Managing z/OS guests on z/VM with IBM Operations Manager for z/VM

Version Date: September 2020

Emily Hugenbruch

Senior Software Developer, z/VM

Walter Church IV

Staff Software Developer, z/VM

Special Notices

This document reflects the author's understanding of z/OS and Operations Manager for z/VM program products. This document is presented "As-Is" and IBM does not assume responsibility for the statements expressed herein. It reflects the opinions of the author, based on her experiences. If you have questions about the contents of this document, please contact the author.

Trademarks

Special thanks to the following people who contributed information to this effort:

- Tracy Dean
- Alan Altmark
- Alan Kline

Problem statement:

Operations Manager is a systems management and automation tool used by many z/VM customers. One of its greatest advantages is the ability to monitor consoles and send commands to guests. While this works well with Linux and TPF guests, z/OS guests pose special challenges to Operations Manager, which relies on the Single Console Image Facility (SCIF). Most users know SCIF via the `CP SET OBSERVER`, `SET SECUSER`, and `SEND` commands.

This paper will describe how to configure z/OS such that the master console is viewable and usable by Operations Manager. It will further discuss the Operations Manager function that enables it to use the z/OS master console. Finally, there is a discussion of problems encountered and some suggested ways to use Operations Manager to control z/OS guests.

Line Mode and The Integrated Console

The method to allow Operations Manager to be able to watch a z/OS guest's console is to have the guest go into line mode. For z/OS this can be done via the virtual integrated console, a simulation of the Operating System Messages task on the HMC.

This picture shows how different consoles may be configured for a z/OS guest:

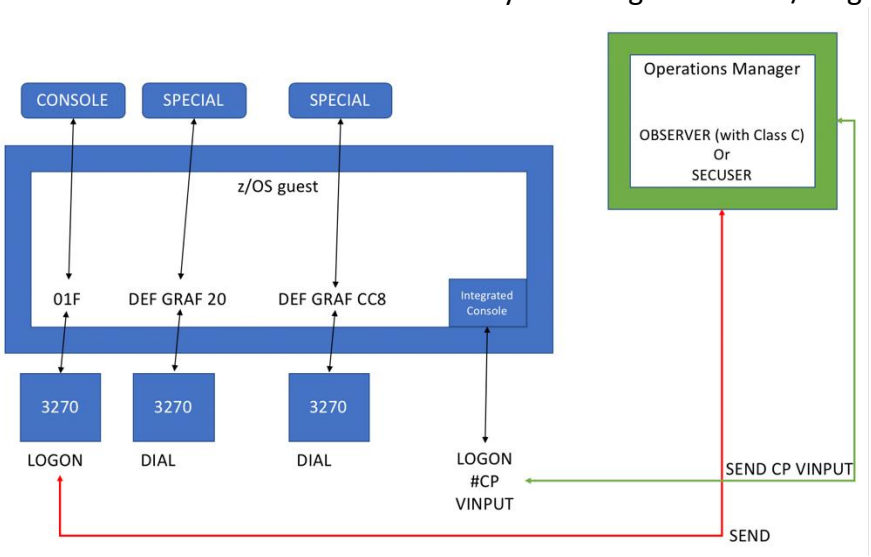


Figure 1 - Console Layout of a z/OS Guest and Operations Manager

As seen in the figure above, only commands sent with the `CP VINPUT` prefix will go to the integrated console where z/OS can receive them. The `CP SEND` command does not work because it can communicate only with a virtual 3215 console, which z/OS does not use or support.

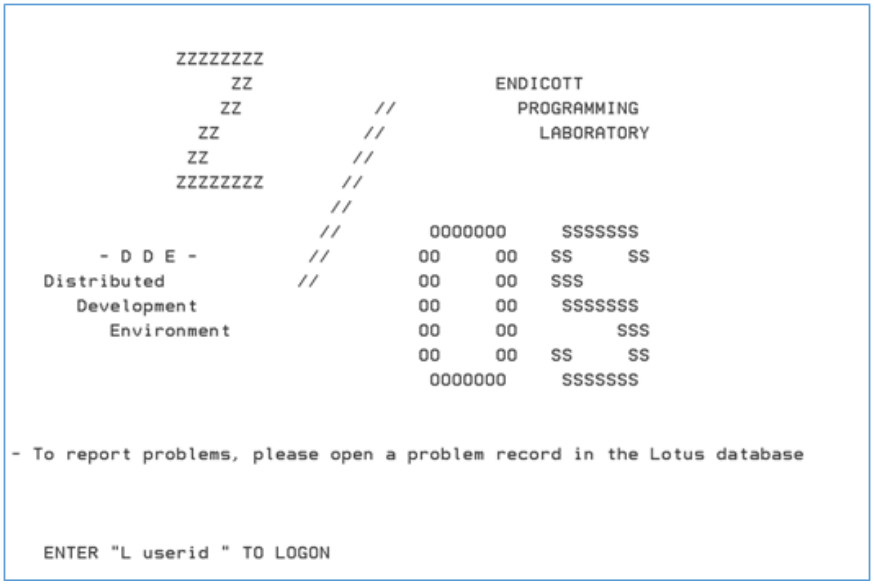
Console configuration in z/OS

First, log on to a z/OS guest that is already IPLed. If it has already IPLed successfully you should be able to go to the command line of a z/VM logon screen and DIAL into the z/OS guest, optionally with a console number, in order to get to the TSO login screen.

The below example shows dialing into the z/OS guest `DCEIMGCA` on z/VM system `GDLPKVM6`, specifying the console `0CC8`.

Screenshot 1 - z/VM Login Screen with Dial Command

This is the resultant TSO logo screen for this system. TSO logo screens may be customized, so your logo is likely different:



Screenshot 2 - Login screen for z/OS

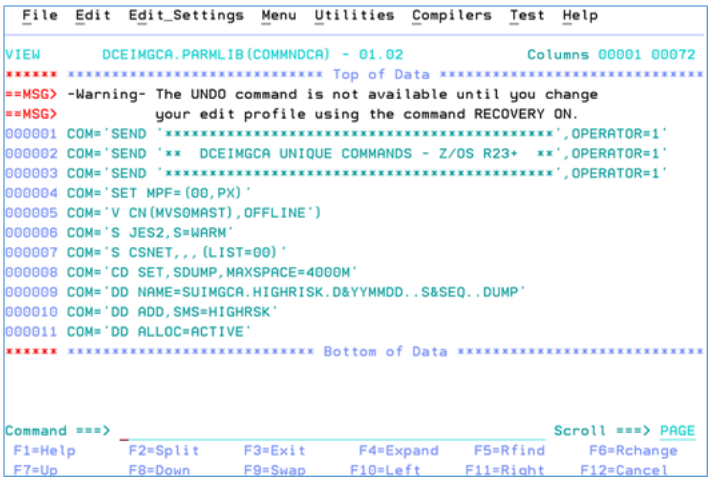
If the guest is not configured to allow dialing in, or the console specified is not fully activated, you may see a message like this:

```
d dceimgca cc8
DIALED TO DCEIMGCA 0CC8
```

Screenshot 3 - Dialed into z/OS but the console is not activated

See Appendix A for some troubleshooting help on getting your z/OS guest running.

