

SAFTRACE Demystified!

Mark Nelson, CISSP®, CSSLP®
IBM

November 2019
Session **FK**



Agenda

- **What is SAFTRACE?**
- **When should SAFTRACE be used?**
- **Who uses SAFTRACE?**
- **Where does the tracing occur?**
- **How is SAFTRACE used?**
- **A SAFTRACE Example**
- **Where to get more information**

A big thank you to Russ Hardgrove and John Reale for creating the original versions of this session and to the RACF Service Team who really know how to use SAFTRACE!



What is SAFTRACE?

- **SAFTRACE is a RACF-provided tracing facility that allows the tracing of:**
 - RACROUTEs
 - RACF callable services
 - RACF database requests (“ICHEINTY”)
- **Creates a trace record before and after each service is called**
- **Writes records to the z/OS Generalized Trace Facility (GTF)**
- **Formatted with IPCS, using IPCS exit IRRUSR57 (alias AMDUSR57) which is shipped by RACF.**



When should SAFTRACE be used?

- **When you need/desire to know what security calls are being made by a resource manager**
 - Unexpected actions being taken by resource managers after a RACF call
 - Absence/excessive number of log records
- **If excessive contention for the RACF database is being experienced**
 - ... by tracing ALTER requests to the database
- **Excessive**
 - RACROUTEs
 - RACF callable services
 - I/O RACF database requests (“ICHEINTY”)
 - ENQ contention for the RACF database



Who uses SAFTRACE?

- Intended for use under the direction of the RACF support team
- Requires a ***strong*** working knowledge of RACF interfaces and the z/OS security architecture (ACEEs, RACROUTE...)
- Requires a ***strong*** working knowledge of the resource managers whose RACF requests are being traced
- SAFTRACE is oriented towards the z/OS Systems programmer
- Must be willing to wade through a high volume of data



Where does the Tracing Occur?

- The primary trace point is in the SAF router modules, ICHSFR00 and IRRSFR11
- All calls made using RACROUTE or the RACF Callable Service Interface can be traced
- Calls made to the RACF data base manager interface (“ICHEINTY”) can be traced
- RACF invocations that are made using the “Independent System Macros (RACINIT, RACHECK, RACLIST, RACDEF, FRACHECK, RACSTAT) are not traced, other than by database (“ICHEINTY”)



How is SAFTRACE Used?

- **The seven SAFTRACE steps:**
 1. Ensure that the RACF subsystem is active
 2. Determine what events you want to trace
 3. Set the trace options using the RACF SET command
 4. Start GTF / ensure that GTF is running
 5. Recreate the scenario that is to be traced
 6. Stop the trace/GTF
 7. Formatting and reviewing the trace output



How do I Decide what to Trace?

- **The goal is to trap only those events which are relevant to the investigation.**
- **Key questions:**
 - Do the requests come from a known set of jobs or user IDs?
 - Are z/OS UNIX System Services functions being invoked?
 - Are the requests access control questions (REQUEST=AUTH, REQUEST=FASTAUTH...)?
 - Are they directed to specific classes?
 - Are the requests related to authentication (REQUEST=VERIFY, REQUEST=VERIFYX...)
 - Are other RACF requests potentially involved?
 - Is a trace of RACF data base I/O required?



Setting the SAFTRACE Options

- The RACF subsystem SET command is used to set the TRACE options

```

SET TRACE (
  APPC | NOAPPC
  ASID(asid ... | *) | ALLASIDS | NOASID
  CALLABLE(ALL | NONE | TYPE(type ...)) | NOCALLABLE
  CLASS(class-name ... | *) | ALLCLASSES
  IFCLASS(class-name ... | *)
  NEVERCLASS(class-name ... | *)
  NOCLASS
  NODATABASE | DATABASE(
    ALL | NONE
    ALTER | NOALTER
    ALTERI | NOALTERI
    READ | NOREAD )
    JOBNAME(jobname ... | *) | ALLJOBNAMES | NOJOBNAME
  PDCALLABLE(ALL | NONE | TYPE(type ...)) |
NOPDCALLABLE
  RRSF | NORRSF
  RACROUTE(ALL | NONE | TYPE(type ...)) | NORACROUTE
  SYSTEMSSL | NOSYSTEMSSL
  USERID(userid ... | *) | ALLUSERIDS
  IFUSERID(userid ... | *)
  NEVERUSERID(userid ... | *)
  NOUSERID

```

Do I Need to Worry About Performance?

- From the “SAFTRACE Performance Considerations” in the RACF Diagnosis Guide:

“Security as implemented on the OS/390, z/OS platform includes many calls to the security product. This trace facility can adversely affect system performance by adding to the path length associated with performance sensitive security functions.

“This trace should only be used as a debugging aid. Caution should be exercised when designing the trace (as with any other trace) to impose the least performance penalty. For example, if the address space ID or jobname is known, use these to restrict the scope of the trace.”





SAFTRACE in Action: Example 1

Let's explore why a user can delete a data set to which they have no access

Example: Why Can Bert Delete Ernie's data?

- Meet Bert. He wants to copy member SOMEDATA from 'ERNIE.TOOLS.CNTL' into his own data set 'BERT.TOOLS.CNTL'.
- He has no access to 'ERNIE.TOOLS.CNTL' and so he is denied access and receives a 913-38 abend and an ICH408I message for his troubles.

```

//SAFTEBRW JOB 'D5202P,?' ,MSGLEVEL=(1,1) ,CLASS=A,MSGCLASS=H,
// REGION=0M,USER=BERT,NOTIFY=MARKN
//COPY      EXEC PGM=IEBGENER
//SYSIN     DD   DUMMY
//SYSPRINT  DD   SYSOUT=*
//SYSUT1    DD   DISP=SHR,DSN=ERNIE.TOOLS.CNTL(SOMEDATA)
//SYSUT2    DD   DISP=SHR,DSN=BERT.TOOLS.CNTL(SOMEDATA)

11.36.41 JOB00031 ---- WEDNESDAY, 16 OCT 2019 ----
11.36.41 JOB00031 ICH70001I BERT      LAST ACCESS AT 22:14:30 ON TUESDAY, OCTOBER 15, 2019
11.36.41 JOB00031 $HASP373 SAFTEBRW STARTED - INIT 1      - CLASS A
11.36.41 JOB00031 ICH408I USER(BERT      ) GROUP(SYS1      ) NAME(#####) 621
        621          ERNIE.TOOLS.CNTL CL(DATASET ) VOL(SMSVL1)
        621          INSUFFICIENT ACCESS AUTHORITY
        621          FROM ERNIE.** (G)
        621          ACCESS INTENT(READ  ) ACCESS ALLOWED(NONE  )
11.36.41 JOB00031 IEC150I 913-38,IFG0194E,SAFTEBRW,COPY,SYSUT1,1080,SMSVL1, 622
        622          ERNIE.TOOLS.CNTL(SOMEDATA)
11.36.41 JOB00031 IEA995I SYMPTOM DUMP OUTPUT 623
        623          SYSTEM COMPLETION CODE=913 REASON CODE=00000038
        623          TIME=11.36.41 SEQ=00018 CPU=0000 ASID=0029
  
```

Example: Why Can Bert Delete Ernie's data?

