

IBM Education Assistance for z/OS V2R2

Item: GDG > 255 GDSs
Catalog RAS & GDG SCRATCH/NOSCRATCH support
Modify Command Security Enhancements
GDG PURGE option
Component: Catalog



Agenda

- Trademarks
- Presentation Objectives
- Overview
- Usage & Invocation
- Interactions & Dependencies
- Migration & Coexistence Considerations
- Installation
- Presentation Summary
- Appendix



Presentation Objectives

Discuss the external changes for new Catalog V2R2 enhancements:

- GDG Enhancements
 - GDG > 255 GDSs
 - GDG SCRATCH/NOSCRATCH support
 - GDG PURGE option
- Catalog RAS
 - New CSI fields (in support of the private CAXWA chain)
 - Connect Multi-Line WTOs
 - Catalog Attributes Healthcheck
 - DSS Restore a Catalog to Any Volume
- Modify Command Security Enhancements



Overview: GDG updates

- Problem Statement / Need Addressed
 - 1. Increase the number of GDSs in a GDG beyond 255
 - 2. Provide a default/override for the GDG SCRATCH parmlib variable
 - 3. Provide a method to delete unexpired GDSs during rolloff
- Solution
 - 1. New EXTEND keyword on DEFINE GDG allows for up to 999 active GDSs
 - 2. GDGSCRATCH parmlib variable overrides IDCAMS keyword on GDG DEFINE
 - 3. PURGE keyword added to IDCAMS DEFINE GDG
- Benefit / Value
 - 1. Allows for more than a year's worth of active GDSs
 - 2. Allows system administrator to set default to SCRATCH
 - 3. Allows user to specify what happens to unexpired GDSs during rolloff



Usage & Invocation: GDG Extended (GDGE)

- New keyword for DEFINE GDG
 - EXTENDED keyword allows up to 999 GDS to be active within 1 GDG
 - Example:
 DEFINE SMS.GDG GDG EXTENDED LIMIT(999)
 - Default is NOEXTENDED LIMIT is 255
 - Abbreviations: EXT for extended, NEXT for noextended
- GDGE support must be enabled for a system
 - SYS1.PARMLIB member IGGCATxx must have new GDGEXTENDED variable set to “YES” (default is “NO”)
 - Example:
 GDGEXTENDED(YES)
 - All systems in an installation should be at z/OS V2.2 or higher before enabling GDGEs
 - Coexistence APAR OA46009 prevents access to GDGEs on lower level systems
 - No coexistence can give unpredictable results



Usage & Invocation: GDG SCRATCH Default Parmlib Variable

- New variable for SYS1.PARMLIB member IGGCATxx - GDGSCRATCH
 - Specified as GDGSCRATCH(YES):
 - Causes Catalog to set attribute to SCRATCH in GDG catalog record regardless of the setting in the DEFINE GDG statement
 - Default is “NO”
 - SCRATCH specifies a generation data set (GDS) is to be deleted from disk when uncataloged from the GDG for EMPTY|NOEMPTY processing
 - Does not change original setting once attribute is set in DEFINE GDG
 - Allows system administrator to override IDCAMS DEFINE GDG setting to SCRATCH
 - If not set or parmlib variable is GDGSCRATCH(NO), the keyword specified on the DEFINE GDG takes precedence
- Also applies to GDGEs



Usage & Invocation: GDG PURGE Keyword

- New keyword PURGE for DEFINE and ALTER GDG that specifies that when the SCRATCH is set, unexpired generation data sets (GDSs) are to be deleted during roll off processing
 - Example:
DEFINE SMS.GDG GDG EXTENDED LIMIT(999) PURGE
SCRATCH
 - ALTER SMS.GDG1 PURGE (for existing GDGs)
 - NOPURGE is the default
- New variable for SYS1.PARMLIB member IGGCATxx – GDGPURGE overrides DEFINE GDG setting when the PURGE attribute is set
 - Specified as GDGPURGE(YES)
 - Only used when SCRATCH is also set
 - Default is GDGPURGE(NO)
- Also applies to GDGEs



Migration & Coexistence Considerations: GDG updates

- APAR OA46009 provides limited coexistence by failing access to a GDG EXTENDED (GDGE) or GDSs in a GDGE on pre-z/OS 2.2 systems
- Failure to have OA46009 will produce unpredictable results if access to a GDGE or GDS in a GDGE is attempted from a pre-z/OS 2.2 system
- PTFs for OA46009:
 - UA75975 HDZ1D10
 - UA75976 HDZ2210
- It is strongly advised that all systems be at z/OS V2R2 or higher before enabling GDGE support
- GDG PURGE support is not available on pre-z/OS V2R2 systems although it may have been set using a z/OS V2R2 system to DEFINE the GDG
 - PURGE setting simply ignored



Installation: GDG updates

- For GDG EXTENDED

- We recommend all systems be a z/OS V2R2 or higher. If not, apply APAR OA46009 to pre-z/OS V2R2 systems.
- Enable by adding GDGEXTENDED(YES) to SYS1.PARMLIB(IGGCATxx)

