THE RESOURCE FOR USERS OF IBM MAINFRAME SYSTEMS

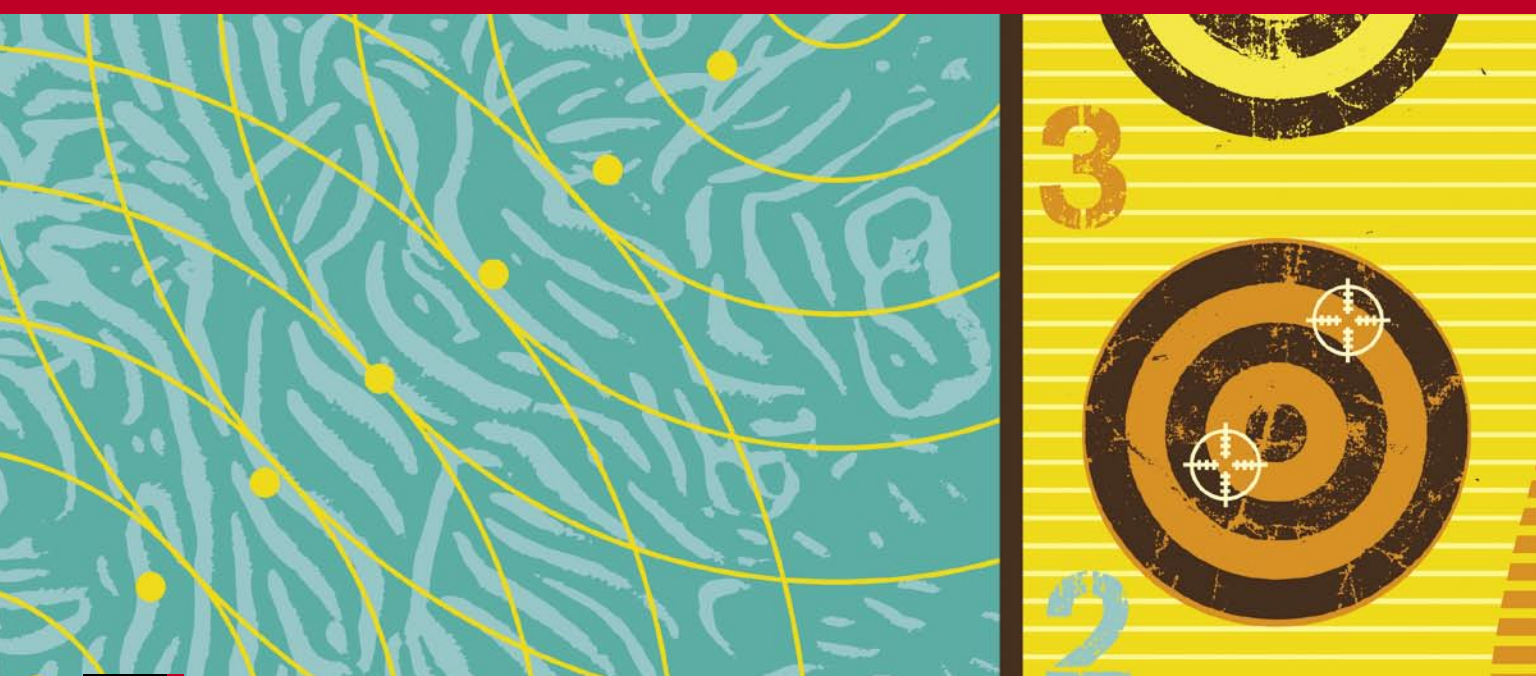
WWW.ZJOURNAL.COM

APRIL/MAY 2006

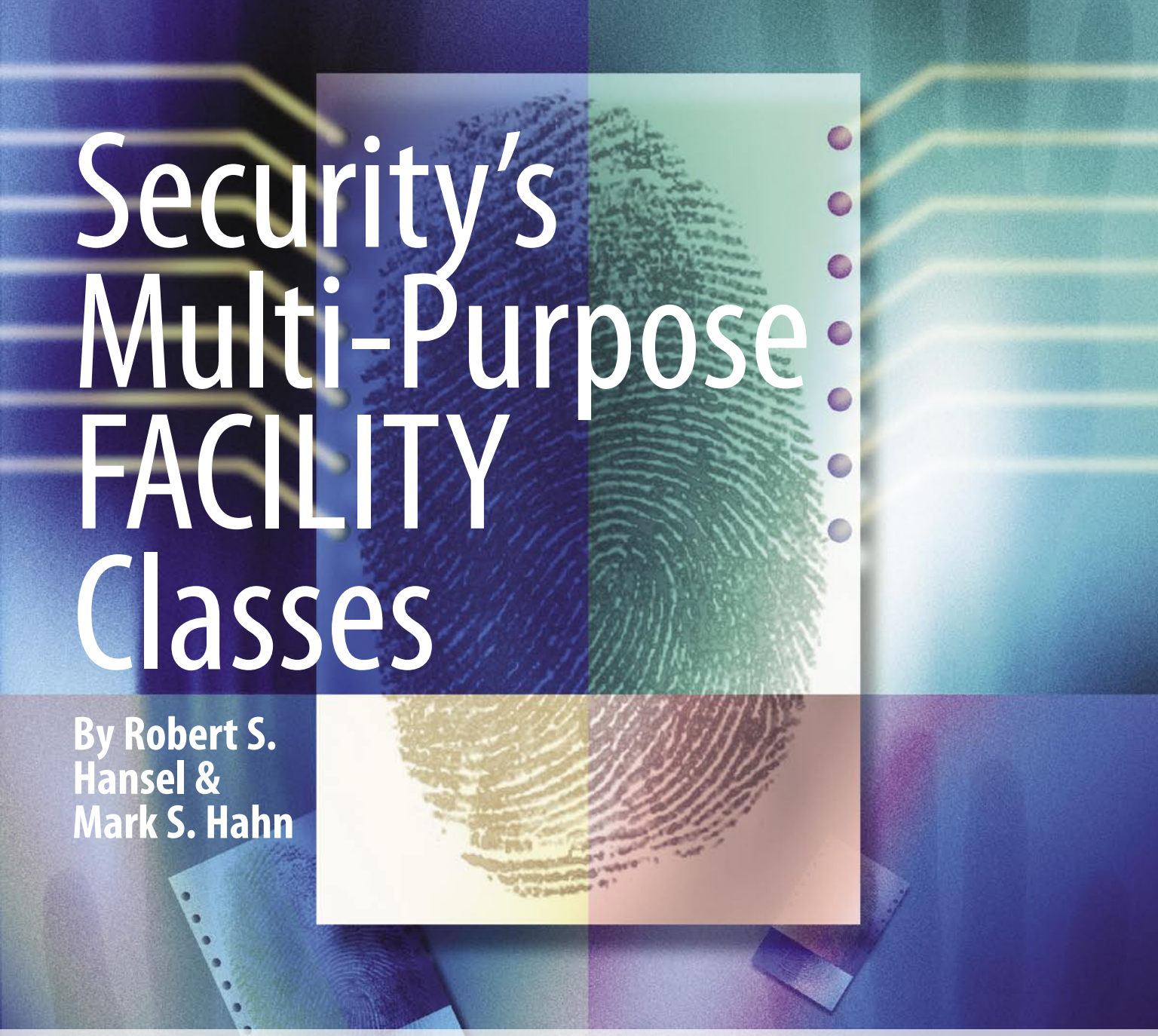
zJOURNAL

Security's Multi-Purpose FACILITY Classes

By Robert S. Hansel & Mark S. Hahn



A tci PUBLICATION



Security's Multi-Purpose FACILITY Classes

By Robert S. Hansel & Mark S. Hahn

Which RACF class has grown from housing a few profiles to housing more than 100? Which class has grown into a major player in the control of many key z/OS services? It also has recently grown from a single class into one with a companion member-grouping class pair. You'd be correct if you answered the FACILITY class. But what services are taking advantage of this long-standing class and why has a new member/grouping class pair been created?

This article explains the base concepts and use of the FACILITY class and its new companion classes, XFACILIT and GXFACILI. These new classes were introduced in z/OS 1.7 and can be rolled back to z/OS 1.4 with APAR OA10774. Currently, only the IBM Health Checker for z/OS uses XFACILIT. While we'll discuss the FACILITY class from a RACF perspective, the information provided applies to other mainframe security products that respond to RACROUTE calls to the System Authorization Facility (SAF). Instead of the FACILITY class, CA-ACF2 knows this as FAC and CA-Top Secret uses IBM-FAC.