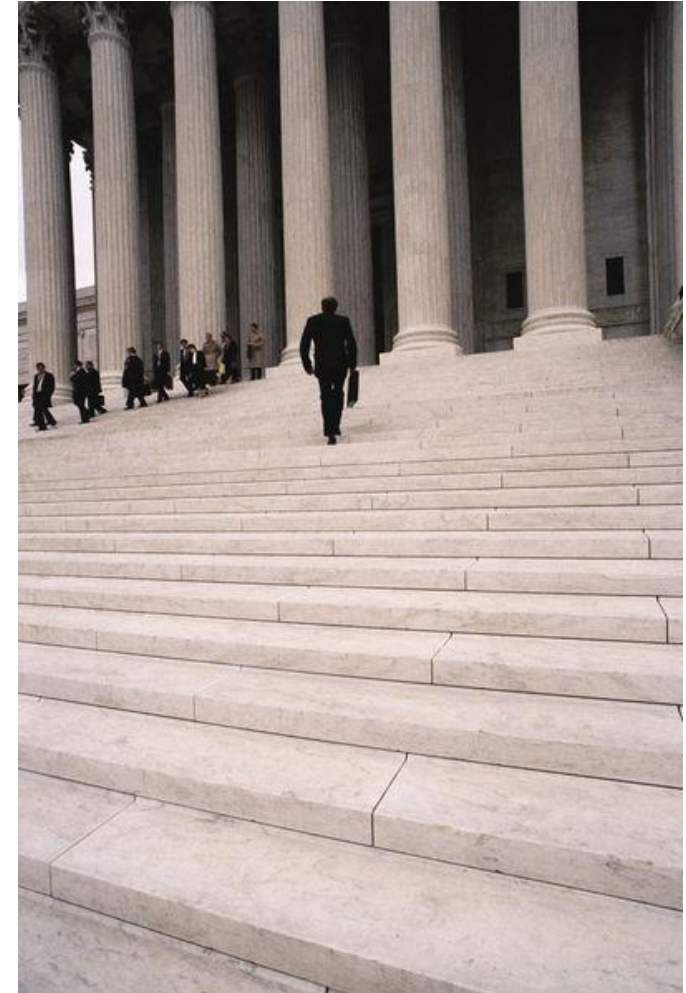
- **But, it turns out that Bert can delete Ernie's data set!**
- **WHY?**

```
//SAFTEDEL JOB 'D5202P,?' ,MSGLEVEL=(1,1) ,CLASS=A,MSGCLASS=H,
// REGION=0M,USER=BERT,NOTIFY=MARKN
//COPY      EXEC PGM=IEFBR14
//DD1      DD   DISP=(SHR,DELETE) ,DSN=ERNIE.TOOLS.CNTL
```

```
ICH70001I BERT      LAST ACCESS AT 12:41:00 ON WEDNESDAY, OCTOBER 16, 2019
IEF236I ALLOC. FOR SAFTEDEL COPY
IGD103I SMS ALLOCATED TO DDNAME DD1
IEF142I SAFTEDEL COPY - STEP WAS EXECUTED - COND CODE 0000
IGD105I ERNIE.TOOLS.CNTL                                DELETED, DDNAME=DD1
IEF373I STEP/COPY      /START 2019289.1255
IEF032I STEP/COPY      /STOP 2019289.1255
```

Let's Apply the Steps...

- **Recall the seven SAFTRACE steps:**
 1. Ensure that the RACF subsystem is active. **Done!**
 2. Determining what events you want to trace. **RACROUTE REQUEST=AUTH (but let's trace all RACROUTEs just to see what is happening with job SAFTEDL). Done!**
 3. Set the trace options using the RACF SET command. **Shown in a moment.**
 4. Start GTF / ensure that GTF is running. **Shown in a moment.**
 5. Recreate the scenario that is to be traced. **Shown in a moment.**
 6. Stopping the trace/GTF. **Shown in a moment.**
 7. Formatting the trace output. **We'll spend a lot of time on this in a moment.**



Step 3: Setting the Trace Options

- The **SET TRACE** command is used to set the trace options. Note that it's issued from any place where console commands can be issued.
- If we wanted **REQUEST=AUTH** only, that's service #1 (**TYPE(1)**). But let's get all of the **RACROUTE**s issued by jobname **SAFTEDEL**

```

-----
-- |   Edit  Options  Help
SD | -----
CO |           System Command Extension
0  |
|   ===> @set trace(RACROUTE(all)) jobname(SAFTEDEL)
0  |   ===>
4  |                                           STORELIMIT
0  | Comment
CO |
4  | Group           Show *           (F4 for list)
0  |                                           More:      +
** | =>  STOP GTF
|   =>  STOP GTFMARKN
|   =>  D A,L
|   =>  S HC
|
|   F1=Help       F3=Cancel       F4=Prompt       F5=FullScr     F6=Details
PF |   F7=Up        F8=Down        F10=Save        F11=Clear      F12=Cancel
*SDSF

```

Step 3: Setting the Trace Options...

- It's a Real Good Idea to ensure that you got the trace set as you wanted it to be set.
- The SET LIST command shows the current trace settings

```

-----
-- |   Edit  Options  Help
SD | -----
CO |           System Command Extension
RE |
I  | ==>> @set list
0  | ==>>
4  |                                           STORELIMIT
0  | Comment
CO |
4  | Group           Show *           (F4 for list)
0  |                                           More:      +
** | => @set trace(RACROUTE(all) jobname(SAFTEDEL))
   | => STOP GTF
   | => STOP GTFMARKN
   | => D A,L
   | => $PI2
   | => $PI1-20
   | => S HC
   | => setprog apf,add,dsn=markn.hrf77C0.load,vol=d94rf4
   |
   | F1=Help      F3=Cancel      F4=Prompt      F5=FullScr    F6=Details
PF | F7=Up        F8=Down       F10=Save      F11=Clear    F12=Cancel
   |
*SDSF

```

Step 3: Setting the Trace Options...