Once dialed in and logged on, you will need to configure the z/OS guest via ISPF and the PARMLIB files. In the following examples the suffix “CA” is used. A selection of pertinent files is shown below.



Screenshot 4 - COMMNDCA PARMLIB file

The COMMNDxx file shows unique commands for the `DCEIMGCA` guest. Take note of the line

```
V CN(MVS0MAST),OFFLINE.
```

This says the `MVS0MAST` console will be offline when z/OS IPLs. You should make other consoles offline to avoid conflicts with the automation services Operations Manager provides. You can use Operations Manager for any tasks that you would normally log on to the master console to do.

```
 File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
VIEW       DCEIMGCA.PARMLIB(CNGRP00) - 01.01              Columns 00001 00072
****** ***************************** Top of Data ******************************
==MSG> -Warning- The UNDO command is not available until you change
==MSG>           your edit profile using the command RECOVERY ON.
000001 GROUP NAME(SYSCON)
000002       MEMBERS(MVS0MAST)
****** **************************** Bottom of Data ****************************
Command ===> _                                            Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F4=Expand    F5=Rfind     F6=Rchange
 F7=Up        F8=Down      F9=Swap      F10=Left     F11=Right    F12=Cancel
```

Screenshot 5 - CNGRP00 PARMLIB file

The CNGRPxx file defines console groups. MVS0MAST here is defined to be part of the SYSCON group, which is a user-defined name. More about console group definitions and how you might use them is available here: https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.1.0/com.ibm.zos.v2r1.ieae200/cngrp.htm. In this case, our CNGRP00 is the first one to receive synchronous messages. This corresponds to the INIT CNGRP(00) statement in the CONSOL00 file, shown below.

```
000120 /*********************************************************************/
000121 /**** INITIALIZATION PARMETERS FOR ALL CONSOLES                   ****/
000122 /*********************************************************************/
000123 INIT     CMDDELIM(;)          /* COMMAND DELIMITER                ****/
000124           MLIM(4000)           /* MAX # OF BUFFERS FOR WTO         ****/
000125           LOGLIM(10000)        /*                                  ****/
000126           MONITOR(DSNAME)      /* ADD DSNAME TO MOUNT MSGS         ****/
000127           MMS(NO)              /* MESSAGE SERVICE NOT ACTIVE       ****/
000128           PFK(00)              /* PFK DEFINITIONS MEMBER SUFFIX    ****/
000129           RLIM(50)             /* MAX # OF BUFFERS FOR WTOR        ****/
000130           UEXIT(N)             /* NO WTO EXIT                      ****/
000131           CNGRP(00)            /* CONSOLE GROUP FOR SYSCONS          */
000132 /****                                                              ****/
000133 /****                                                              ****/
```

Screenshot 6 - INIT statement from the CONSOL00 PARMLIB file

```
 File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
VIEW       DCEIMGCA.PARMLIB(CONSOL00) - 01.03            Columns 00001 00072
000016 /****               DEFINED <CA8> AS FIRST ALTERNATE CONSOLE.   ****/
000017 /*************************************************************************/
000018 /****                                                              ****/
000019 /****                                                              ****/
000020 /*************************************************************************/
000021 /**** MASTER CONSOLE DEFINITION                                   ****/
000022 /*************************************************************************/
000023 CONSOLE DEVNUM(SYSCONS)           /* ADDRESS OF THIS CONSOLE        */
000024         ALLOWCMD(Y)               /* ALLOW COMMANDS FROM SYSCONS    */
000025         AUTH(MASTER)              /* MASTER AUTHORITY               */
000026         INTIDS(Y)                 /* RECEIVE MSGS FOR CONSOLE 0     */
000027         LEVEL(ALL)                /* MESSAGE LEVEL                  */
000028         MONITOR(JOBNAMES-T,SESS-T)       /* MONITOR SELECTED EVENTS */
000029         NAME(SYSCONS)             /* CONSOLE NAME                   */
000030         ROUTCODE(ALL)             /* MESSAGE ROUTING CODES          */
000031         UNKNIDS(Y)                /* RECEIVE MSGS FOR UNKNOWN CONS  */
000032 /****                                                              ****/
000033 /****                                                              ****/
Command ===> _                                            Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F4=Expand    F5=Rfind     F6=Rchange
 F7=Up        F8=Down      F9=Swap      F10=Left     F11=Right    F12=Cancel
```

Screenshot 7 - CONSOL00 PARMLIB file - SYSCONS

The CONSOLxx file defines the consoles. Here is a console definition. Note that the device number is SYSCONS, a reference to the integrated line mode console. This means that the console is not tied to any particular device.

```
 File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
VIEW       DCEIMGCA.PARMLIB(CONSOL00) - 01.03            Columns 00001 00072
000034 /*************************************************************************/
000035 /**** ALTERNATE CONSOLE DEFINITIONS                                ****/
000036 /*************************************************************************/
000037 CONSOLE DEVNUM(CC8)               /* ADDRESS OF THIS CONSOLE        */
000038         UNIT(3270-X)              /* GENERIC DISPLAY DEVICE         */
000039         NAME(MVS0MAST)            /* CONSOLE NAME                   */
000040         AUTH(ALL)                 /* MASTER AUTHORITY               */
000041         USE(FC)                   /* USE FULL CAPABILITIES          */
000042         ROUTCODE(ALL)             /* MESSAGE ROUTING CODES          */
000043         LEVEL(ALL)                /* MESSAGE LEVEL                  */
000044         CON(N)                    /* CONVERSATIONAL MODE            */
000045         DEL(RD)                   /* DELETE MODE                    */
000046         RNUM(5)                   /* ROLL NUMBER OF LINES           */
000047         RTME(1/2)                 /* ROLL TIME THRESHOLD IN SEC.    */
000048         MFORM(J,T)                /* MESSAGE FORMAT                 */
000049         AREA(NONE)                /* DISPLAY AREA NUMBER OF LINES   */
000050         PFKTAB(PFKTAB1)           /* PFKEY DEFINITION TABLE         */
000051         MONITOR(JOBNAMES-T,SESS-T)       /* MONITOR SELECTED EVENTS */
Command ===> _                                            Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F4=Expand    F5=Rfind     F6=Rchange
 F7=Up        F8=Down      F9=Swap      F10=Left     F11=Right    F12=Cancel
```

Screenshot 8 - CONSOL00 PARMLIB file - CC8 definition

Here is the definition for `MVS0MAST`. This is defined at device address `CC8`, which is why this address was used on the `DIAL` command above in Screenshot 1.

- NAME is a user-defined string. Make it different from anything that could be a device number to avoid confusion. You should fill this in because the default is the same as the system name.
- Message LEVEL is what kinds of messages get to this console. To ensure Operations Manager receives all messages, set this to ALL.
- Other possible options, not shown here
    - AUTOACT –> good to specify on one of a group of consoles so only one is automatically activated
    - UNKNIDS(Y) -> receive messages for unknown codes
    - ALLOWCMD(Y) -> can issue commands even when not in problem determination mode. This is only valid for SYSCONS.

You may see more CONSOLE statements; ideally all console device numbers correlate with consoles defined to the z/OS guest via the z/VM directory or z/VM CP commands. See section "Checking the z/VM directory for console definitions" for an example of a z/VM directory for DCEIMGCA.

