RSH CONSULTING

RACF Utilities

RSH RACF Forum

November 2024

RSH Consulting - Robert S. Hansel

RSH CONSULTING

RSH Consulting, Inc. is an IT security professional services firm 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. RSH's services include RACF security reviews and audits, initial implementation of new controls, enhancement and remediation of existing controls, and training.

- www.rshconsulting.com
- 617-969-9050

Robert S. Hansel is Lead RACF Specialist and founder of RSH Consulting, Inc. He began working with RACF in 1986 and has been a RACF administrator, manager, auditor, instructor, developer, and consultant. Mr. Hansel is especially skilled at redesigning and refining large-scale implementations of RACF using role-based access control concepts. He is a leading expert in securing z/OS Unix using RACF. Mr. Hansel has created elaborate automated tools to assist clients with RACF administration, database merging, identity management, and quality assurance.

- 617-969-8211
- R.Hansel@rshconsulting.com
- www.linkedin.com/in/roberthansel
- http://twitter.com/RSH_RACF

RACF Utilities

- Programs providing extended services within the RACF environment
  - Manage and maintain the RACF database and profiles
  - Load the command parsing table
  - Report on security status
  - Unload RACF profiles and SMF access events
- Information provided for each utility
  - Description and function
  - Sample JCL and execution options
  - Usage considerations and recommendations
  - References
- As a general rule, if multiple systems share a RACF database and are at different z/OS release and maintenance levels, use the utilities from the highest-level system
  - This is mandatory for IRRMIN00 and IRRUT400

RACF Utilities

- IRRMIN00 RACF Initialization Utility
- IRRIRA00 RACF Internal Reorganize Alias Utility
- IRRDPI00 RACF Dynamic Parse Initialization
- IRRUT100 RACF Cross Reference Utility
- IRRUT200 RACF Database Verification Utility
- BLKUPD RACF Block Update Utility (a.k.a. IRRUT300)
- IRRUT400 RACF Database Split/Merge/Extend Utility
- ICHDSM00 RACF Data Security Monitor (DSMON)
- IRRDBU00 RACF Database Unload Utility
- IRRRID00 RACF Remove ID Utility
- IRRADU00 RACF SMF Data Unload Utility
- RACFRW RACF Report Writer (not covered)
- Unsupported RACF utilities

IRRMIN00 - RACF Database Initialization Utility

- Initializes new databases and updates templates in existing databases
- PARM=
  - NEW - Initializes new RACF database
  - UPDATE - Installs new templates in existing RACF database
  - ACTIVATE - Activates new templates by loading them into storage (PTF changes)
- Sample JCL

```
//jobname  JOB (account),'username',CLASS=x,MSGCLASS=x
//UPDATE   EXEC PGM=IRRMIN00,PARM=UPDATE
//STEPLIB  DD DSN=SYS1.LINKLIB,DISP=SHR,       <- Library with newest IRRMIN00
//    UNIT=unit,VOL=SER=volser
//SYSPRINT DD SYSOUT=*
//SYSRACF  DD DSN=racf.database,DISP=SHR       <- Database dataset to be updated
```

- Considerations
  - PARM=NEW will not overwrite a RACF database in use on the system where it is executed
  - When updating templates, update them on all primary and backup database datasets
  - Requires ALTER (for NEW) or UPDATE (for UPDATE) access authority to the target database
  - If using RACF sysplex data sharing, execute on a system in the sysplex group
  - SYSRACF must not be allocated in the extended addressing area of a DASD volume (EATTR)

IRRMIN00 - RACF Database Initialization Utility

- RACF templates define record layouts for RACF profiles and segments
  - New releases of RACF and RACF APARs sometimes introduce template changes
- RACF templates reside in memory and in both Primary and Backup databases
  - During RACF initialization, RACF compares the templates in the Primary database with those embedded in its software, specifically module IRRTEMP2
  - If the Primary database templates are down-level, RACF loads IRRTEMP2's version of the templates into memory and issues a console message like the following

```
ICH579E RACF TEMPLATES ON DATABASE ARE DOWNLEVEL:
        HRF77C0 00000223.00000050; USING TEMPLATES AT LEVEL
        OA59196 00000243.00000050 FROM IRRTEMP2.
        RUN IRRMIN00 PARM=UPDATE.
```

  - The RACF console command SET LIST shows the template version in memory, which may not be the template version in the database
- The templates in the Backup database should be kept at the same level as those in the Primary database
- Utility IRRMIN00 should be run with PARM=UPDATE for all databases soon after RACF upgrades and maintenance if not beforehand
- Reference: RACF Systems Programmer's Guide

IRRIRA00 - RACF Internal Reorganize Aliases Utility

- Converts a RACF database the Application Identity Mapping (AIM) structure
  - Adds alias index structure for identity mapping
  - Replaces use of pre-existing identity mapping profiles in classes UNIXMAP (z/OS Unix), NOTELINK (Lotus Notes) and NDSLINK (Novell Directory Services)
- Reasons for converting
  - More efficient lookup of application identities (VLF caching with IRRUMAP and IRRGMAP for z/OS Unix is still recommended for performance)
  - Enables use of SEARCH on UID and GID
  - Alias index requires less space than mapping profiles in both the RACF database and the IRRDBU00 unload
  - Required to implement ...
    - UNIXPRIV SHARED.IDS Stage 2
    - FACILITY BPX.NEXT.USER Stage 2
    - FACILITY BPX.UNIQUE.USER Stage 3
- Reference: RACF Systems Programmer's Guide

IRRIRA00 - RACF Internal Reorganize Aliases Utility

- AIM conversion stages (indicator set in ICB record and RCVT):
  - 0 = Pre-conversion, uses mapping profiles for lookups if mapping class is active
  - 1 = Uses mapping profiles for lookups, builds and maintains alias index
  - 2 = Uses alias index then mapping profiles for lookups; maintains mapping profiles
  - 3 = Uses alias index for lookups and deletes mapping profiles

