Adabas SAF Security

Adabas SAF Security Operations

Version 8.3.1

October 2021

ADABAS & NATURAL

This document applies to Adabas SAF Security Version 8.3.1 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: AAF-OPERATION-831-20210928

Table of Contents

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

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at **https://documentation.softwareag.com**.

Software AG Empower Product Support Website

If you do not yet have an account for Empower, send an email to empower@softwareag.com with your name, company, and company email address and request an account.

Once you have an account, you can open Support Incidents online via the eService section of Empower at **https://empower.softwareag.com/**.

You can find product information on the Software AG Empower Product Support website at **https://empower.softwareag.com**.

To submit feature/enhancement requests, get information about product availability, and download products, go to **Products**.

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the **Knowledge Center**.

If you have any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory at **https://empower.softwareag.com/public_directory.aspx** and give us a call.

Software AG Tech Community

You can find documentation and other technical information on the Software AG Tech Community website at **https://techcommunity.softwareag.com**. You can:

- Access product documentation, if you have Tech Community credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

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 Adabas SAF Security Operations

This document describes the operation of Adabas SAF Security in the following jobs:

- *Operation in the Adabas Nucleus*
- *Operation in Adabas Utilities*
- *Operation in the Adabas System Coordinator Daemon*
- *Operation in Entire Net-Work*

For general operations, refer to the following:

- *Adabas SAF Security Operator Commands*
- *Adabas SAF Security User Exits*

Operation in the Adabas Nucleus

Topics related to the operation of Adabas SAF Security for the protection of Adabas nucleus resources are listed here:

- Adabas Nucleus Start-up
- Adabas Nucleus Administration Functions
- Adabas Nucleus Operator Commands
- Adabas Nucleus Audit Logging for Adabas Security Violations
- Adabas Nucleus Logging On and Logging Off
- Remote Workstation Logon
- Defining Grouped Resource Names With AAFFILE
- Security Violations
- Cross-Level Security Checking
- Caching of Security Checks
- Adabas Basic Services
- Passwords and Cipher Codes
- Adabas and Natural Commands
- Remote Access to Adabas

Adabas Nucleus Start-up

Refer to Resource Names for Adabas Nucleus Start-up in the Reference section for a description of the resource name format used by Adabas SAF Security at nucleus start-up.

When starting a nucleus, Adabas SAF Security uses the access level of the User ID assigned to the nucleus job to determine whether to run in WARN mode or FAIL mode.

The following table describes the actions depending on the permissions of the starting User ID:

User ID Permissions	Action
Resource is not defined	Nucleus will abend U0042
User ID has no access to the resource	Nucleus will abend U0042
User ID has READ access to the resource	Nucleus will run in WARN mode. Failing security checks will be permitted and reported according to the SAFCFG parameter `GWMSGL`.
User ID has UPDATE access to the resource	Nucleus will run in FAIL mode. Failing security checks will be denied and reported according to the SAFCFG parameter `GWMSGL`.

When using ADASAF to protect execution of ADACOM, you must specify a valid Adabas SVC number and a database id in the ADARUN parameters, for example:

```
ADARUN PROG=ADACOM,SVC=249,DBID=55555
```

The database id does not need to be valid; it is simply used by ADASAF to build the resource name to be checked. The above example would result in ADASAF checking for read access to a resource named COM55555.SVC249.

Nucleus Start-up Resource Name Examples

The examples below refer to a nucleus starting for database 1 running under SVC 237 (all possible configuration constructions are shown):

Resource Name	Values for DELIM and DBFLEN Configuration Parameters
`NUC001SVC237`	`DELIM= N, DBFLEN=0`
`NUC001.SVC237`	`DELIM= Y, DBFLEN=0`
`NUC00001SVC237`	`DELIM= N, DBFLEN=1`
`NUC00001.SVC237`	`DELIM= Y, DBFLEN=1`
`NUC1SVC237`	`DELIM= N, DBFLEN=2`
`NUC1.SVC237`	`DELIM= Y, DBFLEN=2`

Adabas Nucleus Administration Functions

Refer to *Resource Names for Adabas Nucleus Administration Functions* in the Reference section for a description of the resource name format and full list of applicable administration functions used by Adabas SAF Security to protect nucleus administration functions.

The protection of Nucleus administration functions is activated using the SAFCFG configuration parameter `DBADMIN`.

This level of protection is available to installations running the following versions:

- Adabas version 8.5 SP1 or above

- Adabas Limited Library (WAL) version 8.5 SP1 or above

The following table shows a sample of the resource names used by Adabas SAF Security in the protection of nucleus administration functions for database 1 with SAFCFG configuration parameters `DELIM=Y` and `DBFLEN=1`.

Admin Function	Resource Name
Nucleus termination	ADANUC00001.OPERCOM_ADAEND
Increase last ASSO/DATA data set size	ADANUC00001.INCREASE
Force a PLOG switch	ADANUC00001.OPERCOM_FEOFPL

Note: When enabling file-level protection using the FILE option of the `DBADMIN` configuration parameter, all component files of an expanded file must be defined with the same permissions.

Adabas Nucleus Operator Commands

Refer to *Resource Names for Adabas Operator Commands* in *Reference* for a description of the resource name format used by Adabas SAF Security for the protection of Adabas operator commands.

Adabas operator commands entered from a z/OS console can be secured by either defining security resources for the operator commands or by defining them as belonging to predefined groups and defining security resources to represent those command groupings.

You must also relink ADAIOR to include ADAEOPV and, if command grouping is required, ADAEOPTB. For more information, refer to *ADASAF installation procedure, step 7*.

Note: Operator commands are allowed or disallowed based on either the User ID of the user who starts the Adabas nucleus, or on the identifier of the started task. The choice is not based on the User ID of the user issuing the console command.

For example, assuming that `DELIM=Y` and `DBFLEN=2`, when the operator issues a `STOPU`=X'`123`' command to database 235, Adabas SAF Security will check that the User ID under which database 235 is executing has read access to the resource OPR235.STOPU.

Adabas Nucleus Audit Logging for Adabas Security Violations

Nucleus audit logging for Adabas Security violations is activated using the SAFCFG configuration parameter `DBAUDIT`.

When an Adabas Security violation occurs, Adabas SAF Security offers the capability of auditing this event to your security product using the general-purpose security-audit request RACROUTE REQUEST=AUDIT. This request records events in system-management-facilities (SMF) type 80 records.

