Natural

SAF Security Kernel

Version 9.2.4

October 2025

ADABAS & NATURAL

This document applies to Natural Version 9.2.4 and all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Document ID: SAK-DOC-924-20251013

Table of Contents

Preface

This documentation is organized under the following headings:

Introduction	Provides an overview of the SAF Security Kernel functionality.
Installation	Describes how to install the SAF Security Kernel.
Operator Commands	Explains the available operator commands for the SAF Security Kernel.
SAF - SAF Daemon Messages*	Describes the SAF daemon messages.
SEFM - ADASAF SAF Interface and SAF Security Kernel Messages*	Describes Adabas SAF Security (ADASAF) and SAF Security Kernel operator console and command messages.
SAF Return Codes	Describes SAF return codes.
SAF Internal Function Codes	Describes SAF internal function codes.
Interpreting Trace Messages	Describes how to interpret SAF trace messages
Security Definitions	Provides a general overview of the definition of resources to RACF, CA-Top Secret and CA-ACF2.

1 About this Documentation

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
`Monospace font`	Identifies service names and locations in the format *`folder.subfolder.service`*, APIs, Java classes, methods, properties.
`Italic`	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
`Monospace font`	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
|	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the | symbol.
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Product Documentation

You can find the product documentation on our documentation website at **https://documentation.softwareag.com**.

Product Training

You can find helpful product training material on our Learning Portal at **https://learn.softwareag.com**.

Tech Community

You can collaborate with Software GmbH experts on our Tech Community website at **https://techcommunity.softwareag.com**. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software GmbH news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at **https://github.com/softwareag** and **https://hub.docker.com/publishers/softwareag** and discover additional Software GmbH resources.

Product Support

Support for Software GmbH products is provided to licensed customers via our Empower Portal at **https://empower.softwareag.com**. Many services on this portal require that you have an account. If you do not yet have one, you can request it at **https://empower.softwareag.com/register**. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

2 Introduction

This documentation describes the SAF Security Kernel and its associated SAF Security Daemon, which may also be referred to as the SAF Server. It covers installation and operation of the kernel and daemon and messages and codes issued by them. The SAF Security Kernel and Daemon are distributed on the Adabas Limited Libraries (product code WAL).

The System Authorization Facility (SAF) is used by z/OS and compatible sites to provide rigorous control of the resources available to a user or group of users. Security packages such as RACF, CA-ACF2, and CA-Top Secret allow the system administrator to

- maintain user identification credentials such as user ID and password; and
- establish profiles determining the datasets, storage volumes, transactions, and reports available to a user.

The resulting security repository and the infrastructure to administer it represent a significant investment. At the same time, the volume of critical information held by a business is constantly growing, as is the number of users referencing the data. The challenge of controlling these ever-increasing accesses requires a solution that is flexible, easy to implement and, above all, one that safeguards the company's investment.

The SAF Security Kernel acts as an agent for other products such as Adabas, Natural, and Entire Net-Work. It allows them to secure resources via a SAF-compliant security system, thus enhancing the scope of the security system to enable:

- a single control and audit system for all resources
- a single definition of userids and passwords
- industry standard protection of resources such as Adabas data and Natural libraries
- maximized return on investment in the security repository

Architecture

A SAF security solution comprises two separate components:

- a product-specific component which is distributed and installed with the product being protected (Adabas, Natural, Entire Net-Work or EntireX)
- a product-independent SAF Security Kernel (the subject of this document) which may be embedded in an authorized product or operate as a separate authorized daemon

Related Documentation

For details on securing specific products such as the following, refer to the relevant product documentation:

- Adabas SAF Security
- Natural SAF Security
- Entire Net-Work
- EntireX Security

Some of these products are distributed with a copy of the SAF kernel. The individual product documentation indicates if this is the case.

Password Phrases

The SAF Security Kernel provides password phrase support with Adabas Limited (WAL) Library Version 8.3.4 (or above) for the following products:

- Adabas SAF Security
- Entire Net-Work
- EntireX Security

In addition, the SAF Security Kernel provides password phrase support with Adabas Limited (WAL) Library Version 8.4.3 Load Update 1 (or above) for the following product:

Natural SAF Security 8.2.7 (or above) in conjunction with fix SF97005.

zIIP Support

The SAF Security Kernel is compatible with the following zIIP implementations:

- Adabas SAF Security Version 8.2.2 (or above) running in a zIIP-enabled Adabas nucleus.

  Refer to the section *Using COR-based Add-ons* in the *Adabas Release Notes* relevant to the Adabas version you are running for any special considerations regarding this type of implementation.
- Adabas SAF Security Version 8.2.2 (or above) running in a zIIP-enabled Adabas System Coordinator daemon (Version 8.3.1 or above).

Refer to the section *Implementing Adabas System Coordinator for zIIP* in the *Adabas System Coordinator z/OS Installation guide* for any special considerations regarding this type of implementation.

Although the current SAF Security Kernel provides compatibility with the above zIIP implementations, enhanced zIIP support is provided with Adabas Limited (WAL) Library Version 8.4.3 Load update 1 (or above) in conjunction with Adabas SAF Security Version 8.2.2 fixes AX822013, AX822014, and AX822015.

Support for ENF Signals

With Adabas Limited Library (WAL) version 8.5 SP4 Patch level 1 and above, the SAF Security Kernel supports ENF signal types 62, 71, and 79 both when installed with another product being protected (Adabas, Natural, Entire Net-Work, EntireX) or when running in a separate authorized daemon.

- An ENF signal type 62 may be issued to listeners when a `SETROPTS RACLIST` command affects in-storage profiles used for authorization checking.
- An ENF signal type 71 may be issued to listeners when a `CONNECT`, `REMOVE`, `ALTUSER REVOKE`, `DELUSER`, or `DELGROUP` command has affected a user's group connections.
- An ENF signal type 79 may be issued to listeners when a `PERMIT`, `RDEFINE`, `RALTER`, or `RDELETE` command has affected a user's or group's authorizations to resources. However, note that the SAF Security Kernel only supports an ENF signal type 79 which affects a user's authorization to resources.

Refer to your security package documentation for detailed information regarding how and when these signal types are issued.

Listening to these signals is implemented using configuration parameters, the default being not to listen. For product specific information, refer to the relevant product documentation.

Signal Listeners are automatically shut-down at job termination. To shut-down any active listeners while the job remains active, use the **SSIGTERM operator command**. Terminated listeners can only be restarted by stopping and restarting the job.

3 Installation

This section describes how to install the SAF Security Kernel.

Prerequisites and Considerations

This section describes the prerequisites and considerations for installing the SAF Security Kernel.

Operating Systems

The SAF Security Kernel is compatible with the z/OS operating system.

Security Systems

The SAF Security Kernel is compatible with all SAF-compliant security systems such as ACF2, RACF, and Top Secret.

Software Prerequisites

The SAF Security Kernel uses the common SAF components provided on the Adabas Limited Library, widely known as the WAL libraries.

Products which require the use of the SAF Security Kernel can be used with any supported version of the WAL library. However, specific product features may require a certain version of the WAL library. The relevant product feature description indicates whenever this is the case.

