RSH CONSULTING

RACF and Started Tasks

November 2023

RSH Consulting - Robert S. Hansel

RSH CONSULTING

RSH Consulting, Inc. is an IT security professional services firm established in 1992 and dedicated to helping clients strengthen their IBM z/OS mainframe access controls by fully exploiting all the capabilities and latest innovations in RACF. RSH's services include RACF security reviews and audits, initial implementation of new controls, enhancement and remediation of existing controls, and training.

- www.rshconsulting.com
- 617-969-9050
- www.linkedin.com/company/rsh-consulting-inc.

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

Started Tasks

- System tasks initiated by MVS operator START or S command:

  START *membername*

- Also known as Started Procedures (PROCs)
- Run in separate Address Spaces
- Can be started by a supervisor-state program using the ASCRE macro (Address Space Create) - need not specify a PROCLIB(member)
- Can optionally be assigned a JOBNAME (also known as Started Jobs)
  - JOBNAME assigned using either ...
    - START *membername*,JOBNAME=*jobname*
    - //*jobname* JOB statement in the Started Task's JCL
    - ASCRE ATTR parameter option JOBSPACE; address space name becomes the *jobname*
  - If not specified, *jobname* defaults to *membername*
  - Format of STARTED profiles is *membername.jobname*

Started Tasks

- Include ...
  - System component address spaces started by z/OS during IPL (e.g., ALLOCAS)
  - Master Scheduler (*MASTER*)
  - Subsystems defined in PARMLIB(IEFSSNxx)
    - Primary subsystem - either JES2 or JES3
    - Secondary subsystems (e.g., RACF)
    - Automatically started unless PARMLIB(IEFSSNxx) entry specifies START(NO)
- PROC IEESYSAS (System Address Space Initiator) can be used to start tasks that only need to execute a program and do not need DD allocations
  - IEESYSAS PROC and ID should be given no access authority except to run programs
  - On the SDSF DA panel, tasks started via IEESYSAS may show IEFPROC for ProcStep
- Tasks starting before RACF is initialized
  - On the SDSF DA panel, show a blank for ProcStep
  - Run as Limited Function Address Spaces with 'BYPASS' authority
  - Matching IDs do not show logon activity unless the task transitions to Full Function

Started Tasks

- Common Started Tasks and System Address Spaces
  - ALLOCAS* Allocation services and data areas
  - APPC APPC/MVS Component
  - CATALOG** Catalog Address Space
  - CEA Common Event Adaptor
  - CONSOLE* Console Task
  - GRS* Global Resource Serialization
  - JES2 Job Entry Subsystem
  - OMVS Open Edition MVS (z/OS UNIX)
  - RACF RACF Subsystem Address Space
  - SMF System Management Facilities
  - SMS Storage Management Subsystem
  - TCPIP Transmission Control Protocol / Internet Protocol
  - VLF Virtual Lookaside Facility
  - WLM Workload Manager
  - VTAM Virtual Telecommunications Access Method (often named NET)

* - Start as Limited Function Address Spaces; ** - Start Limited but transition to Full Function

Started Task Procedure Libraries (PROCLIBs)

- Master JCL specifies //IEFJOBS and //IEFPDSI DD library concatenations used as a source for Started Jobs and Started Task procedures

PARMLIB(IEASYSxx) MSTRJCL=(00 | xx)

Either PARMLIB(MSTJCLxx) or SYS1.LINKLIB(MSTJCLxx)

```
//MSTJCLSV JOB MSGLEVEL=(1,1),TIME=1440
//        EXEC PGM=IEEMB860,DPRTY=(15,15)
//STCINRDR DD SYSOUT=(A,INTRDR)
//TSOINRDR DD SYSOUT=(A,INTRDR)
//IEFJOBS  DD DISP=SHR,DSN=TECHSPT.STCJOBS
//IEFPDSI  DD DISP=SHR,DSN=SYS1.PROCLIB
//SYSUADS  DD DSN=SYS1.UADS,DISP=SHR
//*SYSLBC   DD DSN=SYS1.BRODCAST,DISP=SHR
```

- JES2 Parameter JOBCLASS(STC) PROCLIB=nn specifies JES PROC //PROCnn DD library concatenation used as the source for Started Task procedures

