

# **Adabas SAF Security**

## **Adabas SAF Security Configuration**

Version 8.4.1

October 2022

This document applies to Adabas SAF Security Version 8.4.1 and all subsequent releases.

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

Copyright © 2022 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

**Document ID: AAF-CONFIG-841-20221009**

## Table of Contents

1 About this Documentation .....	1
Document Conventions .....	2
Online Information and Support .....	2
Data Protection .....	3
2 Adabas SAF Security Configuration .....	5
ADASAF Configuration Parameters .....	6
Defining Resource Profiles .....	6
Defining Resources to RACF, CA ACF2, and CA Top Secret .....	7
Using Adabas SQL Server Versions 1.4.2 and 1.4.3 .....	12



# 1

## About this Documentation

---

■ Document Conventions .....	2
■ Online Information and Support .....	2
■ Data Protection .....	3

## Document Conventions

---

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

## Online Information and Support

---

### Product Documentation

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

In addition, you can also access the cloud product documentation via <https://www.software-ag.cloud>. Navigate to the desired product and then, depending on your solution, go to “Developer Center”, “User Center” or “Documentation”.

### Product Training

You can find helpful product training material on our Learning Portal at <https://knowledge.softwareag.com>.

## Tech Community

You can collaborate with Software AG experts on our Tech Community website at <https://tech-community.softwareag.com>. From here you can, for example:

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

## Product Support

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

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

## Data Protection

---

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





## 2      **Adabas SAF Security Configuration**

---

■ ADASAF Configuration Parameters .....	6
■ Defining Resource Profiles .....	6
■ Defining Resources to RACF, CA ACF2, and CA Top Secret .....	7
■ Using Adabas SQL Server Versions 1.4.2 and 1.4.3 .....	12

This document describes the ADASAF configuration.

## ADASAF Configuration Parameters

---

Site-dependent ADASAF configuration parameters are specified using an assembled configuration module called SAFCFG, which is supplied as part of the SAF Security Kernel on the Adabas limited libraries.

Some ADASAF options can be overridden on a nucleus-by-nucleus basis by providing them in a dataset referenced by the DD name DDSAF.

Because of the sensitivity of SAF security, the ability to change the configuration module or the DDSAF dataset must be tightly controlled by the external security system.

For a complete description of ADASAF parameter usage, see the section ADASAF Parameters.

## Defining Resource Profiles

---

Before defining resource profiles, you need to decide which Adabas SAF Security features you will use, what format the resource names should take, and which users and jobs should be allowed to access or update which resources.

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

At the very least - for each type of server protected by Adabas SAF Security - you need to define the following:

- The resource class/type used for standard security checks (SAFCFG parameter DBCLASS).
- The execution profile, to which the user assigned to the server job has read access (to run in WARN mode) or update access (to run in FAIL mode).
- Resource profiles, to which appropriate users or groups must then be granted the necessary access permissions.

Resource profiles are defined using upper case characters only and their format can be influenced by SAFCFG parameters such as DBFLEN and DELIM.

Refer to Adabas SAF Security Operations for information about the protection capabilities offered by Adabas SAF Security when operating in different environments and information about the required resource profiles to implement such protection.

## Defining Resources to RACF, CA ACF2, and CA Top Secret

---

In order to secure Adabas it is necessary to define resource profiles to the security product installed at your site. Each security product provides the necessary facilities required for maintaining resource profiles. This section provides an overview of those facilities for each of the security products.

RACF enables the grouping of similar resource profiles into a resource class. CA ACF2 and CA Top Secret provide resource types which give equivalent functionality.

The default name of the resource class/type for Adabas is ADASEC. This can be changed by the parameter `DBCLASS` for normal checks and `NWCLASS` for cross-level checks.

Refer to Resource Name Maximum Lengths in the Reference section for important information regarding the maximum length of Adabas resource names. It is important that the resource class/type can support the appropriate resource name lengths for the installation configuration.

- [Defining Resources to RACF](#)
- [Defining Resources to CA ACF2 Version 5](#)
- [Defining Resources to CA ACF2 Version 6 and Above](#)
- [Defining Resources to CA Top Secret](#)

### Defining Resources to RACF

This section explains how to add the required resource profiles to RACF. For more details of the procedures to be followed, consult the relevant IBM documentation.

- [Add Classes to Class Descriptor Table](#)
- [Update MVS Router Table](#)
- [Activate New Classes](#)
- [Assign User ID for Started Task](#)
- [Permit Users Access to Resource Profiles](#)

#### Add Classes to Class Descriptor Table

Resource classes used must be added to the RACF class descriptor table. For further details, refer to the *IBM SPLRACF* documentation; an example is given in IBM SYS1.SAMPLIB, member RA-CINSTL. Classes must be allocated the maximum lengths as described above and be defined for discrete and generic profile use. Other attributes control the level of RACF logging and SMF recording when executing RACROUTE calls.