- Sample JCL

```
//jobname  JOB (account),'username',CLASS=x,MSGCLASS=x
//STEP     EXEC PGM=IRRIRA00,PARM=STAGE(1)                  <- Advance to Stage 1
//SYSPRINT DD  SYSOUT=*
```

- PARM=
  - No parms specified: displays current stage
  - STAGE(1 | 2 | 3): upgrade by one level to the level specified

- Must pass through all stages, converting from one stage to the next, one stage at a time, until reaching stage 3; cannot retreat to a previous stage

IRRIRA00 - RACF Internal Reorganize Aliases Utility

- Requires UPDATE access authority to the live RACF database (primary and backup)
  - Runs only against the live database
- Considerations and recommendations
  - Ensure IRRMIN00 PARM=UPDATE was run and system was IPLed before going to stage 1
  - Verify the database has sufficient free space for the new index before converting
  - Maximum of 129 8-character IDs can share an identity (e.g., UID(0))
    - Identify and reduce excessive sharing before conversion; otherwise, IRRIRA00 will fail
  - IRRIRA00 assumes UNIXMAP profiles are accurate and builds the index from them
    - Either identify and fix UNIXMAP profile / OMVS segment mismatch errors before the conversion or run IRRUT400 immediately afterwards to rebuild the index
  - If using RACF sysplex data sharing, execute on a system in the sysplex group
  - IRRIRA00 obtains exclusive use of the database during execution
    - Run during non-peak periods and suspend RACF administrative work
    - Stage 1 conversion may take a long time if there are many mapping profiles
    - To expedite, deactivate backup and create a new backup from the converted primary when done
  - Do the entire conversion (0-to-1, 1-to-2, 2-to-3) in a single session
  - Make an IRRUT200 backup before starting the conversion
    - Check for integrity errors before proceeding
  - Deactivate mapping classes upon reaching stage 3

IRRDPI00 - RACF Dynamic Parse Initialization Utility

- Dynamic parse table provides segment field layouts for RACF commands
- IRRDPI00 is a TSO command that builds the parsing table in memory
- Should be run on all systems sharing a RACF database …
  - At IPL - by a Started Task or the RACF subsystem
  - After activating new templates (IRRMIN00 ACTIVATE) if required by a PTF
  - After creating or modifying Custom Field CFIELD class profile CFDEF segments
- Executed by either …
  - Started Task - typically named IRRDPTAB
  - RACF subsystem (preferred method at IPL)

```
//RACF     PROC
//RACF     EXEC PGM=IRRSSM00,REGION=0M,PARM='OPT=xx'
//RACFPARM DD DSN=racf.parm.library,DISP=SHR
```

    Member IRROPTxx in RACFPARM library contains command IRRDPI00
  - Batch job (preferred method for CFIELD updates)
  - TSO user at READY prompt
    - Requires program IRRDPI00 to be APF-authorized via PARMLIB member IKJTSOxx

IRRDPI00 - RACF Dynamic Parse Initialization Utility

- To execute, requires ...
  - Either …
    - READ access to PROGRAM class profile for IRRDPI00
      - Define and restrict access to PROGRAM profile IRRDPI00 if backstop * or ** profile exists
    - READ access to FACILITY class profile IRRDPI00 if ...
      - PROGRAM not protected, and
      - Executing in TSO
    - RACF SPECIAL if neither profile is defined
    - Recommendation: Define profile IRRDPI00 in both the PROGRAM and FACILITY class and permit Started Task IRRDPTAB, RACF Admins, and z/OS Tech Support access
  - And …
    - READ access to table source code referenced in DD SYSUT1 - usually SYS1.SAMPLIB(IRRDPSDS)
- IRRDPI00 options
  - CHECK - Verifies syntax of CFDEF definitions - run before UPDATE
  - UPDATE - Verifies syntax of CFDEF definitions and updates the table
  - LIST [(profile-type [segment-name [keyword-name]])] - List entries

IRRDPI00 - RACF Dynamic Parse Initialization Utility

- Executed in batch (for Started Task, replace JOB with PROC statement)

```
//jobname  JOB (account),'username',CLASS=x,MSGCLASS=x
//STEP0001 EXEC PGM=IKJEFT01,REGION=2M,
//    PARM='IRRDPI00 UPDATE'
//SYSTSPRT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSUT1   DD DSN=SYS1.SAMPLIB(IRRDPSDS),DISP=SHR
//SYSTSIN DD DUMMY
```

- Executed as a TSO command or by RACF subsystem (IRROPTxx)

```
ALLOCATE FILE(SYSUT1) DATASET('SYS1.SAMPLIB(IRRDPSDS)') SHR
IRRDPI00 UPDATE
FREE FILE(SYSUT1)
```

- SYS1.SAMPLIB(IRRDPSDS) contains the parsing table entries included with RACF

- Reference: RACF Systems Programmer's Guide

IRRDPI00 - RACF Dynamic Parse Initialization Utility

- Validate CFIELD profiles before attempting to add them to the table

```
//xxxxxxxx JOB job card fields
//STEP0010 EXEC PGM=IKJEFT01,REGION=0M,PARM='IRRDPI00 CHECK'
//SYSTSPRT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSUT1   DD  DSN=SYS1.SAMPLIB(IRRDPSDS),DISP=SHR
//SYSTSIN  DD  DUMMY
```

- Install CFIELD profiles if validation is successful

```
//xxxxxxxx JOB job card fields
//STEP0010 EXEC PGM=IKJEFT01,REGION=0M,PARM='IRRDPI00 UPDATE'
//SYSTSPRT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSUT1   DD  DSN=SYS1.SAMPLIB(IRRDPSDS),DISP=SHR
//SYSTSIN  DD  DUMMY
```

- Optionally list CFIELD definitions in the Command Parsing Table

```
//xxxxxxxx JOB job card fields
//STEP0010 EXEC PGM=IKJEFT01,REGION=0M,PARM='IRRDPI00 LIST (USER CSDATA)'
//SYSTSPRT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSUT1   DD  DSN=SYS1.SAMPLIB(IRRDPSDS),DISP=SHR
//SYSTSIN  DD  DUMMY
```

