Application Programming Interfaces

This section describes the application programming interfaces (APIs) provided by Natural SAF Security. It covers the following topics:

- Overview of Application Programming Interfaces
- APIs for User and Password Authentication
- API for Checking Resource Access to Dedicated API Class
- APIs for Maintaining Resource Profiles
- API for Checking Access Rights to a Resource
- API for Obtaining Information from the SAF Server
- API for Maintaining RACF User Definitions
- Natural Security APIs

Overview of Application Programming Interfaces

Natural SAF Security provides the following application programming interfaces (APIs):

Function	**Invoked Subprogram**	**Example Program of how to Invoke the Subprogram**
User and password authentication.	NSFNPAS	PGMSFU01
	NSFNPASZ	PGMSFU02
	NSFNPAX	PGMSFU03
Check resource access to a dedicated API class.	NSFNAPC	PGMSFC*nn*
Maintain resource profiles.	NSFNRES	PGMSFR*nn*
Check access rights to a resource.	NSFNRES, NSFNREX	PGMSFX*nn*
Obtain miscellaneous information from the SAF server.	NSFNINF	PGMSFI*nn*
Maintain user definitions in RACF.	NSFADM	PGMSAF*nn*

The example programs are provided in the Natural Security library SYSSEC.

APIs for User and Password Authentication

- NSFNPAS
- NSFNPASZ
- NSFNPAX

NSFNPAS

The subprogram NSFNPAS can be called from any Natural library to verify the authentication of a user (*USER) and, optionally, establish that the user was already logged on.

Five different sub-calls are available:

#PAS-FUNC	Action
`INDQVER`	Verify user ID (not password) and create ACEE.
`INDQVPW`	Verify user ID and password, creating new ACEE.
`INDQVPO`	Verify user ID and password without creating new ACEE (CA Top Secret only).
`INDQVPT`	Verify user ID and password without creating ACEE (CA Top Secret only).
`INDQVPC`	Verify user ID and password and change password creating new ACEE.

The parameter data area NSFAPAS is available to invoke this subprogram. Its fields are:

Field	Format/Length	Description
`#PAS-FUNC`	B1	Indicates type of verification check required.
`#PAS-RETC`	I2	Return code: 8 = error; 16 = severe error.
`#PAS-POLD`	A8	Existing (old) password.
`#PAS-PNEW`	A8	New password.
`#PAS-ACCN`	A8	Accounting information - *for future use.*
`#PAS-SERR`	B8	Return code (as described in the *SAF Security Kernel* documentation).

NSFNPASZ

To verify the password of any other user ID, the subprogram NSFNPASZ is provided.

The parameters are the same as described for subprogram NSFNPAS above.

In addition, the parameter data area NSFAPAS contains the following fields for NSFNPASZ:

Field	Format/Length	Description
#PAS-PUSER	A8	User ID of user whose password is to be changed.
#PAS-PMSG	A40	Message text returned from the SAF server.

NSFNPAX

To verify and change the password of *USER, the subprogram NSFNPAX is provided.

The parameters are the same as described for the subprogram NSFNPAS above.

In addition, the parameter data area NSFAPAS contains the following fields for NSFNPAX:

Field	Format/Length	Description
#PAS-PUSER	A8	*Not used.*
#PAS-PMSG	A40	Message text returned from the SAF server.

API for Checking Resource Access to Dedicated API Class

The subprogram NSFNAPC can be called from any Natural library to check the access to a general resource profile.

Input Parameters:

Parameter	Content
#RES-PROF	Name of desired profile.
#RES-CLAS	Name of desired class.
#RES-ATTR	Access level to be checked: H’02’ = READ access, H’04’ = UPDATE access; H’08’ = CTL access, H’80’ = ALTER access. If you specify H’00’, the highest access level will be returned.

Output Parameters:

Parameter	Content
#RES-ATTR	If H’00’ was specified as input, this field returns the highest acceptable access level.
#RES-RETC	Return code: 0 = Profile allowed for given access level. 8 = Error (in this case, the field #RES-SERR contains the SAF error code).

APIs for Maintaining Resource Profiles

- NSFNRES
- NSFNREX

NSFNRES

The subprogram NSFNRES can be called from any Natural library to read and maintain security-profile information.

RACF, CA Top Secret and CA-ACF2 enable different levels of functionality to be achieved. The different functions are shown below:

#RES-FUNC	Action
INDQRTV	Retrieve field(s) from user, group, and general profiles of the security system. CA Top Secret and CA-ACF2 allow fields such as PGMRNAME to be read from a base segment.
INDQRDN	Retrieve next resource profile in collating sequence. The name of the resource and selected field(s) can be retrieved. CA Top Secret permits only the USER class to be retrieved in this way. This functionality is currently not available with CA-ACF2.

