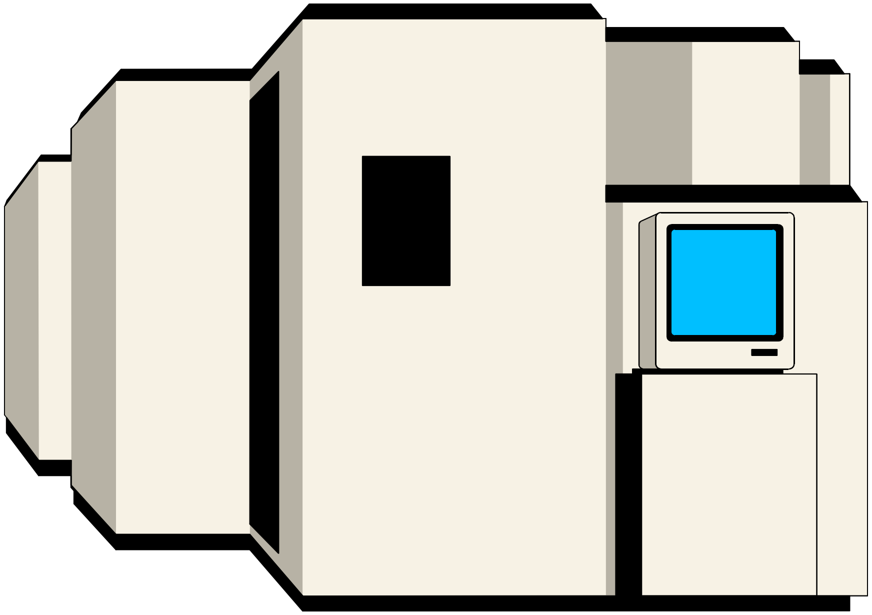
RACF GENERAL RESOURCE FUNDAMENTALS

KOIRUG - May 2007



Robert S. Hansel

RACF Specialist - RSH Consulting, Inc.

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

TOPICS

Introduction to General Resources

RACF Router Table

Class Descriptor Table

CDT Class

Profiles

GENERAL RESOURCES

A General Resource is anything other than a dataset

- Terminals
- CICS Transactions
- DASD Volumes
- Application APPLIDs
- TSO Logon Attributes
- General Purpose Facility
- Programs
- JES Spool
- NJE Nodes
- DB2 System Connections
- MVS and JES Commands
- 3rd Party or Locally Defined

General Resources are identified by their logical names within a specific class

The construct of the resource name is determined by the resource manager

GENERAL RESOURCES

RESOURCE-TYPE	CLASS / GROUPING-CLASS	RESOURCE-NAME
Program	PROGRAM / PMBR	AMASPZAP
TSO Authority	TSOAUTH	OPER
DASD Volumes	DASDVOL / GDASDVOL	SYS001
CICS APPLID	APPL	CICSPRD1
DB2 TSO Connect	DSNR	DB2P.BATCH
Storage Admin	FACILITY	STGADMIN.ADR.DEFRAG
JES2 RJE Reader	JESINPUT	RMT0002.RD1
SDSF Command	SDSF	ISFCMD.DSP.OUTPUT.JES2
MVS Command	OPERCMDS	MVS.HALT.NET
CICS Transaction	TCICSTRN / GCICSTRN	CEMT

GENERAL RESOURCE PROTECTION

When a user attempts to access a resource, the Resource Manager (e.g., CICS) calls RACF for an authorization check

The Resource Manager sends RACF

- **Identity of the user**
- **Class and name of the resource**
- **Access the user is attempting (e.g., Update)**

The Resource Manager uses a RACF Macro to make the call

- **RACHECK or FRACHECK**
- **RACROUTE REQUEST=AUTH or FASTAUTH**
- **RACLIST / RACROUTE REQUEST=LIST build in-storage profiles for FRACHECK / FASTAUTH processing**