- **RACROUTE(ALL) and JOBNAME(SAFTEDEL) are in effect as we requested**

```

-----
SDSF HELD OUTPUT DISPLAY ALL CLASSES LINES 367          13 RESPONSES NOT SHOWN
COMMAND INPUT ===>                                     SCROLL ===> HALF
RESPONSE=SY1
IRRH005I (@) RACF SUBSYSTEM INFORMATION:
  TRACE OPTIONS
- NOIMAGE
- NOAPPC
- NOSYSTEMSSL
- NORRSF
- RACROUTE
  ALL
- NOCALLABLE
- NOPDCALLABLE
- NODATABASE
- NOGENERICANCHOR
- NOASID
- JOBNAME
  SAFTEDEL
- NOCLASS
- NOUSERID
  
```


Step 4: Starting GTF

- **GTF must be started before you run your test scenario. These are the values that I used:**

- **PROCLIB (member GTFMARKN in my PROCLIB concatenation)**

```
//GTFRACF PROC MEMBER=GTFMARKN
//BR14 EXEC PGM=IEFBR14,REGION=512K
//SYSPRINT DD SYSOUT=*
//D DD DISP=(OLD,DELETE),UNIT=3380,VOL=SER=D94RF1,
// DSN=MARKN.GTF.TRACE
//IEFPROC EXEC PGM=AHLGTF,PARM='MODE=EXT,DEBUG=NO,SA=100K,AB=100K',
// REGION=2880K,TIME=NOLIMIT
//IEFRDER DD DSN=MARKN.GTF.TRACE,UNIT=3380,VOL=SER=D94RF1,
// DISP=(NEW,CATLG),SPACE=(TRK,(30,10))
//*EFRDER DD DSN=SYS1.TRACE,UNIT=3390,VOL=SER=TEMP03,
//* DISP=(NEW,CATLG),SPACE=(TRK,(100))
//SYSLIB DD DSN=RACFDRVR.PARMLIB.ZR13(&MEMBER),DISP=SHR
```

- **PARMLIB (member GTFMARKN in my PARMLIB concatenation)**

```
TRACE=USRP
USR=(F44)
END
```

Step 4: Starting GTF...

- Starting GTF using the PARMLIB/PROCLIB shown earlier

```

-----
-- |   Edit  Options  Help                               |  --
SD | -----|
CO |                               System Command Extension |  F
PR |
NP | ==>> s gtfmarkn.gtf,,,noprompt                       |
   | ==>>
   |
   |                               STORELIMIT              |
   | Comment
   |
   | Group                               Show *           (F4 for list)
   |                               More:                +
   |
   | => @set list
   | => @set trace(RACROUTE(all) jobname(SAFTEDEL))
   | => @set trace(RACROUTE(all) jobname(SAFTEDEL))
   | => @set trace(RACROUTE(type(all)) jobname(SAFTEDEL))
   | => @set trace(RACROUTE(type(1)) jobname(SAFTEDEL))
   | => STOP GTF
   | => STOP GTFMARKN
   | => D A,L
   |
   | F1=Help      F3=Cancel      F4=Prompt      F5=FullScr   F6=Details
PF | F7=Up        F8=Down       F10=Save     F11=Clear    F12=Cancel
   |
//SYSLIB DD DSN=PARMLIB.ZR13(&MEMBER),DISP=SHR

```

Step 4: Starting GTF...

- Console messages after a successful start

```

S GTFMARKN.GTF,,,NOPROMPT
IRR813I NO PROFILE WAS FOUND IN THE STARTED CLASS FOR 608
      GTFMARKN WITH JOBNAME GTFMARKN. RACF WILL USE ICHRIN03.
$HASP100 GTFMARKN ON STCINRDR
IRR813I NO PROFILE WAS FOUND IN THE STARTED CLASS FOR 610
      GTFMARKN WITH JOBNAME GTFMARKN. RACF WILL USE ICHRIN03.
IEF695I START GTFMARKN WITH JOBNAME GTFMARKN IS ASSIGNED TO USER STCUSER
, GROUP SYSGRP
$HASP373 GTFMARKN STARTED
IEF188I PROBLEM PROGRAM ATTRIBUTES ASSIGNED
AHL121I TRACE OPTION INPUT INDICATED FROM MEMBER GTFMARKN OF PDS
RACFDRVR.PARMLIB.ZR13
      TRACE=USRP
      USR=(F44)
      END
AHL103I TRACE OPTIONS SELECTED --USR=(F44)
AHL906I THE OUTPUT BLOCK SIZE OF      23476 WILL BE USED FOR OUTPUT 619
AHL906I THE OUTPUT BLOCK SIZE OF      23476 WILL BE USED FOR OUTPUT 619
      DATA SETS:
      MARKN.GTF.TRACE
AHL080I GTF STORAGE USED FOR GTF DATA: 620
      GTFBLOCK STORAGE      68K BYTES (BLOK=      40K)
      PRIVATE STORAGE      1031K BYTES (SIZE=     1024K)
      SADMP HISTORY        45K BYTES (SADMP=      40K)
      SDUMP HISTORY        45K BYTES (SDUMP=      40K)
      ABEND DUMP DATA      0K BYTES (ABDUMP=      0K)
AHL031I GTF INITIALIZATION COMPLETE
  
```

Step 5: Recreating the Scenario

- Our test scenario is straightforward: Submitting our batch job

```

-----
EDIT          MARKN.TOOLS.CNTL(SAFTDEL) - 01.04          Columns 00001 00072
*****  ***** Top of Data *****
002200 //SAFTEDEL JOB 'D5202P,?',MSGLEVEL=(1,1),CLASS=A,MSGCLASS=H,
002300 // REGION=0M,USER=BERT,NOTIFY=MARKN
002400 //COPY      EXEC PGM=IEFBR14
002700 //DD1       DD  DISP=(SHR,DELETE),DSN=ERNIE.TOOLS.CNTL
*****  ***** Bottom of Data *****
  
```

```

Command ==> sub                               Scroll ==> PAGE
F1=Help      F2=Split      F3=Exit      F4=Expand    F5=Rfind     F6=Rchange
F7=Up        F8=Down      F9=Swap     F10=Left    F11=Right    F12=Cancel
  
```

Step 5: Recreating the Scenario...