There are many other possible PARMLIB files, and more information on PARMLIB files is available here:
https://www.ibm.com/support/knowledgecenter/zosbasics/com.ibm.zos.zsysprog/zsysprogc_syslibraries.htm

Console setup in the z/VM guest

Now that the z/OS setup is done, check the z/VM guest's setup to ensure this matches the consoles defined. In this example the console is at device number `CC8`. z/VM guest definitions may be done in one of 2 places: the z/VM user directory, which provides the base definition of the guest and any maximums it is subject to, and dynamic commands applied to the guest.

To dynamically check, go to your z/OS guest and issue the `#CP QUERY VIRTUAL CC8` command. Optionally, you may want a larger view of all the virtual resources available via the `#CP QUERY VIRTUAL ALL` command.

Screenshot 9 - QUERY VIRTUAL CC8

Screenshot 9 shows the output of this command and that it matches our desired settings

- `CC8` is defined as a GRAF device, so it can be used as a 3270 display.
- `NOT DIALED` shows that no user is currently connected to this device. If a user is connected to this device, Operations Manager will only be able to monitor the console if you use the OBSERVER function in SCIF. Refer to the Operations Manager Administration Guide for more details and guidance on SECUSER versus OBSERVER.

Dynamic console activation

Issuing a `DIAL` into `CC8` at this point may not work. This is because the `MVS0MAST` console at device number `CC8` was not defined to be online at IPL time. This command will put the console online:

```
#CP VINPUT VMSG VARY CN(mvs0mast),online
```

Since the 3270 session into which you previously dialed has the integrated line mode terminal for z/OS, the `#CP VINPUT VMSG` prefix is necessary to send the command to the integrated terminal. Simply issuing `VARY CN` will not send the command to z/OS (see Figure 1 for an example of how commands are routed through consoles to z/OS).

```
00:
00: CP VINPUT VMSG VARY CN(MVSOMAST),ONLINE
03:  CNZ4100I 17.12.37 CONSOLE DISPLAY 648
03:  CONSOLES MATCHING COMMAND: D C,U=(0CC8)
03:  MSG:CURR=0    LIM=4000 RPLY:CURR=0    LIM=50    SYS=DCEIMGCA PFK=00
03:  MVSOMAST  TYPE=MCS      STATUS=ACT-DCEIMGCA
03:            DEFINED=(*ALL)
03:            MATCHED=(*ALL)
03:     ATTRIBUTES ON DCEIMGCA
03:        AUTH=(ALL)        CMDSYS=N/A          NBUF=0        SUPSBY=N
03:        DEV=0CC8          LOGON=OPTIONAL      USERID=N/A      TIMEOUT=N/A
03:        MFORM=(T,J)       AREA=(Z)            PFKTAB=PFKTAB1
03:        USE=FC  DEL=RD    RTME=1/2  RNUM=5    SEG=*     CON=N
03:        LEVEL=(ALL)
03:        MONITOR=(JOBNAMES,SESS)               INTIDS=N  UNKNIDS=N
03:        ROUT=(ALL)
```

Screenshot 10 - Example of VARY CN online command response

And now, flipping back to the `CC8` console should show that it is now dialed into the z/OS guest.

```
IEE612I CN=MVSOMAST DEVNUM=0CC8 SYS=DCEIMGCA

IEE163I MODE= RD
```

Screenshot 11- z/OS guest with console activated and dialed in

Note that the console definition that Operations Manager uses as well as the CC8 definition explained above are not password protected. Because authorization is handled here by the z/VM system (i.e., the DCEIMGCA guest password) and Operations Manager.

Checking the z/VM directory for console definitions

So far this paper has focused on dynamic commands, but you should check the z/VM directory entry for the z/OS guest, in case the guest is logged off.

It should have a user ID similar to:

```
USER DCEIMGCA xxxxxxx 2G 4G G
   INCLUDE COMMVST
   ACCOUNT VMACCT1 ZOSE
   SPECIAL 0CC8 3270
   CPU 00 BASE
   CPU 01
   CPU 02
   CPU 03
   IPL CMS PARM AUTOCR
   OPTION CFUSER TODENABLE
   CONSOLE 0680 3215 T OPMGRM1 OBSERVER
   MDISK 04E0 3390 00001 03339 S54001 MWV
   MDISK 04E1 3390 03340 03339 S54001 MWV
   MDISK 0816 3390 2761 00040 CFC990 MR
```

1. The `INCLUDE` statement brings in all the definitions from the `PROFILE COMMVST`, including minidisk links
2. The `SPECIAL` statement defines device `CC8` as a virtual 3270 console
3. The `CONSOLE` statement is required to enable Operations Manager to control this guest. It means:
    a. There is a console at address `680`. The number of this console is not important.
    b. The console is of type 3215, which allows use of SCIF, in conjunction to zVM command `SEND`. See "SCIF – Single Console Image Facility" for more information on this function.
    c. Class T is a placeholder
    d. `OPMGRM1 OBSERVER` enables the `OPMGRM1` userid to be an `OBSERVER` of this userid.
4. The integrated console does not appear in the directory statement, this always exists.

SCIF – Single Console Image Facility

Single Console Image Facility is a way in z/VM for guests to have control of other guests. There are 3 main commands:

- `SET SECUSER` allows one user ID (X) to see another user ID's (Y) console and to use `CP SEND` commands to send commands from X to Y. This will work even if X only has class G privileges. X may be a SECUSER for more than one user ID at the same time. The main drawback of SECUSER is that it will only work if Y is disconnected.
- `SET OBSERVER` allows one user ID (X) to see another user ID's (Y) console, but does not allow use of the `CP SEND` command. X may be an `OBSERVER` of more than one user ID at the same time. `SET OBSERVER` will work even if a user is logged on to Y. Its drawback is that `SET OBSERVER` does not automatically allow X to `SEND` commands to Y unless X also has the appropriate privilege class for the CP SEND command in general. There are other considerations as well. Refer to the Operations Manager Administration Guide for more details.
- The `CP SEND` command allows sending of a command from one user ID (X) to another user ID (Y). If X has privilege class C, it may send commands to any user ID on the system. If X has privilege class G, it may only send commands to user IDs for which it is a `SECUSER`.

Operations Manager Configuration

Some setup must be done in configuration files to define the userid `DCEIMGCA` to Operations Manager. Configuration files for Operations Manager are usually on the `OPMGRM1 198` minidisk and they are of filetype `CONFIG`.

Authorization to view consoles in Operations Manager can be controlled by an External Security Manager, such as RACF, or by Operations Manager itself. The example system uses the latter approach.

The directory entry above allowed Operations Manager itself to be `OBSERVER` of `DCEIMGCA`, but end users must then be separately authorized in Operations Manager for each guest they wish to control. The below statement authorizes an Operations Manager end user id `TLD1` to look at the console of z/OS guest `DCEIMGCA`.

```
AUTH USER TLD1     CONSOLE DCEIMGCA
```

Operations Manager PTF UI68310 for APAR PH22941 is required to enable users to issue z/OS commands to a z/OS guest from the VIEWCON function. Previously, the only way to send commands in real time to the z/OS guest was using the VIEWLOG capability. This is undesirable because VIEWLOG:

- Will show all consoles from all guests
- Requires an additional prefix on all commands:
  CP CMD 'SEND CP DCEIMGCA VINPUT VMSG <command>'
- Requires additional authority in Operations Manager

To issue z/OS commands directly in the VIEWCON function of Operations Manager, you will need to define the guest to Operations Manager as a z/OS guest. In general, Operations Manager does not know what kind of operating system is running in each guest. By default Operations Manager uses CP SEND (without VINPUT VMSG) to send commands to a guest. To enable Operations Manager to send commands correctly to a z/OS guest (using VINPUT VMSG), you must specify which guests are running z/OS.