This auditing capability is available to installations running the following versions:

- Adabas version 8.5 SP1 or above
- Adabas Limited Library (WAL) version 8.5 SP1 or above

Adabas Nucleus Logging On and Logging Off

Normally, logging on to a database is done using an Adabas `OP` command. However, not all applications use an explicit `OP` command. Adabas SAF Security does not make any security check until the user actually attempts to access or update a file.

If multiple Adabas SAF Security targets are being controlled and these targets reside on different physical machines or nodes, each target node must have the same Logon ID and password assignment per user as every other target node.

When users log off a database, they may or may not issue an explicit CL command. By default, a close command indicates the end of Adabas SAF Security validity for that user and the user is logged off the security system. If the user again logs on to Adabas, the user's validity and access rights are checked again, as though the user were logging on for the first time.

However, in databases where users have many short-lived sessions (for example, control databases or system file databases), this imposes a considerable overhead on the security system. To log a user on generally involves reading and updating security information and building up the cached security checks anew.

To avoid these overheads, you can instruct Adabas SAF Security, via the `LOGOFF` parameter, to log users off only when they time out (or are stopped) in Adabas, or never to log users off (with the exception that, if Adabas SAF Security needs to reclaim memory, it will log off the oldest inactive user).

On the other hand, if a user's security profile changes, Adabas SAF Security will continue to use the old profile until the user times out or is stopped. So, if you choose `LOGOFF=TIMEOUT` and a user's profile changes, you should stop the user via the `STOPU` operator command or Adabas Basic Services to bring the new profile into effect. If you choose `LOGOFF=NEVER` and a user's profile changes, use Adabas SAF Security Online Services to forcibly log the user off from the security system.

Remote Workstation Logon

When a new client attempts to access an Adabas SAF secured database, they are prompted, by their local Adabas link routine, to provide a Logon ID and password/passphrase. These user credentials are then encrypted and forwarded to the target for authentication (see the section ***ADASAF User Exits*** for more information on the available encryption/decryption options).

On a successful authentication, the client’s original user call is then sent by the local Adabas link routine to the target, where the user request is executed.

The Logon ID and password/passphrase are prompted by the Adabas link routine included in the Entire Net-Work running on the supported platforms.

Entire Net-Work is a prerequisite for Adabas SAF Security remote workstation support. For more information, refer to the related *Entire Net-Work* documentation.

Note: Refer to Step 6 in the *Installation Procedure* section in the *Installation Documentation* for information regarding which version of the Adabas Limited (WAL) library is required for the support of passphrases.

Alternatives to Remote Workstation Logon

As an alternative to Remote Workstation Logon, you can configure Adabas SAF Security to use either the Entire Net-Work node or link name of the remote user as the SAF Logon ID. This may be useful when the issuer of the remote calls cannot prompt for a User ID and password (for example, if it is a server rather than a client). For more information, see the description of the `REMOTE` parameter.

Defining Grouped Resource Names With AAFFILE

Refer to *Grouped Resource Names for Adabas Files* in the Reference section for a description of the grouping capability provided by AAFFILE.

Security Violations

If the security package does not recognize the user or entity being validated, or the user does not have sufficient access authority, ADASAF returns the following response code to the user:

- 200 when running in fail mode. Application programs that operate on an ADASAF-protected nucleus must check for a non-zero response code
- Zero (0) when running in warn mode

In either case, security violations can optionally be logged in the nucleus `DDPRINT` or `SAFPRINT` output.

Cross-Level Security Checking

At its simplest, ADASAF validates that a user has the necessary authority to access or modify Adabas files. However, additional levels of security are available to protect inadvertent or unauthorized data access.

This is known as cross-level checking and allows both the user's and the job's access permissions to be verified. For example, users may be given access to production data but only when they access it from a production TP monitor or batch job.

To achieve this level of protection, ADASAF performs two security checks against the same resource profile (CMD00001.FIL00456 in the example above), but for different resource classes:

- the user's User ID is checked against the resource in the class defined by the `DBCLASS` parameter
- the originating job's User ID is checked against the resource in the class defined by the `NWCLASS` parameter

If either check fails, the Adabas command is rejected with response 200.

Choosing the XLEVEL Setting

Set `XLEVEL` to

- 0: when users' access rights are not dependent on which environment (job) the user runs in
- 1: when certain jobs (for example, test TP monitors or TSO users) are not allowed to access this database
- 2: when certain jobs (for example, test TP monitors or TSO users) are only allowed to access some files on this database
- 3: when different users have different access requirements depending on which job they are running in

The following is an example of using `XLEVEL=2`.

Assume that user ABC is allowed to update file 456 on database 1 from production CICS but not from TSO; and that user XYZ is allowed to update file 456 on database 1 from production CICS and also from TSO; and that production CICS runs under User ID PCICS.

This would require the definition of the profile CMD00001.FIL00456 in both the `DBCLASS` and `NWCLASS` resource classes and granting these permissions (`DBCLASS=ADASEC` and `NWCLASS=XLVADA`):

User	Class	Profile Name	Access
ABC	ADASEC	CMD00001.FIL00456	Read, Update
ABC	XLVADA	CMD00001.FIL00456	None
PCICS	ADASEC	CMD00001.FIL00456	None
PCICS	XLVADA	CMD00001.FIL00456	Read, Update
XYZ	ADASEC	CMD00001.FIL00456	Read, Update
XYZ	XLVADA	CMD00001.FIL00456	Read, Update

ADASAF performs the following checks:

1. ABC accesses file 456 from production CICS:
   - Does ABC (the individual user) have access to resource ADASEC /CMD00001.FIL00456? Yes.

- Does PCICS (the originating job's user) have access to resource XLVADA /CMD00001.FIL00456? Yes.
- The access is allowed.

2. ABC accesses file 456 from TSO:
   - Does ABC (the individual user) have access to resource ADASEC /CMD00001.FIL00456? Yes.
   - Does ABC (the originating job's user) have access to resource XLVADA /CMD00001.FIL00456? No.
   - The access is rejected and the command receives response 200.
3. XYZ accesses file 456 from TSO
   - Does XYZ (the individual user) have access to resource ADASEC /CMD00001.FIL00456? Yes.
   - Does XYZ (the originating job's user) have access to resource XLVADA /CMD00001.FIL00456? Yes.
   - The access is allowed.

In this way the database resources are protected not only for individuals but also for jobs. A user may only access allowed resources from jobs which also have the necessary access to those resources.

However, suppose the requirement is more complicated:

ABC is allowed to update file 456 on database 1 from production CICS but not from TSO; and user XYZ is allowed to access file 456 on database 1 from TSO but not from production CICS.

