

# ***RACF COMMAND TIPS***

**RACF Users Group of the Northeast (RUGONE) - September 2011**



**Robert S. Hansel**

**Lead RACF Specialist - RSH Consulting, Inc.**

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

# *TOPICS*

**USERS**

**GROUPS**

**DATASETS**

**GENERAL RESOURCES**

**PERMIT**

**LIST**

**SEARCH**

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

# USER

## **ADDUSER Defaults:**

- **OWNER** - Creator's ID
- **DFLTGRP** - Creator's Current Connect Group
- **PASSWORD** - Default Group
- Always specify when creating new ID

**Do not permit access to SPECIAL users' default groups to avoid granting access if DFLTGRP is not specified**

**Group-SPECIAL - ensure user profile OWNER is within scope-of-groups, else will not be able to administer**

**Recommendation: OWNER = DFLTGRP - ensures group authority applies to appropriate users**

**Deleting user does not remove user from access lists or purge associated profiles (e.g., JESSPOOL JES2.userid.\*\*)**

- **IRRRID00 - Build commands to purge user and associated profiles**

# USER - PASSWORD

## SETROPTS PASSWORD( RULEn( *rule* ) )

- Only one rule need apply to satisfy the syntax requirements

### INSTALLATION PASSWORD SYNTAX RULES:

```
RULE 1  LENGTH(6:8)  NNNNNNNN
RULE 2  LENGTH(6:8)  LLLLLLLL
RULE 3  LENGTH(6:8)  VVVVVVVV
RULE 4  LENGTH(6:8)  AAAAAAAA
```

### LEGEND:

```
A-ALPHA C-CONSONANT L-ALPHANUMERIC N-NUMERIC V-VOWEL W-NOVOWEL *-ANYTHING
c-MIXED CONSONANT m-MIXED NUMERIC v-MIXED VOWEL $-NATIONAL
```

- Optionally requiring 6-8 character passwords with embedded numbers (if MIXEDCASE in effect, use MIXEDNUM instead of ALPHANUM)

```
RULE1( LENGTH(6) ALPHA(1,6) ALPHANUM(2:5) )
```

```
RULE2( LENGTH(7) ALPHA(1,7) ALPHANUM(2:6) )
```

```
RULE3( LENGTH(8) ALPHA(1,8) ALPHANUM(2:7) )
```

- Password characters - letters, numbers, national characters ( \$ # @ )
  - For password rules, national characters are included in ALPHANUM, ALPHA, NOVOWEL, NATIONAL, and MIXEDNUM
  - If MIXEDCASE character option (e.g., MIXEDNUM) is specified when NOMIXEDCASE is in effect, rule will be created but no passwords will match

# ***USER - PASSWORD***

**If an ID is REVOKED due to inactivity or bad logon attempts, it is only necessary to RESUME the ID; a password reset is not required unless the password has been forgotten**

- **RESUME resets last logon date but not last connect date**

**Password composition rules do not apply when issuing new password via ALTUSER PASSWORD() unless using NOEXPIRE**

- **NOEXPIRE resets last logon date but not last connect date**

# USER - PASSWORD

Reducing the SETROPTS Password History can result in permanent non-reusable passwords

**SETR PASSWORD( HISTORY( 10 ) )**

|        |     |     |     |     |     |     |     |     |     |      |
|--------|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|
| USERID | PW1 | PW2 | PW3 | PW4 | PW5 | PW6 | PW7 | PW8 | PW9 | PW10 |
|--------|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|

**SETR PASSWORD( HISTORY( 5 ) )**

|        |     |     |     |     |     |     |     |     |     |      |
|--------|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|
| USERID | PW1 | PW2 | PW3 | PW4 | PW5 | PW6 | PW7 | PW8 | PW9 | PW10 |
|--------|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|

- All former passwords are checked when a new password is selected, but the passwords beyond the current HISTORY setting are never updated or deleted
- To clear prior history, fetch and use CUTPWHIS utility from IBM's RACF web site [www.ibm.com/servers/eserver/zseries/zos/racf/](http://www.ibm.com/servers/eserver/zseries/zos/racf/)

