only the version record is affected in normal scratch processing. However, dataset definitions can also be deleted from the DSN Catalog with the DELETE and PURGE functions. DELETE scratches all versions and VTOC entries but does not remove tape volumes from the DSN Catalog. PURGE scratches all versions and VTOC entries, and removes tape volumes from the DSN Catalog.

CA EPIC for z/VSE JCL

CA EPIC for z/VSE changes the way you use JCL. It offers many enhancements to standard VSE job control. It also requires some changes to function properly.

Physical Input/Output Control System and Pre-Open Checking

Programs that use physical IOCS cause conflicts with tape and disk management programs. As the following sections describe, CA EPIC for z/VSE doesn't get control until your program issues a request to the VSE supervisor for open, end-of-volume/extent, or close.

Physical IOCS programs cause conflicts because they don't strictly follow most logical IOCS program conventions. For example, they perform "pre-open checking": they check to see what kind of device is to be used to process a dataset before they open it, before CA EPIC for z/VSE can make the assignment. Most PIOCS programs don't update the DTF with block size, record size, record count and address of the last record written, so CA EPIC for z/VSE can't find the necessary information about the dataset at end-of-volume/extent or close.

However, processing controlled datasets with physical IOCS is not difficult. It requires some special definitions in the JCL and, in some instances, special defaults in the CA EPIC for z/VSE dataset definition (EDD).

Note: For complete requirements, see Physical IOCS Processing (see page 246).

What happens if the job cancels?

At open time, versions are entered into the DSN Catalog with an open status. When close processing is requested, CA EPIC for z/VSE updates the version record and indicates that the version has been closed and is now active.

If the job cancels before a version is closed, the version remains in open status. Open versions cannot be used as input, and they become eligible for scratch differently from closed versions. Generation datasets become eligible for scratch according to the time limits you set for the THLDOPEN and DHLDOPEN parameters of the TSIDUTL program's SCRATCH function. The default is to scratch open disk datasets after two days, tape datasets after one day. Non-generation datasets are automatically scratched the next time an output version is created. You can also use CA EPIC for z/VSE JCL options to specify that non-generation disk datasets are to be deleted immediately if a job cancels.

If a dataset is successfully closed before the job cancels, it becomes an active version, and is handled exactly as if the job had processed normally.

CA EPIC for z/VSE Job Control

CA EPIC for z/VSE job control are as follows.

JCL Enhancements

Because CA EPIC for z/VSE offers JCL enhancements which VSE cannot process, CA EPIC for z/VSE processes JCL options before VSE so that no error is generated. CA EPIC for z/VSE checks dataset status and generates the required VSE label information.

Tape Processing

The following are the various tape processing types.

Open Processing

Open, end-of-volume, and close processing are the most significant processing periods for CA EPIC for z/VSE, and open processing is the most important of the three. In each case, processing is different for tape and disk. Each is discussed separately. When a tape dataset is opened, CA EPIC for z/VSE carries out the following tasks:

CA EPIC for z/VSE determines whether a dataset is tape or disk by checking the JCL and the DSN Catalog.

CA EPIC for z/VSE determines whether the dataset is controlled. If it isn't controlled, CA EPIC for z/VSE passes control to VSE immediately. If the dataset is controlled, it reads the catalog entry to determine how to process this version.

CA EPIC for z/VSE performs AVR processing by searching available drives for the correct input tape or for an CA EPIC for z/VSE scratch tape for output.

If it doesn't find an appropriate tape, it issues a message to the operator and waits for a response.

When the tape is mounted, it assigns the drive to the requesting partition and attaches the drive to the VSE machine if Autoattach is active.

It updates the DSN Catalog (creates a dataset definition if automatic cataloging is required and records information about the version being created). The version is placed in open status.

CA EPIC for z/VSE updates the VSE Label Area. On input, it clears the dataset name from the label and substitutes the correct volume serial number. On output, it adds a period to the VOL1 label to mark the tape as an CA EPIC for z/VSE tape and writes a 99365 expiration in the HDR1 label to protect the tape against accidental overwrites.

CA EPIC for z/VSE passes control to VSE before the dataset is written. VSE then uses the information CA EPIC for z/VSE has written in the VSE Label Area to process the dataset.

End-of-Volume Processing

The steps performed in end-of-volume (EOV) processing are similar to those performed in open processing. CA EPIC for z/VSE sets the stage for reading or writing the next tape. In EOV processing, CA EPIC for z/VSE must gain and pass control to VSE more than once.

CA EPIC for z/VSE releases the assignment made for the drive that was just used and detaches the tape drive from the VSE machine if Autoattach is active.

It performs AVR for the next volume.

If it can't locate the volume, it issues a message to the operator.

When the tape is mounted on an available drive, CA EPIC for z/VSE assigns and attaches the drive.

It updates the DSN Catalog. On output, it writes another volume record for the version, places that volume in open status, and places the previous volume in conditional catalog status. On input, it updates date last accessed, use count, and other statistical information.

It passes final control to VSE.

Closing Tape Datasets

In close processing for tape, CA EPIC for z/VSE gains control from and passes control to VSE more than once.

CA EPIC for z/VSE tells VSE how to position the tape, whether to unload it, rewind it, or leave it in place.

It releases the drive's assignments and detaches the drive from the VSE machine is Autoattach is active.

On output, it writes to the DSN Catalog, placing all volumes in closed status.

CA EPIC for z/VSE passes final control to VSE.

Disk Processing

The following are the various types of disk processing.

Open and End-of-Extent Processing

The tasks required for disk open and end-of-extent processing are similar to those for tape. CA EPIC for z/VSE reads the DSN Catalog, issues messages, makes an assignment, and so on. For disk, however, CA EPIC for z/VSE must make the allocation, which is more complicated than finding the correct tape volume.

Quick Disk Allocation (QDAS)

At every open, VSE reads the VTOC sequentially, which can require thousands of I/O operations. CA EPIC for z/VSE provides the *Quick Disk Allocation System (QDAS)* to dramatically decrease the time required to allocate an output dataset.

QDAS maps disk volume free space to virtual storage. One map is created for each volume when the system is initialized. When disk space is used or freed for any dataset (controlled or uncontrolled), QDAS updates its map in storage.

Operational Notes: QDAS only updates its maps when the VTOCs of the disk volumes under its control are updated using the VSE common VTOC handler ($IJJHCVH). Programs that update the VTOC directly may bypass the QDAS interface and cause its maps to get out of sync with the VTOC. This applies to a very limited number of programs (such as DITTO). However, should this condition occur, you can refresh the QDAS map tables at any time. See the *Installation and System Guide* for procedures.

Because QDAS only maps free space, it is necessary to refresh its tables daily to pick up the space made available by Start Track 1 files that have expired. This is not necessary if all your disk files are catalog controlled. It is recommended that this procedure be performed daily after midnight as part of the CA EPIC for z/VSE daily maintenance procedures. See "Regular Maintenance" in Chapter 4 of the *Installation and System Guide* for further information.

On Output

- CA EPIC for z/VSE locates the pool to which the dataset is assigned. For cataloged datasets, it finds the pool in the DSN Catalog entry. For Start Track 1 datasets, it finds the pool that is specified in the JCL.
- If the dataset is non-generation, CA EPIC for z/VSE deletes the old version. This is simple for cataloged datasets, since the location of the dataset is contained in the DSN Catalog. For Start Track 1 datasets, CA EPIC for z/VSE has to read each VTOC in the pool until it finds the old dataset.

- CA EPIC for z/VSE decides which disk volume in that pool will be allocated using the leveling feature, if active, so that datasets will be allocated evenly across all volumes in the pool.
- CA EPIC for z/VSE checks the VTOC (or QDAS) for a large enough free extent. (CA EPIC for z/VSE does not find free extents through the DSN Catalog.) If it cannot find sufficient space in the pool for the extent, it issues a message and waits for a response from the operator.
- CA EPIC for z/VSE updates the VSE Label Area, creating an extent and assigning the disk volume.

On Input

- CA EPIC for z/VSE locates the dataset. For cataloged datasets, it finds the dataset's location in the DSN Catalog. For Start Track 1 datasets, it has to read each VTOC in the pool until it finds the first extent of the dataset.
- It updates the VSE Label Area, creating an extent and assigning the disk volume.

Cataloged dataset processing requires a great deal less overhead than Start Track 1 processing. Although CA EPIC for z/VSE has to write to the DSN Catalog each time it accesses a cataloged dataset, it uses many fewer I/O operations than would be required to read all the VTOC entries found in your disk pool. We strongly recommend using cataloged datasets instead of Start Track 1 datasets in most cases. We also recommend that you use Start Track 1 datasets only in disk pools that contain fewer than three volumes.

Space Holder Records

Before we can discuss close processing for disk datasets, you need to understand *space holder records*.

Each time CA EPIC for z/VSE allocates an extent for a dataset, it writes a VTOC entry called a space holder record to prevent the same space from being used when allocating another CA EPIC for z/VSE dataset. The VTOC entry for the space holder record contains the name of the dataset with a right parenthesis substituted for the first letter:)ATASET.NAME for DATASET.NAME. The entry is written with today's date as its expiration date.

If QDAS is running, no space holder record is created. Instead, CA EPIC for z/VSE reserves the space in the QDAS map.

Close Processing

When a program requests close processing for its datasets, some records can be left in the buffer. These are written to disk when VSE closes the dataset. In some cases there are too many records in the buffer to fit in the space remaining in the current extent, and another extent is necessary to accommodate the excess.

When CA EPIC for z/VSE closes a disk dataset, it takes the buffer into account. It determines from the buffer size and from the amount of space left in the extent whether another extent may be necessary. If another extent is needed, CA EPIC for z/VSE makes the allocation.

If VSE runs out of room when writing the buffered records, it checks the VSE Label Area for another extent. It will find the extent that CA EPIC for z/VSE has placed there. VSE then reads the VTOC, finds an expired dataset there, and rewrites the VTOC entry with the real dataset name.

If the extent is not necessary, the VTOC entry remains as a space holder record. Since the entry indicates the dataset is expired, the space can be used for any uncontrolled dataset at any time, and CA EPIC for z/VSE reuses space holder extents on an as-needed basis.

You may see space holder records in your VTOC listings. You may also see an extra extent in the DSN Catalog entry for the version. Neither of these adversely affects processing.

When closing an output disk dataset:

CA EPIC for z/VSE determines whether a precautionary extent is required. If so, it writes the space holder record, updates the VSE Label Area with the new extent information, and writes the extent record in the DSN Catalog.

It places the version in closed status in the DSN Catalog.

CA EPIC for z/VSE passes control to VSE.

Chapter 2: CA EPIC for z/VSE Online Manager

This chapter describes the CA EPIC for z/VSE Online Manager.

This section contains the following topics:

Introduction to Manual Cataloging

There are several reasons for doing manual cataloging and catalog management. Automatic cataloging is global in scope, so it will not match the exact requirements of all datasets. You may want to:

- ensure that precise catalog defaults are established before the dataset is written for the first time
- update a definition after it is created in a job
- change a dataset's requirements due to changes in government regulations or datacenter procedures

For manual cataloging, CA EPIC for z/VSE provides a panel–driven system called the Online Manager. Online Manager is easy to use, and its full–screen fill–in–the–blank formats lead you through the catalog management task step–by–step.

CA EPIC for z/VSE also provides a command–driven catalog management program called TSIDMNT for batch cataloging of new datasets and versions, for updating fields in existing dataset entries, and for DSN Catalog reporting. TSIDMNT is discussed in Chapter 3, "DSN Catalog Management Utility".

CA EPIC for z/VSE programs do not require leading zeros; numeric values are zero–filled on the left. However, in Online Manager and TSIDMNT, you can suppress the zero–fill by entering a volume serial number as a six–character string with blanks on the right (Example: "12 ").

Introduction to Online Manager

This chapter gets you started with online management, including:

- accessing Online Manager
- using panels and menus
- getting help
- exiting Online Manager

Online Access Using CICS (VSE only)

Online Manager can be accessed as a CICS transaction. See the CA EPIC for z/VSE *Installation and System Guide* for information on installing the CICS CA EPIC for z/VSE transaction. If online access will be through FAQS only, installing the CICS CA EPIC for z/VSE transaction is not required.

Online Access Using FAQS® (VSE only)

If CA's FAQS product is installed, and CA EPIC for z/VSE is defined to it, the Online Manager can be accessed by entering CA EPIC for z/VSE from any FAQS command line. If online access will be through FAQS only, installing the CICS CA EPIC for z/VSE transaction is not required.

Primary Option Menu

When the Online Manager is accessed, the Primary Option Menu is displayed first.

```
 TSIO0001.3 -------- EPIC - Primary Option Menu ----------------- USER=A35LP030
OPTION  ===>



        OPTION   COMMAND    DESCRIPTION

          M      MANAGE     Search and manage EPIC Catalog
          D      DEFINE     Define EPIC dataset definition (EDD)
          S      SYSTEM     Manage EPIC system configuration
          P      PROFILE    Manage EPIC user profiles
          C      CAPACITY   Display EPIC Catalog statistics

          H      HELP       Learn about online help
          X      EXIT       Exit EPIC




  F1=Help  F3=Return  F5=Recall  F12=Exit
```

Using Menus

When the Primary Option Menu is displayed, press PF1 to read the help information for the primary online features. Select the H (Help) option to learn more about online help capabilities.

Using Online Help

Online Manager provides a number of Help options. Choose the H (Help) option on the Primary Option Menu to learn about these. Whether the cursor is located on a panel's command area or an input field, the PF1 key always provides help information.

Exiting the Online Manager

The X (Exit) command on the Primary Option Menu ends the online session.

Online Catalog Management Commands

Even while using Online Manager panels for your catalog management, you can still execute TSIDMNT functions in command mode. To do this, place the cursor in a panel's command area, and type EPIC, followed by a space and the normal TSIDMNT syntax. See Chapter 3, "DSN Catalog Management Utility", for more information.

DSN Catalog Usage Display

Use CAPACITY to display statistics on current DSN Catalog file use.

Cataloging New Datasets

There are two commands for entering new files in the catalog. Use DEFINE to create new CA EPIC for z/VSE dataset definitions (EDDs). Use ADD to enter new versions to cataloged datasets.

Updating Datasets

There are several commands available for modifying the characteristics of existing DSN Catalog entries.

UPDATE modifies the CA EPIC for z/VSE dataset definition (EDD).

RETAIN modifies creation and expiration data for a version.

CYCLE modifies retention for a selected version and all previous versions.

CLOSE changes open and conditionally cataloged versions to active status.

LOCK locks datasets and versions, UNLOCK unlocks locked datasets and versions, and RELEASE releases locked datasets and versions for one use only.

Scratching Datasets

There are three commands for scratching files: SCRATCH, DELETE, and PURGE. DELETE deletes the dataset definition and all dataset versions. PURGE deletes the dataset definition and all its versions, and deletes associated volume serial numbers from the DSN Catalog. SCRATCH deletes one version only.

Changing the Name of a EDD

Change an CA EPIC for z/VSE dataset definition name with RENAME. Transfer a version from one dataset name to another with TRANSFER. You can make a tape version accessible with two dataset names using MCATALOG.

Tape Management Commands

CLEAN records tape cleaning.

Disk Volume Control

Using TSIDMNT commands, you can temporarily remove disk packs from CA EPIC for z/VSE processing. Use QUIESCE to prevent writing controlled output datasets on a disk volume, while still allowing datasets to be read from it. Use OFFLINE to remove the volume from both input and output processing. ONLINE returns the volume to full processing.

Using the Online Manager

The following section explains how to configure and use the Online Manager.

Configuring Online Security and Controlling Access to CA EPIC for z/VSE Online Manager

Online Manager has its own panel–driven security system. This system allows online catalog management functions and some CA EPIC for z/VSE system functions to be restricted selectively to individual users, and for multiple users to inherit (model) an individual set of security attributes. It also provides a set of security defaults for users who are not defined to CA EPIC for z/VSE online security.

Defining a security profile for every potential user at a large site can be a tedious task. CA EPIC for z/VSE Online Manager security overcomes this problem by allowing flexible combinations of global defaults, group modeling and unique individual profiles. One site may wish to grant wide system access and restrict only a few users, while another may want widespread restrictions and limited user access. All are easily accomplished with CA EPIC for z/VSE online security.

CA EPIC for z/VSE Online Manager is supplied unsecured. A default security profile ($DEFSEC) automatically governs all unsecured users; its global security defaults can be modified as needed. Users with security system access can also use a non–modifiable unsecured profile ($EPIC) as an initial model for additional profiles. The first time you modify one profile that was modeled on another, the modified profile becomes independent. If you modify any profile on which others are modeled, the changes are inherited by the others. In this way, any combination of independent and related security profiles can be created as desired.

Online Help

Online help is provided for the security panels and input fields. To receive help, place the cursor on the first character of the command area or an input field and press PF1.

Planning Your Online Manager Security

There are some issues you need to consider if you plan to use Online Manager security. First, the online security profile names you create must follow the convention appropriate for your operating environment. For example, if you are running CA EPIC for z/VSE under FAQS, user IDs defined for CA EPIC for z/VSE security must be known to FAQS.

Second, you should always reserve exclusive security system access for your security administrator and then secure $DEFSEC itself against security system access (see below).

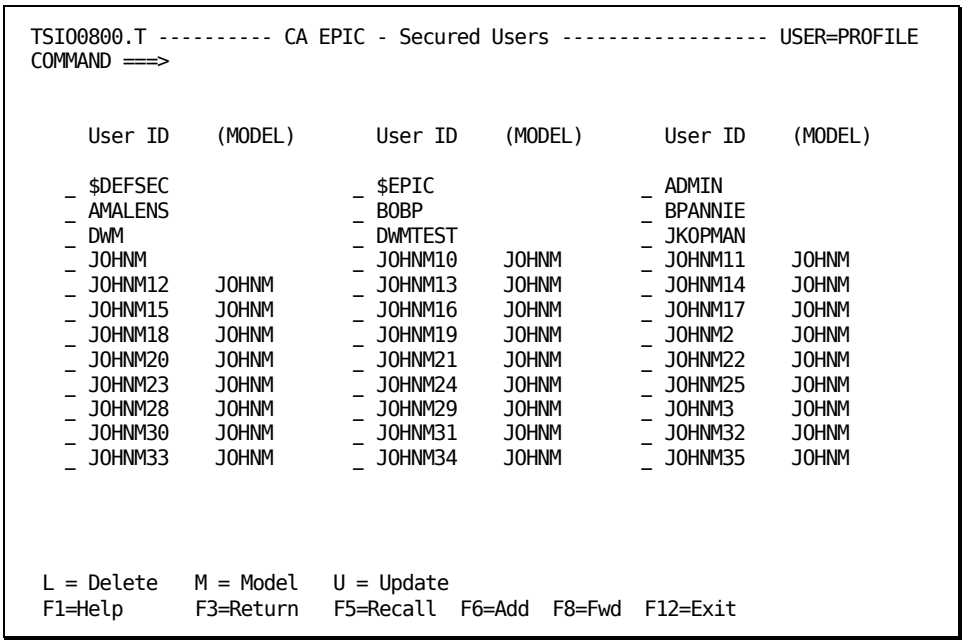
Third, you should determine whether the global default settings ($DEFSEC) will be adequate for most users, with occasional exceptions to be defined, or whether you need a number of different security profiles for specific users or groups of users (by department, for example).

Accessing the Online Security System

To access the security system from the Primary Option Menu, type SECURITY at the command line and press ENTER, or select the SYSTEM option to display the System Configuration Menu. To access the security system from the System Configuration Menu, select the SECURITY option.

Secured Users Panel

The Secured Users panel displays the names of security profiles that are defined to Online Manager. Users authorized to do so can use the Secured Users panel to create, modify, and model security profiles.



```
TSIO0800.T ---------- CA EPIC - Secured Users ------------------ USER=PROFILE
COMMAND ===>


     User ID    (MODEL)       User ID    (MODEL)       User ID    (MODEL)

   _ $DEFSEC                  _ $EPIC                  _ ADMIN
   _ AMALENS                  _ BOBP                   _ BPANNIE
   _ DWM                      _ DWMTEST                _ JKOPMAN
   _ JOHNM                    _ JOHNM10    JOHNM       _ JOHNM11    JOHNM
   _ JOHNM12    JOHNM         _ JOHNM13    JOHNM       _ JOHNM14    JOHNM
   _ JOHNM15    JOHNM         _ JOHNM16    JOHNM       _ JOHNM17    JOHNM
   _ JOHNM18    JOHNM         _ JOHNM19    JOHNM       _ JOHNM2     JOHNM
   _ JOHNM20    JOHNM         _ JOHNM21    JOHNM       _ JOHNM22    JOHNM
   _ JOHNM23    JOHNM         _ JOHNM24    JOHNM       _ JOHNM25    JOHNM
   _ JOHNM28    JOHNM         _ JOHNM29    JOHNM       _ JOHNM3     JOHNM
   _ JOHNM30    JOHNM         _ JOHNM31    JOHNM       _ JOHNM32    JOHNM
   _ JOHNM33    JOHNM         _ JOHNM34    JOHNM       _ JOHNM35    JOHNM



  L = Delete   M = Model   U = Update
  F1=Help      F3=Return   F5=Recall  F6=Add  F8=Fwd  F12=Exit
```

Default Security Profile ($DEFSEC)

The $DEFSEC profile is supplied unsecured and always acts as the default security profile for all unsecured users, so that the security administrator does not have to define a profile for every user. To prevent unauthorized access to Online Manager, $DEFSEC cannot be used as a user ID even if left unsecured. If Bypass the CA EPIC for z/VSE Sign–On Panel is set to 'N' in $DEFSEC, all unsecured users will be locked out of Online Manager unless they know a secured ID and password.

The $EPIC Security Profile

$EPIC is an unmodifiable, unsecured profile that is supplied for use as a model. After $DEFSEC has been secured, you can grant total system access to selected users by modeling their user IDs after $EPIC. Users modeled on $EPIC can themselves be modified to create customized security profiles, as with any other model. To prevent unauthorized access to Online Manager, $EPIC cannot be used as a user ID. Use caution when modeling user profiles after $EPIC and leaving them unmodified, because whoever learns one such profile name could use the Sign–On panel to gain total system access!

Creating a New User Security Profile

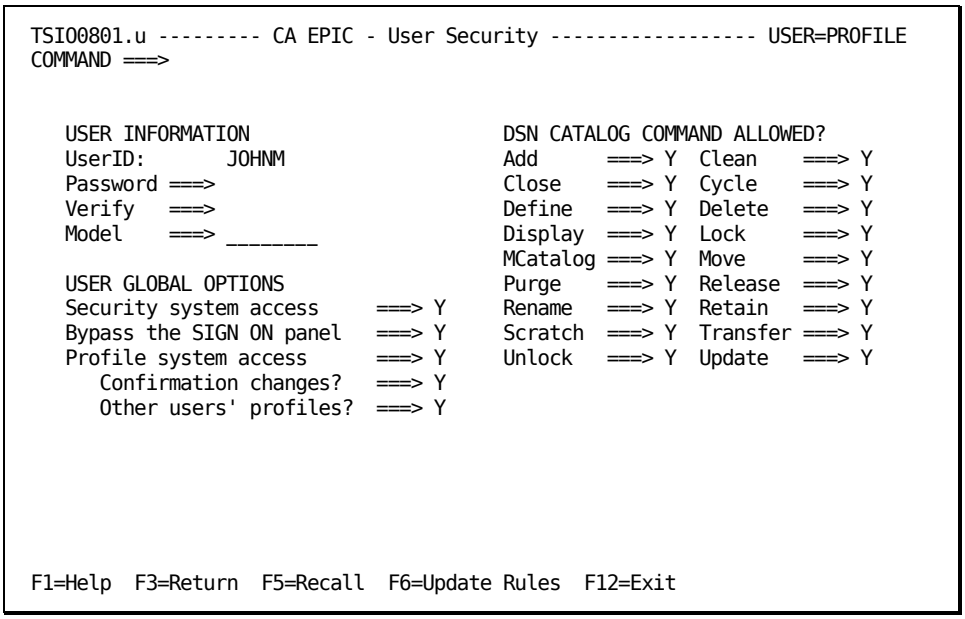
To create a brand new Secured User, use an existing security profile as the initial model.

Example

Enter SECURITY at an Online Manager command line to display the Secured Users panel. Place M (for Model) in front of the $DEFSEC profile, and press ENTER. When prompted for a new profile name, enter USER99. The list displays USER99 and shows that it is modeled on $DEFSEC. USER99 now possess the security attributes of $DEFSEC.

The User Security Panel

Users authorized to do so can use the User Security panel to modify and model security settings for a selected user or group ID. The panel allows changing the security password, setting user global options for security access, and enabling or disabling specific CA EPIC for z/VSE DSN Catalog management functions for a user or group ID.



User Security Panel Fields

USER INFORMATION - When the CA EPIC for z/VSE Sign-On panel has been enabled for a user (see below), a user password will be requested. The Password and Verify fields can be used to create and modify this password. An asterisk (*) at the end of these fields indicates an existing password. For privacy, neither entry is displayed on the screen. The Model field can be used to import security settings from another secured user.

USER GLOBAL OPTIONS - The Security system access field determines whether the selected user can make security system changes. The Bypass the SIGN ON panel field determines whether access to Online Manager is controlled by the CA EPIC for z/VSE Sign-On Panel (see example below). In CA EPIC for z/VSE only, the VSE System option access field controls access to CA EPIC for z/VSE's processing options. The Profile system access field controls access to the user's PROFILE display formats. Subsequent fields can be used to restrict the user's access to their own confirmation settings and other users' profiles, if desired. We recommend that access to Other users' profiles be set to 'N', so that profile modeling cannot be affected by anyone besides the owner of a model profile.

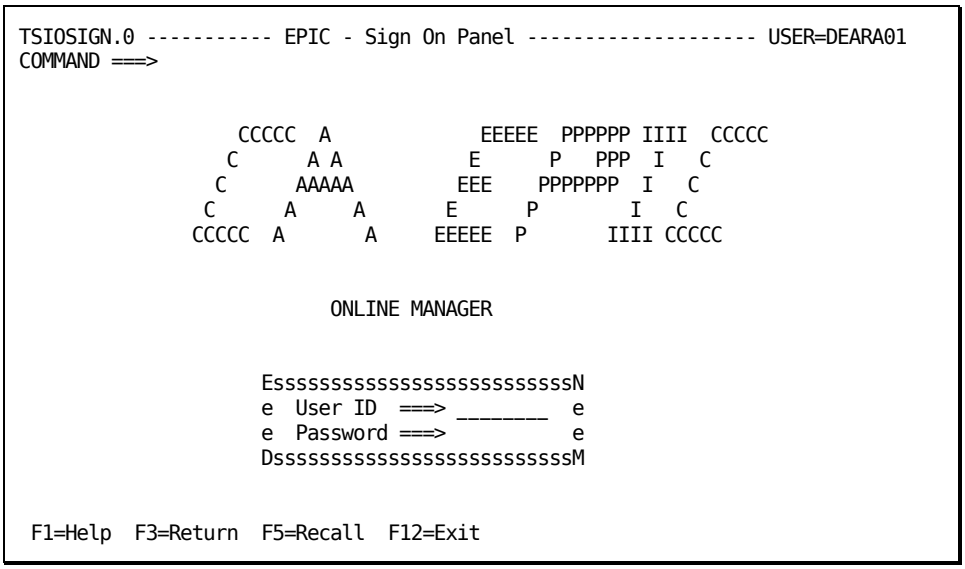
DSN CATALOG COMMANDS - These fields control access to individual DSN Catalog management functions.

Changing Security Settings for One User

Enter SECURITY at an Online Manager command line to display the Secured Users panel. Place U (for Update) in front of the selected user's security profile, and press ENTER. When the User Security panel is displayed, make the necessary changes. Press PF6 to accept the changes. If the security profile was formerly modeled after another, this relationship no longer exists, as reflected in the now–empty Model fields for this user.

Sign–On Panel

The Sign–On panel is used to require a password from a secured user. The panel prompts the user for a secured user ID and the appropriate password. The panel will only recognize user IDs listed on the Secured Users panel. If Bypass the SIGN On panel is set to 'N' in $DEFSEC, all unsecured users are locked out of Online Manager unless they know a secured user ID and password.



Security Administration Example

1. Use your security administrator's user ID as the name for a new security profile modeled on $EPIC.
2. Update the new administrator profile with password protection and enable password prompting by setting Bypass the CA EPIC for z/VSE Sign–On Panel to 'N'. Be sure to update rules with PF6 before exiting from the modified administrator profile (if you do not do this, your administrator's profile may still model $EPIC).
3. Modify $DEFSEC to deny all system access (including security) and catalog update functions to unsecured users.
4. Create any desired group security profiles. For instance, you can create a departmental profile named DEPT1 and provide it with desired catalog update access for that department. Then model several key departmental user IDs after DEPT1.
5. Create any additional customized user security profiles as desired.

Maintaining User Profiles

Online Manager allows users to customize the way dataset definition and version information is displayed. CA supplies two default profiles containing display layouts and confirmation options. The $DEFPROF default profile provides a modifiable layout that applies to all users who are not specifically defined to Online Manager. The $EPIC default profile provides a fixed standard layout. Users can model either default profile under a name of their choice and modify the characteristics of the new profile as desired. Whenever you model an existing user profile, the new profile's attributes can then be modified, but this "disconnects" it from inheriting the model's attributes in the future.

Access

From the Primary Option Menu, choose P for PROFILE. You can also type PROFILE at the command line on most panels and press ENTER.

User Profiles Panel

```
TSIO0300.P ---------- CA-EPIC - User Profiles ------------------ USER=PROFILE
COMMAND ===>


     Profile    (MODEL)       Profile    (MODEL)       Profile    (MODEL)

   _ $DEFPROF               _ $EPIC                  _ ADMIN
   _ AMALENS                _ BPANNIE    $DEFPROF    _ DWM
   _ DWMTEST                _ HEWPROF                _ HPICKEN
   _ JKOPMAN                _ JOHNM                  _ KJOHNSO
   _ RONN




 Active Profile:  $DEFPROF


 A = Activate    L = Delete    M = Model    U = Update
 F1=Help  F3=Return  F5=Recall  F6=Add  F12=Exit
```

From this panel, you can create and modify user profiles with different panel display formats and confirmation options. Press PF1 to display help information on using User Profiles panels.

Select M (for Model) to create a copy of the $EPIC or any other existing profile, and give the copy another name.

Use U (for Update) to modify the attributes of an existing profile. The first of three attribute panels is displayed. After you have changed any of your profile defaults, press ENTER to update your changes. Press PF3 to exit without changes.

Customizing the EDD Display

Use the Modify EDD Display panel to change the screen display format for CA EPIC for z/VSE dataset definitions.

The Current Display Order area provides functions to Move (M) a field to a new location Following (F) or Preceding (P) another field, or to Remove (R) a field from the display. Press PF1 for more information.

```
TSIO0301.u ------ CA-EPIC - User Profile Definition ------------ USER=PROFILE
COMMAND ===>

                          ** MODIFY EDD DISPLAY **

CURRENT DISPLAY FORMAT                                            More: +>

 Dataset Name               Cyc Reten GEN or Owner
  Width ===> 25                        WRK    ID


CURRENT DISPLAY ORDER   Profile: $DEFPROF

  _  1 Cyc           _  2 Reten         _  3 GEN or WRK    _  4 Owner ID
  _  5 Alloc Req     _  6 User ID       _  7 Last Output   _  8 Create Date
  _    Tape Pool     _    System ID     _    Attributes    _    Disk Pool
  _    Vault Method  _    SYSnnn        _    DLA           _    Tape Mode
  _    Comment       _    Reblock Value _    Primary Alloc _    Prime Dataset
  _    Sec Alloc     _    Max Sec       _    AGP


 F = Following   M = Move   P = Preceding   R = Remove
 F1=Help  F3=Return  F6=Update  F8=Fwd  F11=Right  F12=Exit
```

Press F6 to accept the new display order. The Current Display Format area will change to show the new arrangement. Press F8 to scroll to additional User Profile Definition panels.

Customizing the Version Display

Use the Modify Version Display panel to change the screen display format for dataset version information.

The Current Display Order area provides functions to Move (M) a field to a new location Following (F) or Preceding (P) another field, or to Remove (R) a field from the display. Press PF1 for more information.

```
TSIO0302.8 ------- CA-EPIC - User Profile Definition ----------- USER=PROFILE
COMMAND ===>

                        ** MODIFY VERSION DISPLAY **

CURRENT DISPLAY FORMAT                                          More: -+>

 Dataset Name              Version Number-of First  --- Creation ---
  Width ===> 25                    Vols/Exts Volser Date        Time


CURRENT DISPLAY ORDER   Profile: $DEFPROF

  _  1 Version       _  2 Exts/Vols     _  3 First Volser  _  4 Create Date
  _  5 Create Job    _  6 Expire Date   _  7 Reten         _  8 Gen Number
  _  9 Status        _ 10 Last Acc Info _ 11 Media Type    _ 12 Attributes
  _ 13 Extent Info   _ 14 File Seq      _ 15 Vault Method  _ 16 Slot
  _ 17 Blocks        _ 18 Block Size    _ 19 Record Size   _ 20 Tape Pool
  _ 21 Tape Mode     _ 22 GEN or WRK    _ 23 SYSLOG ID     _ 24 CPU ID
  _ 25 Reel          _    Create Info   _    Last Acc Date _    Last Acc Job
  _    Reten Info    _    Cyc           _    DLA           _    Last Backup

 F = Following   M = Move   P = Preceding   R = Remove
 F1=Help  F3=Return  F6=Update  F7=Bkwd  F8=Fwd  F11=Right  F12=Exit
```

Press F6 to accept the new display order. The Current Display Format area will change to show the new arrangement. Press F8 to scroll to the Modify Profile Defaults panel. Press F7 to scroll back to the Modify EDD Display panel.

Customizing the Confirmation Options

Use the Modify Profile Defaults panel to enable or disable confirmation prompts to a variety of CA EPIC for z/VSE functions, and to change the default level for a catalog search. The $EPIC default profile always requests confirmation.

```
TSIO0303.8 ------- CA-EPIC - User Profile Definition ----------- USER=PROFILE
COMMAND ===>

                      ** MODIFY PROFILE DEFAULTS **
                                                                More: -
Profile: $DEFPROF

       CONFIRMATION SETTINGS                LEVEL FOR CATALOG SEARCH

  Add Ver  ===> Y  Clean    ===> Y        Level ===> E     E = EDD
  Close    ===> Y  Cycle    ===> Y                         V = Version
  Define   ===> Y  Delete   ===> Y
  Lock     ===> Y  MCatalog ===> Y
  Move     ===> Y  Purge    ===> Y
  Release  ===> Y  Rename   ===> Y
  Retain   ===> Y  Scratch  ===> Y            SYSTEM PASSWORD PROMPT
  Sys Opts ===> Y  Transfer ===> Y
  Unlock   ===> Y  Update   ===> Y        Prompt on initial entry ===> Y




 F1=Help  F3=Return  F6=Update  F7=Bkwd  F12=Exit
```

Change confirmation settings and Search Catalog level as desired. Press F6 to accept the new settings. Press F7 to scroll back to previous User Profile Definition panels.

Using Online Manager Help

The Online Manager provides a number of types of online help for the user. These include:

- Panel help
- Input field help
- Command help
- Panel navigation help

Help for Help

The H (Help) option of the Primary Option Menu provides a description of the Help system.

Panel Help

Press PF1 in the command line of any panel to display help for that panel.

Field Help

Place the cursor at the beginning of any input field and press PF1 to display online help for that field.

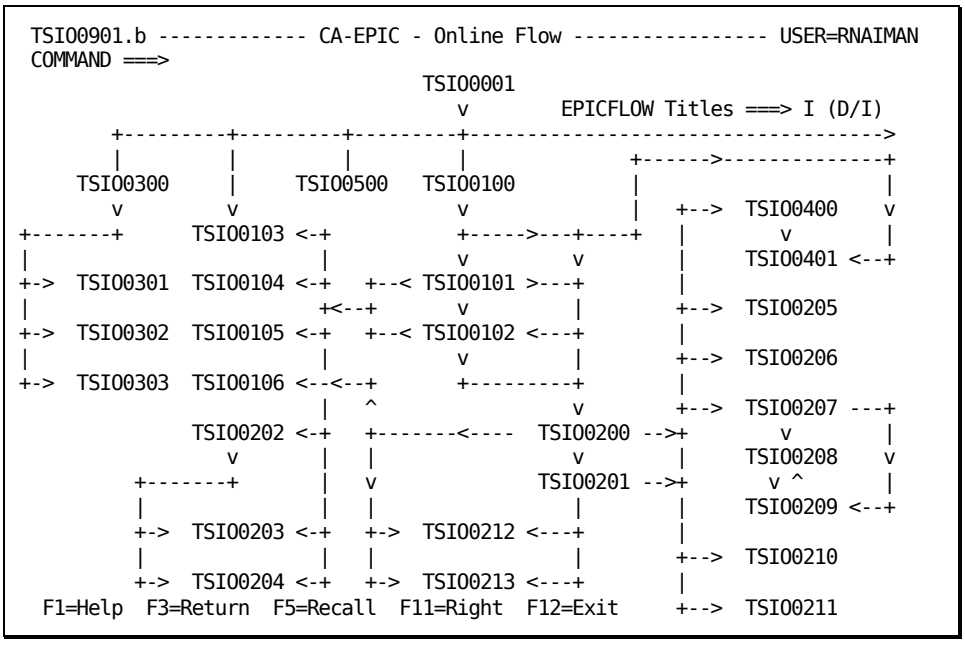
Command Help

The Online Manager provides help in understanding CA EPIC for z/VSE management commands. To view a list of these commands, enter CMDHELP at the command line on any panel.

Panel Navigation Help

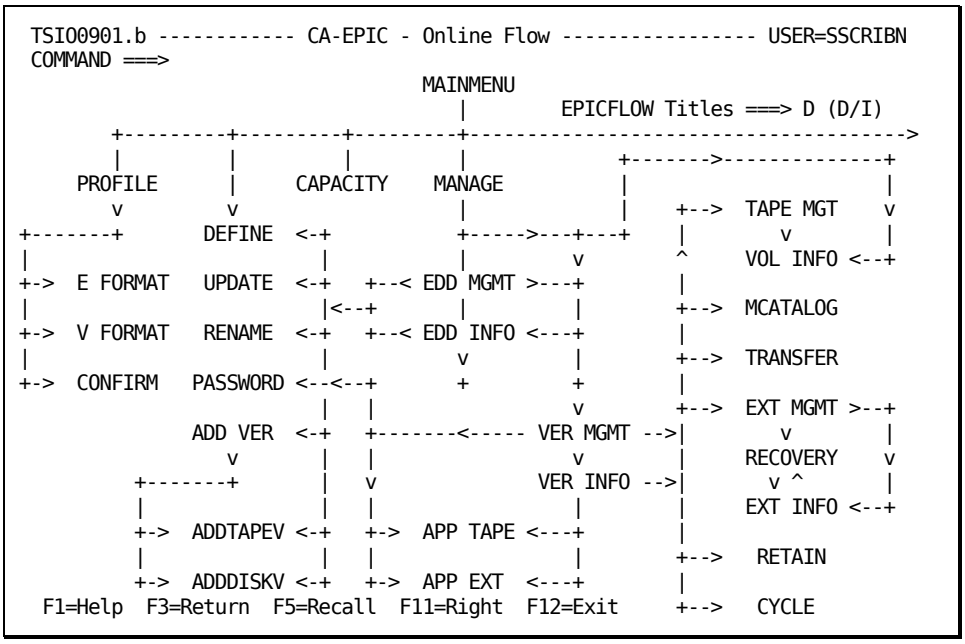
The Online Manager provides help in understanding panel navigation. To view a diagram of its panel system, enter the command EPICFLOW at the command line on any panel. The Online Flow panel below is displayed along with a highlighted path showing how the user got to their current location. Help for any panel on this diagram can be viewed by entering its panel ID on the command line in the Online Flow panel and pressing ENTER.

Panel IDs Display (I)



To view the Online Flow in terms of its panel functions, enter D (for Descriptions) on the EPICFLOW Titles field and press ENTER to display the panel below:

Panel Descriptions Display (D)



Managing CA EPIC for z/VSE Dataset Definitions (EDDs)

CA EPIC for z/VSE maintains a dataset–level description for each unique dataset name it controls, regardless of the number of versions of it that exist. This CA EPIC for z/VSE dataset definition (EDD) can be created automatically during processing, or it can be defined manually.

Displaying CA EPIC for z/VSE Dataset Definitions

Use the Search Catalog panel to search for dataset definitions in the DSN Catalog. You can search for EDDs using the following criteria:

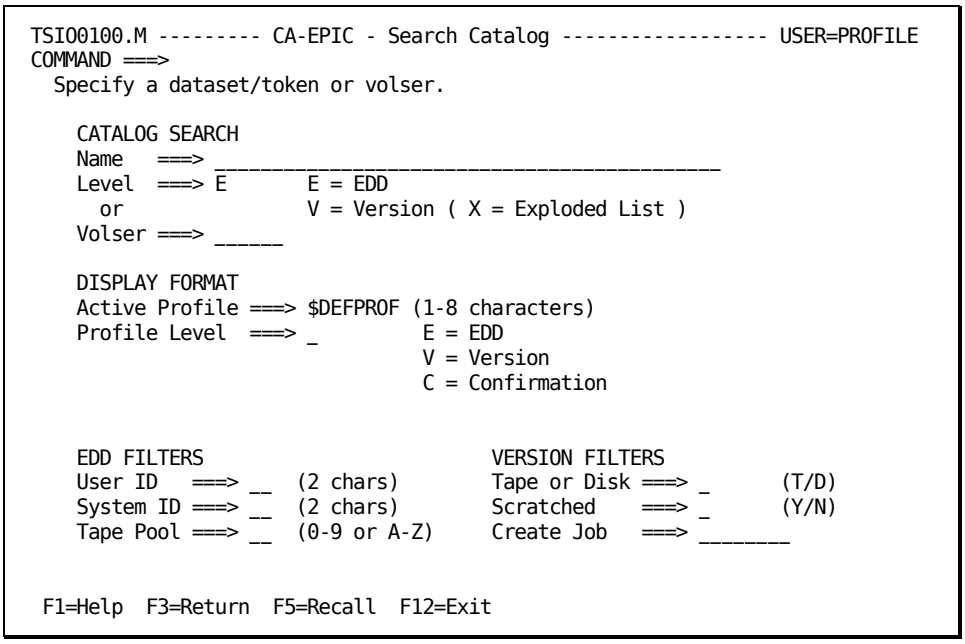
- Two–character user–defined User ID (not your system USERID)
- Two–character user–defined System ID (not the SMFID)
- Tape pool ID

Using the Manage EDDs panel, you can view, modify, and update dataset definitions.

Access

From the Primary Option Menu, select M (for MANAGE) and press ENTER to display the Search Catalog panel.

Search Catalog Panel



```
TSIO0100.M --------- CA-EPIC - Search Catalog ------------------ USER=PROFILE
COMMAND ===>
  Specify a dataset/token or volser.

    CATALOG SEARCH
    Name   ===> ____________________________________________
    Level  ===> E       E = EDD
      or               V = Version ( X = Exploded List )
    Volser ===> ______

    DISPLAY FORMAT
    Active Profile ===> $DEFPROF (1-8 characters)
    Profile Level  ===> _         E = EDD
                                  V = Version
                                  C = Confirmation


    EDD FILTERS                        VERSION FILTERS
    User ID   ===> __  (2 chars)       Tape or Disk ===> _      (T/D)
    System ID ===> __  (2 chars)       Scratched    ===> _      (Y/N)
    Tape Pool ===> __  (0-9 or A-Z)    Create Job   ===> ________


 F1=Help  F3=Return  F5=Recall  F12=Exit
```

Procedure

1. Type a fully or partially qualified dataset name in the Name field. If you specify a partial dataset name, all EDDs that meet the name criteria are displayed. An asterisk (*) can be substituted for dataset levels. A question mark (?) can be substituted for any individual character.
2. Enter an 'E' in the Level field to view selected EDDs.
3. If you want more specific search criteria, use the EDD filters at the bottom of the panel.
4. Press ENTER to display the Manage EDDs panel listing the dataset definitions that match your search criteria. Or press PF3 to exit the Search Catalog panel without listing dataset definitions.

Managing EDDs

The Manage EDDs panel lists dataset definitions matching the dataset name or expression you entered in the Search Catalog panel. From this panel, you can select a dataset definition to display or update. You can add, delete, lock, purge, release or unlock a dataset definition. You can also request a list of its versions.

Access

From the Primary Option Menu, press M (for MANAGE). Complete the Search Catalog panel and press ENTER to display the Manage EDDs panel.

Manage EDDs Panel

```
TSI00101.Z ----------- EPIC - Manage EDDs ----------------------  USER=A35LP031
COMMAND ===>
                                                                    More: ↓→

  Dataset Name               Cyc Reten  DLA ----- Attributes ------ GEN or
   Width: 25                                Lock Rel Rev Unlbl NSub  WRK

_ $$$.IJSYSLN                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYSPH                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS01                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS02                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS03                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS04                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS05                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS06                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS07                  0     0    0  N    N   N    N    N    WRK
_ $$$.SORTWK1                  0     0    0  N    N   N    N    N    WRK
_ $$$.SORTWK2                  0     0    0  N    N   N    N    N    WRK
_ $$$.SORTWK3                  0     0    0  N    N   N    N    N    WRK
_ $$$.SORTWK4                  0     0    0  N    N   N    N    N    WRK
_ ** DSN DELETED **            0     0    0  N    N   N    N    N    GEN
_ ***.VOLUME.DSN.**            0     0    0  N    N   N    N    N    GEN
_ DISKGDG.DISK.1               0     0    0  N    N   N    N    N    GEN
_ HUGH.BLOCK.MAX               0     0    0  N    N   N    N    N    GEN
_ HUGH.BLOCK.OPT               0     0    0  N    N   N    N    N    GEN
_ HUGH.BLOCK.10K               0     0    0  N    N   N    N    N    GEN
_ MULTIEXT.CAT.FILE            0     0    0  N    N   N    N    N    GEN
_ MULTIVOL.FILE001             0     0    0  N    N   N    N    N    GEN
_ MULTIVOL.FILE002             0     0    0  N    N   N    N    Y    GEN
 A = Add-Ver C = Close F = Release I = Info   J = Lock  K = Unlock  L = Delete
 M = Model   P = Purge R = Rename  U = Update V = Versions ( X = Exploded Lst)
 F1=Help  F3=Return  F5=Recall  F8=Fwd  F11=Right  F12=Exit
```

Procedure

1. To manage a dataset definition entry, type the letter of the command you want to execute in front of the dataset name to be processed. You can enter commands for as many entries as you wish.
2. Press ENTER to invoke the commands entered. Press PF3 to exit without performing any commands.

Using TSIDMNT DISPLAY

CA EPIC for z/VSE also provides the DISPLAY command of the TSIDMNT catalog management utility to list dataset definitions. See Chapter 3, "DSN Catalog Management Utility", for information on using the DISPLAY command.

Displaying a Dataset Definition

The EDD Information panel displays the dataset definition attributes. You can update those attributes or create a new definition based on the one displayed.

Access

From the Manage EDDs panel, place an I (Information) in front of a dataset definition entry and press ENTER to display the EDD Information panel.

EDD Information Panel

```
TSIO0102.G ---------- CA-EPIC - EDD Information ---------------- USER=PROFILE
COMMAND ===>

( _ ) Dataset: BG.BW.SYSIN.FILE

   TAPE INFORMATION          DATASET INFORMATION
   Tape Mode:    NO          GEN or WRK: WRK   BLKSZ:       MAX
   Tape Pool:    .           Reten:      0     Create Date: 02/01/2009
   Vault Method: 0           Cycles:     0     Last Output: 02/02/2009
   Owner ID:     .           DLA:        0     System ID:   .
   KMODEL                    SYSnnn:     0     User ID:     .

   DISK INFORMATION          DATASET ATTRIBUTES
   Disk Pool:    POOL01      Prime:      Y     Revolving:   N
   Primary:      5           Locked:     N     Unlabeled:   N
   Secondary:    1           Newsub:     N     Released:    N
   Max Sec:      0           Pswd Prot:  N
   Primary ALR:  -
   Auto Purge:   N           Comment:    .
                             Prime DSN:  .

 A = Add-Ver  C = Close  F = Release  J = Lock    K = Unlock    L = Delete
 M = Model    P = Purge  R = Rename   U = Update  V = Versions
 F1=Help  F3=Return  F5=Recall  F12=Exit
```

Procedure

1. If you want to manage the dataset definition displayed, type the desired command in front of the dataset name.
2. Press ENTER to execute the desired function. Press PF3 to exit without performing any commands.

Using TSIDMNT DISPLAY

CA EPIC for z/VSE also provides the DISPLAY command of the TSIDMNT catalog management utility to list dataset definitions. See Chapter 3, "DSN Catalog Management Utility", for information on using the DISPLAY command.

Creating a Dataset Definition

Use the Define an EDD panel to create new dataset definitions. The attributes on this panel are those of the dataset being modeled if you typed M in the input field of the EDD Information panel.

Access

You can access the Define an EDD panel by entering D (for DEFINE) on the Primary Option Menu. Or enter M (for Model) in front of the dataset definition name on either the Manage EDDs or EDD Information panels.

Define an EDD Panel

```
TSIO0103.D ------------ CA-EPIC - Define an EDD ---------------- USER=PROFILE
COMMAND ===>


  DEFINE

  Dataset ===> ____________________________________________
  Model   ===> ____________________________________________

  TAPE INFORMATION     DISK INFORMATION          DATASET INFORMATION
  Tape Mode  ===> __   Disk Pool   ===> ______   GEN or WRK     ===> ___
  Tape Pool  ===> __   Primary     ===> _______  Retention      ===> ____
  Vault Meth ===> ___  Secondary   ===> _______  Cycles         ===> ___
  Owner ID   ===> __   Max Sec     ===> ___      DLA            ===> ____
  Unlabeled  ===> _    Primary ALR ===> _______  BLKSZ( _ CISZ) ===> _____
  Revolving  ===> N    Auto Purge  ===> _       SYSnnn         ===> ___
  KMODEL     ===> _                             System ID      ===> __
                                                User ID        ===> __
  Comment    ===> ________________________________ Password       ===>
  Prime DSN  ===> ________________                 Encrypt Key    ===>



F1=Help  F3=Return  F5=Recall  F12=Exit
```

Procedure

1. To create a new EDD from scratch, specify a dataset name in the Dataset field. The dataset name is the key CA EPIC for z/VSE uses to locate a record in the DSN Catalog. For this reason, the dataset name must be unique. It must also follow all CMS naming rules. Use the Model field when you are defining a new dataset which is similar to a previously defined dataset.
2. Update any desired attributes for the dataset. Any modifications you make on the Create an EDD panel override modeled attributes.
3. Press ENTER to create the new EDD. All data entered are verified for syntactic correctness and defaults added. Press PF3 to exit the panel without creating the EDD.

Using TSIDMNT CATALOG

CA EPIC for z/VSE also provides the CATALOG command of the TSIDMNT catalog management utility to create an CA EPIC for z/VSE dataset definition (EDD). See Chapter 3, "DSN Catalog Management Utility", for information on using the CATALOG command.

Updating a Dataset Definition

Online Manager provides the Update an EDD panel to update your CA EPIC for z/VSE dataset definitions (EDDs) in the DSN Catalog. The attributes displayed on this panel are the defaults for all output datasets using this dataset name. Any modifications affect subsequent versions only.

Access

Select M (for MANAGE) from the Primary Option Menu, and use the Search Catalog panel to specify dataset name criteria. From the Manage EDDs list displayed, place a U next to the name of the dataset definition you wish to update and press ENTER.

Update an EDD Panel

```
TSIO0104.e ------------ CA-EPIC - Update an EDD ---------------- USER=PROFILE
COMMAND ===>


  UPDATE

  Dataset: BG.BW.SYSIN.FILE

  TAPE INFORMATION     DISK INFORMATION          DATASET INFORMATION
  Tape Mode  ===> NO   Disk Pool   ===> POOL01   GEN or WRK     ===> WRK
  Tape Pool  ===> __   Primary     ===> 5        Retention      ===> 0
  Vault Meth ===> 0    Secondary   ===> 1        Cycles         ===> 0
  Owner ID   ===> __   Max Sec     ===> 0        DLA            ===> 0
  Unlabeled  ===> N    Primary ALR ===> _______  BLKSZ( _ CISZ) ===> MAX
  Revolving  ===> N    Auto Purge  ===> N        SYSnnn         ===> 0
  KMODEL     ===> L                             System ID      ===> __
                                                User ID        ===> __
  Comment    ===> ______________________________ Password       ===>




F1=Help  F3=Return  F5=Recall  F12=Exit
```

Procedure

1. Update any desired attributes for the dataset definition.
   a. To remove the Disk Pool specification, replace the current value with the word "NO".
   b. To remove the "Primary", "Secondary", "Max Sec", or "Primary ALR" value, replace the current value with a zero (0).
2. Press ENTER to update the EDD, or press PF3 to exit without updating.

Using TSIDMNT UPDATE

CA EPIC for z/VSE also provides the UPDATE command of the TSIDMNT catalog management utility to modify an CA EPIC for z/VSE dataset definition. See Chapter 3, "DSN Catalog Management Utility", for information on using the UPDATE command.

Locking a Dataset Definition

Locking an CA EPIC for z/VSE dataset definition prevents its use for output (creating a new version with the same dataset name). You can lock a dataset definition from the Manage EDDs panel.

Access

From the Primary Option Menu, press M (for MANAGE). Complete the Search Catalog panel and press Enter.

Manage EDDs Panel

```
TSIO0101.Z ----------- EPIC - Manage EDDs ----------------------  USER=A35LP031
COMMAND ===>
                                                                    More: ↓→

  Dataset Name               Cyc Reten  DLA ----- Attributes ------ GEN or
   Width: 25                                 Lock Rel Rev Unlbl NSub  WRK

_ $$$.IJSYSLN                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYSPH                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS01                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS02                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS03                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS04                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS05                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS06                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS07                  0     0    0  N    N   N    N    N    WRK
_ $$$.SORTWK1                  0     0    0  N    N   N    N    N    WRK
_ $$$.SORTWK2                  0     0    0  N    N   N    N    N    WRK
_ $$$.SORTWK3                  0     0    0  N    N   N    N    N    WRK
_ $$$.SORTWK4                  0     0    0  N    N   N    N    N    WRK
_ ** DSN DELETED **            0     0    0  N    N   N    N    N    GEN
_ ***.VOLUME.DSN.**            0     0    0  N    N   N    N    N    GEN
_ DISKGDG.DISK.1               0     0    0  N    N   N    N    N    GEN
_ HUGH.BLOCK.MAX               0     0    0  N    N   N    N    N    GEN
_ HUGH.BLOCK.OPT               0     0    0  N    N   N    N    N    GEN
_ HUGH.BLOCK.10K               0     0    0  N    N   N    N    N    GEN
_ MULTIEXT.CAT.FILE            0     0    0  N    N   N    N    N    GEN
_ MULTIVOL.FILE001             0     0    0  N    N   N    N    N    GEN
_ MULTIVOL.FILE002             0     0    0  N    N   N    N    Y    GEN
 A = Add-Ver C = Close F = Release I = Info   J = Lock  K = Unlock  L = Delete
 M = Model   P = Purge R = Rename  U = Update V = Versions ( X = Exploded Lst)
 F1=Help  F3=Return  F5=Recall  F8=Fwd  F11=Right  F12=Exit
```

Procedure

1. Place the cursor in front of the name of the dataset definition you wish to lock.
2. Type J (Lock) on the line and press Enter.

Using TSIDMNT LOCK

CA EPIC for z/VSE also provides the LOCK command of the TSIDMNT catalog management utility to lock an CA EPIC for z/VSE dataset definition. See Chapter 3, "DSN Catalog Management Utility", for information on using the LOCK command.

Releasing a Locked Dataset Definition

Releasing a locked CA EPIC for z/VSE dataset definition permits one use of the EDD for output (creating one new version with the EDD name and attributes). You can release a dataset definition from the Manage EDDs panel. To permanently release an EDD, use the UNLOCK procedure.

Access

From the Primary Option Menu, press M (for MANAGE). Complete the resulting Search Catalog panel and press Enter.

Manage EDDs Panel

```
 TSIO0101.Z ----------- EPIC - Manage EDDs ----------------------  USER=A35LP031
COMMAND ===>
                                                                      More: ↓→

  Dataset Name               Cyc Reten  DLA ----- Attributes ------ GEN or
   Width: 25                                 Lock Rel Rev Unlbl NSub  WRK

_ $$$.IJSYSLN                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYSPH                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS01                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS02                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS03                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS04                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS05                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS06                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS07                  0     0    0  N    N   N    N    N    WRK
_ $$$.SORTWK1                  0     0    0  N    N   N    N    N    WRK
_ $$$.SORTWK2                  0     0    0  N    N   N    N    N    WRK
_ $$$.SORTWK3                  0     0    0  N    N   N    N    N    WRK
_ $$$.SORTWK4                  0     0    0  N    N   N    N    N    WRK
_ ** DSN DELETED **            0     0    0  N    N   N    N    N    GEN
_ ***.VOLUME.DSN.**            0     0    0  N    N   N    N    N    GEN
_ DISKGDG.DISK.1               0     0    0  N    N   N    N    N    GEN
_ HUGH.BLOCK.MAX               0     0    0  N    N   N    N    N    GEN
_ HUGH.BLOCK.OPT               0     0    0  N    N   N    N    N    GEN
_ HUGH.BLOCK.10K               0     0    0  N    N   N    N    N    GEN
_ MULTIEXT.CAT.FILE            0     0    0  N    N   N    N    N    GEN
_ MULTIVOL.FILE001             0     0    0  N    N   N    N    N    GEN
_ MULTIVOL.FILE002             0     0    0  N    N   N    N    Y    GEN
 A = Add-Ver C = Close F = Release I = Info   J = Lock  K = Unlock  L = Delete
 M = Model   P = Purge R = Rename  U = Update V = Versions ( X = Exploded Lst)
 F1=Help  F3=Return  F5=Recall  F8=Fwd  F11=Right  F12=Exit
```

Procedure

1. Place the cursor in front of the name of the dataset definition you wish to release.
2. Type F (Release) on the line and press ENTER.

Using TSIDMNT RELEASE

CA EPIC for z/VSE also provides the RELEASE command of the TSIDMNT catalog management utility to allow one use of a locked CA EPIC for z/VSE dataset definition. See Chapter 3, "DSN Catalog Management Utility", for information on using the RELEASE command.

Unlocking a Dataset Definition

Unlocking an CA EPIC for z/VSE dataset definition allows its unrestricted use for output (creating new versions with the EDD name and attributes). You can lock a dataset definition from the Manage EDDs panel.

Access

From the Primary Option Menu, press M for MANAGE. Complete the Search Catalog panel and press ENTER.

Manage EDDs Panel

```
TSIO0101.Z ----------- EPIC - Manage EDDs ----------------------  USER=A35LP031
COMMAND ===>
                                                                      More: ↓→

  Dataset Name               Cyc Reten  DLA ----- Attributes ------ GEN or
   Width: 25                                 Lock Rel Rev Unlbl NSub  WRK

_ $$$.IJSYSLN                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYSPH                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS01                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS02                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS03                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS04                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS05                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS06                  0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS07                  0     0    0  N    N   N    N    N    WRK
_ $$$.SORTWK1                  0     0    0  N    N   N    N    N    WRK
_ $$$.SORTWK2                  0     0    0  N    N   N    N    N    WRK
_ $$$.SORTWK3                  0     0    0  N    N   N    N    N    WRK
_ $$$.SORTWK4                  0     0    0  N    N   N    N    N    WRK
_ ** DSN DELETED **            0     0    0  N    N   N    N    N    GEN
_ ***.VOLUME.DSN.**            0     0    0  N    N   N    N    N    GEN
_ DISKGDG.DISK.1               0     0    0  N    N   N    N    N    GEN
_ HUGH.BLOCK.MAX               0     0    0  N    N   N    N    N    GEN
_ HUGH.BLOCK.OPT               0     0    0  N    N   N    N    N    GEN
_ HUGH.BLOCK.10K               0     0    0  N    N   N    N    N    GEN
_ MULTIEXT.CAT.FILE            0     0    0  N    N   N    N    N    GEN
_ MULTIVOL.FILE001             0     0    0  N    N   N    N    N    GEN
_ MULTIVOL.FILE002             0     0    0  N    N   N    N    Y    GEN
 A = Add-Ver C = Close F = Release I = Info   J = Lock  K = Unlock  L = Delete
 M = Model   P = Purge R = Rename  U = Update V = Versions ( X = Exploded Lst)
 F1=Help  F3=Return  F5=Recall  F8=Fwd  F11=Right  F12=Exit
```

Procedure

1. Place the cursor in front of the name of the dataset definition you wish to unlock.
2. Type K (Unlock) on the line and press ENTER.

Using TSIDMNT UNLOCK

CA EPIC for z/VSE also provides the UNLOCK command of the TSIDMNT catalog management utility to unlock an CA EPIC for z/VSE dataset definition for unrestricted access. See Chapter 3, "DSN Catalog Management Utility", for information on using the UNLOCK command.

Deleting a Dataset Definition

When you delete an CA EPIC for z/VSE dataset definition, all version information is removed from the DSN Catalog, but tape volume information is kept under the dataset name

'** DSN DELETED **' until the volume serial number is reused or purged. You can remove an EDD from the DSN Catalog using the Manage EDDs panel.

Access

From the Primary Option Menu, press M (for MANAGE). Complete the Search Catalog panel and press ENTER.

Manage EDDs Panel

```
TSIO0101.Z ----------- EPIC - Manage EDDs ----------------------  USER=A35LP031
COMMAND ===>
                                                                    More: ↓→

  Dataset Name              Cyc Reten  DLA ----- Attributes ------ GEN or
   Width: 25                                Lock Rel Rev Unlbl NSub  WRK

_ $$$.IJSYSLN                 0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYSPH                 0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS01                 0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS02                 0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS03                 0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS04                 0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS05                 0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS06                 0     0    0  N    N   N    N    N    WRK
_ $$$.IJSYS07                 0     0    0  N    N   N    N    N    WRK
_ $$$.SORTWK1                 0     0    0  N    N   N    N    N    WRK
_ $$$.SORTWK2                 0     0    0  N    N   N    N    N    WRK
_ $$$.SORTWK3                 0     0    0  N    N   N    N    N    WRK
_ $$$.SORTWK4                 0     0    0  N    N   N    N    N    WRK
_ ** DSN DELETED **           0     0    0  N    N   N    N    N    GEN
_ ***.VOLUME.DSN.**           0     0    0  N    N   N    N    N    GEN
_ DISKGDG.DISK.1              0     0    0  N    N   N    N    N    GEN
_ HUGH.BLOCK.MAX              0     0    0  N    N   N    N    N    GEN
_ HUGH.BLOCK.OPT              0     0    0  N    N   N    N    N    GEN
_ HUGH.BLOCK.10K              0     0    0  N    N   N    N    N    GEN
_ MULTIEXT.CAT.FILE           0     0    0  N    N   N    N    N    GEN
_ MULTIVOL.FILE001            0     0    0  N    N   N    N    N    GEN
_ MULTIVOL.FILE002            0     0    0  N    N   N    N    Y    GEN
 A = Add-Ver C = Close F = Release I = Info   J = Lock  K = Unlock  L = Delete
 M = Model   P = Purge R = Rename  U = Update V = Versions ( X = Exploded Lst)
 F1=Help  F3=Return  F5=Recall  F8=Fwd  F11=Right  F12=Exit
```

Procedure

1. Place the cursor next to the dataset definition you wish to delete.
2. Type L (Delete) on the line and press ENTER.

Using TSIDMNT DELETE

CA EPIC for z/VSE also provides the DELETE command of the TSIDMNT catalog management utility to remove an CA EPIC for z/VSE dataset definition. See Chapter 3, "DSN Catalog Management Utility", for information on using the DELETE command.

Managing Dataset Versions

Displaying Version Information

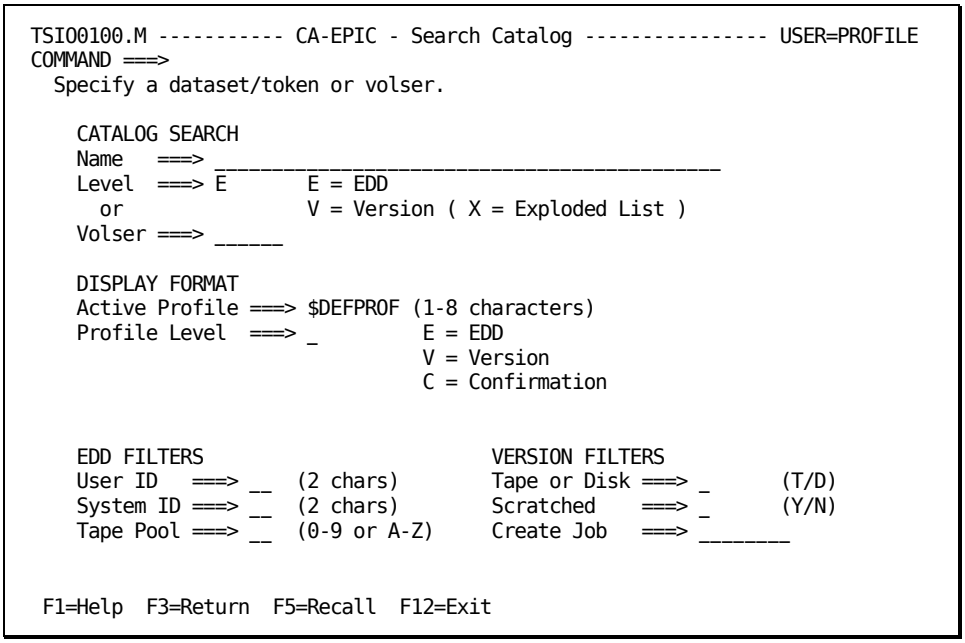
In order to view or maintain a dataset version online, you must first find its entry in the DSN Catalog. Online Manager provides the Search Catalog panel to request a list of dataset versions according to specified criteria. You can search for versions using the following criteria:

- Two–character user–defined User ID (not system USERID)
- Two–character user–defined System ID (not SMFID)
- Whether scratch or active
- Tape pool ID
- Creation job name

Access

From the Primary Option Menu, press M (for MANAGE) to display the Search Catalog panel.

Search Catalog Panel



```
TSIO0100.M ----------- CA-EPIC - Search Catalog ---------------- USER=PROFILE
COMMAND ===>
  Specify a dataset/token or volser.

    CATALOG SEARCH
    Name   ===> ____________________________________________
    Level  ===> E       E = EDD
      or                V = Version ( X = Exploded List )
    Volser ===> ______

    DISPLAY FORMAT
    Active Profile ===> $DEFPROF (1-8 characters)
    Profile Level  ===> _         E = EDD
                                  V = Version
                                  C = Confirmation


    EDD FILTERS                         VERSION FILTERS
    User ID   ===> __  (2 chars)        Tape or Disk ===> _      (T/D)
    System ID ===> __  (2 chars)        Scratched    ===> _      (Y/N)
    Tape Pool ===> __  (0-9 or A-Z)     Create Job   ===> ________


 F1=Help  F3=Return  F5=Recall  F12=Exit
```

Procedure

1. Type a fully or partially qualified dataset name in the Name field. If you specify a partial dataset name, all versions that meet the name criteria are displayed. An asterisk (*) can be substituted for dataset levels. A question mark (?) can be substituted for any individual character.
2. Enter an V in the Level field to view selected versions.
3. If you want more specific search criteria, use the version filters at the bottom of the panel.
4. Press ENTER to display the Manage Versions panel listing the dataset versions that match your search criteria. Or press PF3 to exit the Search Catalog panel without listing dataset versions.

Using TSIDMNT DISPLAY

CA EPIC for z/VSE also provides the DISPLAY command of the TSIDMNT catalog management utility to list version information. See Chapter 3, "DSN Catalog Management Utility", for information on using the DISPLAY command.

Selecting a Version from a List

The Manage Versions panel lists dataset versions based on the search criteria specified on the Search Catalog panel. From Manage Versions, you can select a specific version for display. You can also close, lock, unlock, release, scratch, retain or cycle the version. From Manage Versions, you can also display and maintain tape volume information.

Access

Once you have used the Search Catalog panel to enter your search criteria, the Manage Versions panel is displayed, listing all dataset versions that meet the criteria specified. The listing begins with the dataset name you entered. If you did not enter a dataset name, the listing begins with the first dataset in the DSN Catalog meeting any criteria you specified. The size of the DSN Catalog determines how long it takes to retrieve the data.

Manage Versions Panel

```
TSIO0200.D ----------- CA-EPIC - Manage Versions --------------- USER=PROFILE
COMMAND ===>
                                                                More: >


  Dataset Name              Version Number-of First  --- Creation ---
   Width: 25                        Vols/Exts Volser Date        Time

_ TECHEPI2.CICS22.DUMPA         1        1    EPC229 04/26/2009 11:52
_ TECHEPI2.CICS22.DUMPB         1        1    EPC182 04/15/2009 15:17









 A = Append C = Close  E = Ext/Reels F = Release H = Retain   I = Information
 J = Lock   K = Unlock M = MCatalog  S = Scratch T = Transfer Y = Cycle
 F1=Help  F3=Return  F5=Recall  F11=Right  F12=Exit
```

Procedure

1. To manage a version entry, type the letter of the command you want to execute in front of the version to be processed. You can enter commands for as many entries as you wish.
2. Press ENTER to invoke the commands entered. Press PF3 to exit without performing any commands.

TSIDMNT

CA EPIC for z/VSE provides the TSIDMNT catalog management utility to manage dataset versions. See Chapter 3, "DSN Catalog Management Utility", for information on TSIDMNT.

Displaying Version Attributes

The Version Information panel lists information on a specific dataset version. From this panel you can add a new version or display information on associated volumes.

Access

From the Manage Versions panel, place an I (Information) next to the dataset name and press ENTER.

Version Information Panel

```
TSIO0201.G --------- CA-EPIC - Version Information ------------- USER=PROFILE
COMMAND ===>

( _ ) Dataset: BG6.JOHNM.VAULT2                              Version: 2

  VERSION HISTORY                                          STATUS
  Create Date:   07/26/199   Expire Date: 07/26/199        Scratched:  N
  Create Time:   12:22       DLA:         0                Open:       N
  Create Job:    VAULT       Cycles:                       Locked:     N
  Last Acc Date: 07/26/199   Reten:       0                Released:   N
  Last Acc Job:  VAULT       Last Backup: .                CondCat:    N
                                                           Offloaded:  N
  VERSION INFORMATION
  GEN or WRK:    GEN         Block Size:  0                ATTRIBUTES
  Generation:    2           Record Size: 0                Newsub:     N
  File Seq:      1           Blocks:      0                Revolving:  N
  SYSLOG ID:     BG          Tape Mode:   NO               MCataloged: Y
  CPU ID:        .           Media Type:  TAPE             Unlabeled:  N
  Vault ID:      -           Slot:        -                Cycled:     N
  Tape Pool:     .           Reels:       1                Prime:      Y

 A = Append  C = Close     E = Reels    F = Release   H = Retain  J = Lock
 K = Unlock  M = Mcatalog  S = Scratch  T = Transfer  Y = Cycle
 F1=Help  F3=Return  F5=Recall  F12=Exit
```

Using TSIDMNT DISPLAY

CA EPIC for z/VSE also provides the DISPLAY command of the TSIDMNT catalog management utility to list version information in batch mode. See Chapter 3, "DSN Catalog Management Utility", for information on using the DISPLAY command.

Adding Version Information

Using the Add a Tape Version panel, you can add a new version to an existing dataset definition or you can add new volume serial numbers to an existing dataset version.

Access

To add a new tape version to a controlled dataset definition, access the Manage EDDs panel and enter A (for Add) to display the Add a Tape Dataset panel. To append reels to an existing version, access the Manage Versions panel and enter A (for Append) in the input field and press ENTER.

Add a Tape Version Panel

```
TSIO0203.t ----- CA-EPIC - Add a Tape Dataset Version ---------- USER=PROFILE
COMMAND ===>

  Dataset ===> BG5.EET312.LIBR.BACKUP

     VERSION INFORMATION
     Version Number             ===> 1      (1 - 9999)
     Version Block Size         ===> _____  (1 - 32767)
     File Seq Number            ===> ____   (2 - 9999)
     Offload Seq Number         ===> ____   (1 - 9999)
     Use scratch tapes          ===> _      (Y/N)

     VOLSER(S)                                              More: ›
      1 ______   2 ______   3 ______   4 ______   5 ______   6 ______
      7 ______   8 ______   9 ______  10 ______  11 ______  12 ______
     13 ______  14 ______  15 ______  16 ______  17 ______  18 ______
     19 ______  20 ______  21 ______  22 ______  23 ______  24 ______
     25 ______  26 ______  27 ______  28 ______  29 ______  30 ______
     31 ______  32 ______  33 ______  34 ______  35 ______  36 ______
     37 ______  38 ______  39 ______  40 ______  41 ______  42 ______
     43 ______  44 ______  45 ______  46 ______  47 ______  48 ______


  F1=Help  F3=Return  F5=Recall  F8=Fwd  F11=Disk Add  F12=Exit
```

Procedure

1. Specify the desired attributes and volsers for the new version.
2. Press ENTER to add the tape version.

Using TSIDMNT ADD

CA EPIC for z/VSE also provides the ADD command of the TSIDMNT catalog management utility to add a version to a controlled dataset. See Chapter 3, "DSN Catalog Management Utility", for information on using the ADD command.