To do this, first add the DEFOPTN VCSCGRP statement to the OPMGRM1 CONFIG file to specify a name for the group of z/OS guests. The members of the group will be defined later. For example:

DEFOPTN VCSCGRP *groupname*

```
 OPMGRM1  CONFIG   Z2  V 80  Trunc=80 Size=136 Line=0 Col=1 Alt=0
====>
     ...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....>


00000 * * * Top of File * * *
00001 *
00002 * Sample configuration file for Operations Manager
00003 *
00004 LOGTEXT TEXT(STARTUP FOR OPERATION MANAGER)
00005 *
00006 * Options for Operations Manager
00007 *
00008 DEFOPTN ESM N
00009 DEFOPTN HLDATTR 0060,N,Y,AAI,CAI
00010 DEFOPTN MIXCASE N
00011 DEFOPTN DSTADJ Y
00012 DEFOPTN SYMBOLS <>
00013 DEFOPTN AUTHCMD N
00014 DEFOPTN VCSCGRP ZOSGUEST
00015 *
00016 * Define groups
00017 DEFGROUP NAME SYSADM MEMBER MAINT*
00018 *
00019 * Authorize the SYSADM group for all Operations Manager functions
00020 *
00021 AUTH GROUP SYSADM ACCESS CONTROL
```

Screenshot 12 - Creating the VCSCGRP named ZOSGUEST

In the example on line 14 of Screenshot 12, ZOSGUEST is the name of the group.

Next, add one or more DEFGROUP statements to the configuration file to add members to the group:

```
DEFGROUP NAME groupname MEMBER z/OSguestname
```

```
 TRACY    CONFIG   Z2  V 80  Trunc=80 Size=107 Line=0 Col=1 Alt=0
====>
     ...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....>


00000 * * * Top of File * * *
00001 * Tracy's configuration file for Operations Manager for z/VM
00002 **
00003 * Define this guest to Operations Manager as a z/OS guest
00004 DEFGROUP NAME ZOSGUEST MEMBER DCEIMGCA
```

Screenshot 13 - Defining DCEIMGCA to the ZOSGUEST group

In the example in Screenshot 13, the DEFGROUP statement specifies the group name ZOSGUEST and adds the z/OS guest DCEIMGCA to this group. Note that the group name specified on DEFOPTN must match the group name specified on DEFGROUP. Refer to the Operations Manager Administration Guide for more information on DEFOPTN and DEFGROUP.

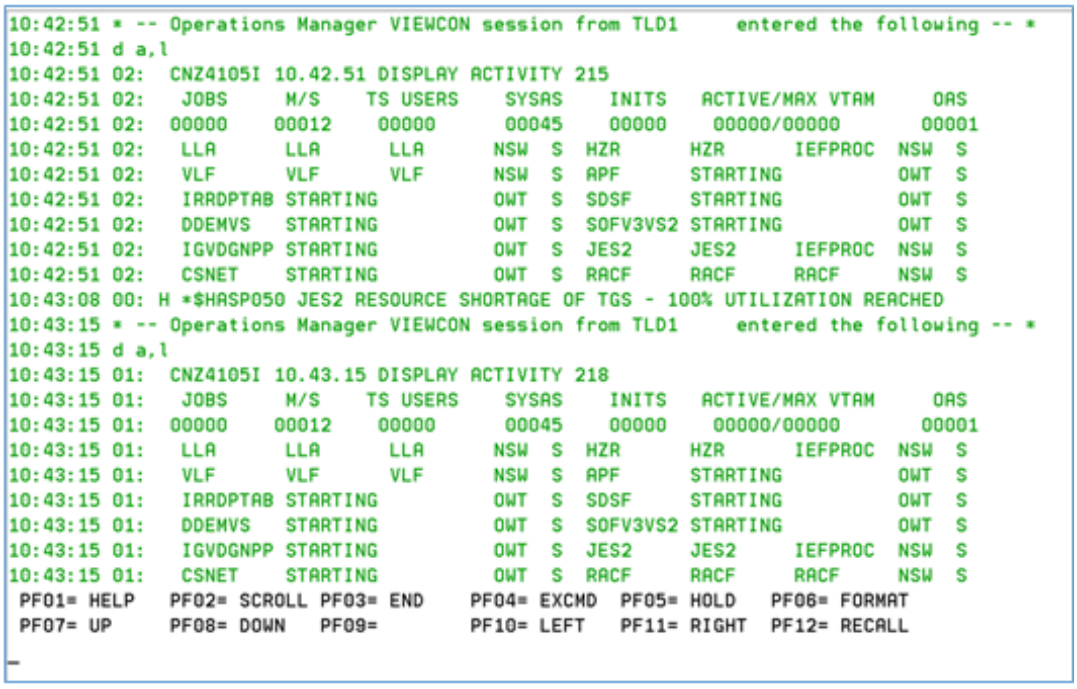
Now that you have made changes, you will need to tell Operations Manager to re-access and reload the configuration files. Run the OMRELOAD EXEC to do this across your SSI cluster.

Now issue the Operations Manager VIEWCON DCEIMGCA command to view the live console of the z/OS guest.

```
10:29:47 01:  $HASP409 INIT     WAS ON STCINRDR ON MEMBER EIMG
10:29:47 01:  $HASP409 INIT     WAS ON STCINRDR ON MEMBER EIMG
10:29:47 02:  $HASP492 JES2 ALL-MEMBER WARM START HAS COMPLETED - z22 MODE
10:29:47 02:  $HASP261 Member EIMG performs deadline scheduling processing
10:29:47 02:  $HASP249 COMMAND RECEIVED FROM INITIALIZATION
10:29:47 02:  $CA,ALL
10:29:47 00:  $HASP000 OK
10:29:47 03:  $HASP249 COMMAND RECEIVED FROM INITIALIZATION
10:29:47 03:  $TNUM,BASE=0001
10:29:47 01:  $HASP609 NUMBER SET TO 1
10:29:47 01:  $HASP355 SPOOL VOLUMES ARE FULL
10:29:47 02:  $HASP249 COMMAND RECEIVED FROM INITIALIZATION
10:29:47 02:  $SI
10:29:47 02:  $HASP892 INIT(1)   STATUS=STARTING,CLASS=ABCDE,NAME=1
10:29:47 02:  $HASP892 INIT(2)   STATUS=STARTING,CLASS=ABCDE,NAME=2
10:29:47 02:  $HASP892 INIT(3)   STATUS=STARTING,CLASS=ABCDE,NAME=3
10:29:47 02:  $HASP892 INIT(4)   STATUS=STARTING,CLASS=ABCDE,NAME=4
10:29:47 02:  $HASP892 INIT(5)   STATUS=STARTING,CLASS=ABCDE,NAME=5
10:29:47 02:  $HASP892 INIT(6)   STATUS=STARTING,CLASS=ABCDE,NAME=6
10:29:47 02:  $HASP892 INIT(7)   STATUS=STARTING,CLASS=ABCDE,NAME=7
10:29:47 02:  $HASP892 INIT(8)   STATUS=STARTING,CLASS=ABCDE,NAME=8
10:29:47 02:  $HASP892 INIT(9)   STATUS=STARTING,CLASS=ABCDE,NAME=9
10:29:47 02:  $HASP892 INIT(10)  STATUS=STARTING,CLASS=ABCDE,NAME=10
PF01= HELP   PF02= SCROLL PF03= END    PF04= EXCMD  PF05= HOLD   PF06= FORMAT
PF07= UP     PF08= DOWN   PF09=        PF10= LEFT   PF11= RIGHT  PF12= RECALL
```

