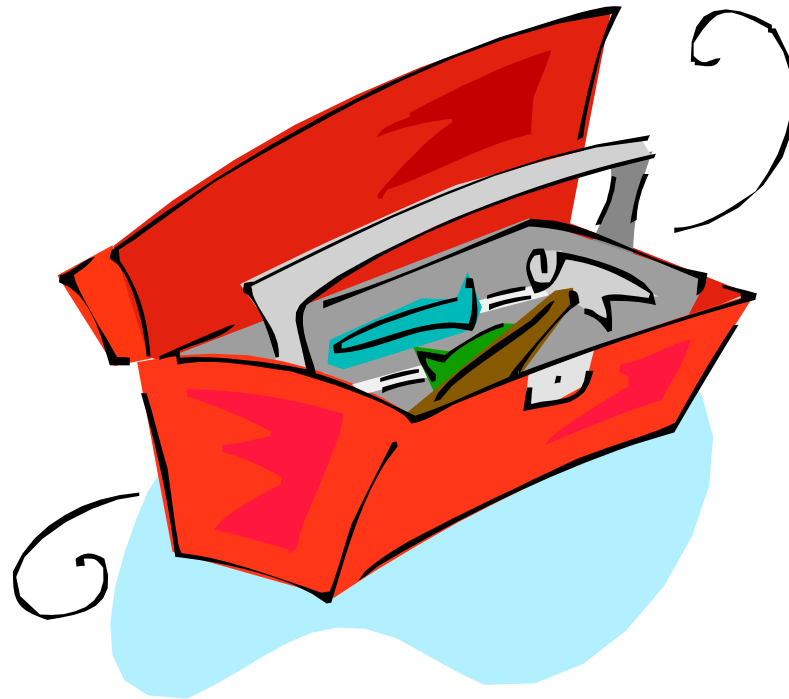


RACF UTILITIES

Chicagoland RACF Users Group - July 2009



Robert S. Hansel

Lead RACF Specialist - RSH Consulting, Inc.

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

RACF Utilities

Programs providing extended services within the RACF environment

- **Manage and maintain the RACF database**
- **Report on the RACF profiles and events**

Information provided for each utility

- **Description & function**
- **JCL & execution options**
- **Usage considerations & recommendations**
- **References**

As a general rule, if multiple systems share a RACF database and are at different z/OS release levels, only use the utilities on the highest level system

RACF, DFSORT and z/OS are Trademarks of the International Business Machines Corporation

RACF Utilities

Utilities covered by this presentation

- **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
- **Unsupported RACF utilities**

RACF Database Initialization Utility (IRRMIN00)

Initializes new databases and updates templates

PARM=

- **NEW** - Initializes new RACF database
- **UPDATE** - Installs new templates in RACF database
- **ACTIVATE** - Loads new templates into storage

PARM=NEW no longer overwrites a RACF database in use on the current system

Templates should be updated in both primary and backup, and in all parts of a split database

Requires ALTER (for NEW) or UPDATE (for UPDATE) access authority to the target RACF database

Reference: RACF Systems Programmer's Guide

RACF Internal Reorganize Aliases Utility (IRRIRA00)

Converts a RACF database to use 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 needed)**
- **Enables use of SEARCH on UID and GID**
- **Alias index requires less space than mapping profiles in both the RACF database and unload**
- **May become a requirement in future releases**

RACF Internal Reorganize Aliases Utility (IRRIRA00)

AIM conversion stage (indicator set in ICB records and RCVT):

- **0 = Pre-conversion database**
- **1 = Builds and maintains, but doesn't use, alias index**
- **2 = Uses alias index for lookup and maintains mapping profiles**
- **3 = Uses alias index for lookup and deletes mapping profiles**

Sample JCL

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

PARM=

- **No parms specified: display current stage**
- **STAGE(1 | 2 | 3): upgrade one level to level specified**

RACF Internal Reorganize Aliases Utility (IRRIRA00)

Requires UPDATE access authority to the live RACF database (primary and backup) - runs only against the live database

Considerations

- **Maximum of 129 8-byte IDs sharing an identity (e.g., UID(0))**
- **Before attempting each stage, make an IRRUT200 backup**
- **Obtains exclusive use of the database during execution**
 - **Run during non-peak periods**
 - **May run long if many mapping profiles**
 - **To expedite, consider deactivating backup and making a fresh backup with a copy of the primary when done**
- **All systems sharing the database must be at OS/390 2.10 or higher**
- **Not needed for databases created with IRRMIN00 PARM=NEW for OS/390 Release 2.10 and later**

Reference: RACF Systems Programmer's Guide

RACF Dynamic Parse Initialization Utility (IRRDPI00)

TSO command to build RACF profile segment parsing table

Should be run ...

