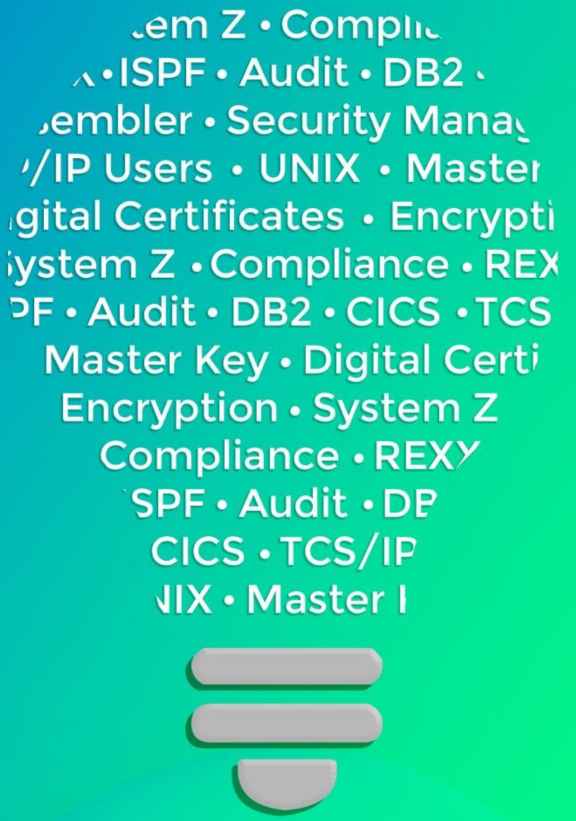
VANGUARD 2023
Security & Compliance

Database Adnalesque Cano: I Sing of a Database and its Records
A Peek Inside the RACF®, Database

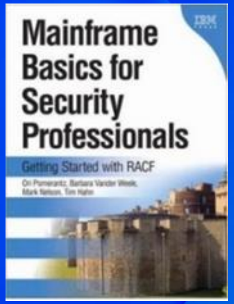
Mark Nelson, CISSP®, CSSLP®,

Senior Technical Staff Member

IBM®, Poughkeepsie


System Z • Complia
X • ISPF • Audit • DB2 •
Assembler • Security Manag
TCP/IP Users • UNIX • Master
Digital Certificates • Encrypti
System Z • Compliance • REX
ISPF • Audit • DB2 • CICS • TCS
Master Key • Digital Certi
Encryption • System Z
Compliance • REXX
ISPF • Audit • DB
CICS • TCS/IP
UNIX • Master K

KNOWLEDGE
is your best defense


Mainframe Basics for Security Professionals
Getting Started with RACF
Ori Pomerantz, Barbara Vander Weele, Mark Nelson, Tim Hahn
IBM Press

Roadmap

- **What is the RACF data base?**
  - What is in it?
  - What's not in it?
  - Required characteristics for a RACF data set

- **RACF Data Set Internals**
  - Data blocks, index blocks, BAM blocks, empty blocks
  - Finding free space in a RACF data set
  - Tying it all together with Inventory Control Block (ICB)

- **The RACF Data Set Utilities**
  - IRRUT200
  - IRRUT400
  - IRRMIN00

What is the RACF Data Base?

- **The RACF data base is the persistent storage location for the vast majority of RACF’s operational and control information.**
- **The RACF data base can consist of a single data set or it can be split across multiple data sets.**
- **Starting with z/OS®, V2.5, RACF offers two options for defining RACF data sets:**
  - “Classic” or “non-VSAM” RACF data sets, where RACF acts as its own “access method”
  - “VSAM” RACF data sets, where IBM’s virtual storage access method (VSAM) is used as the “access method”
      - VSAM allows you to encrypt your RACF data set.

  - **The RACF data base can be configured with an on-line back-up which can be “switched to” to allow continuous operations in the event of data base problems.**

RACF Data Base as the Store of Virtually All RACF Information

- **The RACF Data Base Contains several different types of information:**
  - User, group, data set, and other resource definitions (profiles)
  - System settings (SETROPTS) such as which classes are active/RACLISTed/etc., password options
  - Index information to quickly locate profiles
  - Meta data (templates) that define all of the fields within profiles
  - Control information for the RACF data base

RACF Data Base as the Store of Virtually All RACF Information…

- **Items not Within the RACF Data Base**
  - Location information about the primary RACF data base, which can be found in ICHRDSNT (data set names table), the PARMLIB concatenation (starting in V2.3 (yea!)), or MSTRJCL (ugh!)
  - Location information on the back-up RACF data base (DSNT, PARMLIB)
  - System-level data base options (number of buffers, recording of statistics, sysplex communications mode, data sharing mode)
  - Definitions of “static” classes (static class descriptor table, one for IBM, one for clients)
  - RRSF network topology
  - RRSF parameter library
  - Exits
  - Naming Contentions Table (*not* recommended)

RACF Data Base as the Store of Virtually All RACF Information…

- **Characteristics of a RACF Data Set - “Classic” or “non-VSAM”:**
  - Physical sequential, un-moveable (DSORG=PSU)
  - A single contiguous extent
  - Logical record length of 4096
  - Block size of 4096
  - Cataloged
  - Properly shared amongst all of the systems
    - Must be on a device defined as shared on all systems

- ***RACF is critically dependent on these characteristics!***

RACF Data Base as the Store of Virtually All RACF Information…

- **Characteristics of a RACF Data Set - “VSAM”:**
  - LINEAR format
  - NONSPANNED
  - CISZ(4096)
  - SHAREOPTIONS(3,3)
  - REUSE
  - ERASE (recommended)
  - FREESPACE(0 0)
  - CYL(nnn 0)
  - Cataloged
  - Properly shared amongst all of the systems
    - Must be on a device defined as shared on all systems

- ***RACF is critically dependent on these characteristics!***

RACF Data Base as the Store of Virtually All RACF Information…

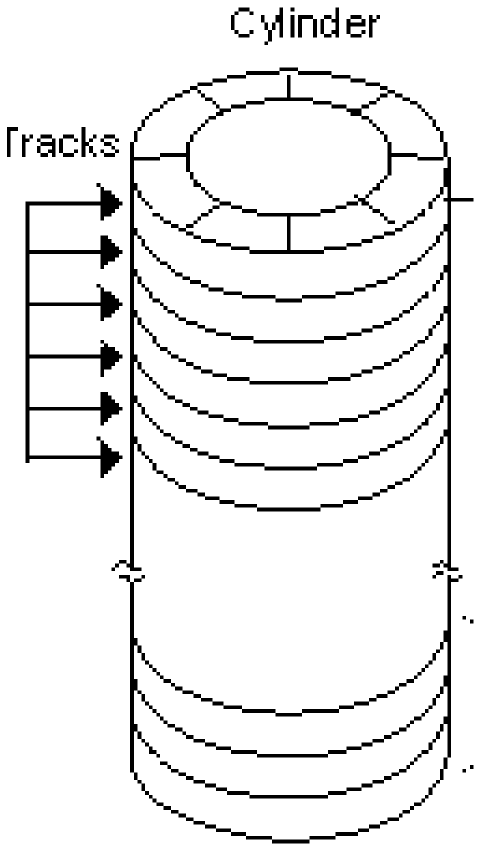
- **All of the information in the RACF data base is located by RACF using the relative byte address (RBA) of the data**
  - The RBA is the offset of the information from the RBA 0, the start of the data set
  - To retrieve or update a profile:
    - Non-VSAM: RACF takes the starting CCHHRR of the RACF data set and adds to it an RBA to determine the CCHHRR of the desired information
    - VSAM: RACF converts the RBA to a record number by dividing the RBA by 4096 and adding 1.
  - Certain RBAs are "defined by the RACF architecture":

RBA	Description
X'000000000000' through X'000000000FFF' '	The "master RACF control record, the RACF Inventory Control Block (**ICB**)
X'000000001000' through X'00000000AFFF"	RACF Data base templates
X'00000000B000' through X'00000000BFFF'	The Segment table
X'00000000C000' through X'000nnnnnFFF'	BAM blocks. The number of BAM blocks depends on the size of the RACF data base. More on these later.

