RACF Snoop

Lennie Dymoke-Bradshaw   
05 December 2021

Document Control

**Version Control:**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Date | Changes |
| 1.0 | Lennie Dymoke-Bradshaw | 2000.09.28 | Original |
| 1.1 | Lennie Dymoke-Bradshaw | 2008.03.10 |  |
| 1.2 | Lennie Dymoke-Bradshaw | 2021.12.05 | Re-designed doc for wider distribution |

Table of Contents

[1. Introduction 4](#_Toc90128309)

[1.1 Restrictions and potential enhancements 4](#_Toc90128310)

[2. Installing RACF Snoop 5](#_Toc90128311)

[3. RACF Snoop usage 8](#_Toc90128312)

[4. Appendix A - GENLIST, and GENLIST2 13](#_Toc90128313)

# Introduction

RACF Snoop is a small suite of programs which can be used to assist in the management of a RACF environment.

The author developed these programs initially to help with the diagnosis of performance problems with Generic profiles. However, the suite has now been extended and can perform other functions.

For example, it is able to answer the following questions about an MVS system.

* Which address spaces are currently using a particular user id?
* Which user id and group are being used by an address space?
* Which address spaces are running with system Special?
* Which address spaces have the TRUSTED or PRIVILEDGED attribute?
* What Generic profiles are being used in a specific address space?
* How many generic profiles exist under a given high-level qualifier?
* Which tasks in an address space have the “dirty-bit” set on?
* Which tasks in an address space are running APF authorised modules?
* Who is the submitter of a job?

The document draws on the author’s experience in working with RACF in an MVS environment since the late 1970s.

The intended audience for this document is MVS systems programmers and MVS security administrators. While some attempt is made to explain some of the control blocks in use the reader is expected to know many terms related to both MVS and RACF.

## Restrictions and potential enhancements

Since this utility was first written about 20 years have passed and RACF and MVS have changed. This means that there are some aspects that of the utility which are no longer complete. The primary two of these are,

* No support for 64bit in-storage profiles. Some address spaces store sets of generic profiles above the 2-gigabyte bar. These will not be shown. Instead the number of profiles is displayed as zero.
* No support for the extended GATE (Generic Anchor Table Entry) support. Since the program was written RACF has been enhanced to allow the number of GATEs to be specified per address space up to a maximum of 99. The RACF command SET GENERICANCHOR can be used to do this.

While RACF Snoop only supports the base 4 GATEs, the GENLIST command will actually display up to 8 GATEs. The reason for this is history[[1]](#footnote-1).

In addition, I think it would be useful to be able to show information about the USS state of each address space. So, for example, whether the address space id dubbed, what the current UID and GID settings are.

# Installing RACF Snoop

These instructions assume you have an installation file called RSP.INSTALL.XMI which is a copy of the install materials for RACF Snoop. This file is in NETDATA format; that is, it has been through the TRANSMIT (or XMIT) TSO command. It will need to be processed by the RECEIVE command before it can be used. When it has been received it must be further unloaded. See instructions below.

RACF Snoop requires two load modules to be placed into an APF-Authorised load library on the system linklist.

RACF Snoop also requires a couple of changes to your definitions in SYS1.PARMLIB, namely to the AUTHCMD section of IKJTSO00, and a CSVLLAxx member to refresh LLA once the programs are installed in the system linklist.

RACF Snoop can be protected by creating a profile in the FACILITY class called RSP.SNOOP. A user will need READ access to this profile to use RACF Snoop. It is recommended that you have the FACILITY class RACLISTed, or else the use of the authority checking process for RACF Snoop may interfere with some of the values it presents.

An alternative is to use the PROGRAM class. More details later.

* 1. Installation Steps

The following steps can be used to install RACF Snoop.

* + 1. Overview

The installation file for RACF Snoop was created by using TSO TRANSMIT to process multiple files into a PDS. Then the PDS was itself processed through TRANSMIT.

You should be starting with a data set which has been FTPed or otherwise stored onto your z/OS system as a flat file in TSO XMIT format. I will assume this has the following name, where xxxxxx is your current userid.

xxxxxx.RSP.INSTALL.XMI

These instructions should take you to a point where RACF Snoop works. It is then your responsibility to organise the libraries and members where it is most appropriate for your installation.

