CA OPS®/MVS Event Management and Automation

Security Guide

Release 12.1

CA Technologies Product References

This document references the following CA Technologies products:

- CA OPS/MVS® Event Management and Automation (CA OPS/MVS)
- CA ACF2™ for z/OS (CA ACF2)
- CA Top Secret® for z/OS (CA Top Secret)

Documentation Changes

The following documentation updates have been made since the last release of this documentation:

Note: In PDF format, page references identify the first page of the topic in which a change was made. The actual change may appear on a later page.

- Updated the How Security Options Interact (see page 13) section.
- Added EXTSECSQLSUFFIX to the Set Parameters that Allow External Security (see page 17) section.
- Updated the Remove SAF Authority Using DEFSAF (see page 21) section.
- Updated the Remove SAF Authority Using DEFSAF CA Top Secret section.
- Updated the Define Profiles Automatically with DEFSAF (see page 27) CA Top Secret section.
- Updated the Create Access Permissions with CA Top Secret (see page 30) section.
- Added the SQL TBL.CMD Names Table (see page 63) section.
- Updated the Define Roles Automatically with DEFSAF (ACF2) (see page 38) section.
- Updated the Commands and Functions that Generate External Security Events (see page 64) section.

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 5: Implementing External Security with RACF 49

Appendix A: Resource Tables and Predefined Resources 61

Appendix B: Troubleshooting External Security 73

Index 75

Chapter 1: Introduction to External Security

Note: External Security is the sole subject of this guide. See the *Administration Guide* and *AOF Rules User Guide* for documentation on rule-based security and the security user exit.

This section contains the following topics:

How to Control Access to Product Resources with External Security

As a security administrator in your mainframe environment, your responsibilities include implementing security in CA OPS/MVS.

You can perform the following tasks using external security:

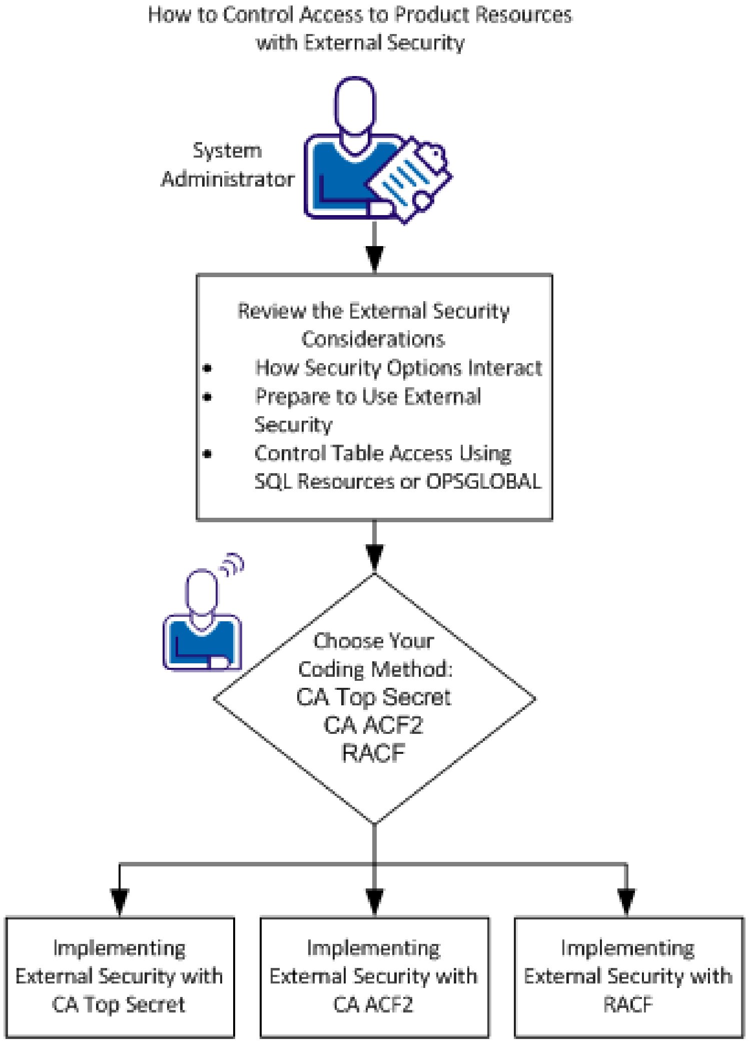
- Check users executing commands and functions on an OSF server.
- Use standard system authorization facility (SAF) calls to control access to CA OPS/MVS resources without coding AOF rules or using the assembler exit. SAF lets you protect commands and features.

  Note: Consider keeping your currently supported methods with the assembler exit and existing rules in place for backward compatibility issues.
- Use the external security interface to implement external security within CA OPS/MVS.

  The following external security packages provide a high degree of control over unique resources used by CA OPS/MVS:

  - CA Top Secret
  - CA ACF2
  - RACF
- Centralize the maintenance of your security.

The following illustration shows the process for using external security to control user access to resources.


How to Control Access to Product Resources with External Security
System Administrator
Review the External Security Considerations
• How Security Options Interact
• Prepare to Use External Security
• Control Table Access Using SQL Resources or OPSGLOBAL
Choose Your Coding Method: CA Top Secret CA ACF2 RACF
Implementing External Security with CA Top Secret
Implementing External Security with CA ACF2
Implementing External Security with RACF

These chapters help you control access to product resources with external security:

- External Security Considerations (see page 13)
  - How Security Options Interact (see page 13)
  - Prepare to Use External Security (see page 16)
  - Control Table Access Using SQL Resources or OPSGLOBAL (see page 22)
- Implementing External Security with CA Top Secret (see page 23)
- Implementing External Security with CA ACF2 (see page 35)
- Implementing External Security with RACF (see page 49)

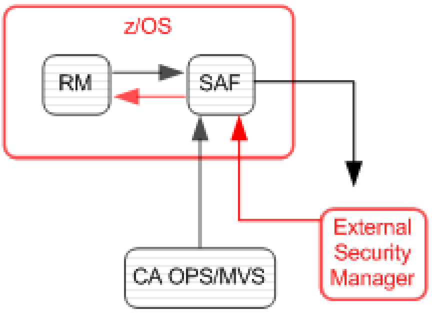
How System Authorization Facility Works

CA OPS/MVS provides external security using System Authorization Facility (SAF) calls to the security product of your choice. CA OPS/MVS makes standard SAF calls to your external security manager using defined resource names. These calls check for security access to its resources.

The System Authorization Facility (SAF) is part of z/OS and initiates the following process:

- Provides the standard interface between z/OS and any external security manager.
- Receives an access request from a resource manager within z/OS and directs control to the external security manager.

The following illustration demonstrates how SAF works:



Chapter 2: External Security Considerations

This section contains the following topics:

How Security Options Interact

An understanding of all the options and how they interact is essential to choosing the right combination for your site. CA OPS/MVS has several security options that can interact in the following ways:

- When a user attempts to access a resource, a security check determines the authority of the user initiating the request.
- The user ID the security check (SAF call) uses depends on where the CA OPS/MVS command or function was issued, for example:
  - Commands and functions that are issued from within a REXX program and initiated from the CA OPS/MVS address space use the user ID assigned to the CA OPS/MVS started task.
  - When users execute commands and functions on one of the OSF servers, security uses the value in OSFCONSOLE or OSFPRODUCT to verify the following resources:
    - The user when OSFSECURITY is set to CHECKUSERID.
    - The user ID associated with the OSF started task when OSFSECURITY is set to NOSECURITY.
- When running CA OPS/MVS with the parameter EXTSECURITY set to ON, also set the following OSF parameters as shown:

```
OSFSECURE   CHECKUSERID
OSFCONSOLE  <site-defined-userid>
OSFPRODUCT  <site-defined-userid>
```

site-defined-userid

Specifies the user ID to authorize for any or all of the CA OPS/MVS facilities secured using external security.

- Security performs the following steps when EXTSECURITY is OFF:
  - Checks for TSO OPER authority.
  - Checks for the existence of security rules and calls the rule when defined.
  - Calls the user exit when no security rule exists for the event.

  Generally, when EXTSECURITY is OFF, the logic flow does not change.

- Security performs the following steps when EXTSECURITY is ON:
  - The SAF call reviews the security for external security resources as follows:
    - If the SAF result is 0 or 4, it reviews your security rules.
    - If the SAF result is 0, it calls the rules. The call is made because the security rules provide a greater degree of refinement than external security.
    - If the SAF result is greater than 4, it rejects the command and it stops further checks.
  - CA OPS/MVS calls the user exit OPUSEX, if available, when no security rule exists for the event.

  When EXTSECURITY is ON its external resource checking takes control except for the security rules, which can still be coded to supplement or refine it.

Review the following security rules, which perform more specific checks:

- Limit Update Authority to Specific Parameters (see page 14, see page 15)
- Limit Specific Users Update Authority (see page 15)

Limit Update Authority to Specific Parameters

You want the user OPSUSR to read and update all CA OPS/MVS parameters except when it involves changing the STATEMAN parameter to a new value. This setting limits the access authority of the user.

Follow these steps:

1. Turn on external security.

   ```
   EXTSECURITY=ON
   ```

2. Grant user OPSUSR UPDATE access to the external security resource OP$MVS.OPSPARM.
3. Use the following security rule to prevent any user from setting the STATEMAN parameter to a new value:

```
)SEC OPSPARM
)PROC
IF SEC.AUPAPANA = 'STATEMAN' THEN
  RETURN REJECT
ELSE
  RETURN NOACTION
```

The external security check permits all other OPSPARM calls.

Limit Specific Users Update Authority

Use security rules to limit user access to resources:

- Grant users OPSADMIN and SSMADMIN access to the CA OPS/MVS RDF table.
- Grant only one user ID, SSMADMIN, authority to read or update the CA OPS/MVS master SSM table (STCTBL).

You can limit update authority to specific users. The following security rule does a more specific security check when you limit the authority of your users.

Follow these steps:

1. Turn on external security.

   ```
   EXTSECURITY=ON
   ```

2. Grant the user OPSADMIN UPDATE access to external security resource OP$MVS.SQL.*.
3. Use the following security rule to allow a specific user to access the STCTBL table:

   ```
   )SEC SQL*
   )PROC
   IF SEC.AUSQTBLS <> 'STCTBL' THEN RETURN 'NOACTION'
   IF SEC.OPAUJBNA = 'SSMADMIN' THEN
     RETURN ACCEPT
   ELSE
     RETURN REJECT
   ```

   External security check permits all other SQL calls.

Prepare to Use External Security

If you want to use external security, set the following OSF parameters as shown:

```
OSFSECURITY = CHECKUSERID
OSFCONSOLE = userid1
OSFPRODUCT = userid2
```

The variables *userid1* and *userid2* are user IDs at your site that are secured using your security package.

Set Parameters that Allow External Security

Before you can use external security, set values for the following parameters:

EXTSECURITY

Turns on or off external security. Specify ON to turn on external security.

EXTSECCLASS

Specifies the resource class name for your site. The value that you specify depends on the external security manager running on your host system. Set this parameter to one of the following values:

- IBMFAC (if your external security manager is CA Top Secret).
- FAC (if your external security manager is CA ACF2).
- FACILITY (if your external security manager is IBM RACF).

You can create and use another resource class name that is based on the security package you have installed on the target z/OS system. See the chapter specific to your security package for information about selecting a resource class name.

