IBM Security zSecure V2.4.0
Service Stream Enhancement (SSE)

# *MQ auditing, Command Audit Trail, compliance automation, and other enhancements*
# *IBM Security zSecure Command Verifier User Guide*

# Chapter 1. About this document

This document describes the documentation updates as a result of the Service Stream Enhancement (SSE) for MQ auditing, Command Audit Trail, compliance automation, and other enhancements (APAR numbers OA59807, OA59823, OA59861, and OA59862).

The following enhancements were made for this zSecure V2.4.0 SSE:

- MQ auditing:
  - New report types:
    - MQ_AUTHINFO to report on MQ authentication information objects.
    - MQ_CHLAUTH to report on MQ channel authentication records.
  - The MQ_REGION reports show the following:
    - Authentication information object for user ID and password authentication.
    - Certificates that the queue manager and queue sharing group use.
    - Presence of various switch profiles.
  - The MQ_CHANNEL report type identifies the security exit and the user data that is passed to it, as well as the channel's certification label.
  - The disposition of inbound transmissions has been added to the MQ_INIT reports.
- STIG controls:
  - Automation of more STIG controls: 17 for RACF, 8 for ACF2, and 8 for Top Secret.
  - Equivalents of STIG controls RACF0570 and RACF0580 that allow for password phrases in addition to passwords are provided in the zSecure Extra standard.
  - General improvements for checking general access and logging requirements.
- Command Verifier:
  - Various enhancements have been made to the Command Audit Trail.
  - Multiple commands can now be specified in a pre-command or post-command policy profile.
- Selection on audit and global audit settings are added to the RA.D and RA.R menu options.
- Db2 102 IFCid 106 events (Security parameters at start-up/reload) are now sent to IBM QRadar SIEM and Micro Focus ArcSight.
- Performance improvements are made for ACF2 TRUSTED reporting.
- ICSF settings are added to the IPL parameters report.
- Automatic sensitivities are added, for example, for inaccessible LPA or linklist libraries.
- New fields FALLBACK_DATASET are added to the SENSDSN report type to identify secondary, duplex, or backup RACF data sets.

The documentation updates apply to zSecure V2.4.0. Each of the following links includes a PDF file with the updates for the subject publication: :

- *zSecure CARLa-Driven Components Installation and Deployment Guide*
- *zSecure Messages Guide*
- *zSecure Admin and Audit for RACF User Reference Manual*
- *zSecure Audit for ACF2 User Reference Manual*
- *zSecure Audit for Top Secret User Reference Manual*
- *zSecure CARLa Command Reference*
- *zSecure Command Verifier User Guide*

**Note:**

- The revision bars in the margin indicate updates since publication of Further Automation Of DISA STIG Resource Controls And Other Enhancements (OA59004, OA59006) on April 11, 2020.
- Referenced or linked topics that have not changed are not included in this document. You can find them in the publication that the chapter applies to.
- The *zSecure (Admin and) Audit User Reference Manuals* and the *zSecure CARLa Command Reference* are available to licensed clients only. To access the zSecure V2.4.0 licensed documentation, you must sign in to the IBM Security zSecure Suite Library with your IBM ID and password. If you do not see the licensed documentation, your IBM ID is probably not yet registered. Send a mail to zDoc@nl.ibm.com to register your IBM ID.

## Incompatibility warnings

### STIG members renamed for controls AAMV0410 and AAMV0420

*Table 1. SCKRCARL generic member names renamed for RACF, ACF2, and Top Secret systems*

| Original member name | Renamed - for RACF system | Renamed - for ACF2 systems | Renamed - for Top Secret systems |
|---|---|---|---|
| C2RGM410 | CKAGM410 | C2AGM410 | CKTGM410 |
| C2RGM420 | CKAGM420 | C2AGM420 | CKTGM420 |

### Multiline mixed SBCS/DBCS strings

With previous versions of CARLa and CKGRACF, within a string literal crossing a line boundary, if a line ended with a shift-in (SI) character and an optional space, and if the next line started with a shift-out (SO) character, the SI character, optional space, and SO character were trimmed away by the parser. This trimming behavior has been extended as follows.