- ... and verifying that we got the result that we want to investigate

```

-----
SDSF OUTPUT DISPLAY SAFTEDDEL JOB00026  DSID      2 LINE 0          COLUMNS 02- 81
COMMAND INPUT ===>                               SCROLL ===> PAGE
***** TOP OF DATA *****
                J E S 2  J O B  L O G  --  S Y S T E M  I B M 2  --  N O

23.11.20 JOB00026 ---- MONDAY,      21 OCT 2019 ----
23.11.20 JOB00026  ICH70001I BERT      LAST ACCESS AT 23:10:16 ON MONDAY, OCTOBER
23.11.20 JOB00026  $HASP373 SAFTEDDEL STARTED - INIT 1      - CLASS A          - SYS
23.11.20 JOB00026  $HASP395 SAFTEDDEL ENDED - RC=0000
----- JES2 JOB STATISTICS -----
  21 OCT 2019 JOB EXECUTION DATE
    4 CARDS READ
    37 SYSOUT PRINT RECORDS
    0 SYSOUT PUNCH RECORDS
    5 SYSOUT SPOOL KBYTES
  0.00 MINUTES EXECUTION TIME
  1 //SAFTEDDEL JOB 'D5202P,?',MSGLEVEL=(1,1),CLASS=A,MSGCLASS=H,
    // REGION=0M,USER=BERT,NOTIFY=MARKN
  2 //COPY          EXEC PGM=IEFBR14

                                IFIND          F6=BOOK
F7=UP                F8=DOWN          F9=SWAP          F10=LEFT          F11=RIGHT          F12=RETRIEVE

```


Step 6: Stopping GTF

- We stop GTF using the MVS STOP command

```

.-----
-- |   Edit  Options  Help
SD | -----
CO |               System Command Extension
009 |
029 | ==> p gtf
029 | ==>
009 |
009 |                                STORELIMIT
009 | Comment
009 |
009 | Group                Show *                (F4 for list)
009 |                                More:      +
009 | =>  D A,L
009 | =>
009 | =>
009 | =>
029 | =>
009 | =>
DUM | =>
DUM | =>
*** |
    |  F1=Help      F3=Cancel    F4=Prompt    F5=FullScr   F6=Details
F   |  F7=Up       F8=Down     F10=Save    F11=Clear    F12=Cancel

```

Step 7: Formatting and Reviewing the Trace

- **IPCS is used to format the SAFTRACE records**

```
//SAFTIPCS JOB 'D5203P,?',
//          'M.NELSON BIN 1016',
//          REGION=0M,
//          MSGLEVEL=(1,1),
//          NOTIFY=&SYSUID,
//          CLASS=A,
//          MSGCLASS=H
//IPCS      EXEC  PGM=IKJEFT01
//IPCSDDIR DD    DISP=SHR,DSN=MARKN.DDIR
//SYSTSPRT DD    SYSOUT=*
//SYSPRINT DD    SYSOUT=*
//SYSTSIN  DD    *
SE 'HELLO, WORLD!'
IPCS NOPARM
SETDEF DSN('MARKN.GTF.TRACE') LIST NOCONFIRM
GTF USR
END
/*
```

Step 7: Formatting and Reviewing the Trace...

- **SAFTRACE output consists three distinct parts**
 1. **Fixed-length header portion**
 2. **The parameters specified on the RACROUTE REQUEST, RACF callable service, or ICHEINTY**
 3. **A “raw” hex dump of the entire GTF record**



Step 7: Formatting and Reviewing the Trace...

- **(1) The Header portion, which contains**

- Date and time of the event
- Service number (RACROUTE or Callable Service)
- Type of request, "RACF", "OMVS", "RACF" appended with "PRE" or "POST"
- Job name/ASID
- Pointers to ACEE
- REQSTOR/SUBSYS for RACROUTE
- RACF return and reason code
- Count of parameters in the request

Following is a formatted R_TRACE record.
 This trace record was generated by IRRTRC00 with IDENT(R_TRACE).

```

Trace Identifier:           00000036
Record Eyecatcher:        RTRACE
Trace Type:                RACFPRE
Ending Sequence:          .....
Calling address:           00000000  8417454E
Requestor/Subsystem:      .....
Primary jobname:          SAFTEDDEL
Primary asid:              00000029
Primary ACEEP:            00000000  009FC870
Home jobname:              SAFTEDDEL
Home asid:                 00000029
Home ACEEP:                00000000  009FC870
Task address:              00000000  009F81A0
Task ACEEP:                00000000  009FC870
Time:                      D6E1A424  4A65BC95
Error class:               .....
Service number:           00000001
RACF Return code:         00000000
RACF Reason code:         00000000
Return area address:       00000000  0000682C
Parameter count:          0000000B
  
```

Step 7: Formatting and Reviewing the Trace...

- (2) The Request Parameter portion, which consists of:
 - The SAF parameter list
 - The function-specific parameter list
 - The individual parameters
 - OFFSETnn is the offset of the next “area value” in the function-specific parameter list
 - In this example, “nn” = x'24', which is the entity name
- The RACF Diagnosis Guide has detailed parameter list diagrams

```

Area length:                00000068

Area value:
00000000  00000000  00A40000  00010000  | .....u..... |
00000000  00000000  000069A4  00000000  | .....u..... |
00000000  00000068  00000000  00000000  | .....       |
00000000  00000000  00000000  00000000  | .....       |
00000000  00000000  00000000  00000000  | .....       |
00000000  00000000  00000000  00000000  | .....       |
00000000  00000000  00000000  00000000  | .....       |

Area length:                0000003C

Area value:
3C000000  9C000000  80000000  00000000  | .....       |
00000000  00000000  00000000  00000000  | .....       |
00000000  00006C48  00006C74  00006C7C  | .....%...%...% |
00000000  00000000  00000000  00000000  | .....       |

Area length:                00000008

Area value:
D6C6C6E2  C5E30024                | OFFSET..     |

Area length:                0000002C

Area value:
1 D7C1C7C5  F0F84BC3  C1E3C1D3  D6C74040  | PAGE08.CATALOG |
40404040  40404040  40404040  40404040  |                |
40404040  40404040  40404040  40404040  |                |

Area length:                00000008

Area value:
D6C6C6E2  C5E30028                | OFFSET..     |

Area length:                00000008

Area value:
07C4C1E3  C1E2C5E3                | .DATASET     |

Area length:                00000008

Area value:
D6C6C6E2  C5E3002C                | OFFSET..     |

Area length:                00000006

Area value:
D7C1C7C5  F0F8                | PAGE08       |

```


Step 7: Formatting and Reviewing the Trace...