EXTSECPREFIX

Specifies the prefix for all security resource names that CA OPS/MVS defined. Use this value as the first or highest level qualifier for all resource names that your external security uses. The examples showing resource names in this guide use the default OP$MVS.

EXTSECSHOW

Sends trace messages to the OPSLOG. Specify ON to turn on trace messages. The trace messages show event checking information from SAF.

The message has the following format:

```
OPS2109T *CKSAF: <userid> <class> <prefix>.<rname>[.<ext>] <access> RC=<SAF rc>
REASON:<reason>
```

userid

Contains the user ID the SAF resource check uses.

class

Contains the SAF resource class.

prefix

Contains the SAF resource prefix or first-level qualifier of the resource name.

rname

Contains the internally defined name of the security event.

ext

(Optional) Specifies an additional qualifier for the resource class. The presence and content depend on the resource name.

access

Requests resource access. Valid values are either READ or UPDATE.

rc

Contains the return code from the SAF call.

reason

Converts into text the reason code from the SAF call.

EXTSECSQLSUFFIX

Specifies the suffix of the SQL security event resource name. One of two formats for the SQL resource name that CA OPS/MVS uses. This parameter accepts option TBL or TBL.CMD.

EXTSECSQLSUFFIX allows a customization suffix to be appended to the resource name on SQL-related CA OPS/MVS security events.

To specify the SQL resource name, use this keyword in one of two ways:

The parameter has the following format:

Method 1:

```
EXTSECSQLSUFFIX=TBL
```

Utilizes the resource name for each SQL security event:

```
<prefix>.SQL.<table>
```

<prefix>

The value that is specified on EXTSECPREFIX.

<table>

The SQL table name that the security event targets.

Method 2:

```
EXTSECSQLSUFFIX=TBL.CMD
```

Utilizes the resource name for each SQL security event:

```
<prefix>.SQL.<table>.<cmd>
```

<prefix>

The value that is specified on EXTSECPREFIX.

<table>

The SQL table name that the security event targets.

<cmd>

A two-letter abbreviation for the command type that the security event drives. For a list of the two-letter command codes, see the SQL TBL.CMD Names Table. (see page 63)

More information:

SAF Resource Names Table (see page 61)

How SAF Resources Are Defined to Use External Security

Before you start using external security, define your SAF resources and groups. You only define your resources and groups once. The REXX program DEFSAF is distributed with CA OPS/MVS. Use DEFSAF to help you define and maintain access to CA OPS/MVS security resources.

Use the following process to define SAF resources and groups to use external security:

1. Generate - The automation expert uses the utility DEFSAF to generate the external security manager commands to a data set member.
2. Execute - The security administrator executes the external security manager commands in batch using the data set member that the automation expert provided in the first step.

More information:

Generate the SAF Resources with CA Top Secret (see page 28)
Batch Execute External Security Manager Commands to Create the Owner and Profiles (see page 29)
Batch Execute External Security Manager Commands to Create the Owner and Profiles (see page 41)
Batch Execute External Security Manager Commands to Create the Owner and Profiles (see page 55)
Generate the SAF Resources with CA ACF2 (see page 40)
Generate the SAF Resources with RACF (see page 54)

Permit SAF Authority Using DEFSAF

After you generate the SAF definitions and execute the commands, you can permit additional users to use the predefined SAF resource groups. You permit additional users in a generalized way through the PERMIT action of the DEFSAF utility.

Follow these steps:

1. Log in to TSO with the user ID that you can use to run the CA OPS/MVS utility program DEFSAF from data set *opshlq.*CCLXEXEC.
2. Execute DEFSAF from either the ISPF or TSO command line with ACT(PERMIT) to permit SAF authority.
   - To execute DEFSAF with permit authority from an ISPF command line, enter the following commands:

     ```
     ISPF EDIT on member DEFSAF in opshlq.CCLXEXEC
     ```

     ```
     !OI OPSSOF ACT(PERMIT) SAFRO(SOFADMIN)
     ```

   - To execute DEFSAF with permit authority from the TSO command line, enter the following command:

     ```
     TSO OX 'opshlq.CCLXEXEC(DEFSAF)' OPSSOF ACT(PERMIT) SAFRW(SOFADMIN)
     ```

     Note: To use the permit action without using groups, add GROUPS(N) at the end of the command string.

You have permitted user SOFADMIN to have full access to all SOF commands.

More information:

Implementing External Security with CA Top Secret (see page 23)
Resource Tables and Predefined Resources (see page 61)
Implementing External Security with CA ACF2 (see page 35)
Implementing External Security with RACF (see page 49)

Remove SAF Authority Using DEFSAF

You can use the UNPERMIT action of the DEFSAF utility to remove users from the predefined SAF resource groups.

This procedure removes authority from the user ID SOFADMIN and removes access to SOF.

Follow these steps:

1. Log in to TSO with the user ID that you can use to run the CA OPS/MVS utility program DEFSAF from data set ops*hlq.CC*LXEXEC.
2. To remove SAF authority, execute DEFSAF from either the ISPF or TSO command line with ACT(UNPERMIT).
   - To execute DEFSAF from an ISPF command line, enter the following commands:

     ```
     ISPF EDIT on member DEFSAF in opshlq.CCLXEXEC
     ```

     ```
     !OI OPSSOF ACT(UNPERMIT) SAFRO(SOFADMIN)
     ```

   - To execute DEFSAF from the TSO command line, enter the following command:

     ```
     TSO OX 'opshlq.CCLXEXEC(DEFSAF)' OPSSOF ACT(UNPERMIT) SAFRO(SOFADMIN)
     ```

Access to SOF has been removed from the user ID SOFADMIN.

Control Table Access Using SQL Resources or OPSGLOBAL

Control access to specific tables under the SQL resources or specific variables under the OPSGLOBAL facility through the resources that are defined to your external security manager.

Prepare to use SQL or GLOBAL to control access to tables by running DEFSAF to get the resources defined to your external security manager as follows:

- Secure access to the CA OPS/MVS STCTBL by creating the resource name *prefix*.SQL.STCTBL for the SAF check. Permit access to the resource name by running DEFSAF as follows:

  ```
  TSO OX 'opshlq.CCLXEXEC(DEFSAF)' OPSSQL.STCTBL ACT(DEFINE)
  TSO OX 'opshlq.CCLXEXEC(DEFSAF)' OPSSQL.STCTBL ACT(PERMIT) SAFRO(myuid)
  GROUPS(N)
  ```

- Secure access to a global variable named GLOBAL1 by creating the resource name *prefix*.OPSGLOBAL.GLOBAL1 for the SAF check. Run DEFSAF as follows:

  ```
  TSO OX 'opshlq.CCLXEXEC(DEFSAF)' OPSGLOBAL.GLOBAL1 ACT(DEFINE)
  TSO OX 'opshlq.CCLXEXEC(DEFSAF)' OPSGLOBAL.GLOBAL1 ACT(PERMIT) SAFRO(myuid)
  GROUPS(N)
  ```

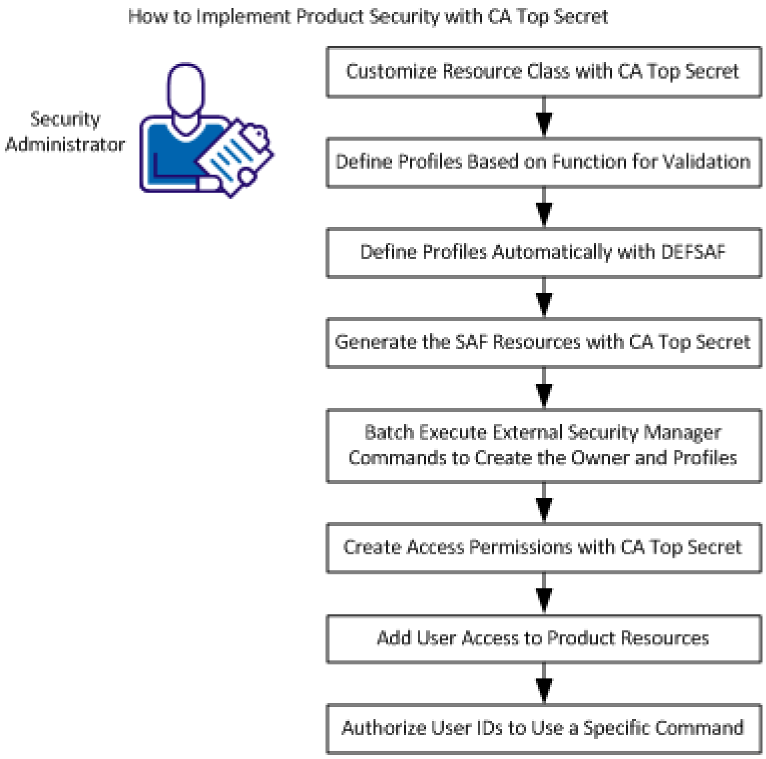
You are ready to use the SQL resources or the OPSGLOBAL facility.

Chapter 3: Implementing External Security with CA Top Secret

This section contains the following topics:

How to Implement External Security with CA Top Secret

As a security administrator in your mainframe environment, your responsibilities include implementing security in CA OPS/MVS with CA Top Secret. You perform the following tasks, which appear in the recommended order. You can perform them in any sequence.


How to Implement Product Security with CA Top Secret
Security Administrator
Customize Resource Class with CA Top Secret
Define Profiles Based on Function for Validation
Define Profiles Automatically with DEFSAF
Generate the SAF Resources with CA Top Secret
Batch Execute External Security Manager Commands to Create the Owner and Profiles
Create Access Permissions with CA Top Secret
Add User Access to Product Resources
Authorize User IDs to Use a Specific Command

- Customize resource class with CA Top Secret (see page 25)
- Define profiles based on function for validation (see page 25)
- Define profiles automatically with DEFSAF (see page 27)
- Generate the SAF resources with CA Top Secret (see page 28)
- Batch execute External Security Manager commands to create the owner and profiles (see page 29)
- Create access permissions with CA Top Secret (see page 30)
- Add user access to CA OPS/MVS resources (see page 32)
- Authorize user IDs to use a specific command (see page 33)

Customize Resource Class with CA Top Secret

The CA OPS/MVS parameter EXTSECCLASS determines the class name that is used to make SAF calls to authorize resources. EXTSECCLASS defaults to IBMFAC, which is a built-in class that is supplied with CA Top Secret.

We recommend using the DEFSAF REXX utility program to create your own resource class. You can have control over both UPDATE and READ access for CA OPS/MVS resources by defining your own resource class to CA Top Secret. After you define the resource class, specify that name on the EXTSECCLASS.

The following steps guide you through customizing CA Top Secret rules under a different resource class name.

Follow these steps:

1. Run the supplied REXX utility DEFSAF.

   DEFSAF creates a TSS resource class for you.

2. Define a new resource class named OPSCLS by executing the following example from the TSO command line:

   ```
   TSO OX 'hql.CCLXEXEC(DEFSAF)' RDT ACT(DEFINE) SAFCL(OPSCLS)
   ```

3. Specify the new resource class name OPSCLS on the EXTSECCLASS parameter before starting CA OPS/MVS. For example:

   ```
   EXTSECCLASS(OPSCLS)
   ```

Your CA Top Secret rules are customized under a different resource class name.

