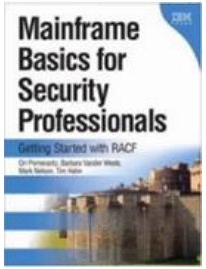
The Life and Times of an ACEE

Session FG

Mark Nelson, CISSP ®, CSSLP®

RACF ® Design and Development, IBM ®



Mainframe Basics for Security Professionals

Getting Started with RACF

Roadmap

- What is an ACEE?
- What is in an ACEE?
- How are they created and deleted?
- How are ACEEs used?
- Can ACEEs be transported?
- What are the challenges with caching ACEEs?
- Why do I care about all of this?

Roadmap Notes

- **Small Print: There are a few simplifying assumptions:**
  - While we focus on RACROUTE REQUEST=VERIFY, there are other ways to create an ACEE (RACINIT, SAF Callable services)
  - Some technical explanations are simplified

Is there Really an ACEE Life Cycle?

- **Just like “the birds and the bees and the flowers and the trees”, control blocks in z/OS have a life cycle**
  - Some are born and live for the life of the IPL
  - Some are born and die with the address space
  - Some exist for the life of the job step
  - Some are around only when the TCB is around
  - Some have shorter lives than that

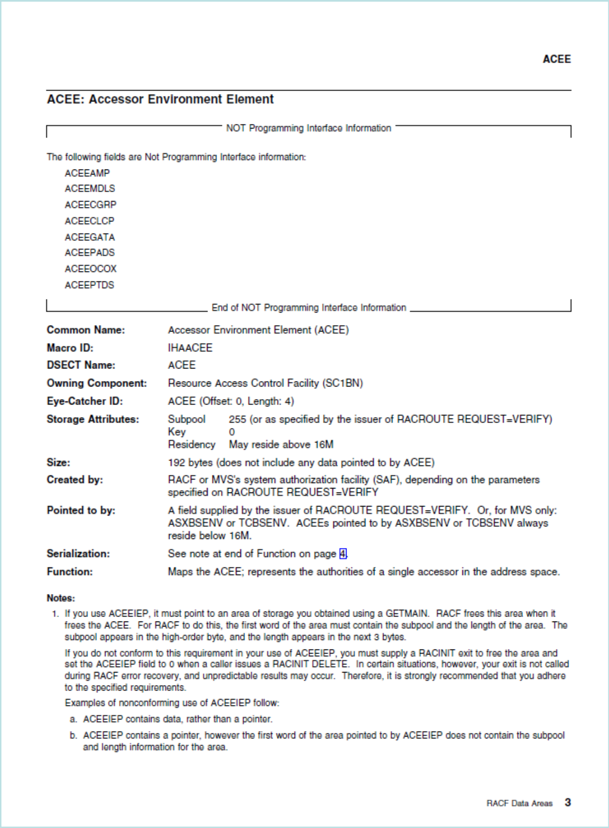
- **Like many other control blocks, ACEEs have a variety of life spans. *A__ll__ <u>of those life spans are outside the control of RACF</u>***

What’s in an ACEE and Why do I Care?

What is an ACEE?

- **An ACcessor Environment Element (ACEE) is an MVS/SAF (*not* RACF) control block**
  - It’s storage….that is central to RACF’s processing!
  - Documented in “RACF Data Areas” and in the ***IHA***ACEE macro in MACLIB
  - Supported by all ESMs

- **The Creator is responsible for managing the lifecycle of the ACEE; If you created it, you:**
  - Must be certain that it is available when it needs to be used
  - ~~Really should~~ Must delete it



ACEE

ACEE: Accessor Environment Element

NOT Programming Interface Information

The following fields are Not Programming Interface information:

ACEEAMP
ACEEMDLS
ACEECGRP
ACEECLCP
ACEEGATA
ACEEPADS
ACEEOCOX
ACEEPTDS

End of NOT Programming Interface Information

Common Name: Accessor Environment Element (ACEE)
Macro ID: IHAACEE
DSECT Name: ACEE
Owning Component: Resource Access Control Facility (SC1BN)
Eye-Catcher ID: ACEE (Offset: 0, Length: 4)
Storage Attributes: Subpool 255 (or as specified by the issuer of RACROUTE REQUEST=VERIFY)
Key 0
Residency May reside above 16M
Size: 192 bytes (does not include any data pointed to by ACEE)
Created by: RACF or MVS's system authorization facility (SAF), depending on the parameters specified on RACROUTE REQUEST=VERIFY
Pointed to by: A field supplied by the issuer of RACROUTE REQUEST=VERIFY. Or, for MVS only: ASXBSENV or TCBSENV. ACEEs pointed to by ASXBSENV or TCBSENV always reside below 16M.
Serialization: See note at end of Function on page 4.
Function: Maps the ACEE; represents the authorities of a single accessor in the address space.

Notes:

1. If you use ACEEIEP, it must point to an area of storage you obtained using a GETMAIN. RACF frees this area when it frees the ACEE. For RACF to do this, the first word of the area must contain the subpool and the length of the area. The subpool appears in the high-order byte, and the length appears in the next 3 bytes.

   If you do not conform to this requirement in your use of ACEEIEP, you must supply a RACINIT exit to free the area and set the ACEEIEP field to 0 when a caller issues a RACINIT DELETE. In certain situations, however, your exit is not called during RACF error recovery, and unpredictable results may occur. Therefore, it is strongly recommended that you adhere to the specified requirements.

   Examples of nonconforming use of ACEEIEP follow:

   a. ACEEIEP contains data, rather than a pointer.

   b. ACEEIEP contains a pointer, however the first word of the area pointed to by ACEEIEP does not contain the subpool and length information for the area.

RACF Data Areas 3

What is an ACEE?...

- **ACEEs may be chained to other system-level control blocks or may exist “free floating” in storage**
  - ASCBASXB->ASXBSENV->ACEE
  - PSATOLD->TCBSENV->ACEE

- **Multi-user address spaces make extensive use of TCB-level ACEEs and “free floating” ACEEs**

- **We’ll talk about the order of checking for ACEEs in a few moments…**

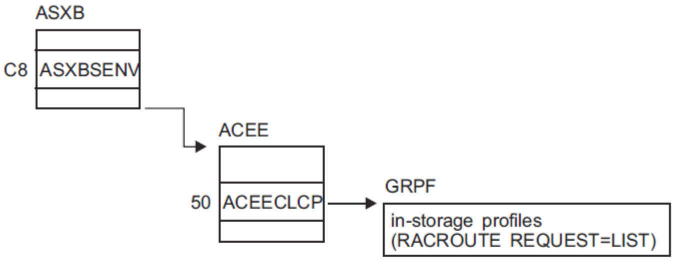


Figure 68. Control block overview: ACEE in a single-user address space

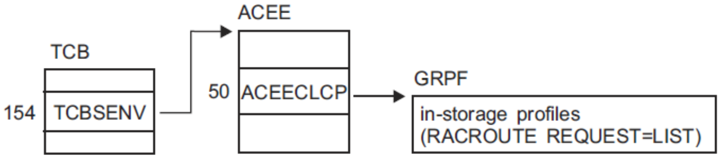


Figure 69. Control block overview: ACEE in a multiple-user address space

What is an ACEE?...

- **ACEEs must be allocated in storage with the appropriate characteristics**
    - **Storage key:** ACEEs should never be in a storage key that can be modified by an unauthorized program
    - **Subpool:** ACEEs should be in a subpool which will not cause the ACEE to be freed prior to its use
        - “Take care in selecting a subpool, as MVS makes certain assumptions about subpool usage and characteristics.”