  - All other data blocks in a RACF data set can be located at any RBA

RACF Data Base as the Store of Virtually All RACF Information…

- **For Data on a disk drive is addressed by its cylinder, head (track), and record (CCHHRR)**
  - For non-VSAM RACF data sets, RACF translates the RBA into the CCHHRR
  - For VSAM data sets, RACF translates the RBA into a record number and VSAM figures out the CCHHRR
- **Each device type (such as 3380, 3390) has its own device characteristics, such as:**
  - The number of cylinders (depends on model as well)
  - The number of tracks per cylinder
  - The maximum number of bytes per track
- **The number of records per track depends on the size of the record**
- **None of this is "real" anymore…. It's all virtualized**



RACF Data Base as the Store of Virtually All RACF Information…

- **When RACF attempts to retrieve a profile in the RACF data base, it must first find the RBA of the profile**
  - Some calls to RACF specify the specific profile
  - Other calls to RACF require RACF to find the “best matching profile”

- **RACF has a high-performance index which allows RACF to quickly find the RBA of any profile in the RACF data set or the best matching profile so that the data in the profile can be read**
  - If the RACF data base is split into multiple data sets, the RACF Range Table (ICHRRNG) determines which data set is used
  - Each RACF data set has its own index structure for the profiles that it contains
  - The index blocks point to data blocks which contain the actual profile data
  - The ICB ties the index components together
    - For some ICB values, the values in the “Master ICB” (the first data set) is used

RACF Can be Configured with a Back-up RACF Data Base

- **RACF supports the definition of a back-up RACF data base**
  - Defined in the RACF data set names table
  - Installations have controls on what level of statistical information is backed up

- **The back-up RACF data base is a "*logical*" copy of the primary data base as it is written in the same manner as the primary RACF data after a successful update to the primary RACF data base**
  - *RACF goes through the same mechanisms to locate the profiles on the back-up data base as it does primary data base after a successful update to the primary RACF data base*

- **Note that while the algorithm is the same, the contents of the RACF data base may not be at the same location in the data set as in the primary RACF database**

Considerations for Backing-Up the RACF Database

- **RACF should run with a primary and back-up data base**
  - RVARY SWITCH can be used to move to the back-up data base in the event of a *physical error* to the primary RACF data base or a logical error in the RACF primary that has not replicated in the back-up RACF database (rare!)
- **The RACF address space should be enabled to allow the issuing of RACF commands from the MVS console**

Considerations for Backing-Up the RACF Database…

- **A copy should be made of the RACF data base (using only IRRUT200 or IRRUT400) and that copy should be duplicated and these copies should be available in the event of a logical error in the RACF database**
  - A logical error would be the deletion of one or more operationally significant profiles or the setting of a system option which cannot be fixed using RACF commands from either a logged on user or as a RACF operator command
  - ***Protect all copies of the RACF data base with the same level of protections that you provide to the live RACF data base***
  - ***Ensure that these data sets have the correct attributes***
    - *See the non-VSAM and VSAM attributes specified earlier*
  - These copies should remain available on the system as they can be brought into the environment by the sequence of commands sometimes called the “RVARY dance”

The RACF Data Base Can be Split Among Data Sets

- **The RACF Data base can be spread across up to 90 data sets**
  - Profiles are distributed among the data sets based on:
    - Users, groups, data sets: The name of the profile
    - General Resources: Class name and profile name

- **The data sets (and their backups) are defined in the RACF Data Set Names Table (ICHRDSNT) or starting in z/OS V2R3, PARMLIB member IRRPRMxx (no assemble or linkage editor needed!)**

- **The allocation of profiles to the RACF data sets is defined by the RACF Data Set Range table (ICHRRNG) or starting in z/OS V2R3, PARMLIB member IRRPRMxx**

- **The back-up RACF database must have the exact same number of data sets as the primary and the ranges will be identical**

The RACF Data Base Can be Split Among Data Sets…

- **The first data set in the split is the “master” RACF DB, from which the ICB and other control information is managed**
  - Each RACF data set has its own ICB with control information about that data set

The RACF Data Base Can be Split Among Data Sets…

- **The first data set in the split is the “master” RACF DB, from which the ICB and other control information is managed**
  - Each RACF data set has its own ICB with control information about that data set

- **The need for a split RACF data base has declined substantially since the split RACF data set was introduced in the early 80s:**
  - Enhancements to system to DASD connectivity, pathing, and caching
  - Improvements in RACF caching, restructuring of RACF database in 1990, RACF data sharing

- **Recommendations:**
  - If you are not already using a split RACF data set, stay with a single RACF data set
  - Consider the user of RACF Data Sharing which can yield performance benefits across a parallel sysplex

RACF Serialization

- **For both non-VSAM and VSAM RACF data sets, RACF provides its own serialization mechanisms, such as:**
  - Serialization bit in the RACF inventory control block (ICB)
    - Controlled by IRRUT400/IRRDBU00 “LOCKINPUT”/”UNLOCKINPUT”
  - In non-RACF datasharing mode, RESERVEs against the device containing the RACF data set(s)
    - Recommendation: Convert to global (“SCOPE=SYSTEMS”) ENQs **if and only if the sharing systems are in the same GRS configuration**
  - SYSZRACF, SYSZRACn ENQs
    - Documented in RACF Systems Programmers Guide

- ***Note: No SYSZDSN ENQ for the RACF data set(s) is held*** *during RACF identification/authentication, authorization check, or logging functions*

- **Utilities which access the RACF data base must honor RACF’s serialization**

- **STRONG Recommendation: Use only the RACF utilities for copying a live RACF data base**
  - ***If you have created an unserialized copy, you might not find out that you have a problem until you are attempting to use the copy and it fails!***

Switching to a Back-up RACF Data Set

- **Switching to the backup RACF Data Base is done with the RVARY SWITCH TSO command**
  - RVARY deactivates the primary and makes it the back-up and makes the back-up the active primary
  - If the RACF data base is shared among systems, the RVARY SWITCH command must be issued on each system
    - *Unless* RACF Sysplex Communication is enabled (***Recommended***)
  - RACF goes through its allocation processing when a data set is RVARYed into an active state
    - This can be used (with extreme caution) to bring a completely new RACF data set into the environment

- **If RACF detects an I/O error on a data set in the primary RACF data base, it can automatically initiate an RVARY SWITCH for that data set**
  - Can only be done if RACF is assured that the device will not be available for future I/O, such as the device being varied offline or "boxed"

RACF Data Set Internals

- **There are five types of blocks in a RACF data set:**
  - **Data blocks:** Contain the actual data for every profile (user, group data set, general resource)
  - **Index blocks:** Define the index structure for the contents of the RACF database
    - **Level 1 Index Block ("Sequence set"):** An alphabetical listing of all profiles in the RACF data set and the associated RBA of the data block
    - **Level 2-9 Index Blocks**: Multilevel index set to locate the RBA of the next lower index block in the chain that leads to the L1 index block
  - **Segment table blocks:** The segment table block contains mappings of individual segments from within a template. These describe the segments associated with the profiles.
  - **Byte allocation mask (BAM) blocks:** Describes the availability of the free space within a RACF database.
  - **Empty blocks:** Blocks which contain no data and are available for use

The RACF Database Internals…

- **Each type of block has a one-byte identifier at the front of the block:**
  - **Data blocks:** X’83’
  - **Index blocks:** X’8A’
  - **Segment table blocks:** X’02’
  - **Byte allocation mask blocks:** X’00’
  - **Empty blocks:** X’C0’