The parameter data area NSFARES is available to invoke this subprogram. Its fields are:

Field	Format/Length	Description
#RES-FUNC	B1	Indicates function type required.
#RES-ATTR	B1	*Not used for this call.*
#RES-RETC	I2	Return code: 0 = call successful ; 4 = profile not found/EOL; 8 = error.
#RES-CLAS	A8	Required resource class/type.
#RES-GRUP	A8	Default user group - returned.
#RES-PROF	A32	Name of resource profile.
#RES-FLDA	A8/1:4	Profile field names (array).
#RES-SERR	B8	8-byte return code (as described in the *SAF Security Kernel* documentation).
#RES-SLOG	A4	*Reserved for future use.*
#RES-DATA	B16/1:16	Profile data input/output area. The data layout is described in detail in the *IBM RACROUTE* documentation.

NSFNREX

The subprogram NSFNREX is an extended version of the subprogram NSFNRES. It allows you to process up to 1024 bytes of data per request.

The parameter data area NSFAREX is available to invoke this subprogram. Its fields are identical to NSFARES (see above), except `#RES-DATA`, whose format/length is `B16/1:64`.

API for Checking Access Rights to a Resource

The subprogram NSFNRES can be called from any Natural library to test a user's authorization to any resource profile, including those used to protect Natural objects.

#RES-FUNC	Action
`INDQCHK` (`#RES-ATTR` supplied)	Check authorization at given level of access.
`INDQCHK` (`#RES-ATTR` zero)	Determine user's maximum access level.

The parameter data area NSFARES is provided to invoke this subprogram. Its fields are:

Field	Format/Length	Description
`#RES-FUNC`	B1	Indicates function type required.
`#RES-ATTR`	B1	Access level to be tested; either zero or determine highest level (as described in the *IBM RACROUTE* documentation).
`#RES-RETC`	I2	Return code: 0 = success; 8 = error.
`#RES-CLAS`	A8	Resource class/type.
`#RES-PROF`	A32	Name of resource profile.
`#RES-SERR`	B8	8-byte return code (as described in the *SAF Security Kernel* documentation).

API for Obtaining Information from the SAF Server

The subprogram NSFNINF is provided to perform a number of functions which may be useful when using Natural SAF Security.

The different functions provided are:

#INFFUNC	Action
INF-1	Determine last "access denied" message for this user.
INF-2	Determine last "access denied" message - internal format.
INF-3	Return invocation count.
INF-4	Return environment code.
INF-5	Read user name and group from values stored.
INF-6	Update user-name/group values; for example, if these are to be reformatted.
INF-7	*Currently not available.*
INF-8	*Currently not available.*
INF-9	Write SMF record.

The parameter data area NSFAINF is provided to invoke this subprogram. The local data area NSFLEQU defines the necessary equate values.

Field	Format/Length	Description
#INFFUNC	B2	Indicates function type required.
#INFRETC	I2	Return code: zero = success.
#INFDATA-SUBR	I4	Error - sub-response.
#INFDATA-TEXT	A72	Last error message.
#INF-COUNT	I4	Invocation count.
#INF-ENV	A1	Current environment code.
#INF-GROUP	A8	Group.
#INF-NAME	A32	User name.
#INF-SMFLEN	B1	Length of SMF data to be written.
#INF-SMFTXT	B255	Data to be written - A15 * 17.

API for Maintaining RACF User Definitions

The subprogram NSFADM can be invoked from any Natural library. It allows you to maintain user definitions contained in RACF from within Natural. It can only be applied to user definitions in RACF, not in other external security systems.

Performing any user maintenance function via NSFADM requires that in RACF you have the appropriate authorization to do so. That is, you can only perform these functions via Natural SAF Security if you are allowed to perform them in RACF itself.

The following functions are provided:

- Add user
- Connect user to a group
- Remove user from a group
- Delete user

For details on how to invoke the subprogram, and on the individual input and output parameters, see the source codes of the example programs PGMSAF*nn*.

Natural Security APIs

When Natural SAF Security is active, the evaluations made by some Natural Security APIs will be based not only on user data defined in Natural Security, but also on user data as defined in the external security system. This affects the following APIs:

- subprogram NSC---L,
- subprogram NSCXR with parameters `POBJ-TYPE='US'` and `SUB-TYPE='GR'`, `'GP'` and `'GM'`.