IRRUT100 - RACF Cross Reference Utility

- Lists all references to specified USERIDs and/or groups in the RACF database in …
  - Standard and Conditional access lists
  - NOTIFY and OWNER fields
  - Group memberships for users
  - Dataset profiles with matching High Level Qualifier (HLQ)
- Does not list instances where …
  - ID is embedded in qualifiers in general resource profiles
  - ID is embedded in dataset profile qualifiers except the HLQ
  - ID is in APPLDATA field (e.g., BPX.DEFAULT.USER)
- Authority to execute
  - System-level SPECIAL / AUDITOR / ROAUDIT - can report on any ID or group
  - Group-level SPECIAL / AUDITOR can report on IDs and groups within their scope
  - Users can generate a report on their own USERID
- Usage Notes
  - Works only with the current (active) RACF database
  - Serializes on one profile at a time - reads every profile
  - Can list up to 1,000 IDs and/or groups

IRRUT100 - RACF Cross Reference Utility

```
//jobname  JOB (account),'username',CLASS=x,MSGCLASS=x
//IRRUT100 EXEC PGM=IRRUT100
//SYSUT1   DD   UNIT=SYSDA,SPACE=(CYL,(10,1))              <<< Work dataset
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
JSMITH1  DJONES
/END                                                      <<< Optional
```

SYSPRINT output

```
OCCURRENCES OF JSMITH1

IN STANDARD ACCESS LIST OF DATASET PROFILE SYSTEST.FDR.OBJLIB
IN STANDARD ACCESS LIST OF DATASET PROFILE SYS1.DASDUTIL.LIB (G)
IN ACCESS LIST OF GROUP USERGRPA
OWNER OF PROFILE HSM.BACKUP.WEEKLY (G)
IN STANDARD ACCESS LIST OF PROGRAM PROFILE AMASPZAP
FIRST QUALIFIER OF PROFILE JSMITH1.TEST.* (G)
USER ENTRY EXISTS

(G) - ENTITY NAME IS GENERIC
```

- Recommendation: Use sparingly if at all
- Reference: RACF Security Administrator's Guide

IRRUT200 - RACF Database Verification Utility

- Identifies inconsistencies in the internal organization of a RACF database
  - Identifies problems with the index-block chains
  - Verifies index entries point to the correct profile
  - Validates and reports errors found in Relative Byte Addresses (RBAs) of all profile segments
  - Reports inconsistencies in the block segments of the database that are actually in use as compared to the block segments listed as allocated in Block Availability Mask (BAM) blocks
  - Validates the database format
  - Issues return codes to indicate validation errors
- Makes an exact, block-by-block, copy of the RACF database
- Optionally creates a formatted index report displaying the 255-byte profile name and profile type information
- Reference: RACF System Programmer's Guide

IRRUT200 - RACF Database Verification Utility

```
//jobname  JOB (account),'username',CLASS=x,MSGCLASS=x
//STEP     EXEC PGM=IRRUT200
//SYSRACF  DD  DSN=SYS1.RACF,DISP=SHR                    <- Input RACF Database
//SYSUT1   DD  DSN=INFOSEC.RACF.COPY,                    <- Output copy or temp file
//             UNIT=SYSDA,SPACE=(CYL,(200),,CONTIG),     <- Match SYSRACF SPACE
//             DCB=(LRECL=4096,RECFM=F)
//SYSUT2   DD  SYSOUT=*                                  <- Output messages
//SYSPRINT DD  SYSOUT=*                                  <- Output report
//SYSIN    DD  *                                         <- Control statements
INDEX FORMAT
MAP ALL
END
```

- SYSIN control statements
  - INDEX [FORMAT] Performs index scan function; FORMAT adds a report of all index blocks
  - MAP [ALL] Maps BAM/allocation; ALL adds an encoded map for each BAM block
  - END Terminates the utility
- To obtain an abbreviated summary listing of just the database status, its percentage full, and number of profiles by class, simply specify:

  MAP

  END
- Requires READ authority to the RACF database being verified

IRRUT200 - RACF Database Verification Utility

- Sample Output (Index Block Verification)

```
TOTAL NUMBER OF NAMES IN RACF DATA SET 00016699
TOTAL NUMBER OF INDEX BLOCKS IN RACF DATA SET 00000313
AVERAGE NUMBER OF NAMES PER INDEX BLOCK 053
AVERAGE NAME LENGTH 007
AVERAGE NUMBER OF UNUSED BYTES PER INDEX BLOCK 2277
TOTAL NUMBER OF LEVEL 01 BLOCKS IN RACF DATA SET 00000305
```

- Sample Output (BAM Block Verification)

```
NUMBER OF BAM BLOCKS DEFINED 006
LAST BAM THAT DEFINES USED SPACE - RBA 00000000D000
RACF DATA SET IS  24 PERCENT FULL.
TOTAL NUMBER OF INDEX BLOCKS IN RACF DATA SET 00000313
TOTAL NUMBER OF LEVEL 01 BLOCKS IN RACF DATA SET 00000305
NUMBER OF GROUP    ENTRIES - 0002174
NUMBER OF USER     ENTRIES - 0003862
NUMBER OF DATASET  ENTRIES - 0004500
NUMBER OF $CHGMAN  ENTRIES - 0000108
NUMBER OF $OMEGCIC ENTRIES - 0000003
NUMBER OF $OMEGDB2 ENTRIES - 0000003
NUMBER OF $OMEGMVS ENTRIES - 0000021
NUMBER OF ACCTNUM  ENTRIES - 0000002
NUMBER OF CDT      ENTRIES - 0000022
```