Locking a Version

Locking a version prevents its use as input. You can lock a version from any panel that allows general version management.

Access

Enter the dataset name and Level V on the Search Catalog panel, and press ENTER to display the Manage Versions panel.

Manage Versions Panel

```
TSIO0200.D ----------- CA-EPIC - Manage Versions --------------- USER=PROFILE
COMMAND ===>
                                                                More: >


  Dataset Name              Version Number-of First  --- Creation ---
   Width: 25                        Vols/Exts Volser Date        Time

_ TECHEPI2.CICS22.DUMPA         1        1    EPC229 04/26/2009 11:52
_ TECHEPI2.CICS22.DUMPB         1        1    EPC182 04/15/2009 15:17









 A = Append C = Close  E = Ext/Reels F = Release H = Retain   I = Information
 J = Lock   K = Unlock M = MCatalog  S = Scratch T = Transfer Y = Cycle
 F1=Help  F3=Return  F5=Recall  F11=Right  F12=Exit
```

Procedure

1. Place the cursor on the selection line of the version you wish to lock.
2. Type J (Lock) on the line and press ENTER.

Using TSIDMNT LOCK

CA EPIC for z/VSE also provides the LOCK command of the TSIDMNT catalog management utility to lock a version. See Chapter 3, "DSN Catalog Management Utility", for information on using the LOCK command.

Releasing a Locked Version

Releasing a version permits one use of a locked version as input. You can release a locked version from any panel that allows general version management.

Access

Enter the dataset name and Level V on the Search Catalog panel, and press ENTER to display the Manage Versions panel.

Manage Versions Panel

```
TSIO0200.D ---------- CA-EPIC - Manage Versions ---------------- USER=PROFILE
COMMAND ===>
                                                                More: >


  Dataset Name              Version Number-of First  --- Creation ---
   Width: 25                        Vols/Exts Volser Date        Time

_ TECHEPI2.CICS22.DUMPA         1        1    EPC229 04/26/2009 11:52
_ TECHEPI2.CICS22.DUMPB         1        1    EPC182 04/15/2009 15:17









 A = Append C = Close  E = Ext/Reels F = Release H = Retain   I = Information
 J = Lock   K = Unlock M = MCatalog  S = Scratch T = Transfer Y = Cycle
 F1=Help  F3=Return  F5=Recall  F11=Right  F12=Exit
```

Procedure

1. Place the cursor on the selection line of the version you wish to lock.
2. Type F (Release) on the line and press ENTER.

Using TSIDMNT RELEASE

CA EPIC for z/VSE also provides the RELEASE command of the TSIDMNT catalog management utility to allow one use of a locked version for input. See Chapter 3, "DSN Catalog Management Utility", for information on using the RELEASE command.

Unlocking a Version

Unlocking a version allows its unrestricted use as input. You can unlock a version from any panel that allows general version management.

Access

Enter the dataset name and Level V on the Search Catalog panel, and press ENTER to display the Manage Versions panel.

Manage Versions Panel

```
TSIO0200.D ---------- CA-EPIC - Manage Versions ---------------- USER=PROFILE
COMMAND ===>
                                                                More: >


  Dataset Name              Version Number-of First  --- Creation ---
   Width: 25                        Vols/Exts Volser Date        Time

_ TECHEPI2.CICS22.DUMPA         1        1    EPC229 04/26/2009 11:52
_ TECHEPI2.CICS22.DUMPB         1        1    EPC182 04/15/2009 15:17









 A = Append C = Close  E = Ext/Reels F = Release H = Retain   I = Information
 J = Lock   K = Unlock M = MCatalog  S = Scratch T = Transfer Y = Cycle
 F1=Help  F3=Return  F5=Recall  F11=Right  F12=Exit
```

Procedure

1. Place the cursor on the selection line of the version you wish to lock.
2. Type K (Unlock) on the line and press ENTER.

Using TSIDMNT UNLOCK

CA EPIC for z/VSE also provides the UNLOCK command of the TSIDMNT batch management utility to unlock a version for unrestricted use as input. See Chapter 3, "DSN Catalog Management Utility", for information on using the UNLOCK command.

Closing a Version

It may be necessary to manually close a dataset version when it is in open status or conditionally cataloged by the abnormal termination of a job. You can close a version from the Manage Versions panel.

Access

Once you have used the Search Catalog panel to enter your search criteria, the Manage Versions panel is displayed, listing all dataset versions that meet the name criteria specified.

Manage Versions Panel

```
TSIO0200.D ---------- CA-EPIC - Manage Versions ---------------- USER=PROFILE
COMMAND ===>
                                                                More: >


  Dataset Name              Version Number-of First  --- Creation ---
   Width: 25                        Vols/Exts Volser Date        Time

_ TECHEPI2.CICS22.DUMPA         1        1    EPC229 04/26/2009 11:52
_ TECHEPI2.CICS22.DUMPB         1        1    EPC182 04/15/2009 15:17









 A = Append C = Close  E = Ext/Reels F = Release H = Retain   I = Information
 J = Lock   K = Unlock M = MCatalog  S = Scratch T = Transfer Y = Cycle
 F1=Help  F3=Return  F5=Recall  F11=Right  F12=Exit
```

Procedure

1. Place the cursor on the selection line of the version you wish to close.
2. Type C (Close) on the line and press ENTER.

Using TSIDMNT CLOSE

CA EPIC for z/VSE also provides the CLOSE command of the TSIDMNT catalog management utility to close a dataset version. See Chapter 3, "DSN Catalog Management Utility", for information on using the CLOSE command.

Removing Version Information from the DSN Catalog

A dataset's version information can be removed from the DSN Catalog by deleting its dataset definition or by purging its tape volume serial number.

Procedure

See Deleting a Dataset Definition or Purging a Tape Volume for information on actions which delete version information.

Using TSIDMNT DELETE and PURGE

CA EPIC for z/VSE also provides the DELETE and PURGE commands of the TSIDMNT catalog management utility to remove a dataset definition or purge a tape volume from the DSN Catalog. See Chapter 3, "DSN Catalog Management Utility", for information on using the DELETE and PURGE commands.

Managing Tapes

Tapes in the DSN Catalog

Every tape used for an CA EPIC for z/VSE-controlled dataset is recorded in the DSN Catalog. Tapes are identified in the DSN Catalog by volume serial number. New tapes are added to the DSN Catalog automatically when CA EPIC for z/VSE-controlled datasets residing on those tapes are processed. You can also enter tape information into the DSN Catalog by using the Online Manager or the TSIDMNT ADD command. Volumes entered into an ACL are added to the dataset name '***.VOLUME.DSN.**' if they do not already exist.

Valid Volsers

Every controlled tape used must have a unique volume serial number. Even unlabeled tapes must be identified with their own volume serial numbers. This strict identification simplifies tape control and retrieval. For controlled datasets opened for output, CA EPIC for z/VSE checks the volume serial number of any new tape against the DSN Catalog and requires you to enter a new one or use a different tape if there is a duplication and still–active datasets on the tape.

Volume serial numbers can be any combination of alphanumeric characters, up to six characters in length. However, volume serial number 000000 is reserved by CA EPIC for z/VSE. It is used as a special marker in the DSN Catalog and cannot be used. Groups of volsers do not have to be consecutive. Any unique volume serial number is valid. Since tape pools are defined by ranges of volume serial numbers, it is helpful to initialize pool tapes with consecutive volume serial numbers.

How CA EPIC for z/VSE Produces Scratch Tapes

Scratch tapes are produced by the SCRATCH functions of the Online Manager and TSIDMNT. In addition, the TSIDUTL SCRATCH function produces a report listing all available scratch tapes. See in this manual for more information. When a tape is scratched in the DSN Catalog, it is automatically scratched in the ACL also.

Note: For more information, see Retention and Scratching (see page 99).

Tape Pooling

You can use tape pooling to group similar types of tapes into tape pools. For instance, your high quality tapes could constitute one pool, short tapes another, long tapes a third, and so on. Datasets are assigned to tape pools through the DSN Catalog. When a pooled dataset is created, CA EPIC for z/VSE requests the operator for a scratch tape from the appropriate pool.

Each pool is given a one–character pool code (A to Z or 0 to 9) and is assigned one or more volume serial number ranges. If a volume serial number falls within a pool's range, the tape belongs to that pool. A pool tape can only be used to write datasets assigned to that pool. If a volume serial number does not fall in any of the defined ranges, the tape is not a pool tape. It can only be used to write datasets which are not assigned to a tape pool. Datasets can be assigned to tape pools by manual or automatic cataloging.

If you use ACL support with pooling, the pools must be defined to the ACL system as "POOL-*x*" where *x* is the one–character CA EPIC for z/VSE pool code. Refer to the ACL vendor's documentation for information about tape pools.

How CA EPIC for z/VSE Selects Tapes for Mounting

CA EPIC for z/VSE controls which tapes can be used for output. It checks the volume serial number of every tape processed. If the mounted volume is controlled, its serial number is compared with its corresponding DSN Catalog entry for scratch status before it can be used as output.

If the tape is uncontrolled, CA EPIC for z/VSE checks to see if the volume serial number is unique within the DSN Catalog. If the tape passes this test, it is added to the DSN Catalog and used automatically. If it does not, the tape is unloaded and a new tape requested.

Single or Prime Dataset Output

If the file has no restrictions, the operator can mount any valid scratch tape which does not belong to a tape pool.

If the file belongs to a tape pool, CA EPIC for z/VSE issues a general request for a scratch or a specific mount from that pool.

If revolving tapes are used, CA EPIC for z/VSE issues a general request for a scratched revolving tape from that dataset.

Subdataset Output

If the tape is not already mounted, CA EPIC for z/VSE requests the operator to mount the volume serial number of the last tape used for that multi–file group.

Uncontrolled Datasets

Ordinarily, no CA EPIC for z/VSE–controlled tape can be used to write an uncontrolled dataset.

ACL Support

If ACL support is activated, CA EPIC for z/VSE always attempts to satisfy a mount using an ACL volume. The ACL volume must meet the criteria of scratch status, mode, drive, availability, and so on, to be selected. If the ACL cannot satisfy the mount request, normal AVR is attempted, followed by the operator mount request message.

ACL cartridges should never be pre–mounted, unlike normal drives which can benefit from AVR.

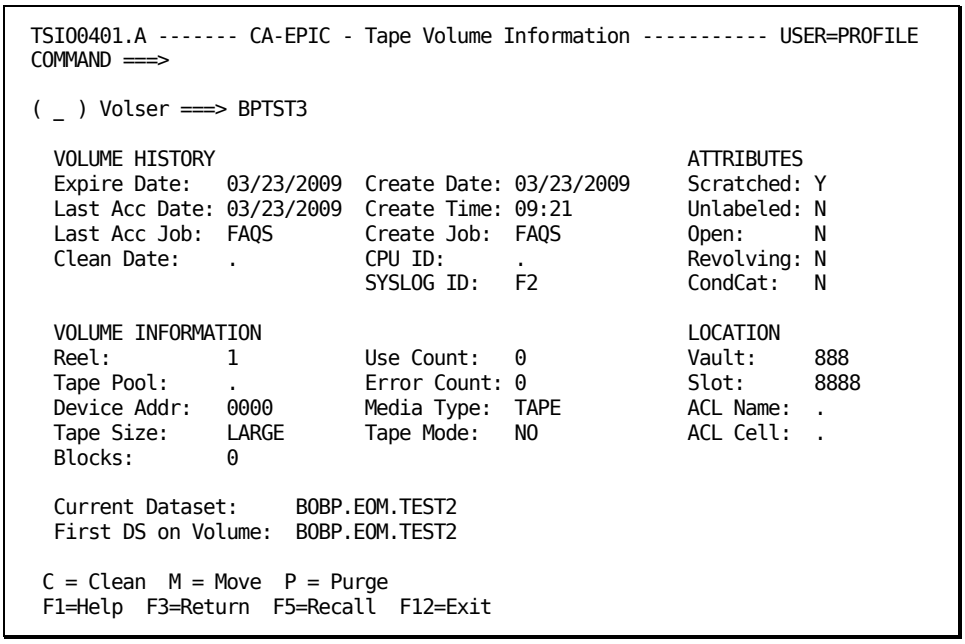
Displaying Tape Volume Information

Online Manager provides the Tape Volume Information panel to display information on any controlled tape volume. You can also execute tape functions from the Tape Volume Information panel.

Access

Choose M (for MANAGE) on the Primary Option Menu to display the Search Catalog panel. Type the desired volume serial number in the Volume Serial Number field. Press ENTER to display the Tape Volume Information panel.

Tape Volume Information Panel



```
TSIO0401.A ------- CA-EPIC - Tape Volume Information ----------- USER=PROFILE
COMMAND ===>

( _ ) Volser ===> BPTST3

  VOLUME HISTORY                                        ATTRIBUTES
  Expire Date:   03/23/2009  Create Date: 03/23/2009     Scratched: Y
  Last Acc Date: 03/23/2009  Create Time: 09:21         Unlabeled: N
  Last Acc Job:  FAQS        Create Job:  FAQS          Open:      N
  Clean Date:    .           CPU ID:      .             Revolving: N
                             SYSLOG ID:   F2            CondCat:   N

  VOLUME INFORMATION                                    LOCATION
  Reel:          1           Use Count:   0             Vault:     888
  Tape Pool:     .           Error Count: 0             Slot:      8888
  Device Addr:   0000        Media Type:  TAPE          ACL Name:  .
  Tape Size:     LARGE       Tape Mode:   NO            ACL Cell:  .
  Blocks:        0

  Current Dataset:     BOBP.EOM.TEST2
  First DS on Volume:  BOBP.EOM.TEST2

 C = Clean  M = Move  P = Purge
 F1=Help  F3=Return  F5=Recall  F12=Exit
```

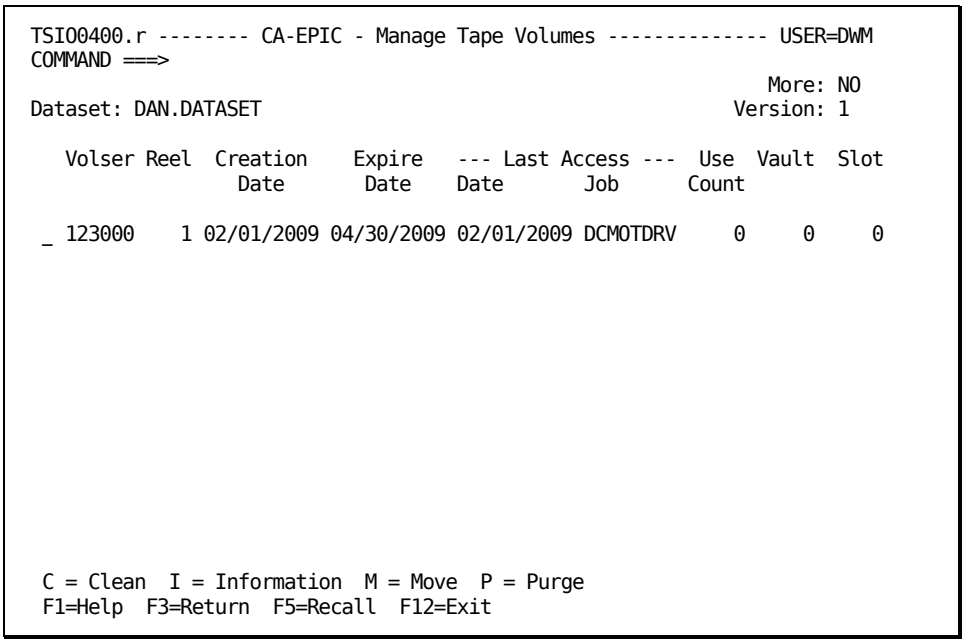
Maintaining Tape Volume Information

Use the Manage Tape Volumes panel to manage tape volume information.

Access

From the Primary Option Menu, enter M (for MANAGE) to access the Search Catalog panel. Enter a volume serial number in the Volser field and press ENTER to display the panel below.

Manage Tape Volumes Panel



```
TSIO0400.r -------- CA-EPIC - Manage Tape Volumes -------------- USER=DWM
COMMAND ===>
                                                                More: NO
Dataset: DAN.DATASET                                          Version: 1

   Volser Reel  Creation    Expire   --- Last Access ---  Use  Vault  Slot
                  Date       Date    Date       Job      Count

 _ 123000    1 02/01/2009 04/30/2009 02/01/2009 DCMOTDRV     0     0     0









 C = Clean  I = Information  M = Move  P = Purge
 F1=Help  F3=Return  F5=Recall  F12=Exit
```

Procedure

Both the Manage Tape Volumes and Tape Volume Information panels provides access to the following functions:

- CLEAN
- MOVE
- PURGE

1. You will be able to purge this volser from the Manage tape Volumes panel. NOTE: If you want CA EPIC for z/VSE to confirm all PURGE requests, make sure that the PURGE field is set to 'Y' in the Modify Profile Defaults (CONFIRM) panel.
2. Press ENTER to execute the requested function.

TSIDMNT

CA EPIC for z/VSE also provides the TSIDMNT catalog management utility to manage tape volumes.

Note: For more information on TSIDMNT, see the chapter DSN Catalog Management Utility.

Purging a Tape Volume

You can purge a tape volume from the DSN Catalog by using the Manage Tape Volumes panel.

Access

Choose M on the Primary Option Menu to display the Search Catalog panel. Type the desired volume serial number in the Volser field. Press ENTER to display the Manage Tape Volumes panel.

Manage Tape Volumes Panel

```
TSIO0400.r -------- CA-EPIC - Manage Tape Volumes -------------- USER=DWM
COMMAND ===>
                                                                More: NO
Dataset: DAN.DATASET                                          Version: 1

   Volser Reel  Creation    Expire   --- Last Access ---  Use  Vault  Slot
                  Date       Date    Date       Job      Count

 _ 123000    1 02/01/2009 04/30/2009 02/01/2009 DCMOTDRV     0     0     0









 C = Clean  I = Information  M = Move  P = Purge
 F1=Help  F3=Return  F5=Recall  F12=Exit
```

Procedure

1. Place a P in the input area to the left of the Volser field and press ENTER.
2. If prompted, type YES to confirm the purge.

Using TSIDMNT PURGE

CA EPIC for z/VSE also provides the PURGE command of the TSIDMNT catalog management utility to remove tape volume information from the DSN Catalog. See Chapter 3, "DSN Catalog Management Utility", for information on using the PURGE command.

Recording Tape Cleaning

You can record tape cleaning using either the Tape Volume Information or Manage Tape Volumes panels.

Access

Select M (for MANAGE) on the Primary Option Menu to display the Search Catalog panel. Enter a volume serial number and press ENTER to display the Manage Tape Volumes panel.

Tape Volume Information Panel

```
TSIO0400.r -------- CA-EPIC - Manage Tape Volumes -------------- USER=DWM
COMMAND ===>
                                                                More: NO
Dataset: DAN.DATASET                                          Version: 1

   Volser Reel  Creation    Expire   --- Last Access ---  Use  Vault  Slot
                  Date       Date    Date       Job      Count

 _ 123000    1 02/01/2009 04/30/2009 02/01/2009 DCMOTDRV     0     0     0









 C = Clean  I = Information  M = Move  P = Purge
 F1=Help  F3=Return  F5=Recall  F12=Exit
```

Procedure

1. Place a C in the input area to the left of the Volser field and press ENTER.
2. Press ENTER to record tape cleaning.

You can also record tape cleaning from the Tape Volume Information panel. Place C in the input area in front of the Volser field and press ENTER.

Using TSIDMNT CLEAN

CA EPIC for z/VSE also provides the CLEAN command of the TSIDMNT catalog management utility to record tape cleaning. See Chapter 3, "DSN Catalog Management Utility", for information on using the CLEAN command.

Retention and Scratching

Understanding Retention Concepts

Retention is how long a dataset is kept before it can be discarded. Retention is one of the most critical areas of information control. Scratching, or making a tape available for reuse, is the physical storage consequence of the expiration of a tape dataset's retention. All automated tape management systems must ensure that retention is monitored and controlled in order to protect information assets.

How CA EPIC for z/VSE Implements Retention

CA EPIC for z/VSE maintains a variety of retention criteria. Retention can expire only when all criteria are met. This implementation protects your datasets against inadvertent scratching.

There are two CA EPIC for z/VSE functions for removing datasets: DELETE and SCRATCH. Single-file tape volumes associated with scratched dataset versions become valid scratch tapes immediately after using either of these functions. Multi–file tape volumes become valid scratch tapes when all associated datasets are scratched.

The DELETE Function

Use the DELETE function to remove a dataset definition and all its versions from the DSN Catalog and from CA EPIC for z/VSE control. All tape volumes associated with the dataset definition become scratch tapes according to the rules for single–file and multi–file tape volumes and are listed in the DSN Catalog under the dataset name '** DSN DELETED **'. See the section on "Managing Dataset Definitions" for further information.

The SCRATCH Function

Use the SCRATCH function to scratch versions of a specific dataset. Associated tape volumes can be used immediately as output tapes if all datasets on the affected tape have been scratched. CA EPIC for z/VSE resequences version numbers as you scratch each version. So if you scratch version number 1, what used to be 2 becomes 1, 3 becomes 2, and so on.

Initiating the Scratch Process

You must run a scratch before CA EPIC for z/VSE expires a dataset or allows a tape to be reused. Scratches can be performed by manual catalog management, using either the Online Manager or the TSIDMNT utility.

The TSIDUTL scratch process searches the entire DSN Catalog for all dataset versions which have satisfied their retention criteria and expires them. However, you can limit the scratch to certain user IDs or system IDs.

Setting Dataset Definition Retention

When a dataset definition is created manually using the Define an EDD panel, a new set of retention criteria is defined which will apply to all future versions of that dataset.

Access

Select D (for DEFINE) from the Primary Option Menu to display the Define an EDD panel.

Define an EDD Panel

```
TSIO0103.D ----------- CA-EPIC - Define an EDD ----------------- USER=PROFILE
COMMAND ===>


  DEFINE

  Dataset ===> ____________________________________________
  Model   ===> ____________________________________________

  TAPE INFORMATION     DISK INFORMATION          DATASET INFORMATION
  Tape Mode  ===> __   Disk Pool   ===> ______   GEN or WRK     ===> ___
  Tape Pool  ===> __   Primary     ===> _______  Retention      ===> ____
  Vault Meth ===> ___  Secondary   ===> _______  Cycles         ===> ___
  Owner ID   ===> __   Max Sec     ===> ___      DLA            ===> ____
  Unlabeled  ===> _    Primary ALR ===> _______  BLKSZ( _ CISZ) ===> _____
  Revolving  ===> N    Auto Purge  ===> _        SYSnnn         ===> ___
  KMODEL     ===> _                             System ID      ===> __
                                                User ID        ===> __
  Comment    ===> _________________________     Password       ===>
  Prime DSN  ===> ________________               Encrypt Key    ===>



F1=Help  F3=Return  F5=Recall  F12=Exit
```

Procedure

Enter the desired criteria to create the new dataset definition.

Using TSIDMNT CATALOG

CA EPIC for z/VSE also provides the CATALOG command of the TSIDMNT catalog management utility to set dataset definition retention. See Chapter 3, "DSN Catalog Management Utility", for information on using the CATALOG command.

Modifying Dataset Definition Retention

When an existing dataset definition is modified using the Update an EDD panel, the existing retention criteria can be changed. The changes will apply to all subsequent versions of that dataset.

Access

Select M (for MANAGE) from the Primary Option Menu to display the Search Catalog panel. After you have entered your search criteria, press ENTER to display the Manage EDDs panel. Place the cursor in front of the dataset definition you wish to modify and type U (for UPDATE). Press ENTER to display the Update an EDD panel.

Update an EDD Panel

```
TSIO0104.e ----------- CA-EPIC - Update an EDD ----------------- USER=PROFILE
COMMAND ===>


  UPDATE

  Dataset: BG.BW.SYSIN.FILE

  TAPE INFORMATION     DISK INFORMATION          DATASET INFORMATION
  Tape Mode  ===> NO   Disk Pool   ===> POOL01   GEN or WRK     ===> WRK
  Tape Pool  ===> __   Primary     ===> 5        Retention      ===> 0
  Vault Meth ===> 0    Secondary   ===> 1        Cycles         ===> 0
  Owner ID   ===> __   Max Sec     ===> 0        DLA            ===> 0
  Unlabeled  ===> N    Primary ALR ===> _______  BLKSZ( _ CISZ) ===> MAX
  Revolving  ===> N    Auto Purge  ===> N        SYSnnn         ===> 0
  KMODEL     ===> L                             System ID      ===> __
                                                User ID        ===> __
  Comment    ===> ______________________________ Password       ===>




F1=Help  F3=Return  F5=Recall  F12=Exit
```

Procedure

To modify the dataset definition, change the values in the desired input fields.

Using TSIDMNT UPDATE

CA EPIC for z/VSE also provides the UPDATE command of the TSIDMNT catalog management utility to modify dataset definition retention. See Chapter 3, "DSN Catalog Management Utility", for information on using the UPDATE command.

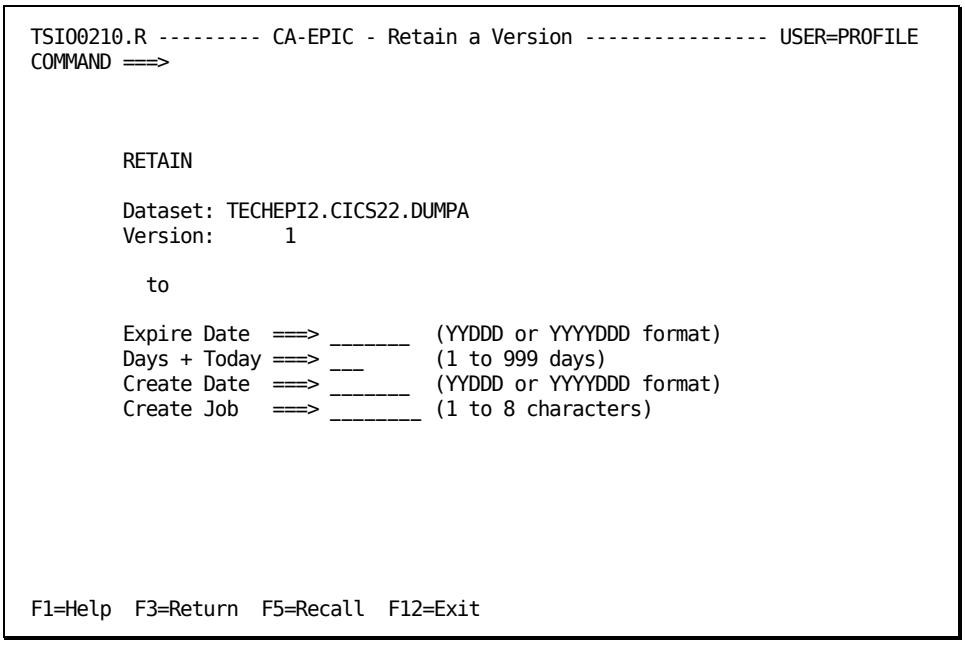
Setting or Modifying Dataset Version Retention

Use the Retain a Version panel to set or modify the expiration date, creation date, and creation job name of a version. If the version is not in active status, it is put in active status automatically.

Access

From the Manage Versions panel, place an H (for Retain) next to a dataset name and press ENTER. The dataset name and version number to be updated are displayed on the Retain a Version panel.

Retain a Version Panel



Procedure

1. Update any desired fields to modify dataset version retention:
2. Press ENTER. All data entered are verified for syntactic correctness and defaults added. Review the data before proceeding.
3. Press ENTER again to update the version retention or press PF3 to exit without updating.

Using TSIDMNT RETAIN

CA EPIC for z/VSE also provides the RETAIN command of the TSIDMNT catalog management utility to set and modify dataset version retention. See Chapter 3, "DSN Catalog Management Utility", for information on using the RETAIN command.

Cycling a Dataset Version

Cycling is a special purpose function for overriding the number of versions (cycles) retained for a dataset. Use it to take one version out of the normal retention cycle and reserve it for special retention. The version you cycle is:

- Removed from cycle retention for the dataset (it no longer counts as a version in the cycle)
- Retained until the expiration date you specify

For example, if you have told CA EPIC for z/VSE to keep three versions (cycles) of a particular dataset and you cycle one version, CA EPIC for z/VSE maintains four versions (the cycled version plus three normal versions) until the cycled version reaches its expiration date. All uncycled versions previous to the version you have just cycled are retained by cycle control only. Days Retention, Days Last Accessed Retention no longer apply. If the dataset is not retained under cycle control, those versions are eligible for scratch at the next run of TSIDUTL. Versions which have been cycled in the past are retained until their expiration date. Cycling a dataset has no effect on versions created after the cycled version. Cycling also allows you to change the creation date and the creation job name. These two fields have no effect on file retention.

From the Online Manager, the cycling function is executed using the Cycle a Version panel.

Difference Between CYCLE and RETAIN

Cycling looks very similar to the retain function. However, its effect is quite different. Using the retain function only affects the version you modify. Using the cycling function affects the version you modify and all previous versions.

Access

From the Manage Versions panel, place a Y next to a version and press ENTER.

Cycle a Version Panel

```
TSIO0211.Y ---------- EPIC - Cycle a Version ------------------  USER=A35LP031
COMMAND ===>


        CYCLE

        Disk Dataset: DISKGDG.DISK.1
        Version:      1

          to

        Expire Date  ===> _______  (YYDDD or YYYYDDD format)
        Days + Today ===> ___      (1 to 999 days)
        Create Date  ===> _______  (YYDDD or YYYYDDD format)
        Create Job   ===> ________ (1 to 8 characters)



F1=Help  F3=Return  F5=Recall  F12=Exit
```

Procedure

1. Update any desired fields to cycle the version.
2. Press ENTER. All data entered are verified for syntactic correctness and defaults added. Review the data before proceeding.
3. Press ENTER again to cycle the version or press PF3 to exit without cycling.

Using TSIDMNT CYCLE

CA EPIC for z/VSE also provides the CYCLE command of the TSIDMNT catalog management utility to cycle dataset versions. See Chapter 3, "DSN Catalog Management Utility", for information on using the CYCLE command.

Scratching a Dataset Version

There are several different status types that affect when a dataset version becomes eligible for scratch. Datasets in open status are eligible for scratch only after they have been closed with the Online Manager. All other versions are eligible for scratch only after all of the retention criteria specified for the dataset version has been satisfied. Scratching a dataset version using the Online Manager or the TSIDMNT SCRATCH command expires a specific dataset version you name. That version is scratched whether it has satisfied its retention criteria or not.

Access

Select M (for MANAGE) from the Primary Option Menu to display the Search Catalog panel. Enter the search criteria for the EDDs you wish to maintain, and press ENTER to display the Manage Versions panel. From this list, type V in front of the EDD whose versions you want to maintain. Press ENTER to display the Manage Versions panel.

Manage Versions Panel

```
TSIO0200.D ---------- CA-EPIC - Manage Versions ---------------- USER=PROFILE
COMMAND ===>
                                                                More: >


  Dataset Name              Version Number-of First  --- Creation ---
   Width: 25                        Vols/Exts Volser Date        Time

_ TECHEPI2.CICS22.DUMPA         1        1    EPC229 04/26/2009 11:52
_ TECHEPI2.CICS22.DUMPB         1        1    EPC182 04/15/2009 15:17









 A = Append C = Close  E = Ext/Reels F = Release H = Retain   I = Information
 J = Lock   K = Unlock M = MCatalog  S = Scratch T = Transfer Y = Cycle
 F1=Help  F3=Return  F5=Recall  F11=Right  F12=Exit
```

Procedure

Place an S (for Scratch) in front of the version you want to scratch and press ENTER.

Using TSIDMNT SCRATCH

CA EPIC for z/VSE also provides the SCRATCH command of the TSIDMNT catalog management utility to scratch dataset versions. See Chapter 3, "DSN Catalog Management Utility", for information on using the SCRATCH command.

Chapter 3: DSN Catalog Management Utility

CA EPIC for z/VSE provides the TSIDMNT utility program for managing the DSN Catalog. The same functions can also be performed using the CA EPIC for z/VSE Online Manager.

This section contains the following topics:

Command Summary

Command	Function
ADD	Add a version of a controlled dataset to the DSN Catalog
CAPACITY	Display DSN Catalog utilization statistics
CATALOG	Define a new EDD in the DSN Catalog (in Online Manager, use the DEFINE command)
CLEAN	Reset tape usage, error counts, and clean date
CLOSE	Change a version's status from "open" or "conditionally cataloged" to "active"
CYCLE	Override the cycle control of a dataset
DELETE	Remove a dataset and all of its versions from the DSN Catalog (tapes are set to scratch status)
DEQ	Dequeue the DSN Catalog
DISPLAY	Display DSN Catalog entries (EDDs, versions, and volumes)
ENQ	Enqueue the DSN Catalog
KMODEL	Catalog model encryption key (KEKL) data
LOCK	Lock an EDD for output or a version from input
MCATALOG	Specify that a tape dataset can be accessed by more than one dataset name
MOVE	Change the vault and slot location of a tape volume

Command	Function
OFFLINE	Prevent a disk volume from being accessed (VSE only)
ONLINE	Enable access to a disk volume (VSE only)
PASSWORD	Specify or reset system password for catalog update
PURGE	Remove an EDD or versions from the DSN Catalog
QUIESCE	Make a disk volume ineligible for use in creating controlled datasets (VSE only)
RELEASE	Enable a locked EDD or version to be used once
RENAME	Change the name of a controlled dataset
RETAIN	Change a version's expiration date, creation date, or creation job, and make the version active
SCRATCH	Scratch a version of a dataset
TRANSFER	Move a version from one dataset to another
UNLOCK	Unlock a locked EDD or version
UPDATE	Change the attributes of an CA EPIC for z/VSE dataset definition (EDD)

CA EPIC for z/VSE System Requirements

The following are the various CA EPIC for z/VSE system requirements.

JCL Requirements

TSIDMNT is designed for use within production job streams, to scratch datasets as necessary and perform other periodic catalog management such as MCATALOG or CYCLE. TSIDMNT functions are submitted as shown below:

```
[// OPTION LOG]
// EXEC TSIDMNT
control-statements
   .
   .
/*
```

Print Options

Print output is controlled using the JCL LOG option. If the LOG option is active, all processed maintenance is printed on SYSLST. If the NOLOG option is in effect, only invalid commands are printed. If your system log option is NOLOG (the usual default), an // OPTION LOG statement is required prior to // EXEC TSIDMNT if a list of submitted commands is desired.

TSIDMNT Commands

Command Syntax

TSIDMNT functions follow this syntax:

command ['*dataset-name*'] [*parameter-expressions*]

The command must always be first. It can be specified in full or abbreviated to its first three letters. For example, you can use either DISPLAY or DIS for the DISPLAY command.

dataset-name is positional. When used, it must always follow the command and be enclosed in apostrophes.

parameter-expressions can follow in any order, except where noted otherwise. A parameter–expression consists of a parameter and its value(s) that are joined by an equal sign (=). Parameter expressions must be separated from each other by a comma or blank spaces. Numeric values are automatically zero–filled on the left if required, so it is not necessary to use leading zeros.

CA EPIC for z/VSE Statements

Statements can be entered in columns 1 through 71, and a statement can span up to three (3) cards. To continue a statement, place a non–blank character in column 72.

Example

```
ADD 'INVENTRY.FILE',                                          X
SER='IN0192,IN2323,IN2987,IN9870,IN4573,IN1029,IN3049'
```

PASSWORD

If your installation has used the PASSWD configuration option to invoke DSN Catalog update protection, you must use the PASSWORD command to enter the required password. The DSN Catalog password is required for all commands except the following:

- CAPACITY
- DISPLAY
- HELP
- PASSWORD

Submit the PASSWORD statement before any protected operation. The password is not printed on SYSLST if the command name (PASSWORD or PAS) begins in column 1. The password is cleared when the program ends.

Example

```
PAS LOLLIPOP
```

ADD

Use the ADD command to add a new version to a controlled dataset. To create a new CA EPIC for z/VSE dataset definition (EDD) for a previously uncontrolled dataset, use the CATALOG command.

ADD can be used to add a new version that was created outside CA EPIC for z/VSE control or to recatalog a deleted version. Once a dataset is controlled, all new versions created under CA EPIC for z/VSE are added to the DSN Catalog automatically.

CA EPIC for z/VSE resequences version numbers as you add new versions. For example, if you add a new version, the previous version 1 becomes 2, 2 becomes 3, and so on. To add more than one version in a specific sequence, make sure to add the oldest version first and the newest version last.

It is possible for some versions of a tape dataset to have standard labels, while other versions are unlabeled. When you add a tape version to the DSN Catalog, its entry automatically receives the label type in the EDD. If a particular tape's label type differs from the EDD's, use the UPDATE command to modify the EDD's label type temporarily. Use ADD to add the new tape version. Then change the EDD back to the previous label type.

If an Automated Cartridge Library (ACL) is supported, ADD sets the tape volume to "non–scratch" status in the ACL inventory.

Important!

- If the VTOC entry for a disk dataset has been deleted, ADD does not re–create it. As long as the VTOC entry is deleted, that disk space is free and can be re–used for another dataset. If this occurs, do the following:
- Ensure that the disk space has not already been reused.
- Use the CA EPIC for z/VSE PACKSCAN ADD function to re–create the VTOC entry as soon as possible.
- Use the ADD command to add the version to the DSN Catalog.

Syntax

```
ADD  'dataset-name'
     [ ,VER=n ]
     [ ,GEN=n ]
     [ ,FIL=n ]
     [ ,BLK=n ]
     [ ,OFL=1 ]
     [ ,MOD=n ]
```

ADD Parameters

Parameter	Required?	Valid Entry	Default
dataset-name	Yes	1 to 44 characters	None
Specify only one of the following two (required):			
SER (or ESER)		1 to 6 characters	None
VOL (or EVOL)		1 to 6 characters	None
Specify only one of the following two (optional):			
VER		1 to 999	1
GEN		1 to 999999	None
FIL	No	2 to 255	None
BLK	No	1 to 65535|MAX|OPT	None
OFL	No	1	None
MOD	No	2 to 4 characters	None

Descriptions

dataset-name specifies the name of the dataset to which the new version belongs.

SER/ESER (Tape versions only) specifies the tape volume serial number(s) on which this version is written. . Specify the volume serial numbers of multi–volume datasets in reel sequence order. Use the ESER parameter instead of SER when a version is written on more than 23 tapes. You can also use ESER to add tapes to a version that is already in the DSN Catalog.

VOL/EVOL (VSE disk versions only) specifies the extents the dataset occupies. volser specifies the volume serial number of the volume that contains the extent.

start-track specifies the address at which the extent starts.

blocks specifies the number of blocks or tracks used for the extent.

VOL subparameters are positional. If there are multiple extents, repeat the subparameters in extent sequence. Use EVOL instead of VOL if the version has more than 10 extents. Also use it to add extents to a version that is already in the DSN Catalog. Supply extent information just as for VOL.

Note: For both VOL and EVOL, TSIDMNT does not access the disk volume(s) to determine whether the version exists or whether its extents match the extents you specify.

VER can be used to assign a specific version number for the dataset. If VER is not specified, VER=1 is assumed.

GEN can be used to specify the absolute generation number for the dataset. A generation number assigned to an existing version is not allowed, but you can specify a previously used generation number from a version that has been deleted. To assign a new (never used) generation number, it must be greater than the next generation number due to be created. For example, if the highest current generation number (usually version 1) is 000120, specify a new generation number of 000122 or greater. The new version is always version 1.

FIL specifies the file sequence number if this version of a tape dataset is not the first dataset in a multi–dataset group.

BLK specifies the block size of this version of the dataset, if you wish. MAX sets the block size to track capacity for disk datasets or to 32K for tape datasets. OPT sets the block size to half track for disk or 16K for tape.

OFL=1 can be used to identify the version as an offload version. While we do not recommend purging offload datasets, the OFL parameter can be used to assign offload status to a disk dataset that has been added to the catalog.

MOD (Tape versions only) can be used to specify the catalog density for the new version. Valid specifications are 2 to 4 byte strings as specified with the TSIDMNT 'CATALOG' command.

Note: For more information on valid specifications, see the MOD parameter of the CATALOG command.

Examples

This example adds version number 3 to the PAYROLL.MASTER dataset. Version 3 is on tapes PY1234 and PY2764, with a block size of 10000.

```
ADD 'PAYROLL.MASTER',VER=3,BLK=10000,SER=PY1234,PY2764
```

This example adds version number 1 to the INVENTRY.237 dataset. Version 1 is on tape E93837.

```
ADD 'INVENTRY.237',SER=E93837
```

CAPACITY

Use the CAPACITY control statement to display DSN Catalog usage. This is displayed as both the number of active/free records, and as a percentage.

Syntax

```
CAPACITY  (no parameters)
```

Results

The following is an example of the CAPACITY display.

```
EP119  TOTAL       CATALOG UTILIZATION           INDEX UTILIZATION
EP119 CAPACITY    ACTIVE     FREE    PCT        ACTIVE     FREE    PCT
EP119    4560        959     3601    21            560     4000    12
```

CATALOG

Use the CATALOG command to create a new CA EPIC for z/VSE dataset definition (EDD) in the DSN Catalog, and to define the attributes to be associated with that dataset name. After the EDD has been created, you can catalog versions of the dataset with the ADD command.

Syntax

```
CATALOG      'dataset-name'
             [ ,AGP=YES|NO ]
             [ ,ALR=n ]
             [ ,BLK=n|MAX|OPT]
             [ ,CIZ=n]
             [ ,COM='comment' ]
             [ ,COP='existing-dataset-name' ]
             [ ,CYC=n ]
             [ ,DEC=NO|key ]
             [ ,DLA=n ]
             [ ,EX2=n ]
             [ ,EXT=n ]
             [ ,KMODEL=kmodel ]
             [ ,LBL=YES|NO ]
             [ ,MOD=<see Supported Tape Modes in Appendix B> ]
             [ ,NSU=''prime-dataset-name'']
             [ ,OWN=xx ]
             [ ,POL=poolname ]
             [ ,PWD=password ]
             [ ,RET=n|PERM ]
             [ ,REV=NO|YES ]
             [ ,SEC=NO|n ]
             [ ,SID=NO|xx ]
             [ ,SIZ=SMALL|LARGE ]
             [ ,SYSnnn ]
             [ ,TPL=x|NO ]
             [ ,UID=NO|xx ]
             [ ,VLT=n ]
             [ ,WRK=NO|YES ]
```

General Parameters

Parameter	Required?	Valid Entry	Default
dataset-name	Yes	1 to 44 characters	None
BLK	No	NO|1 to 65535|MAX|OPT	NO
COM	No	1 to 30 characters	None
COP	No	1 to 44 characters	None
CYC	No	1 to 999	None
DLA	No	1 to 9999	None
PWD	No	1 to 8 characters	NO
RET	No	1 to 9999|PERM	None
SID	No	NO|2 characters	NO
SYS*nnn*	No	000 to 253	000
UID	No	NO|2 characters	NO
WRK	No	YES|NO	NO

Descriptions

dataset-name is used to locate an CA EPIC for z/VSE dataset definition (EDD) in the DSN Catalog. It must be unique. Generation dataset names can be a maximum of 35 characters long.

VSE Only: When adding CPU and partition–independent datasets, the dataset name must contain the specific CPU and partition IDs. For example, to use the dataset name '===.SORTWK1' on a DLBL in a job that will run on CPU 4 in the BG partition, specify the dataset name 'BG4.SORTWK1'.

BLK specifies the blocksize to use for output.

COM can be used to make notes about a dataset's use. The comment can be deleted by specifying COM=' '

COP can be used to define a new dataset that is similar to a previously cataloged dataset. You must specify the COP parameter immediately following the dataset name. Any other parameters specified before the COP parameter are ignored. To make modifications to the copied attributes, you can specify additional parameters after the COP parameter.

PWD can be used to protect the dataset from unauthorized UNLOCK or RELEASE. If you specify NO, no password protection is provided. This parameter does not lock the dataset. To do so, use the LOCK command.

RET, CYC, DLA, WRK (Retention Days, Cycles Retention, DLA Retention, and Generation or Work) work together to determine dataset retention. These are some of the most important attributes of any dataset, and they must be carefully selected.

- RET specifies the number of days to retain the dataset after its creation date.
- CYC specifies the number of cycles (versions) to retain.
- DLA specifies the number of days to retain the dataset after its last access date.

WRK specifies whether this is a generation dataset (WRK=NO) or a non–generation dataset (WRK=YES).

SID and UID (VSE Only) can be used to group your dataset with other datasets for catalog functions. If you specify NO, the ID is inactive.

SYS*nnn* (VSE Only) specifies a logical unit number to be used for the dataset, if you wish. Using this parameter inhibits LUB allocation automatically. If you specify SYS000, the parameter is deactivated.

Tape Parameters

Parameter	Required?	Valid Entry	Default
DEC	No	1 to 8 characters	None
KMOdel	No	1 to 8 characters	$DEFAULT (if defined)
LBL	No	YES|NO	YES
MOD	No	<see Supported Tape Modes in Appendix B>	None
NSU	No	1 to 17 characters	None
OWN	No	1 to 2 characters	None
REV	No	YES|NO	NO
SIZ	No	LARGE|SMALL	LARGE
TPL	No	1 character	NO
VLT	No	1 to 255	None

DEC is used to enter a software encryption key if you want tape versions to be encrypted at creation. The encryption key cannot be added, changed, or deleted for a dataset definition that has existing versions. This is not a password, and you will not be asked to re-enter it.

KMOdel is used to specify the name of the KEKL Model Definition (KMODEL) that will supply encryption key labels for this DSN should it be written to a 3592 tape drive with an encryption density. The KMOdel name must be 8 characters or less and must already have been defined in the EPIC catalog (using the 'KMODEL' command), or this parameter will be rejected.

If a KMOdel with the name '$DEFAULT' is defined, the KMOdel parameter may be omitted. The KEKL stored in '$DEFAULT' will be used automatically whenever a 3592E drive is written to using one of the encryption densities.

Example:

```
// EXEC TSIDMNT
CAT 'TEST.PRIME.DSN'  WRK=NO CYC=3  MOD=1BW5  KMO=MYKMODEL

/*
```

LBL specify NO for unlabeled, YES for standard labeled.

MOD specify the tape density. For a list of supported tape densities, see 'Supported Tape Modes' in Appendix B.

NSU specifies that tape versions of the dataset are part of a multi–dataset group. Specify the name of first dataset on the tape (the prime dataset).

Enter the prime dataset name when the dataset is first cataloged. CA EPIC for z/VSE does not allow you to add a prime dataset name to an existing CA EPIC for z/VSE dataset definition (EDD).

OWN specifies an Owner ID, if you wish. If you also specify a tape pool, make sure the Owner ID falls within the tape pool range. If you specify NO, no Owner ID is used.

REV regulates tape selection and can be used for backup tapes. Revolving tapes and datasets have an exclusive relationship. A revolving tape volume can only be used for its dataset, and a revolving tape dataset can only be written on tape volumes assigned to it. A revolving tape dataset cannot be used in an Automated Cartridge Library (ACL).

SIZ specifies a tape size.

TPL specifies a tape pool ID.

VLT specifies a vault method number, if you want tape vaulting for this dataset.

Disk Parameters

Disk information is only meaningful to CA EPIC for z/VSE, but it can be updated from CA EPIC for CMS.

Parameter	Required?	Valid Entry	Default
AGP	No	Yes or NO	NO
ALR	No	1 to 99999	None
CIZ	No	NO|1 to 32767	None
EXT	No	1 to 999999	DEFEXT
EX2	No	0 to 999999	50% of primary
POL	No	1 to 6 characters	None
SEC	No	0 to 255	255

AGP specifies whether dataset retention is controlled by the Automatic Generation Purge feature.

ALR specifies the number of records to be written to a version, if you wish. Specifying allocation by logical records overrides the primary extent size specified by the EXT parameter if both parameters are specified.

EXT specifies the primary extent size in blocks or tracks. The minimum size is 3 tracks for CKD devices, or the value specified in the CIZ parameter for FBA devices. If the DEFEXT configuration option is set to *YES* for your installation, you can use that as a default primary extent, if you wish.

EX2 specifies the secondary extent size, if you wish. Remember that the secondary extent defaults to 50% of the size of the primary extent. The minimum extent size is 3 tracks for CKD devices, or the value specified in the CIZ parameter for FBA devices.

POL specifies a disk pool name if disk versions are to be created. Use disk pool names defined in the TSIDPOL macro. If the DEFPOL configuration option is set to **YES** for your installation, use that as a default disk pool, if you wish.

SEC specifies the maximum number of secondary extents to be allowed, if you wish. NO (the default) allows 255 secondary extents. To inhibit all secondary allocations, specify zero (0).

Examples

The following example catalogs the PAYROLL.MASTER dataset, copying dataset attributes from the 'AP.MASTER' dataset and changing the dataset password to PASSWORD.

```
CATALOG 'PAYROLL.MASTER',COP='AP.MASTER',PWD='PASSWORD'
```

The following example catalogs the BALANCE.BUDGET dataset, with day retention of 3 and days last accessed retention of 3. Disk datasets are written in disk pool POOLA1 with a primary extent size of 20 and a maximum of 3 secondary extents.

```
CAT 'BALANCE.BUDGET',RET=3,DLA=3,POL=POOLA1,EXT=20,SEC=3
```

The following example results in CA EPIC for z/VSE retaining three versions of the BALANCE.BUDGET dataset at all times. When a new version is created, the oldest version is automatically deleted at end–of–job.

```
CAT 'BALANCE.BUDGET',CYC=3,AGP=YES...
```

CLEAN

Use the CLEAN command to record tape cleaning. CLEAN sets the use and error counts for cleaned tapes to zero.

Syntax

```
CLEAN  SER=volser1,volser2,...
```

Parameters

Parameter	Required?	Valid Entry	Default
volsern	Yes	1 to 6 characters	None

SER specifies the volume serial numbers of the cleaned tapes.

Example

The following example records the cleaning of tapes A38472, A94874, and 857594

```
CLEAN SER=A38472,A94874,857594
```

CLOSE

CA EPIC for z/VSE only accesses active versions of a dataset. Datasets that are in "open" or "conditionally cataloged" status cannot be processed, and they become eligible for scratch differently from active versions. To access or save an open or conditionally cataloged dataset, you must close it with the CLOSE command.

CLOSE only changes the DSN Catalog status of a version. It does not perform close processing, such as writing tape or disk labels.

If an Automated Cartridge Library (ACL) is supported, CLOSE sets the tape volume to "non–scratch" status in the ACL inventory.

Although CLOSE can be used to set a tape volume to "non–scratch" status in the CA EPIC for z/VSE Catalog, we recommend using RETAIN to do this.

Syntax

```
CLOSE  'dataset-name' [ ,SER=volser ] [ ,GEN=nnn ]
```

Parameters

Parameter	Required?	Valid	Default
dataset-name	Yes	1 to 44 characters	None
SER	No	1 to 6 characters	None
GEN	No	1 to 999999	None

dataset-name specifies the name of the dataset.

SER specifies the volume serial number of a tape version.

GEN specifies the generation number of a disk dataset.

If neither SER or GEN is specified, CA EPIC for z/VSE closes the most recently created version of *dataset-name*.

Example

The following example closes the BATCH.BALANCE dataset.

```
CLOSE 'BATCH.BALANCE'
```

CYCLE

The CYCLE command is a special purpose command for overriding the normal Cycles Retention (CYC) of a dataset. The CYCLE command causes one version to be removed from the Cycles Retention and reserved for special retention. For example, you might use it to isolate the correct balance sheet from a series of trial balance runs.

CYCLE looks very similar to the RETAIN command. However, its effect is quite different. Using RETAIN only affects the version you modify. Using CYCLE affects the version you modify and all previous versions.

The version you CYCLE is:

- removed from cycle retention for the dataset (that is, it no longer counts as a version in the cycle).
- retained until the expiration date you specify.

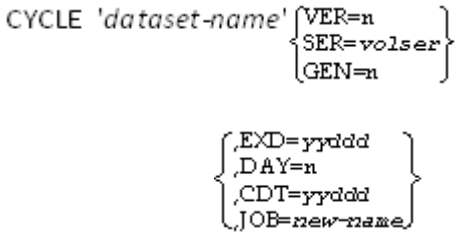
For example, if the dataset has CYC=3 retention and you CYCLE one version, CA EPIC for z/VSE maintains four versions (the cycled version plus three normal versions) until the cycled version reaches its expiration date.

All uncycled versions created before the version that is cycled are retained by cycle control only. Retention days and DLA retention no longer affect cycled versions. If the dataset does not have cycle control, uncycled versions are eligible for scratch.

Versions which have been cycled in the past are retained until their expiration date.

The CYCLE command also allows you to change the creation date and the creation job name. These two fields have no effect on dataset retention.

Syntax



Parameters

Parameter	Required?	Valid Entry	Default
dataset-name	Yes	1 to 44 characters	None
Specify only one of the following three (required):			
VER		1 to 999	None
SER		1 to 6 characters	None
GEN		1 to 999999	None
Specify at least one of the following parameters (required):			
EXD		1 to 99365	None
DAY		1 to 999	None
CDT		1 to 99365	None
JOB		1 to 8 characters	None

dataset-name specifies the dataset name.

Specify either VER, SER, or GEN:

- VER specifies the version number.
- SER specifies the volume serial number of the first (or only) volume of a tape version.

GEN specifies the generation number of a disk version.

At least one of the following parameters must be included in the command statement. Specify dates in Julian format (2–digit or 4–digit year followed by 3–digit day, for example January 1, 1991 = 91001 or 1991001).

EXD specifies the new expiration date of the version that was cycled. If neither an expiration date nor period retention is specified, the version is retained according to the period retention established in the dataset defaults. For dates after the year 1999, the format must be *yyyyddd*.

DAY specifies period retention. CA EPIC for z/VSE calculates the expiration date as today's date plus the number of days entered.

CDT specifies a new creation date. CDT has no effect on dataset expiration.

JOB changes the name of the job that created this version. JOB has no effect on dataset expiration.

Example

The following example cycles the AGED.TRIAL.BALANCE dataset, setting apart version 3 with a 7–day retention and changing its creating job name to TRIALBAL. All versions earlier than version 3 (for example, versions 4–9) are now retained by cycle retention only.

```
CYCLE 'AGED.TRIAL.BALANCE',VER=3,DAY=7,JOB=TRIALBAL
```

DELETE

Use the DELETE command to remove a dataset and all its versions from the DSN Catalog and from CA EPIC for z/VSE control. The CA EPIC for z/VSE dataset definition (EDD) is deleted. All tapes associated with the dataset become scratch tapes and are listed under the dataset name '** DSN DELETED **' on CA EPIC for z/VSE reports and in Online Manager displays. Disk space allocated to the dataset is freed.

If an Automated Cartridge Library (ACL) is supported, DELETE sets the tape volume to "scratch" status in the ACL inventory.

Important! Software encrypted datasets must not be deleted from the DSN Catalog if you ever plan to use them again as input. Once an encrypted version's EDD is deleted, it cannot be decoded and is forever unreadable.

Syntax

```
DELETE  'dataset-name'
```

Parameters

Parameter	Required?	Valid Entry	Default
dataset-name	Yes	1 to 44 characters	None

dataset-name specifies the dataset name to be deleted.

Example

This example deletes the dataset AP.BATCH.BALANCE.TEST.

```
DEL 'AP.BATCH.BALANCE.TEST'
```

DEQ

Use DEQ to dequeue the CA EPIC for z/VSE DSN Catalog after it has been enqueued with the ENQ command.

In CA EPIC for CMS, this command is only meaningful when sharing an CA EPIC for z/VSE DSN Catalog.

Syntax

```
DEQ  (no parameters)
```

DISPLAY

Use the DISPLAY command to display entries in the DSN Catalog. The display lists all cataloged information about the dataset and each of its versions. The summary display contains one line of data for each version of each dataset.

If a version resides on multiple volumes, each volume has its own line in the display. Scratched tape versions continue to be displayed until their tapes are used to write other datasets.

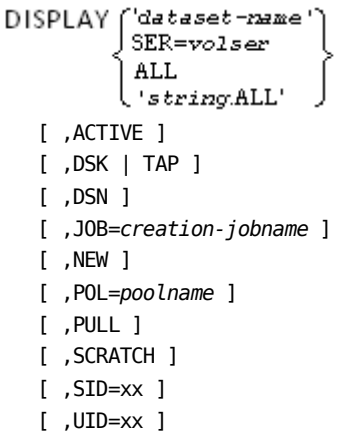
For a detailed explanation of the displays, see Appendix A. Most of the items listed in the reports are self–explanatory. However the ST (file status) and the TYPE (file type) headings require immediate explanation.

A code for dataset status is displayed under the column heading ST. The valid status codes are:

- (blank) - active version
- C - conditional catalog status
- L - active version in a locked status
- M - MCAT version
- O - open status
- S - scratch status

There are three dataset types listed under the TYPE heading: TAPE, DISK, and EMPTY. TAPE and DISK indicate the media on which the dataset is written. EMPTY signifies that the dataset has been cataloged, but it contains no versions. A dataset is empty when it is first cataloged, before any versions are created. A dataset can also be empty if all its versions have been scratched and all tapes have been reused.

Syntax


DISPLAY
'dataset-name'
SER=volser
ALL
'string.ALL'
[,ACTIVE]
[,DSK | TAP]
[,DSN]
[,JOB=creation-jobname]
[,NEW]
[,POL=poolname]
[,PULL]
[,SCRATCH]
[,SID=xx]
[,UID=xx]

Parameters

Parameter	Required?	Valid Entry	Default
Specify one of the following four (required):			
dataset-name		1 to 44 characters	None
SER		1 to 6 characters	None
ALL		ALL	None
string.ALL		1 to 44 characters	None
ACTIVE	No	ACTIVE	OFF
Specify only one of the following two (optional):			
DSK		DSK	OFF
TAP		TAP	OFF
ACTIVE	No	ACTIVE	OFF
DSN	No	DSN	OFF
JOB	No	1 to 8 characters	None
NEW	No	NEW	OFF
POL	No	1 to 8 characters	None
PULL	No	PULL	OFF
SCRATCH	No	SCRATCH	OFF
SID	No	2 characters	None
UID	No	2 characters	None

Primary Parameters

dataset-name requests a detailed display of the dataset named.

SER requests a detailed display of the contents of the specified tape volume.

ALL requests a summary display of all datasets and entries in the DSN Catalog. ALL is the default when you specify any optional parameters.

'*string*.ALL' requests a summary display of all datasets beginning with the specified character string.

Secondary Parameters

The following parameters are valid with the ALL and string.ALL parameters.

ACTIVE displays only active versions.

DSK displays only disk versions.

DSN displays dataset names only (omits dataset summaries).

JOB displays the latest version of all datasets that were created by the specified jobname. It also displays any OPEN or conditionally cataloged datasets created by that job since the latest version was created.

NEW displays all datasets that have been created since the last execution of the TSIDUTL DSPLY NEW function.

POL displays only datasets from the specified disk pool.

PULL displays only the latest version. Place PULL after the ALL or string.ALL parameter, as in this example: DIS 'PB.PAY.ALL',PULL

SCRATCH displays only scratch tapes.

SID displays only those datasets with the specified system ID. If specified, the SID and UID parameters must be the last parameters in the command.

TAP displays only tape versions.

UID displays only those datasets with the specified user ID. If specified, the SID and UID parameters must be the last parameters in the command.

Example

The following example produces a summary display of all active datasets with the creation job name of AR8000.

```
DISPLAY ALL,ACTIVE,JOB=AR8000
```

The following example produces a summary display of all the scratched versions of datasets that begin with the character string SDR3.

```
DIS 'SDR3.ALL',SCRATCH
```

The following example displays a detailed display of the AP.BALANCE dataset.

```
DIS 'AP.BALANCE'
```

ENQ

Use ENQ to enqueue the CA EPIC for z/VSE DSN Catalog and prevent access to it by jobs running in other partitions or CPUs. To enqueue a DSN Catalog that is shared by two or more CPUs, MULTCPU=YES must be specified as a configuration option.

In CA EPIC for CMS, this command is only meaningful when sharing an CA EPIC for z/VSE DSN Catalog.

Syntax

```
ENQ  (no parameters)
```

KMODEL

Use the KMODEL to store Key Encryption Key Label (KEKL) data in the DSN catalog for use when carts are encrypted with the IBM 3592 hardware encryption facility. Although the syntax of CA EPIC for z/VSE's keywords differ slightly, the KMODEL command stores the same information that is specified in the native VSE 'KEKL' Job Control Statement.

For additional information about the meaning of the KEKL fields, see the KEKL description in the IBM 'System Control Statements' manual.

Syntax

```
KMOdel  ‘kmodel’ | ‘$DEFAULT’
        [ ,KL1=’n’,KM1=L|H]  [ ,KL2=’n’,KM2=L|H]
        [ ,DELete]
        [ ,DISplay]
```

The KMODEL command and all parameters may be abbreviated to as few as 3 characters, and the command may be continued up to 3 total statements. To continue a command, place an 'X' in column 72 of the card that is to be continued. Individual parameters may not cross a card boundary.

Define a KMODEL

To define a new KMODEL (KEKL Model Definition)

```
KMODEL 'mykmodel’ KL1=’n’, KM1=’L’, KL2=’n’, KM2=H
```

Where:

- 'kmodel' (required) - 8 characters or less, enclosed in apostrophes.
- KL1 (required) - Up to 64 characters of key label data. It must be enclosed in apostrophes, and imbedded apostrophes are not allowed (an EPIC restriction). The first character within the apostrophes must be non-blank (IBM restriction). Identical to the IBM JCL 'KEKL1=' parameter.
- KM1 (required) - 1 byte which must be either 'L' or 'H', signifying the encoding method used by the key manager with the key label named in the 'KL1' parm. It may be enclosed in apostrophes but they are not required. Identical to the IBM JCL 'KEM1=' parameter.
- KM2 and KL2 are optional and allow the specification of a second key label and encoding method. Although they are optional, if one is coded, they both must be. Identical to the IBM JCL 'KEKL2' and 'KEM2=' parms.
- ‘$DEFAULT’ – Optionally, a default KMODEL may be defined to the catalog. The default KEKL information stored here will be used each time a dataset intended for hardware encryption is created, provided the dataset is defined with an encryption mode and has no KMODEL defined for it.

Example

```
  // EXEC TSIDMNT
  KMODEL  'MYKMODEL',KM1=L,KM2=H,                                          X
 KL1='ABBBBBBBBBBBBBBBBBBBBBBBBBBBBBCDEEEEEEEEEEEEEEEEEEEEEEEEEEEEF' X
KL2='GHHHHHHHHHHHHHHHHHHHHHHHHHHHHHIJKKKKKKKKKKKKKKKKKKKKKKKKKKKKL'
 /*
```

Display KMODEL Information

To display the information stored in an existing KMODEL:

```
KMOdel 'name' ,DISplay
```

Where:

- 'name' (required) - 8 characters or less, enclosed in apostrophes. This is the name of the existing KMODEL that is to be displayed.
- DISplay - If specified, must be the only parm on this commmand. It will print the encryption key data stored in the KMODEL.

Example

```
// EXEC TSIDMNT
KMO 'MYKMODEL',DIS
/*
```

Delete a KMODEL

```
DELete
```

If specified, must be the only parm on this command. The named KMODEL will be deleted from the catalog.

Example

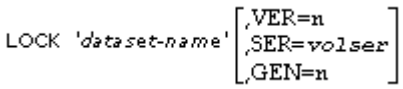
```
// EXEC TSIDMNT
KMODEL 'MYKMODEL',DELETE
/*
```

LOCK

Use the LOCK command to lock a dataset or version.

If you do not specify a version, volume serial number, or generation number, the EDD is locked for output, which means that CA EPIC for z/VSE does not permit additional versions to be created until you release or unlock it. If you do specify a version, volume serial number, or generation number, only that version of the dataset is locked for input, which means that it cannot be read until you release or unlock it.

Syntax



Parameters

Parameter	Required?	Valid Entry	Default
dataset-name	Yes	1 to 44 characters	None
Specify only one of the following three (optional):			
VER		1 to 999	None
SER		1 to 6 characters	None
GEN		1 to 999999	None

dataset-name specifies the name of the dataset to be locked.

To lock a specific version, specify either VER, SER, or GEN:

- VER specifies the version number.
- SER specifies the volume serial number of the first (or only) volume of a tape version.
- GEN specifies the generation number of a disk version.

Example

The following example locks all versions of the PAYROLL.MASTER dataset.

```
LOCK 'PAYROLL.MASTER'
```

The following example locks only version 1 of the 'PAYROLL.MASTER' dataset.

```
LOC 'PAYROLL.MASTER',VER=1
```

MCATALOG

Use the MCATALOG command to specify that a tape dataset can be accessed by more than one dataset name. The new dataset names are known as MCAT aliases. After a version has been MCATALOGed, it is accessible by both its original dataset name and its MCAT alias(es). The dataset is retained until all retention criteria from all associated CA EPIC for z/VSE dataset definitions (EDDs) have been satisfied. MCATALOG can be used to create special retention for a tape version. Subdatasets cannot be MCATALOGed. When a display of an MCATALOGed version is requested by volume serial number, the original dataset record is displayed.

Syntax

```
MCAT   'from-dataset-name' {,VER=from-ver|SER=from-ser},
       'to-dataset-name' [ ,VER=to-ver ]
```

Parameters

Parameter	Required?	Valid Entry	Default
from-dataset-name	Yes	1 to 44 characters	None
One of the following two must be specified:			
from-ver		1 to 999	None
from-ser		1 to 6 characters	None
to-dataset-name	Yes	1 to 44 character	None
to-ver	No	1 to 999	1

MCAT parameters are positional.

from-dataset-name specifies the name of the original dataset.

You must specify either SER or VER:

- *from-ver* specify the version number of the original dataset.

from-ser specify the volume serial number of the first (or only) volume of the original dataset.

to-dataset-name specifies the MCAT alias.

to-ver specifies the version number of the MCAT alias.

Example

Make version 2 of the PAYROLL.MASTER dataset to be also version 1 of the PAYROLL.PERM dataset, so it can be accessed using either dataset name.

```
MCA 'PAYROLL.MASTER',VER=2,'PAYROLL.PERM',VER=1
```

MOVE

Use the MOVE command to change the vault and slot location of a tape volume.

If an Automated Cartridge Library (ACL) is supported and the tape is moved *from* the main vault (Vault 0), the volume is ejected from the ACL at the time the MOVE statement is processed.

Important! Vaulting reports do not reflect changes specified by MOVE commands. If any dataset on the volume is controlled by automatic vaulting, the vaulting program will undo your MOVE and return the tape to its normal place in the rotation.

Syntax

```
MOVE  SER=volser,vault,slot
```

Parameters

Parameter	Required?	Valid Entry	Default
volser	Yes	1 to 6 characters	None
vault	Yes	1 to 255	None
slot	Yes	1 to 9999	None

volser specifies the volume serial number of the tape to be moved.

vault specifies the number of the tape's new vault.

slot specifies the number of the tape's new slot.

vault and *slot* are positional parameters which must be entered in the sequence shown.

Example

The following example moves tape AP3847 to vault 6, slot 234.

```
MOV SER=AP3847,6,234
```

OFFLINE (VSE Only)

Use the OFFLINE command to prevent a disk volume from being accessed by CA EPIC for z/VSE if you must dismount the volume or "device down" the drive. Once a volume is in OFFLINE status, CA EPIC for z/VSE will not use it to allocate either output or input datasets. OFFLINE cannot disable volume access across machines, so you must issue the OFFLINE command from all machines that have access to the device.

Use the ONLINE command to restore CA EPIC for z/VSE access to a volume that has been declared OFFLINE. See ONLINE and QUIESCE.

Syntax

```
OFFLINE volser
```

Parameters

Parameter	Required?	Valid Entry	Default
volser	Yes	1 to 6 characters	None

volser specify the serial number of the volume you wish to remove from CA EPIC for z/VSE access.

Example

The following example removes the disk volume SYSWK1 from CA EPIC for z/VSE access.

```
OFF SYSWK1
```

ONLINE (VSE Only)

Use the ONLINE command to enable CA EPIC for z/VSE to access a disk volume for both input and output datasets. If a disk volume has been placed in OFFLINE or QUIESCE status, use ONLINE to restore access to the volume. ONLINE cannot enable a volume across machines, so you must use the ONLINE command in all machines in which the OFFLINE command was used for the volume.

See OFFLINE and QUIESCE.

Syntax

```
ONLINE volser
```

Parameters

Parameter	Required?	Valid Entry	Default
volser	Yes	1 to 6 characters	None

volser specify the serial number of the volume you want to place online.

Example

The following example gives CA EPIC for z/VSE access to disk volume SYSWK2.

```
ONL SYSWK2
```

PURGE

Use the PURGE command to remove CA EPIC for z/VSE dataset definitions (EDDs), versions, and tape volume serial numbers from the DSN Catalog. With PURGE, you can delete an entire dataset and all its versions (tape and disk) or simply remove a tape volume and the versions it contains.

Datasets which are part of a multi–dataset group cannot be purged until all other datasets in the group are scratched or deleted. In addition, all versions must be in closed status before they can be purged.

If an Automated Cartridge Library (ACL) is supported, PURGE sets the tape volume to "scratch" status in the ACL inventory.

Important!

- Do not use this command for routine deletion of datasets.
- When a tape volume is purged, CA EPIC for z/VSE does not attempt to determine whether it is a valid scratch tape. For this reason, use this command only when a physical tape volume is destroyed or permanently removed from your installation.
- Encrypted dataset versions must not be purged from the DSN Catalog. Once information about an encrypted version is deleted, it cannot be decoded and is forever unreadable.

Syntax

Parameters

Parameter	Required?	Valid Entry	Default
Specify one of the following two (required):			
dataset-name		1 to 44 characters	None
SER		1 to 6 characters	None

dataset-name specifies a dataset name when you wish to delete an EDD and all associated versions from the DSN Catalog. The dataset is no longer controlled.

SER specifies a tape volume serial number when you wish to remove that serial number from the DSN Catalog, but leave the dataset under CA EPIC for z/VSE control. If the volume serial number contains non–alphanumeric characters, enclose it in apostrophes.

Examples

The following example purges PAYROLL.TABLE and all associated tape volumes.

```
PURGE 'PAYROLL.TABLE'
```

The following example purges volume PR3746 and the version written on it.

```
PUR SER=PR3746
```

QUIESCE (VSE Only)

QUIESCE makes disk volumes ineligible for the creation of controlled datasets. CA EPIC for z/VSE will not allocate output datasets on a volume that has been quiesced. Quiesced volumes are still eligible for CA EPIC for z/VSE input processing and for uncontrolled processing. QUIESCE cannot disable a volume across machines, so you must use QUIESCE on all machines that have access to the volume.

Use the ONLINE command to remove a volume from quiesced status. Use the OFFLINE command to remove a volume from both input and output processing.

Syntax

```
QUIESCE volser
```

Parameters

Parameter	Required?	Valid Entry	Default
volser	Yes	1 to 6 characters	None

volser specify the serial number of the volume you want to quiesce.

Example

The following example quiesces the disk volume SYSWK1.

```
QUI SYSWK1
```

RELEASE

Use the RELEASE command to unlock, for one use only, EDDs or versions that have been placed in LOCK status (the UNLOCK command allows unlimited access). If you do not indicate a specific version, the EDD is released for one use as output, so that you can create one more version of the dataset. The new version is placed in locked status automatically.

If a specific version is indicated, that version is released for input, and CA EPIC for z/VSE allows you to read it only once. After a released version is read, it is automatically put back into locked status. RELEASE requires prior use of the LOCK command.

Syntax

```
RELEASE 'dataset-name' [,specifier-expression] [,PWD=password]
```

Parameters

dataset-name is required and can be up to 44 characters long.

specifier-expression is optional. If used, it must be one of the following:

- VER=*version-number*
- SER=*volser*
- GEN=*generation-number*

where *version-number* is an integer from 1 to 999. *volser* identifies the first (or only) volume of a tape version and can be up to 6 characters. *generation-number* is an integer from 1 to 999999.

password must be used if the dataset is password–protected. It can be up to 8 characters.