- **At IPL**
- **After activating new templates (IRRMIN00 with ACTIVATE) if instructed by PTF documentation**

Executed by either ...

- **Started Task - typically IRRDPTAB**
- **RACF subsystem (preferred)**

```
//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

- **TSO user at READY prompt (requires program IRRDPI00 to be APF-authorized via PARMLIB member IKJTSoxx)**

RACF Dynamic Parse Initialization Utility (IRRDPI00)

Requires

- **Either ...**
 - **READ access to PROGRAM IRRDPI00**
 - ◆ **Define PROGRAM profile IRRDPI00 if backstop ** profile exists**
 - **READ access to FACILITY IRRDPI00 if PROGRAM not protected**
 - **RACF SPECIAL, if neither profile is defined**
 - **Recommendation: Define both PROGRAM and FACILITY IRRDPI00 and permit same IDs to both**
- **And ...**
 - **READ access to table source code referenced in DD SYSUT1 - usually SYS1.SAMPLIB(IRRDPSDS)**

Reference: RACF Systems Programmer's Guide

RACF Cross Reference Utility (IRRUT100)

Lists all references to specified USERIDs and/or groups in the RACF database appearing in ...

- **Standard and Conditional access lists**
- **NOTIFY and OWNER fields**
- **Group memberships for users**
- **Dataset 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)**

Use is limited to scope of authority

- **Group-level SPECIAL / AUDITOR can report only on those profiles within their scope of authority**
- **Users can generate a report on their own USERID**

RACF Cross Reference Utility (IRRUT100)

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

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)
IN ACCESS LIST OF GROUP TECHSPT1
IN ACCESS LIST OF GROUP SYS1
OWNER OF PROFILE SYS1.JSMITH.TEST
IN ACCESS LIST OF DASDMGT
OWNER OF PROFILE DASDMGT.KEEP.* (G)
USER ENTRY EXISTS
```

(G) - ENTITY NAME IS GENERIC

RACF Cross Reference Utility (IRRUT100)

Usage Notes

- **Works only with the current (active) RACF database**
- **Serializes on one profile at a time**
- **Should be run at non-peak time periods causing the least system impact**

Recommendation: Use sparingly if at all

Reference: RACF Security Administrator's Guide

RACF Database Verification Utility (IRRUT200)

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 segments of the database that are actually in use as compared to the 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

RACF Database Verification Utility (IRRUT200)

```
//jobname JOB (account),'username',CLASS=x,MSGCLASS=x
//STEP EXEC PGM=IRRUT200
//SYSRACF DD DSN=SYS1.RACF,DISP=SHR
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(10),,CONTIG),
// DCB=(LRECL=4096,RECFM=F)
//SYSUT2 DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
INDEX FORMAT
MAP ALL
END
```

SYSRACF	Input RACF database
SYSUT1	Output database copy
SYSUT2	Output messages
SYSPRINT	Output report

RACF Database Verification Utility (IRRUT200)

Control statements

- **INDEX [FORMAT]**
 - **INDEX** Performs index scan function
 - **FORMAT** Produces a formatted report of all index blocks
- **MAP [ALL]**
 - **MAP** Maps BAM/allocation
 - **ALL** Produces encoded map for each BAM block printed
- **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

RACF Database Verification Utility (IRRUT200)