Early in RACF's evolution, the developers recognized a need for a general-purpose resource class available to any system service with too few resources needing protection to justify a separate class. This was back when the maximum number of resource classes was relatively small and implementation of an additional class required an Initial Program Load (IPL), a constraint that has only recently been lifted with the Dynamic CDT introduced in z/OS 1.6. The FACILITY class was born in response to these original requirements.

Over the years, an increasing number of system services used the FACILITY class to protect their growing list of functions. From its humble beginning of just a few system services and a handful of their related resources, the FACILITY class has grown to dozens of system services and more than 100 different resources. Many third-party products also have chosen to use the FACILITY class because it was readily available

POSIT=8	The POSIT value is a handle linking classes together for managing as a set. Previously, IBM reserved the value 8 solely for the FACILITY class. Now FACILITY shares this value with the new XFACILIT/GXFACILI class pair. When any one of the FACILITY classes is refreshed, the others will be as well.
MAXLNTH=39 FIRST=ANY OTHER=ANY KEYQUAL=0	These parameters govern the format of the profile names. The names are at most 39 characters in length, can start with any character, and can include any character. The XFACILIT and GXFACILI class definitions specify 246 characters as the maximum length of the resource names. KEYQUAL of 0 enables generic characters to be used in any qualifier, including the first.
DFTRETC=4 DFTUACC=NONE OPER=NO	This set of parameters affects security. DFTRETC is the default return code RACF issues when there is no profile protecting a resource. The value 4 indicates RACF makes no decision, and the caller must decide whether or not to grant the access. Some system services deny use of particular resources when this occurs. For example, DFSMSrmm does not permit the use of certain EDGINERS parameters if resource STGADMIN.EDG. INERS.WRONGLABEL is not protected. DFTUACC is the default Universal Access (UACC) set when a profile is first created and the UACC isn't specified in the RDEFINE command. OPER governs whether OPERATIONS authority will grant access, and for the FACILITY class it does not.
GENLIST=ALLOWED RACLIST= ALLOWED RACLREQ=NO	These parameters relate to performance. They allow the class to use either GENLIST or RACLIST caching. GENLIST caches only the generic profiles in the Common Storage Area (CSA); RACLIST caches all profiles in a data space. The two are mutually exclusive. RACLIST isn't required, but certainly recommended. (Some system services will automatically RACLIST the FACILITY class if it isn't already.)

Figure 1: FACILITY Class–RACF Class Descriptor Table Entry Characteristics

```
HZS.SYS1.IBMRACF.RACF_GRS_RNL.QUERY
HZS.SYSA.IBMCNZ.CNZ_AMRF_EVENTUAL_ACTION_MSGS.QUERY
```

Figure 2: Two Examples of the IBM Health Checker for z/OS Resource Names

```
CLASS       NAME
FACILITY    BPX.DEFAULT.USER

LEVEL  OWNER       UNIVERSAL ACCESS  YOUR ACCESS  WARNING
00     SYS1            NONE              NONE        NO
...
 APPLICATION DATA
  $OMDFLU/@OMDFLG
```

Figure 3: Abbreviated Display of BPX.DEFAULT.USER Profile

```
JES2 – EXIT6 - Control job execution characteristics
JOBCLASS.class
JOB.CPU.TIME.EXTEND
STEP.CPU.TIME.EXTEND
WAIT.TIME.EXTEND

DADSM (Direct Access Device Storage Manager) - IGGPRE00 Exit -
Control dataset allocations on non SMS-managed Direct Access
Storage Device (DASD) volumes
$DASDI.volser
$DASDI.NOVOLCHK

SAF - Router Exit - Allow use of surrogate IDs. Preceded introduction
of the SURROGAT class
$SUBMIT.userid
```

Figure 4: Sample Exits for Using FACILITY Class Profiles

in all MVS, OS/390 and z/OS systems, and required no modification to the RACF tables.

FACILITY Classes Characteristics

The FACILITY classes and their characteristics are defined in the RACF Class Descriptor Table (CDT). Figure 1 describes a few of the more pertinent characteristics.