Within a string literal crossing a line boundary, if a continuation line starts with an SO character, optionally preceded by Single-byte Character Set (SBCS) space characters, lines immediately preceding this line are trimmed away if they entirely consist of SBCS spaces. Trailing SBCS spaces in the line before these blank lines, if any, are trimmed away as well. If the trimmed line ends with an SI character and the continuation line starts with an SO character, these SI and SO characters are trimmed away, too.

Double-byte Character Set (DBCS) space characters are typically used for non-Roman character languages, like Japanese.

For more information, see "Syntax rules" in *zSecure CARLa Command Reference*.

# Chapter 2. Auditing commands and policy effects

The following sections were updated:

- “Command audit trail” on page 3
- “Structure of =CMDAUD policy profile” on page 3
- “Format of the Command Audit Trail data display” on page 5
- “Internal format of USRDATA entries” on page 9

## Command audit trail

zSecure Command Verifier provides a function to collect and retain additional data about issued commands in the RACF profiles affected by these commands.

For example, if user C4RTEST issues the command **ALTUSER IBMUSER RESTRICTED**, information is saved in the IBMUSER profile. The information includes an indication for the RESTRICTED attribute, the date and time, and the userid C4RTEST. The information that is retained can be used as a Command Audit Trail. The same information can usually be obtained from SMF audit records. However, the zSecure Command Verifier function has the advantage of finding the same information quicker, and without processing potentially large amounts of SMF data. An example of this Command Audit Trail information for a USER, as maintained by zSecure Command Verifier is shown:

```
Command Audit Trail for USER C4RUSER
Profile:          Created on 20.238/14:24 by C4RTEST
Segment:  CICS    Added on 20.241/03:19 by C4RTEST
                  Changed on 20.241/03:20 by C4RTEST
          TSO     Changed on 20.241/03:19 by C4RTEST
Attrib:   PASSWRD Removed on 20.238/14:24 by C4RTEST
          INTERV  Changed on 20.241/04:42 by C4RTEST
          RESTR   Added on 20.238/14:24 by C4RTEST
Connect:          BCSC Added on 20.238/14:24 by C4RTEST
GrpAttr:  ADSP    BCSC Removed on 20.238/14:24 by C4RTEST
```

*Figure 1. Command Audit Trail data for a user*

The data is maintained in the **USRDATA** fields in each profile. The **USRDATA** fields are normally not shown as part of the regular RACF® commands. When appropriate controls are set, the **USRDATA** fields that are used for the Command Audit Trail are shown as part of the various RACF list commands, like **LISTUSER**. The data is displayed following the regular command output.

Because the Command Audit Trail data is maintained in the affected profile, no information is collected, and all existing information is deleted if the profile is deleted.

### Structure of =CMDAUD policy profile

Use these variable details and examples to define =CMDAUD policy profiles that control the Command Audit Trail function.

The basic structure of the =CMDAUD policy profiles includes five separate sections in the following format:

```
C4R.class.=CMDAUD.data-type.profile-identification
```

The *class*, the *data-type*, and the profile itself (*profile-identification*) are used to select which type of Command Audit Trail is collected. The parts of the =CMDAUD policy profiles are described in the following list.

***class***
: This qualifier in the policy profile describes the resource class of the profile as used in or implied by the command.

**=CMDAUD**
This qualifier of the policy profile must be present exactly as shown. If the best matching generic profile for this policy does not contain this qualifier, zSecure Command Verifier searches for a next best policy profile where the class qualifier is represented by a single generic character (*). For an example, see “Examples” on page 4.

***data-type***
This part of the =CMDAUD policy profile can have any of the following values:

**=SEGMENT**
Information about adding, changing, and deleting segments.

Although technically, the MFA data in the USER profile is not kept in a separate segment, modifications to the MFA data are recorded based on the =SEGMENT policy for the Command Audit Trail.

The =SEGMENT policy profile also controls the creation of the "Profile Created" entry.