IRRUT200 - RACF Database Verification Utility

- Sample Output (BAM Block Verification - with error)

```
                                                ****  BAM BLOCK VERIFICATION  ****
          ****  SYMBOL LEGEND  ****

          *    BAM=ALLOC    , ACTUAL=ALLOC
          0    BAM=UNALLOC  , ACTUAL=UNALLOC
          .    BAM=ALLOC    , ACTUAL=UNALLOC
          +    BAM=UNALLOC  , ACTUAL=ALLOC
          I    INDEX BLOCK WITH LEVEL IN NEXT POSITIONS
          B    BAM BLOCK
          T    TEMPLATE BLOCK
          S    SEGMENT TABLE BLOCK
          F    FIRST BLOCK (ICB)
          -    BAM=UNALLOC  , ACTUAL=ALLOC  I,B,OR F BLK
          $    BAM=UNALLOC  , ACTUAL=ALLOC  SPECIAL BLK
          ?    BAM=ALLOC    , ACTUAL=ALLOC  UNKNOWN BLK
          %    BAM=UNALLOC  , ACTUAL=ALLOC  UNKNOWN BLK
          @    BAM=ALLOC    , DUPLICATE ALLOCATION
          #    BAM=UNALLOC  , DUPLICATE ALLOCATION
          /    UNDEFINED STORAGE
-BLOCK 009 RBA 000000015000
 014 .******* ******** ******** ******** ******** ******** ******** ******** ********
                                  ****   LOCATIONS WITH POSSIBLE CONFLICTS  ***

  BAM BLOCK 009  BYTE 0014  BIT 0  RBA 0000047A6000...
```

IRRUT200 - RACF Database Verification Utility

- Considerations and Recommendations
  - Has exclusive use of the database during the copy phase - avoid using during peak work periods
  - If merely analyzing an active database, always specify a work dataset on the SYSUT1 DD to make a temporary copy for analysis to avoid holding a RESERVE on the database throughout the entire analysis phase
  - If using RACF sysplex data sharing, execute on a system in the sysplex group
  - Device, space, and DCB parameters on SYSUT1 should be exactly the same as the input RACF database; use SPACE=(type(n),,CONTIG)
  - SYSUT1 must not be allocated in the extended addressing area of a DASD volume (EATTR)
  - Optional PARM=ACTIVATE can copy the in-use active primary to the in-use but inactive backup and then automatically RVARY ACTIVE the backup - avoids losing any updates
    - If sharing the database and not using RACF Sysplex Communications, RVARY ACTIVE must be executed on the other systems to activate the backup after running IRRUT200
  - Optional PARM=RENAMEACTIVATE(dsname) is like ACTIVATE but renames the backup
  - Regularly review the percentage full to avoid running out of space

- IRRUT200 creates reliable RACF backups by suspending updates while copying
  - Copies made using other backup utilities may have errors due to updates made during copying
  - For split database, each database dataset must be backed up individually

IRRUT200 - RACF Database Verification Utility

- RACF Database Backup Procedure - Pre-allocate a permanent dataset on an isolated DASD volume with the exact same characteristics as Primary RACF dataset - becomes standby, off-line backup and is used for generating IRRDBU00 and adjunct RACF administration product unloads

```
//jobname  JOB (account),'username',CLASS=x,MSGCLASS=x
//STEP0001 EXEC PGM=IRRUT200    CREATE BACKUP OF PRIMARY DATABASE DATASET
//SYSPRINT DD  SYSOUT=*
//SYSRACF  DD  DSN=racf.primary.database.dsname,DISP=SHR
//SYSUT1   DD  DSN=racf.dataset.copy.dsname,DISP=OLD
//SYSUT2   DD  SYSOUT=*
//SYSIN    DD  DUMMY
//STEP0002 EXEC PGM=IRRUT200    ANALYZE BACKUP
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=racf.dataset.copy.dsname,DISP=SHR
//SYSUT2   DD  SYSOUT=*
//SYSIN    DD  *
 INDEX
 MAP
 END
//STEP0003 EXEC PGM=SORT    COPY BACKUP TO GDG - AT LEAST 30 GENERATIONS
//SYSOUT   DD  SYSOUT=*
//SORTIN   DD  DSN=racf.dataset.copy.dsname,DISP=SHR
//SORTOUT  DD  DSN=racf.dataset.copy.dsname.tape(+1),
//             DISP=(NEW,CATLG,DELETE),UNIT=CART,
//             LABEL=RETPD=366,DCB=(RECFM=FB,LRECL=4096,BLKSIZE=0)
//SYSIN    DD  *
 SORT FIELDS=COPY
```

BLKUPD - RACF Database Block Update Command

- BLKUPD (a.k.a., IRRUT300)
  - APF-authorized TSO command
  - Examines and modifies blocks in a RACF database
  - Enables correction of inconsistencies found by IRRUT200 in the RACF database
- Requires UPDATE access to the RACF database being fixed
- Use with extreme caution and only if RACF commands fail to fix the problem
  - Before using, make a backup of the database in case of a mistake
  - Test the fix on a copy of the data base, vary the fixed copy active, confirm the fix was correct, then fix the “active” data base
- Best to use BLKUPD with assistance and guidance from RACF Level II support
- Reference: RACF Diagnosis Guide

IRRUT400 - RACF Database Split/Merge/Extend Utility

- Copies a RACF database to a larger or smaller sized database
- Copies a database to a different device type
- Redistributes data amongst split RACF database datasets
- Identifies inconsistencies (e.g., duplicate profiles in split database)
- Reorganizes the database and corrects errors
  - Reduces fragmentation by bringing all segments of a profile together and, optionally, kept within a block boundary
  - Compresses index
  - Rebuilds all index blocks - profile and AIM (corrects errors)
  - Rebuilds Block Availability Mask (BAM) blocks (corrects errors)