- For GDG SCRATCH Default

- Add GDGSCRATCH(YES) to SYS1.PARMLIB(IGGCATxx) if desired system wide

- For GDG PURGE

- Add PURGE to DEFINE GDG
- ALTER GDG to PURGE
- Add GDGPURGE(YES) to SYS1.PARMLIB(IGGCATxx) if desired system wide



Overview: Catalog RAS – New CSI Fields

- Problem Statement / Need Addressed:
 - The Catalog Address Space (CAS) uses internal control blocks called CAXWAs to process catalogs. The CAXWAs are in common storage and when they are damaged an IPL is the only way to fix it.
 - Some vendors use the CAXWAs to retrieve information about a catalog.
- Solution:
 - We have created CAXWAs in CAS private storage.
 - Vendors can obtain information about a catalog from new fields supported by the Catalog Search Interface (CSI).
- Benefit / Value:
 - A CAS restart can now be used to repair broken CAXWAs.
 - Dynamic catalog data is available to all thru a supported user interface.



Usage & Invocation: Catalog RAS – New CSI Fields

- The CSI interface is documented in Chapter 11 of the *DFSMS Managing Catalogs* manual.
- A new entry type of “Z” (used in the CSIDTYPES field) is used to tell the CSI that the information desired is dynamic data from CAS control blocks rather than information stored in the catalog.
- No other types are allowed to be specified with the Z entry type
- The master catalog name is returned for every Z entry request
- For Z entry requests, the catalog name is specified in the filter key field (CSIFILTK); the catalog name field (CSICATNM) should be blank



Usage & Invocation: Catalog RAS – New CSI Fields (con't)

The new supported fields are described below:

- CATACT: A 2-byte field containing the catalog activity count (the number of address spaces that have the catalog allocated)
- CATFLAGS: A 2-byte field containing flags that indicate the following:
 - Catalog is open
 - Catalog is the master catalog on this system
 - Catalog is active in In-Storage Cache
 - Catalog is active in VLF
 - Catalog is active in ECS
 - Catalog is open in RLS mode
 - Catalog is deleted
 - Catalog is locked
- CATUCB: A 4-byte field containing the address of the UCB for the volume the catalog resides on



Usage & Invocation: Catalog RAS – New CSI Fields (con't)

Sample Invocation:

```
/* Initialize variables */

CSIFLTK = CatName;                ! Name to be found
CSICATNM = ' ';                  ! Blank the catalog name
CSIRESNM = ' ';                  ! Blank the resume name
CSIDTYP = 'Z';                   ! CB data request
CSICLDI = ' ';                  ! Don't return index
CSIRESUM = ' ';                  ! Not in resume mode
CSIS1CAT = ' ';                  ! Search all cats
CSIOPTNS = 'F';                  ! Use fullword lengths
CSINUMEN = 3;                    ! Retrieve 3 fields
CSIENTS(1) = 'CATACT';           ! Set 1st field name
CSIENTS(2) = 'CATUCB';           ! Set 2nd field name
CSIENTS(3) = 'CATFLAGS';         ! Set 3rd field name
CSIUSRLN = Length(L_Work_Area);
Call IGGCSI00(L_Reason_Area,L_Input_Area,L_Work_Area)
    Retcode(L_CSI_Retcode);

/* Process output here */

Return Code(L_Retcode);
```



Overview: Catalog RAS – Connect Multi-Line WTOs

- Problem Statement / Need Addressed:

Customers requested that large Catalog multi-line operator messages (>255 lines) that were being cut off, be allowed to display to completion.

- Solution:

The behavior was actually a defect for only the MODIFY CATALOG,ALLOCATED command output. It has been fixed.

- Benefit / Value:

- All allocated/open catalogs now appear in the MODIFY CATALOG,ALLOCATED command output.



Usage & Invocation: Catalog RAS – Connect Multi-Line WTOs

- The Catalog Modify command is documented in Chapter 8 of the *DFSMS Managing Catalogs* manual.
- The following sample operator command shows how to display a list of all catalogs currently allocated on the system:

```
F CATALOG,ALLOCATED
```



Overview: Catalog RAS - Catalog Attributes Healthcheck

- Problem Statement

- By default, catalogs are defined with SHAREOPTIONS(3 4). A shared catalog must be defined with SHAREOPTIONS(3 4) and must reside on a shared volume. Catalogs that reside on shared volumes will become damaged if referred to by another system and the shareoptions are inconsistently set i.e not (3 4).

- Solution

- This line item will provide a new Catalog Health Check designed to inspect all the catalogs defined in the environment for shareoptions and volume status (shared or non-shared) and report any inconsistencies between the two. System programmers can decide to redefine or alter the inconsistent catalog(s) with the correct shareoptions during system downtime.

- Benefit / Value

- Damage to incorrectly shared catalogs can now be avoided. Further, for catalogs on non-shared devices, the shareoptions can be set to (3,3), which will provide for more efficient access and better overall performance.