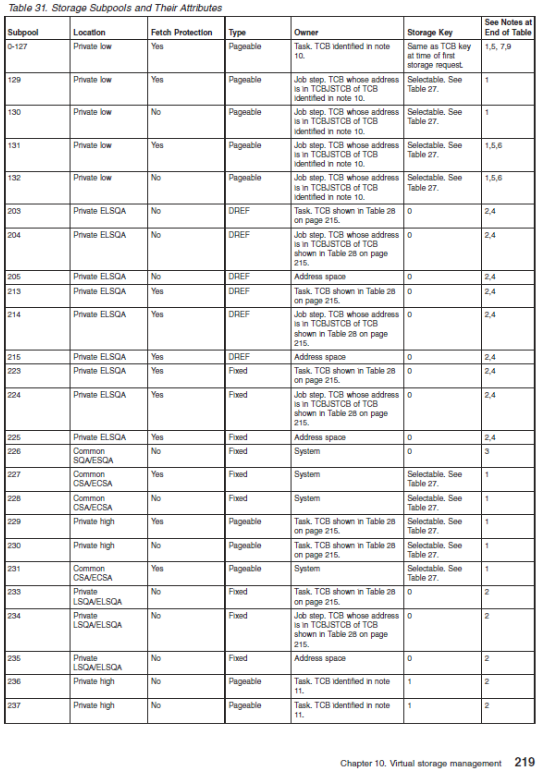


Table 31. Storage Subpools and Their Attributes

Subpool	Location	Fetch Protection	Type	Owner	Storage Key	See Notes at End of Table
0-127	Private low	Yes	Pageable	Task. TCB identified in note 10.	Same as TCB key at time of first storage request.	1,5, 7,9
129	Private low	Yes	Pageable	Job step. TCB whose address is in TCBJSTCB of TCB identified in note 10.	Selectable. See Table 27.	1
130	Private low	No	Pageable	Job step. TCB whose address is in TCBJSTCB of TCB identified in note 10.	Selectable. See Table 27.	1
131	Private low	Yes	Pageable	Job step. TCB whose address is in TCBJSTCB of TCB identified in note 10.	Selectable. See Table 27.	1,5,6
132	Private low	No	Pageable	Job step. TCB whose address is in TCBJSTCB of TCB identified in note 10.	Selectable. See Table 27.	1,5,6
203	Private ELSQA	No	DREF	Task. TCB shown in Table 28 on page 215.	0	2,4
204	Private ELSQA	No	DREF	Job step. TCB whose address is in TCBJSTCB of TCB shown in Table 28 on page 215.	0	2,4
205	Private ELSQA	No	DREF	Address space	0	2,4
213	Private ELSQA	Yes	DREF	Task. TCB shown in Table 28 on page 215.	0	2,4
214	Private ELSQA	Yes	DREF	Job step. TCB whose address is in TCBJSTCB of TCB shown in Table 28 on page 215.	0	2,4
215	Private ELSQA	Yes	DREF	Address space	0	2,4
223	Private ELSQA	Yes	Fixed	Task. TCB shown in Table 28 on page 215.	0	2,4
224	Private ELSQA	Yes	Fixed	Job step. TCB whose address is in TCBJSTCB of TCB shown in Table 28 on page 215.	0	2,4
225	Private ELSQA	Yes	Fixed	Address space	0	2,4
226	Common SQA/ESQA	No	Fixed	System	0	3
227	Common CSA/ECSA	Yes	Fixed	System	Selectable. See Table 27.	1
228	Common CSA/ECSA	No	Fixed	System	Selectable. See Table 27.	1
229	Private high	Yes	Pageable	Task. TCB shown in Table 28 on page 215.	Selectable. See Table 27.	1
230	Private high	No	Pageable	Task. TCB shown in Table 28 on page 215.	Selectable. See Table 27.	1
231	Common CSA/ECSA	Yes	Pageable	System	Selectable. See Table 27.	1
233	Private LSQA/ELSQA	No	Fixed	Task. TCB shown in Table 28 on page 215.	0	2
234	Private LSQA/ELSQA	No	Fixed	Job step. TCB whose address is in TCBJSTCB of TCB shown in Table 28 on page 215.	0	2
235	Private LSQA/ELSQA	No	Fixed	Address space	0	2
236	Private high	No	Pageable	Task. TCB identified in note 11.	1	2
237	Private high	No	Pageable	Task. TCB identified in note 11.	1	2

Chapter 10. Virtual storage management 219

For What are ACEEs Used?