The FACILITY class maximum of 39 characters is quite limiting when creating descriptive names for newer resources. This RACF limitation, which wasn't lifted until the restructured database in RACF 1.9, has been addressed in the new XFACILIT and GXFACILI classes, where resource names can be up to 246 characters. This is needed by the IBM Health Checker for z/OS, since it addresses the increasingly complex system environment by including such information as the system name, owner name and function in the resource name for authorization. Two such examples appear in Figure 2.

Resource Naming Conventions

Most system services using the FACILITY classes follow the naming conventions used by data sets: Names are constructed of one or more qualifiers (i.e., nodes) separated by periods such as BPX.DEFAULT.USER. Most qualifiers are eight characters or less in length. Unlike data sets, the names aren't restricted to alphanumeric and national characters. The class description, shown earlier, allows any character to be used in the first (FIRST=ANY) and all other positions (OTHER=ANY). The IBM Health Checker for z/OS exploits this capability by using the underscore character ('_') in resource names.

With so many system services sharing one class, a means of segregating their different resource names was needed to avoid duplicate profile names. To address this need, each system service selects a specific prefix in naming all their associated resources. Often, these prefixes match the prefix of their program names. Examples include BPX for Unix System Services and IRR for RACF.

Third-party products use the FACILITY class and IBM customer sites also may write their own in-house software using the FACILITY classes. (IBM strongly advises non-IBM users of the FACILITY classes to add a numeric or national character—such as $, #, or @—to their resource name prefixes to pre-

IBM RESOURCES

PREFIX	CALLING SERVICE	DESCRIPTOR
BBM.	CICS	CICSplex controls
BPX.	z/OS Unix	Control access to UNIX services, store default values, and activate select services
CBD.	Hardware Configuration Definition (HCD)	Control IPL and I/O configuration information
CDS.	Open Cryptographic Service Facility (OCSF)	Control which daemons call OCSF services or related processes
CSV----.	z/OS - Control dynamic changes to the system configuration	• CSVAPF. – Authorized Program Facility (APF) libraries • CSVDYLPA. – Link Pack Area • CSVDYNEX. – Exits • CSVDYNL. – Link List • CSVLLA. – Link Library Lookaside (LLA) • CSVRTLS. – Run-Time Library Services (RTLS - replaces JOBLIB and STEPLIB)
DHF----.	CICS	Control various means of binding and connecting
DITTO.	DITTO	Control use of DITTO's functions to manipulate tape and DASD datasets
HZS.	The IBM Health Checker for z/OS (using XFACILIT class profiles)	Protect checks and control changing of checks made by the Checker
ICH----. and IRR(----).	RACF (ICH are the older services and IRR the more recent)	• ICHBLP – Control use of Bypass Label Processing • IRRDPI00 – Control loading the Command Parsing Table • IRR.PASSWORD.RESET – Reset passwords without SPECIAL authority • IRR.PGMSECURITY - Set RACF program protection mode • IRR.R----(.) – RACF Callable Services functions, enabling authorized non-privileged callers to invoke privileged services
IEAABD.	Dump Services Control	Control diagnostic dumps for certain types of address spaces (APF-authorized, etc.)
IHV.	Enterprise Systems Connection (ESCON) Manager	Control the ESCON Manager and related APIs
IOA.	Open Systems Adapter Support Facility (OSA/SF)	Control OSA/SF commands and functions
IXLSTR. and MVSADMIN.	Sysplex	Control Sysplex policy administration and coupling facility structures
NJE.nodename	JES	Assign an ID to a remote node for console command authorization
RJE.workstation and RJP.workstation	JES	Workstation logon validation
STGADMIN.service	DFSMS	Control storage administration functions and privileges, also enable delegation of related authorities without OPERATIONS authority, including: • STGADMIN.ADR. – Data Set Services (DSS) • STGADMIN.ADR.STGADMIN. – DSS ADMINISTRATOR • STGADMIN.ANT. – Data Mover • STGADMIN.ARC.– Automatic Backup And Recovery Service (ABARS) • STGADMIN.ARC. – Hierarchical Storage Manager (HSM) • STGADMIN.EDG. – Removable Media Manager (RMM) • STGADMIN.IDC. – IDCAMS Catalog Management • STGADMIN.IDG. – System Managed Storage (SMS) • STGADMIN.IGG. – IDCAMS Catalog Management

vent conflicts with current or future IBM prefixes.)