Usage & Invocation: Catalog RAS - Catalog Attributes Healthcheck

- Once the IBM Health Checker for z/OS is up and running, an exit routine (IGG0CLHX) adds the new check to the system. If the check is successfully added, it will start running at the interval specified in the exit routine unless it is disabled or the interval is modified. The default interval for the check to run is every 720 hours (30 days). This interval can be changed using the F HZSPROC command.
- For a detailed list of health check commands and how to run, start, stop, activate, deactivate, enable, disable, delete, refresh, modify and update a specific health check, refer to *Chapter 4 Managing Checks* 'IBM Health Checker for z/OS User's Guide Manual'.



Interactions & Dependencies: Catalog RAS - Catalog Attributes Healthcheck

- Software Dependencies
 - The Health Checker Address space for z/OS must be started for the catalog health check to run.
- Hardware Dependencies
 - None



Overview: DFSMSdss Restore of a User Catalog to Any Volume

▪ Problem Statement

- DFSMSdss only allows a user catalog to be restored to the same volume from which it was dumped
 - Volsers of source volume where catalog was dumped from must match restore target volume

▪ Solution

- Allows a user catalog to be restored to any volume
 - Device capacity must match (3380 vs 3390)
 - Applies to logical restore
 - Restriction for physical data set restore remains

▪ Benefit / Value

- Makes recovering a user catalog easier
- When a user catalog is backed up there could be a point where the user catalog is moved to a different volume (volser)



Usage & Invocation: DFSMSdss Restore of a User Catalog to Any Volume

- This function only affects logical data set Restore of ICF Catalogs



Overview: Catalog Modify Command Security Enhancement

- Problem Statement

- Clients have indicated that they would like to be able to provide some people with access to the Catalog Address Space reporting commands without also allowing those same people to issue commands which might alter the catalog configuration. For example, a user via SDSF might be able to issue an "F CATALOG,ALLOCATED" command, but not be able to issue an "F CATALOG,CLOSE(usercat)" command.

- Solution

- This line item will provide a new RACF resource profile that will be checked by catalog command processing to insure that the issuer of a command is authorized to execute that command. This will allow users utilizing SDSF to issue operator reporting commands, the results of which they can view at their terminal, but not permit them to issue commands that change the catalog configuration.

- Benefit / Value

- The Catalog reporting capabilities inherent in the Modify Catalog Command can now be provided to analysts and others needing to monitor the catalog environment while at the same time insuring that accidental or malicious changes to catalog configuration and operation are protected against.



Usage & Invocation: Catalog Modify Command Security Enhancement

- Usage of this facility continues to be provided by the MVS Modify Command interface, and more specifically the MODIFY CATALOG command as Documented in the publication *DFSMS Managing Catalogs*.
- It is anticipated that users with READ authority to the Modify Catalog Command would likely be using the MVS Command capability provided in SDSF, using the documented leading '/' to invoke specific commands.
- For example:

`/F CATALOG,ALLOCATED`

would display the catalogs currently allocated to the system being displayed.



Interactions & Dependencies: Catalog Modify Command Security Enhancement

- Software Dependencies
 - RACF or comparable security product
 - SDSF if command entry through that vehicle is desired
- Hardware Dependencies
 - None
- Exploiters
 - None



Installation: Catalog RAS – Catalog Modify Command Security Enhancement

- The only requirement for activating this facility is to define a new OPERCMDS Resource profile:

MVS.MODIFY.STC.CATALOG.CATALOG.SECURE

and then PERMIT users and operators (as needed) to this resource via the RACF PERMIT command, where either READ or UPDATE authorization is granted to a user or user group.

- Examples of the RACF commands needed to establish the resource profile and permissions are in the publication *DFSMS Managing Catalogs*.



Presentation Summary

- GDG Enhancements
 - GDG > 255 GDSs
 - GDG SCRATCH/NOSCRATCH support
 - GDG PURGE option
- Catalog RAS
 - New CSI fields (in support of the private CAXWA chain)
 - Connect Multi-Line WTOs
 - Catalog Attributes Healthcheck
 - DSS Restore a Catalog to Any Volume
- Modify Command Security Enhancements



Appendix (1 of 2)

▪ Publications:

- *DFSMS Managing Catalogs* (SC23-6853-02)
- *DFSMS Access Method Services Commands* (SC23-6846-02)
- *IBM Health Checker for z/OS User's Guide* (SC23-6843-03)
- *z/OS DFSMSdss Storage Administration* (SC23-6868)



Appendix (2 of 2)

▪ Terminology

- GDG: Generation Data Group. Can have 1 to 255 GDSs, aka GDG Classic
- GDGE: GDG Extended. Can have 1 to 999 GDSs
- GDS: Generation Data Set. Data set whose generations are managed by a GDG or GDGE.
- Roll off processing: What happens to a GDS or GDSs when it or they are uncataloged (removed) from a GDG or GDGE.
-

