Defining Resources in the External Security System and Activating Them

Defining Resources in the External Security System and Activating Them

This section describes which resources have to be defined in the external security system in conjunction with Natural SAF Security, and how they are activated. It covers the following topics:

- Users
- Environments
- Libraries
- Programming Objects
- RPC Services
- User-Defined Resources
- Overview of Resource-Class Definitions
- Translation and Effects of Access Levels
- Examples of Resource Definitions

Note on terminology:
Some external security systems use the term "resource profile", others the term "rule". In this documentation the term "resource profile" is used.
Some external security systems use the term "resource class", others the term "resource type". In this documentation the term "resource class" is used.

Users

External Security System Definitions

The existing user definitions in the external security system can be used. No additional user-specific definitions have to be made in the external security system.

Natural SAF Security Definitions

If the NSF user options "NSF *GROUP" and "NSF *USER-NAME" are set to "Y", the user’s group and user name as defined in the external security system are passed to Natural SAF Security.

Environments

SAF Server Definitions

The resource-class name for Natural environments is defined with the macro parameter NACLSF in the configuration module of the SAF server (see Step 4 of the Natural SAF Security installation procedure). The default name is "SAGNSF".

External Security System Definitions

A Natural environment is determined by the combination of the Natural system files FNAT, FDIC, FSEC and FUSER. For each system-file combination that is to be protected, a resource profile has to be defined in the external security system.

The identification of the resource profile must be a 40-digit number corresponding to the database ID / file number (DBID/FNR) combinations of the four system files. The database IDs and file numbers must be specified in the following sequence:

1. FNAT DBID and FNR,
2. FDIC DBID and FNR,
3. FSEC DBID and FNR,
4. FUSER DBID and FNR.

Each DBID and FNR must be specified as a 5-digit number (padded with leading zeros).

For example, the following environment:

```
FNAT=(00011,00035),
FDIC=(00011,00033),
FSEC=(00011,00034),
FUSER=(00011,00032)
```

would have to be specified as follows:

```
0001100035000110003300011000340001100032
```

The access level specified in the resource profile determines whether a user has access to the environment. A user needs at least READ access to be able to access a Natural environment.

Natural SAF Security Definitions

The NSF environment option "Protect Environments" determines if access to a Natural environment is to be controlled by Natural SAF Security. If this option is set to "Y", the access level defined for the environment in the external security system determines whether a user has access to it or not.

Libraries

With Natural SAF Security, Natural libraries can be protected to control users’ access to them. You can protect a Natural library:

- independently of the environment, or
- in specific environments.

Environment-Independent Access to a Library

SAF Server Definitions

The resource-class name for Natural libraries is defined with the macro parameter NACLTC in the configuration module of the SAF server (see Step 4 of the Natural SAF Security installation procedure). The default name is "SAGNTC".

External Security System Definitions

If a Natural library is to be protected, a resource profile has to be defined for it in the external security system. The resource-profile name must correspond to the library ID and may be up to 8 characters long.

The access level specified in the resource profile is checked when a user logs on to a Natural library. A user needs at least READ access to be able to log on to a library.

Natural SAF Security Definitions

The NSF library option "Protect Libraries" determines if access to Natural libraries is to be controlled by Natural SAF Security. If this option is set to "Y", the access level defined for a library in the external security system determines whether a user can log on to the library or not.

Access to a Library in Specific Environments

SAF Server Definitions

The resource-class name for Natural libraries is defined with the macro parameter NACLTC in the configuration module of the SAF server (see Step 4 of the Natural SAF Security installation procedure). The default name is "SAGNTC".

External Security System Definitions

If a Natural library is to be protected in a specific Natural environment (Natural system-file combination), a resource profile for the environment-library combination has to be defined in the external security system. A Natural environment is determined by a one-character alias. The resource-profile name must consist of the alias and the library ID (up to 8 characters), separated by a period:

a.library-ID

Natural Security Definitions

In Natural Security, you have to define an environment profile for the environment. In the environment profile, the alias to be used has to be specified.

Natural SAF Security Definitions

The NSF library option "Protect Libraries" has to be set to "Y" to activate Natural SAF Security's library-access control.

The environment-specific library-access check is activated by setting the NSF library option "with Environment" to "Y". Access to the library is then only possible in environments to which the user has READ access.

Use of System Commands in a Library

If the NSF library option "Disable Natural Commands" is set to "Y", the access level defined for the library (or library-environment combination) in the external security system also determines whether or not users may use Natural system commands within the library. A user needs at least CONTROL access to use system commands.