Screenshot 14 - VIEWCON DCEIMGCA

Notice you can issue z/OS commands (such as `D A,L`) normally with no prefix needed:



Screenshot 15 - Issuing D A,L via the VIEWCON interface

Tips and Tricks for z/OS Startup with Operations Manager

You will want to include some rules and actions to help control your z/OS guest. Operations Manager can have rules and actions that are targeted at a specific user, or common for all users being managed.

```
 TRACY    CONFIG   Z2  V 80  Trunc=80 Size=107 Line=6 Col=1 Alt=0
====>
     ...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....>

00005
00006 ** Automate startup of z/OS guest DCEIMGCA
00007 * Watch for z/OS guest DCEIMGCA to come up and when ready, issue commands to
00008 * vary the system console on as needed.  Then display a message re spool
00009 * consumption
00010 DEFRULE NAME MVSRDR +
00011  MATCH '*IEE389I MVS COMMAND PROCESSING AVAILABLE*' +
00012  USER DCEIMGCA +
00013  ACTION MVSCONS1
00014
00015 DEFACTN NAME MVSCONS1 +
00016  COMMAND 'CP SEND CP DCEIMGCA VINPUT VMSG VARY CN(*),ACTIVATE' +
00017  ENV LVM +
00018  INPUT AHI +
00019  NEXTACTN MVSCONS2 +
00020  NEXTDELY 00:10
00021
00022 DEFACTN NAME MVSCONS2 +
00023  COMMAND 'CP SEND CP DCEIMGCA VINPUT VMSG V CN(*),AROUT=(1-128)' +
00024  ENV LVM +
00025  NEXTACTN SHZOSSPL +
00026  NEXTDELY 00:30
```

Screenshot 16 - Operations Manager Rules for z/OS guest startup

For example, Screenshot 16 shows a rule and 2 actions that vary the z/OS console online during the z/OS IPL process so it is ready to receive commands. The first part is defining a rule called `MVSRDR`. It specifies that if Operations Manager sees message `IEE389I` for user `DCEIMGCA`, it should execute the action called `MVSCONS1`. Message `IEE389I` is a message issued by z/OS during IPL when initialization is complete and the system is ready to process commands. This first rule brings the console online as soon as MVS command processing allows.

> Note on `ENV LVM` vs. specifying a particular SVM ID
>
> `ENV LVM` will have the command run on OPMGRM1. You can instead specify a particular action processing server (default OPMGRS1-OPMGRS4) to issue the command instead. Set this in accordance with which ID is set up to have the correct privilege to `CP SEND` commands to the z/OS user. Action processing servers are recommended for longer running tasks, such as EXECs, so the task will not tie up the main Operations Manager server.

The action `MVSCONS1` issues a `CP SEND` command that will tell z/OS to activate the console (see Appendix A for more information on how to activate a dialed in console). Normally z/OS would not activate the integrated console for command input, so this is necessary. The `INPUT`

AHI option ensures the input is highlighted, which makes it easier to see important messages in the console log.

After action MVSCONS1 there will be a 10 second delay before the next action, MVSCONS2. This is another VARY CONSOLE that routes all messages from any console to this console. This ensures that no messages are missed.

```
 TRACY    CONFIG   Z2  V 80  Trunc=80 Size=107 Line=28 Col=1 Alt=0
====>
      ...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....>

00027
00028 DEFACTN NAME SHZOSSPL +
00029  COMMAND 'CP SEND CP DCEIMGCA VINPUT VMSG $D SPL' +
00030  ENV LVM
00031
00032 **
00033 * Highlight spool utilization message and change to red
00034 * Will create automation later to clean up spool if > 75% full
00035 DEFRULE NAME SPLUTIL +
00036  MATCH '*$HASP646*PERCENT SPOOL UTILIZATION*' +
00037  USER DCEIMGCA +
00038  ACTION CKZOSSPL
00039
00040 DEFACTN NAME CKZOSSPL +
00041  COMMAND 'EXEC CKZOSSPL &T' +
00042  ENV SVM +
00043  INPUT AHI,CRE
00044
00045 DEFACTN NAME HILITE +
00046  INPUT AHI,CRE
00047
```

Screenshot 17 - Operations Manager Actions to Check Spool Usage

Screenshot 17 shows the action SHZOSSPL. This issues a command to the z/OS guest to display the status of the spool. There is a rule and several actions to display and clean up spool space usage in z/OS. An exec, CKZOSSPL, has been created and is executed on an action server (SVM). The EXEC is shown below issues a message if spool is more than 75% full.

```
/*  Parse the message from z/OS re spool utilization and send a message to */
/*  OPMGRM1 to take an action if spool is too full                          */

Parse Arg . '$HASP646' pctfull . "."
/* $HASP646 34.4000 PERCENT SPOOL UTILIZATION */
If pctfull > 75 Then Say 'Clean up spool on DCEIMGCA'
Exit
```

This is an example of what the previous set of commands will look like on the screen. It queries the spool usage on the system, and highlights the resulting message in red:

```
13:40:05 00:  $HASP893 VOLUME(SYS001)  STATUS=ACTIVE,PERCENT=11
13:40:05 00:  $HASP646 11.4666 PERCENT SPOOL UTILIZATION
13:40:05 * -- Operations Manager Action CKZOSSPL scheduled for execution -- *
```

Screenshot 18- Querying Spool Utilization in z/OS via Operations Manager

Another rule you should define is one to clear out held messages. Held messages are kept by z/VM for display using the CP QUERY VMSG or CP QUERY PVMSG commands. z/VM will only keep a limited number of messages thus, when the maximum is reached, no further messages will be displayed. That includes messages that are not held and the message warning you no further messages can be written. Thus, to avoid missed messages, you should set up Operations Manager to delete held messages.

