

IBM Security zSecure V2.3.1 Service Stream Enhancement

*Enhancements for zSecure Command
and Ticket Logging (OA56705,
OA56718, OA56801, and OA58254)
Documentation updates*

IBM

IBM Security zSecure V2.3.1 Service Stream Enhancement

*Enhancements for zSecure Command
and Ticket Logging (OA56705,
OA56718, OA56801, and OA58254)
Documentation updates*

IBM

Chapter 1. About this document

This document describes the documentation updates as a result of the IBM Security zSecure Command and Ticket Logging Service Stream Enhancement (SSE - OA56705, OA56718, OA56801, and OA58254). All the documentation updates apply to V2.3.1 zSecure Admin and zSecure Command Verifier.

The following publications are updated:

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

Note: Referenced topics that have not changed are not included in this document. You can find them in the publication that the chapter applies to.

Incompatibility warning

This zSecure Service Stream Enhancement (SSE) provides support for a new function. The zSecure ISPF user interface has new fields to specify ticket identifiers and descriptions. In the original SSE, this support is activated automatically. If PTF UJ00783 (APAR OA58254) has been applied as well, the new support is activated only if profile CKR.CKXLOG.** in the configured resource class (default XFACILIT) has been defined.

Individual functions related to command logging can be controlled by creating more specific profiles. Creating a CKR.CKXLOG.** profile with UACC(READ), and not defining more specific profiles, has the following effects:

- It changes the command confirmation level (as specified in SE.4) for all zSecure Admin users.
- It enforces that a ticket ID and description are entered for every generated command.

In the absence of PTF UJ00783, the same situation might also arise from a top generic profile, like CKR.**

If PTF UJ00783 was applied, and the CKR.CKXLOG.** profile is not defined, the new fields and options are not available in the zSecure ISPF user interface. Other parts of the zSecure Command Logging application are not affected by the presence or absence of the CKR.CKXLOG.** profile.

The zSecure ISPF user interface enforces new prompts if a matching profile is found for resource CKR.CKXLOG.ID.PROMPT. If you have a matching profile (for example CKR.** or CKR.CKXLOG.**), but do not want to activate the additional prompting, define profile CKR.CKXLOG.ID.PROMPT with a UACC(NONE), and do not PERMIT groups or users on this profile.

The zSecure ISPF user interface displays fields allowing a ticket ID and description to be entered. If you have a matching profile (for example CKR.** or CKR.CKXLOG.**), but do not want to display these fields, define profile CKR.CKXLOG.ID.SHOW with a UACC(NONE), and do not PERMIT groups or users on this profile.

Chapter 2. zSecure CARLa-Driven Components Installation and Deployment Guide

This chapter lists the documentation updates for *zSecure CARLa-Driven Components Installation and Deployment Guide* as a result of the zSecure Command and Ticket Logging SSE.

Chapter 8. Setup for production

Added: section **Setup of zSecure Admin Command Logger**.

Appendix B. Security setup for zSecure

The intro of Appendix B was updated. See **Security setup for zSecure**.

In section “**Resources that configure which options are shown**”, a resource name was added to the list of resource names for the main panel:

CKR.OPTION.CR for COMMAND REVIEW

The following section was added: **Other ISPF interface controls**.

Setup of zSecure Admin Command Logger

zSecure™ Admin provides an option to log all RACF commands that are issued through the zSecure interface in a central repository. This section provides an overview of the functions and describes the required installation and activation steps.

About this task

The command log records are kept in a z/OS log stream. Using zSecure functions, it is possible to select and report about the recorded RACF commands. The z/OS log stream must be defined in the LOGR policy data set. The actual recording of the commands in the log stream is done through a started task (CKXLOG) that runs on each system (LPAR) where RACF commands are run. The record of the command can be annotated with a ticket identification and a ticket description. The zSecure Admin ISPF interface and zSecure Command Verifier send the commands to the zSecure Admin Command Logger.

When zSecure Admin is installed and it is verified that it is working properly, the following steps are required for the setup of the zSecure Admin Command Logger function:

- Step : Define a log stream (or log stream model) for use by the Command Logger
- Step : Update C2R\$PARM to specify a value for CKXCUST.
- Step : Update the CKXCUST member CKXPARM for installation parameters.
- Step : Define the STC user ID and STARTED profile for the CKXLOG started task.
- Step : Copy the STC procedure to your STC proclib.
- Step : Authorize the STC user ID to allocate and write the log stream.

- Step : Start the CKXLOG started task.
- Step : Define the Command Log policies for components and users.
- Step : Configure the zSecure Admin ISPF user interface for command logging.
- Step : Configure zSecure Command Verifier to ensure logging of commands that are issued in a batch job or from outside the zSecure Admin ISPF interface.

In the above steps, you define new profiles to activate and control various parts of the zSecure Admin Command Logger. The following picture shows a global overview of the flow of a log record, indicating where these different profiles are used:

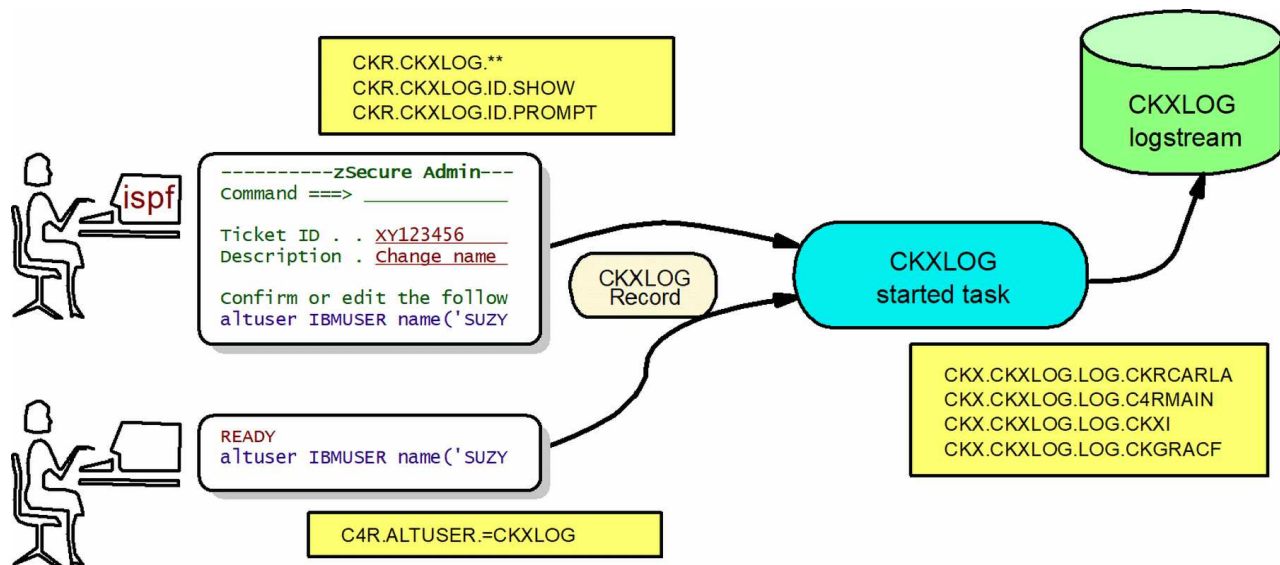


Figure 1. Global overview of the flow of a log record

The CKXLOG started task uses resources that start with CKX, as shown in the rectangular box. Access to these resources is used to determine if a log record is written to the logstream or not. Step 8 on page 8 describes these resources. The first administrator uses the zSecure Admin ISPF interface to enter changes. The ISPF interface uses the `CKR.CKXLOG.**` profile and resources that start with CKR as shown in the associated rectangular box. The presence and access of these profiles and resources determine if the zSecure Admin ISPF interface passes command records to the CKXLOG started task. Step 9 on page 8 describes these resources.

The second administrator enters a command directly without using the zSecure Admin ISPF interface. These commands might be intercepted by zSecure Command Verifier. In that case, logging is controlled through resources that start with C4R, of which a sample is shown in the associated rectangular box. Step 10 on page 9 describes these resources.

Note that commands that are executed through the zSecure Admin ISPF interface might also pass through zSecure Command Verifier, and might thus be passed to the CKXLOG started task twice: once by zSecure Admin, and once by zSecure Command Verifier. It is the CKXLOG started task that determines whether or not the passed CKXLOG records are actually recorded in the CKXLOG log stream.

Depending on your environment, you might need profiles for one or more of these types of resources, using generic profiles where possible.

Procedure

1. Define a log stream (or log stream model) for use by the Command Logger. The zSecure Admin Command Logger writes the command records to a system-defined log stream. The CKXPARM member defines the name of the log stream. Before a log stream can be used, it must be defined in the system LOGR policy data set using the IXCMIAPU utility. The zSecure Command Logger supports two methods for defining the log stream:
 - The log stream is defined by the systems programmer using the exact name that is specified in CKXPARM.
 - The CKXLOG program itself defines the log stream, based on a predefined log stream model.

The second method is preferred. The log stream must be defined once and can then be used on all systems that share that LOGR policy data set. The CKXLOG started task tries to open the log stream as it is defined in the CKXPARM. If the log stream does not exist, CKXLOG defines the log stream using the specified (or default) log stream model.

The log stream can be defined as a DASDONLY log stream; the log stream is dedicated to a single system. It is recommended to use &SYSNAME as its first qualifier. The log stream name is specified in the CKXPARM that you create or update in step 3. Member CKXZLGST in SCKRSAMP is provided as an example for defining a DASDONLY model log stream.

It is also possible to define a COUPLING FACILITY log stream; the log stream is shared between multiple systems. Member CKXZLGCF in SCKRSAMP is provided as an example for defining a shared model log stream.

When defining the log stream model, you must consider the retention period and the size of the stream. These are largely dependent on your installation policy and the number of commands issued.

2. Update C2R\$PARM to specify the value for CKXCUST. The CKXLOG procedure refers to include member C2R\$PARM. C2R\$PARM is a sample member that contains JCL SET statements and it is usually referred to as the zSecure configuration. In the zSecure configuration file, the variable CKXCUST is used to name the configuration file for the CKXLOG procedure

```
// SET CKXCUST='your.prefix.CKRPARM'
```

You normally substitute *your.prefix.CKRPARM* with the installation CKRPARM data set, but you can specify any partitioned data set with record format FB, and a logical line length of 80. Position 73 to 80 in the records are ignored. This data set must contain the member indicated by the XPARM parameter in the JCL procedure.

3. Update the CKXCUST member CKXPARM for installation parameters. The CKXPARM member in the CKXCUST data set can be used to override the default values for certain parameters. The parameters are set using the SETUP statement. The following SETUP statements are supported:

SETUP L\$NAME(log-stream-name)

The log-stream-name specifies the name of the log stream that the CKXLOG started task is to use. The qualifier CKXLOG is used to identify the log stream as related to the Command Logger application. It is suggested that the first qualifier indicates the system or sysplex. The log-stream-name can contain system symbols, like &SYSNAME or &SYSPLEX. The default value is &SYSPLEX..CKXLOG

SETUP L\$MODEL(model-log-stream-name)

The model-log-stream-name specifies the name of the model that the CKXLOG started task is to use if the log stream that the L\$NAME

specifies does not exist. The qualifier CKXLOG is used to identify the model log stream as related to the Command Logger application. The suggested value for the first qualifier is MODEL. The log-stream-name can contain system symbols, like &SYSNAME or &SYSPLEX. The default value is MODEL.CKXLOG.

SETUP TICKETEXPIRE(unused-time)

The CKXLOGID command can be used to set a ticket ID and description. The CKXLOGID command can also be used to clear such a setting. If a ticket ID has not been referenced, it can be automatically removed to free up associated resources. The unused-time specifies the period (HHMM) after which an unreferenced ticket ID is removed from the CKXLOG started task. The default value is 0100 (1 hour).

Update the CKXPARM member to specify values that are appropriate for your environment.

4. Define the STC user ID and STARTED profile for the CKXLOG started task. If you want to use a dedicated user ID for the CKXLOG started task, you must define the user ID. The user ID and group ID can be assigned to the started task through the STDATA segment of the applicable STARTED profile. The user ID does not need to have any non-standard attributes.
5. Copy the STC procedure to your STC proclib. The example CKXLOG procedure must be available in one of the system STC proclibs. Copy and adapt the CKXLOG procedure from the SCKRPROC library.
6. Authorize the STC user ID to allocate and write the log stream. The CKXLOG started task writes record to the specified log stream. If the log stream does not exist, it defines the log stream. For these actions, the STC user ID needs sufficient authorization. Use the following table for the required authorization:

Table 1. Authorization of the STC user ID

Action	Type of log stream	Required authorization	Resource class	Resource name
DEFINE	DASDONLY	ALTER	LOGSTRM	<i>log_stream_name</i>
	COUPLING FACILITY	ALTER	LOGSTRM	<i>log_stream_name</i>
		UPDATE	FACILITY	<i>IXLSTR.model_structure_name</i>
CONNECT		UPDATE	LOGSTRM	<i>log_stream_name</i>

7. Start the CKXLOG started task. The zSecure Admin Command Logger task is started from the operator console by a START command. The command executes the procedure from the applicable system proclib. It is possible to specify startup parameters. These parameters can be given at the START command itself. An example of such a START command is:

```
S CKXLOG,,DEBUG
```

The following START parameters are supported:

DEBUG

Specifies that diagnostic messages must be issued during startup and running of the CKXLOG task. This setting is in effect until a subsequent NODEBUG command is issued from the operator console.

FORCE

Specifies that, regardless of a previous execution, initialization must continue. Only use the FORCE option if you cannot start the zSecure Command Logger normally. The FORCE option might enable starting

the zSecure Command Logger without the need to IPL the system. During normal operation, the FORCE option is never required.

DEBUG-FORCE

Specifies that both the DEBUG and FORCE options must be active at startup.

To terminate the started task, the console operator can issue the STOP command. z/OS supports the use of the P command as an alias of the STOP command; for example:

```
P CKXLOG
```

The STOP command can also be issued as the parameter on the MODIFY command. z/OS supports use of the F command as an alias of the MODIFY command. An example of such a command is:

```
F CKXLOG,STOP
```

While the CKXLOG started task is active, the console operator can also issue commands to modify the zSecure Command Logger. These commands can be issued through the MODIFY console command. For example:

```
MODIFY CKXLOG,DEBUG
```

The following list describes the zSecure Command Logger operator commands that can be used in the MODIFY console command:

ABND

This command causes an immediate abend of the CKXLOG started task, with recovery. It is intended for IBM internal testing only.

CRSH This command causes an immediate abend of the CKXLOG started task, without any recovery. It is intended for IBM internal testing only.

DEBUG

This command controls the diagnostic and monitoring messages that the program can generate; the zSecure Messages Guide describes all messages. The command is effective immediately. The DEBUG command does not support any additional keywords.

NODEBUG

Deactivates issuing of diagnostic and monitoring messages. The NODEBUG command does not support any additional keywords.

SIPL This command is used during the migration to a new release. It results in freeing all in-memory data structures, loss of a system level linkage index (LX), and marking the address space as non-reusable. System level LX's are a limited resource and cannot be recovered without an IPL of the system. When upgrading from one release of the zSecure Command Logger to another, the installation instructions require that you shut down the previous version using this SIPL command. The SIPL command does not support any additional keywords.

