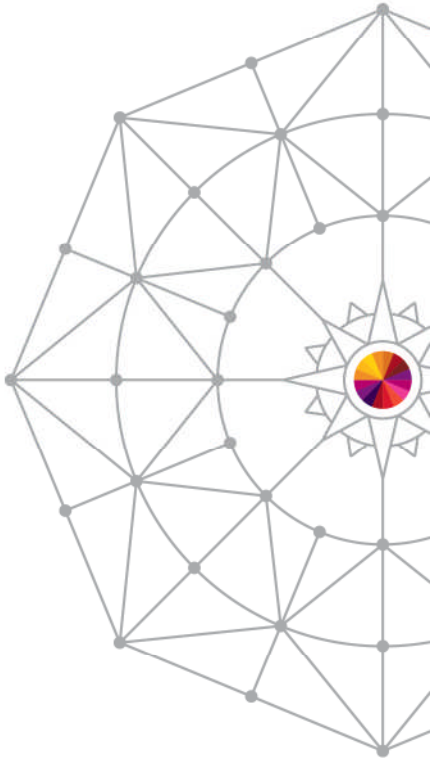
DFSMS Security: Unveiling the Inner Truth



Chris Taylor (IBM Corporation)
Marty Hasegawa(Rocket Software)

August 7, 2014 – 1:30pm
Session Number 15767

Copyright (c) 2014 by SHARE Inc. Except where otherwise noted, this work is licensed under http://creativecommons.org/licenses/by-nc-sa/3.0/

Legal Disclaimer

NOTICES AND DISCLAIMERS

Product information and data has been reviewed for accuracy as of the date of initial publication. Product information and data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or programs(s) described herein at any time without notice.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Consult your local IBM representative or IBM Business Partner for information about the product and services available in your area.

Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectually property rights, may be used instead. It is the user's responsibility to evaluate and verify the operation of any non-IBM product, program or service.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. IBM is not responsible for the performance or interoperability of any non-IBM products discussed herein.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not necessarily tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Trademarks

The following are trademarks of the *International Business Machines Corporation:*

IBM, DFSMS/MVS, DFSMShsm, DFSMSrmm, DFSMSdss, DFSMSopt, DFSMS Optimizer, z/OS, eServer, zSeries, MVS, FlashCopy®

The information contained in this presentation is distributed on an 'AS IS' basis without any warranty
either expressed or implied, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose. The use of this information is a customer responsibility and depends on the customer's ability to evaluate and integrate it into the customer's operational environment.

Session Abstract

This is a follow-on to a session presented at Share in Anaheim. Both Storage and Security Administrators share a common concern regarding access to sensitive data. In the DFSMS arena, some protection mechanisms often remain overlooked and unexploited.

This session will describe the various resource classes and how they can be effectively established and ensure compliance. The presenters will also discuss how RACF can be leveraged to monitor permissions to storage management functions.

Agenda

- Security roles
- Storage Administration & User tasks
- SAF & RACF Facility classes
- Some product definitions
  - DFSMSdss
  - DFSMShsm
- Setting up group auditing
- Monitoring access to resources
- Additional DFSMSdfp and other considerations

Thanks to…..

- Previous presentations on some of the topics
  - Tony Pearson
  - Ed Baker
  - Tom Conley

Reason for this presentation

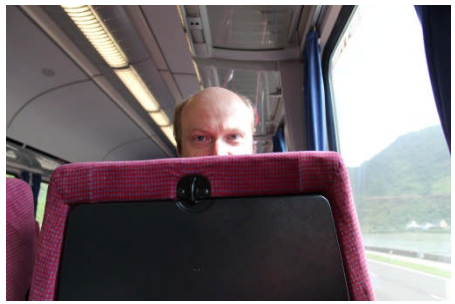
- Presentation at Share in Anaheim
  - How to Protect the z/OS Storage Environment from Prying Eyes and Still Get Your Work Done
    - Session 15071
- Questions that were asked during the presentation
  - “What can I do as a storage administrator to assist with compliance?”
    - *“How can I monitor this?”*
  - “How can we ensure that storage administration and users have **sufficient** access to data?”
- The presentation will attempt to look at security requirements from a storage administrator’s perspective

Role of the Security Administrator (partial)

- Security administrators are responsible for maintaining and enforcing the corporate security policy
- Ensure that data is only accessible to those who require access
  - PHI, SPI, etc.
- Ensure that systems can only be accessed by those who need access
- Maintain an audit trail to ensure compliance if requested for auditing purposes

Roles of the various departments

- Security will (and should) always be a primary concern in any organization
- Ideally only allow enough access for the users to be able to perform their function effectively
- Setup security access in such a way that tweaks do not always require major paperwork
  - Effectively integrate new products or processes without major upheaval.
- Nowadays, beware of the internal hacker



Storage Administrator Tasks

- Storage Admin typical day
  - Backup and Recover data sets
  - Move data sets from one volume to another
  - Full disk volume functions
  - Copy data sets to a new name
  - Report on data sets and environment
  - Resize data sets
  - Make changes to SMS constructs
- What access does the Storage Admin need to perform these tasks?

User Tasks

- User typical day (from a storage perspective)
- Recall datasets
  - Batch or TSO recalls
  - Backup and Recover data sets
  - Copy data sets to a new name
  - Report on data sets
    - Maybe?

- What access levels does the user need to perform these tasks?