APF Authorization

The SAF Security Kernel load library and any other step libraries in its loading environment must be APF authorized.

Installation Datasets

The SAF Security Kernel is supplied as a component of the Adabas Limited Libraries (product code WAL).

Dataset	Description
WAL*vrs*LOAD	Standard load library containing modules (prefixed SAF*) required to operate the SAF Security Kernel.
WAL*vrs*SRCE	Standard source library containing Assemble macros (prefixed NA2M*) source books and, sample input (prefixed SAF*).
WAL*vrs*JOBS	Standard jobs library containing sample installation jobs (prefixed SAFI*).

Installation Procedure

This section describes how to install the SAF Security Kernel.

Step 1: Creating the SAFCFG Configuration Module

(Sample Job SAFI010)

The SAFCFG configuration module is created by assembling a source member. Refer to the sample SAFPARMS member supplied on the SRCE library. This source member invokes the SAFCFG macro, also supplied on the SRCE library, specifying your site-specific options and requirements.

The SAF Security Kernel uses the settings in SAFCFG to determine:

- The protected resources for each product
- Security classes for resource checking
- The composition of different resource profile names
- The caching requirements

The resulting load module SAFCFG must be available to any job that includes the SAF Security Kernel. If appropriate, you may decide to maintain different SAFCFG modules for different secured products.

The following SAFCFG parameters are the minimum required for an initial installation of the SAF Security Kernel.

Note: Refer to the appropriate product documentation for which the SAF Security Kernel is being installed to determine what product specific SAFCFG parameters are required to be set in addition to, or in conjunction with, the following parameters.

GWSIZE: Storage Size for Caching User Information

Parameter	Description	Syntax
GWSIZE	The amount of storage in kilobytes used for caching user information. Generally, size this parameter based on approximately 512 bytes per user. Individual product caching requirements affect the amount of storage required. Refer to the respective product documentation for any specific cache sizing estimations.	GWSIZE={256|*nnnn*}

GWMSGL: Trace Level for Security Checking

Parameter	Description	Syntax
GWMSGL	The tracing level used for security checks. Valid values are: ■ 0 – no tracing ■ 1 – trace violations only ■ 2 – trace successful checks only ■ 3 – trace all checks Use the parameter SAFPRINT to control where the trace messages are written and, for an interpretation of the trace message content, refer to section *Interpreting Trace Messages* in the *SAF Security Kernel* documentation. These trace messages are retained for as long as the job or the dataset, to which they have been written, remains available. Deleting the job or dataset deletes the trace messages. For diagnostic and troubleshooting purposes, the content of the trace message includes the SAF User ID for which access is requested.	GWMSGL={0|1|2|3}

GWSTYP: SAF Security System Type

Parameter	Description	Syntax
GWSTYP	The type of SAF Security system in use. Valid values are: ■ 1 – RACF ■ 2 – Top Secret ■ 3 – ACF2	GWSTYP={1|2|3}

SAFPRINT: Security Check Trace Message Printing

Parameter	Description	Syntax
SAFPRINT	The location where the security check trace messages (see parameter GWMSGL) are written. Valid values are: ■ N – security check trace messages are written to DD DDPRINT ■ Y – security check trace messages are written to DD SAFPRINT If you specify SAFPRINT=Y, but you miss to provide a SAFPRINT dataset, trace messages are written to DDPRINT. You must define the SAFPRINT dataset in the SAF Security Daemon JCL. The dataset might refer to a SYSOUT dataset or to a file defined with RECFM=F (or FB) and LRECL=121.	SAFPRINT={N|Y}

Step 2: Creating the SAFPSEC Security Module

(Sample Job SAFI020)

Create the SAFPSEC security module using the sample assembly job SAFI020. Specify the appropriate STY= and REL= parameter values for your security system.

The resulting load module SAFPSEC must be available to any job that includes the SAF Security Kernel.

STY: SAF Security System Type

Parameter	Description	Syntax
STY	The type of SAF Security system in use. Valid values are: ■ RACF ■ TSS – Top Secret ■ ACF2	STY={RACF|TSS|ACF2}

REL: Release level of parameter list generated by RACROUTE

Parameter	Description	Syntax
REL	The release level of the parameter list generated by the SAFPSEC RACROUTE macros.	REL={7730|*nnnn*}

Step 3: Creating the SAFPMAC environment module

(Sample Job SAFI021)

Create the SAFPMAC environment module using the sample assembly job SAFI021 and the SRCE member SAFPOS. The resulting load module SAFPMAC must be available to any job that includes the SAF Security Kernel.

Step 4: Identifying the Appropriate Installation Mode

You can install the SAF Security Kernel in two modes:

- Embedded within a product. The SAF Security Kernel runs in the same address space as the product.
- Running in its own Daemon address space as a target in the network.

The following table shows the installation mode for the most common products:

Product	Installation Mode
Adabas SAF Security	Embedded installation mode
Natural SAF Security	Daemon installation mode

Identify which installation mode is appropriate for the product you are installing and continue with Step 4a or Step 4b accordingly.

Note: For both installation modes, the SAF Security Kernel must run under a defined user ID. This user ID must have sufficient authority to invoke the `AUTH`, `VERIFY`, and `EXTRACT` functions of the RACROUTE external security manager interface and to issue third-party checks on behalf of all users.

- Step 4a: Embedded Installation Mode
- Step 4b: Daemon Installation Mode

Step 4a: Embedded Installation Mode

This section describes the steps for installing the SAF Security Kernel in embedded mode.

For embedded installation mode, you must only add the load library containing the SAF Security Kernel (SAFKRN) and the three load modules SAFCFG, SAFPSEC, and SAFPMAC, created in Steps 1 to 3 above, to the step library concatenation applicable to the installation of the product.

Step 4b: Daemon Installation Mode

(Sample Job SAFI024)

This section describes the steps for installing the SAF Security Kernel in daemon mode.

For daemon installation mode, the SAF Security Kernel runs in its own Daemon address space using Adabas modules to establish itself as a target in the network. Consequently, the SAF Security Daemon (and therefore its Kernel) can be accessed remotely via Entire Net-Work if configured appropriately.

It is recommended that you run the SAF Security Daemon as a started task, although it can be run as a batch job. The SAF Security Daemon must run APF-authorized, therefore all step libraries must be APF-authorized. Sample JCL to execute the SAF Security Daemon is provided in SAFI024 in the JOBS library.

Daemon Runtime Parameters

You configure the SAF Security Daemon with runtime parameters in a similar way to an Adabas nucleus – using a DDCARD input dataset at startup. A sample DDCARD input is provided in the SRCE member SAFDDCAR.

Following is an explanation of the daemon runtime parameters.

CT Parameter: Command Timeout Limit

Parameter	Description	Minimum	Maximum	Syntax
CT	The maximum time in seconds for interregion communication of results from the daemon to the user.	1	2147483647	CT={60|*nn...*}

The maximum number of seconds (more precisely, units of 1.048576 seconds) that can elapse from the time a user request has been completed until the results are returned to the user through inter-region communication.

Use this parameter to prevent a command queue element (see NC parameter) and an attached buffer (see NABS parameter) from being held for a long period of time for a user who has terminated abnormally.