STOP Stop execution of the zSecure Command Logger started task. This results in an orderly shutdown of the task. Some memory remains reserved after termination of the task, to enable reuse of some critical system resources during a subsequent restart of the started task. The effect of the STOP MODIFY command is identical to that of the z/OS STOP command. The STOP command does not support any additional keywords.

8. Define the Command Log policies for zSecure components. The zSecure Admin Command Logger checks if commands that are issued through supported zSecure components must be logged. In the absence of any profile, commands are not logged. To activate command logging, you must define one or more profiles. These switch profiles have the following format: CKX.CKXLOG.LOG.*component-name*, where *component-name* can be one of the following options:

CKRCARLA

The zSecure common program.

CKXI

zSecure Admin Command Execution module CKX. This is used for queued commands and for commands that are issued through the zSecure Server.

CKGRACF

The zSecure Admin RACF special functions program.

C4RMAIN

zSecure Command Verifier.

CKXLOGID

Special logging program when testing the LOG function of the Command Logger.

You can use a generic profile to match these various component names.

When commands are issued using the zSecure Admin interface, the same command can be logged multiple times, because it passes multiple components. By defining these profiles, you can control for which components the commands must be logged and for which components logging the commands must be suppressed. In the absence of any profile, commands are not logged. To activate logging, the user that issues the command must have at least READ access to the resource, for example:

```
CKX.CKXLOG.LOG.* ID(*) ACCESS(READ)
```

9. Configure the zSecure Admin ISPF user interface for command logging.

Command logging from the zSecure Admin ISPF panels is enabled by defining a profile CKR.CKXLOG.**. Individual functions related to command logging can be controlled by also creating more specific profiles as described in subsequent paragraphs. Creating this CKR.CKXLOG.** profile with UACC(READ), and not defining more specific profiles, changes the command confirmation level (as specified in SE.4) for all zSecure Admin users, and enforces that a ticket ID and description are entered for every generated command.

To allow users to specify ticket information, the following resource is used: CKR.CKXLOG.ID.SHOW. Users with at least READ access to the matching XFACILIT profile can use the CKXLOGID primary ISPF command and can specify or confirm ticket information during command confirmation. Command confirmation is active according to the user specified setting in SETUP CONFIRM (SE.4). The matching profile for this resource can be generic, for example the CKR.CKXLOG.** profile.

To require that a user fills in the ticket ID and description, the following resource is used: CKR.CKXLOG.ID.PROMPT. Users with at least READ access to the matching XFACILIT profile always see a command confirmation panel when commands are generated, irrespective of the value that is specified in SETUP CONFIRM (SE.4 or SE.D.4). They must supply ticket information for the generated commands, optionally through the CKXLOGID primary ISPF

command. The matching profile for this resource can also be generic, for example the CKR.CKXLOG.** profile. It is expected that most organizations will define a discrete profile with UACC(NONE) and will not require all users to specify or confirm the ticket ID and description for every generated command.

Note: Additional logging of RACF commands can be provided by zSecure Command Verifier, independent of the command logging functions of zSecure Admin. See the step for setup instructions. Installations can decide to log through the zSecure Admin interface, through zSecure Command Verifier, or through both logging mechanisms. Logging is finally controlled by the user's access to CKX.CKXLOG.component profiles, as described in step .

10. Configure zSecure Command Verifier to ensure logging of commands that are issued in a batch job or from outside the zSecure Admin ISPF interface.

You can use zSecure Command Verifier to record most RACF commands. To activate this function, you must define one or more profiles in the XFACILITY class. The profiles have the following format: C4R.command.=CKXLOG, where *command* is the unabbreviated RACF command. Generics can be used, except for the =CKXLOG qualifier, which must be present exactly as shown. zSecure Command Verifier logs all RACF commands through the zSecure Admin Command Logger for users that have at least READ access to the applicable profile. To suppress warning messages due to missing ticket information, or an inactive CKXLOG server, users must have at least UPDATE access to the C4R.command.=CKXLOG resource.

For more information about zSecure Command Verifier, see *zSecure Command Verifier User Guide*.

Security setup for zSecure

Use these guidelines to help set up security for zSecure.

The APF-authorized functions of zSecure are protected by resources in the XFACILIT class, unless you changed this setup. (See Site module for instructions).

zSecure uses SAF to configure menus, action commands, and to determine what data (profiles, rules, SMF records) users are allowed to see:

- To activate these controls, you can create catchall (backstop) profiles or rules for resources with high level qualifiers CKF, CKG, CKR, C2R, and C2X.
- To disable all functions for most users, specify default access NONE. Grant READ access only to those people entitled to use all functions of zSecure.
- To activate use of the zSecure Admin Command Logging function, define the profile CKR.CKXLOG.** and profiles for the resources as described in "Other ISPF interface controls" on page 10.

All these resources use the XFACILIT resource class or the resource class that is configured in your CKRSITE module (see Site module).

To enhance performance, zSecure does not always read the entire RACF database. To ensure that scope-related reports are correct, ensure the following:

- CKGRACF scope profiles must start with non-generic characters for the CKG qualifier.
- RACF delegated authority profiles must start with non-generic characters for the IRR qualifier.

For further refining, see the remaining information in this Appendix.

Other ISPF interface controls

This section lists additional resources that control the way that the ISPF interface behaves:

CKR.CONTROL.MASK

This resource controls the capability of users to ask for all profiles (no mask or name). If a matching profile is defined, and the user does not have at least READ access, most queries in the ISPF interface must specify at least three non-generic characters in the mask for the RACF profile or in an other query-specific field on the selection panel. An example is the RA.U query that requires at least three non-generic characters in the mask for the Userid or Name field. This resource forces the use of the 31-bit version of the CKRCARLA program in stead of the 64-bit version. It also forces the use of the **Use IO in preference to storage** option in **SE.0 Run Options**. The control function is inactive for users with READ access or higher.

CKR.CKXLOG.**

This profile enables the zSecure Admin Command Logger functions in the zSecure Admin ISPF user interface. If this profile is defined, the CKR.CKXLOG.ID.SHOW and CKR.CKXLOG.ID.PROMPT resources can be used to control the individual command logger functions. In the absence of this profile, the command logger functions are not available in the zSecure Admin ISPF environment.

CKR.CKXLOG.ID.SHOW

This resource controls which users can use the CKXLOGID primary command to set ticket information. READ access to this resource also activates use of the ticket ID and description in the command confirmation panels. This resource can be used to selectively implement the use of tickets to annotate commands that are logged using the zSecure Admin Command Logger (CKXLOG).

CKR.CKXLOG.ID.PROMPT

This resource controls which users must use the CKXLOGID primary command or specify ticket information in the zSecure Admin command confirmation panels. The control is used for all users that have at least READ access to the matching profile. This resource can be used to selectively require the use of tickets to annotate commands that are logged using the zSecure Admin Command Logger (CKXLOG).

Chapter 3. zSecure Messages Guide

This chapter lists the documentation updates for *zSecure Messages Guide* as a result of the zSecure Command and Ticket Logging SSE.

The following messages were added or updated:

- **CKR messages**
- **CKX messages from 100 to 199**
- **CKX messages from 200 to 299**
- **CKX messages from 300 to 399**
- **CKX messages from 700 to 799**
- **CKX messages from CKX900 to CKX999**
- **C4R messages**

CKR messages

CKR0467 *operation not allowed with format field name at ddname line number*

| **Explanation:** This message indicates that zSecure
| found a manipulation operation that is specified for a
| field value that does not have the right format. An
| EXTRACTDN operation can be applied only to a
| distinguished name in X.509-DN format. Some
| CONVERT operations only work on a particular date
| format. The other operations can be applied only to
| character-format fields.

Note: When field value operation functions are nested,
the format that is shown might be an intermediate
result. For example, the resulting format after
EXTRACTDN is Char, so the function can be applied
only once.

Severity: 12

CKR1415 **ALLOC TYPE=ACCESS file skipped
because not licensed - ddname volume dsn**

| **Explanation:** An ALLOC TYPE=ACCESS statement for
| the indicated data set is ignored because both zSecure
| Admin and zSecure Audit for RACF® are not installed
| or have been disabled in IFAPRDxx.

Severity: 00

CKR1593 **Unrecognized SLFN operation hex to
format (description) on field name source**

| **Explanation:** The field manipulation operation syntax
| checker encountered an unknown type of DEFINE ...
| CONVERT operation. The nested operation syntax
| checking stops and the manipulation operation is
| allowed. The return code 24 during parse stops the

| CARLa query from executing, unless the message is
| suppressed.

| **User response:** See the Electronic Support Web site for
| possible maintenance associated with this message. If
| you cannot find applicable maintenance, follow the
| procedures described in Contacting IBM® Support to
| report the problem.

| **Severity:** 24

CKR1594 **Unrecognized SLFN operation hex
(description) on field name source**

| **Explanation:** The field manipulation operation syntax
| checker encountered an unknown type of field
| manipulation operation. The nested operation syntax
| checking stops and the manipulation operation is
| allowed. The return code 24 during parse stops the
| CARLa query from executing, unless the message is
| suppressed.

| **User response:** See the Electronic Support Web site for
| possible maintenance associated with this message. If
| you cannot find applicable maintenance, follow the
| procedures described in Contacting IBM Support to
| report the problem.

| **Severity:** 24

CKR2057 **ALLOC TYPE=CKXLOG file skipped
because not licensed - ddname volume dsn**

| **Explanation:** An ALLOC TYPE=CKXLOG statement
| for the indicated data set is ignored because zSecure
| Admin is not installed or has been disabled in
| IFAPRDxx.

| **Severity:** 04

| **CKR2058** Started [re-]read of TYPE=CKXLOG
| [PADS] file *source*

| **Explanation:** The engine has successfully opened the
| indicated command-log file and has started reading.

| **Severity:** 00

| **CKR2059** Read *number* records [with *errornum*
| errors] from TYPE=CKXLOG file *source*

| **Explanation:** This message indicates how many
| command-log records have been read from the
| indicated file.

| **Severity:** 00

| **CKR2061** Excessive errors in TYPE=CKXLOG
| input; skipping rest of file *source*

| **Explanation:** There were already more than a
| threshold of five errors found in the TYPE=CKXLOG
| input file. The rest of the file is skipped.

| **Severity:** 16

| **CKR2063** Non-PADS access required to read
| CKXLOG data, skipping *source*

| **Explanation:** The program does not support scoping;
| that is, restricting who can see which command-log
| records, based on the user's scope of control. The field
| content is skipped.

| **Severity:** 08

| **CKR2064** Unexpected short CKXLOG record *recno*
| in file *source*

| **Explanation:** An unsupported record length was
| found at the indicated record number. The record is
| skipped and processing continues until a limit is
| reached and CKR2061 is issued.

| **Severity:** 16

| **CKR2065** Unsupported CKXLOG file format *hex*
| record *recno* for file *source*

| **Explanation:** An unsupported record format was
| encountered at record *recno* in the indicated file. The
| rest of the file is skipped.

| **User response:** Check whether this is actually a
| TYPE=CKXLOG file, and adapt allocation to not
| include the data set.

| **Severity:** 16

| **CKR2067** ALLOC TYPE=CKXLOG ACTIVE
| requested but CKXLOG is not active -
| skipped

| **Explanation:** The CKXLOG started task is not active
| or not returning an active log stream name. The
| allocation request is suppressed.

| **Severity:** 04

| **CKR2068** Annotating system *system1* with complex
| *complex version* default *system2* because
| no CKFREEZE file matching *dsn* or *path*

| **Explanation:** This informational message is issued
| when a CKXLOG record contained *system1*, no
| CKFREEZE was found for this system, and SIMULATE
| ACCESS_FALLBACK_DEFAULT was specified to direct the
| selection of the default system *system2* from the
| complex. It is only issued when a COMPLEX= parameter
| is present on an ALLOC TYPE=CKXLOG statement.

| **Severity:** 00

| **CKR2069** Annotating system *name1* with default
| *name2* because no CKFREEZE file
| matching *ddname* *volser* *dsn*

| **Explanation:** CKXLOG records were read that
| contained a system SMF ID for which there is no
| CKFREEZE file present in the set of input files. Because
| the SIMULATE ACCESS_FALLBACK_DEFAULT
| statement was specified, the settings from the default
| system are used for these CKXLOG records.

| **User response:** No action required.

| **Severity:** 00

| **CKR2070** No CKFREEZE file found for system
| SMFid in file *ddname* *volser* *dsn*

| **Explanation:** CKXLOG records were read that
| contained a system SMF ID for which there is no
| CKFREEZE file present in the set of input files.

| **User response:** Connect a CKFREEZE file for the
| indicated system. If no such CKFREEZE exists, for
| example, because the CKXLOG file has been modified,
| you can use the SIMULATE
| ACCESS_FALLBACK_DEFAULT statement to use
| settings from the default system.

| **Severity:** 08

| **CKR2071** Annotating system *system* with
| COMPLEX/VERSION *complex1* *version1*
| instead of *complex2* *version2*

| **Explanation:** The message warns that a COMPLEX= or
| VERSION= parameter was not honored for a
| TYPE=CKXLOG file because that would result in
| lookup fields being empty. Instead, the indicated

| database (complex and version) is used.
 | **User response:** Verify that this is what you meant to
 | do.
 | **Severity:** 00

| **CKR2072** **Complex name used for system *smfid***
 | **records in *ddname volser dsn***

| **Explanation:** This message is issued once for each
 | system ID *smfid* that the user is allowed to see in each
 | CKXLOG input file processed. The message is issued to
 | help you understand unexpected failures. For example,
 | when using lookups with incomplete sets of input or
 | user-specified complex names, the message identifies
 | which complex (RACF database) the lookup uses. In
 | this message, the complex name is followed by the
 | VERSION if VERSION is specified in the ALLOC
 | command.

| **Severity:** 00

| **CKR2076** **Unexpected record order in CKXLOG**
 | **input merge found:**
 | **ETOD hex record *recno1* in file *ddname1***
 | ***source1***
 | **ETOD hex record *recno2* in file *ddname2***
 | ***source2***

| **Explanation:** During the input file merge for files
 | identified by ALLOC TYPE=CKXLOG, an unexpected
 | sort order was found. The message identified the two
 | records that had the unexpected order. If they are from
 | different files, either of those files could be out of order.

| **Severity:** 04

| **CKR2156** **No storage left for COMMON_PREFIX,**
 | **processing SUMA *address* prefix length**
 | ***length* prefix *prefix***

| **Explanation:** An out of storage condition occurred
 | while processing a COMMON_PREFIX (or CPRX)
 | modifier. Summary processing at this level stops; even
 | if the message is suppressed.

| **Severity:** 08

| **CKR2157** **No storage left for COMMON_PREFIX**
 | **work buffer, processing SUMA *address***
 | ***prefix* length *length* prefix *prefix***

| **Explanation:** An out of storage condition occurred
 | while processing a COMMON_PREFIX (or CPRX)
 | modifier. Summary processing at this level stops; even
 | if the message is suppressed.

| **Severity:** 08

CKX messages from 100 to 199

| **CKX100A** **zSecure Admin Command Logger**
 | **inactive**

| **Explanation:** The zSecure Admin Command Logger
 | started task stopped.

| **CKX101E** **Error return code from CKXCLEAN**
 | ***retcode***

| **Explanation:** This message represents an internal
 | error. See the Electronic Support Web site for possible
 | maintenance associated with this message. If you
 | cannot find applicable maintenance, follow the
 | procedures described in Contacting IBM Support to
 | report the problem. Create a problem record and
 | provide the complete list of error messages.

| **CKX102W** **FRR routine invoked for ABEND *xxx***

| **Explanation:** An ABEND occurred in the CKXPCRTN
 | module that is used as interface between the command
 | intercept and the log stream writer. The ABEND was
 | recovered and execution continued.

| **CKX103E** **IKJSCAN *retcode*=*retcode***

| **Explanation:** An unexpected error occurred during
 | processing the CKXPARM statements. The program
 | stops.

| **CKX104I** **Received STOP command**

| **Explanation:** This message is issued in response to the
 | operator STOP command.

| **CKX105I** **Received command *command-string***

| **Explanation:** This message is issued in response to the
 | operator MODIFY command.

| **CKX106I** **Received STOP command**

| **Explanation:** This message is issued in response to the
 | operator MODIFY command to STOP the CKXLOG
 | started task. A normal stop is performed, allowing
 | reuse of system resources.

| **CKX107I** **Received SIPL command**

| **Explanation:** This message is issued in response to the
 | operator MODIFY command to STOP the CKXLOG
 | started task using total removal mode. Program
 | cleanup also removes the CKXC control block in ECSA
 | and the pointer to it. This results in loss of the system
 | LX that was used for the current instance of the started
 | task.

| **CKX108E** **Crashing without ESTAEX**

| **Explanation:** The CRSH command is processing. The

CKX109E • CKX118I

| CRSH command is intended only for IBM internal testing purposes.