ABC's security requirements are satisfied, but XYZ can access file 456 from production CICS, even though it is not desired (because once a user has access to a resource, ADASAF will allow that access from any job which also has the necessary permissions).

To achieve this level of security, it is necessary to set the `XLEVEL` parameter to 3, which instructs ADASAF to verify a user's access to a resource profile of the format:

```
uuuuuuuu.dddddddd.ffffffff
```

where:

uuuuuuuu	is the User ID of the originating job
dddddddd.ffffffff	is the Database ID and file number, as in a standard ADASAF resource profile

The resource profile length must be defined to the security system as 26 rather than 17. Therefore, the following definitions must be made in the security system:

User	Class	Profile Name	Access
ABC	ADASEC	PCICS.CMD00001.FIL00456	Read, Update
XYZ	ADASEC	PCICS.CMD00001.FIL00456	None
XYZ	ADASEC	XYZ.CMD00001.FIL00456	Read

And disallow access to undefined resources (`DBUNI=N`) or define a profile name ABC.CMD00001.FIL00456 and give user ABC no access to it.

ADASAF now performs the following checks:

1. ABC accesses file 456 from production CICS
   - Does ABC have access to resource ADASEC /PCICS.CMD00001.FIL00456? Yes.
   - The access is allowed.
2. ABC accesses file 456 from TSO
   - Does ABC have access to resource ADASEC /ABC.CMD00001.FIL00456? No.
   - The access is rejected and the command receives response 200.
3. XYZ accesses file 456 from TSO
   - Does XYZ have access to resource ADASEC /XYZ.CMD00001.FIL00456? Yes.
   - The access is allowed.
4. XYZ accesses file 456 from production CICS
   - Does XYZ have access to resource ADASEC /PCICS.CMD00001.FIL00456? No.
   - The access is rejected and the command receives response 200.

Caching of Security Checks

The results of data access and update checks, both successful and unsuccessful, are cached by ADASAF. There are two levels of caching:

- A generalized resource cache, which contains a given number of user-based entries and holds the profile names for resources that have been successfully checked for this SAF user. Both the number of entries and the number of profile names per entry are configurable by parameter. This cache is particularly effective in conjunction with the `LOGOFF=TIMEOUT/NEVER` parameter as it precludes the need to log on repetitively to the security system (and re-populate the cached resources) in databases where users frequently log on to Adabas, do a small amount of work and logoff again. Each user entry is (256 + (`DBNCU`*17) + (`NWNCU`*17)) bytes in size and, if there are more users than entries, the oldest entry is overwritten when a new entry is required. The total size of this cache is specified by the `GWSIZE` parameter.
- The second cache is a quick look-up cache and contains an entry for each Adabas user (the number of entries is set to the value of the Adabas `NU` parameter, plus 25%, so if `NU` is 200, this cache will have 250 entries). Each entry contains 128 bytes of fixed information and eight times

the value of the `MAXFILES` parameter for holding information about files that the user has attempted to access. Whenever a user accesses or updates an Adabas file, this cached file list is checked to determine whether the user already has the necessary access level.

Adabas Basic Services

Refer to Resource Names for Adabas Basic Services in the Reference section for a description of the resource name format used by Adabas SAF Security for the protection of Adabas Basic Services.

You can use ADASAF to incorporate protection of Adabas Basic Services into your SAF security repository. This option can be activated on a nucleus-by-nucleus basis using the `ABS` parameter. There are two levels of security, as follows:

Level	Description
1	Only the main functions are protected. If a user has read access to a main function, all subfunctions are automatically permitted (`ABS=1`).
2	Subfunctions are also protected. The user must have access to the main function and the subfunction (`ABS=2`).

The resource check is performed against the resource class specified by the `DBCLASS` parameter and the resource name is built in accordance with the settings of the `DBFLEN` and `DELIM` parameters. Access to undefined resources is governed by the `DBUNI` parameter.

Passwords and Cipher Codes

Assuming that an Adabas command satisfies the appropriate security checks, ADASAF can automatically apply Adabas passwords and cipher codes if the SAF security system is RACF. At nucleus initialization, ADASAF extracts the INSTDATA field from the RACF profiles for all files in the current database (if `PRMDELIM=N`, the ACC prefixed profiles are used, otherwise the CMD prefixed profiles are used) and subsequently applies them to any command for the relevant file. You must define an ACC or CMD prefixed profile (for example ACC123FIL45 or CMD00123.FIL00045) for each file that needs a cipher code or password. Cipher codes should be specified as `C=`*`nnnnnnnn`* where *`nnnnnnnn`* is the eight-digit cipher code. Passwords should be specified as `P=`*`xxxxxxxx`*, where *`xxxxxxxx`* is the password. If a file has both, they should be separated by a comma, for example `C=12345678,P=PASSWORD`. A file may have only one cipher code and one password.

The Adabas password and cipher code can be provided by a user exit rather than being stored in RACF. This is activated by specifying `P=USEREXIT` (or `C=USEREXIT`) in the RACF INSTDATA field for the relevant file's profile. Then, whenever a command passes security checks, ADASAF invokes the user exit and uses the returned information as password or cipher code. Member ADASAFX1 in AAFvrs.SRCE contains a sample user exit, a description of the interface, and instructions for installing the exit.

As an alternative to using RACF INSTDATA, or for SAF security systems other than RACF, passwords and cipher codes may be provided at nucleus initialization time by user exit ADASAFX2. If ADASAFX2 is linked with SAFADA, no attempt is made to extract passwords and cipher codes from the security system. Instead, any passwords and cipher codes for files in the current database must be supplied by ADASAFX2.

See the section ***Password / Cipher Code Exits*** for more information.

Adabas and Natural Commands

ADASAF recognizes three categories of Adabas direct call commands:

- Data access commands (`Lx`, `Sx` and `HI`)
- Data update commands (`Ax`, `Ex` and `Nx`)
- Transaction data commands

The equivalent categories of Natural commands are

- Data access commands (`READ`, `HISTOGRAM`, `FIND`)
- Data update commands (`UPDATE`, `DELETE`, `STORE`)
- Transaction data commands (`END TRANSACTION` with operand1, `GET TRANSACTION DATA`, generated `OP` and `CL` commands with option 2 set to E). For more information, see the description of the `ETDATA` parameter.

Only these types of calls have significance for ADASAF and the related security package. ADASAF recognizes and classifies all database calls according to one of the command categories described above and performs the authorization check appropriate to the command category (that is, `ATTR=READ` for access commands and `ATTR=UPDATE` for update commands).