**=ATTR**
Information about adding and deleting attributes.

**=CONNECT**
Information about adding, changing, and deleting user to group connections.

**=ACL**
Information about use of the **PERMIT** command to manage Access List entries.

**=MEMBER**
Information about adding and deleting members in a grouping resource class profile.

**=MAINT**
Controls display and removal of the Command Audit Trail data.

***profile-identification***
This part of the =CMDAUD policy profile is dependent on the *class* of the target profile. For USER and GROUP profiles, it includes the owner of the profile. For other profiles, it is the resource profile itself.

**USER**
*owner.userid*

**GROUP**
*owner.group*

**resource**
*resource-profile*

### Examples

In this first example of an =CMDAUD policy profile, recording in the Command Audit Trail of changes to a segment of the USER profile IBMUSER, owned by the GROUP SYS1, is controlled through policy

```
C4R.USER.=CMDAUD.=SEGMENT.SYS1.IBMUSER
```

It is also possible to define a more generic policy profile. For example, if you want to activate the Command Audit Trail for all profiles in all resource classes, you can define a policy profile Policy Profile A (**PPA**):

```
PPA:  C4R.*.=CMDAUD.*.**
```

In both of these examples, the required =CMDAUD qualifier is present.

When you want to restrict management of FACILITY class profiles to certain administrators, you probably also define additional policy profiles; for example, Policy Profile B (**PPB**):

```
PPB:  C4R.FACILITY.**
```

However, updating the Command Audit Trail for a PERMIT command to FACILITY profile BPX.SUPERUSER is controlled by the following policy:

```
C4R.FACILITY.=CMDAUD.=ACL.BPX.SUPERUSER
```

The best matching generic profile for this policy is **PPB**. Because this policy profile does not contain the required qualifier =CMDAUD, zSecure Command Verifier bypasses this best matching profile, and instead tries to locate the best matching policy profile for

```
C4R.*.=CMDAUD.=ACL.BPX.SUPERUSER
```

In this example, **PPA** is used instead of the regular best matching profile **PPB**.

## Format of the Command Audit Trail data display

Use this information to understand how to suppress, filter, and interpret Command Audit Trail data.

The example in Figure 1 shows the output of the **C4RCATMN** command. This output is the same as the lines appended at the end of the regular RACF list commands. For the RACF list commands, the information is shown if the user has READ access to the =CMDAUD.=MAINT policy profile. If the RACF list command specifies multiple RACF profiles, the Command Audit Trail information for the specified profiles is shown after all RACF information for all profiles. Examples of such list commands are:

```
   LISTDSD   DA(dsn1,dsn2)
   LISTUSER  (user1,user2)
```

Each Command Audit Trail section is identified by a header line, like:

```
   C4R736I Command Audit Trail for USER user1
```

The Command Audit Trail is not included if the RACF list command specifies a pattern or prefix for the profiles to be shown. Examples of such list command are:

```
   LISTDSD   PREFIX(user1)
   RLIST     FACILITY *
```

If the terminal user has READ access to =CMDAUD.=MAINT policy profile, the Command Audit Trail information is shown. There is no option on the RACF list commands to suppress these additional lines. There are two indirect ways to suppress the Command Audit Trail information:

- Issue the **C4RCATMN** command with the NOMSG keyword. The Command Audit Trail information is no longer shown. It is still possible to show the information by using the **C4RCATMN** command, but it requires a higher authorization than the regular RACF commands need. You can use the **C4RCATMN MSG** command to reactivate showing the Command Audit Trail. The MSG / NOMSG setting is saved across sessions. The initial setting of the MSG / NOMSG setting if you did not issue the **C4RCATMN (NO)MSG** command is MSG.
- Allocate a ddname (`=filename`) with the name C4RNOCAT. This ddname does not need to be allocated to a particular data set, sysout class, or device. The preferred allocation is to DUMMY. The allocation of this ddname is sufficient to suppress display of all Command Audit Trail information as part of the regular RACF list commands. It is still possible to show this information by using the **C4RCATMN** command, although it requires a higher authorization to the =CMDAUD.=MAINT policy profile.

The Command Audit Trail information consists of several sections.

- **The Header**

  Shows the class and profile that is listed.

