RSH CONSULTING

RACF Command Tips

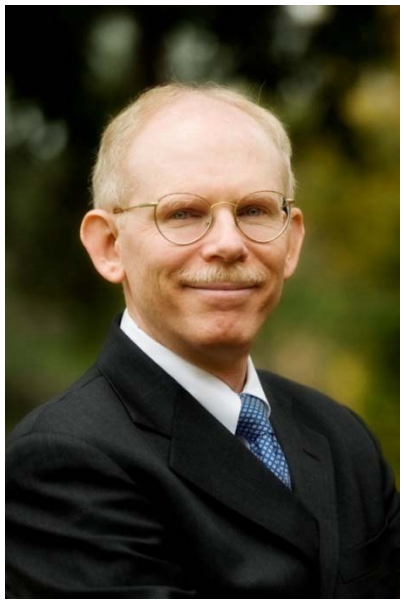
SHARE - March 2015

Session 18875

RSH Consulting - Robert S. Hansel

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

Topics

- User Commands
- Group Commands
- Dataset Command
- General Resource Commands
- PERMIT Command
- Generic Profile Refresh
- List Commands
- SEARCH Command
- Console Command Entry
- Building Commands with Microsoft Excel

User Commands

- ADDUSER Defaults:
  - OWNER - Creator's ID
  - DFLTGRP - Creator's Current Connect Group
  - PASSWORD - Pre-z/OS 2.2: Default Group
    z/OS 2.2: NOPASSWORD
  - Always specify when creating new ID
- Do no 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 Commands - RESUME & Password Reset

- 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(*newpassword*) unless using NOEXPIRE
  - NOEXPIRE resets last logon date but not last connect date

- To force user to change their password ...
  - Pre-APAR OA43999-z/OS 2.2 ALTUSER *userid* PASSWORD(*newpassword*)
  - Post-APAR OA43999-z/OS 2.2 ALTUSER *userid* EXPIRED

User Commands - Password History

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

  SETROPTS PASSWORD(HISTORY(10))

USERID	PW1	PW2	PW3	PW4	PW5	PW6	PW7	PW8	PW9	PW10

  SETROPTS 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 excess prior history entries, ...
  - Pre-APAR OA43999-z/OS 2.2 CUTPWHIS utility available via IBM's RACF web site
  - Post-APAR OA43999-z/OS 2.2 ALTUSER *userid* PWCLEAN
    - RWCLEAN also clears the entire history of a PROTECTED ID

User Commands - PROTECTED

- 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 assign the ID a 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-03.ibm.com/systems/z/os/zos/features/racf/

User Commands - INACTIVE

- SETROPTS INACTIVE(##) - automatic revoke if no logon in ## days
  - Inactive IDs not actually revoked until next attempted logon
  - 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)
    - PROTECTED IDs are exempt from inactive revoke
  - 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 resume select IDs

User Commands - Blocking Reactivation

- Situation: A REVOKED USERID is within the scope of a non-SPECIAL administrator , and you do not want the latter to reactivate the ID
- Block users with Group-SPECIAL access
  - Change user profile owner to a user or group not within scope (e.g. DEADGRP)
- Block users permitted FACILITY IRR.PASSWORD.RESET
  - REVOKE ID's Default Group Connect (e.g., CONNECT REVOKE) to inhibit logon
  - Set REVOKE(*today's-date*) on the ID to cause immediate re-REVOKE on logon
  - Make ID PROTECTED
- Block users permitted FACILITY IRR.PWRESET.OWNER.*owner* or IRR.PWRESET.TREE.*group*
  - REVOKE ID's Default Group Connect (e.g., CONNECT REVOKE) to inhibit logon
  - Set REVOKE(*today's-date*) on the ID to cause immediate re-REVOKE on logon
  - Change user profile owner to a user or group not within scope
  - Make ID PROTECTED
  - Exclude user with FACILITY profile IRR.PWRESET.EXCLUDE.*userid*

Group Commands

- 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 synchronized

- Deleting group does not remove group from access lists
  - IRRRID00 - Build commands to purge group

Group Commands - Broken CONNECT

- Interrupted execution of a CONNECT or REMOVE command or deletion of a UNIVERSAL group can leave USER and GROUP profiles with incomplete connect 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)
  - If broken connection is user's Default Group, temporarily change the Default Group
  - May need to recreate formerly existing user or group to perform the CONNECT