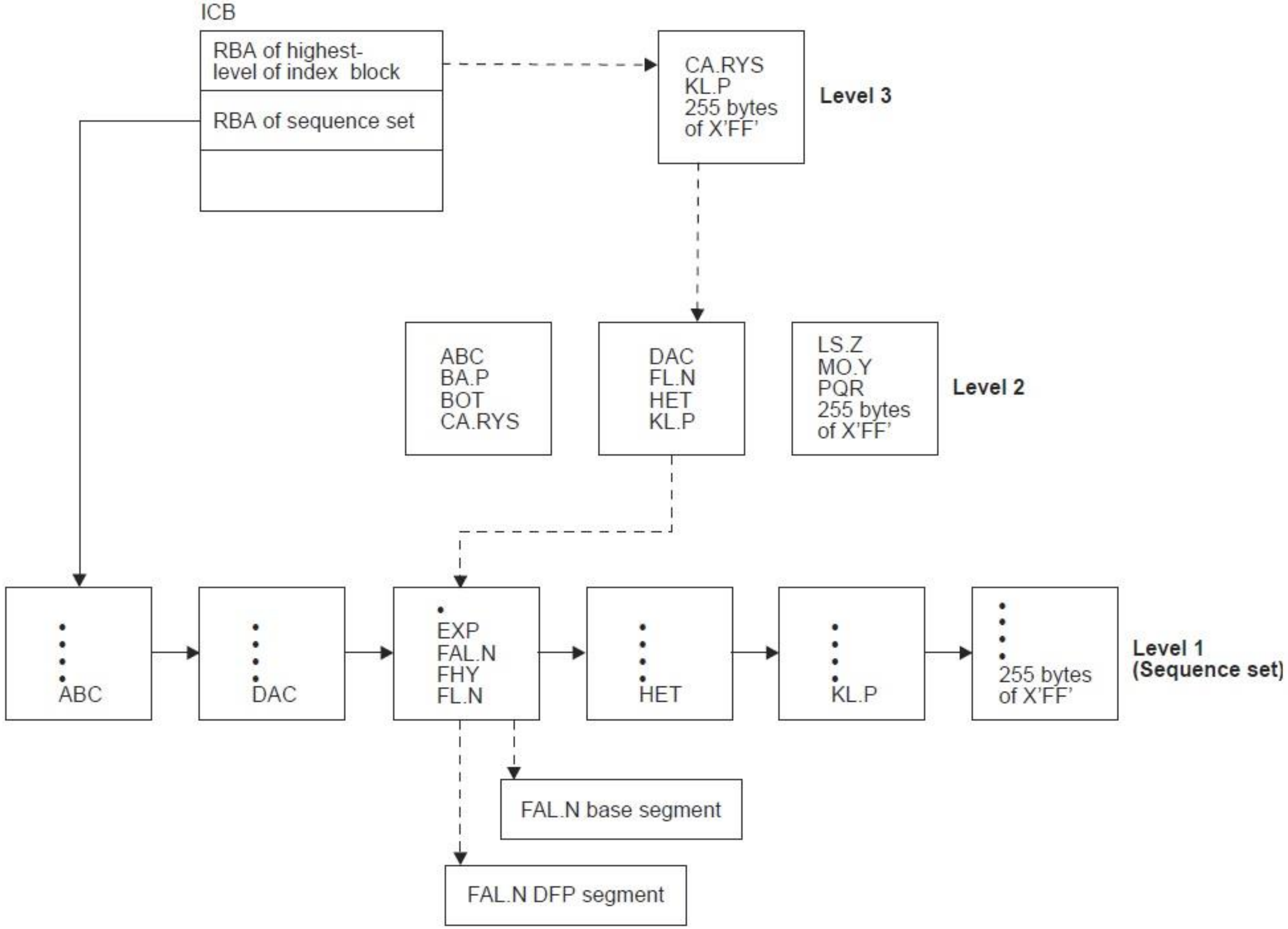
RACF Data Set Internals: Index Entries

- **Index entries for users, groups, and data sets are just the profile name**
- **Index entries for general resource profiles are prefixed with the class name and a dash (“-”) followed by the profile name**
  - Class names less than eight (8) characters are padded on the right with blanks
- **Index blocks “compress” the index entries by omitting leading characters from the index entry as compared to the first index entry**
- **Prior to the restructuring of the RACF database in RACF 1.9 (1990), RACF performed ‘back-end’ compression for common last qualifiers (.CNTL, .LOAD etc.)**

```
OFFSET   COMP.                          ENTRY NAME                                           RBA             BAM
         COUNT                                                                                        BLOCK  BYTE BIT
 00E     0000  DIGTCERT-01                                                             00000000E000   00    030   0
 02C     0009  DIGTCERT-326                                                            00000001E000   00    050   0
 042     0004  DIGTRING-CERTOWNR.RING00007                                             000000017000   00    042   0
```

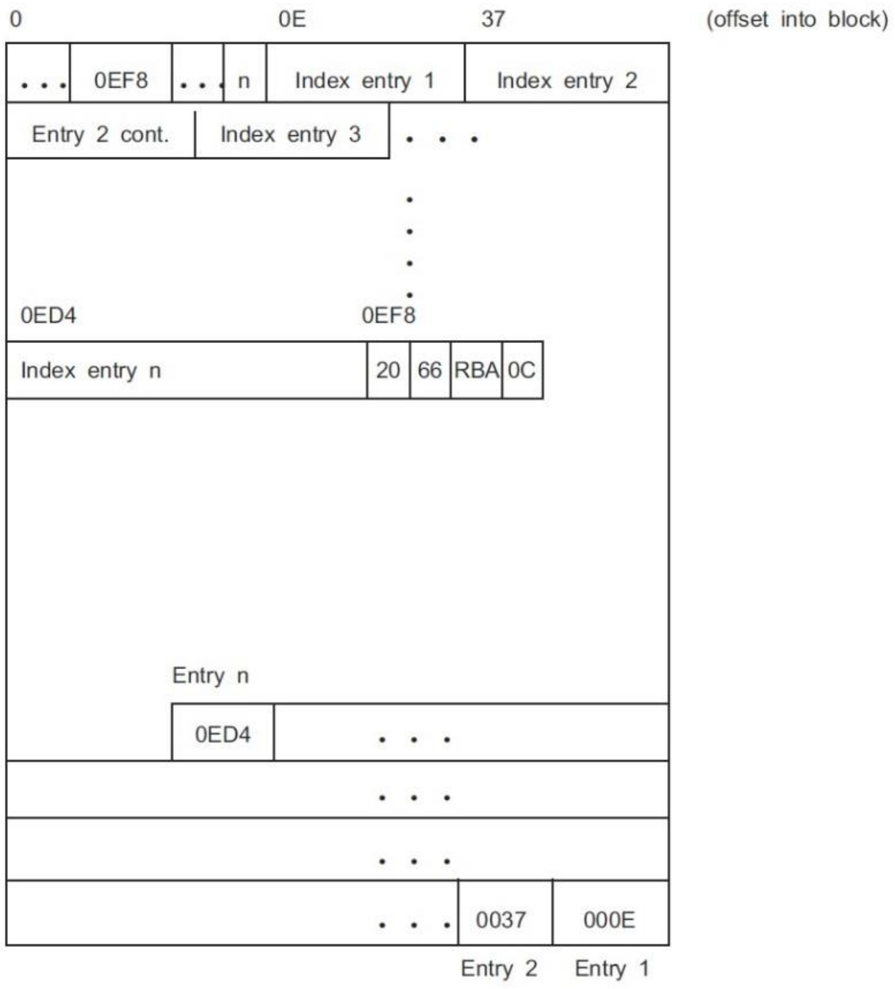
The RACF Data Set Internals: Index

- **When searching for a profile (example: “FAL.N”) , RACF searches the top level index block to find the first entry which is equal to or greater.**
- **The process is repeated in each of the blocks down to the index block.**
- **The index block is searched for the first matching entry, which contains the RBA of the data block**
- **Adding profiles results in additions to the L1 index and may result in additions to higher level blocks**



The RACF Data Set Internals: The Level 1 Index (Sequence Set)