If the CT limit is exceeded:

- The command queue element and attached buffer are released.
- A message ADAM93 is printed.
- Response code 254 (ADARSP254) is returned to the calling user if the user is not terminated.

DEFAULT Parameter: Default Product

Parameter	Description	Syntax
DEFAULT	The default product to which user requests are passed.	DEFAULT=SAF

The DEFAULT parameter should be set to SAF.

FORCE Parameter: Allow Daemon ID Table Overwrite

Parameter	Description	Syntax
FORCE	Specifies whether the daemon can overwrite an existing ID table entry.	FORCE={NO|YES}

When a daemon starts up, it scans the ID table to ensure that no entry exists for the ID specified by the NODE parameter. You can use the FORCE parameter to indicate whether the daemon can overwrite an existing entry.

Caution: Do not use the FORCE parameter unless absolutely necessary or the integrity of the daemon could be lost. Ensure that no server of any kind is active for the ID table entry being overwritten.

LOCAL Parameter: Local Daemon

Parameter	Description	Syntax
LOCAL	Specifies whether the daemon is isolated and available for local use only.	LOCAL={NO|YES}

If LOCAL=NO, the daemon is reachable by Entire Net-Work and therefore accessible from remote users.

If LOCAL=YES, the daemon is unreachable by Entire Net-Work and therefore not accessible from remote users.

LU Parameter: Maximum Size of User Request

Parameter	Description	Minimum	Maximum	Syntax
LU	The maximum allowable size of a user request to the daemon.	none	none (see note)	LU={65535|*nn..*}

The LU parameter specifies the maximum allowable size of a user request to the daemon. It should be set to 65535.

Note: An error occurs if the LU parameter is equal to a value greater than the byte count implied by the NABS (number of attached buffers) parameter. LU cannot exceed a value greater than that produced by the following calculation: NABS_value*4096.

MPMWTO Parameter: Daemon Informational Messages

Parameter	Description	Syntax
MPMWTO	Specifies whether the daemon issues informational messages to the operator console.	MPMWTO={NO|YES}

If MPMWTO=NO, the daemon does not send informational messages to the operator console.

If MPMWTO=YES, the daemon sends informational messages to the operator console.

NABS Parameter: Number of Attached Buffers

Parameter	Description	Minimum	Maximum	Syntax
NABS	The number of attached buffers used.	1	32767	NABS={16|*nn..*}

The number of 4K storage blocks to be used for transmitting information between the user and the daemon. If a request to the daemon fails with a response code 255, increase the value of NABS.

NC Parameter: Number of User Request Queue Elements

Parameter	Description	Minimum	Maximum	Syntax
NC	The maximum number of user request queue elements.	1	32767	NC={20|*nn..*}

The maximum number of user requests that can be queued or be in process at any time. If a request to the daemon fails with a response code 151, increase the value of NC.

NODE Parameter: Daemon Identification

Parameter	Description	Minimum	Maximum	Syntax
NODE	The physical ID of the daemon.	1	65535	NODE=*nnnnn*

The physical ID of this SAF Security Daemon. Products requiring the SAF Security Kernel to run in Daemon installation mode must specify this same Node ID to the SAFCFG parameter GWDBID. Refer to the appropriate product documentation for more information.

PRODUCT Parameter: Product Availability

Parameter	Description	Syntax
PRODUCT	Specifies the products that are available in this daemon.	PRODUCT=SAF

The PRODUCT parameter should be set to SAF.

SAF PARM Parameter: SAF Configuration Module Name

Parameter	Description	Syntax
SAF PARM	The SAF configuration module name.	SAF PARM={SAFCFG|*aaaaaaaa*}

If you need to change the name of the default configuration module SAFCFG (for example, you have different configuration modules with different settings), you can specify the name of the configuration module the daemon uses.

SVC Parameter: SVC Number

Parameter	Description	Syntax
SVC	The Adabas SVC number used.	SVC={0|*nnn*}

The SVC number you specify must correspond to the number used for the Adabas SVC at your installation. Valid SVC values are between 200 and 255.

TIMER Parameter: Daemon Wake Up Frequency

Parameter	Description	Minimum	Maximum	Syntax
TIMER	The frequency in seconds the daemon wakes up and looks for work.	1	none	TIMER={10|*nnn*}

The maximum number of seconds (more precisely, units of 1.048576 seconds) that defines how often the daemon wakes up and looks for work.

Note: The daemon wakes up automatically whenever it receives a user request or an operator command. Only change the default when advised to do so by our support.

4 Operator Commands

MVS operator communication with the daemon is achieved using the z/OS `Modify (F)` command. All operator commands for the SAF Security Kernel are prefixed with SAF. For example:

```
F jobname,SAF SSTAT
```

The available operator commands are:

Command	Description
`SHELP`	Display all possible SAF Kernel operator commands.
`SLOGOFF userid`	Log an individual SAF User ID off from the security system. Any cached security checks for this user are discarded.
`SNEWCOPY`	Restart the SAF Kernel, ensuring that all data held in its cache is flushed (the same as `SREST`). Additionally, reload SAFKRN and its dynamically loaded modules.
`SREST`	Restart the SAF Kernel, ensuring that all data held in its cache is flushed. Any data held by the security system itself in the SAF Kernel address space is also flushed. The operation is transparent to all online and batch users.
`SSHUT`	Perform an orderly shutdown of the SAF Kernel started task. This command should always be used to request an orderly termination. You may also use ADAEND, for example: `F jobname,ADAEND`
`SSNAP hhhhhhhh`	Display a selected portion of the SAF Kernel's memory. Operation is not terminated.
`SSTAT`	Display general statistics on the operator console for the SAF Kernel.
`STRACE {0|1|2|3}`	Dynamically activate or deactivate SAF Kernel diagnostic tracing: ■ 0 – tracing is suppressed ■ 1 – only security violations are traced ■ 2 – only successful checks are traced ■ 3 – all checks are traced

Command	Description
SUSERS	Display a list of active users.
SUSTAT userid	Display statistics for a specified user.
SSIGTERM	All active ENF Signal listeners are terminated. The use of SSIGTERM requires Adabas Limited Library (WAL) version 8.5 SP4 patch level 1 and above.

5 SAF* - SAF Daemon Messages

The following messages may be issued by the SAF daemon.

SAF001I **Unable to load required module: {xxxxxx}**

Explanation The SAF kernel or one of the modules it needs could not be loaded. Ensure that all SAF modules (including those created by installation assembly jobs – SAFCFG, SAFPSEC and SAFPMAC) are available.

SAF002I **Unable to obtain anchor storage**

Explanation A memory allocation failed during initialization. Increase the region size.

SAFD01S **SAF CANNOT INITIALISE, GETMAIN ERROR**

Explanation There is insufficient memory for the SAF daemon to initialize. Increase the region size.

SAFD02S **SAF CANNOT INITIALISE, KERNEL LOAD ERROR**

Explanation Installation error (SAFCKN load module not available). Ensure that all required load modules are available.

SAFD03E **DDCARD open error: ## - terminating**

Explanation The SAF daemon was enable to open DDCARD. Ensure that the DDCARD DD statement is specified correctly.