- **(3) The raw hex dump**

- I rarely use this....

```

Hexadecimal dump of record follows:
+0000 00000036 D9E3D9C1 C3C54040 D9C1C3C6 | ....RTRACE RACF |
+0010 D7D9C540 00000000 00000000 00000000 | PRE ..... |
+0020 00000000 00000000 00000000 00000000 | ..... |
+0030 83C78020 C9C5C5F0 F0F0F3C4 4040C3D6 | cG..IEE0003D CO |
+0040 D5E2D6D3 C5400000 00000000 00000000 | NSOLE ..... |
+0050 00000000 00000000 00000000 009F81A0 | .....a..... |
+0060 00F53700 00000000 00F53700 00000029 | .5.....5..... |
+0070 00000029 E2C1C6E3 C5C4C5D3 E2C1C6E3 | ....SAFTEDELSAFT |
1 +0080 C5C4C5D3 00000000 00000000 009FC870 | EDEL.....H..... |
+0090 009FC870 00000000 009FC870 D6E1A424 | ..H.....H.O.u. |
+00A0 4368CA98 00000000 00000000 009D2F80 | ...q..... |
+00B0 00000007 0000000E 00000068 00000000 | ..... |
+00C0 00000000 00680000 000E5000 03C789A0 | .....&..Gi. |
+00D0 03C78998 009D2D80 00000000 00000000 | .Giq..... |
+00E0 00000068 00000000 00000000 00400000 | ..... |
+00F0 00000000 00000000 00000000 00000000 | ..... |
+0100 00000000 00000000 00000000 00000000 | ..... |
+0110 00000000 00000000 00000000 00000000 | ..... |
+0120 00000000 0000000C 03C78D50 00000000 | .....G.&.... |
+0130 000C0000 00000008 D6C6C6E2 C5E30000 | .....OFFSET.. |
+0140 00000008 D6D7C5D9 C3D4C4E2 000000C0 | ....OPERCMDS...{ |
+0150 C1C3C5C5 FF0000C0 03E1CE2F 00000000 | ACEE...{..... |
+0160 00000000 04C2C5D9 E3404040 4004E2E8 | .....BERT .SY |
+0170 E2F14040 40400101 0019289F 40404040 | S1 ..... |
+0180 40404040 00000000 00000000 00000000 | ..... |
+0190 40404040 40404040 00000000 00000000 | ..... |
+01A0 00000000 00000000 40404040 40404040 | ..... |
+01B0 00000000 009FC930 00000000 009FC948 | .....I.....I. |
+01C0 00000000 009FC980 00000000 0119289F | .....I..... |
+01D0 00000000 00200000 00000000 00000000 | ..... |
+01E0 00000000 00000000 009FC9B8 00000000 | .....I..... |
+01F0 00000000 009FCA48 00000000 00000000 | ..... |
+0200 00000000 00000000 00000000 12555720 | ..... |
+0210 00000050 50010007 0403C000 00000000 | ...&&.....{..... |
+0220 00000000 E6C3C340 40404040 D4C1D9D2 | ...WCC MARK |
+0230 D5404040 E6C3C340 40404040 E2E8E2F1 | N WCC SYS1 |
+0240 40404040 C9D5E3D9 C4D94040 00000000 | INTRDR .... |
+0250 00000000 C2C5D9E3 40404040 E2E8E2F1 | ...BERT SYS1 |
+0260 40404040 00000090 C1C3C5E7 032DDC4E | ...ACEX...+ |
+0270 00F54278 00000000 00000000 00000000 | .5..... |
+0280 00000000 00000046 000000AD 00000028 | ..... |
+0290 009FCA98 00000000 00000000 00000000 | ...q..... |
+02A0 00000000 00000000 00000000 0000807B | .....# |
+02B0 00000000 00000000 00000000 00000000 | ..... |
+02C0 00000000 00000000 00000000 00000000 | ..... |
+02D0 00000000 00000000 00000000 00000000 | ..... |
+02E0 00000000 00000000 00CD6C78 00290000 | .....%..... |
+02F0 00000000 00000000 | ..... |

```

SAFTRACE Records for our Simple Batch Job

- Our simple Batch job has created 20 SAFTRACE entries, 10 RACFPRE and 10 RACFPOST. These are the RACFPOST records

Event	REQSTOR	SUBSYS	Return and Reason Codes	Description/Comments
TOKENMAP(XM)	IEE0003D	CONSOLE	0/0/0	Not related to our investigation
EXTRACT(BR)	JBSTXTRT	JES2z204	0/0/0	Not related to our investigation
VERIFYX	JBSTTMAP	JES2z204	0/0/0	Not related to our investigation
AUTH			4/4/0	????? authority to FACILITY/IARRSM.LRGPAGES Not related to our investigation
AUTH			8/8/200	????? Authority to XFACILIT/STGADMIN.IGG.DELAUDIT.PAGE08.CATALOG
AUTH			8/8/0	????? Authority to DATASET/ERNIE.TOOLS.CNTL
AUTH			0/0/0	????? Authority to DATASET/PAGE08.CATALOG
AUTH			8/8/0	????? Authority to DATASET/ERNIE.TOOLS.CNTL
DEFINE			0/0/0	DATASET/ERNIE.TOOLS.CNTL
VERIFY			0/0/0	Not related to our investigation

Note: z/OS V2R4 adds a new check for the resource STGADMIN.IGG.DELAUDIT.catalogname in the XFACILIT class

Step 7: Formatting and Reviewing the Trace...

- Let's look at the last three REQUEST=AUTHs
- The X'3C' length is the function-specific parameter list for RACROUTE REQUEST=AUTH
 - Offset X'04' (value X'0C') indicates LOG=NOFAIL was specified and ENTITY (not ENTITYX) was specified
 - Offset X'08 (value X'80') indicates that ATTR=ALTER was specified
 - OFFSETxx (X'24') is the ENTITY name ("ERNIE.TOOLS.CNTL")
 - OFFSETxx (X'28') is the CLASS name ("DATASET")
 - OFFSETxx (X'2C') is the VOLSER ("SMSVL1")
- Where do we find the parameter list mappings?

```

Area length:                00000068

Area value:
00000008 00000000 00A40000 00010000 | .....u..... |
00000000 00000000 000069A4 00000000 | .....u..... |
00000000 00000068 00000000 00000000 | .....u..... |
00000000 00000000 00000000 00000000 | .....u..... |
00000000 00000000 00000000 00000000 | .....u..... |
00000000 00000000 00000000 00000000 | .....u..... |
00000000 00000000 00000000 00000000 | .....u..... |
00000000 00000000 00000000 00000000 | .....u..... |

Area length:                0000003C

Area value:
3C000000 0C000000 80000000 00000000 | .....u..... |
00000000 00000000 00000000 00000000 | .....u..... |
00000000 00006C48 00006C74 00006C7C | .....%...%...% |
00000000 00000000 00000000 00000000 | .....u..... |

Area length:                00000008

Area value:
D6C6C6E2 C5E30024 | OFFSET.. |

Area length:                0000002C

Area value:
C5D9D5C9 C54BE3D6 D6D3E24B C3D5E3D3 | ERNIE.TOOLS.CNTL |
40404040 40404040 40404040 40404040 | .....u..... |
40404040 40404040 40404040 40404040 | .....u..... |

Area length:                00000008

Area value:
D6C6C6E2 C5E30028 | OFFSET.. |

1 Area length:                00000008

Area value:
07C4C1E3 C1E2C5E3 | .DATASET |

Area length:                00000008

Area value:
D6C6C6E2 C5E3002C | OFFSET.. |

Area length:                00000006

Area value:
E2D4E2E5 D3F1 | SMSVL1 |

Area length:                000000C0

```

Step 7: Formatting and Reviewing the Trace...