ADASAF authorizes use of Adabas data by building a resource name to represent the file being used and instructing the security system to validate the caller's access to that resource name. The format of the resource name is defined by the `DELIM` and `DBFLEN` configuration parameters:

*lvldbid*FIL*nnnnn*	if `DELIM=N`
CMD*dbid*.FIL*nnnnn*	if `DELIM=Y`

where

Value	Description
lvl	is the required access level (ACC for access commands and UPD for update)
dbid	represents the Database ID, which is specified in the format selected by the `DBFLEN` parameter.
nnnnn	represents the file number, which is specified in the format selected by the `DBFLEN` parameter.

For example, assuming that `DELIM=Y` and `DBFLEN=1` (5 digits, with leading zeroes), a `DELETE` against database 1, file 456, must have update access to the resource CMD00001.FIL00456.

The resource name may optionally be preceded by a prefix (as defined by the `AAFPREFX` configuration parameter) and the SAF userid of the job which issued the Adabas call (if the `XLEVEL` configuration parameter is set to 3).

As an alternative to this, you can instruct ADASAF to build grouped resource names for checking access to Adabas files. This gives a number of benefits:

- Optimize security administration by grouping related files under a single resource name, even across multiple databases
- Delegate security administration by grouping related files under the same prefix or major name
- More meaningful resource names
- No need to change the security system if a file is renumbered (the ADASAF file resource name table must be changed though)

The resource name for Adabas files may be considered to contain a number of nodes:

Type	Optional/Required	Setting
Prefix	Optional	As defined by `AAFPREFX`
Job userid	Optional	Only if `XLEVEL=3`
Major	Required	CMD00001, ACC00001, UPD00001 depending on `DELIM` and `DBFLEN`
Minor	Required	FIL00456 depending on `DBFLEN`

You can provide your own values for the Prefix, Major and Minor nodes of the resource name, to group many files together as a single resource

Remote Access to Adabas

- Calls from Inactive External Security Nodes

Remote IBM Peer-to-Peer Database Access

With ADASAF and Entire Net-Work, remote Adabas calls to a multi-user node (MPM) can be validated when ADASAF is active on all participating MVS systems.

Entire Net-Work transports the User ID, which it obtains from the active external system on the host node, to the target node. There, ADASAF uses the User ID to construct the `RACROUTE REQUEST=AUTH` security calls. Support for validation based on dynamic User ID strings or connect groups is available.

Prerequisites for IBM Peer-to-Peer Access

The prerequisites for running ADASAF with Entire Net-Work are as follows:

- All participating Entire Net-Work nodes that make remote calls to an Adabas nucleus with ADASAF active must be running a current version of Entire Net-Work. All Adabas components must be current;
- An external security system like RACF, CA-Top Secret, and CA-ACF2 must be active on every Entire Net-Work /ADASAF MPM node. The external security systems can vary from node to node, since the external security information being transported by Entire Net-Work is in a format acceptable by all systems;
- The Adabas SVC used by Entire Net-Work must be current and must contain the SAF security extensions for ADASAF.

Database Access from Remote Workstations

When ADASAF is active on a multi-user (MPM) node, you can secure remote Adabas calls with Entire Net-Work for Workstations. The Adabas link routines supplied with Entire Net-Work provide the mechanism required for the two-phase logon described in the section **Remote Workstation Logon**. Once logon has been completed, all validation of resources occurs just as it does when the remote user is executing on the mainframe.

Additionally, as described in the section **Alternatives to Remote Workstation Logon**, ADASAF can secure remote Adabas calls by selecting the Entire Net-Work Node name of the remote caller, or the Entire Net-Work Link name used by the remote caller, as the user ID on which security checks are based. If you select either (or both - different databases can use different options) of these mechanisms, you must define the appropriate Node and Link names as users in your security system, with the correct access permissions for the relevant Adabas resources.

Operating and Remote Call Characteristics

The external security User ID that is transported from the host node takes on the profile of the User ID in the external security system, the User ID must be defined with the proper authority to ensure access to only the proper Adabas resources.

Calls from Inactive External Security Nodes

A remote call to a target ADASVC with ADASAF active from an inactive external security node causes a security violation (response code 200) on the calling side.

Operation in Adabas Utilities

Topics related to the operation of Adabas SAF Security for the protection of Adabas utility resources are listed here:

- Adabas Utility Start-up
- Adabas Utility Operator Commands

Adabas Utility Start-up

Refer to Resource Names for Adabas Utilities in the Reference section for a description of the resource name format used by Adabas SAF Security at utility start-up.

Adabas SAF Security offers different levels of utility protection depending on the versions of the following products installed at your site:

- Adabas
- Adabas Limited Library (WAL)

The levels of protection are described in the following sections along with the necessary product version prerequisites:

- **Name-level protection**
- **Function-level protection**
- **Function/File-level protection**

Notes:

1. When starting an Adabas utility, Adabas SAF Security will check that the starting User ID has READ access to an appropriate resource profile determined by the selected level of protection. In all circumstances where this check fails, the utility will abend U0042.
2. A WARN mode capability for utilities is provided by the SAFCFG parameter `FAILUTI`.

Name-level protection

Name-level protection provides protection at the utility name level and is the default protection level.

Name-level protection is available to installations running the following versions:

- Any supported Adabas version
- Any supported Adabas Limited Library (WAL) version

The following table shows example resource names for an ADASAV utility executing against database 1 running under SVC 237 (all possible configuration constructions are shown):

Resource Name	Values for DELIM and DBFLEN Configuration Parameters
SAV001SVC237	DELIM= N, DBFLEN=0
SAV001.SVC237	DELIM= Y, DBFLEN=0
SAV00001SVC237	DELIM= N, DBFLEN=1
SAV00001.SVC237	DELIM= Y, DBFLEN=1
SAV1SVC237	DELIM= N, DBFLEN=2
SAV1.SVC237	DELIM= Y, DBFLEN=2

Function-level protection

Function-level protection provides protection at the utility function level and is activated by specifying SAFCFG configuration parameter `UTI`=2.

Function-level protection is available to installations running the following versions:

- Adabas version 8.5 SP1 or above
- Adabas Limited Library (WAL) version 8.5 SP1 or above

The following table shows example resource names for utilities executing against database 1 with SAFCFG configuration parameters `DELIM=Y` and `DBFLEN=1`.