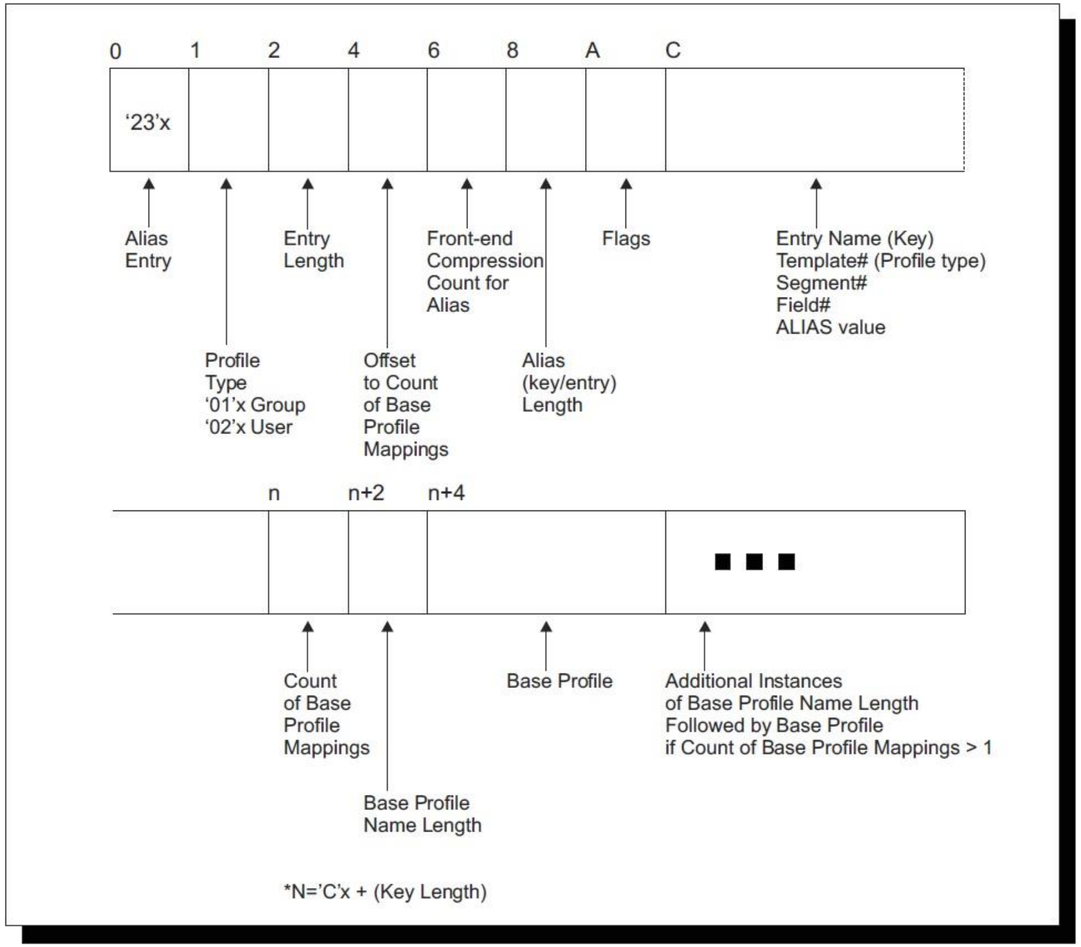
- **Contains a header, a list of index entries, a pointer to the next L1 index entry, and at the end a table of two byte offsets to allow for the quick location of entries or the first free space in the block**



The RACF Data Set Internals: Alias Index

- **RACF also supports an alias index structure for selected fields in the RACF data base, such as OMVS UID and GID**

- **The Alias Index is conceptually similar to the “regular” index, but its index blocks are different**

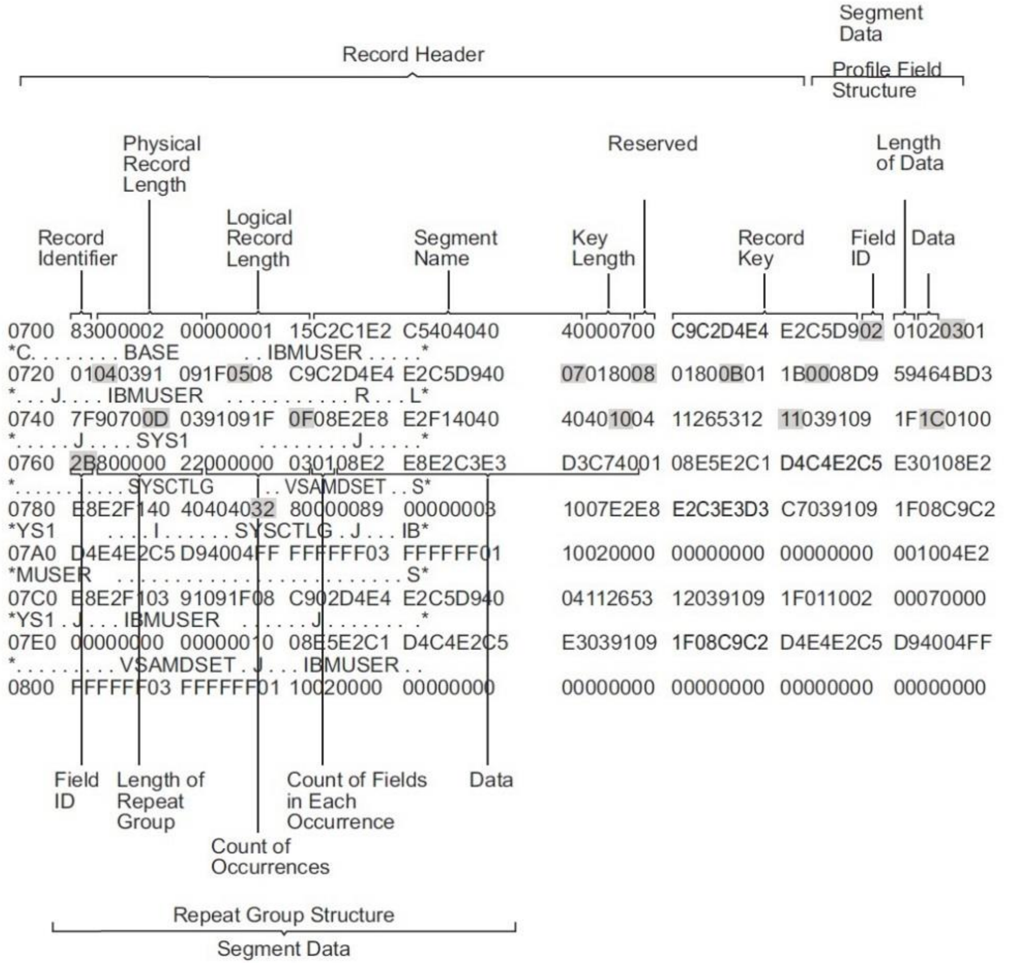


0
1
2
4
6
8
A
C
‘23’x
Alias Entry
Profile Type ‘01’x Group ‘02’x User
Entry Length
Offset to Count of Base Profile Mappings
Front-end Compression Count for Alias
Alias (key/entry) Length
Flags
Entry Name (Key) Template# (Profile type) Segment# Field# ALIAS value
n
n+2
n+4
■ ■ ■
Count of Base Profile Mappings
Base Profile Name Length
Base Profile
Additional Instances of Base Profile Name Length Followed by Base Profile if Count of Base Profile Mappings > 1

*N=’C’x + (Key Length)

The RACF Data Set Internals: Data Blocks

- **RACF profile information is stored in data blocks whose structure is defined in the RACF templates**
- **The data block has a header and then fields defined with a field ID, length, followed by the data**
- **For fields which are not present, the RACF manager returns a default value specified in the templates**
- **Repeat groups include the overall group length and then counts and lengths of each field**



```
0700  83000002  00000001  15C2C1E2  C5404040     40000700  C9C2D4E4  E2C5D902  01020301
*C. . . . . . . . BASE          . . IBMUSER . . . . . .*
0720  01040391  091F0508  C9C2D4E4  E2C5D940     07018008  01800B01  1B0008D9  59464BD3
*. . . J. . . . IBMUSER   . . . . . . . . . . . . R . . . L*
0740  7F90700D  0391091F  0F08E2E8  E2F14040     40401004  11265312  11039109  1F1C0100
*. . . . . . J . . . . . SYS1         . . . . . . . . . . J . . . . . .*
0760  2B800000  22000000  030108E2  E8E2C3E3     D3C74001  08E5E2C1  D4C4E2C5  E30108E2
*. . . . . . . . . . . . SYSCTLG        . . VSAMDSET . . S*
0780  E8E2F140  40404032  80000089  00000003     1007E2E8  E2C3E3D3  C7039109  1F08C9C2
*YS1          . . . . I . . . . . . . . SYSCTLG . J . . . IB*
07A0  D4E4E2C5  D94004FF  FFFFFF03  FFFFFF01     10020000  00000000  00000000  001004E2
*MUSER     . . . . . . . . . . . . . . . . . . . . . . . . . . . . S*
07C0  E8E2F103  91091F08  C902D4E4  E2C5D940     04112653  12039109  1F011002  00070000
*YS1 . J . . . IBMUSER     . . . . . . . J . . . . . . . . . *
07E0  00000000  00000010  08E5E2C1  D4C4E2C5     E3039109  1F08C9C2  D4E4E2C5  D94004FF
*. . . . . . . . . . VSAMDSET . J . . . IBMUSER . .
0800  FFFFFF03  FFFFFF01  10020000  00000000     00000000  00000000  00000000  00000000
```