IRRUT400 - RACF Database Split/Merge/Extend Utility

- PARM=
  - LOCKINPUT | NOLOCKINPUT | UNLOCKINPUT
    - (no default - one must be specified)
    - LOCKINPUT - Locks the input data set to prevent updates and remains locked after execution ends; prevents first logon of the day statistic updates and password changes
    - NOLOCKINPUT - Does not change the status of the database
      - If the live database is used for input, changes made during execution could result in a corrupted copy
    - UNLOCKINPUT - Unlocks database that was previously locked
  - FREESPACE(nn) | NOFREESPACE
    - FREESPACE(nn) - specifies percentage of freespace to be left within the index blocks to accommodate future growth; 'nn' is 0 to 50 - recommended with at least 10
    - NOFREESPACE - is equivalent to FREESPACE(0)
  - ALIGN | NOALIGN
    - ALIGN - forces profiles and segments that occupy multiple 256-byte slots to be placed so they do not span 4096 physical blocks, thereby requiring less I/O to retrieve - recommended
    - NOALIGN - causes no special alignment

IRRUT400 - RACF Database Split/Merge/Extend Utility

- PARM= (continued)
  - RANGE(member) | TABLE(table-name) | NOTABLE
    - RANGE(member) - 'member' is the PDS member with the IRROPTxx-style dataset range table to be used for splitting the database
      - Optional RANGEDD DD statement can specify the dataset where the member is to be found; otherwise, RACF searches for the member in the current system PARMLIB concatenation
    - TABLE(table-name) - 'table-name' is the IRRRNG-style dataset range table load module to be used for splitting the database
      - Optional STEPLIB DD statement can specify the APF-authorized dataset where the table is to be found; otherwise, RACF searches for the module in LINKLIB
    - Ensure the ranges specified in the IRROPTxx member or IRRRNG module used for IPL match those used to create the split database
    - NOTABLE - all profiles written to OUTDD1 dataset
  - DUPDATASETS | NODUPDATASETS
    - DUPDATASETS - all duplicate dataset profiles are allowed and processed
    - NODUPDATASETS - if duplicate dataset profiles are found on multiple input databases when merging databases, only the profile from the lowest numbered input database, identified by INDDxx, is retained

IRRUT400 - RACF Database Split/Merge/Extend Utility

- Reorganizing a copy of a Database

```
//jobname  JOB (account),'username',CLASS=x,MSGCLASS=x
//COPYDB   EXEC PGM=IRRUT400,PARM='NOLOCKINPUT,FREESPACE(20),ALIGN'
//SYSPRINT DD  SYSOUT=*
//INDD1    DD  DSN=SYS1.RACF.COPY,DISP=OLD
//OUTDD1   DD  DSN=SYS2.RACF.REORG,DISP=(,KEEP),VOL=SER=RACVL2,
//             SPACE=(CYL,100,,CONTIG),DCB=DSORG=PSU,UNIT=SYSDA
```

- Splitting a Database

```
//jobname  JOB (account),'username',CLASS=x,MSGCLASS=x
//SPLITDB  EXEC PGM=IRRUT400,PARM='NOLOCKINPUT,TABLE(NEWRNG)'
//SYSPRINT DD  SYSOUT=*
//INDD1    DD  DSN=SYS1.RACF.COPY,DISP=OLD
//OUTDD1   DD  DSN=SYS2.RACF1,DISP=(,KEEP),
//             UNIT=SYSDA,VOL=SER=VOL1,DCB=DSORG=PSU,
//             SPACE=(CYL,50,,CONTIG)
//OUTDD2   DD  DSN=SYS2.RACF2,DISP=(,KEEP),
//             UNIT=SYSDA,VOL=SER=VOL2,DCB=DSORG=PSU,
//             SPACE=(CYL,50,,CONTIG)
//OUTDD3   DD  DSN=SYS2.RACF3,DISP=(,KEEP),
//             UNIT=SYSDA,VOL=SER=VOL3,DCB=DSORG=PSU,
//             SPACE=(CYL,50,,CONTIG)
//STEPLIB  DD  DSN=INSTALL.LINKLIB,DISP=SHR
```

IRRUT400 - RACF Database Split/Merge/Extend Utility

- Merging a split Database (first run)

```
//jobname  JOB (account),'username',CLASS=x,MSGCLASS=x
//MERGEDB  EXEC PGM=IRRUT400,PARM='NOLOCKINPUT,DUPDATASETS'
//SYSPRINT DD  SYSOUT=*
//INDD1    DD  DSN=SYS1.RACF1.COPY,DISP=OLD
//INDD2    DD  DSN=SYS1.RACF2.COPY,DISP=OLD
```

  - First make a test run to identify any possible inconsistencies
  - Dataset entries with identical names, but from different RACF databases, are allowed
  - Correct any inconsistencies and continue with the merge

- Merging a split Database (second run)

```
//jobname  JOB (account),'username',CLASS=x,MSGCLASS=x
//DBMERGE  EXEC PGM=IRRUT400,
//            PARM='NOLOCKINPUT,NODUPDATASETS,FREESPACE(10),ALIGN'
//SYSPRINT DD SYSOUT=*
//INDD1    DD DSN=SYS1.RACF1.COPY,DISP=OLD
//INDD2    DD DSN=SYS1.RACF2.COPY,DISP=OLD
//OUTDD1   DD DSN=SYS2.RACF,DISP=(,KEEP),UNIT=SYSDA,VOL=SER=VOL001,
//            DCB=DSORG=PSU,SPACE=(CYL,100,,CONTIG)
```

IRRUT400 - RACF Database Split/Merge/Extend Utility

- Locking and copying a live Database to a larger Database (Extend)

```
//jobname  JOB (account),'username',CLASS=x,MSGCLASS=x
//EXTEND   EXEC PGM=IRRUT400,PARM='LOCKINPUT,FREESPACE(10),ALIGN'
//SYSPRINT DD  SYSOUT=*
//INDD1    DD  DSN=SYS1.RACF,DISP=OLD
//OUTDD1   DD  DSN=SYS2.RACF,DISP=(,KEEP),UNIT=SYSDA,VOL=SER=VOL1,
//             DCB=DSORG=PSU,SPACE=(CYL,150,,CONTIG)
```

