<>Interview notes:

05/02/2012

- Their is NO Master Console. One or more Consoles can be configured to have MASTER Authority. This is done in the CONSOLxx ParmLib Member.
- Eliminating all NIP Consoles from OSCP configurations in favor of HMC.
- Control over access HMC functions is controlled by assigned roles, but the specifics of each role may not be well understood. As for example when comparing Operator vs. System Programmers.
- Have done some investigation of the HMC Log but no systematic program is in place at this time.
- Assumption is that those that can issue Commands from the HMC are in fact VETED USERS of the resource.
- When the Dispay Console ACTIVE Command is issued the name of HCM Consoles returned is the System Name/Id and NOT the Console Name. The HMC per-say will not show-up as an active console. It appears that System Name/Id is used for logging command issued from the HMC.
- Screen display provided of output from Display Console,Active

```
NC0000000 MAI2 2012123 13:34:56.98 MAI2 00000290 D C,A,CA
LR 291 00000090 CONSOLES MATCHING COMMAND: D C,A,CA
LR 291 00000090 NAME TYPE SYSTEM ADDRESS STATUS
DR 291 00000090 MAANXOCC MCS MAI2 10A2 ACTIVE
DR 291 00000090 MACN10A0 MCS MAI2 10A0 ACTIVE
DR 291 00000090 MACR2080 MCS MAI2 1080 ACTIVE
DR 291 00000090 MFANXOCC MCS MFI3 10A2 ACTIVE
DR 291 00000090 MFCN10A0 MCS MFI3 10A0 ACTIVE
ER 291 00000090 MFCR2180 MCS MFI3 1180 ACTIVE
```

- Does the HMC enjoy some special distinction in the RACF OPER Command Class when compared to other Consoles when checking Console Authority of the console command stream. Ask Mark Nelson - RACF IBM.

<>Overview

The Hardware Management Console or HMC is an acronym frequently used to describe the IBM technology for managing and monitoring IBM mainframe (System z) or IBM UNIX (system p) servers.
The HMC uses its network connections to one or more servers or frames to perform various management functions. As defined by IBM, the HMC technology provides a standard user interface for configuring and operating partitioned (LPARS) and Symmetric Multiprocessing (SMP) systems.
The HMC enables a system administrator to manage configuration and operation of partitions in a system and multi-system complex, as well as to monitor the individual system for hardware and other operational problems.

<>Abstract - Problem Statement

The HMC, what mistakes are you making securing it?
The Hardware Management Console (HMC) is a fantastic facility that allows an installation to configure and dynamically reconfigure the LPARs in one or more zEnterprise Systems. But, the HMC can also issue any operator command you want, with no control by the External Security Manager. So, can you

- Vary a storage volume online? ΓÇô sure!
- Add an APF authorized library? ΓÇô sure!

How many people have authorized access to the HMC? 25, 50, 150? Can they

- Access it remotely?
- Do they need a Digital Certificate to do that?

It used to be that this kind of physical access was severely restricted because you had to be in the ΓÇ£Computer RoomΓÇ¥ to get to the console. But, now, this old kind of access plus the ability to change configuration and even do it remotely, is available to many.

<>Consoles

//IBM z/OS Parallel Sysplex Operational Scenarios - SG24-2079-01

The first step in planning an MVSΓäó console configuration is to define the I/O devices to MVS. Ensure that you define each I/O device that you plan to use as an MCS console with the hardware configuration definition (HCD) program for each MVS system in the installation. Use the HCD Add Device panel to define the device number and other information that identifies the device to MVS.

Normally a locally attached console is used when a system is IPLed. The console is defined as a Nucleus Initialization Program (NIP) console in the operating system configuration (IODF). If none of the consoles specified as a NIP console are available, or if there are none specified for the system, then the system will IPL using the HMC as the SYSCON console as the NIP console. If there is no working Hardware Management Console (HMC) available, then the Support Element on the processor will be used instead.

You would generally only use HMC to IPL a system if there were problems with the consoles defined with master console authority in the CONSOLxx parmlib member.

<>What are the Types of Consoles and/or Console "Masters"

z/OSMF

//z/OS Management Facility, SG24-7851-00

Currently there is no central system management portal for z/OS, and the many interfaces (including HCM) that are available are foreign to users new to platform. In most cases, manual tasks require going through extensive documentation and also require years of z/OS experience to be productive.

z/OSMF is a WEB 2.0 based application on z/OS with direct access to z/OS data and information, and a secure browser interface from the workstation. z/OSMF contains the GUIs and the application code. Everything is installed on the z/OS server. There are no client-side install requirements.

An installation with System z10┬« systems is required, you can use the zAAP on zIIP facility.

You need a security product to use z/OSMF. This can be Resource Access Control Facility (RACF) or a comparable product. z/OS resources that you want to manage with z/OSMF should be secured using the appropriate profiles in RACF.

With z/OS version 1 release 12, z/OSMF secures the access to its tasks using a role-based concept. You must ensure that the users you define in z/OSMF are also defined as RACF users in z/OS.

>>--> PRR's note: Comming to an HMC like facility near you in V2R1, or later.

The Unified Resource Manager:

//IBM zEnterprise Unified Resource Manager: Building an Ensemble, SG24-7921-00

For the first time it is possible to deploy an integrated hardware platform that brings mainframe and distributed technologies together: a system that can start to replace individual islands of computing and that can work to reduce complexity, improve security, and bring applications closer to the data that they need.