Finding Free Space

- **When RACF needs to allocate a new data block or index block, RACF has to find space for the new block**

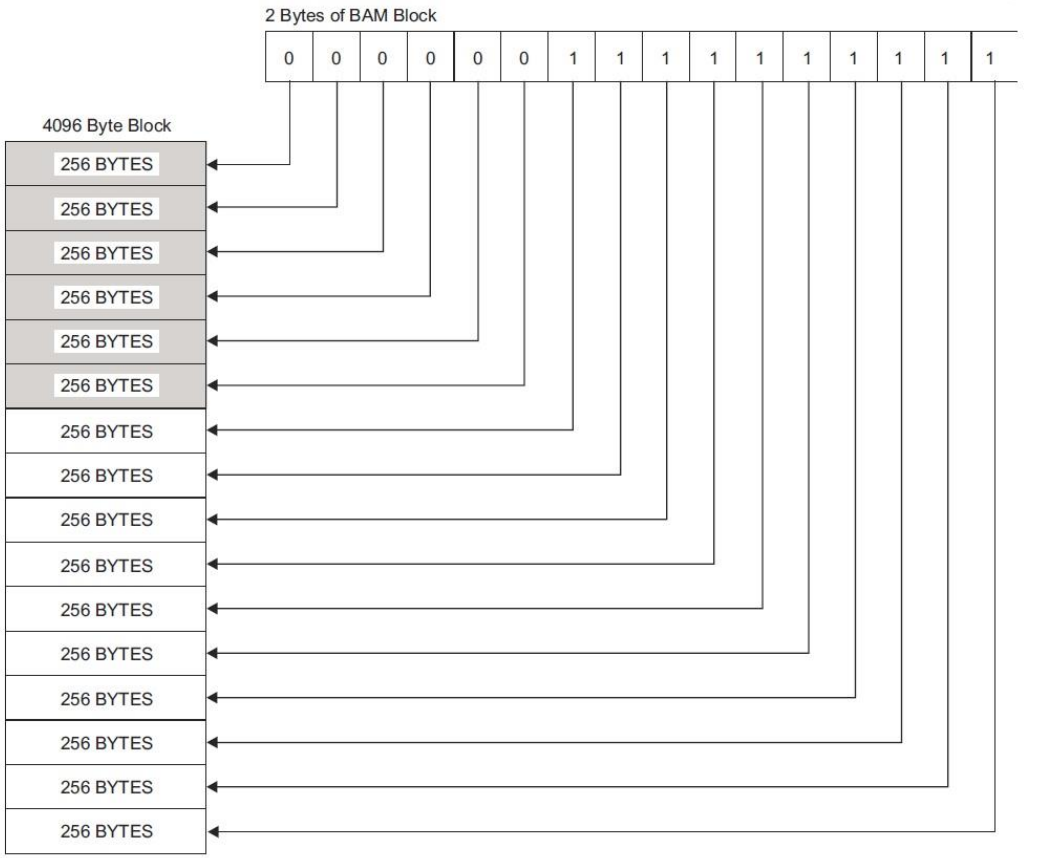
- **Each 4K RACF data base block is divided into 16 256-byte areas called a *slot***
  - Slots are the basic allocation unit for RACF profiles
  - Index blocks, BAM blocks, segment table blocks, and free space blocks are allocated in 16 slot “chunks” on a block boundary

- **Each slot is mapped by a bit in the Byte Allocation Mask (BAM) control blocks**

The RACF Database Internals: Byte Allocation Mask (BAMs)

- **Each BAM block starts with a header which has the RBA of the prior and next BAM blocks, a count of the number of 4K blocks defined by this BAM block, followed by a two byte mask for every block**

- **Each two-byte mask describes the allocation status of each 256-byte slot in the 4,096 byte block**
  - ‘1’B=FREE



Tying it all Together – The ICB

- **The ICB contains information which ties together all of the RACF data set control blocks:**
  - **ICCIBRBA:** RBA of the top level index block
  - **ICISSRBA:** RBA of the first block of the sequence set
  - **ICBAMRBA:** RBA of the first BAM block
  - **ICBBAMNO:** Number of BAM blocks in the data set
  - **ICBAMHWM:** BAM high water mark
  - **ICTMPCNT:** Number of templates
  - **ICBALRBA:** RBA of the top level alias index block
  - **ICBASRBA:** RBA of the first block of the alias sequence set
  - **ICTSEGRB:** RBA of the segment table
  - **ICTREGRB:** length of the segment table

The RACF Data Set Utilities: IRRMIN00 (PARM=NEW)

- **The RACF IRRMIN00 utility (PARM=NEW) formats a data set into a RACF data set, setting:**
  - The “base” ICB
  - The RACF data base templates, which define the format of the data blocks
  - The byte allocation mask blocks, the number of which is determined by the size of the RACF data set

- **IRRMIN00 (PARM=NEW) will not format a “live” RACF data set (either primary or backup)**

- ***IRRMIN00 does not add profiles***
  - RACF initialization adds IBMUSER, SYS1, the system-defined SECLABELS, and the IBM-shipped certificate authority certificates
  - Starting with z/OS V2.3, only a limited number of these CA Certs (3) will be added during initialization
    - RACF will not delete your CA Certs if you are migrating from an earlier release

The RACF Data Set Utilities: IRRMIN00 (PARM=UPDATE)

- **The RACF IRRMIN00 utility (PARM=UPDATE) updates the RACF templates in the RACF database with the version that is shipped with RACF**
  - Templates are taken from CSECT IRRTEMP2

  - Templates contain a three-part indicator consisting of:
    - A 7-character FMID (such as ‘HRF77A0’) or APAR number
    - An 8-digit release level
    - An 8-digit APAR level
      - For example: `OA50735 00000219.00000032`
    - The RACF SET LIST command shows the template version information

  - IRRMIN00 will not downlevel the RACF templates if IRRMIN00 is run from a downlevel system

- **IRRMIN00 (PARM=UPDATE) should be run on all data sets (all primary and all backup) in your RACF data base**

The RACF Data Set Utilities: IRRMIN00 (PARM=ACTIVATE)

- The RACF IRRMIN00 utility (PARM=ACTIVATE) compares the template level of the RACF database to the level of the templates being used by the system.
- If the level of the templates on the RACF database is higher, IRRMIN00 reads the templates from the data base and makes them the current templates.

The RACF Data Base Utilities: IRRUT200

- **The RACF Database Verification Utility (IRRUT200) copies a RACF data set (to an identical size/device type) and identifies inconsistencies in a RACF data set:**
  - Compares the index profile name to the data block profile name
  - Validates the “free” status of the BAM entries
  - Creates a “map” of the BAM blocks in the data set

- **Control statements for IRRUT200 (specified in the SYSIN DD statement) are:**
  - INDEX [FORMAT]
    - Requests the index scanning function. FORMAT indicates that a formatted listing of index blocks is requested. Only one blank can separate INDEX and FORMAT
  - MAP [ALL]
    - Indicates that BAM allocation verification is requested. ALL indicates that you want the encoded map for each BAM block in the RACF DB printed. Only one blank can separate MAP and ALL.