Traditional methods - Data set level access

- Grant ALTER access to all data sets in the environment?
  - Allows access to data set contents
  - Requires maintaining a large number of access lists
  - Missing profile access could hinder work
  - Auditors will not probably not like this level of access to all data sets

- “Administrators need to treat data like CARGO, and do not need access to information inside”

Traditional methods - Volume level checking

- Allows defined users to perform maintenance tasks without having access to data set profiles
- DASDVOL class in RACF
- Different function keywords might require different access levels
  - DUMP with DELETE vs. DUMP without DELETE
- Used as well with ICKDSF and IDCAMS ALTER
- Does not work with SMS volumes
  - Big Problem!
  - Use OPERATIONS instead

Traditional methods – OPERATIONS Attribute

- Allows access to most storage admin functions
- Also allows access to data sets and DASDVOL
- SETROPTS OPERAUDIT can be used to audit users
- OPERATIONS will allow access as long as no access list counteracts it.
  - e.g. ALTER access required but Storage Admin is defined with READ access
- Need is reduced through the use of facility classes

RACF Facility Class

- Calls are made using System Authorization Facility (SAF)
  - CA-Top Secret and CA-ACF2 support SAF calls
- Security Administrators using other products generally know how to translate rules for their environment
- The examples provided are based on RACF definitions
  - Not every individual profile is described in this presentation
- Assumes Enhanced Generic Naming is in use
  - E.g. dataset.**

RACF Facility Class

- Referenced when an action takes place
- Typically used by program products or components
- Commands for Facility Classes

```
SETROPTS RACLIST(FACILITY)

RDEFINE FACILITY STGADMIN.x.y.z    UACC(NONE)

PERMIT STGADMIN.x.y.z CLASS(FACILITY)    USER(userid) ACCESS(read)

RALTER FACILITY STGADMIN.x.y.z    GLOBALAUDIT(SUCCESS)
```

- For most of the STGADMIN.** profiles, READ access is sufficient

STGADMIN.**

- Profiles for DFSMS functions begin with STGADMIN.**
  - Some vendors also use this format for their checks
- Calls are made using System Authorization Facility (SAF)
  - CA-Top Secret and CA-ACF2 support SAF calls
- Some examples
  - STGADMIN.IDC.**
  - STGADMIN.IGG.**
  - STGADMIN.ADR.**
  - STGADMIN.ARC.**
  - STGADMIN.EDG.**
  - STGADMIN.IGD.**
  - STGADMIN.ICK.**

DFSMSdss Facility Classes

DFSMSdss for Storage Administrators

- DFSMSdss allows for Storage Administrators to process data (“CARGO”) without individual data set checking
- RACF Facility Class must be active
- Facility Class Profiles must be defined
  - STGADMIN.ADR.STGADMIN.** command
- Userid must have READ access to profile
- Userid must specify ADMIN keyword on batch job

```
COPY DATASET(INCLUDE(MYDATSET)) -
     LOGINDDNAME(DASD1) OUTDDNAME(DASD2)
     DELETE CATALOG ADMIN
```

DFSMSdss for Storage Administrators

- ADMIN functions protected using profiles
  - Compress
  - Consolidate
  - Copy
  - Defrag
  - Dump
  - Print
  - Release
  - Restore

DFSMSdss for Regular Users

- Other DFSMSdss functions can be protected using RACF
- RACF Facility class needs to be active
- Profiles defined as STGADMIN.ADR.**
- Allows protection of DFSMSdss functions and certain keywords

DFSMSdss facility classes

- DUMP function
  - Additional keyword protection for
    - Concurrent copy
      - *STGADMIN.ADR.DUMP.CNCURRNT*
    - INCAT processing
      - *STGADMIN.ADR.DUMP.INCAT*
    - NEWNAMEUNCONDITIONAL
      - *STGADMIN.ADR.DUMP.NEWNAME*
    - Process SYS1 data sets
      - *STGADMIN.ADR.DUMP.PROCESS.SYS*
    - Dump without serialization
      - *STGADMIN.ADR.DUMP.TOLERATE.ENQF*

DFSMSdss facility classes

- COPY function
  - Additional keyword protection for
    - Bypass ACS routines
      - *STGADMIN.ADR.COPY.BYPASSACS*
    - Concurrent copy
      - *STGADMIN.ADR.COPY.CNCURRNT*
    - FCCGFREEZE (Source volume part of FC Consistency Grp)
      - *STGADMIN.ADR.COPY.FCFREEZE*
    - FCFASTREVERSERESTORE
      - *STGADMIN.ADR.COPY.FCFRR*
    - FCSETGTOK
      - *STGADMIN.ADR.COPY.FCSETGT*

DFSMSdss facility classes

- COPY function (cont)
  - Additional keyword protection for
    - FCTOPPRCPRIMARY
      - *STGADMIN.ADR.COPY.FCTOPPRCP*
    - Flashcopy with COPY
      - *STGADMIN.ADR.COPY.FLASHCPY*
    - INCAT with COPY
      - *STGADMIN.ADR.COPY.INCAT*
    - COPY SYS1 data sets
      - *STGADMIN.ADR.COPY.PROCESS.SYS*
    - Bypass serialization
      - *STGADMIN.ADR.COPY.TOLERATE.ENQF*

DFSMSdss facility classes

- RESTORE Function
  - Additional keyword protection for
    - Bypass ACS routines
      - *STGADMIN.ADR.RESTORE.BYPASSACS*
    - Delete catalog entry
      - *STGADMIN.ADR.RESTORE.DELCATE*
    - IMPORT with RESTORE
      - *STGADMIN.ADR.RESTORE.IMPORT*
    - Bypass serialization
      - *STGADMIN.ADR.RESTORE.TOLERATE.ENQF*

