DFSMSdfp Storage Administration (SC23-6860)

Chapter 17. Protecting the Storage Management Subsystem

Storage administration (STGADMIN) profiles in the FACILITY class or XFACILIT class

Command and keyword related properties

STGADMIN.SMS.ALLOW.DATASET.SEQ.ENCRYPT
…

STGADMIN.SMS.DADSM.UNMAP.PREFER
prevents access to residual data by releasing extents and unmapping media blocks when a data set is deleted, or space is released. If the device does not support this facility, residual data is erased by overwriting with zeros. SMS-managed, non-SMS managed, and temporary data sets are supported. A data set is eligible for UNMAP when erase is requested using RACF command SETROPTS ERASE or an ERASE keyword is specified at allocation time or included as a parameter to DADSM SCRATCH or PARTREL.

Authorize access to the DADSM UNMAP facility. Do this by defining a profile in the FACILITY class:
RDEFINE FACILITY STGADMIN.SMS.DADSM.UNMAP.PREFER UACC(NONE)
and refresh the CLASS facility:
SETROPTS RACLIST(FACILITY) REFRESH

DFSMS Using Data Sets (SC23-6855)

Chapter 5. Protecting Data Sets

Erasure of Residual Data
When you release media space, you can erase your data.

Writing and Erasing Data on HDD vs SSD

When you write to a data set that resides on a hard disk drive (HDD), whatever data that previously existed is replaced with the new write data. Data written to HDDs are saved in fixed-size sectors and these sectors can be rewritten repeatedly. When you delete a data set or release space, the data remains accessible on the HDD. The only way to prevent access to residual data on HDDs is to overwrite the sectors.

Solid state drives (SSD) work differently. The media is arranged in fixed-size (4 kilobyte) pages, and pages are grouped into blocks. Writes are only permitted to a blank page. Pages can never be overwritten. When you write a flash device, the device must allocate a new, blank page for the new data and the new page is noted as the current location in the logical block address (LBA) table. The original page is marked as invalid.

When you delete a data set the data remains accessible on the SSD. You could overwrite with zeros; however, this would result in a potentially large data transfer and cause the device to allocate additional pages. This is not desirable because flash devices have a finite number of “program/erase” (P/E) cycles before they need to be replaced.

Flash drives provide an UNMAP command which is a more efficient solution to prevent access to residual data. UNMAP is a SCSI command that mark pages as invalid. It requires no allocation of SSD pages and zero data transfer. By marking the pages as invalid, they cannot be read by any program running under control of an IBM operating system.

Periodically, the device erases an entire block of invalid pages which makes these pages available for new writes.

Erasing DASD Data
<last paragraph>

Prior to z/OS V2R1, EOS issued one channel program to erase on track. This process repeated until all tracks are erased. If the data set is large, this process could cause a large number of I/O requests. Starting in V2R1, the system can erase up to 255 tracks in a single channel program.

Unmapping DASD Data
DADSM UNMAP prevents access to residual data by having the system instruct the device to guarantee all tracks in the data set are initialized. After the channel program completes, space is released and media space is unmapped. The combination of these actions ensures that residual data cannot be read from a host system.

The system programmer can enable the DADSM UNMAP feature with the RACF FACILITY class profile STGADMIN.SMS.DADSM.UNMAP.PREFER. When this profile is defined, the system will first attempt to use DADSM UNMAP when a data set is scratched or space is released, and the data set has been identified as needing erasure (e.g., using RACF SETROPTS ERASE). If the device is unable to guarantee tracks are initialized, the system will erase residual data by overwriting with zeros.

Temporary data sets do not need to be identified as needing erasure. If the DADSM UNMAP FACILITY class profile is defined, the system will automatically instruct the device to guarantee tracks are initialized. If this is not possible, the data remains accessible.

z/OS Security Server RACF System Programmer's Guide (SA23-2287)

Chapter 2. Performance considerations

Protecting data during its life cycle
Providing protection of data requires protecting data throughout its lifecycle. In addition to the access controls provided by RACF and DFSMS, IBM provides four mechanisms to provide additional levels of protection for DASD data sets:

- **"Encryption of data at rest" on page xx**
- **"DFSMS considerations" on page xx**
- **"Erase-on-Scratch with overwrite" on page xx**
- **"DADSM UNMAP" on page xx**

These mechanisms are intended to address the threat of:

Table 1. Mechanisms intended to address threats to data during its life cycle

Threat	Addressed by Encryption of Data at Rest	Addressed by Erase on Scratch and DADSM UNMAP
The loss of physical control of the media, either through malevolent actions (such as theft) or non-malevolent actions (such as maintenance).	Yes	No
The movement of data within the storage device, such as for data recovery or performance	Yes	No
Residual data being read after deletion	No	Yes

DADSM UNMAP

DADSM UNMAP prevents access to residual data by instructing the device to guarantee tracks are initialized. After the channel program completes, space is released for full extents and media space is unmapped. The combination of these actions ensures that residual data cannot be read from a host system.

Both EOS (overwrite with zeros) and DADSM UNMAP prevent access to residual data. This protection means that no one can allocate a new data set at the same location, open it for input, and read old data.

Unlike overwriting with zeros, DADSM UNMAP should not have a material impact on system performance. For more information refer to Writing Data on HDD vs SSD in publication DFSMS Using Data Sets.

You can specify when the system should use EOS or DADSM UNMAP. To continue using EOS, no changes are required.

To have the system use DADSM UNMAP, define the FACILITY class profile for DADSM UNMAP as described in DFSMSdfp Storage Administration. Any data set requested to be erased via SETROPTS ERASE, JCL parameter, are eligible.

Interplay between FACILITY class profile and SETROPTS ERASE

Use Case	RACF SETROPTS ERASE specified	DADSM UNMAP FACILITY class profile defined	System Behavior
No data erasure	No	No	Default today
Erase by overwriting residual data with zeros	Yes	No	No change to EOS behavior
Release space and unmap media for data sets specified using SETROPTS ERASE	Yes	Yes	Better performance solution using DADSM UNMAP. Overwrite with zeros if DADSM UNMAP is not supported.
Release space and unmap media only for temporary data sets	No	Yes	Better solution for temporary data sets No overwrite if DADSM UNMAP is not supported.