- **The PROFILE section**

  Contains information about who created the profiles (User, Group, Dataset, General Resource profile). The first line starts with the word `Profile:`, followed by information about when the profile was

created and which user ran the command. It also contains the highest non-zero return code from the pre-, RACF, and post-command.

Collection is controlled by the policy profile

```
C4R.class=CMDAUD.=SEGMENT.profile-identification
```

- **The Segments section**

  Contains the information about the last change to non-base segments. The first line starts with the word *Segment:*, followed by an abbreviated name for the segment. The remainder of the line contains information about the type of change, like add, change, delete, when the change was made, and which user ran the command. It also contains the highest non-zero return code from the pre-command, RACF, and post-command. For modifications to existing segments, only the last change is shown.

  Collection is controlled by the policy profile

  ```
  C4R.class=CMDAUD.=SEGMENT.profile-identification
  ```

  A separate block (add, change, delete) is shown for each segment that was modified. The following segments and pseudo-segment are currently supported.

  **USER**

  | | | |
  |---|---|---|
  | CICS®<br>DCE<br>DFP<br>CSDATA<br>EIM<br>KERB | LANGUAGE<br>LNOTES<br>MFA<br>NDS<br>NETVIEW<br>OMVS | OPERPARM<br>OVM<br>PROXY<br>TSO<br>WORKATTR |

  **GROUP**
  CSDATA, DFP, OMVS, OVM, TME

  **DATASET**
  CSDATA, DFP, TME

  **General Resource**

  | | | |
  |---|---|---|
  | CFDEF<br>CDTINFO<br>CSDATA<br>DLFDATA<br>EIM<br>ICSF<br>ICTX | IDTPARMS<br>JES<br>KERB<br>MFA<br>MFPOLICY<br>PROXY | SESSION<br>SIGVER<br>SSIGNON<br>STDATA<br>SVFMR<br>TME |

- **The Attributes section**

  Contains the attributes and the information about the last change to the attributes. The first line starts with the word *Attrib:*, followed by an abbreviated name for the attribute. The remainder of the line contains information about the type of change such as add or remove, when the change was made, and which user ran the command. It also contains the highest non-zero return code from the pre-, RACF, and post-command. If the profile already has the attribute, a possible *confirmation* command is not shown. The information that is shown reflects the date, time, and ID that changed the profile.

  Collection is controlled by the policy profile

  ```
  C4R.class=CMDAUD.=ATTR.profile-identification
  ```

  A separate block (add, change, remove) is shown for each attribute that was modified. The following attributes are currently supported.

**USER**

| | | |
|---|---|---|
| ADSP<br>AUDITOR<br>CATEGORY<br>CLAUTH<br>DFLTGRP<br>EXPIRED<br>GRPACC<br>INSTDATA<br>INTERVAL | MODEL<br>NAME<br>OIDCARD<br>OPERATIONS<br>OWNER<br>PASSWORD<br>PHRASE<br>RESTRICTED | RESUME<br>REVOKE<br>ROAUDIT<br>SECLEVEL<br>SECLABEL<br>SPECIAL<br>UAUDIT<br>WHEN |

**GROUP**

INSTDATA, MODEL, OWNER, SUPGRP, TERMUACC, UNIVERSAL

**DATASET**

| | | |
|---|---|---|
| ACL<br>AUDIT<br>CATEGORY<br>ERASE<br>FROM | GAUDITINSTDATA<br>LEVEL<br>NOTIFY<br>OWNER | SECLEVEL<br>SECLABEL<br>UACC<br>WARNING |

**General Resource**

| | | |
|---|---|---|
| ACL<br>APPLDATA<br>AUDIT<br>CATEGORY<br>FROM<br>GAUDIT | INSTDATA<br>LEVEL<br>NOTIFY<br>OWNER<br>SECLEVEL<br>SECLABEL | SINGLEDS<br>TVTOC<br>TIMEZONE<br>UACC<br>WARNING<br>WHEN |

- **The Connects section**

  Contains the Groups, the Authorizations, and the UACC together with information about the last change to the connect.