The zEnterprise Unified Resource Manager (URM) is firmware that executes on the HMC and SE. It is comprised of six management areas:

1. Operational controls (Operations) - includes extensive operational controls for various management functions.
2. Virtual server lifecycle management (Virtual servers) - enables directed and dynamic virtual server provisioning across hypervisors from a single uniform point of control.
3. Hypervisor management (Hypervisors) - enables the management of hypervisors and support for application deployment.
4. Energy management(Energy) - provides energy monitoring and management capabilities that can be used to better understand the power and cooling demands of the zEnterprise System.
5. Network management (Networks) - creates and manages virtual networks, including access control, which allows virtual servers to be connected together.
6. Workload Awareness and platform performance management (Performance) - provides management of CPU resource across virtual servers hosted in the same hypervisor instance to achieve workload performance policy objectives.

The Unified Resource Manager provides energy monitoring and management, goal-oriented policy management, increased security, virtual networking, and data management for the physical and logical resources of a given ensemble.

The zEnterprise Ensemble is a collection of highly virtualized heterogeneous systems that can be managed as a single logical entity and where heterogeneous workloads can be deployed. Some characteristics of an ensemble are:

- A zEnterprise CPC, with or without an attached zBX, is called a node.
- One ensemble can contain from one to eight nodes.
- A single node can be a member of only one ensemble.
- Dedicated integrated networks for management and data.

An ensemble is managed as a single logical virtualized system by the Unified Resource Manager, through the use of a Hardware Management Console (HMC). The HMC is used to create and manage the resources of the ensemble.

Some of the benefits of the zEnterprise and ensemble:

- Integration, monitoring, and management of multiple platform resources as a single, logical, virtualized system.
- End-to-end management capabilities for the diverse systems in the ensemble and enhanced IT platform management.
- Enterprise wide virtualization across systems, storage, networks, and applications.

- Controlled access to data and information to protect network and physical infrastructure. Energy savings with extensive monitoring of energy consumption, key environmental parameters and Integrated Energy Management Controls.

The URM is composed of and utilizes three types of LANs that attach to the zEnterprise system each with redundant connections. They are:

1. The intraensemble data network (IEDN)
2. The intranode management network (INMN)
3. The client managed data network

The intranode management network (INMN) is required for platform management within a node. The network allows the HMC to communicate to the hypervisors within the managed ensemble. An ensemble must have an INMN and it is used by Unified Resource Manager components.

URM Security - The isolation of networks from each other is a basic method of network security. In the ensemble the two internal networks, the INMN and the IEDN are physically isolated from each other.

An external client data network cannot connect directly to the INMN. The INMN is accessible only through the HMC, and the HMC is locally secured on a private LAN with authentication and authorization. If accessed remotely, the HMC is secure with firewall filtering through a secure SSL (Secure Sockets Layer) connection and further discrete authorizations. URM functions are further secured with discrete authorizations to its special functions.

The INMN is entirely private and can be accessed only through the HMC, by its connection to the SE. Standard HMC security also still applies. There are additions to Unified Resource Manager ΓÇ£role-basedΓÇ¥ security, so that not just any user can reach the Unified Resource Manager panels even if that user can perform other functions of the HMC. Very strict authorizations for users and programs controls who is allowed to take advantage of the INMN.

System/Support Element:

Regardless of what hardware is in your mainframe complex, the hardware can be managed by using either the Support Elements (SE) that are directly attached to the server. The SE is located inside of the same frame that the central processor complex is located and performs such management tasks as:

- testing the hardware before the operating system is loaded
- loading the operating systems
- concurrent repair
- concurrent upgrade
- reporting of and recovering from errors
- and other task

Hardware Management Console:

The HCM communicates with each Central Processor Complex (CPC) through the CPCΓÇÖs Support Element (SE). When tasks are performed at the HCM, the commands are sent to one or more support elements which then issue commands to their CPCs.

To ensure the integrity and security of the environment, the HMC and SE operate

on a closed platform, which means that the customer is not provided access to
the underlying operating platform and is not permitted to install or run additional
applications on the underlying operating platform that the HMC or SE
are using. The sole purpose of the HMC and SE is to provide a platform for the
execution of the HMC and SE application.
One HMC can control up to 100 SE and one SE can be controlled by 32 HMCs.
Resource access and Management functions that can be performed from the
HMC Include the following interface access points:

- HMC Administration
- z and/or p Server Management
- Image/LPAR Management
- ESCON Directors
- Sysplex Timers
- Fiber Savers
- Other Resources

> Built in HCM Security

Evaluation Assurance Level 5 (EAL5) is the System z grade (EAL1 ΓÇôEAL7)
that the System z achieves after completing ongoing assessments, by an
independent authority, against an international standard call Common
Criteria. The purpose of this standard is to ensure that systems-security
features are reliably implemented. These features inclued the following:
1 - The HMC can only communicate with CPC support elements that have the
same domain name and domain password as the HMC. Assigning a unique
domain name and password to a HMC and the CPCs that are defined to
it will isolate those CPCs from any other HMCs connected to the same
Local Area Network (LAN).
Domain Security Data Security - defined as the definitions (domain name
and password) for your HMC and CPC support elements in your processor
complex. Individual remote users can be configured to have restricted
access in the same way as they would be configured on a local HMC.
2 - Logon security for a web browser is provided by the HMC user logon
procedures. Certificates for secure communications are provided, and
can be changed if desired.
3 - HMC and SE log the following:
□□
a) Local and remote accesses by user ID and if a remote access, then
by IP address also
b) Disruptive actions by user ID
c) All profile and configuration changes (including what was changed)
by user ID
Event Log - The event log function tracks all system events, which
are individual activities that indicate when processes occur, begin,
end, succeed, or fail. When an event occurs, the date and time that
it occurs and a brief description of the event is recorded in the
event log. Through the Monitor System Events task, you can create
and manage event monitors. Event monitors listen for events from
objects that the Hardware Management Console manages.
There are three types of events:

- Hardware messages
- State changes
- Operating system messages.