JES2 PROC

```
//JES2     PROC  PROC01=PAY.PROCLIB
//IEFPROC  EXEC  PGM=HASJES20
//PROC00   DD  DSN=SYS1.PROCLIB,DISP=SHR
//        DD  DSN=TECHSPT.PROCLIB,DISP=SHR
//PROC01   DD  DSN=USER1.PROCLIB,DISP=SHR
//        DD  DSN=&PROC01,DISP=SHR
//HASPPARM DD  DSN=SYS1.PARMLIB(JES2PARM),DISP=SHR
//HASPLIST DD  DDNAME=IEFRDER
```

HASPPARM

```
PROCLIB(PROC02) DD(1)=(DSN=TEST.USER.PROCLIB),
                DD(2)=(DSN=TEST.APPL.PROCLIB)
JOBCLASS(STC) PROCLIB=00,
```

Started Task Procedure Libraries (PROCLIBs)

- When START *membername* is issued, z/OS searches for a matching membername in the following DD statement sequence: (Note - Master address space is available before JES)
  - MSTJCLxx IEFJOBS Started JOBs
  - MSTJCLxx IEFPDSI Started JOBs and PROCs
  - JESx PROCxx Started PROCs

- First member found in the various DD concatenations whose name matches that specified in the START command is executed (same applies to PROCs for batch jobs and TSO logon)

- Best Practices
  - Do no include application and user PROCLIBs in the MSTJCLxx DD concatenations
  - Use separate JES2 PROCnn concatenations to segregate Started Task and system PROCLIBs from application and user PROCLIBs

DSMON - Started Task Report

```
R A C F     S T A R T E D     P R O C E D U R E S     T A B L E     R E P O R T

FROM PROFILES IN THE STARTED CLASS:
-------------------------------------------------------------------------------
PROFILE                 ASSOCIATED  ASSOCIATED
NAME                    USER        GROUP       PRIVILEGED  TRUSTED  TRACE
-------------------------------------------------------------------------------
CASAM                   CASAM                   NO          NO       NO
CICSP01.* (G)           CICSPRD1    STASKGP     NO          NO       NO
CICST01.CICSTEST        =MEMBER     STCTEST     YES         NO       NO
CICS* (G)               =MEMBER     CICSTSKS    NO          NO       YES
DUMPSRV.* (G)           MVSSYST     STASKGP     NO          YES      NO
HSERVER.* (G)                                   NO          NO       YES
NETA.* (G)              -STDATA NOT SPECIFIED, ICHRIN03 WILL BE USED-
** (G)                  DFLTSTC     STASKGP     NO          NO       YES
```

=MEMBER - assign ID matching PROC name
If assigned USERID does not exist, runs with no ID
Report not generated if STARTED is not active

```
R A C F     S T A R T E D     P R O C E D U R E S     T A B L E     R E P O R T

FROM THE STARTED PROCEDURES TABLE (ICHRIN03)
-------------------------------------------------------------------------------
PROCEDURE       ASSOCIATED      ASSOCIATED
NAME            USER            GROUP           PRIVILEGED     TRUSTED
-------------------------------------------------------------------------------
JES2            JES2                            YES            YES
CICSTOR         CICSPRD         CICSSYS         NO             NO
CICSAOR         CICSPRD         CICSSYS         NO             NO
NETA            $SNETA          NTWKSTC         NO             NO
NETB            $SNETB          NTWKSTC         NO             NO
RCVRY           SYSRCVRY                        YES            NO
*               =                               YES            NO
```

* - all PROCs not specified above
= - assign ID matching PROC name

Started Tasks - USER ID Assignment

- ID is assigned at time of START from either (in sequence) …
  - STARTED Class profile STDATA segment [e.g. CICSP1.* STDATA(USER(*userid*))]
  - Started Task Table (ICHRIN03) entry - membername (e.g., CICS01)
    - ICHRIN03 is used if ...
      - The STARTED class is not active and RACLISTed
      - There is no matching STARTED profile
      - The STARTED profile has no STDATA segment
      - The STDATA segment has no USER
- A Started Task will run without an ID (a.k.a., Undefined User) if ...
  - Neither a STARTED profile nor an ICHRIN03 entry assign an ID
  - The ID assigned does not exist
  - The Group assigned does not exist
  - The assigned ID is not connected to the assigned Group
  - Owner assigned is set by SETROPTS JES(UNDEFINEDUSER) - default is ++++++++
- A Started Task will start even if the assigned ID or the ID's logon group connect is REVOKED