Utility Function	Resource Name
ADASAV SAVE	ADASAV00001.SAVE
ADAORD REORDB	ADAORD00001.REORDB
ADADBS OPERCOM DUQA	ADADBS00001.OPERCOM_DUQA

Refer to Resource Names for Adabas Utilities in the Reference section for additional information and a full list of applicable utility functions used by Adabas SAF Security at utility start-up.

Function/File-Level protection

Function/File-level protection provides protection at the utility function/file level and is activated by specifying SAFCFG configuration parameter `UTI`=3.

Function/File-level protection extends function-level protection to include files for those utility functions that are file-related.

For utility functions that are not file-related, the protection level reverts to **Function-level protection**.

Function/File-level protection is available to installations running the following versions:

- Adabas version 8.5 SP1 or above
- Adabas Limited Library (WAL) version 8.5 SP1 or above

The following table shows example resource names for utilities executing against database 1 with SAFCFG parameters `DELIM=Y` and `DBFLEN=1`.

Utility Function	Resource Name
ADASAV SAVE FILES=3	ADASAV00001.SAVE.UFL00003
ADAORD REORFASSO FILE=10	ADAORD00001.REORFASSO.UFL00010
ADADBS OPERCOM DLOCKF=12	ADADBS00001.OPERCOM_DLOCKF.UFL00012

Refer to *Resource Names for Adabas Utilities* in the Reference section for additional information and a full list of applicable utility functions used by Adabas SAF Security at utility start-up.

Adabas Utility Operator Commands

Resource names used in the protection of operator commands issued to Adabas utilities follow the same format used for the protection of Adabas nucleus operator commands. Refer to *Resource Names for Adabas Operator Commands* in *Reference* for a description of the resource names.

Adabas operator commands entered from a z/OS console can be secured by either defining security resources for the operator commands or by defining them as belonging to predefined groups and defining security resources to represent those command groupings.

You must also relink ADAIOR to include ADAEOPV and, if command grouping is required, ADAEOPTB. For more information, refer to *ADASAF installation procedure, step 7*.

Note: Operator commands are allowed or disallowed based on the User ID of the user who starts the Adabas utility. Protection is not based on the User ID of the user issuing the console command.

For example, assuming that `DELIM=Y` and `DBFLEN=2`, when the operator issues a DSTAT command to an ADASAV utility running against database 235, Adabas SAF Security will check that the User ID under which the ADASAV job is executing has read access to the resource OPR235.DSTAT.

Operation in the Adabas System Coordinator Daemon

Topics related to the operation of Adabas SAF Security for the protection of online administration of COR-based Add-on products are listed here:

- Activating the security service in the System Coordinator daemon
- Activating security protection of online administration for SAF Security
- Activating security protection of online administration for Fastpath
- Activating security protection of online administration for Transaction Manager
- Activating security protection of online administration for Vista
- Activating security protection of online administration for System Coordinator
- Operator commands for the security service in the System Coordinator daemon

Activating the security service in the System Coordinator daemon

To activate:

1. Create a SAFCFG module (if you need different settings to the database).
2. Ensure all required libraries are in the STEPLIB concatenation. Refer to Check the STEPLIB Concatenation in the installation documentation.
3. Add PRODUCT=AAF to the daemon DDCARD dataset.

Activating security protection of online administration for SAF Security

Refer to Resource Names for Adabas SAF Security Administration Services in the Reference section for a description of the resource name format used by Adabas SAF Security for the protection of Adabas SAF Security administration services.

Once you have a running daemon security service (see **Activating the security service in the System Coordinator daemon**), use option 2 of System Settings (see System Settings in *Adabas SAF Security Online Services*) to activate SAF Security online administration protection. These settings are stored in the configuration file assigned to LFILE 152. Therefore you can use different LFILEs to vary your security behaviors. You **must** ensure the ability to change LFILE 152 (in the Natural parameter module or in Natural dynamic parameters, etc) is restricted to authorized people only!

Activating security protection of online administration for Fastpath

Refer to Resource Names for Adabas Fastpath Administration Services in the Reference section for a description of the resource name format used by Adabas SAF Security for the protection of Adabas Fastpath administration services.

Once you have a running daemon security service (see **Activating the security service in the System Coordinator daemon**), logon to the Fastpath online administration library SYSAFP and use option 2 of System Settings (see System Settings in *Adabas SAF Security Online Services* which also applies to other online administration libraries) to activate Fastpath online administration protection. These settings are stored in the configuration file assigned to LFILE 152. Therefore you can use different LFILEs to vary your security behaviors. You **must** ensure the ability to change LFILE 152 (in the Natural parameter module or in Natural dynamic parameters, etc) is restricted to authorized people only!

Activating security protection of online administration for Transaction Manager

Refer to Resource Names for Adabas Transaction Manager Administration Services in the Reference section for a description of the resource name format used by Adabas SAF Security for the protection of Adabas Transaction Manager administration services.

Once you have a running daemon security service (see **Activating the security service in the System Coordinator daemon**), logon to the Transaction Manager online administration library SYSATM and use option 2 of System Settings (see System Settings in *Adabas SAF Security Online Services* which also applies to other online administration libraries) to activate Transaction Manager online administration protection. These settings are stored in the configuration file assigned to LFILE 152. Therefore you can use different LFILEs to vary your security behaviors. You **must** ensure the ability to change LFILE 152 (in the Natural parameter module or in Natural dynamic parameters, etc) is restricted to authorized people only!

Activating security protection of online administration for Vista

Refer to Resource Names for Adabas Vista Administration Services in the Reference section for a description of the resource name format used by Adabas SAF Security for the protection of Adabas Vista administration services.

Once you have a running daemon security service (see **Activating the security service in the System Coordinator daemon**), logon to the Vista online administration library SYSAVI and use option 2 of System Settings (see System Settings in *Adabas SAF Security Online Services* which also applies to other online administration libraries) to activate Vista online administration protection. These settings are stored in the configuration file assigned to LFILE 152. Therefore you can use different LFILEs to vary your security behaviors. You **must** ensure the ability to change LFILE 152 (in the Natural parameter module or in Natural dynamic parameters, etc) is restricted to authorized people only!

Activating security protection of online administration for System Coordinator

Refer to *Resource Names for Adabas System Coordinator Administration Services* in the Reference section for a description of the resource name format used by Adabas SAF Security for the protection of Adabas System Coordinator administration services.

