Adabas

SAF Security Kernel

Version 7.4.4

September 2009

This document applies to Adabas Version 7.4.4 and to all subsequent releases.

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

Table of Contents

1 SAF Security Kernel

This document describes the SAF Security Kernel and its associated Daemon. 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).

2 Introduction

The System Authorization Facility (SAF) is used by OS/390 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 Software AG 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

This chapter covers the following topics:

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.

3 Installation

This section describes how to install the SAF Security Kernel.

This chapter covers the following topics:

Prerequisites

The following are prerequisites:

- OS/390 or z/OS
- SAF-compliant security system

Preparing for Installation

Before installing the SAF Security Kernel, review all possible configuration options for the kernel itself and for the product(s) it will secure.

If the kernel will execute as a daemon, in its own address space, allocate a unique node number to it.

Authorization

The kernel load library and any other step libraries in the kernel's loading environment must be APF authorized.

Modes of Operation

The kernel may be embedded with a product (that is, it may run in the same address space). This is the case for Adabas and Entire Net-Work. To implement this mode of operation, you simply need to add the kernel load library (and any load libraries used as the target of installation assembly and link jobs) to the step library concatenation, ensuring that they are APF authorized.

For products other than Adabas and Entire Net-Work, the kernel operates under a daemon, in its own address space as a target in the Software AG network. This mode of operation is described in more detail below.

For both modes of operation, 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 RACROUTE and to issue third-party checks on behalf of all users.

Installation Datasets

The SAF Security Kernel is supplied as a component of the Adabas Limited Libraries

WALvrs.LOAD

WALvrs.LOAD is a standard load library containing modules needed to operate the SAF Security Kernel.

This library must be APF-authorized and available on the loading environment of any job that uses the SAF Security Kernel. Jobs that include the SAF Security Kernel are:

- The SAF Security Daemon, used by Natural SAF Security and EntireX Security
- Adabas nuclei protected by Adabas SAF Security
- Entire Net-Work nodes protected by Entire Net-Work SAF Security
- Adabas SQL SMARTS servers protected by Adabas SAF Security

The WALvrs.LOAD modules pertaining to SAF Security all have names beginning with SAF, with the exception of:

SVCLIST	Program to list active Adabas SVCs
SVCSAF	Adabas router security extensions used by Adabas SAF Security and EntireX Security

WALvrs.SRCE

WALvrs.SRCE is a standard source library containing Assembler macros (names beginning NA2M) and source books (SAFCFG, SAFPOS and SAFPSEC) which must be assembled as part of the SAF Security Kernel installation. There are also several example members:

SAFAEXT	CA-ACF2 extract for Natural RPC protection
SAFRCLSN	RACF class definitions for Natural SAF Security
SAFRCLSX	RACF class definitions for EntireX Security
SAFTEXT	CA-Top Secret extract for Natural RPC protection
SAFDDCAR	Daemon DDCARD input
SAFPARMS	Sample SAFCFG

WALvrs.JOBS

WALvrs.JOBS is a standard source library containing example jobs for installing the SAF Security Kernel. These examples have names beginning SAF.

Installation Procedure

This section describes how to install the SAF Security Kernel.

Step 1 Assemble the Configuration Mode

The configuration module defines the required installation options. Only general options are described here. For information about product-specific options, see the relevant product documentation. A sample job is provided in SAFI010 in the jobs library.

The resulting load module, SAFCFG, must be available to any job that includes the SAF Security Kernel and, in the case of EntireX, to the jobs being secured. You may decide to maintain different SAFCFG modules for different secured products. However, it is critical that the daemon use exactly the same configuration module as EntireX jobs secured by that daemon.

Set the following parameters to the appropriate values:

GWDBID=nnnnn	Node ID of SAF server
GWSIZE=nnnnn	Buffer size in K (approximately 512 bytes per user)
GWMSGL={0, 1 ,2,3}	Message level
GWSTYP={ 1 ,2,3}	Security repository type
SAFPRINT={ N ,Y}	Write trace messages to DDPRINT (N) or SAFPRINT (Y)

Message level indicates which diagnostic messages will be written to DDPRINT or SAFPRINT:

1 (the default)	only security violations are traced
2	only successful checks are traced
3	all checks are traced
0	tracing is suppressed

Security repository type identifies the SAF security system in use:

1 (the default)	RACF
2	CA-Top Secret
3	CA-ACF2

SAFPRINT specifies where security check trace messages should be written:

N (the default)	DDPRINT
Y	SAFPRINT

If you specify Y, but do not provide a SAFPRINT dataset, the trace messages will be written to DDPRINT. The SAFPRINT dataset must be defined in the JCL and may refer to a SYSOUT dataset or to a file defined with RECFM=F (or FB) and LRECL=121.

Step 2 Assemble the RACROUTE Macros

The SAF Security Kernel requires the same version of the RACROUTE macros as used at the customer site. Sample job SAFI020 is provided to assemble the module containing these macros.

