Enabling the FACILITY Class for Use by RACF for z/VM

Prepared by the IBM z/VM Development Lab
Valid through: z/VM 6.4
Brian W. Hugenbruch, CISSP
IBM z Systems Virtualization Security
bwhugen@us.ibm.com

Change Summary:

- 10 October 2016 Updates based on field feedback (for z/VM® 6.4)
- 03 December 2014 Updates based on field feedback (for z/VM 6.3)
- 30 November 2011 Initial version (for z/VM 6.2)

Introduction

IBM's RACF® for z/VM provides a mechanism for the control of resources associated with the z/VM environment. These external controls expand z/VM's security management and auditing capabilities in scope and granularity. RACF for z/VM maintains its control of resources through the use of profiles – a definition of a resource to the system. Each profile is associated with a Class – a category grouping for how a type of resource is handled.

The FACILITY Class is a category for miscellaneous resource profiles. Such profiles will normally be associated with program products or components – for example, IBM Operations Manager for z/VM and IBM Tape Manager for z/VM both make use of the FACILITY Class. Similarly, the RACROUTE program is authorized through the FACILITY Class.

This paper will discuss how to enable the FACILITY Class for use by such a program.

Notes for Readers

1. This paper assumes that RACF for z/VM has been installed and is currently active in the environment. Further, it assumes that the environment being protected is a single z/VM system or single SSI cluster using a single RACF database, and that any pertinent product configuration steps to enable RACF control have been correctly followed.
2. The RACF commands listed here must be sent to RACF using the RAC exec. Prepend any sample command with the token "RAC" to issue the command. The RAC exec will automatically propagate commands to other members of an SSI cluster, so there is no need to issue the RACF command on other members to have it take effect. Commands can also be issued from a RACF command session, but an error message will be issued for any command that must be propagated to other SSI members.
3. It is recommended that generic profiles be created to protect or authorize all commands, and then define discrete profiles to grant or deny specific authority. This is useful for cases where distinct groups of users (for example, operators or general users) have different scopes of responsibilities. For example, a group of operators with equal authority to issue TAPE or OPERATOR commands could be placed in a group and given authority to the generic profile. General users could be granted authority to use certain commands by granting the userid authority to them via a discrete profile.

   Note: when using both generic and discrete profiles, discrete profiles supersede any

security authorization granted by generic profiles—generic access does not imply discrete access.

For more information about generic profiles, please consult the z/VM RACF Security Server Security Administrator's Guide (SC24-6218), Chapter 6 (“Defining Resources”).

4. To prevent an escalation of authority, validate whether or not any accesses granted require either READ or UPDATE authority. The sample commands in this document may use either, but these may not be accurate for local security requirements. Consult appropriate product manuals to verify access requirements to the FACILITY class.
5. Be advised that virtual machines granted READ or UPDATE authority to the FACILITY class as described in this paper may or may not be the same users (or enrolled in the same GROUP) as the administrators who own or CONTROL the profile. The examples in this document may not necessarily be in accordance with previously defined security policy inside of RACF for z/VM. Consult with local documentation when assigning an owner to newly created security profiles.

Enabling the FACILITY Class

Verifying RACF Administrator Authority

For the context of this paper, the commands which follow should be issued from a virtual machine defined to RACF with the SPECIAL attribute. This is commonly a virtual machine belonging to the security administrator.

To determine if your virtual machine has this authority, issue the following RACF command and check the 'Attribute' field:

```
LISTUSER userid
```

If your virtual machine does not have SPECIAL authority, please consult with RACF for z/VM documentation for more detailed configurations for security administration.

Setup for RACF Profile Creation

To enable generic profiles for FACILITY class and enable GRPLIST where access is based on the authority of any group to which the user is connected, issue the following commands:

```
SETROPTS GRPLIST GENERIC(FACILITY)
```

Next, validate that NOADDCREATOR is specified. This will prevent the user creating FACILITY profiles from being added to the Access Control Lists for those resources. To determine whether or not NOADDCREATOR is in effect, issue:

```
SETROPTS LIST
```

NOADDCREATOR is the default for new RACF databases in z/VM 6.4. If it is not currently enabled and you do not wish your issuing user added to the ACLs, also issue the following:

```
SETROPTS NOADDCREATOR
```

Defining a Resource Profile to the FACILITY Class

Definition of resource profiles to the FACILITY Class can be handled prior to activating the Class. This allows for the creation of security policy prior to enforcement – and avoids inadvertent rejections due to incomplete policy. The FACILITY Class is broad in its usage, so the specifics of required profiles may vary from program to program.

