

**Hands-on Lab:
Operations Manager for z/VM
Lab Exercises
SHARE, August 2014
Session 15731**



Reference information for all Operations Manager labs:

You each have a copy of the Operations Manager Administration Guide in PDF format on your desktop. The following chapter and page numbers will help you find documentation on each of the commands and configuration options used in these labs.

Help can be accessed by issuing:

HELP AGOM <command>

cms cmd()	<i>Chapter 5, page 42</i>
config	<i>Chapter 5, page 43</i>
viewcon	<i>Chapter 7, page 123</i>
viewlog	<i>Chapter 7, page 127</i>
viewspl	<i>Chapter 7, page 132</i>
defactn	<i>Chapter 6, page 71</i>
defrule	<i>Chapter 6, page 91</i>
defmmon	<i>Chapter 6, page 85</i>
suspend	<i>Chapter 6, page 117</i>
resume	<i>Chapter 6, page 110</i>
defsmon	<i>Chapter 6, page 99</i>
defpmon	<i>Chapter 6, page 88</i>

All Operations Manager commands that you issue from your user ID to the Operations Manager server are preceded by:

GOMCMD OPGMRM1

For example, from your user ID, the VIEWCON command would look like this:

GOMCMD OPGMRM1 VIEWCON USER(XXXXXXXXXX)

For your convenience, we have created the following shortcuts (via EXECs) that allow you to enter the most common commands more easily:

- **VIEWCON <userid>**
- **VIEWLOG**
- **IEWSPL**

In these labs, we'll be using the following user IDs:

- USER01 through USER15 – you'll each be assigned one of these user IDs to use during class.
- CLASSADM – this will be used by the instructor
- OPERPOT – think of this as the OPERATOR ID.
- DIRMPOT – think of this as the DIRMAINT user ID.
- RACFPOT – think of this as the RACFVM user ID.
- TCPOT – think of this as the TCPIP user ID.
- TEC3LNX and TEC4LNX – sample Linux guests
- OPMGRM1 – the master server for Operations Manager
- OPMGRS1 through OPMGRS4 – the action processing servers for Operations Manager

Summary of Labs:

Everyone must complete Lab 1:

- Lab 1: Update Operations Manager configuration information and use console, log, and spool file viewing

After Lab 1, you can choose which of the remaining labs you would like to do, depending on your interest in the various functions of Operations Manager.

For these labs, we'll use the concept of a central operations console (the user ID **OPERPOT**) for receiving important messages whenever a rule or monitor is triggered. Operations staff can then use Operations Manager to view/monitor just this central console for general status. You can think of this console as an "alternative OPERATOR" console. Once an alert/message appears on this central console, system programmers or operations staff can use Operations Manager to view individual service machine or Linux guest consoles to investigate the problem further.

With Operations Manager you can also take automated actions when rules or monitors are triggered. For example:

- Send an email to someone
- Send an alert to an enterprise event console (such as IBM Netcool/OMNibus)
- Issue some commands (or a REXX EXEC) to actually address the problem

In the lab environment, we don't have access to an external network, so emails and other types of alerts are not possible. See the instructor if you'd like a copy of a presentation that shows how to implement many of these other types of actions.

Following is a short description of each available lab. Once you complete Lab 1, you do these in any order.

- Lab 2: Customize console, log, and spool file viewing
- Lab 3: Take an action based on a console message
- Lab 4: Customize the display of console messages
- Lab 5: Monitor user ID's for logoff status, suspend and resume rules
- Lab 6: Monitor spool space
- Lab 7: Monitor page space

Lab 1: Update Operations Manager configuration information

The detailed steps of the lab start on page 6, but we'll first discuss some general information about this lab. Remember, don't try to issue commands until you get to the detailed steps on page 6.