Task Log - The task log function tracks the last performed 1000 tasks,
which includes the objects of the task. It also tracks how many times
each task was used and the date and time of the last use, for example,
the HMC has a workplace that supports functions, facilities, and controls
that monitor and operate different entities, such as servers,
logical partitions, groups of servers, and logical partitions.
4 - You can copy a security log in ASCII format to a DVD-RAM. By

offloading the security log to the DVD-RAM, you can use any ASCII editor to view the entire log or print a hardcopy of the log for reference.

The HMC should be considered an appliance, whose purpose is to provide management and control of System z┬« servers and associated operating systems and devices.

- The HMC is an orderable feature of a System z┬« server. At least one HMC
is required in order for all the capabilities of a System z┬« server to be fully operational. The HMC feature consists of a standard PC hardware platform, which includes a keyboard, mouse, and display; along with preloaded
Licensed Internal Code that is used to perform its various functions.
- The HMC is a closed platform. Specifically this means that the customer is not given access to the underlying operating platform and is not allowed
to install and run other applications on the HMC. All configuration of the HMC is accomplished using tasks provided by the application.
- The HMC is intended, and required, to be a network attached device, since
this is the path the HMC uses to communicate with the various System z┬« resources. The HMC Licensed Internal Code application provides the controls
used to configure the network for the HMCΓÇÖs use. It is expected that the HMC is tested for network security using the normal procedures including
periodic network scans, etc.
- The HMC hardware is not serviced by the end user; IBM service personnel are
responsible for this task.
- The HMC is not an operating platform that is directly usable by the end user;
the HMC Licensed Internal Code, which provides the HMC application, is the only feature of the HMC that is to be used by the end user.

You can operate a z/OS system or an entire sysplex using the operating system
message facility of the Hardware Management Console (HMC). This facility is
also known as the SYSCONS console and is considered an EMCS type of console.

You would generally only use this facility if there were problems with the consoles defined with master console authority in the CONSOLxx parmlib member.

The procedures for using the HMC to operate z/OS are not unique to a sysplex.

See also then Hardware Management Console Operations Guide, SC28-6837

Remote Hardware Management Consoles:

//System z:Hardware Management Console Operations Guide, SC28-6857-01

Each Hardware Management Console contains a web server that can be configured
to allow remote access for a specified set of users. A web browser gives an
enabled user access to all the configured functions of a local HCM, except for those functions that require physical access. Such a web base configuration
might be an off-hours monitor from home by an operator or system programmer.

A remote HCM has the same function, setup and maintenance requirements as a local HCM Console. A remote HCM needs LAN TCP/IP connectivity to each managed object (Support Element) that is to be managed. Firewall that may exist between the remote Hardware Management Console and its managed objects must permit HCM to SE communications to occur.

Security for a remote Hardware Management Console is provided by the Hardware Management Console user logon procedures in the same way as a local Hardware Management Console. As with a local Hardware Management Console, all communication between a remote Hardware Management Console and each Support Element is encrypted. Certificates for secure communications are provided, and can be changed by the user if desired

System Console:

In this context, the term hardware (or system) consoles refers to the interface provided by the Hardware Management Console (HMC) on an IBM System z processor. It is often referred to as the System Console (SYSCONS).

The System Console is considered an EMCS Console that has been specify by DEVNUM(SYSCONS) in the CONSOLxx Parmlib Member.

MCS Consoles:

MCS consoles are display devices that are attached to a z/OS system to provide communication between operators and z/OS. MCS consoles are defined to a local non-SNA control unit (for example an OSA Integrated Console Controller, or 2074). Currently you can define a maximum of 99 MCS consoles for the entire Parallel Sysplex.

MCS consoles are locally attached to the system through control devices that do not support Systems Network Architecture (SNA) protocols.

EMCS Consoles:

Extended MCS consoles are defined and activated by authorized programs acting as operators. An extended MCS console is actually a program that acts as a console.

SMCS Consoles:

SMCS consoles are MCS consoles that use z/OS┬« Communications Server SNA and TCP/IP services for input and output. SMCS consoles provide most of the same functions as MCS consoles with the following exceptions:

- Synchronous WTO/R, also known as disabled console communication facility (DCCF), is not supported for SMCS consoles. The system console or an MCS console must be used instead.
- SMCS consoles are not available during NIP. The system console or an MCS console must be used instead.
- z/OS Communications Server must be active for SMCS to be active. The system console and MCS consoles do not rely on z/OS Communications Server, and these can be used before z/OS Communications Server is active.
- SMCS consoles must be activated differently than MCS consoles. The activation process depends on the console definitions, but in all cases, VARY CONSOLE and VARY CN, ONLINE do not work for SMCS.
- SMCS does not support output-only (message stream and status display) consoles. SMCS consoles must always be full-capability consoles.
- SMCS does not support printer consoles.

Because an SMCS console is connected through a network and uses z/OS Communications Server services, the z/OS Communications Server commands VARY NET and HALT NET, as well as network problems, can affect console operations. SMCS consoles are not defined to HCD.

Note - DEVNUM Specify SMCS on the CONSOLE statement.