## Update MVS Router Table

Update the MVS router table; refer to the *IBM SPL RACF* documentation. An example is given in IBM SYS1.SAMPLIB, member RACINSTL, section RFTABLE.

## Activate New Classes

Activate new resource classes with SETROPTS (see *IBM RACF Command Language Reference* documentation). For example, activate class ADASEC:

```
SETROPTS CLASSACT(ADASEC)
```

```
SETROPTS GENCMD(ADASEC)
```

```
SETROPTS GENERIC(ADASEC)
```

## Assign User ID for Started Task

The Adabas nucleus started task or batch jobs require a User ID having the relevant RACF authorizations including the ability to perform `RACROUTE,TYPE=EXTRACT` calls on profiles belonging to the above classes.

## Permit Users Access to Resource Profiles

Add profiles to protect the different resources and permit users the required level of access. The following RACF commands add resource profile `CMD00001.FIL01234` and grant User ID `user2` READ access:

```
RDEFINE ADASEC CMD00001.FIL01234 UACC(NONE)
```

```
PERMIT CMD00001.FIL01234 CLASS(ADASEC) ACCESS(READ) ID(user2)
```

## Defining Resources to CA ACF2 Version 5

This section explains how to define resources to ACF2 version 5. For more details, consult the relevant ACF2 documentation.

- [Add Task User ID to ACF2](#)
- [Activate the SAF interface](#)
- [Insert the SAFSAVE Record](#)
- [Insert the SAFPROT Record](#)
- [Insert the SAFMAPS Record](#)

- Define Resource Rules

### Add Task User ID to ACF2

Adabas normally executes as an MVS started task. Define user ID of started task to ACF2 with the following attributes. To avoid the NON-CNCL attribute, APAR TW95626 must be applied.

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

### Activate the SAF interface

```
GSO OPTS - SAF
```

### Insert the SAFSAVE Record

Switch off all SAF checks:

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

### Insert the SAFPROT Record

Switch on the SAF security checks for Adabas:

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

### Insert the SAFMAPS Record

For the general resource class names used by ADASAF define a 3-character ACF2 resource type code:

```
SAFMAPS MAPS(ADA/ADASEC)
```

### Define Resource Rules

Define the required security profiles in ACF2 using the new type code.

The following example shows resource CMD00001.FIL01234 being added allowing read access for user ID user2:

```
$KEY(CMD00001.FIL01234) TYPE(ADA) UID(user2) ALLOW SERVICE(READ)
```

## Defining Resources to CA ACF2 Version 6 and Above

This section explains how to define resources to ACF2 version 6. For details of the procedures required, consult the current ACF2 documentation.

- [Add Task User ID to ACF2](#)
- [Insert the SAFDEF Record](#)
- [Insert the CLASMAP Record](#)
- [Define Resource Rules](#)

### Add Task User ID to ACF2

Adabas normally executes as an MVS started task. Define User ID of started task to ACF2 with the following attributes:

```
MUSASS,STC
```

Both ACF2 6.1 and 6.2 no longer need TW95626 since the current versions are more SAF-compliant.

### Insert the SAFDEF Record

The SAFDEF record must be inserted as follows:

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

### Insert the CLASMAP Record

For the general resource class names used by ADASAF define a 3-character ACF2 resource type code:

```
CLASMAP ENTITYLN(0) MUSID() RESOURCE(ADASEC) RSRCTYPE(ADA)
```

### Define Resource Rules

Define the required security profiles in ACF2 using the new type code.

The following example shows resource CMD00001.FIL01234 being added allowing read access for User ID user2:

```
$KEY(CMD00001.FIL01234) TYPE(ADA) UID(user2) ALLOW SERVICE(READ)
```

## Defining Resources to CA Top Secret

This section describes the definition of resources to CA-Top Secret. For more details, consult the relevant CA-Top Secret documentation.

- [Defining Resources to CA-Top Secret Facility](#)
- [Assign User ID for Started Task](#)
- [Add Procedure Name for Started Task](#)
- [Add Resource Type to Resource Definition Table](#)
- [Assign Ownership of Resources](#)
- [Permit Access to Defined Resources](#)

### Defining Resources to CA-Top Secret Facility

ADASAF issues authorization checks against a specific CA-Top Secret facility. The default facilities are batch and STC. Different facilities can be defined enabling, for example, development and production environments to be secured separately. Additional facilities are added by modifying predefined models. The following attributes are important and should be assigned when modifying facilities for ADASAF:

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

### Assign User ID for Started Task

Add one User ID for each instance of the Adabas 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)
```

### Add Procedure Name for Started Task

The procedure name under which the Adabas 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)
```

### Add Resource Type to Resource Definition Table

Resource types must be added to the CA-Top Secret resource definition table (RDT). Resource definitions relating to Adabas are kept in resource type ADASEC. Refer to *CA-Top Secret Reference Guide* for a detailed explanation of the following command and arguments:

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

### 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) ADASEC(CMD00001.FIL01234)
```

This makes user user1 owner of resource CMD00001.FIL01234.

### Permit Access to Defined Resources

Access to resource profiles is granted as follows. The example gives User ID user2 read access to database 1, file 1234.