Examples

The following example releases the PAYROLL.MASTER dataset, supplying its password HARRY, and permits one more version to be created.

```
RELEASE 'PAYROLL.MASTER',PWD=HARRY
```

The following example releases the version of the PAYROLL.MASTER dataset which starts on volume serial number PR9485, and permits that version to be read once as input.

```
REL 'PAYROLL.MASTER',SER=PR9485
```

RENAME

The RENAME command allows you to change the name of a cataloged dataset. All existing versions of the original dataset are referenced by the new dataset name, and all dataset characteristics remain the same.

If the dataset is a disk dataset, its VTOC entries are also renamed.

Important! Do not rename prime datasets or offload datasets. Results are unpredictable.

Syntax

```
RENAME 'old-dataset-name','new-dataset-name'
```

Parameters

Parameter	Required?	Valid Entry	Default
old-dataset-name	Yes	1 to 44 characters	None
new-dataset-name	Yes	1 to 44 characters	None

old-dataset-name specifies the name of the dataset you want to rename.

new-dataset-name specifies the new dataset name.

Example

The following example changes the name of the dataset CURRENT.MASTER to APRIL.MASTER.

```
REN 'CURRENT.MASTER','APRIL.MASTER'
```

RETAIN

Use the RETAIN command to alter a version's expiration date, creation date, or creation job name. If the version is not in active status, it is put in active status automatically.

If an Automated Cartridge Library (ACL) is supported, RETAIN sets the tape volume to "non–scratch" status in the ACL inventory. RETAIN also changes the status of a version from scratch, open or conditional catalog to active status.

Syntax

```
RETAIN 'dataset-name',specifier-expression,modifier-expression[,0F2]
```

Parameters

dataset-name is required.

specifier-expression is required. It must be one of the following:

- VER=nnn where nnn is an integer from 1 to 999.
- SER=xxxxxx where xxxxxx identifies the first (or only) volume of a tape version and can be up to 6 characters.
- GEN=nnnnnn where nnnnnn is an integer from 1 to 999999.

modifier-expression is required. It must be one of the following:

- EXD=yyyyddd or EXD=CDT+nnn

  EXD=yyyyddd

  where yyyyddd is the new (julian) expiration date of the version. For dates after the year 1999, it must be in the format yyyyddd (not yyddd).

  Example: EXD=2008001 <= expires Jan 1, 2008

  EXD=CDT+nnn

  where the constant *CDT* indicates that the expiration will be calculated as a number of days after the version's creation date. NNN is the 3 digit number of days to be added to the create date to determine the new expiration date.

  Example: EXD=CDT+005 <= expires 5 days after creation

  EXD=CTM=hhmm

  where the constant CTM indicates that the version’s creation time is to be updated to the hour and minute (hhmm) specified.

  Example: EXD=CTM=1430 <= changes creation time to 14:30 hours.

- DAY=*retention-days* where retention days specifies period retention. CA EPIC for z/VSE calculates the expiration date as today's date plus the number of retention days.
- CDT=*creation-date* where creation date is the Julian date on which the version was created.
- JOB=*jobname* where *jobname* is the name of the job that created this version of the dataset.

OF2 is an optional keyword parameter. If used, it goes at the end of the RETAIN statement. It can be used to identify the RETAINed version as having FAVER2 offload format. While we do not recommend purging any offload datasets, the OF2 parameter can be used to assign FAVER2 offload format to a tape after it has already been ADDed to a dataset with the OFL parameter.

Examples

The following example retains version 3 of the AGED.TRIAL.BALANCE dataset. Version 3 is closed and is retained for seven days from today, and its creating job name is changed to TRIALBAL.

```
RETAIN 'AGED.TRIAL.BALANCE',VER=3,DAY=7,JOB=TRIALBAL
```

The following example retains the version of PAYROLL.UNEDITED.TRANSACTIONS which starts on volume PR5867. The version is closed and expires on November 23, 1997.

```
RET 'PAYROLL.UNEDITED.TRANSACTIONS',SER=PR5867,EXD=97327
```

SCRATCH

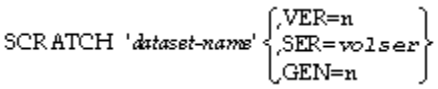
Use the SCRATCH command to scratch a version of a dataset. The version must be in active status to be scratched. If it is in open or conditionally cataloged status, you must first use the CLOSE command to make it active.

If an Automated Cartridge Library (ACL) is supported, SCRATCH sets the tape volume to "scratch" status in the ACL inventory.

In CA EPIC for z/VSE, you can use the TSIDMNT SCRATCH command in a job stream to scratch disk datasets without supplying the dataset's password. As long as the current job name and partition are the same as the creating job's name and partition, CA EPIC for z/VSE disregards its password protection.

Important! Encrypted dataset versions must not be scratched. If an encrypted version is scratched, it cannot be decoded and is forever unreadable.

Syntax



Parameters

Parameter	Required?	Valid Entry	Default
dataset-name	Yes	1 to 44 characters	None
Specify only one of the following three (required):			
VER		1 to 999	None
SER		1 to 6 characters	None
GEN		1 to 999999	None

dataset-name specifies the dataset name.

You must specify either VER, SER, or GEN:

VER specifies the version number.

SER specifies the volume serial number of the first (or only) volume of a tape version.

GEN specifies the generation number of a disk version.

Scratching by volume serial number is not valid for NSU subdatasets or MCAT aliases.

Examples

The following example scratches version 1 of the INVENTRY.SANDIEGO.3847 dataset.

```
SCRATCH 'INVENTRY.SANDIEGO.3847',VER=1
```

The following example scratches the version of the INVENTRY.SANDIEGO.3847 dataset which starts on volume IN4658.

```
SCR 'INVENTRY.SANDIEGO.3847',SER=IN4658
```

TRANSFER

Use the TRANSFER command to move a version from one dataset to another. The version transferred is no longer associated with its original dataset name. A transferred version retains all of its original characteristics, except for version number.

VSE Only: If the version is a disk dataset, the associated VTOC entry is also renamed.

Syntax

```
TRANSFER 'from-dataset-name', from-specifier,

         'to-dataset-name'[,VER=to-version]
```

Parameters

from-dataset-name is required.

from-specifier is required. It must be one of the following:

- VER=*version-number* where *version-number* is an integer from 1 to 999.
- SER=*volser* where *volser* is the serial number of the first (or only) volume of a tape version.

to-dataset-name is required.

to-version is optional. If specified, it must be an integer from 1 to 999.

Example

The following example transfers version 7 of the IRVINE.INVENTRY dataset to version 1 of the SANDIEGO.INVENTRY dataset.

```
TRA  'IRVINE.INVENTRY',VER=7,'SANDIEGO.INVENTRY'
```

UNLOCK

Use the UNLOCK command to permanently unlock a locked dataset or version. To unlock a dataset or version for one use only, use the RELEASE command instead. Use of UNLOCK requires prior use of the LOCK command.

To permit unlimited access to a specific version, use UNLOCK with a version qualifier. To permit creation of an unlimited number of new versions, use UNLOCK without a version qualifier.

Syntax

```
UNLOCK  'dataset-name',dataset-specifier [,PWD=password]
```

Parameters

dataset-name is required.

dataset specifier is required. It must be one of the following:

- VER=*version-number*
- SER=*volser*
- GEN=*generation-number*

password must be used if the dataset has been password–protected.

Examples

The following example unlocks the PAYROLL.MASTER dataset, supplying its dataset password HARRY. Unlimited versions of the dataset can now be created.

```
UNLOCK 'PAYROLL.MASTER',PWD=HARRY
```

The following example unlocks version 6 of the INVENT.958575 dataset. It can now be read as input an unlimited number of times.

```
UNL 'INVENT.958575',VER=6
```

UPDATE

Use UPDATE to change the characteristics associated with an EDD in the DSN Catalog. The updated characteristics will only affect versions created after the update. To update previously created versions, use CLOSE, CYCLE, or RETAIN.

The parameters available on the UPDATE command are nearly identical to the CATALOG command parameters. However, the prime dataset name and encryption password parameters cannot be updated once the dataset is cataloged. To change processing for a dataset once these parameters have been entered, you must create a new dataset name for it in your DSN Catalog and change existing JCL.

To remove the following parameters if they are already cataloged, set them to NO:

- disk pool
- kmodel
- password
- system ID
- tape pool
- user ID

Syntax

```
UPDATE      'dataset-name'
            [ ,AGP=YES|NO ]
            [ ,ALR=n ]
            [ ,BLK=n|MAX|OPT]
            [ ,CIZ=n]
            [ ,COM='comment' ]
            [ ,CYC=n ]
            [ ,DLA=n ]
            [ ,EX2=n ]
            [ ,EXT=n ]
            [ ,KMOdel=kmodel ]
            [ ,LBL=YES|NO ]
            [ ,MOD=<see Supported Tape Modes in Appendix B> ]
            [ ,OWN=xx ]
            [ ,POL=poolname|NO ]
            [ ,PWD=password|NO ]
            [ ,RET=n|PERM ]
            [ ,REV=NO|YES ]
            [ ,SEC=NO|n ]
            [ ,SID=xx|NO ]
            [ ,SIZ=SMALL|LARGE ]
            [ ,SYSnnn ]
            [ ,TPL=x|NO ]
            [ ,UID=xx|NO ]
            [ ,VLT=n ]
            [ ,WRK=NO|YES ]
```

General Parameters

Parameter	Required?	Valid Entry	Default
dataset-name	Yes	1 to 44 characters	None
BLK	No	NO|1 to 65535|MAX|OPT	NO
COM	No	1 to 30 characters	None
COP	No	1 to 44 characters	None
CYC	No	1 to 999	None
DLA	No	1 to 9999	None
PWD	No	1 to 8 characters	NO
RET	No	1 to 9999|PERM	None
SID	No	NO|2 characters	NO
SYS*nnn*	No	000 to 253	000
UID	No	NO|2 characters	NO
WRK	No	YES|NO	NO

Descriptions

dataset-name is used to locate an CA EPIC for z/VSE dataset definition (EDD) in the DSN Catalog. It must be unique. Generation dataset names can be a maximum of 35 characters long.

VSE Only: When adding CPU and partition–independent datasets, the dataset name must contain the specific CPU and partition IDs. For example, to use the dataset name '===.SORTWK1' on a DLBL in a job that will run on CPU 4 in the BG partition, specify the dataset name 'BG4.SORTWK1'.

BLK specifies the blocksize to use for output.

COM can be used to make notes about a dataset's use. The comment can be deleted by specifying COM=' '

COP can be used to define a new dataset that is similar to a previously cataloged dataset. You must specify the COP parameter immediately following the dataset name. Any other parameters specified before the COP parameter are ignored. To make modifications to the copied attributes, you can specify additional parameters after the COP parameter.

PWD can be used to protect the dataset from unauthorized UNLOCK or RELEASE. If you specify NO, no password protection is provided. This parameter does not lock the dataset. To do so, use the LOCK command.

RET, CYC, DLA, WRK (Retention Days, Cycles Retention, DLA Retention, and Generation or Work) work together to determine dataset retention. These are some of the most important attributes of any dataset, and they must be carefully selected.

- RET specifies the number of days to retain the dataset after its creation date.
- CYC specifies the number of cycles (versions) to retain.
- DLA specifies the number of days to retain the dataset after its last access date.

WRK specifies whether this is a generation dataset (WRK=NO) or a non–generation dataset (WRK=YES).

SID and UID (VSE Only) can be used to group your dataset with other datasets for catalog functions. If you specify NO, the ID is inactive.

SYS*nnn* (VSE Only) specifies a logical unit number to be used for the dataset, if you wish. Using this parameter inhibits LUB allocation automatically. If you specify SYS000, the parameter is deactivated.

Tape Parameters

Parameter	Required?	Valid Entry	Default
DEC	No	1 to 8 characters	None
KMOdel	No	1 to 8 characters	$DEFAULT (if defined)
LBL	No	YES|NO	YES
MOD	No	<see Supported Tape Modes' in Appendix B>	None
NSU	No	1 to 17 characters	None
OWN	No	1 to 2 characters	None
REV	No	YES|NO	NO
SIZ	No	LARGE|SMALL	LARGE
TPL	No	1 character	NO
VLT	No	1 to 255	None

DEC is used to enter an encryption key if you want tape versions to be encrypted at creation. The encryption key cannot be added, changed, or deleted once it is entered in an CA EPIC for z/VSE dataset definition. This is not a password, and you will not be asked to re–enter it.

KMOdel is used to specify the name of the KEKL Model Definition (KMODEL) that will supply encryption key labels for this DSN should it be written to a 3592 tape drive with an encryption density. The KMOdel name must be 8 characters or less and must already have been defined in the CA EPIC for z/VSE catalog (with the 'KMODEL' command), or this parameter will be rejected.

If a KMOdel with the name '$DEFAULT' is defined, the KMOdel parameter may be omitted. The KEKL stored in '$DEFAULT' will be used automatically whenever a 3592 drive is written to using one of the encryption densities.

To remove an existing association of a dataset with a KMODEL, specify 'KMODEL=NO'.

Example

```
// EXEC TSIDMNT
UPD 'TEST.PRIME.DSN',MOD=1BW5,KMO=MYKMODEL

/*
```

LBL specify NO for unlabeled, YES for standard labeled.

MOD specify the tape density. For a list of supported tape densities, see 'Supported Tape Modes' in Appendix B.

NSU specifies that tape versions of the dataset are part of a multi–dataset group. Specify the name of first dataset on the tape (the prime dataset).

Enter the prime dataset name when the dataset is first cataloged. CA EPIC for z/VSE does not allow you to add a prime dataset name to an existing CA EPIC for z/VSE dataset definition (EDD).

OWN specifies an Owner ID, if you wish. If you also specify a tape pool, make sure the Owner ID falls within the tape pool range. If you specify NO, no Owner ID is used.

REV regulates tape selection and can be used for backup tapes. Revolving tapes and datasets have an exclusive relationship. A revolving tape volume can only be used for its dataset, and a revolving tape dataset can only be written on tape volumes assigned to it. A revolving tape dataset cannot be used in an Automated Cartridge Library (ACL).

SIZ specifies a tape size.

TPL specifies a tape pool ID.

VLT specifies a vault method number, if you want tape vaulting for this dataset.

Disk Parameters

Disk information is only meaningful to CA EPIC for z/VSE, but it can be updated from CA EPIC for CMS.

Parameter	Required?	Valid Entry	Default
AGP	No	Yes or NO	NO
ALR	No	1 to 99999	None
CIZ	No	NO|1 to 32767	None
EXT	No	1 to 999999	DEFEXT
EX2	No	0 to 999999	50% of primary
POL	No	1 to 6 characters	None
SEC	No	0 to 255	255

AGP specifies whether dataset retention is controlled by the Automatic Generation Purge feature.

ALR specifies the number of records to be written to a version, if you wish. Specifying allocation by logical records overrides the primary extent size specified by the EXT parameter if both parameters are specified.

EXT specifies the primary extent size in blocks or tracks. The minimum size is 3 tracks for CKD devices, or the value specified in the CIZ parameter for FBA devices. If the DEFEXT configuration option is set to *YES* for your installation, you can use that as a default primary extent, if you wish.

EX2 specifies the secondary extent size, if you wish. Remember that the secondary extent defaults to 50% of the size of the primary extent. The minimum extent size is 3 tracks for CKD devices, or the value specified in the CIZ parameter for FBA devices.

POL specifies a disk pool name if disk versions are to be created. Use disk pool names defined in the TSIDPOL macro. If the DEFPOL configuration option is set to *YES* for your installation, use that as a default disk pool, if you wish.

SEC specifies the maximum number of secondary extents to be allowed, if you wish. NO (the default) allows 255 secondary extents. To inhibit all secondary allocations, specify zero (0).

Examples

The following example updates the PAYROLL.EDITED.TRANSACTIONS dataset, changing its vault method to number 3.

```
UPDATE 'PAYROLL.EDITED.TRANSACTIONS',VLT=3
```

The following example updates the SR3.INVENTRY.45768 dataset, changing its default block size to 10,000 and changing its default density to 1600 bpi.

```
UP 'SR3.INVENTRY.45768',BLK=10000,MOD=3
```

Chapter 4: Accessing the DSN Catalog

This chapter discusses several ways to access the DSN Catalog.

This section contains the following topics:

Displaying DSN Catalog Entries (TSIDUTL DSPLY)

The DSPLY command creates reports from the data maintained in the DSN Catalog. Reports can be created displaying data associated with all the datasets, all the version entries, specific datasets, or specific versions which match specified criteria. These reports can also be produced in different sort sequences.

DSPLY produces reports listing all datasets or groups of datasets in the DSN Catalog. In most cases, you receive a full report of each dataset's catalog entry. The CA EPIC for z/VSE dataset definition can be made bold by overprinting for easier viewing.

Note: For report examples, see *TSIDUTL Reports* in Appendix A of this guide.

UPSI Values

Setting	Action
1xxxx	Lists dates in European format (DD/MM/YY)
x1xxx	Suppresses dataset name overprinting. Use this setting if the report will only be viewed at a terminal or if the report will be printed on a printer that does not support overprinting.
xxx1x	Suppresses catalog update when using the NEW subparameter
xxxx1	Bypasses CA EPIC for CMS Archive datasets

Dataset Requirements

SORTWK1 is required if you use the SEQ parameter.

Syntax

```
DSPLY dataset-qualifier [,report-qualifier]
         [,SEQ=sort-sequence]
```

Parameters

dataset-qualifier is required. It must be one of the following:

ACTIVE

Reports only active datasets.

ALL

Reports all datasets in the DSN Catalog.

DISK

Reports only on disk datasets.

DSN=*'dataset-name'*

Reports only the specified dataset.

NEW

Reports dataset versions created since the last time the DSPLY NEW report was run. NEW can be used to track the datasets created within a given time frame. NEW updates the catalog so that at the next run it can distinguish the new datasets. If you do not want the update, execute TSIDUTL with UPSI xxx1x.

PULL

Reports the latest version of all datasets.

PULL,DSN='*dataset-name*'

Reports the latest version of the named dataset.

SCRATCH

Reports scratch tapes.

TAPE

Reports only on tape datasets.

TAPEPOOL

Reports the tapes in each tape pool. Page breaks occur between tape pools.

VAULT

Reports the tapes stored in each offsite vault. Page breaks occur between vaults.

report-qualifier is optional. If used, it must be one of the following:

Values	Reports Only...
OWNER	Datasets with the specified owner ID
SYSID	Datasets with the specified system ID
TPL	Tapes within the specified pool
USERID	Datasets with the specified user ID
VLT	datasets stored in the specified vault.

SEQ=*sort-sequence* defines the sort sequence of your report. Valid sort sequences are listed below. SEQ=*sort-sequence* is an optional parameter. If used, it must be the last parameter specified. If you do not use it, datasets are listed in the order they are found in the index. You might want to sort the index by using the INDEX function before running reports without the SEQ parameter.

If SEQ is used, *sort-sequence* must be one of the following:

Value	Sorts By...
CJOB	Creation jobname
CDATE	Creation date
DSN	Dataset name
ERROR	Total errors
SERIAL	Tape volume serial number
VAULT	Vault and slot numbers
TAPEPOOL	Tape pool ID
XDATE	Expiration date

Examples

The following job produces a report listing all datasets belonging to the System ID TS and the User ID MR. They are listed in the order they are found in the catalog's index.

```
// JOB DISPLAY DSN CATALOG
// EXEC TSIDUTL
DSPLY ALL,SYSID=TS,USERID=MR
/*
/&
```

The following job produce a report listing all vaulted tapes. Tapes are listed in dataset name sequence within the vault.

```
// JOB DISPLAY DSN CATALOG
// ASSGN SYS001,150
// DLBL SORTWK1,'===.SORTWK1,(X)'
// EXTENT SYS001
// EXEC TSIDUTL
DSPLY VAULT,SEQ=DSN
/*
/&
```

Accessing DSN Catalog Records (TSIDAC2)

You may want to access DSN Catalog information with your own programs to produce your own reports or for other reasons. The TSIDAC2 subroutine is provided for this purpose and is located in the CA EPIC for z/VSE installation library as member name TSIDAC2.OBJ.. A sample COBOL program (TSIDACC2) is located in the CA EPIC for z/VSE installation library as member name TSIDACC2.C.

TSIDAC2 is not reentrant and should not be used in an online CICS environment.

DSN Catalog Record Structure

Each dataset's DSN Catalog entry consists of two types of records:

- A single dataset definition (EDD) record (called the *master record*).
- Individual version-level records (called *detail records*). One record exists for each version, whether it is valid, open, conditionally cataloged, or scratched. If the version is multi-volume, there is a detail record for each additional volume.

The DSN Catalog dataset also includes an index. Each time a new record is added to the DSN Catalog file, another index entry is added to the end of the index area. This means that information obtained sequentially may not be in alphabetical order. To re-sequence the index, use the INDEX function of TSIDUTL.

About Sequential and Random Access

In random access, TSIDAC2 returns the master record for a specific dataset the first time it is called for that dataset. On each subsequent call, TSIDAC2 returns a detail record for that dataset.

In sequential access, TSIDAC2 returns each record for a single dataset in order, and then moves to the next dataset in the DSN Catalog's index sequence and begins returning its records, one at a time. Detail records are always returned current version first, then each successively older version.

TSIDAC2 Calling Procedure

TSIDAC2 returns DSN Catalog information for any program language or report writer.

TSIDAC2 is a relocatable module in the CA EPIC for z/VSE installation library. The library that contains TSIDAC2 must be available when you catalog your calling program.

The procedure for using the TSIDAC2 subroutine is:

Step 1

Define the storage area to be used for the parameter list during a TSIDAC2 call.

This storage area begins with a one-byte field that is used as a return code from the TSIDAC2 subroutine. The rest of the storage area contains the data being passed to and received from the TSIDAC2 subroutine.

The TSIDAC2.C and TSIDAC2.A copybooks contain the layout of the storage area, and are in the CA EPIC for z/VSE installation library. Use one of these copybooks to define the storage area.

Step 2

Place the appropriate argument in the USR-KEY field to tell TSIDAC2 what function to perform. Function calls are described in detail in the next section.

Step 3

Call TSIDAC2 from your program.

This call requires that a parameter list be passed to TSIDAC2. This parameter list consists solely of the storage area discussed in Step 1.

Step 4

Following the TSIDAC2 call, you should check the return code that is passed back in the USER-RETURN-CODE field.

Return Codes

The following return codes are used:

0 indicates that the requested function has been successfully completed and the data has been returned in USER-CATALOG-DATA.

1 indicates that end-of-file has been reached. This return code is issued when reading the entire DSN Catalog via Sequential Access or when reading all versions of a specific dataset using Random Access.

2 indicates that the requested data was not found in the DSN Catalog during a Random Access call.

3 indicates that the CA EPIC for z/VSE system is not active.

Data is returned only if the return code is 0.

Example

This is an example of the COBOL coding that can be used to invoke the TSIDAC2 subroutine.

```
WORKING-STORAGE SECTION.
01 ACC-RECORD COPY TSIDAC2.
PROCEDURE DIVISION.
    MOVE (search argument) TO USR-KEY.
     CALL 'TSIDAC2' USING ACC-RECORD.
     IF USER-RETURN-OK GO TO PROCESS-THE-RECORD.
     IF USER-RETURN-EOF GO TO END-OF-FILE.
     IF USER-RETURN-NOT-ACTIVE GO TO SYSTEM-DOWN.
     IF USER-RETURN-NOT-FOUND GO TO NOT-FOUND.
PROCESS-THE-RECORD.
```

TSIDAC2 Search Arguments and Calls

The following pages detail the specific procedures necessary to gain different types of access to the DSN Catalog using TSIDAC2. Implement these procedures as steps 2 and 3 of the procedure outlined in the previous section.

For the sake of discussion, we use the COBOL copybook (TSIDAC2.C) to name the data fields described in this section, and all examples are coded in COBOL.

Random Access

To request a complete dataset entry

1. Move the dataset name to the USR-KEY field.
2. Issue multiple calls to TSIDAC2.

Example

```
PROCEDURE DIVISION.
     .
     .
     .
DSN-LOOP.
     MOVE 'DATA.SET.NAME.A' TO USR-KEY.
     CALL 'TSIDAC2' USING ACC-RECORD.
     IF USER-RETURN-CODE = 1 GO TO END-OF-DSN
     (process the catalog information here)
     GO TO DSN-LOOP.
```

To request a master record only

1. Move the dataset name to the USR-KEY field.
2. Call TSIDAC2 just once.

Example

```
PROCEDURE DIVISION.
     .
     .
     .
     MOVE 'DATA.SET.NAME.A' TO USR-KEY.
     CALL 'TSIDAC2' USING ACC-RECORD.
     .
     .
     .
```

To request a detail record by volume serial number

1. Move a hex 02 to USR-KEY-POS-1 and move the six-digit volume serial number to USR-KEY-SER.
2. Call TSIDAC2.

Example

```
WORKING-STORAGE SECTION.
01 HEX-NUMBER   PIC S9(4) COMP VALUE +2.
01 HEX-REDEF REDEFINES HEX-NUMBER.
   05 FILLER    PIC X.
   05 HEX02     PIC X.
      .
      .
      .
PROCEDURE DIVISION.
      .
      .
      .
     MOVE HEX02 TO USR-KEY-POS-1.
     MOVE 'EP3452' TO USR-KEY-SER.
     CALL 'TSIDAC2' USING ACC-RECORD.
      .
      .
      .
```

Sequential Access

To read the entire DSN Catalog sequentially

1. Move spaces to the USR-KEY field.
2. Issue multiple calls to TSIDAC2.

Example

```
PROCEDURE DIVISION.
      .
      .
      .
SEQUENTIAL-READ.
     MOVE SPACES TO USR-KEY.
     CALL 'TSIDAC2' USING ACC-RECORD.
     IF USER-RETURN_CODE = 1
              GO TO END-OF-CATALOG.
(process the catalog information here)
GO TO SEQUENTIAL-READ.
```

To specify a starting point for sequential processing

1. Move the starting dataset name into the USR-KEY field.
2. Call TSIDAC2.
3. Move spaces to the USR-KEY field.
4. Issue multiple calls to TSIDAC2.

Example

```
PROCEDURE DIVISION.
     .
     .
     .
     MOVE 'DATA.SET.NAME.B' TO USR-KEY.
     CALL 'TSIDAC2' USING ACC-RECORD.
SEQUENTIAL-READ.
     MOVE SPACES TO USR-KEY.
     CALL 'TSIDAC2' USING ACC-RECORD.
     IF USER-RETURN_CODE = 1
              GO TO END-OF-CATALOG.
(process the catalog information here)
GO TO SEQUENTIAL-READ.
```

Terminating Sequential Access

Sequential access to the DSN Catalog is normally terminated at end-of-file. In certain circumstances, however, you may wish to terminate sequential processing prior to end-of-file and allow your programs to change between sequential and random processing. No records are returned when performing this type of call.

To terminate sequential processing prior to end-of-file

1. Move high-values (hex FF) to the USR-KEY field.
2. Call TSIDAC2.

Example

```
PROCEDURE DIVISION.
     .
     .
     .
     MOVE HIGH-VALUES TO USR-KEY.
     CALL 'TSIDAC2' USING ACC-RECORD.
     .
     .
     .
```

Releasing GETVIS

The first time TSIDAC2 is called, it acquires partition GETVIS storage for various purposes. If you want to release this storage:

1. Move low-values (hex 00) to USR-KEY.
2. Call TSIDAC2.

Example

```
PROCEDURE DIVISION.
     .
     .
     .
     MOVE LOW-VALUES TO USR-KEY.
     CALL 'TSIDAC2' USING DSN-RECORD.
     .
     .
     .
```

ADDRESS EPIC

The ADDRESS EPIC command provides an interface to CA EPIC for z/VSE from CA's Global Subsystem (GSS) product.

Return Codes

Code	Explanation
0	Function completed normally
-1	ADDRESS EPIC command is invalid
-2	CA EPIC for z/VSE is not active
-4	Invalid keyword on ADDRESS EPIC command line
-5	Invalid keyword value for numeric operand
-6	STEM keyword is missing
-8	ID keyword is missing
-9	Period is missing from STEM operand
-10	STEM name is invalid
-11	Insufficient getvis for VBUF work area
-12	GRXEXCOM error return
-13	GETDSA failure

Command Forms

The following form returns the current CA EPIC for z/VSE release. The stem n will be in the format vv.rr.mm.

```
ADDRESS EPIC EXTRACT RELEASE STEM n
```

The following form returns the master record for the dataset specified by ii. If the dataset name has embedded blanks, it must be enclosed in single quotes (') or double quotes ("). If stem n.0 is 0, the requested dataset was not found.

```
ADDRESS EPIC EXTRACT ID ii STEM n
```

The following form returns the master record for datasets whose dataset name starts with the character string ii. The stem n.0 indicates the total number of records returned as stem variables. If n.0 is 0, there were no datasets found whose name starts with the specified character string ii. The COUNT keyword is required describing the maximum number of records to return. A specification of COUNT * will return all matching records. The SKIP keyword may optionally be used specify the number of records that will be bypassed before returning them as stem variables.

```
ADDRESS EPIC EXTRACT ID ii STEM n. COUNT c SKIP s
```

The following form returns the master record for all datasets in the CA EPIC for z/VSE catalog. The stem n.0 indicates the total number of records returned as stem variables. If n.0 is 0, there were no datasets found. The COUNT keyword may optionally be used to specify the maximum number of records to return. A specification of COUNT *, the default when ID is *, will return all records. The SKIP keyword may optionally be used to specify the number of records that will be bypassed before returning them as stem variables.

```
ADDRESS EPIC EXTRACT ID * STEM n. COUNT c SKIP s
```

The following form returns the detail record for the dataset ii. The stem n.0 indicates the total number of records returned as stem variables. If n.0 is 0, there were no detail records found for the dataset ii or the datasets is not found in the CA EPIC for z/VSE catalog. The COUNT keyword may optionally be used to specify the maximum number of records to return. A specification of COUNT *, the default, will return all detail records associated with the datasets. The SKIP keyword may optionally be used to specify the number of records that will be bypassed before returning them as stem variables.

```
ADDRESS EPIC EXTRACT DETAIL ID ii STEM n. COUNT c SKIP s
```

The following form returns the detail record for the tape volume vvvvvv. If stem n.0 is 0, the requested volume serial was not found.

```
ADDRESS EPIC EXTRACT VOL vvvvvv STEM n
```

Processing Output

If you need to use data records extracted by the EXTRACT ID, EXTRACT DETAIL, or EXTRACT VOL instructions, they can be decoded by including the GREXX member EPICREC.OAL in your IMODs. This member is loaded into the MON PDS during Online Manager installation.

Chapter 5: CA EPIC for z/VSE TLBL and DLBL Statements

This chapter describes CA EPIC for z/VSE's TLBL and DLBL statements. CA EPIC for z/VSE TLBL statements are slightly different from VSE TLBLs and provide additional options. CA EPIC for z/VSE DLBL statements have the same format as VSE DLBLs and also provide additional options. CA EPIC for z/VSE provides an MVS-style DD statement for VSE. See CA EPIC for z/VSE DD Statements (see page 177) in this guide for further information. Each type of CA EPIC for z/VSE label statement provides its own benefits. Which you choose depends largely on what you want to do with the dataset you are processing.

This section contains the following topics:

TLBL Statements

CA EPIC for z/VSE provides modifications to the standard VSE TLBL statement. Similarities and differences are shown in the table below.

All TLBL fields are positional.

TLBL Format

Pos	CA EPIC for z/VSE Format and Terminology	VSE Format and Terminology
1	// TLBL *dtfname*	// TLBL *filename*
2	[,'*dataset-name* [,*alpha-opts*]']	[,'*fileid*']
3	[,*retention*]	[,*retention*]
4	[,*volser*]	[,*file-serial-number*]
5	[,*reel-number*]	[,*volume-sequence-number*]
6	[,*file-sequence-number*]	[,*file-sequence-number*]
7	[,*version-number*]	[,*generation-number*]
8	[,*numeric-options*]	[,*version-number*]
9	[,*disposition*]	[,*disposition*]

Parameters

Parameter	Required?	Valid Entry	Default
dtfname	Yes	1 to 7 characters	None
dataset-name	For output	See description below	Blanks
alpha-opts	No	See description below	None
retention	No	1 to 9999 or yyyy/ddd	None
volser	No	1 to 6 characters	None
reel-number	No	1 to 256	1
file-sequence-number	No	1 to 9999	None
version-number	No	1 to 999	1
numeric-options	No	0 to 99	0

dtfname

dtfname specifies the symbolic name that the program uses for the dataset, for example TAPEIN and SORTIN. IBM calls this field the *filename*.

Dataset Name Field

The *dataset name field* consists of *dataset-name* and any desired *alpha-opts*. A beginning and ending apostrophe are required. This parameter is required for output. For input, the dataset name field can be omitted and a volume serial number supplied instead. However, if you want CA EPIC for z/VSE to select the tape, specify this parameter. The maximum length of the dataset name field (including *dataset-name*, commas, options, and parentheses) is 58 characters.

dataset-name

dataset-name specifies the name of the actual dataset to be used. IBM calls this the *fileid*. For CA EPIC for z/VSE to control the dataset, *dataset-name* must be identical to an EDD name in the DSN Catalog. The maximum length of *dataset-name* for a prime dataset in a multi-file group is 17 characters. Single dataset or subdataset names can be up to 44 characters in length.

alpha-opts

alpha-opts are provided for compatibility with tape management products from other vendors. The JCLOPT configuration option must be YES to use these options. You can specify alphabetic codes, version requests, and density requests (in any order), as long as they are separated from *dataset-name* and each other by commas. For alphabetic codes, see TLBL Options (see page 169). A specific version can be requested by specifying a parenthetical expression that contains a minus sign (-) followed by an integer that is one less than the version desired. For example, to retrieve catalog version 2, specify (-1).

Note: For a list of supported tape densities that can be specified, see 'Supported Tape Modes' in Appendix B.

retention

retention specifies the number of days to retain this version.

volser

volser requests a specific tape volume serial number. IBM calls this the file serial number. This field is optional, and *we recommend that it be omitted*.

reel-number

reel-number specifies the reel number to be processed first. Use this parameter to skip initial reels in a multi-volume group. If you specify a volume serial number, *reel-number* is ignored.

file-sequence-number

file-sequence-number specifies the file number of the dataset on the tape.

version-number

version-number requests a specific version. Use the number assigned to the version in the DSN Catalog for the version. If you are requesting version number 2, specify '2' here. If you specify a volume serial number, *version-number* is ignored. This parameter replaces the generation number in the VSE format.

numeric-options

numeric-options specifies numeric option codes. See "TLBL Options (see page 169)" below for the numeric option codes and their meanings. This parameter replaces the version number in the VSE format.

disposition

disposition specifies file disposition as DISP=OLD, DISP=NEW, or DISP=MOD. It is identical to the VSE parameter.

Note: For more information, see the IBM TLBL documentation..

VSE LOGSRC Option

The VSE LOGSRC option has no effect on TLBL statements. CA EPIC for z/VSE performs the symbolic parameter substitution and will only display the TLBL after substitution, not before.

Examples

In the following example, CA EPIC for z/VSE selects the tape volume BACKUP.STR (version 1) from the DSN Catalog.

```
// TLBL TAPEIN,'BACKUP.STR'
```

In the following example, CA EPIC for z/VSE selects the BACKUP.STR (version 3) from the DSN Catalog.

```
// TLBL TAPEIN,'BACKUP.STR,(-2)'
```

In the following example, CA EPIC for z/VSE honors the request for volume EP6173.

```
// TLBL TAPEIN,,,EP6173
```

In the following example, CA EPIC for z/VSE selects BACKUP.STR, and specifies "Rewind and unload at close" and mode '08'.

```
// TLBL TAPEIN,'BACKUP.STR,U,08'
```

TLBL Options

TLBL option codes provide special CA EPIC for z/VSE processing that is unavailable with standard VSE TLBL parameters. See "TLBL Statements (see page 165)" above for instructions and restrictions on specifying parameters.

Alphabetic option codes (called *alpha options*) are provided for compatibility with other tape management products. Alpha options can only be specified within the TLBL dataset name field.

Numeric codes can only be specified in the TLBL *numeric-options* parameter. To specify multiple numeric codes, total the different codes together. For example, if you want to specify option 3 (Do not rewind) and option 64 (Inhibit LUB allocation), specify 67 in the numeric-options parameter.

Numeric options can be used for uncontrolled datasets.

Option Codes

Alpha	Numeric	Action
	0	No options in effect
W	1	Rewind at close
U	2	Rewind and unload at close
	3	Do not rewind
R	4	Scratch at close
H	8	Hold the drive assignment at close
D	16	Data set drop
	32	Conditional catalog
	48	Concatenating datasets
	64	Inhibit LUB allocation

0 (No Options in Effect)

When no options are coded, CA EPIC for z/VSE honors the rewind code in your program.

Rewind options in the TLBL override rewind specifications in your program.

1 or W (Rewind)

Rewinds the tape to load point at close. This eliminates operator intervention when the same tape is reused in another step or job. If a cartridge is being used on an ACL drive, CA EPIC for z/VSE unloads it at close unless the "Hold assignment at close" option (8 or H) is specified.

2 or U (Unload)

Rewinds and unloads the tape at close.

3 (Do not rewind)

Prevents rewinding the tape at close. Use this option when you are creating multi-dataset tapes.

4 or R (Scratch at close)

Scratches the dataset when it is closed as input. This option does not affect subdatasets.

8 or H (Hold the drive assignment at close)

Prevents CA EPIC for z/VSE from releasing the tape drive at close. If Autoattach is on, it also prevents the drive from being detached. The drive will be released at end-of-job, when the logical unit is assigned to another device, or when the tape drive is deviced down, whichever occurs first.

16 or D (Dataset Drop)

Use this option when your program issues multiple "open" requests for the same DTF name. While VSE alone would process the original label each time the same DTF name is opened with the drop option, CA EPIC for z/VSE processes the next label in the JCL with that DTF name. (Dataset drop differs from concatenation, since concatenation processes multiple datasets with one open.)

32 (Conditional catalog)

Puts the version in the DSN Catalog with conditional catalog status.

48 (Concatenate datasets)

Reads multiple files using only one open. The files are read as a single dataset. This process is called dataset concatenation. Use concatenation to read multiple files when your program performs only one open. This option is valid for controlled datasets only.

Note: For coding, examples, and limitations, see Concatenating Datasets (see page 180) under Specifying the DD Statement in this guide.

64 (Inhibit LUB allocation)

Inhibits LUB allocation for the dataset. Inhibiting LUB allocation is often necessary when you are running programs that use physical IOCS, or the CA EPIC for z/VSE TSIDASS program. You do not have to specify this option when LUB allocation is turned off automatically or when a SYS number is specified in the dataset definition.

DLBL Statements

The CA EPIC for z/VSE DLBL statement is identical to the VSE DLBL statement, with additional options provided for special CA EPIC for z/VSE processing.

CA EPIC for z/VSE DLBL options are specified within the file-id (dataset-name) parameter, beginning after the dataset name itself and ending with the closing apostrophe ('). Options must be separated from the dataset name, and from each other, by a comma. The options(s) specified must be enclosed in parentheses.

CA EPIC for z/VSE replaces the options, commas, and parentheses with blanks before creating label information, so that only the dataset name appears in VTOC entries.

The maximum length allowed for the dataset name, including options, parentheses, and commas, is 44 characters.

Option Codes

Code	Action
A	Align allocation on cylinder boundaries
B=n	Reblock (n=1 to 32767|MAX|OPT)
C	Conditionally catalog
D	Delete after closed as input
E	Concatenate datasets
F	Allow fragmentation
G=n	Access the specified generation (n=1 to 65535)
H	Prevent automatic fragmentation
I	Ignore this dataset
L	Inhibit LUB allocation
M	Issue messages open/close
N	Bypass truncation
O	Use extent size in EXTENT statement
P	Delete (Purge) at End of Job (non-generation files only)
R=n	Allocate by Logical Records (n=1 to 999999)
S	Reuse same space
T	Dataset drop
V=n	Access the specified version (n=1 to 65535)
X	Delete at close or EOJ (non-generation files only)

DLBL Options

The following are the various DLBL options.

A

Aligns the disk allocation on cylinder boundaries. Valid for CKD devices only. Allocations for direct access (DA) and ISAM datasets are aligned automatically.

B=xxx

Changes the block size of your datasets without recompiling your programs. Valid for datasets with either fixed or variable record formats. Reblock with caution when running programs that use physical IOCS; such programs must be able to support reblocking. Specify a numeric block size, MAX for full-track blocking, or OPT for half-track blocking.

C

Puts the version in the DSN Catalog in conditional catalog status.

D

Deletes the dataset after it is closed as input. It is not required for compiler work areas or SORTWK files (they are deleted automatically).

E

Invokes dataset concatenation, which causes the problem program to read two or more files as if they were a single file (using just one open and one close). Dataset concatenation can only be performed for cataloged datasets.

Note: For coding examples and limitations, see Concatenating Datasets (see page 180) under Specifying the DD Statement in this guide.

F

When the requested amount of space is not available, this option allows CA EPIC for z/VSE to allocate the largest available contiguous space. Do not use for TYPEFLE=WORK files (for example, compiler work areas and FORTRAN disk files) or for any other files that do not support secondary extents. If you do not specify the F option, and CA EPIC for z/VSE cannot find the requested contiguous extent in the pool, message EP038 is issued, to which operator can respond *FRAG* to invoke fragmentation.

G=n

Accesses the specified generation of the dataset. This is useful when you want to access a specific version of a dataset without reference to its relative version number. The generation number of a version is listed in the TSIDUTL DSPLY and catalog maintenance DISPLAY reports. It is valid for cataloged datasets only.

H

Overrides the configuration option FRAG=YES, preventing automatic file fragmentation if the space requested for allocation is not available. The EP038 message is issued, so FRAG can still be specified in reply to continue the allocation.

I

Overrides CA EPIC for z/VSE's control of dataset. Valid ASSGN and EXTENT statements must be included to process the dataset.

L

Inhibits LUB allocation for the dataset. Inhibiting LUB allocation is often necessary when you are running programs that use physical IOCS. It is required with the CA EPIC for z/VSE TSIDASS utility. See "LUB allocation" for more information.

M

Overrides configuration options ALLMALL=NO (default) and ALLMSG=NO for this dataset. Allocation messages are issued for the dataset and for all other disk datasets opened after it.

For output datasets, EP039 is issued after the dataset has been opened, displaying the extents allocated. EP052 is issued after the file has been closed, displaying the size of the extent after truncation.

For input datasets, EP053 is issued to display the extent that has been opened, and EP051 is issued to indicate that the dataset has been closed.

This option is ignored for system datasets (such as IJSYSLN) held open across job step boundaries.

N

All unused space between the end-of-file record and the end of the last extent remains allocated to the dataset. Use this option when additional records may be written to a file after it is closed if the file is accessed later as a DA file. This option is required for any dataset processed with a TYPEFLE=WORK DTF. This option is also required when you use the 'S' option code (reuse same space).

O

Accepts the size specified in the EXTENT statement as the primary extent rather than the size specified in the DSN Catalog. Pool name and secondary extent size are still taken from the CA EPIC for z/VSE dataset definition. Use this option when the volume of data you are processing for a dataset is significantly different from normal. This option is valid for cataloged datasets only.

P

Deletes the dataset at end of job. Use this option for temporary datasets that are not closed as input. This option is automatically activated for all SORTWK files. This option affects only cataloged, non-generation datasets. Configuration option JOBACCT=YES is required.

R=nnn

Specifies the number of logical records the file is to contain. CA EPIC for z/VSE uses this number to calculate the size of the extent required.

S

The next version of the dataset is to be written in the same extent space the current version occupies. This option is useful when you want to create datasets under a fixed head for better performance. This option affects single extent non-generation datasets only. When you use this option, you must also inhibit truncation (option 'N') for the dataset.

T

Deletes the label information for this dataset from the label area (not from the VTOC) when the dataset is closed. If the problem program subsequently issues another open for the file, this label will not be found. This option is useful for programs that issue more than one open for the same DTF name. Dataset drop differs from dataset concatenation (option 'E') in that concatenation causes two or more datasets to be processed as a single dataset when a single open is issued by the program.

V=n

Specifies a specific version to access. This option is valid for cataloged datasets only. Remember that the latest version is always version 1.

X

Deletes the dataset at close if it is closed as input or at end of job if it still exists. This option provides the same function as the combination of the 'P' and the 'D' options. The configuration option JOBACCT=YES is required for this option to work. This option is valid only for cataloged, non-generation datasets.

Example

```
// DLBL INPUT  'INVENTORY,(C,F,E)'
```

Chapter 6: CA EPIC for z/VSE DD Statements

The CA EPIC for z/VSE DD statement is an MVS-style dataset label for VSE installations. For many users, its primary benefit is compatibility with MVS, because it simplifies moving JCL between the two operating systems. The DD statement also offers several options which are unavailable with standard VSE labels, even when using CA EPIC for z/VSE TLBL and DLBL options. If the TSIDOSJ module is not in the SVA, the DD statements are processed by VSE as comments.

This section contains the following topics:

Specifying the DD Statement

You can specify the DD statement as follows.

DD Statement Identifier (* //)

All DD statements follow the same format. The identifying characters are '* //', which must be specified in columns 1-4. These are followed immediately by the DD name.

Format

```
* //ddname  DD  parameter-expressions
```

DD Name

ddname performs the same function as the DTF name (*filename*). It specifies the program's logical name for the dataset. It must begin in column 5.

If you use a DD statement for a DTF that your program defines as a unit record DTF (DTFCD or DTFPR), *ddname* is the SYS number that is specified in that DTF. A unit record DTF is generated for COBOL programs when the SELECT statement contains an assignment name that specifies the class *UR*.

Do not use *ddname* in continuation statements or in the second and following statements of a concatenation.

Some DTF names are invalid for DD statements. Some have special meanings.

DD (operation)

DD specifies the DD operation. Place at least one blank immediately before and after it. Do not repeat the DD operation in continuation statements.

Additional Parameters

You can specify many MVS-compatible parameters to define your datasets. The parameters that follow the DD operation specify the type of processing to be done for the dataset. Some parameters are different for:

- Tape and disk datasets (including GDIUR datasets)
- Standard system datasets
- SYSIPT data
- Dummy datasets

Not all MVS DD parameters are supported by the CA EPIC for z/VSE DD facility. CA EPIC for z/VSE ignores unsupported and unrecognized parameters and subparameters. Supported parameters are discussed in the following sections.

DD Statement Syntax

Specifying Parentheses and Commas

You can specify multiple parameters after the DD operation. Some parameters have multiple subparameters. Separate multiple parameters and subparameters with commas. Enclose multiple subparameters in parentheses.

Many subparameters are positional. If you omit a subparameter, a comma must be specified in the omitted position for all but the last subparameter, just as in VSE and MVS.

Examples

```
* //SYS007  DD  DSN=SORTWK,UNIT=SYSDA,DISP=(NEW,DELETE),
* //            SPACE=(TRK,(5,5),RLSE),VOL=SER=DOSRES
VOL=(,RETAIN,,,REF=*.TAPEOUT)
```

Specifying Continuations

All DD statements and continuations can extend through column 71.

Format

```
* //ddname  DD  parameter,...,parameter,
* //           parameter,...,parameter
```

Step 1

Interrupt the current statement before column 71. The break must occur on a keyword boundary. Do not split subparameters. End the statement with a comma. Do not specify a comment after the comma.

Step 2

Begin the next statement with the statement identifier '* //' in columns 1-4. Omit *ddname* and the DD operation code.

MVS Compatibility

In MVS, the first parameter in a continuation line must begin in or before column 16. If you are specifying an CA EPIC for z/VSE DD statement for conversion to MVS, begin your VSE continuations in or before column 18 to maintain compatibility. To move the statement to MVS, delete the asterisk and the space that follows it ('* '), shifting the rest of the statement two columns to the left.

Example

```
* //SYS007  DD  DSN=SORTWK,DISP=(NEW,DELETE),
* //            UNIT=SYSDA,VOL=SER=DOSRES,
* //            SPACE=(TRK,(5,5),RLSE)
```

Specifying Dataset Drop

To take advantage of CA EPIC for z/VSE's dataset drop option, use multiple DD statements with the same ddname. No special coding is required. *This is not valid under MVS. In addition, VSAM datasets (including VSAM-managed* SAM *datasets) are not eligible for dataset drop.*

Example

```
* //TAPEIN  DD  DSN=INVENTRY.1QTR(0),DISP=(OLD,KEEP)
* //TAPEIN  DD  DSN=INVENTRY.2QTR(0),DISP=(OLD,KEEP)
* //TAPEIN  DD  DSN=INVENTRY.3QTR(0),DISP=(OLD,KEEP)
```

Concatenating Datasets

CA EPIC for z/VSE concatenates all versions of a single input dataset automatically, as does MVS. If you do not specify a single version on input, all versions are read sequentially, beginning with the current version.

Example

```
* //TAPEIN  DD  DSN=INVENTRY.1QTR,DISP=(OLD,KEEP)
```

To concatenate several versions of the same dataset or several different datasets, use a DD statement for each version or dataset and omit the ddname on all but the first statement. Submit the statements in the order you want the datasets processed. If block sizes differ, the version with the largest BLKSIZE *must* be specified first. *VSAM datasets (including VSAM-managed SAM datasets) are not eligible for concatenation.*

Example

```
* //TAPEIN  DD  DSN=INVENTRY.1QTR(0),DISP=(OLD,KEEP)
* //        DD  DSN=INVENTRY.2QTR(0),DISP=(OLD,KEEP)
* //        DD  DSN=INVENTRY.3QTR(0),DISP=(OLD,KEEP)
```

Syntax Checking

You can have CA EPIC for z/VSE check the syntax of DD statements before you run the job. Place a TYPRUN=SCAN or TYPRUN=SCANS parameter on your JOB statement. See TYPRUN for details.

Special DD Names

Many ddnames have special purposes in MVS. This affects how the CA EPIC for z/VSE DD facility processes them.

Unsupported DD Names

Statements containing unsupported ddnames are ignored by the CA EPIC for z/VSE DD system and treated as comments by VSE. Use standard VSE JCL to define datasets processed with these names.

The following ddnames are not supported:

- JOBLIB
- STEPLIB
- SYSCHK
- SYSCKEOV
- SYSOUT
- SYSPRINT

DD Names with Special Meanings

The following ddnames have special meaning to CA EPIC for z/VSE. If a program's DTF uses one of these names for a different purpose, use standard VSE JCL to define the dataset.

- JOBCAT
- STEPCAT
- SYSABEND
- SYSMDUMP
- SYSUDUMP
- SYSIN

JOBCAT and STEPCAT

JOBCAT and STEPCAT define VSAM catalogs to be used for VSAM clusters. In the CA EPIC for z/VSE DD facility (but not in MVS), they also define catalogs to be used for VSAM/SAM datasets.

JOBCAT defines the catalog for the entire job stream. STEPCAT defines the catalog for a step. STEPCAT overrides a JOBCAT definition for that step.

A JOBCAT DD must immediately follow the JOB statement. A STEPCAT DD must precede all VSAM labels which require that catalog.

Example

```
//JOB  VSAM UPDATE
* //JOBCAT  DD  DSN=VSAM.JOB.CATALOG,DISP=SHR
.


* //STEPCAT DD  DSN=VSAM.STEP.CATALOG,DISP=SHR
vsam labels
// EXEC PROGRAM
/&
```

SYSABEND, SYSUDUMP, and SYSMDUMP

SYSABEND, SYSUDUMP, and SYSMDUMP determine the kind of dump to be generated when a job cancels. They are similar to the DUMP and PARTDUMP parameters of the VSE OPTION statement. SYSABEND and SYSUDUMP specify a standard partition dump. SYSMDUMP specifies a full supervisor and partition dump.

The SYSOUT parameter must be specified with the SYSABEND, SYSUDUMP, and SYSMDUMP ddnames. SYSOUT identifies a print class in MVS. In VSE, class A indicates SYSLST, class B indicates SYSPCH.

Example

The following statement calls for a standard partition dump. '// OPTION PARTDUMP' is the VSE JCL equivalent.

```
* //SYSABEND  DD  SYSOUT=A
```

The following statement calls for a full supervisor and partition dump. '// OPTION DUMP' is the VSE JCL equivalent.

```
* //SYSMDUMP  DD  SYSOUT=A
```

SYSIN

SYSIN is used to assign a logical unit to SYSIPT data.

DD Statements for Tape and Disk Datasets

Introduction to the Parameters

In this section, we present a brief guide to the parameters and their purposes. Following this section, each parameter is examined in detail in a section of its own. Those sections are arranged in alphabetical order for quick reference.

The parameters available for defining tape and disk datasets are:

```
* //ddname  DD  DSN=dataset-name
                [ ,AMP=x ]
                [ ,DCB=x ]
                [ ,DISP=x ]
                [ ,FREE ]
                [ ,LABEL=x ]
                [ ,OPTION=x ]
                [ ,SPACE=x ]
                [ ,UNIT=x ]
                [ ,VOL=x ]
```

AMP

The AMP parameter is used to define a VSAM cluster or a VSAM/SAM dataset.

DCB

The DCB parameter defines the dataset's record characteristics. DCB is an MVS term that stands for *dataset control block. I*t determines the block size, logical record length, density (for tape), and dataset organization (for example, sequential, direct access, or indexed sequential for disk datasets).

DISP

Use the DISP parameter for almost all datasets. It specifies the current status of the dataset and its disposition at normal or abnormal termination.

The DISP parameter specifies whether a dataset is new or already exists, whether it should be retained at close or deleted, whether it should be cataloged or left uncontrolled, and so on.

Because the DISP parameter says so much about the dataset being processed, it also serves as documentation for your JCL, and it is sometimes used explicitly for that purpose even when only the defaults are specified.

DSN

DSN defines the name of the dataset to be processed. It is required for all tape and disk datasets. It also defines which version or versions of the dataset are processed. If you do not specify a particular version, all versions are read in on input (automatic concatenation).

You can also request CPU and partition independence or common work dataset support with the DSN parameter (these options are not available with MVS).

FREE

The FREE parameter specifies how tape drive assignments are to be handled when the dataset is closed. When doing multi-dataset processing, use FREE to hold the assignment until all datasets have been processed. FREE is only valid for tape datasets.

LABEL

The LABEL parameter determines how long output datasets are retained. In most cases, you will use it for uncataloged datasets. You can use it for a cataloged dataset when you wish to override the default retention for a version.

OPTION

OPTION is an CA EPIC for z/VSE DD parameter that has no counterpart in MVS. It allows you to specify options available in VSE and CA EPIC for z/VSE which have no equivalents in MVS.

SPACE

The SPACE parameter specifies disk space requirements for output datasets. Among other things, it specifies how much space to use and how to process space after the dataset is closed.

SPACE is optional for cataloged datasets. You can use it to specify an allocation size that is different from the size in the DSN Catalog, or to inhibit truncation for a version.

SPACE is required for Start Track 1 and uncontrolled datasets.

UNIT

UNIT specifies the type of device (tape or disk, 3380 or 3390, and so on) to be used for an output dataset.

VOLUME

The VOLUME (VOL) parameter specifies which tape volume, disk volume, or disk pool to allocate for an output dataset. For tape datasets, it can be used to specify rewind options for multi-dataset processing.

What to do for Different Types of Datasets

This section describes the particular DD parameters to use when you are processing different kinds of datasets.

Tape

For tape processing, pay particular attention to the VOLUME, FREE, and DISP parameters. VOLUME has several uses for tape. FREE holds and releases tape drive assignments. DISP=(x,KEEP) unloads tapes at close.

By default a tape is rewound at close, but left mounted. If you specify DISP=(x,KEEP), VOL=PRIVATE, or VOL=(,RETAIN) the rewind default is overridden.

For multi-dataset requirements, see "Multi-Dataset Processing,".

Disk

The SPACE parameter plays a particularly important role in disk processing. Use it to determine the extent allocations for datasets not yet entered in the DSN Catalog. SPACE is always required for Start Track 1 and uncontrolled datasets. Reuse the same extent with DISP=MOD. UNIT is required for disk output.

Cataloged

If the dataset is already defined in the catalog, specify DISP=(x,KEEP) for both input and output. You can override dataset defaults for the version by specifying the equivalent parameter. For instance, you can override period retention by specifying LABEL=RETPD=n. Otherwise, parameters do not have to be specified if that information is already in the dataset definition. If you want to modify the current dataset definition, specify DISP=(x,CATLG). You must also set the JCLOVRDE configuration option to YES.

To autocatalog a new output dataset, specify DISP=(NEW,CATLG). It is cataloged using the data in the DD statement. DD statement parameters cover most of the dataset definition, so you can do a thorough job of defining the dataset in your JCL. You cannot use DD statements to autocatalog an input dataset.

Note: If a dataset qualifies for autocataloging according to the configuration options you have specified, it will be cataloged automatically even if you don't specify DISP=(x,CATLG). See "Automatic Cataloging" in Chapter 1 for more information on autocataloging with configuration options. The recommended method is to use the DISP=(x,CATLG) parameter to autocatalog all datasets defined with DD statements.

Example

The following example shows a new, cataloged dataset. Its default block size is overridden: its actual block size is 4080.

```
* //TAPEOUT  DD  DSN=SARS.BACKUP,DISP=(NEW,KEEP),
* //             DCB=BLKSIZE=4080
```

The following example shows a new, uncataloged dataset. It is automatically cataloged, and the specified values for block size, period retention and density are entered in the dataset definition.

```
* //TAPEOUT  DD  DSN=SARS.BACKUP,DISP=(NEW,CATLG),
* //          DCB=(BLKSIZE=4080,DEN=F8),LABEL=RETPD=7
```

Start Track 1 Datasets

To create a Start Track 1 dataset, use the SPACE parameter and specify a relative starting address (SPACE=TRK, SPACE=CYL, or SPACE=alr). You must also use the VOLUME parameter, to assign the dataset to a disk pool. Do not catalog the dataset or specify DISP=(x,CATLG).

Example

```
* //DISKOUT DD  DSN=TWORK.BG,DISP=(NEW,PASS),
* //            SPACE=(TRK,(10,5),RLSE),VOL=SER=POOL01,UNIT=DISK
```

Uncontrolled Datasets

To create an uncontrolled (non-VSAM) disk dataset, use the SPACE=ABSTR and VOL parameters. No special parameters are required to create an uncontrolled tape dataset. Do not specify DISP=(x,CATLG), or the dataset will be cataloged and therefore controlled.

Configuration Options

To use CA EPIC for z/VSE DD statements for uncontrolled disk datasets, UNCDSK=YES must be specified.

Example

```
* //OPTOUT   DD  DSN=EPIC.LIBRARY,SPACE=(ABSTR,(100,20)),
* //             DISP=NEW,VOL=SER=SYSWK1,UNIT=DISK
```

Sequential and Direct Access Datasets

See the corresponding topics in "Types of Datasets".

VSAM/SAM

To process a VSAM/SAM dataset, use the AMP, DISP, and SPACE parameters for both input and output. You can use the VOLUME parameter to specify a disk volume. Identify the VSAM catalog with a STEPCAT or JOBCAT DD statement.

Example

In the following example, the STEPCAT DD statement defines the catalog for the VSAM/SAM dataset. AMP=AMORG indicates that a VSAM or VSAM/SAM dataset is being defined. DISP=SHR allows the dataset to be accessed by other programs as it is processed. This example uses ALR to allocate space to the dataset: it indicates a record size of 240 bytes and allocates a primary extent large enough for 1000 records and a secondary large enough for 500 records.

```
* //STEPCAT DD  DSN=CUSTOMER.CAT,DISP=SHR
* //VSAMSAM DD  DSN=CUSTOMER.MASTER,DISP=SHR,AMP=AMORG,
* //            SPACE=(240,(1000,500))
```

VSAM

You can use existing VSAM datasets using CA EPIC for z/VSE DD statements. To process an already existing VSAM dataset, include the AMP parameter in your statement. Define the VSAM catalog with a STEPCAT or JOBCAT DD statement.

Example

```
* //JOBCAT  DD  DSN=CUSTOMER.CAT,DISP=SHR
* //VSAMCLU DD  DSN=CUSTOMER.MASTER,DISP=SHR,AMP=AMORG
```

AMP Parameter

Use the AMP parameter to define a VSAM cluster or a VSAM/SAM dataset.

Syntax

```
AMP=(AMORG [ ,BUFSP=x ])
```

Subparameters

Subparameter	Required?	Valid	Default
AMORG	Yes	None	None
BUFSP	No	0 to 9999999	None

For more information about using special masking characters to unload data,

AMORG specifies that this is a VSAM cluster or VSAM/SAM dataset.

BUFSP specifies the number of bytes to be allocated as buffer space. If omitted, VSAM will determine the size of the buffer.

DCB Parameter

Use the DCB parameter to define the characteristics of your dataset. It overrides the DSN Catalog entry on output, but not on input.

Syntax

```
DCB=( [ BLKSIZE=x ][ ,DEN=x ][ ,DSORG=x ][ ,LRECL=X ] )
```

Subparameters

Subparameter	Required?	Valid	Default
BLKSIZE	No	1 to 65534 or MAX|OPT	None
DEN	No	See Supported Tape Modes in Appendix B	None
DSORG	No	See DSORG table (below)	PS
LRECL	No	1 to 65534	None

BLKSIZE specifies a block size. It is valid for sequential datasets only. MAX requests full track blocking on disk, 64K blocking on tape. OPT requests half-track blocking on disk, 32K blocking on tape.

DEN

Specifies a recording density. Any of the TLBL modes listed in Appendix B 'Supported Tape Modes' are valid. In addition, the following density codes are accepted in DD statements:

Code	Density
2	800 bpi
3	1600 bpi
4	6250 bpi
A	8809 streaming short gap
B	8809 start-stop long gap
C	8809 start-stop short gap
D	8809 streaming long gap
E	3480/3490 non-IDRC buffered write mode
F	3480/3490 IDRC buffered write mode
G	3424/9348 1600 bpi buffered write mode
H	3424/9348 1600 bpi unbuffered write mode
I	3424/9348 6250 bpi buffered write mode
J	3424/9348 6250 bpi unbuffered write mode
K	3490E cartridge device
L	3490E IDRC cartridge device
M	3590 buffered write mode
O	3590 IDRC buffered write mode
P	3590 unbuffered write mode
Q	3590 IDRC unbuffered write mode

DSORG

Specifies the organization of the dataset. It is required for direct access and indexed sequential datasets. Valid values are:

Code	Organization
DA and DAU	Direct access
IS and ISU	Indexed sequential
PS and PSU	Sequential

MVS Compatibility

There are two options for each type of dataset organization. In MVS, the 'U' indicates that the dataset is unmovable, so the same extents are used every time. When 'PSU', 'DAU', or 'ISU' is specified, and the dataset is a disk work dataset, CA EPIC for z/VSE allocates the same extents to the dataset that it already occupies. If you are going to convert to MVS, choose your dataset organization carefully.

LRECL

Specifies the logical record length of the dataset. The number specified must be less than or equal to the record length specified in the program's DTF. If it is greater, the LRECL subparameter is ignored.

DISP Parameter

DISP identifies how a dataset is to be opened and what is to be done to the dataset when it is closed. DISP subparameters define the usage, normal termination and abnormal termination disposition of a dataset. If more than one subparameter is used, the subparameter list must be enclosed in parentheses. When a subparameter inside the list is not specified (to use its default), a comma must still be used for all but the last subparameter.

Syntax

```
DISP=[usage-status][,normal-end-status][,abnormal-end-status]
```

Variables

usage-status is optional. If used it must be one of the following:

- NEW specifies that this is a new dataset or a new version of a dataset.
- OLD specifies that this is an existing dataset
- SHR specifies that this is an existing dataset
- MOD specifies that new data is to be added to an existing version. A new version will be created if there is no existing version

When you use MOD with a disk dataset, CA EPIC for z/VSE will reuse the extents used for the last version. For disk datasets, you can only specify MOD for non-generation disk datasets.

When used with tape, MOD causes new data to be written at the end of the dataset, over the existing EOF record. A new EOF will be created.

Important! *MOD does not extend an existing disk dataset as it does for tape, or as MVS does. If MOD is specified for an existing disk dataset that is opened as output, the same extent is used but records are written at the beginning of the dataset. Any data already in the dataset is overwritten and lost.*

If *usage-status* is not specified, NEW is assumed.

normal-end-status is optional. If used it must be one of the following:

- CATLG
- CONDCAT
- DELETE
- KEEP
- PASS

If *normal-end-status* is not specified, PASS is assumed if usage-status is NEW or MOD. KEEP is assumed if usage-status is OLD, SHR, or SHARE.

abnormal-end-status is optional. If used it must be one of the following:

- CATLG
- DELETE
- KEEP

If *abnormal-end-status* is used without both *usage-status* and *normal-end status*, *normal-end status* will be used for *abnormal-end-status*.

Normal Termination Subparameters

The following are the termination subparameters.

DELETE

Causes datasets to be scratched automatically after they have been used. Based upon the dataset usage, the scratching may be done at close, end of step (EOS) or at end of job (EOJ). Several factors determine when and if the DELETE disposition will be effective or bypassed. The factors are:

- work vs. generation dataset
- used for input or output
- for disk, if concatenation is being used

In the cases where the DELETE disposition is bypassed, it will function the same as the PASS disposition.

Deleting Datasets Using DDGEN and the DISP Parameter

The DDGEN configuration option controls when a dataset is to be deleted based upon its DISP DD parameter. The table below shows when deletion occurs based upon the possible DDGEN and DISP combinations. Where there is no entry in the table, no deletion occurs.

For simplicity, the table below describes the basic DISP combinations. For those combinations not explicitly defined in the table, the following rules apply:

- If the *usage-status* is SHR, it will be treated the same as OLD.
- If the *usage-status* is MOD, it will be treated the same as OLD if there is an existing version. If there is no existing version, it will be treated the same as NEW.
- The CONDCAT *normal-end-status* will be treated the same as KEEP.
- The CATLG *normal-end-status* and *abnormal-end-status* will be treated the same as KEEP.

When DISP=	If DDGEN=NO, then…	If DDGEN=YES, then…
NEW	deletes at end-of-job	deletes at end-of-step
NEW,KEEP		
NEW,KEEP,KEEP		
NEW,KEEP,DELETE	deletes at end-of-job only if job abends	deletes at end-of-step only if step abends
NEW,DELETE	deletes at end-of-job	deletes at end-of-step
NEW,DELETE,KEEP	deletes at end-of-job only if good eoj	deletes at end-of-step only if good eoj
NEW,DELETE,DELETE	deletes at end-of-job	deletes at end-of-step
NEW,PASS	deletes at end-of-job	deletes at end-of-job
NEW,PASS,KEEP	deletes at end-of-job only if good eoj	deletes at end-of-job only if good eoj
NEW,PASS,DELETE	deletes at end-of-job	deletes at end-of-job
OLD		
OLD,KEEP	(Note 1)	(Note 1)
OLD,KEEP,KEEP	(Note 1)	(Note 1)
OLD,KEEP,DELETE	deletes at end-of-job only if job abends	deletes at end-of-step only if step abends
OLD,DELETE	deletes at close (Note 2)	deletes at end-of-step
OLD,DELETE,KEEP	deletes at end-of-job only if good eoj	deletes at end-of-step only if good eoj
OLD,DELETE,DELETE	deletes at close (Note 2)	deletes at end-of-step
OLD,PASS		
OLD,PASS,KEEP		
OLD,PASS,DELETE	deletes at end-of-job only if job abends	deletes at end-of-step only if job abends

NOTE 1

DISP=(OLD,KEEP) and DISP=(OLD,KEEP,KEEP) will automatically reset any delete at end-of-step or delete at end-of-job action currently in effect for the specified dataset.

NOTE 2

Normal deletion will occur at file close but, when used in conjunction with the automatic concatenation feature, deletion will occur at end-of-step. If usage is MOD, deletion will occur at end-of-job.

PURBYP Override

In cases of abnormal termination, the end-of-step and end-of-job deletions above can be overridden. To do this, set the PURBYP configuration option to YES.

EPS049 Option

CA EPIC for z/VSE does not delete datasets at end-of-step or end-of-job unless they have been opened. This can be overridden to allow the deletions even though there has not been an open. To do this, set both the DDGEN configuration option and the EPS049 special option to YES.

Additional DISP Settings

For both normal and abnormal terminations, other DISP settings are treated as follows:

Setting	Is Treated Like...
DISP=MOD	DISP=NEW
DISP=SHR	DISP=OLD.
DISP=(xxx,CATLG) DISP=(xxx,CONDCAT)	DISP=(xxx,KEEP)
DISP=(,xxx)	defaults to DISP=(NEW,xxx).

KEEP

For tape datasets, if KEEP is explicitly specified, the tape will be unloaded when it is closed If KEEP is not explicitly specified, the tape will not be unloaded at close. As a default, KEEP does not unload the tape. VOL=RETAIN overrides KEEP.

CATLG

(1) Catalogs the dataset automatically if no entry for it already exists in the DSN Catalog or (2) updates the dataset definition if one exists only when configuration option JCLOVRDE=YES is used.

CATLG can cause the following fields to be updated:

Tape/Disk	Tape Only	Disk Only
Retention	Tape pool	Disk pool
Block size		CISIZE
Density		Primary extent size
		Secondary extent size
		Maximum number of secondary extents

CA EPIC for z/VSE uses the information specified in your DD statement to create or update the dataset definition.

By default, tape datasets are always cataloged as generation datasets, and disk datasets as non-generation datasets. The exception to this occurs when LABEL=RETPD=xx is specified for a disk dataset; in that case the disk dataset will be cataloged as a generation dataset. If the startup option DDGEN is set to YES, tape and disk datasets will both be automatically cataloged as non-generation datasets unless a plus sign (+) is found in the DSN parameter. See the DDGEN configuration option in the *Installation and System Guide* for more information.

Note: The automatic cataloging configuration options (AUTOCAT and AUTOLOG) do not affect CATLG. However CATLG's updating function is permitted by the JCLOVRDE configuration option. Cataloging datasets as generation or non-generation is affected by the DDGEN configuration option.

CONDCAT conditionally catalogs the version. This is not a valid subparameter in MVS.

PASS retains the dataset for use in a later job step.

Abnormal Termination Subparameters

The default disposition for abnormal termination depends on both the normal termination subparameter and the PURBYP configuration option.

KEEP and *CATLG* Retain the dataset if the job cancels.

The abnormal termination disposition is KEEP unless both of the following are true:

- Normal termination disposition is DELETE
- PURBYP is set to NO

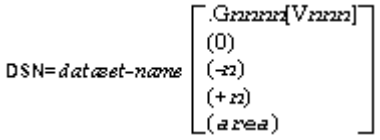
DSN Parameter

The DSN parameter specifies the name of the dataset being processed. It is required for all tape and disk datasets. It also determines which version or versions of the dataset will be used for input.

The rules for dataset names with DD statements are slightly different from those with standard VSE JCL. Both tape and disk dataset names can be up to 44 characters long. Generation dataset names are limited to 35 characters. If the dataset name contains blanks, enclose it in apostrophes.

Only the first 17 characters of the dataset name are written on a tape's HDR1 label.

Syntax



or

DSN=&[&]*dsname*

Subparameters

Subparameter	Required?	Valid	Default
dataset-name	Yes	1 to 44 characters	None
.G*nnnn*[V*nnn*]	No	Generation number	None
n	No	Zero or signed integer	See text
area	No	INDEX|PRIME|OVFLOW	None

dataset-name specifies the dataset name.

.G*nnnn*[V*nnn*] specifies a disk version by generation number. Specify the generation number after the G, but before the V, if any. The V*nnn* portion of the parameter is ignored by CA EPIC for z/VSE and can be omitted. It is included for MVS compatibility only.

(0), (-*n*), or (+*n*) specify a version by version number. (0) indicates the current version. A minus sign (-) indicates a previous version for input. A plus sign (+) indicates a new version for output.

Important! *For input, if no version is specified in the dataset name, all versions of a generation dataset will be concatenated and read!*

For compatibility with MVS, the current version is indicated by a zero instead of a one as in the DSN Catalog. That means that when you enter a version number with a minus sign, it should be one less than the version number displayed in the DSN Catalog. For example, if the version number in the DSN Catalog is 3, specify DSN=*dataset-name*(-2).

Also note that specifying '(+n)' is not necessary for output datasets. Specifying the dataset name is sufficient to cause a new version to be created. However, specifying '(+n)' with DISP=(x,CATLG) causes disk datasets to be cataloged as generation datasets, if the DDGEN configuration option is set to YES. If DDGEN is set to NO, and you do not specify LABEL=RETPD or LABEL=EXPDT, the dataset is cataloged as a non-generation dataset.

(area)

Specifies which area of an ISAM dataset is being defined. INDEX indicates the cylinder index component. PRIME indicates the prime data area. OVFLOW indicates the overflow area.

&&dsname, &dsname

Can be used to specify work datasets. In MVS, placing two ampersands (&&) before a dataset name indicates a temporary dataset. In CA EPIC for z/VSE, two ampersands (&&) indicate CPU and partition independence. A single ampersand (&) indicates CPU independence only. The dataset name can be up to 44 characters.