NIP Consoles:

The Nucleus Initialization Process console is MCS(System or HCM Console) whose device address is defined in the NIPCON Statememt in the IODF being used during the system IPL.

<>Using the HMC to IPL a System

To use the SYSCONS console on the HMC, you must select the Operating System Messages (OSM) task and the appropriate system on the HMC. The HMC will open a window which will be the SYSCONS console for the system. During an IPL process, the messages are automatically displayed on the SYSCONS console. If there are any replies required during the NIP portion of the IPL, the operator can reply using the Respond button on the window.

If you need to use the SYSCONS console for command processing, you can use the Send button to send a command to z/OS. You must first enter the

```
VARY CN(*),ACTIVATE
```

command to allow the SYSCONS console to send commands and receive messages and is considered to be in Problem Determination (PD) mode.

This command can only be entered at the SYSCONS console. If you try to enter any other z/OS command prior to this command, you receive a reply stating that you must enter the VARY CONSOLE command to enable system console communications.

Messages will now displayed from systems as specified in the MSCOPE parameter for the SYSCONS.

Almost any z/OS command can now be entered, with a few restrictions.

If there is no response to the command, it may indicate that the system is not active, or the interface between z/OS, or the Support Element (SE) is not working or the ROUTCDE setting on the SYSCONS is set to NONE.

The SYSCONS console for a system may be accessed on multiple HMCs, you do not have to issue the VARY CONSOLE command on each HMC. It only needs to be entered once for the system. It remains active for the duration of the IPL, or until the

```
VARY CN,DEACT
```

command (to deactivate the system console) is entered.

To display the SYSCONS console status for all systems in the sysplex, use the DISPLAY CONSOLE command

```
D C,M
```

A typical reply (for a single console in this example) would appear as follows:

```
CONSOLE ID --------------- SPECIFICATIONS ---------------
AAILSYSC COND=A,PD AUTH=MASTER
SYSCONS MFORM=M LEVEL=ALL,NB
AAIL ROUTCDE=NONE
CMDSYS=AAIL
MSCOPE=*ALL
AUTOACT=--------
INTIDS=N UNKNIDS=N
```

For each system that is active or has been active in the sysplex since the

sysplex was initialized, there is a SYSCONS console status displayed, along
with all the other consoles inthe sysplex.
The COND status for the SYSCONS has three possible values. They are:
- A The system is active in the sysplex, but the SYSCONS is not available.
The status of A indicates that the system has been IPLed, but there has not
been a VARY CONSOLE command issued from the SYSCONS for the system.
- A,PD The SYSCONS is available, and in Problem Determination mode.
The status of A,PD indicates that the system is IPLed, and there has been a
VARY CONSOLE command issued from the SYSCONS for the system.
- N The system is not active in the sysplex, therefore no SYSCONS is available.
The status of N indicates that the system associated with the SYSCONS is not
active in the sysplex. The console appears in the list because the system
had been active in the sysplex. It could also indicate that the interface
between z/OS and the SE is not working.
<>Defining RACF Profiles
//z/OS V1R11.0 MVS Planning Operations z/OS V1R10.0-V1R11.0 - SA22-7601-11
To determine whether a particular user (an operator) is allowed to access
a particular resource (a command or a console), security profiles are used.
The security administrator can define a security profile for:
- Each user of a console
- Each console that is to be automatically logged on
- Each MVSΓäó command issued from a console
- Each user of the SMCS application that is able to enter a command.

SMCS will support the protecting of the SMCS application via the APPL class
of a security product. If the user is defined and authorized by the security
product and the APPL class is not active or the APPL class is active but no
profile matches the SMCS APPLID, access will be granted. If the APPL class
is active and a profile matching the SMCS APPLID exists, the name the user
is logging on with must be defined in the profile's access list with at
least READ authority for access to be granted. If the console has been
defined with LOGON(AUTO), the console name must be in the access list.
Using RACF┬« to authorize commands means that each operator requires an
individual user profile. (TSO/E users of extended MCS consoles should
already have a security profile in order for them to log on to TSO.) This
user profile establishes the userid of the individual operator, and the
userid identifies the operator when the operator logs on to the system.
You can define the operator's or TSO/E user's authority to access resources
by userid, but you can also establish access authority through a security
group. For example, if you have several operators or TSO/E users with
identical access requirements, you can have the security administrator
create a security group and define the access for the individual operators
or TSO/E users through the group.
If you want an MCS console to be automatically logged on when you specify
LOGON(AUTO), you must ensure that each console has a user profile
established for it. Your security administrator can define a user profile
by console name. When LOGON(AUTO) is in effect, the console is automatically
logged on when it is activated.
Resources, such as commands, MCS or SMCS consoles, and TSO terminals,
also require security profiles. These profiles establish the access

requirements for the resource ΓÇö such as who can issue the command or use the console or terminal ΓÇö and the level of security auditing your installation requires. For example, you might need to audit all uses of commands or want to audit only unauthorized uses of commands. For specific information using RACF, see Defining Commands with RACF and Defining Consoles with RACF. For an example of defining a TSO/E terminal as a resource, see Controlling Extended MCS Consoles Using RACF.

You need to work with the security administrator to set up the security profiles and options to implement your installation's security goals. z/OS Security Server RACF Security Administrator's Guide includes RACF-related information about securing access to system commands and consoles.

RACF Access Authorities

In RACF profiles that protect resources, the MCS authority ΓÇ£translatesΓÇ¥ to a RACF access authority. This RACF access authority is specified for a user or console in an access list of the resource profile and determines the command authority of the user or console.