CKX109E CKXLOG must run as started task

| **Explanation:** The zSecure Admin Command Logger program (CKXLOG) must run as a started task. Running these programs as part of a batch job is not supported. Execution is terminated.

CKX110E Error return from IEANTDL, RC=*retcode*

| **Explanation:** The IEANTDL service failed with a return code *retcode*.

| **User response:** Restart the task with the debug option. See the Electronic Support Web site for possible maintenance associated with this message. If you cannot find applicable maintenance, follow the procedures described in Contacting IBM Support to report the problem. Create a problem report including the output of the started task.

CKX111I Removed Active Name/Token Pair

| **Explanation:** This debug only message is issued as progress indicator for the program cleanup process.

CKX112I IEAVRLS RC=*retcode*

| **Explanation:** This debug only message is issued as progress indicator for the program cleanup process.

CKX113E Error return from IFAEDDRG, RC=*retcode*

| **Explanation:** A problem occurred with the deregistration during the termination of the program.

| **User response:** This situation should not occur. See the Electronic Support Web site for possible maintenance associated with this message. If you cannot find applicable maintenance, follow the procedures described in Contacting IBM Support to report the problem.

CKX114I IFAEDDRG for *product-name* succeeded

| **Explanation:** This debug only message is issued as progress indicator for the program cleanup process.

CKX115I Purge Latch Set token *latchsettoken*

| **Explanation:** This debug only message is issued as progress indicator for the program cleanup process.

CKX116I Return from purge latch set, RC=*retcode*

| **Explanation:** This debug only message is issued as progress indicator for the program cleanup process.

CKX117E IEAVRLS parse_client RC=*retcode*

| **Explanation:** This message might be issued during program termination. When issued as message CKX117E, releasing the client task failed. See also message CKX117I.

| **User response:** The CKX117E message indicates a problem. See the Electronic Support Web site for possible maintenance associated with this message. If you cannot find applicable maintenance, follow the procedures described in Contacting IBM Support to report the problem.

CKX117I IEAVRLS parse_client RC=*retcode*

| **Explanation:** This message might be issued during program termination. When issued as message CKX117I, it indicates that a request to write a log-record was being processed during termination. See also message CKX117E.

| **User response:** Although this message is not normally issued, it does not necessarily indicate an error. Only when it occurs during every regular CKXLOG shutdown, see the Electronic Support Web site for possible maintenance associated with this message. If you cannot find applicable maintenance, follow the procedures described in Contacting IBM Support to report the problem.

CKX118E IEAVRLS parse_task RC=*retcode*

| **Explanation:** This message might be issued during program termination. When issued as message CKX118E, releasing a subtask failed. See also message CKX118I.

| **User response:** The CKX118E message indicates a problem. See the Electronic Support Web site for possible maintenance associated with this message. If you cannot find applicable maintenance, follow the procedures described in Contacting IBM Support to report the problem.

CKX118I IEAVRLS parse_task RC=*retcode*

| **Explanation:** This message might be issued during program termination. When issued as message CKX118I, it indicates that a request to write a log-record was being processed during termination. See also message CKX118E.

| **User response:** Although this message is not normally issued, it does not necessarily indicate an error. Only when it occurs during every regular CKXLOG shutdown, see the Electronic Support Web site for possible maintenance associated with this message. If

| you cannot find applicable maintenance, follow the
| procedures described in Contacting IBM Support to
| report the problem.

| **CKX119I Use model logstream *LSModel***

| **Explanation:** During initialization, it was found that
| the specified log stream was not defined. The CKXLOG
| started task attempts to define it using the indicated
| *LSModel* as model for the actual log stream.

| **CKX120W Unknown command**

| **Explanation:** The operator command shown in
| message CKX105I was not recognized as a valid
| operator command for the product.

| **User response:** Verify the correct syntax and spelling
| of the command you were trying to issue.

| **CKX121E Task is not APF authorized, exit**

| **Explanation:** The task is not APF-authorized. APF
| authorization is required for successful execution.

| **User response:** Ensure that the library, from which the
| modules are loaded, is marked as APF-authorized.
| When using a STEPLIB, ensure that all libraries in the
| concatenation are APF-authorized.

| **CKX123E Error return code from CKXINIT *retcode***

| **Explanation:** This message represents an internal
| error. See the Electronic Support Web site for possible
| maintenance associated with this message. If you
| cannot find applicable maintenance, follow the
| procedures described in Contacting IBM Support to
| report the problem. Create a problem record and
| provide the complete list of error messages.

| **CKX124I Start option FORCE specified**

| **Explanation:** This is an informational message
| confirming the use of the FORCE keyword as a start
| option.

| **CKX125I Start option DEBUG specified**

| **Explanation:** This is an informational message
| confirming the use of the DEBUG keyword as a start
| option.

| **CKX126E Product registration problem, see
| CKXDEBUG**

| **Explanation:** A problem occurred during registration
| of the zSecure Admin product.

| **User response:** Check the detailed error messages in
| the CKXDEBUG file. Ensure that the appropriate
| products are not disabled in your IFAPRDxx parmlib
| member.

| **CKX127E Product disabled here or install error,
| see CKXDEBUG**

| **Explanation:** A problem occurred during registration
| of the zSecure Admin product.

| **User response:** Check the detailed error messages in
| the CKXDEBUG file. Ensure that the appropriate
| products are not disabled in your IFAPRDxx parmlib
| member.

| **CKX128E Task already active**

| **Explanation:** During the inspection of the
| environment, the CKXLOG program was found to be
| already active. The current task is terminated.

| **User response:** If this message is issued incorrectly
| (that is, no other instance of CKXLOG is active in the
| system), you might be able to recover from this
| situation by using the FORCE startup option. Such a
| situation could result from the use of the FORCE
| operator command to stop a previous instance of the
| task.

| **CKX129I Force restart**

| **Explanation:** This is an informational message issued
| during initialization to confirm use of the FORCE
| option to recover from previous unrecoverable errors. It
| is issued if the FORCE option was used to bypass a
| previous 'Incorrect version' or 'Incorrect length' error.

| **CKX130E Error return code from ESTAEX,
| RC=*retcode-reascode***

| **Explanation:** During specification of an abend exit
| routine, a system error occurred. The error and reason
| codes were *retcode- reascode*.

| **User response:** See the Electronic Support Web site for
| possible maintenance associated with this message. If
| you cannot find applicable maintenance, follow the
| procedures described in Contacting IBM Support to
| report the problem.

| **CKX131I Waiting for subtasks to terminate**

| **Explanation:** During a normal stop of the task, a short
| waiting period is used to allow the subtasks to stop
| normally. At the end of the waiting period, the subtasks
| are terminated.

| **CKX132E Error attaching command parser,
| RC=*retcode***

| **Explanation:** While attaching the subtask that is used
| to parse and normalize RACF commands, a system
| error occurred. The ATTACH error code was *retcode*.

| **User response:** See the Electronic Support Web site for
| possible maintenance associated with this message. If

CKX133I • CKX143E

| you cannot find applicable maintenance, follow the
| procedures described in Contacting IBM Support to
| report the problem. Create a problem report including
| the complete error message.

CKX133I Successful attach command parser

| **Explanation:** This debug only message is issued after
| successful start of the RACF command parsing subtask.

CKX134E Error attaching expiry routine, RC=*retcode*

| **Explanation:** While attaching the subtask that is used
| to expire and remove unused ticket information, a
| system error occurred. The ATTACH error code was
| *retcode*.

| **User response:** See the Electronic Support Web site for
| possible maintenance associated with this message. If
| you cannot find applicable maintenance, follow the
| procedures described in Contacting IBM Support to
| report the problem. Create a problem report including
| the complete error message.

CKX135I Successful attach expiry routine

| **Explanation:** This debug only message is issued after
| successful start of the ticket expiration subtask.

CKX136E Control Block error, exit

| **Explanation:** During internal consistency verification,
| the main control block was found to be corrupted.
| Because this is detected during the startup, the control
| block must have been created during a previous
| execution. This situation should never occur.

| **User response:** See the Electronic Support Web site for
| possible maintenance associated with this message. If
| you cannot find applicable maintenance, follow the
| procedures described in Contacting IBM Support to
| report the problem. You might be able to recover from
| this situation by using the FORCE keyword on the
| START command.

CKX137E CKXC Incorrect version

| **Explanation:** The CKXLOG version as recorded in the
| CKXC control block during the previous execution of
| the product does not match the current version of the
| product. The CKXC control block is not usable.
| Execution of the current task is terminated.

| **User response:** This situation is most likely caused by
| an upgrade of the product code without an appropriate
| shutdown of the previous instance of the started task. It
| might also be caused by a destructive overlay of the
| control block. See the Electronic Support Web site for
| possible maintenance associated with this message. If
| you cannot find applicable maintenance, follow the
| procedures described in Contacting IBM Support to

| report the problem. It might be possible to recover from
| this situation via use of the FORCE startup parameter.

CKX138E CKXC Incorrect length

| **Explanation:** During internal consistency verification,
| CKXLOG detected an incorrect length of the CKXC
| control block. The CKXC control block is not usable.
| Execution of the current task is terminated.

| **User response:** This situation is most likely caused by
| an upgrade of the product code without an appropriate
| shutdown of the previous instance of the started task. It
| might also be caused by a destructive overlay of the
| control block. See the Electronic Support Web site for
| possible maintenance associated with this message. If
| you cannot find applicable maintenance, follow the
| procedures described in Contacting IBM Support to
| report the problem. It might be possible to recover from
| this situation via use of the FORCE startup parameter.

CKX139I Previous execution detected

| **Explanation:** This is an informational message that the
| CKXLOG program has detected that it has run before
| since IPL of the system. It indicates that some resources
| are reused from that previous execution. This pertains
| mainly to the common communications area and the
| Linkage Index (LX).

CKX140I Reuse CKXC area at *address*

| **Explanation:** This debug only message is issued to
| provide the address of the common communications
| control block (C2PC) that is reused from a previous
| instance of the started task.

CKX141W Improper shutdown detected, attempting cleanup

| **Explanation:** The last instance of the started task that
| initialized the C2PC control block was not able to mark
| its termination in the C2PC control block. This is
| probably caused by a program error, or failure to use
| the STOP command to terminate the previous instance
| of the task.

CKX142I Obtained CKXC storage at *address*

| **Explanation:** This debug only message is issued to
| provide the address of the new common
| communication control block (CKXC) that is created.

CKX143E Error return from IEANTCR, RC=*retcode*

| **Explanation:** An error occurred during creation of the
| Named Token that is used as anchor for the Common
| Communication Control Block (CKXC). The error
| return code is *retcode*.

| **User response:** See the Electronic Support Web site for

| possible maintenance associated with this message. If
 | you cannot find applicable maintenance, follow the
 | procedures described in Contacting IBM Support to
 | report the problem. Create an error report including the
 | retcode. You might be able to recover from this
 | situation via use of the FORCE startup parameter.

| **CKX144E Error return from IEANTRT, RC=*retcode***

| **Explanation:** An error occurred during retrieval of the
 | Named Token that is used as anchor for the Common
 | Communication Control Block (CKXC). The error
 | return code is *retcode*.

| **User response:** See the Electronic Support Web site for
 | possible maintenance associated with this message. If
 | you cannot find applicable maintenance, follow the
 | procedures described in Contacting IBM Support to
 | report the problem. Create an error report including the
 | retcode. You might be able to recover from this
 | situation via use of the FORCE startup parameter.

| **CKX145E Internal error, terminate**

| **Explanation:** An unspecified error occurred during
 | retrieval of the Named Token that is used as anchor for
 | the Common Communication Control Block (CKXC).

| **User response:** See the Electronic Support Web site for
 | possible maintenance associated with this message. If
 | you cannot find applicable maintenance, follow the
 | procedures described in Contacting IBM Support to
 | report the problem. You might be able to recover from
 | this situation via use of the FORCE startup parameter.

| **CKX146E Error return from IKJTSOEV RC=*retcode***

| **Explanation:** An error occurred during initialization of
 | the TSO environment necessary for command
 | interpretation and execution. The error return code is
 | *retcode*.

| **User response:** See the Electronic Support Web site for
 | possible maintenance associated with this message. If
 | you cannot find applicable maintenance, follow the
 | procedures described in Contacting IBM Support to
 | report the problem. Create an error report including the
 | retcode.

| **CKX147I TRANSWAP done**

| **Explanation:** This debug only message informs of a
 | successful transition of the product's task to
 | non-swappable. This is required for allowing cross
 | memory services.

| **CKX148I Need a new LX**

| **Explanation:** This debug only message indicates that
 | no previous LX was found. A new system Linkage
 | Index (LX) will be allocated. System LX is a
 | non-reclaimable critical system resource and may be

| issued only once. A new System LX is needed during
 | the first start after a system IPL. The new System LX
 | will be saved for future reuse by subsequent instances
 | of the started task.

| **CKX149I Obtained LX *lxnum***

| **Explanation:** This debug only message provides the
 | number of the new system Linkage Index (LX) that was
 | allocated. It is saved for future reuse by subsequent
 | instances of the started task.

| **CKX150I Found existing LX**

| **Explanation:** This debug only message indicates that
 | an LX was found from a previous instance of the
 | started task. The existing system Linkage Index (LX) is
 | reused. System LX is a non-reclaimable critical system
 | resource and can be issued only once.

| **CKX151I LX was *lxnum***

| **Explanation:** This debug only message provides the
 | number of the existing system Linkage Index (LX) that
 | is used. It is a logical continuation of message CKX150I.

| **CKX152I Need a new ET**

| **Explanation:** This debug only message can be issued
 | during initialization to indicate that a new Entry Table
 | (ET) is created, which contains the Program Call
 | definitions. The Entry Table is returned to the system
 | when the task ends.

| **CKX153I Obtained ET token**

| **Explanation:** This debug only message provides the
 | token for the Entry Table (ET) that was created.

| **CKX154I Found existing ET**

| **Explanation:** This debug only message can be issued
 | if an existing Entry Table (ET) was found during
 | initialization. The existing ET is reused. This message
 | should not occur, because the Entry Table is returned to
 | the system at normal task termination.

| **CKX155I ET was *etnum***

| **Explanation:** This debug only message provides the
 | token for the Entry Table (ET) that was found from a
 | previous instance of the started task. This message is a
 | logical continuation of message CKX154I.

| **CKX156I Create latch set of *nn* latches**

| **Explanation:** This debug only message is issued as
 | progress indicator for the program startup process.

