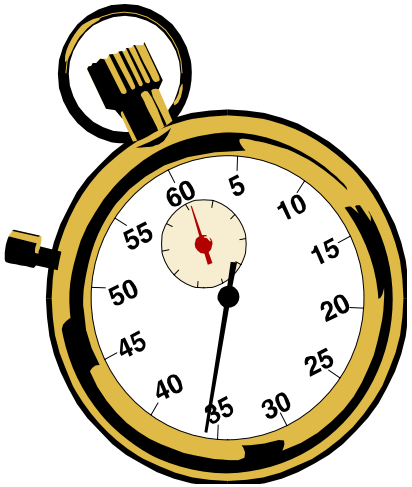
RACF PERFORMANCE TUNING

SHARE - August 2010



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**

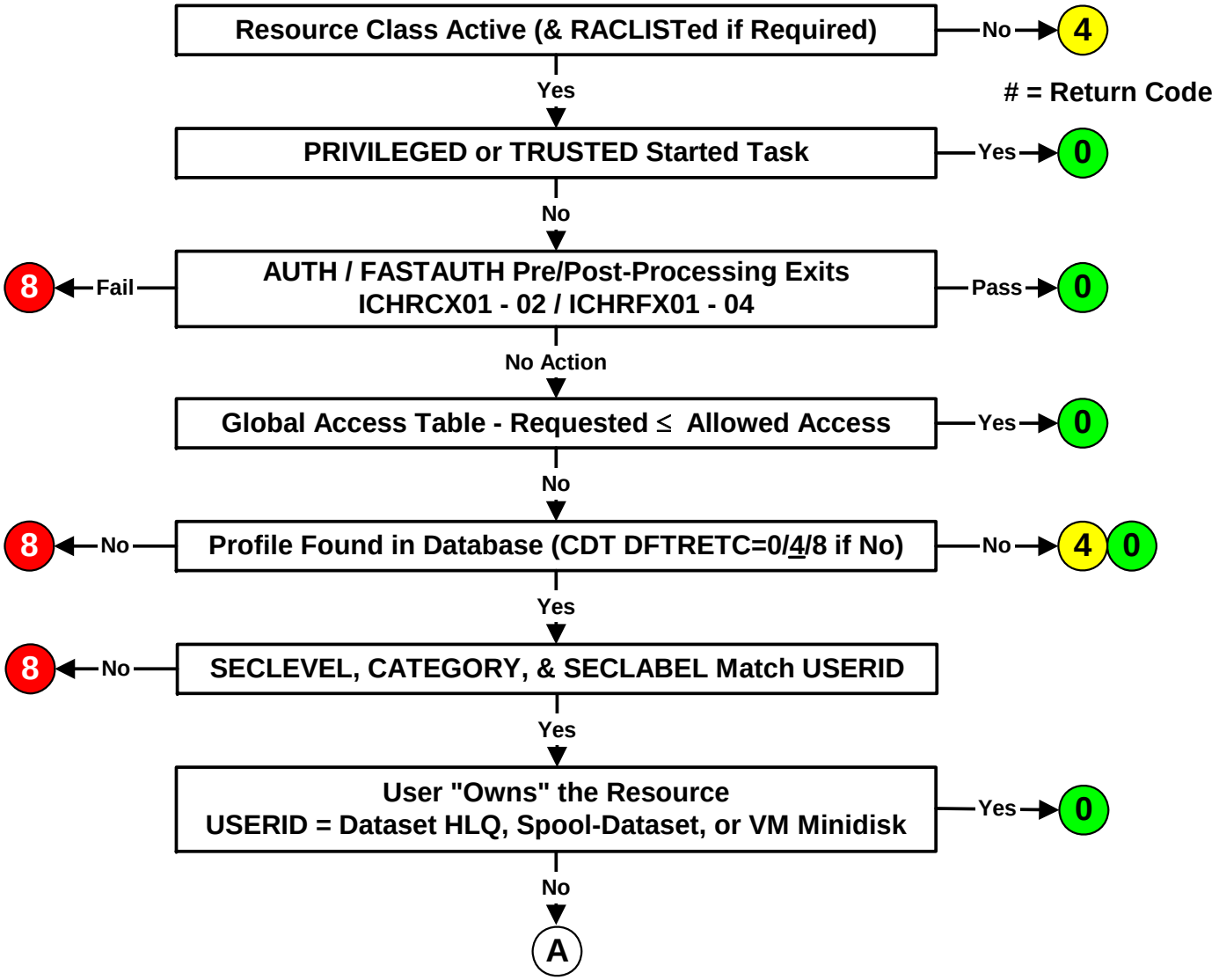
OBJECTIVES

Optimize Access Authorizations

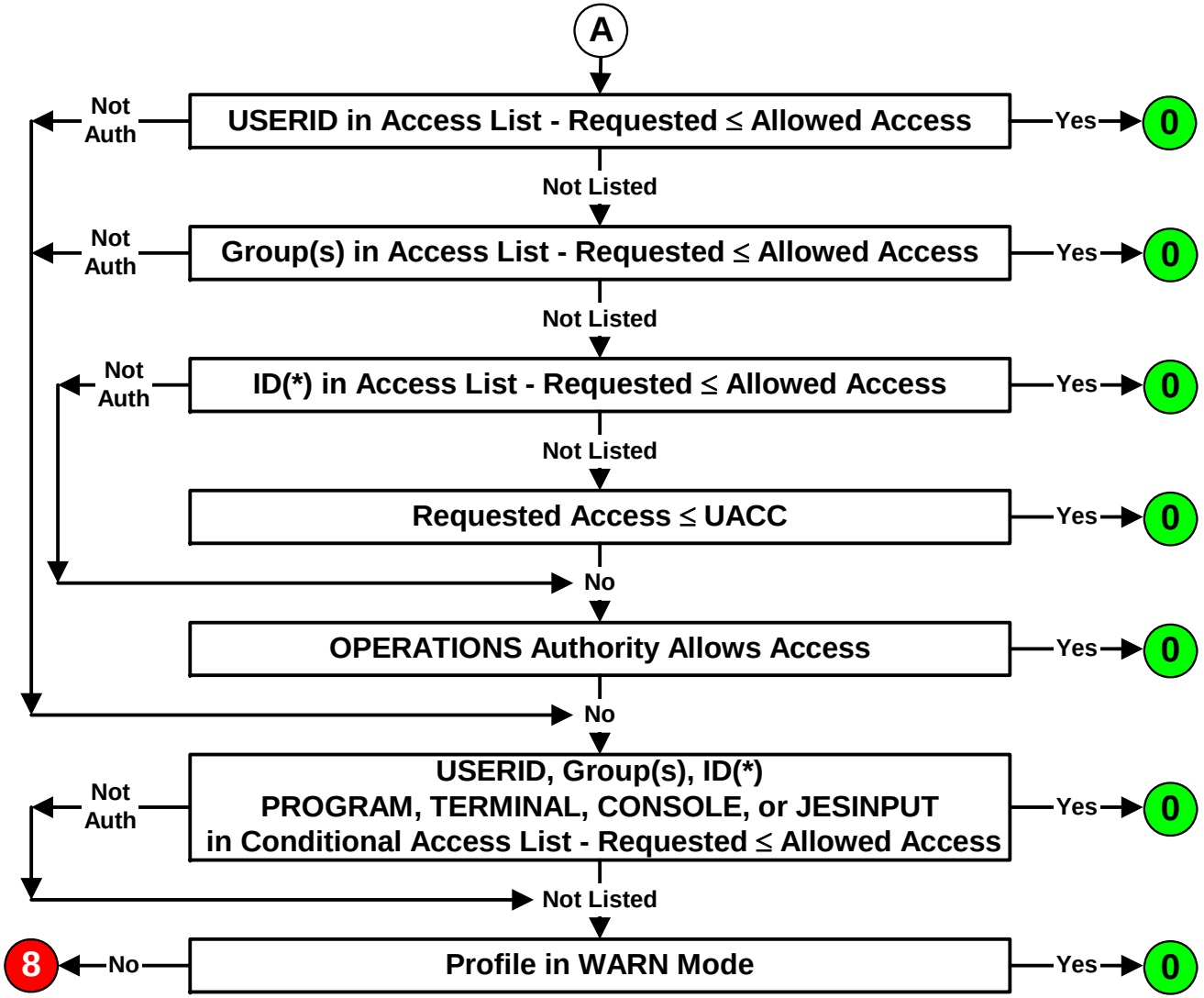
Expedite the Logon Process

Minimize I/O Operations

RACF 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**
- **Minimize the number of group connects per user**
- **Remove redundant entries (e.g., access allowed equals UACC)**
- **Remove obsolete residual entries - run IRRRID00**