- **ACEEs are used for several purposes:**
  - Identification of users
  - Privilege information
  - Authorization information
  - Environment information
  - Installation information
  - Logging
  - Control

- **While processing "work", every address space has at least one ACEE**

ACEE Contents: Identification Information

- **The ACEE associates identity with work**
  - **ACEEUSER:** 1-8 character user ID
    - **Some special user IDs values:**
      - ***BYPASS*:** Auditable work that bypasses authorization checking
      - ***:** No user id
  - **ACEE5PTR**: X.500 name pair structure for logging
  - **ACEEICTX:** Identity context extension
  - **ACEEIDID:** Distributed identity data

ACEE Contents: Privilege Information

- **An ACEE holds privilege information:**

ACEE Field	Description
ACEESPEC	RACF SPECIAL
ACEEOPER	RACF OPERATIONS
ACEEAUDT	RACF AUDITOR
ACEEROA	Read-only AUDITOR
ACEEPRIV	Started task with the privileged attribute
ACEEUATH	CLAUTH(USER)
ACEEDASD	CLAUTH(DASDVOL)
ACEETAPE	CLAUTH(TAPEVOL)
ACEETERM	CLAUTH(TERMINAL)

ACEE Contents: Environmental Information

- **How did this user/work enter the system?**

  - The Port of Entry (POE) describes how the users entered the system

  - Consists of a name (1-8 characters) and a class:
    - TERMINAL
    - CONSOLE
    - JESINPUT
    - APPCPORT
    - SERVAUTH
  - Used in RACF’s conditional access list processing

ACEE Contents: Authorization Information

- **ACEERACF:** Is this a RACF-defined user?
- **ACEERASP:** Is this the RACF address space?
- **ACEECGRP:** The groups to which this user is connected. Can be changed by REQUEST=AUTH
- **ACEEFGRP:** The groups to which this user is connected. Fixed at time of ACEE creation (used by REQUEST=FASTAUTH).
- **ACEEDSLP:** Categories to which the user is authorized
- **ACEESLVL:** Maximum security level for this user

ACEE Contents: Authorization Information…

- **ACEETRMP:** Address of terminal ID
- **ACEETOKP:** Pointer to UTOKEN

ACEE Contents: Installation Information

- **Installation information can be placed in the ACEE for quick reference**
  - **ACEEIEP:** Reserved for installation use. Must point to a one byte subpool, followed by a three byte length followed by data
  - **ACEEINSTP:** Installation data from the user profile
  - **ACEEAPLV:** APPL profile level
  - **ACEETRLV:** Port of entry level

ACEE Contents: Logging Information

- **An ACEE controls certain aspects of logging**
  - **ACEELOGU:** User is to have most RACF functions logged (UAUDIT attribute**)**
  - **ACEEDALY:** User logged into an application which only records daily logon stats

ACEE Contents: Control Information

- **ACEEACEE:** Eyecatcher
- **ACEESP/ACEELEN/ACEEVRSN:** ACEE subpool, length and version
- **ACEECLCP:** Pointer to GLOBAL=NO profiles RACLISTed by this user
- **ACEEMODE:** 24 or 31 bit ACEE data areas
- **ACEEPADS:** Pointer to list of data sets accessed by controlled programs executed by this user
- **ACEEPTDS:** Pointer to first temporary data set table
- … and many more!

The Birth and Death of the ACEE

The Birth of ACEEs

- **The three primary means of creating an ACEE are:**
  - RACROUTE REQUEST=VERIFY,ENVIR=CREATE
  - The InitACEE Callable Service
    - “This service is only intended for use by the z/OS UNIX kernel or other MVS servers that do not use Z/OS UNIX”
  - RACINIT ENVIR=CREATE
    - Deprecated… use RACROUTE REQUEST=VERIFY

- **Some products have higher-level interfaces which call these services**

The Birth of ACEEs

- **The creator of an ACEE can specify the desired identity as:**
  - **An Explicitly specified User ID:** The classic 1 to 8 character z/OS identification of a user
  - **A TOKEN:** A TOKEN is an 80-byte “synopsis” of an identity that contains the user ID, group ID, and a minimal set of privileging information
  - **A Kerberos credential**
  - **A digital certificate**
  - **A RACF Identity Token**
  - **An Environment Object (“RACO”):** More on these later…