MCS Authority	RACF Access Authority
MASTER	CONTROL
ALL(SYS,IO,CONS)	UPDATE
INFO	READ

These access authorities are the same for extended MCS console users. The security administrator can define resource profiles for MCS, SMCS and extended MCS consoles using RACF commands. (See Controlling Extended MCS Consoles Using RACF.)

Defining Users with RACF

Your installation's security policy determines how you define the operators, MCS consoles, or SMCS consoles for automatic logon. If your installation's security policy requires you to audit all operator commands according to the identity of the user, then all operators must be defined as individual users. If your installation uses the LOGON(AUTO) option in CONSOLxx to automatically log on MCS and SMCS consoles when they are activated, you must ensure that a user profile exists for each console to be logged on.

You can also grant access to commands to groups of operators. A RACF group defines a set of related individuals who have similar security requirements.

Defining access authority by group minimizes changes to the RACF profiles when individual users change job responsibilities or leave a particular job.

To create profiles for operators, the RACF security administrator needs to know

- Who the operators are
- Which operators fall into groups with identical access requirements.

To create profiles for consoles to be automatically logged on, the RACF security administrator needs to know the names of the consoles defined in CONSOLxx.

Changes made to the access authority while a system is running may not take effect until the security data for the console(s) is reset in MVS. This occurs during LOGON for MCS or SMCS consoles and during MCSOPER ACTIVATE for EMCS consoles. For instance, if an active user is connected to a new group, the user must log off and then log back on again to have the authority associated with that new group.

Defining TSO/E Users of Extended MCS Consoles with RACF

Your TSO or RACF security administrator should define user profiles for all

TSO/E users of extended MCS consoles. TSO/E logon can be controlled through TSO/E or RACF, and like operators, you can define TSO/E users by individual or group profiles. Your installation authorizes the TSO/E user to be able to issue the TSO/E CONSOLE command. This command initiates an extended MCS console session. For an example of how to define a TSO/E user to initiate an extended MCS console, see Controlling Extended MCS Consoles Using RACF.

Defining Commands with RACF

Your installation's security policy determines which commands you must protect. A RACF profile for the command in the OPERCMDS class protects the command. When an operator logs on to a console and issues an MVS command that requires a higher authority than the console allows, RACF can check the access list of the command profile to determine if the user is authorized to issue the command.

To link the command the operator issues with the profile that protects the command, MVS provides a construct, or structure, called a resource-name for each command.

The resource-name for an MVS command has the following parts:

MVS.command.command-qualifier.command object

where:

- MVS

Is the high-level qualifier that defines the command as a system command. MVS is a required part of the resource-name. Subsystem commands use a different high-level qualifier, such as JES2 or JES3.

- command

Specifies the command or a specific variation of the command. To protect an individual command, this part of the resource-name is required. It also allows you to control significant variations of a command separately. For example, FORCE without the ARM operand has a different effect than does FORCE with the ARM operand; you can thus specify either FORCE or FORCEARM to control the two uses separately.

- command-qualifier

Specifies a subfunction of the command. This part of the resource-name is optional. It allows you to protect specific command subfunctions separately.

For example, the following resource-name protects all functions of the TRACE command:

MVS.TRACE.**

In contrast, the following resource-names protect each function of the TRACE command separately:

MVS.TRACE.ST
MVS.TRACE.MT
MVS.TRACE.CT
MVS.TRACE.STATUS

- command-object

Specifies the object or target of the command. This part of the resource-name is optional. Examples of objects or targets include:

The device on a CANCEL command
The jobname on a MODIFY command
The membername on a START command

MVS Commands, RACF Access Authorities, and Resource Names in z/OS MVS System Commands defines the MVS commands and their corresponding resource-names. It also shows the RACF access authority associated with each command. To define resource profiles for system commands, the RACF security administrator can use the resource-names exactly as shown in MVS Commands, RACF Access Authorities, and Resource Names, or replace the optional fields with asterisks or, for command-object, specific values. In the command profile, the security administrator also defines the auditing requirements and the users or groups allowed to issue the command in the profile's access list.

When an operator issues an MVS command with a RACF profile, MVS determines the resource-name that matches the command and passes that resource-name to RACF. RACF uses the resource-name to locate the profile for the command and verifies that the operator is allowed to issue the command by checking the access list in the profile. If RACF authorizes the access, MVS processes the command; if RACF denies the access, MVS rejects the command. If your installation has user-written commands that you must protect, use the CMDAUTH macro.

To create profiles for MVS system commands that you do not have to change frequently, it is a good idea to end each name with two asterisks, which indicate that the profile protects all commands that match the specified portion of the resource-name, regardless of whether there are additional qualifiers or how many additional qualifiers there are. For example, use:

```
MVS.SET.**
```

to protect all SET commands with a single profile.

Defining Consoles with RACF

You can use a RACF profile in the CONSOLE class to determine which userids are authorized to log on to a particular console. The commands in the following example define a RACF profile for console CON1 (CON1 is defined in CONSOLxx), and authorize userid CONSID1 to log on to that console.

```
RDEF CONSOLE CON1 UACC(NONE)
PERMIT CON1 CLASS(CONSOLE) ID(CONSID1) ACCESS(READ)
SETROPTS CLASSACT(CONSOLE)
```

Setting DEFAULT LOGON Requirements for MCS and

Once you have established the RACF profiles your installation requires, you use the LOGON keyword on the DEFAULT statement in CONSOLxx to establish your MCS console operator LOGON requirements. You can:

- Have the system automatically log each console on as the console is activated. Operators can log on but are not required to do so.
- Require each operator to log on to the system before issuing commands.
- Allow MCS console command authorization to control access to commands.