Once you have a running daemon security service (see ***Activating the security service in the System Coordinator daemon***), logon to the System Coordinator online administration library SYSCOR and use option 2 of System Settings (see *System Settings* in *Adabas SAF Security Online Services* which also applies to other online administration libraries) to activate System Coordinator online administration protection. These settings are stored in the configuration file assigned to LFILE 152. Therefore you can use different LFILEs to vary your security behaviors. You **must** ensure the ability to change LFILE 152 (in the Natural parameter module or in Natural dynamic parameters, etc) is restricted to authorized people only!

Operator commands for the security service in the System Coordinator daemon

Use the z/OS Modify (`F`) command. All operator commands must be prefixed with *AAF*. For example:

```
F CORDAEMN,AAF SSTAT
```

You may use the same ADASAF operator commands for the daemon service as for a database (see ***ADASAF Operator Commands***).

Note: There are no security checks for ADASAF daemon operator commands.

Operation in Entire Net-Work

Topics related to the operation of Adabas SAF Security for the protection of Entire Net-Work resources are listed here:

- Entire Net-Work Start-up
- Entire Net-Work Administration Functions

- Entire Net-Work Operator Commands

Entire Net-Work Start-up

Refer to *Resource Names for Entire Net-Work Start-up* in the Reference section for a description of the resource name format used by Adabas SAF Security at Entire Net-work start-up.

When starting an Entire Net-Work job, Adabas SAF Security uses the access level of the User ID assigned to the job to determine whether to run in WARN mode or FAIL mode.

The following table describes the actions depending on the permissions of the starting User ID:

User ID Permissions	Action
Resource is not defined	Entire Net-work job will abend U0042
User ID has no access to the resource	Entire Net-Work job will abend U0042
User ID has READ access to the resource	Entire Net-Work job will run in WARN mode. Failing security checks will be permitted and reported according to the SAFCFG parameter `GWMSGL`.
User ID has UPDATE access to the resource	Entire Net-Work job will run in FAIL mode. Failing security checks will be denied and reported according to the SAFCFG parameter GWMSGL.

Refer to the Entire-Net-Work section in *Software Prerequisites* for the necessary prerequisites for providing this start-up protection.

Entire Net-Work Start-up Resource Name Examples

The examples below refer to an Entire Net-Work job with a target ID of 1 running under SVC 237 (all possible configuration constructions are shown):

Resource Name	Values for `DELIM` and `DBFLEN` Configuration Parameters
NET001SVC237	`DELIM=N, DBFLEN=0`
NET001.SVC237	`DELIM=Y, DBFLEN=0`
NET00001SVC237	`DELIM=N, DBFLEN=1`
NET00001.SVC237	`DELIM=Y, DBFLEN=1`
NET1SVC237	`DELIM=N, DBFLEN=2`
NET1.SVC237	`DELIM=Y, DBFLEN=2`

Entire Net-Work Administration Functions

Refer to *Resource Names for Entire Net-Work Administration Functions* in the Reference section for a list of the applicable Entire Net-Work administration functions and the corresponding resource names used by Adabas SAF Security to protect such functions.

Adabas SAF Security provides protection for Entire Net-Work administration functions ensuring unauthorized use of such functions is not permitted. This protection is available for all means of administration:

- Operator commands issued from the console
- Operator commands issued using Entire Net-Work's Programmable Command Interface
- Using Adabas Manager

The protection of Entire Net-Work administration functions is activated using the SAFCFG configuration parameter `NETADMIN`.

Refer to the Entire-Net-Work section in *Software Prerequisites* for the necessary prerequisites for providing this administration function protection.

The following table shows a sample of the resource names used by Adabas SAF Security in the protection of Entire Net-Work administration functions for a target ID of 1 with SAFCFG configuration parameters `DELIM=Y` and `DBFLEN=1`.

Admin Function	Resource Name
Job termination	NETWRK00001.CONTROL
Display Nodes	NETWRK00001.DISPLAY
Set CQTIMER	NETWRK00001.MODIFY

Entire Net-Work Operator Commands

Resource names used in the protection of operator commands issued to Entire Net-Work from the console or from Entire Net-work's Programmable Command Interface, follow the same format used for the protection of **Entire Net-work Administration Functions**.

Refer to *Resource Names for Entire Net-Work Administration Functions* in the Reference section for a list of the applicable Entire Net-Work administration functions and the corresponding resource names used by Adabas SAF Security to protect such functions.

Note: The protection of Entire Net-Work operator commands is controlled by the `NETADMIN` configuration parameter and not by ADAIOR having been relinked with ADAEOPV. If `NETADMIN=Y`, then operator command protection will be enabled regardless of whether ADAIOR has been relinked with ADAEOPV or not.

Adabas SAF Security Operator Commands

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

```
F ADA123,AAF SSTAT
```

Note: The Adabas SAF Security operator commands are enabled by ADAEOPV. So you need to link ADAEOPV (and optionally ADAEOPTB) with ADAIOR (see installation procedure, step 7) and define command security rules.

Refer to *Resource Names for Adabas SAF Security Operator Commands* in *Reference* for a description of the resource name format used by Adabas SAF Security for the protection of its own operator commands.

The following table describes the Adabas SAF Security operator commands.

Command	Description
`SREST`	The `SREST` command can be used to effect security permissions changes immediately. All cached security information held by Adabas SAF Security (or by the security system itself in the job's address space) is discarded and subsequently dynamically recached. The operation is transparent to all online and batch users. **Note:** If the FASTAUTH option is specified by the SAFCFG parameters `DBCLASS` and/or `NWCLASS` then the process to effect security permission changes is as follows: 1. The security administrator must issue a SETROPTS RACLIST(*classname*) REFRESH to cause the globally shared in-storage profiles to be refreshed. 2. Issue SREST operator commands to all relevant jobs operating with Adabas SAF Security in order to discard any locally cached security information.
`SSTAT`	Display general statistics on the operator console for Adabas SAF Security. These statistics are the same as those available using Online Services.
`SUSERS`	Display a list of active users.
`SUSTAT user-id`	Display statistics for a specified user. These statistics are the same as those available from Online Services.
`SSNAP hhhhhhhh`	Display 256 bytes of the SAF Security Kernel's memory starting from address *hhhhhhhh*. The memory display is written immediately to the console as multiple SEFM916 messages.
`SSNAP=AGL`	Display the Adabas SAF Security global work area. The display is triggered by the next user command following the `SSNAP=AGL` operator command and is written to DDPRINT (regardless of the setting of `SAFPRINT`).