Reduce reliance on OPERATIONS authority by implementing Storage Administration authorities

- **FACILITY STGADMIN profiles**
- **DASDVOL profiles**
- **ALTER access to Catalogs**

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 to improve performance
- 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.BRODCAST	READ	
DATASET	SYS1.HELP	READ	
DATASET	SYS1.MACLIB	READ	
DATASET	SYS1.RACF	NONE	(precludes access)
DATASET	SYS%.**	READ	(avoid - too broad)
JESJOBS	SUBMIT.*.&RACUID*. &RACUID	READ	
JESJOBS	CANCEL.*.&RACUID.*	ALTER	
JESSPOOL	*.&RACUID.**	ALTER	
JESSPOOL	*.*.$JESNEWS.**	READ	
FACILITY	STGADMIN.ARC.ENDUSER.**	READ	
TSOAUTH	JCL	READ	
TSOAUTH	RECOVER	READ	
OPERCMDS	MVS.MCSOPER.&RACUID	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

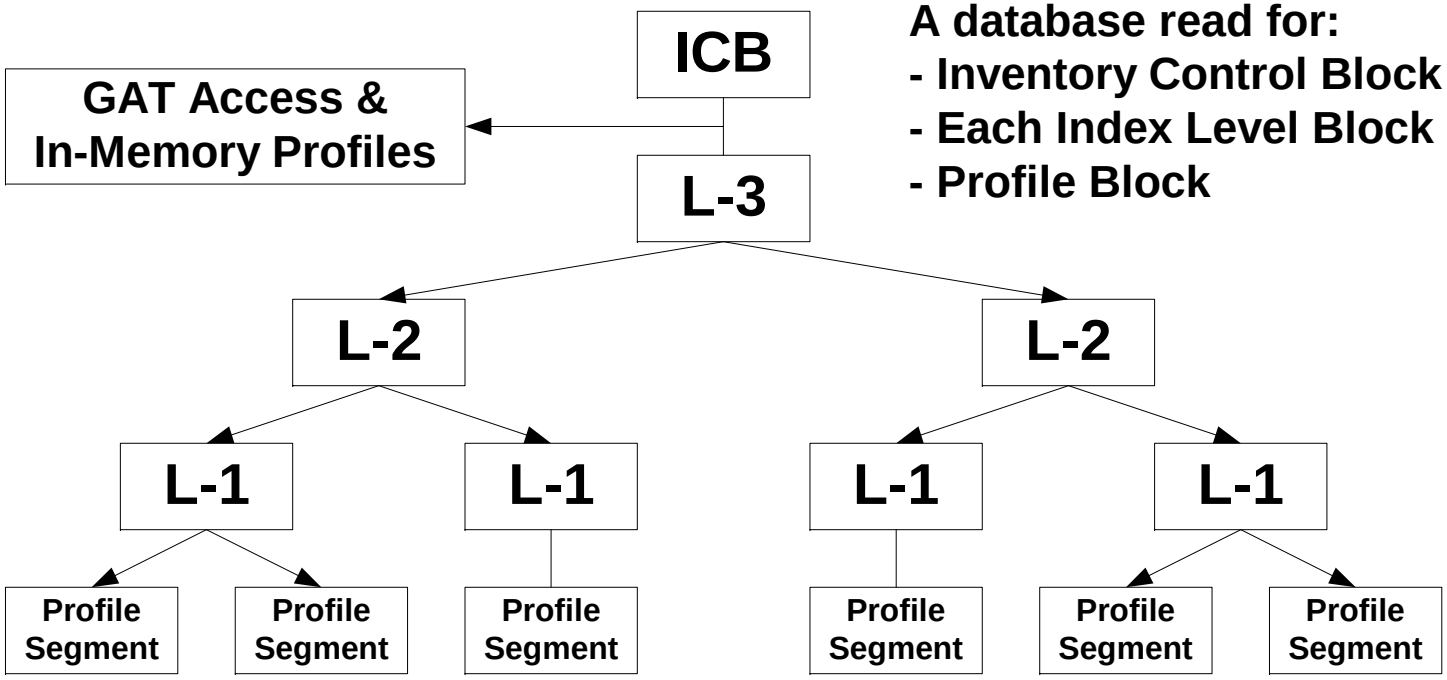
- **Not checked in RACROUTE REQUEST=FASTAUTH processing**
- **Not checked in RACROUTE REQUEST=VERIFY processing for APPL, TERMINAL, JESINPUT, CONSOLE, APPCPORT, and SERVAUTH resources**