SAFD04I **Input parameter: {xxx}**

Explanation The daemon echoes the values of the supplied DDCARD parameters.

SAFD05E **Invalid parameter: {##########}**

Explanation DDCARD contained an invalid parameter. The SAF daemon terminates. Correct the parameter in error.

SAFD06E **Product parameter not specified**

Explanation DDCARD did not contain `PRODUCT=SAF`. Ensure that `PRODUCT=SAF` and `DEFAULT=SAF` are both specified.

SAFD07E **MPM failure - function: ## error ##**

Explanation The SAF daemon received an error from ADAMPM. This message will usually be preceded by an explanatory message. If in doubt, contact your technical support representative for assistance.

SAFD08E **IOR failure - function: ## error ##**

Explanation An error occurred during an ADAIOR service call. Contact your technical support representative for assistance.

SAFD09I **Shutdown requested**

Explanation The SAF daemon has been instructed to shut down.

SAFD10E **Getmain for command queue failed**

Explanation The SAF daemon failed to allocate its command queue. Increase the region size.

SAFD11I **SAF Kernel is active on node {nnnnn sss }CIB={aaaaaaaa}**

Explanation The daemon is now active and ready to receive security requests; nnnnn is the node ID, sss is the SVC number, and aaaaaaaa is the address of the daemon's main storage area.

SAFD12I **Oper type in: SAF {xxxxx}**

Explanation Message 12I is issued before processing of an operator command.

SAFD13E **Invalid operator command**

Explanation An invalid operator command was entered.

SAFD14I **Target {nnnnn} termination in progress**

Explanation Message 14I is issued during daemon termination (nnnnn is the daemon's node ID).

SAFD15I **Target {nnnnn} ended normally**

Explanation Message 15I is issued during daemon termination (nnnnn is the daemon's node ID).

SAFD22E **Load for module: {########} failed**

Explanation The indicated module could not be loaded. Ensure that it is available.

SAFD25E **{###} is an invalid product name**

Explanation An invalid `PRODUCT= parameter` was specified in DDCARD.

SAFD26E **Proxy module SAFPXY was not found, product SAF will not be called**

Explanation Ensure that SAFPXY and all other required load modules are available.

SAFD30E **Getmain for product parm block failed**

Explanation A memory allocation failed during initialization. Increase the region size.

SAFD31E **Cab allocation error in module syscoru**

Explanation A memory allocation failed during initialization. Increase the region size.

SAFD34E **UAB allocation error in module syscoru**

Explanation A memory allocation failed during initialization. Increase the region size.

SAFD40S **Abend {code} Psw {pppppppp pppppppp}**

Explanation Message 40S is issued during abnormal termination. It shows the abend code, Program Status Word, module that abended and register contents.

In the event of an abend, please ensure you collect the messages, the dump and any trace messages or snaps that have been generated.

SAFD42S **Module {module} entry {entry-point} offset {offset}**

Explanation Message 42S is issued during abnormal termination. It shows the abend code, Program Status Word, module that abended and register contents.

In the event of an abend, please ensure you collect the message, the dump and any trace messages or snaps that have been generated.

SAFD43S **Regs 00-03 {register contents}**

Regs 04-07 {register contents}

Regs 08-11 {register contents}

Regs 12-15 {register contents}

Explanation Message 43S is issued during abnormal termination. It shows the abend code, Program Status Word, module that abended and register contents.

In the event of an abend, please ensure you collect the message, the dump and any trace messages or snaps that have been generated.

6 SEFM* - ADASAF SAF Interface and SAF Security Kernel Messages

ADASAF displays an eight-byte code containing various return codes from SAF. This information is shown in a number of messages denoted *ssssssss*.

The ADASAF return code "ssssssss" contains the following structure:

Position	Information Content
Byte: 1	SAF return code
Byte: 2	Function code. ADASAF internal function codes (hex) include: ■ 04 - Authorize Adabas access ■ 44 or 6C - Authenticate user
Byte: 3	Return code from security system, for example RACF
Byte: 4	Reason code from security system, for example RACF
Bytes: 5 - 8	SAF reason code

Refer to the IBM manual External Security Interface (RACROUTE) Macro Reference manual for z/OS for a thorough explanation of all possible return/reason codes. CA-Top Secret and CA-ACF2 can provide different return code values in some circumstances.

BLS0334 **SYMBOL 'NETSAF' CANNOT BE FOUND. LOADING ABORTED**

Explanation This message should be ignored.

SEFM001 **{ssssssss} : {user} : {resource}**

Explanation The security system determined that the user identified in the message (*user*) does not have authorization for the resource listed in the message (*resource*). System return and reason codes are given in the hexadecimal string *ssssssss*. This message is displayed when access has been denied to a particular resource.

SEFM002 ***{XX} to request FF : {user} : {resource}**

Explanation An unexpected response code (*XX*) was received from the SAF Security Kernel for the user identified in the message (*user*) when requesting function *FF* to be performed on the resource specified in the message (*resource*).

SEFM004 ***Natural programs not extracted**

Explanation The SAF Security Kernel was not able to extract a list of protected program objects from the security system on behalf of Natural users.

Action Obtain a trace of SAF call RACROUTE EXTRACT from the security system and contact your technical support representative. ACF2 and Top Secret users should ensure that the protected programs have been extracted from the security system and supplied

to the SAF Security Kernel via the SEFEXT DD statement in the daemon started task JCL.

SEFM006 ***ADARSP {XX}({xx}) to request FF : {user}**

Explanation The SAF Security Kernel returned the Adabas response code (*XX*) and subcode (*xx*) shown in the message to request *FF* for the user shown in the message (*user*).

Action Ensure that the SAF Kernel started task is active. Check its output for error messages. Take the necessary remedial action indicated by the Adabas response code.

SEFM008 ***SAF Gateway (V{v.r}) started**

*** SAF Security Kernel (V{x.x.x} - BUILD {xxxx}) started**

Explanation Entire Net-Work SAF Security Interface (ADASAF) startup completed or the SAF Security Kernel initialized successfully.

Action No action is required for this informational message.

SEFM009 **Module {module-name} not loaded**

Explanation Entire Net-Work SAF Security Interface could not load the module listed in the message (*module-name*).

Action Ensure that the module is in the STEPLIB and that the region size is sufficient.

SEFM013 ***Less {memory | storage} acquired than specified**

Explanation The SAF Security Kernel or the Entire Net-Work SAF Security Interface (ADASAF) were not able to allocate all the memory or storage required to satisfy the buffer size specified in its parameters. Operation continues.

Action Ensure that the region size is sufficient and the parameters are appropriate.

SEFM014 ***No {memory | storage}could be acquired**

Explanation Entire Net-Work SAF Security Kernel or the SAF Security Interface (ADASAF) could obtain no storage or memory at system startup. Operation has terminated.

Operation has terminated.

Action Ensure that the region size is sufficient and system parameters are appropriate.

SEFM015	***Logic error - {XXXX} for request FF : {user}**
Explanation	The SAF Security Kernel suffered an internal error. A general restart is performed and the operation continues.
Action	Keep all information written to DDPRINT and contact your technical support representative.