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
```

...

RACF Database Verification Utility (IRRUT200)

Requires READ authority to the RACF database being verified

Considerations & Recommendations

- **If only analyzing an active database, specify a work data set on the SYSUT1 DD to make a temporary copy for analysis to avoid holding a RESERVE throughout the entire analysis phase**
- **Device, space, and DCB parameters on SYSUT1 should be exactly the same as input RACF database; use SPACE=(type(n),,CONTIG)**
- **With PARM=ACTIVATE, can copy in-use active primary directly to in-use backup without losing any updates**
- **Regularly review the percentage full to avoid running out of space**
- **If split database, each individual database must be backed up**

IRRUT200 is the only way to make a valid RACF database backup - it suspends updates during the copy process

Reference: RACF System Programmer's Guide

RACF Database Block Update Command (BLKUPD)

The BLKUPD TSO command

- **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 after other commands fail

- **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**

Reference: RACF Diagnosis Guide

RACF Database Split/Merge/Extend Utility (IRRUT400)

The 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 databases**
- **Identifies inconsistencies (e.g., duplicate profiles)**
- **Physically reorganizes the database**
 - **Reduces fragmentation by bringing all segments of a profile together and, optionally, kept within a block boundary**
 - **Compresses index**
 - **Rebuilds upper-level index blocks (corrects errors)**
 - **Rebuilds BAM blocks (corrects errors)**

RACF Database Split/Merge/Extend Utility (IRRUT400)

PARAM=

- **LOCKINPUT / NOLOCKINPUT / UNLOCKINPUT**
 - (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

- **TABLE / NOTABLE**
 - **TABLE(table-name)** - user written range table will be used; table-name is the dataset range table load module to be used for splitting the database
 - **NOTABLE** - forces selection of OUTDD1 for all profiles

- **FREESPACE(nn) / NOFREESPACE**
 - **FREESPACE** - specifies percentage of freespace to be left within the index blocks to accommodate future growth; 'nn' is 0 to 50
 - **NOFREESPACE** - is equivalent to **FREESPACE(0)**

RACF Database Split/Merge/Extend Utility (IRRUT400)

PARM= (continued)

- **ALIGN / NOALIGN**

- **ALIGN** - forces segments that occupy multiple 256-byte slots to be placed so they do not span 4096 physical blocks. Decreases I/O to process
- **NOALIGN** - causes no special alignment

- **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

RACF Database Split/Merge/Extend Utility (IRRUT400)

Copying a Database

```
//jobname JOB (account), 'username', CLASS=x, MSGCLASS=x
//COPYDB EXEC PGM=IRRUT400, PARM='NOLOCKINPUT, FREESPACE(20) '
//SYSPRINT DD SYSOUT=*
//INDD1 DD DSN=SYS1.RACF5, DISP=OLD
//OUTDD1 DD DSN=SYS2.RACF5, DISP=(, KEEP), VOL=SER=RACVL2,
// SPACE=(CYL, 10, , CONTIG), DCB=DSORG=PSU, UNIT=SYSDA
```

Splitting a Database

```
//jobname JOB (account), 'username', CLASS=x, MSGCLASS=x
//SPLITDB EXEC PGM=IRRUT400, PARM='NOLOCKINPUT, TABLE(NEW RNG) '
//SYSPRINT DD SYSOUT=*
//INDD1 DD DSN=SYS1.RACF, DISP=OLD
//OUTDD1 DD DSN=SYS2.RACF1, DISP=(, KEEP),
// UNIT=SYSDA, VOL=SER=VOL1, DCB=DSORG=PSU,
// SPACE=(CYL, 5, , CONTIG)
//OUTDD2 DD DSN=SYS2.RACF2, DISP=(, KEEP),
// UNIT=SYSDA, VOL=SER=VOL2, DCB=DSORG=PSU,
// SPACE=(CYL, 20, , CONTIG)
//OUTDD3 DD DSN=SYS2.RACF3, DISP=(, KEEP),
// UNIT=SYSDA, VOL=SER=VOL3, DCB=DSORG=PSU,
// SPACE=(CYL, 5, , CONTIG)
//STEPLIB DD DSN=INSTALL.LINKLIB, DISP=SHR
```

RACF Database Split/Merge/Extend Utility (IRRUT400)

Merging a 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, DISP=OLD
//INDD2 DD DSN=SYS1.RACF2, 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 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, DISP=OLD
//INDD2 DD DSN=SYS1.RACF2, DISP=OLD
//OUTDD1 DD DSN=SYS2.RACF, DISP=(, KEEP), UNIT=SYSDA, VOL=SER=VOL001,
//          DCB=DSORG=PSU, SPACE=(CYL,10,, CONTIG)
```

RACF Database Split/Merge/Extend Utility (IRRUT400)