  Collection is controlled by the policy profile

  ```
  C4R.class=CMDAUD.=CONNECT.profile-identification
  ```

  The Connects section is only present for USER profiles. It is not included for GROUP profiles. The first line in this section starts with the word *Connect:*. Each line shows the GROUPNAME, followed by the UACC, the GROUP-Authority, the date and time when the change was made, which user ID executed the command, and the highest non-zero return code from the pre-, RACF and post-command. If both the UACC and the GROUP-Authority have their default value (that is, UACC=NONE and AUTH=USE) their values are not explicitly shown. This makes it easier to spot non-default settings. For more information about the UACC and AUTH settings, see the *RACF Security Administrator 's Guide* and the *RACF Command Language Reference*.

  Because of size limitations, only the last 64 changes to the connect groups are shown.

- **The Group-Attributes section**

  This section immediately follows the Connect section and it contains information about the last change to any GROUP-attribute. The first line starts with the word *GrpAttr:*, followed by an abbreviated name for the attribute.

  Collection is controlled by policy profile

```
C4R.class=CMDAUD.=CONNECT.profile-identification
```

The Group-Attributes section is only present for User profiles. It is not included in Group profiles. The lines show the attribute, followed by the GROUP name, when the change was made, and which user ran the command. It also shows the highest non-zero return code from the pre-, RACF, and post-command. There can be multiple lines for the same attribute, if the attribute was added and removed. The lines for each attribute are in date/time sequence, so the last line reflects the status.

Because of size limitations, only the last 64 changes to the connect groups are shown. The following attributes are currently supported.

```
ADSP, SPECIAL, OPERATIONS, REVOKE, GRPACC, AUDITOR, RESUME
```

- **The Access List section**

  Contains access list entries and the information about the last change to the access list entries. The lines show the access level that was granted, followed by when the change was made, and which user ran the command. It also shows the highest non-zero return code from the pre-, RACF, and post-command. There is only one line for each user, group, or profile. The last instance of granting or removing access is shown. If a user was removed from the access list, the value `Removed` is shown. The special ID **ALL** is used to reflect the use of the RESET keyword on the PERMIT command. Because of size limitations, only the last 64 changes to the access list are collected.

  Collection is controlled by policy profile

  ```
  C4R.class=CMDAUD.=ACL.profile-identification
  ```

- **The Member section**

  Contains members that are part of a grouping class profile. The lines reflect adding or removing entries to and from the member list of grouping class profiles. Each line has one member, followed by when the change was made, and which user ran the command. It also shows the highest non-zero return code from the pre-, RACF, and post-command. There is only one line for each member, reflecting the last action. Because of size limitations, only the last 64 changes to the member list are shown. Also, only the first 128 bytes of the member name are collected and thus included in the display.

  Collection is controlled by policy profile

  ```
  C4R.class=CMDAUD.=MEMBER.profile-identification
  ```

An example for a user profile is shown here:

```
Command Audit Trail for USER C4RUSER
Profile:          Created on 19.238/14:24 by C4RTEST
Segment:  CICS    Added on 19.241/03:19 by C4RTEST
                  Changed on 19.241/03:20 by C4RTEST
          TSO     Changed on 19.241/03:19 by C4RTEST
Attrib:   PASSWRD Removed on 19.238/14:24 by C4RTEST
          INTERV  Changed on 19.241/04:42 by C4RTEST
          RESTR   Added on 19.238/14:24 by C4RTEST
          WHEN    Added on 19.238/14:24 by C4RTEST
Clauth:           USER Added on 19.241/10:04 by C4RTEST
                  TCICSTRN Removed on 19.241/10:05 by C4RTEST
Connect:          C4RGRP1 Added on 19.238/14:24 by C4RTEST
GrpAttr:  ADSP    C4RGRP1 Removed on 19.238/14:24 by C4RTEST
```