SEFM016	***SAF logoff failed {ssssssss} ACEE AAAA : {user}**
Explanation	The SAF Security Kernel was unable to logoff *user* from the security system. The SAF error code is *ssssssss*.
Action	Contact your technical support representative.

SEFM017	***Insufficient space to initialize - make Natural buffer {XX}**
Explanation	The Natural SAF interface requires a larger value to be specidied for NSFSIZE.
Action	Increase the Natural NSFSIZE parameter.

SEFM020	***GETMAIN failed / IDSIZE error**
Explanation	The Natural SAF interface could not acquire storage from the designated IDMSBUF.
Action	Increase Natural region and/or thread size.

SEFM021	***Illegal storage use / relocation problem**
Explanation	Internal problem in Natural SAF storage use.
Action	Contact your technical support representative.

SEFM025	***Natural IDMSBUF parameter is not defined**
Explanation	The Natural NSFSIZE parameter has not been specified.
Action	Ensure NSFSIZE is set correctly in the Natural parameters.

SEFM026	***Natural protected programs not extracted code: {XX}**
Explanation	The list of protected programs could not be returned from the SAF Security Kernel to Natural.
Action	Ensure the same copy of the configuration module SAFCFG is used by all system components. Check that the GWSTYP parameter defined in SAFI010 and STY parameter in SAFI020 are both correctly set for the installed security system and that all installation requirements have been met.

SEFM028 ***System files not found in environment table**

Explanation The current Natural system files were not matched in the table defining all possible system file sets.

Action Ensure that the environment definitions in Natural Security are correct.

SEFM029 ***Error in communications layer - check installation procedure**

Explanation Possible reasons for error: Adabas link module installed into this component is not reentrant.

SEFM030 ***SQL table / VIEview could not be identified for file ({XX},{YY})**

Explanation Interface could not identify table name for DBID/FNR of an SQL request.

Action Ensure interface is correctly installed, then contact your technical support representative.

SEFM031 ***DBID / FNR identified with SQL request not recognized {XXXX}**

Explanation Interface component could not determine the DBID/FNR associated with this SQL request.

Action Contact your technical support representative.

SEFM041 ***Interface installed for Net-work**

Explanation The interface is installed for operation with Entire Net-Work.

Action No action is required for this informational message.

SEFM049 ***User type T not permitted by installed options**

Explanation The SAF Kernel will not permit user type *T* to operate using the currently installed options.

SEFM050 ***Error writing SMF record : {XX}**

Explanation The stated error occurred when an SMF record was being written.

SEFM051 ***SAFPRINT dataset not defined, DDPRINT will be used**

Explanation `SAFPRINT=Y` is set in SAFCFG, but no SAFPRINT dataset is defined.

SEFM060 *** RACLIST REQUESTED FOR CLASS cccccccc**

Explanation The specified security class cccccccc has been configured to use FASTAUTH in SAFCFG. This message indicates the start of the RACLISTing process for this class.

Action No action is required for this informational message.

SEFM061 *** RACLIST SUCCESSFUL FOR CLASS cccccccc**

Explanation This message indicates RACLISTing has been successful for class cccccccc.

Action No action is required for this informational message.

SEFM062 *** RACLIST FAILED FOR CLASS cccccccc ERROR ssssssss**

Explanation This message indicates RACLISTing has failed for class cccccccc. The FASTAUTH option for this class cannot be honoured, authorization checks will be performed using RACROUTE REQUEST=AUTH (the default).

Action Refer to *Adabas SAF Security* > *Adabas SAF Security Messages and Codes* > *SAF Return Codes* for information on how to interpret the error code ssssssss.

SEFM070 *** LISTENER ACTIVE FOR SIGNAL ENF-{xx}**

Explanation This message indicates the signal listener has been successfully activated and is listening for ENF signal type *xx*.

Action No action is required for this informational message.

SEFM071 *** UNABLE TO ACTIVATE LISTENER FOR SIGNAL ENF-{xx} ERROR {eeeeeeee}**

Explanation This message indicates the signal listener could not be activated for ENF signal type *xx*. The error code is *eeeeeeee*.

Action Contact your technical support representative.

SEFM072 *** LISTENER TERMINATED FOR SIGNAL ENF-{xx}**

Explanation This message indicates the signal listener has been successfully de-activated and is no longer listening for ENF signal type *xx*.

Action No action is required for this informational message.

SEFM073 *** SIGNAL LISTENERS REQUESTED BUT INCOMPATIBLE RUNTIME**

Explanation This message indicates that signal listeners have been requested by configuration but there is an incompatibility between the Adabas SAF Security and the Adabas Limited Library (WAL) runtimes. The job continues to run, but without any active ENF Signal listeners.

Action Ensure Adabas SAF Security is at least version 8.4 SP1, and the Adabas Limited Library (WAL) is at least version 8.5 SP4 Patch level 1.

SEFM074	*** UNABLE TO ALLOCATE REQUIRED MEMORY FOR SIGNAL LISTENERS**
Explanation	This message indicates a memory shortage when attempting to establish the signal listeners. The job continues to run, but without any active ENF Signal listeners.
Action	Review the `REGION` size for the job. Review the `SIGNQSZ` parameter.

SEFM075	*** UNABLE TO TERMINATE SIGNAL LISTENERS. PLEASE TRY LATER**
Explanation	SAF daemon only. This message indicates that Adabas SAF Security in the daemon has been unable to cleanly terminate the ENF Signal listeners in response to a `SSIGTERM` command. The ENF Signal listeners are asynchronous processes that were likely busy at the time of termination.
Action	Retry the `SSIGTERM` command at a later time

SEFM205	***CPU identity : {cpuid}**
Explanation	The interface component linked to Entire Net-Work displays the CPU ID of the host machine.
Action	No action is required for this informational message.

SEFM210	***SAF Gateway is active for Entire Net-Work**
Explanation	The Entire Net-Work SAF Security Interface is active.
Action	No action is required for this informational message.

SEFM255	***Unauthorized use of request**
Explanation	Attempted illegal use of security request.
Action	Contact your technical support representative.

Operator Command Messages (SEFM900+ Series) Adabas SAF Security operator command messages SAF Security Kernel operator command messages

The following messages are displayed in response to operator commands:

SEFM900	*** Operator issued command: {command}**
Explanation	Entire Net-Work SAF Security Interface (ADASAF) or the SAF Security Kernel received the operator command identified in the message.
Action	No action is required for this informational message.