CKX157I • CKX171I

| **CKX157I** **Latch set create RC=*retcode***

| **Explanation:** This debug only message shows the return code from the latch set create service.

| **CKX158I** **Latch set token =*latchsettoken***

| **Explanation:** This debug only message shows the generated token for the latch set.

| **CKX159E** **Error return from IEANTCR, RC=*retcode***

| **Explanation:** An error occurred during creation of the Named Token that is used to provide information for the command-issuing applications. The error return code is *retcode*. The program terminates.

| **User response:** See the Electronic Support Web site for possible maintenance associated with this message. If you cannot find applicable maintenance, follow the procedures described in Contacting IBM Support to report the problem. Create an error report including the *retcode*. You might be able to recover from this situation via use of the FORCE startup parameter.

| **CKX160I** **Connect to logstream, RC=*retcode***

| **Explanation:** This debug only message shows the return code from the system service to connect to the log stream.

| **CKX161I** **Logstream *LSname* not defined, try create**

| **Explanation:** This message is issued when the indicated log stream does not exist. The CKXLOG program continues with defining the log stream based on the model log stream.

| **CKX162I** **Define *Comm_LSname* RC=*retcode***

| **Explanation:** This debug only message shows the return code from the system service to define the new log stream, based on the model log stream.

| **CKX163I** **Retry connect after define**

| **Explanation:** This debug only message shows progress connecting to the specified log stream.

| **CKX164I** **Connect to logstream, RC=*retcode***

| **Explanation:** This debug only message shows the return code from the system service to connect to the just-defined log stream.

| **CKX165E** **Connect after define failed, RC=*retcode-reascode***

| **Explanation:** The connect to the newly defined log stream fails with an unexpected *retcode*. The program terminates.

| **User response:** To determine the reason for the error, check the "Return and reason codes" section for IXGCONN in *z/OS MVS Programming: Assembler Services Reference, Volume 2 (IAR-XCT)*.

| **CKX166E** **Define *LSName* failed, MODEL logstream not defined**

| **Explanation:** This message is issued when the definition of the indicated log stream failed because the specified or defaulted model log stream does not exist. The program terminates.

| **CKX167E** **Define *LSName* failed, RC=*retcode-reascode***

| **Explanation:** An unexpected error occurred during defining of log stream *LSName*. The program terminates.

| **User response:** To determine the reason for the error, check the "Return and reason codes" section for IXGINVNT in *z/OS MVS Programming: Assembler Services Reference, Volume 2 (IAR-XCT)*.

| **CKX168I** **Start of cleanup previous execution**

| **Explanation:** During initialization, a previous instance of the started task was detected. This debug only message indicates that possible residual data of a previous instance is being removed.

| **CKX169E** **Error return from IEANTDL, RC=*retcode***

| **Explanation:** This message might be issued during cleanup of a previous instance of the program. It indicates an unexpected error in removing a residual Name Token.

| **User response:** Restart the task with the debug option. See the Electronic Support Web site for possible maintenance associated with this message. If you cannot find applicable maintenance, follow the procedures described in Contacting IBM Support to report the problem. Create a problem report including the output of the started task.

| **CKX170I** **Removed non-persistent NT**

| **Explanation:** This debug only message is issued as progress indicator for the program cleanup process.

| **CKX171I** **Removed PC-Routine Address**

| **Explanation:** This debug only message is issued as progress indicator for the program cleanup process.

CKX172I Made task swappable again

Explanation: This debug only message is issued as progress indicator for the program cleanup process.

CKX173I Found existing ET *etnum*

Explanation: This debug only message is issued as progress indicator for the program cleanup process.

CKX174I ET destroyed

Explanation: This debug only message is issued as progress indicator for the program cleanup process.

CKX175I No ET found, next

Explanation: This debug only message is issued as progress indicator for the program cleanup process.

**CKX176I Disconnect from logstream,
RC=*retcode-rsncode***

Explanation: This debug only message is issued as progress indicator for the program cleanup process. Normally, both *retcode* and *rsncode* are 0.

CKX177I Special (Total) cleanup mode

Explanation: This informational message shows that a complete shutdown of the CKXLOG program is being done. All resources are released, which includes some non-reusable system resources. Usually, this is only done when processing the SIPL command in preparation for a software release migration.

**CKX178I zSecure Admin Command Logger
release initialization successful**

Explanation: This informational message shows that initialization of release release of the zSecure Admin Command Logger has completed. The program is now ready to log records to the CKXLOG log stream.

CKX179I Delete N/T

Explanation: This debug only message is issued as progress indicator for the 'total' cleanup process.

CKX180E Error return from IEANTDL, RC=*retcode*

Explanation: An unexpected error occurred during deletion of a named token during the 'total' cleanup process. Program cleanup continues.

CKX181I Freemain CKXC

Explanation: This debug only message is issued as progress indicator for the 'total' cleanup process.

CKX182E Incorrect length of CKXC, attempt free

Explanation: During the 'total' cleanup process, the storage for the main communication control block is returned to the system. However, the internal length field does not match that of the current version of the code. The program uses the internal length to free storage. This might ABEND.

User response: Most often, this is caused by changing the release of the CKXLOG program code without a proper SIPL of the previous instance of the CKXLOG started task. Verify that you are using the correct version of the code and that you followed the documented migration procedure.

CKX183E Impossible length of CKXC

Explanation: During the 'total' cleanup process, the storage for the main communication control block is returned to the system. However, the internal length field is incorrect. Program cleanup cannot continue.

User response: See the Electronic Support Web site for possible maintenance associated with this message. If you cannot find applicable maintenance, follow the procedures described in Contacting IBM Support to report the problem.

CKX184I End of cleanup previous execution

Explanation: During initialization, a previous instance of the started task was detected. This progress message indicates that removal of all potential data from that previous instance is complete.

CKX185I Ticket expiry timer Loop

Explanation: This debug only message is issued every minute to mark the start of detecting inactive ticket information.

CKX186I Declare expired

Explanation: This debug only message is issued to show the information of a ticket that is detected to be inactive and expired. It is followed by a hex dump of the first 4 lines of the internal control block representing the ticket information.

**CKX187E Error attaching Expire Timer routine,
RC=*retcode***

Explanation: While attaching the timer task that is used to trigger inspection and removal of inactive ticket

information, a system error occurred. The ATTACH error code was *retcode*.

User response: See the Electronic Support Web site for possible maintenance associated with this message. If you cannot find applicable maintenance, follow the procedures described in Contacting IBM Support to report the problem. Create a problem report including the complete error message.

CKX188I Successful attach Expire Timer routine

Explanation: This debug only message is issued after successful start of the timer task that triggers the inspection and removal of inactive ticket information.

CKX189I Read from CKXPARM

Explanation: This debug only message is issued to mark the start of processing the statements in CKXPARM.

CKX190I Start of CKXC

Explanation: This debug only message indicates that the next 16 lines show the hex dump of the CKXC control block.

CKX191I Base ptr: *address*

Explanation: This debug only message shows the starting location for the hexadecimal data that is shown in subsequent CKX392I messages.

CKX192I *address : hexdata*

Explanation: This debug only message shows hexadecimal data that IBM support personnel can use to diagnose problems.

CKX193I Completed processing CKXPARM file

Explanation: This debug only message is issued to mark the end of processing the statements in CKXPARM.

CKX194E Required ddname CKXPARM not allocated

Explanation: The CKXPARM ddname is not allocated. The CKXPARM ddname is used for CKXLOG initialization statements.

User response: See the Installation and Deployment Guide for information about the required ddnames and the contents of the allocated files.

CKX195E Command Logger {User|System} abend ({Dec|Hex}) *abndcode-reascode*

Explanation: An abend occurred during processing of the CKXLOG started task. For a User abend, the *abndcode* and *reascode* are shown in decimal format. For a System abend, the codes are shown in hex format.

User response: See the Electronic Support Web site for possible maintenance associated with this message. If you cannot find applicable maintenance, follow the procedures described in Contacting IBM Support to report the problem. Create a problem report including the complete error message.

CKX196E Cannot perform cleanup

Explanation: An abend occurred during processing of the CKXLOG started task. The failing task does not contain specific recovery code. The abend exit routine attempted to perform general program cleanup. However, the CKXLOG started task was in an inconsistent state, which prohibits further processing.

User response: See the Electronic Support Web site for possible maintenance associated with this message. If you cannot find applicable maintenance, follow the procedures described in Contacting IBM Support to report the problem.

CKX197I Attempting cleanup

Explanation: An abend occurred during processing of the CKXLOG started task. The failing task does not contain specific recovery code. The abend exit routine starts to perform general program cleanup.

CKX198I Cleanup completed

Explanation: An abend occurred during processing of the CKXLOG started task. The failing task does not contain specific recovery code. The abend exit routine completed general program cleanup.

CKX199E STACK failed; RC=*retcode*

Explanation: An unexpected error occurred during manipulation of the command input stack.

User response: Verify that the required CKQPARM statements were read and processed. You might want to stop and start the CKQLOG started task. See the Electronic Support Web site for possible maintenance associated with this message. If you cannot find applicable maintenance, follow the procedures described in Contacting IBM Support to report the problem. Create a problem report including the complete error message.

CKX messages from 200 to 299

<p> CKX200I <i>input_statement</i></p> <p> Explanation: This debug only message echoes the statement as read from the CKXPARMS ddname.</p>	<p> intended only for IBM internal testing purposes.</p>
<p> CKX201I Command found is <i>cmdname</i></p> <p> Explanation: This debug only message shows that the first word of the input statement was determined to be <i>cmdname</i>.</p>	<p> CKX210I LSNAME =<i>lsname_with_symbols</i></p> <p> Explanation: This debug only message shows the name of the log stream as it is read from the CKXPARMS input file.</p>
<p> CKX202E Command not recognized, <i>cmdname</i></p> <p> Explanation: The first word of the input statement (<i>cmdname</i>) was not recognized as a valid input statement.</p>	<p> CKX211I LSNAME =<i>lsname_resolved_symbols</i></p> <p> Explanation: This debug only message shows the name of the log stream after resolving system symbols. It is the name of the actual log stream that is used.</p>
<p> CKX203E Invalid parameter, Parse-RC=<i>retcode</i></p> <p> Explanation: The input statement did not contain valid keywords and parameters.</p>	<p> CKX212I LSMODEL =<i>lsmode_with_symbols</i></p> <p> Explanation: This debug only message shows the name of the model definition for the log stream as it is read from the CKXPARMS input file.</p>
<p> CKX204I No command found (? or /*)</p> <p> Explanation: This debug only message indicates that the input statement did not contain a valid command. It is probably a comment line, or contains a ?, asking for second level help messages.</p>	<p> CKX213I LSMODEL =<i>lsmode_resolved_symbols</i></p> <p> Explanation: This debug only message shows the name of the model log stream after resolving system symbols. It is the name of the actual model that is used.</p>
<p> CKX205E Command not recognized, scancode=<i>scan-flag</i></p> <p> Explanation: The input statement did not start with a valid command.</p>	<p> CKX214I Tickets are expired after (HHMM) <i>HHMM</i></p> <p> Explanation: This debug only message shows the interval after which inactive ticket information is discarded.</p>
<p> CKX206E Error return code from ESTAEX, RC=<i>retcode-rsncode</i></p> <p> Explanation: During specification of an abend exit routine, a system error occurred. The error code was <i>retcode- rsncode</i>.</p> <p> User response: See the Electronic Support Web site for possible maintenance associated with this message. If you cannot find applicable maintenance, follow the procedures described in Contacting IBM Support to report the problem.</p>	<p> CKX215E Ticket expiration time invalid <i>HHMM</i></p> <p> Explanation: This interval specified in CKXPARMS is invalid. The default value of one (1) hour is used instead.</p>
<p> CKX207I Activate retry routine at <i>retry-addr</i></p> <p> Explanation: An abend occurred during processing of the CKXLOG started task. The failing task contains a specific recovery code. The abend exit routine passes control to the recovery code at address <i>retry-addr</i>.</p>	<p> CKX216E IXGWRITE Failed, RC=<i>retcode</i></p> <p> Explanation: Writing the record to the log stream failed.</p> <p> User response: Check the return code of the IXGWRITE service. If this does not point to an obvious error, see the Electronic Support Web site for possible maintenance associated with this message. If you cannot find applicable maintenance, follow the procedures described in Contacting IBM Support to report the problem.</p>
<p> CKX208E Abend with ESTAEX</p> <p> Explanation: This message indicates that the ABND command is processing. The ABND command is</p>	<p> CKX217I IXGWRITE RC=<i>retcode</i></p> <p> Explanation: This debug only message shows the return code of writing the record to the log stream.</p>

CKX218I IXGWRITE *blockid=blockid*
Explanation: This debug only message shows the unique block identifier of the record that was just written.

CKX220E Define *LSName* failed, STRUCTNAME missing on MODEL logstream
Explanation: The define of the log stream *LSName* failed because the MODEL log stream was not a DASDONLY log stream, but lacked the specification of the STRUCTNAME.
User response: Correct the MODEL log stream definition and restart the CKXLOG task.

CKX221E Coupling Facility STRUCTNAME not defined in CFRM
Explanation: The define of the log stream failed because the STRUCTNAME specified on the MODEL log stream was not defined in the Coupling Facility Resource Manager (CFRM) data set
User response: Correct the LOGR and/or CFRM definitions and restart the CKXLOG task.

CKX222E No Coupling Facility available for logstream *LSName*
Explanation: The connect to the log stream *LSName* failed because there was no suitable coupling facility available.
User response: Check the CFRM definitions and ensure that a suitable coupling facility can be used. Restart the CKXLOG task.

CKX223E DASDONLY logstream *LSName* cannot be shared in sysplex
Explanation: The connect to the log stream *LSName* failed because it was defined as a DASDONLY log stream, and it is already in use in the SYSPLEX.
User response: Correct the LOGR definitions and the input statements in CKXLOG. Restart the CKXLOG task.

CKX messages from 300 to 399

CKX301E No command to log
Explanation: The LOG function was requested, but no command was specified. Command processing is terminated.

CKX302E Command Logger not available
Explanation: The zSecure Command Logger started task was not active. The requested function was not executed.

CKX224E Connect to logstream failed, RC=*retcode-rsncode*
Explanation: An unexpected error occurred when connecting to the log stream.

User response: Check the return code of the IXGCONN service. If this does not point to an obvious error, see the Electronic Support Web site for possible maintenance associated with this message. If you cannot find applicable maintenance, follow the procedures described in Contacting IBM Support to report the problem.

CKX225E Define *LSName* failed, STRUCTNAME not defined in LOGR inventory
Explanation: The define of the log stream *LSName* failed because the STRUCTNAME that is specified on the MODEL log stream was not defined in the LOGR inventory.
User response: Correct the LOGR definitions and restart the CKXLOG task.