Group Commands - CONNECT

- 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*)

- CONNECT Default - UACC(NONE)
  - CONNECT UACC is used as the Default UACC when creating profiles for DATASET or General Resource classes whose CDT entry has DFTUACC / DEFAULTUACC of ACEE

CBIND	CPSMOBJ*	CPSMXMP	DASDVOL*	DSNR
INFOMAN*	NETCMDS	NETSPAN	PTKTVAL	RODMMGR
SERVER	SOMDOBJS*	TAPEVOL	TERMINAL*	(* - also Grouping)

  - Also applies to ADDUSER for initial DFLTGRP connection
  - Always allow to default to NONE

Dataset Commands

- 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
```

- Enhanced Generic Naming (EGN)
  - SETROPTS EGN | NOEGN
  - Enables use of ** masking character and changes behavior of * character
  - Only applies to DATASET profiles

Dataset Commands - Common Errors

- Common Error - forget to use apostrophes:
  - ADDSD SYS1.DATA ➔ *tso-id-prefix*.SYS1.DATA
  - ADDSD 'SYS1.DATA' ➔ SYS1.DATA
  - An alterative to having to add apostrophes is to first set your TSO prefix to NOPREFIX with the PROFILE command

Dataset Commands - Common Errors

- Common Error - undercutting access authority
  - Creating new profiles can inadvertently undermine existing authorized access or grant unintended access
  - Example #1:
    - Existing Profile SYS1.SFTWR.*.** UACC(READ) GROUPA(ALTER)
    - New Profile SYS1.SFTWR.CONFIG.** UACC(NONE)
    - Result 1) GROUPA no longer has ALTER access
      2) Everyone loses READ access
  - Example #2:
    - Existing Profile APP.*.LIB.** UACC(NONE)
    - New Profile APP.UTIL.** UACC(READ)
    - Result Everyone has READ access to APP.UTIL.LIB.** datasets
  - Before creating new profiles:
    - Examine existing profile protection
    - Copy current UACC and access list if appropriate

Dataset Commands - Discretes

- 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 missing 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 Resource Commands

- RDEFINE Defaults
  - OWNER - Creator's ID
  - UACC - Specified by DEFAULTUACC in CDT entry for the class, either:
    - Specific level - IBM classes are NONE, except APPCSI is READ
    - 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 Resource Commands - Common Errors

- Common Error - inadvertent use of quotes in creating a profile:
  - RDEFINE *class* '*profile*' - Resource name contains quote (') characters

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

  - RDEFINE *class profile* - Correct method

- Common Error - accidentally create profile with embedded quote (') or parentheses '(' characters
  - RDEFINE *class* (PROFILE AUDIT(ALL DATA('TEST PROF') - forgot ')' after PROFILE

```
SEARCH CLASS(FACILITY)
  AUDIT(ALL
  DATA('TEST
  PROF'
  PROFILE
```

  - RDELETE *class* DATA('TEST - Fails because no matching ' or)
  - RDELETE *class* (DATA('TEST ') - Succeeds (ignore error about nonexistent ')
  - RDELETE *class* (AUDIT(ALL X) - Succeeds (ignore error about nonexistent X)

General Resource Commands - Common Errors

- Common Error - discretes with generic characters:
  - Forget to activate GENCMD or GENERIC
  - Create discrete profiles with generic characters - % * **
  - Subsequently activate GENCMD or GENERIC
  - Profiles are meaningless and cannot be administered
  - Profiles appear in SEARCH results with the indicator (UNUSABLE)

```
SR CLASS(OPERCMDS)