• RACF Diagnosis Guide Mapping of the SAF parameter List

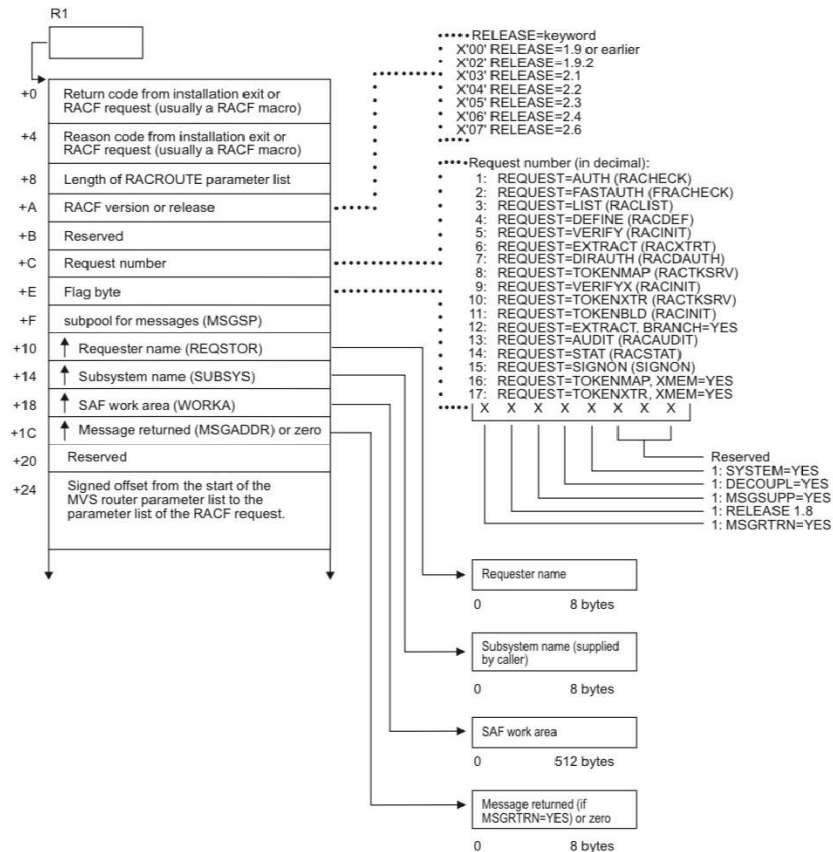


Figure 26. RACROUTE Parameter List (Part 1 of 2)

These areas show up on both MVS and VM systems, but they are used only on VM.

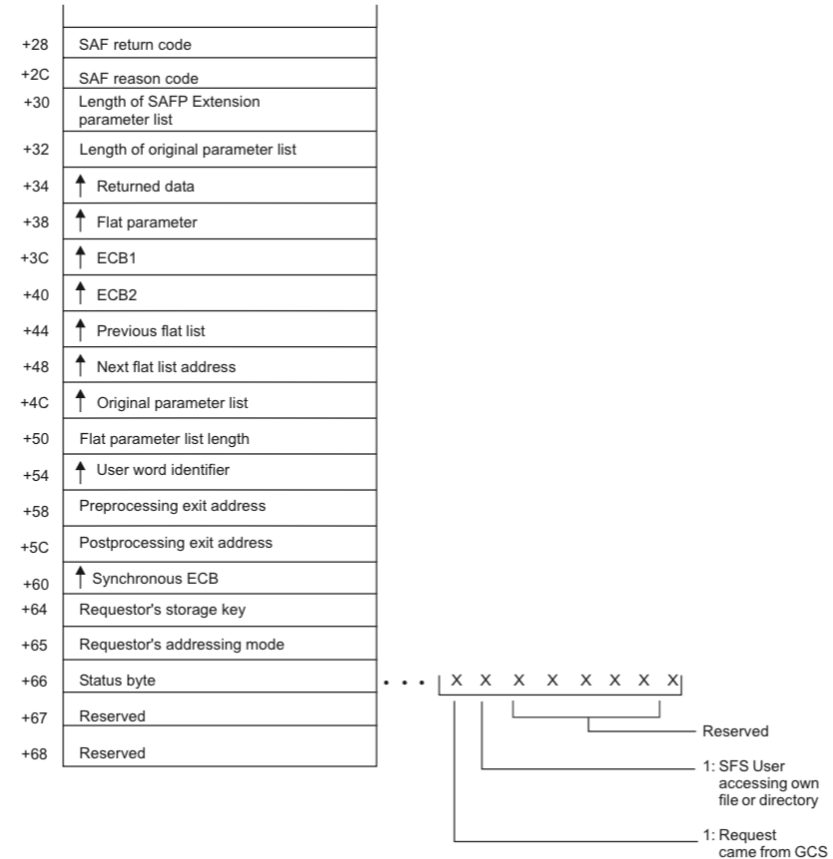


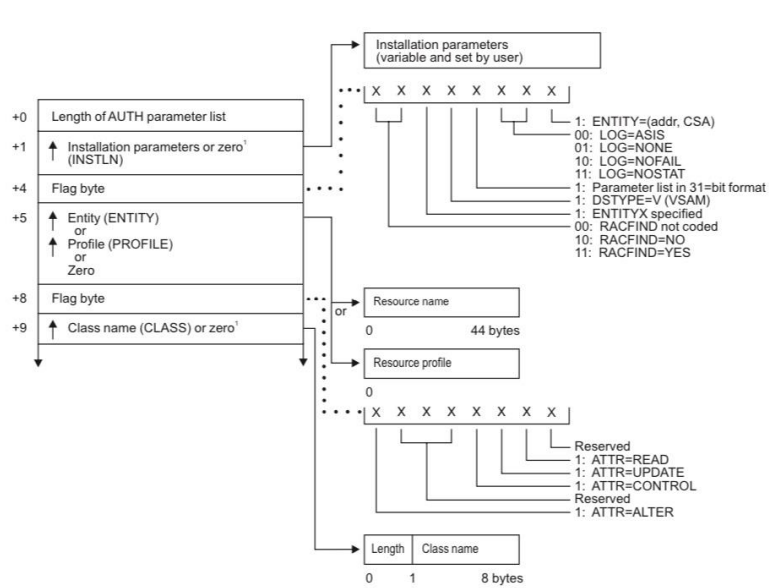
Figure 26. RACROUTE Parameter List (Part 2 of 2)

Step 7: Formatting and Reviewing the Trace..