Operations Manager allows you to update its configuration dynamically. This means you can add, delete, or change many configuration settings, such as rules, monitors, actions, and schedules, while the server is running. In this class, we are using the Shared File System to store all the Operations Manager configuration files. Each user has their own configuration file (such as USER01 CONFIG, USER02 CONFIG, etc.) in the following SFS directory:
OPSSFS:OPMGRM1.CONFIGURATION

For each USER nn , this directory is accessed as D.

To make changes to your configuration file, you will issue:

```
xedit usernn config d
```

replacing nn with your user number.

These USER nn configuration files are imbedded in the master configuration file (**OPMGRM1 CONFIG**), also located on your D disk. We've used the **CONFIG** command within the master configuration file to imbed each of your configuration files. As part of the lab, you will view this master file but you do not have the authority to update it.

Once you have made your updates to your user configuration file, you can tell Operations Manager to load your updates by issuing the following command:

```
gomcmd opmgrm1 config fn(opmgrm1),ft(config),fm(e),clear(all)
```

You **must** specify the main configuration file (OPMGRM1 CONFIG). **Do not** change it to the name of your configuration file.

Note that Operations Manager (the user ID OPMGRM1) accesses the OPMGRM1.CONFIGURATION directory as E, thus the E is specified as the file mode above.

We have updated your PROFILE EXEC to assign function keys to this command (and others) that will be useful throughout the labs. **F03 is set to the CONFIG command above.**

So as you go through the lab steps starting on page 6, to reload the configuration file, press F03 and hit ENTER.

When you create rules, schedules, actions, etc., you must give them unique names. In order to avoid names that conflict with names that other users in the class are using, please start all your rule, schedule, and action names with your user number. **For example, if you are USER05, start all your rule names with the digits 05,** such as:

```
DEFRULE NAME(05LAB1) ...
```

The format of the commands in your configuration file is flexible, and each person may have their own style.

- The operands on the commands can be in any order. Remember to use commas between each operand.
- Commands can be strung out on one line, or can be broken down into multiple lines (indenting lines 2+ if you prefer). Use the plus sign (+) as the continuation character.

For example, the following 2 entries have the same result:

- ```
DEFRULE NAME(16LAB3R) , +
 MATCH(*test text*) , +
 ACTION(16LAB3A)
```
- ```
DEFRULE NAME(16LAB3R) , ACTION(16LAB3A) , MATCH(*test text*)
```

Steps for Lab 1:

1. Edit your configuration file:
`xedit <USERnn> config d`
2. Add or update the comments to add some of your own text. All comments start with *.

Note: We aren't actually adding any configuration information yet. We're just updating the file with a comment. We'll add configuration information in other labs.
3. Save and exit:
`file`
4. Review the master configuration file
`xedit opmgrm1 config d`
5. Find the `CONFIG` statement that imbeds your file (near the bottom.)
6. Review other statements which we'll learn more about later, if you like.
7. Exit without saving:
`qquit`
8. Use **F3** to tell Operations Manager to reload the master configuration file, which imbeds your new file:
`gomcmd opmgrm1 config fn(opmgrm1),ft(config),fm(e),clear(all)`

Using the `VIEWCON` command, Operations Manager allows you to view the messages on any monitored console (such as a z/VM service machine or Linux guest) to which you are authorized. Remember that this is the "filtered" view of the console. All rules and associated actions have been applied, so messages may be suppressed, highlighted, held, etc.

In addition, you can use `VIEWLOG` to look at a copy of the Operations Manager log. The log contains all messages from all consoles being monitored. This view is unfiltered. The log also contains all messages from the Operations Manager (OPMGRM1) console, including confirmation messages about new configuration information, error messages, and messages indicating when a rule, action, or schedule is triggered.

Within `VIEWCON` and `VIEWLOG` you can scroll up and down, scroll left and right, search for text forward or backward, and toggle in and out of auto scroll mode. In addition, in `VIEWCON`, you can turn on or off the date and/or time, and acknowledge held messages.

You can also use the `VIEWSP` command to look at a list of spool files on the system.