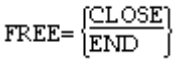
CPU and partition independence can also be indicated with equal signs (=) or dollar signs ($), as with TLBL and DLBL statements (see "CPU and Partition Independence" and "Common Work Datasets"). Please note, however, that equal signs and dollar signs are not supported by MVS. The MULTCPU configuration option affects CPU independence.

FREE Parameter

Specifies when a tape drive assignment is to be released. FREE is valid for tape datasets only. If FREE is not specified, the assignment is released when the dataset is closed.

When you are processing multi-dataset tapes, you must specify FREE=END for every dataset except the last one processed in the job.

Syntax



Values

CLOSE releases the assignment when the dataset is closed. This is the default.

END holds the assignment beyond the close of the dataset. The assignment is released at the end of the job.

Example

The following example shows how the FREE parameter is used to hold the assignment until the last dataset is closed.

```
* //INPUT1  DD  DSN='INPUT.DSN.1'
* //            UNIT=TAPE,
* //            DISP=(OLD,KEEP),
* //            FREE=END,
* //            VOL=(,RETAIN)
* //INPUT2  DD  DSN='INPUT.DSN.2'
* //            UNIT=TAPE,
* //            DISP=(OLD,KEEP),
* //            FREE=END,
* //            VOL=(,RETAIN)
* //INPUT3  DD  DSN='INPUT.DSN.3',
* //            UNIT=TAPE,
* //            DISP=(OLD,KEEP)
```

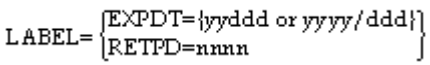
LABEL Parameter

The LABEL parameter can be used to define the dataset's retention. You can specify an explicit expiration date or specify the number of days the dataset is to be retained.

The LABEL parameter overrides the CYCL and RETEN configuration options. If a dataset is being autocataloged for the first time, the new EDD will take the LABEL value. If a dataset is already cataloged, a new version takes the LABEL value instead of the EDD value.

LABEL is valid only with DISP=NEW. If you specify LABEL=EXPDT or LABEL=RETPD with DISP=(NEW,CATLG), the dataset is cataloged as a generation dataset.

Syntax



Subparameters

Subparameter	Required?	Valid	Default
Specify only one of the following (required):			
EXPDT		00001 to 99365 2000/001 to 2099/365	None
RETPD		1 to 9999	None

EXPDT specifies an expiration date for the dataset. Enter the date in Julian format (for example, 92365 for December 31, 1992). If the date is January 1, 2000 or later, you can specify the full year with a slash between the year and the day (for example, 2000/001), or you can omit the slash (2000001).

RETPD specifies the number of days to retain the dataset.

Examples

```
* //OUTPUT  DD  DSN=OUT.DATA.SET,LABEL=EXPDT=98365,
* //            DISP=(NEW,KEEP)
* //OUTPUT  DD  DSN=OUT.DATA.SET,LABEL=RETPD=5,
* //            DISP=(NEW,KEEP)
```

OPTION Parameter

The OPTION parameter can be used to invoke an CA EPIC for z/VSE feature applicable for which there is no equivalent MVS JCL parameter. OPTION can be used to:

- selectively activate disk allocation messages
- specify the name of an applicable VSAM catalog
- specify an FBA CISIZE
- specify a specific logical unit number
- specify a tape pool ID
- base current disk allocation size on the size used by a different dataset
- selectively remove a VSE label from the label area
- override rewind options specified by DISP or VOL parameters

When multiple subparameters are used, they must be separated by commas and the list must be enclosed in parentheses. The subparameters are not positional, so it is not necessary to use commas in place of unspecified subparameters.

Syntax

```
          [ ,ALLMSGS ]
          [ ,CAT=ddname ]
          [ ,CISIZE=n ]
          [ ,NOREW ]
OPTION=   [ ,RESET ]
          [ ,REW ]
          [ ,RUN ]
          [ ,SPACE=*.ddname[x] ]
          [ ,SYSnnn ]
          [ ,TPL=x ]
```

Subparameters

Subparameter	Required?	Valid	Default
ALLMSGS	No		OFF
CISIZE=	No	512, 1024, 2048, 3072, 4096	None
NOREW	No		see below
RESET	No		OFF
REW	No		see below
RUN	No		see below
SPACE=	No	1 to 15 characters	None
nnn	No	000 to 254|RDR|IPT|LST|PCH	None
TPL=	No	A to Z or 0 to 9	None

ALLMSGS

Overrides configuration options ALLMALL=NO (default) and ALLMSG=NO for this dataset in the current job step only. It enables all allocation messages for the disk dataset. This subparameter performs the same function as the CA EPIC for z/VSE DLBL option 'M'. The ALLMSGS subparameter is valid for disk datasets only and is ignored if specified for a tape dataset.

CAT=ddname

Specifies the DD name used in a previous DD or DLBL statement to define this dataset's VSAM catalog. Use this parameter only when you are defining a VSAM dataset. It will override any JOBCAT or STEPCAT statement for the dataset.

CISIZE=nnnn

Specifies the CISIZE to be used for an FBA disk dataset according to IBM rules. NOTE: If the dataset is a Start Track 1 dataset and CISIZE= is specified, specification of the maximum number of secondary extents in the VOLUME parameter is ignored.

NOREW

OPTION=NOREW can be used to not rewind the tape at close. It overrides any rewind option defaulted to by the DISP or VOL parameters.

RESET

Specifies that the control of a dataset by the Generic Device Independence for Unit Record (GDIUR) feature is to be ended. If OPTION=RESET is used, it must be the only parameter specified on the DD statement, and no other OPTION subparameter can be used.

A DD statement with OPTION=RESET is only required when the job contains no TLBL, DLBL, or DD statements in any step following the DD statement that was used to invoke GDIUR control of the dataset. The ddname on the DD statement that specifies OPTION=RESET must be the same as the ddname on the DD statement that was used to invoke GDIUR control of the dataset.

RESET Example

In this example, SYSIPT control statements are read by the first LIBR execution from the tape or disk dataset "SYSIPT.DATA". The second DD statement for IJSYSIN with the OPTION=RESET insures that the SYSIPT control statements for the second LIBR execution will be read from the instream data, not from the "SYSIPT.DATA" dataset.

```
* //IJSYSIN  DD  DSN=SYSIPT.DATA,DISP=OLD
// EXEC LIBR
/*
* //IJSYSIN  DD  OPTION=RESET
// EXEC LIBR
 LD L=IJSYSRS
/*
```

REW

OPTION=REW can be used to rewind the tape at close. It overrides any rewind option defaulted to by the DISP or VOL parameters.

RUN

OPTION=RUN can be used to unload the tape at close. It overrides any rewind option defaulted to by the DISP or VOL parameters.

SPACE=

*.ddname[x]

Specifies that the primary extent allocation is to be based on the amount of space used by a dataset that was defined previously in the same job step. Identify the dataset you are referring to by specifying its ddname in the SPACE parameter. RLSE is automatically invoked with SPACE refer-back.

There are several ways in which the size of the allocation can be varied:

1. To specify that the dataset is to be exactly the same size as the referenced dataset, use the format:

```
SPACE=*.ddname
```

2. To specify that a dataset is to be larger than the referenced dataset by an exact number of tracks or FBA blocks, use the format:

```
SPACE=*.ddname+n
```

where n is the number of tracks or FBA blocks to be added to the size of the referenced dataset.

3. To specify that a dataset is to be smaller than the referenced dataset by an exact number of tracks or FBA blocks, use the format:

```
SPACE=*.ddname-n
```

where n is the number of tracks or FBA blocks to be subtracted from the size of the referenced dataset. Note that an error will result if the calculated allocation is not a positive number.

4. To specify that the size of the dataset is to be a percentage of the size of the referenced dataset, use the format:

```
SPACE=*.ddname*n
```

where n is a number between 000.01 and 999.99 (leading zeroes are not required).

If the referenced dataset is cataloged, the allocation will be based on the amount of space the referenced dataset actually occupies (the sum of its primary extent space and all secondary extent spaces). If the referenced dataset is uncataloged, the allocation will be based on the amount of space in the referenced dataset's primary extent only.

SPACE Examples

The following DD statement defines the referenced dataset:

```
* //REFFILE DD DSN=dataset-name,DISP=SHR
```

To allocate a new dataset one track larger than the space the referenced dataset occupies, specify:

```
* //FILEOUT DD DSN=dataset-name,DISP=(NEW,CATLG)
* //           UNIT=SYSDA,OPTION=SPACE=*.REFFILE+1
```

To allocate a new dataset five tracks smaller than the space the referenced dataset occupies, specify:

```
* //FILEOUT DD DSN=dataset-name,DISP=(NEW,CATLG)
* //           UNIT=SYSDA,OPTION=SPACE=*.REFFILE-5
```

To allocate a new dataset half of the space occupied by the referenced dataset, specify:

```
* //FILEOUT DD DSN=dataset-name,DISP=(NEW,CATLG)
* //           UNIT=SYSDA,OPTION=SPACE=*.REFFILE*.5
```

To allocate a new dataset twice the space occupied by the referenced dataset, specify:

```
* //FILEOUT DD DSN=dataset-name,DISP=(NEW,CATLG)
* //           UNIT=SYSDA,OPTION=SPACE=*.REFFILE*2
```

SYSnnn

Causes the specified SYS number to be assigned to this dataset. Replace *nnn* with the appropriate programmer logical unit number or 'RDR', 'IPT', 'PCH', or 'LST'.

You can use OPTION=SYSnnn to satisfy the pre-open checking that is performed by some programs, such as IBM's SORT/MERGE. For most jobs, this eliminates the need to execute TSIDASS ASSGN for tape datasets or to use // ASSGN statements for disk datasets before such programs. For disk datasets, ensure that the pool name is specified in the DSN Catalog or on the VOL=SER parameter.

SYS*nnn* Example

```
// JOB IDCAMS  REPRO DISK TO TAPE
* //IN    DD DSN=DISK.SEQ,DISP=OLD,
* //         UNIT=DISK,OPTION=SYS007
* //OUT   DD DSN=DISK.SEQ.BACKUP,OPTION=SYS005,
* //         DISP=(NEW,KEEP,DELETE)
// EXEC IDCAMS,SIZE=AUTO
   REPRO INFILE(IN -
   ENVIRONMENT (RECFM(FIXBLK) BLKSZ(800) RECSZ(40) )) -
   OUTFILE(OUT -
   ENVIRONMENT -
   ( PDEV(2400) RECFM(FIXBLK) BLKSZ(8000) RECSZ(40) ))
/*
/&
```

TPL=x

Specifies a tape pool for new tape datasets. It is only honored when a dataset is automatically cataloged using DISP=(NEW,CATLG).

SPACE Parameter

SPACE specifies the allocation of a disk dataset. Space is used to:

- specify allocation size
- permit or inhibit truncation at close for an output dataset
- invoke dataset fragmentation if contiguous space is unavailable
- align allocation on track or cylinder boundaries on CKD devices

Parentheses are used to enclose the positional subparameters of the SPACE parameter. Additional parentheses are used to enclose the actual allocation amounts when supplying a prime allocation size with a secondary size and/or an index size. SPACE has two types of syntax, one for CA EPIC for z/VSE-controlled datasets and one for uncontrolled datasets.

Configuration Options

Using DD statements for uncontrolled disk datasets is permitted by the UNCDSK configuration option.

Controlled Dataset Syntax

SPACE=(*type*,(*size-spec*),RLSE,*frag-control*,ROUND)

type is required. It must be one of the following:

- TRK specifies allocation in tracks or FBA blocks.
- CYL specifies allocation in cylinders and forces the dataset to begin and end on cylinder boundaries.
- *block-length* specifies the dataset's block size for allocation by number of logical records (ALR). *block-length* can also be used to specify the record size of VSAM/SAM datasets.

size-spec is required. It must be enclosed in parentheses and takes the form:

primary-qty[,*secondary-qty*][,*index*]

where *primary-qty* specifies the amount of the primary allocation in tracks, FBA blocks, cylinders, or number of logical records depending on the value specified by type. To use the SPACE parameter to indicate an option for a cataloged dataset (such as fragmentation, no truncation, or align on cylinder boundaries) without affecting the allocation size, specify zero (0) in *primary-qty*. When *primary-qty* is zero, CA EPIC for z/VSE uses the allocation size in the dataset definition.

secondary-qty specifies the size of the secondary allocation. To specify a maximum number of secondary allocations, use the VOLUME parameter.

index specifies the size of the index component for an ISAM dataset. Use it only when both the prime and the index areas are defined with a single set of parameters. If you use *index* but omit *secondary-qty*, be sure to specify the comma for *secondary-qty.*

RLSE is optional. RLSE causes truncation of the dataset at close. The default is no truncation if RLSE is not specified, even for cataloged datasets (when the SPACE parameter is specified). However, if the SPACE parameter is not specified for cataloged datasets, they are truncated at close.

frag-control is optional. If used, it must be one of the following:

- CONTIG specifies that each allocation must be made using consecutive tracks or blocks for the entire allocation amount. This is the same as the DLBL option 'H'.
- MXIG and ALX both specify fragmentation without operator intervention. This is the same as the DLBL option 'F'.

Specify only one of the CONTIG, MXIG, or ALX subparameters.

MVS Compatibility

MXG and ALX have different meanings in MVS. For more information, see an MVS JCL manual.

ROUND is optional. It specifies alignment on a cylinder boundary.

Controlled Dataset Examples

In the following example, the dataset asks for primary and secondary allocations of 5 cylinders. Unused space will be released. The dataset will be automatically cataloged.

```
* //DISKIN  DD  DSN=INVENTRY.FILE,SPACE=(CYL,(5,5),RLSE),
* //            DISP=(NEW,CATLG)
```

In the following example, exactly the same space requirements are set, but because the DISP parameter doesn't tell CA EPIC for z/VSE to automatically catalog the dataset, it is treated as a Start Track 1 dataset. Note that the VOLUME parameter is also specified because it is required for a Start Track 1 dataset.

```
* //DISKIN  DD  DSN=INVENTRY.FILE,SPACE=(CYL,(5,5),RLSE),
* //            DISP=(NEW,KEEP),VOL=SER=SYSWK1
```

Uncontrolled Dataset Syntax

SPACE=(ABSTR,(*size-specification*))

ABSTR is required. It specifies the allocation requirements for an uncontrolled dataset. The dataset will be allocated at the address you specify in the next parameters.

size-specification is required. It must be enclosed within parentheses and takes the form:

primary-qty, *start-address* [,*index*]

where *primary-qty* specifies the size of the primary allocation in tracks or FBA blocks.

start-address specifies the starting relative track or FBA block (as in the VSE EXTENT statement).

index is only used when both the prime and index data areas are defined using a single set of parameters. It specifies the size of the index component for an ISAM dataset.

primary-qty, *start-address*, and *index* values are positional and must be enclosed in parentheses.

Example

The following example shows an uncontrolled dataset. CA EPIC for z/VSE creates a label indicating the extent begins at relative track 1680 and is 5 tracks long. It is on the SYSWK1 volume.

```
* //DISKIN  DD  DSN=INVENTRY.FILE,SPACE=(ABSTR,(5,1680)),
* //            DISP=(NEW,KEEP),VOL=SER=SYSWK1
```

UNIT Parameter

Specifies a device or device type for processing. It is usually specified for output, and it is required for disk output datasets. It is optional for input datasets that are controlled by the DSN Catalog. If UNIT is not specified, UNIT=TAPE is assumed unless the dataset name begins with "$$$." (common work dataset) in which case UNIT=DISK is forced.

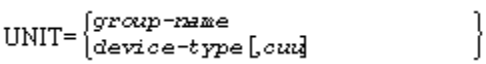
You can also request a unit by a group name. For disk datasets, CA EPIC for z/VSE chooses a volume from the disk pool specified on the VOLUME parameter. For all tape datasets, CA EPIC for z/VSE selects a suitable tape drive.

For uncontrolled disk datasets, you can use a group name such as 'UNIT=DISK'. However, you must also specify a particular device with the VOLUME parameter.

MVS Compatibility

In MVS, you can specify a pool name in the UNIT parameter. With CA EPIC for z/VSE DD statements, you must specify the VOL=SER parameter instead.

Syntax



Parameters

Subparameter	Required?	Valid	Default
Specify one of the following (required):			
group-name		DISK|SYSDA|TAPE|IDRC|CART	TAPE
device-type		See text	None

group-name specifies a generic request for a device group. This is the most generic kind of request. In MVS it specifies a particular category of device set up by your installation. In CA EPIC for z/VSE, valid groups are:

- DISK or SYSDA specify any disk device.
- TAPE specifies any tape device.
- IDRC specifies an IDRC drive.
- CART specifies any cartridge drive.

device-type specifies a device type, such as FBA, 3390, 3480. Valid device types are listed in the IBM *System Control Statements* manual. See the tape and disk device-classes for the ASSGN statement.

cuu specifies the address of a particular device. If *cuu* is specified, it must be separated from device-type by a comma, and both subparameters must be enclosed in parentheses. For example, UNIT=(CART,3F0).

Example

In the example below, the UNIT parameter indicates that this is a tape dataset. Any tape drive can be used.

```
* //TAPEOUT  DD  DSN=INVENTRY.FILE,UNIT=TAPE,
* //             DISP=(NEW,CATLG)
```

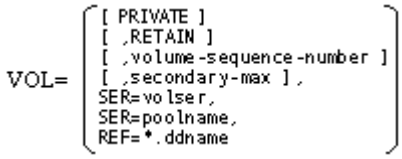
VOLUME Parameter

For tape datasets, use VOLUME to specify tape rewind options and to select a tape volume to process. These options are especially important for multi-dataset processing.

For disk datasets, use VOLUME to specify a disk pool for allocation and to specify a maximum number of secondary allocations. VOLUME is required for Start Track 1 datasets.

VOLUME's first four subparameters are positional. The last subparameter, SER or REF, is both keyword and positional. If specified alone, no commas are required before it. If other subparameters are specified as well, SER or REF must be the last positional subparameter.

Syntax



Subparameters

Subparameter	Required?	Valid	Default
PRIVATE	No		None
RETAIN	No		None
volume-sequence-number	No	1 to 255	1
secondary-max	No	1 to 255	255
SER=*volser*	No	1 to 6 characters	None
SER=*poolname*	For ST1	1 to 6 characters	None
REF=*.*ddname*	No	1 to 7 characters	None

PRIVATE causes the tape to be unloaded when it is closed. PRIVATE is overridden by the RETAIN parameter. If PRIVATE is not specified, the tape is rewound when it is closed. PRIVATE is ignored for disk datasets.

RETAIN prevents rewinding the tape when it is closed. This subparameter is most often used when processing multi-datasets. If neither PRIVATE nor RETAIN is specified, the tape is rewound when it is closed. RETAIN is ignored for disk datasets.

volume-sequence-number selects the first reel to process in a multi-volume group. It is ignored for disk datasets.

secondary-max specifies the maximum number of secondary allocations which can take place for this disk dataset. It is ignored for tape datasets.

SER=*volser* selects a volume by volume serial number. You can use this subparameter to specify a disk volume for uncontrolled disk datasets.

SER=*poolname* selects a disk pool. It is required for Start Track 1 datasets.

Example

In the following example, the VOL parameter indicates the pool where the Start Track 1 dataset is located.

```
* //DISKIN  DD  DSN=INVENTRY.FILE,SPACE=(CYL,(5,5),RLSE),
* //            DISP=(NEW,KEEP),VOL=SER=SYSWK1
```

REF=*.*ddname* associates a subdataset with a prime dataset. Use it with DISP=(NEW,CATLG) when you want to automatically catalog a subdataset. You can also use it when you want to refer to a prime dataset other than the one specified in the subdataset's dataset definition. Both the prime dataset and the subdataset must be controlled, and both must be defined with DD statements. The prime dataset's DD statement must precede all DD statements for related subdatasets.

Specify the ddname that is on the prime dataset's DD statement. For example, specify:

```
REF=*.TAPEOUT
```

to refer to a prime dataset defined with the ddname TAPEOUT.

Example

In the following example, the tape is unloaded at close.

```
* //TAPEIN  DD  DSN=INVENTRY.FILE,
* //            DISP=(NEW,KEEP),VOL=(PRIVATE)
```

The following example automatically catalogs a prime dataset and two new subdatasets. The datasets are "stacked" on one tape to create a multi-dataset group. 'PRIME.DSN.1' is the first dataset on the tape, 'NEWSUB.DSN.2' is the second dataset and 'NEWSUB.DSN.3' is the third dataset.

```
* //TAPEDUM  DD  DSN='PRIME.DSN.1(+0)',
* //             OPTION=SYS010,
* //             UNIT=TAPE
* //             DISP=(NEW,CATLG),
* //             FREE=END,
* //             VOL=(,RETAIN)
* //TAPEOUT  DD  DSN='NEWSUB.DSN.2(+0)',
* //             OPTION=SYS010,
* //             UNIT=TAPE,
* //             DISP=(NEW,CATLG),
* //             FREE=END,
* //             VOL=(,RETAIN,,,REF=*.TAPEDUM)
* //TAPEOUT  DD  DSN='NEWSUB.DSN.3(+0)',
* //             OPTION=SYS010,
* //             UNIT=TAPE,
* //             DISP=(NEW,CATLG),
* //             VOL=(PRIVATE,,,,REF=*.TAPEDUM)
// EXEC PROGX
```

The VOL=REF=*subparameter* tells CA EPIC for z/VSE the name of the prime dataset. VOL=(RETAIN,,,REF=*.TAPEDUM) tells CA EPIC for z/VSE that both of the following are true:

- this dataset is a subdataset and part of a multi-dataset group
- the prime dataset associated with this subdataset is 'PRIME.DSN.1'

See "Multi-Dataset Processing" for further explanation of prime datasets and subdatasets.

DD Statements For Printer and Punch Datasets

You can use DD statements to assign printer and punch datasets.

Format

```
* //ddname  DD  SYSOUT=x [ ,OPTION=SYSnnn ]
```

Parameters

Subparameter	Required?	Valid	Default
PRIVATE	No		None
RETAIN	No		None
x	Yes	A to Z or 0 to 9	None
SYS*nnn*	No	SYS000 to SYS254	None

SYSOUT specifies the class to be used for printer and punch data. For punch data, specify B. For print data, specify any other character.

SYS*nnn* specifies the programmer logical unit. Use this subparameter if the ddname itself is not a logical unit specification.

For example, if your COBOL program includes the statement

```
SELECT PRINT-OUT ASSIGN SYS006-UR-1403-S-PRINT.
```

then the DD statement would be:

```
* //PRINT  DD  SYSOUT=A,OPTION=SYS006
```

If no external name is used (the ddname is itself a logical unit specification), do not use the OPTION parameter. For example, if your COBOL program includes the statement

```
SELECT PRINT-OUT ASSIGN SYS006-UR-1403-S.
```

then the DD statement would be:

```
* //SYS006  DD  SYSOUT=A
```

DD Statements for In-Stream Datasets

When a job uses in-stream data, you can use a DD statement to assign a logical unit to the data stream. You must still place the in-stream data after the EXEC statement, as in VSE.

Format

```
* //ddname  DD  specifier  [,OPTION=SYSnnn]
```

Parameters

Subparameter	Required?	Valid	Default
specifier	Yes	* or DATA	None
SYS*nnn*	No	SYS000 to SYS254	None

* or DATA specifies that this statement defines an instream dataset.

SYS*nnn* specifies a programmer logical unit. Use this parameter if the ddname itself is not a logical unit specification.

For example, if your COBOL program includes the following statement:

```
SELECT CARD-IN ASSIGN SYS005-UR-2540R-S-CARDIN.
```

then the DD statement would be:

```
* //CARDIN  DD  *,OPTION=SYS005
```

or

```
* //CARDIN  DD  DATA,OPTION=SYS005
```

If no external name is used (the ddname is itself a logical unit specification), do not use the OPTION parameter.

For example, if your COBOL program includes the following statement:

```
SELECT CARD-IN ASSIGN SYS005-UR-2540R-S.
```

then the DD statement would be:

```
* //SYS005  DD  *
```

or

```
* //SYS005  DD  DATA
```

Examples

In the following example, the AJS00987 statement is read as in-stream data using SYS005. Because the ddname is itself a logical unit specification, no OPTION statement is necessary.

```
* //SYS005  DD  *
// EXEC PGM
AJS00987
/*
```

In the following example, LOG JE0987 and SP 9876 are read as in-stream data using logical unit SYS005.

```
* //CARDIN  DD  DATA,OPTION=SYS005
// EXEC PGM
LOG JE0987
SP 9876
/*
```

DD Statements for Dummy Datasets

You can define dummy datasets with DD statements.

Format

```
* //ddname  DD  DUMMY  [,OPTION=SYSnnn]
```

Parameters

Subparameter	Required?	Valid	Default
DUMMY	Yes		None
SYS*nnn*	No	SYS000 to SYS254	None

DUMMY specifies that this is a dummy dataset.

SYS*nnn* specifies the programmer logical unit and is required if the ddname itself is not a logical unit specification.

Example

This example defines the dataset belonging to DTF name (and logical unit) SYS005 as a dummy dataset. It is equivalent to a VSE // ASSGN SYS005,IGN. Input and output to the dataset is ignored.

```
* //SYS005  DD  DUMMY
```

Chapter 7: Running Jobs With CA EPIC for z/VSE

This section contains the following topics:

Overview

CA EPIC for z/VSE is designed to work with your existing JCL. However, JCL changes are sometimes necessary and sometimes helpful. Some situations require special entries in the CA EPIC for z/VSE dataset definition (EDD). This chapter discusses how to use CA EPIC for z/VSE and JCL with products from IBM and other third-party vendors. It describes new requirements and the use of CA EPIC for z/VSE label options. Recommendations for JCL and EDDs are presented for each topic when relevant. Please refer to the $INDEX SAMPJOB file as a reference to the many job samples provided on the CA EPIC for z/VSE installation tape. Familiarity with the TLBL and DLBL statements and CA EPIC for z/VSE DD statements is assumed.

Unsupported Datasets

CA EPIC for z/VSE does not control datasets that are defined in the JCL with the following DTF names:

- IJSYSNR
- IJSYSPC
- IJSYSRS

CA EPIC for z/VSE also does not control datasets processed with the following system utilities:

- POWER OFFLOAD and LST tapes
- DITTO basic and control functions

CA EPIC for z/VSE can control DITTO file functions.

JCL or DSN Catalog?

For output datasets, JCL parameters always override corresponding DSN Catalog information. However, JCL period retention must be greater than zero to override a DSN Catalog value.

For input datasets, DSN Catalog values override any corresponding JCL parameters, with the following exceptions:

- EXTENT size (DLBL or DD SPACE parameter)
- density (TLBL)
- block size or CISIZE (DLBL or DD DCB parameter)
- period retention (days) and expiration date

Input Dataset Selection

When you use TLBL and DLBL statements for input, CA EPIC for z/VSE selects the current version by default.

When you use DD statements for input, the entire dataset is concatenated if you do not specify a version number. All versions of the dataset are read, beginning with the oldest version and ending with the current version.

JCL can be used to override these defaults. You can request a specific tape version or a specific disk version or generation.

Note: For more details on overriding defaults, see TLBL Statements (see page 165), DLBL Statements (see page 172), and DD Statements in this guide.

Defining Your Datasets With TLBL, DLBL, and DD Statements

For every controlled dataset that it processes, CA EPIC for z/VSE uses the information in the DSN Catalog entry to update the VSE label area when the dataset is opened. Because this information is recorded automatically, JCL can be very concise. Using the DSN Catalog instead of JCL can be:

- simpler
- less error–prone
- easier to change

In many circumstances, all you need to specify for an input dataset is the DTF name (or the ddname that the program uses) and the dataset name.

Examples

```
// TLBL TAPEOUT,'SYSTEM.BACKUP'
* //TAPEOUT   DD   DSN=SYSTEM.BACKUP
```

On DD statements for output datasets, you may also need to specify a device type.

```
* //DISKOUT   DD   DSN=SYSTEM.BACKUP,UNIT=DISK
```

Rules For Dataset Name

Rules for dataset names are specified at and in this manual.

For generation datasets, specify the dataset name as it appears in the EDD. *Do not specify the G=nnnnnn extension that appears in the VTOC.*

LUB Allocation and the ASSGN Statement

The following are ASSGN statement and LUB allocation information.

When an ASSGN Statement is Not Required

Using LUB allocation and AVR, CA EPIC for z/VSE does the work of an ASSGN statement. LUB allocation selects the SYS number, and AVR selects the drive. As long as LUB allocation is active, ASSGN statements are not necessary and should be removed from your JCL.

When an ASSGN Statement is Required

ASSGN statements must be used whenever LUB allocation is inhibited. When LUB allocation is inhibited, the drive specified in the ASSGN statement is used.

LUB allocation is automatically inhibited:

- for direct access datasets
- if the logical unit number is currently assigned IGNORE
- for system logical units
- if the logical unit is assigned in the DSN Catalog

and for the following DTF names:

- UIN
- UOUT
- IJSYSxx
- SYS000
- SORTxxx (unless TSIDSRT is executed in place of SORT)

You can add other DTF names to this list by specifying them in the TSIDMAP macro. See the *Installation and System Guide* for details.

In addition, you must inhibit LUB allocation for datasets opened by TSIDASS and datasets which will be processed with PIOCS. This includes datasets processed with many utility programs and any dataset opened with a DTFPH. In most circumstances, the TSIDASS utility can replace the use of a VSE ASSGN statement.

Uncontrolled tape datasets do not need ASSGN statements. CA EPIC for z/VSE will request the operator to specify to a tape drive address.

Inhibiting LUB Allocation

You can inhibit LUB allocation for a dataset in several ways:

- the "L" option in the dataset name on a DLBL
- option "64" in the numeric options on a TLBL
- OPTION=SYSnnn parameter on a DD statement
- a default SYS number in the CA EPIC for z/VSE dataset definition

Disk Processing

Controlled Datasets

EXTENT Statements

For most controlled datasets, the EXTENT statement is ignored. An EXTENT statement can be required when you process direct access (DA) datasets or any datasets with programs that use PIOCS processing. See "Physical IOCS Processing" in this chapter for details.

To change the extent size from the size in the EDD, use the DLBL statement with the "O" option in the dataset name and use the EXTENT statement as you would for VSE. All parameters except the number of tracks/blocks on the EXTENT are ignored.

Example

```
// DLBL DISKIN,'==.WORKFILE.1,(O)'
// EXTENT SYS004,SYSWK1,1,0,5,210
```

This dataset is created with a primary extent size of 210 tracks, overriding the primary extent size in the EDD.

Start Track 1 Datasets

To define a Start Track 1 dataset, you must specify allocation information.

DLBL Statements

EXTENT statements are required for Start Track 1 datasets. The EXTENT statement for a Start Track 1 dataset differs only slightly from the standard VSE EXTENT statement. To use a Start Track 1 EXTENT, do the following:

- Specify an CA EPIC for z/VSE disk pool instead of a volume serial number.
- Specify a relative starting track/block of 1.

Example

A Start Track 1 dataset is allocated in POOL01 with 20 tracks.

```
// EXTENT SYS003,POOL01,1,0,1,20
```

If you specify a Start Track 1 EXTENT for a controlled dataset, CA EPIC for z/VSE ignores the EXTENT information.

Note: For Start Track 1 datasets defined with CA EPIC for z/VSE DD statements, see Start Track 1 Datasets (see page 186) under DD Statements for Tape and Disk Datasets in this guide.

Reusing the Same Space

You can write a new version of a disk dataset in the same location.

DSN Catalog

If you want to write a new version of a disk dataset in the same location (reusing the same space), catalog the dataset as work ("WRK" using Online Manager, or WRK=YES using TSIDMNT) and zero secondary allocations.

JCL Requirements

1. Each time the dataset is created or modified, inhibit truncation. This leaves space in the extent for a larger dataset the next time it is processed. To inhibit truncation with a DLBL, specify the "N" option in the dataset name; with a DD statement, use the SPACE parameter.

2. Each time the dataset is modified, you must indicate that you want to reuse the same space. With a DLBL, specify option "S"; with a DD statement, specify DISP=(MOD,KEEP).

Examples

```
// DLBL OUTPUT,'DATA.FILE,(N,S)'
* //OUTPUT  DD  DSN=DATA.FILE,UNIT=DISK,
* //            DISP=(MOD,KEEP),SPACE=(TRK,0)
```

Processing Tape Datasets

You can process tape datasets as follows.

Processing Internal Tape Labels

CA EPIC for z/VSE supports both labeled and unlabeled tape processing. CA EPIC for z/VSE automatically changes the DTF for an unlabeled tape (FILABL=NO or "LABEL RECORDS OMITTED") to a DTF for a labeled tape (FILABL=STD or "LABEL RECORDS STANDARD") when you supply a TLBL with the DTF name. This allows you to convert to labeled processing without making program changes. CA EPIC for z/VSE does not affect user header label processing.

DSN Catalog

To indicate standard label processing, specify LBL=YES on the TSIDMNT CATALOG statement. To indicate unlabeled processing, specify LBL=NO.

JCL Requirements

When processing a controlled dataset on an unlabeled tape, supply a TLBL or a DD statement for the dataset. Standard VSE processing does not require a TLBL for unlabeled tape datasets.

Rewind Options and MTC Statements

You do not need to use MTC statements to rewind your tapes. CA EPIC for z/VSE will position the tape automatically for input processing. You can also use CA EPIC for z/VSE's rewind options for output or input processing. These label options will override equivalent options in your program.

See PRIVATE and RETAIN at in this manual, and also TLBL options 1, 2, and 3 for information on tape rewind options.

Multi-Volume Processing

You can process up to 255 volumes per dataset.

JCL Requirements

None.

Multi-Dataset Processing

This section explains the concept of *multi-dataset groups* and describes their processing requirements and options.

A multi-dataset group consists of a prime dataset and up to 9,998 subdatasets. A multi-dataset group can reside on a single volume or on multiple volumes.

Prime Datasets

The first dataset on the first reel of a multi-dataset group is called the *prime dataset*. The prime dataset "owns" all the volumes in the group and controls their physical attributes.

All the datasets that follow the prime dataset in the group are called *subdatasets*. Subdatasets must be associated with the prime dataset. CA EPIC for z/VSE supports two methods for this:

- Automatically (using the DSN Catalog)
- Explicitly (using JCL)

If the datasets are associated using TLBL statements, the length of the prime dataset name is limited to 17 characters, although the subdataset names are not limited. This type of association is made by specifying the prime dataset name when the subdataset names are controlled.

If the datasets are associated using DD statements, the length of the prime dataset name can be up to 44 characters. This type of association is made by a referback to the prime dataset name on the DD statement for the subdataset. Therefore, a DD statement for the prime dataset name is necessary when the subdataset is created. This method allows the additional flexibility of associating the same subdatasets with different prime datasets depending on job requirements.

Subdatasets

Each subdataset has its own EDD, but some subdataset attributes cannot be specified because they are always inherited from the prime dataset. These attributes are:

- Density
- Tape labeling
- Tape size
- Tape pool
- Owner ID
- Vaulting method

Scratching Multi-Dataset Tapes

Multi-dataset tapes are scratched when all active datasets on the tape are scratched. None of the volumes in a multi-dataset/multi-volume group is eligible for scratch until all active dataset on all active volumes are scratched.

DSN Catalog

1. If you want the group to be written to a particular tape pool, or with a particular density, or you want to specify any other physical property or handling of the tape, these characteristics must be specified in the prime definition. Applicable parameters on the TSIDMNT CATALOG statement include:

- density (MOD)
- label type (LBL)
- owner ID (OWN)
- tape size (SIZ)
- vaulting method (VLT)
- tape pool (TPL)

2. Several important limitations apply when creating a subdataset EDD.

You must catalog the subdatasets using Online Manager (or TSIDMNT), or autocatalog them using DD statements. You cannot use TLBL statements to autocatalog subdatasets.

You must enter the prime dataset name correctly the first time for a subdataset EDD, because you cannot change the prime dataset name field in an existing subdataset EDD.

You can supply independent retention criteria, passwords, and block sizes for each subdataset.

You may find it helpful to give all the datasets in each multi-dataset group a single, unique user ID or system ID. This makes it easier to report or scratch them as a group, rather than as individual datasets.

JCL Requirements

1. Inhibit LUB allocation for all datasets using either of the following:

- option "64" in a TLBL statement
- SYSnnn option in a DD statement

If you are creating subdatasets with a single program that opens multiple DTFs with different logical units, you must inhibit LUB allocation through the DSN Catalog. To force CA EPIC for z/VSE to use the same logical unit for each dataset, specify the same SYS number in each EDD.

If you are creating subdatasets with different programs using DTFs with different logical units, you can use a VSE // ASSGN statement to assign the SYS number from the previous step to the SYS number in the current step. This may be necessary when using different utility programs to create subdatasets with different SYS numbers.

2. CA EPIC for z/VSE corrects the file sequence number on your TLBL, if it is incorrect. You can specify the file sequence number or omit it as you wish.

3. If your program uses multiple open requests for the same DTF name, you must specify TLBL option 16 (dataset drop). Dataset drop is automatic when you use multiple DD statements with the same ddname.

4. Prevent rewind on close for each dataset processed except the last. This can be accomplished with either of the following:

- option code "3" in a TLBL statement
- VOL=(,RETAIN) in a DD statement

For the last dataset, you may want to specify a rewind or unload option code. Rewind can be accomplished with either of the following:

- option "1" (rewind) in a TLBL statement
- FREE=CLOSE (rewind) in a DD statement
- Unload can be accomplished with either of the following:
- option "2" (unload) in a TLBL statement
- VOL=(PRIVATE) (unload) in a DD statement

5. For each dataset processed except the last, prevent the release of assignment. This can be accomplished with either of the following:

- option "8" in a TLBL statement
- FREE=END in a DD statement

For the last dataset, you usually want the assignment to be released at close. This can be accomplished by omitting the option code.

6. On output, CA EPIC for z/VSE continues to write new datasets at the end of the current multi-dataset group. When you want to create a new group, you must start by creating a new version of the prime dataset.

7. On input, you can process the datasets in any order. Because of the physical limitations of tape media, it is most efficient to process the datasets in ascending order by file sequence number, but this is not mandatory.

8. To automatically catalog a subdataset, you must use DD statements.

If the length of the prime dataset name is 17 characters or less, the prime dataset name is entered in the Comment field of the subdataset EDD. If the prime dataset name in the comment field is longer than 17 characters, CA EPIC for z/VSE puts XXXXXXXXXXXXXXXXX in the subdataset's Comment field.

Be sure the dataset is not already controlled before you attempt to catalog it using this option. If the prime dataset or any subdataset is already controlled, DISP=(x,CATLG) is ignored.

Multi-Dataset Processing Examples

Example 1

The first example is one of the simplest possible situations for creating a multi-dataset group. Your program creates datasets with different DTF names, but each DTF uses the same logical unit. The datasets are controlled in the first step. They are created in the second step. The second dataset on the tape is read in the third step. In the cataloging step, the prime dataset is defined first. It is given a tape pool which applies to all its subdatasets. All other datasets are defined with the NSU parameter, which makes them subdatasets. SUB1 and SUB2 have their own retention parameters specified. SUB3 is assigned the CA EPIC for z/VSE system's default retention. All datasets are defined with the user ID TS, so that they can be easily identified as a group. The last dataset does not require the tape and drive to be held, nor does it require dataset drop.

```
// JOB XYZ
*
* ------------------------------------------------------------+
*  CATALOG A PRIME DSN AND THREE SUB DSN'S                     +
* ------------------------------------------------------------+
*
// EXEC TSIDMNT
 CAT 'PRIME' CYC=2 UID=TS TPL=A
 CAT 'SUB1' NSU='PRIME' RET=14 UID=TS
 CAT 'SUB2' NSU='PRIME' CYC=3 UID=TS
 CAT 'SUB3' NSU='PRIME' UID=TS
/*
*
* ------------------------------------------------------------+
*  CREATE A PRIME AND 3 SUB DSN'S. PROGX PERFORMS 4 OPENS, +
*  ONE EACH FOR TAPE1, TAPE2, TAPE3 AND TAPE4. . . . . . . . . . . . . +
* ------------------------------------------------------------+
*
// TLBL TAPE1,'PRIME',,,,,,75          64 + 8 + 3
// TLBL TAPE2,'SUB1',,,,,,75           64 + 8 + 3
// TLBL TAPE3,'SUB2',,,,,,75           64 + 8 + 3
// TLBL TAPE4,'SUB3',,,,,,65           64 + 1
// EXEC PROGX
/*
*
* ------------------------------------------------------------+
*  READ FILE=2 ONLY                                            +
* ------------------------------------------------------------+
*
// TLBL TAPEIN,'SUB1'
// EXEC PROGY
/*
/&
```

Options Used

Code	Action
1	Rewinds the tape (for the last dataset)
3	Inhibits rewind at close (for all but the last dataset)
8	Holds the drive assignment at close (for all but the last dataset)
64	Inhibits LUB allocation (for all datasets)

Example 2

This example creates multiple datasets with the same DTF name. This process requires dataset drop.

```
// JOB XYZ
*
* ------------------------------------------------+
*  CATALOG A PRIME AND 2 SUB DSN'S                +
* ------------------------------------------------+
*
// EXEC TSIDMNT
 CAT 'PRIME' CYC=3 UID=KM
 CAT 'SUB1' NSU='PRIME' UID=KM CYC=4
 CAT 'SUB2' NSU='PRIME' UID=KM RET=14
/*
*
* ------------------------------------------------+
*  CREATE A PRIME AND 2 SUB DSN'S. PROGX PERFORMS+
*  3 OPENS FOR THE DTFNAME TAPEOUT. . . . . . . . . . . . . +
* ------------------------------------------------+
*
// TLBL TAPEOUT,'PRIME',,,,,,91        64 + 16 + 8 + 3
// TLBL TAPEOUT,'SUB1',,,,,,91         64 + 16 + 8 + 3
// TLBL TAPEOUT,'SUB2',,,,,,66         64 + 2
// EXEC PROGX
/*
*
* ------------------------------------------------+
*  READ A PRIME AND 2 SUB DSN'S. PROGY PERFORMS  +
*  3 OPENS FOR THE DTFNAME TAPEOUT. +
* ------------------------------------------------+
*
// TLBL TAPEIN,'PRIME',,,,,,91        64 + 16 + 8 + 3
// TLBL TAPEIN,'SUB1',,,,,,91         64 + 16 + 8 + 3
// TLBL TAPEIN,'SUB2',,,,,,66         64 + 2
// EXEC PROGY
/*
*
/&
```

The options in use are:

Code	Action
1	Rewinds the tape (for the last dataset)
2	Dismounts the tape (for the last dataset)
3	Inhibits rewind at close (for all but the last dataset)
8	Holds the drive assignment at close (for all but the last dataset)
16	Drops the TLBL (for all but the last dataset)
64	Inhibits LUB allocation (for all datasets)

The last dataset does not require the tape and drive to be held, nor does it require dataset drop.

Example 3

This example creates multiple datasets with different DTF names. Because the different DTFs use different logical units, a logical unit must be supplied in the dataset definitions.

```
// JOB XYZ
*
* ------------------------------------------------------+
*  CATALOG A PRIME AND 2 SUB DSN'S                       +
* ------------------------------------------------------+
*
// EXEC TSIDMNT
 CAT 'PRIME' SYS010
 CAT 'SUB1' NSU='PRIME' SYS010
 CAT 'SUB2' NSU='PRIME' SYS010
/*
*
* ------------------------------------------------------+
*  CREATE A PRIME AND 2 SUB DSN'S. PROGX PERFORMS 3 OPENS, +
*  ONE EACH FOR TAPE1, TAPE2 AND TAPE3. . . . . . . . . . . . . . . . . . . . +
* ------------------------------------------------------+
*
// TLBL TAPE1,'PRIME',,,,,,11          8 + 3
// TLBL TAPE2,'SUB1',,,,,,11           8 + 3
// TLBL TAPE3,'SUB2',,,,,,2            2
// EXEC PROGX
/*
*
/&
```

The options used are:

Code	Action
2	Dismounts the tape (for the last dataset)
3	Inhibits rewind at close (for all but the last dataset)
8	Holds the drive assignment at close (for all but the last dataset)

Because LUB allocation is inhibited through the dataset definition (SYS010 specified in the definitions of both the prime dataset and the subdatasets), 64 does not have to be specified on the TLBL.

Example 4

This example is very similar to Example 2. It may seem more complicated than Example 2, but the CA EPIC for z/VSE DD facility offers greater flexibility than TLBLs.

```
// JOB XYZ
* ------------------------------------------------------+
*  CATALOG A PRIME DSN AND THREE SUB DSN'S             +
* ------------------------------------------------------+
*
// EXEC TSIDMNT
 CAT 'PRIME'            CYC=2
 CAT 'SUB1' NSU='PRIME' CYC=2
 CAT 'SUB2' NSU='PRIME' CYC=2
/*
*
* ------------------------------------------------------------+
*  CREATE A PRIME AND 2 SUB DSN'S USING EPIC/VSE DD FACILITY. +
*  PROGX PERFORMS 3 OPENS FOR DTFNAME TAPEOUT. . . . . . . . . . . . . . +
* ------------------------------------------------------------+
*
* //TAPEOUT  DD  DSN='PRIME(+0)',
* //             OPTION=(SYS010),
* //             UNIT=TAPE,
* //             FREE=END,
* //             DISP=(NEW,KEEP),
* //             VOL=(,RETAIN)
* //TAPEOUT  DD  DSN='SUB1(+0)',
* //             OPTION=(SYS010),
* //             UNIT=TAPE,
* //             FREE=END,
* //             DISP=(NEW,KEEP),
* //             VOL=(,RETAIN)
* //TAPEOUT  DD  DSN='SUB2(+0)',
* //             OPTION=(SYS010),
* //             UNIT=TAPE,
* //             DISP=(NEW,KEEP),
* //             VOL=(PRIVATE)
// EXEC PROGX
/*
* ------------------------------------------------------------+
*  READ A PRIME AND 2 SUB DSN'S USING EPIC/VSE DD FACILITY. +
*  PROGY PERFORMS 3 OPENS FOR DTFNAME TAPEIN. +
* ------------------------------------------------------------+
* //TAPEIN  DD  DSN='PRIME(0)',
* //             OPTION=(SYS010),
* //             UNIT=TAPE,
* //             FREE=END,
* //             DISP=(OLD,PASS),
* //             VOL=(,RETAIN)
```

```
* //TAPEIN  DD  DSN='SUB1(0)',
* //             OPTION=(SYS010),
* //             UNIT=TAPE,
* //             FREE=END,
* //             DISP=(OLD,PASS),
* //             VOL=(,RETAIN)
* //TAPEIN  DD  DSN='SUB2(0)',
* //             OPTION=(SYS010),
* //             UNIT=TAPE,
* //             DISP=(OLD,KEEP)
// EXEC PROGY
/*
/&
```

In the example above, the options used in the creation step are:

Option	Action
DSN='*xxxx*(+0)'	The "+0" in parentheses indicates that this is a generation dataset (see the DDGEN configuration option in the *Installation and System Guide*).
OPTION=(SYS010)	Inhibits LUB allocation for this dataset. This is equivalent to TLBL option code 64. In addition, this forces the use of SYS010 (which has no TLBL equivalent).
UNIT=TAPE	Tells CA EPIC for z/VSE to create a tape label. This would be determined in standard VSE JCL by using a TLBL.
FREE=END	Holds the assignment at close for all but the last dataset processed. This is equivalent to TLBL option code 8.
DISP=(NEW,KEEP)	The NEW subparameter indicates that this dataset version will be created as output. The KEEP subparameter causes the tape to be unloaded at close unless VOL=(,RETAIN).
VOL=(,RETAIN)	Inhibits rewinding the tape at close (overrides the DISP subparameter KEEP). This is equivalent to TLBL option code 3.
VOL=(PRIVATE)	Causes the tape to be unloaded at close. This is equivalent to TLBL option code 2.

The option used in the read step are:

Option	Action
DSN='*xxxx*(0)'	The zero (0) in parentheses indicates that the current version (version 1) is to be processed.
DISP=(OLD,PASS)	The OLD subparameter indicates that this dataset version already exists and will be read as input. The PASS subparameter indicates that this tape is used later in the step or in a subsequent step.

Example 5

Whenever a subdataset such as VERY.LONG.NAME.FOR.NEWSUB.3 is accessed, the prime dataset name must be referenced in the JCL. The following example shows how VERY.LONG.NAME.FOR.NEWSUB.3 can be read as input:

```
// JOB XYZ
*
* ------------------------------------------------------------+
*  READ VERY.LONG.NAME.FOR.NEWSUB.3 AS INPUT. THE PRIME DATASET +
*  NAME (VERY.LONG.NAME.FOR.PRIME.1) MUST BE SPECIFIED AND  +
*  REFERRED TO BY THE DD FOR THE SUBDATSET. PROGY+
*  PERFORMS 1 OPEN FOR DTFNAME TAPEIN. . . . . .
* ------------------------------------------------------+
*
* //DUMMY    DD  DSN='VERY.LONG.NAME.FOR.PRIME.1(0)',
* //             UNIT=TAPE
* //TAPEIN  DD  DSN='VERY.LONG.NAME.FOR.NEWSUB.3(0)',
* //             UNIT=TAPE,
* //             DISP=(OLD),
* //             VOL=REF=*.DUMMY
// EXEC PROGY
/*
/&
```

Note the use of the DTF name DUMMY in DD statements. This is also known as the ddname. Any ddname except TAPEIN can be used because PROGY performs an open for TAPEIN, not DUMMY.

Types of Datasets

The following types of datasets are available.

Sequential Datasets

For disk, allocation is on track boundaries for CKD devices. If you specify DLBL option "A" or the SPACE=(...,ROUND) parameter on a DD statement, CA EPIC for z/VSE makes the allocation on cylinder boundaries.

Otherwise, there are no special requirements for sequential dataset processing.

Direct Access Datasets

For direct access datasets:

- LUB allocation is inhibited automatically
- Allocation is on cylinder boundaries and datasets are not truncated
- Secondary extents are not supported for controlled datasets

Configuration Options

For *uncontrolled* DA datasets, the following apply:

- When UNCDSK=NO, there is no limit to the number of extents.
- When UNCDSK=YES, there is a limit of 5 extents.

DSN Catalog

No special EDD requirements.

JCL Requirements

The VSE job control program requires you to use a DLBL and a complete EXTENT statement. For CA EPIC for z/VSE purposes, however, only the file type (access method "DA") on the DLBL and the extent sequence numbers on the EXTENT statement must be correct. For both cataloged and Start Track 1 datasets, the extent sequence numbers must begin with 1. VSE also requires that secondary extents reside on separate volumes.

Examples

The following example shows the EXTENT statement for a controlled dataset. CA EPIC for z/VSE supplies the volume serial number, starting address, and number of tracks/blocks from information in the catalog.

```
// DLBL RESDA,'dataset.name',,DA
// EXTENT SYS020,,1,1
```

The following example shows the EXTENT statement for a Start Track 1 dataset. CA EPIC for z/VSE supplies the actual volume serial number and starting address.

```
// DLBL RESDA,'dataset.name',,DA
// EXTENT SYS020,POOL01,1,1,1,15
```

On CA EPIC for z/VSE DD statements, specify DCB=DSORG=DA or DCB=DSORG=DAU. This example could be used to read or update a controlled DA dataset.

```
* //RESDA  DD  DSN=INVOICE.TICKETS,DISP=(OLD,KEEP),
* //           DCB=DSORG=DA
```

ISAM Datasets

CA EPIC for z/VSE can control ISAM datasets. The following VSE restrictions affect ISAM dataset processing with CA EPIC for z/VSE:

- CA EPIC for z/VSE can control only one prime data area, one overflow area, and one index per ISAM dataset.
- ISAM datasets are automatically allocated on cylinder boundaries.
- If UNCDSK=YES is specified as a configuration option, or if you specify DLBL options in your JCL, uncontrolled ISAM datasets are limited to a maximum of 5 extents total.

DSN Catalog

There are no special requirements for the catalog entry. Extent size specified in the catalog refers to the prime data area. Multiple extents are not allowed. Extent information for the index and overflow areas is taken from JCL. The disk pool applies to all areas, prime data, index, and overflow.

JCL Requirements

EXTENT statements are always required for all three areas, as follows:

1. For catalog controlled datasets, the number of tracks to allocate for the prime area is taken from the catalog, unless you specify the DLBL 'O' option. The number of tracks to allocate for index and overflow areas must be specified on the appropriate extent statement, as shown in the following two examples:

```
// DLBL MASTER,'EPIC.ISAM.MASTER.FILE',,ISE
// EXTENT SYS040,,4,1        Index
// EXTENT SYS041,,1,2        Prime
// EXTENT SYS042,,2,4        Overflow
// DLBL MASTER, 'EPIC.ISAM.MASTER.FILE',,ISC
// EXTENT SYS040,,4,1,,15      Index
// EXTENT SYS041,,1,2          Prime
// EXTENT SYS042,,2,4,,30      Overflow
```

2. For Start Track 1, the SYS number for the overflow extent must be different from the SYS number for the index and prime data areas, as shown in the following example:

```
// DLBL MASTER,'EPIC.ISAM.MASTER.FILE',,ISE
// EXTENT SYSO4O,POOL80,4,1,1,15    Index
// EXTENT SYS041,POOL80,1,2,1,90    Prime
// EXTENT SYSO42,POOL80,2,3,1,30    Overflow
```

3. For each DD statements, specify DCB=DSORG=IS or DCB=DSORG=ISU.

The three areas of an ISAM dataset (prime, index, and overflow areas) are all defined as part of the same DD statement. However, the overflow area must be defined using its own parameters (DSN, DCB, DISP, SPACE, VOL, and so on). You can define the prime and index areas together (with one set of parameters) or independently.

Examples

In the following example, each area is defined with its own set of parameters. Space and other information about the area follows the DSN parameter identifying the area. Because it is defined alone, index area space is defined with the primary-qty subparameter, SPACE=(TRK,1) and not the index-qty, SPACE=(TRK(,,1)).

```
* //DATA    DD  DSN=CUSTOMER.MASTER(PRIME),
* //            DISP=(NEW,KEEP)
* //            UNIT=DISK
* //            DCB=DSORG=IS,
* //            SPACE=(TRK,45)
* //            VOL=SER=POOL01
* //      DD    DSN=CUSTOMER.MASTER(INDEX),
* //            DISP=(NEW,KEEP)
* //            UNIT=DISK,
* //            DCB=DSORG=IS,
* //            SPACE=(TRK,1)
* //            VOL=SER=POOL01
* //      DD    DSN=CUSTOMER.MASTER(OVFLOW),
* //            DISP=(NEW,KEEP)
* //            UNIT=DISK,
* //            DCB=DSORG=IS,
* //            SPACE=(TRK,15)
* //            VOL=SER=POOL01
```

In the following example, one set of parameters is used to define both the prime and the index areas. In this case, SPACE is defined for the index area by using the index-qty subparameter.

```
* //DATA  DD  DSN=CUSTOMER.MASTER,
* //            DISP=(NEW,KEEP),
* //            UNIT=DISK,
* //            DCB=DSORG=IS,
* //            SPACE=(TRK,(45,,1)),
* //            VOL=SER=POOL01
```

VSAM and VSAM-Managed SAM Datasets

CA EPIC for z/VSE does not control VSAM or VSAM-managed SAM datasets. Therefore no changes are required in your JCL. However, you can use DD statements to access an existing VSAM cluster or VSAM-managed SAM dataset. See the IBM documentation for further information.

You can have VSAM dataset location and activity recorded in the DSN Catalog. If you catalog the VSAM cluster using Online Manager (or TSIDMNT), this information will appear on the CA EPIC for z/VSE reports and displays, just as it does for non-VSAM datasets. At close time, CA EPIC for z/VSE logs the allocations used for any suballocated or unique VSAM clusters.

Configuration Options

The VSAM configuration option enables logging of VSAM datasets.

Work Datasets

The following are the various types of work datasets.

Temporary Datasets

CA EPIC for z/VSE can release disk space or tapes used for controlled datasets at close or at end-of-job.

CA EPIC for z/VSE cannot automatically delete Start Track 1 datasets at end-of-job if the dataset is not closed, as may occur if the dataset is processed with PIOCS or the job step cancels. We recommend that you catalog all your temporary datasets instead of using Start Track 1.

You can find common work dataset (CWDS) support or CPU/partition independence helpful with work datasets. See "Common Work Datasets" and "CPU and Partition Independence" for details.

DSN Catalog

Catalog the EDD as work ("WRK" using Online Manager, or WRK=YES using TSIDMNT).

JCL Requirements

For TLBLs, specify option "4".

For DLBLs used for controlled datasets, specify option "D", "P", or "X" in the dataset name. For Start Track 1 DLBLs, specify option "D".

For DD statements, specify DISP=(NEW,DELETE).

CPU and Partition Independence

CPU and partition independence are normally used with work datasets. They allow you to use a single set of JCL to run in any partition or CPU. Because CA EPIC for z/VSE inserts a partition and/or CPU ID into the dataset name to make the dataset name unique, there will be no duplication of the dataset names in different partitions or machines.

For example, in your JCL you could specify ===.SORTWRK1 as the dataset name. The three equal signs designate both CPU and partition independence. When you run the job in the BG partition of CPU ID 1, CA EPIC for z/VSE uses the dataset name 'BG1.SORTWRK1'. When you run it in the F1 partition of CPU ID 2, CA EPIC for z/VSE uses the name 'F12.SORTWRK1'.

DSN Catalog

Catalog a separate dataset definition for each partition/CPU ID combination you intend to use. Specify the partition/CPU ID first, then a period, and then the dataset name. For the example above, you would create a definition for BG1.SORTWRK1 and F12.SORTWRK1. When using both CPU and partition independence, the partition ID precedes the CPU ID.

Note: If you actually specify === in the dataset name when it is controlled, the === is replaced by the partition and CPU ID in which the catalog operation occurs.

JCL Requirements

1. For both DLBL and DD statements, prefix your dataset name with equal signs (=) and a period (.), as follows:

Format	Specifies...
=.dataset.name	CPU independence only
==.dataset.name	Partition independence only
===.dataset.name	Both CPU and partition independence

2. For DD statements, you can substitute two ampersands (&&) for the equal signs and omit the period. If you are concerned about MVS compatibility, using the ampersands can be an advantage because they signify a temporary dataset to MVS. Note that MVS requires that ampersands be specified as the first two characters of the dataset name. See the DSN parameter for more information.

Example

This dataset will have a unique name in any partition or CPU in which it used.

```
// DLBL WORKOUT,'===.WORKFILE.1'
```

Common Work Datasets (CWDS)

The easiest way to control sort work, compiler work areas and any other dataset that does not need to be kept after end-of-job is to use the common work dataset (CWDS) feature.

When a dataset name begins with three dollar signs and a period ($$$., or £££. in the U.K.), CA EPIC for z/VSE recognizes it as a CWDS. The $$$ (or £££) in the dataset name are replaced by the CPU ID and partition ID in which the job is executed. This substitution is the reverse of "===" datasets, to avoid any conflicts if both are used with the same dataset name. In addition, a CWDS is automatically deleted at end-of-job, even if the job cancels.

Since each version is deleted at the end of every job, no versions are recorded in the DSN Catalog or kept on disk. If you attempt to open a version as input in a new job, the job cancels with an appropriate error message.

CWDS requires that you catalog only one dataset name, unlike CPU/partition independence which requires you to catalog a partition/CPU independent dataset name for every partition and machine combination in which the dataset could be used.

DSN Catalog

1. Catalog the dataset name beginning with '$$$.' ('£££.' in the U.K.). Use the remaining 40 characters to name the dataset.

2. Specify as work ("WRK" using Online Manager, or WRK=YES using TSIDMNT).

3. Cycle, retention period, and DLA retention are ignored if you specify them.

Example

This control statement catalogs a CWDS using TSIDMNT.

```
CAT   '$$$.SORTWK1',WRK=YES,POL=POOL01,EXT=120
```

JCL Requirements

Specify the dataset name beginning with '$$$' ('£££'. in the U.K.). No delete options are necessary on the dataset's DLBL or DD statement.

TYPEFLE=WORK DTFs

Datasets that have TYPEFLE=WORK DTFs are processed differently from other sequential datasets. Common TYPEFLE=WORK DTFs include compiler work datasets and FORTRAN disk datasets.

CA EPIC for z/VSE can process TYPEFLE=WORK DTFs on disk only (DTFSD). You can, of course, still use TYPEFLE=WORK DTFs with tape if the tape dataset is uncontrolled.

TYPEFLE=WORK DTFs can be controlled as Start Track 1 or cataloged datasets. However, we strongly recommend that you use the Catalog to eliminate many allocation problems. For example, CA EPIC for z/VSE automatically reuses the same extents used by an earlier version of a cataloged dataset, if one exists. . If you use Start Track 1 datasets, the old dataset is deleted the next time a dataset with the same dataset name is opened.

Secondary extents are not allowed.

DSN Catalog

1. Catalog the dataset as work ("WRK" using Online Manager, or WRK=YES using TSIDMNT).

2. You can inhibit LUB allocation for the dataset by specifying a SYS number in the Catalog.

JCL Requirements

Inhibit LUB allocation if this is not done through the Catalog. For DLBLs, specify the DLBL option "N" in the dataset name. For DD statements, include the OPTION=SYSnnn parameter.

Compiler Work Datasets

Compiler work datasets have the same requirements as TYPEFLE=WORK DTF datasets. Refer to "TYPEFLE=WORK DTF's" above, for those requirements.

In addition, observe the following requirements described below.

DSN Catalog

Compiler work datasets should be controlled with a disk pool that contains volumes of the same device type as those to which SYS001, SYS002,SYS003, SYS004, and SYSLNK are permanently assigned.

JCL Requirements

1. Each compiler does a pre-open check to determine the device type prior to opening its work datasets. Logical units SYS001, SYS002, SYS003, SYS004, and SYSLNK must therefore be pre-assigned to the same device type to be used for the compiler work datasets. You can use permanent or temporary assignments to accomplish this.
2. All compiler work dataset labels must be submitted before the compiler is executed.
3. LUB allocation is inhibited automatically, since the DTF names begin with IJSYS.

Example

```
// JOB COMPILE AND LINK EDIT
// ASSGN SYS001,SYSWK1
// ASSGN SYS002,SYSWK1
// ASSGN SYS003,SYSWK1
// ASSGN SYS004,SYSWK1
// ASSGN SYSLNK,SYSWK1
// DLBL IJSYS01,'$$$.IJSYS01'
// EXTENT SYS001
// DLBL IJSYS02,'$$$.IJSYS02'
// EXTENT SYS002
// DLBL IJSYS03,'$$$.IJSYS03'
// EXTENT SYS003
// DLBL IJSYS04,'$$$.IJSYS04'
// EXTENT SYS004
// DLBL IJSYSLN,'$$$.IJSYSLN'
// OPTION CATAL
 PHASE PROGX,*
// EXEC FCOBOL
 (COBOL SOURCE)
/*
// EXEC LNKEDT
/&
```

Uncontrolled Datasets

Any dataset that is not controlled using the DSN Catalog or the Start Track 1 features is called an uncontrolled dataset. With a few exceptions, uncontrolled datasets are not provided any additional features than are provided using standard VSE.

For uncontrolled tape datasets, CA EPIC for z/VSE provides protection for all output tapes to ensure that any tape volume that is still in an active status is not used. Additionally, if the configuration option UNCTL is set to NO, uncontrolled tape datasets cannot use any tape in the DSN Catalog even if it is in a scratch status. VSE ASSGN statements are not required for uncontrolled tape datasets. If there is no assignment, the operator is prompted to specify an available tape drive which can be used. Unlike controlled tape datasets, the assignment is not released at close unless the configuration option UNCTAP is set to YES.

Uncontrolled disk datasets can be placed within the extents of a disk pool provided they are created with valid expiration dates. ASSGN and EXTENT statements are normally required, but if the configuration option UNCDSK is set to YES, the ASSGN statement is optional.

JCL Requirements

No special options are required for uncontrolled datasets.

Physical IOCS Processing

VSE system utilities, sort programs, report writers, database utilities, and many other programs process datasets with physical IOCS (PIOCS).

Many different issues can come into play when a program uses PIOCS to process datasets that are under CA EPIC for z/VSE control. If you familiarize yourself with those described below, you can make any PIOCS program work with CA EPIC for z/VSE-controlled datasets. When you are familiar with these points, please review the sections on program-specific instructions.

PIOCS issues

1. When a program requires a specific logical unit, LUB allocation must be inhibited for every dataset processed in that job step. . . There are some cases in which LUB allocation is inhibited automatically. See "LUB Allocation and the ASSGN Statement" in this chapter for details.

2. Satisfy pre-open checking:

   a. Make an assignment to a device type, using any of the following techniques:

      - make a permanent assignment, as for SYS001 through SYS005 and SYSLNK
      - make a temporary assignment
      - use the DD OPTION=SYSnnn parameter
      - execute TSIDSRT (for SORT) or TSIDASS ASSGN

      If you make a disk assignment using the VSE ASSGN statement, assign the dataset to a device of the same type to be used for the dataset. For example, you can assign a disk dataset with the following statement:

      ```
      // ASSGN SYS003,DISK,VOL=SYSWK1,SHR
      ```

      If SYSWK1 is a 3380, for example, your dataset can be created on any 3380 volume.

   b. For other programs like DITTO, some IDCAMS functions, and IBM's SM2 sort, include an EXTENT card with the assigned logical unit number.

3. Many programs that use PIOCS do not issue an "open" request that CA EPIC for z/VSE can recognize. In this case, you can satisfy CA EPIC for z/VSE's "open" processing requirements by executing the TSIDASS OPEN function to simulate the opening of both input and output datasets. This is required only when the program does not issue an open, or does I/O to the dataset before it is opened. Examples include some DITTO functions, LIBR stand-alone backups and most FORTRAN compilers.

4. Use caution when using CA EPIC for z/VSE's concatenation facility. CA EPIC for z/VSE cannot recognize end-of-file processing for many PIOCS programs, so CA EPIC for z/VSE's dataset concatenation may not work for these programs.

5. Most PIOCS programs do not honor CA EPIC for z/VSE's dataset reblocking facility. Alternate block sizes must usually be specified by submitting control statements to the program. For sorts, TSIDSRT can be used to automatically update the control statements with the CA EPIC for z/VSE block size.

6. Generic device independence (GDI) *cannot* be used with PIOCS.

7. Some PIOCS programs do not provide the address of the last record written, so CA EPIC for z/VSE may not be able to truncate the dataset.

Utility Programs

CA EPIC for z/VSE provides the following utility programs to facilitate processing with PIOCS programs:

Program	Can be used for...
TSIDASS	Pre-open assignment checking and simulation of open processing
TSIDSRT	Sort programs
TSIDDEL	Deleting disk work datasets that have not been closed

These programs are discussed in detail in Chapter 8, "CA EPIC for z/VSE Utilities"

Sorting with CA EPIC for z/VSE

This section describes the considerations necessary to use CA EPIC for z/VSE controlled datasets with three widely used VSE sort programs:

- SYNCSORT
- CA SORT
- DFSORT

All VSE sort programs do pre-open assignment checking and use PIOCS to read and write their datasets. Therefore, some special considerations are usually necessary when accessing CA EPIC for z/VSE-controlled datasets with a sort program. These considerations fall into three categories:

- Satisfaction of pre-open assignment check requirements
- Deletion of the SORTWK dataset(s)
- Acquisition and recognition of necessary DTF information by CA EPIC for z/VSE

Pre-open assignment checking requirements can be met using JCL changes. CA EPIC for z/VSE automatically deletes controlled SORTWK datasets at EOJ. DTF information must be supplied by the sort for any datasets it has opened.

CA EPIC for z/VSE Sort Interface (TSIDSRT)

The CA EPIC for z/VSE sort interface program (TSIDSRT) addresses all the considerations described above, and usually requires no JCL changes except the phase name on the EXEC statement. TSIDSRT does the following:

1. Reads the sort control statements and modifies the input BLKSIZE to match the block size in the CA EPIC for z/VSE Catalog.
2. Assigns tape datasets to a "dummy" tape device and opens disk datasets to satisfy pre-open checking.
3. Calls the sort program.
4. Uses the sort program's E35 exit routine to write the SORTOUT dataset. This allows access to DTF information necessary to support such CA EPIC for z/VSE options as disk dataset truncation, reblocking, and logging of block size, record size and record count.
5. Closes SORTWK datasets to enable them to be deleted.

   The recommended method of using sort programs with CA EPIC for z/VSE-controlled datasets is to use TSIDSRT.

Sort Specifics

Many CA EPIC for z/VSE features rely upon DTF information that is automatically provided by the LIOCS access methods. CA EPIC for z/VSE also relies upon the standard open/close/end-of-extent/end-of-reel processing used by LIOCS. Because all sort programs use PIOCS to process data, the functionality of CA EPIC for z/VSE features varies depending upon which sort product is in use.

The following table details the operation of specific CA EPIC for z/VSE features when used with the sort programs. Its information is based upon DFSORT release 3.4, CA-SORT release 9.0, and SYNCSORT release 2.2c. These considerations apply only to datasets that are opened directly by the respective sort program. In addition:

When executing sort using the VSE EXEC statement, these considerations apply to SORTIN, SORTOUT and SORTWK.

When the sort is called internally by another program, these considerations usually apply only to SORTWK.

When executing TSIDSRT, these considerations apply only to SORTIN and SORTWK.

CA EPIC for z/VSE feature	DF SORT	CA-SORT	SYNCSORT	TSIDSRT
Reblocking	No	No	Yes	Yes
Truncation of disk output dataset	No	No	Yes	Yes
Deletion of SORTWK (using close)	Yes	Yes	Yes	Yes
DTF information posting	No	No	Yes	Yes
LUB allocation	Yes	Yes	Yes	Based on sort program.
Disk dataset concatenation	Yes	Yes	Yes	Based on sort program.
Tape dataset concatenation	Yes	No	Yes	Based on sort program.
Secondary SORTWK allocation	Yes	Yes	Yes	Based on sort program.
Limited pre-open assignment checks	Yes	Yes	Yes	Yes

SYNCSORT and CA SORT

CA-SORT, DFSORT, and SYNCSORT offer options to improve compatibility with CA EPIC for z/VSE. See the sort product's installation and configuration instructions for details about these options. See the *Installation and System Guide* for details about which SYNCSORT options to apply.

In addition, there are CA EPIC for z/VSE special options which must be applied for use with both sorts. See the *Installation and System Guide*, "Compatibility With Other Software" for details.

With these options activated, LUB allocation and secondary SORTWK allocations are supported, and SORTWK datasets are closed to allow them to be deleted. In addition, SYNCSORT can communicate directly with CA EPIC for z/VSE to allow truncation of disk output datasets, and to allow block/record sizes and counts to be stored in the DSN Catalog and on tape HDR2 labels.

CA-SORT, DFSORT, and SYNCSORT also use an improved type of pre-open checking which reduces the need for pre-open assignments, thereby simplifying JCL. Before checking assignments, these sorts access the VSE label area to determine whether the SORTINx and SORTOUT datasets are on tape or disk. If a matching disk or tape label is found, pre-open assignment checking is bypassed if the associated SYS number is assigned to a disk or tape device, or set to unassigned (UA). The sort then correctly assumes that CA EPIC for z/VSE will make the necessary assignment when the dataset is opened.

Recommendations:

If datasets that have been reblocked by CA EPIC for z/VSE will be read by a sort program, TSIDSRT must be executed if the block size specified in the sort control statement is incorrect.

If CA EPIC for z/VSE reblocking is not being used, SYNCSORT and CA SORT can be executed directly. However, CA SORT users may prefer to use TSIDSRT for disk output datasets to enable dataset truncation.

IBM DFSORT

Beginning with release 3.3, DFSORT offers File Management System options that offer improved handshaking with CA EPIC for z/VSE. On older releases of DFSORT or without the FMS=YES option (in the ILUINST macro), see the section on IBM SM-2 (below) for recommendations.

With the FMS=YES option, DFSORT users can take advantage of improved pre-open checking, LUB allocation for SORTWK files, and secondary allocation for SORTWK files. Unless EPIC reblocking of input files is required, you should be able to execute DFSORT directly to enjoy improved performance compared to TSIDSRT.

IBM SM-2

We recommend you use TSIDSRT to satisfy the processing requirements of IBM's sort program (SM-2). TSIDSRT satisfies pre-open assignment checking, makes actual device assignments, allows truncation, and closes the SORTWK datasets so that they can be deleted.

To execute the SM-2 sort directly, use the TSIDASS ASSGN function to make the necessary pre-open assignments for tape datasets. For disk datasets, the pre-open assignments can be permanent or can be made in the JCL with VSE ASSGN statements. In addition, an EXTENT statement that contains the correct SYS number is required for all disk datasets defined with DLBL statements.

In either case, LUB allocation is inhibited automatically if you use DTF names beginning with "SORT". If you use alternate DTF names, you must inhibit LUB allocation.

SORTWK Datasets

To provide optimal utilization of disk work space, SORTWK datasets should be defined as temporary datasets and deleted as soon as they are no longer in use.