CKX226E No SAF authorization to connect to logstream *LSName*
Explanation: The CKXLOG started task user ID does not have at least UPDATE access to the *LSName* in the LOGSTRM resource class.

User response: Grant UPDATE access to the *LSName* in the LOGSTRM resource class and restart the CKXLOG task.

CKX227E No SAF authorization to define logstream *LSName*
Explanation: The CKXLOG started task user ID does not have at least ALTER access to the *LSName* in the LOGSTRM resource class.

User response: Grant ALTER access to the *LSName* in the LOGSTRM resource class and restart the CKXLOG task. For a coupling facility log stream, you might also need to grant UPDATE access to the STRUCTURE defined for the model log stream

CKX304E CKXLOGID must be invoked as a TSO command
Explanation: The CKXLOGID command cannot be run as a stand alone program. It requires to be invoked as a TSO command. Command processing is terminated.

CKX306I SET option used
Explanation: This DEBUG only message shows trace information about the flow of a request.

CKX307I CLEAR option used

Explanation: This DEBUG only message shows trace information about the flow of a request.

CKX308I DISPLAY option used

Explanation: This DEBUG only message shows trace information about the flow of a request.

CKX309I LOG option used

Explanation: This DEBUG only message shows trace information about the flow of a request.

CKX310E Missing function

Explanation: The function keyword of the CKXLOGID could not be located or was not recognized. Command processing is terminated.

CKX311I ID keyword

Explanation: This DEBUG only message shows trace information about the flow of a request.

CKX312W ticket_id truncated

Explanation: The specified ticket ID without quotation marks is longer than 32 characters. The truncated ticket ID is used.

CKX313W ticket_id truncated

Explanation: The specified ticket ID within quotation marks is longer than 32 characters. The truncated ticket ID is used.

CKX314E Ticket_ID value missing

Explanation: The value for the ticket ID could not be located. Command processing is terminated.

CKX315I Ticket_ID entered = *ticket_id*

Explanation: This DEBUG only message shows trace information about the flow of a request.

CKX316I Description keyword

Explanation: This DEBUG only message shows trace information about the flow of a request.

CKX317W Description truncated

Explanation: A specified ticket description without quotation marks is longer than 255 characters. The truncated ticket description is used.

CKX318W Description truncated

Explanation: A specified ticket description within quotation marks is longer than 255 characters. The truncated ticket description is used.

CKX319W Description value missing

Explanation: The value for the ticket description could not be located. The ticket ID is registered without a ticket description.

CKX320I Description entered = *ticket_description*

Explanation: This is a DEBUG only message showing trace information about the flow of a request.

CKX321I Command keyword

Explanation: This DEBUG only message shows trace information about the flow of a request.

CKX322W Command truncated

Explanation: A specified command within quotation marks is longer than 32768 characters. The truncated command is used.

CKX323I Command entered = *command*

Explanation: This DEBUG only message shows trace information about the flow of a request.

CKX324E Command value missing

Explanation: The value for the command to be logged could not be located. Command processing is terminated.

CKX325I From keyword

Explanation: This DEBUG only message shows trace information about the flow of a request.

CKX326I Node name = *node_name*

Explanation: This DEBUG only message shows trace information about the flow of a request.

CKX327E Node name missing

Explanation: The value of the from node could not be located. The node is the value before the period in the FROM parameter. Command processing is terminated.

CKX328I • CKX392I

CKX328I **User name =** *user_name*

Explanation: This DEBUG only message shows trace information about the flow of a request.

CKX329E **User name missing**

Explanation: The value of the from user in the LOG function could not be located. The node is the value following the period in the FROM parameter. Command processing is terminated.

CKX330I **Component keyword**

Explanation: This DEBUG only message shows trace information about the flow of a request.

CKX331I **Component =** *component_name*

Explanation: This DEBUG only message shows trace information about the flow of a request. The value for the component name is always CKXLOGID.

CKX332I **End of parameter parse**

Explanation: This DEBUG only message shows trace information about the flow of a request.

CKX333W **No ticket identifier set**

Explanation: The DISPLAY or LOG function is used, but the CKXLOG server does not have any value for the ticket ID. For the LOG function, the specified command is logged without any ticket information.

CKX334E **Writing command log record failed**

Explanation: The LOG function is used, but the CKXLOG encountered an error while writing the log stream record.

User response: Inspect the job log of the CKXLOG started task for additional information.

CKX335I **Ticket_id is** "*id-value*"

Explanation: This message is the response to the DISPLAY request to show the currently registered ticket information. The *id-value* is shown within double quotation marks, which are not part of the registered value.

CKX336I **Ticket_desc is** "*desc-value*"

Explanation: This message is the response to the DISPLAY request to show the currently registered ticket information. The *desc-value* is shown within double quotation marks, which are not part of the registered value.

CKX337I **Logstream name is** "*lstream*"

Explanation: This message is the response to the LSNAME request to show the name of the currently used log stream. The *lstream* is shown within double quotation marks, which are not part of the current log stream name.

CKX338E **Incorrect parameters, command terminated**

Explanation: Parsing the CKXLOGID command failed. This message is accompanied by other messages showing the invalid parameters. Command processing is terminated.

User response: Verify that correct parameters and keywords are used.

CKX339I **Show LSNAME option**

Explanation: This DEBUG only message shows trace information about the flow of a request.

CKX340I **Retcode keyword**

Explanation: This DEBUG only message shows trace information about the flow of a request.

CKX341I **Specified retcode =** *retcode*

Explanation: This DEBUG only message shows trace information about the flow of a request.

CKX342E **Retcode value missing**

Explanation: The value of the return code in the LOG function could not be located. Command processing is terminated.

CKX391I **Base ptr:** *address*

Explanation: This DEBUG only message shows the starting location for the hexadecimal data that is shown in subsequent CKX392I messages.

CKX391I **Base ptr:** *address*

Explanation: This DEBUG only message shows the starting location for the hexadecimal data that is shown in subsequent CKX392I messages.

CKX392I **address :** *hexdata*

Explanation: This DEBUG only message shows hexadecimal data that IBM support personnel can use to diagnose problems.

CKX messages from 700 to 799

CKX797I **Unsupported call type**

Explanation: This message indicates a mismatch in an internal API. Verify that the product install and setup ended successfully.

Severity: 16

Severity: 16

CKX798I **Internal version mismatch**

Explanation: This message indicates a mismatch in an internal API. Verify that the product install and setup ended successfully.

Severity: 16

CKX messages from 900 to 999

CKX962D **Out of memory**

Explanation: UNIX System Service spawn encountered an out of memory condition.

User response: Increase REGION (or possibly MEMLIMIT).

Severity: 8

Severity: 08

CKX962H **Spawn failed**

Explanation: UNIX System Service spawn failed.

User response: Look in the SYSPRINT or CKXPRINT for more details about the exact error condition.

Severity: 08

CKX962K **Wait failed**

Explanation: UNIX System Service subprocess wait failed.

Severity: 08

CKX962J **TSOXUSS RC=xxxxxxxx**

Explanation: UNIX System Service interface failed with hexadecimal return code *xxxxxxxx*.

CKX962Q **Quoting error**

Explanation: UNIX System Service command parameter quoting is not understood.

Severity: 08

CKX962V **No command**

Explanation: UNIX System Service command is a null string.

Severity: 08

C4R messages

C4R400W **No ticket identifier set**

Explanation: The terminal user has READ access to the C4R.command.=CKXLOG resource, but has not issued a CKXLOGID command to set a ticket identification.

User response: Either issue a CKXLOGID command to set ticket information, or contact your system administrator to verify correct access to the =CKXLOG resources.

Administrator response: Verify that terminal users have correct access to the =CKXLOG resources. Avoid logging the RACF command to the command log by ensuring that they do not have READ access or higher to the appropriate =CKXLOG resource, or suppress the warning message by ensuring that they have at least UPDATE to the appropriate =CKXLOG resource.

C4R401E **Writing command log record failed**

Explanation: This warning message is issued when the terminal user has at least READ access to the C4R.command.=CKXLOG resource and writing the command log record failed.

User response: Contact your system administrator to investigate the reason for this error.

C4R402W **CKXLOG started task not active**

Explanation: The terminal user has READ access to the C4R.command.=CKXLOG resource, but the CKXLOG started task is not active.

User response: Contact your system administrator to verify correct access to the =CKXLOG resources, and to verify that the CKXLOG started task is active.

| **Administrator response:** Verify that terminal users
| have correct access to the =CKXLOG resources. Avoid
| logging the RACF command to the command log by
| ensuring that they do not have READ access or higher
| to the appropriate =CKXLOG resource, or suppress the
| warning message by ensuring that they have at least
| UPDATE to the appropriate =CKXLOG resource. If
| logging is required, verify that the CKXLOG started
| task is active.

Chapter 4. zSecure Admin and Audit for RACF User Reference Manual

This chapter lists the documentation updates for *zSecure Admin and Audit for RACF User Reference Manual* as a result of the zSecure Command and Ticket Logging SSE.

Updated: UI

- In the main menu, the **CR** option was added:

```
CR  Command review  Review and run commands
  1  Libraries      Review and run commands from library
  2  CKXLOG        Review and re-run commands in commands execution log
```

For the corresponding documentation, see the new section **Command logging facility** in Chapter 2. RACF Admin Guide.

- A section **Ticket identifier** was added with two new fields to several panels:

```
Ticket identifier
ID . . . . _____ More _
Description _____
```

These fields were added to the following panels:

Section	Panels
"Line commands on profile displays"	GROUP COPY panel RESOURCE COPY-CLASS panel USER RECREATE panel
"RA.4.0 Copy user - Copy an existing user to a new user"	User Multiple copy panel User Multiple copy panel - user selection
"RA.4.1 Copy group - Copy existing groups to new groups"	Group multiple copy panel
"RA.4.2 Copy dataset - Copy dataset profiles to another High-level qualifier"	Dataset multiple copy panel
"RA.4.3 Copy resource - Copy general resource profiles to another class"	Resource multiple copy panel
"RA.4.6 Recreate user - Recreate users"	User Recreate panel
"RA.4.7 Recreate grp - Recreate groups"	Group recreate panel
"RA.4.8 Recreate ds- Recreate data set profiles"	Data set Recreate panel
"RA.4.9 Recreate res - Recreate general resource profiles"	Resource Recreate panel
"RA.4.C Copy CICS - Copy CICS prefixed profiles or members"	Copy CICS prefixed profiles panel

- In the SE.4 SETUP - Confirm panel, the **Add** option was added and the descriptions for the **Confirmation** options were updated:

```
Confirmation . . . . 5 1. None 2. Deletes 3. Passwords 4. All 5. Add
```

Table 2. Setup - Confirm panel - Confirmation command options

Confirmation setting	Description
All	All commands must be confirmed.

Table 2. Setup - Confirm panel - Confirmation command options (continued)

Confirmation setting	Description
Add	All commands must be confirmed, even the "Add new profile" commands that RA.U, RA.G, RA.D, and RA.R generate.

Chapter 2. RACF Admin Guide

- Added: section **Command logging facility**.
- Added: section **CR.1 Run RACF commands from library**
- Updated: section **"Interactive reports"**. CKRDQXL was added:

Table 3. CARLa scripts to create Interactive reports

Report	Meaning
CKRDQXL	Sample display of CKXLOG records. The CR option in zSecure uses this script.

- Updated: section **"Batch CARLa scripts"**. CKRLQXL was added:

Table 4. CARLa scripts for creating reports through batch processing

CARLa script	Function
CKRLQXL	Sample print format output of CKXLOG records.

Chapter 11. Calling zSecure

- Updated: section **"IBM Security zSecure jobs"**. The following jobs were added:

C2RJXUNL

Unloads and selects yesterday's command execution log records from the current sysplex-level CKXLOG log stream.

- Updated: section **"Naming convention"**. CKX was added:

Table 5. Product prefixes for CARLa scripts

Prefix	Component or feature
CKX	zSecure Admin Execution Log

- Updated: section **"Standard CARLa scripts"**. C2RL* was updated and CKRLXSUM and C2RLXUNL were added:

CKRLXSUM - CKXLOG record summary table

Overview of unloaded CKX log records.

C2RL* - Unload and report statistics

These members provide unload and report statistics.

C2RLXUNL- Unload CKXLOG

This member unloads CKXLOG records and merges them to the specified output dataset.

Chapter 15. Setup and Library

Updated: section **"SE.1 SETUP - Input files"**. Three rows were added to the table "Input Files: available data set types" and the File type menu for Setup files panel was updated:

Table 6. Input Files: available data set types

If you want to use...	use TYPE...	Notes
The live CKXLOG log stream	ACT.CKXLOG	Admin only.
A specific CKXLOG log stream	CKX.LOGSTR	Admin only.
A CKXLOG data set	CKXLOG	Admin only.

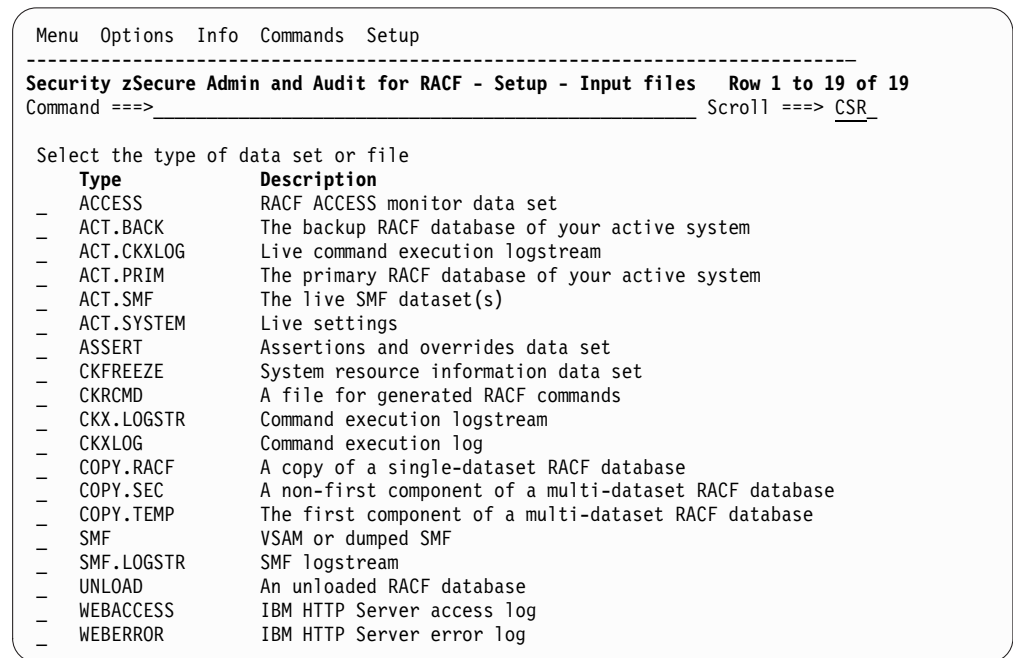


Figure 2. File type menu for Setup files

Command logging facility

zSecure Admin provides an option to log all RACF commands that are issued through the zSecure interface in a central repository. This section provides an overview of the functions and describes the reporting options.

The command repository is defined in a z/OS log stream. Using zSecure functions, it is possible to select and report about the recorded RACF commands. The actual recording of the commands in the log stream is done through a started task (CKXLOG) that runs on each system (LPAR) where RACF commands are run. The record of the command can be annotated with a ticket identification and a ticket description. The zSecure Admin ISPF interface and zSecure Command Verifier send commands to the zSecure Admin Command Logger.