Continued steps for Lab 1:

9. View the console of user ID `OPERPOT` by issuing the following command:
`viewcon operpot`

Notice in the lower right corner you are in autoscroll mode and the console you are viewing belongs to `OPERPOT`.

10. Use F6 to toggle through the options for turning on and off the date and/or time information.
11. Use F7 and F8 to scroll up and down. Notice that this takes you out of auto scroll mode.
12. Use F10 and F11 to scroll left and right.
13. Put a number on the command line when using F7, F8, F10, or F11. Notice that it will scroll the specified number of rows or columns.
14. Use the `\` (**backslash**) character to search backwards in the console for some specific text. The text you are looking for will be at the top of the window. Again, notice that this takes you out of auto scroll mode.
15. Use the `/` (**forward slash**) character to search forward in the console for some specific text. The text you are looking for will be at the top of the window.
16. Use the F2 key to toggle back to auto scroll mode.
17. Issue the command `ID` on the command line. Notice that it's the user ID `OPERPOT`, not your user ID, because the `ID` command ran on user `OPERPOT`.
18. Issue the command `Q DISK` on the command line. Notice that these are the disks accessed by the user ID `OPERPOT` which will be different than the disks you have accessed.
19. Use F3 to exit `VIEWCON`.
20. View the console of a Linux guest (`TEC3LNX` or `TEC4LNX`) by issuing the following command:
`viewcon TECnLNX`
- Replace "n" with 3 or 4.
21. Issue a simple Linux command to confirm you are on the Linux guest:
`echo hello`
- Notice the guest responds asking for the login password. We won't login and issue any real Linux commands since these are running guests.
22. Use F3 to exit `VIEWCON`.

23. View the Operations Manager log by issuing the following command:
viewlog
24. Similar to **VIEWCON**, use function keys, /, and \ to move around in the log.
25. Use F3 to exit **VIEWLOG**.
26. View the spool files currently on the system by issuing the following command:
viewspl
27. Similar to **VIEWLOG**, use function keys F7 and F8 to scroll up and down.
28. Sort the list of files on any of the columns by placing the cursor on the column header and using F5 (sort ascending) and F6 (sort descending).
29. Place the cursor on the appropriate row and use F2 to view the contents of any spool file that does not have an asterisk next to it. (Those with an asterisk are open spool files and can not be viewed until they are closed.)
30. Use F3 to exit **IEWSPL**.

******* End of Lab 1 *******

Lab 2: Customize console, log, and spool file viewing

You can customize your **VIEWCON**, **VIEWLOG**, and **IEWSPL** sessions by creating a **PROFILE** that specifies how function keys operate. You can also customize the default date/time display settings for **VIEWCON** and the default sort order for **IEWSPL**.

Default **PROFILES** for each of the commands already exist on the Y-disk.

Steps for Lab 2:

1. Create your own **VIEWCON** profile to customize your settings. Issue the commands:
copyfile viewcon profile y = = a
xedit viewcon profile a
2. Update the following line to customize your default date/time display:
FORMAT=n
where *n* is 0, 1, 2, 3, 4, or 5. Refer to the **VIEWCON** command in the Admin Guide for details. A value of 1 will result in only the time being displayed by default.
3. Update the following line to modify if or how the Function keys are displayed:
SHOWPF=Y,Y,AAI,CAI
Refer to the **VIEWCON** command in the Admin Guide for more details or if you want to choose different options for showing or hiding the Function keys. The statement above will show 12 Function keys and highlight them.
4. Update or insert the following line (repeat as necessary) to assign or reassign functions to specific keys:
PFnn=action
where *nn* is a 2-digit function key number and *action* is a valid Function key action or any valid command during a **VIEWCON** session. Refer to the **VIEWCON** command in the Admin Guide for details.
5. Save your changes and close the file. Issue:
file
6. Issue **VIEWCON** again. Notice and use your customizations.
7. Perform similar customization of your **VIEWLOG** session by creating a **VIEWLOG PROFILE** on your A-disk, copying your **VIEWCON PROFILE** or the **VIEWLOG PROFILE** from the Y-disk.