MVS.FORCE.* (UNUSABLE)
MVS.LOG
MVS.*.BAT (G)
MVS.** (G)
```

  - Can delete such profiles using RDELETE command with NOGENERIC keyword

    RDEL OPERCMDS MVS.FORCE.* NOGENERIC

General Resource Commands - Common Errors

- Common Error - undercutting access authority:
  - Creating new profiles can inadvertently undermine existing authorized access
  - Example:
    - Existing Profile DASDVOL ** GROUPA - ALTER
    - New Profile DASDVOL TSO*
      - or - GDASDVOL TSODISKS ADDMEM(TSO*)
      - or - 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
  - Be mindful that creating profiles with embedded generics, especially in early qualifiers, are at greater risk of being mistakenly undercut
    - JESSPOOL *node*.*.*jobname*.**
    - JESSPOOL *node*.*userid*.**

General Resource Commands - ADDMEM

- Member sequence sometimes matters
  - RACFVARS - first match is used; issue if one entry is the substring of another
  - NODES - first entry is used (should only be one entry)
- RDEFINE adds members in the order they are listed in the command (FIFO)
  - RDEFINE *class profile* ADDMEM(A B) results in a member list of: A B
- RALTER adds members in the reverse order they are listed in the command and adds them to the front of the list (LIFO)
  - RALTER *class profile* ADDMEM(X Y) results in a member list of : Y X A B
- RLIST lists the members is alphanumeric sequence
- IRRDBU00 Database Unload lists the members in actual sequence
- Administrative tip - when adding new members to a RACFVARS profile, delete (RDELETE) and then recreate (RDEFINE) the profile with the desired member sequence as opposed to trying to modify (RALTER) the list

General Resource Commands - POSIT

- CDT entry for every class has a POSIT number which, if shared, links classes together as a set
  - IMS 4 AIMS GIMS TIMS
  - CICS 5 ACICSPCT BCICSPCT CCICSCMD DCICSDCT ...
  - Member / Grouping class pairs (e.g., 0 - DASDVOL / GDASDVOL)

- SETROPTS class-related options STATISTICS, CLASSACT, AUDIT, GENERIC, GENCMD, RACLIST, GENLIST, GLOBAL, and LOGOPTIONS affect the entire set
  - EX: SETROPTS NOCLASSACT(FCICSFCT) deactivates all CICS classes

- SETROPTS REFRESH of one class effects all classes linked by POSIT

- Class Authorization - User Attribute - ADDUSER/ALTUSER CLAUTH(*class*)
  - Allows profile creation in all classes with matching POSIT
  - LISTUSER only shows specified class added to the ID, not all applicable ones

General Resource Commands - RACLIST

- Performance improvement - profiles loaded into data space
- Required to exploit grouping class profiles (e.g., GDASDVOL)
- Required for FASTAUTH - XFACILIT - HealthChecker profiles
- CDT RACLREQ=YES / RACLIST(REQUIRED) - RACLIST is required
  - If class is not RACLISTed, profiles are ignored

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

- SETROPTS 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
- SETROPTS RACLIST vs. GLOBAL=YES RACLIST ONLY = class
  - RACROUTE LIST GLOBAL - no warning of need to REFRESH

General Resource Commands - DEFAULTRC(8)

- Return Code (RC) for a profile 'not found' is determined by the CDT DFTRETC / DEFAULTRC parameter - setting are 0 | 4 | 8
- DEFAULTRC(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*
MXADMN*	MXNLIST*	MXPROC	MXQUEUE*
MXTOPIC*	PROCACT	PROCESS	PSFMPL
RACFHC	ROLE	SECLABEL	SFSCMD
SERVER	SOMDOBJS*	TEMPDSN	TMEADMIN
WRITER	XCSFKEY	XFACILIT*	

- Before executing SETROPTS CLASSACT(*class*)
  - JES & CONSOLE - SETROPTS GENERIC(*class*) and define ** UACC(READ)
  - MQADMIN - define discrete profiles *queue*.NO.SUBSYS.SECURITY

PERMIT Command - Deny / Restrict Access

- Undercut group access authority where user is a member of the group
  - PERMIT *profile* [CLASS(*class*)] ID(*group*) ACCESS(UPDATE)
  - PERMIT *profile* [CLASS(*class*)] ID(*userid*) ACCESS(READ) - takes precedence

- Undercut UACC
  - Profile has UACC(READ)
  - PERMIT *profile* [CLASS(*class*)] ID(*userid* | *group* | *) ACCESS(NONE)

- Block or limit OPERATIONS Authority
  - USERID has OPERATIONS
  - CONNECT *userid* GROUP(#NOOPER)
  - PERMIT *profile* [CLASS(*class*)] ID(#NOOPER) ACCESS(< ALTER)
  - Best Practice - create an OPERATIONS blocking group and permit access to ...
    - DATASET profiles for Catalogs - permit same as ID(*) or UACC
    - DASDVOL profiles - permit NONE

PERMIT Command - RESET

- Clears existing access list(s) of a profile