DSN Catalog

1. Catalog SORTWK datasets as work ("WRK" using Online Manager, or WRK=YES using TSIDMNT).
2. Use the CWDS facility or partition/CPU-independent datasets so that the datasets can be processed in any partition without conflict.

JCL Requirements

Various CA EPIC for z/VSE options can be used to delete SORTWK areas upon completion of a sort.

1. Controlled SORTWK datasets are deleted automatically at end of job.
2. Common work datasets ($$$) are also deleted at end of job.
3. If the sort program closes SORTWK (as CA SORT and SYNCSORT do), use the DLBL "D" option or the DD DISP=(,DELETE) parameter to delete SORTWK at close.
4. If the sort program does not close SORTWK, (SM-2 does not), execute TSIDDEL after every sort step.

For jobs that contain multiple sort steps, SORTWK datasets should be deleted after each sort step, using either (3) or (4) above as applicable.

Pre-Open Checking

All sorts check specific SYS numbers to determine the device type for each dataset that is to be opened. The sort searches the following sources (in the order listed) for the SYS number to be used for each dataset. The search stops as soon as a SYS number is found.

1. The SYS number found in the VSE label area for disk datasets (from the EXTENT statement or the DD OPTION=SYSnnn parameter).
2. Sort control statements (SORTIN, SORTOUT, SORTWK keywords); which TSIDSRT can modify if necessary.
3. Sort defaults (standard or user-specified: SYS001 for SORTOUT, SYS002 for SORTIN1, SYS003 for SORTWK, and so on).

Note: If LUB allocation is active for sort datasets, the SYS numbers checked by the sort in its pre-open processing may not be the same SYS numbers assigned by CA EPIC for z/VSE when the open actually occurs.

JCL Requirements

Pre-open assignments can be met by any of the following means:

Media	Method
Disk	1. Permanent assignments (in partition start-up procedure) 2. Temporary assignments (in each job)
Tape	Execute TSIDASS with the ASSGN function.
Tape or Disk	1. Use the DD OPTION=SYSnnn parameter. 2. Execute TSIDSRT.

One popular method of satisfying pre-open requirements for the sort (and compilers) is to permanently assign SYS001 through SYS005 to disk in every partition in which the compiler may be used. The disk assigned must be the same device type as the device type in the CA EPIC for z/VSE disk pool. This allows the processing of up to five datasets on disk (and tape with SYNCSORT and CA SORT) with no additional JCL necessary.

Example: SYNCSORT, CASORT, SM-2

This example executes TSIDSRT, which automatically makes all the necessary pre-open assignments. In addition, all CA EPIC for z/VSE features are fully supported.

```
// JOB SORT
// TLBL SORTIN1,'DAILY.TRANS',,,,,,2
// DLBL SORTOUT,'SORTED.DAILY.TRANS'
// DLBL SORTWK1,'$$$.SORTWK1,(D)'
// EXEC TSIDSRT
 SORT FIELDS=(33,4,A),FORMAT=BI,WORK=1,FILES=1
 RECORD TYPE=F,LENGTH=80
 INPFIL BLKSIZE=1600
 OUTFIL BLKSIZE=4000
 END
/*
/&
```

Example: SYNCSORT, CASORT, DFSORT

In this example, SORT is executed directly, for best performance. The pre-open assignment for the tape dataset is made using the TSIDASS ASSGN function. Pre-open assignments for the disk datasets are made with VSE ASSGN statements. The "D" option on the DLBL for SORTWK1 causes the SORTWK1 dataset to be deleted when it is closed by the sort.

```
// JOB SORT
// TLBL SORTIN1,'DAILY.TRANS',,,,,,2
// ASSGN SYS001,DISK,VOL=SYSWK1,SHR
// DLBL SORTOUT,'SORTED.DAILY.TRANS'
// ASSGN SYS003,DISK,VOL=SYSWK1,SHR
// DLBL SORTWK1,'$$$.SORTWK1,(D)'
// EXEC TSIDASS
 ASSGN SYS002
/*
// EXEC SORT
 SORT FIELDS=(33,4,A),FORMAT=BI,WORK=1,FILES=1
 RECORD TYPE=F,LENGTH=80
 INPFIL BLKSIZE=1600
 OUTFIL BLKSIZE=4000
 END
/*
/&
```

Example: SYNCSORT, CASORT, DFSORT (FMS=YES)

This example assumes that SYS001, SYS002 and SYS003 have been permanently assigned to disk. Therefore, no assignments are necessary in the jobstream. (This requires the TMSYS=YES option for SYNCSORT.)

```
// JOB SORT
// TLBL SORTIN1,'DAILY.TRANS',,,,,,2
// DLBL SORTOUT,'SORTED.DAILY.TRANS'
// DLBL SORTWK1,'$$$.SORTWK1,(D)'
// EXEC SORT
 SORT FIELDS=(33,4,A),FORMAT=BI,WORK=1,FILES=1
 RECORD TYPE=F,LENGTH=80
 INPFIL BLKSIZE=1600
 OUTFIL BLKSIZE=4000
 END
/*
/&
```

Example: SYNCSORT, CASORT, SM-2

In this example, the DD OPTION=SYSnnn parameter is used to satisfy pre-open checking. This job works with all three sort programs, with no special considerations.

```
// JOB XYZ
*
* //SORTOUT DD DSN='SORTED.DAILY.TRANS',
* /            DISP=(NEW,KEEP,DELETE),
* //           UNIT=DISK,
* //           OPTION=SYS001
* //SORTIN1 DD DSN='DAILY.TRANS',
* //           DISP=(OLD,DELETE,KEEP),
* //           UNIT=TAPE,
* /            OPTION=SYS002
* //SORTWK1 DD DSN='$$$.SORTWK1'
* //           DISP=(NEW,DELETE,DELETE),
* //           UNIT=DISK,
* //           OPTION=SYS003
// EXEC SORT
 SORT FIELDS=(33,4,A),FORMAT=BI,WORK=1,FILES=1
 RECORD TYPE=F,LENGTH=80
 INPFIL BLKSIZE=1600
 OUTFIL BLKSIZE=4000
 END
/*
/&
```

With SM-2, the following step should be included to delete the SORTWK1 dataset.

```
// EXEC TSIDDEL
SORTWK1
/*
```

SM-2 Example

In this example, sort is executed directly. To satisfy SM2's requirements, "dummy" EXTENT statements are included for the disk datasets. Pre-open assignments are made using the TSIDASS ASSGN function for the tape dataset and VSE ASSGN statements for the disk datasets. Lastly, TSIDDEL is executed to delete the SORTWK1 dataset.

```
// JOB SORT
// TLBL SORTIN1,'DAILY.TRANS',,,,,,2
// ASSGN SYS001,DISK,VOL=SYSWK1,SHR
// DLBL SORTOUT,'SORTED.DAILY.TRANS'
// EXTENT SYS001
// ASSGN SYS003,DISK,VOL=SYSWK1,SHR
// DLBL SORTWK1,'$$$.SORTWK1,(D)'
// EXTENT SYS003
// EXEC TSIDASS
 ASSGN SYS002
/*
// EXEC SORT
 SORT FIELDS=(33,4,A),FORMAT=BI,WORK=1,FILES=1
 RECORD TYPE=F,LENGTH=80
 INPFIL BLKSIZE=1600
 OUTFIL BLKSIZE=4000
 END
/*
// EXEC TSIDDEL
SORTWK1
/*
/&
```

IDCAMS BACKUP and RESTORE

To avoid the following IDCAMS considerations, we recommend that you use CA's FAVER product as your VSAM backup/restore utility. FAVER is compatible with EPIC and easy to use.

DSN Catalog

All tape datasets must be controlled (cataloged).

JCL Requirements

LUB allocation must be inhibited for both input and output tape datasets. IDCAMS uses SYS005 for the tape during a backup and SYS004 for the tape during a restore. You can define an alternate DTF name for the tape dataset using the IDCAMS STDLABEL parameter.

To satisfy pre-open assignment checking for the tape, execute TSIDASS or use the DD OPTION=SYSnnn parameter.

Restrictions

Due to IDCAMS restrictions, you cannot use CA EPIC for z/VSE multi-dataset groups for IDCAMS BACKUP datasets. Also, GDI cannot be used to route backup datasets to disk. They can only be written to tape.

Selective Dataset RESTORE

When you do a selective dataset restore from an IDCAMS BACKUP tape, the CA EPIC for z/VSE mount request will conflict with the IDCAMS mount request for secondary volumes if only one TLBL is used in the job step. This occurs because the IDCAMS selective dataset RESTORE opens and closes each volume as a separate dataset, expecting to process only the volume containing the dataset to be restored.

This requires a manual check of the backup object cross reference report to determine which volume to mount. Most users prefer to let IDCAMS read through all the volumes to locate the dataset to be restored. However, the separate opens for each volume cause CA EPIC for z/VSE to always request that the first volume be mounted.

To avoid this problem, we recommend that you restrict backups to a single volume, or restore all the datasets on the backup tape. This ensures complete CA EPIC for z/VSE control without any special considerations. If this is not possible, the following job can be used.

Include a TLBL specifying the volume sequence number for every volume in the backup dataset version. Arrange the TLBLs in ascending sequence by volume sequence number beginning with the first volume (or the starting volume sequence number if it is known). Additional TLBLs beyond the actual number of volumes that may be read are permissible and will not affect processing.

Since the volume serial number of the volume that contains the dataset to be restored is not known, the TLBLs begin with volume sequence number 1. The TLBL option code "82" tells CA EPIC for z/VSE to inhibit LUB allocation, unload each volume and drop the TLBL at close. This causes CA EPIC for z/VSE to request the next volume upon each successive open.

Reply ENTER to any IDC402A MOUNT VOLUME nnn OR HIGHER messages that occur. CA EPIC for z/VSE's AVR reassigns SYS004 as necessary for each volume.

IDCAMS REPRO

You can use CA EPIC for z/VSE-controlled multi-dataset groups for IDCAMS REPRO datasets on tape. See "Multi-Dataset Processing" in this chapter for details.

IDCAMS REPRO datasets can also be written to disk. However, unlike tape, the IDCAMS REPRO function inspects both the VSE label area and the assignment for SYS005 prior to issuing an open, so additional JCL considerations may be necessary.

DSN Catalog

Every output tape dataset must be controlled. Secondary datasets must be cataloged as subdatasets.

JCL Requirements for Tape

Each tape dataset must be defined with a TLBL or a DD statement.

LUB allocation must be inhibited.

To satisfy pre-open assignment checking for the tape dataset, use the DD OPTION=SYSnnn parameter or execute the TSIDASS ASSGN function. Specifying SYS005 for output, SYS004 for input.

JCL Requirements for Disk

Each dataset must be defined with a DLBL and EXTENT (for controlled datasets, only the SYS number is required on the EXTENT), or with a DD statement.

LUB allocation is supported.

To satisfy pre-open assignment checking for sequential disk datasets, use the DD OPTION=SYSnnn parameter or include a generic VSE ASSGN statement for the SYS number that is specified in the EXTENT statement.

IDCAMS IMPORT/EXPORT

DSN Catalog

No special entries are required.

JCL Requirements

Use the OPTION=SYSnnn parameter or execute the TSIDASS ASSGN function to satisfy pre-open assignment checking.

Inhibit LUB allocation for the tape dataset in each step.

LIBR

CA EPIC for z/VSE can control LIBR backup tapes. JCL requirements vary depending on the VSE release.

DSN Catalog

No special entries are required.

JCL Requirements

For all releases: Specify the following TLBL options:

- Inhibit LUB allocation (64)
- Hold the assignment (8)

You can also include an MTC command to unload the tape.

For all releases:

- (Backup only) Execute the TSIDASS OPEN function if the LIBR RESTORE=STANDALONE parameter is specified.
- (Restore only) Do not execute TSIDASS at all.

Control Statements

For all releases: Always restrict control statements to a single backup per LIBR execution.

Use the TAPELABEL parameter on the BACKUP and RESTORE statements.

DITTO

CA EPIC for z/VSE can control DITTO file functions. However, we do not recommend using DITTO. We recommend using TSIDDTD or TSIDDMP for equivalent functions with simpler JCL requirements.

Note

If you create uncontrolled disk datasets with DITTO file functions, and you want to process them later as controlled, you must renumber their extent sequence numbers with the TSIDVUT RENUM command.

DSN Catalog

No special entries are required.

JCL Requirements

For output tape:

1. Inhibit LUB allocation.
2. Execute the TSIDASS OPEN function before executing DITTO.

For input tape:

1. Inhibit LUB allocation.
2. Execute the TSIDASS OPEN function before executing DITTO.

You must respond ACCEPT to CA EPIC for z/VSE message EP049. Two EP005 label messages are issued. If you use the CA EPIC for z/VSE reblocking option, special configuration option EPS022 must be set to YES.

For output disk:

1. Inhibit LUB allocation.
2. Provide an EXTENT statement with a SYS number and an ASSGN statement assigning that SYS number to a device of the same type as the CA EPIC for z/VSE disk pool to be used.

Alternatively, you can use a DD statement with the OPTION=SYSnnn parameter.

For input disk:

1. Provide an EXTENT statement with a SYS number.
2. Execute the TSIDASS OPEN function before executing DITTO.

Alternatively, you can use a DD statement with the OPTION=SYSnnn parameter.

Two EP053 messages are issued. If you use the CA EPIC for z/VSE reblocking option, special configuration option EPS022 must be set to YES.

COPYING TAPES WITH DITTO TT FUNCTION

The DITTO TT function can be used in conjunction with CA EPIC for z/VSE to copy tapes which do not follow standard LIOCS tape format conventions (for example, tapes created by some utility programs using PIOCS). Use TSIDDTD to copy standard LIOCS formatted tapes. (See "Supported Tape Formats" under TSIDDTD for further information). When copying tapes with DITTO TT, the following restrictions and considerations apply:

1. The copied data MUST physically fit onto a single volume (since no EOV processing is done by DITTO TT). If it does not, cancel the job and try another tape.
2. The number given for NFILES= must be sufficiently large to copy all the data segments on the tape, up to and including the EOF or EOV labels. DITTO considers every data segment followed by a tape mark as one "file". Some PIOCS programs write multiple "files" (by this definition) as a single logical dataset. If you are unsure how many "files" are on the input tape, specify a number high enough to cause DITTO to copy until it encounters an unrecoverable I/O error. This usually signifies the end of all data on the tape.
3. To copy tapes containing multiple datasets (by VSE's definition), the following three-step job must be run for each dataset and the number given for NFILES= must be exactly correct.

POWER Job Accounting

CA EPIC for z/VSE can control POWER job accounting tapes.

DSN Catalog

No special entries are required.

JCL Requirements

The following must be supplied in system or partition standard labels or in the POWER start-up job stream, using a TLBL with the correct DTF name. Specify TLBL option "66" to inhibit LUB allocation, drop the label at close, and rewind the tape at close.

You cannot use CA EPIC for z/VSE DD statements for POWER Job Accounting tapes.

When you dump the job accounting dataset, specify the DTF name as part of the command:

```
J cuu,,dtfname
```

Processing with COBOL II

CA EPIC for z/VSE requires that COBOL II programs are compiled with the Library Management Feature (the RES CBL option) in order to function properly.

In some instances, you may have COBOL II software which was not compiled with the Library Management Feature active, and you have no access to the source code. To run these programs with CA EPIC for z/VSE, special JCL is required.

DSN Catalog

No special entries are required.

JCL Requirements

For all controlled disk datasets: Pre-open each dataset with the TSIDASS OPEN function.

For all uncontrolled disk datasets: The proper volume serial number must be specified on the EXTENT statement, and you must make an assignment to the proper device with a VSE ASSGN statement.

For all tape datasets: Assign the dataset to a tape drive with a VSE ASSGN statement or with the TSIDASS ASSGN function.

Examples

The following example processes a controlled input disk dataset. The dataset is pre-opened with TSIDASS.

```
// DLBL DISKIN, 'TEST.INPUT.DISK, (L)'
// EXEC TSIDASS
 OPEN DISKIN,SYS020,INPUT
/*
// EXEC PROGA
```

The following example processes an uncontrolled input disk dataset. VSE ASSGN and EXTENT statements are used to satisfy the COBOL II pre-open process.

```
// ASSGN SYS020,154
// DLBL DISKIN,'TEST.INPUT.DISK'
// EXTENT SYS020,VOL154,1,0,15,75
// EXEC PROGA
```

The following example processes a controlled tape dataset. The TSIDASS ASSGN function is used to satisfy the COBOL II pre-open process.

```
// DLBL TAPEIN,'TEST.TAPE',,,,,64
// EXEC TSIDASS
 ASSGN SYS011
/*
// EXEC PROGA
```

Using CA EPIC for z/VSE's JCL Features

You can use various JCL features as follows.

Restarting Programs from a Checkpoint

To prepare to restart programs from a checkpoint:

- Close all output datasets with the TSIDMNT CLOSE function.
- Use the console log to determine which input and output tape volumes have been used and which checkpoints have been taken.

DSN Catalog

All datasets must be cataloged. Start Track 1 datasets cannot be used with checkpoint/restart. You must specify the SYS number in the catalog entry for disk datasets. You can also specify it for tape datasets.

JCL Requirements

To prepare the restart job:

1. Ensure that you use the same job name, supervisor, and partition.
2. Execute the TSIDASS RSTRT function in the step just prior to the VSE RESTRT command. See "Pre-Open Processing" for details.
3. Define each controlled dataset in the TSIDASS RSTRT step. Remember that the output datasets are now the current versions in the DSN Catalog, so submit all output RSTRT statements first, input statements last.

   For tape, specify the volume sequence number on the label, if you restart with volume 2 or later. If you are processing a multi-dataset group, CA EPIC for z/VSE positions the tape automatically. Inhibit LUB allocation.

   For disk, supply the EXT parameter reflecting the last extent opened.
4. Follow the TSIDASS step with a VSE ASSGN statement for the checkpoint dataset. Use the logical unit number that was used when the dataset was opened in the previous execution. See the EP039 and EP053 messages to find these numbers.
5. The SYS number used for the checkpoint dataset must be assigned with a VSE ASSGN statement if checkpoints are not embedded in the output dataset.
6. Execute the restart job.

Example

In this example, TSIDASS restarts both input and output datasets at volume 2, using the same SYS numbers that were used when the checkpoint was taken. The TLBL option "64" inhibits LUB allocation. Because the checkpoints are embedded (being written to the same output tape as the output dataset using an assignment of SYS019), a VSE ASSGN statement is used to assign SYS019 to the checkpoint dataset. The job is restarted at checkpoint 3.

```
// JOB TESTCHKP
// TLBL TAPIN,'TEST.FILE1',,,2,,,64
// TLBL TAPOUT,'TEST.FILE2',,,2,,,64
// EXEC TSIDASS
RSTRT TAPOUT,SYS021,OUTPUT
RSTRT TAPIN,SYS020,INPUT
/*
// ASSGN SYS019,SYS021
// RSTRT SYS019,3
/&
```

Choosing the Device Type

You can choose the device type as follows.

Generic Device Independence

Generic Device Independence (GDI) and Generic Device Independence for Unit Record Devices (GDIUR), sometimes called storage media independence, allows you to choose the type of media on which your dataset will be stored. You can process the dataset on either tape and disk, no matter how the dataset is defined in your program.

In addition, Generic Device Independence for Unit Record devices (GDIUR), allows you to use disk or tape for datasets that have been defined in your programs as unit record devices (DTF types DTFCD, DTFPR, DTFCP, and DTFDI). GDIUR can be used for logical units SYS000 through SYS255, and for SYSIPT, SYSLST, and SYSPCH.

GDIUR greatly simplifies the use of disk for VSE datasets that use the logical units SYSIPT, SYSLST, and SYSPCH. The following standard VSE JCL could be used to create a disk dataset on logical unit SYSPCH, and then read that dataset with logical unit SYSIPT:

```
// DLBL IJSYSPH,'COBOL.PROGRAM',0
// EXTENT SYSPCH,VOL251,1,0,200,2000
   ASSGN SYSPCH,DISK,VOL=VOL251,SHR
// EXEC DFHECP1$
.
.
   CLOSE SYSPCH,PUNCH
// DLBL IJSYSIN,'COBOL.PROGRAM'
// EXTENT SYSIPT,VOL251
   ASSGN SYSIPT,DISK,VOL=VOL251,SHR
// EXEC FCOBOL
.
.
   CLOSE SYSIPT,SYSRDR
```

There are two problems with this approach:

- If the first step (DFHECP1$) cancels, SYSPCH will remain permanently assigned to disk. Manual intervention or additional JCL is needed to re-assign SYSPCH correctly.
- If the second step (FCOBOL) cancels, SYSIPT remains permanently assigned to disk. Manual intervention or additional JCL is needed to re-assign SYSIPT correctly.

In addition, the JCL required is complex and lengthy, especially when compared to the following CA EPIC for z/VSE JCL which accomplishes the same task:

```
* //IJSYSPH DD DSN=$$$.COBOL,UNIT=DISK,DISP=(,KEEP,DELETE)
// EXEC DFHECP1$
.
.
* //IJSYSIN DD DSN=$$$.COBOL,DISP=(OLD,DELETE)
```

Note that no ASSGN or CLOSE statements are needed, because GDIUR ensures that system logical units can never be left erroneously assigned to disk.

Programs that use GDIUR can also take advantage of CA EPIC for z/VSE's reblocking facility to save disk space and improve performance.

Eligibility

You can use GDI or GDIUR for uncontrolled and controlled datasets. A dataset normally processed as a Start Track 1 disk dataset would become uncontrolled if processed on tape with GDI.

GDIUR does not support RPG-II SYSLST processing.

GDIUR datasets cannot have secondary extents.

GDI applies to tape and disk sequential datasets (DTFMT or DTFSD). VSAM, direct access, and ISAM datasets are not eligible.

GDI and GDIUR datasets must be processed with logical IOCS.

Configuration Options

Generic device independence is controlled by the configuration options GDI and GDIUR.

Processing

Device independence can be invoked in any of the following ways:

- In response to a message. An insufficient space message or a request for a scratch tape can be answered with TAPE or DISK to force the use of a different device type. (GDI only)
- Automatically for input datasets. CA EPIC for z/VSE allocates the correct device type when controlled datasets are processed as input. (GDI only)
- Using JCL. The desired device type is indicated by your job control statements. For example, use a TLBL instead of a DLBL to specify that the dataset is to be on tape instead of disk. On a DD statement, specify UNIT=DISK instead of UNIT=TAPE to specify that the dataset is to be on disk instead of tape. (Both GDI and GDIUR)

DSN Catalog

Required information for tape and disk must be supplied in the dataset definition for controlled datasets. This includes pooling and allocation information. If it is not included, CA EPIC for z/VSE prompts the operator for the necessary input.

JCL Requirements

For uncontrolled tape or disk datasets, use the proper label (TLBL, DLBL, or a DD statement with the UNIT parameter).

For controlled datasets, no JCL changes are required for input (GDI only). For output GDI, and GDIUR input and output, invoke GDI by changing the label type (TLBL or DLBL) or device type (UNIT=type). No volume or allocation information needs to be supplied in the JCL.

When using GDIUR, it may be necessary to insert the OPTION=RESET parameter on the DD statement after the step to terminate GDIUR processing if subsequent steps reference the same system dataset DTF name and you do not need to control it further using GDIUR.

Note: For more information, see the RESET (see page 310) subparameter of the OPTION DD Statement parameter.

Example

This example routes punch output to a disk dataset using GDIUR.

```
* //IJSYSPH  DD  DSN='CARD.OUTPUT',DISP=(NEW,PASS),
* //            UNIT=DISK
// EXEC PROGRAM
/*
* //IJSYSPH  DD  OPTION=RESET
```

System Datasets and VSE Device Independence

Although we recommend that you use GDIUR, you can also control a system dataset that is routed to disk or tape using VSE device independence.

DSN Catalog

No special entries are required.

JCL Requirements

The dataset cannot have secondary extents.

The dataset must be permanently assigned, and label information must precede the assignment.

CA EPIC for z/VSE assigns VSE system datasets to the correct disk volume, so the assignment can specify any volume in the disk pool.

Example

This example routes SYSPCH output to an CA EPIC for z/VSE controlled work dataset, which is then processed using SYSIPT in the subsequent step.

```
// JOB ASSEMBLE AND CATALOG AN OBJECT MODULE
// DLBL IJSYSPH,'==.WORKFILE.1'
ASSGN SYSPCH,DISK,VOL=SYSWK1,SHR
// EXEC ASSEMBLY
  (SOURCE CODE)
/*
CLOSE SYSPCH,PUNCH
// DLBL IJSYSIN,'==.WORKFILE.1'
ASSGN SYSIPT, DISK,VOL=SYSWK1,SHR
// EXEC LIBR
CLOSE SYSIPT,READER
/&
```

Dummy Datasets and ASSGN IGNORE

Dummy processing and ASSIGN IGNORE are only supported for LIOCS sequential access methods.

DSN Catalog

No special entries are required.

JCL Requirements

Assign the logical unit to IGNORE, so that CA EPIC for z/VSE ignores the dataset.

Use an EXTENT statement for disk datasets defined with DLBLs.

For DD statements, specify the DUMMY parameter instead of the DSN parameter.

Examples

```
// ASSGN SYS020,IGN
// TLBL TAPEOUT,'DATA.SET.A'
// ASSGN SYS004,IGN
// DLBL DISKOUT,'DATA.SET.B(L)'
// EXTENT SYS004
* //TAPEOUT DD DUMMY
```

Concatenating Datasets

Dataset concatenation reads multiple versions of a dataset or multiple datasets with one open. This allows you to process a number of datasets as if they were a single dataset.

All versions of concatenated datasets must reside on the same device type. If the datasets have different block sizes, you must read the dataset with the largest block size first.

DSN Catalog

No special entries are required.

JCL Requirements

With TLBL or DLBL statements: Use a separate TLBL or DLBL statement for each dataset to be read. Datasets are processed in the order in which their label statements appear. Use the same DTF name for each statement and supply the concatenation option code for all except the last version: TLBL option "48" and DLBL option "E".

Important! Do not specify the concatenation code on the last label statement in a concatenation group!

Examples

The following example processes versions 1 through 5 of the DAILY.TRANS dataset on disk with one open. It also processes WEEKLY.TRANS and WEEKLY.RERUN on tape with another single open. Although there are mixed device types in the job, they are not mixed within a single open, so they meet the requirements for concatenation.

```
// DLBL INPUT1,'DAILY.TRANS,(E,V=5)'
// DLBL INPUT1,'DAILY.TRANS,(E,V=4)'
// DLBL INPUT1,'DAILY.TRANS,(E,V=3)'
// DLBL INPUT1,'DAILY.TRANS,(E,V=2)'
// DLBL INPUT1,'DAILY.TRANS'
// TLBL INPUT2,'WEEKLY.TRANS',,,,,,48
// TLBL INPUT2,'WEEKLY.RERUN'
// EXEC PROGRAM
```

With DD statements: Concatenation is automatic for all versions of a single dataset on input. Simply specify DSN=data.set.name without specifying a particular version. When all versions of a single dataset are concatenated, datasets are read in order from the most current version to the oldest.

In the following example, all versions of DAILY.TRANS are read with one open. For input with different dataset names, request concatenation explicitly. Specify the version number of each version to be read in the DSN parameter. Specify a ddname for the first statement only.

```
* //INPUT1  DD  DSN='DAILY.TRANS',DISP=(OLD,KEEP)
```

The following example concatenates versions 1 through 4 of the DAILY.TRANS dataset and version 1 of the DAILY.DEP dataset.

```
* //INPUT1  DD  DSN='DAILY.TRANS(3)',DISP=(OLD,KEEP)
* //        DD  DSN='DAILY.TRANS(2)',DISP=(OLD,KEEP)
* //        DD  DSN='DAILY.TRANS(1)',DISP=(OLD,KEEP)
* //        DD  DSN='DAILY.TRANS(0)',DISP=(OLD,KEEP)
* //        DD  DSN='DAILY.DEP(0)'DISP=(OLD,KEEP)
// EXEC PROGRAM
```

Processing Multiple Datasets with the Same DTF Name (Dataset Drop)

CA EPIC for z/VSE allows you to open multiple datasets with the same DTF name. Each dataset is opened and closed separately, in the order in which its label is submitted.

DSN Catalog

No special entries are required.

JCL Requirements

Specify the dataset drop option for all but the last dataset defined.

For...	Use...
DLBLs	T
TLBLs	16 or D
DD statements	Automatic. Just use multiple DD statements with the same ddname. *This technique is not compatible with MVS.*

Examples

Dataset drop is invoked for three datasets using the ddname TAPEIN.

```
* //TAPEIN DD DSN=PAYROLL.WEEKLY(2),DISP=(OLD,KEEP)
* //TAPEIN DD DSN=PAYROLL.WEEKLY(1),DISP=(OLD,KEEP)
* //TAPEIN DD DSN=PAYROLL.WEEKLY(0),DISP=(OLD,KEEP)
```

Dataset drop is invoked for three datasets using the DTF name TAPEIN. Option 16 is used for all but the last dataset with that DTF name.

```
// TLBL TAPEIN,'FILE1',,,,,16
// TLBL TAPEIN,'FILE2',,,,,16
// TLBL TAPEIN,'FILE3'
```

Running Test Jobs With Production Datasets

CA EPIC for z/VSE's test job feature allows you to run test jobs with production datasets without affecting production processing. Test job names are defined by the TSTJOB configuration option. The TSTJOB configuration option can include wildcard characters to allow greater flexibility in creating test job names. A question mark (?) specifies that any character in that position is a valid character. An asterisk (*) specifies that all characters in that position and any following are valid.

All versions of controlled datasets created using test jobs are recorded in the DSN Catalog with a status of "conditionally cataloged". Only such versions can be accessed by subsequent test job executions. Production jobs will not use the conditionally cataloged versions. These versions are retained until the following day, when they become eligible for scratch. If you close a conditionally cataloged version using Online Manager (or TSIDMNT) during that time, it becomes an active version (version 1), can be accessed by production jobs, and is maintained according to its EDD retention criteria.

DSN Catalog

No special entries are required.

JCL Requirements

To test a single dataset, specify a conditional catalog option on its label:

For...	Use...
DLBLs	C
TLBLs	32
DD statements	DISP=(xxx,CONDCAT)

To test an entire job, specify a valid test job name on the VSE JOB statement.

Examples

If TSTJOB=A*, the test job feature is activated by any job name beginning with A.

If TSTJOB=A?AAAA, the test job feature is activated by the job names ABAAAA, AXAAAA, A1AAAA, and so on.

If TSTJOB=A?A*, the test job feature is activated by the job names ABAAAA, AXA, ACAXX, and so on.

Chapter 8: CA EPIC for z/VSE Utilities

This section contains the following topics:

Overview

This chapter describes how to use CA EPIC for z/VSE JCL utilities that control files that are processed with physical IOCS (PIOCS) instead of logical IOCS (LIOCS). For more information about the recommended use of these utilities, see the chapter Running Jobs with CA EPIC for z/VSE. In addition, this chapter presents the dataset copy and dump utilities TSIDDTD and TSIDDMP and the VSE label access subroutines (TSIDGLB and TSIDPLB).

Pre-Open Processing (TSIDASS)

Use TSIDASS to process controlled files with programs that do not issue OPEN requests or that do pre-open checking of assignments. You can also use TSIDASS to make assignments to SYSRDR, SYSIPT, SYSLST, and SYSPCH.

Its functions are determined by control statements:

- ASSGN satisfies pre-open assignment checking for tape datasets
- OPEN opens datasets for programs which do not issue an OPEN request
- RSTRT assists with using IBM checkpoint restart facility

TSIDASS is executed immediately before the program that does pre-open assignment checking. Inhibit LUB allocation for all datasets in the TSIDASS step by using the appropriate option codes on the TLBL and DLBL statements in the step.

Note: Pre-open assignment checking can also be satisfied with the DD parameter OPTION=SYSnnn.

JCL Requirements

```
dataset labels
// EXEC TSIDASS
control statements
/*
// EXEC <program that performs pre-open assignment checking>
```

UPSI Values

None.

Dataset Requirements

Place the labels for all datasets *before* the EXEC statement for TSIDASS. Do not place any labels between the EXEC statement for TSIDASS and the EXEC statement for your program.

ASSGN Statements

Use the ASSGN function for tape datasets only. It satisfies pre-open assignment checking by assigning the logical unit to a dummy tape device. When the program issues an OPEN request for the dataset, CA EPIC for z/VSE performs its usual device assignment to the real device.

For sort programs, we recommend using TSIDSRT to perform this function instead of TSIDASS.

Note: For more information and JCL examples, see Sorting with CA EPIC (see page 248).

You can also use ASSGN to make assignments to the system logical units SYSRDR, SYSIPT, SYSLST, and SYSPCH. When you make these assignments, be sure to submit their control statements last, because TSIDASS stops processing control statements after one of these assignments is made.

To make the assignment to a system logical unit, you must supply a dataset label for the dtfname IJSYSIN, IJSYSLS, or IJSYSPH. If a SYSPCH or SYSLST dataset is created on tape, you must place it in closed status before you can use it as input. Use the CLOSE function of TSIDMNT to close the dataset after it is created.

To simplify the JCL necessary to process system files (SYSIPT, SYSLIST and SYSPCH), use the Generic Device Independence for Unit Record Devices (GDIUR) feature. The GDIUR feature eliminates the need for extra TSIDASS execution, VSE CLOSE statements and most of the TSIDMNT CLOSE function executions. GDIUR also protects your default system file assignments in case the job terminates abnormally. Refer to "Choosing the Device Type" in Chapter 7, "Running Jobs With EPIC" for additional information about using GDIUR.

Syntax

```
[//]  ASSGN SYSxxx
```

Example

```
// JOB MAKE ASSIGNMENT WITH TSIDASS
// TLBL TAPIN1,'DAILY.TRANS',,,,,64
// TLBL TAPOUT,'DAILY.BILLING',,,,,64
// EXEC TSIDASS
 ASSGN SYS021
 ASSGN SYS022
/*
// EXEC BILLING
/&
```

In this example, the program BILLING uses SYS021 for TAPIN1 and SYS022 for TAPOUT. TSIDASS makes the assignment for both of these files before the BILLING program is executed. The TLBL option code 64 inhibits LUB allocation.

OPEN Statements

Use TSIDASS OPEN when you process controlled datasets with a program that does not issue OPEN requests. TSIDASS OPEN issues a logical open for the dataset, so that CA EPIC for z/VSE can gain control of the dataset.

Syntax

OPEN *dtfname*,SYS*xxx*,*type*

Parameters

Parameter	Required?	Valid Entry	Default
dtfname	Yes	1 to 7 characters	None
xxx	Yes	000-254	None
type	Yes	INPUT or OUTPUT	None

dtfname identifies the dataset to be opened. Specify the dtfname or ddname as it appears on the dataset's label statement.

SYSxxx specifies the logical unit to assign.

INPUT specifies that the file is opened as input.

OUTPUT specifies that the file is opened as output.

Example

```
// JOB OPEN TAPE DATASET
// TLBL TAPEIN,'AP.DAILY.TRANS',,,,,64
// EXEC TSIDASS
 OPEN TAPEIN,SYS020,OUTPUT
/*
// EXEC PROGRAM
/&
```

In this example, TSIDASS issues an open for the AP.DAILY.TRANS dataset. LUB allocation is inhibited for that dataset.

RSTRT Statements

Use TSIDASS RSTRT when you are restarting a job that has been checkpointed by VSE. Like OPEN, it issues a logical open for named datasets, allowing CA EPIC for z/VSE to maintain control of its datasets. It also allows CA EPIC for z/VSE to do AVR processing and assign tape drives.

Note: For detailed restart procedures and for a JCL example, see Restarting Programs from a Checkpoint (see page 264).

LUB allocation must be inhibited for all restarted datasets. In addition, the same SYS number that was in use when the checkpoint was taken must be specified.

RSTRT Parameters

Parameter	Required?	Valid	Default
dtfname	Yes	1 to 7 characters	None
xxx	Yes	000-254	None
Specify one of the following two (required):			
INPUT		None	OFF
OUTPUT		None	OFF
EXT	No	1 to 255	1

dtfname identifies a dataset that will be used when the program is restarted. Specify the DTF name or DD name as it appears on the dataset's label statement.

SYSxxx specifies the logical unit to assign. It must be the same logical unit number that was in use when the checkpoint was taken.

INPUT specifies that the dataset was last opened as input.

OUTPUT specifies that the dataset was last opened as output.

EXT specifies the sequence number of the extent to be re-opened. This is only necessary for disk datasets, and only if the dataset is to be opened at an extent other than the primary extent. This number will be equal to the number of CA EPIC for z/VSE EP053 messages that were issued for the dataset prior to the checkpoint from which the program will be restarted.

Using the CA EPIC for z/VSE Sort Interface (TSIDSRT)

TSIDSRT solves the problems that can occur at both open and close with some SORT utility programs. TSIDSRT is not a sort program. Instead, it is an interface program between CA EPIC for z/VSE and your sort program. It calls your sort program to do the sort, but it allows CA EPIC for z/VSE to satisfy pre-open assignment checking, to truncate the SORTOUT file, and to close and delete SORTWK areas. TSIDSRT provides full support for CA EPIC for z/VSE's generic device independence and reblocking features.

If you use unlabeled tape datasets in your sort, these datasets must be controlled.

You can use only one SORTOUT dataset.

Note: For other specific information on executing a sort with controlled datasets and for JCL examples, see Sorting with CA EPIC (see page 248).

Configuration Options

GDI=YES must be specified as a configuration option.

JCL Requirements

```
// <labels for sort datasets>
// EXEC TSIDSRT
sort control statements
/*
```

UPSI Values

None.

Dataset Requirements

Supply labels for the datasets required by your sort program.

Control Statements

Supply the control statements required for your sort program.

Example

```
// JOB EXECUTE SORT
// TLBL SORTIN1,'DAILY.TRANS',,,,,,2
// TLBL SORTOUT,'SORTED.DAILY.TRANS',,,,,,1
// DLBL SORTWK1,'$$$.SORTWK1,(D)'
// EXEC TSIDSRT
 SORT FIELDS=(33,4,A),FORMAT=BI,WORK=1,FILES=1
 RECORD TYPE=F,LENGTH=80
 INPFIL BLKSIZE=1600
 OUTFIL BLKSIZE=4000
 END
/*
/&
```

Deleting Datasets (TSIDDEL)

Some programs that use PIOCS do not close their work datasets. TSIDDEL can be used to delete disk work datasets that are not closed by the program that opens them. At any point in the job or job step, TSIDDEL can be used to free up their disk space for other processing. These datasets must be controlled, and they can be either Start Track 1 or cataloged datasets.

TSIDDEL is available as a batch program and as a callable subroutine. When using a DLBL, specify the DLBL 'D' option. When using an CA EPIC for z/VSE DD statement, specify DISP=(NEW,DELETE).

If you are already using TSIDSRT as the interface to your sort program, TSIDDEL execution is not necessary.

TSIDDEL does not delete generation disk datasets.

JCL Requirements

```
// EXEC TSIDDEL
[ control statements ]
/*
```

UPSI Values

None.

Dataset Requirements

The datasets must have been created in the step immediately prior to the TSIDDEL step, with no labels between the EXEC statement for the program and the EXEC statement for TSIDDEL.

Control Statements

Control statements specify which datasets to delete. Each control statement specifies one dataset to delete, but you can submit as many control statements as you wish for each execution of TSIDDEL. Control statements must begin in column 1.

Syntax

dtfname

TSIDDEL Parameter

Parameter	Required?	Valid	Default
dtfname	Yes	1 to 7 characters	None

dtfname specifies the DTF name or DD name of the dataset to be deleted.

Example

The following job creates the SORTWK1 dataset in the first step. The 'D' option is included on the DLBL to qualify the dataset for deletion. In the second step, TSIDDEL deletes the SORTWK1 file and frees the space it occupied.

```
.
.
.
// DLBL SORTWK1,'$$$.SORTWK1,(D)'
// EXTENT SYS003
// EXEC SORT
[sort control statements]
/*
// EXEC TSIDDEL
SORTWK1
/*
.
.
.
```

Using TSIDDEL as a Subroutine

When you use TSIDDEL as a subroutine, you must use a parameter list to specify which file to delete. In the parameter list, place the 1 to 7-character DTF name or DD name as it appears on the dataset's label statement. You can delete only one file per call.

Control is passed to the subroutine using standard IBM linkage conventions. TSIDDEL does not provide any return codes or issue any messages.

Assembler Example

```
...
       CALL TSIDDEL,(FILEID)
...
FILEID DC   CL7'SORTWK1'
...
```

COBOL Example

```
...
WORKING-STORAGE SECTION.
10 FILEID PIC X(7) VALUE 'SORTWK1'.
...
PROCEDURE DIVISION.
...
CALL 'TSIDDEL' USING FILEID.
...
```

Copying and Merging Datasets (TSIDDTD)

TSIDDTD is a dataset-to-dataset copy/merge utility program. It can be used to copy a single dataset or to merge multiple input datasets into a single output dataset. TSIDDTD uses standard LIOCS access methods, so the input and output datasets can be completely controlled by CA EPIC for z/VSE.

TSIDDTD supports any sequential dataset with a fixed, variable, spanned or undefined record format. However, TSIDDTD can only process tape datasets that follow the standard LIOCS format conventions (see "Tape Formats Supported" below).

TSIDDTD performs one copy operation per execution. To copy a multi-file tape, execute TSIDDTD once for every dataset on the tape, and include input and output labels for each dataset.

Configuration Options

To use TSIDDTD for disk datasets, GDI=YES must be specified.

JCL Requirements

```
input dataset label(s)
output dataset label
// EXEC TSIDDTD
[optional control statements ]
/*
```

UPSI Values

None.

Dataset Requirements

You must submit a label for at least one input dataset and one output dataset. The default DTF names are INPUT and OUTPUT, but you can specify other DTF names in your control statements.

The output dataset's block size will be the same as the input dataset's block size unless you specify a new block size in the output dataset's definition. Uncontrolled disk datasets can be reblocked using the 'B=n' option on the DLBL statement.

When you are copying uncontrolled datasets, use SYS005 for the output dataset and SYS004 for the input dataset.

If more than one input dataset is being copied to the output dataset, specify the 'T' option (dataset drop) on every input dataset label except the last. With the 'Drop' option, you can merge tape and disk versions into a single dataset. If all the input is on the same device type, you can also use the concatenation feature of the CA EPIC for z/VSE DD statement.

Tape Formats Supported

Because TSIDDTD uses standard LIOCS access methods, it can only process tapes which follow standard LIOCS format conventions.

For a labeled tape, the format is:

```
VOL/HDR labels, tapemark, data, tapemark, EOV/EOF labels, tapemark
```

For an unlabeled tape, the format is:

```
[tapemark], data, tapemark
```

Some utility programs using PIOCS create tapes with additional tape marks within the data area. Utilities known to do this include IDCAMS BACKUP (but not REPRO) and LIBR BACKUP. However, when LIOCS encounters one of these additional tape marks, it issues the message "4103D EOF OR EOV INQUIRY?" because no EOV/EOF label was found. Although this message is normal when processing unlabeled tapes, it should not occur when processing labeled tapes.

If this message does occur when copying a labeled tape, or before the actual end-of-data when copying an unlabeled tape, it indicates that the tape format does not follow standard LIOCS conventions and cannot be copied successfully with TSIDDTD. To copy non-standard format tapes, you can use the DITTO TT function. Refer to the description of the use of DITTO in Chapter 7 to keep the copied tape under CA EPIC for z/VSE control.

Control Statements

Control statements are used to define dataset characteristics and specify TSIDDTD processing options.

For controlled datasets, control statements are not normally necessary. TSIDDTD will get the record type, record size and block size from the version record in the DSN Catalog.

For uncontrolled and Start Track 1 datasets, control statements specifying the record type, record size and block size are required.

The TSIDDTD maximum block size for tape is 64K. For uncontrolled labeled input tapes, TSIDDTD will use HDR2 information. If record type, record size and block size are not specified and cannot be obtained from the DSN Catalog, TSIDDTD defaults to an undefined record format with a blocksize of 65534. Record type, record size, and block size may not be in the catalog for some controlled datasets that were created by programs that use PIOCS.

Control statement parameters can be used to specify alternate DTFnames, control unlabeled tape processing, limit total output, select specific records only and end processing before end-of-file (EOF) is reached on input.

All control statements will be read and processed before TSIDDTD begins the copy operation. Parameters can be placed in columns 1 through 71. Separate parameters with commas. Multiple statements can be submitted; but parameters must be completed on the statement on which they begin.

Syntax

```
[ BLKSIZE=nnnnn ]
[ FEOF=start,length,operand,type,value ]
[ HDRINFO ]
[ INNAME=dtfname ]
[ INPUT=n ]
[ LRECL=nnnnn ]
[ NOREWIN ]
[ NOREWOUT ]
[ NOTPMK ]
[ OBLKSIZE=nnnnn ]
[ OUTNAME=dtfname ]
[ RECFM=x ]
[ RUNIN ]
[ RUNOUT ]
[ RECS=nnnnnn ]
[ SEL=start,length,operand,type,value ]
[ SKIP=nnnnnn ]
[ UNLBLIN ]
[ UNLBLOUT ]
```

Control Statement Parameters

Parameter	Required?	Valid	Default
BLKSIZE	No	1 to 65534 (or track capacity for disk)	input blksize
FEOF	No	See subparameters	None
FEOF Subparameters			
start		1 to 999999	None
length		1 to 999999	None
operand		EQ|NE|GT|LT|GE|LE	None
type		C or X	C
value		See text	None
HDRINFO	No	None	OFF
INNAME	No	1 to 7 characters	INPUT
INPUT	No	1 to 65534	None
LRECL	No	1 to 99999	None
OBLKSIZE	No	1 to 65534	BLKSIZE
OUTNAME	No	1 to 7 characters	OUTPUT
NOREWIN/NOREWOUT	No	None	OFF
NOTPMK	No	None	OFF
RECFM	No	F|V|S|U	U
RECS	No	1 to 999999	All
RUNIN/RUNOUT	No	None	OFF
SEL	No	See FEOF	All
SKIP	No	1 to 999999	0
UNLBLIN/UNLBLOUT	No	None	OFF

BLKSIZE defines the block size for input datasets. If BLKSIZE is not specified, it will be set to one of the following:

- the value of LRECL
- undefined (if LRECL is not specified). In this case, the input dataset is treated as unblocked.

For controlled datasets, the version block size for input datasets and the EDD reblock size for output datasets are used instead if available.

FEOF specifies a condition that causes TSIDDTD to act as if EOF had been reached on input. FEOF must be the last (or only) parameter on a control statement. If more than one FEOF parameter is specified on a single control statement, all of that statement's selection criteria must be met (AND) for a match to occur. If multiple control statements contain FEOF parameters, and any single statement's selection criteria are met (OR), a match occurs. You can use a total of 20 FEOF parameters per execution.

When FEOF is specified, all FEOF subparameters are required and positional:

start specifies the starting position in the record (relative to 1) of the field to be tested.

length specifies the length of the field to be tested.

operand specifies the type of test.

Use...	To specify that the input data is...
EQ	equal to value
NE	not equal to value
GT	greater than value
LT	less than value
GE	greater than or equal to value
LE	less than or equal to value

type specifies the type of data to be used in the test. C indicates that the value is in character format. X indicates that the value is in hexadecimal format.

value specifies the constant to be used in the test. Specify the same number of characters as the length value for character comparisons, or two times that number for hexadecimal comparisons.

HDRINFO causes VSE to display the tape header information on the operator console when opening the input datasets.

INNAME specifies an alternate input DTF name.

INPUT specifies the number of input datasets to be merged into a single output.

LRECL defines the dataset record size. For catalog-controlled datasets, the LRECL in the DSN Catalog is used instead if available. If LRECL is not specified, it will be set to one of the following:

- the value of BLKSIZE
- undefined (if BLKSIZE is not specified). In this case, the input dataset is treated as unblocked.

NOREWIN prevents rewinding of input tapes at close.

NOREWOUT prevents rewinding of the output tape at close.

NOTPMK suppresses the leading tape marks normally written before the data on unlabeled output tapes.

OBLKSIZE defines the block size for the output dataset. If OBLKSIZE is not specified, the BLKSIZE value will be used.

OUTNAME specifies an alternate output DTF name.

RECFM specifies the record format of both input and output datasets.

Code	Record Format
F	fixed length
V	variable length
S	spanned
U	undefined

RUNIN unloads input tapes at close.

RUNOUT unloads the output tape at close.

RECS specifies the number of records to be copied.

SEL specifies selection criteria for copying. If the data matches the expression, it is copied. If it does not match, it is skipped. SEL must be the last (or only) parameter on a control statement. If more than one SEL parameter is specified on a single control statement, all of that statement's selection criteria must be met (AND) for a match to occur. If multiple control statements contain SEL parameters, and any single statement's selection criteria are met (OR), a match occurs. You can use a total of 20 SEL parameters per execution.

When SEL is specified, all SEL subparameters are required. SEL subparameters are identical to FEOF subparameters (see above).

SKIP specifies the number of records to skip before beginning the copy operation.

UNLBLIN defines the input as an unlabeled uncontrolled tape dataset (not required if the dataset is controlled).

UNLBLOUT defines the output as an unlabeled uncontrolled tape dataset (not required if the dataset is controlled).

Example

This job copies a controlled disk dataset to a controlled tape dataset. No TSIDDTD control statements are required.

```
// JOB TSIDDTD COPY DISK TO TAPE
// DLBL INPUT,'input.data.set'
// TLBL OUTPUT,'output.data.set'
// EXEC TSIDDTD
/*
/&
```

Example

This job merges two input disk datasets and two input tape datasets to a single output tape dataset. All of the datasets are controlled.

```
// JOB TSIDDTD MERGE 2 DISK AND 2 TAPE DATASETS TO A SINGLE OUTPUT
// DLBL INPUT,'input.data.set.1,(T)'
// DLBL INPUT,'input.data.set.2,(T)'
// TLBL INPUT,'input.data.set.3',,,,,,18
// TLBL INPUT,'input.data.set.4',,,,,,2
// TLBL OUTPUT,'output.data.set'
// EXEC TSIDDTD
INPUT=4
/*
/&
```

Note the use of the 'Drop' option (T) on the DLBLs and Drop (16) and Rewind (2) options on the TLBLs.

Printing Datasets (TSIDDMP)

TSIDDMP is a dataset dump and print utility program. It can be used to completely or selectively dump or print a dataset in hex or character format, and will also print report files that have been written to disk or tape. TSIDDMP uses standard LIOCS access methods, so the input dataset can easily controlled by CA EPIC for z/VSE.

TSIDDMP supports any sequential dataset with a fixed, variable, spanned or undefined record format. However, TSIDDMP can only process tape datasets that follow the standard LIOCS format conventions (see "Tape Formats Supported" for TSIDDTD above).

TSIDDMP performs one dump or print operation per execution.

Configuration Options

To use TSIDDMP for disk datasets, GDI=YES must be specified as a configuration option.

JCL Requirements

```
input dataset label
// EXEC TSIDDMP
[ control statements ]
/*
```

UPSI Values

None.

Dataset Requirements

You must submit a label for the input dataset. The default DTF name is INPUT, but you can specify an alternate DTF name in your control statements.

For uncontrolled tape datasets, use SYS004 for the input dataset.

Control Statements

Control statements are used to define dataset characteristics and specify TSIDDMP processing options.

Control statements are not normally necessary for controlled datasets, because TSIDDMP will get the record type, record size and block size from the DSN Catalog.

Control statements specifying the record type, record size and block size are required for uncontrolled and Start Track 1 datasets.

If record type, record size and block size are not specified and cannot be obtained from the DSN Catalog, TSIDDMP will use an undefined record format with a blocksize of 65534.

Note: Record type, record size, and block size may not be available in the catalog for some catalog-controlled datasets if they were created by programs that use PIOCS.

Control statement parameters can be used to limit total output, select specific records only and end processing before end-of-file (EOF) is reached on input. Various formatting options are also provided.

All control statements will be read and processed before TSIDDMP begins the dump operation. Parameters can be placed in columns 1 through 71. Separate parameters with commas. Multiple statements can be submitted; but parameters must be completed on the statement on which they begin.

Syntax

```
[ BLKSIZE=nnnnn ]
[ CHARONLY ]
[ CTLCHAR ]
[ FEOF=start,length,operand,type,value ]
[ HDRINFO ]
[ INNAME=dtfname ]
[ LRECL=nnnnn ]
[ NOREW ]
[ NOSCALE ]
[ RECFM=x ]
[ RECS=nnnnnn ]
[ RUN ]
[ SEL=start,length,operand,type,value ]
[ SKIP=nnnnnn]
[ UNLBL ]
```

Parameters

Parameter	Required?	Valid	Default
BLKSIZE	No	1 to 99999	See text
CHARONLY	No	None	OFF
CTLCHAR	No	None	OFF
FEOF	No	See subparameters	None
FEOF Subparameters			
start		1 to 999999	None
length		1 to 999999	None
operand		EQ|NE|GT|LT|GE|LE	None
type		C or X	C
value		See text	None
HDRINFO	No	None	OFF
INNAME	No	1 to 7 characters	INPUT
LRECL	No	1 to 65534	None
NOREW	No	None	OFF
NOSCALE	No	None	OFF
RECFM	No	F|V|S|U	U
RECS	No	1 to 999999	All
RUN	No	None	OFF
SEL	No	See FEOF	All
SKIP	No	1 to 999999	0
UNLBL	No	None	OFF

Descriptions

BLKSIZE defines the block size for the input dataset. If BLKSIZE is not specified, it will be set to one of the following:

- the value of LRECL

- undefined (if LRECL is not specified). In this case, the input dataset is treated as unblocked.

For controlled datasets, the block size in the catalog will be used instead if available.

CHARONLY specifies that only the character representation is printed. If CHARONLY is omitted, both character and hexadecimal representations are printed.

CTLCHAR specifies that the dataset has an ASA print control character in the first byte of each record to control printer spacing. Use CTLCHAR for print files that have been written to disk or tape. CLTCHAR forces CHARONLY and NOSCALE printing.

Note: If CTLCHAR is used for a dataset that does not contain a valid ASA print character in the first byte of every record, TSIDDMP can cancel with an I/O error. To verify the print characters, print the file in hex mode. The first character should be one of the following:

space(x'40'), 0-9, A-C, +, or -

FEOF specifies a condition that causes TSIDDMP to act as if EOF had been reached on input. FEOF must be the last (or only) parameter on a control statement. If more than one FEOF parameter is specified on a single control statement, all of that statement's selection criteria must be met (AND) for a match to occur. If multiple control statements contain FEOF parameters, and any single statement's selection criteria are met (OR), a match occurs. You can use a total of 20 FEOF parameters per execution.

When FEOF is specified, all FEOF subparameters are required and positional:

start specifies the starting position in the record (relative to 1) of the field to be tested.

length specifies the length of the field to be tested.

operand specifies the type of test.

Use...	To specify that the input data is...
EQ	equal to value
NE	not equal to value
GT	greater than value
LT	less than value
GE	greater than or equal to value
LE	less than or equal to value

type specifies the type of data to be used in the test. C indicates that the value is in character format. X indicates that the value is in hexadecimal format.

value specifies the constant to be used in the test. Specify the same number of characters as the length value for character comparisons, or two times that number for hexadecimal comparisons.

HDRINFO causes VSE to display the tape header information on the operator console when opening the input datasets.

INNAME specifies an alternate input DTF name.

LRECL defines the dataset record size. For catalog-controlled datasets, the LRECL in the DSN Catalog is used instead if available. If LRECL is not specified, it will be set to one of the following:

- the value of BLKSIZE
- undefined (if BLKSIZE is not specified). In this case, the input dataset is treated as unblocked.

NOREW Prevents rewind of the input tape at close.

NOSCALE Suppresses (printing of) the position scale, which normally appears under the print record.

RECFM specifies the record format of both input and output datasets.CodeRecord FormatFfixed lengthVvariable lengthSspannedUundefined

RECS specifies the number of records to be printed.

RUN Unloads input tape at close.

SEL specifies selection criteria for copying. If the data matches the expression, it is copied. If it does not match, it is skipped. SEL must be the last (or only) parameter on a control statement. If more than one SEL parameter is specified on a single control statement, all of that statement's selection criteria must be met (AND) for a match to occur. If multiple control statements contain SEL parameters, and any single statement's selection criteria are met (OR), a match occurs. You can use a total of 20 SEL parameters per execution.When SEL is specified, all SEL subparameters are required. SEL subparameters are identical to FEOF subparameters (see above).SKIP specifies the number of records to skip before beginning the copy operation.

UNLBL defines the input as an unlabeled uncontrolled tape dataset (not required if the dataset is controlled).

Examples

The following job will dump the entire contents of a catalog-controlled disk dataset in hex format. No control statements are required.

```
// JOB TSIDDMP  DUMP DISK DATASET IN HEX
// DLBL INPUT,'input.data.set'
// EXEC TSIDDMP
/*
/&
```

The following will print the first 100 lines of a report file that has been routed to a controlled tape dataset by using GDIUR.

```
// JOB TSIDDMP  PRINT BLOCKED REPORT TAPE
// TLBL INPUT,'input.data.set'
// EXEC TSIDDMP
CTLCHAR,RECS=100
/*
/&
```

The following job prints a start track one disk dataset using an alternate DTF name, in character-only format, with the position scale suppressed. Only records containing the characters 10794 in positions 1-5 will be selected. End of file (and end of job) will be forced as soon as high values are found in positions 7-9. Note that the record format, record size and block size information is required for an uncataloged disk dataset.

```
// JOB TSIDDMP  SELECTIVE DUMP OF S.T. 1 FILE
// DLBL TPDATA,'input.data.set'
// EXTENT SYS005,POOL01,1,0,1,300
// EXEC TSIDDMP
INNAME=TPDATA
RECFM=F,BLKSIZE=4000,LRECL=80
CHARONLY,NOSCALE
SEL=1,5,EQ,C,10794
FEOF=7,3,EQ,X,FFFFFF
/*
/&
```

Accessing the VSE Label Area (TSIDGLB and TSIDPLB)

TSIDGLB and TSIDPLB are subroutines that can be called by your program to read and write VSE labels. TSIDGLB is used to read system-standard, partition-standard, and user labels.

TSIDPLB is used to modify a label that was acquired by TSIDGLB. TSIDPLB writes the label to an area maintained by CA EPIC for z/VSE. When CA EPIC for z/VSE is active, it checks for the label in this area first before searching the VSE label area.

Using the Subroutines

You can call both subroutines at any time, before or after a dataset is opened or closed. Both are called using standard IBM linkage conventions.

You can read and modify the label of any dataset defined in your JCL. However, tape dataset names are restricted to 17 characters. If you want to create a new label, first read a TLBL or DLBL that has no parameters except the DTF name.

Sample assembler and COBOL programs (TSIGLBL.A and TSIGLBL.C) are provided as examples. These can be assembled or compiled and run to demonstrate the coding techniques required to process VSE labels. These programs can be found in the CA EPIC for z/VSE installation library.

Data Area

Use the assembler copybook (TSIDGLB.A), or the COBOL copybook (TSIDGLB.C), for both subroutines. The discussion below refers to the data names in the assembler copybook.

Specify FILENAME and LBLRETCD as parameter list entries on your CALL statement.

FILENAME contains the DTF name of the label to be processed. The same DTF name must be specified in JCL.

Both subroutines return a 1-byte binary value in LBLRETCD. A zero (0) indicates that the label was returned by TSIDGLB or written by TSIDPLB. Any non-zero value indicates the label was not found or was not written.

In LBLLEN, TSIDGLB returns a 2-byte, unaligned, binary field that contains the length of the label returned. A length of 80 bytes indicates the label is for tape. Any length greater than 80 indicates the label is for disk. TSIDPLB uses the label length from this field.

TSIDGLB returns the label beginning at LBLFILNM. TSIDPLB writes the label from the same area.

To Read a Label

1. Move the DTF name of the label to be processed to FILENAME.
2. Call TSIDGLB, specifying FILENAME and LBLRETCD.

TSIDGLB Example

```
...
MVC FILENAME,=CL7'INPFILE'
CALL TSIDGLB,(FILENAME,LBLRETCD)
...
COPY TSIDGLB
...
```

To Write a Label

1. Read the label using the TSIDGLB procedure.
2. Modify the label information (see warning below).
3. Move the DTF name of the label to be processed to FILENAME.
4. Move the length of the label to LBLLEN (80 for tape; 200 for disk).
5. Call TSIDPLB, specifying FILENAME and LBLRETCD.

Important! TSIDPLB does not validate the data on the label you write, so you must ensure that it is correct. Although the label can be written successfully, results are unpredictable if its data is incorrect.

TSIDPLB Example

```
...
MVC  FILENAME,=CL7'OUTFILE'
MVC  LBLLEN,=H'80'
CALL TSIDPLB,(FILENAME,LBLRETCD)
...
COPY TSIDGLB
...
```

Chapter 9: Job Management Control Language

CA EPIC for z/VSE's Job Management Control Language (JMCL) gives you greater control over the execution of your job. The JMCL facility is activated using the JCLLANG=YES configuration option as described in the *Installation and System Guide*.

This section contains the following topics:

JMCL Statements

JMCL parameters are specified on VSE JCL statements. You can specify JMCL parameters on:

- EXEC statements
- JOB statements
- OPTION statements
- Comment statements

Most JMCL parameters are valid on only one or two of the JCL statement types. Check the description of the particular parameter to see where it can be specified.

JMCL parameters are in keyword format. You can specify more than one JMCL parameter on a single statement. Parameters must be separated with commas. Because many JMCL parameters are interdependent, be sure to check the end of this section for an example of the parameters working together.

Using JOB, EXEC, and OPTION Statements

On JOB, EXEC, and OPTION statements, the JMCL parameters must:

- follow at least one VSE keyword (or a comma between two spaces)
- precede any comments

Syntax

```
// JOB jobname parameters comments
// EXEC program parameters comments
// OPTION option parameters comments
```

Example

In this example, the job name is PAYROLL. RESET is a JMCL parameter in its correct position. CHECKS is a comment.

```
// JOB PAYROLL RESET CHECKS
```

Using Comment Statements

VSE comment statements can invoke JMCL by using special JMCL parameters.

Syntax

```
* JMCL parameters
```

Example

```
* JMCL NAME=BGSTEP
```

JMCL Parameters

JMCL parameters are specified on VSE JCL statements.

JCL Syntax Checking

With JMCL, you can retain datasets that have been flagged for purge (HOLD). You can cancel a job if it exceeds a specified time limit (TIME=n) or specify special processing for restarting canceled jobs (TYPRUN=RESTART).

In addition, you can prevent some job cancellations by requesting DD statement syntax checking (TYPRUN=SCAN and TYPRUN=SCANS).

HOLD

Use HOLD to prevent datasets that have been flagged for purge at EOJ from being purged if the job cancels.

HOLD is valid on JMCL statements and is effective for the duration of the job.

Syntax

```
HOLD
```

TIME

Use TIME to specify a time limit for job execution. If your job exceeds the limit you specify, it is terminated without a dump and the "CC" condition code (see "Conditions and IF") is set to 1.

TIME is valid on JOB statements.

Important! *The TIME parameter will be overridden by any program which uses IBM's SETIME instruction. If this occurs, your job may not cancel as expected.*

Syntax

```
TIME=nnnn
```

nnnn specifies the time period in minutes. Valid time periods are from 1 minute to 1440 minutes. No error message is issued if you specify a number out of this range. If you specify a number greater than 1440, the time limit is set to 1440 (unlimited time execution).

Example

```
// JOB XYZ TIME=60
```

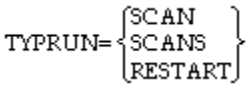
TYPRUN

TYPRUN has two separate uses. It can be used to specify:

- syntax checking for DD statements
- special processing for job restarts

TYPRUN is valid on JOB statements.

Syntax



TYPRUN Parameters

SCAN checks DD statement syntax and prints all JCL and any error messages. SCAN scans the JCL until the first /& statement, after which normal processing resumes. The job is not executed.

If there is SYSIPT data in your job stream you must also specify the DATA parameter.

Note: For more information, see DATA (see page 304) under JMCL Parameters.

SCANS checks DD statement syntax and prints only DD statements in error and the resulting error messages. The job is not executed. SCANS scans the JCL until the first /& statement, after which normal processing resumes.

If there is SYSIPT data in your job stream you must also specify the DATA parameter.

Note: For more information, see DATA (see page 304) under JMCL Parameters.

RESTART When a job cancels, you may want to restart it at a job step other than the first step of the job. You can branch to the desired step with a JMCL GOTO parameter. However, if you have specified the JMCL parameter HOLD or the configuration option PURBYP=YES, disk datasets flagged for purge are not purged when a job cancels. If you bypass steps with GOTO, work datasets created in those steps need to be specially purged. TYPRUN=RESTART purges those datasets when they are flagged for purge with the DLBL option P or X or DISP=(x,DELETE).

Example

```
// JOB XYZ TYPRUN=SCANS
```

Condition Checking and Branching

The GOTO facility allows you to bypass JCL statements without deleting them or changing them to comments before you submit the job. Condition checking, with the IF parameter, allows you to check whether certain conditions exist before you branch to a new part of the job.

GOTO

GOTO supports unconditional branching and conditional branching when used in conjunction with the JMCL IF parameter. Use GOTO for unconditional branching when you are restarting a job.

GOTO only branches forward in a job. GOTO cannot branch back to an earlier statement.

Mark the destination of the branch with the JMCL NAME parameter. If you bypass SYSIPT data during the branch, you must also use the DATA parameter. See "NAME" and "DATA" below.

If you are restarting a job, you may need to use the TYPRUN=RESTART parameter. Please note that if the step you execute in the restart checks for a condition code, that code may be invalid at restart time.

GOTO is valid on OPTION and JMCL statements.

Syntax

```
GOTO=label
```

label specifies the NAME label to branch to. The value must match the label specified on the NAME parameter (see below). If you specify GOTO=JOBEND, GOTO branches to end-of-job (/&) and a NAME parameter is not required.

Example

```
* JMCL GOTO=SORT3
```

NAME

Use NAME to mark the destination of a GOTO branch. Place the NAME statement immediately before the statement at which you want to resume processing.

NAME is valid on JMCL statements and EXEC statements.

Syntax

```
NAME=label
```

label specifies a label name. The value specified here must match the label specified on the GOTO parameter.

Example

```
* JMCL NAME=SORT3
```

DATA

Use DATA to bypass SYSIPT data during a GOTO branch. Because SYSIPT data is processed specially, CA EPIC for z/VSE cannot bypass it automatically.

Place a DATA parameter on the EXEC statement of each program that is followed by SYSIPT data. DATA is valid only on EXEC statements.

Syntax

```
DATA [ =n ]
```

Branching Subparameters

Parameter	Required?	Valid	Default
GOTO	No	1 to 8 characters	None
NAME	No	1 to 8 characters	None
DATA	No	1 to 127	1

n specifies the number of SYSIPT data streams to bypass for a single EXEC. The default is one, so you only need to specify n when there is more than one data stream for the EXEC.

Example

```
// EXEC PAYROLL DATA
<SYSIPT data here>
/*
```

Conditions and IF

Use IF to test for a condition. IF can test the current condition code, the highest condition code, the VSE cancel code, the date, time, CPU, partition, and a global constant you set. You can also set the current and highest condition codes.

Note: For more information, see Setting Values (see page 308).

You can test multiple conditions using the IF parameter. Both "AND" and "OR" conditions are supported.

When IF is used with GOTO, the branch occurs if the condition is true. When IF is used on an EXEC, the step is executed only if the condition is true. If it is false, CA EPIC for z/VSE bypasses the EXEC.

The IF parameter has three subparameters, which must be separated by at least one space. To test AND conditions, specify separate sets of subparameters on the same IF parameter and separate them with commas. To test OR conditions, use separate IF parameters.

When used with the GOTO parameter, IF is valid on OPTION and JMCL statements. It is also valid on EXEC statements, without GOTO. "AND" conditions are valid on OPTION, EXEC, and JMCL statements. "OR" conditions are valid on OPTION and JMCL statements only.

Syntax

```
IF=(type operand value [ ,type operand value, ... ] )
```

Parameters

For...	Valid values are...
CC, HC, and DOSCC	1 to 999
PART	2 character partition ID
CPU	0 to 9, A to Z
DATE and TIME	2 to 8 characters
Gnnn	1 to 3 characters

type specifies the type of input data to be tested.

Use...	To test the...
CC	current condition code
CPU	CPU ID
HC	highest condition code
DOSCC	VSE cancel code
PART	partition ID
DATE	current date
TIME	current time
Gnn	global constant

operand specifies the kind of test.

Use...	To test whether the input data is...
EQ	equal to *value*
NE	not equal to *value*
GT	greater than *value*
LT	less than *value*
GE	greater than or equal to *value*
LE	less than or equal to *value*

value specifies the constant to be used in the comparison.

Global constants are set with the SETG JMCL parameter. Identify the global constant you are testing by specifying 01 through 15 for nn.

DOSCC values are decimal codes generated by VSE. For normal EOJ the code is 16. You can check the DOSCC across job boundaries.

The CPU ID is the same number generated when you use CPU independence—that is, the last digit specified for the CPUIDS configuration option.

In partition comparisons, only the EQ and NE operands are valid. For example IF=(PART EQ BG) is valid. IF=(PART LE BG) is invalid.

Date is specified in yy-mm-dd format. Time is specified in hh-mm-ss format. If you wish, you can test only a portion of the date and time. To do so, supply only the portion you wish to test. Supply asterisks (*) as filler for beginning parts of the field. For example, to test for the first day of the month, code IF=(DATE EQ **-**-01). To test for January, code IF=(DATE EQ **-01).

Example

```
* JMCL IF=(HC EQ 99),GOTO=JOBEND
* JMCL IF=(PART NE BG,CPU EQ 5),GOTO=BGSTEP
// EXEC PHASE1 IF=(CC LT 16)
```

The first statement tests the highest condition code. If the highest condition code is 99, CA EPIC for z/VSE branches to the end of the job. The second statement uses an "AND" condition. If the job is not running in the BG partition but is running on CPU 5, a branch is made to BGSTEP. If neither of these tests is true, the job continues with the EXEC statement. However, PHASE1 only executes if the current condition code is less than 16.

Setting Values

You can establish a number of constants with JMCL parameters. These constants can be tested with the IF parameter.

Use SETC to set the current condition code. Use SETG to set up to 15 global constants.

SETC

SETC sets the current condition code to the number you specify. This also updates the highest condition code, if appropriate.

The current condition code remains in effect until a new value is set by another SETC parameter, until a RESET is encountered, or until a new value is set by the TSIDJC2 subroutine.

SETC is valid on OPTION and JMCL statements, where it is in effect as soon as the statement is processed. It is also valid on EXEC statements, where the condition code is set only if the program executes successfully.

Syntax

SETC=*nnn*

nnn specifies the number for the current condition code.

SETG

SETG sets global constants. You can have up to 15 global constants. Each constant remains in effect until you IPL VSE or until you set a new value with SETG.

SETG is valid on JMCL and OPTION statements only.

Syntax

SETGnn='constant'

nn identifies which global constant you are setting. Valid constants are 1-15 or 1-16 characters.

constant specifies the value of the constant. You can use spaces.

Parameters

Parameter	Required?	Valid	Default
nnn	No	1 to 255	None
nn	No	01 to 15	None
constant		1 to 16 characters	None

Symbolic Replacements

Symbolic replacements "replace" a set of characters that are found in your JCL.

Symbolic replacements are useful for JCL subparameters or comments.

CA EPIC for z/VSE inserts the replacement value into the statement and shifts to the right any data following the replacement symbol. For example:

```
* JMCL &1='INVENTRY.INPUT'
// TLBL SORTOUT,'&1',,,,,,2
```

would result in:

```
// TLBL SORTOUT,'INVENTRY.INPUT',,,,,,2
```

Important! *You must take care to avoid truncation of the statement. The statement must not exceed 71 columns with the replacement value inserted.*

Symbolic replacements are not valid on POWER JECL statements or in SYSIPT data. Any other statement (including JMCL statements) can use a symbolic replacements.

Assign values to symbolic replacements on OPTION statements.

&n

The symbolic replacement parameter has the format "&*n*", where *n* is a number between 0 and 9. Use it to replace characters with the value you assign to the symbol.

You can have up to 10 of these parameters in effect at the same time. Each one remains in effect until it is re-assigned in a JMCL or OPTION statement or it is nullified with RESET, or end-of-job occurs.

Syntax

```
&n='constant'
```

n identifies symbol being assigned.

constant specifies the value assigned to the symbol. The symbol must be enclosed in apostrophes and can contain spaces.

Example

In the following example, the DLBL is changed to

```
// DLBL INPUT,'PAYROLL.INPUT'
* JMCL &1='PAYROLL.INPUT'
// DLBL INPUT,'&1'
```

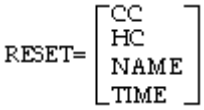
RESET

RESET nullifies most JMCL parameters. You can use it to set the current or highest condition code to zero, nullify GOTO branches, and nullify TIME control over a job.

We recommend that you use the global RESET parameter on the JOB statements and at the end of all jobs that do not have a logical dependency on previous jobs.