Modifications on FUSER System File

If the NSF library option "Set FUSER Read-Only" is set to "Y", the access level defined for the library (or library-environment combination) in the external security system also determines whether or not a user may make modifications on the FUSER system file from within the library. A user needs at least ALTER access to make modifications on the FUSER file.

Programming Objects

With Natural SAF Security, Natural programming objects in libraries can be protected to control users' execution of them.

Only programming objects which are defined in the external security system can be protected; undefined programming objects can be executed by any user.

The protection of programming objects requires that Natural SAF Security has been installed appropriately; see Step 2 of the installation procedure.

SAF Server Definitions

The resource-class name for Natural programming objects is defined with the macro parameter NACLPG in the configuration module of the SAF server (see Step 4 of the Natural SAF Security installation procedure). The default name is "SAGNPG".

At start-up, the SAF server retrieves a list of defined programming objects from the external security system, so that Natural SAF Security can determine which objects require authorization checks without continually accessing the SAF repository. Whenever programming-object definitions in the external security system are changed, the SAF server has to be restarted for these changes to take effect.

Special Requirements for CA-ACF2 and CA Top Secret

For CA–ACF2 and CA Top Secret, the list of defined programming objects cannot be obtained directly from the security repository. Instead, it must be provided in an intermediate dataset into which the information is written by CA-ACF2 and CA Top Secret. The dataset must be allocated to the DD name "SEFEXT" in the SAF daemon's JCL. Examples are provided in the Adabas Limited Libraries source library: members SAFAEXT for CA-ACF2 and SAFTEXT for CA Top Secret. Whenever

programming-object definitions in CA-ACF2 or CA Top Secret are changed, the dataset has to be newly created for these changes to take effect.

External Security System Definitions

If a Natural programming object is to be protected, a resource profile has to be defined for it in the external security system.

The resource-profile name must correspond to the library ID, and the module name of the programming object, each of which may be up to 8 characters long and which must be separated by a period:

library-ID.module-name

The *library-ID* you choose depends on how you set the Natural SAF Security Definitions (see below).

If a Natural library is protected in a specific Natural environment (see *Access to a Library in Specific Environments* above), the programming objects contained in the library can also protected in this environment. For this purpose, a corresponding resource profile has to be defined in the external security system, identifying the environment (determined by a one-character alias), library and programming object:

a.library-ID.module-name

The access level specified in the resource profile determines whether a user can execute the programming object or not. A user needs at least READ access to be able to execute a Natural programming object.

Natural SAF Security Definitions

The NSF library option "Protect Natural Modules" determines if the exexution of a Natural programming object is to be controlled by Natural SAF Security, in which case the access level defined for the programming object in the external security system determines whether a user can execute or not.

The "Protect Natural Modules" option is only evaluated if the NSF library option "Protect Libraries" has been set to a value other than "N".

An authorization check is performed when a user attempts to execute a programming object. Depending on the setting of the "Protect Natural Modules" option, the authorization check is performed either for the user’s current library (as determined by the current value of the Natural system variable *LIBRARY-ID) or for the library in which the programming object is stored.

For example, let us assume the following situation:

- A user is logged on to the library SALARY, which contains a program BONUS.
- The library SALARY has a steplib PAYGENRL, which contains a program PAYMENTS.
- The user invokes the program BONUS, which in turn invokes the program PAYMENTS.

If the option "Protect Natural Modules" is set to "Y", Natural SAF Security checks if the user is allowed to execute

- program BONUS in library SALARY, and
- program PAYMENTS in library SALARY;

this means, the following resource profiles would be checked:

- SALARY.BONUS, and
- SALARY.PAYMENTS.

If the option "Protect Natural Modules" is set to "X", Natural SAF Security checks if the user is allowed to execute

- program BONUS in library SALARY, and
- program PAYMENTS in library PAYGENRL;

that means, the following resource profiles would be checked:

- SALARY.BONUS and
- PAYGENRL.PAYMENTS.

If authorization checks are to be environment-specific, the NSF library option "with Environment" has to be set to "Y".

Natural Application Programming Considerations

If the execution of programming objects in controlled by Natural Security, error NAT0963 is issued in the case of a user attempting to execute a programming object for which he/she has no authorization. If it is controlled by Natural SAF Security, however, error NAT0972 is issued in such a case instead.

RPC Services

With Natural SAF Security, Natural RPC services can be protected against unauthorized use. You can protect a Natural RPC service:

- independently of the environment, or
- in specific environments.

Environment-Independent Use of an RPC Service

SAF Server Definitions

The resource-class name for Natural RPC services is defined with the macro parameter NACLRP in the configuration module of the SAF server (see Step 4 of the Natural SAF Security installation procedure). The default name is "SAGNRP".