DFSMSdss generic example

```
zSecure Admin+Audit for RACF General resource overview
Command ===> _____________________________________________ Scroll===> CSR
Class FACILITY, like STGADMIN.adr.**              4 Mar 2014 14:55
   Class    Profile key                              T UACC     Owner    S/F W
__ FACILITY STGADMIN.ADR.CONSOLID                     NONE     SYS1      R  _
__ FACILITY STGADMIN.ADR.DEFRAG                       NONE     SYS1      R  _
__ FACILITY STGADMIN.ADR.DUMP.**                    G NONE     SYS1      R  _
__ FACILITY STGADMIN.ADR.*                          G NONE     IBMUSER   R  _
****************************** Bottom of Data ********************************
```

DFSMShsm facility classes

DFSMShsm for Storage Administrators

- DFSMShsm previously provided rudimentary authorization using AUTH command
  - Either Storage Admin or End-user
- SAF interface introduced for z/OS DFSMS V1R5
- Allowed more control for user access
- Storage Administrators
  - STGADMIN.ARC.command
- Storage End Users defined differently
  - STGADMIN.ARC.ENDUSER.*
- Described in DFSMShsm Implementation and Customization Guide

DFSMShsm profiles

Profile	Function
STGADMIN.*	System level storage administrator command protection. Generic profile provides default access if other DFSMShsm profiles are not defined
STGADMIN.ARC.*	DFSMShsm command protection, generic profile for all DFSMShsm commands
STGADMIN.ARC.command	DFSMShsm authorized command protection, discrete profile for specific DFSMShsm authorized command
STGADMIN.ARC.ENDUSER.*	DFSMShsm end user command protection
STGADMIN.ARC.ENDUSER.h_command	DFSMShsm end user command protection, discrete profile protects specific DFSMShsm end user command
STGADMIN.ARC.ENDUSER.h_command.parameter	Discrete profile protects specific DFSMShsm end user command with specific parameter

RACF Authorized Commands

- ***STGADMIN.ARC.**** can be used to protect all DFSMShsm authorized commands
  - User or group requires ACCESS(READ) to issue command
  - ACCESS(NONE) means that user or group can't issue command

- ***STGADMIN.ARC.command*** or ***STGADMIN.ARC.command.parameter*** can be used to restrict the use of any authorized command

RACF Storage Admin command profiles

```
zSecure Admin+Audit for RACF General resource overview
Command ===> _____________________________________________ Scroll===> CSR
Class FACILITY, like stgadmin.arc.*              4 Mar 2014 08:44
   Class    Profile key                                T UACC     Owner    S/F W
__ FACILITY STGADMIN.ARC.BACKVOL                       NONE     SYS1      R  _
__ FACILITY STGADMIN.ARC.DELVOL                        NONE     SYS1      R  _
__ FACILITY STGADMIN.ARC.EXPIREBV                      NONE     SYS1      R  _
__ FACILITY STGADMIN.ARC.**                          G NONE     SYS1      R  _
****************************** Bottom of Data ********************************
```

RACF End-user commands

- ***STGADMIN.ARC.ENDUSER.**** can be used to protect all DFSMShsm end user commands
  - User or group requires ACCESS(READ) to issue command
  - ACCESS(NONE) means that user or group can't issue command

- ***STGADMIN.ARC.ENDUSER.h_command*** *or*

STGADMIN.ARC.ENDUSER.h_command.parameter can be used to restrict the use of any end user command

- The use of end user commands will also require RACF authorization to data sets for:
  - HDELETE, HMIGRATE, HRECALL, HRECOVER, HBDELETE, HLIST, or HQUERY

RACF End-user command profiles

```
zSecure Admin+Audit for RACF General resource overview
Command ===> _____________________________________________ Scroll===> CSR
Class FACILITY, like stgadmin.arc.enduser.**    4 Mar 2014 08:46
   Class    Profile key                                T UACC     Owner    S/F W
__ FACILITY STGADMIN.ARC.ENDUSER.HBACKDS.RETAINDAYS      NONE     SYS1      R
__ FACILITY STGADMIN.ARC.ENDUSER.HBACKDS.**              G NONE     SYS1      R
__ FACILITY STGADMIN.ARC.ENDUSER.HDELETE                 NONE     SYS1      R
__ FACILITY STGADMIN.ARC.ENDUSER.HRECALL                 NONE     SYS1      R
__ FACILITY STGADMIN.ARC.ENDUSER.HRECOVER                NONE     SYS1      R
__ FACILITY STGADMIN.ARC.ENDUSER.**                      G NONE     SYS1      R
****************************** Bottom of Data ********************************
```

Data Set Access in DFSMShsm

- HRECALL
  - EXECUTE access to data set being recalled
- HMIGRATE
  - UPDATE access to data set(s) being migrated
- HBACKDS
  - UPDATE access to data set being backed up
- HRECOVER
  - ALTER access, if NEWNAME not being used
  - If NEWNAME is used, READ access to original data set and ALTER access to the new name data set
- HDELETE
  - ALTER access to data set
- HBDELETE
  - ALTER access to data set

Violations

- Facility class will be checked first for Hxxxxxx commands
  - Data set access performed after initial check
- Resource not authorized:

```
ARC1710E USER ADCDZ NOT AUTHORIZED FOR RESOURCE
STGADMIN.ARC.ENDUSER.HRECALL
ARC1001I P390.$DCT.BACKUP  RECALL FAILED, RC=0004, REAS=1710
ARC1604I COMMAND NOT AUTHORIZED FOR USER
```