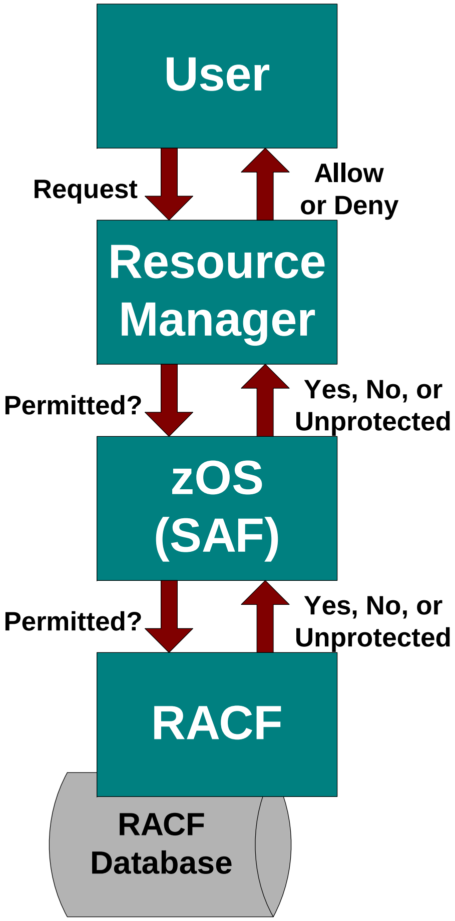
GENERAL RESOURCE PROTECTION

RACF determines whether the user is authorized to access the resource at the requested level of access

RACF makes this determination based on General Resource Profiles defined in its database

RACF sends a Return Code (RC) back to the calling Resource Manager indicating the results of the authorization check

0	Authorized
4	Not-Protected
8	Not-Authorized



GENERAL RESOURCE PROTECTION

General Resource classes must first be defined to RACF

- **RACF Class Descriptor Table (CDT) or CDT Class Profiles (z1.6)**
- **RACF Router Table (RRT) [not required if using CDT profiles]**
- **There is a set of IBM entries and installation-defined entries**

Classes must be activated to enable protection

- **SETROPTS CLASSACT(*class*)**
- **SETROPTS WHEN(PROGRAM)**

Certain classes must be RACLISTed to enable protection as determined by CDT parameter RACLREQ / RACLIST(REQUIRED)

RACFVARS APPCSERV APPCTP CSFKEYS FIELD

CSFSERV DEVICES NODES OPERCMDS

PROPCNTL PSFMPL PTKTDATA SECLABEL

STARTED SYSMVIEW UNIXPRIV VTAMAPPL

GENERAL RESOURCE PROFILES

General Resource Profile contains

- **Identifying information**
- **Control options**
- **Auditing specifications**
- **Access permissions**

General Resource Profile names incorporate the class and name of the resource

• **DASD Volume**	**TSO003**
• **Class & Profile**	**DASDVOL TSO***

Length, character composition, and letter case of profile names are determined by CDT parameters

GENERAL RESOURCE PROFILES

Class types

• Member	TCICSTRN
• Grouping	GCICSTRN

Profile types

• Discrete	TCICSTRN CEMT
• Generic (RACFVARS)	TCICSTRN C*
• Grouped (members)	GCICSTRN CICSCMD1 ADDMEM(CEDF)

RACFVARS

- Variable text strings used in Generic profiles
- Prefixed with an ampersand '&' (e.g., &RACLNDE)
- Ex: JES2.LOCAL.&PAYPRTR, where &PAYPRTR = PRT05 & PRT44

GENERAL RESOURCE PROFILES

RACF uses the most specific profile or grouping class member (i.e., closest match) for determining access authorization

```
ISFCMD.DSP.STATUS.JES2
ISFCMD.DSP.&SDSFOPR.JES2
ISFCMD.DSP.*
ISFCMD.*
**
```

Profiles with masking characters further from the front are considered to be more specific

PROFILE NOT FOUND

The RC for a profile ‘not found’ is determined by CDT parameter DFTRETC / DEFAULTRC

- **DFTRETC parameter 0 | 4 | 8 (Auth | Unknown | Not auth)**
- **DFTRETC=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	SFSCMD	SERVER	SOMDOBJS*
TEMPDSN	TMEADMIN	WRITER	XFACILIT*

Calling process decides how to react to Return Code

GENERAL RESOURCE PERMISSIONS

Access permissions are specified in three ways

- **Standard Access List**
- **Conditional Access List**
- **Universal Access (UACC)**

Access can be permitted to

- **USERID**
- **Group**
- **ID(*)** **- Grants access to all RACF Defined users**

Each permission specifies an Access Level

GENERAL RESOURCE PERMISSIONS

Access levels

- **ALTER**
- **CONTROL**
- **UPDATE**
- **READ** **(often equates to USE)**
- **EXECUTE**
- **NONE**

The meanings of the levels varies depending on the class of resources being protected

OPERATIONS authority may grant ALTER access to resources in a specific class depending on CDT parameter OPER / OPERATIONS

GENERAL RESOURCE PERMISSIONS

Conditional Access - grants access when a condition is met

Types of conditional access

- **WHEN(JESINPUT(device))**
- **WHEN(TERMINAL(terminal-id))**
- **WHEN(APPCPORT(partner-lu-name))**
- **WHEN(CONSOLE(console-id))**
- **WHEN(SYSID(smf-id))- PROGRAM profiles only**

EX: WHEN(CONSOLE(SDSF))