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



A database read for:
- **Inventory Control Block (ICB)**
- **Each Index Level Block (L-n)**
- **Profile Block**

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) for processing

Frequently used blocks tend to stay in these buffers

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

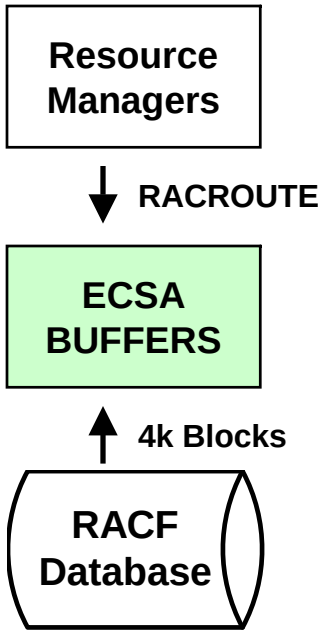
AL1(1)	Number of databases
CL44'RACF.PRIMARY'	Primary DB name
CL44'RACF.BACKUP'	Backup DB name
AL1(100)	# of Resident Data Blocks
XL1'xx'	Flags

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



DATABASE CACHING

RACF Sysplex Data Sharing

- Uses coupling facility as large store-through cache - caches ICB, index, & 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
- Cache size
  - Recommend enough to at least hold all index blocks and data blocks for non-RACFLISTed resource classes
  - To assist in calculating the coupling facility size for RACF, go to http://www.ibm.com/systems/support/z/cfsizer/racf/

RACLIST

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**

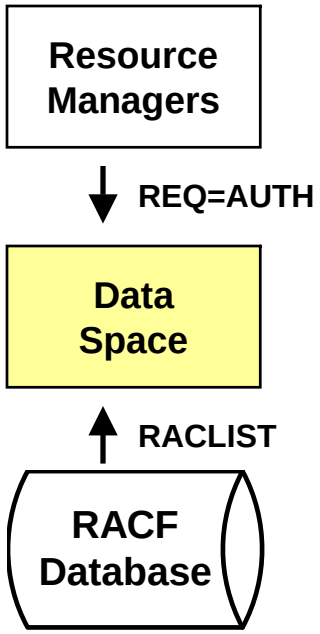
  CICS IMS DB2 MQSeries
- **Updated with SETROPTS RACLIST(class) REFRESH**

RACLIST-required classes

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	



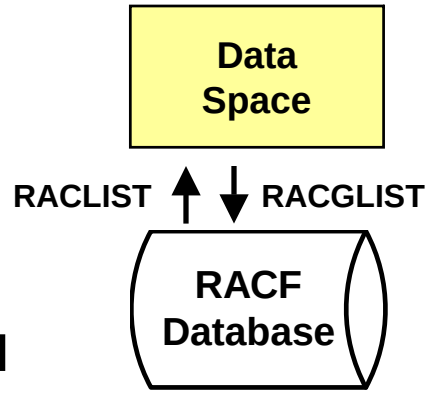
RACGLIST CLASS

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

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 U*nnn* and G*nnn*, 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, 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 <u>all</u> 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)
```

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 SHARING

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

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

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**

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

RACF COMMANDS

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