- Data set access not authorized

```
ARC1001I P390.ACBTEMP  RECALL FAILED, RC=0039, REAS=0008
ARC1139I ERROR PROCESSING RACF PROTECTED DATA SET,
RECOVERY/RECALL/DELETE
ARC1139I (CONT.) TERMINATED
```

ARCCATGP

- Normally, UNCATALOG, RECATALOG or DELETE NOSCRATCH will cause recall of migrated data set
- MIGRAT catalog entry may point to non-existent MCDS entry
- Connect storage admin userids to group ARCCATGP
- LOGON to system specifying this group
- Following jobcard could also be used

  //JOBNAME JOB (*accounting information*),'USERNAME',
  //USER=*userid*,GROUP=ARCCATGP,PASSWORD=*password,*
  // EXEC PGM=….

ARCCATGP Group definition

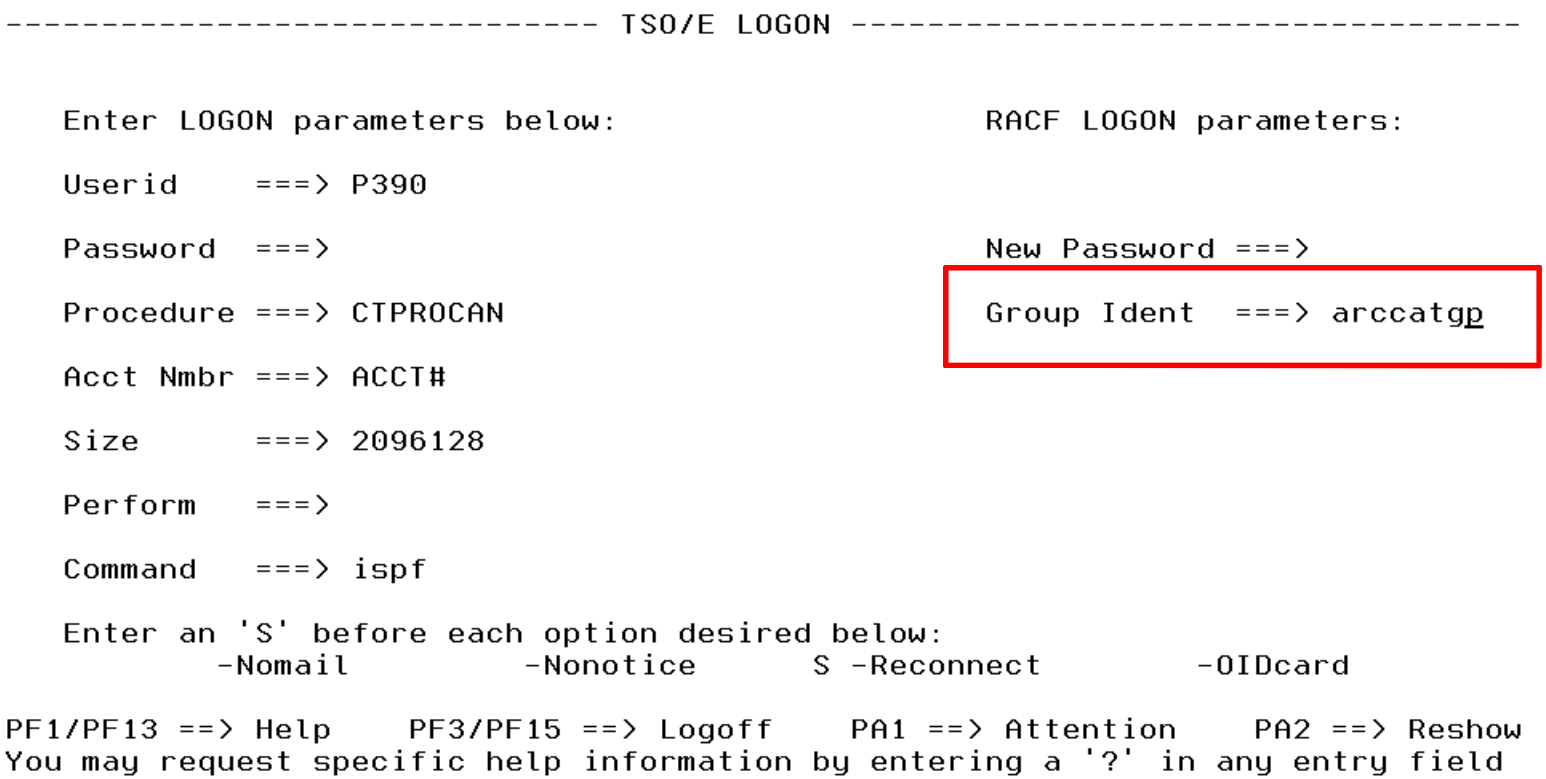
```
zSecure Admin+Audit for RACF GROUP ARCCATGP Overview
Command ===> _____________________________________________ Scroll===> CSR
like ARCCATGP                                  4 Mar 2014 08:52

_ Identification                                                    SYS1
_ RACF group name                ARCCATGP
_ Superior group                 SYS1
_ Owner                          SYS1
  Installation data              ____________________________________________

  User/Grp Auth     R SOA AG Uacc     Revokedt     Resumedt     Name
_ IBMUSER  USE      _ ___ __ NONE     ____________ ____________
_ LHANNA   USE      _ ___ __ NONE     ____________ ____________ LOUIS
_ P390     USE      _ ___ __ NONE     ____________ ____________ CHRIS


  Safeguards                              Statistics
  Terminal use authorization     No       Creation date                21May09
  Universal access authority     NONE     Universal group              No
  Data set model profile name             ____________________________________
```

Logon with group ARCCATGP



Group auditing

How can the Storage Administrator help?