Profiles

Profiles are the records in the RACF database whose names match, in whole or part, the names of the resources specified by the calling system services. Profiles in the FACILITY classes are general resources and the records are formatted accordingly. Profiles in the FACILITY classes can be used in multiple ways:

Access authorization: Access authorization is the most common use of the FACILITY class profiles. When a user attempts to perform a function such as locking an ICF catalog, the catalog system service issues a RACROUTE call to see if the user has authority to use the function. In this case, READ-level access to the resource IGG.CATLOCK is required. RACF locates the profile most closely matching this resource and interrogates the access list to see if the user has sufficient authority.

Access authorization checks both discrete and generic profiles. The use of generics assumes the class has been activated for generics via the RACF SETROPTS command. Discrete profiles are exact matches to the resource name; generics use masking characters (&racfvar, %, *, and **) to match multiple resource names. The definition of the FACILITY classes allows generic masking in any qualifier (KEYQUAL=0). RACF finds the profile most specifically matching the resource name when making the access authorization decision. (**Tip:** To find the profile protecting a specific resource, issue the command: RLIST FACILITY resource).

The minimum level of access authority needed to use a given resource is defined by the caller. Resource names generally relate to specific functions or operations and the caller wants to know if the user may perform the requested function. Usually, having READ-level access allows the request and an access level of NONE denies the request. For example, READ access to CSVDYNEX. LIST lets the user list the dynamic system exits.

Some services choose different levels of access to a particular resource to control varying levels of capability. READ, UPDATE, and CONTROL-level access to the DITTO.DISK.FULLPACK resource determines, in part, which DASD volumes a user may read or update using DITTO's utilities.

In permitting access to profiles with-

in the FACILITY classes, be careful when granting ALTER level of access to discrete profiles. In addition to allowing all other levels of authority, ALTER access also lets the user change the access list. There are very few resources requiring ALTER-level access to perform the related function.

Option activation: The existence of a specific profile causes the caller to activate or de-activate a particular feature or function. Typically, a discrete profile is required for option activation. Further, the profile's access list has no impact on use of the feature. It's the existence of the profile that prompts action. One example is the profile BPX.SAFFASTPATH. The existence of this profile causes the bypassing of normal SMF logging of successful accesses in z/OS Unix. This can be quite valuable in terms of saving the CPU time required to build and write the records and reducing the volume of SMF records written when performing maintenance on z/OS Unix. However, it also eliminates the audit trail of successful accesses.

There's an exception to the requirement for a fully qualified discrete profile in option activation. The Run-Time Library Service (RTLS) will bypass security checks to a logical library if the profile CSVRTLS.NOSECCONNECT.library.version is defined. Generic profiles can be used to bypass checks for multiple libraries or multiple versions, provided the profile has a prefix of at least "CSVRTLS.NOSEC." Anything shorter (for example, CSVRTLS.*) isn't considered a match for the NOSECCONNECT function and will be ignored.

Control information storage: A few profiles are designed to pass control information to RACF via a specific data field. Similar to option activation profiles, a discrete profile is required and the access list isn't referenced. One example is the BPX.DEFAULT.USER profile. Its APPLDATA field stores the name of a user ID and/or group name whose associated OMVS segments provide default values such as uid, gid and other parameters to users and groups needing, but not having, their own OMVS segment. Figure 3 shows an abbreviated display of this record.

With many other general resource classes, it's often wise to create a "catch-all" profile such as '**' to cover resources not more specifically defined by other profiles. This isn't the case with the FACILITY class. If a profile covers resources RJE.workstation or RJP.workstation, RACF performs the validation of RJE and RJP logons instead of JES. If not properly set up, RJE and RJP connections will be blocked. There's no minimum length prefix required when checking for these profiles, so the '**' profile would match.

Another reason for not having a catch-all profile (or backstop profile) is handling of the undefined resource return code 4 by different system services. In some cases, they deny access. If a catch-all is defined with a UACC of READ or greater, previously restricted functions may become accessible to everyone. Even if the profile has its UACC set to NONE, there's a concern, since at least one non-IBM product uses access of

Third-Party Products Listed by Profile Prefixes

Several vendors have made use of FACILITY class profiles. Find the prefix of the resource name in question to locate the vendor and product(s) likely using the profile in question.