To control how operators can log on to MCS consoles, use the following keyword on the DEFAULT statement in CONSOLxx:

//MVS Initialization and Tuning Reference - V1R13, SA22-7592-22

LOGON

Controls the logon for operators of MCS or SMCS consoles

Options you can specify for LOGON are as follows:

- AUTO Specifies that the console is automatically logged on by its console name. In addition, operators can optionally log on to the console.
- REQUIRED Specifies that operators must log on before the system allows them to enter commands. If a system includes SMCS consoles, LOGON(REQUIRED) is recommended.
- OPTIONAL Specifies that operators can optionally log on to the console; otherwise, MCS console authority is in effect.

The LOGON keyword affects only full-capability display consoles. It does not prevent the operator from receiving synchronous messages. However, the LOGON keyword setting may prevent the operator from receiving synchronous write-to-operator-with-reply (WTOR) messages. For details, ee individual topics about LOGON values: Automatic LOGON, Required LOGON, and Optional LOGON.

Regardless of the LOGON value set on the DEFAULT statement, individual consoles can override the value. For more information, see Setting LOGON Requirements for Individual MCS or SMCS Consoles.

Setting LOGON Requirements for Individual MCS or SMCS Consoles

With z/OS┬«, the LOGON keyword on the CONSOLE statement in CONSOLxx can override the console LOGON default on the DEFAULT statement.

To control how operators can log on to specific MCS or SMCS consoles, specify the following keywords on the CONSOLE statement in CONSOLxx:

- LOGON

Controls the logon for operators of MCS and SMCS consoles

Options you can specify for LOGON are as follows:

- AUTO Specifies that the console is automatically logged on.
- REQUIRED Specifies that the console must be logged on before commands can be issued.
- OPTIONAL Specifies that the console does not need to be logged on.
- DEFAULT Specifies that the console is to use the LOGON value on the DEFAULT statement. If you specify DEFAULT and the DEFAULT statement does not contain a LOGON value, the system issues an error message and uses LOGON(OPTIONAL) for an MCS console and LOGON(REQUIRED) for SMCS.

For an SMCS console, see Defining SMCS Consoles.

Automatic LOGON

To control and audit command activity by console, specify LOGON (AUTO).

When LOGON (AUTO) is in effect and RACF is active, the system automatically issues a LOGON for each MCS or SMCS console as the console is activated. The automatic LOGON uses the console name as the logon userid.

To ensure that the console is automatically logged on, the security administrator must define a user profile for each console by console name. Your installation must define the name of the system console as a valid USERID to RACF. IBM┬« recommends that if you plan to use LOGON (AUTO) for your installation, you define the system console in CONSOLxx and do not use the system default name as the name of the system console.

To define access requirements for the console, the security administrator defines a resource profile for the console in the RACF CONSOLE class. The CONSOLE class must be active when console resource profiles are used.

When automatic LOGON is in effect, operators can log on to the system but are not required to do so. The system issues an automatic LOGON for the console whenever RACF is active and the following conditions occur:

- The console is activated either during system initialization, as a result of the VARY command or if an SMCS console is logged on.
- The console is switched from message-stream or status display mode to full capability mode.
- An operator who had logged on issues the LOGOFF command.

Once the console is logged on, operators can use it to issue commands at the level defined for the userid. This could be the level defined in the OPERCMDS class for the userid, or lacking an OPERCMDS definition matching the command, the authority of the console (originally defined in CONSOLxx).

If you have some consoles, perhaps those not in secure areas, that you want to require LOGONs, LOGON (AUTO) and RACF profiles allow you to control operator logon. If an operator wishes to issue a command requiring a higher level of authorization, and the operator (through RACF checking of OPERCMDS profiles) has the required level of authorization, the operator must log on to the console to be able to issue the command successfully. The operator authority (defined in the OPERCMDS class) then replaces the console authority. When the operator logs off, the system automatically issues the LOGON for the console name, thus reverting back to the original console authority.

When using LOGON(AUTO), you should ensure that at least one operator is logged on with master authority to be able to communicate with the system. Synchronous WTORs can be displayed on LOGON(AUTO) consoles only after the consoles have been logged on.

Required LOGON

To audit all command activity by operator userid or to control which commands individual operators may issue, specify LOGON(REQUIRED) on either the CONSOLE statement or the DEFAULT statement. Specifying LOGON(REQUIRED) is especially important for SMCS consoles. Before setting LOGON(REQUIRED), your installation must define RACF profiles for all operators and for the commands and consoles you want to protect. When protecting commands and consoles with RACF resource profiles, both the OPERCMDS and CONSOLE class must be active. Also, before setting LOGON(REQUIRED), your installation must define the name of the system console as a valid USERID to RACF. IBM recommends that, if you plan to use LOGON(REQUIRED) for your installation, you define the system console in CONSOLxx and do not use the system default name as the name of the system console.

When LOGON(REQUIRED) is in effect, all operators must log on before issuing commands, and your installation can limit the commands they can issue. If an operator tries to issue a command without logging on, the system rejects the command and issues a message. The system also rejects any command the operator is not authorized to issue. To change LOGON(REQUIRED) on the DEFAULT statement, you must re-IPL the system. You can use the VARY CN command to change LOGON(REQUIRED) on the CONSOLE statement.

During system initialization, the system accepts commands only from a master authority console (or the system console) until RACF is fully initialized and able to process LOGON requests. Allowing commands from a master authority console before RACF is fully initialized allows an operator to intervene if required to complete RACF initialization.