```
TSS PER(user2) ADASEC(CMD00001.FIL01234) FAC(fac) ACCESS(READ) ↵
```

The facility argument can be omitted if Adabas operates under the master facility Batch or STC.

## Using Adabas SQL Server Versions 1.4.2 and 1.4.3

---

This section describes how to use Adabas SAF Security to protect calls received from Adabas SQL Server, based on the SQL table name. This feature is only available for Adabas SQL Server versions 1.4.2 and 1.4.3 operating on z/OS, OS390 or compatible platforms.

Using ADASAF, together with Entire Service Manager and Adabas SQL Server, it is possible to

- Authenticate Adabas SQL server clients
- Authorize SQL table access
- Authorize SQL table update

Security authorization is sought for all requests processed by Adabas SQL Server. Commands are routed to their final Adabas target only if the originating user's User ID and password are authenticated and the user has sufficient authorization for the requested SQL objects. Otherwise, the request is returned to the calling application with SQL response code -3004.



SQL objects are defined to the host security repository. Users are given the required access level. The SQL object name is prefixed with the name of the database as defined in the SQL catalog. Using this convention development users can, for example, be restricted to development databases. Access to data is controlled according to an individual's read/update authorization for each SQL object within the SQL database.

The following tasks are necessary when securing access through Adabas SQL Server:

- define Adabas SQL Server protection configuration options, as described below
- attach security component to Entire Service Manager (ESG)
- make the necessary definitions in your SAF security system, as described below
- configure an ADASAF-protected nucleus to provide SQL table security



**Note:** The SQL `connect` statement must be utilized by the calling application to provide User ID and password

## Configuration Options

In addition to the configuration options described in this document and the general options described in the *SAF Security Kernel* documentation, the following SAFCFG options are relevant for protection of Adabas SQL Server tables:

SAFCFG Option	Description
SQNCU=0	Specify the number of SQL table checks to be buffered per user, in the cache defined by GWSIZE.
SQUNI=Y   N	Whether or not to allow access to undefined tables (where the security system differentiates between undefined and disallowed resources).
SQTBCAT=Y   N	Determine table names from SQL catalog. You should set this to Y.
SQCLASS=ADASEC	Specify the name of the resource class to be used for checking SQL tables. This name should be up to eight alphanumeric characters and may be the same as or different from that used for database checks (as specified by the DBCLASS parameter).
GWDBID=nnnnn	Specify an ADASAF-protected database for which SQL table authorization is to be performed.

## Entire Service Manager (ESG) Considerations

The security component attached to ESG requires 32K of storage from the ESG thread.

You will need a reentrant ADALNK (created from ADALNKR in the Adabas source library). Ensure that the equates LNUINFO, LRVINFO, LEXITB, LEXITA and LXITAFP are all 0 and that SVCNR is set to your Adabas SVC number.

You must then link the Adabas SQL Server security module, NA2PESQ, using the following linkage editor control statements:

Statement	Description
INCLUDE AAFLIB(NA2PSQ1)	supplied ADASAF load library
INCLUDE CFGLIB(SAFCFG)	your assembled SAFCFG module
INCLUDE ADALIB(ADALNKR)	reentrant ADALNK
NAME NA2PESQ	

Finally, ensure that NA2PESQ and ESQUEX5 (from the ADASAF load library) are available in one of the load libraries in ESG's COMPLIB concatenation and that they are both named in ESG's SYSPARM member as RESIDENTPAGE modules.

## Configuring the ADASAF-Protected Nucleus

The nucleus specified with GWDBID must be configured to handle SQL table authorization requests. The SQL table names (together with their corresponding Database IDs and file numbers) must first be extracted from the SQL catalog. This can be done using ESQBIF in a batch job, with SYSIN similar to the following:

```
SELECT CAT02_TABLE_ID, CAT02_DB_NR, CAT02_FILE_NR, CAT02_SCHEMA_ID
```

```
FROM DEFINITION_SCHEMA.ESQCAT1 WHERE CAT02_TABLE_TYPE EQ 'BASE TABLE';
```

Direct the output (SYSPRINT) to a dataset and allocate this dataset to DD name SEFDD1 in the JCL that executes the specified nucleus.

If your SQL catalog regularly changes, you can add a step to execute ESQBIF before executing the Adabas nucleus and pass the output through to the nucleus step's SEFDD1 DD statement.

## Security Definitions

ADASAF authorizes SQL tables against the class specified for SQCLASS (default ADASEC), using resource names of the format:

```
Database-name.table-name
```

For example

```
ESQ1.CRUISE
```



**Note:** ADASAF always operates in fail mode for SQL table protection.