GENERAL RESOURCE PERMISSIONS

Conditions must be explicit (e.g., JESINPUT NJENODE1); generics cannot be used

The 'WHEN' General Resource class (e.g., JESINPUT) must be active and a profile matching the resource (e.g., NJENODE1) must be defined to RACF (except SYSID)

If more than one condition is specified in a PERMIT command, each is stored and treated as a separate permission and are not used in combination

```
WHEN( PROGRAM(PAYUPDT)  TERMINAL(35) )
```

Conditional access permissions can not be used to DENY access

GENERAL RESOURCES

General
Resource
Profile

Class
General Resource Name/Mask

Profile Type
Resource Members (Grouping)
Profile Owner
WARN mode flage
Auditing
UACC
Installation-Data
Application-Data

Standard Access List
- User(s) - Access
- Group(s) - Access
- * - Access

Conditional Access List - WHEN
- User(s) - Access - Condition
- Group(s) - Access - Condition
- * - Access - Condition

Segment
- STDATA (STARTED profiles)
- SESSION (APPCLU profiles)

GENERAL RESOURCE PROFILE

```
RLIST GCICSTRN TSPT$CMD ALL
CLASS      NAME
-----      ----
GCICSTRN   TSPT$CMD

MEMBER CLASS NAME
------ ----- ----
TCICSTRN

RESOURCES IN GROUP
--------- -- -----
CEMT
CEDA
CEDF
CSM*

LEVEL  OWNER      UNIVERSAL ACCESS   YOUR ACCESS  WARNING
-----  --------   ----------------   ---- ------  -------
 00    CICSSPT         NONE               NONE     NO

INSTALLATION DATA
---------------------------------------------------
CICS TECH SPT SYSTEM COMMANDS

APPLICATION DATA
----------------
NONE

SECLEVEL
--------
NO SECLEVEL

CATEGORIES
----------
NO CATEGORIES
```

GENERAL RESOURCE PROFILE

```
SECLABEL
--------
NO SECLABEL

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

CREATION DATE  LAST REFERENCE DATE  LAST CHANGE DATE
(DAY) (YEAR)        (DAY) (YEAR)      (DAY) (YEAR)
-------------  -------------------  ----------------
 270    92           282    92         282    92

ALTER COUNT  CONTROL COUNT  UPDATE COUNT  READ COUNT
-----------  -------------  ------------  ----------
  000000        000000         000000        000000

USER      ACCESS    ACCESS COUNT
----      ------    ------ -----
RJONES2   ALTER
CICSSPT   UPDATE
DASDMGT   READ
JWILLS2   NONE

   ID     ACCESS ACCESS COUNT   CLASS         ENTITY NAME
-------- ------- ------------   --------  -----------------------
NO ENTRIES IN CONDITIONAL ACCESS LIST
```

RACF ROUTER TABLE (RRT)

Controls the action taken by the RACF router module ICHRFR00 when envoked by the RACROUTE macro

RRT Modules

ICHRFR0X	IBM supplied router entries (pre-z1.6)
ICHRFR01	Installation-defined router entries

ICHRFR0X contains (pre-z1/6)

- **Entry corresponding with each entry in the Class Descriptor Table**
- **Entry for DATASET, USER, GROUP, and CONNECT classes**
- **Entries specifying REQSTOR and SUBSYS for special cases**

```
CLASS=PROGRAM,REQSTOR=CKPGMDSN,SUBSYS=CONTENTS
```

RACF ROUTER TABLE (RRT)

ICHRFR01 is loaded during IPL

ICHRFRTB assembly macro is used to define ICHRFR01 entries

```
[label]   ICHRFRTB      [ CLASS=class-name ]
                        [ ,REQSTOR=requestor-name ]
                        [ ,SUBSYS=subsystem-name ]
                        [ ,ACTION=NONE | RACF ]
                                  - or -
                        [ TYPE=END ]
```

Place frequently referenced classes first

As of z1.6, RRT is only needed to specify ACTION=NONE

RRT - ICHRFRTB Macro Parameters

CLASS=	**Class**	**Name of the resource class; same as corresponding CDT entry**
REQSTOR=	**Requestor**	**Optional 1-8 character name**
SUBSYS=	**Subsystem**	**Optional 1-8 character name**
ACTION=	**NONE**	**Take no action**
	RACF	**Call RACF**
TYPE=	**END**	**Last entry of the table**

RRT - ICHRFRTB & RACROUTE Macros

REQSTOR and SUBSYS

- Optional parameters
- Combined with CLASS to form 24-character string
- Used to trace and segregate specific RACF calls
- Installation names should use prefix #, @, $
- Included in RACROUTE macro call
- Pre z1.6, required for DB2 DSNR Class entries

