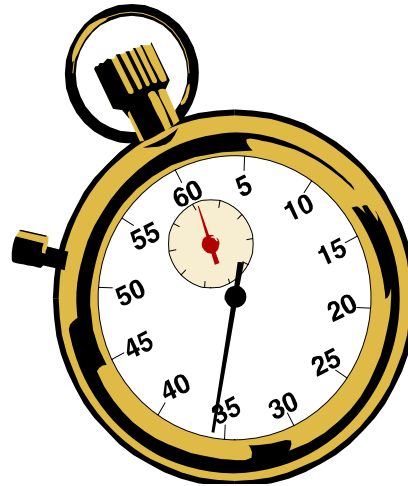


ACHIEVING PEAK RACF PERFORMANCE

Vanguard Security & Compliance 2011 - AST11 - June 2011



Robert S. Hansel

Lead RACF Specialist - RSH Consulting, Inc.

R.Hansel@rshconsulting.com - 617-969-9050 - www.rshconsulting.com

RSH PRESENTER



Robert S. Hansel is Lead RACF Specialist and founder of RSH Consulting, Inc., a firm he established in 1992 and dedicated to helping clients strengthen their IBM z/OS mainframe access controls by fully exploiting all the capabilities and latest innovations in RACF. He has worked with IBM mainframes since 1977 and in information systems security since 1981. Mr. Hansel began working with RACF in 1986 and has been a RACF administrator, manager, auditor, instructor, developer, and consultant. He has reviewed, implemented, and enhanced RACF controls for major insurance firms, financial institutions, utilities, payment card processors, universities, hospitals, and international retailers. Mr. Hansel is especially skilled at redesigning and refining large-scale implementations of RACF using role-based access control concepts. He has also created elaborate automated tools to assist clients with RACF administration, database merging, identity management, and quality assurance.

Contact and background information:

- 617-969-8211
- R.Hansel@rshconsulting.com
- www.linkedin.com/in/roberthansel

OBJECTIVES

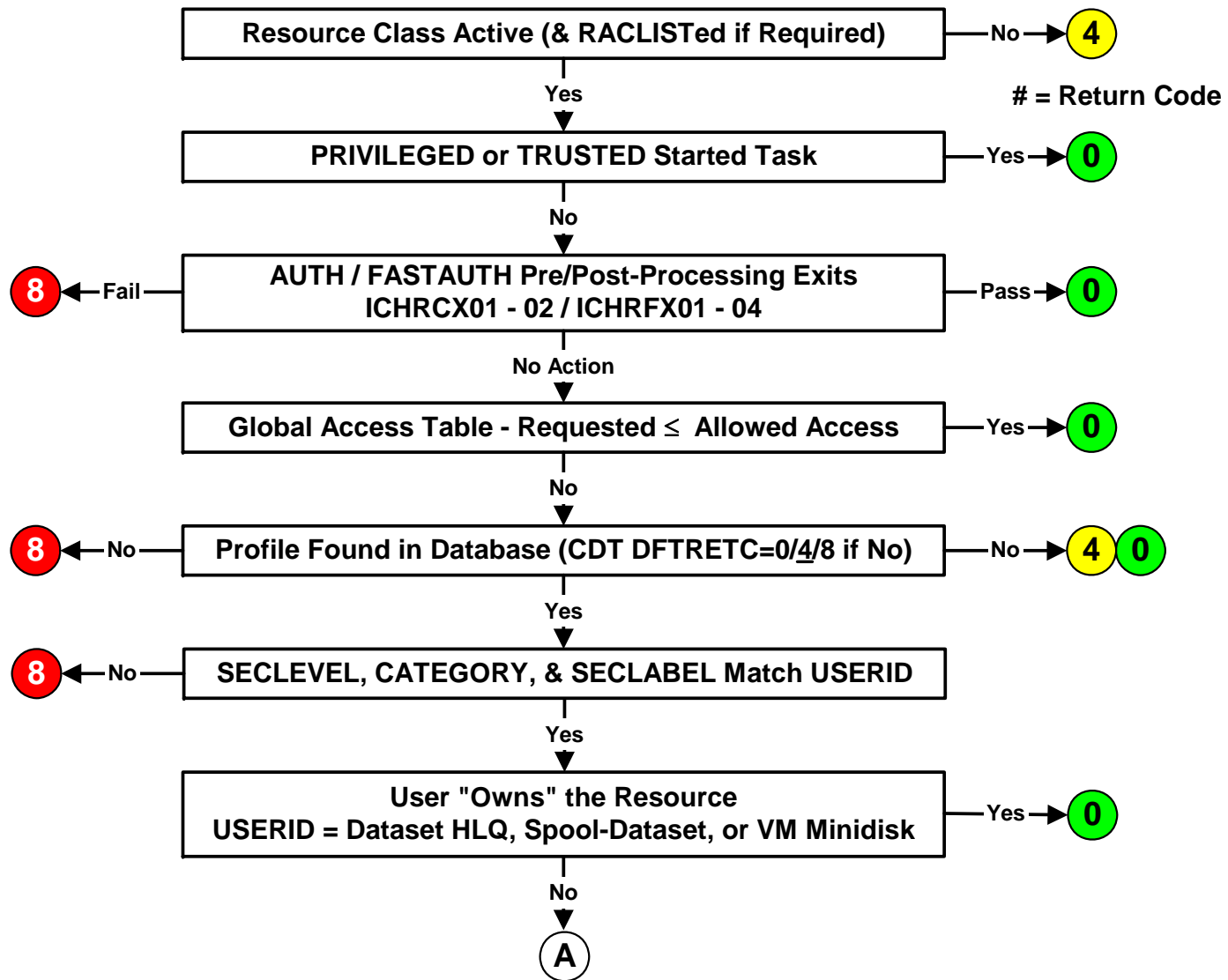
Optimize Access Authorizations

Expedite the Logon Process

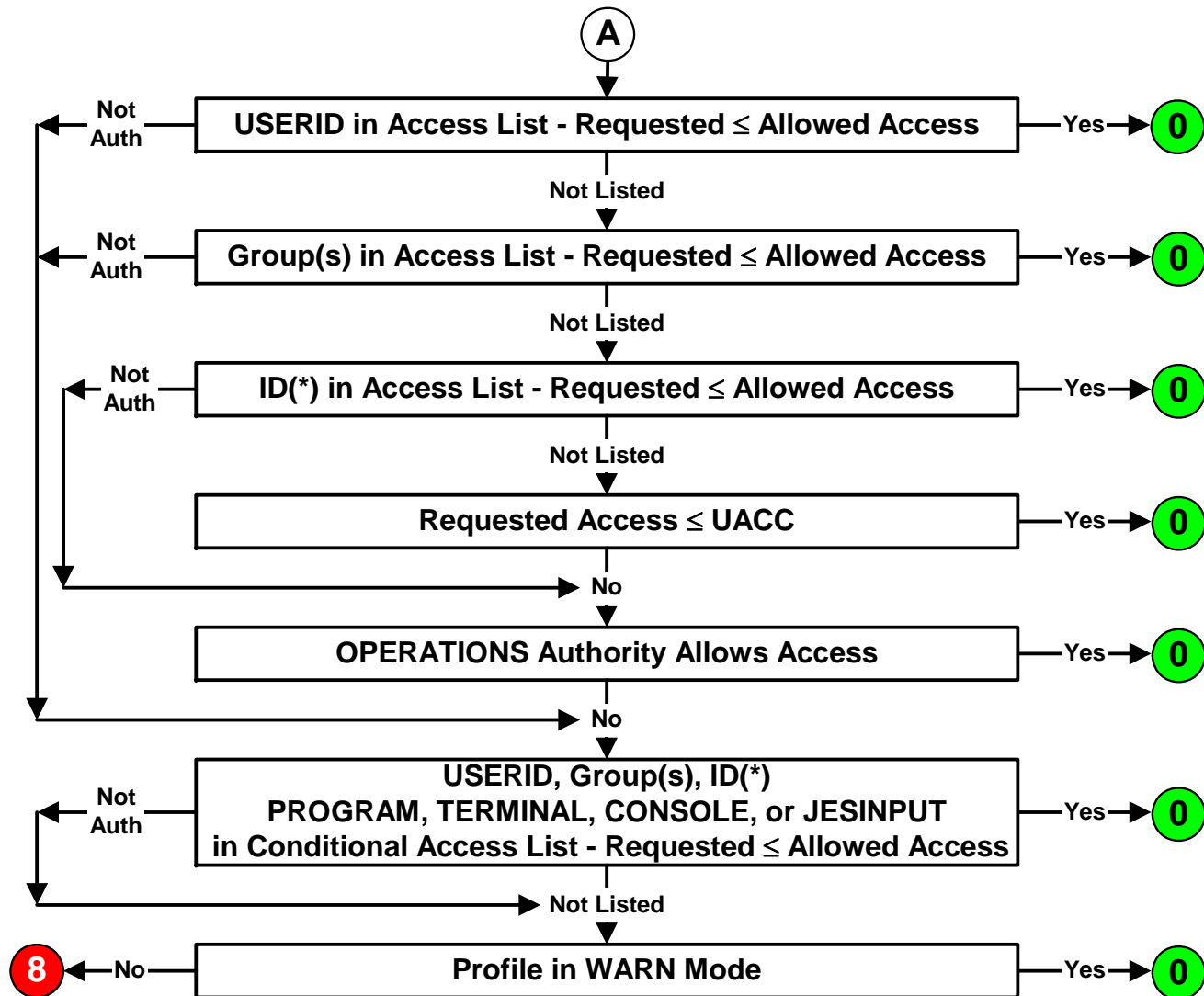
Minimize I/O Operations

RACF, z/OS, CICS, IMS, and DB2 are Trademarks of the International Business Machines Corporation