STARTED Profiles and STDATA Segments

```
RDEFINE STARTED BPXAS.* OWNER(STCGRP) +
      DATA('UNIX ADDR SPACE INITIATOR') +
      STDATA( USER(OMVSKERN) GROUP(OMVSGRP) TRUSTED(YES) )
SETROPTS RACLIST(STARTED) REFRESH
```

```
RLIST STARTED BPXAS.* NORACF STDATA

CLASS       NAME
-----       ----
STARTED     BPXAS.* (G)

STDATA INFORMATION
------------------
USER= OMVSKERN
GROUP= OMVSGRP
TRUSTED= YES
PRIVILEGED= NO
TRACE= NO
```

STARTED Profiles and STDATA Segments

- STARTED Class profile
  - Resource - *membername.jobname* (e.g. CICS01.CICSP)
    - Can optionally assign different IDs to the same *membername* with different *jobnames*
  - STDATA Segment
    - USER(*userid* | =MEMBER) | NOUSER
    - GROUP(*groupid* | =MEMBER) | NOGROUP
    - PRIVILEGED(YES | NO)
    - TRUSTED(YES | NO)
    - TRACE(YES | NO)
  - =MEMBER is substituted for the PROCLIB Membername
    - Typically used for USER - assigns an ID that matches the Started Task name
  - STDATA GROUP
    - Optional - if not specified, ID's Default Group is used
    - Can specify any of the ID's connect groups, not just the Default Group
    - Becomes the "current connect group" at logon
  - TRACE can be used to identify Started Tasks logons that occur before SMF initialization and, hence, have no corresponding SMF 30 record

STARTED Profiles and STDATA Segments

- STARTED Class profile (continued)
  - As a General Resource class, an * at the end of a STARTED profile matches any characters that follow, including multiple qualifiers
    - EX: DB2P* could be used instead of DB2P*.* (recommend using .* to avoid confusion)
  - Profile UACC and access list
    - READ - List profile, but not the STDATA segment
    - ALTER - Modify or delete discrete profile, but cannot change the STDATA segment
    - Recommend UACC(NONE) and no permissions
  - To view and manage STDATA segments without SPECIAL, use FIELD class profiles

- STARTED Class
  - RACLIST Required
  - Profile changes require a SETROPTS RACLIST(STARTED) REFRESH
  - STDATA segment changes take affect immediately - no REFRESH required
    - STDATA segment is fetched from the database for each start

STARTED Profiles - Implementation Strategies

- USER Started Task ID assignment - options
  - Unique ID for each Started Task (e.g. FTPSERVE)
  - Shared ID for sets of related Started Tasks (e.g., DB2 subsystem tasks - DB2P)

- USER Started Task ID naming convention - options
  - IDs match Started Task PROC membernames (favored)
    - Enables use of USER(=MEMBER)
    - Easier to identify corresponding Started Task in access list entries and ICH408I messages
  - IDs follow naming convention unique to Started Task IDs but dissimilar to PROC membernames (e.g., $STCBMC1)
    - Recommend including procedure-name in the ID's NAME field
      - NAME appears in ICH408I messages
      - Facilitates identifying the associated Started Task

STARTED Profiles - Implementation Strategies

- Optionally (recommended) define a STARTED catch-all profile - ** or *.* - with either of the following settings …

Assign ID = PROC USER(=MEMBER) GROUP(*no-authority-groupid*)

- Intent is for most Started Tasks to use this profile
- Assigns IDs to Started Tasks that match PROC names
- Need only define Started Tasks requiring other characteristics (e.g., TRUSTED)
- All IDs share a common logon group - permit group no access
- PROCs without matching IDs will run as undefined users

Assign default ID USER(*no-authority-id*) GROUP(*no-authority-groupid*) (favored)

- All known Started Tasks must be defined with STARTED profiles
- Default ID is assigned to unknown, undefined Started Tasks
- Default ID should be PROTECTED, REVOKED, and RESTRICTED, with UAUDIT, OMVS(NOUID), and no permissions
- GROUP is a unique group with OMVS(NOGID), no users other than the default ID, and no permissions
- Will cause unknown tasks to fail