To define a resource profile for the FACILITY Class, the following commands will be used:

```
ADDGROUP groupname OWNER(userid) SUPGROUP(supergroupname)
CONNECT userid GROUP(groupname) OWNER(owner)
RDEFINE FACILITY profilename OWNER(groupname) UACC(acc)
```

- ADDGROUP creates a new group which can own resources created using RDEFINE. This step is not necessary if you already have a userid or group defined to own the profiles being created for the FACILITY class.
- The CONNECT command adds users to the group. This step is not necessary if you already have the appropriate users connected to a group that will own the profiles being created for the FACILITY class.
- The RDEFINE command creates a profile of profilename in the currently-disabled FACILITY Class, owned by groupname and with a universal access of acc. The owner of a given profile may either be a group (in this case, groupname) or a userid with SPECIAL authority – for example, the userid issuing these RACF commands. Universal access can be any valid value such as NONE or READ.

To grant access to the profile created by the RDEFINE command above, use the PERMIT command, specifying READ or UPDATE authority, depending upon the program requirements. Access can be granted on a GROUP basis:

```
PERMIT profilename CLASS(FACILITY) ID(groupname)
ACCESS({READ|UPDATE})
```

Access can also be granted to the profile on a per-userid basis:

```
PERMIT profilename CLASS(FACILITY) ID(userid1 userid2 useridn)
ACCESS({READ|UPDATE})
```

To specify default access for ALL userids (known as universal access), remember to include an appropriate value on the RDEFINE command, as follows:

```
RDEFINE FACILITY GOM.OPSMGR.COMMAND.command UACC(READ)
RDEFINE FACILITY EUM.TAPEMGR.CMND.command UACC(READ)
```

Use a combination of generic and discrete profiles to set up your RACF authorities to meet local security requirements. See Chapter 5 in the *IBM Tape Manager for z/VM Installation and Administration Guide (SC18-9344)* or Chapter 2 in the *IBM Operations Manager for z/VM Administration Guide (SC18-9347)* to find a more granular list of profiles to define.

If you are creating FACILITY class profiles for z/VM Related Products such as Operations Manager for z/VM or Tape Manager for z/VM, the following examples can be used as a guide.

Example Definition: Operations Manager for z/VM

1. First put all users who will have full authority to Operations Manager in the same RACF/VM group OPSADMS. For the owner, *secadmin* is the userid or group of the z/VM security administrator.

   ADDGROUP OPSADMS OWNER(secadmin) SUPGROUP(SYS1)
   CONNECT userid GROUP(OPSADMS) OWNER(secadmin)

2. Define generic FACILITY class profile GOM.OPSMGR. Then allow all users in group OPSADMS to issue all Operations Manager commands.

   RDEFINE FACILITY GOM.OPSMGR.* OWNER(OPSADMS) UACC(NONE)
   PERMIT GOM.OPSMGR.* CLASS(FACILITY) ID(OPSADMS) ACCESS(READ)

3. Define GOM.OPSMGR.CONFIG and permit *userid1* to have authority to issue only configuration commands:

   RDEFINE FACILITY GOM.OPSMGR.CONFIG OWNER(OPSADMS) UACC(NONE)
   PERMIT GOM.OPSMGR.CONFIG CLASS(FACILITY) ID(userid1) ACCESS(READ)

4. Define GOM.OPSMGR.VIEWCON and permit *userid2* to have full authority to view and issue commands on monitored consoles:

```
RDEFINE FACILITY GOM.OPSMGR.VIEWCON OWNER(OPSADMS) UACC(NONE)
PERMIT GOM.OPSMGR.VIEWCON CLASS(FACILITY) ID(userid2) ACCESS(UPDATE)
```

5. Define GOM.OPSMGR.VIEWSPL and permit *userid3* to have full authority to view all spool files, purge all spool files, and alter any spool file's external attributes:

```
RDEFINE FACILITY GOM.OPSMGR.VIEWCON OWNER(OPSADMS) UACC(NONE)
PERMIT GOM.OPSMGR.VIEWSPL CLASS(FACILITY) ID(userid3) ACCESS(UPDATE)
```