The Birth of ACEEs…

- **A few rules for creating ACEEs:**
  - If ACEE= is not specified, the address of the newly created ACEE is stored in the TCBSENV field of the task control block. *The TCBSENV field is set unconditionally if ACEE= is not specified and ASXBSENV is non-zero*
  - If there is no address-space level ACEE (ASXBSENV=0), *the new ACEE address is stored in ASXBSENV and will be returned if ACEE= specified*
  - If any RACF exits are AMODE(24), then all of the ACEEs that are created will be "below the line" so that they can be accessed by the RACF exits
  - Do not create an ACEE above 16MB and place the address in ASXBSENV or TCBSENV

The Death of ACEEs

- **RACROUTE REQUEST=VERIFY,ENVIR=DELETE, the InitACEE Callable Service, and RACINIT ENVIR=DELETE are the primary ways for deleting ACEEs**
  - The ACEE that is to be deleted is identified by its address
  - If there is no ACEE address specified, then the “current” ACEE (TCBSENV or ASXBSENV) is deleted
    - Probably not what you wanted.

- **ACEEs which have been deleted will have the eyecatcher set to ‘acee’**

Using ACEEs

Using ACEEs

- **All of the RACROUTE REQUESTs allow the specification of an ACEE except for REQUEST=STAT, TOKENBLD, TOKENMAP, and VERIFYX**

- **Many z/OS services and other products use the ACEE**

- **If no ACEE is specified, many of these services will:**
  - If there is a task-level ACEE (TCBSENV), it is used
  - If there is no task-level ACEE, the address space level ACEE (ASXBSENV) is used

Using ACEEs…

- **Originally, ACEEs were intended for use only within the address space**

- **Some RACF services and some SAF callable services allow the referencing of ACEEs in other address spaces by specifying the ALET of the address space which contains the ACEE**

- **ACEEs can be “transformed” by RACF and by SAF Services into other formats for various purposes**

Welcome to the REQUEST=AUTH ACEE Party!

- **RACROUTE REQUEST=AUTH allows you to “specify” an identity/ACEE in three different ways:**
  - **First-Party:** You don’t specify any identity information. RACF locates the ACEE (TCBSENV and then ASXBSENV).
  - **Second-Party:** You provide the ACEE
  - **Third-party:** You provide UTOKEN, USERID, or GROUPID and RACF creates an ACEE for that user ID and uses it for the REQUEST=AUTH
    - RACF anchors the address of this ACEE in the ACEE associated with the request (ACEE3PTY) so that if you do a similar third-party check, the ACEE does not have to be recreated.

Nested ACEEs

- **Specifying NESTED=YES or NESTED=COPY on a REQUEST=VERIFY,ENVIR=CREATE causes the ACEE which is being created to contain information about the current address space.**
  - **NESTED=YES** causes the encapsulation of ACEE information from the current address space ACEE in the ACEE which is being created
  - **NESTED=COPY** causes the copying of the encapsulated information that is in the current address space ACEE into the ACEE which is being created
- **The encapsulated information is in the form of a RACF environment object (RACO)**

Transporting ACEEs

Transporting ACEEs

- ACEEs contain addresses, which complicates how they can be moved or copied

- RACF had solved this problem with RACF/VM with the concept of a “flattened” RACROUTE REQUEST parameter list (including its data) to be transported from the requestor to the RACF/VM service machine

- This concept was extended to the RACF Environment Object (“RACO”)

Transporting ACEEs: The Environment Object

- **The RACF Environment Object is a representation of the ACEE that can be moved from one address space to another and can be very quickly transformed into an ACEE**.