# ***USER - PASSWORD***

**PROTECTED prevents logon with an ID where password entry is required**

**Making an ID PROTECTED deletes its current password**

***ALTUSER userid NOPASSWORD***

**To remove PROTECTED, simply reset the password**

***ALTUSER userid PASSWORD(password)***

**Concern: What if after making an ID PROTECTED you discover the password was being used for logon by some process or other platform, and it may be difficult to quickly find out what the prior password was in order to reset it**

**Precautionary Measure: Before making an ID PROTECTED, save a copy of its current password for subsequent restoration should this be necessary using IBM's PWDCOPY utility - for a copy, visit IBM's RACF web site [www.ibm.com/servers/eserver/zseries/zos/racf/](http://www.ibm.com/servers/eserver/zseries/zos/racf/)**

# ***USER - REVOKE & RESUME***

## **ALTUSER REVOKE / RESUME**

- **Pre-z1.7 - wipes out automatic REVOKE / RESUME dates**
- **z1.7-Later - ALTUSER NOREVOKE / NORESUME removes dates**

## **SETR INACTIVE(##) - automatic revoke if no logon in ## days**

- **Inactive IDs not actually revoked until next attempted logon**
- **As of z1.7, applies to newly created IDs that have never logged on**
- **Be mindful of infrequently run batch job IDs and IDs belonging to infrequent users, especially if the IDs have SPECIAL authority (Started Tasks will start regardless)**
- **Logon statistics are only updated on the system where logon occurs; when systems are synchronized using RRSF, an ID that is typically active on only one system will appear to be inactive on the other and will get revoked on both if attempts to log onto the other system**
- **Consider creating a program to select and hard-revoke IDs as an alternative or adjunct process, or periodically resumes select IDs**

# ***USER - REVOKE & RESUME***

## **Blocking RESUMEs by Group-SPECIAL users**

- **Change user profile owner to ID not within scope-of-groups**
  - **ALTUSER SMITH01 OWNER(DEADGRP)**

## **Blocking RESUMEs by FACILITY IRR.PASSWORD.RESET users**

- **REVOKE ID's Default Group Connect**
  - **CONNECT SMITH01 GROUP(FINAN1) REVOKE**
- **Make ID PROTECTED**
  - **ALTUSER SMITH01 NOPASSWORD**
- **Set a REVOKE(date) on the ID**
  - **ALTUSER SMITH01 REVOKE(*today*)**

# USER - REVOKE & RESUME

**Blocking RESUMEs by FACILITY IRR.PWRESET.OWNER.owner or IRR.PWRESET.TREE.group users**

- **Change user profile owner to ID not within scope-of-groups**
  - ALTUSER SMITH01 OWNER(DEADGRP)
- **REVOKE ID's Default Group Connect**
  - CONNECT SMITH01 GROUP(FINAN1) REVOKE
- **Make ID PROTECTED**
  - ALTUSER SMITH01 NOPASSWORD
- **Set a REVOKE(date) on the ID**
  - ALTUSER SMITH01 REVOKE(*today*)
- **Exclude user with FACILITY profile**
  - RDEF FACILITY IRR.PWRESET.EXCLUDE.*userid* UACC(NONE)

**IRR.PWRESET.EXCLUDE.*userid***

- **SAG indicates READ permit sufficient to allow reset to proceed**
- **Actually, same permission level as needed for reset is required**

# GROUP

## **ADDGROUP defaults:**

- **OWNER** - Creator's ID
- **SUPGROUP** - Creator's Current Connect Group
- **Always specify when creating new group**