- Storage Administrators can monitor how data is being managed
  - Other tools can show what actual functions are being performed against data and what jobs are accessing particular data sets
    - DFSMShsm, DFSMSdss, IDCAMS, IEBGENER, etc.
- They can be given access to see which users and groups are allowed access to storage admin functions
- They could be allowed to update security definitions within their realm
  - Not recommended

How can they do this?

- Define storage administrators access as GROUP AUDITOR to STGADMIN profiles
- Allows them to see who has access to various resources
- Does not give access to the resource
  - Still need a specific PERMIT to the function
- Does not allow them to make updates to the resources
  - Requires GROUP SPECIAL
  - Should still be done by formal security request

Setup for GROUP AUDITOR

```
//CREATE   EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  *
ADDGROUP (MH#STG) SUPGROUP(SYS1) OWNER(SYS1)
ADDGROUP (MH#AUD) SUPGROUP(SYS1) OWNER(SYS1)
RDEFINE FACILITY STGADMIN.ARC.HOLD OWNER(MH#AUD)*
PERMIT  STGADMIN.ARC.HOLD CLASS(FACILITY) ID(P390) DELETE
CONNECT (ADCDA) GROUP(MH#STG) AUTHORITY(USE) UACC(NONE)
CONNECT (ADCDA) GROUP(MH#AUD) AUTHORITY(USE) UACC(NONE) AUDITOR*
SETROPTS REFRESH RACLIST(FACILITY)
/*
```

* Auditor attribute needs to be connected to owner group

Output from RLIST FACILITY command – No auditor access to owner group

```
RLIST FACILITY STGADMIN.ARC.HOLD ALL
CLASS      NAME
-----      ----
FACILITY   STGADMIN.ARC.HOLD

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  -----------  -------
 00    MH#AUD          NONE              NONE     NO
```

<snip>

```
USER       ACCESS    ACCESS COUNT
----       ------    ------ -----
NO USERS IN ACCESS LIST
```

Output from RLIST FACILITY command – With auditor access to owner group

```
RLIST FACILITY STGADMIN.ARC.HOLD ALL
CLASS      NAME
-----      ----
FACILITY   STGADMIN.ARC.HOLD

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  -----------  -------
 00    MH#AUD          NONE              READ     NO
```

<snip>

```
USER       ACCESS    ACCESS COUNT
----       ------    ------ -----
MH#STG     READ         000000
```

Reporting on access to resources

Reporting methods - Sources

- IRRDBU00
  - RACF Database unload utility
- IRRADU00
  - RACF Audit Utility
- IRRICE
  - RACF reporting using ICETOOL
- Documented in z/OS Security Server RACF Auditor's Guide
  - SA23-2290-00 for z/OS Version 2 Release 1

Reporting methods – IRRDBU00

- RACF Database unload utility
- Shows userid and group information
- List last recorded date and time of RACROUTE=VERIFY
  - Issued when user is logging onto a system or when batch job enters the system
- For performance reasons, this should be run against backup copy or backup RACF database
- Userid running utility needs UPDATE authority against the input data set
- Used as input to DFSORT ICETOOL utility
  - IRRICE in SYS1.SAMPLIB

Reporting methods – IRRADU00

- RACF Audit utility
- Extracts SMF records relating to RACF to a sequential file
- Can be viewed directly or post-processed
- Can be loaded into DB2 and allows tailored reports to be created
- Used as input to DFSORT ICETOOL utility
  - IRRICE in SYS1.SAMPLIB

Reporting methods – IRRICE

- Source in member IRRICE in SYS1.SAMPLIB
- Sample JCL in $$CNTL$$ executes most of the reports
  - Executes RACFICE proc
- For more information, see
  - RACF Power Tools – Using IRRICE and Rexx on IRRADU00 and IRRDBU00 from Share Anaheim 2012

Reporting methods – IRRICE

- Information can also be loaded into DB2
- DDL to create tablespace and tables can be found in member IRRADUTB in SAMPLIB
- DB2 Load Utility statements are in IRRADULD in SAMPLIB

Examples

- Example for UGRP report:

```
- 1 -        UGRP: Users With Extraordinary Group Authorities        14/07/19

User ID    Group Name   Group Special   Group Operations   Group Auditor
--------   ----------   -------------   ----------------   -------------
ADCDA      MH#AUD       NO              NO                 YES
```

- Example for OPER report:

```
- 1 -        OPER: Accesses Allowed Because of OPERATIONS        14/07/19

Time       Date         User ID    Resource Name                          Volume
--------   ----------   --------   ------------------------------------   ------
09:47:52   2014-07-18   P390       VTFM.JRNL                              SMS007
09:47:53   2014-07-18   P390       VTFM.JRNL                              SMS007
09:47:53   2014-07-18   P390       VTFM.JRNL                              SMS007
09:47:54   2014-07-18   P390       VTFM.JRNL                              SMS007
09:48:14   2014-07-18   P390       VTFM.JRNL                              SMS007
09:48:34   2014-07-18   P390       VTFM.JRNL                              SMS007
```