At the end of this process

* + 1. Detailed Instructions

|  |  |
| --- | --- |
| 1 | You will initially have a file on your PC called  RACF Snoop.zip  When this is unzipped, it will contain various documents (including this one) and a file called  rsp.install.xmi. |
| 2 | Use file transfer to move rsp.install.xmi to your mainframe. When doing this the following items are essential.  Firstly, ensure it is transferred using a BINARY transfer.  Secondly ensure it is transferred to a file with a LRECL of 80 bytes which has a RECFM of FB. |
| 3 | I will assume the file is stored on the mainframe as  userid.RSP.INSTALL.XMI  Issue the TSO command  RECEIVE INDA(RSP.INSTALL.XMI)  Follow the instructions to store the file  userid.RSP.INSTALL |
| 4 | This install data set contains 5 members. Four of them are the XMIT formatted versions of the four libraries you will need for this product.  The fifth member is called #INSTALL.  This is a REXX exec. Execute this to copy the XMIT files from the INSTALL library to your disk. |
| 5 | After the previous step you should have the following data sets on your disk.  userid.RSP.ASM.XMI  userid.RSP.CLIST.XMI  userid.RSP.CNTL.XMI  userid.RSP.INSTALL  userid.RSP.ISPPLIB.XMI  Issue the following commands to create the target libraries.  RECEIVE INDA(RSP.ASM.XMI)  RECEIVE INDA(RSP.CLIST.XMI)  RECEIVE INDA(RSP.CNTL.XMI)  RECEIVE INDA(RSP.ISPPLIB.XMI) |
| 6 | At this point you can delete the XMI data sets,  userid.RSP.ASM.XMI  userid.RSP.CLIST.XMI  userid.RSP.CNTL.XMI  userid.RSP.ISPPLIB.XMI  userid.TSP.INSTALL.XMI |
| 7 | In the data set  Userid.RSP.CNTL  Examine the member called #README and determine if you need to rename the members. This will only apply if you have a system at level z/OS 1.4 or older.  You will find a job called ASSEMBLE.  Set the target load library for the two load modules. This should be an APF authorised library on your system linklist.  Run the ASSEMBLE job and ensure it end successfully and stores the two modules in the library you have chosen marked with AC=1. |
| 8 | Once the modules are assembled refresh LLA. You can use the command,  F LLA,REFRESH |
| 9 | You will need to add the two modules GENLIST and GENLIST2 to the AUTHCMD table in your IKJTSOxx member.  The IKJTSOxx member can be made active with the TSO command PARMLIB. |
| 10 | Security  In order to control access to RACF Snoop you may create a profile in the FACILITY class called RSP.SNOOP.  I recommend that you create this profile as a generic profile and then have the FACILITY class RACLISTed using the following TSO RACF commands,  RDEFINE FACILITY RSP.SNOOP GENERIC UACC(N) -OWNER(SYS1)  PERMIT RSP.SNOOP GENERIC CLASS(FACILITY) -ID(userid/group)  SETROPTS RACLIST(FACILITY)  If the FACILITY class is not RACLISTed then each time that one of the RACF Snoop modules performs a check it may change the state of the generic profiles within the running address space (i.e. your TSO userid). If this is not a problem then you may miss out the last step above.  An alternative is to create the profile RSP.SNOOP as a discrete profile. This will avoid all interference with generic profile reporting, but with have a very minor performance hit.  If the profile RSP.SNOOP is not created then blanket access is granted to RACF Snoop.  It is also possible to create PROGRAM class profiles to protect RACF Snoop. If you do this then the modules to be protected are GENLIST and GENLIST2. However, even if you do this, the check for RSP.SNOOP will still be issued, and so the potential to interfere with generic profile processing within the current address space still exists. |

Once these steps are complete you may invoke RACF Snoop from any ISPF panel using the following command,

EXEC ‘hlq.RSP.CLIST(RSP)’

where ‘hlq’ is the high-level qualifier you chose in step 4 above.

# RACF Snoop usage

Once installed RACF Snoop can be accessed using the following command,

EXEC ‘hlq.RSP.CLIST(RSP)’

where ‘hlq’ is the high-level qualifier you chose during installation.

This should display the following panel

-------------------------------- RACF Snoop ----------------------------------

Command ===> Scroll ==> PAGE

A/S name mask: \* STC A/S: ATX A/S: NoACEE A/S:

Userid mask: \* SYS A/S: INIT A/S: ALL A/S:

Max A/S: 500 TSO A/S: Y Batch A/S: REFRESH?: Y

A/S Name Type User GATE-1 GATE-2 GATE-3 GATE-4

Asid Attr Group

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Bottom of data \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. RACF Snoop entry panel

The entry panel is pre-filled with values which will select TSO address spaces up to a maximum of 500. If you press <Enter> at this point then the lower part of the panel will be populated with information from TSO address spaces. This is a table which can be scrolled using the normal ISPF scroll commands.