STARTED Profiles - Implementation Strategies

- GROUP assignment - options
  - Common/shared default/logon group for all or most Started Tasks (e.g., STCGRP)
    - Ensure no access is permitted to the group
      - Difficult to manage - access often permitted mistakenly and remediation can be complicated
    - Assign unique OMVS GID - ensure no Unix access is permitted to the GID
  - Unique default/logon group for each Started Task (e.g., GCICSP1) (favored)
    - Shared default group for set of like tasks can be used instead of a shared ID
      - Group could be used to permit access
    - Assign unique OMVS GID to each group
  - Combination - common/shared group for certain Started Tasks (e.g., OMVSGRP) and unique default/logon group for all other Started Tasks
    - Assign unique OMVS GIDs
  - GROUP(logon-group) need not be the ID's default group and instead could be set to any of the ID's other connect groups (not favored)

STARTED Profiles - Implementation Strategies

- STARTED profile design - options
  - Fully qualified - CICST0A.CICST0A
    - Use if Started Task PROCs with specific jobnames should be assigned unique IDs
  - Generic for JOBNAME qualifier - CICST0A.* (favored)
  - Generics in MEMBERNAME qualifier - CICST*.* (use with caution)
    - Ensure the profile only matches the desired Started Task membernames (e.g., CICSTST1 and CICSTORP)
    - If specifying USER(=MEMBER), be mindful of IDs matching the membername generic that are not intended to be used as Started Task IDs
    - Best to also specify GROUP to avoid aforementioned issues
  - Generic for JOBNAME can be either .* or .** - opt for .* to avoid undercutting

    SAMPSTC.** - Defined first

    SAMPSTC.* - Defined second - takes precedence - might change assigned ID or attributes

ICHRIN03

- Created and maintained using hard-coded assembly MACRO
- Static - IPL needed to change
- Stored in SYS1.LPALIB concatenation
- Required in order for RACF to initialized during IPL
  - IBM provides a placeholder table with no entries
- Can be reviewed using DSMON
- Table entries
  - PROCNAME | * 8 characters (* - catch-all entry - matches any PROCNAME)
  - USERID | = 8 characters (= - assign USERID matching PROCNAME)
  - Group 8 characters (optional)
  - Privileged 1 byte flag - x'80'
  - Trusted 1 byte flag - x'40'
  - Zero-fill 7 bytes
- Catch-all * entry must be placed last
  - Specifying Group recommended, especially if '=' is specified for USERID

ICHRIN03 - Sample

```
NUMEN    EQU   ((LAST-ENTRY1)/32)+X'8000'
ENTRIES  DC    AL2(NUMEN)
ENTRY1   EQU   *
         DC    CL8'A01PKTIP',CL8'PKTMAIN ',CL8'SYS1    ',XL1'40',XL7'00'
         DC    CL8'DBQADBM1',CL8'DB2     ',CL8'SYS1    ',XL8'00'
         DC    CL8'DBQADIST',CL8'DB2     ',CL8'SYS1    ',XL8'00'
         DC    CL8'DBQAIRLM',CL8'DB2     ',CL8'SYS1    ',XL8'00'
         DC    CL8'DBQAMSTR',CL8'DB2     ',CL8'SYS1    ',XL8'00'
         DC    CL8'DBQASPAS',CL8'DB2     ',CL8'SYS1    ',XL8'00'
         DC    CL8'GTF     ',CL8'GTF     ',CL8'SYS1    ',XL8'00'
         DC    CL8'INETD   ',CL8'OMVSKERN',CL8'OMVSGRP ',XL8'00'
         DC    CL8'TCPIPL62',CL8'TCPIP   ',CL8'SYS1    ',XL8'00'
         DC    CL8'*       ',CL8'=       ',CL8'        ',XL8'00'
LAST     EQU   *
```

- Converting ICHRIN03 entries to STARTED profiles
  - SYS1.SAMPLIB(ICHSPTCV)
  - Reference - RACF Systems Programmer Guide