**Recommendation: OWNER = SUPGROUP - keeps group hierarchy and group authority equal and comprehensible**

**Deleting group does not remove group from access lists**

- **IRRRID00 - Build commands to purge group**

# GROUP - CONNECTS

**Interrupted execution of Connect or Remove command can leave USER and GROUP profiles with partial information**

```
LISTGRP ABC
```

```
  USER235          USE          ----- NO CONNECT ENTRY -----
```

```
LISTUSER USER349
```

```
ICH30003I GROUP GRP888 USER CONNECTION NOT INDICATED
```

```
  GROUP=GRP888      AUTH=?          CONNECT-OWNER=USER234    CONNECT-DATE=99.195
```

```
  CONNECTS=        00  UACC=NONE      LAST-CONNECT=UNKNOWN
```

```
  CONNECT ATTRIBUTES=NONE
```

```
  REVOKE DATE=NONE   RESUME DATE=NONE
```

```
LISTGRP GP1 - no member entry displayed for USERSAM2 (not UNIVERSAL group)
```

```
LISTUSER USERSAM2 - no group entry displayed for GP1
```

```
IRRDBU00 - 0205 USERSAM2 GP1 ... record, but no 0102 or 0203 records
```

**To correct, re-execute CONNECT (and then REMOVE)**

- **CONNECT** *userid* GROUP(*group*) OWNER(*owner*) [ AUTH(USE) ]
- **REMOVE** *userid* GROUP(*group*)

# GROUP - CONNECTS

## CONNECT Default - OWNER(*connector's-id*)

- **CONNECT OWNER** serves no purpose and conveys no authority
- **If allow to default to connector's ID ...**
  - **May offer hint as to who executed the CONNECT**
  - **Becomes a cleanup nightmare when connector's ID is to be deleted**
- **Consider setting OWNER to be same as group to which CONNECT is being made to avoid cleanup issue**

**CONNECT *userid* GROUP(*group*) OWNER(*group*)**

# DATASET

## ADDSD Defaults

- OWNER - Creator's ID
- UACC - UACC on Creator's Current Connect Group
- Always specify when creating new profile (recommend OWNER(*hlq*) )

```
USER=JSMITH1 NAME=JOHN SMITH OWNER=SECGRP1 CREATED=05.067
DEFAULT-GROUP=USRGRPA PASSDATE=10.130 PASS-INTERVAL= 30 PHRASEDATE=N/A
...
GROUP=USRGRPA AUTH=USE CONNECT-OWNER=SECUSR02 CONNECT-DATE=05.067
CONNECTS= 3,234 UACC=READ LAST-CONNECT=10.135/11:35:22
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE RESUME DATE=NONE
```

## Common Error #1 - forget to use apostrophes:

- ADDSD SYS1.DATA → *tso-id-prefix*.SYS1.DATA
- ADDSD 'SYS1.DATA' → SYS1.DATA
- An alternative to having to add apostrophes is to first set your TSO prefix to NOPREFIX with the PROFILE command

# ***DATASET***

## **Common Error #2 - undercutting access authority:**

- **Creating new profiles can inadvertently undermine existing authorized access**
- **Example:**
  - **Existing Profile**            **PAY.PROD.\*.\*\*            GROUPA - ALTER**
  - **New Profile**                **PAY.PROD.MASTER.\*\***
  - **Result**                        **GROUPA no longer has access**
- **Before creating new profiles:**
  - **Examine existing profile protection**
  - **Copy current UACC and access list if appropriate**

# ***DATASET***

## **Getting rid of orphaned Discrete profile**

- Dataset was deleted but discrete profile remains
- Corrective action

```
DELDSD 'profile' NOSET [ UNIT(type) VOLUME(volser) ]
```

## **Turning off RACF bit on a dataset with no Discrete profile**

- Dataset exists but discrete profile was deleted
- Corrective action

```
ADDSD 'profile' NOSET [ UNIT(type) VOLUME(volser) ]  
DELDSD 'profile' [ UNIT(type) VOLUME(volser) ]
```

# ***GENERAL RESOURCES***

## **RDEFINE Defaults**

- **OWNER** - Creator's ID
- **UACC** - Specified in CDT entry for the class, either:
  - Specific level (e.g., NONE)
  - UACC on Creator's Current Connect Group
- Always specify when creating new profile

## **Enhanced Generic Naming is always in effect**

- **SETROPTS EGN | NOEGN** only applies to DATASET profiles

# GENERAL RESOURCES

## Common Error #1 - inadvertent use of apostrophes:

- **RDEFINE *class 'profile'*** - Resource name contains 's

```
SEARCH CLASS(FACILITY)  
  'BPX.FILEATTR.APF'  
BPX.DAEMON
```