Examples

DB2 example:

```
SELECT
ACC_DATE_WRITTEN,ACC_TIME_WRITTEN,ACC_EVT_USER_ID,ACC_JOB_NAME,
   ACC_EVENT_QUAL, ACC_RES_NAME,ACC_NAME FROM USER01.ACCESS
     WHERE ACC_RES_NAME LIKE 'STGADMIN.ARC%';
   COMMIT;
```

2014-06-18	16.20.42	AUTOMAT	HSM	SUCCESS STGADMIN.ARC.ENDUSER.HBACKDS.RETAINDAYS
2014-06-18	16.20.42	AUTOMAT	HSM	SUCCESS STGADMIN.ARC.ENDUSER.HBACKDS.RETAINDAYS
2014-06-18	16.20.42	AUTOMAT	HSM	SUCCESS STGADMIN.ARC.ENDUSER.HBACKDS.RETAINDAYS
2014-06-18	16.20.42	AUTOMAT	HSM	SUCCESS STGADMIN.ARC.ENDUSER.HBACKDS.RETAINDAYS
2014-06-18	16.20.42	AUTOMAT	HSM	SUCCESS STGADMIN.ARC.ENDUSER.HBACKDS.RETAINDAYS
2014-06-18	18.30.23	ADCDZ	HSM	INSAUTH STGADMIN.ARC.ENDUSER.HRECALL
2014-06-18	18.31.27	ADCDZ	HSM	SUCCESS STGADMIN.ARC.ENDUSER.HRECALL
2014-06-19	00.07.17	AUTOMAT	HSM	SUCCESS STGADMIN.ARC.ENDUSER.HBACKDS.TARGET
2014-06-19	08.01.10	AUTOMAT	HSM	SUCCESS STGADMIN.ARC.LIST

Examples

DB2 example:

```
SELECT
ACC_DATE_WRITTEN,ACC_TIME_WRITTEN,ACC_EVT_USER_ID,ACC_JOB_NAME,
   ACC_EVENT_QUAL, ACC_RES_NAME,ACC_NAME FROM USER01.ACCESS
     WHERE ACC_RES_NAME LIKE 'STGADMIN.ARC%';
   COMMIT;
```

2014-06-18	16.20.42	AUTOMAT	HSM	SUCCESS	STGADMIN.ARC.ENDUSER.HBACKDS.RETAINDAYS
2014-06-18	16.20.42	AUTOMAT	HSM	SUCCESS	STGADMIN.ARC.ENDUSER.HBACKDS.RETAINDAYS
2014-06-18	16.20.42	AUTOMAT	HSM	SUCCESS	STGADMIN.ARC.ENDUSER.HBACKDS.RETAINDAYS
2014-06-18	16.20.42	AUTOMAT	HSM	SUCCESS	STGADMIN.ARC.ENDUSER.HBACKDS.RETAINDAYS
2014-06-18	16.20.42	AUTOMAT	HSM	SUCCESS	STGADMIN.ARC.ENDUSER.HBACKDS.RETAINDAYS
2014-06-18	18.30.23	ADCDZ	HSM	INSAUTH	STGADMIN.ARC.ENDUSER.HRECALL
2014-06-18	18.31.27	ADCDZ	HSM	SUCCESS	STGADMIN.ARC.ENDUSER.HRECALL
2014-06-19	00.07.17	AUTOMAT	HSM	SUCCESS	STGADMIN.ARC.ENDUSER.HBACKDS.TARGET
2014-06-19	08.01.10	AUTOMAT	HSM	SUCCESS	STGADMIN.ARC.LIST

Note

- Make sure to audit for both success and failures

```
rlist FACILITY STGADMIN.ARC.ENDUSER.HRECALL all
CLASS      NAME
-----      ----
FACILITY   STGADMIN.ARC.ENDUSER.HRECALL

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  -----------  -------
 00    SYS1            NONE              ALTER    NO
```

<snip>

```
AUDITING
--------
ALL(READ)
```

DFSMSdfp Facility Classes

VSAM Dataset RACF Checking

- CLASS(DATASET) checking for VSAM datasets is always on the “Sphere” name, (i.e. the primary cluster name).
- RACROUTE AUTH includes DSTYPE=V, and VOLSER is n/a.

```
USER.KSDS1 (C)          ← Used for Access Authorization
- USER.KSDS1.DATA (D)
- USER.KSDS1.INDEX (I)

APPLIC.SPHERE (C)       ← Used for Access Authorization
- APPLIC.SPHERE.DATA (D)
- APPLIC.SPHERE.INDEX (I)
APPLIC.SPHERE.AIX1 (G)
- APPLIC.SPHERE.AIX1.PATH1 (R)
- APPLIC.SPHERE.AIX1.DATA (D)
- APPLIC.SPHERE.AIX1.INDEX (I)
```

Metadata Overview

ICF Catalog Structure: 3 Components

- **BCS:** Basic Catalog Structure
  - Is a Usercatalog (or Master Catalog, or VOLCAT).
  - Is a special KSDS, usually with Dataset Names in the key.
- **VVDS:** VSAM Volume Dataset
  - Resides on each DASD volume as an ESDS named 'SYS1.VVDS.Vxxxxxx' where "xxxxxx" is the VOLSER.
  - Serves as an extended supplement to the VTOC.
- **VTOC:** Volume Table of Contents
  - OS-VTOC (not really a dataset at all)
  - VTOC-Index (not a VSAM dataset, but a pseudo ESDS).