Copying 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, 15, , CONTIG)
```

Unlocking the 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**

RACF Database Split/Merge/Extend Utility (IRRUT400)

Not intended to merge RACF databases from different systems

Output database cannot be the active RACF database

Recommendations

- **Run at a time when updates are not likely to be made to the RACF database**
- **Copy the database using IRRUT200 and verify integrity**
- **Reorganize using this copy**
- **RVARY the reorganized copy online**

Requires UPDATE access to the input RACF database

Reference: RACF System Programmer's Guide

RACF Data Security Monitor (DSMON - ICHDSM00)

Produces reports on the status of ...

- **System and security environment**
- **Certain RACF controls**

Requires

- **Either ...**
 - **READ access to PROGRAM ICHDSM00**
 - ◊ **Define PROGRAM profile ICHDSM00 if backstop ** profile exists**
 - **System-level AUDITOR if program is not defined**
- **And ...**
 - **READ access to FACILITY ICHDSM00.SYSCAT to list user catalogs with SYSCAT (allowed if not defined)**

Reference: RACF Auditor's Guide

RACF Data Security Monitor (DSMON - ICHDSM00)

```
//jobname JOB (account), 'username', CLASS=x, MSGCLASS=x
//DSMON EXEC PGM=ICHDSM00
//SYSPRINT DD SYSOUT=*
//SYSUT2 DD SYSOUT=*
//SYSIN DD *
FUNCTION SYSAPF
```

Functions (in report generated sequence)

SYSTEM	System and RACF identification [Recommendation - specify with every execution]
SYSPT	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

RACF Database Unload Utility (IRRDBU00)

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 programs, such as**
 - **REXX**
 - **SAS**
 - **DFSORT & ICETOOL (see SYS1.SAMPLIB(IRRICE))**
- **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 from ...

- **A copy of a RACF Database (preferred)**
- **The active backup RACF Database**
- **The active primary RACF Database**

RACF Database Unload Utility (IRRDBU00)

```
//jobname JOB (account),'username',CLASS=x,MSGCLASS=x
//UNLOAD EXEC PGM=IRRDBU00,PARM=NOLOCKINPUT
//INDD1 DD DSN=input.racf.database,DISP=SHR
//OUTDD DD DSN=output.racf.unload,UNIT=DISK,
// 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

If split database, can specify multiple input databases with INDDn statements to create combined unload file OUTDD

- Range table on the system where IRRDBU00 is executed must be identical to that used in creating the split database

RACF Database Unload Utility (IRRDBU00)

Uses Enhanced Generic Naming (EGN) setting and Class Descriptor Table (CDT) of the system where it executes

- **Can result in missing profiles for classes not defined on execution system**

Recommendation: Create a copy of the RACF database with the IRRUT200 utility and use as input to the IRRDBU00 utility

- **Prevents interference if LOCKINPUT used on active database**
- **Prevents integrity errors if NOLOCKINPUT used on active database**

Reference: RACF Security Administrator's Guide

RACF Database Unload Utility (IRRDBU00)

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 field
- 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 ONORATO 1998-06-18 SYSADMIN NO YES YES NO 030 ...
```

- Reference: Security Server Macros and Interfaces Guide

RACF Remove ID Utility (IRRRID00)

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**
- **Building commands to remove all references to them**

IRRRID00 looks for references to USERIDs and groups in ...

- **Profiles names in DATASET, FACILITY, and most general resource classes**
- **Standard and conditional access lists**
- **OWNER and NOTIFY fields**
- **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)

RACF Remove ID Utility (IRRRID00)

```
//jobname JOB (account),'username',CLASS=x,MSGCLASS=x
//STEP001 EXEC PGM=IRRRID00,REGION=25M
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//INDD DD DSN=racf.database.unload.file,DISP=SHR
//SORTOUT DD SPACE=(CYL,(200,20),RLSE)
//SYSUT1 DD SPACE=(CYL,(200,20),RLSE)
//OUTDD DD DSN=racf.remove.commands.clist.file,
// 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 * <- delete select IDs
USERAB1
GRP002
USERNM USERBB <- second ID is replacement (e.g., OWNER)
```

Input

- **INDD** - Output from the IRRDBU00 utility
- **SYSIN**- List of USERIDs and groups to be removed (optional)

RACF Remove ID Utility (IRRRID00)

Sample Output

```
CONNECT MBOM17 GROUP(REVOKE) OWNER(?MBOMB17)
PERMIT 'AP.DS.I.FTP.**' GENERIC ID(SLAU97V ) DELETE
PERMIT 'BD.DS.M.ISPF.**' GENERIC ID(BTAY51V ) DELETE
RALTER STARTED BACKUPV.* STDATA( USER(?BACKUPV ))
CONNECT ?SYSKCMD GROUP(SYST )
```