- **Ends with a return code of zero (0) if there are no problems and a non-zero return code (4,8, 12, or 20, depending on the severity) if there is an error.**

IRRUT200 Output: The Index Blocks

- **The top level index block is the first piece of information that IRRUT200 displays.**

```
1                                         ****  INDEX BLOCK VERIFICATION  ****
-                                     ****  SCAN OF INDEX BLOCKS AT LEVEL 03  ****

 BLOCK WITH RBA OF 000000209000

 OFFSET  COMP.                          ENTRY NAME                                     RBA           BAM
         COUNT                                                                                BLOCK  BYTE BIT
  00E    0000  DIGTRING-CERTOWNR.RING01751                                       000000018000   00   044   0
  03C    0000  255 X'FF's                                                        000000208000   00   424   0

 TOTAL NAMES IN THIS BLOCK-002. UNUSED BYTES-3757. AVERAGE NAME LENGTH-141.
 LEVEL NUMBER-03. DISPLACEMENT TO LAST KEY-003C. DISPLACEMENT TO FREE SPACE-014F
 (G) - ENTITY NAME IS GENERIC
```

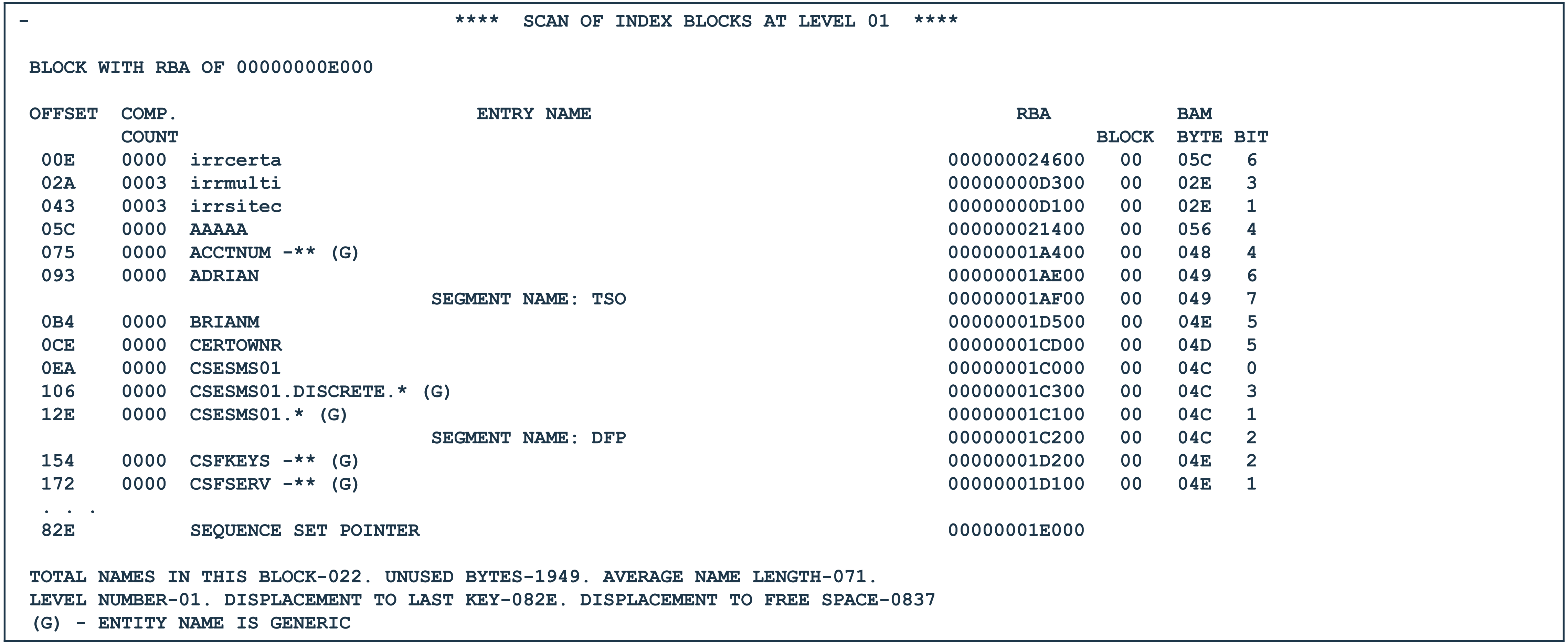
IRRUT200 Output: The Index Blocks…

- **Next, IRRUT200 displays the next lower level index blocks**

```
-                                    ****  SCAN OF INDEX BLOCKS AT LEVEL 02  ****

 BLOCK WITH RBA OF 000000018000

 OFFSET  COMP.                         ENTRY NAME                                    RBA           BAM
         COUNT                                                                           BLOCK  BYTE BIT
  00E    0000  DIGTCERT-01                                                     00000000E000   00   030   0
  02C    0009  DIGTCERT-326                                                    00000001E000   00   050   0
  042    0004  DIGTRING-CERTOWNR.RING00007                                     000000017000   00   042   0

 TOTAL NAMES IN THIS BLOCK-003. UNUSED BYTES-3981. AVERAGE NAME LENGTH-012.
 LEVEL NUMBER-02. DISPLACEMENT TO LAST KEY-0042. DISPLACEMENT TO FREE SPACE-006D
 (G) - ENTITY NAME IS GENERIC


 BLOCK WITH RBA OF 000000208000

 OFFSET  COMP.                         ENTRY NAME                                    RBA           BAM
         COUNT                                                                           BLOCK  BYTE BIT
  00E    0000  JESSPOOL-ARCAE (G)                                              0000002CD000   00   5AE   0
  02F    0000  255 X'FF's                                                      000000023000   00   05A   0

 TOTAL NAMES IN THIS BLOCK-002. UNUSED BYTES-3770. AVERAGE NAME LENGTH-134.
 LEVEL NUMBER-02. DISPLACEMENT TO LAST KEY-002F. DISPLACEMENT TO FREE SPACE-0142
 (G) - ENTITY NAME IS GENERIC
```

IRRUT200 Output: The Index Blocks…

- **The Sequence Set (Level 1 Index) is the shown last…**



```
-                                     ****  SCAN OF INDEX BLOCKS AT LEVEL 01  ****

 BLOCK WITH RBA OF 00000000E000

 OFFSET  COMP.                       ENTRY NAME                                             RBA           BAM
         COUNT                                                                                    BLOCK  BYTE BIT
  00E    0000  irrcerta                                                                000000024600   00    05C   6
  02A    0003  irrmulti                                                                00000000D300   00    02E   3
  043    0003  irrsitec                                                                00000000D100   00    02E   1
  05C    0000  AAAAA                                                                   000000021400   00    056   4
  075    0000  ACCTNUM -** (G)                                                         00000001A400   00    048   4
  093    0000  ADRIAN                                                                  00000001AE00   00    049   6
                                   SEGMENT NAME: TSO                                   00000001AF00   00    049   7
  0B4    0000  BRIANM                                                                  00000001D500   00    04E   5
  0CE    0000  CERTOWNR                                                                00000001CD00   00    04D   5
  0EA    0000  CSESMS01                                                                00000001C000   00    04C   0
  106    0000  CSESMS01.DISCRETE.* (G)                                                 00000001C300   00    04C   3
  12E    0000  CSESMS01.* (G)                                                          00000001C100   00    04C   1
                                   SEGMENT NAME: DFP                                   00000001C200   00    04C   2
  154    0000  CSFKEYS -** (G)                                                         00000001D200   00    04E   2
  172    0000  CSFSERV -** (G)                                                         00000001D100   00    04E   1
  . . .
  82E          SEQUENCE SET POINTER                                                    00000001E000

 TOTAL NAMES IN THIS BLOCK-022. UNUSED BYTES-1949. AVERAGE NAME LENGTH-071.
 LEVEL NUMBER-01. DISPLACEMENT TO LAST KEY-082E. DISPLACEMENT TO FREE SPACE-0837
 (G) - ENTITY NAME IS GENERIC
```

IRRUT200 Output: The Index Blocks…

- **The Sequence Set (Level 1 Index) blocks are the shown last…**

```
BLOCK WITH RBA OF 00000001E000

OFFSET  COMP.                          ENTRY NAME                                     RBA           BAM
        COUNT                                                                                BLOCK  BYTE BIT
 00E    0000  DIGTCERT-01.premium-server@thawte.com.CN=Thawte¢Premium¢Serv      000000010200   00   034   2
              er¢CA.OU=Certification¢Services¢Division.O=Thawte¢Consulting
              ¢cc.L=Cape¢Town.SP=Western¢Cape.C=ZA
                                   SEGMENT NAME: CERTDATA                       000000010400   00   034   4
 0C5    0012  DIGTCERT-01.server-certs@thawte.com.CN=Thawte¢Server¢CA.OU=C      00000000DE00   00   02F   6
              ertification¢Services¢Division.O=Thawte¢Consulting¢cc.L=Cape
              ¢Town.SP=Western¢Cape.C=ZA
                                   SEGMENT NAME: CERTDATA                       00000000F300   00   032   3
 . . .
 . . .
 . . .
 617          SEQUENCE SET POINTER                                              000000017000

TOTAL NAMES IN THIS BLOCK-010. UNUSED BYTES-2508. AVERAGE NAME LENGTH-127.
LEVEL NUMBER-01. DISPLACEMENT TO LAST KEY-0617. DISPLACEMENT TO FREE SPACE-0620
(G) - ENTITY NAME IS GENERIC
```