Note that you can not use the **FORMAT** option in **VIEWLOG PROFILE**, but the **SHOWPF** and **PFnn=** options are still available.
8. Issue **VIEWLOG** again. Notice and use your customizations, then exit.

9. Use the **VIEWLOG** command with the **DATE()** option to retrieve a copy of yesterday's log. Using **RDRLIST**, find the file Operations Manager has sent you. **PEEK** it to see its contents.
10. Perform similar customization of your **IEWSPL** session by creating a **IEWSPL PROFILE** on your A-disk, copying your **VIEWCON PROFILE** or the **IEWSPL PROFILE** from the Y-disk.

Note that you can not use the **FORMAT** option in **IEWSPL PROFILE**. However, you can use a **SORT** statement. For example, the following line in **IEWSPL PROFILE** will sort the list of files in ascending order based on owning user ID.

```
SORT=OWNER,A
```

Once you are in **IEWSPL**, you can still change the sorting as needed. This is just the default.

11. Issue **IEWSPL** again. Notice and use your customizations, then exit.
12. Filter the list of spool files displayed by adding the **USER**, **DATE**, or **SIZE** parameter. For example:

```
viewspl size(1m)
```

Refer to the **IEWSPL** command in the Admin Guide for details. Remember the Admin Guide is on your desktop.

13. Use **F3** to exit **IEWSPL**.

Optional steps to see new functions in V1.5.

14. View a specific date/time range of the console of user ID **OPERPOT** by issuing the following command:

```
viewcon operpot daterng(2014/mm/dd-hh:mm 2014/mm/dd-hh:mm)
```
15. Use **F3** to exit **VIEWCON**.
16. View a specific date/time range of the Operations Manager log by issuing the following command:

```
viewlog daterng(2014/mm/dd-hh:mm 2014/mm/dd-hh:mm)
```
17. Use **F3** to exit **VIEWLOG**.

18. Query then change the CP date format for your user ID from SHORT format to ISO format:

```
cp query dateformat  
cp set dateformat isodate
```

19. View the console of user ID `OPERPOT`, view the Operations Manager log, and view spool files. Notice the date format displayed is now in ISO format instead of SHORT format.

```
viewcon operpot  
<F3 to exit>  
viewlog  
<F3 to exit>  
viewspl  
<F3 to exit>
```

******* End of Lab 2 *******

Lab 3: Take an action based on a console message

In this exercise, you will configure Operations Manager to take an action, such as send a message to another user or send an e-mail to yourself, when a specific message appears on a monitored console. An example might be to send a message to a central console when a specific error occurs on a Linux guest or z/VM service machine.

For this lab, we'll use the concept of a central operations console (the user ID `OPERPOT`) for receiving important messages whenever a rule or monitor is triggered. Operations staff can then use Operations Manager to view/monitor just this central console for general status. You can think of this console as an "alternative OPERATOR" console. Once an alert/message appears on this central console, system programmers or operations staff can use Operations Manager to view individual service machine or Linux guest consoles to investigate the problem further.

As Operations Manager monitors consoles, we'll create rules to look for specific messages. We'll start with simple text matching and then expand to columns and restricting it to specific user IDs. Exclusion can also be specified, but won't be covered in the lab. **Keep in mind that you won't need to specify a value for all the options on `DEFRULE` and `DEFACTN`.** Just use the options you need for each step in the procedure.

To understand text matching, it's important to remember how wildcard characters work in z/VM:

- * means 0 to n characters
- % means one and only one character. You can use multiple of these to represent, for example, exactly 2 characters
- # means one and only one numeric character (0-9)
- @ means one and only one hex character (0-9, A-F, a-f)
- & means one and only one alphabetic character (A-Z, a-z)
- \" is the escape character for matching any of the above characters

NOTE: Unless you are matching the full text of the message or the full text in the specified columns, you must specify the * character before and after the text specified in the **MATCH** value.