A typical set of TSO users is shown in figure 2

-------------------------------- RACF Snoop ---------------------- Row 1 of 14

Command ===> Scroll ==> PAGE

A/S name mask: \* STC A/S: ATX A/S: NoACEE A/S:

Userid mask: \* SYS A/S: INIT A/S: ALL A/S:

Max A/S: 500 TSO A/S: Y Batch A/S: REFRESH?: Y

A/S Name Type User GATE-1 GATE-2 GATE-3 GATE-4

Asid Attr Group

USER1 TSO USER1 D-CATALOG D-ISP D-TS D-SYS1

0058 ESASSIST 3 1 24 134

USER2 TSO USER2 D-USER2 D-ISP D-IMSIVP D-SYS1

0063 ESASSIST 4 1 1 134

**USER3 TSO .swapout**

**006E .swapout**

USER4 TSO .swapout

0070 .swapout

1. RACF Snoop entry panel

You may alter the options at the top and a table of address spaces will be shown below.

The options are pre-set to display only TSO address spaces. However, you may select from the following classifications of address spaces:-

1. TSO
2. Started Task
3. System
4. APPC
5. Initiators
6. Batch

In addition, you may select address spaces by address space name, or you may select address spaces, by the userid in use in the address space. You may make use of generic characters in the address space name and userid fields. See the help panels for further details.

To demonstrate what can be seen, let us assume you just press enter at the initial panel. This will build a table of TSO address space, similar to that shown in figure 5.

-------------------------------- RACF Snoop ---------------------- Row 1 of 11

Command ===> Scroll ==> PAGE

A/S name mask: \* STC A/S: ATX A/S: NoACEE A/S:

Userid mask: \* SYS A/S: INIT A/S: ALL A/S:

Max A/S: 500 TSO A/S: Y Batch A/S: REFRESH?: Y

A/S Name Type User GATE-1 GATE-2 GATE-3 GATE-4

Asid Attr Group

USER1 TSO USER1 D-USER1 D-SYS1 D-OPCESA D-DRL150

006F SWARE 1 134 11 1

USER2 TSO USER2 D-TS D-SYS1 D-DRL140 D-DRL

0070 A CS 24 134 1 1

DEREK TSO DEREK D-SYS1 D-CATALOG D-ISP D-TS

0092 CS 134 3 1 24

USER3 TSO USER3 D-USER3 D-ISP D-TS D-SYS1

00A2 CS 5 1 24 134

USER4 TSO USER4 D-OPCESA D-CATALOG D-ISP D-TS

00B1 ESASSIST 11 3 1 24

1. List of TSO address spaces

On the left of the panel is shown the address space name with the address space id (known as the ASID) shown below it. The next column shows the type of address space and under it a field showing security attributes for the address space.

The address space type will be one of the following,

|  |  |
| --- | --- |
| SYS | System address space |
| STC | Started task |
| TSO | Time sharing user |
| BAT | Batch address space |
| ATX | APPC address space |
| INIT | Idle Initiator, including APPC initiators and USS initiators |

The “Attr” field shows what special attributes are in use for this user. Thus it is possible to see if the user has the SPECIAL attribute (S), the operations attribute (O), the auditor attribute (A), or if the address space has been marked as privileged (P), or trusted (T). Any combinations of these letters may be shown.

The third column shows the address space Racf userid with the address of the ASCB underneath it.

The fourth column shows the active Racf group for the address space.

To the right are the four columns marked GATE-1 to GATE-4. These are the four generic access tables. Each line is formatted as one of the following:-

D-HHHHHHHH

where ‘HHHHHHHH’ represents a dataset high-level qualifier.

Or

G-RRRRRRRR

where ‘RRRRRRRR’ represents a general resource class name.

Under each of these entries is shown the number of profile names in storage for each of these GATE entries.

(For further details of this information see the author’s document on Tuning Racf Generic profiles.)

Each of the lines may be selected with a T or and S

-------------------------------- RACF Snoop ---------------------- Row 1 of 10

Command ===> Scroll ==> PAGE

A/S name: USER1 Asid: 00A2 Type: TSO ACEE Addr: 009FD810

User: USER1 User Name: \*\*\* TSO USER \*\*\* ACEE Date: 20.10.2000

Group: CS Seclabel: .nosecl.