External Security System Definitions

If a Natural RPC service is to be protected, a resource profile has to be defined for it in the external security system.

The resource-profile name must correspond to the library ID and subprogram name, each of which may be up to 8 characters long and which must be separated by a period:

library-ID.subprogram-name

The access level specified in the resource profile determines whether a user can use the service or not. A user needs at least READ access to be able to execute a Natural subprogram via RPC.

Natural SAF Security Definitions

The NSF RPC option "Protect Services" determines if access to Natural RPC services is to be controlled by Natural SAF Security, in which case the access level defined for the RPC service in the external security system determines whether a user can use the service or not.

Use of an RPC Service in Specific Environments

SAF Server Definitions

The resource-class name for Natural RPC services is defined with the macro parameter NACLRP in the configuration module of the SAF server (see Step 4 of the Natural SAF Security installation procedure). The default name is "SAGNRP".

External Security System Definitions

If a Natural RPC service is to be protected in a specific Natural environment (Natural system-file combination), a resource profile has to be defined for the environment-service combination in the external security system. A Natural environment is determined by a one-character alias. The resource-profile name must consist of the alias, the library ID (up to 8 characters), and the subprogram name, separated from one another by periods:

a.library-ID.subprogram-name

Natural Security Definitions

In Natural Security, you have to define an environment profile for the environment. In the environment profile, the alias to be used has to be specified.

Natural SAF Security Definitions

The NSF RPC option "Protect Services" has to be set to "Y" or "F" to activate Natural SAF Security’s service-access control.

The environment-specific service-access check is activated by setting the NSF RPC option "with Environment" to "Y". Use of the RPC service is then only possible in environments to which the user has READ access.

User-Defined Resources

With Natural SAF Security, user-defined resources can be protected against unauthorized use. You can protect a user-defined resource:

- independently of the environment, or
- in specific environments.

Environment-Independent Use of a User-Defined Resource

SAF Server Definitions

The resource-class name for user-defined resources is defined with the macro parameter NACLAP in the configuration module of the SAF server (see Step 4 of the Natural SAF Security installation procedure). The default name is "SAGNPG".

External Security System Definitions

If a user-defined resource is to be protected, a resource profile has to be defined for it in the external security system.

The name of a resource profile can, for example, consist of a library ID, main function and subfunction. The library ID may be up to 8 characters long, the main function is usually (but not necessarily) the name of the programming object, and the subfunction is a 3-character code identifying the function to be performed. Each of the three must be separated from one another by a period:

library-ID.main-function.sub-function

The resource profile determines whether a user may access a user-defined resource or not.

Natural SAF Security Definitions

The necessary security requests are handled via application programming interfaces provided by Natural SAF Security.

Use of a User-Defined Resource in Specific Environments

SAF Server Definitions

The resource-class name for user-defined resources is defined with the macro parameter NACLAP in the configuration module of the SAF server (see Step 4 of the Natural SAF Security installation procedure). The default name is "SAGNPG".

External Security System Definitions

If a user-defined resource is to be protected in a specific Natural environment (Natural system-file combination), a resource profile has to be defined for the environment-resource combination in the external security system. A Natural environment is determined by a one-character alias. The resource-profile name is composed as above, prefixed by the alias, for example:

a.library-ID.main-function.sub-function

The resource profile determines whether a user may access a user-defined resource in that environment or not.

Natural Security Definitions

In Natural Security, you have to define an environment profile for the environment. In the environment profile, the alias to be used has to be specified.

Natural SAF Security Definitions

The environment-specific resource-access check is activated by setting the NSF user-resource option "with Environment" to "Y".

The necessary security requests are handled via application programming interfaces provided by Natural SAF Security.

Overview of Resource-Class Definitions

The following table summarized the resource-class definitions to be made in the configuration module of the SAF server:

Resource	Macro Parameter in Configuration Module	Default Name	Length of Resource-Profile Name
Environments	NACLSF	SAGNSF	40
Libraries	NACLTC	SAGNTC	10
Programming objects	NACLPG	SAGNPG	19 or 23 (*)
RPC services	NACLRP	SAGNRP	19
User-defined resources	NACLAP	SAGNPG	23

* Both NACLPG and NACLAP have the same default name, SAGNPG. If only NACLPG is used, a length of 19 is sufficient. If both NACLPG and NACLAP are used, a length of 23 is required.