For example:

```
DEFRULE NAME( 05LAB1 ),MATCH(*text*)...
```

Refer to the `DEFRULE` and `DEFACTN` commands in the Admin Guide for syntax and details about the required commands for this lab.

Steps for Lab 3:

1. Edit your configuration file:
x usernn config d
2. Add a rule (using **DEFRULE**) that looks for the text similar to “**USERnn says DIRMPOT is having a problem**”, replacing *nn* with your user number. Assign an action to that rule. Remember to put your user number at the beginning of both the rule and action name.
3. Add an action (using **DEFACTN**) that issues the **CP MSG** or **CP MSGNOH** command to send a message to the central operations console (user ID **OPERPOT**.) Say something like “**USERnn says error on DIRMPOT.**” Be sure to put your user name/number in the text of your message to make it unique.

Important note: Be sure that this message is different than the message text you are looking for in the rule from Step 2. Otherwise, an infinite loop will occur in Operations Manager. More advanced users can look at the **EXUSER** option on **DEFRULE** and the **&T** substitution variable available in **DEFACTN** if you want to forward the message to **OPERPOT** as-is.

The syntax of **CP MSG** is simply **CP MSG <userid> <text of the message>**

4. File your changes.
5. User F3 to tell Operations Manager to load your new rule and action using the **CONFIG** command.
6. Force a message on the console of some monitored service machine, **DIRMPOT** in our case:

```
msg dirmpot USERnn says DIRMPOT is having a problem, notify operations
```

Be sure to replace *nn* with your user number and make sure the message text includes at least the text you specified in your rule in step 2 above.

7. To view the console of the service machine having the problem (**DIRMPOT**), issue:
viewcon dirmpot

You should see the message you specified in step 6 above: **USERnn says DIRMPOT is ...**

Exit **VIEWCON**.

8. To view the central operations console (**OPERPOT**), issue:
viewcon operpot

You should see the message you specified in step 3 above: **USERnn says error on DIRMPOT.**

Exit **VIEWCON**.

9. If you didn't see your message, you can debug the problem by:

- Viewing the console of **DIRMPOT** (using "VIEWCON") to make sure it received your message and the text of the message is what you intended.
- Viewing the Operations Manager log (using "VIEWLOG") to make sure your rule and action were loaded successfully and to see if your rule was triggered.
- Viewing your rule in the **USERnn CONFIG** file to correct your **MATCH** parameter.

10. Update your rule to restrict the "USERnn ..." text to columns 1 through 65.

11. Use F3 to tell Operations Manager to load your updated rule.

12. Issue:

```
msg dirmpot USERnn says DIRMPOT is having a problem, testing columns
```

13. View the console of **OPERPOT** to see if your updated rule was triggered and the message forwarded to **OPERPOT**. You should see the message you specified in step 3 above. Verify the timestamp to make sure this was triggered by the message in step 12 and not from step 6.

14. Issue:

```
msg dirmpot here are some extra columns before USERnn says ...
```

Be sure to include the full text of the message.

15. Again, view the console of **OPERPOT** to see if your updated rule was triggered and the message forwarded to **OPERPOT**. This time you should **not** see another occurrence of the message you specified in step 3 above because the text you're looking for is not in the right columns.

16. Create a new rule that looks for a message containing the word "abend", but only on the console of **DIRMPOT**. (Hint: see the **USER** parameter on **DEFRULE**.) Be sure the rule also looks for your user ID in the text of the message to make it unique. Assign an action to the rule.

17. Create an action (using **DEFACTN**) that calls the **SMTPNOTE EXEC** to send an e-mail to your user ID or the user id **MAILDEMO**. (We don't have access to the external network from this lab system.) The email will contain the user ID that had the error (**DIRMPOT**) and the text of the error message. The syntax for **SMTPNOTE** is:

```
SMTPNOTE <email> AT <nodeid> <userid> <errtype> <message text>
```

In this case, we'll just use:

```
SMTPNOTE <USERnn or MAILDEMO> AT ZTECZVM <userid> <errtype> <message text>
```

View (XEDIT) the **SMTPNOTE EXEC** to see a description of each parameter. It's on your D disk.