PREFIX	VENDOR	PRODUCT
$C2R.	Consul Risk Management	zAudit, zSecure Suite
$C2F.	Consul Risk Management	zCollect, zAudit and zSecure Suite
$CNG.	Consul Risk Management	zAdmin and zSecure Suite
DSM.	Allen Software Group	ASG-Admin for Security Server
ESP.	Cybermation	ESP Workload Manager
FATAR.	Innovation Data Processing	Fast Analysis of Tape and Recovery (FATAR)
FATS.	Innovation Data Processing	Fast Analysis of Tape Surface (FATS)
IOF.	Fisher International	Interactive Output Facility (IOF)
MGCKEYS.	ASPG	Megacryption
MXICMD.	RocketSoftware	MVS Extended Information (MXI)
RA2.	Software AG	RACF Administrator 2
SPACEFINDERWORKBENCH.	TeraCloud	SpaceFinder
SQLNKACC/SQLNKAUT	Data Direct	SequelLink Server
STGADMIN.DMS.STGADMIN	CA	CA-Disk
VRA$.	Vanguard Integrity Professionals	RACF Advisor
VRAxxx$.	Vanguard Integrity Professionals	RACF Administrator
VIP$.	Vanguard Integrity Professionals	RACF Administrator
VSA$.	Vanguard Integrity Professionals	RACF Analyzer
VSR.	Vanguard Integrity Professionals	RACF Advisor
XCOM.	CA-XCOM	Data Transport
XDC.	Cole Software	Extended Debugging Controller (XDC)

From its humble beginning of just a few system services and a handful of their related resources, the FACILITY class has grown to dozens of system services and more than 100 different resources.

NONE to enable privileges.

Tip: Instead of defining ** with AUDIT(ALL) to find out which undefined resources are being used in your system, use the SETROPTS command to set LOGOPTIONS(ALWAYS(FACILITY)). This gives a comprehensive picture of FACILITY class resource checks issued, including those for undefined profiles or requested by TRUSTED started tasks. The one exception being REQUEST=FASTAUTH—which are not logged. For most sites, this setting doesn't generate a significant volume of SMF records. However, your system may differ, so activate with caution and watch for excessive SMF record counts.

When it comes to documentation, the FACILITY classes are unlike other RACF classes. Other classes tend to be used exclusively by a single system service. A few manuals provide all the information you need about a class and its resources. Conversely, a multitude of z/OS services, third-party products, and locally designed installation software use the FACILITY classes. It's necessary to search the manuals of all these individual services and products to discover how they use these classes.

FACILITY Class Resource Highlights

It's impractical within the space constraints of this article to list and describe the many FACILITY class resources and their use. The accompanying sidebars provide a brief description of many resource prefixes and their calling service to facilitate further research.

Customizations

Years ago, IBM provided several sample exits for using FACILITY class profiles for various controls not available at that time. Some of these exits are still in use at a few sites. In some cases, the profiles still exist long after the exit was removed (see Figure 4).

Summary

The FACILITY class started as a small collection of profiles for multiple applications and now fills a significant role in the control of many strategic system functions. When the need for longer names grew apparent, two new classes were created, XFACILIT and GXFACILI, to continue and expand the support. FACILITY class profiles fulfill one of three roles: access authorization, option activation, or control data storage. While there's no single reference manual for the profiles, the prefix of each profile provides a strong indication of the system service, vendor, or installation using the profiles—and suggests where to look for more detailed information. z

References

- "Understanding Your FACILITY Class Profiles," *Technical Support*, Mark S. Hahn, January 1999
- "FACILITY Class Overview," Kurt Meiser, June 1999
- "FACILITY Class," RSH Consulting, Inc., 2005 (visit www.rshconsulting.com for the latest version).

About the Authors

Robert S. Hansel is president of RSH Consulting, Inc., a firm specializing in RACF. He has worked with RACF for nearly 20 years in the roles of manager, administrator, auditor, instructor, and consultant. He supports several of the RACF User Groups throughout the U.S.
e-Mail: r.hansel@rshconsulting.com
Website: www.rshconsulting.com

Mark S. Hahn is a senior RACF consultant at RSH Consulting, Inc. He has been in the mainframe computer security field for almost 30 years. He presents at technical conferences and participates in the SHARE Security and Audit Project. He recently rekindled the Southern California RACF User Group (SCRUG).
e-Mail: m.hahn@rshconsulting.com