RESET is valid on JOB, EXEC, JMCL, and OPTION statements. When it is specified on a JOB statement, it is processed before any other JMCL parameters.

Syntax



Subparameters

CC resets the current condition code only.

HC resets the highest condition code only.

NAME nullifies any branch in effect. Processing resumes with the current statement.

TIME nullifies TIME control over the job.

When no subparameters are specified, all values are reset to the defaults.

JMCL Example

The following example shows JMCL parameters working together.

```
// JOB EXAMPLE TYPRUN=RESTART,RESET,TIME=240
* JMCL HOLD
// OPTION LOG &1=PAYROLL
* JMCL &2=PAYROLL
* JMCL GOTO=STEP2
// ASSGN SYS001,DISK,VOL=VOLUME,SHR
// DLBL SORTWK1,'==.SORTWK1(P)'
// EXTENT SYS001
// DLBL SORTIN,'&1.INPUT'
// DLBL SORTOUT,'&2.OUTPUT'
// EXEC SORT DATA
sort statements
/*
* JMCL NAME=STEP2
* JMCL IF=(DATE EQ **-**-01,TIME LT 19-00-00),GOTO=STEP3
// DLBL INPUT,'&1.INPUT'
statements
* JMCL NAME=STEP3
statements
/&
```

This example is a restart job.

The RESET parameter is included to reset any JMCL parameters established in a previous job. The TIME parameter limits the execution of this job to 4 hours.

Since it is a restart job, the GOTO bypasses the first step, which completed successfully in a previous execution. (GOTO is placed after the symbolic replacement definitions, so that they remain in effect.) The DATA parameter allows CA EPIC for z/VSE to bypass the sort control statements (SYSIPT data). The TYPRUN=RESTART parameter is specified so that the sort work area in the bypassed step is deleted. It was retained in the previous execution because of the JMCL HOLD parameter, specified in the second statement.

The * JMCL NAME=STEP2 statement defines the point at which processing should resume after the GOTO branch. The next statement tests the date and time. If it is the first of the month and earlier than 7:00 p.m., CA EPIC for z/VSE branches to the statement * JMCL NAME=STEP3. If either condition is false, processing continues with the // DLBL INPUT statement.

Both the &1 and &2 symbols are replaced with PAYROLL.

Using the JMCL Interface (TSIDJC2)

TSIDJC2

A batch program can interface with CA EPIC for z/VSE's Job Management Control Language by calling the TSIDJC2 subroutine. TSIDJC2 can be used to retrieve the following information:

- DOS job name
- program name
- current date (MM/DD/YY)
- current time (HH.MM.SS)
- cpu id character
- current JMCL condition code
- highest JMCL condition code
- task time limit
- JMCL symbolic values
- JMCL global constants

TSIDJC2 can update the following information:

- current JMCL condition code
- highest JMCL condition code
- JMCL symbolic values
- JMCL global constants

Parameter List

TSIDJC2 uses a parameter list to indicate the function being performed (Get or Put) in addition to providing the data areas for returning or updating data fields. Two modules are provided in the source library:

- TSIDJC2.C for COBOL programs
- TSIDJC2.A for Assembler programs

The TSIDJC2 subroutine can also be called from any language that uses standard calling conventions.

Retrieving JMCL Data (GET REQUEST)

GET REQUEST (G)

Data can be retrieved from the JMCL system by passing a character 'G' to TSIDJC2 in the request byte of the parameter list (field JCSREQ in TSIDJC2.And JCS-REQ in TSIDJC2.C Control is returned to the next instruction following the CALL.

COBOL Example

```
WORKING-STORAGE  SECTION.
COPY TSIDJC2.
.
.
.
PROCEDURE DIVISION.
MOVE   'G' TO JCS-REQ.
CALL   'TSIDJC2' USING JCS-START.
```

Assembler Example

```
MVI    JCSREQ,JCSREQG
CALL   TSIDJC2,(JCSSTART)
```

Updating JMCL Data (PUT REQUEST)

PUT REQUEST (P)

JMCL data can updated by passing a character 'P' to TSIDJC2 in the request byte of the parameter. This byte is labeled JCSREQ in TSIDJC2.A and JCS-REQ in TSIDJC2.C. A Put request updates the current JMCL condition code, the highest JMCL condition code, all JMCL symbolics and all global constants. All Put requests should be preceded by a Get request to ensure that all data fields that are not explicitly being modified are updated to their previous data values. Control is returned to the next instruction following the CALL.

To update current or highest JMCL condition code

Move the new condition code to the appropriate data field as a 2-byte binary value. During the update process, the highest condition code is set first. Then the current condition code is set. When the current condition code is set, it is checked against the highest condition code to determine whether the new current condition code is now also the new highest condition code. The JMCL current condition code is labeled JCSCC in TSIDJC2.A and JCS-CURR-COND-CODE in TSIDJC2.C. The JMCL highest condition code is labeled JCSHC in TSIDJC2.A and JCS-HIGH-COND-CODE in TSIDJC2.C.

To update a symbolic

Move the new 1 - 20 character symbolic into its appropriate data field. Symbolics &0 thru &9 are labeled JCSSYM0 thru JCSSYM9 in A.TSIDJC2 and JCS-SYMBOLIC-0 thru JCS-SYMBOLIC-9 in C.TSIDJC2. If the new symbolic is less than 20 characters, it must be left-justified and padded with spaces on the right. To remove a symbolic, clear the symbolic data field with hex zeros or LOW-VALUES.

To update a global constant

Move the new 1 - 16 character constant into its appropriate data field. Global constants G01 thru G15 correspond to the parameter data fields JCSGBL1 thru JCSGBL15 in A.TSIDJC2 and JCS-GLOBAL-CONS-1 thru JCS-GLOBAL-CONS-15 in C.TSIDJC2. If the new global constant is less than 16 characters, it must be left justified and padded with spaces on the right. To remove a global constant, clear the global constant data field with hex zeros or LOW-VALUES.

COBOL Example

```
WORKING-STORAGE  SECTION.
COPY TSIDJC2.
.
.
.
PROCEDURE DIVISION.
MOVE   'G' TO JCS-REQ.
CALL   'TSIDJC2' USING JCS-START.
MOVE   0 TO JCS-CURR-COND-CODE.
MOVE   'MONTHLY' TO JCS-SYMBOLIC-2.
MOVE   LOW-VALUES TO JCS-GLOBAL-CONSTANT-10.
MOVE   'P' TO JCS-REQ.
CALL   'TSIDJC2' USING JCS-START.
```

Assembler Example

```
MVI    JCSREQ,JCSREQG
CALL   TSIDJC2,(JCSSTART)
MVC    JCSCC,=XL2'00'
MVC    JCSSYM2,=CL20'MONTHLY'
XC     JCSGBL10,JCSGBL10
MVI    JCSREQ,JCSREQP
CALL   TSIDJC2,(JCSSTART)
```

TSIDJC2 Return Codes

TSIDJC2 returns a value in the return code byte of the parameter list. This byte is labeled JCSRC in TSIDJC2.A and JCS-RC in TSIDJC2.C. All return codes are one-byte character values.

Values

Code	Explanation
0	Successful completion of the G(et) or P(ut) request.
1	Either CA EPIC for z/VSE or the JMCL system are not active.
2	A secondary routine (TSIDJCS) could not be loaded from the CA EPIC for z/VSE system library. The normal cause of this problem is not having the CA EPIC for z/VSE system library in the search chain of the partition.
3	There was insufficient Partition GETVIS available to load the TSIDJCS routine. The simplest way to correct this problem is to add a SIZE= parameter to the EXEC statement.
4	The request code in JCSREQ or JCS-REQ was not a G or P.

Executing CP Commands

You can execute any VM/CP command in your VSE job stream by using JMCL's CP control statement.

The CP control statement must begin in column 1.

Configuration Options

For CP commands to be valid, the VM configuration option must be set to YES.

Syntax

```
* CP command
```

where *command* specifies which VM/CP command to execute.

Example

The following statement lists all tape drives on the operator console.

```
* CP Q T ALL
```

Chapter 10: Job Scheduling

CA EPIC for z/VSE's job scheduling system can be used to control execution across all partitions and VSE machines sharing the same DSN Catalog.

This section contains the following topics:

Event Posting

Introduction

When a job, step, or process has completed, you can direct CA EPIC for z/VSE to *post an event*. CA EPIC for z/VSE writes the name of the event in the DSN Catalog as a version of the dataset *EPIC.EVENT.DSN*. You assign an event a name, 1 to 8 characters long. You then use this name to identify which event has completed.

Like versions of a regular dataset, posted events become eligible for scratch according to the retention criteria cataloged for the dataset.

If *EPIC.EVENT.DSN* is not already in the catalog, CA EPIC for z/VSE adds it automatically the first time an event is posted. Then automatic day and cycle retention is used to retain posted events. You may want to change this retention with manual catalog maintenance.

Posted events can be deleted with the SCRATCH functions of TSIDUTL or TSIDMNT. They are normally deleted using the TSIDUTL utility.

Job scheduling subroutines put in the event name in the SYSPARM value, and COBOL programs do special processing when SYSPARM values include A or D. Avert potential conflict by not using these letters in event names processed with job scheduling subroutines.

Scheduling Functions

There are four job scheduling functions:

- Posting an event and releasing all matching jobs
- Posting an event only
- Checking for a posted event and releasing the requesting job
- Deleting a posted event

For each job scheduling function, CA EPIC for z/VSE provides one batch utility, one standard processing subroutine, and one CICS subroutine.

The batch utilities and subroutines share the same names. CICS subroutine names begin with TSID*C*.

Posting an Event and Releasing Matching Jobs (TSIDREL and TSIDCREL)

TSIDREL and TSIDCREL post events and automatically release jobs with matching job names from the POWER reader queue.

To be released, a job must be in disposition "H" or "L" in the POWER reader queue, and its job name must match the event name posted.

If the event name is 8 characters, CA EPIC for z/VSE releases the job with the POWER command:

R RDR,*event-name*

If the event name is less than 8 characters, it releases the job with:

R RDR,**event-name*

If jobs are released into the same partition and have the same priority, they execute in the order they are released.

Posting an Event Only (TSIDLOC)

TSIDLOC and TSIDCLOC post events. They do *not* release jobs. When you use TSIDLOC or TSIDLOC, dependent jobs must themselves issue a check for the posted event using TSIDCHK or TSIDCCHK.

Checking for a Posted Event (TSIDCHK and TSIDCCHK)

TSIDCHK and TSIDCCHK are used within a job to check to see if an event has been posted before continuing with a dependent task. If the event is posted your job continues uninterrupted.

If you check for the event with TSIDCHK (in either batch or subroutine) and the event is not posted, execution is interrupted. TSIDCHK issues a message to the operator and ask whether the job should be canceled or suspended. If the operator responds WAIT, the job is suspended, and the partition is stopped until the event is posted. When the event is posted, the partition is restarted automatically.

The CICS version, TSIDCCHK, does *not* interrupt processing if the event is not posted. You must check the return code passed, and process accordingly.

Deleting a Posted Event (TSIDUNL)

TSIDUNL and TSIDCUNL delete posted events. Use them to delete an event after all dependent tasks have completed. This is important if another processing cycle occurs before the event is scratched automatically.

Remember that if you do not delete them, events become eligible for scratch as described under "About Posted Events."

Executing in Batch Jobs

TSIDREL, TSIDLOC, TSIDCHK, and TSIDUNL can be executed in batch jobs.

JCL Requirements

```
// OPTION SYSPARM='event-name'
// EXEC TSIDxxx
// OPTION SYSPARM='        '
```

Control Statements

Two OPTION statements are required: one to specify the event name as a SYSPARM and one to clear the SYSPARM so it does not affect later processing.

event-name specifies the name of the event to be posted (the name of the job to be released). It may be 1 to 8 characters long.

TSID*xxx* specifies the utility you are executing. Specify TSIDREL, TSIDLOC, TSIDCHK, or TSIDUNL.

Example

This job executes TSIDREL. It posts an event with the name PAY and releases all jobs in the POWER reader queue which have job names beginning with PAY.

```
// JOB RELEASE JOBS
// OPTION SYSPARM='PAY'
// EXEC TSIDREL
// OPTION SYSPARM='        '
/&
```

Executing Standard Processing Subroutines

TSIDREL, TSIDLOC, TSIDCHK, and TSIDUNL are callable subroutines for non-CICS programs. Standard linkage conventions apply to all four subroutines.

Processing Requirements for TSIDREL, TSIDLOC, and TSIDUNL

1. Perform the "event" task.
2. Move the event name to the EVENT-NAME field. See "Posted Events," for details about event name limitations.
3. Issue a call to the appropriate subroutine.

Example

This task calls TSIDUNL. It deletes the PAY event after PAYROLL checks have been processed.

```
...

WORKING-STORAGE SECTION.
01 EVENT-NAME         PIC X(8) VALUE SPACES.
PROCEDURE DIVISION.
     PERFORM PAYROLL-CHECKS.
     MOVE 'PAY' TO EVENT-NAME.
     CALL 'TSIDUNL' USING EVENT-NAME.
...
```

Processing Requirements for TSIDCHK

1. Move the event name to the EVENT-NAME field. See "Posted Events," for details about event-name limitations.
2. Issue a call to TSIDCHK.
3. Perform the dependent task.

Example

The following task calls TSIDCHK, with PAY as the EVENT-NAME. If that event is posted, the task performs PAYROLL-CHECKS.

```
...

WORKING-STORAGE SECTION.
01 EVENT-NAME         PIC X(8) VALUE SPACES.
PROCEDURE DIVISION.
   MOVE 'PAY' TO EVENT-NAME.
   CALL 'TSIDCHK' USING EVENT-NAME.
   IF EVENT-RETURN-CODE = 'Y'
   PERFORM PAYROLL-CHECKS.
...
```

Executing in CICS

Using the Subroutines

TSIDCREL, TSIDCLOC, TSIDCCHK, and TSIDCUNL are CICS subroutines that can be called from a CICS command-level program.

The program must link to them using a 69–byte communications area with the following format:

Byte	Contents
1–8	EVENT–NAME
9	EVENT–RETURN–CODE
10–69	EVENT–POWER–MESSAGE

For TSIDCREL, when EVENT–RETURN–CODE is "P", bytes 10 through 69 contain the following:

Byte	Contents
10–12	CTL Spool Return Code
13–21	reserved
15–22	XPCC Function That Failed
23–36	reserved
37–38	VSE Return Code
39	reserved
40–41	Reason Code
42–54	reserved
55–56	POWER Return Code
57	reserved
58–59	POWER Feedback Code
60–69	reserved

Return Codes

TSIDCREL

Code	Explanation
A	The event has already been posted. No jobs are released.
F	The DSN Catalog is full and the event cannot be posted.
M	The event name specified in EVENT–NAME is missing or invalid.
P	The event was posted, but there was an error in releasing jobs from the POWER reader queue.
W	The event was posted, but POWER is in a busy state and jobs were not released.
X	The event was posted, but the XECB supervisor table is full.
Y	The event was posted and the jobs were released successfully.
5	The event was posted, but the POWER XPCC function failed.
6	The event was posted, but POWER is being terminated.
7	The event was posted, but POWER has terminated abnormally.
8	The event was posted, but TSIDCREL could not make a connection to POWER within 30 seconds.
9	The event was posted, but the POWER file could not be located.

Return codes 5–9 indicate highly unusual situations and should not occur.

Each of the numeric return codes (n) is associated with an CA EPIC for z/VSE message, numbered EPV3n. For example, to find the message associated with return code 5, see message EPV35 in *Messages and Codes*.

TSIDCLOC

Code	Explanation
A	The event has already been posted.
F	The DSN Catalog is full and the event cannot be posted.
M	The event name is missing or invalid.
Y	The event was posted successfully.

TSIDCCHK

Code	Explanation
M	The event name is missing or invalid.
N	The event is not posted.
Y	The event is posted.

TSIDCUNL

Code	Explanation
E	The *EPIC.EVENT.DSN* dataset definition has been deleted from the DSN Catalog.
F	The DSN Catalog is full.
M	The event name is missing or invalid.
N	The event was not posted.
Y	The event was deleted successfully.

Processing Requirements

1. Move the event name to the EVENT–NAME field. See "Posted Events," for details about event–name limitations.
2. Link to the subroutine.
3. Check the return code passed and process accordingly.

Example

```
WORKING-STORAGE SECTION.
01 EVENT.
   02 EVENT-NAME            PIC X(08).
   02 EVENT-RETURN-CODE     PIC X(01).
      88 SUCCESS      VALUE 'Y'.
      88 FAILURE        VALUE 'N'.
   02 EVENT-POWER-MESSAGE PIC X(60)
PROCEDURE DIVISION.
   MOVE 'PAY' TO EVENT-NAME.
   EXEC CICS
      LINK PROGRAM ('TSIDCCHK')
           COMMAREA (EVENT)
           LENGTH (69)
   END EXEC.
   IF SUCCESS
       GOTO PERFORM
   ELSE
       GOTO WAIT.
...
```

This example checks for a posted event with TSIDCCHK. If the event is posted, it performs the dependent task. If the event is not posted, it goes to the WAIT routine.

Creating Job Setup/Run Sheets (TSIDFAV)

UPSI Settings

```
// UPSI   1XXXX  Print comment cards.

// UPSI   X1XXX  Print user data cards.

// UPSI   XX1XX  Scan all VSE jobs for creation of input files.

// UPSI   XXX1X  Scan all POWER JOBS for creation of input files.

// UPSI   XXXX1  Input JCL submitted backwards (POWER MODE).
```

Modes of Execution

TSIDFAV can be executed in VSE mode or in POWER mode. The execution mode is determined by UPSI bit 4 (relative to 0).

To execute in VSE mode, set this UPSI bit to 0 (off).

To execute in POWER mode, set this UPSI bit to 1 (on).

SET Control Statement

The SET control statement can be used to define the partition ID and CPUID values used to replace equal signs (=) in the dataset name. The SET control statement must precede all job controls submitted. It must begin on column 1 whether execution is in VSE or POWER mode.

Format

```
SET PART=xx CPUID=y
```

Parameter Values Explained

xx defines the 2-character partition ID for replacing the first two equal signs. If this parameter is omitted, the first two equal signs are replaced by the partition ID where TSIDFAV is being executed.

y defines the 1-digit CPUID for replacing the third equal sign. If this parameter is omitted, the third equal sign is replaced by the last digit of the CPUID where TSIDFAV is being executed.

VSE Mode

In VSE mode, all job control is submitted to TSIDFAV in the order in which it is executed. The JCL is submitted either as instream data (following the EXEC statement), or via tape or disk input using the DTF name DISKIN. If POWER JECL is included in the jobstream, the input must be submitted via tape or disk. If the input is submitted after the EXEC statement, SYS004 must be assigned to SYSIPT. Under all types of input, the last record must be the EOF flag as determined by TYPE=EOF in the TSIDFAV macro, or defaulted to two dashes (--).

JCL Examples for VSE Mode

```
// JOB TSIDFAV
* INPUT INSTREAM DATA
// ASSGN SYS004,SYSIPT           REQUIRED
// UPSI XXXX0
// EXEC TSIDFAV
 VSE JCL
--
/*
/&

// JOB TSIDFAV
* DISK INPUT
// DLBL DISKIN,'FAV.INPUT.JCL'    REQUIRED
// ASSGN SYS004,DISK,...
// UPSI XXXX0
// EXEC TSIDFAV
/*
```

POWER Mode

In POWER mode, the job control must be submitted backwards (in reverse order of execution) and upside down so that card column 1 is submitted to TSIDFAV in card column 80. This prevents POWER from processing any POWER JECL submitted as instream data. The JCL can be submitted either as instream data (following the EXEC statement), or via tape or disk input using the DTF name DISKIN. If the input is submitted after the EXEC statement, SYS004 must be assigned to SYSIPT.

The EOF flag must be submitted in columns 1 and 2 (not reversed).

JCL Examples for POWER Mode

```
// JOB TSIDFAV
* INPUT INSTREAM DATA
// ASSGN SYS004,SYSIPT           REQUIRED
// UPSI XXXX1
// EXEC TSIDFAV
 VSE JCL AND POWER JECL READ WITH 9 EDGE IN
--                      (EOF FLAG)
/*
/&

// JOB TSIDFAV
* DISK INPUT
// DLBL DISKIN,'FAV.INPUT.JCL'    REQUIRED
// ASSGN SYS004,DISK,...
// UPSI XXXX1
// EXEC TSIDFAV
/*
```

TSIDFAV reads the VSE procedures and POWER SLIs from the appropriate VSE library. However, TSIDFAV does not support either PROC or SLI overwrites. Nested SLIs are supported, but nested PROCs are not supported.

Chapter 11: Job Activity Reports

This section contains the following topics:

Overview

CA EPIC for z/VSE provides several job accounting programs. These programs produce job accounting statistics using data collected from the Recorder File. All job accounting is affected by the JOBACCT and RECORD configuration options.

Recorder History Dataset

Recorder File backups can be used to create job activity reports. For example, you may want to combine daily backups into a single Recorder History dataset. You can create a new Recorder History dataset weekly, monthly, or for whatever time period you wish.

Step 1

Create a new Recorder History dataset, using a tape-to-tape copying utility such as TSIDDTD. The Recorder backup format is fixed-blocked, with a record length of 338 and a block size of 5746.

```
// JOB INITIALIZE RECORDER HISTORY FILE
// TLBL INPUT,'DAILY.RECORDER'
// TLBL OUTPUT,'RECORDER.HISTORY'
// EXEC TSIDDTD
/*
/&
```

Step 2

Combine the new Recorder File backups into an existing Recorder History dataset. The example below uses a sort program, but you can use whatever mechanism you prefer.

```
// JOB MERGE VSE RECORDER FILES
// TLBL SORTIN1,'VSE.DAILY.BACKUP',,,,,,2
// TLBL SORTIN2,'VSE.MNTHLY.BACKUP',,,,,,2
// TLBL SORTOUT,'VSE.MNTHLY.BACKUP',,,,,,2
// DLBL SORTWK1,'===.SORTWK1'
// EXTENT SYS004
// EXEC TSIDSRT
   SORT FIELDS=(25,2,A,21,6,A,17,4,A),FILES=2,FORMAT=BI
   RECORD TYPE=F,LENGTH=338
   INPFIL BLKSIZE=5746
   OUTFIL BLKSIZE=5746
   OPTION PRINT=ALL
   END
/*
/&
```

Creating the Activity Log Report (TSIDARP)

TSIDARP produces the Activity Log report, which shows open and close activity for controlled datasets. Open and close activity is sorted by:

- System ID
- Date and time
- Job name

The Activity Log report can be limited to activity generated on any CPU or occurring during a specific time period. TSIDARP also produces the cross-reference dataset that is used as input to TSIDBRP.

Control Statements

There are two control statements for TSIDARP: EXCLUDE and TSIDARP. Both are optional.

JCL Requirements

```
// ASSGN SYS003,SYSIPT
// ASSGN SYS004,SYSLST
// ASSGN SYS001,DISK,VOL=volser,SHR
// TLBL SYS005,'daily.recorder.backup'
// DLBL SYS002,'recorder.cross.ref'
// DLBL SORTWK1,'===.sortwk1'
// EXTENT SYS001
// EXEC TSIDARP
control statements
/*
```

UPSI Values

None.

Dataset Requirements

SYS005 identifies the dataset to be used as input. You can use the Recorder backup dataset created by TSIDRFS, or you can use a Recorder history dataset.

SYS002 identifies the cross-reference dataset produced as output. This dataset is used as input for TSIDBRP.

SORTWK1 identifies a sort work area for an internal sort. You must provide an ASSGN and a dummy EXTENT statement.

EXCLUDE Control Statement

The EXCLUDE control statement specifies a job to be excluded from the report. If used, it must come before the TSIDARP statement. You can use up to 20 EXCLUDE statements, each statement specifying one 1 to 8–character job name.

Syntax

EXCLUDE=*jobname*

TSIDARP Control Statement

The TSIDARP control statement specifies what dates, CPUs, and partitions to include in the report. It is optional, and if it is omitted, the report contains all relevant data.

The TSIDARP statement is a 25-byte record, consisting of five fields:

Bytes	Contents
1–7	'TSIDARP'
8–12	Starting date, in Julian format
13–17	Ending date, in Julian format
18–23	CPU ID
24–25	Partition ID

The statement must begin with 'TSIDARP'. All other fields are optional. To omit a field, leave the field blank. If a field is omitted, the report will contain all relevant data for that field.

Syntax

TSIDARP*nnnnnnnnnnnnnnnxx*

Example

The following job produces an Activity Log report for jobs run from December 1, 2009 to December 31, 2009. It includes only those jobs that were run on CPU 55555 in the BG partition. It excludes all datasets from the CICSPR16 job.

```
// JOB TSIDARP ACTIVITY LOG REPORT
// ASSGN SYS003,SYSIPT
// ASSGN SYS004,SYSLST
// ASSGN SYS001,DISK,VOL=WORK01,SHR
// TLBL SYS005,'DAILY.RCDR.BACKUP'
// DLBL SYS002,'RECORDER.CROSS.REF'
// DLBL SORTWK1,'==.SORTWK1'
// EXTENT SYS001
// EXEC TSIDARP
EXCLUDE=CICSPR16
TSIDARP943359436555555BG
/*
/&
```

Creating the DSN/Job Cross-Reference Report (TSIDBRP)

TSIDBRP generates two reports: the DSN/JOB Cross-Reference Report and the Accepted Tapes Report.

The DSN/JOB Cross-Reference Report lists tape activity and is a cross-reference to the Activity Log report created by TSIDARP.

The Accepted Tapes Report lists the controlled tapes that were accepted as input with the ACCEPT response to CA EPIC for z/VSE message EP009.

TSIDBRP uses one of three datasets as its input, depending on the UPSI value specified. For UPSI 000, 010, and 100 it uses the output dataset created by TSIDARP, called the Recorder Cross-Reference dataset. For UPSI 101, it uses either the Recorder Backup dataset created by TSIDRFS or the Recorder History dataset.

TSIDBRP uses the entire input dataset to create the contents of its reports. If you use the dataset created by TSIDARP, the time period for the reports is the same as that used for TSIDARP. If you use the Recorder Backup or History file, the time period for the reports is the time period covered by the Backup or History file.

If you use the Recorder Cross-Reference dataset as your input dataset, you may wish to run TSIDARP and TSIDBRP in the same job stream.

JCL Requirements

```
// ASSGN SYS004,SYSLST
// ASSGN SYS001,DISK,VOL=volser,SHR
// DLBL SORTWK1,'===.sortwk1'
// EXTENT SYS001
// DLBL SYS002,'recorder.cross.ref'
// DLBL SYS005,'recorder.xref.work.file'
// UPSI 000
// EXEC TSIDBRP
/*
```

or

```
// ASSGN SYS004,SYSLST
// ASSGN SYS001,DISK,VOL=volser,SHR
// DLBL SORTWK1,'===.sortwk1'
// EXTENT SYS001
// TLBL SYS005,'daily.recorder.backup'
// UPSI 101
// EXEC TSIDBRP
/*
```

UPSI Values

UPSI 000 creates both reports. It uses the Recorder Cross-Reference dataset (created by TSIDARP) as input.

UPSI 010 creates only the DSN/JOB Cross-Reference report. It uses the Recorder Cross-reference dataset as input.

UPSI 100 creates only the Accepted Tapes report. It uses the Recorder Cross-Reference dataset as input.

UPSI 101 creates only the Accepted Tapes report. It uses the Recorder Backup or History dataset as input.

Dataset Requirements

SYS002 identifies the Cross-Reference dataset created by TSIDARP. It is required with UPSI 000, 100, and 010.

SYS005 identifies the Recorder Backup or History dataset when used with UPSI 101. It also identifies the Recorder Cross-Reference work dataset when used with UPSI 000.

SORTWK1 identifies a sort work area for an internal sort. You must provide an ASSGN and a dummy EXTENT statement.

Control Statements

None.

Examples

The following job creates the DSN/JOB Cross-Reference Report. It uses the Recorder Cross-Reference dataset created by TSIDARP as input.

```
// JOB TSIDBRP CROSS-REFERENCE REPORT
// ASSGN SYS004,SYSLST
// ASSGN SYS001,DISK,VOL=volser,SHR
// DLBL SORTWK1,'==.SORTWK1(N,D,P)'
// EXTENT SYS001
// DLBL SYS002,'RECORDER.CROSS.REF'
// UPSI 010
// EXEC TSIDBRP
/*
/&
```

The following job creates the Accepted Tapes Report. It uses the Recorder File Backup as input.

```
// JOB  TSIDBRP CROSS-REFERENCE REPORT
// ASSGN SYS004,SYSLST
// ASSGN SYS001,DISK,VOL=volser,SHR
// DLBL SORTWK1,'==.SORTWK1(N,D,P)'
// EXTENT SYS001
// TLBL SYS005,'DAILY.RCDR.BACKUP'
// UPSI 101
// EXEC TSIDBRP
/*
/&
```

Job Accounting Reports by Job (TSIDJAC)

TSIDJAC produces the Job Time Activity report, which reports machine use statistics by job. Statistics are detailed by job step with totals for each job. In addition, a grand total is generated for all jobs.

Use this report to find out when a particular job was run (did they really run payroll twice last night?) and what resources it used (my inventory program used 100% of my CPU?). Statistics include start and end times, duration, partition ID, CPU time, overhead, and I/O activity.

JCL Requirements

```
// ASSGN SYS004,SYSLST
// ASSGN SYS001,DISK,VOL=volser,SHR
// DLBL SORTWK1,'==.sortwk1'
// EXTENT SYS001
// TLBL SYS005,'daily.recorder.backup'
// EXEC TSIDJAC
/*
```

UPSI Values

None.

Dataset Requirements

SYS005 identifies the dataset to be used as input. You can use either the Recorder Backup or History dataset.

SORTWK1 identifies a sort work area for an internal sort. You must provide an ASSGN and a dummy EXTENT statement.

Control Statements

None.

Example

```
// JOB  TSIDJAC JOB ACTIVITY
// ASSGN SYS004,SYSLST
// ASSGN SYS001,DISK,VOL=WORK01,SHR
// DLBL SORTWK1,'===.SORTWK1'
// EXTENT SYS001
// TLBL SYS005,'DAILY.RCDR.BACKUP'
// EXEC TSIDJAC
/*
/&
```

Job Accounting Reports by Partition (TSIDTAC)

TSIDTAC produces the Date Time Activity report, which reports machine use statistics by partition and generates a grand total for machine use in all partitions.

Use this report to find out what was running in a partition at a particular time. Statistics include start and end times, duration, partition ID, CPU time, overhead, and I/O activity.

JCL Requirements

```
// ASSGN SYS004,SYSLST
// ASSGN SYS001,DISK,VOL=volser,SHR
// DLBL SORTWK1,'===.sortwk1'
// EXTENT SYS001
// TLBL SYS005,'daily.recorder.backup'
// EXEC TSIDTAC
/*
```

UPSI Values

None.

Dataset Requirements

SYS005 identifies the dataset to be used as input. You can use either the Recorder Backup or the Recorder History dataset.

SORTWK1 identifies a sort work area for an internal sort. You must provide an ASSGN and a dummy EXTENT statement.

Control Statements

None.

Example

```
// JOB  TSIDTAC PARTITION ACTIVITY
// ASSGN SYS004,SYSLST
// ASSGN SYS001,DISK,VOL=WORK01,SHR
// DLBL SORTWK1,'===.SORTWK1'
// EXTENT SYS001
// TLBL SYS005,'DAILY.RCDR.BACKUP'
// EXEC TSIDTAC
/*
/&
```

Maintaining the Job Accounting Database (TSIDJLD)

TSIDJLD creates and maintains a job accounting database.

You can access this database through CICS with the TOJA transaction. You can also develop your own reports and CICS transactions to access this data. To help you develop your own access to this data, a COBOL copy book (TSIDOJA.C) is provided.

TSIDJLD uses a VSAM KSDS to store its data. See the *Installation and System Guide*, "Initializing the Job Accounting Database" to create the database.

JCL Requirements

```
// ASSGN SYS003,SYSIPT
// ASSGN SYS004,SYSLST
// ASSGN SYS001,DISK,VOL=volser,SHR
// DLBL SORTWK1,'===.sortwk1'
// EXTENT SYS001
// DLBL TSIJADB,'tsijadb.job.account',,VSAM,CAT=ucat
// TLBL SYS005,'daily.recorder.backup'
// EXEC TSIDJLD
control cards
/*
```

UPSI Values

None.

Dataset Requirements

SYS005 identifies the input dataset. You can use either the Recorder Backup or the Recorder History dataset.

TSIJADB identifies the database file that will be produced as output. This dataset is a VSAM KSDS, and must be defined prior to running TSIDJLD.

SORTWK1 identifies a sort work area for an internal sort. You must provide an ASSGN and a dummy EXTENT statement.

Control Statements

Control statements indicate what data is to be added to or deleted from the database. You can submit up to 95 control statements per execution. All fields are positional.

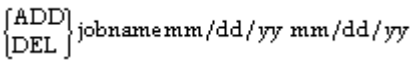
The control statement is optional, but if it is used, the first field is required. All other fields are optional. When a field is omitted, TSIDJLD uses all relevant data.

You can submit both ADD and DEL control statements in the same execution. All deletes are performed before new information is added.

The control statement is 28 bytes long and contains five fields:

Bytes	Contents
1–3	ADD or DEL
4–11	Job name
12–19	Jobs started on or after this date
20	Blank
21–28	Jobs started on or before this date

Syntax



Parameters

Parameter	Required?	Valid	Default
Specify one of the following two (required):			
ADD		None	OFF
DEL		None	OFF
jobname	No	1 to 8 characters	None
mm/dd/yy	No	8 characters	None

ADD adds data from the Recorder Backup or History dataset. If ADD is the only parameter on the statement, all data are added. If other parameters are specified, only data matching those parameters are added.

DEL deletes matching data from the data base. At least one other parameter must be specified. (To delete all records from the data base, use IDCAMS to delete and redefine the database.)

jobname specifies a particular job to be added to or deleted from the data base. You can specify a generic job name by ending the name with an asterisk (for example, specify TST* to process all jobs beginning with TST). All occurrences of that job or those jobs are added or deleted. If jobname is omitted, all jobs on the Recorder Backup or History dataset are added or all jobs matching the dates specified in positions 12 through 28 are deleted.

*mm**/**dd**/**yy* (Positions 12 through 19) Adds or deletes jobs started on or after this date. You must use 2 digits for month and day (for example, 01/05/09).

*mm**/**dd**/**yy* (Positions 21 through 28) Adds or deletes jobs started on or before this date. You must use 2 digits for month and day (for example, 01/05/09).

Examples

The following job adds all information from the Recorder Backup dataset to the data base.

```
// JOB  TSIDJLD UPDATE THE JOB ACCTNG DATA BASE
// ASSGN SYS003,SYSIPT
// ASSGN SYS004,SYSLST
// ASSGN SYS001,DISK,VOL=WORK01,SHR
// TLBL SYS005,'DAILY.RCDR.BACKUP'
// DLBL SORTWK1,'===.SORTWK1'
// EXTENT SYS001
// DLBL TSIJADB,'TSIJADB.JOB.ACCOUNT',,VSAM,CAT=UCAT
// EXEC TSIDJLD,SIZE=TSIDJLD
ADD
/*
/&
```

The following job deletes all jobs started on or before December 31, 2009 and adds all information from the Recorder Backup dataset to the database.

```
// JOB  TSIDJLD UPDATE THE JOB ACCTNG DATA BASE
// ASSGN SYS003,SYSIPT
// ASSGN SYS004,SYSLST
// ASSGN SYS001,DISK,VOL=WORK01,SHR
// TLBL SYS005,'DAILY.RCDR.BACKUP'
// DLBL SORTWK1,'===.SORTWK1'
// EXTENT SYS001
// DLBL TSIJADB,'TSIJADB.JOB.ACCOUNT',,VSAM,CAT=UCAT
// EXEC TSIDJLD,SIZE=TSIDJLD
ADD
DEL                 12/31/09
/*
/&
```

Online Access to the Job Accounting Database (TOJA)

TOJA, a CICS transaction, gives you online access to the job accounting database created by TSIDJLD.

TOJA has four panels. The Job Selection panel is an entry panel. It allows you to select a job or a set of jobs for display. The rest of the panels display job accounting information at increasing levels of detail.

Panels

```
            EPIC/VSE ONLINE JOB ACCOUNTING FACILITY

ENTER JOB NAME TO DISPLAY

    ENTER (EOB) FOR ALL JOBNAMES FROM START OF FILE
    ENTER PARTIAL JOBNAME FOR GENERIC DISPLAY
    ENTER JOBNAME FOR SINGLE JOB DISPLAY
```

The highest level, the Job Overview panel, displays an overview of one or more jobs.

```
                EPIC/VSE ONLINE JOB ACCOUNTING FACILITY
.ALL                                                                  PAGE 16
JOB NAME   PHASE NAME  START DATE  START TIME   DURATION  CC  PT  AR  CPU ID
$HEXDUMP    BIMUTIL     05/06/09    11:59:42    00:00:35  10  BG 000  000001
$HEXDUMP    DYNSEL      05/01/09    17:04:27    00:00:37  10  BG 002  000002
$HEXDUMP    DYNSEL      05/03/09    10:35:39    00:01:44  10  FA 002  000002
$HEXDUMP    DYNSEL      05/13/09    12:18:16    00:03:25  10  F5 002  000002
$HEXDUMP    DYNSEL      05/06/09    12:00:20    00:03:01  24  F6 001  000002
$MOVE       BIMUTIL     05/10/09    12:25:43    00:00:21  10  BG 000  000001
$MOVE       DYNSEL      05/10/09    11:26:07    00:00:54  10  BG 000  000002
$MOVE       DYNSEL      05/10/09    12:26:04    00:05:36  10  F7 004  000002
$PKG        BIMUTIL     05/01/09    03:31:28    00:00:18  10  FA 000  000001
$PKG        BIMUTIL     05/07/09    13:24:11    00:00:20  10  FA 000  000001
$PKG        BIMUTIL     05/11/09    01:02:12    00:00:24  10  FA 000  000001
$PKG        BIMUTIL     05/01/09    22:37:36    00:00:03  10  FA 000  000002
$SELECT     BIMUTIL     05/03/09    09:06:21    00:00:19  10  BG 000  000001
$SELECT     BIMUTIL     05/14/09    10:33:08    00:00:29  10  BG 000  000001
$SELECT     BIMUTIL     05/14/09    21:55:41    00:00:14  10  BG 000  000001
$SELECT     DYNSEL      05/07/09    00:49:10    00:11:18  23  F5 001  000001
$SELECT     DYNSEL      05/07/09    19:57:25    00:06:38  10  F5 004  000001
$SELECT     DYNSEL      05/14/09    10:18:38    00:00:42  10  F5 000  000001
$SELECT     LISTLOG     05/14/09    10:33:47    00:00:07  10  FA 000  000002
            DYNSEL      05/14/09    10:33:54    00:09:53  10     004
     PF1=FORWARD  PF2=BACKWARD  PF3=DETAIL  PF4=END
```

At the next level, the Program Overview panel displays an overview of the steps executed in a single job.

```
                EPIC/VSE ONLINE JOB ACCOUNTING FACILITY
$SEL                                                                  PAGE 01

 JOB NAME --------- $SELECT         PHASE NAME ---------- DYNSEL
 CPUID  ----------- 000001          PARTITION ----------- F5
 START DATE ------- 05/07/09        END DATE ------------ 05/07/09
 START TIME ------- 00:49:10        CPU TIME ------------ 00:00:02:0
 STOP TIME -------- 01:00:28        OVERHEAD ------------ 00:00:00:9
 DURATION  -------- 00:11:18        ALLBOUND ------------ 00:00:26
 CANCEL CODE (HEX) ------ 23        PHASE CODE -- L (S=STEP L=LAST)
                                    PHASE SEQ --- 0001
          ********** INPUT OUTPUT TABLES **********
   00C      00E      100      158      750      01F      756
 00000064 00000125 00000468 00000145 00000228 00000017 00000065
   390      391      392      393      395      492      890
 00000006 00000015 00000042 00000008 00000042 00000024 00000024
   891      893      896      897      645
 00000018 00000005 00000003 00000006 00000003
          ****************************************
  USER INFORMATION --------
  NUMBER OF AUDIT RECORDS FOR THIS PHASE -- 006
     USE PF3 TO VIEW AUDIT RECORDS

     PF1=FORWARD  PF2=BACKWARD  PF3=DETAIL  PF4=END
```

Finally, the Program Audit panel displays the datasets used by a single program.

```
                EPIC/VSE ONLINE JOB ACCOUNTING FACILITY
 FILEID           REC CREATION OPEN      EXPIRY    SERIAL  REEL    END   DTF
 PROCESSED       TYPE   TIME  CLOSE CUU   DATE    ACCEPTED TRACK  TRACK  TYPE
*******************************************************************************
 250.SII.PAFS      TI 20:02:07  O   809  12/31/99  C18609  000001         14
 250.SII.PAFS      TO 20:02:56  O   492  12/31/99  C06092  000001         12
 250.SII.PAFS      TI 20:03:53  C   809  12/31/99  C18609  000001         14
 250.SII.PAFS      TO 20:03:55  C   492  12/31/99  C06092  000001         12
 MT   0: 0: 0
 MT   0: 0: 0

                   PF2=BACKWARD              PF4=END
```

Accessing TOJA

1. From CICS, enter the TOJA transaction. The Job Selection panel is displayed.
2. If you want to view all jobs, press ENTER.
   a. If you want to view several jobs beginning with the same character string, enter the string and press ENTER.
   b. If you want to view a single job, enter the job name and press ENTER.

Navigating in TOJA

To move forward in a panel, press PF1.

To move backward in a panel, press PF2.

To move to the next level of detail, place the cursor on the item you wish to display and press PF3.

To move back a level of detail, press ENTER.

Exiting TOJA

From any panel, press the CLEAR key or PF4.

Chapter 12: Managing Tape Volumes

This section contains the following topics:

Overview

To protect controlled datasets effectively, CA EPIC for z/VSE controls the tape volumes they reside on. CA EPIC for z/VSE also protects the tape volumes it controls. Every tape that contains an CA EPIC for z/VSE-controlled dataset is recorded in the DSN Catalog. Tapes are identified by volume serial number and controlled at the version level under the name of the dataset they contain. Tape volumes are cataloged individually, as they are used for controlled datasets or initialized with TSIDINIT. CA EPIC for z/VSE provides utilities to assist you in labeling, initializing, pooling, and vaulting controlled tapes. This chapter describes CA EPIC for z/VSE tape control and tape protection utilities.

Tape Encryption

CA EPIC for z/VSE provides two facilities for encrypting user tape data. The first is a **software based encryption**, an intercept that serves as an encryption/decryption layer between user programs and the VSE data management routines. The second is an interface with the IBM 3592 tape drive that provides **hardware based encyption** of tape data. Each facility has unique advantages, and the encryption method chosen should depend upon individual needs.

Software Based Encryption

Software based encryption is invoked for a dataset by including the 'DEC=' parameter in the CA EPIC for z/VSE Dataset Definition. Encrypted tapes may only be decrypted under the control of the same DSN catalog that was used to write them. It is intended to prevent tape datasets from being read at external locations, but does not prevent unauthorized internal use. The advantage of this method is that it has no specific hardware requirements.

Note: For more information, see the TSIDMNT CATALOG 'DEC' tape parameter description in the Chapter DSN Catalog Management Utility.

Important! The software method requires the use of LIOCS by the program accesssing the tape information. CA EPIC for z/VSE's software facility cannot provide encryption or decryption for utilities that use PIOCS such as LIBR 'BACKUP' or SORT packages. Careful planning and testing should be conducted for each application that uses this feature.

Hardware Based Encryption

CA EPIC for z/VSE also provides an interface with the hardware encryption feature on IBM 3592 tape drives. Unlike software encryption, this method will work with any program, including PIOCS. It uses fewer CPU cycles than the software method because the encryption work is performed by the tape control unit. As a result, the hardware method is preferrable to software if the required hardware is available.

VSE's support of 3592 encryption hardware requires an Encryption Key Manager (EKM) that runs on a JAVA platform and communicates related information to the control unit. With native VSE, applications specify encryption preferences to the EKM through 'KEKL' JCL statements. 'KEKL' syntax requires that a tape drive be pre-assigned, which negates the operational advantages of EPIC's AVR feature. Therefore, CA EPIC for z/VSE provides a facility for the specification of 'KEKL' information without the need for 'KEKL' JCL statements.

CA EPIC for z/VSE allows the storage of enryption key labels in it's catalog with the TSIDMNT 'KMODEL' command. Along with the KEKL information, the 'KMODEL' provides a name which is used to collectively refer to the encryption key data.

For hardware tape encryption to occur, two requirements must be met:

1) A MODE (density) that supports encryption must be selected for the tape through the catalog or JCL.

2) A KMODEL must be associated with the dataset so that EPIC will know which KEKL information to use. The KMODEL for a dataset is specifed with the 'KMODEL=' parameter of the TSIDMNT 'CATALOG' or 'UPDATE' commands. If a KMODEL is defined with the name of '$DEFAULT', that will be used when no KMODEL name is supplied for a dataset.

It should be emphasized that the specification of a KMODEL name for a dataset will not by itself trigger encryptrion.

When CA EPIC for z/VSE determines that an output tape should be encrypted, it selects a compatible tape drive that supports encryption, retrieves the KEKL information from the catalog, and passes it to VSE. In addition to the standard console traffic, message 'EP175' will be written to signal that an encrypted tape is being created. For input, the KEKL information is stored on the tape and the key exchange is handled by VSE without EPIC involvement. CA EPIC for z/VSE only needs to select an appropriate tape drive for the operation.

Tape Volume Serial Numbers

Every tape volume used to write an CA EPIC for z/VSE-controlled dataset must have a unique volume serial number. Even unlabeled tapes must be identified with their own volume serial numbers. This strict identification is essential to tape volume control.

Volume serial numbers can be any six-character combination of alphanumeric characters. Groups of volume serial numbers do not need to be consecutive; any unique volume serial number is valid. Tape pools are defined within volume serial number ranges, so it can be helpful to initialize pooled tapes with consecutive volume serial numbers.

Recognizing and Controlling Tape Volumes

When a tape volume is brought under CA EPIC for z/VSE control, CA EPIC for z/VSE gives it a special VOL1 label which identifies the volume as controlled and protects it against accidental overwrites.

CA EPIC for z/VSE looks for a controlled tape volume when it performs AVR processing. First, it looks for the special VOL1 label. If it does not find that label, it checks the DSN Catalog to see if the tape was created by CA EPIC for CMS; if it was, the volume is accepted. If the volume fails these tests, CA EPIC for z/VSE asks the operator to specify a tape drive and to verify the volume serial number of the tape on that drive.

Unlabeled tapes do not have this special VOL1 label, so CA EPIC for z/VSE requires the operator to specify the drive that contains the volume, and to verify the volume serial number every time unlabeled processing is requested.

If an uncontrolled tape volume has a label that has expired, CA EPIC for z/VSE can use the tape, but again the operator is asked to verify the volume serial number. If that volume serial number duplicates one already in the DSN Catalog, the operator is asked to enter a new volume serial number.

HDR1 Labels and Retention

CA EPIC for z/VSE writes an expiration date of 99365 (permanent retention) on every HDR1 label created under its control. This helps to protect the tape from being overwritten outside CA EPIC for z/VSE's control (for example, when CA EPIC for z/VSE has not been started or has been deactivated). CA EPIC for z/VSE uses the DSN Catalog, not the HDR1 label, to determine each tape volume's status as active or scratch.

Scratch Tapes

Scratch tapes are produced by the SCRATCH functions of TSIDUTL, TSIDMNT, and Online Manager. The TSIDUTL "DSPLY SCRATCH" function produces a report listing all available scratch tapes. The TSIDUTL "PSCRATCH" function creates a report projecting which tapes will be eligible to be scratched on a certain date.

See the *Installation and System Guide* for information about TSIDUTL's JCL requirements. Also see the TSIDMNT SCRATCH function.

Tape Pooling

You can use tape pooling to group similar types of tapes into tape pools. For example, your high quality tapes could constitute one pool, short tapes another, long tapes a third, and so forth. You can also use pools to separate cartridge tapes from reel tapes. Datasets are assigned to tape pools through the DSN Catalog. When a pooled dataset is created, CA EPIC for z/VSE automatically requests the operator to mount a scratch tape from the appropriate pool.

Tape pools are established using the TSIDPOL macro and CA EPIC for z/VSE configuration options. Each pool is given a one character pool code (A to Z or 0 to 9) and is assigned one or more volume serial number ranges.

If a tape's volume serial number falls within a pool's ranges, the tape belongs to that pool. A pool tape can only be used to write datasets assigned to that pool. If a volume serial number does not fall in any of the defined ranges, the tape is not a pool tape. It can only be used to write datasets that are not assigned to a tape pool.

Datasets can be assigned to tape pools through the Online Manager, the TSIDMNT program, or the use of the OPTION=TPL=x parameter on a DD statement.

Tape Ownership

The tape ownership feature provides another form of tape pooling. An owner ID is a two character code that is used as the first two characters in a tape's volume serial number.

When a dataset is cataloged with an owner ID, it can only be written on tapes with volume serial numbers which begin with the code specified.

Tape ownership is different from tape pooling because a pool tape can only be used for datasets assigned to that pool. Tape ownership does not restrict the use of tapes in that way. Only the dataset is restricted. The tape can be used for any dataset.

Specifying Tape Size

You can restrict datasets to small or large tapes with the size specification feature.

Small tapes are distinguished by their volume serial numbers, which must end with S. Large tapes have no restrictions to their volume serial numbers. You can designate that a dataset should be written to small tapes with online or manual catalog maintenance, using the Tape Size field.

Revolving Tape Datasets

In some cases, you may want to use a specific set of tapes for a dataset, such that all versions of that dataset are contained on that set of tapes only.

The TSIDMNT CATALOG command contains a parameter (REV) that allows you to define revolving datasets.

Specifying REV=YES for a dataset establishes an exclusive relationship between the dataset and the tapes it uses: a revolving tape can only be used to write its own dataset, and a revolving tape dataset can only be written on its own tapes.

If the dataset is new or it does not own enough tapes, new tapes can be added with the ADD command and then scratched with the SCRATCH command (using Online Manager or the TSIDMNT program).

Multi-Dataset/Multi-Volume Processing

CA EPIC for z/VSE supports multi-dataset, multi-volume, and multi-dataset/multi-volume processing. A tape volume that contains a dataset that is a member of a multi-dataset group cannot be scratched until all datasets in the group have been scratched.

Work Tapes

A work tape dataset is one that has been cataloged with the attribute WRK=YES or defined with DISP=(NEW,DELETE) on a DD statement. A work tape dataset can be written on any available scratch tape. The data on a work tape is not usually saved after the end of the job in which it was created, so there is usually no need to place an external dataset label on a work tape. For this reason, the EP005 message (**LABEL**) is not issued for work tapes unless special option EPS043=YES.

Assigning Tape Drives

CA EPIC for z/VSE uses a process called *Automatic Volume Recognition (AVR)* to automatically search available tape drives for a required tape volume whenever a program issues an OPEN request. When AVR locates the required volume, CA EPIC for z/VSE makes the necessary assignment. When the program issues a CLOSE request, CA EPIC for z/VSE releases the assignment and makes the drive available to other programs unless directed otherwise by a parameter on the TLBL or DD statement.

It is the operator's job to ensure that tape drives are available to CA EPIC for z/VSE when it is searching for a tape. A tape drive is available if it:

- Contains a valid tape (the volume serial number is in the DSN Catalog, or the tape has been initialized by the TSIDINIT or DITTO program)
- Is not in the "device down" status
- Is not attached to any virtual machine (if Autoattach is on), or is attached to the machine running the job (if Autoattach is off)
- Is not assigned to any partition
- Is ready and at load point
- Supports the mode setting selected for the dataset

Under most circumstances, you do not have to make drive assignments with the VSE ASSGN statement. If Autoattach is on, you should avoid using the VM/CP ATTACH or DETACH commands to control the tape drives. Instead, DETACH every tape drive and let CA EPIC for z/VSE do the work for you.

If Autoattach is on and you do use the ATTACH command, remember to issue the VM/CP DETACH command for the tape drive when you are finished, so that CA EPIC for z/VSE can use the drive in another virtual machine.

CA EPIC for z/VSE only uses AVR for controlled tapes. If the required tape is not an CA EPIC for z/VSE-controlled tape, the operator will be asked to specify the address of the tape drive to be used.

Attention Routine Commands

Certain CA EPIC for z/VSE services can be invoked by commands that the operator issues to the VSE Attention Routine (AR).

Command Summary

The CA EPIC for z/VSE Attention Routine commands are:

Command	Action
EP ? (or EP HELP)	Displays help information for CA EPIC for z/VSE AR commands.
EP END ST*name*	Terminates an CA EPIC for z/VSE Dispatcher subtask. *This command should be used only when directed by CA EPIC for z/VSE Technical Support. name* is the name of the subtask being ended. ACL is a valid subtask name.
EP END TASK	Ends CA EPIC for z/VSE Dispatcher processing.
EP REW,*cuu*	Rewinds the selected tape drive.
EP RUN,*cuu*	Unloads the selected tape drive.
EP STATUS[,*cuu*]	Displays status of all accessible tape drives, or of a particular tape drive.
EP TRQ CANCEL *n*	Deletes a work request that has not been processed by the CA EPIC for z/VSE Dispatcher. *n* is the identification number of the request to be deleted. It is obtained using the EP TSTAT command.
EP TSTAT	Displays any work requests waiting to be processed by the CA EPIC for z/VSE Dispatcher.

How to Enter the Commands

CA EPIC for z/VSE Attention Routine commands must begin with "EP" followed by a space. For example, to unload the tape on drive 881, enter:

```
EP RUN,881
```

EP930 Message

The message EP930 UNABLE TO OBTAIN LOCK - COMMAND BYPASSED may be issued on some systems with heavy activity (particularly when two physical CPU's are active with SHARE=YES). The Attention Routine cannot afford to wait for a successful lock, so CA EPIC for z/VSE issues this message when it cannot obtain one immediately. Try the command again.

FREE and STATUS Commands

In addition to providing status information using the AR command facility, CA EPIC for z/VSE also allows the operator to respond FREE or STATUS to many CA EPIC for z/VSE messages.

The FREE command lists all tape drives that are:

- In READY status or with no tape mounted
- Not assigned to any partition
- Not attached to any other machine

The STATUS command lists the current status of each tape drive.

Mounting Tapes

Output

CA EPIC for z/VSE controls which tapes can be used for output. It checks the volume serial number of every tape processed. If the tape mounted is controlled, its volume serial number is checked with its corresponding DSN Catalog entry to ensure that it is in scratch status before it can be used as output.

If the tape is uncontrolled, CA EPIC for z/VSE checks to see if the tape's volume serial number exists in the DSN Catalog. If it does not yet exist, the tape is cataloged and used automatically. If the volume serial number already exists in the catalog, CA EPIC for z/VSE issues a message to allow the operator to enter a new volume serial number. You do not have to pre-initialize all your tapes to bring them under CA EPIC for z/VSE's control.

The configuration option UNCTAP affects uncontrolled tape handling.

Input

Requesting the Previous Version

To request the previous version of a dataset, type RERUN in response to CA EPIC for z/VSE's request for a tape mount. A new tape mount request will then be issued for the previous version.

Example

If a tape mount request is issued for version 1 and you reply RERUN, a new request is issued for version 2. You can reply RERUN as many times as there are versions for the dataset.

Requesting a Specific Version

To request a specific version, type RERUN,n in response to a request for a tape mount, where *n* is the number of the version requested.

Example

To issue a new tape mount request for version number 3, type RERUN,3.

Using ACCEPT

The operator can be given the option to reply ACCEPT to a mount request to force CA EPIC for z/VSE to use any desired volume. Each time ACCEPT is used, a notation is made in the Accepted Tapes Report generated by TSIDBRP. The use of the ACCEPT reply is governed by the ACCEPT configuration option.

Important! *ACCEPT=YES disables CA EPIC for z/VSE's supervision of the input process and places responsibility for tape selection in the hands of the operator.*

Initializing Tapes (TSIDINIT)

TSIDINIT initializes tapes with CA EPIC for z/VSE VOL1 labels and enters them in the DSN Catalog under the dataset name '***.VOLUME.DSN.**'. The tapes are then ready for use as controlled scratch tapes.

Up to ten tape drives can be used in the initialization process.

TSIDINIT can be executed from JCL or from the operator console. If it is executed from the console, the operator is queried for all processing information.

To execute from the console, enter:

```
EXEC TSIDINIT
```

after any VSE '// PAUSE' statement.

JCL Requirements

```
 [// UPSI xxxx]
// EXEC TSIDINIT [,PARM='CONSOLE']
         cuu statement
         volser statements
/*
```

Types of Initialization

When more than one tape is to be initialized, you can specify each tape's volume serial number in individual control statements, and the statements can be placed in any order. This is referred to as *random initialization*. You can also specify the starting volume serial number, and TSIDINIT will initialize successive tapes in ascending volume serial number sequence. This is referred to as *sequential initialization*. In sequential initialization, TSIDINIT continues to initialize tapes until the operator replies STOP to the mount message.

Sequential initialization requires that the last four characters of the volume serial number be numeric (for example, EP1200 or A67019). Random initialization places no restrictions on the volume serial number.

TSIDINIT normally checks each tape's actual volume serial number before writing in the tape to ensure that an active tape is not initialized by mistake. It also protects against the creation of duplicate volume serial numbers. If one of the volume serial numbers specified already exists in the DSN Catalog, CA EPIC for z/VSE issues an error message and request a new volume serial number. If you are initializing brand new tapes that do not have tape marks, you must specify *forced initialization*.

If it is necessary to reinitialize a tape that is currently in scratch status, execute TSIDINIT with UPSI xxx1.

Important! *Caution should be exercised with forced initialization. It bypasses tape label checking, so active data can be lost if the wrong tape is mounted. Forced initialization also eliminates protection against duplicate volume serial numbers. If the volume serial number already exists in the DSN Catalog, TSIDINIT issues an error message, but the tape is initialized with the duplicate volume serial number.*

UPSI Values

To specify...	Use UPSI setting...
Sequential initialization	1xxx
Random initialization	0xxx
Forced initialization	x1xx
Scratch tape reinitialization	xxx1
Inclusion of the CUU statement	xx1x

PARM Values

PARM='CONSOLE' can be specified on the VSE EXEC statement to allow the operator to control initialization. This has exactly the same effect as entering EXEC TSIDINIT after a VSE '// PAUSE' statement.

CUU Statement

Normally, the operator is asked to specify the tape drives to be used. If specific tape drives are to be used, operator intervention can be eliminated by using a CUU statement. Up to ten drives can be specified. The format of the CUU statement is as follows:

Syntax

CUU=*cuu*,*cuu*,*cuu*,...

The CUU statement requires the use of UPSI xx1x in your JCL.

VOLSER Statements

VOLSER statements identify which volume serial numbers are written in the VOL1 label.

If random initialization is requested, one VOLSER statement is required for each volume serial number. If sequential initialization is requested, use only one statement to specify the starting volume serial number.

Examples

In this example, TSIDINIT performs sequential initialization of uncontrolled tapes that have labels or tape marks. It initializes the first tape with the volume serial number EP0001, the second tape with EP0002, the third tape with EP0003, and so on until the operator stops the process. It does label checking to ensure that the tape does not contain an active dataset. It also checks the DSN Catalog to ensure that there is no duplication of volume serial number before it initializes each tape.

```
// JOB INITTAPE
// UPSI 1000
// EXEC TSIDINIT
    EP0001
/*
/&
```

In this example, TSIDINIT performs random initialization of uncontrolled tapes that do not have labels or tape marks. It initializes the first tape with the volume serial number PR0001, the second tape with PR0024, the third tape with PR0047. It does not prevent duplication of volume serial number and does not do label checking to ensure that the tapes mounted are valid scratch tapes.

```
// JOB INITTAPE
// UPSI 0100
// EXEC TSIDINIT
    PY0001
    PY0024
    PY0047
/*
/&
```

Printing External Volume Serial Labels (TSIDLBL)

TSIDLBL creates external volume serial number labels in large block characters. These labels contain the volume serial number and a short optional comment.

Standard spacing for each label is 1 line for the comment, 7 lines for the volume serial number, and two blank lines between labels.

Example

```
THIRTY CHARACTER COMMENT
****      ****      ****      ****      ****      *
*         *  *      *  *      *  *      *  *     **
*         *  *      *  *      *  *      *  *      *
****      ****      *  *      *  *      *  *      *
*         *         *  *      *  *      *  *      *
*         *         *  *      *  *      *  *      *
****      *         ****      ****      ****      *
```

JCL Requirements

```
 [// UPSI 1]
// EXEC TSIDLBL
control statements
/*
```

UPSI Values

UPSI 1 causes the printer to skip to channel 1 (top-of-form) after each label. This allows you to use labels of any size. When UPSI 1 is used, TSIDLBL assumes that the printer has a correctly loaded forms control buffer or carriage control tape that defines the size of the labels.

Syntax

```
volser[-volser] [ comment ]
```

Parameters

Parameter	Required?		Valid Entry
volser-volser	Yes		1 to 6 characters
comment	No	1 to 30 characters	None

volser **[** *-volser* **]** specifies the range of volume serial numbers for the labels you want to create. If only one label is to be created, omit the second *volser*. The last four characters in the volume serial number must be numeric.

comment specifies the comment to be printed on each of the labels.

Example

In this example, EP0001-EP0100 DSN CONTROLLED indicates that one label is to be created for each volume serial number from EP0001 to EP0100, inclusive. Each label in this series will contain the comment 'DSN CONTROLLED'.

PR0001-PR0050 PAYROLL indicates that one label is to be created for each volume serial number from PR0001-PR0050, inclusive. Each label in this series will contain the comment 'PAYROLL'.

```
// JOB PRINTLBL
// EXEC TSIDLBL
EP0001-EP0100 DSN CONTROLLED
PR0001-PR0050 PAYROLL
/*
/&
```

Printing Dataset Labels (TSIDBPR and TSIDPRT)

With CA EPIC for z/VSE, you can create gummed dataset labels automatically when datasets are opened for output.

You can create labels in a batch partition or through CICS. The TSIDBPR program creates labels in a batch partition. The TSIDPRT transaction creates labels through CICS.

Automatic label creation requires you to write a subroutine. For installation and subroutine requirements, see the *Installation and System Guide*. Dataset label creation is enabled using the RMNTPRT configuration option.

Label Creation in a Batch Partition

Execute TSIDBPR in any partition. Be sure to use a partition which is not needed for other jobs, since TSIDBPR remains active in that partition until it is deactivated.

To deactivate TSIDBPR, execute TSIDPEN in any partition. This terminates TSIDBPR and releases the label printer.

Label Creation in CICS

TSIDPRT can be activated automatically through the CICS PLTPI table. If it is activated using the PLTPI, enter the TPRT transaction in CICS. When you activate TSIDPRT this way, the terminal you use is not released until the TPRT is deactivated or until CICS is terminated.

To deactivate TPRT, enter the TPRE transaction in CICS or execute TSIDPEN in any available partition. This releases the label printer and any terminal used by TPRT.

Tape Vaulting

CA EPIC for z/VSE's vaulting subsystem controls tape vaulting automatically. This section presents an overview of the programs and procedures involved in tape vaulting.

Components

Vault locations are established using the VAULTDEF macro, which defines the name and, if necessary, the size of each vault and the number of slots it contains. Vaults are identified by integers. The main library is always Vault 0 (zero). CA EPIC for z/VSE assigns the other vault numbers to each vault in the sequence defined by the VAULTDEF macro.

Vaulting methods are established using the METHOD macro. Each method defines a distinct sequence of vault locations for all versions of a dataset vaulted with that method.

The VAULTDEF and METHOD macros are documented in the *Installation and System Guide*.

How Vaulting Works

Datasets are designated for vaulting using the DSN Catalog. If a dataset has a vaulting method assigned to it, its versions are vaulted automatically whenever the vault movement program (TSIDVLT) is run.

TSIDVLT moves cataloged tape volumes to their new vault locations (and slot numbers if defined). In addition, TSIDVLT creates two vault movement reports. Each report lists only the tapes which need to be moved, sorted by vault location. These reports are:

Report	displays...
Picking List	Tapes listed by their current vault and slot location and indicating the vault and slot destination to which each tape is to be moved.
Receiving List	Tapes listed by their newly assigned vault and slot destinations and indicating the current vault and slot where each tape can be found.

Vaulting ACL Cartridges

If an Automated Cartridge Library (ACL) is supported, TSIDVLT also writes the volume serial numbers of all cartridges moved from Vault 0, in the ACL Inventory Management Library. At a later time, these volumes can be ejected from the ACL using the EP ACL EJECT command or the TSIDUTL program's ACLEJECT control statement. See the *Installation and System Guide* for more information on these commands.

Additional Reports

Reports listing current vault inventories can be created using TSIDUTL. TSIDPRV creates a report that cross references datasets with the vaulting methods assigned to them. See the *Installation and System Guide* for more information on TSIDUTL.

Assigning Vaulting Methods to Datasets

After vaults and vaulting methods have been defined, they must be assigned to datasets before automatic vaulting can occur. This is done using the DSN Catalog.

Vaulting methods are assigned to datasets using the CATALOG or UPDATE command (in Online Manager or TSIDMNT).

In Online Manager, place the vault method number in the Vault Method field of the EDD.

In TSIDMNT, use the VLT parameter to supply the vault method number to the EDD.

Moving Volumes (TSIDVLT)

TSIDVLT should be executed every day. TSIDVLT moves tapes from one vault to another and creates movement reports. TSIDVLT can be instructed to:

- Perform actual vaulting actions. Movement from one vault/slot to another occurs and a report describing the activity is produced.
- Perform projected vaulting actions. A report is produced describing what movements would take place, but no actual vault/slot movement occurs.

JCL Requirements

```
// DLBL SORTWK1...
// EXTENT SYS001
// EXEC TSIDVLT
  control statements
```

Control Statements

Four input control statements are provided to tell TSIDVLT what action to take. The control statements are read from SYSIPT and can begin in any column. Only one control statement can be used in any execution of TSIDVLT. A control statement must be present; there is no default.

Control Statement	Function
MOVESLT	Do vault movement and produce a report sorted by vault and slot number.
MOVESER	Do vault movement and produce a report sorted by volume serial number.
PROJSLT	Do not do vault movement, but produce a report of what movement would occur, sorted by vault and slot number.
PROJSER	Do not do vault movement, but produce a report of what movement would occur, sorted by volume serial number.

Dataset Requirements

SORTWK1 is required.

Example

```
// JOB VAULTMOV
// DLBL SORTWK1,'===.SORTWK1,(D)'
// EXTENT SYS001
// EXEC TSIDVLT
MOVESLT
/*
```

Reporting Current Locations of Vaulted Tapes

The DSPLY function of TSIDUTL creates reports which provide vault information for tape volumes. These reports can select and display tape datasets by vault.

Parameter	Action	Example
ALL	Lists vault information for all tape datasets.	DSPLY ALL,SEQ=DSN
VAULT	Lists the tapes contained in every vault except Vault 0 (main library). When SEQ=VAULT is specified, tapes are listed in vault and slot sequence and a page break occurs after each vault.	DSPLY VAULT,SEQ=VAULT
VLT	Lists the contents of a single vault.	DSPLY ALL,VLT=2,SEQ=DSN

Manually Controlled Vaults

A *manually controlled vault* is any vault whose number is greater than the value of the AVAULTS parameter in the VAULTDEF macro. Since manually controlled vaults are not known to CA EPIC for z/VSE, they must be tracked manually.

Any version of any dataset can be moved to a manually controlled vault. Tapes are moved to or from manually controlled vaults using the MOVE function of Online Manager or TSIDMNT.

Important! *If a dataset is under the control of an automatic vaulting method, TSIDVLT executions override any manual vaulting.*

Vault To Dataset Cross-Reference Report (TSIDPRV)

TSIDPRV produces the Vault Method Report. This report lists the vaulting method for each dataset, in dataset name sequence. For a dataset to appear on this report, it must have a vaulting method assigned to it in the DSN Catalog.

JCL Requirements

```
// DLBL SORTWK1...
// EXTENT SYS001
// EXEC TSIDPRV
```

UPSI Values

None.

Dataset Requirements

SORTWK1 is required.

Example

```
// JOB VLTMETH
// DLBL SORTWK1'===.SORTWK1,(D)'
// EXTENT SYS001
// EXEC TSIDPRV
/&
```

Dataset to Tape Report (TSIDDIS)

TSIDDIS produces a summary report correlating dataset versions and tape volume serial numbers. It is listed in dataset name order. There is one entry for each dataset version. If a version spans more than one volume, there is an entry for each volume.

Note: For an example of this report, see Appendix A.

JCL Requirements

```
// DLBL SORTWK1
// EXEC TSIDDIS
```

UPSI Values

None.

Dataset Requirements

SORTWK1 is required.

Example

```
// JOB TAPE DATASET LISTING
// DLBL SORTWK1,'==.SORTWK1'
// EXTENT SYS001
// EXEC TSIDDIS
/&
```

Tape to Dataset Report (TSIDSDS)

TSIDSDS produces a summary report correlating dataset versions and tape volume serial numbers. It is listed in volume serial number order. There is at least one entry for each version of a dataset. If a version spans more than one volume, there is an entry for each volume.

Note: For an example of this report, see Appendix A.

JCL Requirements

```
// DLBL SORTWK1
// EXEC TSIDSDS
```

UPSI Values

None.

Dataset Requirements

SORTWK1 is required.

Example

```
// JOB TAPE VOLSER LISTING
// DLBL SORTWK1,'==.SORTWK1'
// EXTENT SYS001
// EXEC TSIDSDS
/&
```

Chapter 13: Disk Space Management

This section contains the following topics:

Overview

This chapter discusses how CA EPIC for z/VSE can help you to maximize and manage your disk space.

Dataset Fragmentation

When you create a dataset with a large primary allocation, there may not be a large enough free space in the pool to contain the entire primary extent. CA EPIC for z/VSE can respond to this condition by allocating the largest contiguous free space to the primary extent and allocating secondary extents when the primary extent becomes full. This process is called *dataset fragmentation*.

Fragmentation can be enabled globally by specifying the configuration option FRAG=YES. Fragmentation can be enabled for an individual dataset by:

- Specifying the "F" option on a DLBL statement
- Specifying the MXIG or ALX subparameter in the SPACE parameter on a DD statement
- Allowing the operator to respond FRAG to the EP038 message

Secondary Allocation

CA EPIC for z/VSE automatically allocates secondary extents to eligible datasets when the primary extent has been filled.

Secondary allocation is not available for direct access or TYPEFLE=WORK datasets. Secondary allocation is available for SORTWK datasets only if the SORT program specifically allows it.

For cataloged datasets, you can specify the size of the secondary extents and the maximum number of secondary extents (up to 255). If you use DD statements for your disk datasets, the size of the secondary extents can be specified in the SPACE parameter, and the maximum number of secondary extents can be specified in the VOLUME parameter. If you do not define a dataset's secondary allocation requirements in one of these ways, CA EPIC for z/VSE automatically allocates up to 255 secondary extents, and each extent will be one-half the size of the primary extent.

Truncation

Any allocated extent that was not completely filled can be truncated so that the unused portion of the extent is freed. CA EPIC for z/VSE performs truncation for controlled datasets by altering both the VTOC and the DSN Catalog to show only the space actually used. Extents are always truncated on track or control interval boundaries.

Truncation occurs only on output for sequential disk datasets. It should be inhibited for TYPEFLE=WORK DTFs and for datasets that are opened as sequential I/O but are not accessed sequentially (for example, DL/1 datasets). Truncation is automatic for datasets that you define with DLBL statements. It must be requested using the RLSE subparameter of the SPACE parameter when using CA EPIC for z/VSE DD statements.

Truncation can also be inhibited by the DLBL option "N".

Converting Uncontrolled Disk Datasets

Existing uncontrolled disk datasets can be placed under CA EPIC for z/VSE control. Any existing dataset can be controlled, as long as all its extents reside in a single defined disk pool. You can convert a disk dataset to catalog control or Start Track 1 control.

Convert to Catalog Control

1. Catalog the dataset name as a work file using the Online Manager (specify "WRK" on the Define an EDD panel) or the TSIDMNT CATALOG command (specify the WRK=YES parameter).
2. Renumber the dataset's extents using the TSIDVUT RENUM function. CA EPIC for z/VSE-controlled disk files must begin with extent sequence number 1.
3. Change the dataset's VTOC expiration date to permanent using the TSIDVUT RETAIN function. Catalog-controlled disk files are scratched according to the retention criteria specified in the EDD, not the VTOC expiration date.
4. Add the version to the DSN catalog using the Online Manager or the TSIDMNT ADD command. Be sure to specify the version block size, if it is known.
5. If generation retention for the dataset has been specified, update the file type from WRK to GEN using the Online Manager or TSIDMNT UPDATE command after the existing version has been added to the DSN catalog.