Submittor: .nosusr. Sgroup: .nosgrp. from: .nosnod. Surrogat: .nosuro.

POE: PYDT6P03 Node: PYDT6P03 appl: .noappl.

Attribs: NOSPECIAL NOOPER NOAUDITOR NOPRIV NOTRUST NOUAUDIT NOADSP NOPROT

TCBaddr Seclbl User GATE-1 GATE-2 GATE-3 GATE-4

ACEEaddr Attr Group

009FE240

00000000

009FF1D8

00000000

009FDE88

00000000

009D4C98

00000000

1. Details of one address space using the ‘S’ line command

-------------------------------- RACF Snoop ---------------------- Row 1 of 10

Command ===> Scroll ==> PAGE

A/S name: USER1 Asid: 00A2 Type: TSO ACEE Addr: 009FD810

User: USER1 User Name: \*\*\* TSO USER \*\*\* ACEE Date: 20.10.2000

Group: CS Seclabel: .nosecl.

Submittor: .nosusr. Sgroup: .nosgrp. from: .nosnod. Surrogat: .nosuro.

POE: PYDT6P03 Node: PYDT6P03 appl: .noappl.

Attribs: NOSPECIAL NOOPER NOAUDITOR NOPRIV NOTRUST NOUAUDIT NOADSP NOPROT

TCBaddr JSTCB OwnerTCB nextTCB ACEE 1st PGM Flags

009FE240 009FE240 00000000 009FF1D8 00000000 IEAVAR00 S

009FF1D8 009FF1D8 009FE240 009FDE88 00000000 IEAVTSDT S

009FDE88 009FDE88 009FE240 009D4C98 00000000 IEESB605 S

009D4C98 009D4C98 009FDE88 009FD980 00000000 IKJEFT01 DAS

009FD980 009D4C98 009D4C98 009CE3B0 00000000 IKJEFT02 DS

009CE3B0 009D4C98 009FD980 009CE218 00000000 IKJEFT09 DAS

009CE218 009D4C98 009CE3B0 009CEE88 00000000 PDF DS

009CEE88 009D4C98 009CE218 009CEA98 00000000 ISPTASK DS

009CEA98 009D4C98 009D4C98 009D4490 00000000 IKJEFT02 DS

009D4490 009D4C98 009CEA98 00000000 00000000 GENLIST2 AS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Bottom of data \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Details of one address space using the ‘T’ line command

# Appendix A - GENLIST, and GENLIST2

The RACF Snoop package consists of some CLISTs, some ISPF panels, and two TSO commands. These two commands can be run outside of the panels if wish. Alternatively, they may be used to produce other monitoring displays.

GENLIST

The syntax of GENLIST is:-

GENLIST -  
 JOBMASK(jobmask) USER(usermask)-  
 TSO STC BATCH ATX INIT SYS ALL -  
 NOACEE -  
 ESTAE | NOESTAE  
 ABENDMSG

This command lists security information from other address spaces. The meanings of the parameters are as follows:-

JOBMASK(jobmask)

1. The jobmask may be a fully specified address space name, or may be a mask used to select multiple address spaces. The characters \* and % may be used in the mask, and they have the same meanings as they normally do in RACF generic profiles. The \* may only be used as the last character in the jobmask, and represents any number of following characters up to the maximum of 8. A % matches any single character in the specified position. If the ‘jobmask’ string matches the name for the address space then the address space is selected.

USERMASK(usermask)

1. The usermask may be a fully specified userid, or may be a mask used to select multiple userids. The characters \* and % may be used in the mask, and they have the same meanings as they normally do in RACF generic profiles. The \* may only be used as the last character in the usermask, and represents any number of following characters up to the maximum of 8. A % matches any single character in the specified position. If the ‘usermask’ string matches the userid for the address space then the address space is selected.

TSO

1. If this keyword is specified then TSO address spaces will be selected for display.

STC

1. If this keyword is specified then started task address spaces will be selected for display.

BATCH

1. If this keyword is specified then batch address spaces will be selected for display.

ATX

1. If this keyword is specified then APPC address spaces will be selected for display.

INIT

1. If this keyword is specified then idle initiator address spaces will be selected for display. Note that you will probably need to specify NOACEE in order to display these address spaces.