- **RDEFINE *class profile*** - Correct method

# GENERAL RESOURCES

## Common Error #2 - discretets with generic characters:

- **Example:**
  - RDEFINE \$CLSX RSH.RES.\*\* - Created as Discrete
  - SETR GENCMDS(\$CLSX)
  - Cannot administer RSH.RES.\*\* - RACF looking for Generic profile
- **To detect, look for 'generic' profiles missing (G) with SEARCH**

```
SEARCH CLASS($CLSX)
RSH.RES.**
RSH.RACF.** (G)
RSH.RES.** (G)
RSH.** (G)
```

- **To correct:**
  - **Option 1 (pre-z1.12):**
    - (1) Turn off Generics - SETR NOGENERIC(*class*) NOGENCMD(*class*)
    - (2) Delete bogus profile - RDELETE *class profile*
    - (3) Turn on Generics - SETR GENERIC(*class*) -or- GENCMD(*class*)
  - **Option 2 (pre-z1.12):** IBM's RACKILL utility (IBM's RACF web site)
  - **Option 3 (z1.12):** RDELETE *class profile* **NOGENERIC**

# GENERAL RESOURCES

## Common Error #3 - undercutting access authority:

- Creating new profiles or modifying existing ones can inadvertently undermine existing authorized access
- Example:
  - Existing Profile      DASDVOL \*\*      GROUPA - ALTER
  - New ...
    - Profile      DASDVOL TSO\*
    - Member      GDASDVOL TSODISKS ADDMEM(TSO\*)
    - Variable String      DASDVOL &T\*      RACFVARS &T ADDMEM(TSO)
  - Result      GROUPA no longer has access
- Before creating new profiles:
  - Examine existing profile protection
  - Copy current UACC and access list if appropriate

# **GENERAL RESOURCES - PROGRAMS**

**SETR WHEN(PROGRAM) - not CLASSACT(PROGRAM)**

**PROGRAM profiles must specify where the programs reside**

- **Library dataset name(s) - listed individually by full name**
- **Library dataset location (Optional) - if omitted means any volume**
  - **volser**            **-- specific DASD volume-serial number**
  - **\*\*\*\*\***           **-- IPL volume**

**Profile anomalies**

- **SETROPTS GENERIC(PROGRAM) not required and has no effect**
- **% generic character may not be used**
- **\*\* can be used, but only alone**
- **“Best fit” considers resident library as well as program name**
- **WARNING has no effect**
- **ALTER with WHEN(PROGRAM) conditional access is not reliable**

**Use READ level access, not EXECUTE, for UACC or permits**

# GENERAL RESOURCES - POSIT

**CDT entry for every class has a POSIT number - links classes for SETROPTS option processing - treated as a set**

- **IMS**
- **CICS**
- **Member / Grouping class pairs**

**Effects SETR - STATISTICS, CLASSACT, AUDIT, GENERIC, GENCMD, RACLIST, GENLIST, GLOBAL, LOGOPTIONS**

- **EX: SETR NOCLASSACT(FCICSFCT) deactivates all CICS classes**

**SETR REFRESH of one class effects all linked by POSIT**

**CLAUTH( *class* ) - Class Authorization - User Attribute**

- **Allows profile creation in all classes with matching POSIT**
- **LISTUSER only shows specified class, not all applicable ones**

# GENERAL RESOURCES - RACLIST

Performance improvement - profiles loaded into data space

Required to exploit grouping class profiles (e.g., GDASDVOL)

CDT RACLREQ=YES / RACLIST(REQUIRED) - 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 |

Required for FASTAUTH - XFACILIT - HealthChecker profiles

SETR REFRESH required before changes take effect

- Ensure REFRESH is performed on all systems sharing database
- With RACF Sysplex Communications - one REFRESH does all systems
- With RRSF Automatic Direction - one REFRESH does all RRSF nodes