SEFM901 *** SAF server - General statistics (at {hhhhhhhh})**

*** SAF Security Kernel - General statistics (at {hhhhhhhh})**

Explanation The operator command for general statistics was issued. Here is an example of the statistics messages produced for the SAF server:

```
SEFM901 * SAF SERVER - GENERAL STATISTICS (AT hhhhhhhh)
SEFM902 * RESOURCE   CHECK(+VE) CHECH(-VE) CHECK SAVED OVERWRITES ↩
LEN
SEFM903 * APPLICATION         1          0          0           ↩
0    8
SEFM903 * ADABAS              0          0          0           ↩
0   32
SEFM903 * SYSMAIN             0          0          0           ↩
0   13
SEFM903 * SYSTEM FILE         2          0          0           ↩
0   24
SEFM903 * PROGRAM             0          0          0           ↩
0   17
SEFM903 * BROKER              0          0          0           ↩
0   32
SEFM903 * NET-WORK            0          0          0           ↩
0    0
SEFM903 * SQL SERVER          0          0          0           ↩
0    0
SEFM904 * USERS - ACTIVE: 1 FREE: 55 OVEWRITES: 0
```

Here is an example of the statistics messages produced for the SAF Security Kernel:. The address in the first line is the address of the SAF Kernel's storage cache.

```
SEFM901 * SAF SECURITY KERNEL - SERVER STATISTICS (AT 12C47000)
SEFM902 * RESOURCE   CHECK(+VE) CHECH(-VE) CHECK SAVED OVERWRITES ↩
LEN
SEFM903 * APPLICATION        10          0          0           ↩
0    8
SEFM903 * DBMS CHECK          0          0          0           ↩
0   17
SEFM903 * SYSMAIN             0          0          0           ↩
0   21
SEFM903 * SYSTEM FILE         2          0          0           ↩
0   40
SEFM903 * PROGRAM             0          0          0           ↩
0   17
SEFM903 * BROKER              0          0          0           ↩
0   68
SEFM903 * NET-WORK            0          0          0           ↩
0   17
SEFM903 * SQL SERVER          0          0          0           ↩
0   32
SEFM904 * CACHED USERS:       1 HIGH WATERMARK:     1 MAX USERS: ↩
```

```
 5545
SEFM905 * OVERWRITES:         0 AUTHENTICATED:      0 DENIED:    ↩
    0
```

Action No action is required for this informational message.

SEFM902 - 905 **{statistics}**

Explanation Various statistics for the SAF server and the SAF Security Kernel are displayed. See message **SEFM901**.

Action No action is required for this informational message.

SEFM909 *** {SAF Gateway | SAF Security Kernel} - shutdown initiated**

Explanation The operator issued a command to shut down Entire Net-Work SAF Security Interface or the daemon started task (SAF Security Kernel). This message is also issued when a secure Adabas nucleus, Net-Work node or Adabas SQL server terminates.

Action No action is required for this informational message.

SEFM910 ***{SAF Server | SAF Security Kernel} - list all active users**

Explanation The operator issued a command to display a list of currently active users.

The following is a sample of the output produced for the SAF server:

```
SEFM910 * SAF SERVER - LIST ALL ACTIVE USERS
SEFM911 * USERID     CHECK(+VE) CHECH(-VE) CHECK SAVED OVERWRITES ↩
BUFF
SEFM912 * K11079              3          0          0           ↩
0   0
```

The following is a sample of the output produced for the SAF Security Kernel:

```
SEFM910 * SAF GATEWAY - LIST ALL ACTIVE USERS
SEFM911 * USERID CHECK(+VE) CHECH(-VE) CHECK SAVED OVERWRITES BUFF
SEFM912 * K11079         3          0          0           0   0
```

Action No action is required for this informational message.

SEFM911 ***{userid} . . .**

Explanation The operator issued a command to display statistics specific to a currently active user.

The following is a sample of the output produced for the SAF server:

```
SEFM911 * NXB        CHECK(+VE) CHECH(-VE) CHECK SAVED OVERWRITES ↩
 BUFF
SEFM912 * APPLICATION         1          0          0           ↩
0    0
SEFM912 * DBMS CHECK          0          0          0           ↩
0    0
SEFM912 * SYSMAIN             0          0          0           ↩
0    0
SEFM912 * SYSTEM FILE         2          0          0           ↩
0    0
SEFM912 * PROGRAM             0          0          0           ↩
0    0
SEFM912 * BROKER              0          0          0           ↩
0    0
SEFM912 * NET-WORK            0          0          0           ↩
0    0
SEFM912 * SQL SERVER          0          0          0           ↩
0    0
```

The following is a sample of the output produced for the SAF Security Kernel:

```
SEFM911 * SJU        CHECK(+VE) CHECH(-VE) CHECK SAVED OVERWRITES ↩
 BUFF
SEFM912 * APPLICATION        10          0           0          ↩
 0   10
SEFM912 * DBMS CHECK          0          0           0          ↩
 0    0
SEFM912 * SYSMAIN             0          0           0          ↩
 0    0
SEFM912 * SYSTEM FILE         2          0           0          ↩
 0    2
SEFM912 * PROGRAM             0          0           0          ↩
 0    0
SEFM912 * BROKER              0          0           0          ↩
 0    0
SEFM912 * NET-WORK            0          0           0          ↩
 0    0
SEFM912 * SQL SERVER          0          0           0          ↩
 0    0
```

Action No action is required for this informational message.

SEFM913 *** No active users found in SAF {Server | Gateway | Security Kernel}**

Explanation No active users were found in Entire Net-Work SAF Security Interface (ADASAF) or in the SAF Security Kernel.

Action No action is required for this informational message.

SEFM914 *** Requested user {userid} not found in SAF {Server | Gateway | Security Kernel}**

Explanation The requested user was not found in the Entire Net-Work SAF Security Interface (ADASAF) or in the SAF Security Kernel.

Action No action is required for this informational message.

SEFM915 **SEFM915 * SAF Security Kernel - snap of server memory**

Explanation This message is issued in response to an `SSNAP` operator command and is followed by a sequence of SEFM916 messages.

Action No action is required for this informational message.

SEFM916 *** {hhhhhhhh hhhhhhhh hhhhhhh hhhhhhhh hhhhhhhh.x..X.Y}/**

Explanation This message contains the results of an `SSNAP` command. Each SSNAP snaps up to 256 bytes and shows the address, the hexadecimal storage contents, and the interpretation.

Action No action is required for this informational message.

SEFM918 *** Supplied address is outside of legal range**

Explanation An attempt was made to snap storage outside the bounds of the SAF Kernel's cache.

SEFM919 ***Operator command did not contain required arguments**

Explanation A required parameter was omitted from an operator command. For example, `SUSTAT` with no userid specified.

Action Correct the operator command and try again.

SEFM920 **{command1}, {command2}, {command3}, and so on**

Explanation This message is issued in response to an `SHELP` operator command and lists the available SAF Kernel operator commands.

Action No action is required. This message is informational.

SEFM921	*** Memory allocation failure - users cannot be logged off**
Explanation	The SAF Kernel was unable to obtain temporary storage (approximately 16Kb) to log users off in response to an SREST, SNEWCOPY or SLOGOFF operator command.
Action	Increase the region size.

SEFM922	*** User {userid} logged off**
Explanation	This message is issued in response to an SLOGOFF operator command. The indicated user has been logged off from the security system.
Action	No action is required for this informational message.

SEFM923	*** User {userid} not logged off - user not found**
Explanation	This message is issued in response to an SLOGOFF operator command. The requested user could not be found in the SAF Kernel's cache.
Action	Verify the correct user ID was specified.

SEFM924	*** User {userid} not logged off - return code {ZZ}**
Explanation	This message is issued in response to an SLOGOFF operator command. An internal error (indicated by ZZ) occurred while attempting to log the requested user off.
Action	Evaluate the return code to determine the cause of the error.

SEFM928	*** Invalid trace setting - must be 0, 1, 2 or 3**
Explanation	The STRACE operator command was issued with an invalid trace setting.
Action	Correct the trace setting and try again.