Define Profiles Based on Function for Validation

Profiles in CA Top Secret are used to do the following tasks:

- Group together access requirements, which are common to more than one user. A profile can be used as a grouping mechanism to represent multiple users with identical or similar functional requirements or access authority.
- Permit the resources and accesses to a profile once, and then add the profile to the users.
- Permit access to a CA OPS/MVS resource.

We recommend that you define a department to CA Top Secret to be the owner of all CA OPS/MVS resources. That department can then become a central point of ownership and administrative responsibility for CA OPS/MVS resources. Users in any department who require access to the resources grouped into a profile can then be attached to these profiles.

You can define profiles that are based on function to use for validation processing.

Follow these steps:

1. Define your job function, or roles, using any criteria necessary. For example, create a TSS profile named OPSADMIN for the CA OPS/MVS administrators.

   Note: The REXX program DEFSAF does not define function-based groups.

2. Populate the functional profile with the resources needed for an administrator.
3. Attach this profile to users as their job roles demand.

Your functional profiles are defined.

Define Profiles Automatically with DEFSAF

Profiles let you add and remove users from a single point for validation processing. You can automatically define CA Top Secret profiles with the DEFSAF REXX utility.

By default, the DEFSAF program defines SAF resource names and roles. If you decide not to use roles, specify the parameter GROUPS(N) on the DEFSAF utility. The resource names are still defined but the default group names are not generated.

Follow these steps:

1. Log in to TSO.
2. Access the DEFSAF REXX utility distributed in the *opshlq*.CCLXEXEC data set.
3. Run DEFSAF from a CA Top Secret user ID that has sufficient privileges to create and modify users in the department that was created for CA OPS/MVS.

   Member DEFTSS is generated and contains the basic CA Top Secret commands for securing the processing environment under CA Top Secret.
4. Review the example definitions in the member DEFTSS to verify that they meet the security requirements of your site.
5. (Optional) Modify the example definitions by running DEFSAF again using different keywords to generate the definitions to meet the security requirements of your site exactly.

   See the comments in DEFSAF for information on using keywords to customize the definitions.
6. Use the tailored definitions as batch input to CA Top Secret.

   Note: Member BATTSS in *opshlq*.OPS.CNTL is provided as a sample that allows submission of the member DEFTSS for batch execution.

Example: DEFSAF Execution

These examples generate the *opshlq*.OPSS.DEFSAF(DEFTSS) file containing all of the required resource definitions to begin using CA OPS/MVS external security with CA Top Secret.

- This example uses the default profile:

  ```
  TSO OX ‘opshlq.OPSnnn.CCLXEXEC(DEFSAF)’ ALL SEC(TSS) ACT(DEFINE) BATCH(Y)
  ```

- This example does not use the default profile:

  ```
  TSO OX ‘opshlq.OPSnnn.CCLXEXEC(DEFSAF)’ ALL SEC(TSS) ACT(DEFINE) GROUPS(N)
  BATCH(Y)
  ```

  ```
  OPS0996I DEFSAF Security product is TSS.
  OPS0996I DEFSAF CA OPS/MVS subsystem OPSS is active.
  OPS0996I DEFSAF 'OPSHLQ.OPSS.DEFSAF(DEFTSS)' has been generated.
  ***
  ```

Note: For a complete example of DEFSAF execution, see the data set member *opshlq*.OPSS.DEFSAF(DEFTSS).

Generate the SAF Resources with CA Top Secret

You generate SAF resources to protect CA OPS/MVS commands and features. The automation expert generates the SAF resources.

Follow these steps:

1. Temporarily set your external security to off by issuing the following command:

   ```
   EXTSECURITY=OFF
   ```

2. Log in to a user ID that you can use to run the CA OPS/MVS utility program DEFSAF from data set *opshlq*.CCLXEXEC.

   Note: We recommend executing DEFSAF while your CA OPS/MVS subsystem is active. You can then retrieve the default values for EXTSECCLASS and EXTSECPREFIX from the running subsystem.

3. Run the DEFSAF REXX utility distributed in the *opshlq*.CCLXEXEC data set.

   This utility defines all the resources and groups for using external security.

   a. Log in to TSO with the user ID logged on in Step 2.

   b. Execute DEFSAF from either the ISPF or TSO command line.

      - Execute DEFSAF from an ISPF command line by entering the commands:

        ```
        ISPF EDIT on member DEFSAF in opshlq.CCLXEXEC
        !OI ALL ACT(DEFINE) BATCH(Y)
        ```

      - Execute DEFSAF from the TSO command line by entering the command:

        ```
        TSO OX 'opshlq.CCLXEXEC(DEFSAF)' ALL ACT(DEFINE)BATCH(Y)
        ```

        Executing the command string creates a dynamically allocated data set named *tsoid*.OPS*x*.DEFSAF. The default value for *x* is the subsystem ID of S.

      The created member DEFTSS contains the external security manager commands to define the resources and groups for CA OPS/MVS. Pass this member to a security administrator on the target host who has the CA Top Secret authority necessary to execute the external security manager commands in the member.

The SAF resources are generated.

Batch Execute External Security Manager Commands to Create the Owner and Profiles

Executing external security manager commands in batch creates the basic security product owner and profiles. Batch execution requires a user ID with sufficientCA Top Secret authority. The user requires authority to execute all of the external security manager commands contained in the DEFTSS and PERTSS members. The CA OPS/MVS administrator generates these members.

Follow these steps:

1. Log in to a user ID that has sufficient authority with your z/OS security package to define resources and groups.
2. Locate the DEFTSS member containing the commands your CA OPS/MVS Administrator generated.
3. Run a batch job to execute the external security manager member DEFTSS. The SAMPLE JCL distributed with CA OPS/MVS in data set *opshlq*.CCLXCNTL contains the BATTSS member.

   Note: You can customize the sample batch BATTSS to execute the external security manager commands member DEFTSS.

You have executed in batch the external security manager commands.

More information:

Resource Tables and Predefined Resources (see page 61)

Create Access Permissions with CA Top Secret

Connecting or permitting site-defined user names authorizes them to the groups or profiles that either you or DEFSAF defined. Execute the REXX DEFSAF utility to define the resource names and groups. Execute DEFSAF again to add user names to those groups.

Use CA Top Secret to create access permissions. You can perform these steps in any order.

Note: The DEFSAF commands provided in the following steps default the BATCH option to YES. The YES value generates member PERTSS in opqhlq.OPSx.DEFSAF. Send this member to your security administrator to run the CA Top Secret commands in the member from an authorized user ID. If you specify BATCH(N) on DEFSAF, the commands are issued directly. In this mode, the running user ID requires the CA Top Secret authorities to execute successfully.

Follow these steps:

1. Execute the following command to provide the OPSUSER with READ access to all CA OPS/MVS protected resources:

   ```
   ISPF EDIT on member DEFSAF in opshlq.CCLXEXEC
   !OI ALL ACT(PERMIT) SAFRO(OPSUSER)
   ```

   or

   ```
   !OX 'opshql.CCLXEXEC(DEFSAF)' ALL ACT(PERMIT) SAFRO(OPSUSER)
   ```

2. Execute the following command to provide the OPSOPER with UPDATE access to all CA OPS/MVS protected resources:

   ```
   ISPF EDIT on member DEFSAF in opshlq.CCLXEXEC
   !OI OPSAOF ACT(PERMIT) SAFRW(OPSOPER)
   ```

   or

   ```
   !OX 'opshlq.CCLXEXEC(DEFSAF)' OPSAOF ACT(PERMIT) SAFRW(OPSOPER)
   ```

3. Execute the following command to provide OPSSQL1 with READ access to all SQL commands:

   ```
   ISPF EDIT on member DEFSAF in opshlq.CCLXEXEC
   !OI OPSSQL ACT(PERMIT) SAFRO(OPSSQL1)
   ```

   or

   ```
   !OX ‘opshlq.CCLXEXEC(DEFSAF)’ OPSSQL ACT(PERMIT) SAFRO(OPSSQL1)
   ```

   Note: The BATCH(Y) option generates the member in *opqhlq*.OPS*x*.DEFSAF named PERTSS. Send this member to your security administrator to run the CA Top Secret commands in the member from an authorized user ID. If the user ID where you run the DEFSAF command has sufficient authority, specify BATCH(N) and then issue the commands directly from DEFSAF.

4. Execute OPSSQL1 either with or without GROUPS as follows:

- Execute OPSSQL1 with GROUPS(Y) and with UPDATE access to all aspects of SQL commands:

  ```
  ISPF EDIT on member DEFSAF in opshlq.CCLXEXEC
  ```

  ```
  !OI OPSSQL ACT(PERMIT) SAFRW(OPSSQL1) GROUPS(Y)
  ```

  or

  ```
  !OX ‘opshlq.CCLXEXEC(DEFSAF)’ OPSSQL ACT(PERMIT) SAFRW(OPSSQL1) GROUPS(Y)
  ```

- Execute OPSSQL1 with GROUPS(N) to permit the same access without using groups:

  ```
  TSO OX 'opshlq.CCLXEXEC(DEFSAF)' OPSSQL ACT(PERMIT) SAFRO(OPSSQL1) GROUPS(N)
  ```

  ```
  ISPF EDIT on member DEFSAF in opshlq.CCLXEXEC
  !OI OPSSQL ACT(PERMIT) SAFRO(OPSSQL1) GROUPS(N)
  ```

  or

  ```
  !OX ‘opshlq.CCLXEXEC(DEFSAF)’ OPSSQL ACT(PERMIT) SAFRO(OPSSQL1) GROUPS(N)
  ```

5. If you did not use DEFSAF, provide OPSSQL1 with READ access to all SQL commands by issuing the following sample CA Top Secret PERMIT command:

   ```
   TSS PERMIT(OPSSQL1) FAC(OP$MVS.SQL.) ACCESS(READ)
   ```

You have created your access permissions with CA Top Secret.

Add User Access to Product Resources

As the administrator, you need two users added to your CA Top Secret external security package with the following access privileges:

- User1 with READ access to address AOF (permitting safe verbs such as LIST)
- User2 with UPDATE access to address AOF (permitting all verbs)

Use one of the following methods to implement the needed access permissions to address AOF.

- Run the supplied DEFSAF REXX utility either with or without GROUPS as follows:
  - Execute the supplied REXX program DEFSAF without GROUPS(N):

    ```
    TSO OX ‘opshlq.CCLXEXEC(DEFSAF)’ OPSAOF ACT(PERMIT) SAFRO(user1)
    TSO OX ‘opshlq.CCLXEXEC(DEFSAF)’ OPSAOF ACT(PERMIT) SAFRW(user2)
    ```

  - Execute the supplied REXX program DEFSAF with GROUPS(N) to permit the access without using groups:

    ```
    TSO OX 'opshlq.CCLXEXEC(DEFSAF)' OPSAOF ACT(PERMIT) SAFRO(user1) GROUPS(N)
    TSO OX 'opshlq.CCLXEXEC(DEFSAF)' OPSAOF ACT(PERMIT) SAFRW(user2) GROUPS(N)
    ```

  Note: Be sure that you ran DEFSAF earlier in batch mode (BATCH=Y) to create the basic CA Top Secret owner and profiles.