RACF AUTHORIZATION DECISION LOGIC



RACF AUTHORIZATION DECISION LOGIC



RACF AUTHORIZATION DECISION LOGIC

Deactivate unused classes (be mindful of POSITs when deactivating)

- Resource classes, including SECDATA & SECLABEL classes
- Global Access Table classes

Make access list processing efficient

- Minimize the number of entries in access lists
 - Grant end-user access via groups instead of USERIDs
 - Remove obsolete residual entries - run IRRRID00
 - Remove redundant entries (e.g., access allowed equals UACC)
- Minimize the number of group connects per user

Reduce reliance on OPERATIONS authority by implementing Storage Administration authorities

Write efficient exit code

Implement the Global Access Table

GLOBAL ACCESS TABLE

Performance enhancement tool

- **Grants immediate access without referring to the profile and without logging**
- **Used to grant access to common shared resources**

GLOBAL Class

- **Profile - Class name [RDEF GLOBAL DATASET]**
- **Members - resource/access [ADDMEM('CTLG.USER'/UPDATE)]**
- **Resource**
 - **Discrete or Generic - General Resource generic profile rules**
 - **Need not match existing profiles**
- **Access-levels - ALTER | CONTROL | UPDATE | READ | NONE**

Special Variables - Used in resource names

- **&RACUID Substitute with requesting user's USERID**
- **&RACGPID Substitute with user's current connect group**

GLOBAL ACCESS TABLE

Sample entries (DSMON Report)

DATASET	&RACUID.*.**	ALTER	
DATASET	&RACGPID.*.**	UPDATE	(avoid - unintended access)
DATASET	CATALOG.MASTER	READ	
DATASET	CATALOG.USER	UPDATE	
DATASET	ISPF.LIBRARY	READ	
DATASET	SDSF.LIBRARY	READ	
DATASET	SYS1.BROADCAST	READ	
DATASET	SYS1.HELP	READ	
DATASET	SYS1.MACLIB	READ	
DATASET	SYS1.RACF	NONE	(precludes access)
DATASET	SYS%.**	READ	(avoid - too broad)
DATASET	*.**.#SMSTEST	ALTER	
FACILITY	IEC.TAPERING	READ	
FACILITY	STGADMIN.ARC.ENDUSER.**	READ	
JESJOBS	SUBMIT.*.&RACUID*. &RACUID	READ	
JESJOBS	CANCEL.*.&RACUID.*	ALTER	(not needed - post rtoken check)
JESSPOOL	*.&RACUID.**	ALTER	
JESSPOOL	*.*.\$JESNEWS.**	READ	
MQQUEUE	MQS*.ISF.USER.&RACUID.**	ALTER	
OPERCMDS	MVS.MCSOPER.&RACUID	READ	
TSOAUTH	JCL	READ	
TSOAUTH	RECOVER	READ	

GLOBAL ACCESS TABLE

Activated and managed via SETROPTS

- **SETROPTS GLOBAL(class) | NOGLOBAL(class) [REFRESH]**
- **Must be refreshed if updated**

Can be used for most resource classes

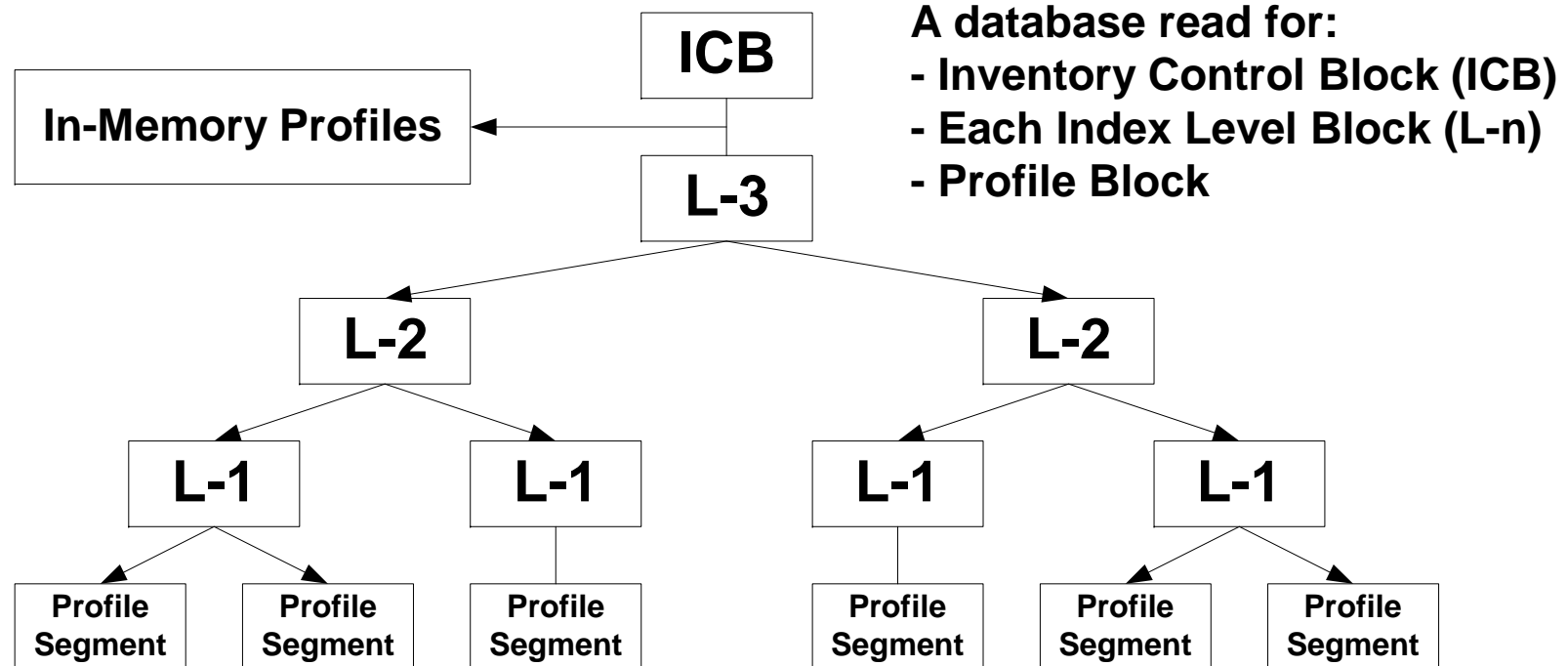
- **Not checked in RACROUTE REQUEST=FASTAUTH processing**
- **Not checked in RACROUTE REQUEST=VERIFY processing for APPL, TERMINAL, JESINPUT, CONSOLE, APPCPORT, and SERVAUTH resources**