```
EXIT
```

```
RDELETE SURROGAT GENG51V.SUBMIT
DELDSD 'DB2T.DSNDBC.KTAR52V.*.**' GENERIC
DELDSD 'TSS.BTAY51V.BOSGET.*.**' GENERIC
```

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

RACF Remove ID Utility (IRRRID00)

Types of commands generated

- **PERMIT DELETE**
 - Remove access list entries; can usually be executed unchanged
- **ALTDSD / RALTER NONOTIFY**
 - Can 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, commands have new ID instead of ?owner
- **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 STARTED profile is obsolete and should be removed
 - If former STARTED STDATA(USER or GROUP) is recreated as the other type of ID, 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
- **DELUSER / DELGROUP**
 - Commands related to IDs specified for deletion

RACF Remove ID Utility (IRRRID00)

Recommendations

- **Steps**
 - **Copy the database using IRRUT200**
 - **Unload the copy using IRRDBU00**
 - **Process the unload using IRRRID00**
- **Run with no SYSIN on a regular basis to keep database free of obsolete residual entries**
- **Run with SYSIN IDs as standard procedure for deletion, especially for UNIVERSAL groups**
- **Archive output from both the IRRDBU00 run and the IRRRID00 utility for future reference / recovery**
- **Carefully review profile and member deletions - some may not be desirable**
- **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

RACF SMF Data Unload Utility (IRRADU00)

Reads SMF data and produces sequential text dataset where the output can be used to extract information by ...

- **Browsing directly as is**
- **Processing by installation-developed reporting programs**
- **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) & 5 (Termination)**
- **80 RACF processing**
- **81 RACF initialization**
- **83 RACF audit: Subtypes 1 (Dataset SECLABEL), 2 (EIM), 3 (LDAP), & 4 (R-auditx)**

Executed as user exits within the SMF dump procedure

Reference: RACF Auditor's Guide

RACF SMF Data Unload Utility (IRRADU00)

```
//SMFUNLD JOB (001),'HANSEL RS',CLASS=A,NOTIFY=&SYSUID
//STEP0001 EXEC PGM=IFASMFDP
//SYSPRINT DD SYSOUT=*
//DUMPIN DD DSN=SMF.MONTHLY.DUMP.ARCHIVE,DISP=SHR
//DUMPOUT DD DUMMY
//ADUPRINT DD SYSOUT=* <<< Messages
//XMLFORM DD DSN=RSH.SMF.XMLFORM,DISP=(NEW,CATLG,DELETE), <<< XML #1
//          SPACE=(CYL,(100,10),RLSE),UNIT=SYSDA,DCB=(LRECL=8192,RECFM=VB)
//XMLOUT DD DSN=RSH.SMF.XMLOUT,DISP=(NEW,CATLG,DELETE), <<< XML #2
//          SPACE=(CYL,(100,10),RLSE),UNIT=SYSDA,DCB=(LRECL=8192,RECFM=VB)
//OUTDD DD DSN=RSH.SMF.UNLOAD,DISP=(NEW,CATLG,DELETE), <<< Text
//          SPACE=(CYL,(100,10),RLSE),UNIT=SYSDA,DCB=(LRECL=8192,RECFM=VB)
//SYSIN DD *
          ABEND(NORETRY)
          USER2(IRRADU00) <<< SMF Unload
          USER3(IRRADU86) <<< SMF Unload
```

Only one form of SMF unload output will be generated

Output DDs are processed in sequence: (1) XMLFORM, (2) XMLOUT, (3) OUTDD

First DD found in above sequence will be the output type generated

RACF SMF Data Unload Utility (IRRADU00)

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
- **XML format - includes <> tags for each field**
 - **XMLOUT** - One line per event
 - **XMLFORM** - One line per tagged element

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

- **Reference: RACF Macros and Interfaces Guide**

Unsupported Utilities

Various programs provided “as is” with no formal support

Available via the 'Downloads' menu option on the RACF webpage via www.ibm.com/racf

Examples:

- CDT2DYN - Convert ICHRRCDE to Dynamic CDT profiles
- CUTPWHIS - Remove old password history entries
- irrhsfu - C program to unload HFS FSPs, like IRRDBU00
- PWDCOPY - Copy passwords between RACF data bases
- RACFDB2 - Migrate DB2 access control to RACF profiles
- RACKILL - Unconditionally deletes profiles

Detailed instructions included with each utility on website