



CONSULTING

RACF & Started Tasks

November 2017 - RUGONE



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

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 via the ASCRE macro (Address Space Create)

- Can optionally be assigned a 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*
 - Also known as Started Jobs
 - If not specified, *jobname* defaults to *membername*

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 Started 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

* - 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.BROADCAST,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)



- z/OS searches for a matching membername in the following DD statement sequence:
 - MSTJCLxx IEFJOBS
 - MSTJCLxx IEFPDSI
 - JESx PROCxx

- Datasets defined in the Master Scheduler JCL that do not specify VOL=SER= must be cataloged in the Master Catalog

- First member found in the various DD concatenations whose name matches that specified in the START command is executed (applies to batch, Started Tasks, TSO logons)

- Best Practice - use separate 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 NAME	ASSOCIATED USER	ASSOCIATED 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 NAME	ASSOCIATED USER	ASSOCIATED 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



- USERID is assigned at time of START from (in sequence) ...
 - STARTED Class profile - membername.jobname (e.g. CICS01.CICSP)
 - Started Task Table (ICHRIN03) - 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 the STARTED profile nor 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(++))

- A Started Task will start even if the assigned ID or the ID's logon group connect is REVOKED

Started Tasks



```
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 Tasks



- STARTED Class profile
 - Resource - membername.jobname (e.g. CICS01.CICSP)
 - STDATA Segment
 - ❖ USER(*userid* | =MEMBER) | NOUSER
 - ❖ GROUP(*groupid* | =MEMBER) | NOGROUP
 - ❖ PRIVILEGED (YES | NO)
 - ❖ TRUSTED(YES | NO)
 - ❖ TRACE(YES | NO)
 - =MEMBER is substituted for the membername
 - STDATA GROUP
 - ❖ Optional - if not specified, ID's Default Group is used
 - ❖ Can be any of the ID's connect groups, not just the Default Group
 - ❖ Becomes the "current connect group" at logon
 - 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
 - To view and manage STDATA segments without SPECIAL, use FIELD class profiles

Started Tasks



- STARTED Class profile (continued)
 - TRACE can be used to identify Started Tasks logons that occur before SMF initialization and, hence, have no corresponding SMF 30 record
 - 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 .*)

- 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 Tasks



■ Implementation strategies

• ID assignment - options

- ❖ Unique USERID for each Started Task
- ❖ Shared USERID for sets of Started Tasks (e.g., DB2 subsystem tasks)

• ID naming convention - options

- ❖ USERID matches name of Started Task procedure-name (PROC) (favored)
- ❖ IDs follow naming convention unique to Started Task IDs but dissimilar to PROC (e.g., \$STCBMC1)

• Define a catch-all profile - ** or *.* - with either of the following options ...

Assign PROC ID - USER(=MEMBER) GROUP(*groupid*)

Need only define Started Tasks requiring other characteristics

Assign bogus ID - USER(*no-auth-id*) GROUP(*no-auth-group*)

Give *no-auth-id* REVOKE, RESTRICTED, UAUDIT, and
OMVS(NOUID)

Must define all Started Tasks with specific profiles

Started Tasks



- Implementation strategies (continued)
 - Group assignments - options
 - ❖ Common default/logon group for all or most Started Tasks (favored)
 - Ensure no access is permitted
 - Assign unique OMVS GID - ensure no Unix access is permitted
 - ❖ Unique default/logon group for each Started Task
 - Shared default group for set of like tasks in lieu of shared USERID
 - Could be used to permit access
 - Assign unique OMVS GID
 - ❖ Unique default/logon group for certain Started Tasks (e.g., OMVSGRP)
 - Assign unique OMVS GID
 - ❖ Default group and logon group (GROUP=) could be different (not favored)
 - Profile design - options
 - ❖ Fully qualified - CICST0A.CICST0A (use if Started Task PROCs with specific jobnames should have unique IDs)
 - ❖ Generic for JOBNAME qualifier - CICST0A.* (recommended)
 - ❖ Generics in MEMBERNAME qualifier - CICST*.* (Use with caution)
 - ❖ For Generics, profiles can end in .* or .** (.* recommended to avoid undercutting)



- Created and maintained using hard-coded assembly MACRO
- Static - IPL needed to change
- Required in order to IPL - IBM provides a placeholder table with no entries
- Stored in SYS1.LPALIB
- 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'DBQADB1',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'OMVSRP  ',XL8'00'  
          DC     CL8'TCPIPL62',CL8'TCPIP    ',CL8'SYS1    ',XL8'00'  
          DC     CL8'*        ',CL8'=        ',CL8'      ',XL8'00'  
LAST     EQU    *
```


Started Tasks



- Best Practices
 - Use the STARTED class for assigning IDs to Started Tasks
 - 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 for Limited Function Address Spaces that do not transition to Full Function to reserve the name (e.g., CONSOLE)
 - Make Started Task IDs PROTECTED
 - Isolate Started Task IDs in their own default and access groups
 - As a general rule, permit access to the IDs themselves, not to their groups
 - Specify GROUP(...) parameter for STARTED profiles with generics in the first qualifier, especially the catch-all profile
 - ❖ Prevent use of ID not intended for logon as a Started Task
 - ❖ Also specify GROUP in ICHRIN03 = entry

Started Tasks



■ Best Practices

- Use TRACE to track ID assignments for profiles with a generic in the first qualifier, including the catch-all profile

```
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
```

- ICHRIN03 - Include backup entries for critical Started Tasks just in case STARTED class is disabled or essential profiles are deleted

CATALOG	DUMPSRV	IEEVMPCR	IOSAS	IXGLOGR
JES2 or JES3	JESXCF	LLA	OMVS	RACF
RMF	RMFGAT	SMF	SMS	SMSVSAM
TCPIP	TSO	TCAS	VLF	VTAM
XCFAS				

- Strictly control UPDATE access to Started Task PROCLIB libraries

PRIVILEGED and TRUSTED Authority



- Grants unrestricted access to nearly all resources
- If the ID has a 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)
- 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 (1) Optional (2) If using z/OSMF ISPF

APSWPROx ⁽¹⁾	CATALOG	CEA ⁽²⁾	DFHSM ⁽¹⁾	DFS ⁽¹⁾
DUMPSRV	GPMSERVE ⁽¹⁾	HIS	IEEVMPCR	IOSAS
IXGLOGR	JESn	JESXCF	JES3AUX	LLA
OMVS ⁽¹⁾	NFS	RACF	RMF	RMFGAT
SMF	SMS	SMSPDSE1	SMSVSAM ⁽¹⁾	TCPIP
VLF	VTAM	WLM	XCFAS	ZFS ⁽¹⁾

■ Remediation of excessive 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

Common Findings - Started Tasks



- Dissimilar Started Tasks sharing IDs rather than individual ID assignments
 - Most prevalent when a common, shared, default Started Task ID is assigned via a catch-all profile
- Use of PRIVILEGED authority instead of TRUSTED
- Excessive use of TRUSTED authority - assigned to non-IBM-designated tasks
- GROUP not specified for STARTED profiles with generics in the first qualifier, especially with the catch-all STARTED profile and/or ICHRIN03 = entry
- ICHRIN03 entries inconsistent with STARTED profiles
 - Especially problematic if STARTED profiles fail to assign IDs, thereby using ICHRIN03
- JES JOBCLASS(STC) PROCLIB=nn concatenation includes application and user libraries
- UPDATE access to Started Task libraries is not strictly controlled