

**Hands-on Lab  
Infrastructure Suite for z/VM and Linux**

**Operations Manager for z/VM  
Lab Exercises**

**August/September 2015**

## 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>**

<b>cms cmd()</b>	<b><i>Chapter 5, page 42</i></b>
<b>config</b>	<b><i>Chapter 5, page 43</i></b>
<b>viewcon</b>	<b><i>Chapter 7, page 127</i></b>
<b>viewlog</b>	<b><i>Chapter 7, page 131</i></b>
<b>viewspl</b>	<b><i>Chapter 7, page 136</i></b>
<b>defactn</b>	<b><i>Chapter 6, page 71</i></b>
<b>defrule</b>	<b><i>Chapter 6, page 94</i></b>
<b>suspend</b>	<b><i>Chapter 6, page 121</i></b>
<b>resume</b>	<b><i>Chapter 6, page 114</i></b>
<b>defsmon</b>	<b><i>Chapter 6, page 102</i></b>
<b>defpmon</b>	<b><i>Chapter 6, page 90</i></b>

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

```
GOMCMD OPMGRM1
```

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

```
GOMCMD OPMGRM1 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**
- **VIEWSP**

In these labs, we'll be using the following user IDs on system TIVLP3:

- **USER01** through **USER20** – you'll each be assigned one of these user IDs to use during class. Your **password** is **IBMCLASS**.
- CLASSADM – this will be used by the instructor
- OPERDEMO – think of this as the OPERATOR ID.
- DIRMDEMO – think of this as the DIRMAINT user ID.
- RACFDEMO – think of this as the RACFVM user ID.
- TCPDEMO – think of this as the TCPIP user ID.
- RHEL1, RHEL4, and RHEL5 – sample Linux guest
- OPMGRM1 – the master server for Operations Manager
- OPMGRS1 through OPMGRS4 – the action processing servers for Operations Manager

## Summary of Labs:

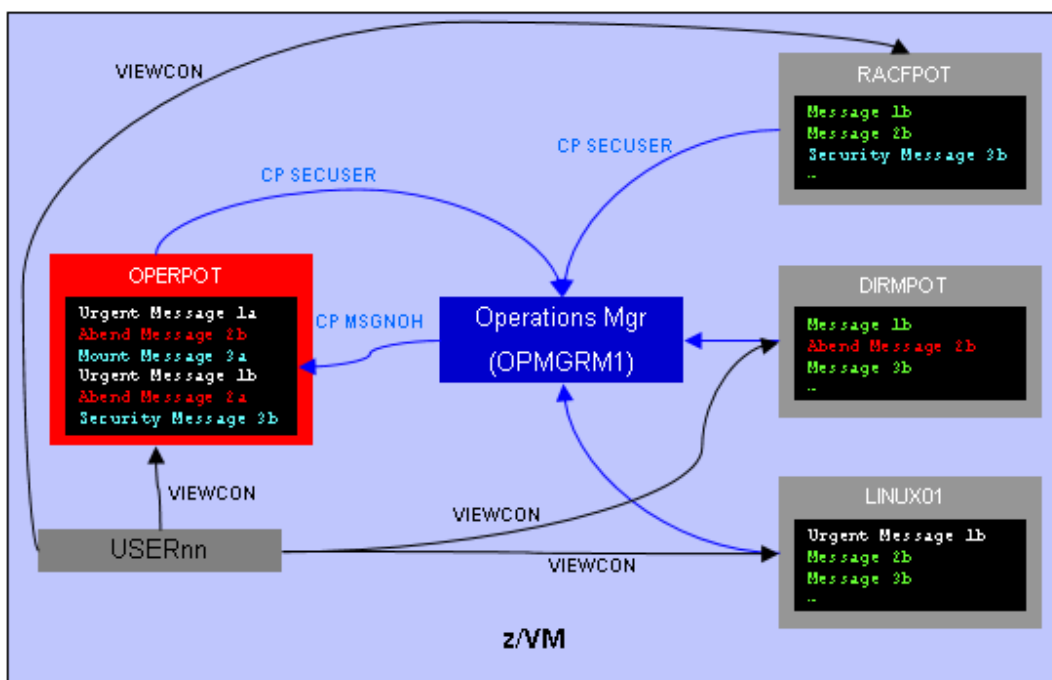
### Everyone must complete Lab 1:

- 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 **OPERDEMO**) 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 the "OPERATOR" console for this Proof of Technology (POT.) 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.

During the labs you will be configuring Operations Manager to send alerts to **OPERDEMO**. You can then view the console of **OPERDEMO** or any other service machines specifically for the POT (**xxxxxPOT**) from your userid **USERnn**.



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 can do these in any order. You will **not** have time to complete all of them.

- Lab 2: View consoles and 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 spool space and suspend rules/monitors
- Lab 6: 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 USERnn, 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 USERnn 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 (also set to F3):

```
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. **F3 is set to the CONFIG command above.**

**So as you go through the lab steps starting on page 8, to reload the configuration file, press F3 and press 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)`
9. Additional steps for Lab 1 are continued on the next page. But first, some background on the next section.