When issuing commands through the zSecure ISPF user interface, the command passes several components that can request the command to be logged. Each of these components might provide its unique information that is not available otherwise. For example, when you issue a permit through CKGRACF, you can specify that the permit command is to be executed at a later date. At the time that you enter the command, CKRCARLA logs it, including the scheduled date. Later, when the permit command is executed, CKGRACF logs it just before it is executed. And after execution, zSecure Command Verifier might log it, including the return code from RACF. The Command Logger started task uses RACF profiles to control

if records from a component are recorded or not. For more information about these CKX.CKXLOG.LOG.*component-name* profiles, see *zSecure CARLa-Driven Components Installation and Deployment Guide*.

The ISPF reporting interface (option **CR.2**) provides an option to suppress records for RACF commands that might also be recorded through zSecure Command Verifier. If you write your own CARLa queries, you can use the IN_C4R_SCOPE flag to include or exclude these records.

To use these functions, the systems programmer must first install and activate them. For more information, see section “Setup of zSecure Admin Command Logger” in *zSecure CARLa-Driven Components Installation and Deployment Guide*.

Using the Command Logger functions involves the following steps:

- Set the ticket ID and description either using the zSecure ISPF interface or the CKXLOGID TSO command. Setting the ticket ID and description is described in “Setting the ticket ID and description in ISPF.” Use of the CKXLOGID command is described in “Setting the ticket ID and description outside ISPF” on page 32. Recording through the zSecure ISPF interface is available only for users with at least READ access to resource CKR.CKXLOG.ID.SHOW. Setting a ticket ID is required for users with READ access or higher to resource CKR.CKXLOG.ID.PROMPT.
- Issue commands either using the zSecure interface or directly using TSO. If commands are issued using the zSecure ISPF interface, they are sent to the CKXLOG started task for recording in the log stream. RACF commands that are issued outside of the zSecure ISPF interface can also be recorded through the CKXLOG started task if zSecure Command Verifier is installed and active. RACF commands are recorded if the user has at least READ access to C4R.*command*.=CKXLOG. For more information about recording by zSecure Command Verifier, see *zSecure Command Verifier User Guide*.

The zSecure Admin Command Logger logs commands depending on the component that passes the command and the user's access to this component. Switch profiles of the form CKX.CKXLOG.LOG.*component-name* control how commands are logged. For additional information about these switch profiles, see *zSecure CARLa-Driven Components Installation and Deployment Guide*.

The commands as recorded by the CKXLOG started task do not contain values for passwords, phrases, and similar sensitive fields. These are replaced by asterisks.

- Reporting about the commands that have been issued and run. Reporting can be done using CARLA scripts, or using option CR (Command Review) on the zSecure ISPF interface. As input to the reports, you can use the current active log stream or data sets that contain offloaded records. In SETUP FILES, you can specify the active log stream, a log stream data set with additional LOGR parameters, or a sequential data set that contains CKXLOG records. See SE.1 SETUP - Input files for more information about specifying CKXLOG input sources. “CR.2 Review zSecure Command Logger logs” on page 34 describes the available CKXLOG reports.
- Additional functions, like offloading or creating extracts of the log stream data sets.

Setting the ticket ID and description in ISPF

The ticket ID and description can be set in the zSecure ISPF interface by using the CKXLOGID command. If you issue CKXLOGID as a primary command, a panel

similar to the following is displayed.

```
Menu      Options      Info      Commands      Setup
-----
zSecure Suite - Ticket ID/Description
Command ==> _____
Ticket ID
_____
Description
_____
_____
_____
Press ENTER to accept changes.
```

This panel provides the option to set the ticket ID and the ticket description for all subsequent commands. The information in this panel is saved between sessions. When you are generating commands using the zSecure ISPF interface, and you have access to the CKR.CKXLOG.ID.PROMPT resource, you are prompted to confirm that the ticket ID and description are correct. This is done through the regular command confirmation panels. You must specify a value for the ticket ID. An example of a confirmation panel with the ticket information is shown in Figure 3.

The confirmation panel is also shown if you have requested command confirmation through the SETUP CONFIM option. If you have access to CKR.CKXLOG.ID.SHOW, the **Ticket ID** and **Description** fields are available to specify ticket information. In this case, specifying a value is not required.

```
Menu      Options      Info      Commands      Setup
-----
zSecure Admin+Audit for RACF - Confirm command
Command ==> _____
Ticket ID . . Test 1234_____
Description . Another test example_____ More. _
Confirm or edit the following command
altuser CRMBGU2 special_____
_____
Command execution . 1 1. EXECUTE RACF command
                     2. EXECUTE CKGRACF command (allows use of Reason)
                     3. ASK administrator to execute CKGRACF command
                     4. REQUEST CKGRACF command for later execution
                     5. WITHDRAW CKGRACF command
Reason . . . . . _____
Press ENTER to continue or END to cancel the command.
```

Figure 3. Confirm command

If additional space is needed for the ticket description, check the **Show More** field, and the ticket specification panel with space for up to 255 characters is shown. When the command is confirmed, it is directly issued or queued, according to your SETUP CONFIRM settings. If a ticket ID has been set, the TSO CKXLOGID

command is issued right before the other commands, to ensure that the command as the CKXLOG started task logs it tagged with the right ticket information.

The first 12 characters of the ticket ID are also shown on the right in the top area of the screen on most interactive reports where you can issue commands. The following figure shows an example display with the current ticket ID in bold:

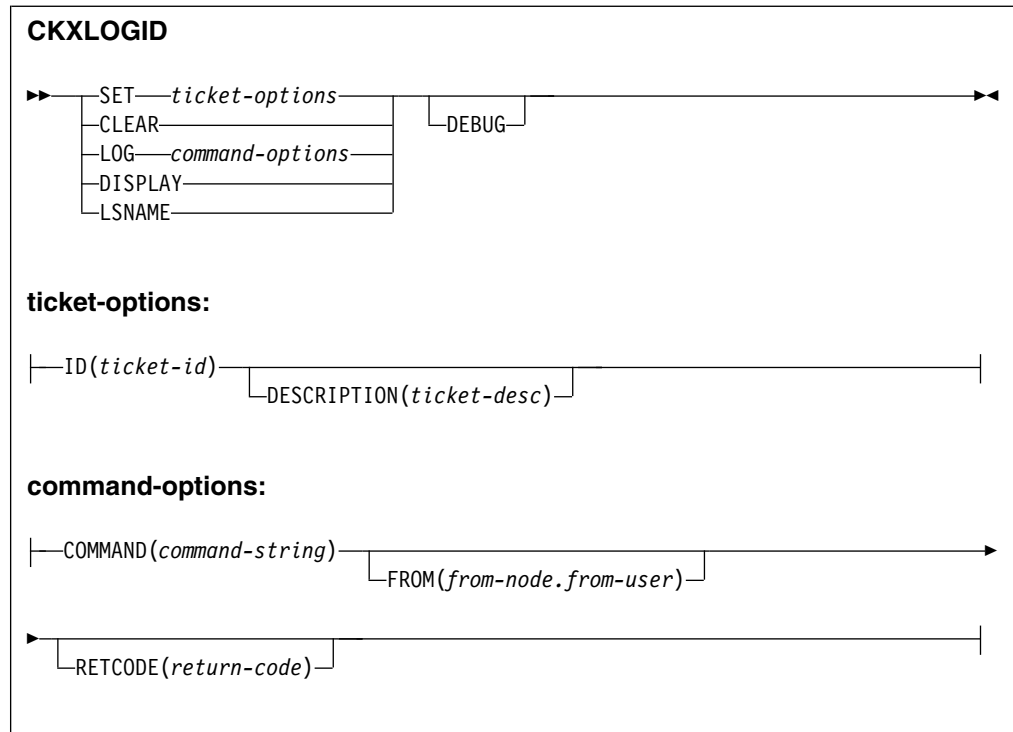
```
zSecure Admin+Audit for RACF USER IBMUSER overview                               Line 1 of 1
Command ==>>                                                                    Scroll==>> CSR
Users like IBMUSER                                                                12 Dec 2018 18:00 Change#12345
  User      Complex  Name                               DfltGrp  Owner    RIRP SOAR gC CX MF
  IBMUSER   PIPL84   TO BE REVOKED-LATER OMVS      IBMUSER  SO      C
***** Bottom of Data *****
```

The ticket information is also included in the command stream that is sent to remote systems when using the zSecure server for command distribution. The ticket information is not transferred when using RRSF. If commands are queued, for example, when using the COPY USER command, the CKXLOGID command is queued before the other commands.

Setting the ticket ID and description outside ISPF

If you are generating and issuing commands through the zSecure ISPF interface, the application takes care of setting the ticket information and passing command records to the CKXLOG started task. However, it is also possible to directly set the ticket ID and description using the TSO CKXLOGID. This can be used for batch jobs, or in combination with for example zSecure Command Verifier to log RACF commands.

The syntax of the TSO CKXLOGID command is as follows:



The main purpose of the CKXLOGID is to SET and CLEAR the ticket information. For testing purposes, it is also possible to display the current ticket information, to show the name of the log stream that the CKXLOG started task uses, and to write records to the log stream. A component name identifies each record that is added to the log stream. When the CKXLOGID command is used to write a command log record, it is identified by the value CKXLOGID for the component name. The following keywords are supported:

SET

The SET function is used to specify the ticket ID and ticket description for subsequent commands that the CKXLOG started task logs:

ID(*ticket-id*)

A string of maximum 32 characters identifies the ticket. This string is normally enclosed in single quotation marks. If the string itself includes quotation marks, they must be doubled, as is standard for TSO commands. If the string does not include any blanks or special characters, the surrounding quotation marks can be omitted.

DESCRIPTION(*ticket-desc*)

The ticket description is an optional string of maximum 255 characters that can be used to further identify or describe the ticket. The string is normally enclosed in single quotation marks. If the string itself includes quotation marks, they must be doubled, as is standard for TSO commands. If the string does not include any blanks or special characters, the surrounding quotation marks can be omitted.

CLEAR

The CLEAR function is used to reset a previously specified ticket ID and ticket description. The CLEAR function does not support any sub-options.

LOG

The LOG function is intended for testing purposes. It calls the CKXLOG started task to write a record for the specified command. The log records can

be identified through the component name. For records originating from CKXLOGID, the component value is CKXLOGID.

COMMAND(*command-string*)

Specifies the command that is to be logged. The *command-string* must be enclosed in single quotation marks. The maximum length of the string is approximately 32k.

FROM(*from-node.from-user*)

Optionally specifies the original source of the command. This would normally be the user that submitted a batch job, or the client user that issued a command through the zSecure server. When specified on the CKXLOGID command, it must be two strings of maximum eight (8) characters, separated by a single period.

RETCODE(*return-code*)

Specifies an optional return code of the command that must be recorded in the log record. The *return-code* can be any number between 0 and 255.

DISPLAY

The DISPLAY function can be used to show the current setting of the ticket ID and the optional ticket description. The DISPLAY function does not support any sub-options.

If a ticket ID has been set, the response from the DISPLAY function is similar to the following:

```
CKX335I Ticket_id is "CHANGE 12345678"  
CKX336I Ticket_desc is "Extra description text for change 12345678"
```

If no ticket ID has been set, the following message is issued:

```
CKX333W No ticket identifier set
```

LSNAME

The LSNAME function can be used to show the name of the current log stream used by the CKXLOG started task. The LSNAME function does not support any sub-options.

If the CKXLOG is active, the response from the LSNAME function is similar to the following:

```
CKX337I Logstream name is PLEX1.CKXLOG
```

If the CKXLOG started task is not active, the following message is issued:

```
CKX302I Command Logger not available
```

DEBUG

The DEBUG keyword specifies that the CKXLOGID program issues additional messages. The DEBUG keyword has no additional parameters.

CR.2 Review zSecure Command Logger logs

The **CR.2 CKXLOG** option that you can select on the zSecure main menu facilitates viewing and reporting of zSecure Admin command log records that the CKXLOG started task writes.

```

Menu          Options      Info      Commands      Setup
-----
                                zSecure Admin for RACF - Main menu
Option ==>> _____

SE  Setup          Options and input data sets
RA  RACF           RACF Administration
AU  Audit          Audit security and system resources
RE  Resource       Resource protection reports
AM  Access         RACF Access Monitor
CR  Command review Review and run commands
  1  Libraries     Review and run commands from library
  2  CKXLOG        Review and re-run commands in command execution log
CO  CARLa          Work with CARLa queries and libraries
IN  Information    Information and documentation
LO  Local          Locally defined options
X   Exit           Exit this panel

```

Figure 4. zSecure main menu

Select the **CR.2** menu option to display the main Figure 5. Use this panel to enter selection criteria in on or more fields to limit the logged commands. When you specify selection criteria, the output includes only those logged commands that match all the selection criteria. Filters can be used in some of the selection fields. To see if a field supports filters, use the field-sensitive help function (PF1). You can also select output and run options in the Command review selection panel, or select no options to process logged commands data as soon as you press Enter. The overview panel that is displayed shows a summary of the logged commands that match your selection criteria, optionally summarized by one of the predefined summary types.

```

Menu          Options      Info      Commands      Setup
-----
                                zSecure Admin - Command review
Command ==>> _____

Show command execution log records that fit all of the following criteria
Userid/logonid/ACID . . _____ (userid or EGN mask)
Job name . . . . . _____ (job name or EGN mask)
System . . . . . _____ (system name or EGN mask)
Profile . . . . . _____ (profile or EGN mask)
Class . . . . . _____ (class or EGN mask)
Ticket id . . . . . _____ (search)

Advanced selection criteria
_ Date time          _ Other

Output/run options
Summarize by _ 1. Ticket id  3. User      5. Command  7. Class+Profile
                2. System   4. Orig. user 6. Component 8. Complex
Ticket ID width . . (1-32)
_ Concise report    _ Suppress pre-exec ESM cmds
_ Recreate commands _ Set ticket id
_ Print format      _ Send as e-mail  _ Background run

```

Figure 5. zSecure Admin command selection criteria panel

In the generated report, several commands might seem to be shown multiple times. This is probably the effect of a single command passing multiple zSecure components. Each component provides unique information that is not available otherwise. For example, if a RACF command is executed remotely using the zSecure Server, the CKXI line (for the command execution module CKX) shows the system where the command originated. The C4RMAIN line (for zSecure Command Verifier) shows the return code of the RACF command.

On the command selection criteria panel, you can use the **Suppress pre-exec ESM cmds** option to suppress certain records created by zSecure components other than zSecure Command Verifier. Records are suppressed if the *command_verb* is one of the RACF commands that might be recorded through zSecure Command Verifier. These criteria work independent of the status of Command Verifier, including whether it is actually installed or not.