IRRUT200 Output: The Sequence Set

- **Next, the Sequence Set RBAs are shown along with data set statistics**

```
 1                                           ****  SEQUENCE SET RBAS  ****
       RBA  00000000E000
       RBA  00000001E000
       RBA  000000017000
       RBA  0000002CD000
       RBA  000000023000


1                                      ****  INDEX FUNCTION STATISTICS  ****

0TOTAL NUMBER OF NAMES IN RACF DATA SET 00000129
 TOTAL NUMBER OF INDEX BLOCKS IN RACF DATA SET 00000008
 AVERAGE NUMBER OF NAMES PER INDEX BLOCK 016
 AVERAGE NAME LENGTH 046
 AVERAGE NUMBER OF UNUSED BYTES PER INDEX BLOCK 2938
 TOTAL NUMBER OF LEVEL 01 BLOCKS IN RACF DATA SET 00000005
```

IRRUT200 Output: BAM Map

- **Last, the analysis of all of the blocks in the data base blocks is shown**

```
1                                           ****  BAM BLOCK VERIFICATION  ****
0             ****  SYMBOL LEGEND  ****

          *   BAM=ALLOC   , ACTUAL=ALLOC
          0   BAM=UNALLOC , ACTUAL=UNALLOC
          .   BAM=ALLOC   , ACTUAL=UNALLOC
          +   BAM=UNALLOC , ACTUAL=ALLOC
          I   INDEX BLOCK WITH LEVEL IN NEXT POSITIONS
          B   BAM BLOCK
          T   TEMPLATE BLOCK
          S   SEGMENT TABLE BLOCK
          F   FIRST BLOCK (ICB)
          -   BAM=UNALLOC , ACTUAL=ALLOC  I,B,OR F BLK
          $   BAM=UNALLOC , ACTUAL=ALLOC  SPECIAL BLK
          ?   BAM=ALLOC   , ACTUAL=ALLOC  UNKNOWN BLK
          %   BAM=UNALLOC , ACTUAL=ALLOC  UNKNOWN BLK
          @   BAM=ALLOC   , DUPLICATE ALLOCATION
          #   BAM=UNALLOC , DUPLICATE ALLOCATION
          /   UNDEFINED STORAGE
-BLOCK 000 RBA 00000000C000
 014 FFFFFFFF FFFFFFFF TTTTTTTT TTTTTTTT TTTTTTTT TTTTTTTT TTTTTTTT TTTTTTTT TTTTTTTT TTTTTTTT TTTTTTTT TTTTTTTT TTTTTTTT
 021 TTTTTTTT TTTTTTTT TTTTTTTT TTTTTTTT TTTTTTTT TTTTTTTT TTTTTTTT TTTTTTTT TTTTTTTT SSSSSSSS SSSSSSSS BBBBBBBB BBBBBBBB
 02E ******** ******** I1111111 11111111 ******** ******** ******** ******** ******** ******** ******** ******** ********
 03B ******** ******** ******** ******** ******** ******** ******** I1111111 11111111 I2222222 22222222 ******** ********
 048 ******** ******** ******** ******** ******** ******** ******** ******** I1111111 11111111 ***00000 00000000 000000**
 055 ******** ******** ******** ******** ******** I1111111 11111111 ******** ******** ******** ******** ******** ********
 062 ******** ******** *0000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

IRRUT200 Output: BAM Map…

- … followed by statistics