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:

10. View the console of user ID `OPERDEMO` by issuing the following command:

```
viewcon OPERDEMO
```

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

11. Use F6 to toggle through the options for turning on and off the date and/or time information.

12. Use F7 and F8 to scroll up and down. Notice that this takes you out of auto scroll mode.

13. Use F10 and F11 to scroll left and right.

14. 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.

15. 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.

16. 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.

17. Use the F2 key to toggle back to auto scroll mode.

18. Issue the command `ID` on the command line. Notice that it's the user ID `OPERDEMO`, not your user ID, because the `ID` command ran on user `OPERDEMO`.

19. Issue the command `Q DISK` on the command line. Notice that these are the disks accessed by the user ID `OPERDEMO` which will be different than the disks you have accessed.

20. Use F3 to exit `VIEWCON`.

21. View the console of a Linux guest `RHEL5` by issuing the following command:

```
viewcon rhel5
```

Replace "5" with 4 to see an additional Linux guest consoles.

22. **NOTE:** Do **not** enter long running commands like “top” or commands that require a full screen (vs line mode) interface. Also remember this guest is being used by other members of the class so don’t issue any commands that disable it.

Issue a simple Linux command to confirm you are on the Linux guest:

```
free -m  
df -h
```

Notice the line that Operations Manager adds to the console indicating that your user ID issued the command.

23. Optionally, use IBM Wave to recycle the Linux guest and watch its console in Operations Manager:

- Start the Wave interface in a browser by going to: <http://192.84.47.77>
- Login with userid ADMnn (nn is 01-20) and password IC15WAVE (case sensitive).
- In the upper left pane click on “Hardware Viewer”, then select the TIVLP3 LPAR.
- A graphical view of users on the LPAR will appear, based on some predefined groups.
- In the upper right of the main pane, click on “Show Filter Panel” which displays a new pane on the bottom of the screen.
- In the filter pane, change “Account” to “Guest name” and change the filter value to RH\*, then click Go.
- Double click on the USER-LOCAL (TIVLP3) group to see the guests in this group that start with RH.
- Right click on RHEL5 and select Recycle, then Go.
- Return to your 3270 session and watch the guest’s console (using VIEWCON) as it shuts down and restarts.

24. Use F3 to exit **VIEWCON**.

25. View two user consoles together with messages from each interspersed in data and timestamp order:

```
viewcon users(operdemo,rhell)
```

Notice the additional column that indicates which user each message came from.

26. View the Operations Manager log by issuing the following command:

```
viewlog
```

27. Similar to **VIEWCON**, use function keys, /, and \ to move around in the log.

28. Use F3 to exit **VIEWLOG**.

29. View the spool files currently on the system by issuing the following command:

**viewspl**

30. Note the following notations on some of the spool file numbers:

- \* (asterisk) means the spool file is still open and therefore the contents can not be viewed
- + (plus sign) means the spool file is not owned by this member of the SSI cluster. However, it is visible on this member because the owning user is currently logged onto this member

31. Similar to **VIEWLOG**, use function keys F7 and F8 to scroll up and down.

32. 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).

33. Place the cursor on one of the rows and use F2 to view the contents of any spool file that does not have an asterisk next to it. (Remember, those with an asterisk are open spool files and can not be viewed until they are closed.)

34. Use F3 to exit **VIEWSP**.

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

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

You can customize your **VIEWCON**, **VIEWLOG**, and **VIEWSP** 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 **VIEWSP**.

Default **PROFILES** for each of the **VIEWxxx** commands already exist on the Y-disk. In this lab you will create your own on your A-disk to be used instead of the one 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. A value of 1 will result in only the time being displayed by default. If you change the value to 2, then you will see date and time by default in **VIEWCON**.

Refer to the **VIEWCON** command in the Admin Guide for more details about the options.

3. Update the following line to modify if or how the Function keys are displayed:  
**SHOWPF=Y, Y, AAI, CAI**  
The statement above will show 12 Function keys and highlight them. To show 12 Function keys and make them red, for example, specify:  
**SHOWPF=Y, Y, AAI, CRE**

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

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. For example, you can specify the following to set F4 to go to the top of the console view:  
**PF04=TOP**

Refer to the **VIEWCON** command in the Admin Guide for details. Usage note 10 on page 129 lists all the available actions.

5. Save your changes and close the file. Issue:  
**file**

6. Issue **VIEWCON** again. Notice and use your customizations.
7. F3 to exit **VIEWCON**.
8. Perform similar customization of your **VIEWLOG** session by creating a **VIEWLOG PROFILE** on your A-disk, copying the **VIEWLOG PROFILE** from the Y-disk.  
**copyfile viewlog profile y = = a**  
**xedit viewlog profile a**

Note that you can not use the **FORMAT** option in **VIEWLOG PROFILE**, but the **SHOWPF** and **PFnn=** options are still available.