In zSecure Admin command review panel, you can select **Advanced selection criteria**. When you select **Date time**, the following panel is displayed:

```

Menu      Options      Info      Commands      Setup
-----
zSecure Admin - Events - Date selection
Command ==> _____

Include CKXLOG records that fit any of the following criteria

Time          From          Until          <  Calendar  >
Date          _____ : _____  December 2018
Weekday       _____ : _____  Mo Tu We Th Fr Sa Su
                                           1 2
                                           3 4 5 6 7 8 9
                                           10 11 12 13 14 15 16
Time zone 1 1. Local (here)    17 18 19 20 21 22 23
                2. Originating system 24 25 26 27 28 29 30
                3. UTC                    31

```

Figure 6. Date/time Command review selection panel

In zSecure Admin command review panel, when you select **Other**, the following panel is displayed:

```

Menu      Options      Info      Commands      Setup
-----
zSecure Admin - Command review
Command ==> _____

Show command execution log records that fit all of the following criteria
Command verb . . . . . _____ (command or EGN mask)
Command return code . . _____ (operator + return code)
Logging component . . . _____ (component or EGN mask)
Originating user . . . _____ (userid or EGN mask)
Ticket description . . _____ (search)

```

Figure 7. Other Command review selection panel

The data for the Command review report is available only when a CKXLOG source is allocated in SETUP FILES.

Detailed field information is available by pressing PF1 on the Command review selection panels or on any field in the display panels.

The following figure shows a sample overview display for the Command review display report:


```

Command history                                     Line 1 of 4
Command ==> _____ Scroll==> CSR
                                           3 Dec 2018 12:11
TimestampHere  Ticket                                Verb  User  Jobname
___ 3Dec2018 12:10 Test 123                            PERMIT  CRMXXX1  CRMXXX1
___ 3Dec2018 12:10 Test 123                            PERMIT  CRMXXX1  CRMXXX1
___ 3Dec2018 12:10 Test 123                            SETROPTS CRMXXX1  CRMXXX1
___ 3Dec2018 12:10 Test 123                            SETROPTS CRMXXX1  CRMXXX1
***** Bottom of Data *****

```

Figure 8. Sample command review overview display

The display includes more information to the right. Scroll to the right to view the following columns:

```

Command history                                     Line 1 of 4
Command ==> _____ Scroll==> CSR
                                           3 Dec 2018 12:11
OrgNode  OrgUser  Comp  RC  Command
___ PLEX1    CRMXXX1  CKXI   0    PERMIT  CKR.CKXLOGID.USE CLASS(XFACILIT) ID(
___ PLEX1    CRMXXX1  CKXI   0    SETROPTS REFRESH RACLIST(XFACILIT) /* POSIT 8
___ PLEX1    CRMXXX1  C4RMAIN 0    SETROPTS RACLIST(XFACILIT) REFRESH
***** Bottom of Data *****

```

Figure 9. Sample command review overview display (2)

In the Command review overview display (Figure 8), you can use the following line commands:

- S (Show additional information) displays the following panel:

```

Command history                                     Line 1 of 25
Command ==> _____ Scroll==> CSR
                                           3 Dec 2018 12:11

Ticket
Ticket id                Test 123
Ticket description       Set CKXLOGID usage

Origin
Originating node        PLEX1
Originating user        CRMXXX1
System name in CVT (SYSNAME) NMPIPL84
System name             ZS13
Security complex name   NMPIPL84
Jobname                 CRMXXX1
User id                 CRMXXX1  ZSECURE TESTER
Component               CKXI

When
Local time of last occurrence 3 Dec 2018 12:10:07.00
Time here of last occurrence 3 Dec 2018 12:10:07.00
Command return code if known

Class Profile
XFACILIT CKR.CKXLOGID.USE

Command
PERMIT  CKR.CKXLOGID.USE CLASS(XFACILIT) ID(CRMB) ACCESS(READ)
***** Bottom of Data *****

```

Figure 10. Show additional information

- R (run command) displays the following panel:

```

zSecure Admin - Confirm command
Command ==> _____

Ticket ID . . Test 1234_____
Description . Another test example_____ More. _

Confirm or edit the following command
PERMIT CKR.CKXLOGID.USE CLASS(XFACILIT) ID(CRMB) ACCESS(READ)_____
_____
_____

Command execution . 1 1. QUEUE RACF command
                        2. QUEUE CKGRACF command (allows use of Reason)
                        3. ASK administrator to execute CKGRACF command
                        4. REQUEST CKGRACF command for later execution
                        5. WITHDRAW CKGRACF command

Specify date for command to be executed
Start date . . . . . _____ (ddmmmyyyy, yyyy-mm-dd or TODAY)
Until/for . . . . . _____ (ddmmmyyyy, yyyy-mm-dd or number of days)
Reason . . . . . _____
Press ENTER to continue or END to cancel the command

***** Bottom of Data *****

```

Figure 11. Run command confirmation panel

Use output option **Recreate commands** to generate command images into CKRCMD for all CKXLOG records that match selection criteria. CKRCMD is shown after you press PF3 from the display of selected CKXLOG records.

When commands were issued multiple times, even with syntax failures, these will be recreated into CKRCMD, ordered by date and time. Before using the RUN command, verify these commands and remove duplicates. When used with output option **Set ticket id** selected, the commands are preceded by CKXLOGID SET with the ticket ID that was originally used for the command. For commands that ran without a valid ID, an empty CKXLOGID SET command is presented.

```

***** Top of Data *****
=NOTE= Enter GO or RUN to execute commands, SUB or SUBMIT to generate batch job
=NOTE= You can also press PF3, enter R at the cursor location, and press ENTER.
000001 /* CKRCMD file CKR1CMD complex IP01 0 generated 12 Apr 2019 02:15 */
000002 /* Pre-execution log not suppressed. Duplicate commands can occur. */
000003
000004 CKXLOGID SET ID('') DESCRIPTION('')
000005
000006 RDEFINE XFACILIT CKNDSN.*.*.LOGR-SMF
000007 RDEFINE XFACILIT CKNDSN.*.*.LOGR-SMF
000008 RDEFINE XFACILIT CKNDSN.*.*.LOGR-SMF
000009 RDEFINE XFACILIT CKNDSN.*.*.LOGR-SMF
000010 SETROPTS REFRESH RACLIST(XFACILIT)
000011 SETROPTS RACLIST(XFACILIT) REFRESH
000012 PERMIT CKNDSN.*.*.LOGR-SMF CLASS(XFACILIT) ID(IBMUSER) ACCESS(NONE)
000013 PERMIT CKNDSN.*.*.LOGR-SMF CLASS(XFACILIT) ID(IBMUSER) ACCESS(NONE)
000014 SETROPTS REFRESH RACLIST(XFACILIT) /* POSIT 8 also affects GXFACILI -
000015 FACILITY */
000016 SETROPTS RACLIST(XFACILIT) REFRESH
000017 RDELETE XFACILIT CKNDSN.*.*.LOGR-SMF
000018 RDELETE XFACILIT CKNDSN.*.*.LOGR-SMF
000019 SETROPTS REFRESH RACLIST(XFACILIT) /* POSIT 8 also affects GXFACILI -

```

Figure 12. CKXLOGID SET

Offloading zSecure command log streams

The zSecure command logger records are written to a z/OS log stream. Data in a log stream is kept according to the specified retention period (RETPD keyword) when defining the log stream. If you need to retain records for a longer period, you can offload the records to a sequential dataset. The offloaded data can be used directly as input source for zSecure reporting. The offloading process can be done using IEBGENER or CKRCARLA. The main advantage of using CKRCARLA instead of IEBGENER is that all the select and exclude criteria for all fields in the CKXLOG newlist are available for selection. You can make a specifically tailored extract of the CKXLOG.

Offloading using IEBGENER

You can use the QSAM interface through the LOGR subsystem to copy records to a sequential data set using IEBGENER. Figure 13 shows an example job. The advantage of using this interface is that it is fast and it supports some selection of the data to be offloaded. Although all *subsys-options* described following the example job can be used, using the FROM and TO keywords is most applicable to an offload process. For example, you can select the records from a specific month and offload to a monthly data set. If you use IBM Workload Scheduler to schedule a monthly offload on the first day of the month, you can use IWS builtin and temporary variables to specify the name of the offload dataset and, for example, use OYYYY-nMO and ODDD-nMO to specify the offload period.

```
//IEBGENER EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT1 DD DSNAME=SYSPLEX.CKXLOG,
// RECFM=VB,LRECL=32756,BLKSIZE=32760,
// SUBSYS=(LOGR,IXGSEXIT,'subsys-options')
//SYSUT2 DD DISP=(NEW,CATLG),DSN=<your.offload.ckxlog>,
// SPACE=(CYL,(10,10)),UNIT=SYSDA,
// RECFM=VB,LRECL=32756,BLKSIZE=32760
```

Figure 13. Example: offload using IEBGENER

For the *subsys-options*, you can specify the following values:

FROM=starting_time

Indicates the starting time of the first log stream block to be processed based on the log stream view that the VIEW keyword specifies. The first block is the one with a time stamp later than or equal to the specified time.

OLDEST

Indicates the first block read is the oldest block on the log stream. OLDEST is the default.

yyyy/ddd

Specifies the start date. If the date is omitted, the current date is assumed. yyyy is a 4-digit year number and ddd is a 3-digit day number from 001 through 366 (366 is valid only on leap years). For example, code February 20, 2000 as 2000/051, and code December 31, 1996 as 1996/366.

hh:mm[:ss]

Specifies the start time. If the time is omitted, the first block written after midnight is used. hh is a 2-digit hour number from 00 to 23, mm is a two digit minute number from 00 to 59, and ss is a 2-digit second number from 00 to 59. The seconds field and associated : delimiter can be omitted if it is not required by the log stream owner.

The FROM keyword is mutually exclusive with the DURATION keyword. The following shows an example FROM specification:

```
'FROM=(2018/331,12:00)'
```

TO=ending_time

Indicates the ending time of the last log stream block that is to be processed based on the log stream view that the VIEW keyword specifies. The last block is the one with a time stamp earlier than or equal to the specified time.

YOUNGEST

Indicates that the last block that is read is the youngest block on the log stream at the time that the DD is allocated. YOUNGEST is the default.

yyyy/ddd

Specifies the end date. If the date is omitted, the current date is assumed. *yyyy* is a four-digit year number and *ddd* is a three-digit day number from 001 through 366 (366 is valid only on leap years). For example, code March 7, 2001 as 2001/066, and code November 12, 2000 as 2000/317.

hh:mm[:ss]

Specifies the end time. If the time is omitted, the last block written before midnight is used. If the end date is the same as the current day, then the youngest block on the log stream at the time the DD is allocated is used. *hh* is a two-digit hour number from 00 to 23, *mm* is a two-digit minute number from 00 to 59, and *ss* is a two-digit second number from 00 to 59. The seconds field and associated: delimiter can be omitted if it is not required by the log stream owner.

The TO keyword is mutually exclusive with the DURATION keyword.

Note: Note: If the value that is specified for the FROM keyword is greater than the value that is specified for the TO keyword, the system ends the job step with a JCL error.

The following shows an example TO specification:

```
'TO=(2019/001,00:00:00)'
```

DURATION=(nnnn,HOURS)

Specifies which blocks are to be processed. Each *n* is a numeric from 0 to 9. Specifying *(nnnn,HOURS)* requests the blocks for the last *nnnn* hours up to the youngest block that is to be processed based on the log stream view that the VIEW keyword specifies. The last *nnnn* hours are calculated from the current time of the allocation for the DD. The first block is the one with a time stamp greater than or equal to the calculated start time. The last block that is read is the youngest block on the log stream when the DD is allocated.

The DURATION keyword is mutually exclusive with the TO and the FROM keywords.

The following shows an example DURATION specification:

```
'DURATION=(48,HOURS)'
```

VIEW=ACTIVE|ALL|INACTIVE

This keyword does not apply to the zSecure Command Logger log stream. The zSecure Command Logger does not create INACTIVE log data.

GMT|LOCAL

Specifies whether the time is local time (based on the time zone offset at the time the log was written) or GMT time. GMT is the default.

Offloading using CKRCARLA UNLOAD

You can also use CKRCARLA to offload data from the active log stream to a sequential data set. You can use an explicit allocation through JCL, as described for the IEBGENER offload, or use one of the supported options as described for the CARLa ALLOCATE statement (see *zSecure CARLa Command Reference*).

The following is JCL sample C2RJXUNL that shows a CKRCARLA unload job that unloads yesterday's CKXLOG records:

```
//JCLLIB  JCLLIB ORDER=(your.prefix.CKRPARM,
//        #thlq.SCKRPROC)
//*
//C2RJUNFL EXEC C2RC,DSTAT=(NEW,CATLG),MEMBER=C2RLXUNL,ALLOECKF=0,
//        CONFIG=C2R$PARM
//*
//CKXLOGIN DD DSNAME=&SYSPLEX..CKXLOG,
//        RECFM=VB,LRECL=32756,BLKSIZE=32760,DISP=SHR,
//        SUBSYS=(LOGR,IXGSEXIT,'DURATION=(48,HOURS)') Yesterday
//*
//CKXLUNLD DD DISP=(&DSTAT),
//        DSN=&DPREF..&SYSPLEX..D&LYR2.&LJDAY..XL,
//        UNIT=&UNIT,VOL=SER=&VOLSER,
//        SPACE=(27998,(300,300),RLSE,,ROUND),
//        DCB=(LRECL=32760,BLKSIZE=&BLKSIZE,RECFM=VBS)
```

Note that the Julian date in the data set name is the unload date, so one day later than the timestamp of the records in the data set.

Offloading using CKRCARLA LIST

You can also use CKRCARLA LIST to offload data from the active log stream to a sequential data set. You can use an explicit allocation through JCL, as described for the IEBGENER offload, or use one of the supported options as described for the CARLa ALLOCATE statement.

The following is an example showing a CKRCARLA offload job:

```
//CKRCARLA EXEC PGM=CKRCARLA,REGION=64M
//STEPLIB  DD DISP=SHR,DSN=<your.SCKRLOAD>
//SYSPRINT DD SYSOUT=*
//CKXUNLD DD DISP=(NEW,CATLG),DSN=<your.offload.ckxlog>,
//        SPACE=(CYL,(10,10)),UNIT=SYSDA,
//        RECFM=VB,LRECL=32756,BLKSIZE=32760
//SYSIN    DD DATA,DLM='\/'
           print dd=ckxunld
           suppress ckfreeze
           suppress racf
           alloc type=ckxlog dsn=adcdpl.ckxlog,
           subsys=(LOGR,IXGSEXIT,'FROM=(2018/244,00:00),TO=(2018/274,00:00)')
           n type=ckxlog nopage
           /* insert additional select statements */
           list record(0,asis)
\
```

CR.1 Run RACF commands from library

Commands that were generated by a zSecure panel query can be saved into a member of a partitioned data set (PDS), by using the W line command on the RESULTS panel in front of the CKRCMD entry. Third party applications can also write TSO commands into PDS members, so RACF administrators can later run these commands.

The CR.1 Libraries option enables you to review members, modify commands in members, run these commands on the local RACF database, use zSecure Network Server to send commands to a remote node, or submit a job to run the commands. The CR.1 option shows a list of data set names that was previously entered by a user or maintained by the installation administrator using option SE.D.8.

```
zSecure Admin+Audit for RACF                               Row 1 of 3
Select TSO command library or work with a library (S, E, R, I, or D)

TSO command library          Type      Status
DD:CKRCMD                   CKRCMD
'RACFADM.A.CLIST'           CKRCMD
'TEAM.RACF.COMMAND.QUEUE'  CKRCMD
***** Bottom of data *****
```

Figure 14. Command library list in CR.1

The data sets can be partitioned or sequential, although most users would prefer PDS(E). Library type CKRCMD defines the contents as RACF and other TSO commands. The following line commands are recognized:

- S** Show members of PDS, or edit contents of sequential data set.
- E** Same as S.
- E** Delete entry from list.
- I** Insert empty entry in list.
- R** Repeat entry.