Review the **DEFACTN** statement in the Operations Manager Admin Guide to learn about special

variables that are available (such as &U and &T). This will help you pass the user ID that displayed the message and the message text to the **SMTPNOTE EXEC**.

18. Use F3 to tell Operations Manager to load your updated rule.

19. Issue:

```
msg dirmpot hello from USERnn this is an abend
```

20. View the console of **DIRMPOT**. You should see the message you specified in step 19 above. You should also see that the message triggered your rule.

21. Verify that your user ID has received the email, or have the instructor verify that MAILDEMO received your email. It should contain the text of the message. Alternatively, you can view the spool files of the user MAILDEMO and see if yours is there.

```
viewspl user(maildemo)
```

22. Issue:

```
msg racfpot hello from USERnn abend testing for specific user
```

23. View the console of **RACFPOT**. You should see the message you specified in step 22 above, but no rule should be triggered. You should not get an e-mail because the message was on **RACFPOT** and your rule should only apply to **DIRMPOT**.

Keep in mind that the above examples use a very simple action of sending a message to another console or calling an EXEC to send an e-mail. The CP **MSG** and **SMTPNOTE** commands in the **DEFACTN** statement can be replaced with any other CP or CMS command, or a call to a REXX EXEC.

******* End of Lab 3 *******

Lab 4: Customize the display of console messages

In this exercise, you will configure Operations Manager to suppress, highlight, and hold specific messages in the console view. Note that this will not affect the view of the log.

For this lab, we'll use the concept of a central operations console (the user ID **OPERPOT**) for receiving important messages whenever a rule or monitor is triggered. Operations staff can then use Operations Manager to view/monitor just this central console for general status. You can think of this console as an "alternative OPERATOR" console. Once an alert/message appears on this central console, system programmers or operations staff can use Operations Manager to view individual service machine or Linux guest consoles to investigate the problem further.

Refer to the **DEFRULE** and **DEFACTN** commands in the Admin Guide for syntax and details about the required commands for this lab.

Steps for Lab 4:

1. Create a rule that looks for the text "**extra noise*USERnn**" on the console of user ID **OPERPOT**. Be sure to include your user number in the text of the message to avoid conflict with other members of the class. Assign a new action to this rule.
2. Using the **INPUT** parameter on **DEFACTN**, create an action that suppresses the message from the console view of **OPERPOT**.
3. Use F3 to tell Operations Manager to load your new rule and action.
4. Issue:
msg operpot please show this message for USERnn

Be sure to replace *nn* with your user number.
5. View the console of **OPERPOT**.
6. You should see the message you sent in step 4 above. Exit **VIEWCON**.
7. Issue:
msg operpot this message is just extra noise for USERnn

Be sure to replace *nn* with your user number.
8. View the console of **OPERPOT**.
9. You should **not** see the message you sent in step 7 above because your rule and action suppressed it. Exit **VIEWCON**.

10. Create a new rule and corresponding action to highlight a message on **OPERPOT**, or change it to red. Test your new message. Be sure to include "USERnn" in the **MATCH** parameter of your rule and put your user name/number in the message you send to **OPERPOT**.
11. Create a new rule and corresponding action to hold a message and make it pink on **OPERPOT**. Again, put your user name/number in the text of the message you send. Test your message using **VIEWCON**. You will need to send multiple other messages that don't have the word "hold" in them to see them flow by while the held message stays at the top.

You can acknowledge (or unhold) the message in **VIEWCON** by placing the cursor on the message and hitting the F5 (or whatever function key you assigned to **HOLD** in **VIEWCON PROFILE**.) **Be sure to only acknowledge your message, so other users will still see their messages held.**

******* End of Lab 4 *******

Lab 5: Monitor user ID's for logoff status, suspend and resume rules

In this exercise, you will configure Operations Manager to automatically start a service machine if it is in logged off status. This can be used to start service machines and guests in a specific order.