• RACF Diagnosis Guide Mapping of the REQUEST=AUTH parameter list

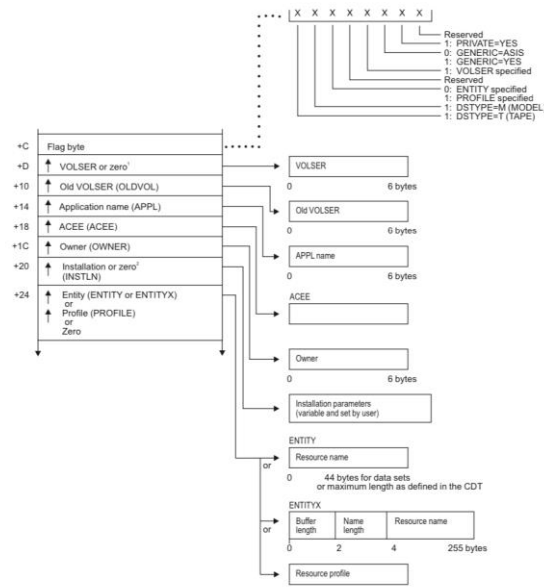
AUTH parameters

AUTH service parameter list



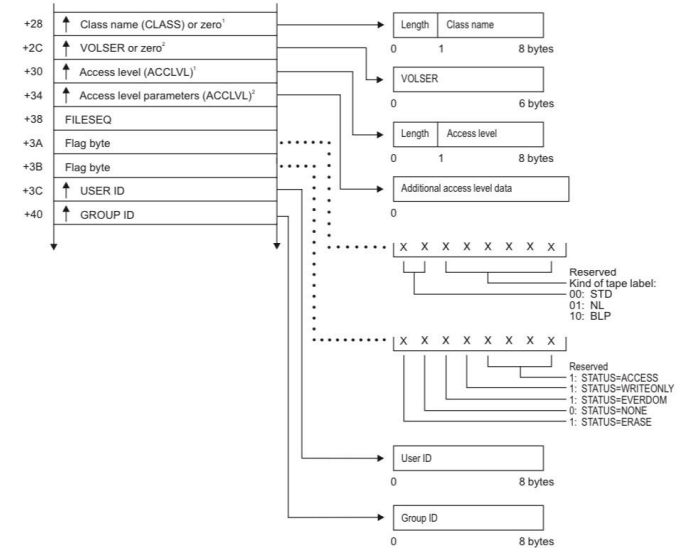
¹ This field is zero if bit 4 of the flag byte at offset 4 is one (input was RACROUTE REQUEST=AUTH).

Figure 28. AUTH Service Parameter List (Part 1 of 4)



¹ This field is zero if bit 4 of the flag byte at offset 4 is one (input was RACROUTE REQUEST=AUTH).
² This field is zero if bit 4 of the flag byte at offset 4 is zero (input was RACHECK macro).

Figure 28. AUTH Service Parameter List (Part 2 of 4)



¹ This field is zero if bit 4 of the flag byte at offset 4 is one (input was RACROUTE REQUEST=AUTH).
² This field is zero if bit 4 of the flag byte at offset 4 is zero (input was RACHECK macro).

Figure 28. AUTH Service Parameter List (Part 3 of 4)

Step 7: Formatting and Reviewing the Trace...

- Let's look at the second of the last three REQUEST=AUTHs
- The X'3C' length is the function-specific parameter list for RACROUTE REQUEST=AUTH
 - Offset X'04' (value X'9C') indicates LOG=ASIS, RACFIND=NO, and DSTYPE=V and ENTITY (not ENTITYX) were specified
 - Offset X'08 (value X'80') indicates that ATTR=ALTER was specified
 - OFFSETxx (X'24') is the ENTITY name ("PAGE08.CATALOG")
 - OFFSETxx (X'28') is the CLASS name ("DATASET")
 - OFFSETxx (X'2C') is the VOLSER ("PAGE08")

```

Area length:                00000068

Area value:
00000000 00000000 00A40000 00010000 | .....u..... |
00000000 00000000 000069A4 00000000 | .....u..... |
00000000 00000068 00000000 00000000 | .....u..... |
00000000 00000000 00000000 00000000 | .....u..... |
00000000 00000000 00000000 00000000 | .....u..... |
00000000 00000000 00000000 00000000 | .....u..... |
00000000 00000000 00000000 00000000 | .....u..... |
00000000 00000000 00000000 00000000 | .....u..... |

Area length:                0000003C

Area value:
3C000000 9C000000 80000000 00000000 | .....u..... |
00000000 00000000 00000000 00000000 | .....u..... |
00000000 00006C48 00006C74 00006C7C | .....%...%...% |
00000000 00000000 00000000 00000000 | .....u..... |

1 Area length:                00000008

Area value:
D6C6C6E2 C5E30024 | OFFSET... |

Area length:                0000002C

Area value:
D7C1C7C5 F0F84BC3 C1E3C1D3 D6C74040 | PAGE08.CATALOG |
40404040 40404040 40404040 40404040 | .....u..... |
40404040 40404040 40404040 40404040 | .....u..... |

Area length:                00000008

Area value:
D6C6C6E2 C5E30028 | OFFSET... |

Area length:                00000008

Area value:
07C4C1E3 C1E2C5E3 | .DATASET |

Area length:                00000008

Area value:
D6C6C6E2 C5E3002C | OFFSET... |

Area length:                00000006

Area value:
D7C1C7C5 F0F8 | PAGE08 |
  
```

Step 7: Formatting and Reviewing the Trace...

- Let's look at the third of the last three REQUEST=AUTHs, which is for 'ERNIE.TOOLS.CNTL'
- The X'3C' length is the function-specific parameter list for RACROUTE REQUEST=AUTH
 - Offset X'04' (value X'8E') indicates RACFIND=NO, and DSTYPE is not V and LOG=NOSTAT was specified and ENTITY (not ENTITYX) was specified
 - Offset X'08 (value X'02') indicates that ATTR=READ was specified
 - OFFSETxx (X'24') is the ENTITY name ("ERNIE.TOOLS.CNTL")
 - OFFSETxx (X'28') is the CLASS name ("DATASET")
 - OFFSETxx (X'2C') is the VOLSER ("SMSVL1")

```

<SAP Plist omitted>

Area length:                0000003C

Area value:
3C000000  8E000000  02000000  00000000  | ..... |
00000000  00000000  00000000  00000000  | ..... |
00000000  009C4A58  009C4D30  009C403C  | ..... |
00000000  00000000  00000080  | ..... |

Area length:                00000008

Area value:
D6C6C6E2  C5E30024  | OFFSET.. |

Area length:                0000002C

Area value:
C5D9D5C9  C54BE3D6  D6D3E24B  C3D5E3D3  | ERNIE.TOOLS.CNTL |
40404040  40404040  40404040  40404040  | |
40404040  40404040  40404040  | |

Area length:                00000008

Area value:
D6C6C6E2  C5E30028  | OFFSET.. |

Area length:                00000008

Area value:
07C4C1E3  C1E2C5E3  | .DATASET |

Area length:                00000008

Area value:
D6C6C6E2  C5E3002C  | OFFSET.. |

Area length:                00000006

Area value:
E2D4E2E5  D3F1  | SMSVL1 |
  
```

Step 7: Formatting and Reviewing the Trace...