- Unlocking a locked Database

```
//jobname  JOB (account),'username',CLASS=x,MSGCLASS=x
//UNLOCK   EXEC PGM=IRRUT400,PARM='UNLOCKINPUT'
//SYSPRINT DD  SYSOUT=*
//INDD1    DD  DSN=SYS1.RACF,DISP=OLD
```

  - Does not make a copy - simply unlocks the database

IRRUT400 - RACF Database Split/Merge/Extend Utility

- Requires UPDATE access to the input RACF database
- Output database cannot be the active RACF database
- OUTDDn must not be allocated in the extended addressing area of a DASD volume (EATTR)
- Recommendations
  - Run at a time when updates are not likely to be made to the RACF database
  - Process sequence
    - Copy the database using IRRUT200 and verify integrity
    - Reorganize using this copy so LOCKINPUT is not required
    - RVARY the reorganized copy online to become the active primary, then use IRRUT200 PARM=ACTIVATE to copy the reorganized primary to the backup
- Not intended to merge RACF databases from different systems
- Reference: RACF System Programmer's Guide

ICHDSM00 - RACF Data Security Monitor (DSMON)

- Produces reports on the status of …
  - System and security environment
  - Certain RACF controls

- Requires
  - Either …
    - READ access to PROGRAM profile ICHDSM00
      - Define and restrict access to PROGRAM profile ICHDSM00 if backstop * or ** profile exists
        - Permit access to RACF Admins, Auditors, and Tech Support
    - System-level AUDITOR or ROAUDIT if program is not defined
  - And …
    - READ access to FACILITY ICHDSM00.SYSCAT to list user catalogs with function SYSCAT or ALL (allowed if not defined)

- Run on every system to collect system-specific configuration information

- Reference: RACF Auditor's Guide

ICHDSM00 - RACF Data Security Monitor (DSMON)

```
//jobname  JOB (account),'username',CLASS=x,MSGCLASS=x
//DSMON    EXEC PGM=ICHDSM00
//SYSPRINT DD  SYSOUT=*                          <- DSMON report
//SYSUT2   DD  SYSOUT=*                          <- Messages
//SYSIN    DD  *
  FUNCTION option
  LINECOUNT 55 | 0 | 40 - 99                     < 0 = page break only at the beginning
                                                       of each report
  USEROPT USRDSN dsname(s) [VOL=volser(s)]       <- User specified dataset
  USEROPT RACGRP SYS1 | group                    <- Partial group tree
```

- Function options (in report generated sequence)

SYSTEM	**System and RACF identification**	**[Recommendation - specify with every execution]**
SYSPPT	**Program Properties Table (PPT) entries**	
RACAUT	**RACF Authorized Caller Table entries**	
RACEXT	**RACF Exits**	
RACUSR	**RACF User Attribute Summary Report**	
RACSPT	**RACF STARTED Class and Started Task Table entries**	
RACCDT	**RACF Class Descriptor Table entries and status**	
RACGAC	**RACF Global Access Checking Table Report**	
RACGRP	**RACF Group Tree**	
SYSAPF	**APF library protection**	
SYSLNK	**Linklist library protection**	
SYSSDS	**System dataset report**	
SYSCAT	**Catalog dataset protection**	
RACDST	**RACF database protection**	
ALL	**Produces all reports listed above**	

IRRDBU00 - RACF Database Unload Utility

- Unloads the RACF database into a sequential text dataset where the output can be used to extract information by …
  - Browsing directly as is
  - Processing by installation-developed report programs (e.g., REXX, SAS, ICETOOL)
    - DFSORT and ICETOOL - see SYS1.SAMPLIB(IRRICE) for samples
  - Loading to a database manager such as DB2 for SQL queries
- Performs diagnostic checks down to the field level - run periodically just to check for data errors
- Input to the utility can be …
  - The active Primary or Backup RACF Database
    - Must read and process every record - can degrade performance
    - If using RACF sysplex data sharing, execute on a system in the sysplex group
  - A copy of a RACF Database (preferred)
    - May fail if the copy is not allocated as contiguous (CONTIG)

IRRDBU00 - RACF Database Unload Utility

```
//jobname  JOB (account),'username',CLASS=x,MSGCLASS=x
//UNLOAD   EXEC PGM=IRRDBU00,PARM=NOLOCKINPUT
//INDD1    DD  DSN=input.racf.database,DISP=SHR            <- Input RACF database - DASD
//OUTDD    DD  DSN=output.racf.unload,UNIT=DISK,           <- Output RACF unload
//             DISP=(NEW,CATLG,DELETE),DCB=(RECFM=VB,LRECL=4096),
//             SPACE=(CYL,(100,50),RLSE),
//SYSPRINT DD  SYSOUT=*
```

- PARM= (no default - one must be specified)
  - LOCKINPUT - Locks the input data set to prevent updates; remains locked after execution ends
  - NOLOCKINPUT - Does not change the status of the database
  - UNLOCKINPUT - Unlocks database that was previously locked
- Requires access to the source RACF database ...
  - UPDATE - If using LOCKINPUT or UNLOCKINPUT
  - READ - If using NOLOCKINPUT
- With split RACF database, can specify multiple input datasets with INDDn statements to create a combined unload file OUTDD; however, ...
  - The datasets must be listed in the same order as is specified in the dataset name table, and
  - The range table on the system where IRRDBU00 is executed must be identical to that used in creating the split database; otherwise, each dataset must be unloaded separately

IRRDBU00 - RACF Database Unload Utility

- Results are affected by the system where IRRDBU00 is executed
  - Uses Enhanced Generic Naming (EGN) setting on the system where it executes
  - Uses Class Descriptor Table (CDT) of the system where it executes
    - Can result in missing profiles for classes not defined on execution system
- If the database templates are downlevel, new profile fields will not be unloaded (run IRRMIN00 UPDATE)
- If an identical unusable "Generic" profile and valid Generic profile coexist (e.g., ABC.**), the 0500 record will flag them, but not other 05xx records
  - Find and delete all unusable "GENERIC" profiles
- Recommendation: Create a copy of the RACF database with the IRRUT200 utility and use the copy as input to the IRRDBU00 utility
  - Prevents interference if LOCKINPUT used with an active database
  - Prevents integrity errors if NOLOCKINPUT used with an active database
  - Use the Primary database for the copy because the Backup may not be identical
    - If SETROPTS NOINACTIVE is set, logon statistics are not replicated in the Backup