RACROUTE macro calls

- CLASS=class[,REQSTOR=requestor,SUBSYS=subsystem]
- DECOUPL= NO | YES

NO	Require corresponding REQSTOR and SUBSYS parameters
YES	Do not require entries and bypass RRT checking (request processing will be performed unconditionally)

RRT - ICHRFR01 Example

```
$ENDEVOR   ICHRFRTB   CLASS=$ENDEVOR,ACTION=RACF

DSNRDB2A   ICHRFRTB   CLASS=DSNR,REQSTOR=IDENTIFY,
                      SUBSYS=DB2A,ACTION=RACF

T$CTSTRN   ICHRFRTB   CLASS=T$CTSTRN,ACTION=RACF

G$CTSTRN   ICHRFRTB   CLASS=G$CTSTRN,ACTION=RACF

END        ICHRFRTB   TYPE=END
```

CLASS DESCRIPTOR TABLE (CDT)

Contains information that governs the management and processing of general resources

RACF references the CDT whenever it receives a RACROUTE request for a class name other than DATASET, USER, or GROUP

CDT Modules

ICHRRCDX	**IBM-supplied class entries**
ICHRRCDE	**Installation-defined class entries**

CDT Class - supplement / replace ICHRRCDE

- **Profile - name of class**
- **CDTINFO segment - specifies class characteristics**
- **Must be RACLISTed**

CLASS DESCRIPTOR TABLE (CDT)

CDT has a maximum of 1024 entries

- **586 reserved for IBM**
- **438 available of installation-defined classes**

OS images sharing a database

- **May have different CDT tables (different subset of classes)**
- **Identically named classes must have same attributes**

Pre z1.6, CDT entries require corresponding RRT entries when:

- **RACLIST=ALLOWED is specified in the CDT**
- **REQSTOR= and SUBSYS= are required**
- **RACROUTE DECOUPL=NO is selected**

CDT - ICHRRCDE

ICHRRCDE must reside in SYS1.LINKLIB or a library in the linklist concatenation

ICHRRCDE is loaded during IPL

ICHERCDE assembly macro is used to define ICHRRCDE entries

Code ICHERCDE with no parameters as the last entry in the table to mark its end

Pre z1.6, place frequently referenced classes first

Note: Completely deactivate a class and delete all related profiles before removing the class from the CDT

CDT - ICHERCDE Macro Parameters

```
[label] ICHERCDE  CLASS=class-name
             ,ID=number
             ,POSIT=number
             [ ,GROUP=grouping-class |
                 ,MEMBER=member-class ]

             [ ,PROFDEF=YES | NO ]
             [ ,MAXLNTH=8 | number ]
             [ ,MAXLENX=number ]
             [ ,GENERIC=ALLOWED |
                 DISALLOWED ]
             [ ,FIRST=ALPHA | NUMERIC |
                 ALPHANUM | ANY |
                 NONATABC | NONATNUM ]
             [ ,OTHER=ALPHA | NUMERIC |
                 ALPHANUM | ANY |
                 NONATABC | NONATNUM ]
             [ ,CASE=UPPER | ASIS )

[ ,DFTRETC=0 | 4 | 8 ]
[ ,DFTUACC=ALTER | CONTROL |
      UPDATE | READ | NONE ]
[ ,OPER=YES | NO ]

[ ,GENLIST=ALLOWED | DISALLOWED ]
[ ,RACLIST=ALLOWED | DISALLOWED ]
[ ,SIGNAL=YES | NO ]
[ ,RACLREQ=YES | NO ]
[ ,KEYQUAL=0 | nnn ]

[ ,SLBLREQ=YES |NO ]
[ ,EQUALMAC=YES | NO   |
      ,RVRSMAC=YES | NO ]
```

CDT - ICHERCDE Macro Parameters

CLASS=	**Class**	**Name of the resource class; use #, $, @, or numeric in installation-defined names to distinguish from IBM**
ID=	**1 - 255**	**Number stored in profile 1-127 - Reserved for IBM 128-255 - Installation use Serves no purpose Need not be unique**
POSIT=	**0 - 1023**	**Identify sets of classes to be managed collectively (e.g., SETROPTS ACTIVE(class)) 0-18,57-127,528-1023 - IBM 19-56,128-527 - Installation use**

CDT - ICHERCDE Macro Parameters

GROUP= MEMBER=	Class	Name of companion Grouping or Member resource class
PROFDEF=	YES | NO	Allow profiles to be defined
MAXLNTH= MAXLENX=	1 - 246 (8)	Maximum resource name length Maximum ENTITYX keyword length
GENERIC=	ALLOWED DISALLOWED	(z1.8) SETROPTS GENERIC or GENCMD are allowed
FIRST= OTHER=	ALPHA NUMERIC ALPHANUM ANY NONATABC NONATNUM	Type of characters allowed for first and remaining characters in the resource name ALPHA, ALPHANUM, and ANY include #, $, and @; whereas, NONATxxx excludes them