- Issue the following CA Top Secret commands from TSO or from JCL:

  ```
  TSS ADDTO(user1) PROFILE(OPSAOFR)
  TSS ADDTO(user2) PROFILE(OPSAOF)
  ```

  Note: Be sure that you ran DEFSAF earlier in batch mode (BATCH=Y) to create the basic CA Top Secret owner and profiles.

- Issue the following CA Top Secret commands to permit a rule for OP$MVS.OPSAOF to accessor IDs *user1* and *user2*. *User1* is given READ access and *user2* is given UPDATE access to the resource:

  ```
  TSS PERMIT(user1) IBMFAC(OP$MVS.OPSAOF) ACCESS(READ)
  TSS PERMIT(user2) IBMFAC(OP$MVS.OPSAOF) ACCESS(UPDATE)
  ```

Authorize User IDs to Use a Specific Command

Group names are derived from the facility names that are associated with the security event that the group name protects. The resources of some facilities have READ access, other facilities have UPDATE access, and other facilities have both READ and UPDATE verbs. Therefore, the group names are encoded following this pattern:

Facility	Group Name
READ access only	The group name is the same as the facility name or derived from the facility name.
UPDATE access only	The group name is the same as the facility name or derived from the facility name.
READ and UPDATE access	The group name for update access is derived from the facility name for UPDATE. The group name for read access is derived from the facility name with an appended R.

You can authorize a specific user ID or group of user IDs to use a particular CA OPS/MVS command either manually or using DEFSAF. The first procedure explains how to add authorizations manually. The second procedure explains how to execute the CA OPS/MVS REXX program DEFSAF.

Follow these steps:

1. Look up the command or function in Commands and Functions that Generate External Security. In the row where you find your command or function, make the following notes:
   - The associated Facility name
   - Whether your command includes verbs that generate either Read or Update access.
2. Find the corresponding row that matches both your Facility name and access value. In that row, make a note of the CA Top Secret profile name.
3. Permit the user ID or profile you want to authorize to the CA Top Secret resource.
4. Issue this CA Top Secret command to add your user IDs to the CA Top Secret profile name:

   ```
   TSS ADDTO("userid") PROFILE(profile)
   ```

   You have manually added your authorizations.

Follow these steps:

1. Look up the command or function in Commands and Functions that Generate External Security. In the row where you find your command or function, make the following notes:
   - Note the associated Facility name.
   - Note whether your command includes verbs that generate either READ or UPDATE access.
2. Find your access value and do the following tasks:
   - If your access value is READ, then execute DEFSAF as follows:

     ```
     DEFSAF <facility> ACT(PERMIT) SAFRO(<userid>)
     ```

   - If your access value is UPDATE, then execute DEFSAF as follows:

     ```
     DEFSAF <facility> ACT(PERMIT) SAFRW(<userid>)
     ```

     <facility>

     Specify the facility name.

     <userid>

     Specify the user ID or group you want to authorize.

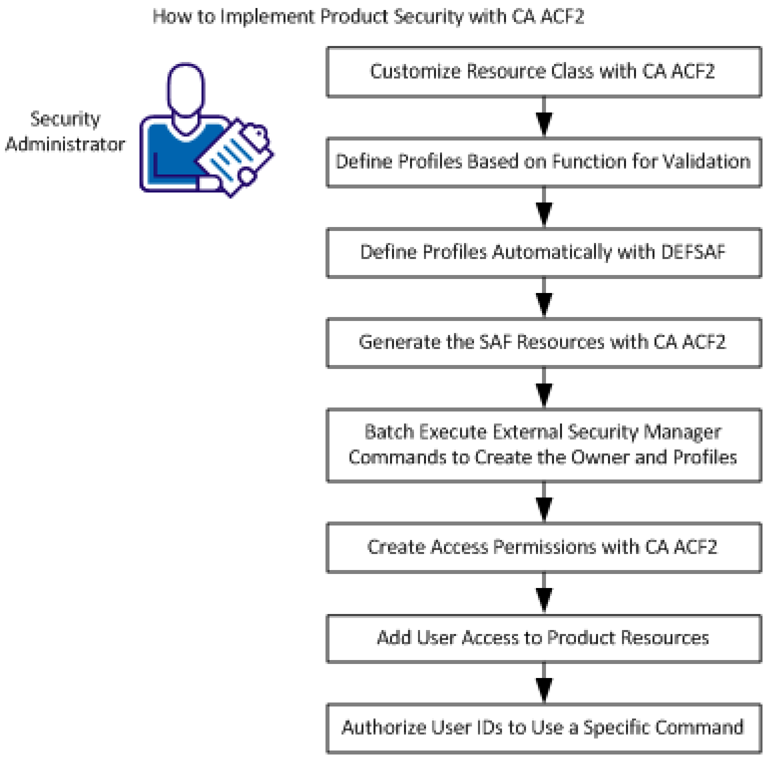
   Your user IDs are authorized.

Chapter 4: Implementing External Security with CA ACF2

This section contains the following topics:

How to Implement External Security with CA ACF2

As a security administrator in your mainframe environment, your responsibilities include implementing security in CA OPS/MVS with CA ACF2. You perform the following tasks, which appear in the recommended order. You can perform them in any sequence:


How to Implement Product Security with CA ACF2
Security Administrator
Customize Resource Class with CA ACF2
Define Profiles Based on Function for Validation
Define Profiles Automatically with DEFSAF
Generate the SAF Resources with CA ACF2
Batch Execute External Security Manager Commands to Create the Owner and Profiles
Create Access Permissions with CA ACF2
Add User Access to Product Resources
Authorize User IDs to Use a Specific Command

- Customize resource class with CA ACF2 (see page 37)
- Define roles based on function for validation (see page 37)
- Define roles automatically with DEFSAF (see page 38)
- Generate the SAF resources with CA ACF2 (see page 40)
- Batch execute External Security Manager commands to create the owner and profiles (see page 41)
- Create access permissions with CA ACF2 (see page 42)
- Add user access to CA OPS/MVS resources (see page 44)
- Authorize user IDs to use a specific command (see page 45)

Customize Resource Class with CA ACF2

The CA OPS/MVS parameter EXTSECCLASS determines the class name that is used to make SAF calls to authorize resources. EXTSECCLASS defaults to the three-character type code FAC. FAC maps to the FACILITY resource class supplied with CA ACF2. With CA ACF2, you can create a GSO CLASMAP record to map to a different three-character type code to address your unique site requirements.

The following steps guide you through customizing CA ACF2 rules under a different resource class name.

Follow these steps:

1. Issue the ACF command SHOW CLASMAP.

   SHOW CLASMAP verifies the three-character type CA ACF2 uses.

   Note: For more information, see the *CA ACF2 for z/OS Administrator Guide*.

2. Specify this three-character resource type code on the EXTSECCLASS parameter before starting CA OPS/MVS.

You have finished customizing CA ACF2 rules under a different resource class name.

Define Roles Based on Function for Validation

The purpose of roles in CA ACF2 is for validation processing. With CA ACF2, a role is a group of users or a group of groups. You define a group name and then add your groups of sources, resources, or roles to that group one time. You then reuse the group name to specify that group.

Use your defined role as a grouping mechanism to represent multiple users with identical or similar functional requirements or access authority. Adding one group entry to access lists rather than many user IDs simplifies both access and maintenance.

You can define functional roles to use for validation processing.

Follow these steps:

1. Define your job function, or roles, using any criteria necessary. For example, create a functional role named OPSADMIN for the CA OPS/MVS administrators.

   Note: The REXX program DEFSAF does not define function-based groups.

2. Assign users to the function roles you defined.
3. Define individual users to a group.
4. Assign to that group a role group name in an XREF role group (X-ROL) record.

   The CA ACF2 X-ROL record can specify either a list users or a list of groups. Use the include and exclude parameters plus masking to include many users with fewer statements.

5. Use the resource rules to specify the role you want to access the resource. You can specify one of the following resources:

   - A role that is a group of users.
   - A role that is a group of groups.
   - An individual user.

Your functional roles are defined.

Note: For more information about roles and XREF, see the *CA ACF2 for z/OS Administrator Guide*.

Define Roles Automatically with DEFSAF

Roles let you add and remove users from a single point for validation processing. You can automatically define CA ACF2 roles with the DEFSAF REXX utility.

By default, the DEFSAF program defines SAF resource names and roles. If you decide not to use roles, specify the parameter GROUPS(N) on the DEFSAF utility. The resource names are still defined but the default group names are not generated.

Follow these steps:

1. Log in to TSO.
2. Access the DEFSAF REXX utility distributed in the *opshlq*.CCLXEXEC data set.

3. Run DEFSAF from an CA ACF2 logon ID that has the SECURITY permission.

   The following actions occur:

   - Roles are created based on the CA OPS/MVS facility names. Roles are named appropriately to match the CA OPS/MVS facility they secure.
   - Member DEFACF2 is generated and contains the basic ACF2 commands for securing the processing environment under CA ACF2.

4. Review and modify the example definitions to meet the security requirements of your site.

5. Use the tailored definitions as batch input to CA ACF2.

   Note: Member BATACF2 in *opshlq*.OPS.CNTL is provided as a sample. For more information about executing CA ACF2 commands in batch, see the *CA ACF2 for z/OS Reports and Utilities Guide*.

Example: DEFSAF Execution

This example generates the *opshlq*.OPSS.DEFSAF(DEFACF2) file containing all of the required resource definitions to begin using CA OPS/MVS external security with CA ACF2.

```
TSO OX 'opshlq.opsnnn.CCLXEXEC(DEFSAF)' 'ALL SEC(ACF2) ACT(DEFINE) BATCH(Y)'
```

```
OPS0996I #DEFSAF Security product is ACF2.
OPS0996I #DEFSAF CA OPS/MVS subsystem OPSS is active.
OPS0996I #DEFSAF 'OPSHLQ.OPSS.DEFSAF(DEFACF2)' has been generated.
***
```

Note: For a complete example of DEFSAF execution, see the contents of data set member OPSHLQ.OPSS.DEFSAF(DEFACF2).

More information:

Resource Tables and Predefined Resources (see page 61)

Generate the SAF Resources with CA ACF2

You generate SAF resources to protect CA OPS/MVS commands and features. The automation expert generates the SAF resources.

Follow these steps:

1. Temporarily set your external security to off by issuing the following command:

   ```
   EXTSECURITY=OFF
   ```

2. Log in to a user ID that you can use to run the CA OPS/MVS utility program DEFSAF from data set *opshlq*.CCLXEXEC.

   Note: We recommend executing DEFSAF while your CA OPS/MVS subsystem is active. You can then retrieve the default values for EXTSECCLASS and EXTSECPREFIX from the running subsystem.

3. Run the DEFSAF REXX utility distributed in the *opshlq*.CCLXEXEC data set.

   This utility defines all the resources and groups for using external security.

   a. Log in to TSO with the user ID logged on in Step 2.

   b. Execute DEFSAF from either the ISPF or TSO command line.

      - Execute DEFSAF from an ISPF command line by entering the commands:

        ```
        ISPF EDIT on member DEFSAF in opshlq.CCLXEXEC
        !OI ALL ACT(DEFINE) BATCH(Y)
        ```

      - Execute DEFSAF from the TSO command line by entering the command:

        ```
        TSO OX 'opshlq.CCLXEXEC(DEFSAF)' ALL ACT(DEFINE)BATCH(Y)
        ```

        Executing the command string creates a dynamically allocated data set named *tsoid*.OPS*x*.DEFSAF. The default value for *x* is the subsystem ID of S.