Because multiple people will be creating these same monitors at the same time, we won't actually restart a service machine if it's logged off. Instead, we'll issue a message to the central operations console **OPERPOT**. The instructor can demonstrate how to automate the restart response if you are interested.

Since we aren't actually restarting the service machine, we want to only activate your rule temporarily. Otherwise, it will continue to be triggered and messages will continue to be sent to **OPERPOT**.

Here's the general order of events you are trying to create:

- Create a new machine monitor for this lab to check the status of **TCPOT** at the defined intervals, and trigger an action if **TCPOT** is found logged off.
- In your configuration file, suspend this monitor to it's generally not active.
- From the CMS command line, issue a command to dynamically resume your monitor.
- When Operations Manager sees that **TCPOT** is logged off, it will trigger an action that sends a message to **OPERPOT**.
- When Operations Manager sees the message on **OPERPOT**, it will suspend your rule that checks for logoff status of **TCPOT**. This will prevent continued messages on **OPERPOT** since we aren't really restarting **TCPOT**.

Refer to the **DEFMMON**, **DEFACTN**, **SUSPEND**, and **RESUME** commands in the Admin Guide for syntax and details about the required commands for this lab.

Steps for Lab 5:

1. In your configuration file, create a machine monitor (using **DEFMMON**) that checks the status of user ID **TCPOT** and triggers an action if the user ID is logged off. Assign a new action to this monitor.
2. Create an entry in your configuration file to suspend this monitor, using the **SUSPEND** command. Make sure this entry is listed in the configuration file after (or below) the monitor you created in step 1. I.e. you can't suspend a monitor that hasn't been defined yet.
3. Create an action (use the name you gave it in step 1) that sends a message to the user ID **OPERPOT** indicating that **TCPOT** is logged off. Use an Operations Manager substitution variable to specify **TCPOT**, rather than hardcoding the logged off user ID in the text of the message. Put your user name/number in the text of your message to make it unique.
4. Create a rule that looks for your message text (from step 3) on user ID **OPERPOT**. Assign a new action to this rule.

5. Create an action (use the name you gave it in step 4) that suspends your monitor from step 1. **Note** you must specify GOM for the ENV value on the DEFACTN statement in order for the command to be executed as an Operations Manager command.

6. User F3 to tell Operations Manager to load your new monitor, rules, and actions.

7. From the CMS command line, tell Operations Manager to resume the machine monitor you created in Step 1. For example:

```
gomcmd opmgrml resume machine(XXXXXXXXXX)
```

This will resume the monitor that checks to see if TCPOT is logged off.

8. View the console of OPERPOT.

9. You should see your message from step 3 above. Remember that this can be delayed up to 30 seconds based on how often Operations Manager checks the monitors.

10. Keep watching the console of OPERPOT. You should **not** see your message at the interval (the default is 1 minute) because the machine monitor should be suspended.

11. If it didn't work, debug by looking at the log and updating your configuration file as needed. Note that in order to test it again, you will have to resume your machine monitor again (Step 7).

***** End of Lab 5 *****

Lab 6: Monitor spool space

In this exercise, you will configure Operations Manager to automatically send a message to the central console if the percent of spool space used reaches a threshold range.

For this lab, we'll use the concept of a central operations console (the user ID **OPERPOT**) for receiving important messages whenever a rule or monitor is triggered. Operations staff can then use Operations Manager to view/monitor just this central console for general status. You can think of this console as an "alternative OPERATOR" console.

Refer to the **DEFSMON**, **DEFACTN**, and **SUSPEND** commands in the Admin Guide for syntax and details about the required commands for this lab.

Steps for Lab 6:

1. Use Operations Manager to determine how full the spool is on our class system:
viewspl

The percent full is in the upper center part of the panel.

2. In your configuration file, create a spool monitor (using **DEFSMON**) that checks the percent full of the spool. Specify a range that includes the current percent full. Assign a new action to this spool monitor.