CDT - ICHERCDE Macro Parameters

CASE=	UPPER ASIS	Upper or mixed case characters
DFTRETC=	0 | 4 | 8	Default Return Code
DFTUACC=	Access Level	Default UACC when not specified in RDEFINE; if DFTUACC is omitted, RACF uses default UACC in user's ACEE (from current connect group)
OPER=	YES | NO	OPERATIONS authority applies in granting access
GENLIST= RACLIST=	ALLOWED DISALLOWED	SETROPTS GENLIST and RACLIST are allowed
SIGNAL=	YES | NO	Send an ENF type 62 signal when RACLIST command effects profiles

CDT - ICHERCDE Macro Parameters

RACLREQ=	YES | NO	RACLISTing is required
KEYQUAL=	0 - 123	Number of matching qualifiers in generic profiles to load into user's storage (0 is all qualifiers)
SLBLREQ=	YES | NO	SECLABELs are required
EQUALMAC=	YES | NO	Equal MAC check is required wherein SECLABEL of resource must match user's SECLABEL
RVRSMAC=	YES | NO	Reverse MAC check is to be performed wherein SECLABEL of resource must dominate user's SECLABEL

CDT - ICHRRCDE Example

```
$ENDEVOR  ICHERCDE  CLASS=$ENDEVOR,
                    ID=141,
                    MAXLNTH=246,
                    FIRST=ALPHANUM,
                    OTHER=ANY,
                    POSIT=20,
                    RACLIST=ALLOWED,
                    GENLIST=ALLOWED,
                    DFTUACC=NONE
```

CDT - ICHRRCDE Example

```
T$CTSTRN  ICHERCDE CLASS=T$CTSTRN,
                   GROUP= G$CTSTRN,
                   ID=145,
                   MAXLNTH=13,
                   FIRST=ANY,
                   OTHER=ANY,
                   POSIT=130,
                   DFTUACC=NONE,
                   OPER=NO
```

```
G$CTSTRN  ICHERCDE  CLASS=G$CTSTRN,
                    MEMBER= T$CTSTRN,
                    ID=147,
                    MAXLNTH=13,
                    FIRST=ANY,
                    OTHER=ANY,
                    POSIT=130,
                    DFTUACC=NONE,
                    OPER=NO
```

CDT Class

Introduced with z/OS 1.6 to provides a means of dynamically defining and reconfiguring classes using RACF commands and without requiring an IPL

CDT class profiles are class names (Discrete profiles only)

```
RDEFINE CDT $USRCLS
```

Class attributes are defined in the CDTINFO segment

```
RALTER CDT $USRCLS CDTINFO(DEFAULTRC(8))
```

Class names can now be …

- Less than 4 characters
- Start with a number (unlike the ICHERCDE macro)

SETROPTS LIST displays classes in alphabetic sequence

CDT Class - CDTINFO Segment

POSIT(number | 500)

GROUP(grouping-class) |
MEMBER(member-class)

PROFILESALLOWED(YES | NO)

MAXLENGTH(number | 8)

MAXLENX(number)

GENERIC(ALLOWED |
DISALLOWED)

FIRST(ALPHA | NUMERIC |
NATIONAL | SPECIAL)

OTHER(ALPHA | NUMERIC |
NATIONAL | SPECIAL)

CASE(UPPER | ASIS)

DEFAULTRC(0 | 4 | 8)

DEFAULTUACC(ALTER | CONTROL |
UPDATE | READ | NONE | ACEE)

OPERATIONS(YES | NO)

GENLIST(DISALLOWED | ALLOWED)

RACLIST(DISALLOWED | ALLOWED |
REQUIRED)

SIGNAL(YES | NO)
KEYQUALIFIERS(number | 0)

MACPROCESSING(NORMAL | EQUAL |
REVERSE)

SECLABELSREQUIRED(YES | NO)

CDT Class - CDTINFO Segment

```
CDTINFO INFORMATION
-------------------
CASE = UPPER
DEFAULTRC = 004
DEFAULTUACC = NONE
FIRST = ALPHA NATIONAL
GENLIST = DISALLOWED
GROUP =
KEYQUALIFIERS = 0000000000
MACPROCESSING = NORMAL
MAXLENGTH = 008
MAXLENX = NONE
MEMBER =
OPERATIONS = NO
OTHER = ALPHA NATIONAL
POSIT = 0000000500
PROFILESALLOWED = YES
RACLIST = DISALLOWED
SECLABELSREQUIRED = NO
SIGNAL = NO
```