IRRDBU00 - RACF Database Unload Utility

- Unload records
  - All fields are unloaded with two exceptions
    - Encrypted fields
    - RESERVED fields
  - Fields are decoded into a readable format
    - UACC is output as “READ”, “UPDATE”, “CONTROL”, “ALTER” rather than a bit mask
  - One record type per segment per repeat group, identified by a 4-byte record type
    - 0100 series - groups
    - 0200 series - users
    - 0400 series - datasets
    - 0500 series - general resources

```
0200 HANSELR  1998-06-18  SYSADMIN NO YES YES NO 030 …
0200 JOHNSON  1998-06-18  SYSADMIN NO YES YES NO 030 …
```

  - Utility Reference: RACF Security Administrator’s Guide
  - Record Layout Reference: Security Server Macros and Interfaces Guide

IRRRID00 - RACF Remove ID Utility

- Helps keep the RACF database current by …
  - Either …
    - Finding all references to any USERIDs and groups that no longer exist
    - Finding all references to select USERIDs and groups slated for deletion
  - And building commands to remove and optionally replace all references to them

- IRRRID00 looks for references to USERIDs and groups in …
  - Profile qualifiers in DATASET, FACILITY, and most general resource classes
  - Standard and conditional access lists
  - OWNER and NOTIFY fields
  - Superior groups, subordinate groups, default groups, connections
  - APPLDATA field of certain general resource profiles
  - STDATA segment of STARTED class profiles

- IRRRID00 will not build commands to remove key RACF entities (e.g., IBMUSER, SYS1, irrcerta)

IRRRID00 - RACF Remove ID Utility

```
//jobname  JOB (account),'username',CLASS=x,MSGCLASS=x
//STEP0001 EXEC PGM=IRRRID00,REGION=25M
//SYSPRINT DD  SYSOUT=*                                  <- Status report
//SYSOUT   DD  SYSOUT=*                                  <- Messages
//INDD     DD  DSN=racf.database.unload.file,DISP=SHR    <- IRRDBU00 output file
//SORTOUT  DD  SPACE=(CYL,(200,20),RLSE)                 <- Work dataset
//SYSUT1   DD  SPACE=(CYL,(200,20),RLSE)                 <- Work dataset
//OUTDD    DD  DSN=racf.remove.commands.clist.file,      <- Output commands
//             DISP=(NEW,CATLG,DELETE),
//             UNIT=DISK,
//             SPACE=(CYL,(10,5),RLSE),
//             DCB=(RECFM=VB,LRECL=259)
   - either -
//SYSIN    DD  DUMMY           <- Look for obsolete entries
   - or -
//SYSIN    DD  *               <- List of USERIDs and groups to be deleted
USERAB1
GRP002
USERNM  USERBB                 <- second ID is replacement (e.g., OWNER)
```

IRRRID00 - RACF Remove ID Utility

- Sample Output

```
CONNECT MBOM17 GROUP(PGMRGRPA) OWNER(?RCADM17)
PERMIT  'AP.DS.I.FTP.**' GENERIC  ID(SLAU97V ) DELETE
PERMIT  'BD.DS.M.ISPF.**' GENERIC  ID(BTAY51V ) DELETE
PERMIT  'SYS2.DB2.**' GENERIC  ID(DSNDBC  ) DELETE
RALTER  FACILITY BPX.SUPERUSER OWNER(?BPXGRPA )
RALTER  STARTED  BACKUPV.* STDATA( USER(?BACKUPV ))

EXIT

RDELETE SURROGAT SLAU97V.SUBMIT
DELDSD  'DB2T.DSNDBC.*.**' GENERIC
DELDSD  'TSO.BTAY51V.BOSGET.*.**' GENERIC
```

In the example above, unknown IDs were found in access lists (e.g., DSNDBC), which prompts IRRRID00 to build DELDSD and RDELETE commands for any profile with a qualifier matching the IDs

- Failsafes
  - ? - Are embedded within operands and commands fail if run without modification
  - EXIT statement - is inserted before the delete commands – preventing accidental “fall through” execution *if executed as a CLIST but **not** when executed via TSO batch SYSTSIN DD*

IRRRID00 - RACF Remove ID Utility

- Types of commands generated
  - PERMIT DELETE
    - Remove access list entries; can usually be executed unchanged
  - ALTDSD / RALTER NONOTIFY
    - Can executed as is to remove NOTIFY or optionally be modified to specify new NOTIFY(userid)
  - ALTUSER / ALTGROUP / ALTDSD / RALTER / CONNECT OWNER(?owner)
    - Replacement group or USERID must be entered for the ?owner
    - If replacement ID was specified via SYSIN, commands have new ID instead of ?owner
    - If CONNECT OWNERs were set to user executing command, can generate many replacement commands
  - RALTER STARTED STDATA(USER(?user)) | GROUP(?group))
    CONNECT ?userid GROUP(group) - USER was not found
    CONNECT userid GROUP(?group) - GROUP was not found
    - Replacement started task USERID or group must be selected, or if STARTED profile is obsolete, delete it
    - If former STARTED STDATA(USER or GROUP) has since been recreated as the other type of ID (e.g., was an ID, now a group), PERMIT DELETEs and DELUSER / DELGROUP are still created for original ID
  - DELDSD / RDELETE / RALTER DELMEM(member)
    - Member or profile assumed to be obsolete if ID matches the member or a qualifier
    - Carefully scrutinize - profiles may still be valid even though a qualifier matches an obsolete ID
  - DELUSER / DELGROUP
    - Commands related to IDs specified for deletion via SYSIN