Held messages can be identified by the string "H " in front of the message. Here is an example of a held message:

```
00: H *$HASP493 JES2 ALL-MEMBER WARM START IS IN PROGRESS - z22 MODE
```

Here is an example of a rule and action that will delete held messages whenever a held message is seen. Please note that the match here assumes that the z/OS guest has `TERMINAL TIMESTAMP OFF`. It also assumes the z/OS guest has multiple CPUs and so the message will be prefixed with the CPU number it was issued on. Be sure to look at some examples on your system before you write your rules. Also note that the rule may catch other messages if they start with " `H` ", but excess VDELETE commands will not hurt anything.

```
DEFRULE NAME HOLDMSG +
 MATCH '##: H *' +
 USER DCEIMGCA +
 ACTION DELMSGS

DEFACTN NAME DELMSGS  +
 COMMAND 'CP SEND CP &U VDELETE VMSG 1-16' +
 ENV LVM +
 INPUT AHI,CRE
```

Note that this will clear out any held messages, but may generate excess error messages similar to:

```
HCPPCX6532I There is no non-priority message currently outstanding
with message number 0011 assigned to it.
```

To suppress these messages, so they do not appear in VIEWCON output, you will want to create a rule and action similar to these.

```
DEFRULE NAME SUPPMSG +
 MATCH '*6532I There is no non-priority message currently *' +
 USER DCEIMGCA +
 ACTION SUPMSGS
```

```
DEFACTN NAME SUPMSGS  +
 INPUT SUP
```

You may also want to have other rules that kick in after the z/OS guest is up and running. You may have a heartbeat type rule that checks in with z/OS periodically to make sure it is still operating.

Tips and Tricks for z/OS Shutdown with Operations Manager

There are many different ways to shut down a z/OS guest. The exact shutdown procedure is dependent on what is running inside z/OS. This paper offers one method.

First, find out what is running, using `D A,L` (Display All List). See Screenshot 15 above for an example of `D A,L` output. From this output you will extract the exact list of things that need to be shut down. In the example system uses the exec in Appendix D – Example SHUTZOS EXEC. This completes some shutdown tasks through shutting down OMVS. To issue an exec, you must

1. Load the EXEC on to a disk to which Operations Manager has access (e.g., `OPMGRM1 198`)
2. Add an action that will run the EXEC, similar to the one in screenshot 19:

```
00032 **
00033 * Add in an action here that runs our shutdown exec
00034 * To run this EXEC use GOMCMD OPMGRM1 RUN ACTION SHUTZOS PARM <username>_
00035 DEFACTN NAME SHUTZOS  +
00036  COMMAND 'EXEC SHUTZOS &P' +
00037  ENV SVM
00038
```

Screenshot 19 – Action to call the SHUTZOS exec

This command can then be run by an end user of Operations Manager by issuing the following command:

```
GOMCMD OPMGRM1 RUN ACTION SHUTZOS PARM DCEIMGCA
```

Once the EXEC completes, there are some additional shutdown actions that need to be done. Actions in the EXEC can be done all at once, while the additional shutdown options need to happen in a particular order, keyed off of certain messages. Here is an example of some Operations Manager rules and actions to finish z/OS shutdown:

```
00056 ** Automate shutdown of z/OS guest DCEIMGCA
00057 * Watch for z/OS guest DCEIMGCA to shut down and when ready, issue commands to
00058 * complete the process and log it off.
00059 DEFRULE NAME OMVSDOWN +
00060  MATCH '*$HASP099 ALL AVAILABLE FUNCTIONS COMPLETE' +
00061  USER DCEIMGCA +
00062  ACTION STOPJES2
00063
00064 DEFACTN NAME STOPJES2 +
00065  COMMAND 'CP SEND CP DCEIMGCA VINPUT VMSG $P JES2' +
00066  ENV LVM +
00067  INPUT AHI
00068
00069 DEFRULE NAME JESDOWN +
00070  MATCH '*IEF404I JES2 - ENDED*' +
00071  USER DCEIMGCA +
00072  ACTION STOPRACF
00073
00074 DEFACTN NAME STOPRACF +
00075  COMMAND 'CP SEND CP DCEIMGCA VINPUT VMSG RACF STOP' +
00076  ENV LVM +
00077  INPUT AHI +
00078  NEXTACTN HALTMVS
```

Screenshot 20 - Shutting down JES2 and RACF

The first one watches for message HASP099 that OMVS has completed shutdown and then shuts down JES2 with the STOP command. Once JES2 is completely shut down, it then stops RACF.

```
00080 DEFACTN NAME HALTMVS +
00081  COMMAND 'CP SEND CP DCEIMGCA VINPUT VMSG Z EOD' +
00082  ENV LVM +
00083  INPUT AHI
00084
00085 DEFRULE NAME MVSHALT +
00086  MATCH '*IEE334I HALT EOD SUCCESSFUL*' +
00087  USER DCEIMGCA +
00088  ACTION STOPMVS
00089
00090 DEFACTN NAME STOPMVS +
00091  COMMAND 'CP SEND CP DCEIMGCA VINPUT VMSG QUIESCE' +
00092  ENV LVM +
00093  INPUT AHI
00094
00095 DEFRULE NAME MVSSTOP +
00096  MATCH '*BLW002I *' +
00097  USER DCEIMGCA +
00098  ACTION MVSLGOFF
00099
00100 DEFACTN NAME MVSLGOFF +
00101  COMMAND 'CP SEND DCEIMGCA LOGOFF' +
00102  ENV LVM
```

Screenshot 21 - HALT EOD, QUIESCE and LOGOFF in Operations Manager

After RACF is shutdown, it's safe to halt the system using Z EOD (https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.3.0/com.ibm.zos.v2r3.ieag100/iea3g1_Syntax17.htm). When the HALT is completed, the guest is quiesced. And then when the quiesce has completed, it is safe to logoff the guest, which appears like this on the console:

```
11:30:12 03:  IEE600I REPLY TO 01 IS;END
11:30:12 01:  $HASP085 JES2 TERMINATION COMPLETE
11:30:12 02:  IXZ0003I CONNECTION TO JESXCF COMPONENT BROKEN
11:30:12 02:           GROUP DCEIMGCA MEMBER DCEIMGCA$EIMG
11:30:12 01:  IEF450I JES2 JES2 - ABEND=S02D U0000 REASON=07D1F240
11:30:12 01:          TIME=11.30.12
11:30:12 03:  IEF404I JES2 - ENDED - TIME=11.30.12
11:30:12 * -- Operations Manager Action STOPRACF scheduled for execution -- *
11:30:12 * -- Operations Manager Action HALTMVS  scheduled for execution -- *
11:30:12 01:  IEE305I RACF     COMMAND INVALID
11:30:12 02:  IEE334I HALT EOD SUCCESSFUL
11:30:12 * -- Operations Manager Action STOPMVS  scheduled for execution -- *
11:30:12 01:  HCPGSP2629I The virtual machine is placed in CP mode due to a SIGP stop from CPU 00.
11:30:12 02:  HCPGSP2629I The virtual machine is placed in CP mode due to a SIGP stop from CPU 00.
11:30:12 03:  HCPGSP2629I The virtual machine is placed in CP mode due to a SIGP stop from CPU 00.
11:30:12 01:  HCPGIR450W CP entered; disabled wait PSW 00020000 80000000 00000000 00000CCC
11:30:12 02:  HCPGIR450W CP entered; disabled wait PSW 00020000 80000000 00000000 00000CCC
11:30:12 03:  HCPGIR450W CP entered; disabled wait PSW 00020000 80000000 00000000 00000CCC
11:30:12 00:  HCPPCM6533A Following is a priority message received by the service processor - use the VINPUT command to respond:
11:30:12 00:  *BLW002I SYSTEM WAIT STATE 'CCC'X - QUIESCE FUNCTION PERFORMED
11:30:12 * -- Operations Manager Action MVSLGOFF scheduled for execution -- *
11:30:12 00:  CONNECT= 01:32:20 VIRTCPU= 007:10.74 TOTCPU= 007:48.79
11:30:12 00:  LOGOFF AT 11:30:12 EDT FRIDAY 06/12/20
PF01= HELP   PF02= SCROLL PF03= END    PF04= EXCMD  PF05= HOLD   PF06= FORMAT
PF07= UP     PF08= DOWN   PF09=        PF10= LEFT   PF11= RIGHT  PF12= RECALL
```

Screenshot 22 - Quiesce completed

Summary

IBM Operations Manager for z/VM can be an effective tool for monitoring and managing the consoles of z/OS guests on z/VM. However, care must be taken to properly configure the console settings such that all pertinent messages will go to Operations Manager. The VIEWCON facility of Operations Manager can be used to send commands to z/OS with a few simple configuration updates.