CDT Class

Access permissions to CDT profiles (UACC & Access List)

READ	List CDT profile
ALTER	Change permissions & delete CDT profile

- Has no influence over access to resources defined to the class

Delegation of administration

- CLAUTH(CDT)
- FIELD class profiles - CDT.CDTINFO.*field* (e.g., CDTOPER)

CDT profiles override entries provided in ICHRRCDE

CDT class must be ACTIVE and RACLISTed

Use CDT2DYN (RACF Downloads) to migrate to CDT class

CDT Class

Be mindful of other classes sharing same POSIT value

Change with <u>extreme</u> care (may need to change profiles too)

- **POSIT**
- **PROFILESALLOWED**
- **MAXLENGTH | MAXLENX**
- **GENERIC**
- **GROUP | MEMBER**
- **FIRST | OTHER**
- **CASE**
- **GENLIST | RACLIST**
- **DEFAULTRC**

Changes only take effect after …

- **SETR RACLIST(CDT) REFRESH**
- **IPL**

ICHERCDE / CDT CLASS COMPARE

ICHERCDE	CDT Class	ICHERCDE	CDT Class
ID	n/a	DFTRETC	DEFAULTRC
POSIT	POSIT	DFTUACC	DEFAULTUACC
GROUP	GROUP	OPER	OPERATIONS
MEMBER	MEMBER	GENLIST	GENLIST
PROFDEF	PROFILESALLOWED	RACLIST	RACLIST
MAXLNTH	MAXLENGTH	RACLREQ	RACLIST(REQUIRED)
MAXLENX	MAXLENX	KEYQUAL	KEYQUALIFIERS
GENERIC	GENERIC	SLBLREQ	SECLABELREQUIRED
FIRST	FIRST	EQUALMAC	MACPROCESSING(EQUAL)
OTHER	OTHER	RVRSMAC	MACPROCESSING(REVERSE)
CASE	CASE		

RED signifies difference in default settings

PROFILES

Discrete

Generic

Grouped

DISCRETE PROFILES

One-to-one relationship of profile to resource

Profile name exactly matches full resource name

Unaffected by resource creation or deletion

Note: ALTER access to a discrete profile allows the user to change the access list

GENERIC PROFILES

One-to-many relationship of profile to resources

Activated by SETROPTS Options

- SETROPTS GENCMD(*class*) - Allows creation of profiles
- SETROPTS GENERIC(*class*) - Activates profiles

Uses masking characters (e.g., **)

Masking character order of precedence - & % * **

Enhanced Generic Naming (EGN) is automatically in effect for general resources -- no option exists to activate or deactivate it

GENERIC PROFILES

%	**Single substitute character**	**PAY%%%%**
*****	**Substitute for**	
	(1) A single qualifier	**JES2.*.JOB**
	(2) Any or no characters at the end of a qualifier	**JES2.G*.JOB**
	(3) Any characters after the end	**JES2.***
******	**Substitute for**	
	(1) Zero or more qualifiers within a resource name	**JES2.**.JOB**
	(2) Zero or more characters after the end	**ISFCMD.****

GENERIC PROFILES

Masking characters may be used in combination

ISFCMD.*.   JES%.***

Masking constraints

- **Profiles ending in %* are not permitted**
- **Only one ** is allowed in a profile**

Masking characters may be used in the first qualifier or as the only qualifier

**** catch-all profile recommended for most resource classes (exceptions - FACILITY and PROPCNTL)**

GENERIC PROFILES

Profile	Matches
PAY%	PAY1
TERM0*	TERM0010 TERM0
JES2.*	JES2.GDISPLAY.JOB JES2.ROUTE
ISFCMD.*.*.JES2	ISFCMD.DSP.ACTIVE.JES2
JES2.**	JES2.GDISPLAY.JOB JES2.ROUTE JES2
ISFCMD.**.JES2	ISFCMD.DSP.ACTIVE.JES2 ISFCMD.MAINT.JES2
**.JES2	ISFCMD.DSP.ACTIVE.JES2

GENERIC & GENCMD ISSUE

Common mistake:

- **Create discrete profiles with generic characters - % * ****
- **Subsequently activate GENCMD or GENERIC**
- **Profiles are meaningless <u>and</u> cannot be administered**
- **Profiles appear in SEARCH results without `(G)` generic indicator**

Corrective action:

- **Deactivate GENERIC and GENCMD**
- **Delete 'generic' discrete profiles**
- **Reactivate GENCMD or GENERIC**
- **Recreate generic profiles**