Once RACF is initialized, the LOGON prompt appears on all MCS display consoles. The LOGON prompt requires the operator to log on by supplying at least a userid and password. The LOGON prompt also appears:

- When a console changes from status display or message stream to full capability
- When the console is brought on line by a VARY command
- When an SMCS console is activated
- When the current operator logs off
- When LOGON(REQUIRED) is in effect

No operator should leave the console unattended without first issuing the LOGOFF command. Issuing LOGOFF leaves the console in a secure, unattended state. For an MCS console, messages continue to appear on the console, but the system does not accept any command from that console until an operator logs on to the console. For SMCS consoles, the console session is terminated.

When using LOGON(REQUIRED), you should also ensure that at least one operator is logged on with master authority to be able to communicate with the system.

Synchronous WTORs can be displayed on LOGON(REQUIRED) consoles only after the consoles have been logged on.

Optional LOGON

If you do not need special command auditing, you can specify LOGON(OPTIONAL). LOGON(OPTIONAL) allows console command authorization (defined by AUTH on the CONSOLE statement) to determine whether the system is to accept the command being issued on the console.

Synchronous WTORs can be displayed on LOGON(OPTIONAL) consoles that are active, even if the consoles are not logged on.

CONSOLxx Parmlib Member Special Considerations:
CONSOLE DEVNUM Must re-IPL Identifies SMCS, SYSCONS, subsystem, or
the 3-digit or 4-digit device number for the MCS console
CONSOLE UNIT Must re-IPL Defines the unit device for the MCS console
CONSOLE NAME Must re-IPL Defines the console name
CONSOLE AUTH OPERPARM AUTH VARY CN,AUTH Defines command groups or
authority
INIT CMDDELIM Must re-IPL System Defines the command delimiter for
entering multiple messages on MCS consoles
DEFAULT LOGON Must re-IPL Specifies default LOGON attributes for MCS
and SMCS consoles
DEFAULT HOLDMODE Must re-IPL System Specifies whether the operator can
freeze the display on MCS console screens
DEFAULT ROUTCODE Must re-IPL System Assigns routing codes for messages
without a target console
The NIP console can be a 3270 that goes away if it is not also defined in
CONSOLxx
CONSOLxx is closely related to the Add Device panel of hardware
configuration
definition (HCD). The device number you specify in CONSOLxx must match the
device number on the panel.
ALLOWCMD(Y|N) ΓÇô Added V1R13
Specifies when the system console is able to issue commands.
Y = The system console is permitted to issue commands even if it is not
running
in problem determination (PD) mode.
N = The system console is restricted from issuing commands until it enters
PD
mode. This is the default value.
An extended MCS console is a program that acts as a console. It can issue
MVSΓäó
commands, and receive command responses, unsolicited message traffic, and
the
hardcopy message set. There are two ways to use extended consoles:
- Interactively through IBM┬« products such as TSO/E and Netview.
- Through an application program that you write. Examples of application
program uses are:
1) Receiving automated message traffic
2) Defining a unique presentation service for messages to consoles
To establish a program as an extended MCS console, the program must issue
the MCSOPER macro. Once activated as an extended MCS console, a program
can
receive messages and command responses by issuing the MCSOPMSG macro, and
can issue commands by issuing the MGCRE macro. Unlike a standard MCS
console,
the extended MCS console can control which command responses it receives.
You must have a CONSOLE statement for each device that you want to use as
a
console.
- Use the Add Device panel in hardware configuration definition (HCD)
to specify the device number and the unit type for MCS consoles only.
- SMCS consoles require the terminals to be defined for use via z/OS
Communications Server.
All consoles, except for the system console, require a console name to be
specified. If no name is specified, the console definition is rejected.
For
the system console, if a name is not specified, a name is generated by the
system.
A console defined to more than one system can be active on only one system
in

the sysplex at a time. If different attributes for the same console are defined in separate CONSOLxx members on different systems:

- In shared mode, the console attributes defined in the first active system in the sysplex take effect.
- In distributed mode, the same console can have different attributes on different systems.

CNZ_Console_Operating_Mode ΓÇô Medium ΓÇô Once at startup

Identifies installations running in 'Shared' console service operating mode. 'Distributed' mode is the preferred mode of operations and 'Shared' mode will be removed in a future release.

DEFAULT statement: CONSOLxx

If you do not code the DEFAULT statement, MCS consoles do not require LOGON.

SMCS consoles will require logon.

CON=NONE: IESASYSxx

If no CONSOLxx member is supplied, the system messages go to the system console and the system uses the defaults for the other values in CONSOLxx. IBM strongly suggests that you specify a name for the system console in CONSOLxx. Select a unique name for the system console that cannot be confused with a valid device number.

The CONSOLE Keyword

indicates the beginning of a statement that defines the characteristics of a console.

```
- DEVNUM {(devnum) }
{(SUBSYSTEM) }
{(SYSCONS) }
{(SMCS) }
```

DEVNUM specifies the type of console. DEVNUM is required and must be the first keyword on the CONSOLE statement.

- devnum must be the same as the number that was specified for the device on the Add Device panel in HCD.

Notes:

1. The system pins UCBs for console devices defined in CONSOLxx at IPL time, and the UCBs are only unpinned when a console definition is removed using the console removal definition service. This means that to delete a console device with HCD, an IPL is required, unless the console definition is deleted using IEARELCN.