Keep list of entries short and efficient to minimize search

Drawbacks

- **Precludes logging (except SETR AUDIT(class) resource defines)**
- **Undermines protection if allows more access than profile UACCs**

RACF PROFILE RETRIEVAL - LOGICAL



Data is written and retrieved in 4K blocks

Individual profiles and profile segments can be greater than 4K in size and span multiple contiguous blocks, each of which requires I/O to fetch - keep profiles as small as possible

RESIDENT DATA BLOCKS

RACF maintains buffers in ECSA to hold copies of most recently used blocks (index, BAM, and profiles)

Frequently used blocks tend to stay in these buffers

Desired number of resident blocks is specified in the Database Name Table - ICHRDSNT

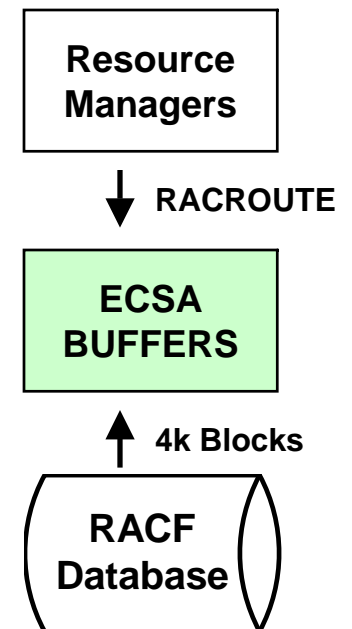
AL1(1)	Number of databases
CL44'RACF.PRIMARY'	Primary DB name
CL44'RACF.BACKUP'	Backup DB name
AL1(100)	# of Resident Data Blocks
XL1'xx'	Flags

Default/minimum number of blocks

10 / 0	Non-Sysplex (<u>none</u> for backup database)
50 / 50	Sysplex (+ additional 20% for backup database)

Maximum number - 255 (recommended)

Sysplex - first system to IPL sets number of blocks



GENERIC PROFILES STORED IN MEMORY

Sets of generic profiles are stored in each individual user's address space memory

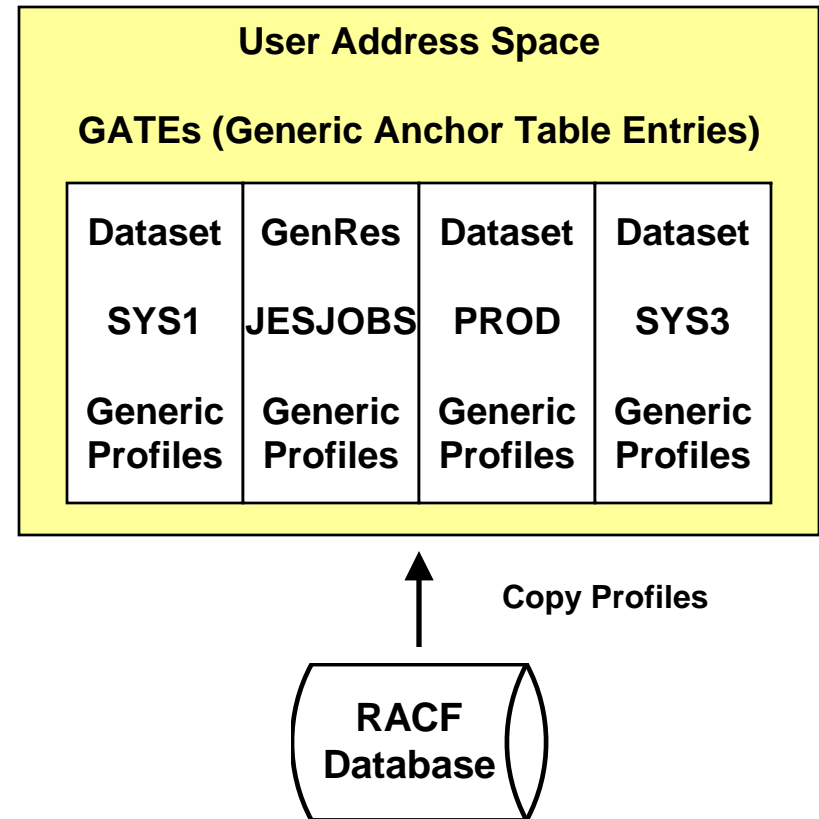
Each set is comprised of generic profiles for either:

- Dataset HLQ
- General Resource class

Upon first access to a resource class or HLQ, a list of all the associated generic profiles are retrieved and loaded into memory

Individual generic profiles are retrieved as needed for authorization checking and retained in memory thereafter

Profiles in memory are used for authorization checking - not those in the RACF database



GENERIC PROFILES STORED IN MEMORY

Once all sets of generic profiles are filled, when the next new resource class or HLQ is accessed, the set with the least recently used profiles is dropped and replaced with the new one

Users accessing many different HLQs and/or general resources could experience thrashing (i.e. constant replacement) among the sets

Dataset HLQs or general resources classes with many generic profiles take more I/O and CPU time to retrieve and load

Prior to z/OS 1.12, RACF kept 4 sets of profiles

With z/OS 1.12, RACF can optionally keep up to 99 sets of profiles

- **Changed with the SET command**
- **Can define for SYSTEM or JOBNAME(jobname jobname* ...)**
- **Minimum/Default is 4**

GENERIC PROFILES STORED IN MEMORY

Additions or changes to generic profiles requires in-memory copies to be refreshed before they become effective; refresh methods ...

- **User must logoff and logon to renew the in-memory profiles**
- **User can execute a LISTDSD GENERIC command to refresh all profiles for the HLQ**

LISTDSD 'HLQ.anything' GENERIC

- **SETROPTS GENERIC(class) REFRESH - this immediately drops all in-memory profile sets for the designated class for all active users and requires every user to reload them upon next access**

I/O is still required for ...

- **Datasets if the RACF indicator bit is on**
- **General resources to check for a discrete profile before generics are checked**

Can avoid having to retrieve and load profiles into user memory by:

- **Granting access using the Global Access Table**
- **Loading profiles into memory using GENLIST and RACLIST**

SETROPTS GENLIST & RACLIST

Cause profiles to be stored in memory for rapid reference and to avoid I/O to the database

Mutually exclusive SETROPTS options set for specific classes

Effects all classes with the same POSIT value

GENLISTed and RACLISTed classes do not consume any of a user's in-memory generic profile sets

GENLIST(class)

- **Retrieval of first Generic profile prompts retrieval and storage of a list of all Generic profiles for the class in ECSA**
- **Generic profiles are individually retrieved on first reference and retained in ECSA for subsequent reference**
- **I/O still required to check for discrete profile**
- **Class must be defined in the CDT with GENLIST=ALLOWED**
- **Refreshed with SETROPTS GENERIC(class) REFRESH**
- **Recommendation - VM related classes**

SETROPTS GENLIST & RACLIST

RACLIST(class)

- All profiles for a specified class are stored in a shared dataspace
 - SETROPTS RACLIST(class), if RACLIST=ALLOWED in CDT
 - RACROUTE REQUEST=LIST,GLOBAL=YES by certain applications

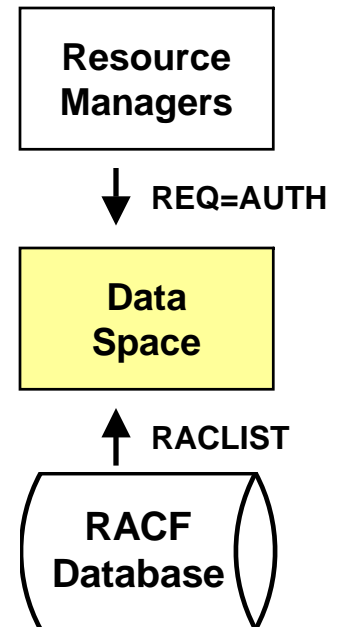
CICS	IMS	DB2	MQSeries
------	-----	-----	----------
 - Updated with SETROPTS RACLIST(class) REFRESH
 - Profile segments are not stored in memory (e.g., STDATA)
 - Required to exploit grouping classes (e.g., DASDVOL / GDASDVOL)