SETR RACLIST vs. GLOBAL=YES RACLIST ONLY = class

- RACROUTE LIST GLOBAL - no warning of need to REFRESH

# GENERAL RESOURCES - DFTRETC

Return Code (RC) for a profile 'not found' is determined by the CDT DFTRETC / DEFAULTTRC parameter - 0 | 4 | 8

## DEFAULTTRC(8) Classes ( \* - includes grouping class)

|          |          |         |           |
|----------|----------|---------|-----------|
| APPCSERV | APPCTP   | CBIND   | CONSOLE   |
| DCEUUIDS | DIRACC   | DIRAUTH | DIRECTRY  |
| DIRSRCH  | FILE     | FSOBJ   | FSSEC     |
| IPCOBJ   | JESINPUT | JESJOBS | JESSPOOL  |
| KEYSMSTR | MQADMIN* | MQCHAN* | MQCMDS    |
| MQCONN   | MQNLIST* | MQPROC* | MQQUEUE*  |
| PROCACT  | PROCESS  | PSFMPL  | ROLE      |
| SECLABEL | SERVER   | SFSCMD  | SOMDOBJ*  |
| TEMPDSN  | TMEADMIN | WRITER  | XFACILIT* |

### Before executing SETR CLASSACT(*class*)

- JES & CONSOLE - SETR GENERIC(*class*) and define \*\* UACC(READ)
- MQADMIN - define discrete profiles *queue.NO.SUBSYS.SECURITY*

# *PERMIT - RESET*

**Clears existing access list(s) of a profile**

**RESET( ALL | STANDARD | WHEN)**

**EX: PERMIT 'SYS1.\*\*' RESET**

# PERMIT - DENY / RESTRICT ACCESS

## Undercut Group authority

- PERMIT *resource* ID( *group* ) ACCESS(UPDATE)
- PERMIT *resource* ID( *userid* ) ACCESS(READ)

## Undercut UACC

- ALTDSD/RALTER *resource* UACC(READ)
- PERMIT *resource* ID( *userid* | *group* | \* ) ACCESS(NONE)

## Undercut OPERATIONS Authority

- ALTUSER *userid* OPERATIONS
- CONNECT *userid* GROUP( #NOOPER )
- PERMIT *resource* ID( #NOOPER ) ACCESS( < ALTER )

# ADDSD / RDEFINE / PERMIT - FROM

**ADDSD & RDEFINE FROM - copies access list and all attributes (e.g., UACC, LEVEL) from existing profile to new one, except for those attributes specified in operands entered with the command**

## Command Operands

- FROM( *profile* )
- FCLASS( *class* ) - Default assumes same class
- FGENERIC - Fully-qualified generic dataset profile
- FVOLUME( *profile-volser* ) - If discrete dataset profile

## Examples

- ADDSD 'PAY.MSTFILE.\*\*' FROM('PAY.\*\*')
- RDEFINE GCICSTRN L2CMDS FROM(CEMT) FCLASS(TCICSTRN)

**PERMIT FROM - adds new IDs but retains existing ID permits**

- Include RESET to replace access list in its entirety

# GENERIC PROFILE REFRESH

## Profiles held in memory for each user

- Used for access authorization
- Not automatically updated when profiles are added or changed
- Need to refresh to activate changes

## Options:

- User logoff / logon to renew profiles
- For datasets, attempt to list a generic profile associated with the same HLQ

```
LISTDSD DATASET('hlq.anything') GENERIC
```

- SETR GENERIC( *class* ) REFRESH
  - Flushes generics for all users for the associated class
  - Avoid if re-logon is viable

**GENERIC REFRESH not necessary if class is RACLISTed (only need to REFRESH the latter)**

# AUDITING

**Unintended changes to AUDIT options - replacing as opposed to augmenting existing options**

- **Example**

**Was:** AD '*dsname*' AUDIT( FAILURES( READ ) )

**Change:** ALD '*dsname*' AUDIT( SUCCESS( UPDATE ) ) [ Add successes ]

**Result:** Now auditing successful updates but not violations

**Fix:** ALD '*dsname*' AUDIT( SUCCESS( UPDATE ) FAILURES( READ ) )

- **Applies to both Datasets and General Resource profiles**
- **Error avoidance - SETR LOGOPTIONS(FAILURES(*class*))**