Value Range: 1 to 4 hexadecimal digits

- SUBSYSTEM indicates that this console is reserved for subsystem use.
- SYSCONS indicates that this console is the system console attached to this processor. The use of the SYSCONS keyword is optional. The first time you put the system console into problem determination (PD) mode (by issuing VARY CN(*),ACTIVATE from the system console), its attributes will be taken from the CONSOLE statement with DEVNUM(SYSCONS). If there is no such statement, a default set of attributes will be used.

```
- AUTH {(MASTER) }
{(INFO) }
{([SYS][,IO][,CONS])}
{(ALL) }
```

AUTH specifies the group of operator commands that can be entered from the console. IBM strongly suggests using a security product, such as RACF, to control commands instead of using AUTH, especially with SMCS.

- MASTER indicates that this is a console with master-level authority. From a console with master authority, you can enter all MVS operator commands.
- INFO specifies that any informational commands can be entered from this console.
- SYS specifies that system control commands and informational commands

may be entered from this console.

- IO specifies that I/O control commands and informational commands may be entered from this console.
- CONS specifies that console control commands and informational commands may be entered from this console.
- ALL specifies that information, system control, I/O control, and console control commands may be entered from this console.

CMDSYS {(sysname)}

{(*) }

- sysname indicates the system (in the sysplex) where commands entered on this console are to be sent for processing.
- An asterisk (*) indicates that commands entered on this console are to be processed on the system where this console is defined.

Note: A value of * is recommended for SMCS consoles since the console is not tied to a system, and the use of z/OS Communication Server Generic Resources may place the console on different systems at different times. The default is (*).

LOGON {(REQUIRED)}

{(OPTIONAL)}

{(AUTO)}

{(DEFAULT)}

Use this optional parameter to override the LOGON value specified on the DEFAULT statement (if any). The system treats the system console as LOGON(OPTIONAL) no matter what LOGON value is specified on the DEFAULT statement.

IBM suggests that SMCS consoles be LOGON(REQUIRED), either by the system-wide specification on the DEFAULT statement or by the individual CONSOLE statement.

- REQUIRED specifies that an operator must log on to a console before issuing commands from that console. For MCS consoles, commands may be issued without the operator logging on under the following condition:
- When issuing commands from a console with master authority before a security product is active.

If an operator is not logged on to the console, the system rejects commands issued from that console.

- OPTIONAL specifies that the operators can optionally log on to the console.
- AUTO- specifies this console is automatically logged on when the console is activated. The userid will be the console name.
- DEFAULT specifies that this console will use the LOGON specification on the DEFAULT statement.

ALLOWCMD

ALLOWCMD specifies when the system console is able to issue commands.

- (Y) The system console is permitted to issue commands even if it is not running in problem determination (PD) mode.
- (N) The system console is restricted from issuing commands until it enters PD mode. This is the default value.

Notes:

1. ALLOWCMD is only valid when DEVNUM is SYSCONS.
2. If there are no entries with DEVNUM(SYSCONS), the system uses the default ALLOWCMD(N).

The IBM Health Checker for z/OS - Console Checks

//IBM Health Checker for z/OS V1R13 User's Guide V1R13, SA22-7994-12

Consoles checks (IBMCNZ)

- CNZ_AMRF_Eventual_Action_Msgs
- CNZ_Console_MasterAuth_Cmdsys
- CNZ_Console_Mscope_And_Routcode
- CNZ_Console_Operating_Mode
- CNZ_Console_Routcode_11
- CNZ_EMCS_Hardcopy_Mscope
- CNZ_EMCS_Inactive_Consoles
- CNZ_OBSOLETE_MSGFLD_AUTOMATION
- CNZ_Syscons_Allowcmd
- CNZ_Syscons_Mscope
- CNZ_Syscons_PD_Mode
- CNZ_Syscons_Routcode
- CNZ_Task_Table

SIDE NOTE: bit.listserv.ibmtcp-l
A nice feature of OSA-ICC is that the OSC CHP can be shared across LPARs
on
the same CP. Thus one OSC port may replace several 3174s. And one
workstation
with multiple TN3270E sessions could act as NIPCONS and MCS consoles for
your
whole environment. Although that falls into the category of "all of your
eggs
in one basket".
Rob Mergner
Independence Blue Cross
May 10, 2006 9:01AM
SIDE NOTE: RACF-L@LISTSERV.UGA.EDU
You can issue any RACF command from ANY console IF you first stand up the
RACF address space.
RACF commands still require system special or in other words just because
you're using a racf command from master mcs console for example does not
mean
someone without the proper RACF authority can issue an ADDUSER command.
I would highly recommend doing this for recoverability purposes. I've been
there when someone makes a bad mistake with a RACF profile that prevents
an
ipl. Then you'll need someone with system special to login to a console to
correct the situation. But you can't do that if you don't run the RACF
address space. Why not just use TSO you say? Well consider the scenario
where the ipl can NOT get far enough for TSO to even start.
So when replying to the operator prompt from rvary switch or rvary act
commands if you reply from a master console then the password is not
needed.
And this is by design for a worse case recovery situation.
Sincerely,
Joel Tilton
April 12, 2012 3:33:54 AM PDT
SMP
Short for Symmetric Multiprocessing, a computer architecture that provides
fast performance by making multiple CPUs available to complete individual
processes simultaneously (multiprocessing). Unlike asymmetrical
processing,
any idle processor can be assigned any task, and additional CPUs can be
added
to improve performance and handle increased loads. A variety of
specialized
operating systems and hardware arrangements are available to support SMP.
Specific applications can benefit from SMP if the code allows
multithreading.
SMP uses a single operating system and shares common memory and disk
input/output resources. Both UNIX and Windows NT support SMP.