IRRRID00 - RACF Remove ID Utility

- Recommendations
  - If RACF database is split, ensure IRRDBU00 was created using all database datasets
  - Run with SYSIN DUMMY on a regular basis (monthly) to keep database free of obsolete residual entries
  - Run with SYSIN IDs as standard procedure for deletion, especially for UNIVERSAL groups
  - Carefully review profile and member deletions - many may not be desirable
  - Archive output from both the IRRDBU00 run and the IRRRID00 utility for future reference and recovery in case something was mistakenly changed or deleted
  - Consider removing 05xx STARTED class records from input unload file when running initial execution to check for obsolete IDs

- Reference: RACF Security Administrator's Guide

IRRADU00 - RACF SMF Data Unload Utility

- Reads SMF data and produces a text or XML output dataset where the output can be used to extract information by …
  - Browsing directly as is
  - Processing by installation-developed report programs (e.g., REXX, SAS, ICETOOL)
    - DFSORT and ICETOOL - see SYS1.SAMPLIB(IRRICE) for samples
  - Loading to a database manager such as DB2 for SQL queries
  - Viewing with a web browser (XML formatted output)

- SMF data types processed:
  - 30 Common A.S. Work: Subtypes 1 (Initiation) and 5 (Termination)
  - 80 RACF processing
  - 81 RACF initialization
  - 83 RACF Audit - Subtypes 1 (Dataset SECLABEL), 2 (EIM), 3 (LDAP), 4 (R-auditx), 5 (WebSphere), 6 (TKLM)

- Executed as user exits to the SMF dump programs IFASMFDP and IFASMFDL

Reporting Tools - SMF Unload - IFASMFDP

```
//DUMPSMFU JOB (001),'HANSEL RS',CLASS=A,NOTIFY=&SYSUID,REGION=0M
//STEP0001 EXEC PGM=IFASMFDP
//SYSPRINT DD SYSOUT=*
//DUMPIN   DD DSN=SMF.MONTHLY.ARCH(0),DISP=SHR
//DUMPOUT  DD DUMMY
//SYSIN    DD *
   ABEND(NORETRY)
   USER2(IRRADU00) USER3(IRRADU86)
//ADUPRINT DD SYSOUT=*
//XMLFORM  DD DSN=RSH.SMF.XMLFORM,DISP=(NEW,CATLG,DELETE),
//            SPACE=(CYL,(100,10),RLSE),UNIT=SYSDA,
//            DCB=(LRECL=12888,RECFM=VB,BLKSIZE=0)
//XMLOUT   DD DSN=RSH.SMF.XMLOUT,DISP=(NEW,CATLG,DELETE),
//            SPACE=(CYL,(100,10),RLSE),UNIT=SYSDA,
//            DCB=(LRECL=12888,RECFM=VB,BLKSIZE=0)
//OUTDD    DD DSN=RSH.SMF.UNLOAD,DISP=(NEW,CATLG,DELETE),
//            SPACE=(CYL,(100,10),RLSE),UNIT=SYSDA,
//            DCB=(LRECL=12888,RECFM=VB,BLKSIZE=0)
```

- Generates only one form of SMF unload output - DD statements are shown in order of precedence
  - XMLFORM - One line per XML < > data tag (i.e., field)
  - XMLOUT - One line per event with all data tags
  - OUTDD - Non-XML text (like IRRDBU00)
- Unloads the entire DUMPIN dataset - no selection criteria

IRRADU00 - RACF SMF Data Unload Utility

- Unload records
  - Commands and events are translated into text format, example:
    - ACCESS - Resource access
    - ADDUSER - ADDUSER command
  - Event Codes are decoded into 8-character strings, examples:
    - INVPSWD - Invalid password
    - REVKUSER - User has been revoked

```
ACCESS  SUCCESS 16:43:00 1999-06-24 SYSA NO  NO  NO JOHNSON SYSP  YES …
ACCESS  SUCCESS 16:43:01 1999-06-24 SYSA NO  NO  NO JOHNSON SYSP  YES …
ACCESS  SUCCESS 16:43:01 1999-06-24 SYSA NO  NO  NO JOHNSON SYSP  YES …
ACCESS  SUCCESS 16:43:01 1999-06-24 SYSA NO  NO  NO JOHNSON SYSP  YES …
ACCESS  SUCCESS 16:43:02 1999-06-24 SYSA NO  NO  NO JOHNSON SYSP  YES …
```

- If SMF unload programs from a down-level system are used to process SMF records created on an up-level system with newer SMF record fields, errors message will be generated and the newer fields will not be unloaded

- Utility Reference: RACF Auditor's Guide
- Record Layout Reference: RACF Macros and Interfaces Guide

Unsupported Utilities

- Various programs provided “as is” with no formal support
- Software and detailed instructions available via:
  - Github https://github.com/IBM/IBM-Z-zOS/tree/master/zOS-RACF/Downloads
  - IBM FTP ftp://public.dhe.ibm.com/eserver/zseries/zos/racf/
- Examples:
  - RacfUnixCommands - Unix security administration REXX EXECs
  - CDT2DYN - Convert installation ICHRRCDE defined classes to Dynamic CDT profiles
  - DSNT2PRM - Converts ICHRDSNT and ICHRRNG into a RACF PARMLIB member
  - DBSYNC - Builds RACF commands to synchronize databases
  - IRRHFSU - C program to unload Unix File Security Packets (FSPs), like IRRDBU00
  - IRRXUTIL - REXX program using the IRRXUTIL R_admin callable service interface
  - LISTCDT - Lists the contents of the RACF Class Descriptor Table (CDT)
  - PWDCOPY - Copy cyphered passwords between RACF data bases
  - RACFDB2 - Migrate DB2 access controls to RACF profiles
  - RACKILL - Unconditionally deletes profiles
  - RACSEQ - Invokes R_admin (IRRSEQ00) and displays every profile field