      The created member DEFACF2 contains the CA ACF2 commands to define the resources and groups for CA OPS/MVS. Pass this member to a security administrator on the target host who has the CA ACF2 authority to execute the external security manager commands in the member.

The SAF resources are generated.

Batch Execute External Security Manager Commands to Create the Owner and Profiles

Executing external security manager commands in batch creates the basic security product owner and profiles. Batch execution requires a user ID with sufficient CA ACF2 authority. The user requires authority to execute all of the external security manager commands contained in the DEFACF2 members. The CA OPS/MVS administrator generates these members.

Follow these steps:

1. Log in to a user ID that has sufficient authority with your z/OS security package to define resources and groups.
2. Locate the DEFACF2 member containing the external security manager your CA OPS/MVS Administrator generated.
3. Run a batch job to execute the external security manager member. The SAMPLE JCL distributed with CA OPS/MVS in data set *opshlq*.CCLXCNTL contains the BATACF2 member.

   Note: You can customize the sample batch jobs named BATACF2 to execute the external security manager commands member.

You have executed in batch the external security manager commands.

More information:

Resource Tables and Predefined Resources (see page 61)

Create Access Permissions with CA ACF2

Connecting or permitting site-defined user names authorizes them to the groups or profiles that either you or DEFSAF defined. Execute the REXX DEFSAF utility to define the resource names and groups. Execute DEFSAF again to add user names to those groups.

Use CA ACF2 to create access permissions. You can perform these steps in any order.

Note: The DEFSAF commands provided in the following steps default the BATCH option to YES, which generates member PERACF2 in *opqhlq*.OPSx.DEFSAF. Send this member to your security administrator to run the CA ACF2 commands in the member from an authorized user ID. If you specify BATCH(N) on DEFSAF, the commands are issued directly. In this mode, the running user ID requires the CA ACF2 authorities to execute successfully.

Follow these steps:

1. Execute the following command to provide the OPSUSER with READ access to all CA OPS/MVS protected resources:

   ```
   ISPF EDIT on member DEFSAF in opshlq.CCLXEXEC
   !OI ALL ACT(PERMIT) SAFRO(OPSUSER)
   ```

   or

   ```
   !OX ‘opshlq.CCLXEXEC(DEFSAF)’ ALL ACT(PERMIT) SAFRO(OPSUSER)
   ```

2. Execute the following command to provide the OPSOPER with UPDATE access to all CA OPS/MVS protected resources:

   ```
   ISPF EDIT on member DEFSAF in opshlq.CCLXEXEC
   !OI OPSAOF ACT(PERMIT) SAFRW(OPSOPER)
   ```

   or

   ```
   !OX ‘opshlq.CCLXEXEC(DEFSAF)’ OPSAOF ACT(PERMIT) SAFRW(OPSOPER)
   ```

3. Execute the following command to provide OPSSQL1 with READ access to all SQL commands:

   ```
   ISPF EDIT on member DEFSAF in opshlq.CCLXEXEC
   !OI OPSSQL ACT(PERMIT) SAFRO(OPSSQL1)
   ```

   or

   ```
   !OX ‘opshlq.CCLXEXEC(DEFSAF)’ OPSSQL ACT(PERMIT) SAFRO(OPSSQL1)
   ```

   Note: The BATCH(Y) option generates the member in *opqhlq*.OPS*x*.DEFSAF named PERACF2. Send this member to your security administrator to run the CA ACF2 commands in the member from an authorized user ID. If the user ID where you run the DEFSAF command has sufficient authority, specify BATCH(N) and then issue the commands directly from DEFSAF.

4. Execute OPSSQL1 either with or without GROUPS as follows:

- Execute OPSSQL1 with GROUPS(Y) and with UPDATE access to all aspects of SQL commands:

  ```
  ISPF EDIT on member DEFSAF in opshlq.CCLXEXEC
  !OI OPSSQL ACT(PERMIT) SAFRW(OPSSQL1) GROUPS(Y)
  ```

  or

  ```
  !OX ‘opshlq.CCLXEXEC(DEFSAF)’ OPSSQL ACT(PERMIT) SAFRW(OPSSQL1) GROUPS(Y)
  ```

- Execute OPSSQL1 with GROUPS(N) to permit the same access without using groups:

  ```
  TSO OX 'opshlq.CCLXEXEC(DEFSAF)' OPSSQL ACT(PERMIT) SAFRO(OPSSQL1) GROUPS(N)
  ISPF EDIT on member DEFSAF in opshlq.CCLXEXEC
  !OI OPSSQL ACT(PERMIT) SAFRO(OPSSQL1) GROUPS(N)
  ```

  or

  ```
  !OX ‘opshlq.CCLXEXEC(DEFSAF)’ OPSSQL ACT(PERMIT) SAFRO(OPSSQL1) GROUPS(N)
  ```

5. If you did not use DEFSAF, provide OPSSQL1 with READ access to all SQL commands by issuing the following sample CA ACF2 commands:

   ```
   SET RES(FAC)
    RECKEY OP$MVS ADD(SQL.- USER(OPSSQL1) SERVICE(READ) ALLOW)
    STORE
   ```

You have created your access permissions with CA ACF2.

Add User Access to Product Resources

As the Administrator, you need two users added to your CA ACF2 external security package with the following access privileges:

- Provide user1 with READ access to address AOF (permitting safe verbs such as LIST).
- Provide user2 with UPDATE access to address AOF (permitting all verbs).

Use one of the following methods to implement the needed access permissions to address AOF.

- Run the supplied DEFSAF REXX utility either with or without GROUPS as follows:
  - Execute the supplied REXX program DEFSAF without GROUPS(N):

    ```
    TSO OX 'opshlq.CCLXEXEC(DEFSAF)’ OPSAOF ACT(PERMIT) SAFRO(user1)
    TSO OX 'opshlq.CCLXEXEC(DEFSAF)’ OPSAOF ACT(PERMIT) SAFRW(user2)
    ```

  - Execute DEFSAF with GROUPS(N) to permit the access without using groups:

    ```
    TSO OX 'opshlq.CCLXEXEC(DEFSAF)' OPSAOF ACT(PERMIT) SAFRO(user1) GROUPS(N)
    TSO OX 'opshlq.CCLXEXEC(DEFSAF)' OPSAOF ACT(PERMIT) SAFRW(user2) GROUPS(N)
    ```

  Note: Be sure that you ran DEFSAF earlier in batch mode (BATCH=Y) to create the basic CA ACF2 groups.

- Issue the following CA ACF2 commands from TSO or from JCL:

  ```
  TSO ACF
   SET XREF(ROL)
   CHANGE OPSAOFR INCLUDE(user1) ROLE ADD
   CHANGE OPSAOF INCLUDE(user2) ROLE ADD
   F ACF2,NEWXREF,TYPE(ROL)
  ```

  You can use masking to specify many users.

  Note: Be sure that you ran DEFSAF earlier in batch mode (BATCH=Y) to create the basic CA ACF2 groups.

- Issue the following CA ACF2 commands to add a rule line to permit user1 access to the resource OP$MVS.OPSAOF:

  ```
  TSO ACF
  SET RESOURCE(FAC)
  RECKEY OP$MVS ADD(OPSAOF.- USER(user1) ALLOW SERVICE(READ))
  RECKEY OP$MVS ADD(OPSAOF.- USER(user2) ALLOW SERVICE(READ,UPDATE))
  F ACF2,REBUILD(FAC)
  ```

Authorize User IDs to Use a Specific Command

Group names are derived from the facility names that are associated with the security event the group name protects. The resources of some facilities have READ access, other facilities have UPDATE access, and other facilities have both READ and UPDATE verbs. Therefore, the group names are encoded following this pattern:

Facility	Group Name
READ access only	The group name is the same as the facility name or derived from the facility name.
UPDATE access only	The group name is the same as the facility name or derived from the facility name.
READ and UPDATE access	The group name for update access is derived from the facility name for UPDATE. The group name for read access is derived from the facility name with an appended R.

You can authorize a specific user ID or group of user IDs to use a particular CA OPS/MVS command either manually or using DEFSAF. The first procedure explains how to add authorizations manually. The second procedure explains how to execute the CA OPS/MVS REXX program DEFSAF.

Follow these steps:

1. Look up the command or function in Commands and Functions that Generate External Security. In the row where you find your command or function, make the following notes:
   - Note the associated Facility name.
   - Note whether your command includes verbs that generate either READ or UPDATE access.
2. Find the corresponding row that matches both your Facility name and access value. In that row, make a note of the role name.

   Note: If you changed the role names from the DEFSAF generated values, be sure to specify those actual role names.
3. Permit the user ID or role name you want to authorize to the CA ACF2 role.
4. Issue this CA ACF2 command to add your user IDS to the CA ACF2role:

   ```
   ACF
     SET XREF(ROL)
     CHANGE ACF2_role_name INCLUDE(userid) ROLE ADD
   ```

   You have manually added your authorizations.

   Note: Add additional role_names instead of individual users when the role record is for a group of groups.

Follow these steps:

1. Look up the command or function in Commands and Functions that Generate External Security. In the row where you find your command or function, make the following notes:

   - Note the associated Facility name.
   - Note whether your command includes verbs that generate either READ or UPDATE access.

2. Find your access value and do the following tasks:

   - If your access value is READ, then execute DEFSAF as follows:

     ```
     DEFSAF <facility> ACT(PERMIT) SAFRO(<userid>)
     ```

   - If your access value is UPDATE, then execute DEFSAF as follows:

     ```
     DEFSAF <facility> ACT(PERMIT) SAFRW(<userid>)
     ```

     <facility>

     Specify the facility name.

     <userid>

     Specify the user ID or role name you want to authorize.

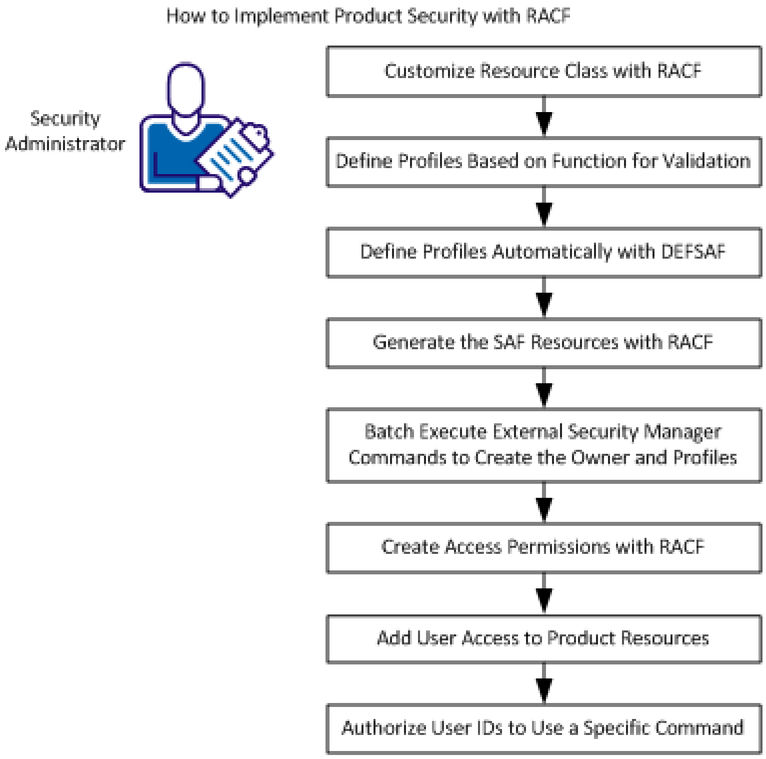
   Your user IDs are authorized.