If you don't specify an **INTERVAL**, then by default Operations Manager will check every 1 minute.

3. Create an action (use the name you gave it in step 2) that sends a message to the user ID **OPERPOT** indicating that the spool is getting full. Include in the message:
 - The actual percent full of the spool – you can do this by using Operations Manager substitution variables. You can find these substitution variables (such as **&4**) in the documentation for **DEFSMON**.
 - Your user name/number in the text of your message to make it unique.
4. Use **F3** to tell Operations Manager to load your new monitor and action.
5. View the console of **OPERPOT**. You should see your message from step 3 above. Remember that this can be delayed up to 30 seconds based on how often Operations Manager checks the monitors.
6. Once you have seen your message at least once (two or three times if you chose a small interval and want to wait), suspend your monitor to avoid cluttering the console of **OPERPOT**:
gomcmd opmgrml suspend spool(<your spool monitor name>)

7. For more advanced functions, update your spool monitor to use the LIMIT operand to restrict your messages on **OPERPOT** to a maximum of 3 per hour. Remember to reload the configuration file once you have updated your spool monitor.
8. View the console of **OPERPOT** to confirm the message is displayed only 3 times.

***** **End of Lab 6** *****

Lab 7: Monitor page space

In this exercise, you will configure Operations Manager to automatically send a message to the central console if the percent of page space used reaches a threshold range.

For this lab, we'll use the concept of a central operations console (the user ID **OPERPOT**) for receiving important messages whenever a rule or monitor is triggered. Operations staff can then use Operations Manager to view/monitor just this central console for general status. You can think of this console as an "alternative OPERATOR" console.

Refer to the **DEFPMON**, **DEFACTN**, and **SUSPEND** commands in the Admin Guide for syntax and details about the required commands for this lab.

Steps for Lab 7:

1. To determine how full the page space is on our class system, you can use the `Q ALLOC PAGE` command. However, this command requires privilege class D, which your user ID does not have. Operations Manager (OPMGRM1) does have the authority to issue this command, so to issue the command from Operations Manager, do the following

- a. View the Operations Manager log (which is really the console of OPMGRM1):
viewlog
- b. Query the page space from OPMGRM1 (since your user ID is not authorized to issue this command):
cp cmd(q alloc page)

The percent full is on the summary line. You will probably have to scroll to the right to see it.

2. Create a page monitor (using **DEFPMON**) that checks the percent full of the page space. Specify a range that includes the current percent full (even if you have to specify a minimum of 000.) Assign a new action to this page monitor.

If you don't specify an **INTERVAL**, then by default Operations Manager will check every 1 minute.

3. Create an action (use the name you gave it in step 2) that sends a message to the user ID **OPERPOT** indicating that the page space is getting full. Include in the message:

- The actual percent full of the page space – you can do this by using Operations Manager substitution variables. You can find these substitution variables (such as &4) in the documentation for **DEFPMON**.
- Your user name/number to make the message unique.

4. Use F3 to tell Operations Manager to load your new monitor and action.

5. View the console of **OPERPOT**. You should see your message from step 3 above. Remember that this can be delayed up to 30 seconds based on how often Operations Manager checks the monitors.
6. Once you have seen your message at least once (two or three times if you chose a small interval and want to wait), suspend your monitor to avoid cluttering the console of **OPERPOT**:
gomcmd opmgrm1 suspend page(<your page monitor name>)
7. For more advanced functions, update your page monitor to use the **LIMIT** operand to restrict your messages on **OPERPOT** to a maximum of 3 per hour. Remember to reload the configuration file once you have updated your page monitor.
8. View the console of **OPERPOT** to confirm the message is displayed only 3 times.

******* End of Lab 7 *******

Answers:

You can find sample solutions to the labs in the file `USER16 CONFIG D`. If you'd like a copy of this file, email Tracy Dean (tld1@us.ibm.com) or bring a flash drive to the instructor and he/she can copy it to your flash drive.