```
F50 //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// ////////
F5D //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// ////////
F6A //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// ////////
F77 //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// ////////
F84 //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// ////////
F91 //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// ////////
F9E //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// ////////
FAB //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// ////////
FB8 //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// ////////
FC5 //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// ////////
FD2 //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// ////////
FDF //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// ////////
FEC //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// //////// ////////
FF9 //////// //////// //////// //////// //////// //////// ////////
**** MAP FUNCTION STATISTICS ****
NUMBER OF BAM BLOCKS DEFINED 001
LAST BAM THAT DEFINES USED SPACE - RBA 00000000C000
RACF DATA SET IS 4 PERCENT FULL.
TOTAL NUMBER OF INDEX BLOCKS IN RACF DATA SET 00000003
TOTAL NUMBER OF LEVEL 01 BLOCKS IN RACF DATA SET 00000002
NUMBER OF GROUP ENTRIES - 0000003
NUMBER OF USER ENTRIES - 0000016
NUMBER OF DATASET ENTRIES - 0000002
NUMBER OF DASDVOL ENTRIES - 0000003
NUMBER OF DIGTCERT ENTRIES - 0000025
NUMBER OF SECLABEL ENTRIES - 0000004
```

The RACF Data Base Utilities: IRRUT400

- **The RACF Database Split/Merge Utility (IRRUT400) copies a RACF data set to a larger or smaller data set on a different type of volume and:**
  - Redistributes the profiles among data sets (split/merge)
    - Identifying inconsistencies (such as duplicate profiles)
  - Physically reorganizes the database by bringing all segments of a given profile together
  - Recreates the higher level index blocks from the sequence set
  - Recreates the BAM blocks and corrects entries which IRRUT200 has flagged as “not valid”
  - “Compresses” the index entries

The RACF Data Base Utilities: IRRUT400…

- **IRRUT400 allows you to control how much free space is allocated in the *level 1* index blocks**
  - Extra space allows for new profiles to be defined without requiring an L1 index block split, which could ripple into an L2 index split, which could ripple into an L3 index split, etc.
  - FREESPACE(0) is the default
  - FREESPACE(30) is the recommendation
  - ***Applies only to the L1 index block;*** Upper level index blocks get approximately seven (7) percent free space

- **IRRUT400 allows you to control how segments which span a 256-byte slot are allocated in the new data set**
  - ALIGN forces segments that occupy multiple 256-byte slots to be placed so that they do not span 4096-byte physical blocks.
  - NOALIGN is the default

The RACF Data Base Utilities: IRRDBU00

- **The RACF Database Unload Utility (IRRDBU00) reads a primary, back-up or offline copy of the RACF database**
  - Requires READ authority to the input RACF data set(s) for PARM=NOLOCKINPUT, UPDATE authority for PARM=LOCKINPUT
  - Consider permitting access to the RACF data set using program control and program access to data sets (PADS)

- **IRRDBU00 reads every profile in the RACF database, which means that it is implicitly validating the index structure for every profile and the data structure for each profile.**
  - Index errors which are encountered by the RACF data manager are surfaced by the RACF data manager
  - Data errors result in “unexpected” IRRDBU00 output

And in Conclusion…

- **Understanding the internal structure of a RACF data set is essential in:**
  - Determining when a RACF data set reorganization is needed
  - Understanding the output of the RACF data set utilities
  - Operating your RACF data base at peak performance

- **For more information see:**
  - RACF Security Systems Programmer's Guide
  - RACF Diagnosis Guide
  - RACF Macros and Interfaces

VANGUARD SECURITY & COMPLIANCE

Questions

How to Contact Me

Mark Nelson

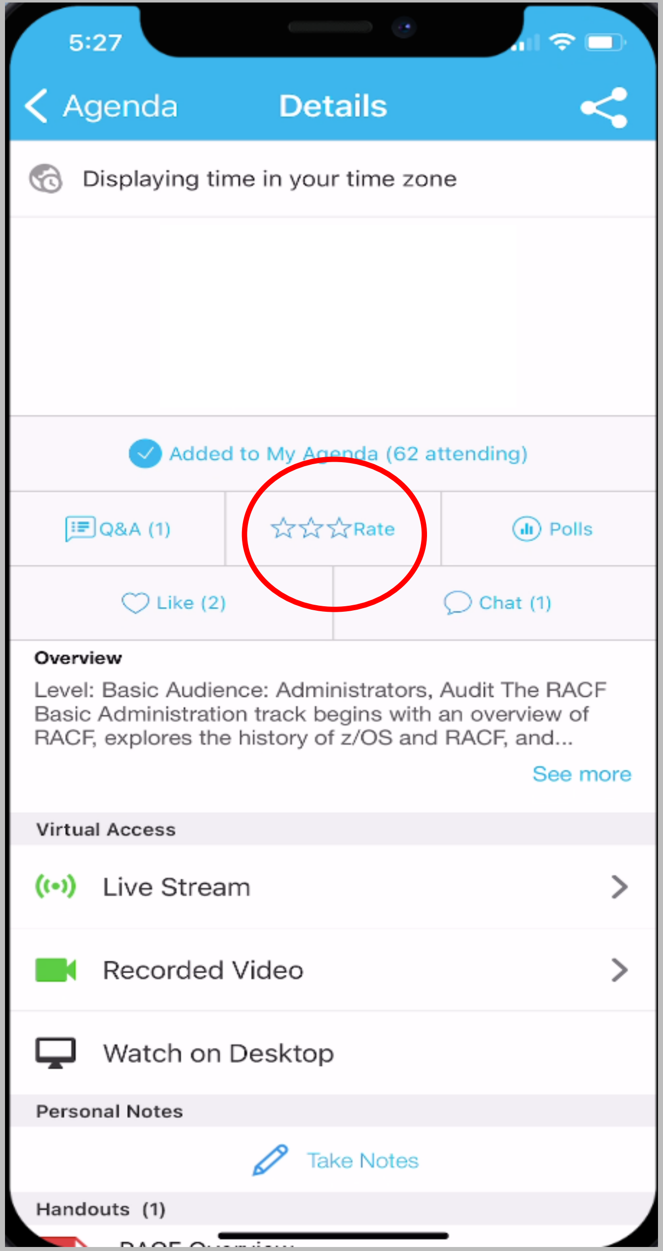
markan@us.ibm.com

VANGUARD SECURITY & COMPLIANCE

Session Evaluation

Be sure to rate your experience using the VSC2023 app.

Your opinion helps us bring you the best experience.
Please let us know your thoughts.



5:27

< Agenda Details

Displaying time in your time zone

Added to My Agenda (62 attending)

Q&A (1) | ☆☆☆Rate | Polls

Like (2) | Chat (1)

Overview

Level: Basic Audience: Administrators, Audit The RACF Basic Administration track begins with an overview of RACF, explores the history of z/OS and RACF, and...

See more

Virtual Access

Live Stream >

Recorded Video >

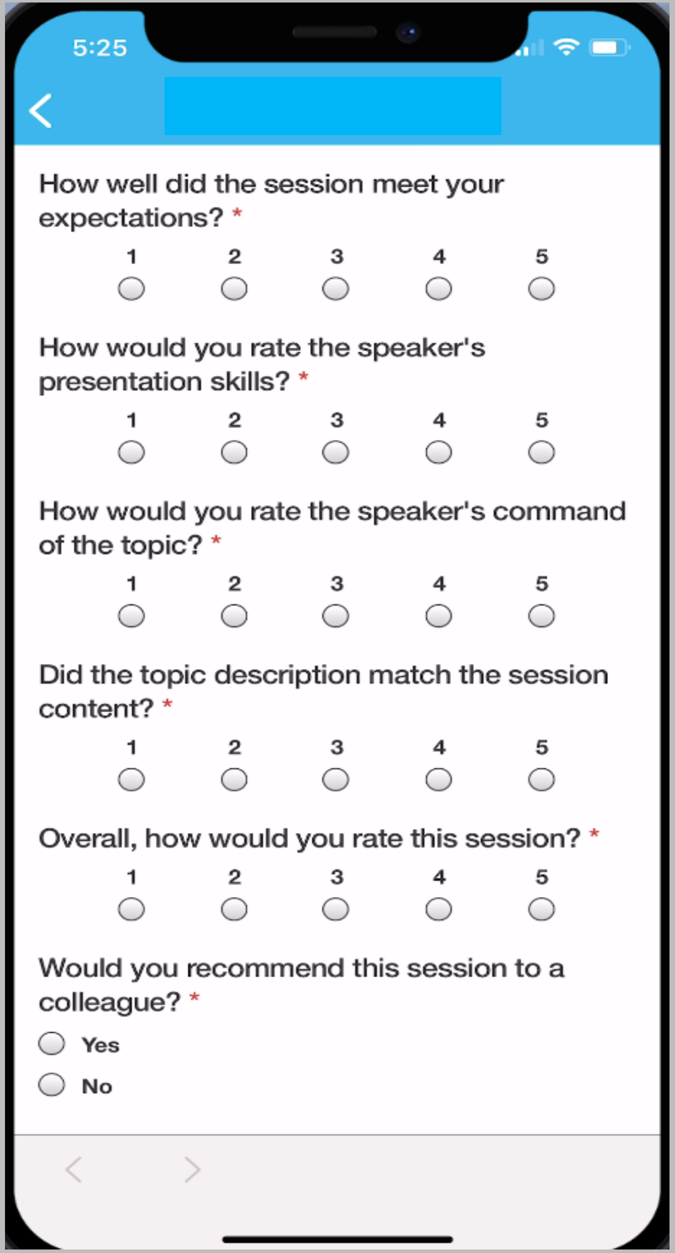
Watch on Desktop

Personal Notes

Take Notes

Handouts (1)

RACF Overview



5:25

How well did the session meet your expectations? *

1 ○ 2 ○ 3 ○ 4 ○ 5 ○

How would you rate the speaker's presentation skills? *

1 ○ 2 ○ 3 ○ 4 ○ 5 ○

How would you rate the speaker's command of the topic? *

1 ○ 2 ○ 3 ○ 4 ○ 5 ○

Did the topic description match the session content? *

1 ○ 2 ○ 3 ○ 4 ○ 5 ○

Overall, how would you rate this session? *

1 ○ 2 ○ 3 ○ 4 ○ 5 ○

Would you recommend this session to a colleague? *

○ Yes

○ No

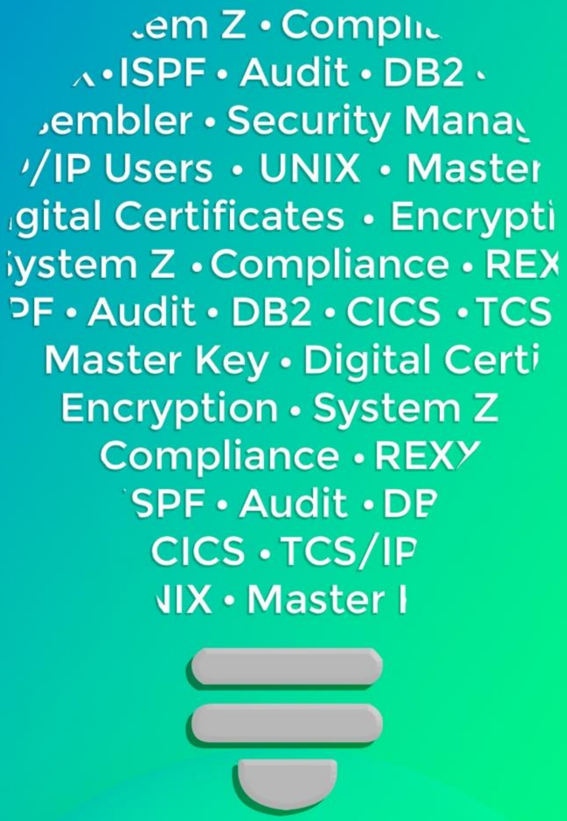
VANGUARD 2023
Security & Compliance

Database Adnalesque Cano: I Sing of a Database and its Records
A Peek Inside the RACF®, Database

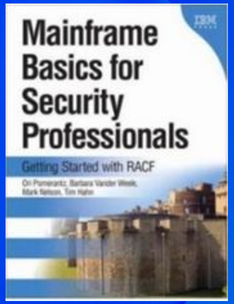
Mark Nelson, CISSP®, CSSLP®,

Senior Technical Staff Member

IBM®, Poughkeepsie



KNOWLEDGE
is your best defense


Mainframe Basics for Security Professionals

Getting Started with RACF

Ori Pomerantz, Barbara Vander Weele, Mark Nelson, Tim Hahn