Chapter 5: Implementing External Security with RACF

This section contains the following topics:

How to Implement External Security with RACF

As a security administrator in your mainframe environment, your responsibilities include implementing security in CA OPS/MVS with RACF. You perform the following tasks, which appear in the recommended order. You can perform them in any sequence.


How to Implement Product Security with RACF
Security Administrator
Customize Resource Class with RACF
Define Profiles Based on Function for Validation
Define Profiles Automatically with DEFSAF
Generate the SAF Resources with RACF
Batch Execute External Security Manager Commands to Create the Owner and Profiles
Create Access Permissions with RACF
Add User Access to Product Resources
Authorize User IDs to Use a Specific Command

- Customize resource classes with RACF (see page 51)
- Define groups based on function for validation (see page 52)
- Define groups automatically with DEFSAF (see page 52)
- Generate the SAF resources with RACF (see page 54)
- Batch execute External Security Manager commands to create the owner and profiles (see page 55)
- Create access permissions with RACF (see page 56)
- Add user access to product resources (see page 58)
- Authorize user IDs to use a specific command (see page 59)

Customize Resource Classes with RACF

The CA OPS/MVS parameter EXTSECCLASS determines the class name that is used to make SAF calls to authorize resources. EXTSECCLASS defaults to FACILITY, which is a built-in class that is supplied with RACF. You can separate the RACF resource profiles under a resource class name for CA OPS/MVS.

Note: Third-party products use the FACILITY class when they do not need to create a user class.

The following steps guide you through customizing RACF rules under a different resource class name.

Follow these steps:

1. Access the local RACF class descriptor table (CDT).
2. Add new resource classes to the CDT.

   The CDT contains two parts:

   - A system-defined part
   - An installation-defined part named ICHRRCDE
3. Add new resource classes to ICHRRCDE using one of the following methods:

   - Code the ICHERCDE macro
   - Dynamically, using the CDT class definition process

   By default, all of the resources you defined to RACF for CA OPS/MVS are added to the IBM built-in FACILITY class.
4. Define your resource class dynamically by running DEFSAF with the following arguments:

   ```
   CDT ACT(DEFINE) SAFCL(yourname)
   ```

   The resource class is created.

   Note: The REXX program DEFSAF is distributed with CA OPS/MVS to help define and maintain access to CA OPS/MVS security resources.
5. Specify the new resource class name on the EXTSECCLASS parameter before starting CA OPS/MVS.

The RACF rules are customized under a different resource class name.

Note: See the following IBM guides:

- *SA22-7681 Security Server RACF System Programmer's Guide* for instructions on updating ICHRRCDE
- *SA22-7683 Security Server RACF Security Administrator's Guide* for instructions on adding a dynamic CDT class

Define Groups Based on Function for Validation

The primary purpose of a group in RACF is for validation processing. Use your defined group to represent multiple users with identical or similar functional requirements or access authority. Adding one group entry to access lists rather than many user IDs simplifies both access and maintenance.

You can define and use functional groups to describe job functions or groups.

Follow these steps:

1. Define your job function, or groups, using any criteria necessary. For example, create a RACF group named OPSADMIN for the CA OPS/MVS administrators.

   Note: The REXX program DEFSAF does not define function-based groups.

2. Populate the functional group with all the facilities needed for an administrator.
3. Connect or remove users from this group as their job roles demand.

   The users acquire or lose the authority of the group without needing to refresh the profile.

   Note: A user with CONNECT group authority for a specific group can use the CONNECT and REMOVE commands to change the members of that group. This capability eliminates using the PERMIT command to change the access list of the affected profiles.

Your functional groups are defined.

Define Groups Automatically with DEFSAF

Groups let you add and remove users from a single point for validation processing. You can automatically define RACF administrative groups with the DEFSAF REXX utility.

By default, the DEFSAF program defines SAF resource names and groups and adds them to the RACF database. If you decide not to use RACF groups, specify the parameter GROUPS(N) on the DEFSAF utility. The resource names are still defined but the default group names are not generated.

Follow these steps:

1. Log in to TSO.
2. Access the DEFSAF REXX utility that is distributed with CA OPS/MVS in the *opshlq*.CCLXEXEC data set.

3. Execute DEFSAF from a user ID that has the RACF SPECIAL attribute.

   The following actions occur:

   - Creates groups that are based upon the CA OPS/MVS facility names.
   - Names the created groups appropriately to match the CA OPS/MVS facility they secure.

     Generates member DEFRACF containing the basic RACF commands that are used to secure the CA OPS/MVS processing environment under RACF.

4. Review and modify the example definitions to meet the security requirements of your site.

5. Use the tailored the definitions as batch input to RACF.

   Note: Member BATRACF in *opshlq*.OPS.CNTL is provided as a sample. For more information about executing RACF commands in batch, see the Security Server RACF Command Language Reference (SA22-7687-15).

Example: DEFSAF Execution

These examples generate the *opshlq*.OPSS.DEFSAF(DEFRACF) file containing all of the required resource definitions to begin using CA OPS/MVS external security with RACF.

- This example uses the default group:

  ```
  TSO OX 'opshlq.opsnnn.CCLXEXEC(DEFSAF)' ALL SEC(RACF) ACT(DEFINE) BATCH(Y)
  ```

- This example does not use the default group:

  ```
  TSO OX 'opshlq.opsnnn.CCLXEXEC(DEFSAF)' ALL SEC(RACF) ACT(DEFINE) GROUPS(N)
  BATCH(Y)
  ```

  ```
  OPS0996I #DEFSAF Security product is RACF.
  OPS0996I #DEFSAF 'OPSHLQ.OPSS.DEFSAF(DEFRACF)' has been generated.
  ***
  ```

Note: For a complete example of DEFSAF execution with RACF security, review the contents of data set member OPSHLQ.OPSS.DEFSAF(DEFRACF).

More information:

Commands and Functions that Generate External Security Events (see page 64)

Generate the SAF Resources with RACF

You generate SAF resources to protect CA OPS/MVS commands and features. The automation expert generates the SAF resources.

Follow these steps:

1. Temporarily set your external security to off by issuing the following command:

   ```
   EXTSECURITY=OFF
   ```

2. Log in to a user ID that you can use to run the CA OPS/MVS utility program DEFSAF from data set *opshlq*.CCLXEXEC.

   Note: We recommend executing DEFSAF while your CA OPS/MVS subsystem is active. You can then retrieve the default values for EXTSECCLASS and EXTSECPREFIX from the running subsystem.

3. Run the DEFSAF REXX utility distributed in the *opshlq*.CCLXEXEC data set.

   This utility defines all the resources and groups for using external security.

   a. Log in to TSO with the user ID logged on in Step 2.

   b. Execute DEFSAF from either the ISPF or TSO command line.

      - Execute DEFSAF from an ISPF command line by entering the commands:

        ```
        ISPF EDIT on member DEFSAF in opshlq.CCLXEXEC
        !OI ALL ACT(DEFINE) BATCH(Y)
        ```

      - Execute DEFSAF from the TSO command line by entering the command:

        ```
        TSO OX 'opshlq.CCLXEXEC(DEFSAF)' ALL ACT(DEFINE)BATCH(Y)
        ```

        Executing the command string creates a dynamically allocated data set named *tsoid*.OPS*x*.DEFSAF. The default value for *x* is the subsystem ID of S.

      The created member DEFRACF contains the IBM RACF commands to define the resources and groups for CA OPS/MVS. Pass this member to a security administrator on the target host who has the RACF authority to execute the external security manager commands in the member.

The SAF resources are generated.

Batch Execute External Security Manager Commands to Create the Owner and Profiles

Executing external security manager commands in batch creates the basic security product owner and profiles. Batch execution requires a user ID with sufficient RACF authority. The user requires authority to execute all of the external security manager commands contained in the DEFRACF members. The CA OPS/MVS administrator generates these members.

Follow these steps:

1. Log in to a user ID that has sufficient authority with your z/OS security package to define resources and groups.
2. Locate the DEFRACF member containing the external security manager your CA OPS/MVS Administrator generated.
3. Run a batch job to execute the external security manager member. The SAMPLE JCL distributed with CA OPS/MVS in data set *opshlq*.CCLXCNTL contains the BATRACF member.

   Note: You can customize the sample batch job BATRACF to execute the external security manager commands member.

You have executed in batch the external security manager commands.

More information:

Resource Tables and Predefined Resources (see page 61)

Create Access Permissions with RACF

Connecting or permitting site-defined user names authorizes them to the groups or profiles that either you or DEFSAF defined. Execute the REXX DEFSAF utility to define the resource names and groups. Execute DEFSAF again to add user names to those groups.

Use IBM RACF to create access permissions. You can perform these steps in any order.

Follow these steps:

1. Execute the following command to provide the OPSUSER with READ access to all CA OPS/MVS protected resources:

   ```
   ISPF EDIT on member DEFSAF in opshlq.CCLXEXEC
   !OI ALL ACT(PERMIT) SAFRO(OPSUSER)
   ```

   or

   ```
   !OX ‘opshlq.CCLXEXEC(DEFSAF)’ ALL ACT(PERMIT) SAFRO(OPSUSER)
   ```

2. Execute the following command to provide the OPSOPER with UPDATE access to all CA OPS/MVS protected resources:

   ```
   ISPF EDIT on member DEFSAF in opshlq.CCLXEXEC
   !OI OPSAOF ACT(PERMIT) SAFRW(OPSOPER)
   ```

   or

   ```
   !OX ‘opshlq.CCLXEXEC(DEFSAF)' OPSAOF ACT(PERMIT) SAFRW(OPSOPER)
   ```

3. Execute the following command to provide OPSSQL1 with READ access to all SQL commands:

   ```
   ISPF EDIT on member DEFSAF in opshlq.CCLXEXEC
   !OI OPSSQL ACT(PERMIT) SAFRO(OPSSQL1)
   ```

   or

   ```
   !OX ‘opshlq.CCLXEXEC(DEFSAF)’ OPSSQL ACT(PERMIT) SAFRO(OPSSQL1)
   ```

   Note: The BATCH(Y) option generates the member in *opqhlq*.OPS*x*.DEFSAF named PERRACF. Send this member to your security administrator to run the RACF commands in the member from an authorized user ID. If the user ID where you run the DEFSAF command has sufficient authority, specify BATCH(N) and then issue the commands directly from DEFSAF.