*Figure 2. Command Audit Trail data for a user profile*

An example for a data set profile is shown in the following figure. In this example, a DFP segment was added, the profile was placed in WARNING mode, and several access list entries were changed or removed. On 14 September 2019 (19.257) the entire access list was reset by IBMUSER by using the **PERMIT RESET** command.

```
Command Audit Trail for DATASET C4RUSER.**
Profile:          Created on 19.234/09:39 by C4RTEST
Segment:  DFP     Added on 19.245/05:21 by C4RTEST
Attrib:   FROM    DATASET C4RUSER.TEST.** on 19.234/09:39 by C4RTEST
          WARNING Added on 19.245/05:20 by C4RTEST
          AUDIT   SUCCESS Removed on 19.245/08:41 by C4RTEST
                  FAILURES Changed on 19.246/01:30 by C4RTEST
          GAUDIT  SUCCESS Changed on 19.245/09:38 by C4RTEST
                  FAILURES Changed on 19.245/09:38 by C4RTEST
Access:           DATASET C4RUSER.TEST.** access Added on 19.234/09:39 by C4RTEST
                  C4RGRP1 access READ on 19.234/09:39 by C4RTEST
                  C4RGRP2 access READ on 19.234/09:39 by C4RTEST
                  C4RTEST access READ on 19.234/09:39 by C4RTEST
                  SYS1 access READ on 19.234/09:39 by C4RTEST
                  IBMUSER access READ on 19.234/09:39 by C4RTEST
                  * access UPD on 19.234/09:39 by C4RTEST
                  CRMBGUS access Removed on 19.234/09:39 by C4RTEST
                  **ALL** access Removed on 19.257/15:06 by C4RTEST
```

*Figure 3. Command Audit Trail data for a data set profile*

The following example shows the Command Audit Trail information for adding and removing members from a profile in a grouping resource class.

```
Command Audit Trail for GCICSTRN CICSA.SPRO
Member:           CICSA.CEDA Added on 19.249/14:21 by C4RTEST
                  CICSA.CEMT Removed on 19.249/14:21 by C4RTEST
```

*Figure 4. Command Audit Trail data for managing members in a profile in a grouping resource class*

The information about a segment or attribute is presented in date/time sequence. The last line that is shown for a particular segment or attribute is the last recorded action. If an attribute was granted and later removed, the first line shows who granted the attribute and the last line shows who removed the attribute.

For Access List entries and Member Lists, only the last 64 changes are retained. This restriction is mainly for profile size and performance reasons. Only the last action for each ID or member is recorded.

## Internal format of USRDATA entries

The information in this section is only relevant for people who want to inspect the USRDATA entries as maintained by zSecure Command Verifier manually, or who must diagnose problems in these fields.

In each profile, relevant information is kept in multiple USRDATA fields. The USRDATA is accessed as a name-value pair. The USRNAME field describes the information kept in the corresponding USRDATA field. The following USRNAME values are used:

```
$C4RSseg    profile segment seg
$C4RAatt    profile attribute att
$C4RAFRM    from profile class and name
$C4RAAUD    audit settings
$C4RAGAU    globalaudit settings
$C4RCLAU    class authority
$C4RCONN    connect groups
$C4RCatt    connect group attribute att
$C4RPACL    access list
$C4RRMEM    member list
```

The corresponding data fields contain the information in EBCDIC format. The information in these data fields is specific for the profile class. For instance, for USERs, the attribute might be SPECIAL (abbreviated to SPC), while for GROUPS, the TERMUACC attribute might be present (represented by $C4RATRM).

The data fields for each segment or attribute are treated as a block of data that contains multiple statistics. The different events (add, change, remove) for that particular attribute or segment are kept in one statistics block. For the access-list-related field, the last 64 `userid` values are kept together in one block. The format of the data is:

**$C4RS*seg***
This field is used to retain information about one segment. The field has four subfields that are separated from each other by a comma. Information about adding (A), changing (C), and deletion (D) of the segment is separated by a semicolon.

Value BAS in $C4RSBAS usrdata is used to represent the BASE segment of the profile, which is used to record information about the creation of the profile.