Translation and Effects of Access Levels

The following table shows how CA-ACF2 translates RACF attributes, and also gives an overview of the effects of the access levels:

RACF Attribute	CA-ACF2 Resource Rule	Disabling of Natural Commands	Read-Only FUSER System File
READ	READ	Commands are disabled (same as profile parameter NC=ON).	FUSER file is read-only.
UPDATE	UPDATE	Commands are disabled (same as profile parameter NC=ON).	FUSER file is read-only.
CONTROL	DELETE	Commands are allowed (same as profile parameter NC=OFF).	FUSER file is read-only.
ALTER	ADD	Commands are allowed (same as profile parameter NC=OFF).	Modification on FUSER file are allowed.

Examples of Resource Definitions

This section provides examples of how resources are defined in the external security system:

- Example of Resource Definitions in RACF
- Example of Resource Definitions in CA-ACF2
- Example of Resource Definitions in CA Top Secret

Example of Resource Definitions in RACF

This is an example of how to define resources in RACF.

For details on RACF features, see IBM’s RACF documentation. See also the *SAF Security Kernel* documentation.

Adding a Class to the Class Descriptor Table

For details on how to add a resource class to the RACF class descriptor table, see IBM’s SPL RACF manual; for an example, see "IBM SYS1.SAMPLIB", member RACINSTL. Allocate a maximum length of 40 for the class. Define the class to enable discrete and generic profile use. 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 member RACFCLSX.

Add the resource class "SAGNSF" for Natural environments.

Updating the z/OS Router Table

Update the z/OS router table, as described in IBM’s SPL RACF manual; for an example, see "IBM SYS1.SAMPLIB", member RACINSTL, section RFTABLE.

Activating a New Resource Class

Activate the new resource class "SAGNSF" with SETROPTS (see IBM’s RACF Command Language Reference manual):

```
SETROPTS CLASSACT(SAGNSF)
SETROPTS GENCMD(SAGNSF)
SETROPTS GENERIC(SAGNSF)
```

Adding a Resource Profile for Environments and Permitting Access to it

Assume the following Natural environment (system-file combination) to be protected:

```
FNAT=(76,225)
FDIC=(76,148)
FSEC=(76,223)
FUSER=(76,1000)
```

To add a resource profile for the above environment, and grant READ access to user ID "ADE", issue the following RACF commands:

```
RDEFINE SAGNSF 000760022500076001480007600230000 7601000 UACC(NONE)
PERMIT 000760022500076001480007600230000 7601000
   CLASS(SAGNSF) ACCESS(READ) ID(ADE)
```

Example of Resource Definitions in CA-ACF2

This is an example of how to define resources in CA-ACF2.

For details on CA-ACF2 features, see Computer Associates' CA-ACF2 documentation. See also the *SAF Security Kernel* documentation.

Adding a CLASMAP Record for Environments

Add a CLASMAP record for Natural environments as follows:

```
ENTITYLN(0) MUSID() RESOURCE(SAGNSF) RSRCTYPE(NSF)
```

Defining a Resource Rule for an Environment and Allowing Access to it

Assume the following Natural environment (system-file combination) to be protected:

```
FNAT=(76,225)
FDIC=(76,148)
FSEC=(76,223)
FUSER=(76,1000)
```

To allow the above environment for all user IDs, define the following rule:

```
$KEY(000760022500076001480007600230000 7601000)
    TYPE(NSF) UID(*) SERVICE(READ,UPDATE) ALLOW
```

Disallowing Access to an Environment

To disallow access to the above environment for user ID "ADE", define the following rule:

```
$KEY(000760022500076001480007600230000 7601000)
   TYPE(NSF) UID(ADE) SERVICE(READ,UPDATE) PREVENT
```

Example of Resource Definitions in CA Top Secret

This is an example of how to define resources in CA Top Secret.

For details on CA Top Secret features, see Computer Associates' CA Top Secret documentation. See also the *SAF Security Kernel* documentation.

Adding a Resource Type for Environments to the Resource Definition Table

To add the resource type "SAGNSF" for Natural environments to the CA Top Secret resource definition table (RDT), issue the following command (see Computer Associates' CA Top Secret Reference Guide for details):

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

Adding a Resource Profile for an Environment and Assigning Ownership

Ownership must be assgined to a resource profile, before access to it can be permitted.

Assume the following Natural environment (system-file combination) to be protected:

```
FNAT=(76,225)
FDIC=(76,148)
FSEC=(76,223)
FUSER=(76,1000)
```

To define a resource profile for the above environment, and assign user "USER1" as owner to this resource profile, issue the following command:

```
TSS ADD(USER1) SAGNSF(000760022500076001480007600230000 7601000)
```

Permitting Access to an Environment

To grant user "ADE" READ access to the above environment, issue the following command:

```
TSS PER(ADE) SAGNSF(000760022500076001480007600230000 7601000)
   FAC(fac) ACCESS (READ)
```