4. Execute OPSSQL1 either with or without GROUPS as follows:

   - Execute OPSSQL1 without GROUPS(Y) and with UPDATE access to all aspects of SQL commands:

     ```
     ISPF EDIT on member DEFSAF in opshlq.CCLXEXEC
     !OI OPSSQL ACT(PERMIT) SAFRW(OPSSQL1) GROUPS(Y)
     ```

     or

     ```
     !OX ‘opshlq.CCLXEXEC(DEFSAF)’ OPSSQL ACT(PERMIT) SAFRW(OPSSQL1) GROUPS(Y)
     ```

- Execute OPSSQL1 with GROUPS(N) to permit the same access without using groups:

  ```
  TSO OX 'opshlq.CCLXEXEC(DEFSAF)' OPSSQL ACT(PERMIT) SAFRO(OPSSQL1) GROUPS(N)
  ISPF EDIT on member DEFSAF in opshlq.CCLXEXEC
  !OI OPSSQL ACT(PERMIT) SAFRO(OPSSQL1) GROUPS(N)
  ```

  or

  ```
  !OX ‘opshlq.CCLXEXEC(DEFSAF)’ OPSSQL ACT(PERMIT) SAFRO(OPSSQL1) GROUPS(N)
  ```

5. If you did not use DEFSAF, provide OPSSQL1 with READ access to all SQL commands by issuing the following sample RACF command:

   ```
   PERMIT OP$MVS.SQL.* CLASS(FACILITY) ID(OPSSQL1) ACCESS(READ)
   ```

You have created your access permissions with IBM RACF.

Add User Access to Product Resources

As the Administrator, you need two users added to your RACF external security package with the following access privileges:

- Provide user1 with READ access to address AOF (permitting safe verbs such verbs as LIST).
- Provide user2 with UPDATE access to address AOF (permitting all verbs).

Use one of the following methods to implement the desired access permissions to address AOF.

- Run the supplied DEFSAF REXX utility either with or without GROUPS as follows:
  - Execute the supplied REXX program DEFSAF without using GROUPS(N):

    ```
    TSO OX 'opshql.OPS.CCLXEXEC(DEFSAF)’ OPSAOF ACT(PERMIT) SAFRO(user1)
    TSO OX 'opshql.OPS.CCLXEXEC(DEFSAF)’ OPSAOF ACT(PERMIT) SAFRW(user2)
    ```

  - Execute DEFSAF with GROUPS(N) to permit the same access without using groups:

    ```
    TSO OX 'opshlq.CCLXEXEC(DEFSAF)' OPSAOF ACT(PERMIT) SAFRO(user1) GROUPS(N)
    TSO OX 'opshlq.CCLXEXEC(DEFSAF)' OPSAOF ACT(PERMIT) SAFRO(user2) GROUPS(N)
    ```

  Note: Be sure that you ran DEFSAF earlier in batch mode (BATCH=Y) to create the basic RACF groups.

- Issue the following RACF commands from TSO or from a self-styled REXX:

  ```
  CONNECT (user1)  GROUP(OPSAOF)  OWNER(OPSS)
  CONNECT (user1)  GROUP(OPSAOFR) OWNER(OPSS)
  ```

  Note: Be sure that you ran DEFSAF earlier in batch mode (BATCH=Y) to create the basic RACF groups.

- Issue the following RACF PERMIT command to add user1 directly under the resource OP$MVS.OPSAOF:

  ```
  PERMIT ‘OP$MVS.OPSAOF’   CLASS(FACILITY) ACCESS(READ)   ID(user1)
  PERMIT ‘OP$MVS.OPSAOF’   CLASS(FACILITY) ACCESS(UPDATE) ID(user2)
  ```

Authorize User IDs to Use a Specific Command

Group names are derived from the facility names that are associated with the security event the group name protects. The resources of some facilities have READ access, other facilities have UPDATE access, and other facilities have both READ and UPDATE verbs. Therefore, the group names are encoded following this pattern:

Facility	Group Name
READ access only	The group name is the same as the facility name or derived from the facility name.
UPDATE access only	The group name is the same as the facility name or derived from the facility name.
READ and UPDATE access	The group name for update access is derived from the facility name for UPDATE. The group name for read access is derived from the facility name with an appended R.

You can authorize a specific user ID or group of user IDs to use a particular CA OPS/MVS command either manually or using DEFSAF. The first procedure explains how to add authorizations manually. The second procedure explains how to execute the CA OPS/MVS REXX program DEFSAF.

Follow these steps:

1. Look up the command or function in Commands and Functions that Generate External Security. In the row where you find your command or function, make the following notes:
   - Note the associated Facility name.
   - Note whether your command includes verbs that generate either READ or UPDATE access.
2. Find the corresponding row that matches both your Facility name and access value. In that row, make a note of the group name.
3. Add the user ID or group of user IDs you want to authorize to the RACF group.
4. Issue this RACF command:

   ```
   CONNECT(userid) GROUP(RACF_group_name)
   ```

   You have manually authorized your user IDs.

Follow these steps:

1. Look up the command or function in Commands and Functions that Generate External Security. In the row where you find your command or function, make the following notes:

   - Note the associated Facility name.

   - Note whether your command includes verbs that generate either READ or UPDATE access.

2. Find your access value and do the following tasks:

   - If your access value is READ, then execute DEFSAF as follows:

     ```
     DEFSAF <facility> ACT(PERMIT) SAFRO(<userid>)
     ```

   - If your access value is UPDATE, then execute DEFSAF as follows:

     ```
     DEFSAF <facility> ACT(PERMIT) SAFRW(<userid>)
     ```

     <facility>

     Specify the facility name.

     <userid>

     Specify the user ID or group you want to authorize.

   You have authorized your user IDs using DEFSAF.

Appendix A: Resource Tables and Predefined Resources

This section contains the following topics:

SAF Resource Names Table

You can use the list of resource names to define your own profiles to RACF. The following table contains all resource names.

SAF Resource Name	SAF Access	Command or Function Subcommands	Facility
OP$MVS.OPSAOF	Read	(INDEX, LISTINST, LIST, LISTSRC, LISTCOMP)	OPSAOF
OP$MVS.OPSAOF	Update	(SETAUTO, DISABLE, ENABLE, COMPILE, DELCOMP, RESETAUTO)	OPSAOF
OP$MVS.AP	Update	.	OPSAP
OP$MVS.OPSBRW	Read	.	OPSBRW
OP$MVS.OPSCMD	Update	.	OPSCMD
OP$MVS.OPSCTL.COF	Read	.	OPSCTCOF
OP$MVS.OPSCTL.COF	Update	.	OPSCTCOF
OP$MVS.OPSCTL.ECF	Read	ECF	OPSCTECF
OP$MVS.OPSCTL.MSF	Read	MSF	OPSCTMSF
OP$MVS.OPSCTL.MSF	Update	MSF	OPSCTMSF
OP$MVS.OPSCTL.OPSLOG	Read	OPSLOG	OPSCTLOG
OP$MVS.OPSCTL.OPSLOG	Update	OPSLOG	OPSCTLOG
OP$MVS.OPSCTL.OSF	Read	OSF	OPSCTOSF
OP$MVS.OPSCTL.OSF	Update	OSF	OPSCTOSF

SAF Resource Name	SAF Access	Command or Function Subcommands	Facility
OP$MVS.OPSDOM	Update	.	OPSDOM
OP$MVS.OPSEPI	Update	.	OPSEPI
OP$MVS.OPSGLOBAL.[AUGLDENA]	Read	.	OPSGLOBAL
OP$MVS.OPSGLOBAL.[AUGLDENA]	Update	.	OPSGLOBAL
OP$MVS.OPSHFI	Update	.	OPSHFI
OP$MVS.OPSLOG	Read	.	OPSLOG
OP$MVS.OPSPARM	Read	(SHOW)	OPSPARM
OP$MVS.OPSPARM	Update	(SET)	OPSPARM
OP$MVS.OPSOSF.OSF	Update	.	OPSOSF
OP$MVS.OPSOSF.OSFTSL	Update	.	OPSOSTSL
OP$MVS.OPSOSF.OSFTSP	Update	.	OPSOSTSP
OP$MVS.OPSREPLY	Update	.	OPSREP
OP$MVS.OPSREQ	Update	.	OPSREQ
OP$MVS.OPSRMT	Update	.	OPSRMT
OP$MVS.OPSSMTBL	Read	(List)	OPSSSM
OP$MVS.OPSSMTBL	Update	(Add, Change, Delete, Post)	OPSSSM
OP$MVS.OPSVIEW	Read	.	OPSVIEW
OP$MVS.OPSWTO	Update	.	OPSWTO
OP$MVS.SOF	Read	(QUERY, PPRCCMD DISPLAY)	OPSSOF
OP$MVS.SOF	Update	(COMMAND, FIND, LOG, READ, WRITE, DELETE, TERM, TERMINATE, PPRCCMD SETUP/DELETE/FREEZE/RUN)	OPSSOF
OP$MVS.SQL.[tbl].[cmd]	Read	See SQL TBL.CMD Names Table (see page 63)	OPSSQL
OP$MVS.SQL.[tbl].[cmd]	Update	See SQL TBL.CMD Names Table (see page 63)	OPSSQL

SAF Resource Name	SAF Access	Command or Function Subcommands	Facility
OP$MVS.SUBSYSDSN	Update	.	OPSSUB
OP$MVS.USS	Update	.	OPSUSS

SQL TBL.CMD Names Table

Value	SQL command
CT	Create table
IN	Insert rows
UP	Update rows
SE	Select
DE	Delete rows
DC	Declare cursor
DT	Drop table
CA	Alter table add column
CD	Alter table drop column
CI	Create index
DI	Drop index

Commands and Functions that Generate External Security Events

The table in this section provides the following information:

- All CA OPS/MVS commands
- The SAF access level that is required to execute the command
- The SAF resource name that is used to validate the command issuers
- The group name, which is derived from the facility names that is associated with the security event the group name protects
- The facility name for DEFSAF

The first qualifier of the resource name (OP$MVS in the table SAF Resource Names Table (see page 61)) is the default resource name prefix value. You can override the prefix value with the EXTSECPREFIX parameter to meet local naming standards.

Review the following explanations of the **var* and ***table* variables in the table column SAF Resource Name:

***var**

Contains the name of the global variable being read or updated. When the variable starts with one of the global stem prefixes, the *var* appended to the resource name is the variable name itself. If the variables do *not* start with one of the product-defined global stems, the prefix GLVTEMPG gets added before *var*. For sysplex variables, the recognized prefix is GLVPLXTx.

Note: The global stem prefixes recognized by CA OPS/MVS are GLOBAL, GLOBALx, or GLVTEMPx.

For example:

OPSVALUE(MYVAR)

Checks using resource name OP$MVS.OPSGLOBAL.GLVTEMPG.MYVAR for READ access

OPSVALUE(GLOBAL.1)

Checks using resource name OP$MVS.OPSGLOBAL.GLOBAL.1 for READ access

OPSSETV(MYVAR ('1'))

Checks using resource name OP$MVS.OPSGLOBAL.GLVTEMPG.MYVAR for UPDATE access

OPSVASRV(“CREATE NAME(GLVPLXT1.TESTVAR) DATAVAL(Testcase)”)

Checks using resource name OP$MVS.OPSGLOBAL.GLVPLXT1.TESTVAR for UPDATE access.

****table**

Most simple SQL statements have only one table reference. In those cases, SQLTBL contains that table name and checks only that one resource.

More complex SQL statements (such as joins and subselect clauses) can reference more than one table. In those cases, SQLTBL treats each table as a separate resource with potentially its own access requirements. A separate SAF call initiates for each table referenced.

For example:

Address SQL select name from table

Checks using resource name OP$MVS.SQL.TABLE for READ access.

Address insert into t1 select * from t2

Checks the resource name OP$MVS.SQL.T1 for UPDATE, and then verifies OP$MVS.SQL.T2 for READ access.