- The final REQUEST= in our job was a REQUEST=DEFINE The X'68' length is the function-specific parameter list for RACROUTE REQUEST=AUTH
 - Offset X'04' (value X'88') indicates that this is a TYPE=DELETE and DSTYPE is not V or M
 - OFFSETxx (X'0C') is the VOLSER name ("SMSVL1")
 - OFFSETxx (X'10') is the CLASS name ("DATASET")
 - OFFSETxx (X'34') is the ENTITY ("ERNIE.TOOLS.CNTL")

```

Area length:                00000068
Area value:
68000000  88000000  00000000  009C403C  | ...h..... |
009C4D30  00000000  00000000  00000000  | ..(..... |
00000000  00000080  00000000  00000000  | ..... |
00000000  009C4A58  00000000  00000000  | .....c..... |
00000000  00000000  00000000  00000000  | ..... |
00000000  00000000  00000000  00000000  | ..... |
00000000  00000000  00000000  00000000  | ..... |

Area length:                00000008
Area value:
D6C6C6E2  C5E3000C  | OFFSET.. |

Area length:                00000006
Area value:
E2D4E2E5  D3F1  | SMSVL1 |

Area length:                00000008
Area value:
D6C6C6E2  C5E30010  | OFFSET.. |

Area length:                00000008
Area value:
07C4C1E3  C1E2C5E3  | .DATASET |

Area length:                00000008
Area value:
D6C6C6E2  C5E30034  | OFFSET.. |

Area length:                0000002C
Area value:
C5D9D5C9  C54BE3D6  D6D3E24B  C3D5E3D3  | ERNIE.TOOLS.CNTL |
40404040  40404040  40404040  40404040  | |
40404040  40404040  40404040  | |

```

SAFTRACE Records for our Simple Batch Job

- Now we can fill in our table...

Event	REQSTOR	SUBSYS	Return and Reason Codes	Description/Comments
TOKENMAP(XM)	IEE0003D	CONSOLE	0/0/0	Not related to our investigation
EXTRACT(BR)	JBSTXTRT	JES2z204	0/0/0	Not related to our investigation
VERIFYX	JBSTTMAP	JES2z204	0/0/0	Not related to our investigation
AUTH			4/4/0	READ authority to FACILITY/IARRSM.LRGPAGES Not related to our investigation
AUTH			8/8/200	READ Authority to XFACILIT/STGADMIN.IGG.DELAUDIT.PAGE08.CATALOG
AUTH			8/8/0	ALTER Authority to ERNIE.TOOLS.CNTL with LOG=NOFAIL
AUTH			0/0/0	ALTER Authority to PAGE08.CATALOG with LOG=ASIS
AUTH			8/8/0	READ Authority to ERNIE.TOOLS.CNTL with LOG=NOSTAT
DEFINE			0/0/0	TYPE=DELETE for ERNIE.TOOLS.CNTL
VERIFY			0/0/0	Not related to our investigation

RACROUTE and Callable Service Numbers

Table 17. Callable services type numbers (continued)

IRRSIU00	1	1
IRRSU00	2	2
IRRSMF00	3	3
Reserved	4	4
IRRSMM00	5	5
IRRSKA00	6	6
IRRSKP00	7	7
IRRSUM00	8	8
IRRSGM00	9	9
IRRSYG00	A	10
IRRSU00	B	11
IRRSU00	C	12
IRRSYG00	D	13
IRRSYG00	E	14
IRRSYG00	F	15
IRRSYG00	10	16
IRRSYG00	11	17
IRRSYG00	12	18
IRRSYG00	13	19
IRRSYG00	14	20
IRRSYG00	15	21
IRRSYG00	16	22
IRRSYG00	17	23
IRRSYG00	18	24
IRRSYG00	19	25
IRRSYG00	1A	26
IRRSYG00	1B	27
IRRSYG00	1C	28
IRRSYG00	1D	29
IRRSYG00	1E	30
IRRSYG00	1F	31
IRRSYG00	20	32
IRRSYG00	21	33
IRRSYG00	22	34
IRRSYG00	23	35
IRRSYG00	24	36
IRRSYG00	25	37
IRRSYG00	26	38
IRRSYG00*	27	39
IRRSYG00	28	40
IRRSYG00	29	41

Table 17. Callable services type numbers (continued)

IRRSYG00	2A	42
IRRSYG00	2B	43
IRRSYG00	2C	44
IRRSYG00	2D	45
IRRSYG00	2E	46
IRRSYG00	2F	47
IRRSYG00	30	48
IRRSYG00	31	49
IRRSYG00	32	50
IRRSYG00	33	51
IRRSYG00	34	52
IRRSYG00	35	53

Note: Callable Service IRRSEQ00, R_Admin, has its own trace facility.

Table 18. RACROUTE CALL= service type number

RACROUTE CALL=	Service / Type Number in Hex	Service / Type Number in Decimal
AUTH	1	1
FASTAUTH	2	2
LIST	3	3
DEFINE	4	4
VERIFY	5	5
EXTRACT	6	6
DIRAUTH	7	7
TOKENMAP	8	8
VERIFYX	9	9
TOKENXTR	A	10
TOKENBLD	B	11
EXTRACT, BR=YES	C	12
AUDIT	D	13
STAT	E	14
SIGNON	F	15
TOKENMAP, XMEM	10	16
TOKENXTR, XMEM	11	17

A Few Last Points

- Due to nesting of some services PRE and POST trace records might not be in sequential order. For example, you might see two PRE calls and then two POST calls.
- Not all parameters appear in the trace. For example, passwords don't appear in the trace.
- When tracing z/OS UNIX System Services calls, you need to have a "*" in the jobname to catch any spawned address spaces.



Please submit your session feedback!

- Do it online at <http://conferences.gse.org.uk/2019/feedback/FK>
- This session is **FK**



1. What is your conference registration number?

This is the three digit number on the bottom of your delegate badge

2. Was the length of this presentation correct?

1 to 4 = "Too Short" 5 = "OK" 6-9 = "Too Long"

1 2 3 4 5 6 7 8 9

3. Did this presentation meet your requirements?

1 to 4 = "No" 5 = "OK" 6-9 = "Yes"

1 2 3 4 5 6 7 8 9

4. Was the session content what you expected?

1 to 4 = "No" 5 = "OK" 6-9 = "Yes"

1 2 3 4 5 6 7 8 9

SAFTRACE Demystified!

Mark Nelson, CISSP®, CSSLP®

IBM

November 2019

Session **FK**