- CDT RACLREQ=YES - Required

APPCSERV	APPCTP	CRYPTOZ	CSFKEYS
CSFSERV	DEVICES	DIGTCIRT	DIGTNMAP
FIELD	IDIDMAP	NODES	OPERCMDS
PROPCNTL	PSFMPL	PTKTDATA	RACFHC
RACFVARS	RDATALIB	SECLABEL	SERVAUTH
STARTED	SYSMVIEW	UNIXPRIV	VTAMAPPL

- Considerations / Recommendations(*):

APPL*	CDT*	DASDVOL	DIGT Classes*
DSNR	FACILITY*	JES classes	LDAPBIND*
LOGSTRM	PRINTSRV*	RRSFDATA*	TSO classes*
TERMINAL*	SDSF	SURROGAT	



RACGLIST CLASS

Stores RACLISTed profiles in pre-processed form for quick re-loading at IPL, RACROUTE REQUEST=LIST, and REFRESH

During RACLIST REFRESH for z/OS images sharing a database with Sysplex communications, first image fetches, merges, and stores a copy of processed member and grouping profiles for other images to simply retrieve and load

Activated by class - profiles are class names

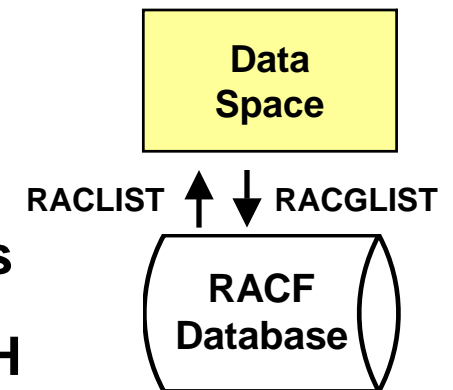
- SETROPTS CLASSACT(RACGLIST)
- RDEFINE RACGLIST class-name

Especially beneficial for CICS, IMS, & DB2 classes

Updated by SETROPTS RACLIST(class) REFRESH

Ensure database has sufficient space for RACGLIST profiles

Note: IPLs no longer cause refresh of RACGLISTed classes



NAMING CONVENTION TABLE

Can be used to convert and restructure dataset names prior to RACF profile checking

For example, could convert from/to

PROD.PAY	>>	PAY.PROD
TAPE.PAY	>>	PAY.TAPE
TEST.PAY	>>	PAY.TEST
VSAMP.PAY	>>	PAY.PRODV
VSAMT.PAY	>>	PAY.TESTV

Hard-coded Macro Table - ICHNCV00

Benefits

- Consolidating an application's files under a common HLQ can reduce the number of generic profiles required to protect the data
- Eliminating HLQs shared by several applications can reduce the I/O necessary to fetch all the different generic profiles

DATABASE REORGANIZATION

Over time, administrative actions have the following effect

- **Index entry additions and profile expansions fill a block to overflowing requiring a block split**
- **Profile and segment deletions empty all but small percentage of a block, wasting both database and buffer space**
- **Newly added profile segments get stored in different blocks than the related profile requiring more I/O to fetch, especially during logon**
- **Creating and deleting profiles causes fragmentation of free space making it difficult for RACF to find contiguous blocks for storing large profiles**

IRRUT400 utility - reorganizes the database - run periodically

- **Aligns index and associated profile blocks in sequential order**
- **Fills in data blocks eliminating wasted space and fragmentation**
- **Optionally places all profile segments in same block when possible**
- **Compresses the index and corrects upper level index errors**
- **Optionally adds free space to index blocks for subsequent growth**
- **Rebuilds BAM blocks, thereby eliminating any prior errors**

DATABASE PLACEMENT

Place on higher performance DASD devices

Use cached control units or solid state devices where possible

Provide device and channel separation between the Primary and Backup databases

Use control units with multiple, redundant channels

Place each database dataset on a separate volume

Isolate the database datasets from other files or place them with infrequently accessed files

DATABASE SHARING

Global Resource Serialization (GRS) ENQs rather than DASD hardware RESERVEs

- **Avoids contention & monopolization**
- **PARMLIB(GRSRNLxx) conversion entry - SYSZRACF**

DATABASE CACHING

RACF Sysplex Data Sharing

- **Uses coupling facility as large store-through cache for the Resident Data Blocks - caches ICB, index, & profile data blocks (can improve performance for single system)**
- **Enabled by ICHRDSNT flag on first database entry**
 - **XL1'x0' No Sysplex**
 - **XL1'x8' Sysplex without data sharing**
 - **XL1'xC' Sysplex with data sharing**
- **Coupling Facility Resource Manager (CFRM) sets cache policy**
- **To assist in calculating the coupling facility size for RACF, go to <http://www.ibm.com/systems/support/z/cfsizer/racf/>**
- **If feasible, specify size large enough to hold all index blocks plus all data blocks for non-RACLISTed resource classes**