SYS

1. If this keyword is specified then system address spaces will be selected for display.

ALL

1. This keyword is equivalent to specifying TSO STC BATCH ATX INIT ALL.

NOACEE

1. The normal use of this command is to display information about address spaces and the GATEs within them. If an address space has no ACEE (ASXBSENV is zero) then it will not normally be selected for display. If this parameter is specified then these address spaces will be included, subject to the selection rules specified.

ESTAE | NOESTAE

1. During normal operation some address space may be inaccessible, due to swapping operations. It is normal for occasional abends to occur during operation. Hence the normal setting for this parameter is ESTAE. Only use NOESTAE as directed by the systems programmer in the diagnosis of problems with the command.

ABENDMSG

1. This parameter will force GENLIST to issue messages for all abends, including those from which it recovers successfully. It would normally only be used in problem diagnosis.

The report produced by GENLIST will consist of multiple sets of lines similar to the following:-

Start TSO DYMOKEL Asid 0033 Ascb 00FB5B00  
Userid DYMOKEL Group SYSPROG SO  
Gate: dsn-hlq DYMOKEL 00000003  
Gate: dsn-hlq CATALOG 00000001  
END TSO DYMOKEL Asid 0033 Ascb 00FB5B00

The number of “Gate:” lines may vary between 0 and 8.

GENLIST2

The synax of GENLIST is:-

GENLIST2 -   
 ASID(nnnn)   
 ADDRNAME(nnnnnnnn)   
 ESTAE | NOESTAE   
 ABENDMSG   
 GATES | NOGATES

This command lists security information from other address spaces. The meanings of the parameters are as follows:-

ASID(nnnn)

1. This is the ASID of the address space to be displayed.

ADDRNAME(nnnnnnnn)

1. The address space name corresponding to the ASID. If this is not specified it is not checked. If t is specified then it will be checked and if found incorrect message RSP0005W will be issued, warning of the mismatch.

ESTAE | NOESTAE

1. during normal operation some address space may be inaccessible, due to swapping operations. It is normal for occasional abends to occur during operation. Hence the normal setting for this parameter is ESTAE. Only use NOESTAE as directed by the systems programmer in the diagnosis of problems with the command.

ABENDMSG

1. This parameter will force GENLIST to issue messages for all abends, including those from which it recovers successfully. It would normally only be used in problem diagnosis.

The report produced by GENLIST2 will consist of multiple sets of lines similar to the following:-

Command: GENLIST2 ASID(0098) ADDRNAME(DYMOKEL)

START DYMOKEL 0098 TSO 00F68180 008FD810  
USER DYMOKEL CS .nosecl. 30.01.2001  
NAME L DYMOKE-BRADSHAW  
ATTR NOSPECIAL NOOPER NOAUDITOR NOPRIV NOTRUST NOUAUDIT NOADSP  
PORT PYDT6P02 PYDT6P02 .noappl.  
SURROGAT .nosuro.  
SUBMITTOR .nosusr. .nosgrp. .nosnod.  
TCB 008FE240 008FE240 00000000 008FF1D8 00000000 IEAVAR00 S  
TCB 008FF1D8 008FF1D8 008FE240 008FDE88 00000000 IEAVTSDT S  
TCB 008FDE88 008FDE88 008FE240 008D4C48 00000000 IEESB605 S  
TCB 008D4C48 008D4C48 008FDE88 008FD980 00000000 IKJEFT01 DAS  
TCB 008FD980 008D4C48 008D4C48 008CEC70 00000000 IKJEFT02 DS  
TCB 008CEC70 008D4C48 008FD980 008CEAD8 00000000 IKJEFT09 DAS  
TCB 008CEAD8 008D4C48 008CEC70 008CE420 00000000 PDF DS  
TCB 008CE420 008D4C48 008CEAD8 008CE080 00000000 ISPTASK DS  
TCB 008CE080 008D4C48 008CEAD8 008D43F0 00000000 ISPTASK DS  
TCB 008D43F0 008D4C48 008D4C48 008D4098 00000000 IKJEFT02 DS  
TCB 008D4098 008D4C48 008D43F0 00000000 00000000 GENLIST2 AS  
END DYMOKEL 0098 00F68180 008FD810

---------------------------------------- End of Document ----------------------------------------

1. A customer had a zap from IBM to increase the number of GATEs to 8, and was using the GENLIST command standalone to monitor activity on those GATEs. [↑](#footnote-ref-1)