Example Definition: Tape Manager for z/VM

1. Put all users who will have full authority to Tape Manager in the same RACF/VM group TAPEADMS. For the owner, *secadmin* is the userid or group of the z/VM security administrator.

```
ADDGROUP TAPEADMS OWNER(secadmin) SUPGROUP(SYS1)
CONNECT userid GROUP(TAPEADMS) OWNER(secadmin)
```

2. Define generic FACILITY class profile EUM.TAPEMGR. Then allow all users in group TAPEADMS to issue Tape Manager commands.

```
RDEFINE FACILITY EUM.TAPEMGR.* UACC(NONE)
PERMIT EUM.TAPEMGR.* CLASS(FACILITY) ID(TAPEADMS) ACCESS(READ)
```

3. Define EUM.TAPEMGR.AUTH.OPER and permit *userid1* to have only OPERATIONS authority in Tape Manager..

```
RDEFINE FACILITY EUM.TAPEMGR.AUTH.OPER OWNER(TAPEADMS) UACC(NONE)
PERMIT EUM.TAPEMGR.AUTH.OPER CLASS(FACILITY) ID(userid1) ACCESS(READ)
```

Defining ICHCONN to the FACILITY Class

Each virtual server that calls RACF using the RACROUTE interface must be authorized to use the RACROUTE interface. This is controlled with the ICHCONN profile. Define the ICHCONN profile in the FACILITY Class for use by virtual servers.

```
RDEFINE FACILITY ICHCONN OWNER(groupname) UACC(NONE)
```

Authorize each virtual server which will call RACF using the RACROUTE interface by granting it UPDATE authority to the ICHCONN profile. For example, authorize userid OPMGRM1 for Operations Manager and authorize userid TMTMM for Tape Manager, as shown below:

```
PERMIT ICHCONN CLASS(FACILITY) ID(OPMGRM1) ACCESS(UPDATE)
PERMIT ICHCONN CLASS(FACILITY) ID(TMTMM) ACCESS(UPDATE)
```

Note also that it may be advisable to PERMIT users such as OPMGRM1 and TMTMM by GROUP rather than by userid. Consult with local security policy and RACF for z/VM documentation for more details.

Initiating FACILITY Class Usage

When all pertinent profiles have been created, enable the FACILITY Class. All RACF for z/VM Classes are disabled by default and need to be started by a security administrator. To enable the Class, issue the following command:

```
SETROPTS CLASSACT(FACILITY)
```

This command will activate the FACILITY Class and enable access controls for all profiles defined to it. Adding new profiles will not require the reactivation of the FACILITY Class.

Tasks may also need to be performed to enable products such as Operations Manager for z/VM and Tape Manager for z/VM to use RACF as their authorization mechanism. Refer to that product documentation for details.

Performance and Maintenance Considerations

If you have a lot of authorization requests to RACF, you may consider activating SETROPTS RACLIST processing for the FACILITY general resource Class. When you activate this function, you improve performance because I/O to the RACF database is reduced.

```
SETROPTS RACLIST(FACILITY)
```

However, a REFRESH will be needed each time the FACILITY class profiles are added, deleted, or changed. For example, a PERMIT for a userid to a FACILITY class profile:

```
PERMIT GOM.OPSMGR.CONFIG CLASS(FACILITY) ID(userid) ACCESS(READ)
```

--will only take effect after issuing the command:

```
SETROPTS RACLIST(FACILITY) REFRESH
```

Validating Class Activation

The status of the Class can be verified by issuing the following command:

```
SETROPTS LIST
```

The list of all active Classes will be displayed under the 'ACTIVE CLASSES' field. The FACILITY Class should be displayed in this list.

Additionally, one can display the profiles, controls and auditing rules associated with the FACILITY Class by issuing the following:

```
RLIST FACILITY *
```

To see a list of all the userids that have authority to the ICHCONN FACILITY Class, issue:

```
    RLIST FACILITY ICHCONN AUTH
```

Summary

The FACILITY Class can be enabled in a few easy steps, and it allows for control and auditing of a broad variety of programmatic resources. Enablement should only be initiated after all pertinent profiles have been defined.

For more information about customizing the FACILITY Class for use by other programs and components, see Chapter 13 of the *z/VM RACF Security Server Security Administrator's Guide (SC24-6218)*.

Enabling the FACILITY Class for Use by RACF for z/VM
November 2016

IBM Systems
Route 100
Somers, New York 10589
U.S.A.
Produced in the United States of America,
11/2016

ZSW03289-USEN-02