<u>Caution</u>: Determine effect of deactivating generics before doing so and plan accordingly; for certain classes (e.g., JESJOBS), user access might be inhibited

GENERIC PROFILES - PROGRAM CLASS

Profiles must specify the resident library(s)

- Uses ADDMEM / DELMEM to maintain
- Library dataset name - fully qualified (e.g. SYSA.DB2.LOADLIB)
- Volume (optional)
  - *VOLSER number* - e.g., 123456
  - ****** - IPL volume of the current SYSRES
- Specifies PADCHK or NOPADCHK for program pathing control
- Ex: 'SYS1.LINKLIB'//NOPADCHK

Profile anomalies

- SETR GENERIC(PROGRAM) not required
- % may not be used
- * can be used, but only at the end (e.g., RSH*) or alone (e.g., *)
- ** can be used, but only alone
- “Best Fit” considers resident library as well as program name

GROUPING PROFILES

One-to-many relationship of profile to resources

Defined in the Grouping resource classes (e.g., GCICSTRN)

Enable resources with dissimilar names to be protected by a common profile (e.g., CICS transactions PAY1, RPAY, INQP)

Contain members, which are the resources they protect

```
RDEFINE  G$CTSTRN  PGT1.MGRS  ADDMEM( PAY1  RPAY  INQP )
```

Simplifies administration by replacing many individual member class profiles with a fewer number of grouping profiles

GROUPING PROFILES

Grouping profile names

- **Need not match the names of the resources protected**
- **Can conform to a naming standard meaningful to the organization (e.g., PAY.MGR.TRNS)**

A resource can be a member of more than one Grouping profile (caution - increases complexity)

Member entries can either be discrete (e.g., PAY1) or generic (e.g., PA*)

Grouping profiles can be used in combination with Discrete and Generic profiles in the resource class (caution - increases complexity)

GROUPING PROFILES

Grouping by User Role

```
PAY.ADMN     ADDMEM( PAY0 PYR0 )
    PERMIT ID(PAYADM) ACC(READ)
PAY.CLKS     ADDMEM( PAY0 PYR0 PYU1 PYXC )
    PERMIT ID(PAYCLK) ACC(READ)
PAY.MGRS     ADDMEM( PAY0 PYR0 PYU1 PYXC PYU2 )
    PERMIT ID(PAYMGR) ACC(READ)
```

Grouping by Application Function

```
PAY.MAIN     ADDMEM( PAY0 PYR0 )
    PERMIT ID(PAYADM PAYCLK PAYMGR) ACC(READ)
PAY.UPDTA    ADDMEM( PYU1 PYXC )
    PERMIT ID(PAYCLK PAYMGR) ACC(READ)
PAY.UPDTM    ADDMEM( PYU2 )
    PERMIT ID(PAYMGR) ACC(READ)
```

GROUPING PROFILES

Associated member class must be RACLISTed for the Grouping profiles to take effect (and REFRESHed if changed)

- **SETROPTS RACLIST(class) [REFRESH]**
- **RACROUTE REQUEST=LIST [,GLOBAL=YES]**

During RACLISTing, RACF builds a composite list of profiles by merging the Discrete, Generic, and Grouping profiles

- **Grouping class profiles are processed <u>first</u>**
- **Access for each user and group is based on the highest permitted**
- **UACC is based on the lowest UACC**
- **Auditing is set to be the most inclusive**
- **First WARNING Mode setting encountered is applied**

WARNING: If access lists are long, merged list could exceed maximum profile size cause RACLIST to <u>Fail</u>

GROUPING PROFILES

HCICSFCT ACCTFIL1 ADDMEM(VENDMAST)

UACC(READ) AUDIT(FAILURE(READ)) NOWARNING

HCICSFCT ACCTFIL3 ADDMEM(VENDMAST)

UACC(NONE) AUDIT(ALL) WARNING

ID(ACCTPAY) ACC(UPDATE)

FCICSFCT VENDMAST

UACC(NONE) AUDIT(NONE) NOWARNING

ID(ACCTMGT ACCTPAY) ACC(READ)

Composite Profile VENDMAST

UACC(NONE) AUDIT(ALL) NOWARNING

ID(ACCTMGT) ACC(READ)

ID(ACCTPAY) ACC(UPDATE)

GENERIC PROFILES - RACFVARS

RACFVARS - RACF Variables

Define lists of variables to be used in substitution for specific characters strings in a profile

RACFVARS	&FN	ADDMEM(NYC CLE)
NODES	&FN.USERJ.*	NYC.USERJ.FIN003
		CLE.USERJ.CA7USR

Used to build a single profile that can protect multiple resources having dissimilar names

- **Most useful with classes lacking a grouping class**
- **Also valuable when qualifiers exist across multiple classes**