DATABASE SPLIT

Divides database into multiple subset databases

Maximum number of databases

- z/OS - 90
- z/VM - 4

ICHRRNG - RACF Range Table (resides in LPA)

TABLE ENTRY	PROFILE PREFIXES	DATA BASE
F'5' (# of ranges)		
XL44'0000000000',AL1(1)	\$ - B	1
CL1'C',XL43'00',AL1(2)	C - GCICTRM	2
CL8'GCICSTRN',XL36'00',AL1(1)	GCICTRN - M	1
CL1'N', XL43'00',AL1(2)	N - TAPVOK	2
CL7'TAPEVOL' ,XL37'00',AL1(1)	TAPEVOL - 9	1

DATABASE SPLIT

Requires multiple entries in ICHRDSNT table

AL1(2)

Number of databases

CL44'RACF.PRIMARY1'

Primary DB name

CL44'RACF.BACKUP1'

Backup DB name

AL1(255)

of Resident Data Blocks

XL1'xx'

Flags

CL44'RACF.PRIMARY2'

Primary DB name

CL44'RACF.BACKUP2'

Backup DB name

AL1(100)

of Resident Data Blocks

XL1'xx'

Flags

DATABASE SPLIT

DATABASE	DB#1	DB#2	DB#3	DB#4	TOTAL
% FULL	7	16	37	0	Ave 15
GROUP	342	2440	261	0	3048
USER	5264	8281	12419	135	25971
DATASET	1540	2404	10860	23	14827
TERMINAL	30	0	0	0	30
APPL	0	3	0	0	3
DSNR	0	3	0	0	3
FACILITY	0	61	0	0	61
PROGRAM	0	4	0	0	4
TSOPROC	0	102	0	0	102
ACCTNUM	0	35	0	0	35
OMCANDLE	0	30	0	0	30
GCICSTRN	0	0	319	0	319
GDFHTEST	0	0	0	269	269
TSOCMDS	0	276	0	0	276
... Others					
TOTAL:	16404	31427	79135	447	127413

DATABASE SPLIT

Advantages

- **Spread workload across devices**
- **Increase resident data blocks - up to 255 for each database**
- **Reduce index levels - ICHRRNG acts as the highest level index**
- **Additional I/O queues - one for each database - reduces impact of enqueues**

Disadvantages

- **More databases to manage - must unload, backup, and reorganize them individually**
- **Requires entire Sysplex IPL to change**

Planning Considerations

- **Profile naming conventions - determines profile placement**
- **Spread heavily used profiles and resource classes evenly across the databases**
- **Factor in effects of GAT, GENLIST, RACLIST, and RACGLIST**
- **Avoid large numbers of ranges - do not exceed 750**

LOGGING

Log judiciously

- **SETROPTS LOGOPTIONS(ALWAYS(class) | SUCCESSES(class))**
- **SETROPTS OPERAUDIT**
- **SETROPTS AUDIT(class)**
- **Resource AUDIT(ALL | SUCCESSES(level))**
- **Resource GLOBALAUDIT(ALL | SUCCESSES(level))**
- **User UAUDIT**

Reduce logging if related records are not need

- **SETROPTS LOGOPTIONS(NEVER(class))**
- **FACILITY BPX.SAFFASTPATH** - if defined, z/OS Unix will skip RACF calls and related logging if it can determine on its own that access is allowed by permission bits
- **Note** - neither option suppresses logging for users with UAUDIT

STATISTICS

Eliminate the use of Statistics

- **SETROPTS STATISTICS(class) | NOSTATISTICS(class) Option**
- **Access counts kept only on Discrete profiles**
- **Not incremented for GAT or RACLISTed class access**
- **May not be accurate in a shared database environment**
- **Increases CPU processing to calculate and I/O to retain**

Updating Statistics in the backup database - ICHRDSNT flag

XL1'0x'	No updates are duplicated in the backup database (default)
XL1'8x'	Updates other than statistics are duplicated (recommended)
XL1'Cx'	Updates including statistics are duplicated (avoid)

Limit user logon statistics update to only once per day (z1.11)

- **Implemented via APPL class profiles for associated applications**
- **Specify APPLDATA('RACF-INITSTATS(DAILY)') to activate**

z/OS UNIX IDENTITY MAPPING

Mapping required when corresponding identity must be determined (e.g., Unix 'ls' command - display RACF USERID and Group for Unix Owner uid and Group gid)

Options to avoid searching all user and group OMVS segments for each look-up request

- **UNIXMAP Class**
 - Contains profiles in the form *Unnn* and *Gnnn*, where '*nnn*' is a uid or gid
 - Users and groups are 'permitted' access to signify uid and gid assignment
 - Profiles are automatically maintained when OMVS segments are created or altered via RACF commands
 - Class must be activated to be used for mapping
- **Application Identity Mapping (AIM)**
 - Restructured database with mapping index structure
 - Implemented using IRRIRA00 utility
 - Replaces UNIXMAP profiles
 - Enables use of *UID(nnn)* and *GID(nnn)* on SEARCH command