Metadata Overview

Internal VVDS Record Types

- **NVR:** SMS Non-VSAM on a volume.
  - Only for 1st volume of an SMS Non-VSAM Dataset.
  - Secondary volume extents do not have an NVR.
  - Non-SMS Non-VSAM datasets do not have an NVR.
- **VVR:** VSAM Dataset Component on a volume.
  - A KSDS will have at least two VVRs, for DATA, and INDEX.
  - Applicable to all VSAM for both SMS and Non-SMS.
  - Secondary volume extents have a type “Q” VVR.
- Each VVR or NVR includes a BCS-Back-Pointer that identifies the usercatalog that it was originally cataloged under.

SMS Rules for the Common Folk

- VSAM and SMS Non-VSAM Physical Datasets and their Catalog entries are meant to always exist together in lock-step.
- Non-SMS Non-VSAM considered to be usable even when not cataloged, when qualified with the VOLSER(s).
- Certain exceptions can be granted by providing READ access using CLASS(FACILITY) profiles, but at risk of polluting SMS DASD pools with inconsistencies.

FACILITY Profiles - SMS Metadata Management

- Security integrity more concerned with preventing unintentional problems before they happen.
- Enable certain capabilities beyond the context of common usage to maintain DASD integrity from operational side.
- Can be used as a “switch” to enable alternate logic paths for security checking requirements.
- Can serve as a “Yellow Tape” barrier when dealing with broken catalogs.

FACILITY Profiles: STGADMIN.IGG.*

- Recommend RACF profiles have OWNER and SUPGROUP to a Storage Management RACF Group to enable Group-Auditor or Group-Special CONNECT for authorized USERIDs.
- Access to IGG FACILITY profiles are typically READ or NONE.
- Some IGG profiles are access “switches”, where denial is not considered a violation. (LOG=NOFAIL or LOG=NONE).
- Other IGG profiles will issue ICH408I when a violation occurs due to an unauthorized request attempt.

Notable DFP FACILITY Profiles:

Controlled Switches:

```
STGADMIN.IGG.DEFDEL.UALIAS
STGADMIN.IGG.DLVVRNVR.NOCAT
STGADMIN.DPDSRN.<old_dsn_mask>
```

CATLOCK (Recovery):

```
IGG.CATLOCK
```

Controlled Capability:

```
STGADMIN.IGG.DIRCAT
STGADMIN.IGG.DELETE.NOSCRTCH
STGADMIN.IGG.DEFINE.RECAT
```

DFSMSdfp FACILITY Class “Switch”
STGADMIN.IGG.DEFDEL.UALIAS

- “Non-Violation” RACF Checking. (No ICH408I if no access).
- Applicable for DEFINE and DELETE ALIAS commands with RELATE(<usercatalog>), which update Master Catalogs.

- Without READ access to DEFDEL.UALIAS, then ACCESS(UPDATE) to the Master Catalogs is needed to DEFINE/DELETE Aliases.
- READ access to DEFDEL.UALIAS will cause DELETE/DEFINE ALIAS to bypass all security checking against the Master Catalog.
- Regardless, security access to the “RELATE” usercatalog is not checked either way.

CLASS(FACILITY) Practical Example:
STGADMIN.IGG.DEFDEL.UALIAS

Situation:

- TSO administrator needs UPDATE access to Master Catalogs to add catalog aliases for new USERIDs.
- Occasionally the alias step is overlooked or mistyped.
- TSO User datasets unintentionally get cataloged in the master catalog. DEFINE ALIAS fails due to HLQ conflict, and Storage Management often must bail them out.

Solution:

- Give TSO Admin READ access to DEFDEL.UALIAS
- Give TSO Admin READ access to master catalogs.

DFSMSdfp FACILITY Class "Switch"
STGADMIN.IGG.DLVVRNVR.NOCAT

- If Not Defined: UACC(NONE) assumed.
- "Non-Violation" RACF Checking. (No ICH408I if no access).
- Applicable for DELETE VVR/NVR.
- Without READ access to NOCAT, then dataset must not be cataloged in the usercatalog name identified by the VVR/NVR BCS-Back-Pointer field, or the BCS name provided by the CATALOG keyword. ALTER access is required for that catalog.
- READ access to NOCAT will allow DELETE VVR/NVR even if the dataset is cataloged. No access checking is performed against the effective usercatalog.

CLASS(FACILITY) Practical Example:
STGADMIN.IGG.DLVVRNVR.NOCAT

Situation:

- MASTCAT.SYSA-1 Cloned for new z/OS Release.
- SYSA now running on MASTCAT.SYSA-2
- Don’t need PAGE.SYSA.LOCAL.SPARE anymore.
- DELETE can uncatalog page dataset, but does not scratch it.
- DELETE VVR (VSAM Scratch) fails IDC3009I (090-046) because dataset’s BCS back-pointer in the page volume’s VVDS still references MASTCAT.SYSA-1, which is now gone.

Solution:

- ACCESS(READ) for STGADMIN.IGG.DLVVRNVR.NOCAT
- DELETE PAGE.SYSA.LOCAL.SPARE VVR FILE(VPAGE99)

DFSMSdfp FACILITY Class “Switch”

STGADMIN.DPDSRN.<*old_dsmask*>

- Enables bypass of Exclusive SYSDSN Enqueue with renaming a Non-VSAM Non-SMS dataset.
- Does not change the standard existing dataset security checking involved with renaming a dataset .
- Applicable to DADSM RENAME functions, such as IEHPROGM RENAME, or RENAME using ISPF.
- Typically used in context to building new SYSRES volumes.