GENERIC PROFILES - RACFVARS

RACF variables are defined as profiles in RACFVARS Class

```
RDEFINE  RACFVARS  &ABCLST
```

RACFVARS profile names

- Begin with an ampersand '&’
- Up to 8 characters in length (no masking characters)
- Prefix &RAC should be reserved for IBM use
- &RACUID and &RACGPID may not be used

Variable character strings

- Added as members to the RACFVARS profile

  ```
  RALTER  RACFVARS  &ABCLST  ADDMEM( OPN1 )
  ```
- May be up to 39 characters in length (no masking characters)
- Can match one, several, or all qualifiers of a resource name as well as partial qualifiers (e.g., LOCAL.PRT4)

GENERIC PROFILES - RACFVARS

RACFVARS profile UACC determines who can list profile

• READ	- Anyone can list using RLIST
• NONE	- No one can list without administrative authority

RACFVARS profile access list

- Grants no access authority
- ALTER level access grants access list and variable member administrative authority

RACFVARS must be RACLISTed and, if changed, refreshed

```
SETROPTS  RACLIST( RACFVARS )  [ REFRESH ]
```

GENERIC PROFILES - RACFVARS

Within a profile, variable names are terminated by:

• Period .	X.&USERVAR.YZ
• Masking character % * **	X.&USERVAR*.*
• Another variable &	X.&USERVAR&V1.*
• End of the profile	X.&USERVAR
• 8th character following the &	X.&USERVARABC (var + ABC)

May be used in combination and with other masking

&RACLNDE.&ABCLST.*.*

In order of precedence, ‘&’ is considered more specific than other generic characters

GENERIC PROFILES - RACFVARS

&RACLNDE - Local NJE nodes for use with NODES, JESJOBS, and JESSPOOL profiles

RACFVARS	&RACLNDE	ADDMEM(SYSA SYSB)
JESJOBS	CANCEL.&RACLNDE.*	ID(OPERS) ACC(ALTER)
JESSPOOL	&RACLNDE.IBMUSER.**	UACC(NONE)

NODES class is ignored for nodes listed in &RACLNDE - batch USERID propagation is automatic

Recommendation: &RACLNDE should only include names of JES nodes sharing the same RACF database

Always define local node in &RACLNDE - even on a stand-alone system - required for spool reload functions

GENERIC PROFILES - RACFVARS

Define set of JES printers to be managed by a particular group

```
RACFVARS     &PAYP                     ADDMEM( PRT5  PRT23  PRT33 )
WRITER       JES2.LOCAL.&PAYP          ID( PAYROLL ) ACC( ALTER )
```

Allow a group of TSO users access to one another's output

```
GROUP        PAYGRP1                   CO ( HRW  IBS  TU1 )
RACFVARS     &PAYG                     ADDMEM( HRW  IBS  TU1 )
   - either -
JESSPOOL     &RACLNDE.&PAYG.*          ID( PAYGRP1 ) ACC( READ )
   - or -
JESSPOOL     &RACLNDE.*.&PAYG*.*       ID( PAYGRP1 ) ACC( READ )
```

GENERIC PROFILES - RACFVARS

Variables are checked in the order they were added to the profile (RLIST lists the variables in alphanumeric sequence)

RACF attempts to find the sequence of characters matching each variable and stops when the first match is found

Problem #1

- **Intend for resource PAYU.SUBMIT to match profile &U.SUBMIT**
- **&U ADDMEM(PAY PAYU)**
- **RACF will match variable PAY to PAYU - equal to PAY.SUBMIT**
- **Correction - reorder members ADDMEM(PAYU PAY)**

GENERIC PROFILES - RACFVARS

Problem #2

- **Intend for resource A1.ABC to match profile &X%.***
- **&X ADDMEM(A1 A)**
- **RACF will match A1 to A1 in its entirety, leaving no match for %**
- **Correction - reorder members ADDMEM(A A1)**

When combining variables - &A&B - consider all possible combinations and whether an &A member could match an intended &A&B combination

Administrative tip - when adding new members to a RACFVARS profile, delete and recreate in its entirety with the correct member sequence

DETERMINING PROTECTING PROFILE(S)

Finding the protecting profile may require executing the following commands

• **SEARCH CLASS(class)**	**- discrete & generic profiles**
• **RLIST RACFVARS &varname**	**- generics with variables**
• **RLIST mbr-class resource RESGROUP**	**- grouping with discrete**
• **RLIST grp-class profile**	**- grouping with generics**

Analyze resulting profiles and members to determine protection

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 (TSO)
• Result	GROUPA no longer has access	

Before creating new profiles:

- **Examine existing profile protection**
- **Copy current UACC and access list if appropriate**