Convert to Start Track 1

1. Renumber the dataset's extents using the TSIDVUT RENUM function. CA EPIC for z/VSE-controlled disk files must begin with extent sequence number 1.
2. If necessary, change the dataset's VTOC expiration date using the TSIDVUT RETAIN function. Expired files in an CA EPIC for z/VSE disk pool are eligible for immediate reallocation.
3. Change the dataset's EXTENT statements to specify a starting track of 1, or whatever number was specified in the STRTRK configuration option.

Example

```
// EXTENT SYS001,POOL50,1,0,1,10
```

Placing Datasets on Specific Volumes in a Pool

All disk datasets can be assigned to an CA EPIC for z/VSE pool. If your pool is made up of more than one volume, the dataset can be placed on any of those volumes. If you wish, you can place datasets on a specific volume.

Procedure

1. Quiesce all other volumes in the pool.
2. Create the dataset and inhibit truncation.
3. Reuse the same space when re-creating the dataset. For more information, see Reusing the Same Space. (see page 224)

VTOC Utilities

CA EPIC for z/VSE provides the following VTOC utilities: TSIDLVT, TVTO, and TSIDVUT.

Utility	Description
TSIDLVT	Displays VTOCs by CA EPIC for z/VSE pool or by volume, and in extent or dataset sequence. Can display both active and expired datasets, or active datasets only.
TVTO	Online utility. Displays VTOCs by volume only.
TSIDVUT	Allows modification of existing entries. Deletes active datasets, retains a dataset permanently, renames a dataset, or converts an uncontrolled dataset to CA EPIC for z/VSE control. See the *Installation and System Guide* for information.

Important! *Use TSIDVUT to update the entries of uncataloged datasets only.*

Printing VTOC Entries (TSIDLVT)

TSIDLVT produces printed reports of VTOC entries. It can display entries for an entire CA EPIC for z/VSE pool or a specific volume.

TSIDLVT prints expired datasets, unexpired datasets, and free space. Extents are shown in track/block format and dates in MM/DD/YYYY format. Control statements can be used to change the display format.

At the end of each report, TSIDLVT prints the total free space in the pool or volume. This total includes expired extents as well as unused space.

JCL Requirements

```
// DLBL SORTWK1,...
// EXTENT SYS001
// EXEC TSIDLVT
  control statements
/*
```

UPSI Values

None.

Dataset Requirements

SORTWK1 is required for all functions.

Control Statements

You must include one control statement per report. The control statement must specify the volume or pool to be displayed.

You can use control statement parameters to customize the report.

The EXP, UNEXP, and FREE parameters limit the contents of your report.

For example, if your control statement specifies:

```
SER=SYSWK1,UNEXP
```

the report lists only unexpired datasets.

To receive a report containing only expired datasets and free space, use a control statement like this one:

```
SER=SYSWK1,EXP,FREE
```

If none of these parameters are specified, all three types of data will appear on the report.

Control Statement Parameters

```
{SER=volser }
{poolname   }

  [ ,EXP ]
  [ ,UNEXP ]
  [ ,FREE ]
  [ ,DSN ]
  [ ,INPOOL ]
  [ ,EURO ]
  [ ,CCHH ]
```

Parameters

Parameter	Required?	Valid	Default
One of the following two is required:			
volser		1 to 6 characters	None
poolname		1 to 6 characters	None
ALL	No		ON
UNEXP	No		OFF
EXP	No		OFF
FREE	No		OFF
DSN	No		OFF
INPOOL	No		OFF
EURO	No		OFF
CHH	No		OFF

You must specify either SER=*volser* or *poolname*.

volser specifies the disk volume serial number to report.

poolname specifies the name of the CA EPIC for z/VSE disk pool to report.

EXP prints expired datasets.

UNEXP prints unexpired datasets.

FREE prints free space.

DSN prints in dataset name sequence. It is ignored if FREE is also specified.

INPOOL lists extents within pool limits only. If INPOOL is not specified, TSIDLVT ignores pool limits on a volume.

EURO prints dates in European format (dd/mm/yyyy). By default dates are printed in USA format (mm/dd/yyyy).

CCHH prints extents in cylinder/head (cchh) format. If cchh is not specified, extents will be shown in relative track/block format.

Examples

The following job creates a report of the VTOC of volume SYSWK1.

```
// JOB VTOC
// DLBL SORTWK1,'===.SORTWK1,(X)'
// EXTENT SYS001
// EXEC TSIDLVT
  SER=SYSWK1
/*
/&
```

The following job creates two reports. The first one, reporting POOL50, prints the standard report. The second, reporting POLFBA, lists only unexpired and expired datasets. They are listed in dataset name sequence.

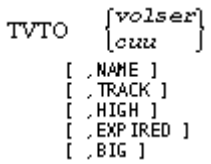
```
// JOB VTOC
// DLBL SORTWK1,'===.SORTWK1,(X)'
// EXTENT SYS001
// EXEC TSIDLVT
  POOL50
  POLFBA,UNEXP,EXP,DSN
/*
/&
```

Online VTOC Displays (TVTO)

The TVTO transaction displays VTOC entries by volume.

The format of the display is tailored by options entered with the TVTO transaction. The options must be separated by a comma or a space. To terminate the transaction, press CLEAR or PA2.

Syntax



Parameters

Parameter	Required?	Valid	Default
One of the following two (required):			
volser		1 to 6 characters	None
cuu		3 characters	None
NAME	No		OFF
TRACK	No		OFF
HIGH	No		OFF
EXPIRED	No		OFF
BIG	No		OFF

volser specifies the volume serial number of the disk volume to display.

cuu specifies the channel and unit number of the disk volume to display.

NAME sorts the display by dataset name. If NAME is not specified, the display is sorted by disk address.

TRACK displays disk addresses in relative track format. If TRACK is not specified, addresses are displayed in cylinder-head format.

HIGH displays the starting and ending disk addresses. If HIGH is not specified, the display shows the starting address and the total space used by the dataset.

EXPIRED displays both expired and active datasets. Expired datasets are marked with an asterisk (*).If EXPIRED is not specified, only active datasets are displayed.

BIG allows display of VTOCs of up to 900 extents. If BIG is not specified, the maximum number of extents TVTO can display is 390. BIG increases CICS storage requirements for the TVTO transaction. If CICS storage is exhausted, the transaction is terminated. If this occurs, you can run the report in batch, try the transaction again when CICS is less active, or increase CICS storage.

Valid PF keys

PF Key	Explanation
PF1/PF13	Display the next screen
PF2/PF14	Display the previous screen
PF3/PF15	Display the first screen
PF4/PF16	Display the last screen
PF5/PF17	Remap the current VTOC and redisplay
PA2 or CLEAR key	Terminate the transaction

```
VOLUME TABLE OF CONTENTS FOR VOL=EPC360 ON 360   10/258 11:41:54 PAGE   1 OF   2
________ FILE NAME ____________________________      LOW     HIGH C SEQ   CRE   EXP
                                FREE.SPACE 00000-01 00017-02
)NKNOC002.CAT.FILE                G=000000 00017-03 00017-03 ? 000 10041*10041
                                FREE.SPACE 00017-04 00926-14
DOS.PAGING.FILE.FF61008A2097                00927-00 01291-14 ? 000 10256+99366
                                FREE.SPACE 01292-00 03022-09
TECHEPIC.CICS23.USER.JOURNAL.B              03022-10 03022-14 S 000 10258*10258
TECHEPIC.CICS23.USER.JOURNAL.A              03023-00 03023-09 S 000 10258*10258
TECHEPIC.CICS23.SYSTEM.LOG.B                03023-10 03023-14 S 000 10258*10258
TECHEPIC.CICS23.SYSTEM.LOG.A                03024-00 03024-09 S 000 10258*10258
TECHEPIC.CICS23.DUMP.DATASET.B              03024-10 03026-09 S 000 10133*10133
TECHEPIC.CICS23.DUMP.DATASET.A              03026-10 03028-09 S 000 10258*10258
TECHEPIC.CICS23.DFHBUXT                     03028-10 03030-09 S 000 10133*10133
TECHEPIC.CICS23.DFHAUXT                     03030-10 03032-09 S 000 10133*10133
                                FREE.SPACE 03032-10 03034-09
TECHEPIC.CICSTS.DFHAUXT                     03034-10 03036-09 S 000 10111*10111
TECHEPIC.CICSTS.DUMP.DATASET.B              03036-10 03038-09 S 000 10111*10111
TECHEPIC.CICSTS.DUMP.DATASET.A              03038-10 03040-09 S 000 10258*10258
ICCF.BKUP                                   03040-10 03074-09 S 000 10111*10111
TECHEPIC.VSE.DUMP.LIBRARY                   03074-10 03114-09 S 000 10112+99366
TECHEPIC.INFO.ANALYSIS.EXT.RTNS.FILE        03114-10 03114-10 S 000 10112+99365
                                FREE.SPACE 03114-11 03114-14
PF:  1=NEXT 2=PREV 3=FIRST 4=LAST 5=REFRESH     39897 FREE TRACKS IN   5 EXTENTS
```

Offload System

CA EPIC for z/VSE provides offload/onload and backup/restore functions for controlled sequential disk dataset versions identified in the CA EPIC for z/VSE catalog. Offload is only available on CA EPIC for z/VSE systems supporting both disk and tape management. Once copied to tape, offload deletes the disk VTOC entry and identifies that version as now residing on tape only whereas backup leaves the disk version in place. The dataset population to be processed by the offload or backup functions are identified by control statements created manually or by the offload system itself.

Onload restores datasets previously offloaded or backed-up to tape. Disk leveling is invoked when restoring multiple datasets as well as multiple extents for a single dataset. After restored to disk, an offloaded dataset version is identified as residing on disk only whereas a backed-up version is always identified as a newly created version 1.

Programs

The offload system consists of three programs:

Program	Function
TSIDOFD	Selects datasets for offload and backup by creating control statements for TSIDOFL.
TSIDOFL	Offloads and backs up selected datasets.
TSIDONL	Restores offload and backup datasets back to disk and creates a listing of datasets on the offload tape.

High Performance I/O

The CA EPIC for z/VSE offload/onload subsystem uses I/O routines from CA's FAVER2™ product. These provide high performance through the use of physical I/O, larger block sizes, and a single tape file. The offload subsystem is implemented automatically as soon as CA EPIC for z/VSE is activated (no special conversion to the FAVER-based subsystem is required). All JCL, control statements, program names, and procedures remain the same. The new onload/restore program automatically processes offload/backup tapes created under previous releases of CA EPIC for z/VSE, so complete access to existing offload and backup datasets is assured.

Under the FAVER-based subsystem, offload datasets are onloaded to temporary disk files instead of being processed directly from the offload tape. These temporary disk files are deleted at the end of the job that uses them. This improves performance and allows concurrent access to multiple files on the same offload tape, which was not possible with the previous offload subsystem.

Important! *The high performance feature requires sufficient disk space whenever any offload datasets are processed. CA EPIC for z/VSE uses the disk pool specified in the EDD. If an "insufficient space" condition occurs while creating the temporary disk files, CA EPIC for z/VSE allocation options (such as FRAG and alternate pools) can be used.*

Selecting Datasets To Be Offloaded (TSIDOFD)

TSIDOFD reads the DSN Catalog and creates control statements for TSIDOFL based on the selection criteria you specify. These control statements determine which datasets will be offloaded or backed up. You specify selection criteria to TSIDOFD using INCLUDE and OMIT control statements.

An internal table is used during the dataset selection process to store each version selected. The capacity of this table is dependent on the amount of excess partition storage available (up to and not including partition GETVIS storage) and will never be less than 100. This internal table size is increased by approximately 21 entries for each 1K of excess partition storage available beyond the first 5K. For example, if SIZE=(TSIDOFD,50K) is specified on the EXEC statement, then the internal table capacity would be about 1066 entries. This internal table is maximized by not specifying the SIZE operand on the EXEC statement and allowing all excess partition storage available to be utilized. If the DATASET SELECTION REPORT is produced, the actual capacity of the internal table for the current processing is stated on the second report heading line. If this program attempts to select more dataset versions than can be held in this internal table, no output will be created and this program will terminate with a return code of 4 after issuing message EP618.

Selection Criteria

Datasets can be included for offload/backup by:

- pool name
- volume serial number
- dataset name
- dataset names beginning with a specific character string
- version number
- versions created since the last time the TSIDUTL program was executed with the DSPLY NEW function
- user ID
- system ID
- access within the last *n* days (SLA)
- no access within the last *n* days (DLA)

SLA and DLA are mutually exclusive.

Datasets can be omitted by:

- pool name
- volume serial number
- dataset name
- version number
- user ID
- system ID
- dataset names beginning with a specific character string

JCL Requirements

```
// xLBL OUTPUT,'sysin.data.set'  ( optional TLBL or DLBL )
//EXEC TSIDOFD {,SIZE=...}
control statements
/*
```

UPSI Values

None.

Dataset Requirements

The control statements created by TSIDOFD are created on SYSPCH if SYS005 is assigned to SYSPCH. Control statements can be written to tape or disk by supplying a TLBL, DLBL, or DD with the DTF name OUTPUT.

Control Statements

An internal table is used during the dataset selection process to store each version selected. The capacity of this table is dependent on the amount of excess partition storage available up to and not including partition GETVIS storage but will never be less than 100. If the DATASET SELECTION REPORT is produced, the actual capacity of the internal table for the current processing is stated on the second report heading line.

Dataset selection criteria are specified using INCLUDE and OMIT control statements. You can specify multiple parameters on any INCLUDE or OMIT statement, but parameter expressions cannot be continued. You can submit up to 25 INCLUDE statements showing the DSN= operand and 25 OMIT statements showing the DSN= operand per execution.

AND logic always applies to multiple selection criteria is specified, regardless of whether submitted on a single control statement, multiple control statements, or both.

One primary set of parentheses must enclose all the parameter expressions on a single control statement. Multiple parameter expressions can be separated by commas or blanks. The parameters ALLVER, VOL, and POL each allow you to specify more than one value; in such cases, the values must be separated by commas within a set of parentheses.

Syntax

```
REPORT=YES|NO

INCLUDE   (
  [ DSN='dataset-name' ]
  [ ,VER=n ]
  [ DSN='string.ALL' ]
  [ ALLVER=n ]
  [ DLA=n ]
  [ UID=x ]
  [ SID=x ]
  [ SLA=n ]
  [ POL=poolid,poolid,... ]
  [ VOL=volid,volid,... ]
  [ NEW ]
           )

OMIT    (
  [ DSN='dataset-name' ]
  [ ,VER=n ]
  [ ALLVER=n ]
  [ DLA=n ]
  [ UID=x ]
  [ SID=x ]
  [ POL=poolid,poolid,... ]
  [ VOL=volid,volid,... ]
          )
```

Parameters

Parameter	Required?	Valid	Default
ALLVER	No	1 to 9999	None
DLA	No	1 to 365	None
DSN	No	1 to 44 characters	None
NEW	No	None	OFF
POL	No	1 to 6 characters	None
SID	No	2 characters	None
SLA	No	1 to 365	None
UID	No	2 characters	None
VER	No	1 to 9999	None
VOL	No	1 to 6 characters	None

REPORT specifies whether the Preliminary Dataset Selection Report is to be printed. This report shows the size of the internal selection table (in heading 2), the selection criteria used, and the actual dataset versions selected.

INCLUDE specifies that the datasets that follow to be included for offload or backup are identified by the subparameters that follow. At least one INCLUDE statement must be used.

OMIT specifies that the datasets to be omitted from offload or backup are identified by the subparameters that follow. OMIT acts on datasets that have been included with an INCLUDE statement.

POL selects datasets that are allocated to the pool specified in their CA EPIC for z/VSE dataset definition (EDD). Up to 10 pool IDs can be specified with INCLUDE, and another 10 can be specified for OMIT. You can specify multiple pool IDs on a single POL parameter (POL=pool1,pool2,pool3...)

The POL parameter is checked against only the *current* pool name in the dataset's definition. If that pool name has been changed, you must use other options to offload those other datasets.

VOL selects datasets from the specified disk volume serial number. Up to 10 volume serial numbers can be specified for INCLUDE, and another 10 can be specified for OMIT. You can specify multiple volume serial numbers on a single VOL parameter (VOL=vol1, vol2, vol3,...)

DSN selects a single dataset name or a set of datasets beginning with the same character string. The value specified must be enclosed in apostrophes. To specify a group of datasets, specify '.ALL' at the end of the dataset name.

VER selects a version number for a particular dataset. It can only be supplied if the DSN parameter is also supplied. If omitted, all versions of the dataset are selected.

The following subparameters can be specified only once for INCLUDE and once for OMIT.

ALLVER selects datasets by version number. ALLVER affects all datasets to be selected (VER affects only a specific dataset). You can specify up to five versions with a single ALLVER parameter. (e.g., ALLVER=n,n,n,n,n).

SID selects datasets with the specified system ID.

UID selects datasets with the specified user ID.

DLA selects datasets which have not been accessed for the specified number of days. DLA is valid only for INCLUDE.

SLA selects datasets which have been accessed within the specified number of days. SLA is valid only for INCLUDE.

NEW selects datasets which have been created since the last time the TSIDUTL program was executed with the DSPLY NEW function. NEW is valid only for INCLUDE.

Offloading and Backing Up Datasets (TSIDOFL)

TSIDOFL backs up or offloads selected datasets according to the settings on an UPSI statement and the contents of control statements.

Control statements indicating which datasets are to be processed can be generated by TSIDOFD or specified manually. They are supplied as SYSIPT data. TSIDOFL sorts the control statements and copies the datasets to tape in order by descending dataset name and version number.

The offload tape is always associated with a cataloged dataset. You can use the default offload dataset OFFLOAD.DATA.SET, or you can create your own dataset. If you create your own dataset, you must catalog it and supply a TLBL or DD statement using the DTF name OUTPUT. The tape is retained until every dataset it contains has expired. If you want to retain the tape longer, you can specify special retention in the offload dataset's catalog entry.

JCL Requirements

When the control statements are supplied as SYSIPT data:

```
[// TLBL OUTPUT,'offload.dataset.name']
// UPSI xx
// EXEC TSIDOFL
[control statements]
/*
```

When the control statements are supplied from a tape or disk dataset (using GDI for SYSIPT):

```
[// TLBL OUTPUT,'offload.dataset.name']
// UPSI xx
* //IJSYSIN DD DSN=sysin.dataset,DISP=(OLD,DELETE,KEEP)
// EXEC TSIDOFL
/*
```

When the control statements are supplied from a tape or disk dataset: (using a standard VSE assignment for SYSIPT):

```
{// TLBL OUTPUT,'offload.dataset.name'}
// DLBL IJSYSIN,'sysin.dataset'
// EXTENT SYSIPT
ASSGN SYSIPT,DISK,VOL=volser,SHR
// UPSI xx
// EXEC TSIDOFL
/*
CLOSE SYSIPT,SYSRDR
```

```
{// TLBL OUTPUT,'offload.dataset.name'}
// TLBL IJSYSIN,'sysin.dataset'
// EXEC TSIDASS
OPEN IJSYSIN,SYS020,INPUT
/*
// ASSGN SYSIPT,SYS020
// UPSI xx
// EXEC TSIDOFL
/*
// RESET SYSIPT
```

UPSI Values

UPSI 0 specifies OFFLOAD.

UPSI 1 specifies BACKUP.

UPSI x1 specifies special handling when the number of control statements exceeds the internal work area used by TSIDOFL. When this UPSI setting is used, the control statements are not sorted, and datasets are offloaded in the order in which the control statements are submitted.

UPSI xx11offers an alternative to UPSI x1 in the handling of extra large numbers of control statements. It will allocate a larger internal work table in 24 bit partition getvis. The default table, without UPSI xx11, holds approximately 2300 statements.

The switches may be coded in several combinations to vary the amount of storage acquired:

UPSI Switch	Additional Partition Getvis Required	Files Supported
xx00xxxx	0 K bytes	2,285 (default)
xx10xxxx	512 K bytes	7,313
xx01xxxx	1024K bytes	14,627
xx11xxxx	1536 K bytes	21,941

Note that the second UPSI switch must be zero (// UPSI x0xxxxxx) in order for these switches to have any meaning.

If the program cannot obtain the requested GETVIS it will issue warning message EP644 and attempt to continue using the default table.

Note that TSIDONL has similar UPSI settings to specify a larger internal work table, but different switches are used in that program.

UPSI xxxx1 specifies that the output tape blocksize be reduced to 32K , instead of the default of 64K. This might be specified if there is a need to make a copy of the output tape, for example, to send offsite. LIOCS programs such as TSIDDTD can process a maximum blocksize of 64K. Note that special handling must be done in order to RESTORE or ONLOAD from a copied image of a TSIDOFL output tape. See the following note for more details.

UPSI xxxx0 specifies that the output tape blocksize is 64K.

Important! *If UPSI x1 is used with an execution of TSIDOFL where the TSIDOFL control statements were not created by the TSIDOFD program, it becomes the user's responsibility to assure that, when multiple versions of the dataset are restored using TSIDONL, the oldest version is restored first, the second oldest version is restored second, and so on.*

1. *Because of the danger of resequencing generations, it is suggested that UPSI X1 -NOT- be specified to bypass statement sorting. Instead, use UPSI x011xxxx to allow sorting with a larger table.*
2. *If the output tape from TSIDOFL is subsequently copied, for example with TSIDDTD, special handling is required in order to restore from the COPIED IMAGE. The copy must be marked in the EPIC catalog as an offload tape. To accomplish this, use the TSIDMNT 'RETAIN' command with the 'OF2' parameter. For example:*

```
// EXEC TSIDMNT
   RET 'COPID.OFFLOAD.TAPE',VER=1,OF2
/*
```

Dataset Requirements

OUTPUT specifies a tape dataset to control the offload tape. *This dataset must be cataloged*. If a label is not supplied for the OUTPUT dataset, TSIDOFL catalogs and uses the dataset name OFFLOAD.DATA.SET.

IJSYSIN specifies that the control statements reside on a tape or disk dataset that was created by the TSIDOFD program. You can use either standard VSE assignments or GDI.

Control Statements

Use control statements to indicate which datasets to offload. These can be generated with TSIDOFD.

Syntax

```
'dataset-name',version-number
```

Parameters

Parameter	Required?	Valid	Default
dataset-name	Yes	1 to 44 characters	None
version-number	No	1 to 9999	1

dataset-name specifies the name of the dataset to be offloaded.

version specifies the version to be offloaded.

Restart Information

TSIDOFL copies each dataset that is to be backed up or offloaded to the output tape. In addition, the OFFLOAD function deletes the VTOC entry for each dataset as soon as the copy function has completed. If for any reason a backup or offload job that involves more than one dataset terminates abnormally and must be restarted, please note the following special restart considerations.

Restarting Backups

Backups must be restarted from the beginning because the control statements are sorted internally by TSIDOFL.

1. Scratch the incomplete backup dataset version.
2. Restart the job in the TSIDOFL step.

Restarting Offloads

The same dataset cannot be offloaded twice, so TSIDOFL automatically restarts with the next dataset to be offloaded. The datasets already offloaded are in the incomplete offload dataset version.

1. Do not scratch the incomplete offload dataset version.
2. Restart the job in the TSIDOFL step. Disregard any error messages for missing datasets. (These were probably offloaded by the run which did not complete.)
3. Keep the TSIDOFL offload report and tapes from both the incomplete and restarted runs.
4. If the offloaded datasets are subsequently onloaded, run TSIDONL with the offload dataset versions from both the incomplete and restarted runs.

Examples

The following job offloads datasets based on the control statements supplied using the dataset SYSIN.DATASET. This input dataset is deleted when it is closed. Because no TLBL or DD for OUTPUT was supplied, all datasets are offload to a tape that is retained under the output dataset name OFFLOAD.DATA.SET.

```
// JOB OFFLOAD
* //IJSYSIN DD DSN=SYSIN.DATASET,DISP=(OLD,DELETE)
// UPSI 0
// EXEC TSIDOFL
/*
/&
```

The following job backs up datasets based on the control statements supplied via the dataset SYSIN.DATASET. This input dataset is deleted when it is closed. Because no TLBL or DD for OUTPUT was supplied, all datasets are backed up to a tape that is retained under the output dataset name BACKUP.DATASET.

```
// JOB  OFFLOAD
* //IJSYSIN DD DSN=SYSIN.DATASET,DISP=(OLD,DELETE)
// TLBL OUTPUT,'BACKUP.DATASET'
// UPSI 1
// EXEC TSIDOFL
/*
/&
```

The following job offloads four datsets named on the control statements supplied as SYSIPT data. All datasets are offloaded to a tape retained under the output dataset name "USER.OFFLOAD.DATA".

```
// JOB OFFLOAD
// TLBL OUTPUT,'USER.OFFLOAD.DATA'
// UPSI 0
// EXEC TSIDOFL
'PAYROLL.CHECKS',2
'BILLING.MASTER',9
'BILLING.MASTER',10
'RECON.TRANS'
/*
/&
```

Onloading and Restoring Datasets (TSIDONL)

TSIDONL recovers offloaded and backed up datasets back to disk. It can also be used to produce a report listing the datasets contained on the offload tape. TSIDONL functions are selected using UPSI values.

You can onload a dataset from an offload tape only. TSIDONL assigns the dataset a current relative version based upon its absolute generation number. You can restore a dataset from either an offload or a backup tape. The restore function assigns the dataset version number 1.

Configuration Options

The Start Track 1 feature is required to automatically access a dataset that has been offloaded. Specify a starting track or block address for this feature using the STRTRK option.

JCL Requirements

```
// UPSI xxxx
[// TLBL INPUT,'user-dataset-name']
// EXEC TSIDONL
[control statements]
/*
```

UPSI Values

Value	Function
000	Onloads all datasets from the offload tape. No control statements are read.
010	Onloads datasets selected by control statements only.
100	Restores all datasets from the offload or backup tape. No control statements are read.
110	Restores datasets selected by control statements only.
xx1	Creates the report only. No datasets are onloaded or restored. No control statements are read.
xxx1	Assumes uncataloged input tapes are in new offload format.
xxxx11	Causes the allocation of a larger internal work table if more than 2285 input files are to be processed.

UPSI xxx1

*UPSI xxx1 should *only* be used when the tape volume(s) accessed are not cataloged *AND* they were created using the new FAVER2 offload engine (CA EPIC for z/VSE 4.1.5 and higher). The following events could cause this situation:

- The tape volume was PURGEd from the catalog
- The tape volume was created at a different site and is not known to the catalog at the site where it is being restored

In all cases, the disk dataset to which a tape is being restored must be defined to the CA EPIC for z/VSE catalog. Instead of using UPSI xxx1 for an uncataloged tape, we recommend that you ADD the tape volume to the CA EPIC for z/VSE catalog and then RETAIN it using the OF2 parameter.

UPSI xxxx11 Causes the allocation of a larger internal work table if more than 2285 input files are to be processed. The larger table is allocated from 24 bit partition GETVIS.

The two UPSI switches may be specified in various combinations to tune the amount of storage acquired:

UPSI Switch	Additional Partition Getvis Required	Files Supported
xxxx00xx	0 K bytes	2,285 (default)
xxxx10xx	512 K bytes	7,313
xxxx01xx	1024K bytes	14,627
xxxx11xx	1536 K bytes	21,941

Note that TSIDOFL also has UPSI settings to specify a larger internal work table, but in that program different switches are used.

Dataset Requirements

INPUT specifies the offload dataset name used when TSIDOFL was executed. If no TLBL or DD statement is supplied, TSIDONL uses version 1 of the dataset 'OFFLOAD.DATASET'. No statements are required for the output datasets.

TSIDONL Control Statements

Control statements are read if UPSI 010 or 110 is specified. There are two types of control statements for TSIDONL:

- A *dataset-name* control statement specifies a dataset to be onloaded or restored.
- A START control statement begins an onload or restore starting at a particular dataset on the tape.

If no control statements are supplied when UPSI x10 is specified, all datasets on the tape are written to disk.

Syntax

```
'dataset-name',generation-number
```

Parameters

Parameter	Required?	Valid	Default
dataset-name	Yes	1 to 44 characters	None
generation-number	see below	1 to 999999	None

dataset-name specifies the name of the dataset to be onloaded or restored.

generation-number specifies the absolute generation number of the tape dataset to be onloaded or restored. To specify all generations of a dataset, code an asterisk (*). The generation number can be omitted for a work dataset.

Restart Information

Each backed up or offloaded dataset is processed as a separate entity. Therefore, if for any reason a multi-dataset restore or onload job terminates abnormally and is restarted, special restart considerations are necessary. In order to avoid dataset duplication and maintain version order integrity, the job must be restarted on the dataset being processed at the time the abend occurred. Use the following procedure.

1. Refer to the restore/onload report and obtain the last dataset name and generation listed.
2. Compare this to the backup/offload report and obtain the next dataset name and generation back-up or offload.
3. Restart TSIDONL using a START control statement specifying the dataset name and generation from Step 2.

Syntax

START='*dataset-name*',*generation-number*

dataset-name specifies the name of the dataset with which to begin onload or restore processing.

generation-number specifies the absolute generation number of the disk dataset. The generation number can be omitted for a work dataset.

Examples

The following job onloads all the datasets on the tape. It uses version 1 of the default offload dataset. The datasets remain assigned to their current relative version number (based upon absolute generation number) unless that version has been scratched. If the version has been scratched, they are assigned to version number 1.

```
// JOB ONLOAD
// EXEC TSIDONL
/*
/&
```

The following job restores all generations of the ACCOUNTS.PAY dataset. Each generation is assigned version number 1 as it is restored. The last version restored is version number 1. Because datasets are copied in reverse version order, the last version restored is the newest version on the tape.

```
// JOB ONLOAD
// UPSI 110
// EXEC TSIDONL
'ACCOUNTS.PAY',*
/*
/&
```

The following job onloads all datasets starting with and written after generation 112 of the ACCOUNTS.PAY dataset. The datasets remain assigned to their current relative version number unless that version has been scratched. If the version has been scratched, they are assigned to version number 1. The job uses version 1 of the USER.DATA.SET offload tape.

```
// JOB ONLOAD
// TLBL INPUT,'USER.DATA.SET'
// UPSI 010
// EXEC TSIDONL
START='ACCOUNTS.PAY',112
/*
/&
```

Differences Between TSIDONL Onload and TSIDONL Restore

Suppose we have the following versions of a dataset called DATASET.A:

Version number	Absolute generation number (AGN)
1	46
2	45
3	44
4	43
5	42
6	41

Absolute generation numbers (AGN) run in inverse sequence to the version numbers within a dataset. At one time (when it was first created), version 2, AGN=45 was version 1. The 45 means that this was the 45th time this dataset was created. The next time a version was created, it was the 46th, hence the AGN=46. The newly created version became version 1, the most current version.

Suppose versions 3 through 6 were offloaded to tape. Now DATASET.A looks as follows:

Version number	Absolute generation number (AGN)
1	46
2	45

Version information still exists in DATASET.A for versions 3 through 6, but now those versions display as tape datasets instead of disk. Versions 3 through 6 are still available for scratching (assuming all retention criteria has been met) even though they have been offloaded.

Suppose versions 3 through 6 are not scratched and two new versions are created. Now DATASET.A looks as follows:

Version number	Absolute generation number (AGN)
1	48
2	47
3	46
4	45

Suppose now we onload our previously offloaded datasets. After doing this, DATASET.A would look as follows:

Version number	Absolute generation number (AGN)
1	48
2	47
3	46
4	45
5 (old ver. 3)	52 (AGN is used to find where the onloaded datasets belong)
6 (old ver. 4)	51
7 (old ver.5)	50
8 (old ver.6)	49

The offloaded datasets were onloaded according to their AGNs, so they are no longer version 3 through 6, but versions 5 through 8.

Now suppose we restore our previously offloaded datasets. After doing this, DATASET.A would look as follows:

'DATASET.A'

Version number	Absolute generation number (AGN)
1 (old ver.3)	52 (old AGN=44)
2 (old ver.4)	51 (old AGN=43)
3 (old ver.5)	50 (old AGN=42)
4 (old ver.6)	49 (old AGN=41)
5	48
6	47
7	46
8	45

Each restored version is restored as version 1. AGN=41 is restored first, AGN=42 second, and so on. As each dataset is restored, it receives a new AGN.

Appendix A: CA EPIC for z/VSE Report Fields and Examples

This section contains the following topics:

Introduction

Standard reports are listed by program name. Programs are presented in alphabetical order.

Customized Reports

In addition to the standard reports, you can use your own report writers to produce customized reports from CA EPIC for z/VSE data.

The DSN Catalog contains all tape and dataset information, including CA EPIC for z/VSE dataset definitions (EDDs) and version data. We have provided the TSIDAC2 subroutine for access to the DSN Catalog. See Chapter 4 for procedures.

The Recorder File contains all DSN Catalog changes, job accounting information, and user comments. You can use the Recorder to report data such as catalog use and machine use. Use the Recorder copybook TSIDREC.C to generate reports from this the Recorder File.

TOJA Report Examples

TOJA produces job accounting information online. For information on producing TOJA reports, see "Online Access to the Job Accounting Database" in Chapter 11.

The report has three distinct levels: the Job Overview panel, the Program Overview panel, and the Program Audit panel. Each displays job accounting statistics in increasing detail.

Job Overview Panel

The Job Overview panel presents an overview of execution based on a jobname. For each job, there is a one line entry for each program executed in that job.

```
                    EPIC/VSE ONLINE JOB ACCOUNTING FACILITY
.ALL                                                                    PAGE 01
JOB NAME   PHASE NAME  START DATE  START TIME   DURATION  CC  PT  AR  CPU ID
BGINIT      TSIDRFS     05/24/10    14:13:46    00:00:00  10  BG 000  61008A
            TSIVON      05/24/10    14:13:47    00:00:01  10     000
            TSIDDEB     05/24/10    14:13:48    00:00:00  10     000
            TSIDDEB     05/24/10    14:13:48    00:00:17  10     000
            DTRIATTN    05/24/10    14:14:05    00:00:01  10     000
            LIBR        05/24/10    14:14:06    00:00:01  10     000
            LIBR        05/24/10    14:14:08    00:01:53  10     000
BGINIT      TSIDRFS     05/25/10    08:04:47    00:00:00  10  BG 000  61008A
            TSIVON      05/25/10    08:04:47    00:00:01  10     000
            TSIDDEB     05/25/10    08:04:48    00:00:00  10     000
            TSIDDEB     05/25/10    08:04:49    00:00:14  10     000
            DTRIATTN    05/25/10    08:05:04    00:00:00  10     000
            LIBR        05/25/10    08:05:05    00:00:00  10     000
            LIBR        05/25/10    08:05:05    00:00:29  10     000
BGINIT      TSIDRFS     07/30/10    08:25:06    00:00:00  10  BG 000  61008A
            TSIVON      07/30/10    08:25:07    00:00:02  10     000
            TSIDDEB     07/30/10    08:25:09    00:00:00  10     000
            TSIDDEB     07/30/10    08:25:10    00:00:15  10     000
            DTRIATTN    07/30/10    08:25:25    00:00:03  10     000
            LIBR        07/30/10    08:25:29    00:00:01  10     000
     PF1=FORWARD              PF3=DETAIL  PF4=END
```

Fields

JOB NAME specifies the name of the job that initiated the open and close activity. Different jobs are separated from each other by a dotted line.

PHASE NAME is the name of the executed phase.

START DATE is the date the program started.

START TIME is the time the program started.

DURATION is the length of time the program ran.

CC is the VSE cancel code, if the program abended.

PT is the partition the job ran in.

AR is the number of audit records for the phase. There is one audit record for each open and one for each close processed by the phase.

CPU ID is the CPU the job ran in.

Program Overview Panel

The Program Overview panel displays statistics at the program level.

```
                  EPIC/VSE ONLINE JOB ACCOUNTING FACILITY
.ALL                                                                  PAGE 01

 JOB NAME --------- BGINIT          PHASE NAME ---------- TSIDRFS
 CPUID  ----------- 61008A          PARTITION ----------- BG
 START DATE ------- 05/24/10        END DATE ------------ 05:24:10
 START TIME ------- 14:13:46        CPU TIME ------------ 00:00:00:1
 STOP TIME -------- 14:13:46        OVERHEAD ------------ 00:00:00:1
 DURATION --------- 00:00:00        ALLBOUND ------------ 00:00:00
 CANCEL CODE (HEX) ------ 10        PHASE CODE -- S (S=STEP L=LAST)
                                    PHASE SEQ --- 0001
          ********** INPUT OUTPUT TABLES **********
   01F      00E      183      209      229      241      247
 00000003 00000006 00000004 00000038 00000004 00000024 00000021
   294      505
 00000004 00000017


          ****************************************
  USER INFORMATION --------
  NUMBER OF AUDIT RECORDS FOR THIS PHASE -- 000
     USE PF3 TO VIEW AUDIT RECORDS

     PF1=FORWARD  PF2=BACKWARD  PF3=DETAIL  PF4=END
```

Fields

JOB NAME is the name of the job the program ran in.

CPUID is the CPU the program ran in.

START DATE is the date the program was started.

START TIME is the time the program was started.

STOP TIME is the time the program finished.

DURATION is the amount of time the program ran.

CANCEL CODE (HEX) is the VSE cancel code, if the program canceled.

PHASE NAME is the name of the program (EOJ reported as a phase).

PARTITION is the partition the program ran in.

END DATE is the date the program finished.

CPU TIME is the amount of CPU time the program used.

OVERHEAD is the program's overhead, as defined by IBM.

ALLBOUND is the program's allbound times, as defined by IBM.

PHASE CODE is STEP for a phase record; LAST for an EOJ record.

PHASE SEQ is the step number of the phase.

INPUT OUTPUT TABLES is the I/O counts for each device.

USER INFORMATION is user information from the JOB statement.

NUMBER OF AUDIT RECORDS FOR THIS PHASE is one audit record for each open and one for each close processed by the phase.

The Program Audit Panel

The Program Audit panel displays information about what datasets were processed by the program. If the program is TSIDMNT, it displays what maintenance was done to the DSN Catalog.

```
                    EPIC/VSE ONLINE JOB ACCOUNTING FACILITY
FILEID           REC CREATION OPEN      EXPIRY    SERIAL  REEL    END   DTF
PROCESSED       TYPE   TIME  CLOSE CUU   DATE    ACCEPTED TRACK  TRACK  TYPE
*******************************************************************************
MT  12:30:07  PUR 'JADB.TEST.DSN1'
MT  12:30:12  PUR 'JADB.TEST.DSN2'
MT  12:30:17  PUR 'JADB.TEST.DSN3'
MT  12:30:23  PUR 'JADB.TEST.DSN4'
MT  12:30:29  PUR 'JADB.TEST.DSN5'
MT  12:30:33  PUR 'JADB.TEST.DSN6'
MT  12:30:36  PUR 'JADB.TEST.DSN7'
MT  12:30:40  PUR 'JADB.TEST.DSN8'
```

TSIDARP Report

TSIDARP produces the Activity Log Report listing open and close activity by job name. Job data is listed in sections with information about the job and program information listed in the first line of the section. Dataset activity for that job is listed under the job name.

Fields

JOB NAME specifies the name of the job which initiated the open and close activity. Different jobs are separated from each other by a dotted line.

PHASE NAME is the name of the program being reported.

Under each job and phase name, the Activity Log Report lists the datasets processed by each phase. It indicates what kind of dataset was processed, and whether the dataset was opened or closed.

PART is the partition the job ran in.

CPU TIME is the amount of CPU time the program used.

OVERHEAD is the program overhead, as defined by IBM.

GEN/REEL is the generation number of a disk version or the reel sequence of a tape version.

SERIAL NO is the serial number of the tape or disk pack accessed.

DTF TYPE is the dtf type of the dataset processed.

UNIT is the address of the drive used.

TOTAL I/O is the amount of I/O issued by the program to that device.

START/OPEN is the time (in HH.MM.SS format) the dataset was opened.

END/CLOSE is the time (in HH.MM.SS format) the dataset was closed.

USAGE MIN/SEC is the amount of time the dataset was processed.

D/S CREATION: JOB is the dataset's creation job.

D/S CREATION: DATE is the dataset's creation date.

DISK SPACE ALLOCAT: START is the beginning address of a disk extent.

DISK SPACE ALLOCAT: REQST is the amount of allocation requested for the extent.

DISK SPACE ALLOCAT: TRUNC is the extent size after truncation.

Activity Log Report Example

```
                                          J O B   A C C O U N T I N G   S Y S T E M                  TSIDARP= 5.01.00
 C.P.U. NO. 500082   DATE 02/24/09            A C T I V I T Y   L O G      RUN DATE 03/09/09    TIME 10.04.51 PAGE    1
  JOB     PHASE        C.P.U.    OVER  GEN/  SERIAL  DTF        TOTAL   START/    END/   USAGE  -D/S CREATION----DISK SPACE ALLOCAT-
 NAME     NAME   PART    TIME     HEAD  REEL    NO.  TYPE UNIT   I/O  OPEN    CLOSE  MIN/SEC   JOB    DATE   START  REQST  TRUNC

RECOVER1 TSIDRFS  BG     00.1     00.1                              20.31.55 20.31.58     .03
RECOVER1 TSIDMNT  BG MAINT REC AT 20.31.59    PUR 'REBLD.IJSYS01'
RECOVER1 TSIDMNT  BG MAINT REC AT 20.32.00    PUR 'REBLD.IJSYS02'
RECOVER1 TSIDMNT  BG MAINT REC AT 20.32.00    PUR 'REBLD.IJSYS03'
RECOVER1 TSIDMNT  BG MAINT REC AT 20.32.01    PUR 'REBLD.IJSYSLN'
RECOVER1 TSIDMNT  BG MAINT REC AT 20.32.01    PUR 'REBLD.IJSYS04'
RECOVER1 TSIDMNT  BG MAINT REC AT 20.32.02    CAT 'REBLD.IJSYS01',WRK=YES,EXT=100,POL=POOL50,SYS001
RECOVER1 TSIDMNT  BG MAINT REC AT 20.32.02    CAT 'REBLD.IJSYS02',WRK=YES,EXT=100,POL=POOL50,SYS002
RECOVER1 TSIDMNT  BG MAINT REC AT 20.32.02    CAT 'REBLD.IJSYS03',WRK=YES,EXT=100,POL=POOL50,SYS003
RECOVER1 TSIDMNT  BG MAINT REC AT 20.32.03    CAT 'REBLD.IJSYSLN',WRK=YES,EXT=100,POL=POOL50
RECOVER1 TSIDMNT  BG MAINT REC AT 20.32.03    CAT 'REBLD.IJSYS04',WRK=YES,EXT=100,POL=POOL50,SYS004
RECOVER1 TSIDMNT  BG     00.1     00.1                              20.31.59 20.32.05     .06
RECOVER1 FCOBOL   BG     00.3     00.1                              20.32.05 20.32.45     .40
         DISK OUTPUT  REBLD.IJSYS01
                                       2  EPC181  20   181         20.32.14           RECOVER1 93055   1535    100    100
         DISK OUTPUT  REBLD.IJSYS02
 * INDICATES THAT TOTAL I/O IS FOR MULTIPLE  FILES IN SAME DEVICE

                                          J O B   A C C O U N T I N G   S Y S T E M                  TSIDARP= 5.01.00
 C.P.U. NO. 500082   DATE 02/24/09            A C T I V I T Y   L O G      RUN DATE 03/09/09    TIME 10.04.51 PAGE    2
  JOB     PHASE        C.P.U.    OVER  GEN/  SERIAL  DTF        TOTAL   START/    END/   USAGE  -D/S CREATION----DISK SPACE ALLOCAT-
 NAME     NAME   PART    TIME     HEAD  REEL    NO.  TYPE UNIT   I/O  OPEN    CLOSE  MIN/SEC   JOB    DATE   START  REQST  TRUNC

                                       2  SYSWK0  20   121         20.32.20           RECOVER1 93055    103    100    100
         DISK OUTPUT  REBLD.IJSYS03
                                       2  SYSWK0  20   121         20.32.24           RECOVER1 93055   1120    100    100
         DISK OUTPUT  REBLD.IJSYS04
                                       2  EPC181  20   181         20.32.29           RECOVER1 93055    433    100    100
    CLOSED AS INPUT   REBLD.IJSYS02
                                       2  SYSWK0  20   121                 20.32.41   RECOVER1 93055    103    100    100
    CLOSED AS INPUT   REBLD.IJSYS01
                                       2  EPC181  20   181                 20.32.41   RECOVER1 93055   1535    100    100
    CLOSED AS INPUT   REBLD.IJSYS04
                                       2  EPC181  20   181                 20.32.43   RECOVER1 93055    433    100    100
    CLOSED AS INPUT   REBLD.IJSYS03
                                       2  SYSWK0  20   121                 20.32.44   RECOVER1 93055   1120    100    100

RECOVER1 TSIDMNT  BG MAINT REC AT 20.32.46    CAT 'REBLD.A' CYC=3 POL=POOL50,EXT=100,COM='SINGLE EXTENT DISK'
RECOVER1 TSIDMNT  BG     00.1     00.1                              20.32.46 20.32.50     .04
RECOVER1 KJUTIL   BG     00.2     00.1                              20.32.50 20.34.08    1.18
         DISK OUTPUT  REBLD.A                        G=0000002
                                       2  SYSWK0  20   121         20.32.53           RECOVER1 93055    103    100    100
    CLOSED AS OUTPUT  REBLD.A                        G=0000002
                                       2  SYSWK0  20   121                 20.32.55   RECOVER1 93055    103    100      2
         DISK OUTPUT  REBLD.A                        G=0000003
                                       3  SYSWK0  20   121         20.32.58           RECOVER1 93055    105    100    100
    CLOSED AS OUTPUT  REBLD.A                        G=0000003
                                       3  SYSWK0  20   121                 20.33.01   RECOVER1 93055    105    100      2
         DISK OUTPUT  REBLD.A                        G=0000004
                                       4  EPC181  20   181         20.33.05           RECOVER1 93055   1535    100    100
    CLOSED AS OUTPUT  REBLD.A                        G=0000004
                                       4  EPC181  20   181                 20.33.07   RECOVER1 93055   1535    100      2

RECOVER1 TSIDMNT  BG MAINT REC AT 20.34.09    CAT 'REBLD.B' CYC=3 POL=POOL50,EXT=4,COM='MULTI EXTENT DISK'
RECOVER1 TSIDMNT  BG MAINT REC AT 20.34.09    CAT 'REBLD.B1' CYC=0,UID=KJ,RET=0,POL=POOL50,EXT=4,COM='MULTI EXT'
RECOVER1 TSIDMNT  BG     00.1     00.1                              20.34.08 20.34.10     .02
 * INDICATES THAT TOTAL I/O IS FOR MULTIPLE  FILES IN SAME DEVICE
```

TSIDBRP Reports

TSIDBRP creates the DSN/JOB Cross-Reference Report (a cross-reference to the Activity Log Report) listing open and close activity sequenced by dataset name. TSIDBRP also creates the Accepted Tapes Report which lists each time ACCEPT was used as a response to message EP008 or EP009.

Cross-Reference Fields

Each dataset is reported with a separate line for each open and each close. The last column, which has no header, specifies whether the line refers to open or close activity and whether the file was used as input or output. "OPEN OUTPUT|INPUT NEXT REEL|EXTENT" indicates that an open was initiated for the next reel or extent in the sequence.

FILE NAME is the name of the dataset opened or closed.

JOB NAME is the name of the job which initiated the activity.

PHASE NAME is the name of the program being reported.

CPUID is the CPU the program ran in.

RUN DATE is the date the job ran.

TIME is the time the job started.

PART is the partition the job ran in.

SERIAL NO. is the volume serial number which was opened or closed.

DEVICE: TYPE and NO. is the type of device used and the device address.

REEL NO. is the reel or extent sequence number.

START TRACK is the starting address of disk extents.

NO. OF TRACKS is the number of tracks requested for the allocation.

TRACKS TRUNCT. is the number of tracks after truncation.

Cross-Reference Report Example

```
                                        J O B   A C C O U N T I N G   S Y S T E M                  TSIDBRP= 5.01.00
                                    DSN / JOB CROSS REFERENCE      RUN DATE 03/09/09    TIME 10.13.57     PAGE    1
                  JOB      PHASE   CPUID    RUN                SERIAL   -DEVICE-   REEL  START   NO OF   TRACKS
FILE NAME         NAME     NAME             DATE    TIME  PART   NO.   TYPE  NO.    NO.  TRACK   TRACKS  TRUNCT.
BG8.KJ.DSNBU      RECOVER1 TSIDUTL  500082 02/24/09 22.28.38 BG  QCTT03  TAPE  341     1                        OPENED OUTPUT
                                    500082          22.28.48 BG  QCTT03  TAPE  341     1                        CLOSED OUTPUT

OFFLOAD.FOR.REBLD RECOVER1 TSIDOFL  500082 02/24/09 22.02.54 BG  QCTT07  TAPE  344     1                        OPENED OUTPUT
                                    500082          22.03.03 BG  QCTT07  TAPE  344     1                        CLOSED OUTPUT
                           TSIDONL  500082          22.14.10 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.14.16 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.14.18 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.14.22 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.14.23 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.14.27 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.14.28 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.14.31 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.14.32 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.14.36 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.14.37 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.14.40 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.14.41 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.14.45 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.14.46 BG  QCTT07  TAPE  344     1                        OPENED INPUT
OFFLOAD.FOR.REBLD RECOVER1 TSIDONL  500082 02/24/09 22.14.50 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.14.51 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.14.55 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.14.55 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.14.59 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.15.00 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.15.04 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.15.05 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.15.09 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.15.10 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.15.13 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.15.14 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.15.18 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.15.19 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.15.22 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.15.23 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.15.27 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.15.28 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.15.32 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.15.32 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.15.36 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.15.37 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.15.41 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.15.42 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.15.45 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.15.46 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.15.50 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.15.51 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.15.54 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.15.55 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.16.02 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.16.03 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.16.06 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.16.07 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.16.11 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.16.12 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.16.17 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.16.18 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.16.35 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.16.43 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.17.48 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.17.57 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.18.00 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.18.21 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.18.23 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.18.42 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
                                    500082          22.18.44 BG  QCTT07  TAPE  344     1                        OPENED INPUT
                                    500082          22.19.03 BG  QCTT07  TAPE  344     1                        CLOSED INPUT
```

Accepted Tapes Report Fields

REQUESTED FILE NAME is the name of the dataset requested by the job.

REQUESTED SERIAL NO. is the volume serial number requested by the job.

ACCEPTED FILE NAME is the name of the dataset accepted.

ACCEPTED SERIAL NO. is the volume serial number accepted.

JOB NAME is the job that was running when the operator accepted the tape.

PHASE NAME is the phase that was running when the operator accepted the tape.

DATE is the date the tape was accepted.

TIME is the time the tape was accepted.

PART is the partition the job was running in.

Accepted Tapes Report Example

```
                                          J O B   A C C O U N T I N G   S Y S T E M                 TSIDBRP= 5.01.00
                                         ACCEPTED TAPES REPORT        RUN DATE 03/09/09    TIME 10.14.00     PAGE    1
---------REQUESTED--------  ----------ACCEPTED--------
                  SERIAL                      SERIAL     JOB       PHASE
FILE NAME           NO.    FILE NAME           NO.     NAME      NAME     DATE      TIME  PART

REBLD.FILE        MOOGA1   NEW.REBLD.FILE     MOOGA2    REBUILD   TSIDBLD  03/18/09   08.15  BG
```

TSIDDIS Report

TISDDIS produces a summary report correlating dataset versions and tape volume serial numbers. It is listed in dataset name order. There is one entry for each dataset version. If a version spans more than one volume, there is an entry for each volume.

Note: For information on creating TSIDDIS reports, see Dataset to Tape Report (TSIDDIS (see page 366)).

Fields

DATASET NAME indicates the dataset name. Versions and reels listed between this name and the next belong to this dataset name.

VER is the relative version number of the version.

REEL is the reel sequence number associated with the tape (important for multi-volume versions).

SER NO is the volume serial number of the tape associated with the version.

TSIDDIS Report Example

```
 TSIDDIS= 5.01.00                    D A T A   S E T   M A N A G E M E N T   S Y S T E M                    PAGE    2
TIME 12 24 08                                                                                               DATE 03/09/09

DATA SET NAME                          VER REEL SER NO      DATA SET NAME                          VER REEL SER NO
REBLD.D                                      4 QCTT14
REBLD.E                                1     1 QCTT22
                                             2 QCTT13
                                       2     1 QCTT09
                                       3     1 QCTT12
REBLD.F                                1     1 QCTT19
REBLD.F1                               *** EMPTY ***
REBLD.F2                               *** EMPTY ***
REBLD.G                                1     1 QCTT20
REBLD.GR                               1     1 QCTT25
REBLD.GR1                              1     1 QCTT25
REBLD.GR2                              1     1 QCTT25
REBLD.G1                               1     1 QCTT20
REBLD.G2                               1     1 QCTT20
REBLD.H                                1     1 QCTT06
                                       2     1 QCTT06
REBLD.H1                               *** EMPTY ***
REBLD.H2                               *** EMPTY ***
REBLD.H3                               *** EMPTY ***
REBLD.H4                               *** EMPTY ***
REBLD.IJSYSLN                          *** EMPTY ***
REBLD.IJSYS01                          *** EMPTY ***
REBLD.IJSYS02                          *** EMPTY ***
REBLD.IJSYS03                          *** EMPTY ***
REBLD.IJSYS04                          *** EMPTY ***
RECORDER.XREF                          *** EMPTY ***
RECORDER.XREF.WORK                     *** EMPTY ***
RONNS.TAPE.TEST                        *** EMPTY ***
RONNS.VLT1.TEST                        1     1 VLT101
                                       2     1 VLT102
                                       3     1 VLT103
                                       4     1 VLT104
                                       5     1 VLT105
RONNS.VLT2.TEST                        1     1 VLT201
                                       2     1 VLT202
                                       3     1 VLT203
                                       4     1 VLT204
                                       5     1 VLT205
RONNS.VLT3.TEST                        1     1 VLT301
                                       2     1 VLT302
                                       3     1 VLT303
                                       4     1 VLT304
                                       5     1 VLT305
RONNS.VLT4.TEST                        1     1 VLT401
                                       2     1 VLT402
                                       3     1 VLT403
                                       4     1 VLT404
                                       5     1 VLT405
```

TSIDJAC Report

TSIDJAC creates the Job Time Activity Report, which lists job accounting information by CPU and by job.

Fields

CPU NO. is the CPU being reported.

JOB NAME is the name of the job which initiated the activity.

RUN DATE is the date the job ran.

PHASE NAME is the name of the program being reported.

PART ID is the partition the job ran in.

START TIME is the time (in hh.mm.ss format) the phase started.

END TIME is the time the phase completed.

DURATION TIME is the length of time the phase ran.

CPU TIME is the amount of CPU time the program used.

OVERHEAD is the program overhead, as defined by IBM.

TOTAL I/O is the amount of I/O issued by the program to that device.

In addition, grand totals for DURATION TIME, CPU TIME, OVERHEAD, and TOTAL I/O are listed at the end of the report.

Job Time Activity Report Example

```
                                                  J O B   A C C O U N T I N G   S Y S T E M                 TSIDJAC= 5.01.00
C.P.U. NO. 500082      RUN DATE 03/09/09       *  J O B  T I M E  A C T I V I T Y  *         TIME 10.15.03          PAGE    1
  JOB              RUN          PHASE          PART      START          END       DURATION          CPU           OVER            TOTAL
  NAME             DATE         NAME            ID       TIME          TIME         TIME            TIME          HEAD              I/O

*NO*NAME      02/24/09                          F4      21.04.43      21.04.43     00.00.00        00.00.00     00.00.00
                                                                                   00.00.00*       00.00.00*    00.00.00*              *
                                                                                   00.00.00**      00.00.00**   00.00.00**            **

RECOVER1      02/24/09      TSIDRFS             BG      20.31.58      20.31.58     00.00.00        00.00.00     00.00.00
              02/24/09      TSIDMNT             BG      20.32.05      20.32.05     00.00.00        00.00.00     00.00.00
              02/24/09      FCOBOL              BG      20.32.45      20.32.45     00.00.00        00.00.00     00.00.00
              02/24/09      TSIDMNT             BG      20.32.50      20.32.50     00.00.00        00.00.00     00.00.00
              02/24/09      KJUTIL              BG      20.34.08      20.34.08     00.00.00        00.00.00     00.00.00
              02/24/09      TSIDMNT             BG      20.34.10      20.34.10     00.00.00        00.00.00     00.00.00
              02/24/09      KJUTIL              BG      20.37.13      20.37.13     00.00.00        00.00.00     00.00.00
              02/24/09      KJUTIL              BG      20.39.15      20.39.15     00.00.00        00.00.00     00.00.00
              02/24/09      TSIDMNT             BG      20.39.21      20.39.21     00.00.00        00.00.00     00.00.00
              02/24/09      KJUTIL              BG      21.06.40      21.06.40     00.00.00        00.00.00     00.00.00
              02/24/09      KJUTIL              BG      21.08.26      21.08.26     00.00.00        00.00.00     00.00.00
              02/24/09      KJUTIL              BG      21.12.04      21.12.04     00.00.00        00.00.01     00.00.00
              02/24/09      KJUTIL              BG      21.14.55      21.14.55     00.00.00        00.00.00     00.00.00
              02/24/09      KJUTIL              BG      21.18.45      21.18.45     00.00.00        00.00.01     00.00.00
              02/24/09      TSIDMNT             BG      21.18.48      21.18.48     00.00.00        00.00.00     00.00.00
              02/24/09      KJUTIL              BG      21.49.33      21.49.33     00.00.00        00.00.00     00.00.00
              02/24/09      TSIDMNT             BG      21.49.35      21.49.35     00.00.00        00.00.00     00.00.00
              02/24/09      KJUTIL              BG      21.57.34      21.57.34     00.00.00        00.00.00     00.00.00
              02/24/09      TSIDMNT             BG      21.57.36      21.57.36     00.00.00        00.00.00     00.00.00
              02/24/09      KJUTIL              BG      22.00.17      22.00.17     00.00.00        00.00.00     00.00.00
              02/24/09      KJUTIL              BG      22.02.40      22.02.40     00.00.00        00.00.00     00.00.00
              02/24/09      TSIDOFL             BG      22.08.36      22.08.36     00.00.00        00.00.01     00.00.00
              02/24/09      TSIDMNT             BG      22.08.39      22.08.39     00.00.00        00.00.00     00.00.00
              02/24/09      KJUTIL              BG      22.09.17      22.09.17     00.00.00        00.00.00     00.00.00
              02/24/09      KJUTIL              BG      22.09.34      22.09.34     00.00.00        00.00.00     00.00.00
              02/24/09      TSIDMNT             BG      22.09.39      22.09.39     00.00.00        00.00.00     00.00.00
              02/24/09      KJUTIL              BG      22.11.08      22.11.08     00.00.00        00.00.00     00.00.00
              02/24/09      KJUTIL              BG      22.11.47      22.11.47     00.00.00        00.00.00     00.00.00
              02/24/09      KJUTIL              BG      22.12.06      22.12.06     00.00.00        00.00.00     00.00.00
              02/24/09      KJUTIL              BG      22.14.08      22.14.08     00.00.00        00.00.00     00.00.00
              02/24/09      TSIDONL             BG      22.16.43      22.16.43     00.00.00        00.00.01     00.00.00
              02/24/09      TSIDONL             BG      22.28.06      22.28.06     00.00.00        00.00.02     00.00.01
              02/24/09      TSIDDEB             BG      22.28.11      22.28.11     00.00.00        00.00.00     00.00.00
              02/24/09      TSIDUTL             BG      22.28.32      22.28.32     00.00.00        00.00.00     00.00.00
              02/24/09      TSIDUTL             BG      22.29.35      22.29.35     00.00.00        00.00.00     00.00.00
                                                                                   00.00.00*       00.00.06*    00.00.01*              *
                                                                                   00.00.00**      00.00.06**   00.00.01**            **

                                               *** G R A N D   T O T A L ***       00.00.00        00.00.06     00.00.01
```

TSIDLVT Report

TSIDLVT lists VTOC data in batch. The report format varies depending on the options you choose, and the following report only a sample of the reports you can create.

Note: For information on producing TSIDLVT reports, see Printing VTOC Entries (TSIDLVT) (see page 372).

Fields

DISPLAY OF identifies the content of the report, either SER=volser or the name of the pool.

FILE NAME is the name of the dataset residing in the specified pools of the VTOC.

SERIAL is the volume serial number the extent resides on.

EXT SEQ is the extent sequence number.

EXTENT START is the starting extent address.

EXTENT END is the ending extent address.

EXTENT NUMBER specifies the number of extents occupied by the dataset.

CREATE DATE is the date the dataset was created.

EXPIRE DATE is the expiration date written on the VTOC.

DISK TYPE is the type of disk device.

FILE TYPE is the dataset's file type.

EXT If printed in the far right column, the extent has expired.

TSIDLVT Report Example

```
*
* ----------------------------------------------------------+
*   POOL IN EXTENT SEQUENCE - UNEXPIRED ONLY                |
* ----------------------------------------------------------+
*
*-------------------------------------------------------------------------------------------------------------------------------*
*                                                                                                                               *
*                       REPORT FOR POOL=POOL01                                                                                  *
*                                                                                                                               *
*                       OPTIONS IN EFFECT - PRINT IN EXT SEQUENCE                                                               *
*                                           EXCLUDE EXPIRED EXTENTS                                                             *
*                                           IGNORE POOL LIMITS                                                                  *
*                                           EXCLUDE FREE SPACE                                                                  *
*                                           LIST IN RELATIVE TRACK/BLOCK FORMAT                                                 *
*                                           USE CONVENTIONAL DATE FORMAT MM/DD/YYYY                                             *
*                                           LIST UNEXPIRED EXTENTS                                                              *
*-------------------------------------------------------------------------------------------------------------------------------*
DATE 03/09/09          DISPLAY OF              E P I C      V T O C      R E P O R T             VOLSER EPC229  TSIDLVT 5.01.00
TIME 12 36 15          POOL01             P O O L   I N   E X T E N T   S E Q U E N C E         CUU     229           PAGE   1
                                          SERIAL  EXT   ---------- EXTENT ----------   CREATE      EXPIRE    DISK  FILE
            FILE NAME                             SEQ    START      END      NUMBER     DATE        DATE     TYPE  TYPE

---------- POOL LOWER LIMIT ----------       EPC229              1
EPC229.QDAS.SHARE.FILE                       EPC229    0         1         1         1  08/24/2009  12/32/1999  3380  SD
DEVTEPI2.POWER.QUEUE.FILE                    EPC229    0         2         6         5  11/23/2009  12/31/1999  3380  DA
EPC229.QDAS.REL4.SHARE.FILE                  EPC229    0         7         7         1  02/01/1995  12/32/1999  3380  SD
DOS.LABEL.FILE.FF5200859021.DEVTEPI2         EPC229    0        15        29        15  03/02/1995  12/31/1999  3380  UN
DOS.LABEL.FILE.FF5200829021.TECHEPIC         EPC229    0        30        44        15  11/12/2009  12/31/1999  3380  UN
TECHEPIC.ESA13.POWER.ACCOUNT.FILE            EPC229    0        45        69        25  11/11/2009  12/31/1999  3380  DA
XXX-7.BG.WORK.FILE                           EPC229    1        70        71         2  02/11/1995  12/31/1999  3380  SD
TECHEPIC.HARDCOPY.FILE                       EPC229    0        95       109        15  11/10/2009  12/32/1999  3380  UN
TECHEPIC.RECORDER.FILE                       EPC229    0       110       124        15  11/10/2009  12/32/1999  3380  UN
KJ.OFL.TST3                                  EPC229    1       125       126         2  02/18/1995  12/31/1999  3380  SD
BG7.KEITH.DISKA                              EPC229    1       133       134         2  03/01/1995  12/31/1999  3380  SD
BG7.KEITH.DISKB                              EPC229    1       135       137         3  03/01/1995  12/31/1999  3380  SD
DBK827.STEP4                                 EPC229    1       138       139         2  03/03/1995  12/31/1999  3380  SD
TSIDBLD.TEST.CATALOG                         EPC229    0       263       322        60  02/23/1995  12/31/1999  3380  UN
DBK.FILEOUT.STEP2                            EPC229    1       323       324         2  03/03/1995  12/31/1999  3380  SD
VSE.HARDCOPY.FILE                            EPC229    0       555       614        60  08/24/2009  12/31/1999  3380  UN
VSE.RECORDER.FILE                            EPC229    0       615       674        60  01/26/1995  12/31/1999  3380  UN
DEVTEPI2.POWER.DATA.FILE                     EPC229    0       675      1274       600  11/23/2009  12/31/1999  3380  DA
DEVTEPI2.POWER.ACCOUNT.FILE                  EPC229    0      1275      1364        90  11/23/2009  12/31/1999  3380  DA
Z9999994.VSAMDSPC.TA6935B4.T746489A          EPC229    0      1365      1389        25  11/10/2009  12/32/1999  3380  VSAM
Z9999992.VSAMDSPC.TA6935C0.T1D6583E          EPC229    0      1390      1889       500  11/10/2009  12/32/1999  3380  VSAM
DEVTEPI2.EPIC.TEST.LIBRARY                   EPC229    0      1890      2489       600  08/24/2009  12/32/1999  3380  SD
EPIC/VSE.CATALOG                             EPC229    0      2490      2519        30  02/01/1995  12/32/1999  3380  UN
EPIC/VSE.RECORDER                            EPC229    0      2520      2549        30  02/08/1995  12/32/1999  3380  UN
Z9999992.VSAMDSPC.TA6C2F7A.T0C0E0EF          EPC229    0      3000      3239       240  12/18/2009  12/32/1999  3380  VSAM
7BG.SORTWK1                                  EPC229    1      3240      3839       600  03/09/1995  12/31/1999  3380  SD
VTOC                                         EPC229    0      4485      4499        15                          3380  UN
---------- POOL UPPER LIMIT ----------       EPC229              4499
*-------------------------------------------------------------------------------------------------------------------------------*
*                                                                                                                               *
*                       ** VOLUME STATISTICS **                                                                                 *
*                                                                                                                               *
*                   TOTAL SPACE THIS VOLUME           -              4500                                                       *
*                   TOTAL USED SPACE THIS VOLUME      -              3016                                                       *
*                   TOTAL FREE SPACE THIS VOLUME      -              1484                                                       *
*                                                                                                                               *
*                       ** POOL STATISTICS **                                                                                   *
*                                                                                                                               *
*                   TOTAL POOL SPACE THIS VOLUME      -              4499                                                       *
*                   TOTAL USED POOL SPACE THIS VOLUME -              3015                                                       *
*                   TOTAL FREE POOL SPACE THIS VOLUME -              1484                                                       *
*                                                                                                                               *-
-------------------------------------------------------------------------------------------------------------------------------*
```

TSIDMNT Reports

TSIDMNT can produce a summary report or a detail report. To receive a summary report, request a display of all datasets or a group of datasets. To receive a detail report, request a specific dataset name or a specific volume serial number.

For information on creating TSIDMNT reports, see Chapter 3.

Summary Report

DATASET NAME is the name of the dataset.

SERIAL is the volume serial number(s) associated with the version.

VN is the version number.

RL is the reel or extent sequence number.

ST is the version's status. Valid status codes are:

- (blank)=active version
- C=conditional catalog status
- L=active version in a locked status
- M=MCAT version
- O=open status
- S=scratch status

BLK is the block size of the version.

REC is the record size of the version.

TYPE TAPE and DISK indicate the storage medium of the version.

EMPTY means there are no versions of this dataset.

Detail Report

The first three lines for each dataset report the dataset's defaults. Lower lines present the version level data for the dataset.

CA EPIC for z/VSE Dataset Definition (EDD)

The first field at the top left corner is the dataset name.

RET is the number of days each version is to be retained.

CYC is the number of versions (cycles) to be maintained.

VLT is the number of the vault method assigned to the dataset.

WORK identifies a work dataset.

password is the dataset's Unlock/Release password is displayed at the far right.

REBLK is the default block size.

POOL The disk pool name.

EXT=n is the first value specified is the size of the primary extent. The second value is the size of all secondary extents.

SEC is the maximum number of secondary extents for a version.

ALR is the number of records for allocation by logical records.

SID is the system ID.

UID is the user ID.

(UN)LBL indicates whether this is a standard label tape dataset.

MOD is the default tape density.

SYSnnn is the SYS number assigned to the dataset.

REC is the actual record size of the current version.

BLK is the actual block size of the current version

REV indicates whether this is a revolving tape dataset.

COM is the user comment.

TPL is the tape pool ID.

Version Information

SER-NO is the tape or disk volume serial number associated with this version.

VN is the version number.

RL is the reel or extent sequence number.

CREDT is the creation date in Julian format.

EXPDT is the projected expiration date.

CREAT/JB is the name of the creating job.

CUU is the drive address the version was created on.

ST is the version's status. The codes are:

- C=conditional catalog status
- L=active version in a locked status
- M=MCAT version
- O=open status
- S=scratch status

LAST/JOB is the last job that accessed this version.

LASTDT is the last date this version was accessed.

USE is the number of times the tape was opened since the last clean date.

ERR is the number of read/write errors encountered since the last clean date.

CLN-DAT is the last clean date of this tape.

VLT is the current vault location of this tape.

SL is the current slot location of this tape.

TSIDOFD Report

TSIDOFD produces a report listing the datasets to be offloaded or backed up. To produce this report, see Selecting Datasets To Be Offloaded (TSIDOFD) (see page 379).

Fields

DATASET NAME is the name of the dataset to offload or backup.

VER is the version number of the dataset to offload or backup.

ALL VER indicates all versions were selected.

UID indicates whether the dataset was selected by user ID.

SID indicates whether the dataset was selected by system ID.

VOLUME indicates whether the dataset was selected by volume.

POOL ID indicates whether the dataset was selected by pool name.

S/DLA indicates whether the dataset was selected by SLA or DLA.

NEW indicates that only new datasets were selected.

Dataset Selection Report Example

```
DATE 03/09/09      TIME   13.01.41         D A T A S E T   S E L E C T I O N   R E P O R T                          PAGE 01
TSIDOFD
    INCLUDE PARAMETERS                DATA SET NAME               / VER   ALLVER   UID   SID   VOLUME   POOLID   S/DLA   MISC
                                                                                                          POOL01

    OMIT PARAMETERS                   DATA SET NAME               / VER   ALLVER   UID   SID   VOLUME   POOLID

DATE 03/09/09      TIME   13.01.41         D A T A S E T   S E L E C T I O N   R E P O R T                          PAGE 02
TSIDOFD
                                  THE FOLLOWING DATA SETS HAVE BEEN SELECTED
                        DATA SET NAME                     VER                     DATA SET NAME                     VER

          BG7.KEITH.DISKA                                   1       BG7.KEITH.DISKB                                   1
          DBK.FILEOUT.STEP1                                 1       DBK.FILEOUT.STEP2                                 1
          DBK827.STEP4                                      1       KEITH.GEN                                         1
          KEITH.GEN                                         2       KEITH.GEN                                         3
          KEITH.GEN2                                        1       KEITH.GEN2                                        2
          KEITH.GEN3                                        1       KEITH.GEN3                                        2
          KEITH.GEN3                                        3       VAULT.IJSYSLN                                     1
          TOTAL NUMBER OF VERSIONS SELECTED    14
1S55I  LAST RETURN CODE WAS 0000
EOJ TSIDOFD   MAX.RETURN CODE=0000                                    DATE 03/09/09,CLOCK 13/02/05,DURATION   00/00/27
DATE 03/09/09,CLOCK 13/02/05,DURATION   00/00/27
```

TSIDOFL Report

TSIDOFL creates a report listing the datasets offloaded or backed up. To produce this report, see Offloading and Backing Up Datasets (TSIDOFL) (see page 383).

Fields

DATASET NAME is the name of the dataset offloaded or backed up.

VER is the dataset's version number.

SERIAL is the volume serial number the dataset came from.

CUU is the drive address the dataset came from.

EXTENTS START is the former extent starting address.

EXTENTS END is the former extent ending address.

EXTENTS SIZE is the size of the extent.

CREATE DATE is the date the dataset was created.

EXPIRE DATE is the date the dataset is scheduled to expire.

NUMBER RECORDS is the number of records in the dataset.

BLOCK SIZE is the block size of the dataset.

RECORD SIZE is the record size.

TYPE is the file type.

FILE SEQ is the dataset's file sequence on the offload tape.