Started Tasks - Best Practices

- All Started Tasks should be assigned an ID
  - Regularly check SMF 30 records for undefined user tasks
- Use the STARTED class for assigning IDs to Started Tasks instead of ICHRIN03
  - Define a catch-all ** profile with appropriate USER setting
- Avoid sharing IDs among dissimilar Started Tasks
- Do not use Started Task IDs for other purposes (e.g., FTP, Batch)
  - Use PROPCNTL class profiles to prevent Started Task ID propagation onto batch jobs
- Add placeholder IDs and STARTED profiles for Limited Function Address Spaces that do not transition to Full Function simply to reserve the name (e.g., CONSOLE)
  - In NAME and DATA field indicate ID should not be deleted even though inactive

Started Tasks - Best Practices

- Isolate Started Task IDs from other types of IDs in their own default and access groups
- As a general rule, permit access to the IDs themselves, not to their groups
- Specify STDATA GROUP(…) parameter for STARTED profiles with generics in the first qualifier, especially the catch-all profile
  - Prevents assignment of IDs not intended for use with Started Tasks
  - Also specify GROUP in ICHRIN03 entry =
- Except as noted above, do not specify the STDATA GROUP(…) parameter to let default groups be used for logon
  - Facilitates changes to default groups without having to update STDATA segments
- Make Started Task IDs PROTECTED
- Strictly control UPDATE access to Started Task PROCLIB libraries

Started Tasks - Best Practices

- Use TRACE to track ID assignments for profiles with a generic in the first qualifier, including the catch-all profile, and for Started Tasks started before SMF

```
START TRACETST
IRR812I PROFILE TRACE*.* (G) IN THE STARTED CLASS WAS USED
        TO START TRACETST WITH JOBNAME TRACETST.
$HASP100 TRACETST ON STCINRDR
IEF695I START TRACETST WITH JOBNAME TRACETST IS ASSIGNED TO USER RLW
 , GROUP RSHDFLT
$HASP373 TRACETST STARTED
```

- (Optional) Code ICHRIN03 to include backup entries for critical Started Tasks just in case the STARTED class is disabled or essential profiles are deleted

CATALOG	DUMPSRV	IEEVMPCR	IOSAS	IXGLOGR
JES2	JESXCF	LLA	OMVS	RACF
RMF	RMFGAT	SMF	SMS	SMSVSAM
TCPIP	TSO	TCAS	VLF	VTAM
XCFAS				

PRIVILEGED and TRUSTED Authority

- Grants unrestricted access to nearly all resources, even unprotected ones
- If the ID assigned to the Started Task has an OMVS UID, the task will obtain z/OS UNIX Superuser (UID 0) authority
- Only applies to Started Tasks
  - Assigned via STARTED class profiles or ICHRIN03 table entries
  - Authority is assigned to the task itself, not to its ID
  - Authority does not transfer to batch jobs submitted by the Started Task
  - Limited Function Started Tasks automatically run "TRUSTED" (e.g., CONSOLE)
  - When both are specified, PRIVILEGED is used
- TRUSTED can be logged via ...
  - User UAUDIT
  - SETROPTS LOGOPTIONS(ALWAYS(*class*))
- Best Practice:
  - Never assign PRIVILEGED - use TRUSTED instead
  - Only assign TRUSTED as recommended by IBM

PRIVILEGED and TRUSTED Authority

- IBM recommended TRUSTED Started Tasks

APSWPROx(1)	CATALOG	CEA(2)	DFHSM(1)	DFS(1)
DUMPSRV	GPMSERVE(1)	GSKSRVR	HIS	IEEVMPCR
IOSAS	IXGLOGR	JESn	JESXCF	JES3AUX
LLA	NFS	OMVS(1)	RACF	RMF
RMFGAT	SMF	SMS	SMSVSAM(1)	TCPIP
VLF	VTAM	WLM	XCFAS	ZFS(1)

(1) Optional (2) If using z/OSMF ISPF

IBM Manual: MVS Initialization and Tuning Reference (System Tailoring - Assigning the RACF TRUSTED Attribute)

- Remediation of inappropriate PRIVILEGED/TRUSTED assignments
  - Replace PRIVILEGED authority with TRUSTED authority
  - Use UAUDIT to log activity of Started Task IDs to determine access needs
    - Be mindful of Started Tasks that use Unix and generate an excessive volume of SMF records
  - Permit authorized access
  - Remove TRUSTED during a system maintenance period and just prior to an IPL