- **Think of it as an ACEE with all of the data appended at the end and all of the pointers replaced by offsets to that data**

- ***The environment object should only be used on a system which shares the same security database***

Caching

Caching in RACF

- **To minimize I/O to the RACF data base, RACF utilizes two primary types of caches**
  - **Database:** Caching of 4K data blocks from the RACF database
    - **(E)CSA:** Up to 256 4K buffers for each RACF data set
    - **Coupling Facility:** Up to the entire contents of the RACF data set
  - **Object:** Explicit caching of RACF “objects” in the Virtual Look-Aside Facility (VLF)
    - **Group tree in storage**
    - **GID mapping**
    - **ACEEs**
- **The RACF InitACEE callable service can be used to cache ACEEs (“managed ACEEs”)**
- **Applications augment object caching with their own caching mechanisms**

The RACF Caching Challenge: Keeping it Current

- **I/O caching uses the relative byte address of the block to determine when the local ((E)CSA) or CF buffers**
  - When a block is changed, the block in the CF is updated and when other systems in the plex attempt to access that block in the CF, they are told that the block has been invalidated in their local cache which causes them to retrieve the current version from the CF

- **ACEE caching in VLF requires that profile changes do not result in the ACEE caching returning outdated results**

The VLF ACEE Cache

- **The ACEEs (in the form of RACF Environment Objects) are stored as objects in the IRRACEE VLF class**

- **One VLF object may contain multiple ACEEs for a user, based on:**
  - Default group name
  - Port of entry (POE), for example, terminal ID or console ID
  - Application name
  - Security label
  - Session type

The ACEE Caching Challenge…

- **If RACF knows what user is affected by a profile change, only that user’s VLF ACEE is deleted**
  - Changing a security-sensitive field in the RACF database causes that users ACEE object to be purged on the local system

- **If RACF can’t tell what user would be affected, *all ACEE VLF objects are purged***
  - If RACF Sysplex Communication has not been enabled, an update on any security sensitive field made on another system
  - Group membership changes on another system sharing the RACF database
  - Certain SETROPTS changes (e.g. ADSP)

- **There has been APAR activity in this area! Look for RACPWENCR/K FIXCAT keyword**

The ACEE Caching Challenge…

- **SETROPTS CLASSACT, NOCLASSACT, RACLIST REFRESH, or NORACLIST of any of these classes causes a purging of all VLF ACEE objects:**
  - APPCPORT, APPL, CONSOLE, GTERMINL, JESINPUT, SECLABEL, SERVAUTH, TERMINAL
  - FACILITY (SETROPTS MLS only)

Sample ACEE Lives

TSO

1. **User issues application command (Anything defined in a USSTAB for a TSO application - many installations use LOGON on the VTAM screen)**

2. **VTAM creates address space and gives control to the specified command, in this case TSO LOGON.**

3. **IF PASSWORDPREPROMPT(ON) is specified:**
   1. VERIFY CREATE is done for *BYPASS*
   2. VERIFY DELETE for *BYPASS*
   3. VERIFY CREATE is done for entered user ID only if password doesn't match
   4. If a valid Passticket or valid MFA user, CREATE would have been successful so VERIFY DELETE is done

4. **If PASSWORDPREPROMPT(NO) is in effect, VERIFY CREATE is called. In fullscreen mode, panel is displayed and then VERIFY is called**

5. **At the time of LOGOFF, TSO does a REQUEST=VERIFY,ENVIR=DELETE to free the ACEE**.

And in Conclusion…

Why do I Care about all of this?

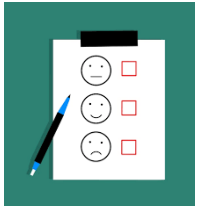
- **Understanding the role of the ACEE is essential in understanding:**
  - The nature of identification, authorization and access control on z/OS
  - The performance impacts of RACF administrative actions
  - The risks involved in allowing nefarious code to manipulate the ACEE

We want your feedback!

- Please submit your feedback online at ….
  - ➢http://conferences.gse.org.uk/2018/feedback/FG

- Paper feedback forms are also available from the Chair person

- This session is FG



GSE UK Conference 2018

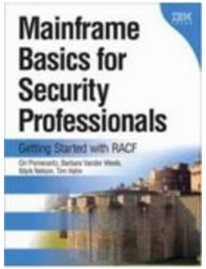
Better, stronger, faster; The Mainframe….. the Machine!

The Life and Times of an ACEE

Session FG

Mark Nelson, CISSP®, CSSLP®

RACF® Design and Development, IBM®


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