Additionally, cache uid and gid mappings in VLF

VIRTUAL LOOKASIDE FACILITY (VLF)

VLF can cache RACF information for reuse

- **Accessor Environment Elements (ACEEs)**
- **Group tree**
- **z/OS Unix mappings of uids and gids to USERIDs and Groups**
- **z/OS Unix User Security Packets (USPs)**

MAXVIRT - VLF Maximum Virtual Storage

- **Optionally specified in PARMLIB(COFVLFxx) for each VLF CLASS**
- **MAXVIRT(*nnnnnn*) - 4K block increments**
 - **Default: 4096**
 - **Range: 256 - 524288**
- **Monitor VLF use - SMF record type 41, subtype 3**
- **Default normally sufficient**

VIRTUAL LOOKASIDE FACILITY (VLF)

Accessor Environment Elements (ACEEs)

- Created during logon process - contains user's attributes, lists of groups, and logon characteristics (e.g., Point-of-Entry (POE), application)
- Caching avoids repeated retrieval of user profile for subsequent logons
- PARMLIB(COFVLFxx) entry
 CLASS NAME(IRRACEE)
 EMAJ(ACEE)
- Altering a user profile causes purge of all cached ACEEs for that user
- Refresh of logon-related classes causes purge of all cached ACEEs

Group tree

- Used to determine scope-of-groups for Group-level authorities
 SPECIAL OPERATIONS AUDITOR
- Caching avoids repeated retrieval of group profiles and tree reconstruction
- Implement only if group authority is used extensively
- PARMLIB(COFVLFxx) entry
 CLASS NAME(IRRGTS)
 EMAJ(GTS)

VIRTUAL LOOKASIDE FACILITY (VLF)

z/OS Unix mappings of uids and gids to USERIDs and Groups

- **Caching avoids repeated retrieval of mapping information**
- **Needed even with AIM restructured database**
- **PARMLIB(COFVLFxx) entry**

CLASS NAME(IRRGMAP)

EMAJ(GMAP)

CLASS NAME(IRRUMAP)

EMAJ(UMAP)

z/OS Unix User Security Packets (USPs)

- **Created when user dubs (invokes z/OS Unix function)**
- **Caching avoids repeated rebuilding of USPs during subsequent dubbing**
- **Especially helpful for applications using thread level security**
- **PARMLIB(COFVLFxx) entry**

CLASS NAME(IRRSMAP)

EMAJ(SMAP)

ENQUEUE RESIDENCY - ERV

Enqueue contention issue - low priority TSO user or batch job gets swapped out while still holding an enqueue on SYSZRACF thereby holding up other address spaces waiting on RACF

Solution - grant more CPU Service Units to address spaces enqueued on system resources enabling them to complete work before being swapped out

PARMLIB(IEAOPTxx) - ERV parameter

- **Range: 0 - 999999**
- **Default: 500**
- **Recommended: 40000 - 50000**

CICS

Keep in mind - single-threaded logon - one queue for everyone

Deactivate authorization checks for unused resource classes in the CICS Systems Initialization Table (SIT) - set *Xparm=NO*

Consider assigning the region's own USERID to Inter-Region Communication CONNECTION definitions for remote regions to make them "equivalent systems" free from link security checking

USRDELAY(#minutes)

- SIT parameter
- Determines life of user ACEEs built in an AOR by ATTACHSEC(IDENTIFY) for routed transactions initiated by users logged on to an FOR
- Default is 30
- If set too short can cause frequent rebuilding of ACEEs
- z/OS 1.11 + CICS TS 4.1 - CICS is notified of user profile changes and immediately purges the ACEE so that it can be recreated; enables longer USRDELAY to be set

STORAGE ADMINISTRATION

Use Storage Administration related authorities in lieu of individual dataset access and OPERATIONS authority

DASDVOL profiles

- **Perform backups and restores on entire non-SMS managed DASD volumes without checking access for each dataset**

FACILITY STGADMIN.ADR.STGADMIN profiles

- **Allows use of ADMINistrator keyword on ADRDSSU utility jobs**
- **Allows backups, restores, compresses, etc. on all target datasets without checking access for each dataset**

RACF COMMANDS & UTILITIES

Avoid use of commands and utilities that are I/O or processing intensive during peak system activity periods

LD ID(), PREFIX(), or DSNS

SR NOMASK, AGE, USER, or WARNING

LU * LG * RL class *

ICHDSM00 IRRDBU00 BLKUPD

IRRUT100 IRRUT200 IRRUT400

SETROPTS GENERIC(class) REFRESH [especially DATASET]

SETROPTS RACLIST(class) REFRESH [classes with many profiles]

Large batches of commands - especially CONNECTs & REMOVEs

Specify parameter NOYOURACC (or NOY) on RLIST commands to avoid retrieval and RACLIST processing of all grouping class profiles simply to determine your access

Keep the RACF database clean of unnecessary profiles