Before running SAFI020, set the parameter STY to RACF, TSS, or ACF2 as appropriate and ensure that the REL parameter is set to the correct RACF version number. CA-Top Secret and CA-ACF2 require the equivalent RACF version number (for example 1.9, 2.1, or 2.2) and not the version of ACF2 or Top Secret itself. The resulting load module, SAFPSEC, must be available to any job that includes the SAF Security Kernel.

Step 3 Assemble the Operating System Services Module

Sample job SAFI021 is provided to assemble the operating system services module, SAFPOS. The resulting load module, SAFPMAC, must be available to any job that includes the SAF Security Kernel.

Embedded SAF Security Kernel

For those products (Adabas and Entire Net-Work) that use an embedded SAF Security Kernel, you need only add the load library containing the kernel (SAFKRN) and the three load modules created above to the step library concatenation.

Installing the SAF Security Daemon

For those products (Natural and EntireX) that need a SAF Security Kernel running in a separate, authorized address space, you must install a SAF Security Daemon.

The SAF Security Daemon runs in its own address space, using Adabas modules to establish inter-process communication. It signs on to the Adabas SVC as a target and is therefore accessible in the same way as an Adabas database. Consequently, the SAF Security Daemon (and its Kernel) can be accessed remotely, via Entire Net-Work.

Software AG recommends that you run the SAF Security Daemon as a started task, although it may be run as a batch job. The SAF Security Daemon must run APF-authorized, therefore all step libraries must be APF-authorized.

Additionally, the SAF Security Daemon must run under a userid with sufficient authority to invoke the RACROUTE AUTH, EXTRACT and VERIFY functions and to make third-party checks on behalf of other users.

Sample JCL to execute the daemon is provided in SAFI024 in the jobs library.

Daemon Configuration

The daemon is configured by parameter input. The parameters are read from the DDCARD dataset at startup. An example dataset is provided in SAFDDCAR in the source library. Following is a description of valid parameters, with default value and meaning.

Parameter	Default	Meaning
NODE	None	Identifies this SAF Security Daemon. Must be a number between 1 and 65535 and must be unique among all targets.
PRODUCT	None	Defines which products are available in this server. Specify SAF.
FORCE	NO	Defines whether or not an existing ID table entry for the same node should be overwritten. Valid values are YES and NO. Specify YES only when advised to by Software AG.
LOCAL	NO	Defines whether or not this server is to be accessible from remote users, via Entire Net-Work. Valid values are YES (the server is not accessible) and NO (the server is accessible).
NC	20	Defines the maximum number of concurrent requests that can be processed by the server. Specify a number between 1 and 32767. If a request to the server fails with response code 151, increase NC.

Parameter	Default	Meaning
NABS	16	Defines the number of 4K storage blocks to be used for transmitting information between clients and the server. Specify a number between 1 and 32767. If a request to the server fails with response code 255, increase NABS.
LU	65535	Defines the maximum total length of data for a request to the server. Do not change this parameter value unless advised to by Software AG.
TIMER	10	Defines how often the server is to wake up and look for work (note that the server wakes up anyway whenever it receives a request or operator command). Specify a value in seconds.
CT	60	Defines how many seconds the server will allow for a client to accept a completed request. If the client fails to acknowledge receipt of the request within this time, the server issues an ADAM93 USER GONE message and the client receives response 254. If you get response 254 frequently, increase the value of CT (the maximum is 32767) and also of NC and NABS.
SVC	0	Defines which SVC number is to be used. Specify your Adabas SVC.
MPMWTO	NO	Defines whether the server should send informational messages to the operator console or not. You should specify YES until you are satisfied that the server is operating correctly.
DEFAULT	None	Defines the default product to which requests will be passed. Specify SAF.
SAF PARM	SAFCFG	If you need to change the name of the configuration module (for example, you have different configuration modules with different settings), you can specify the name of the configuration module the daemon is to use. For example: SAF PARM=CFGDAEM

4 Operator Commands

MVS operator communication with the daemon is achieved using the OS/390 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
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`
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.
SSTAT	Display general statistics on the operator console for the SAF Kernel.
SUSERS	Display a list of active users.
SUSTAT User-ID	Display statistics for a specified user.
SSNAP hhhhhhhh	Display a selected portion of the SAF Kernel's memory. Operation is not terminated.
SHELP	Display all possible SAF Kernel operator commands.

5 SAF Messages

This section contains a description of the SAF Security Kernel messages.

This document covers the following topics:

Messages Displayed on the Operator Console and System Message Datasets

The following messages are displayed on the operator console and system message datasets. The messages may be issued by the SAF Security Kernel component (in a daemon, an Adabas nucleus, an Entire Net-Work node, or an Adabas SQL server) or by another product into which SAF Security is installed, such as Natural, Entire Broker, Entire Net-Work, or Adabas SQL Server.

SEFM001 ***SSSSSSSS : user : resource**

Explanation The security system determined *user* does not have authorization for *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 *user* when requesting function *FF* to be performed.

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 Software AG 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 Adabas response *XX* and subresponse *xx* to request *FF* for *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 Security Kernel (Vx.x) started**

Explanation The SAF Security Kernel initialized successfully.