Once the console configuration changes for z/OS and Operations Manager are completed, there are many z/OS operational tasks that can be automated. In particular, startup and shutdown can be completed by Operations Manager without requiring human intervention. In all, Operations Manager makes managing any type of guest on z/VM a breeze.

Appendix A: z/OS Guest Startup Troubleshooting

Sometimes when starting a z/OS guest, startup does not completely finish because of messages waiting for replies. This section describes how to troubleshoot the issue while directly logged on to the guest. Note that in the command examples, all z/OS commands are prefaced with `#CP VINPUT VMSG`. `#CP` is necessary to escape and route the command to the z/VM control program. # is the escape character, which is settable in your system configuration file under CHARACTER_DEFAULTS statement.

You may need to activate the console manually through VARY CN commands:

```
#CP VINPUT VMSG V CN(*),ACTIVATE
#CP VINPUT VMSG V CN(*),AROUT=(1-128)
```

Then you can display current outstanding activity:

```
#CP VINPUT VMSG D  A,L
```

```
00:
00: CP VINPUT VMSG V CN(*),ACTIVATE
02:  IEE712I VARY CN  PROCESSING COMPLETE
00:
00: CP VINPUT VMSG V CN(*),AROUT=(1-128)
02:  IEE712I VARY CN  PROCESSING COMPLETE
00:
00: CP VINPUT VMSG D A,L
01:  CNZ4105I 15.19.47 DISPLAY ACTIVITY 611
01:    JOBS     M/S    TS USERS    SYSAS    INITS   ACTIVE/MAX VTAM     OAS
01:   00000    00013    00000      00032    00023    00000/10000       00008
01:    LLA      LLA      LLA      NSW  S  HZR      HZR      IEFPROC  NSW  S
01:    VLF      VLF      VLF      NSW  S  SDSF     SDSF     SDSF     NSW  S
01:    SOFV3VS2 SOFV3VS2         OWT  S  IGVDGNPP IGVDGNPP IGVDGNPP OWT  S
01:    JES2     JES2     IEFPROC  NSW  S  CSNET    CSNET    VTAM     NSW  S
01:    RACF     RACF     RACF     NSW  S  SDSFAUX  SDSFAUX  SDSFAUX  NSW  S
01:    APPC     APPC     APPC     NSW  S  ASCH     ASCH     ASCH     NSW  S
01:    TSO      TSO      STEP1    OWT  S
```

Screenshot 23 - Activating console and displaying current activity

Since nothing appears here, you can check what is pending for VTAM:

```
#CP VINPUT VMSG D NET, PENDING
```

```
00:
00: CP VINPUT VMSG D NET, PENDING
03:  IEE341I VTAM                NOT ACTIVE
```

Screenshot 24 - VTAM not active

This output shows that VTAM is not active. So you can use this command to show pending messages:

```
#CP VINPUT VMSG D R,R
```

```
00:  CP VINPUT VMSG D R,R
02:   IEE112I 15.18.55 PENDING REQUESTS 086
02:   RM=1     IM=2      CEM=2      EM=1      RU=0     IR=0     AMRF
02:   ID:R/K      T MESSAGE TEXT
02:            01 R *01 $HASP454 SHOULD JES2 BYPASS THE MULTI-MEMBER
02:                  INTEGRITY LOCK? ('Y' OR 'N')
```

Screenshot 25 – Pending message

To reply to this message (note the 01) you can use:

```
#CP VINPUT VMSG R 1,Y
```

```
00:  CP VINPUT VMSG R 1,Y
03:   IEE600I REPLY TO 01 IS;Y
01:   $HASP478 INITIAL CHECKPOINT READ IS FROM CKPT1
01:            (SYS1.HASPCKPT ON SYS001)
01:            LAST WRITTEN MONDAY, 15 JUN 2020 AT 19:17:26 (GMT)
01:  H *$HASP493 JES2 ALL-MEMBER WARM START IS IN PROGRESS - z22 MODE
```

Screenshot 26 - Replying to a pending message

In this case a second message appears you can reply with:

```
#CP VINPUT VMSG R 2,Y
```

```
02:   $HASP470 MEMBER  NOT DORMANT
03:  H *02 $HASP471 REPLY 'Y' TO CONTINUE WARM START, 'N' TO TERMINATE JES2
00:
00:  CP VINPUT VMSG R 2,Y
03:   IEE600I REPLY TO 02 IS;Y
02:   $HASP537 THE CURRENT CHECKPOINT USES 512 4K RECORDS
00:   IXZ0001I CONNECTION TO JESXCF COMPONENT ESTABLISHED,
00:            GROUP SYSJ2$XD MEMBER DCEIMGCA$EIMG$$$
```

Screenshot 27 - Replying to a second pending message

Now the z/OS guest is up and running and can be logged into:

```
              ZZZZZZZZ
                   ZZ                     ENDICOTT
                  ZZ           //           PROGRAMMING
                ZZ            //              LABORATORY
               ZZ            //
              ZZZZZZZZ      //
                           //
                          //       0000000     SSSSSSS
         - D D E -       //       00     00   SS     SS
   Distributed          //        00     00   SSS
      Development                 00     00    SSSSSSS
         Environment              00     00         SSS
                                  00     00   SS     SS
                                   0000000     SSSSSSS

- To report problems, please open a problem record in the Lotus database


   ENTER "L userid " TO LOGON
```

Screenshot 28 - z/OS logon screen

Appendix B – More on z/OS console definitions

z/OS needs 3 definitions to open a valid console: IOCP, NIPCON, and CONSOLxx. The main part of this paper goes over how to set up the CONSOLxx definitions. The address to be used for the SYSCONS master console must not be the same as the with IOCP and NIPCON definitions. During z/OS IPL, SYSCONS echoes the NIP (Nucleus Initialization Program) messages and stops, waiting for ASID CONSOLE to gain control. To continue using SYSCONS this terminal needs to be put in Problem Determination mode (mode=PD). To do it use the command:

```
V CN(*),ACTIVATE
```

in that terminal.