**Use LEVEL(##) to flag resource profiles for report selection**

- **## = 0 to 99; 0 is the default**
- **Appears in list profile displays and database unload records 0400 and 0500**
- **Recorded in SMF type 80 access event records**

# ***LIST COMMANDS - NORACF & AUTHUSER***

**NORACF - Avoid listing entire base profile when only seeking segment info**

- **LISTUSER SMITH01 NORACF TSO**
- **LISTGRP PAYGRP1 NORACF OMVS**
- **LISTDSD DA('SYS1.\*\*') NORACF DFP**
- **RLIST STARTED TMON.\*\* NORACF STDATA**

**AUTHUSER - Minimize info from base profile when only seeking access list info**

- **LISTDSD DA('SYS1.\*\*') AUTHUSER**
- **RLIST TCICSTRN CEMT AUTHUSER**

# LISTDSD COMMAND

## Finding protecting profile:

- LISTDSD DA('dsname') [ VOL(volser) ] - Discrete profile(s)
- LISTDSD DA('dsname') GEN - Generic profile
- Also examine Global Access Table

## LISTDSD selection options

- LISTDSD DATASET('dsname-or-profile') - Examples - DA('SYS1.RACF')
- LISTDSD PREFIX(*partial.profile*) PRE(SYS1.P) -or- (SYS)
- LISTDSD ID(*hlq*) ID(SYS1)

## Finding all the cataloged datasets protected by a profile

- LISTDSD DA('profile') DSNS [ GEN ] [ NORACF ]

## Executing LISTDSD for dataset in ISPF 3.4 DSLIST panel display

- LD DA(/) GEN ALL Enter on same line as dsname; overwrite dsname

# RLIST COMMAND

List a profile in a class that is part of a member/grouping pair without confirming own access (improve performance)

- RLIST *member-class profile* NOYOURACC

List all the grouping profiles where a particular discrete resource is a member

- RLIST *member-class resource* RESGROUP

```
RLIST TCICSTRN CEMT RESGROUP
CLASS      NAME
-----
TCICSTRN   CEMT

GROUP CLASS NAME
-----
GCICSTRN

RESOURCE GROUPS
-----
CICSCMD2
CCMDSSP
```

Catch-all profile \* vs. \*\* (prefer \*\*)

- RLIST *class* \* - lists all profiles, last one being \*
- RLIST *class* \*\* - lists only \*\* profile

# SEARCH COMMAND

Used to find lists of profiles

- SEARCH or SR
- EX: SEARCH CLASS(DATASET) MASK(SYS)

General Operands:

CLASS( DATASET | *class* )

MASK( *string-1* | \* [ ,*string-2* ] ) - MASK(SM)  
| NOMASK - MASK(\*,01)  
- MASK(\$,BAT)

- or (mutually exclusive options) -

FILTER( *filter-string* ) - FILTER(PAY.\*\*.LIB.\*)  
- FILTER(%%BAT\*)

MASK defaults:

- DATASET class - *your-userid*
- General Resource classes - NOMASK

# SEARCH COMMAND

## Users:

- **SR CLASS(USER) AGE(##)**
  - Number of days since last logon or greater
  - Combine with CLIST('ALU ' ' REVOKE') to hard-revoke inactive IDs
  - NJE and RJE link IDs and DB2 DRDA IDs will be listed even if active
- **SR CLASS(USER) UID(##)**
  - Lists all users with a specified uid
  - Requires Application Identity Mapping (AIM) database structure

## Groups:

- **SR CLASS(GROUP) USER(*userid*)**
  - Lists all groups user owns or has CONNECT, JOIN, or Group-SPECIAL Authority (will not work with a USERID that is revoked)
- **SR CLASS(GROUP) GID(##)**
  - Lists all groups with a specified uid
  - Requires Application Identity Mapping (AIM) database structure