SEFM929	*** Invalid SAF Security Kernel operator command**
Explanation	An invalid SAF Security Kernel operator command was entered.
Action	Specify a valid SAF Security Kernel operator command.

SEFM930	*** SIGNAL SREST INTERVAL COUNTS:**
Explanation	When the ENF Signal Listener is active, this message is appended to the general statistics displayed by message SEFM901 as a result of an SSTAT or SREST command. Together with SEFM931, counts are displayed relating to the number of ENF Signal conditions which have resulted in an SREST operation being requested, since the time of the last completed SREST operation.
Action	No action is required for this informational message.

SEFM931 *** QUEUE FULL: nnnnnnnn ENF-62: nnnnnnnn**

*** ENF-71: nnnnnnnn ENF-79: nnnnnnnn**

Explanation When the ENF Signal Listener is active, this message is appended to the general statistics displayed by message SEFM901 as a result of an SSTAT or SREST command. Counts are displayed relating to the number of ENF Signal conditions which have resulted in an SREST operation being requested, since the time of the last completed SREST operation.

QUEUE FULL

The number of times a free entry in the ENF Signal Listener Queue could not be found.

ENF-*nn*

The number of times the ENF Signal Listener was unable to identify the signal type *nn* as belonging to a specific user.

Action No action is required for this informational message.

7 SAF Return Codes

ADASAF and the SAF Security Kernel display an eight-byte code containing various return and reason codes from SAF. This information is shown in a number of messages denoted "SSSSSSSS".

The SAF and ADASAF return codes contains the following structure:

Position Within Message Code	Information Content
Byte: 1	SAF return code (R15 after RACROUTE)
Byte: 2	Function code (see section *SAF Internal Function Codes*)
Byte: 3	RACROUTE return code
Byte: 4	RACROUTE reason code
Byte: 5-8	Internal reason code

The SAF trace messages written to DDPRINT, when GWMSGL is not 0, include the first four bytes of the following information, printed as eight hexadecimal digits. The ADASAF trace messages include the first four bytes of the following information, also printed as eight hexadecimal digits:

Position Within Trace Message	Information Content
Digits 1 and 2	SAF return code (R15 after RACROUTE)
Digits 3 and 4	Function code (see section *SAF Internal Function Codes*)
Digits 5 and 6	RACROUTE return code
Digits 7 and 8	RACROUTE reason code

Refer to the *IBM Security Server RACROUTE Macro Reference* manual for a thorough explanation of all possible return/reason codes. CA-Top Secret and CA-ACF2 can provide different return code values in some circumstances.

8 SAF Internal Function Codes

SAF Security Kernel and ADASAF internal function codes include:

Function Code (Hex)	Description
00	Authorize Natural Library
04	Authorize Adabas access
08	Authorize SYSMAIN function
0C	Authorize Natural system files
10	Authorize Natural program execution
14	Authorize Broker service
18	Authorize Entire Net-Work access (Net-Work SAF Security) or Adabas cross-level access (Adabas SAF Security) or RPC execution (Natural SAF Security).
1C	Authorize SQL Server access
44 or 6C	Authenticate user

9 Interpreting Trace Messages

The SAF Kernel may optionally write trace messages to DDPRINT (or SAFPRINT). These trace messages have the following format:

```
Time   Jobname  Result     Return Code Type SAF Userid Level Resource Name
13:19:19 DAEFCODE SEF DENIED    08040800 RQ 02 :USERA   : (02) CMD00153.FIL00005
```

Field	Explanation
Time	Time the security check was made.
Jobname	Job that requested the security check. For Adabas and Net-Work SAF Security this is the job that issued the Adabas call being checked.
Result	SEF DENIED: the security system rejected the access attempt. SEF PERMITTED: the security system allowed the access.
Return Code	The return code consists of 4 hexadecimal bytes which contain the following information. The numbers in brackets refer to the values in the example trace message above. ■ Byte 1 (08) - R15 after RACROUTE ■ Byte 2 (04) – internal function code (see table above) ■ Byte 3 (08) – RACROUTE return code ■ Byte 4 (00) – RACROUTE reason code The return code can be interpreted by checking the RACROUTE manual referred to above for the appropriate RACROUTE function (AUTH for an authorize function; VERIFY for authenticate). For a RACROUTE AUTH, R15 of 8 with return code 8 and reason code 0 means the user is not authorized to use the requested resource. This is a normal security violation. For PERMITTED security checks, the return code contains 00000000 or 00000001. 00000001 indicates that the security check was satisfied from the SAF Kernel's cache (that is, the same user had previously requested the same resource access and the SAF Kernel had cached the security system's successful response).

Field	Explanation
Type	The internal SAF Kernel request type. This may be: ■ 01 – authorize Natural library ■ 02 – authorize Adabas access ■ 03 – authorize SYSMAIN function ■ 04 – authorize Natural system files ■ 05 – authorize Natural program execution ■ 06 – authorize Broker service ■ 07- authorize Net-Work (or Adabas cross-level) access ■ 08 – authorize SQL server access ■ 13 – authenticate user ■ 23 – authorize Natural RPC execution
SAF Userid	The SAF User ID for which access was requested.
Level	The access level requested: ■ 02 – read ■ 04 – update ■ 08 – control ■ 80 – alter
Resource Name	The name of the resource for which access was requested. For successful user authentications, resource name contains: ■ XXNEWU – user successfully authenticated or ■ XX - user already logged on

In the example trace message shown above: at 13:19:19, SAF user USERA in job DAEFCODE attempted to read Adabas file 5 in database 153 but did not have the necessary security access.

10 Security Definitions

SAF Security is implemented by defining resource classes and profiles and permitting users the necessary access to those profiles. Specific requirements for class and profile definitions and access levels are described in the individual product documentation.

This section describes in general how to define resources to RACF, CA-Top Secret and CA-ACF2.

Defining Resources to RACF

This section describes how the resources are defined to RACF. For exact details of the procedures to be followed for the installed RACF version, consult the relevant IBM manuals.

Overview of tasks

- Add classes to Class Descriptor Table
- Update z/OS Router Table
- Activate new classes
- Assign user ID for the SAF Security Started Task
- Permit user access to resource profiles

To add classes to Class Descriptor Table

1. Add the resource classes to the RACF Class descriptor table. Refer to the *IBM SPL RACF* manual. For an example, see IBM SYS1.SAMPLIB, member RACINSTL.
2. For flexibility, allocate maximum length for the classes (80).
3. Define the classes to enable discrete and generic profile use.
4. Check further attributes controlling the level of RACF messages generated when performing RACROUTE calls, as well as the required level of SMF recording. Sample definitions are provided in source members SAFRCLSN and SAFRCLSX.

To update the z/OS Router Table

- Update the z/OS router table as described in the *IBM SPL RACF* manual. For an example, see the IBM SYS1.SAMPLIB, member RACINSTL, section RFTABLE.