Lastly the NIP message is:

```
IEE389I MVS COMMAND PROCESSING AVAILABLE
```

Appendix C – More About Consoles

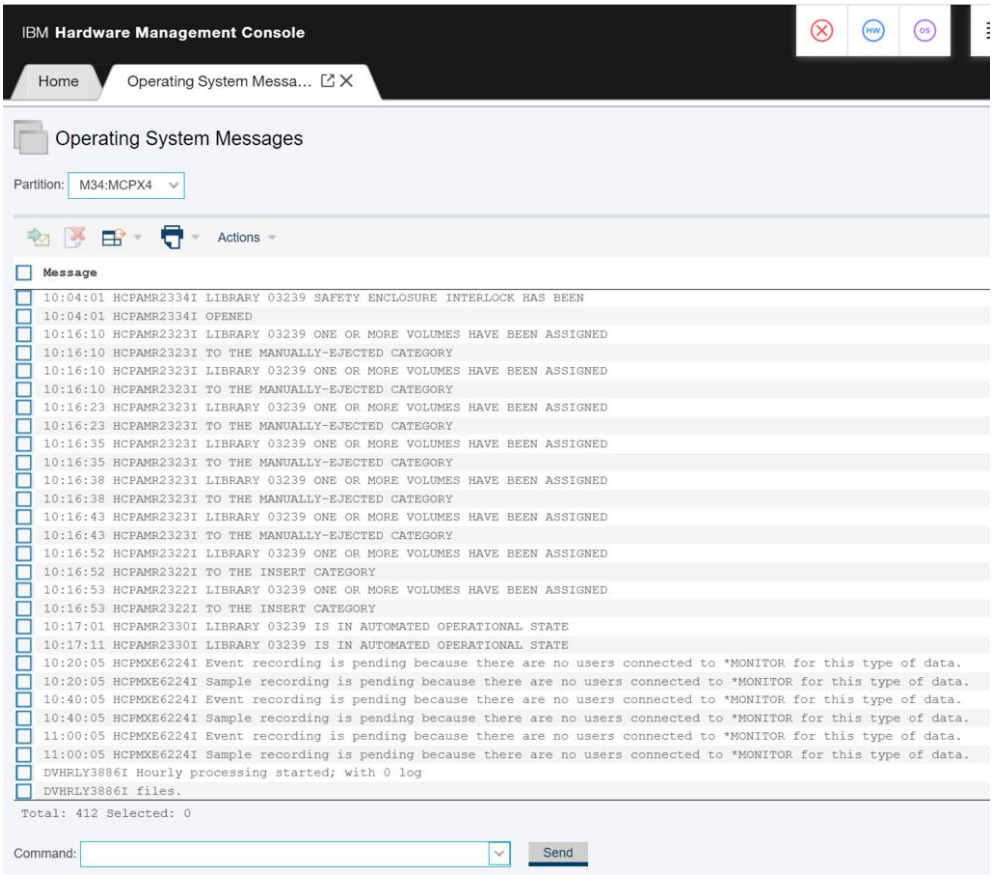
In the "real" world, an operating system can speak to the Hardware Management Console (HMC) in line mode via the **Operating System Messages** task. Aliases for this function include:

- Integrated console
- Line mode console
- Integrated line mode console
- System console

When an OS uses it, it is **NOT** a 3215 I/O device such as is defined for a virtual machine console for CMS or Linux. Sometimes the OS will let you tell it to use the integrated console, and sometimes the OS will only use it in an emergency (no better choice).

Below is a z/VM system that has been IPLed with "`LOADPARM SYSC`", where `SYSC` is the name CP gives to the integrated console. You may also hear talk about `SYSG`, which is the Integrated 3270 console on the HMC.

What you see is the "primary system operator", whose ID can be changed in the `SYSTEM_USERIDS` statement in the z/VM system configuration file. The default id for this is `OPERATOR`.



Screenshot 29 - Operating System Messages on the Hardware Management Console (HMC)

Note the checkboxes and the command line. Each message can be deleted individually since CP sends each line separately, not as a formal multi-line messages. You can type CMS or CP commands in the command line and see the responses.

This interface is virtualized in VM. But because a virtual machine can have both a virtual console AND connectivity to the integrated console, there has to be a way to cause commands to go to one instead of the other. Normally you type commands and they go to the virtual console device (at virtual address 01F or 009, for example). But to get that data to go to the virtual line mode console, you use the `#CP VINPUT VMSG` command.

You can use SCIF to send line-mode data to a virtual machine 3215 console or integrated console by choosing between

```
CP SEND userid query disk
```
(for 3215)

and

```
CP SEND CP userid VINPUT VMSG query disk
```
(for integrated console)

Knowing which flavor of CP SEND to use involves the target guest's configuration. CMS, for example, only knows how to talk to the virtual 3215 console.

Appendix D – Example SHUTZOS EXEC

```
/* Sample shutdown exec */
parse arg user
/*Now, stop JES2 initiators:*/
'CP SEND CP' user 'VINPUT VMSG $PI'
/*Next, stop the NFS client:*/
'CP SEND CP' user 'VINPUT VMSG P CSMVSNFS'
'CP SEND CP' user 'VINPUT VMSG C CORCHECK'
'CP SEND CP' user 'VINPUT VMSG P IGVDGNPP'
/*Now stop the rest of the TCP/IP stuff:*/
'CP SEND CP' user 'VINPUT VMSG P TN3270'
'CP SEND CP' user 'VINPUT VMSG P CS390IP'
'CP SEND CP' user 'VINPUT VMSG P PAGENT'
'CP SEND CP' user 'VINPUT VMSG P CSFTPD1'
'CP SEND CP' user 'VINPUT VMSG P CSFTPOE1'
'CP SEND CP' user 'VINPUT VMSG C USINETD1'
'CP SEND CP' user 'VINPUT VMSG P SYSLOGD'
/*Now stop APPC, ASCH, and VTAM (including CNET):*/
'CP SEND CP' user 'VINPUT VMSG C ASCH'
'CP SEND CP' user 'VINPUT VMSG C APPC'
'CP SEND CP' user 'VINPUT VMSG  Z NET,QUICK'
/*Stop TSO:*/
'CP SEND CP' user 'VINPUT VMSG  FORCE TSO,ARM'
'CP SEND CP' user 'VINPUT VMSG  P SDSF'
/*Shut down USS (OMVS):*/
'CP SEND CP' user 'VINPUT VMSG  F OMVS,SHUTDOWN'
/*Stop the NJE lines:*/
'CP SEND CP' user 'VINPUT VMSG $PLNE1-*'
/*Stop VLF and LLA:*/
'CP SEND CP' user 'VINPUT VMSG P VLF'
'CP SEND CP' user 'VINPUT VMSG P LLA'
'CP SEND CP' user 'VINPUT VMSG C SOFV3VS2'
'CP SEND CP' user 'VINPUT VMSG P HZR'
'CP SEND CP' user 'VINPUT VMSG D A,L'
```

IBM Corporation
IBM Systems
1 New Orchard Rd.
Armonk, NY 10504

Produced in the United States of America
August 2020

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

Please Recycle

For more information

To learn more about IBM Operations Manager, contact your IBM representative or IBM Business Partner, or visit: https://www.ibm.com/products/operations-manager-for-zvm

To learn more about IBM z/OS, contact your IBM representative or IBM Business Partner, or visit: https://www.ibm.com/it-infrastructure/z/zos