Action Information message only.

SEFM009 ***Module MMMMMMMM not loaded**

Explanation The SAF Security Kernel could not load the stated module.

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

SEFM013 ***Less storage acquired than specified**

Explanation The SAF Security Kernel was not able to allocate all the storage required to satisfy the buffer size specified in its parameters.

Action Operation continues.

Action Ensure region size is sufficient and parameters are appropriate.

SEFM014 ***No storage could be acquired**

Explanation The SAF Security Kernel could obtain no storage at system start-up.

Action Operation has terminated.

Action Ensure 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.

Action A general restart is performed and the operation continues.

Action Keep all information written to DDPRINT and contact your Software AG 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 Software AG technical support representative.

SEFM017 ***Insufficient space to initialize - make Natural buffer XX**

Explanation The Natural SAF interface requires a larger value to be specified for the length of IDMSBUF parameter.

Action Increase the Natural IDSIZE parameter or NSFSIZE if using Natural 4.1 or above.

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 Software AG technical support representative.

SEFM025 ***NATURAL IDMSBUF parameter is not defined**

Explanation The Natural IDSIZE parameter has not been specified.

Action Ensure IDSIZE (or NSFSIZE if using Natural 4.1 or above) 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 / view 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 Software AG 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 Software AG technical support representative.

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.

SEFM255 ***Unauthorized use of request**

Explanation Attempted illegal use of security request.

Action Contact your Software AG technical support representative.

Operator Command Messages

The following messages are displayed in response to operator commands being processed by the SAF Security Kernel.

SEFM900 *** Operator issued command : XXXXXXXX**

Explanation SAF Security Kernel received the stated operator command.

Action Information message only.

SEFM901 *** SAF Security Kernel - general statistics (at hhhhhhhh)**

Explanation

```
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 * USERS - ACTIVE:     1 FREE:  2051  OVEWRITES:         0
```

Operator command for general statistics was issued. The address in the first line is the address of the SAF Kernel's storage cache.

SEFM909 *** SAF Gateway - shutdown initiated**

Explanation Operator issued command to shut-down the daemon started task. This message is also issued when a secure Adabas nucleus, Net-Work node or Adabas SQL server terminates.

SEFM910 ***SAF Gateway - list all active users**

Explanation

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

Operator issued command to display list of currently active users.

SEFM911 ***userid . . .**

Explanation

```
SEFM911 * SJU        CHECK(+VE) CHECH(-VE) CHECK SAVED OVERWRITES  BUFF
SEFM912 * APPLICATION       10          0          0           0   10
EFM912 * DBMS CHECK          0          0          0           0    0
EFM912 * SYSMAIN             0          0          0           0    0
EFM912 * SYSTEM FILE         2          0          0           0    2
EFM912 * PROGRAM             0          0          0           0    0
EFM912 * BROKER              0          0          0           0    0
EFM912 * NET-WORK            0          0          0           0    0
EFM912 * SQL SERVER          0          0          0           0    0
```

Operator issued command to display statistics specific to a currently active user.

SEFM913 *** No active users found in SAF Security Kernel**

Explanation No active users were found in SAF Security Kernel.

SEFM914 *** Requested user xxxxxxxx not found in SAF Security Kernel**

Explanation The requested user was not found in the SAF Security Kernel.

SEFM916 *** 129D2000 D5C1F2E2 C9C1F3F2 B79931B7 NA2SIA32.r..R.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.

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 argument(s)**

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

Daemon Messages

These informational messages are issued by the SAF Security Daemon during initialization:

SAFD04I **Input parameter: XXX**

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

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.

SAFD21I **Operator command processed successfully**

Explanation Message 21I is issued after processing of an operator command.

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).

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 SAF Return and Function Codes

This chapter covers the following topics:

SAF Return Codes

The SAF Security Kernel displays an eight-byte code containing various return and reason codes from SAF.

This information is shown in a number of messages denoted "SSSSSSSS".

Return Code Structure

The SAF return code contains the following structure:

Position Within Message Code	Information Content
Byte: 1	SAF return code (R15 after RACROUTE)
Byte: 2	Function code (see section **Internal Function Code**)
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 this information, 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 **Internal Function Code**)
Digits 5 and 6	RACROUTE return code
Digits 7 and 8	RACROUTE reason code

Refer to the IBM *External Security Interface (RACROUTE) Macro Reference* manual for MVS and VM 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.

Internal Function Codes

SAF Security Kernel 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 Net-Work access
1C	Authorize SQL Server access
44 or 6C	AuthenticateUser

7 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.

This chapter covers the following topics:

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 OS/390 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 OS/390 Router Table

- Update the OS/390 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 ID 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 OS/390. 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 user ID user2:

```
$KEY(etb.policy.quote1) TYPE(NBK) UID(user2) ALLOW SERVICE(READ)
```

To define resources to ACF2 version 6

1 The SAF Security Kernel executes as a normal started task in OS/390. 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
```