- RESET(ALL | STANDARD | WHEN)
- EX: PERMIT 'SYS1.**' RESET

ADDSD / RDEFINE / PERMIT - FROM Operand

- ADDSD and RDEFINE FROM - copies access list and attributes (e.g., UACC, LEVEL) from existing profile to new one, except for those attributes specified in operands entered with the command
- PERMIT FROM - copies access list from another profile
  - Adds new IDs but retains any existing ID permits
  - Include RESET to replace access list in its entirety
- 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 AUDIT(ALL) FROM(CEMT) FCLASS(TCICSTRN)
  - PERMIT BPX.FILEATTR.APF CLASS(FACILITY) RESET FROM(BPX.FILEATTR.PROGCTL)

Generic Profile Refresh

- Profiles held in memory for each individual user (non-RACLISTed classes)
  - Used for access authorization
  - Not automatically updated when profiles in the database are added or changed
  - Need to refresh user-held profiles to activate changes

- Profile refresh options:
  - User logoff / logon to renew all profiles
  - TSO user can execute a LISTDSD GENERIC command to refresh a DATASET profile

    LISTDSD DA('*profile.or.dataset.name*') GENERIC ALL
  - SETROPTS GENERIC(*class*) REFRESH
    - Flushes all generic profiles for all users for the associated class, requiring everyone to retrieve the profiles again
    - Avoid if re-logon is a viable alternative, especially for the DATASET class or classes with many profiles
    - GENERIC REFRESH is meaningless if class is RACLISTed (only need RACLIST REFRESH)

AUDIT, LEVEL, TSO PROFILE

- Common Error - Degrade existing AUDIT options when the intent was to expand them
  - Example

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

    Change: ALD '*dsname*' AUDIT(SUCCESS(UPDATE)) - Replaces prior setting

    Result: Now auditing successful updates but not violations

    Fix: ALD '*dsname*' AUDIT(SUCCESS(UPDATE) FAILURES(READ))
  - Applies to both Dataset and General Resource profiles
  - Error avoidance - SETROPTS 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

- TSO - Avoid prompting to fix command errors
  - Command: PROFILE NOPROMPT

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 - Avoid listing entire profile when only seeking access list info; alternative to specifying ALL
  - LISTDSD DA('SYS1.**') AUTHUSER
  - RLIST TCICSTRN CEMT AUTHUSER

LIST Commands - LISTDSD

- Finding protecting profile:
  - LISTDSD DA('*dsname*') [VOL(*volser*)] - Discrete & Fully-Qualified Generic profile(s)
  - LISTDSD DA('*dsname*') GEN - Generic profile

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

- Finding all the cataloged datasets protected by a profile
  - LISTDSD DA('*profile*') DSNS [NORACF]

- Executing LISTDSD for dataset in ISPF 3.4 DSLIST panel display
  - LD DA(/) ALL [GEN] Enter on same line as dsname; overtype dsname

LIST Commands - RLIST

- List a profile in a member class that is part of a member/grouping pair without confirming own access (improve performance)
  - RLIST *member-class profile* NOYOURACC - Can abbreviate as NOY
- List all the grouping profiles where a particular discrete resource is a member
  - RLIST *member-class resource* RESGROUP -or- ALL

```
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
      - Warning: 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 - Stage 2 or 3

- 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 GID
    - Requires Application Identity Mapping (AIM) database structure - Stage 2 or 3

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 those where any of the user's groups 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 adds quotes for DATASET profiles
    - ➔ 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 characters (e.g., #)

  LOGOFF
- Logged-on ID is shown at bottom of console display
- RACF commands must be prefixed with the designated prefix character(s)
  - Prefix character determined by PARMLIB(IEFSSNxx) INITPARM operand on RACF subsystem definition (IBM manuals use # in their examples)

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

  - If no prefix is specified, it 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

Building Commands in Microsoft Excel

- Generate RACF commands from spreadsheets using Excel formulas such as =concatenate("*text*",*cell*,"*text*",*cell*,...)

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