CLASS(FACILITY) Practical Example:
STGADMIN.DPDSRN.<*old_dsmask*>

Situation:

- Systems Programmer is building RES21B that is currently not in use as a SYSRES on any active system.
- SYS1.LINKLIB on RES21B needs to be resized, i.e. re-allocated.
- SYS1.LINKLIB (on RES21A, etc.) is allocated DISP=SHR by other tasks within the sysplex. SCRATCH on RES21B fails because that DSN is “in use”.

Solution:

- ACCESS(READ) for STGADMIN.DPDSRN.SYS1.* only for Sysprog.
- IEHPROGM: RENAME SYS1.LINKLIB –> SYS1.LINKLIBX on RES21B.
- IEHPROGM: SCRATCH SYS1.LINKLIBX on RES21B.

DFSMSdfp FACILITY Class
IGG.CATLOCK

- Enables ability to issue ALTER <bcs> LOCK or UNLOCK, else command failed with ICH408I and IDC3009I (186-002).
- Allows access to a catalog in a LOCK state, else catalog request is failed with error IDC3009I (186-002).

- z/OS 2.1: Also enables ALTER <bcs> SUSPEND or RESUME.
- Allows access to a catalog in a SUSPEND state, else the catalog request goes into a WAIT until ALTER <bcs> RESUME occurs.
- Jobs on pre-z/OS 2.1 systems see a shared catalog in SUSPEND state, as instead being in LOCK state.

DFSMSdfp FACILITY Class

IGG.CATLOCK - Policies

- If IGG.CATLOCK is Not Defined: as if everybody has ACCESS(NONE).
- READ access should only be assigned to Userids that will be creating, moving, deleting, or repairing catalogs.
- System tasks (Catalog Address Space is n/a), production and non-production jobs, online systems, Userids, etc. that are not specifically involved in catalog recovery should have ACCESS(NONE). Otherwise, catalog updates can be lost, bad information can be returned, and catalog recovery procedures can be corrupted.
- IGG.CATLOCK UACC(READ) is a bad idea.

DFSMSdfp FACILITY Class

Miscellaneous Controlled Capabilities

- STGADMIN.IGG.DIRCAT
  - Enables CATALOG() keyword for DELETE, DEFINE, and ALTER.
  - Else possible ICH408I and IDC3009I(190-002)
- STGADMIN.IGG.DELETE.NOSCRATCH
  - Allows an uncat or DELETE NOSCRATCH of SMS data set.
  - Else Request Failed with ICH408I and IDC3009I (190-010).
- STGADMIN.IGG.DEFINE.RECAT
  - Empowers RECATALOG keyword for DEFINE, and ALTER.
  - Else possible ICH408I for ALTER access to Dataset name.

DFSMSdfp FACILITY Class
Miscellaneous Controlled Capabilities

- STGADMIN.IGG.LIBRARY
  - DEFINE, DELETE or ALTER library or volume contents.
- STGADMIN.IGG.ALTER.UNCONVRT
  - Controls ability to convert VSAM from SMS to non-managed.
- STGADMIN.IGG.DELGDG.FORCE
  - DELETE FORCE for GDG base with existing SMS GDSs.

- STGADMIN.IGG.*
  - UACC(NONE) Catch-All.
  - Insufficient access with most other profiles will produce ICH408I message and IDC3009I (190-nnn) errors.

DFSMSdfp FACILITY Class
"Enigmatic" Profiles

The following documented profiles are not yet fully understood. Recommended access rules and practical examples of usage are not available at this time.

- STGADMIN.IGG.DEFNVSAM.NOBCS
- STGADMIN.IGG.DELNVR.NOBCSCHK
- STGADMIN.IGG.DEFNVSAM.NONVR

Research and investigation continues …

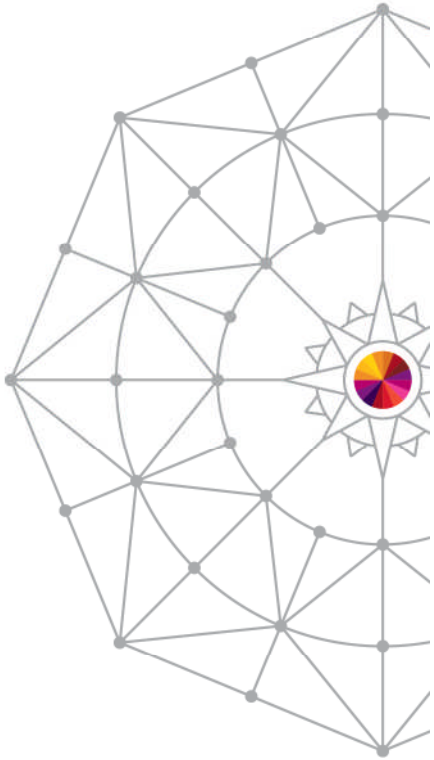
DFSMSdfp FACILITY Class

Manuals:

- DFSMS Access Method Services
  SC23-6846-00 in z/OS DFSMS V2 R1
- DFSMS Managing Catalogs
  SC23-6853-00 in z/OS DFSMS V2 R1
- DFSMSdfp Advanced Services
  SC23-6861-00 in z/OS DFSMS V2 R1

Questions?

DFSMS Security: Unveiling the Inner Truth



Chris Taylor (IBM Corporation)
Marty Hasegawa (Rocket Software)

August 7, 2014
Session Number 15767

Copyright (c) 2014 by SHARE Inc. Except where otherwise noted, this work is licensed under http://creativecommons.org/licenses/by-nc-sa/3.0/