Command	Description
`SSNAP=CFA`	Display the Adabas SAF Security user file cache area. The display is triggered by the next user command following the SSNAP=CFA operator command and is written to DDPRINT (regardless of the setting of `SAFPRINT`).
`SHELP`	Display all possible operator commands.
`SNEWCOPY`	Restart Adabas SAF Security, discarding all cached security information and reload modules too. The `SNEWCOPY` command is used if Adabas SAF Security parameters or cipher codes or passwords held in the RACF security system need to be changed, without interrupting operation. The restart is triggered by the next user command following the issuing of the `SNEWCOPY` operator command. **Note:** If the FASTAUTH option is specified by the SAFCFG parameters `DBCLASS` and/or `NWCLASS` and the purpose of the `SNEWCOPY` is to also effect security permission changes then the correct process is as follows: 1. The security administrator must issue a SETROPTS RACLIST(*classname*) REFRESH to cause the globally shared in-storage profiles to be refreshed. 2. Issue `SNEWCOPY` operator commands to all relevant jobs in order to restart Adabas SAF Security.
`SLOGOFF userid`	All cached security information held by `Adabas SAF Security` (or by the security system itself in the job's address space) for the specified user is discarded. **Note:** If the FASTAUTH option is specified by the SAFCFG parameters `DBCLASS` and/or `NWCLASS` then it may be necessary for the security administrator to issue a SETROPTS RACLIST(*classname*) REFRESH in order to cause the globally shared in-storage profiles to be refreshed.
Note: All of the above commands can be issued using Online Services.	
`TRACE= {0|1|2|3}`	`TRACE=0`: suppress security trace `TRACE=1`: trace security violations `TRACE=2`: trace successful security checks `TRACE=3`: trace all security checks

Adabas SAF Security User Exits

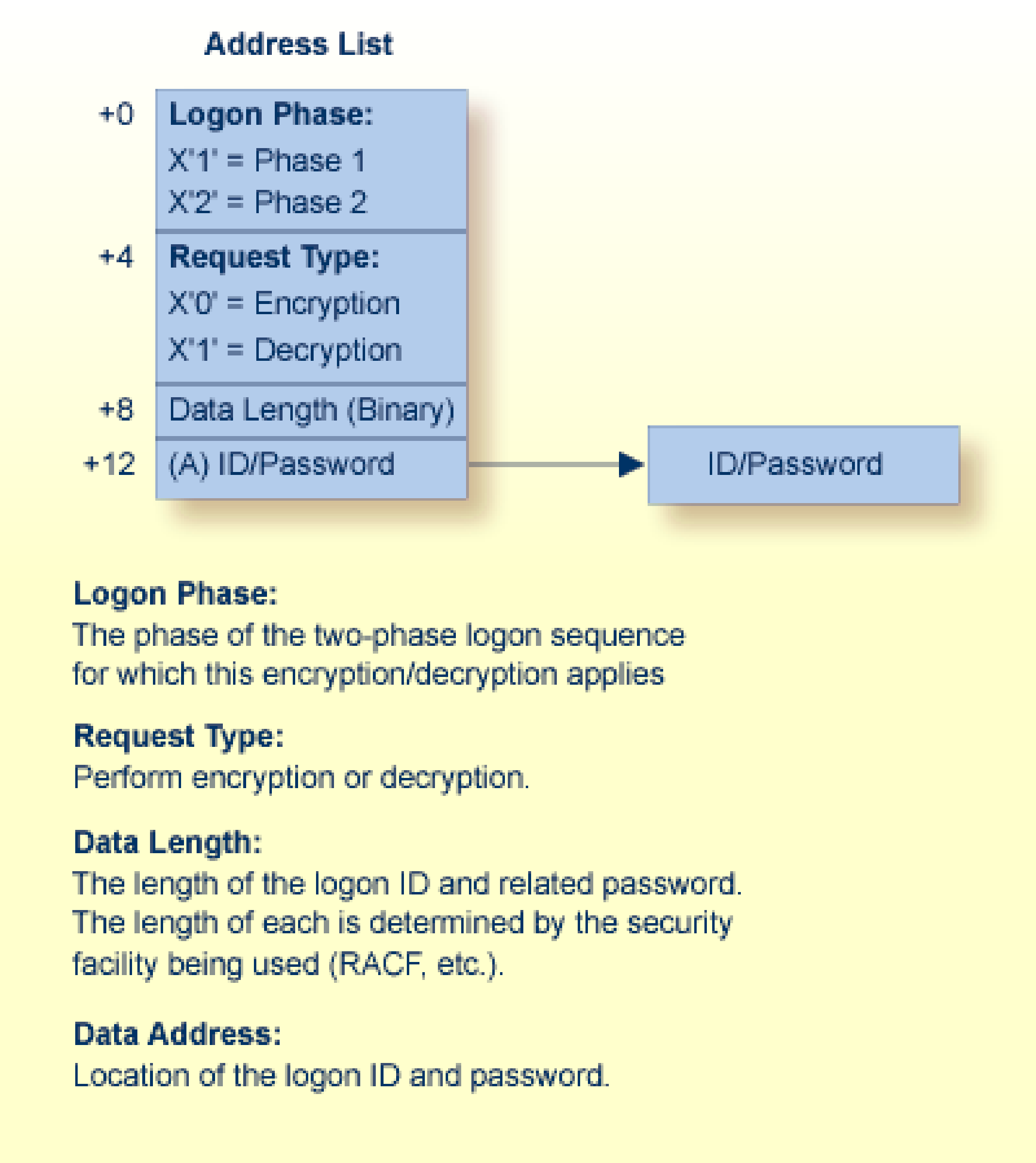
- Encryption / Decryption Exit
- Password / Cipher Code Exits

Encryption / Decryption Exit

ADASAF provides an exit for encrypting and decrypting the user Logon ID and password during the two-phase remote logon process. The encryption/decryption algorithms that are used must produce the same result on the workstation as in the Adabas ADASAF mainframe user exit. Information is provided in the appropriate *Entire Net-Work* documentation.

If a user-provided exit is not used with ADASAF, ADASAF uses its own internal encryption/decryption routines during the logon. If a user exit is used, the user exit CSECT must be "ESIEXIT" and must be linked to the SAFADA module.

The following graphic illustrates the parameter list that ADASAF passes to the ADASAF user exit:



Logon Phase:
The phase of the two-phase logon sequence for which this encryption/decryption applies

Request Type:
Perform encryption or decryption.

Data Length:
The length of the logon ID and related password. The length of each is determined by the security facility being used (RACF, etc.).

Data Address:
Location of the logon ID and password.