# SEARCH COMMAND

## Dataset & General Resources:

- Profile type selection:
  - ALL | GENERIC | NOGENERIC
  - Dataset only: MODEL | TAPE | VSAM | NONVSAM
  - Examples:
    - SR NOMASK NOGENERIC      - List all DATASET discrete profiles
    - SR NOMASK MODEL            - List all DATASET model profiles
  
- USER( userid )
  - Lists all profiles user owns or has READ or higher access except where a user's group has access of NONE
  - Will not work with a USERID that is revoked
  
- WARNING
  - Profiles in WARN mode
  
- LEVEL( ## )
  - Profiles with LEVEL set to value ##

# SEARCH COMMAND - CLIST

## Automatically build RACF administration commands:

```
SR CLASS(USER) CLIST('LU ' ' TSO') NOLIST
```

### CLIST output dataset:

```
00000010CONTROL ASIS
00000020LU $OEDFLU TSO
00000030LU BWO TSO
00000040LU BWO1 TSO
00000050LU CICSUSER TSO
00000060LU CLRLOG TSO
00000070LU DSN1WLM1 TSO
00000080LU FTPD TSO
```

- Creates output dataset: *tso-id-prefix*.EXEC.RACF.CLIST
- To execute, enter either:

```
EXEC EXEC.RACF.CLIST
```

- prefixed with ID

```
EXEC 'tso-id-prefix.EXEC.RACF.CLIST'
```

# SEARCH COMMAND - CLIST

- **CLIST( 'string-1' [ , 'string-2' ] ) LIST | NOLIST**
  - **CLIST('LU ')**  
    ➔ **LU *userid***
  - **CLIST('RALTER FACILITY ' ' OWNER(SYSPROG)')**  
    ➔ **RALTER FACILITY *profile* OWNER(SYSPROG)**
  - **CLIST('LU ' ' NORACF TSO')**  
    ➔ **LU *userid* NORACF TSO**
  - **CLIST('ALD ' ' UACC(NONE)')** - automatically inserts profile quotes  
    ➔ **ALD '*ds-profile*' UACC(NONE)**
- **Inserted profiles will abut the string(s) - include spaces within quotes as necessary to separate text from the profile**

# ENTERING COMMANDS VIA CONSOLE

Requires RACF subsystem

Execution sequence at console

**LOGON *userid*** - will be prompted for password  
**#*racf-command*** - include prefix character (e.g., #)  
**LOGOFF**

Logged-on ID is shown at bottom of console display

Prefix character determined by PARMLIB( IEFSSNxx ) INITPARM operand on RACF subsystem definition

```
SUBSYS SUBNAME(RACF)  
INITRTN(IRRSSI00) INITPARM( '#' )
```

If no prefix is specified, the prefix defaults to the RACF subsystem name plus a blank - e.g., 'RACF '

```
RACF LISTUSER IBMUSER
```

# ENTERING COMMANDS VIA CONSOLE

Can optionally register prefix with Command Prefix Facility (CPF) to ensure it is reserved

- INITPARM('prefix,scope')            scope = M or X
  - M - Reserve within system image
  - X - Reserve within Sysplex (only one subsystem in Sysplex can use)
- If registered, can be listed with operator command:  
DISPLAY OPDATA,PREFIX

Commands are protected by OPERCMDS profiles

- *racf-subsystem-name.racf-command*
  - READ            Execute SETROPTS LIST and all other commands
  - UPDATE        Execute SETROPTS with parameters other than LIST
  - Normal RACF authority (e.g., SPECIAL) is also required

Periodically test to ensure other subsystems do not interfere with command execution

# EXCEL - CONCATENATE

Generate RACF commands from user spreadsheet request

| Cols:  | A      | B    | C    | D        | E         |
|--------|--------|------|------|----------|-----------|
| Row#1: | Userid | Name | Dept | Password | Employee# |

Col: F

```
=concatenate("AU ",a1," NAME(",b1,"") PASSWORD(",d1,  
") DFLTGRP(G@",c1,") OWNER(G@",c1,  
") OMVS( UID(222",e1,"0)")
```

Copy the column with Concatenate results to a text file, upload, and execute