The following subfields are present:

**Action**
Character that indicates if information is about adding (A), changing (C), or deleting (D) the segment.

**DATETIME**
10 characters when the command was issued. The format is *yyddd/hhmm*.

**User ID**
Maximum eight-character userid that handled the segment.

**RC**
Two-digit maximum return code of the RACF command or the pre-command and post-command.

An example entry for a TSO segment might be:

```
A,09220/0801,CRMBTST,00;C,09221/0815,IBMUSER,00
```

**$C4RA*att***
This field is used to retain information about attributes that were added or removed from the profile. The field has four subfields that are separated from each other by a comma. Information about different actions is separated by a semicolon. The following subfields are present:

**Action**
Character that indicates if information is about adding (A), changing (C), or deleting (D) the attribute.

**DATETIME**
10 characters when the command was issued. The format is *yyddd/hhmm*.

**User ID**
Maximum eight-character userid that last handled the attribute

**RC**
Two-digit maximum return code of the RACF command or the pre-command and post-command.

An example entry for the Special attribute might be:

```
A,09181/0917,IBMUSER,00;D,09181/0920,IBMUSER,00
```

**$C4RAFRM**
This field is used to retain information about the FROM class and profile of dataset and general resource profiles. The field has five subfields that are separated from each other by a comma. The following subfields are present:

**Class**
Profile class name that was used to create a new profile. This field represents the FCLASS field value of the ADDSD/RDEFINE command.

**Profile-Name**
Profile that was used as a model profile to create a new profile.

**Action**
A character that indicates that the profile is created by using the FROM(F) keyword.

**DATETIME**
10 characters that show when the command was issued. The format is *yyddd/hhmm*.

**User ID**
Maximum eight-character ID of the userid that created the profile by using an explicit model profile.

**RC**
Two-digit maximum return code of the RACF command or the pre-command and post-command.

An example entry might be:

```
XFACILIT,C4RUSER.TEST.**,F,20140/1304,C4RUSER,00
```

**$C4RAAUD**
This field is used to retain information about the audit setting of dataset and general resource profiles. The field has five subfields that are separated from each other by a comma. Information for success and failure auditing is kept in two entries, separated by a semicolon. The following subfields are present:

**SUCCESS or FAILURES**
The audit setting for successful or failed access to the resource.

**Action**
Character that indicates if auditing was added (A), changed (M), or removed (D).

**DATETIME**
10 characters that show when the command was issued. The format is *yyddd/hhmm*.

**User ID**
Maximum eight-character userid that last changed this audit setting.

**RC**
Two-digit maximum return code of the RACF command or the pre-command and post-command.

An example entry might be:

```
SUCCESS,D,19214/0841,C4RTEST,00
```

**$C4RAGAU**
This field is used to retain information about the globalaudit setting of dataset and general resource profiles. The field has five subfields that are separated from each other by a comma. Information for success and failure auditing is kept in two entries, separated by a semicolon. The following subfields are present:

**SUCCESS or FAILURES**
The globalaudit setting for successful or failed access to the resource.

**Action**
Character that indicates if auditing was added (A), changed (M), or removed (D).

**DATETIME**
10 characters that show when the command was issued. The format is *yyddd/hhmm*.

**User ID**
Maximum eight-character userid that last changed this audit setting.

**RC**
Two-digit maximum return code of the RACF command or the pre-command and post-command.

An example entry might be:

```
FAILURES,M,19214/0938,IBMUSER,00
```

**$C4RCLAU**
This field is used to retain information about the class authorization (CLAUTH) of users. Only the last 64 changes are retained in the profile. The field has five subfields that are separated from each other by a comma. Information for different resource classes is separated by a semicolon.The following subfields are present:

**Class**
The class for which the CLAUTH was added or removed.

**UN or DD**
Two placeholder characters that are used to indicate if the CLAUTH was added or removed.

**DATETIME**
10 characters that show when the command was issued. The format is *yyddd/hhmm*.

**User ID**
Maximum eight-character userid that last changed this connect.

**RC**
Two-digit maximum return code of the RACF command or the pre-command and post-command.

An example entry might be:

```
USER,UN,19211/1004,IBMUSER,00
```

**$C4RCONN**
This field is used to retain information about the connection of users to groups. It is kept in the user profile. Only the last 64 changes are retained in the profile. The field has five subfields that are separated from each other by a comma. Information about different connect groups is separated by a semicolon. The following subfields are present:

**Group**
The group to which the user is connected.

**Auth and UACC**
These characters represent the Authority in the group and the UACC for new data sets when the user is logged on using this GROUP as the current connect group. The Authority can be U (use), R (create), C (connect), or J (join). The UACC can be N (none), E (execute), R (read), U (update), C (control), or A (alter).