To activate new classes

- Activate new resource classes with SETROPTS (see *IBM RACF Command Language Reference* manual). For an example, activate class NBKSAG:

```
SETROPTS CLASSACT(NBKSAG)
SETROPTS GENCMD(NBKSAG)
SETROPTS GENERIC(NBKSAG)
```

To assign user ID for the SAF Security Started Task

- The SAF Security Kernel runs either in its own Started Task or in an Adabas or Entire Net-Work started task. Assign a user ID to these jobs with the relevant RACF authorizations, including the ability to perform `RACROUTE, TYPE=EXTRACT, TYPE=AUTH` and `TYPE=VERIFY` calls on profiles belonging to the defined classes.

To permit user access to resource profiles

- After adding profiles to protect the different resources, permit users the required level of access, using the relevant RACF Commands. The following example adds resource profile ETB.POLICY.QUOTE1 and grants READ access to user USER2 and CONTROL access to USER3. USER2 represents a client and requires READ access to execute while USER3 represents a server component that needs CONTROL access to register:

```
RDEFINE NBKSAG ETB.POLICY.QUOTE1 UACC(NONE)
PERMIT ETB.POLICY.QUOTE1 CLASS(NBKSAG) ACCESS(READ) ID(USER2)
PERMIT ETB.POLICY.QUOTE1 CLASS(NBKSAG) ACCESS(CONTROL) ID(USER3)
```

Defining Resources to CA-TOP SECRET

This section describes how the resources are defined to TOP SECRET. For exact details of the procedures to be followed for the installed version of TOP SECRET, consult the relevant CA-TOP SECRET manual.

Overview of tasks

- Add CA-TOP SECRET Facility
- Assign user ID for the SAF Security Started Task
- Add procedure name for the Started Task
- Add resource type to Resource Definition Table
- Assign ownership of resources
- Permit defined resources to Users

To add CA-TOP SECRET facility

- CA-TOP SECRET enables a set of authorization checks to be made against a certain facility. For example, this can be used to secure the development environment SAGDEV separately from the production environment SAGPROD. Alternatively, a default facility of batch can be used.

  When adding additional facilities, use the following attributes:

```
AUTHINIT,MULTIUSER,NONPWR,PGM=ADA,NOABEND
```

To assign a user ID for the SAF Security Started Task

- Add one user ID for each instance of the SAF Security Started Task.

  If required, different facilities can be assigned to development and production tasks.

  The designated facility is assigned to the Started Task user ID:

```
TSS CRE(user-id) DEPT(dept) MASTFAC(fac)
```

To add a procedure name for the SAF Security Started Task

- The procedure name under which the SAF Security Started Task executes must be defined to CA-Top Secret. Different procedure names are suggested when securing different environments separately with the use of non default CA-Top Secret facilities:

```
TSS ADD(STC) PROC(proc) USER(user-id)
```

To add resource types to Resource Definition Table

- Add the resource types to the CA-TOP SECRET Resource Definition Table (RDT). Below is an example for resource type NBKSAG. Refer to the CA-TOP SECRET Reference Guide for a detailed explanation of the following commands and arguments:

```
TSS ADD(RDT) RESCLASS(NBKSAG)
RESCODE(HEXCODE)
ATTR(LONG)
ACLST(NONE,READ,CONTROL)
DEFACC(NONE)
```

To assign ownership of resources

- Assign ownership to a particular resource as shown in the following example. This must be done before permitting access to defined resource profiles:

```
TSS ADD(user1) NBKSAG(ETB.POLICY.QUOTE1)
```

  This makes user user1 the owner of the Broker service etb.policy.quote1.

To permit defined resource to users

- Permit access to a resource profile as in the following example. In the example, user user2 is permitted READ access to the Broker service etb.policy.quote1. This enables the user to execute as a client and issue requests to this Broker service:

```
TSS PER(user2) NBKSAG(ETB.POLICY.QUOTE1) FAC(fac) ACCESS(READ)
```

Defining Resources to ACF2

This section describes the definition of resources to ACF2 versions 5 and 6. For details of the procedures required for the current software version, please consult the relevant ACF2 manual.

Note: ACF2 provides insufficient return codes to determine whether a resource profile does not exist or simply the user does not have access to it. Therefore, if access is denied by ACF2, the SAF Security Kernel will always report "Access denied resource not allowed" in the error message.

Consequently the SAF Security configuration options such as BKUNI=Y to allow access to undefined resources are not applicable where ACF2 is used.

To define resources to ACF2 version 5

1 The SAF Security Kernel executes as a normal started task in z/OS. Define the user ID of the server task to ACF2 with the following attributes:

```
MUSASS,NON-CNCL,STC
```

To avoid the NON-CNCL attribute, APAR TW95626 must be applied.

2 Activate the SAF Interface using the command:

```
GSO OPTS -  SAF
```

3 Switch off all SAF checks by inserting the SAFSAVE record as follows:

```
SAFSAVE CLASSES(-) CNTLPTS(-) SUBSYS(-)
```

4 Switch on the SAF security checks for the SAF Security Kernel by inserting the SAFPROT record as follows:

```
CLASSES(-) CNTLPTS(-) SUBSYS(ADARUN)
```

5 For the general resource class name used by SAF Security product options, define a 3-character ACF2 resource type code by inserting a SAFMAPS record as follows:

```
SAFMAPS MAPS(NBK/NBKSAG)
```

6 Define the required resource profiles to ACF2 using the new type code.

The following example shows the addition of a Broker service etb.policy.quote1, allowing READ access for USER2:

```
$KEY(ETB.POLICY.QUOTE1) TYPE(NBK) UID(user2) ALLOW SERVICE(READ)
```

To define resources to ACF2 version 6 and above

1 The SAF Security Kernel executes as a normal started task in z/OS. Define the user ID of the server task to ACF2 with the following attributes:

```
MUSASS,STC
```

ACF2 version 6.1 and 6.2 no longer require TW95626,as these versions are more SAF-compliant.

2 Insert SAFDEF records as follows:

```
SAFDEF.EXS1
FUNCRET(4) FUNCRSN(0) ID(ENTIREX) MODE(GLOBAL)
RACROUTE(REQUEST=VERIFY SUBSYS=ADARUN REQSTOR=-)
RETCODE(4)
```

```
SAFDEF.EXS2
FUNCRET(4) FUNCRSN(0) ID(ENTIREX) MODE(GLOBAL)
RACROUTE(REQUEST=AUTH SUBSYS=ADARUN REQSTOR=-)
RETCODE(4)
```

```
SAFDEF.EXS3
FUNCRET(4) FUNCRSN(0) ID(ENTIREX) MODE(GLOBAL)
RACROUTE(REQUEST=EXTRACT SUBSYS=ADARUN REQSTOR=-)
RETCODE(4)
```

3 For the general resource class names used by SAF Security product options, define a 3-character ACF2 resource type code by inserting a CLASMAP record as follows:

```
CLASMAP
ENTITYLN(0) MUSID() RESOURCE(NBKSAG) RSRCTYPE(NBK)
```

4 Define the required security profiles to ACF2 using the new type code. The following example shows the addition of a Broker service etb.policy.quote1, allowing READ access only for user ID user2:

```
$KEY(ETB) TYPE(NBK)
POLICY.QUOTE1 UID(user2) SERVICE(READ)   ALLOW
POLICY.QUOTE1 UID(-)                     PREVENT
```

Index