Linking User Exit ESIEXIT into ADASAF

The following example shows how to link the ADASAF user exit ESIEXIT module into the ADASAF module:

```
//JOB
//LKESI       EXEC PGM=IEWL,PARM='XREF,LET,LIST,NCAL,REUS'
//SYSPRINT    DD SYSOUT=*
//SYSUT1      DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSLMOD     DD DSN=YOUR.APF.LOADLIB,DISP=SHR (target execution loadlib)
//AAFLIB      DD DSN=AAFvrs.LOAD,DISP=SHR      (distributed ADASAF loadlib)
//YOURLIB     DD DSN=YOUR.USER.EXIT.LOADLIB,DISP=SHR (user exit loadlib)
//SYSLIN      DD *
INCLUDE  AAFLIB (SAFADA)                             (ADASAF module)
INCLUDE YOURLIB (ESIEXIT)        (your encryption/decryption module)
NAME SAFADA (R)
/*
```

Password / Cipher Code Exits

If you want ADASAF to provide Adabas passwords and cipher codes, but for any reason these cannot be stored in RACF (or you use a different security system), you may use user exits to return the passwords and cipher codes to ADASAF. These exits are only invoked if you have set the `PASSWORD` or `CIPHER` configuration parameter to Y.

ADASAFX1

The ADASAFX1 user exit is used to supply passwords/cipher codes at Adabas command execution time. It is invoked for every file where the RACF profile's INSTDATA specifies `P=USEREXIT` or `C=USEREXIT`. The user exit must be re-entrant and must have a CSECT name of ADASAFX1. Addressing mode on entry is 31-bit and the exit must return in the same mode.

Example

To link the exit into ADASAF, use a job similar to the following:

```
//LINKSAF  EXEC PGM=IEWL,
//      PARM=`MAP,LET,LIST,XREF,NCAL,REUS'
//SYSPRINT DD   SYSOUT=X
//SYSUT1   DD   UNIT=SYSDA,SPACE=(CYL,(1,1))
//AAFLOAD  DD   DSN=AAFvr1.LOAD,DISP=SHR
//EXITLOAD DD   DSN=your.LOAD,DISP=SHR
//SYSLMOD  DD   DSN=your.LOAD,DISP=SHR          must be APF-authorized
//SYSLIN   DD   DDNAME=SYSIN
//SYSIN    DD     *
MODE AMODE(31),RMODE(ANY)
INCLUDE AAFLOAD(SAFADA)
INCLUDE EXITLOAD(ADASAFX1)
NAME SAFADA(R)
```

Registers

The registers on entry to ADASAFX1 are as follows:

R1	Address of the parameter address list
RD	Address of two consecutive 18-word save areas
RE	Return address
RF	ADASAFX1 base address

All registers must be restored to their contents on entry before returning to ADASAF.

R1 on entry contains the address of a six-word address list:

Word 1	Address of call type. Call type is a single byte. If set to X"80", ADASAF expects a password; if set to X"40", ADASAF expects a cipher code.
Word 2	Address of return code. Return code is a full word. If set to X"00000000", ADASAF uses the value returned by the exit as password or cipher code. Otherwise, ADASAF leaves the Adabas control block unchanged.
Word 3	Address of the database ID. The database ID is a two-byte binary number.
Word 4	Address of the file number. The file number is a two-byte binary number.
Word 5	Address of the returned password/cipher code. This is an eight-byte field containing binary zeros on entry. It should be set to the desired password or cipher code, which ADASAF inserts into the Adabas control block if the return code in parameter 2 is 0.
Word 6	Address of the Adabas parameter list for the command being processed. The first word of this parameter list contains the address of the Adabas Control Block, when running in an Adabas Version 7 nucleus, or of the Extended Adabas Control Block, when running in an Adabas Version 8 (or higher) nucleus.

ADASAFX2

The ADASAFX2 user exit is used to supply passwords/cipher codes at nucleus initialization time. It is invoked by ADASAF during nucleus initialization and may return a password and or a cipher code for as many files as required (providing the value of `MAXPC` is not exceeded). The user exit must be re-entrant and must have a CSECT name of ADASAFX2. Addressing mode on entry is 31-bit and the exit must return in the same mode.

Note: If ADASAFX2 is used, ADASAF does not attempt to extract passwords and cipher codes from RACF INSTDATA.

Example

To link the exit into ADASAF, use a job similar to the following:

```
//ADASAF   EXEC PGM=IEWL,
//      PARM='MAP,LET,LIST,XREF,NCAL,REUS'
//SYSPRINT DD   SYSOUT=X
//SYSUT1   DD   UNIT=SYSDA,SPACE=(CYL,(1,1))
//AAFLOAD  DD   DSN=AAFvrl.LOAD,DISP=SHR
//EXITLOAD DD   DSN=your.LOAD,DISP=SHR
//SYSLMOD  DD   DSN=your.LOAD,DISP=SHR          must be APF-authorized
//SYSLIN   DD   DDNAME=SYSIN
//SYSIN    DD   *
 MODE AMODE(31),RMODE(ANY)
 INCLUDE AAFLOAD(SAFADA)
 INCLUDE EXITLOAD(ADASAFX2)
 NAME SAFADA(R)
```

Registers

The registers on entry to ADASAFX2 are as follows:

R1	Address of the parameter address list
RD	Address of an 18-word save area
RE	Return address
RF	ADASAFX2 base address

All registers must be restored to their contents on entry before returning to ADASAF.

R1 on entry contains the address of a five-word address list:

Word 1	Address of Database ID. The Database ID is a two-byte binary number. ADASAF passes the current Database ID to the exit.
Word 2	Address of return code. The return code is a four-byte binary number and must be set by ADASAFX2. ADASAF will call the exit repetitively until the return code is not 0. If set to 0, ADASAF will use the values returned by the exit. If not 0, ADASAF will not call the exit again (but will still use the values returned on previous calls).
Word 3	Address of file number. The file number is a two-byte binary number. The first time in, this will be X"0000". On subsequent calls it will contain the most recently returned file number. The exit must set it to the file number to which the returned password or cipher code applies.
Word 4	Address of code type. This is a one-byte binary field. The exit must set this to X"40" when returning a cipher code and to X"80" when returning a password.
Word 5	Address of an eight-byte password/cipher code. The exit must set this to the appropriate password or cipher code. If ADASAFX2 sets this to USEREXIT, ADASAF will subsequently invoke ADASAFX1 to provide a password or cipher code at Adabas command execution time.