Note: The utility DEFSAF eliminates the need to remember group names. If you use these SAF group names outside of DEFSAF, be sure to use the correct group name when adding user IDs to groups.

The facilities are divided into the following three types:

- Facilities containing a single READ group

  For example: OPSBRW

- Facilities containing a single UPDATE group

  For example: OPSAP

- Facilities containing two groups – one for READ and one for UPDATE. In this case, the READ group has an appended R

  For example: OPSAOFR

Command or Function (subcommand verb)	Description	SAF Access	SAF Resource Name	Group Name	Facility Name (for DEFSAF)
address AOF (INDEX, LISTINST, LIST, LISTSRC, LISTCOMP)	Access AOF (Rule or Rule sets)	Read	OP$MVS.OPSAOF	OPSAOFR	OPSAOF
address AOF (SETAUTO, DISABLE, ENABLE, COMPILE, DELCOMP, RESETAUTO)	Modify AOF (Rule or Rule sets)	Update	OP$MVS.OPSAOF	OPSAOF	OPSAOF

Command or Function (subcommand verb)	Description	SAF Access	SAF Resource Name	Group Name	Facility Name (for DEFSAF)
address AP (all verbs)	All OPS-AP Interface commands	Update	OP$MVS.AP	OPSAP	OPSAP
OPSLOG() function (all verbs)	Retrieve information from OPSLOG	Read	OP$MVS.OPSBRW	OPSBRW	OPSBRW
address OPER (all verbs)	Issuing z/OS commands	Update	OP$MVS.OPSCMD	OPSCMD	OPSCMD
address TSO OPSCMD (all verbs)	Issuing z/OS commands	Update	OP$MVS.OPSCMD	?OPSCMD?	OPSCMD
address OPSCTL COF (LIST)	Access COF components	Read	OP$MVS.OPSCTL.COF	OPSCTCFR	OPSCTCOF
address OPSCTL COF (ACTIVATE, DEACTIVATE, DEFINE, DELETE)	Modify COF components	Update	OP$MVS.OPSCTL.COF	OPSCTCOF	OPSCTCOF
address OPSCTL ECF (all verbs)	Access ECF components	Read	OP$MVS.OPSCTL.ECF	OPSCTECF	OPSCTECF
address OPSCTL MSF (LIST)	Access MSF components	Read	OP$MVS.OPSCTL.MSF	OPSCTMSR	OPSCTMSF
address OPSCTL MSF (ACTIVATE, DEACTIVATE, DEFAULT, DEFINE, DELETE, START, STOP)	Modify MSF operations	Update	OP$MVS.OPSCTL.MSF	OPSCTMSF	OPSCTMSF
address OPSCTL OPSLOG (LIST)	Access OPSLOG management and control	Read	OP$MVS.OPSCTL.OPSLOG	OPSCTLGR	OPSCTLOG

Command or Function (subcommand verb)	Description	SAF Access	SAF Resource Name	Group Name	Facility Name (for DEFSAF)
address OPSCTL OPSLOG (ACTIVATE, DEACTIVATE, DEFINE, DELETE, RESET, SETLIVE)	Modify OPSLOG management and control	Update	OP$MVS.OPSCTL.OPSLOG	OPSCTLOG	OPSCTLOG
address OPSCTL OSF (EXECSTATS, QUEUE, LIST)	Access OSF components	Read	OP$MVS.OPSCTL.OSF	OPSCTOSR	OPSCTOSF
address OPSCTL OSF (RESETQ, STOP)	Modify OSF components	Update	OP$MVS.OPSCTL.OSF	OPSCTOSF	OPSCTOSF
address TSO OPSDOM or OPSDOM()	Deleting an Operator Message	Update	OP$MVS.OPSDOM	OPSDOM	OPSDOM
address EPI (all verbs)	Using External Product Interface	Update	OP$MVS.OPSEPI	OPSEPI	OPSEPI
OPSVALUE() function (O, E, F, I, J, K, L, N, O, S, T)	Access global variables	Read	OP$MVS.OPSGLOBAL.*var**	OPSGLOB	OPSGLOBAL
OPSVALUE() function (6, A. C, D, R, U, V)	Modify global variables	Update	OP$MVS.OPSGLOBAL.*var**	OPSGLOBR	OPSGLOBAL
OPSHFI function or command processor (all verbs)	Access global variables from a VSAM data set	Update	OP$MVS.OPSHFI	OPSHFI	OPSHFI
OPSLOG API	Access OPSLOG using OPSLOG API	Read	OP$MVS.OPSLOG	OPSLOG	OPSLOG
OPSPARM() function (SHOW)	Accessing CA OPS/MVS parameters	Read	OP$MVS.OPSPARM	OPSPAR	OPSPARM

Command or Function (subcommand verb)	Description	SAF Access	SAF Resource Name	Group Name	Facility Name (for DEFSAF)
OPSPARM() function (SET)	Modifying CA OPS/MVS parameters	Update	OP$MVS.OPSPARM	OPSPARM	OPSPARM
address TSO OPSREPLY (all verbs)	Reply to WTORs	Update	OP$MVS.OPSREPLY	OPSREP	OPSREP
address TSO OPSREQ (all verbs)	Invoke AOF request (REQ) rules	Update	OP$MVS.OPSREQ	OPSREQ	OPSREQ
address TSO OPSRMT (all verbs)	Send commands to remote OPS	Update	OP$MVS.OPSRMT	OPSRMT	OPSRMT
address OSF (all verbs)	Send command to an OSF server	Update	OP$MVS.OPSOSF.OSF	OPSOSF	OPSOSF
address OSFTSL (all verbs)	Send command to OSFTSL server	Update	OP$MVS.OPSOSF.OSFTSL	OPSOSTSL	OPSOSTSL
address OSFTSP (all verbs)	Send command to OSFTSP server	Update	OP$MVS.OPSOSF.OSFTSP	OPSOSTSP	OPSOSTSP
OPSSMTBL() function (LIST)	Accessing STATEMAN definitions	Read	OP$MVS.OPSSMTBL	OPSSSMR	OPSSSM
OPSSMTBL() function (ADD, CHANGE, DELETE, POST)	Modifying STATEMAN definitions	Update	OP$MVS.OPSSMTBL	OPSSSM	OPSSSM
address SOF (QUERY, PPRCCMD DISPLAY)	SOF commands	Read	OP$MVS.SOF	OPSSOFR	OPSSOF

Command or Function (subcommand verb)	Description	SAF Access	SAF Resource Name	Group Name	Facility Name (for DEFSAF)
address SOF (COMMAND, FIND, LOG, READ, WRITE, DELETE, TERM, TERMINATE, PPRCCMD SETUP/DELETE/ FREEZE/RUN)	SOF commands	Update	OP$MVS.SOF	OPSSOF	OPSSOF
address SQL or OPSSQL function (SELECT, DECLARE, OPEN, FETCH, CLOSE)	Accessing SQL tables	Read	OP$MVS.SQL.*table***	OPSSQLR	OPSSQL
address SQL or OPSSQL function (CREATE, INSERT, UPDATE, DELETE, DROP, ADD)	Modifying SQL tables	Update	OP$MVS.SQL.*table***	OPSSQL	OPSSQL
SUBSYSTEM data set OPEN Request	Allocate OPSS SUBSYS data set	Update	OP$MVS.SUBSYSDSN	OPSSUB	OPSSUB
address USS or OPSUSS() function (all verbs)	All UNIX System Services	Update	OP$MVS.USS	OPSUSS	OPSUSS
OPSVIEW command	Access ISPF interface	Read	OP$MVS.OPSVIEW	OPSVW	OPSVIEW
address WTO (all verbs)	Send a WTO	Update	OP$MVS.OPSWTO	OPSWTO	OPSWTO
OPSWTO command(all verbs)	Send a WTO	Update	OP$MVS.OPSWTO	OPSWTO	OPSWTO

Predefined Resources Used by External Security

The CA OPS/MVS external security feature uses the following resources to make SAF calls:

- OP$MVS.AP - ADDRESS AP Host command
- OP$MVS.OPSAOF - ADDRESS AOF command
- OP$MVS.OPSBRW - OPSLOG Browse request
- OP$MVS.OPSCMD - ADDRESS OPER (MVS, VM, JES3, IMS)
- OP$MVS.OPSCTL.COF - ADDRESS OPSCTL COF request
- OP$MVS.OPSCTL.ECF - ADDRESS OPSCTL ECF request
- OP$MVS.OPSCTL.MSF - ADDRESS OPSCTL MSF request
- OP$MVS.OPSCTL.OPSLOG - ADDRESS OPSCTL OPSLOG request
- OP$MVS.OPSCTL.OSF - ADDRESS OPSCTL OSF request
- OP$MVS.OPSDOM - DOM message request
- OP$MVS.OPSEPI - ADDRESS EPI command or EPI request
- OP$MVS.OPSGLOBAL.*global_variable_name* - Global variable access and update request
- OP$MVS.OPSHFI - Shared file I/O request
- OP$MVS.OPSLOG - OPSLOG API request
- OP$MVS.OPSOSF - OPSOSF request (OSF command request)
- OP$MVS.OPSPARM - OPSPARM set parameters request
- OP$MVS.OPSREPLY - OPSREPLY (WTO/WTOR) request
- OP$MVS.OPSREQ - Attempt to execute a REQUEST rule
- OP$MVS.OPSRMT - SEND a command to a server request
- OP$MVS.OPSSMTBL - STATETBL request
- OP$MVS.OPSVIEW - OPSVIEW request
- OP$MVS.OPSWTO - OPSWTO and ADDRESS WTO (WTO, WTP, WTOR request
- OP$MVS.SOF - SOF command request
- OP$MVS.SQL - SQL/RDF request
- OP$MVS.SUBSYSDN - Subsystem data set open request
- OP$MVS.USS - ADDRESS USS command

Note: The two facilities OPSGLOBAL and OPSSQL have an additional suffix appended to their resource names. By default DEFSAF ALL only creates generic resources to control these two resources. You can control access to specific tables under the SQL resources or specific variables under the OPSGLOBAL facility. Simply run DEFSAF with the required parameters to get the resources defined to your external security manager.

More information:

How SAF Resources Are Defined to Use External Security (see page 19)
Control Table Access Using SQL Resources or OPSGLOBAL (see page 22)
Commands and Functions that Generate External Security Events (see page 64)

Appendix B: Troubleshooting External Security

This section contains the following topics:

CA Top Secret Options that Affect Product Access Requests

CA Top Secret has many possible configurations and combinations. Some of the configurations affect the CA OPS/MVS requests for access authorization. Review the following troubleshooting topics.

Access Granted Without Reference to Access Profiles

Symptom:

My CA Top Secret batch job allows access requests to CA OPS/MVS resources without reference to the CA Top Secret defined resource access profiles.

Solution:

Setting the batch mode of CA Top Secret to DORM or WARN can affect CA OPS/MVS commands or functions the z/OS batch jobs issue.

Change the value of MODE for the user to IMPL or FAIL.

ACID Bypassed Security Checking

Symptom:

My CA Top Secret ACIDs allow access requests to CA OPS/MVS resources without reference to the CA Top Secret defined resource access profiles.

Solution:

Verify that the NORESCHK attribute is not attached to the individual ACIDs.

```
TSS REMOVE(acid) NORESCHK
```

NORESCHK

Lets an ACID bypass security checking for all owned resources except data sets and volumes.

Index