**DATETIME**
10 characters that show when the command was issued. The format is *yyddd/hhmm*.

**User ID**
Maximum eight-character userid that last changed this connect.

**RC**
Two-digit maximum return code of the RACF command or the pre-command and post-command.

An example entry might be:

```
SYS1,JR,09245/0545,C4RTEST,08
```

**$C4RC*att***
This field is used to retain information about the group attributes of users. It is kept in the user profile. Only the last 64 changes are retained in the profile. The field has five subfields that are separated from each other by a comma. Information about different connect groups is separated by a semicolon. The following subfields are present:

**Group**
The group to which this attribute applies.

**Action**
Character that indicates if information is about adding (A) or deleting (D) the attribute.

**DATETIME**
10 characters when the command was issued. The format is *yyddd/hhmm*.

**User ID**
Maximum eight-character userid that last changed this connect.

**RC**
Two-digit maximum return code of the RACF command or the pre-command and post-command.

An example entry might be:

```
SYS1,A,09245/0550,C4RTEST,00;SYS1,D,09245/0555,C4RTEST,00
```

**$C4RPACL**

This field is used to retain information about the access list of data sets and general resource profiles. Only the last 64 changes are retained in the profile. The field has five subfields that are separated from each other by a comma. Information about different users/groups in the access list is separated by a semicolon. The following subfields are present:

**User ID or Class/Profile**
The access list entry, which can be a RACF user ID, group ID, an asterisk, class/profile, or the special value &RACUID.

**Access level**
Character for the access level granted: N(one), E(xecute), R(ead), U(pdate), C(opy), A(lter), D(elete), or F(rom).

The F character indicates that the access list of the profile is copied from the profile that is mentioned in the FROM(F) keyword.

**DATETIME**
10 characters when the command was issued. The format is *yyddd/hhmm*.

**User ID**
Maximum eight-character userid that last changed this access list entry

**RC**
Two-digit maximum return code of the RACF command or the pre-command and post-command.

An example entry might be:

```
IBMUSER,R,09245/0545,C4RTEST,00;DATASET/C4RUSER.TEST.**,F,20156/0036,C4RTEST,00
```

**$C4RRMEM**

This field is used to retain information about the member list for profiles in a grouping resource class. Only the last 64 changes are retained in the profile. The field has four subfields that are separated from each other by a comma. Information about different members is separated by a semicolon. The following subfields are present:

**Member**
The member name. This name usually has the format of a profile in the corresponding member (non-grouping) class.

**Action**
Character that indicates if information is about adding (A) or deleting (D) the member.

**DATETIME**
10 characters when the command was issued. The format is *yyddd/hhmm*.

**User ID**
Maximum eight-character userid that last added or removed this member.

**RC**
Two-digit maximum return code of the RACF command or the pre-command and post-command.

An example entry might be:

```
'SYS1.LINKLIB'//NOPADCHK,A,09249/1419,C4RTEST,00
```

or

```
TEST.CEMT,A,09249/1421,C4RTEST,00;TEST.CEDA,A,09249/1421,C4RTEST,00
```

# Chapter 3. Policy profiles

The following section was updated. For a list of the supported variables, see the *zSecure Command Verifier User Guide*.

**RACF command replacement**

The last paragraph for C4R.*command*.=PSTCMD.keyword-qualification was updated:

The APPLDATA of =PRECMD and =PSTCMD profiles can be used to specify the command that is to be run before and after the original RACF command. The Command Verifier policy can specify multiple commands, that are separated by a semicolon (;), up to 255 characters. Command Verifier executes each command in the order that it is provided before moving on to the next command. If a command fails, all the following commands are not executed. Because of the way that RACF handles the **APPLDATA** field, the value that is entered is folded to uppercase. In the specified command string, variables can be used to refer to parts of the original RACF command. Variables are prefixed by an ampersand (&) sign.