You can select a library by typing the S line command in front of the library of your choice. If the entry specifies a sequential data set, the contents is displayed into ISPF Edit. For a PDS, the member list is shown and members can be selected. Special purpose entry DD:CKRCMD provides access to commands from the most recent zSecure panel query. If the user has multiple COMPLEX values selected from SETUP INPUT (SE.1), a table will be shown with a CKRCMD file for each COMPLEX, and each file can be edited separately.

When reviewed, the commands can be modified using normal ISPF Edit commands. To run the commands from the editor, enter G0 or RUN in the command line. The location where the commands will run depends on several criteria:

- With SETUP CONFIRM (SE.4), the Command Routing option can be set to 3 (Local only); commands are then always applied to the local RACF database.
- If Command Routing is set to 1 (Ask), a list of destinations is shown, including remote zSecure Network Server nodes.
- If Command Routing is set to 2 (Normal), the selected Set of Input Files from SETUP INPUT (SE.1) determines where the commands execute.

|
|
|

- With the SUBMIT command, a batch job is generated to run the commands. The execution node is retrieved from the parameters in the SETUP INPUT (SE.1) entry, or from the contents of the selected UNLOAD data set.

Chapter 5. zSecure CARLa Command Reference

This chapter lists the documentation updates for *zSecure CARLa Command Reference* as a result of the zSecure Command and Ticket Logging SSE.

Chapter 1. CARLa Command Language

- Updated: section “**ALLOCATE**”:
 - Updated: section “Explicit allocation mode”. The following rows were updated:

Table 7. File types for allocating data sources for zSecure

CKXLOG	Command-log. The CKXLOG started task creates these log records. They can be formatted with NEWLIST TYPE=CKXLOG. Use of the keyword ACTIVE selects the last 24 hours from the log stream that the CKXLOG started task currently uses.
SMF SMFSTREAM	Use TYPE=SMFSTREAM for allocating an SMF log stream. This file is for SMF NEWLIST processing. You cannot specify parameters to limit the number of returned SMF records. If you must specify a limit, use TYPE=SMF with the SUBSYS keyword. If SMF is written to a log stream, the keyword ACTIVE selects the last 24 hours from the log stream.

- Updated: section “Current system input source parameters”. The following sentence was updated:
Specifying to the program that these data set names are to be dynamically retrieved can be done for RACF, ACF2, CKXLOG, and SMF data sets.
- Updated: section “**LIST family of commands**”:
 - Updated: section “Conditional field include”.
 - Updated: section “Format names”. Output format KERBSECLVL was added:

KERBSECLVL

This format is in use for newlist IP_FTP_REGION fields like SECURE_CTRL_CONN and SECURE_DATA_CONN. It can be specified in CARLa in order to generate test cases for compliance rules with an overriding DEFINE statement. Possible values are:

- 1: NEVER
- 2: CLEAR
- 3: PRIVATE
- 4: SAFE
- other: ?

- Updated: section “**UNLOAD**”: The following information was updated:

RACF systems

ACCESS, ASSERT, RACF, SMF, CKXLOG, and deftype files

The unload of SMF, ACCESS, CKXLOG, and deftype data requires that the UNLOAD statement is inside the scope of a NEWLIST statement.

Chapter 2. SELECT/LIST Fields

Added: section “CKXLOG: command-log records” was added for the new CKXLOG newlist.

Conditional field include

This section describes the construct to conditionally include output fields specified on LIST family commands.

The syntax of the conditional field include is as follows:

```
( symbolic=value ? fields_and_operators )
```

or

```
( symbolic<>value ? fields_and_operators )
```

The *symbolic* must be defined with the SYMBOLIC statement. The *value* assigned to it provides the condition for including the *fields_and_operators* in the report.

If the *symbolic* is not defined or has a different value:

- For relational operator =: the *fields_and_operators* are not included.
- For relational operator <>: the *fields_and_operators* are included.

The operators that are allowed within the *fields_and_operators* part are |, /n, and /.

To conditionally include part of a LIST family statement, surround it with brackets and include the *symbolic=value* clause after the opening bracket, followed by the question mark (?) separator.

CKXLOG: command-log records

Administration z/OS		Manager for RACF z/VM	Compliance and Auditing z/OS				Adapters for SIEM
Admin	Visual		Audit for RACF	Audit for ACF2	Audit for TSS	Alert	
.							

This report type formats command-log records. These are allocated with ALLOC TYPE=CKXLOG. Each record shows the command that a component issued. If you have the zSecure Command Verifier product installed, the RACF commands can be logged twice; once as it is issued by a zSecure Admin function like CKX (shown as component CKXI), CKGRACF, or CKRCARLA, and once as it gets modified and extended by Command Verifier (component C4RMAIN). There are two similar records, mainly differing in the component field as shown in the record. In the ISPF user interface, you can use the option **Suppress pre-exec ESM cmds**. This field exploits the field IN_C4R_SCOPE to exclude all records that zSecure Admin logs for RACF commands that might also be logged by Command Verifier.

The log files are assumed to be sorted chronologically by ETOD clock. If the input is not in ascending ETOD sequence, CKRCARLA issues a warning message. Duplicate records with the same ALLOC VERSION are suppressed. The LIST command provides a storage efficient way to create a chronological report. The UNLOAD command can be used to offload a log stream to a data set. It can also be used to create a merged data set copy of multiple offload data sets, or a subset of it by using the SELECT command.

CKXLOG fields

Admin	Audit for RACF	Audit for ACF2	Audit for Top Secret	Manager for RACF z/VM	Command syntax
▪					[CLASS, C] COMMAND COMMAND_RC COMMAND_VERB COMPLEX COMPONENT DATETIME DATETIME_RUNTZ DDNAME ETOD IN_C4R_SCOPE JOBNAME ORIGIN_NODE ORIGIN_USER PROFILE RECNO RECORD SYSNAME SYSTEM TICKET_DESC TICKET_ID TOD USER

CKXLOG field descriptions

The CKXLOG newlist provides the following fields for reporting.

CLASS, C

An eight-character class name as derived from the command, or empty.

For the RACF SETROPTS command, the value setropts is shown if the command specifies multiple classes or if the command does not apply to a specific class.

COMMAND

The security administration command logged by zSecure. If the command is recognizable as a RACF command, it will be normalized to make it easier to search for keywords. If you have zSecure Command Verifier installed, RACF commands can be intercepted and logged even when issued from the TSO command line. In a detail DISPLAY, the field supports the action R (Run) to issue the command. The default width is 64.

COMMAND_RC

The return code of the command. Whether this field is filled in depends on which component created the logging record. Only C4RMAIN (component name for zSecure Command Verifier) fills the return code; the other components log the command before it is passed to the operating system. The default width is 3. The value 254 implies RC 254 or higher; only one byte is stored in the log record.

COMMAND_VERB

The command verb of the logged command. The default width is 8. Note that for UNIX commands, the verb is !UNIX.

COMPLEX

Identifies the security complex name.

If the ALLOC COMPLEX parameter is specified, that value is used, except when the ZSECSYS=* or ZSECNODE=* parameter is used and the ALLOCATE statement results in the allocation of multiple nodes.

For z/OS RACF systems:

- The default value is the ZSECNODE name if the data is obtained via the zSecure server.
- The default value is taken from the CKFREEZE or UNLOAD data sets if the data is obtained directly from these data sets.
- The default value is the RRSFNODE name of the system if the system uses RRSF, the information is available, and the RRSFNODE is not specified as LOCAL.

Note: The RRSF node name can be obtained only if the z/OS release is 2.2 or higher.

- The default value is taken from the SYSPLEX name if there is no valid distinct RRSFNODE name available, the information is available, and the SYSPLEX name is not specified as LOCAL.
- Otherwise, the default value is taken from the SYSNAME.

In restricted mode, complex names can be overridden only to match either the ZSECNODE, RRSF node (for z/OS RACF systems), SYSPLEX name, or SMF ID.

If the ALLOC statement for a CKFREEZE data set contains a VERSION= parameter, a blank and the 4-character version are appended to the 8-character complex name on output. To display the version in the report output, use an output length modifier on the COMPLEX field and specify a value of 13 or greater, or 0. For more information, see Modifying output length.

COMPONENT

The product component that logs the command. This can be CKXI (command execution module CKX), CKGRACF, CKRCARLA, C4RMAIN, or CKXLOGID. The default width is 8.

DATETIME

Date and time in centi-second precision (SMF timestamp format) of the event, in the time zone of the system where the command was logged. The default width is 15.

DATETIME_RUNTZ

Date and time in centi-second precision (SMF timestamp format) of the event, in the time zone of the system that the report runs on. The default width is 15.

DDNAME

File name from which the record was read. See also RECNO. The default width is 8.

ETOD

The ETOD clock of when the log request reached CKXLOG, in hexadecimal format.

Presumably, the ETOD is UTC/GMT. This field is part of the record key for the CKXLOG newlist. Also, CKXLOG input files are assumed to contain records in ascending ETOD order. So if you want a chronological report, instead of SORTLIST, you can use a LIST command, which uses less storage. The default width is 32.

IN_C4R_SCOPE

Flag field that is true when the command verb is within the scope of the zSecure Command Verifier product. If Command Verifier is running then the RACF command as Command Verifier modified it can also turn up in a component C4RMAIN record. The flag is not influenced by whether or not Command Verifier is actually installed and running, or the effects of Command Verifier policies.

JOBNAME

The job name of the address space that issues the command, on the system where the command is logged. The default width is 8.

ORIGIN_NODE

The node name where the command originated. For example, this can be the submitting JES node for a batch job, a remote zSecure system for a command that is issued through CKNSERVE, or an RRSF node name. The default width is 8.

| **ORIGIN_USER**

| The user ID that issued the original command. For example, this can be the
| submitting user ID for a batch job or a remote user ID for a command that is
| issued through CKNSERVE. The default width is 8.

| **PROFILE**

| A character profile name as derived from RACF commands, up to 246
| characters. The default width is 44.

| For the RACF SETROPTS command, the value setropts is shown.

| **RECNO**

| Record number within the DDNAME. The default width is 10.

| **RECORD**

| The as-is content of the input record. The field is meant for debugging
| purposes. The default width is 72.

| **SYSNAME**

| The z/OS SYSNAME of the system where the command was issued and
| logged. The default width is 8.

| **SYSTEM**

| The SMF ID of the system where the command was issued and logged. The
| default width is 8.

| **TICKET_DESC**

| A description for the current ticket identification as set by the user at the time
| that the command was issued. The default width is 64.

| **TICKET_ID**

| The current ticket identification as set by the user at the time that the
| command was issued. The default width is 32.

| **TOD**

| The TOD clock in UTC format when the command was logged. The default
| width is 13.

| **USER**

| The current user ID at the time that the command was issued. The default
| width is 8.

Chapter 6. zSecure Command Verifier User Guide

This chapter lists the documentation updates for *zSecure Command Verifier User Guide* as a result of the zSecure Command and Ticket Logging SSE.

In Chapter 4. Auditing commands and policy effects, the section **Using the zSecure Admin Command Logger** was added.

Using the zSecure Admin Command Logger

Aside from the Command Audit Trail (CAT) that the previous section describes, and the command auditing functions that the next section describes, zSecure Command Verifier also provides the option to write the complete command text to the zSecure Admin Command Logger (CKXLOG).

The CAT has an advantage that it records the essential part of a profile change in the profile itself. There is no need to scan a large amount of data to quickly find who changed the profile and when that was done. The SMF support, as the next section describes, allows a separate record of the command as the user enters it, and the command as it is modified and approved by the zSecure Command Verifier policy profiles. The zSecure Admin Command Logger function is a mix between these two options. The complete command, resulting from the application of the policy profiles, is written to a log stream. Depending on the record retention period, the log stream might be large, and you must scan many records to locate the relevant ones. The log stream contains all the commands, while the CAT only contains information about the latest change. The CKXLOG log stream is smaller than the full SMF files.

The zSecure Admin Command Logger logs commands depending on the component that passes the command and the user's access to this component. Switch profiles of the form CKX.CKXLOG.LOG.*component-name* control how commands are logged. The *component-name* for Command Verifier is C4RMAIN. For additional information about these switch profiles, see *zSecure CARLa-Driven Components Installation and Deployment Guide*. The zSecure Admin Command Logger uses these switch profiles independently of the Command Verifier policy profiles that are described in this section. To successfully log a RACF command, the terminal user must have the appropriate access to the Command Verifier =CKXLOG policy profile, and to the CKXLOG switch profile.

The record in the CKXLOG log stream might contain information about the origin of the RACF command: For commands routed through RRSF, the terminal user's node and user IDs are included. If the command was issued from a batch job, the submitting node and user IDs are included. The record also contains the RACF return code from the command. Reporting about the RACF commands can be done either by browsing records offloaded from the log stream, or using zSecure Admin. zSecure Admin provides the CKXLOG newlist and the CR menu option in the ISPF interface. For more information about installing the required CKXLOG started task, see *zSecure CARLa-Driven Components Installation and Deployment Guide*. For more information about reporting through zSecure Admin see *zSecure Admin and Audit for RACF User Reference Manual*.

zSecure Command Verifier logs the RACF commands through CKXLOG after the commands are executed. For example, if RACF already rejected a command because of invalid syntax, the command is not passed to zSecure Command Verifier and the command is not logged. This in contrast to the zSecure Admin interface, which logs the commands before they are executed.

If the CKXLOGID command was issued before the RACF command, the record in the CKXLOG log stream is annotated with the specified ticket identification and description. If the CKXLOGID command was not issued, or if the ticket information expired, a warning message might be issued to inform the terminal user that no ticket information was added. This message can be suppressed by allowing UPDATE access to the policy profile. For more information about the CKXLOGID command see *zSecure Admin and Audit for RACF User Reference Manual*.

Recording RACF commands to the zSecure Admin Command Logger (CKXLOG) is controlled through profiles of the following format:

```
C4R.command.=CKXLOG
```

This profile specifies that the RACF command, as optionally modified and approved by the zSecure Command Verifier policy profiles, is recorded in the CKXLOG log stream. If the command is rejected, zSecure Command Verifier does not record the command.

The *command* is the non-abbreviated RACF command as the terminal user issues it. The qualifier =CKXLOG in the policy profile cannot be covered by generic characters. It must be present in the exact form as shown.

It is expected that many installations define a generic policy profile with UACC(READ) to cover all RACF commands and separate policy profiles for the RACF LIST commands with UACC(NONE):

Profile	Universal access
C4R.*.=CKXLOG	READ
C4R.LIST*.=CKXLOG	NONE
C4R.RLIST.=CKXLOG	NONE

The following access rules apply:

No Profile Found

This control is not implemented. The command is not logged through CKXLOG.

NONE

This control is not active for the terminal user. The command is not logged through CKXLOG.

READ The command as optionally modified and approved by zSecure Command Verifier is logged through CKXLOG. If no ticket information is present, or if the CKXLOG server is not active, a warning message is issued.

UPDATE

The command as optionally modified and approved by zSecure Command Verifier is logged through CKXLOG. Warning messages about missing ticket information or an inactive CKXLOG server are suppressed.

CONTROL

Same as UPDATE.



Printed in USA