Dataset Offload Report Example

```
DATE 03/09/09      TIME   16.20.32          D A T A S E T   O F F L O A D   R E P O R T                          PAGE 01
TSIDOFL

                                                          -------EXTENTS--------CREATE EXPIRE NUMBER    BLOCK  RECORD      FILE
               DATASET NAME                 VER SERIAL CUU   START     END    SIZE DATE   DATE   RECORDS    SIZE    SIZE TYPE  SEQ
KJ.VARIABLE                       G=0000002   2 VSDAT1 180    3032     3036     5 93068  93068   1200   32760     994   V     1
KJ.VARIABLE                       G=0000003   1 VSDAT1 180    3037     3042     6 93068  93068   1500   32760     994   V     2
KJ.UNDEF                          G=0000002   1 EPC551 551    5778     5781     4 93068  93068    100   32760   32760   U     3
KJ.SPAN                           G=0000002   2 EPC551 551    5775     5777     3 93068  93068    125   32760     500   S     4
KJ.SPAN                           G=0000003   1 VSDAT1 180     932      934     3 93068  93068    125   32760     500   S     5
KJ.REBLK1                         G=0000002   2 VSDAT1 180    3020     3022     3 93068  93068   1550   32760      40   F     6
KJ.REBLK1                         G=0000003   1 VSDAT1 180    3023     3026     4 93068  93068   2001   32760      40   F     7
KJ.REB.VAR                        G=0000002   2 VSDAT1 180    3043     3047     5 93068  93068   1100   32760   10000   V     8
KJ.REB.VAR                        G=0000003   1 VSDAT1 180    3048     3052     5 93068  93068   1575   32760   10000   V     9
KJ.REB.AFTER                      G=0000002   1 VSDAT1 180    3027     3031     5 93068  93068   1751   32760      40   F    10
KJ.OFL1                           G=0000002   2 VSDAT1 180    3011     3014     4 93068  93068   1501   32760      40   F    11
KJ.OFL1                           G=0000003   1 VSDAT1 180    3015     3019     5 93068  93068   2850   32760      40   F    12
            OFFLOAD DATA SET NAME = OFFLOAD.DATA.SET                                  VOLUME SERIAL = 00LREC
DATE 03/09/09      TIME   16.20.32          D A T A S E T   O F F L O A D   S U M M A R Y                          PAGE 02
TSIDOFL

                                    VOLUME          TRACKS RELEASED          TOTAL EXTENTS

                                    VSDAT1                    45                      10
                                    EPC551                     7                       2
                                    * GRAND TOTAL *           52                      12
```

TSIDONL Report

TSIDONL creates a report listing the datasets that were onloaded. To create this report, see "Onloading and Restoring Datasets."

Fields

DATASET NAME is the name of the dataset onloaded.

OLD VER is the version number of the dataset before it was onloaded.

OLD GEN is the generation number of the dataset before it was onloaded.

NEW VER is the version number of the dataset after it was onloaded.

NEW GEN is the generation number of the dataset after it was onloaded.

NUMBER RECORDS is the number of records in the dataset.

BLOCK SIZE is the block size.

RECORD SIZE is the record size.

RECFM is the record format.

INPUT DATASET NAME is the name of the offload dataset.

INPUT SERIAL is the volser of the offload tape the dataset was copied from.

FILE SEQ is the dataset's file sequence on the offload tape.

Dataset Onload Report Example

```
DATE 03/09/09      TIME   16.24.59            D A T A S E T   O N L O A D   R E P O R T                                PAGE 01
                                          ----OLD----  ----NEW----  NUMBER   BLOCK  RECORD        ----------INPUT---------  FILE
              DATA SET NAME               VER     GEN  VER   GEN  RECORDS   SIZE   SIZE  RECFM   DATA SET NAME   SERIAL  SEQ
KJ.VARIABLE                                 2       2    2     4     1200    994    994    V   OFFLOAD.DATA.SET  00LREC     1
KJ.VARIABLE                                 1       3    1     5     1500    994    994    V   OFFLOAD.DATA.SET  00LREC     2
KJ.UNDEF                                    1       2    1     3      100    800    800    U   OFFLOAD.DATA.SET  00LREC     3
KJ.SPAN                                     2       2    2     4      125    500    500    S   OFFLOAD.DATA.SET  00LREC     4
KJ.SPAN                                     1       3    1     5      125    500    500    S   OFFLOAD.DATA.SET  00LREC     5
KJ.REBLK1                                   2       2    2     4     1550  16000     40    F   OFFLOAD.DATA.SET  00LREC     6
KJ.REBLK1                                   1       3    1     5     2001  16000     40    F   OFFLOAD.DATA.SET  00LREC     7
KJ.REB.VAR                                  2       2    2     4     1100  10000  10000    V   OFFLOAD.DATA.SET  00LREC     8
KJ.REB.VAR                                  1       3    1     5     1575  10000  10000    V   OFFLOAD.DATA.SET  00LREC     9
KJ.REB.AFTER                                1       2    1     3     1751    800     40    F   OFFLOAD.DATA.SET  00LREC    10
KJ.OFL1                                     2       2    2     4     1501    800     40    F   OFFLOAD.DATA.SET  00LREC    11
KJ.OFL1                                     1       3    1     5     2850    800     40    F   OFFLOAD.DATA.SET  00LREC    12
```

TSIDPRV Report

TSIDPRV produces the Vaulting Method Report. The fields in this report are:

Fields

DATASET NAME is the name of the dataset.

NO. CYC is the number of cycles to be maintained for the dataset.

DAYS RET is the number of days each version is to be retained.

METH NO. is the vaulting method assigned to the dataset.

VER NO. and VAULT NAME represent the heart of the vaulting method report. They correlate each version number with its designated vault location.

TSIDSDS Report

TSIDSDS produces a summary report correlating dataset versions and tape volume serial numbers. It is listed in volume serial number order.

Note: For information on creating TSIDSDS reports, see Dataset to Tape Report (TSIDDIS) (see page 366).

Fields

There is at least one entry for each version of a dataset. If a version spans more than one volume, there is an entry for each volume.

SER NO is the volume serial number.

DATASET NAME is the name of the dataset the referenced volume contains.

TSIDSDS Report Example

```
 TSIDSDS= 5.01.00                    D A T A   S E T   M A N A G E M E N T   S Y S T E M                    PAGE    1
TIME 12 28 09                                                                                               DATE 03/09/09

SER NO  --DATA SET NAME--                    SER NO  --DATA SET NAME--
ACCPT1  BG5.KJ.HIST                          VLT403  RONNS.VLT4.TEST
EGB001S DAN901.FILEOUT                       VLT404  RONNS.VLT4.TEST
EGB002  DBK827.STEP2                         VLT405  RONNS.VLT4.TEST
EGB002  DBK827.STEP1                         VLT501  RONNS.VLT5.TEST
JA0001  JUAN.OLD                             VLT502  RONNS.VLT5.TEST
JA0002  JUAN.TRANS                           VLT503  RONNS.VLT5.TEST
JA0012  JUAN.TRANS                           VLT504  RONNS.VLT5.TEST
KLSA02S ** DSN DELETED **                    VLT505  RONNS.VLT5.TEST
KNOTS S BG5.KJ.DISKA                         0ASER1S A
MOOGA1S ** DSN DELETED **                    0OTST3S KJ.OFL.TST3
QCTT01  BG7.KJ.DSNBU                         0SCRA1S KEITH.VAULT4
QCTT02  BG7.KJ.DAILY                         0SCRA2S KEITH.VAULT4
QCTT03  BG7.KJ.DSNBU                         0SCRA3S KEITH.VAULT4
QCTT06  OFFLOAD.FOR.REBLD                    0SCRA4S KEITH.VAULT4
QCTT06  REBLD.H                              0SCRA5S KEITH.VAULT4
QCTT06  REBLD.H                              000X01S BG5.KJ.DISKA
QCTT07  BG5.KJ.SUB3                          990050  BG7.KJ.TPOL
QCTT07  BG5.KJ.SUB2
QCTT07  BG5.KJ.SUB1
QCTT08  REBLD.D
QCTT09  REBLD.E
QCTT10S REBLD.C
QCTT11  REBLD.C
QCTT12  REBLD.E
QCTT13  REBLD.E
QCTT14  REBLD.D
QCTT18  REBLD.D
QCTT19  REBLD.F
QCTT20  REBLD.G
QCTT20  REBLD.G2
QCTT20  REBLD.G1
QCTT21  REBLD.D
QCTT22  REBLD.E
QCTT23  REBLD.D
QCTT25  REBLD.GR
QCTT25  REBLD.GR2
QCTT25  REBLD.GR1
QCTT26  BG5.KJ.SUB2
QCTT26  BG5.KJ.SUB1
VLT101  RONNS.VLT1.TEST
VLT102  RONNS.VLT1.TEST
VLT103  RONNS.VLT1.TEST
VLT104  RONNS.VLT1.TEST
VLT105  RONNS.VLT1.TEST
VLT201  RONNS.VLT2.TEST
VLT202  RONNS.VLT2.TEST
VLT203  RONNS.VLT2.TEST
VLT204  RONNS.VLT2.TEST
VLT205  RONNS.VLT2.TEST
VLT301  RONNS.VLT3.TEST
VLT302  RONNS.VLT3.TEST
VLT303  RONNS.VLT3.TEST
```

TSIDTAC Report

TSIDTAC creates the Date Time Activity Report, which lists job accounting information by CPU and by date and time. It is a cross reference to the Job Time Activity report.

Fields

CPU NO. is the CPU being reported.

RUN DATE is the date the job ran.

JOB NAME is the name of the job which initiated the activity.

PHASE NAME is the name of the program being reported.

PART ID is the partition the job ran in.

START TIME is the time (in hh.mm.ss format) the phase started.

END TIME is the time the phase completed.

DURATION TIME is the length of time the phase ran.

IDLE TIME is the amount of time the CPU was idle while the phase ran. (Idle time uses standard IBM definitions.)

CPU TIME is the amount of CPU time the program used.

OVERHEAD is the program overhead, as defined by IBM.

TOTAL I/O is the amount of I/O issued by the program to that device.

In addition, grand totals for DURATION TIME, IDLE TIME, CPU TIME, OVERHEAD, and TOTAL I/O are listed at the end of the report.

Date Time Activity Report Example

```
                                        J O B   A C C O U N T I N G   S Y S T E M                TSIDTAC= 5.01.00
C.P.U. NO. 500082      RUN DATE 03/09/09      *  D A T E   T I M E   A C T I V I T Y  *        TIME 10.16.23        PAGE    1
  RUN          JOB      PHASE      PART     START       END       DURATION       IDLE        CPU          OVER         TOTAL
  DATE         NAME     NAME       ID       TIME        TIME        TIME         TIME        TIME         HEAD          I/O

02/24/09    RECOVER1    TSIDRFS     BG     20.31.55    20.31.58    00.00.03     .00.00.00   .00.00.00    .00.00.00
02/24/09    RECOVER1    TSIDMNT     BG     20.31.59    20.32.05    00.00.06     .00.00.01   .00.00.00    .00.00.00
02/24/09    RECOVER1    FCOBOL      BG     20.32.05    20.32.45    00.00.40     .00.00.00   .00.00.00    .00.00.00
02/24/09    RECOVER1    TSIDMNT     BG     20.32.46    20.32.50    00.00.04     .00.00.01   .00.00.00    .00.00.00
02/24/09    RECOVER1    KJUTIL      BG     20.32.50    20.34.08    00.01.18     .00.00.00   .00.00.00    .00.00.00
02/24/09    RECOVER1    TSIDMNT     BG     20.34.08    20.34.10    00.00.02     .00.00.00   .00.00.00    .00.00.00
02/24/09    RECOVER1    KJUTIL      BG     20.34.11    20.37.13    00.03.02     .00.00.01   .00.00.00    .00.00.00
02/24/09    RECOVER1    KJUTIL      BG     20.37.13    20.39.15    00.02.02     .00.00.00   .00.00.00    .00.00.00
02/24/09    RECOVER1    TSIDMNT     BG     20.39.16    20.39.21    00.00.05     .00.00.01   .00.00.00    .00.00.00
02/24/09    RECOVER1    KJUTIL      BG     20.39.21    21.06.40    00.27.19     .00.00.00   .00.00.00    .00.00.00
02/24/09    RECOVER1    KJUTIL      BG     21.06.41    21.08.26    00.01.45     .00.00.01   .00.00.00    .00.00.00
02/24/09    RECOVER1    KJUTIL      BG     21.08.27    21.12.04    00.03.37     .00.00.01   .00.00.01    .00.00.00
02/24/09    RECOVER1    KJUTIL      BG     21.12.04    21.14.55    00.02.51     .00.00.00   .00.00.00    .00.00.00
02/24/09    RECOVER1    KJUTIL      BG     21.14.55    21.18.45    00.03.50     .00.00.00   .00.00.01    .00.00.00
02/24/09    RECOVER1    TSIDMNT     BG     21.18.46    21.18.48    00.00.02     .00.00.01   .00.00.00    .00.00.00
02/24/09    RECOVER1    KJUTIL      BG     21.18.49    21.49.33    00.30.44     .00.00.01   .00.00.00    .00.00.00
02/24/09    RECOVER1    TSIDMNT     BG     21.49.33    21.49.35    00.00.02     .00.00.00   .00.00.00    .00.00.00
02/24/09    RECOVER1    KJUTIL      BG     21.49.35    21.57.34    00.07.59     .00.00.00   .00.00.00    .00.00.00
02/24/09    RECOVER1    TSIDMNT     BG     21.57.34    21.57.36    00.00.02     .00.00.00   .00.00.00    .00.00.00
02/24/09    RECOVER1    KJUTIL      BG     21.57.37    22.00.17    00.02.40     .00.00.01   .00.00.00    .00.00.00
02/24/09    RECOVER1    KJUTIL      BG     22.00.18    22.02.40    00.02.22     .00.00.01   .00.00.00    .00.00.00
02/24/09    RECOVER1    TSIDOFL     BG     22.02.40    22.08.36    00.05.56     .00.00.00   .00.00.01    .00.00.00
02/24/09    RECOVER1    TSIDMNT     BG     22.08.36    22.08.39    00.00.03     .00.00.00   .00.00.00    .00.00.00
02/24/09    RECOVER1    KJUTIL      BG     22.08.40    22.09.17    00.00.37     .00.00.01   .00.00.00    .00.00.00
02/24/09    RECOVER1    KJUTIL      BG     22.09.17    22.09.34    00.00.17     .00.00.00   .00.00.00    .00.00.00
02/24/09    RECOVER1    TSIDMNT     BG     22.09.35    22.09.39    00.00.04     .00.00.01   .00.00.00    .00.00.00
02/24/09    RECOVER1    KJUTIL      BG     22.09.39    22.11.08    00.01.29     .00.00.00   .00.00.00    .00.00.00
02/24/09    RECOVER1    KJUTIL      BG     22.11.09    22.11.47    00.00.38     .00.00.01   .00.00.00    .00.00.00
02/24/09    RECOVER1    KJUTIL      BG     22.11.47    22.12.06    00.00.19     .00.00.00   .00.00.00    .00.00.00
02/24/09    RECOVER1    KJUTIL      BG     22.12.06    22.14.08    00.02.02     .00.00.00   .00.00.00    .00.00.00
02/24/09    RECOVER1    TSIDONL     BG     22.14.08    22.16.43    00.02.35     .00.00.00   .00.00.01    .00.00.00
02/24/09    RECOVER1    TSIDONL     BG     22.16.44    22.28.06    00.11.22     .00.00.01   .00.00.02    .00.00.01
02/24/09    RECOVER1    TSIDDEB     BG     22.28.06    22.28.11    00.00.05     .00.00.00   .00.00.00    .00.00.00
02/24/09    RECOVER1    TSIDUTL     BG     22.28.11    22.28.32    00.00.21     .00.00.00   .00.00.00    .00.00.00
02/24/09    RECOVER1    TSIDUTL     BG     22.28.33    22.29.35    00.01.02     .00.00.01   .00.00.00    .00.00.00
                                                                   01.57.25*    .00.00.15*  .00.00.06*   .00.00.01*          *
02/24/09    *NO*NAME                F4     20.31.16    21.04.43    00.33.27     .00.00.00   .00.00.00    .00.00.00
                                                                   00.33.27*    .00.00.00*  .00.00.00*   .00.00.00*          *
                                                                   02.30.52**   .00.00.15** .00.00.06**  .00.00.01**        **
                          *** G R A N D   T O T A L ***             02.30.52     .00.00.15   .00.00.06    .00.00.01
```

TSIDUTL Reports

Display ALL, Dataset Name Sequence

The Display ALL, SEQ=DSN report gives you a detailed listing of all datasets in dataset name sequence. Each dataset is reported at two levels: CA EPIC for z/VSE dataset definition (EDD) and version information.

CA EPIC for z/VSE Dataset Definition

The first two lines for each dataset (beginning with the name) provide the dataset definition. The field titles are listed on these two lines.

SID is the system ID.

UID is the user ID.

RET is the number of days each version is to be retained.

CYC is the number of versions (cycles) to be maintained.

RBLK is the default block size.

REC is the actual record size of the last version.

BLK is the actual block size of the last version.

VLT is the number of the vault method assigned to the dataset.

MODE is the default tape density.

REV indicates whether this is a revolving tape dataset.

UNLBL indicates whether this is a standard label tape dataset.

ENCODE=NO indicates whether tape encryption is activated.

OWN is the owner ID.

TPOOL is the tape pool ID.

In addition to these fields, you may also see the following information in this area of the report:

Any Comment associated with the dataset appears at the end of the first line.

WORK (located between the RET and CYC fields) indicates the dataset is a non-generation dataset.

Version Information

The field titles listed at the top of each page refer to data recorded at the version level. The fields listed are:

SERIAL NO is the tape or disk volume serial number.

VER. NO is the relative version number of the version. There may also be four status codes listed in this column. An "S" means the version is scratched. An "O" means the version is in open status. A "C" means the version is conditionally cataloged. An "M" means the version is the master dataset of an MCAT group.

REEL/EXT is the reel or extent sequence number.

CREATION JOB NAME is the name of the job that created this version.

CREATION DATE is the date the version was created.

EXPIRE DATE is the projected expiration date of the version.

TIME is the time (HH.MM) the version was created.

TP is the dataset's tape pool.

PART is the partition in which the version was created.

PHYS. UNIT is the address of the device the version was created on.

BLOCK COUNT is the number of blocks read the last time the version was accessed.

LAST ACCESS JB NAME is the name of the last job that accessed this version.

LAST ACCESS DATE is the date the version was last accessed.

COUNT USE is the number of times the tape has been opened since it has contained this version.

COUNT ERR is the number of temporary read/write errors on the tape since it has contained this version.

CLEAN DATE is the last clean date of the tape. If the dataset is a disk dataset, the display appears in the following format: EXT=starting address/allocation.actual usage.

TOTAL USE is the total number of times the tape was opened since the last cleaning.

TOTAL ERR is the total number of temporary read/write errors since the last cleaning.

VLT is the current vault location of this tape.

SLT is the current slot location of this tape.

TOTAL ACTIVE is the total number of active tapes on the TSIDUTL report.

TOTAL SCRATCH is the total number of scratch tapes on the TSIDUTL report.

TSIDUTL ALL,DSN Report Example

```
 TSIDUTL= 5.01.00                      D A T A   S E T   M A N A G E M E N T   S Y S T E M                        PAGE    1
 TIME 17.28.11           DSPLY ALL,SEQ=DSN                                                                      DATE 03/11/09
                  SERIAL VER. REEL ---CREATION---   EXPIRE               PHY.  BLOCK --LAST ACCESS-   COUNT  CLEAN  TOTAL
                    NO    NO  /EXT JB NAME  DATE   DATE  TIME TP PART  UNIT  COUNT JB NAME   DATE  USE ERR  DATE USE ERR VLT SLT
$$$.BIG.IJSYSLN   SID=   UID=    RET=   0 CYC=  0 WORK REBLK=      REC=      BLK=      DLA=   0
                  POOL=POOL01    EXT=0000600.0000300   SEC=000     ALR=0000000
$$$.BIG.IJSYS01   SID=   UID=    RET=   0 CYC=  0 WORK REBLK=      REC=      BLK=      DLA=   0
                  VLT= 0 MODE=   REV=NO   UNLBL=NO     ENCODE=NO   OWN=      TPOOL=                SYS001
                  POOL=POOL01    EXT=0000600.0000300   SEC=000     ALR=0000000
$$$.BIG.IJSYS02   SID=   UID=    RET=   0 CYC=  0 WORK REBLK=      REC=      BLK=      DLA=   0
                  VLT= 0 MODE=   REV=NO   UNLBL=NO     ENCODE=NO   OWN=      TPOOL=                SYS002
                  POOL=POOL01    EXT=0000600.0000300   SEC=000     ALR=0000000
$$$.BIG.IJSYS03   SID=   UID=    RET=   0 CYC=  0 WORK REBLK=      REC=      BLK=      DLA=   0
                  VLT= 0 MODE=   REV=NO   UNLBL=NO     ENCODE=NO   OWN=      TPOOL=                SYS003
                  POOL=POOL01    EXT=0000600.0000300   SEC=000     ALR=0000000
$$$.IJSYSLN       SID=   UID=    RET=   0 CYC=  0 WORK REBLK=      REC=      BLK=      DLA=   0
                  POOL=POOL01    EXT=0000060.0000030   SEC=000     ALR=0000000
$$$.IJSYSPH       SID=   UID=    RET=   0 CYC=  0 WORK REBLK=      REC=      BLK=      DLA=   0
                  POOL=POOL01    EXT=0000060.0000030   SEC=000     ALR=0000000
$$$.IJSYS01       SID=   UID=    RET=   0 CYC=  0 WORK REBLK=      REC=      BLK=      DLA=   0
                  VLT= 0 MODE=   REV=NO   UNLBL=NO     ENCODE=NO   OWN=      TPOOL=                SYS001
                  POOL=POOL01    EXT=0000060.0000030   SEC=000     ALR=0000000
$$$.IJSYS02       SID=   UID=    RET=   0 CYC=  0 WORK REBLK=      REC=      BLK=      DLA=   0
                  VLT= 0 MODE=   REV=NO   UNLBL=NO     ENCODE=NO   OWN=      TPOOL=                SYS002
                  POOL=POOL01    EXT=0000060.0000030   SEC=000     ALR=0000000
$$$.IJSYS03       SID=   UID=    RET=   0 CYC=  0 WORK REBLK=      REC=      BLK=      DLA=   0
                  VLT= 0 MODE=   REV=NO   UNLBL=NO     ENCODE=NO   OWN=      TPOOL=                SYS003
                  POOL=POOL01    EXT=0000060.0000030   SEC=000     ALR=0000000
$$$.IJSYS04       SID=   UID=    RET=   0 CYC=  0 WORK REBLK=      REC=      BLK=      DLA=   0
                  VLT= 0 MODE=   REV=NO   UNLBL=NO     ENCODE=NO   OWN=      TPOOL=                SYS004
                  POOL=POOL01    EXT=0000060.0000030   SEC=000     ALR=0000000
$$$.IJSYS05       SID=   UID=    RET=   0 CYC=  0 WORK REBLK=      REC=      BLK=      DLA=   0
                  VLT= 0 MODE=   REV=NO   UNLBL=NO     ENCODE=NO   OWN=      TPOOL=                SYS005
                  POOL=POOL01    EXT=0000060.0000030   SEC=000     ALR=0000000
$$$.IJSYS06       SID=   UID=    RET=   0 CYC=  0 WORK REBLK=      REC=      BLK=      DLA=   0
                  VLT= 0 MODE=   REV=NO   UNLBL=NO     ENCODE=NO   OWN=      TPOOL=                SYS006
                  POOL=POOL01    EXT=0000060.0000030   SEC=000     ALR=0000000
$$$.IJSYS07       SID=   UID=    RET=   0 CYC=  0 WORK REBLK=      REC=      BLK=      DLA=   0
                  VLT= 0 MODE=   REV=NO   UNLBL=NO     ENCODE=NO   OWN=      TPOOL=                SYS007
                  POOL=POOL01    EXT=0000060.0000030   SEC=000     ALR=0000000
```

```
TSIDUTL= 5.01.00                        D A T A   S E T   M A N A G E M E N T   S Y S T E M                          PAGE    5
 TIME 17.28.11            DSPLY ALL,SEQ=DSN                                                                      DATE 03/11/09
                  SERIAL VER. REEL ---CREATION---   EXPIRE                PHY.  BLOCK --LAST ACCESS-   COUNT  CLEAN  TOTAL
                    NO    NO  /EXT JB NAME  DATE   DATE  TIME TP PART  UNIT  COUNT JB NAME   DATE  USE ERR  DATE USE ERR VLT SLT
BG7.TSIDUTL.SCR.WORK
                  SID=   UID=    RET=   0 CYC=  0 WORK REBLK=      REC=      BLK=      DLA=   0
                  POOL=POOL50    EXT=0000100.0000050   SEC=000     ALR=0000000
CIB207.TST.01     SID=   UID=AC  RET=   7 CYC=  3      REBLK=      REC=      BLK=      DLA=   0  **** AUTO CATALOGUE ****
                  POOL=POOL80    EXT=0000060.0000030   SEC=000     ALR=0000000
CIB207.TST.02     SID=   UID=AC  RET=   7 CYC=  3      REBLK=      REC=      BLK=      DLA=   0  **** AUTO CATALOGUE ****
                  POOL=POOL80    EXT=0000060.0000030   SEC=000     ALR=0000000
CIB207.TST.03     SID=   UID=AC  RET=   7 CYC=  3      REBLK=      REC=      BLK=      DLA=   0  **** AUTO CATALOGUE ****
                  POOL=POOL80    EXT=0000060.0000030   SEC=000     ALR=0000000
CIB207.TST.04     SID=   UID=AC  RET=   7 CYC=  3      REBLK=      REC=      BLK=      DLA=   0  **** AUTO CATALOGUE ****
                  POOL=POOL80    EXT=0000060.0000030   SEC=000     ALR=0000000
CIB207.TST.05     SID=   UID=AC  RET=   7 CYC=  3      REBLK=      REC=      BLK=      DLA=   0  **** AUTO CATALOGUE ****
                  POOL=POOL80    EXT=0000060.0000030   SEC=000     ALR=0000000
CIB207.TST.06     SID=   UID=AC  RET=   7 CYC=  3      REBLK=      REC=      BLK=      DLA=   0  **** AUTO CATALOGUE ****
                  POOL=POOL80    EXT=0000060.0000030   SEC=000     ALR=0000000
CIB207.TST.07     SID=   UID=AC  RET=   7 CYC=  3      REBLK=      REC=      BLK=      DLA=   0  **** AUTO CATALOGUE ****
                  POOL=POOL80    EXT=0000060.0000030   SEC=000     ALR=0000000
CIB207.TST.08     SID=   UID=AC  RET=   7 CYC=  3      REBLK=      REC=      BLK=      DLA=   0  **** AUTO CATALOGUE ****
                  POOL=POOL80    EXT=0000060.0000030   SEC=000     ALR=0000000
CIB207.TST.09     SID=   UID=AC  RET=   7 CYC=  3      REBLK=      REC=      BLK=      DLA=   0  **** AUTO CATALOGUE ****
                  POOL=POOL80    EXT=0000060.0000030   SEC=000     ALR=0000000
CIB207.TST.10     SID=   UID=AC  RET=   7 CYC=  3      REBLK=      REC=      BLK=      DLA=   0  **** AUTO CATALOGUE ****
                  POOL=POOL80    EXT=0000060.0000030   SEC=000     ALR=0000000
DAN901.FILEOUT    SID=   UID=    RET=   1 CYC=  0      REBLK=      REC=   36 BLK= 3600 DLA=   0  INCIDENT DAN901 TEST FILE
                  VLT= 0 MODE=   REV=NO   UNLBL=NO     ENCODE=NO   OWN=      TPOOL=                SYS010
                  EGB001 S     1 DAN901   02/08/09 02/09/09 12.17     BG    301     9 DAN901   02/08/09  8  0     8   0   0    0
DBK.FILEOUT.STEP1 SID=   UID=AC  RET=   0 CYC=  0 WORK REBLK=      REC=   36 BLK= 3600 DLA=   0  **** AUTO CATALOGUE ****
                  POOL=POOL01    EXT=0000200.0000100   SEC=000     ALR=0000000
                  EPC247    1  1 DBK865   03/03/09 03/03/09 13.39     BG    247  0 DBK865   03/03/09  2  EXT=0000004/00200.00002
DBK.FILEOUT.STEP2 SID=   UID=AC  RET=   0 CYC=  0 WORK REBLK= 7200 REC=   36 BLK= 7200 DLA=   0  **** AUTO CATALOGUE ****
                  POOL=POOL01    EXT=0000200.0000100   SEC=000     ALR=0000000
                  EPC229    1  1 DBK865   03/03/09 03/03/09 13.41     BG    229  0 DBK865   03/03/09  2  EXT=0000323/00200.00002
```

Volume Serial Number and Vault Sequences

This report lists tape datasets. Each tape volser is listed along with data for the version the tape contains. Dataset definitions are not listed. The first field listed is the version's dataset name. Refer to "Version Data" under "Display ALL, Dataset Name Sequence" for an explanation of the other fields contained in this report.

Example

```
TSIDUTL= 5.01.00                    D A T A   S E T   M A N A G E M E N T   S Y S T E M                    PAGE    1
TIME 17.28.11          DSPLY ALL,SEQ=SER                                                    DATE 03/11/09
                SERIAL VER. REEL ---CREATION---   EXPIRE               PHY.  BLOCK --LAST ACCESS-   COUNT  CLEAN  TOTAL
                  NO    NO  /EXT JB NAME  DATE   DATE    TIME TP PART  UNIT  COUNT JB NAME  DATE USE ERR  DATE USE ERR VLT SLT
BG5.KJ.HIST       ACCPT1    1  1 RFAST    02/05/09 02/05/09 23.56  BG    343   0 RFAST    02/05/09  1  0     1   0   0    0
DAN901.FILEOUT    EGB001 S     1 DAN901   02/08/09 02/09/09 12.17  BG    301   9 DAN901   02/08/09  8  0     8   0   0    0
DBK827.STEP2      EGB002    1  1 DBK827   03/03/09 03/10/09 14.55  BG    343   1 DBK827   03/03/09  1  0     1   0   0    0
DBK827.STEP1      EGB002    1  1 DBK827   03/03/09 03/03/09 14.55  BG    229   1 DBK827   03/03/09  5  0     5   0   0    0
KEITH.GEN3        EPC229    1  1 PURGEN   02/03/09 02/03/09 23.57  BG    229   0 PURGEN   02/03/09  1   EXT=0000367/00090.00002
DBK.FILEOUT.STEP2 EPC229    1  1 DBK865   03/03/09 03/03/09 13.41  BG    229   0 DBK865   03/03/09  2   EXT=0000323/00200.00002
REBLD.H2          EPC229    6  1 RECOVER1 02/23/09 02/23/09 18.44  BG    229   0 RECOVER1 02/23/09  4   EXT=0002569/00300.00002
REBLD.H4          EPC229    8  1 RECOVER1 02/23/09 02/23/09 18.46  BG    229   0 RECOVER1 02/23/09  4   EXT=0002554/00300.00002
REBLD.H3          EPC229    4  1 RECOVER1 02/23/09 02/23/09 18.45  BG    229   0 RECOVER1 02/23/09  4   EXT=0002562/00300.00002
REBLD.H3          EPC229    2  1 RECOVER1 02/23/09 02/23/09 18.45  BG    229   0 RECOVER1 02/23/09  4   EXT=0002564/00300.00003
REBLD.H           EPC229    1  1 RECOVER1 02/23/09 02/23/09 18.42  BG    229   0 RECOVER1 02/23/09  1   EXT=0002552/00300.00002
REBLD.H1          EPC229    4  1 RECOVER1 02/23/09 02/23/09 18.42  BG    229   0 RECOVER1 02/23/09  4   EXT=0002575/00300.00002
REBLD.H2          EPC229    4  1 RECOVER1 02/23/09 02/23/09 18.44  BG    229   0 RECOVER1 02/23/09  4   EXT=0002571/00300.00002
REBLD.H2          EPC229    2  1 RECOVER1 02/23/09 02/23/09 18.44  BG    229   0 RECOVER1 02/23/09  4   EXT=0002573/00300.00002
REBLD.A           EPC229    3  1 RECOVER1 02/23/09 02/23/09 18.37  BG    229   0 RECOVER1 02/23/09  1   EXT=0000133/00100.00002
REBLD.H1          EPC229    2  1 RECOVER1 02/23/09 02/23/09 18.43  BG    229   0 RECOVER1 02/23/09  4   EXT=0002577/00300.00002
REBLD.H4          EPC229    4  1 RECOVER1 02/23/09 02/23/09 18.47  BG    229   0 RECOVER1 02/23/09  3   EXT=0002558/00300.00002
REBLD.H4          EPC229    2  1 RECOVER1 02/23/09 02/23/09 18.47  BG    229   0 RECOVER1 02/23/09  3   EXT=0002560/00300.00002
REBLD.B           EPC229    3  2 RECOVER1 02/23/09 12/31/99 18.38  BG    229   0 RECOVER1 02/23/09  1   EXT=0000129/00002.00002
REBLD.B           EPC229    1  2 RECOVER1 02/23/09 12/31/99 18.39  BG    229   0 RECOVER1 02/23/09  1   EXT=0000072/00002.00002
REBLD.B           EPC229    2  1 RECOVER1 02/23/09 02/23/09 18.38  BG    229   0 RECOVER1 02/23/09  1   EXT=0000011/00004.00004
REBLD.B           EPC229    2  3 RECOVER1 02/23/09 12/31/99 18.38  BG    229   0 RECOVER1 02/23/09  1   EXT=0000353/00002.00002
REBLD.A           EPC229    1  1 RECOVER1 02/23/09 02/23/09 18.38  BG    229   0 RECOVER1 02/23/09  1   EXT=0000135/00100.00002
KEITH.GEN         EPC229    3  1 PURGEN   02/03/09 02/03/09 23.58  BG    229   0 PURGEN   02/03/09  1   EXT=0000375/00090.00002
KEITH.GEN         EPC229    1  1 PURGEN   02/03/09 02/03/09 23.58  BG    229   0 PURGEN   02/03/09  1   EXT=0000377/00090.00002
KEITH.DISK        EPC229 O     1 DISK     02/17/09 02/17/09 11.36  BG    229   0 DISK     02/17/09  1   EXT=0002550/00300.00300
REBLD.H4          EPC229    6  1 RECOVER1 02/23/09 02/23/09 18.46  BG    229   0 RECOVER1 02/23/09  4   EXT=0002556/00300.00002
BG7.KEITH.DISKA   EPC229    1  1 DITTO    03/01/09 03/01/09 17.55  BG    229   0 DITTO    03/01/09  3   EXT=0000133/00030.00002
DBK827.STEP4      EPC229    1  1 DBK827   03/03/09 03/03/09 15.29  BG    229   0 DBK827   03/03/09  1   EXT=0000138/00100.00002
BG7.KEITH.DISKB   EPC229    1  1 DITTO    03/01/09 03/01/09 17.56  BG    229   0 DITTO    03/01/09  1   EXT=0000135/00030.00003
REBLD.H2          EPC229    8  1 RECOVER1 02/23/09 02/23/09 18.43  BG    229   0 RECOVER1 02/23/09  3   EXT=0002567/00300.00002
VAULT.IJSYSLN     EPC247    1  1 VAULT    03/09/09 03/09/09 12.55  BG    247   0 VAULT    03/09/09  2   EXT=0000008/00100.00100
REBLD.H4          EPC247    5  1 RECOVER1 02/23/09 02/23/09 18.46  BG    247   0 RECOVER1 02/23/09  4   EXT=0000008/00300.00002
DBK.FILEOUT.STEP1 EPC247    1  1 DBK865   03/03/09 03/03/09 13.39  BG    247   0 DBK865   03/03/09  2   EXT=0000004/00200.00002
REBLD.H4          EPC247    7  1 RECOVER1 02/23/09 02/23/09 18.46  BG    247   0 RECOVER1 02/23/09  4   EXT=0000006/00300.00002
REBLD.H2          EPC247    1  1 RECOVER1 02/23/09 02/23/09 18.44  BG    247   0 RECOVER1 02/23/09  4   EXT=0000025/00300.00002
REBLD.H3          EPC247    1  1 RECOVER1 02/23/09 02/23/09 18.45  BG    247   0 RECOVER1 02/23/09  4   EXT=0000017/00300.00002
REBLD.H3          EPC247    3  1 RECOVER1 02/23/09 02/23/09 18.45  BG    247   0 RECOVER1 02/23/09  4   EXT=0000014/00300.00003
REBLD.H4          EPC247    1  1 RECOVER1 02/23/09 02/23/09 18.47  BG    247   0 RECOVER1 02/23/09  4   EXT=0000012/00300.00002
REBLD.H4          EPC247    3  1 RECOVER1 02/23/09 02/23/09 18.47  BG    247   0 RECOVER1 02/23/09  4   EXT=0000010/00300.00002
REBLD.H2          EPC247    7  1 RECOVER1 02/23/09 02/23/09 18.44  BG    247   0 RECOVER1 02/23/09  4   EXT=0000019/00300.00002
REBLD.H2          EPC247    3  1 RECOVER1 02/23/09 02/23/09 18.44  BG    247   0 RECOVER1 02/23/09  4   EXT=0000023/00300.00002
REBLD.H1          EPC247    3  1 RECOVER1 02/23/09 02/23/09 18.42  BG    247   0 RECOVER1 02/23/09  4   EXT=0000027/00300.00002
REBLD.A           EPC247    2  1 RECOVER1 02/23/09 02/23/09 18.38  BG    247   0 RECOVER1 02/23/09  1   EXT=0002452/00100.00002
REBLD.H1          EPC247    1  1 RECOVER1 02/23/09 02/23/09 18.43  BG    247   0 RECOVER1 02/23/09  4   EXT=0000029/00300.00003
REBLD.B           EPC247    3  3 RECOVER1 02/23/09 12/31/99 18.38  BG    247   0 RECOVER1 02/23/09  1   EXT=0002454/00002.00002
REBLD.B           EPC247    3  1 RECOVER1 02/23/09 02/23/09 18.38  BG    247   0 RECOVER1 02/23/09  1   EXT=0002625/00004.00004
REBLD.B           EPC247    2  2 RECOVER1 02/23/09 12/31/99 18.38  BG    247   0 RECOVER1 02/23/09  1   EXT=0002456/00002.00002
REBLD.B           EPC247    1  3 RECOVER1 02/23/09 12/31/99 18.39  BG    247   0 RECOVER1 02/23/09  1   EXT=0002462/00002.00002
REBLD.B           EPC247    1  1 RECOVER1 02/23/09 02/23/09 18.39  BG    247   0 RECOVER1 02/23/09  1   EXT=0002458/00004.00004
```

Expiration Date, Creation Date, and Creation Job Sequences

Each version is listed with all version data. Dataset definitions are not listed. The first field listed is the version's dataset name. Refer to "Version Data" under "Display ALL, Dataset Name Sequence" for an explanation of the other fields contained in this report.

Display Active

This is a detailed report listing the active versions of each dataset. Refer to the "Display ALL, Dataset Name Sequence" for an explanation of the fields contained in this report.

Display NEW Report

This report lists all datasets created since the last time you ran this report. The fields in this report are described under "Display ALL, Dataset Name Sequence".

TSIDUTL NEW Report Example

```
 TSIDUTL= 5.01.00                      D A T A   S E T   M A N A G E M E N T   S Y S T E M                          PAGE    2
 TIME 15.55.19           DSPLY NEW                                                                                 DATE 03/09/09
                 SERIAL VER. REEL ---CREATION---   EXPIRE                PHY.  BLOCK --LAST ACCESS-   COUNT  CLEAN  TOTAL
                   NO    NO  /EXT JB NAME  DATE     DATE  TIME TP PART  UNIT  COUNT JB NAME DATE  USE ERR  DATE USE ERR VLT SLT
BG3.KEITH.SORTIN  SID=   UID=    RET=   0 CYC=  0 WORK REBLK=      REC=   40 BLK=  800 DLA=   0
                  POOL=POOL01    EXT=0000100.0000050   SEC=000     ALR=0000000
                  VSDAT2    1  1 SORTD    02/21/09 02/21/09 11.10     BG    181   0 SORTD    02/21/09  3 EXT=0005524/00100.00002
BG3.KEITH.SORTOUT SID=   UID=    RET=   0 CYC=  0 WORK REBLK=      REC=      BLK=      DLA=   0
                  POOL=POOL01    EXT=0000100.0000050   SEC=000     ALR=0000000
                  EPC551    1  1 SORTD    02/21/09 02/21/09 11.11     BG    551   0 SORTD    02/21/09  2 EXT=0008105/00100.00100
BG3.KEITH.SYSIPT  SID=   UID=    RET=   0 CYC=  0 WORK REBLK=OPT   REC=   80 BLK=23440 DLA=   0
                  POOL=POOL01    EXT=0000500.0000250   SEC=000     ALR=0000000
BG3.KEITH.SYSLST  SID=   UID=    RET=   0 CYC=  0 WORK REBLK=OPT   REC=   80 BLK=23440 DLA=   0
                  POOL=POOL01    EXT=0000500.0000250   SEC=000     ALR=0000000
BG3.KEITH.SYSPCH  SID=   UID=    RET=   0 CYC=  0 WORK REBLK=OPT   REC=   80 BLK=23440 DLA=   0
                  POOL=POOL01    EXT=0000500.0000250   SEC=000     ALR=0000000
BG3.KJ.DAILY      SID=   UID=    RET=   0 CYC=  1      REBLK=      REC=  338 BLK= 5746 DLA=   0
BG3.KJ.DSNBU      SID=   UID=    RET=   0 CYC=  1      REBLK=      REC=  205 BLK=20500 DLA=   0
BG3.KJ.TAPE       SID=   UID=    RET=   0 CYC=  1      REBLK=      REC=      BLK=      DLA=   0
BG3.KJ.TPOL       SID=   UID=    RET=   0 CYC=  1      REBLK=      REC=   40 BLK=  800 DLA=   0
                  VLT= 0 MODE=   REV=NO   UNLBL=NO     ENCODE=NO   OWN=      TPOOL=A                SYS000
                  990050 S     1 POLTP    02/02/09 02/02/09 11.32  A  BG    340     0 POLTP    02/02/09  1  0     3   0   0    0
BG3.PRIME         SID=   UID=    RET=   0 CYC=  2      REBLK=      REC=   40 BLK=  800 DLA=   0
BG3.SUB1          SID=   UID=    RET=   0 CYC=  2      REBLK=      REC=   40 BLK=  800 DLA=   0  NSU FILE OF BG3.PRIME
BG3.SUB2          SID=   UID=    RET=   0 CYC=  2      REBLK=      REC=      BLK=      DLA=   0  NSU FILE OF BG3.PRIME
BG3.SUB3          SID=   UID=    RET=   0 CYC=  2      REBLK=      REC=      BLK=      DLA=   0  NSU FILE OF BG3.PRIME
BG3.SUB4          SID=   UID=    RET=   0 CYC=  2      REBLK=      REC=      BLK=      DLA=   0  NSU FILE OF BG3.PRIME
BG3.SUB5          SID=   UID=    RET=   0 CYC=  2      REBLK=      REC=      BLK=      DLA=   0  NSU FILE OF BG3.PRIME
BG3.SUB6          SID=   UID=    RET=   0 CYC=  2      REBLK=      REC=      BLK=      DLA=   0  NSU FILE OF BG3.PRIME
BG3.SUB7          SID=   UID=    RET=   0 CYC=  2      REBLK=      REC=      BLK=      DLA=   0  NSU FILE OF BG3.PRIME
BG3.TSIDUTL.SCR.WORK
                  SID=   UID=    RET=   0 CYC=  0 WORK REBLK=      REC=   50 BLK= 4000 DLA=   0
                  POOL=POOL50    EXT=0000100.0000050   SEC=000     ALR=0000000
```

Display PULL Report

This report lists only the most current version for each dataset. The fields in this report are described under "Display ALL, Dataset Name Sequence".

Example

```
 TSIDUTL= 5.01.00                      D A T A   S E T   M A N A G E M E N T   S Y S T E M                        PAGE    1
 TIME 15.46.51           DSPLY PULL,SEQ=DSN                                                                    DATE 03/09/09
                  SERIAL VER. REEL ---CREATION---   EXPIRE               PHY.  BLOCK --LAST ACCESS-   COUNT  CLEAN  TOTAL
                    NO    NO  /EXT JB NAME  DATE     DATE  TIME TP PART  UNIT  COUNT JB NAME DATE  USE ERR  DATE USE ERR VLT SLT
*EPIC.EVENT.DSN*  SID=   UID=    RET=   7 CYC=  3      REBLK=      REC=      BLK=      DLA=   0
       G=0000002 *EVENT    1  1 EVENT    02/10/93 02/17/93 12.19  F4    000     0 EVENT7   02/10/93  1  0     1   0   0    0
BG2.KEITH.SORTIN  SID=   UID=    RET=   0 CYC=  0 WORK REBLK=      REC=   40 BLK=  800 DLA=   0
                  POOL=POOL01    EXT=0000100.0000050   SEC=000     ALR=0000000
                  EPC551    1  1 SORTD   09/22/93 09/22/93 16.33  BG    551     0 SORTD    09/22/93  3   EXT=0008700/00100.00002
BG2.KEITH.SORTOUT SID=   UID=    RET=   0 CYC=  0 WORK REBLK=      REC=      BLK=      DLA=   0
                  POOL=POOL01    EXT=0000100.0000050   SEC=000     ALR=0000000
                  VSDAT2    1  1 SORTD   09/22/93 09/22/93 16.33  BG    181     0 SORTD    09/22/93  2   EXT=0004400/00100.00100
BG2.KJ.SUB1       SID=   UID=    RET=   0 CYC=  1      REBLK=      REC=   40 BLK=  800 DLA=   0  NSU FILE OF BG2.KJ.PRIME
       F=0000002 QCTT07    1  1 DAU293  01/26/09 01/26/09 21.50  BG    341     0 DAU293   01/26/09  1  0         1   0   0    0
BG2.KJ.SUB2       SID=   UID=    RET=   0 CYC=  1      REBLK=      REC=   40 BLK=  800 DLA=   0  NSU FILE OF BG2.KJ.PRIME
       F=0002457 QCTT07    1  1 DAU293  01/26/09 01/26/09 21.50  BG    341     0 DAU293   01/26/09  1  0         1   0   0    0
BG3.KEITH.CKDT    SID=   UID=    RET=   0 CYC=  0 WORK REBLK=      REC= 9530 BLK=19060 DLA=   0
                  POOL=POOL50    EXT=0000010.0000005   SEC=000     ALR=0000000
                  VSDAT2    1  1 TRUNC   01/22/09 01/22/09 13.10  BG    181     0 TRUNC    01/22/09  2   EXT=0007247/00010.00002
BG3.KEITH.DISKD   SID=   UID=    RET=   0 CYC=  0 WORK REBLK=      REC=  500 BLK=  500 DLA=   0
                  POOL=POOL50    EXT=0000200.0000100   SEC=000     ALR=0000000
                  VSDAT2    1  1 DMP5S   02/19/09 02/19/09 13.22  BG    181     0 DMP5S    02/19/09  7   EXT=0006230/00200.00006
BG3.KEITH.FRAG    SID=   UID=    RET=   0 CYC=  2      REBLK=      REC=   40 BLK=  800 DLA=   0
                  VLT= 0 MODE=   REV=NO   UNLBL=NO     ENCODE=NO   OWN=      TPOOL=                  SYS020
                  POOL=POOL50    EXT=0999999.0499999   SEC=000     ALR=0000000
       G=0000004 EPC551    1  1 FRAG    01/26/09 01/26/09 15.02  BG    551     0 FRAG     01/26/09  2   EXT=0002709/00228.00002
BG3.KEITH.SORTIN  SID=   UID=    RET=   0 CYC=  0 WORK REBLK=      REC=   40 BLK=  800 DLA=   0
                  POOL=POOL01    EXT=0000100.0000050   SEC=000     ALR=0000000
                  VSDAT2    1  1 SORTD   02/21/09 02/21/09 11.10  BG    181     0 SORTD    02/21/09  3   EXT=0005524/00100.00002
BG3.KEITH.SORTOUT SID=   UID=    RET=   0 CYC=  0 WORK REBLK=      REC=      BLK=      DLA=   0
                  POOL=POOL01    EXT=0000100.0000050   SEC=000     ALR=0000000
                  EPC551    1  1 SORTD   02/21/09 02/21/09 11.11  BG    551     0 SORTD    02/21/09  2   EXT=0008105/00100.00100
BG3.KEITH.TAPEZ   SID=   UID=    RET=   0 CYC=  3      REBLK=      REC=   40 BLK=  800 DLA=   0
                  POOL=POOL50    EXT=0000200.0000100   SEC=000     ALR=0000000
       G=0000003 EPC551    1  1 CONCAD  01/23/09 01/23/09 10.34  BG    551     0 CONCAD   01/23/09  2   EXT=0000409/00200.00002
BG3.KEITH.TRUNC   SID=   UID=    RET=   1 CYC=  0      REBLK=      REC=   40 BLK=  800 DLA=   0
                  POOL=POOL50    EXT=0000300.0000150   SEC=000     ALR=0000000
       G=0000003 VSDAT2    1  1 TRUNC   01/22/09 01/23/09 13.09  BG    181     0 TRUNC    01/22/09  1   EXT=0001001/00300.00300
BG3.KJ.NOWN       SID=   UID=    RET=   0 CYC=  1      REBLK=      REC=   40 BLK=  800 DLA=   0
                  OWMMM     1  1 OWNTP   09/14/93 09/14/93 13.29  BG    884     0 OWNTP    09/14/93  1  0         1   0   0    0
```

Display Vault Report

This report lists the vault and slot locations of all tapes located in offsite vaults. Refer to "Version Data" under "Display ALL, Dataset Name Sequence" for an explanation of the fields contained in this report.

Example

```
 TSIDUTL= 5.01.00                        D A T A   S E T   M A N A G E M E N T   S Y S T E M                          PAGE    1
 TIME 15.43.35            DSPLY VAULT                                                                        DATE 03/09/09
                  SERIAL VER. REEL ---CREATION---   EXPIRE                PHY.  BLOCK --LAST ACCESS-   COUNT  CLEAN  TOTAL
                    NO    NO  /EXT JB NAME  DATE     DATE  TIME TP PART  UNIT  COUNT JB NAME DATE  USE ERR  DATE USE ERR VLT SLT
 RONNS.VLT1.TEST  SID=   UID=    RET=   0 CYC= 10      REBLK=      REC=      BLK=      DLA=   0
                  VLT= 1 MODE=   REV=NO   UNLBL=NO     ENCODE=NO   OWN=      TPOOL=                 SYS000
                  VLT101    1  1 VAULT    03/09/09 03/09/09 14.20     BG    000     0 VAULT    03/09/09  0  0     0   0   1    1
                  VLT102    2  1 VAULT    03/09/09 03/09/09 14.20     BG    000     0 VAULT    03/09/09  0  0     0   0   2   20
                  VLT103    3  1 VAULT    03/09/09 03/09/09 14.20     BG    000     0 VAULT    03/09/09  0  0     0   0   3   30
                  VLT104    4  1 VAULT    03/09/09 03/09/09 14.20     BG    000     0 VAULT    03/09/09  0  0     0   0   4   40
                  VLT105    5  1 VAULT    03/09/09 03/09/09 14.20     BG    000     0 VAULT    03/09/09  0  0     0   0   5   50
 RONNS.VLT2.TEST  SID=   UID=    RET=   0 CYC= 10      REBLK=      REC=      BLK=      DLA=   0
                  VLT= 2 MODE=   REV=NO   UNLBL=NO     ENCODE=NO   OWN=      TPOOL=                 SYS000
                  VLT201    1  1 VAULT    03/09/09 03/09/09 14.20     BG    000     0 VAULT    03/09/09  0  0     0   0   5   51
                  VLT202    2  1 VAULT    03/09/09 03/09/09 14.20     BG    000     0 VAULT    03/09/09  0  0     0   0   4   41
                  VLT203    3  1 VAULT    03/09/09 03/09/09 14.20     BG    000     0 VAULT    03/09/09  0  0     0   0   3   31
                  VLT204    4  1 VAULT    03/09/09 03/09/09 14.20     BG    000     0 VAULT    03/09/09  0  0     0   0   2   21
 RONNS.VLT3.TEST  SID=   UID=    RET=   0 CYC= 10      REBLK=      REC=      BLK=      DLA=   0
                  VLT= 3 MODE=   REV=NO   UNLBL=NO     ENCODE=NO   OWN=      TPOOL=                 SYS000
                  VLT301    1  1 VAULT    03/09/09 03/09/09 14.21     BG    000     0 VAULT    03/09/09  0  0     0   0   1    2
                  VLT302    2  1 VAULT    03/09/09 03/09/09 14.21     BG    000     0 VAULT    03/09/09  0  0     0   0   2   22
                  VLT303    3  1 VAULT    03/09/09 03/09/09 14.21     BG    000     0 VAULT    03/09/09  0  0     0   0   3   32
                  VLT304    4  1 VAULT    03/09/09 03/09/09 14.21     BG    000     0 VAULT    03/09/09  0  0     0   0   4   42
                  VLT305    5  1 VAULT    03/09/09 03/09/09 14.21     BG    000     0 VAULT    03/09/09  0  0     0   0   5   52
 RONNS.VLT4.TEST  SID=   UID=    RET=   0 CYC= 10      REBLK=      REC=      BLK=      DLA=   0
                  VLT= 1 MODE=   REV=NO   UNLBL=NO     ENCODE=NO   OWN=      TPOOL=                 SYS000
                  VLT401    1  1 VAULT    03/09/09 03/09/09 14.21     BG    000     0 VAULT    03/09/09  0  0     0   0   1    3
                  VLT402    2  1 VAULT    03/09/09 03/09/09 14.21     BG    000     0 VAULT    03/09/09  0  0     0   0   2   23
                  VLT403    3  1 VAULT    03/09/09 03/09/09 14.21     BG    000     0 VAULT    03/09/09  0  0     0   0   3   33
                  VLT404    4  1 VAULT    03/09/09 03/09/09 14.21     BG    000     0 VAULT    03/09/09  0  0     0   0   4   43
                  VLT405    5  1 VAULT    03/09/09 03/09/09 14.21     BG    000     0 VAULT    03/09/09  0  0     0   0   5   53
 RONNS.VLT5.TEST  SID=   UID=    RET=   0 CYC= 10      REBLK=      REC=      BLK=      DLA=   0
                  VLT= 2 MODE=   REV=NO   UNLBL=NO     ENCODE=NO   OWN=      TPOOL=                 SYS000
                  VLT501    1  1 VAULT    03/09/09 03/09/09 14.21     BG    000     0 VAULT    03/09/09  0  0     0   0   5   54
                  VLT502    2  1 VAULT    03/09/09 03/09/09 14.21     BG    000     0 VAULT    03/09/09  0  0     0   0   4   44
                  VLT503    3  1 VAULT    03/09/09 03/09/09 14.21     BG    000     0 VAULT    03/09/09  0  0     0   0   3   34
                  VLT504    4  1 VAULT    03/09/09 03/09/09 14.21     BG    000     0 VAULT    03/09/09  0  0     0   0   2   24
                  VLT505    5  1 VAULT    03/09/09 03/09/09 14.21     BG    000     0 VAULT    03/09/09  0  0     0   0   1    4
         TOTAL ACTIVE TAPES      24  TOTAL SCRATCH TAPES
```

Display Scratch Report

This report lists scratch tapes along with data specific to the version the scratch tape contains. Refer to "Version Data" under "Dataset Name Sequence" for an explanation of the fields contained in this report.

Example

```
 TSIDUTL= 5.01.00                      D A T A   S E T   M A N A G E M E N T   S Y S T E M                        PAGE    1
 TIME 17.28.11           DSPLY SCRATCH                                                                    DATE 03/11/09
                  SERIAL VER. REEL ---CREATION---   EXPIRE                PHY.  BLOCK --LAST ACCESS-   COUNT  CLEAN  TOTAL
                    NO    NO  /EXT JB NAME  DATE   DATE  TIME TP PART  UNIT  COUNT JB NAME   DATE  USE ERR  DATE USE ERR VLT SLT
** DSN DELETED ** KLSA02 S     1 TFMRGTF  02/03/09 02/03/09 01.12     BG    343     5 TFMRGTF  02/03/09  3  0     3   0   0    0
** DSN DELETED ** MOOGA1 S     1 UTL001   02/23/09 02/23/09 17.05     BG    342     0 UTL001   02/23/09  1  0     1   0   0    0
A                 0ASER1 S     1 XYZ      02/09/09 02/09/09 00.17     BG    000     0 XYZ      02/09/09  0  0     0   0   0    0
BG5.KJ.DISKA      000X01 S     1 RFAST    02/06/09 02/06/09 00.07     BG    000     0 RFAST    02/06/09  0  0     0   0   0    0
BG5.KJ.DISKA      KNOTS  S     1 RFAST    02/06/09 02/06/09 00.07     BG    343     0 RFAST    02/06/09  2  0     2   0   0    0
DAN901.FILEOUT    EGB001 S     1 DAN901   02/08/09 02/09/09 12.17     BG    301     9 DAN901   02/08/09  8  0     8   0   0    0
KEITH.VAULT4      0SCRA5 S     1 VAULTTF  02/03/09 02/03/09 22.48     BG    000     0 VAULTTF  02/03/09  0  0     0   0   0    0
KEITH.VAULT4      0SCRA4 S     1 VAULTTF  02/03/09 02/03/09 22.48     BG    000     0 VAULTTF  02/03/09  0  0     0   0   0    0
KEITH.VAULT4      0SCRA1 S     1 VAULTTF  02/03/09 02/03/09 22.48     BG    000     0 VAULTTF  02/03/09  0  0     0   0   0    0
KEITH.VAULT4      0SCRA2 S     2 VAULTTF  02/03/09 02/03/09 22.48     BG    000     0 VAULTTF  02/03/09  0  0     0   0   0    0
KEITH.VAULT4      0SCRA3 S     3 VAULTTF  02/03/09 02/03/09 22.48     BG    000     0 VAULTTF  02/03/09  0  0     0   0   0    0
KJ.OFL.TST3       0OTST3 S     1 OFL1     02/05/09 02/05/09 19.15     BG    000     0 OFL1     02/05/09  0  0     0   0   0    0
REBLD.C           QCTT10 S     1 RECOVER1 02/23/09 02/23/09 18.48     BG    340     0 RECOVER1 02/23/09  1  0     1   0   0    0
         TOTAL ACTIVE TAPES          TOTAL SCRATCH TAPES      13
```

Projected Scratch (PSCRATCH) Report

This report lists all tapes that becomes scratch tapes by the specified date. It also reports current DSN Catalog utilization statistics.

The projected date you requested is shown next to the report title PSCRATCH and written in Julian format. For instance PSCRATCH 96001 indicates a projected scratch date of January 1, 1996.

At the end of the report, message EP119 is issued to report on DSN Catalog utilization.

Fields

Name	Description
SERIAL NO	Volume serial number of the tape
FILE NAME	Dataset name of the version written on this tape
CREATION DATE	Date this version was created
EXPIRATION DATE	Date retention criteria may be satisfied for this version
TAPE ERRORS	Number of temporary read/write errors encountered on this tape while it contained this version
TOT-ERR	Total number of errors since the tape was last cleaned
TOT-USED	Total number of times the tape was used since the last cleaning
DT-CLEAN	Date the tape was last cleaned
POOL	
VLT	Current vault location of the tape
SLOT	Current slot location of the tape

PSCRATCH Report Example

```
 TSIDUTL= 5.01.00                    D A T A   S E T   M A N A G E M E N T   S Y S T E M                    PAGE    1
 TIME 15.28.04          PSCRATCH 99095                                                                  DATE 03/09/09
SERIAL                                          CREATION  EXPIRATION   TAPE
  NO     FILE NAME                                 DATE       DATE     ERRORS TOT-ERR TOT-USED  DT-CLEAN  POOL     VLT   SLOT
JA0001  JUAN.OLD                                 02/25/09   02/25/09       0       0        0    00000              0      0
JA0002  JUAN.TRANS                               02/25/09   02/25/09       0       0        0    00000              0      0
JA0012  JUAN.TRANS                               02/25/09   02/25/09       0       0        0    00000              0      0
QCTT01  BG7.KJ.DSNBU                             02/23/09   02/23/09       0       0        1    00000              0      0
```

TSIDVLT Reports

TSIDVLT produces two reports listing tapes which should be moved. One is a Picking List which is sorted by the tape's current location. The other is a Distribution List which is sorted by the tapes' new destination. Both reports list the current location of tape reels and their new location. They are designed to assist vault workers in the physical movement of tapes from vault to vault.

To produce these reports, see the topic Printing VTOC Entries (TSIDLVT) in this guide.

Fields

Name	Description
VAULT INFORMATION & SLOT	Number and name of the vault and the number of the slot from which the tape should be moved
SER-NO	Volume serial number of the tape
VER	Number of the version on the tape
REEL	Reel sequence number of the tape (important for multi-volume versions)
DATASET NAME	Dataset name of the version on the tape
VAULT INFORMATION & SLOT	Number and name of the vault and the number of the slot to which the tape should be moved

Vault Movement Report Example

```
TSIDVLT                                   VAULT MOVEMENT REPORT
DATE 03/09/09                                                                                          TIME 14 25 07
----------F R O M---------                                                              -----------T O-----------
VAULT INFORMATION            SLOT  SER-NO  VER REEL   DATA SET NAME                     VAULT INFORMATION           SLOT
  0 MAIN LIBRARY                   VLT101   1    1   RONNS.VLT1.TEST                    1 VAULT 1                     1
  0 MAIN LIBRARY                   VLT102   2    1   RONNS.VLT1.TEST                    2 VAULT 2                    20
  0 MAIN LIBRARY                   VLT103   3    1   RONNS.VLT1.TEST                    3 VAULT 3                    30
  0 MAIN LIBRARY                   VLT104   4    1   RONNS.VLT1.TEST                    4 VAULT 4                    40
  0 MAIN LIBRARY                   VLT105   5    1   RONNS.VLT1.TEST                    5 VAULT 5                    50
  0 MAIN LIBRARY                   VLT201   1    1   RONNS.VLT2.TEST                    5 VAULT 5                    51
  0 MAIN LIBRARY                   VLT202   2    1   RONNS.VLT2.TEST                    4 VAULT 4                    41
  0 MAIN LIBRARY                   VLT203   3    1   RONNS.VLT2.TEST                    3 VAULT 3                    31
  0 MAIN LIBRARY                   VLT204   4    1   RONNS.VLT2.TEST                    2 VAULT 2                    21
  0 MAIN LIBRARY                   VLT301   1    1   RONNS.VLT3.TEST                    1 VAULT 1                     2
  0 MAIN LIBRARY                   VLT302   2    1   RONNS.VLT3.TEST                    2 VAULT 2                    22
  0 MAIN LIBRARY                   VLT303   3    1   RONNS.VLT3.TEST                    3 VAULT 3                    32
  0 MAIN LIBRARY                   VLT304   4    1   RONNS.VLT3.TEST                    4 VAULT 4                    42
  0 MAIN LIBRARY                   VLT305   5    1   RONNS.VLT3.TEST                    5 VAULT 5                    52
  0 MAIN LIBRARY                   VLT401   1    1   RONNS.VLT4.TEST                    1 VAULT 1                     3
  0 MAIN LIBRARY                   VLT402   2    1   RONNS.VLT4.TEST                    2 VAULT 2                    23
  0 MAIN LIBRARY                   VLT403   3    1   RONNS.VLT4.TEST                    3 VAULT 3                    33
  0 MAIN LIBRARY                   VLT404   4    1   RONNS.VLT4.TEST                    4 VAULT 4                    43
  0 MAIN LIBRARY                   VLT405   5    1   RONNS.VLT4.TEST                    5 VAULT 5                    53
  0 MAIN LIBRARY                   VLT501   1    1   RONNS.VLT5.TEST                    5 VAULT 5                    54
  0 MAIN LIBRARY                   VLT502   2    1   RONNS.VLT5.TEST                    4 VAULT 4                    44
  0 MAIN LIBRARY                   VLT503   3    1   RONNS.VLT5.TEST                    3 VAULT 3                    34
  0 MAIN LIBRARY                   VLT504   4    1   RONNS.VLT5.TEST                    2 VAULT 2                    24
  0 MAIN LIBRARY                   VLT505   5    1   RONNS.VLT5.TEST                    1 VAULT 1                     4
TSIDVLT                                   VAULT MOVEMENT REPORT
DATE 03/09/09                                                                                          TIME 14 25 07
-----------T O-----------                                                               ----------F R O M---------
VAULT INFORMATION            SLOT  SER-NO  VER REEL   DATA SET NAME                     VAULT INFORMATION           SLOT
  1 VAULT 1                     1 VLT101    1    1   RONNS.VLT1.TEST                    0 MAIN LIBRARY                0
  1 VAULT 1                     2 VLT301    1    1   RONNS.VLT3.TEST                    0 MAIN LIBRARY                0
  1 VAULT 1                     3 VLT401    1    1   RONNS.VLT4.TEST                    0 MAIN LIBRARY                0
  1 VAULT 1                     4 VLT505    5    1   RONNS.VLT5.TEST                    0 MAIN LIBRARY                0
TSIDVLT                                   VAULT MOVEMENT REPORT
DATE 03/09/09                                                                                          TIME 14 25 07
-----------T O-----------                                                               ----------F R O M---------
VAULT INFORMATION            SLOT  SER-NO  VER REEL   DATA SET NAME                     VAULT INFORMATION           SLOT
  2 VAULT 2                    20 VLT102    2    1   RONNS.VLT1.TEST                    0 MAIN LIBRARY                0
  2 VAULT 2                    21 VLT204    4    1   RONNS.VLT2.TEST                    0 MAIN LIBRARY                0
  2 VAULT 2                    22 VLT302    2    1   RONNS.VLT3.TEST                    0 MAIN LIBRARY                0
  2 VAULT 2                    23 VLT402    2    1   RONNS.VLT4.TEST                    0 MAIN LIBRARY                0
  2 VAULT 2                    24 VLT504    4    1   RONNS.VLT5.TEST                    0 MAIN LIBRARY                0
TSIDVLT                                   VAULT MOVEMENT REPORT
DATE 03/09/09                                                                                          TIME 14 25 07
-----------T O-----------                                                               ----------F R O M---------
VAULT INFORMATION            SLOT  SER-NO  VER REEL   DATA SET NAME                     VAULT INFORMATION           SLOT
  3 VAULT 3                    30 VLT103    3    1   RONNS.VLT1.TEST                    0 MAIN LIBRARY                0
  3 VAULT 3                    31 VLT203    3    1   RONNS.VLT2.TEST                    0 MAIN LIBRARY                0
  3 VAULT 3                    32 VLT303    3    1   RONNS.VLT3.TEST                    0 MAIN LIBRARY                0
  3 VAULT 3                    33 VLT403    3    1   RONNS.VLT4.TEST                    0 MAIN LIBRARY                0
  3 VAULT 3                    34 VLT503    3    1   RONNS.VLT5.TEST                    0 MAIN LIBRARY                0
TSIDVLT                                   VAULT MOVEMENT REPORT
DATE 03/09/09                                                                                          TIME 14 25 07
-----------T O-----------                                                               ----------F R O M---------
VAULT INFORMATION            SLOT  SER-NO  VER REEL   DATA SET NAME                     VAULT INFORMATION           SLOT
  4 VAULT 4                    40 VLT104    4    1   RONNS.VLT1.TEST                    0 MAIN LIBRARY                0
  4 VAULT 4                    41 VLT202    2    1   RONNS.VLT2.TEST                    0 MAIN LIBRARY                0
  4 VAULT 4                    42 VLT304    4    1   RONNS.VLT3.TEST                    0 MAIN LIBRARY                0
  4 VAULT 4                    43 VLT404    4    1   RONNS.VLT4.TEST                    0 MAIN LIBRARY                0
  4 VAULT 4                    44 VLT502    2    1   RONNS.VLT5.TEST                    0 MAIN LIBRARY                0
TSIDVLT                                   VAULT MOVEMENT REPORT
DATE 03/09/09                                                                                          TIME 14 25 07
-----------T O-----------                                                               ----------F R O M---------
VAULT INFORMATION            SLOT  SER-NO  VER REEL   DATA SET NAME                     VAULT INFORMATION           SLOT
  5 VAULT 5                    50 VLT105    5    1   RONNS.VLT1.TEST                    0 MAIN LIBRARY                0
  5 VAULT 5                    51 VLT201    1    1   RONNS.VLT2.TEST                    0 MAIN LIBRARY                0
  5 VAULT 5                    52 VLT305    5    1   RONNS.VLT3.TEST                    0 MAIN LIBRARY                0
  5 VAULT 5                    53 VLT405    5    1   RONNS.VLT4.TEST                    0 MAIN LIBRARY                0
  5 VAULT 5                    54 VLT501    1    1   RONNS.VLT5.TEST                    0 MAIN LIBRARY                0
```

TVTO Report

TVTO displays VTOC reports online.

Note: For information on producing this display, see Online VTOC Displays (TVTO) (see page 375).

Fields

FILE NAME is the name of the file occupying the extent. This can also be free space.

LOW is the starting address of the extent.

SIZE is the size of the extent.

C is the type of file: sequential, VSAM, direct access, etc.

SEQ is the sequence number of the extent.

CREATION is the date the file was created.

EXPIRATION is the expiration date on the VTOC.

Example

```
VOLUME TABLE OF CONTENTS FOR VOL=EPC360 ON 360   10/258 11:41:54 PAGE   1 OF   2
________ FILE NAME ____________________________      LOW     HIGH C SEQ   CRE   EXP
                                FREE.SPACE 00000-01 00017-02
)NKNOC002.CAT.FILE                G=000000 00017-03 00017-03 ? 000 10041*10041
                                FREE.SPACE 00017-04 00926-14
DOS.PAGING.FILE.FF61008A2097                00927-00 01291-14 ? 000 10256+99366
                                FREE.SPACE 01292-00 03022-09
TECHEPIC.CICS23.USER.JOURNAL.B              03022-10 03022-14 S 000 10258*10258
TECHEPIC.CICS23.USER.JOURNAL.A              03023-00 03023-09 S 000 10258*10258
TECHEPIC.CICS23.SYSTEM.LOG.B                03023-10 03023-14 S 000 10258*10258
TECHEPIC.CICS23.SYSTEM.LOG.A                03024-00 03024-09 S 000 10258*10258
TECHEPIC.CICS23.DUMP.DATASET.B              03024-10 03026-09 S 000 10133*10133
TECHEPIC.CICS23.DUMP.DATASET.A              03026-10 03028-09 S 000 10258*10258
TECHEPIC.CICS23.DFHBUXT                     03028-10 03030-09 S 000 10133*10133
TECHEPIC.CICS23.DFHAUXT                     03030-10 03032-09 S 000 10133*10133
                                FREE.SPACE 03032-10 03034-09
TECHEPIC.CICSTS.DFHAUXT                     03034-10 03036-09 S 000 10111*10111
TECHEPIC.CICSTS.DUMP.DATASET.B              03036-10 03038-09 S 000 10111*10111
TECHEPIC.CICSTS.DUMP.DATASET.A              03038-10 03040-09 S 000 10258*10258
ICCF.BKUP                                   03040-10 03074-09 S 000 10111*10111
TECHEPIC.VSE.DUMP.LIBRARY                   03074-10 03114-09 S 000 10112+99366
TECHEPIC.INFO.ANALYSIS.EXT.RTNS.FILE        03114-10 03114-10 S 000 10112+99365
                                FREE.SPACE 03114-11 03114-14
PF:  1=NEXT 2=PREV 3=FIRST 4=LAST 5=REFRESH     39897 FREE TRACKS IN   5 EXTENTS
```

Appendix B: Supported Tape Modes

Cartridge tape modes/densities supported by CA EPIC for z/VSE:

Drive Type	Buffered Uncompacted	Buffered Compacted*	Unbuffered Uncompacted	Unbuffered Compacted	Tracks	Encryption Capable?
3480/3490	00	08	20	28	18	No
3490E	F0	F8	-	-	36	No
3590	10M	18M	30M	38M	128	No
3590E	10E	18E	30E	38E	256	No
3590H	10H	18H	30H	38H	384	No
3592	10W	18W	30W	38W	512	No
3592-E05	10W5	18W5	30W5	38W5	896	No
3592-E05	13W5	1BW5	33W5	3BW5	896	Yes
3592-E06	10W6	18W6	30W6	38W6	1152	No
3592-E06	13W6	1BW6	33W6	3BW6	1152	Yes

* Typically provides the best performance.

Modes F0 and F8 are used internally by CA EPIC for z/VSE to distinguish 3490E tape drives from 3480/3490. Because VSE does not support these modes directly, CA EPIC sends mode 00 or 08 to the 3490E hardware whenever modes F0 or F8 are specified.

3592-E05 and 3592-E06 modes 13nn, 1Bnn, 33nn, and 3Bnn, used in conjunction with a valid KEKL (Key Encryption Key Label), enable hardware tape encryption. See the section titled ‘Tape Encryption‘ in Chapter 12 for more information on encrypting tapes.

Reel tape modes/densities supported for 3410/3420/3422/3430 devices:

Mode/Density	Bytes Per Inch (BPI)
DO	6250
CO	1600
C8	800

Other older tape devices may still be supported. Contact CA Technical Support for information on devices not listed here.

Index

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z