9. Save your changes and close the file. Issue:  
**file**
10. Issue **VIEWLOG** again. Notice and use your customizations, then exit.
11. F3 to exit **VIEWLOG**.
12. Use the **VIEWLOG** command with the **DATE( )** option to retrieve a copy of yesterday's log.  
**viewlog date(2015/mm/dd)**
13. Find the file Operations Manager has sent you.  
**rdrlist**
14. Next to the file, type the following command (or use F11) to see the contents:  
**peek**  
  
F3 to exit **PEEK** and F3 again to exit **RDRLIST**.
15. Perform similar customization of your **IEWSPL** session by creating a **IEWSPL PROFILE** on your A-disk, copying the **IEWSPL PROFILE** from the Y-disk.  
**copyfile viewspl profile y = = a**  
**xedit viewspl profile a**

Note that you can not use the **FORMAT** option in **IEWSPL PROFILE**. However, you can use a **SORT** statement. 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.

To change the default to sort on the size column, in descending order, specify:

**SORT=SIZE,D**

Refer to the **IEWSPL** command in the Admin Guide for more options – see item 2 on page 141.

16. Save your changes and close the file. Issue:  
**file**

17. Issue **VIEWSPPL** again. Notice and use your customizations, then exit.

18. Filter the list of spool files displayed by adding the **USER**, **DATE**, or **SIZE** parameter.

For example, the following shows only spool files whose size are 1MB or larger:

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

Either of the following shows only spool files owned by user IDs with with **DEMO** in their name:

```
viewspl user(*demo*)
```

```
viewspl *demo*
```

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

19. Use **F3** to exit **VIEWSPPL**.

#### **Optional steps to see additional functions:**

20. View a specific date/time range of the console of user ID **OPERDEMO** by issuing the following command. Specify a day and time within the last 24 hours:

```
viewcon OPERDEMO daterng(2015/mm/dd-hh:mm 2015/mm/dd-hh:mm)
```

21. Use **F3** to exit **VIEWCON**.

22. View a specific date/time range of the Operations Manager log by issuing the following command. Again, specify a day and time within the last 24 hours:

```
viewlog daterng(2015/mm/dd-hh:mm 2015/mm/dd-hh:mm)
```

23. Use **F3** to exit **VIEWLOG**.

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

```
cp query dateformat
```

```
cp set dateformat isodate
```

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

```
viewcon OPERDEMO
```

```
<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 `OPERDEMO`) 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 DIRMDEMO 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.

For example (replace *nn* with your user number):

```
*
DEFRULE NAME(nnLAB3A),+
  MATCH(*USERnn says DIRMDEMO is having a problem*),+
  ACTION(nnLAB3A1)
```

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 **OPERDEMO**.) Say something like “**USERnn says error on DIRMDEMO.**” 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 **OPERDEMO** as-is.

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

For example (replace *nn* with your user number):

```
*
DEFACTN NAME(nnLAB3A1),+
  COMMAND(CP MSGNOH OPERDEMO USERnn says error on DIRMDEMO.),+
  ENV(SVM)
```

4. Save your changes and exit:  
**file**
5. Use 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, **DIRMDEMO** in our case:

```
msg DIRMDEMO USERnn says DIRMDEMO 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 (**DIRMDEMO**), issue:  
**viewcon DIRMDEMO**



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

F3 to exit `VIEWCON`.

8. To see if your rule and action were triggered, view the central operations console (`OPERDEMO`):  
`viewcon OPERDEMO`

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

F3 to exit `VIEWCON`.

9. If you didn't see your message, you can debug the problem by:
  - Viewing the console of `DIRMDEMO` (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.
  - Editing your rule in the `USERnn` CONFIG file to correct your MATCH parameter.

10. If your first test was successful, proceed to the rest of the lab.

11. Edit your configuration file:  
`x usernn config d`

12. Update your existing rule to restrict the "`USERnn ...`" text to columns 1 through 65.

For example (replace `nn` with your user number):

```
*  
DEFRULE NAME(nnLAB3A) ,+  
  MATCH(*USERnn says DIRMDEMO is having a problem*),+  
  MCOL(001:065) ,+  
  ACTION(nnLAB3A1)
```

13. Save your changes and exit:  
`file`

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

15. To test your updated rule issue:  
`msg DIRMDEMO USERnn says DIRMDEMO is having a problem, testing columns`

16. To verify if it worked, view the console of the service machine having the problem (`DIRMDEMO`):  
`viewcon DIRMDEMO`

17. You should see the message you specified in step 12 above: `USERnn` says `DIRMDEMO` is ...

F3 to exit `VIEWCON`.

18. To see if your rule and action were triggered, view the central operations console (`OPERDEMO`):  
`viewcon OPERDEMO`

You should see the message you specified in step 15 above:  
`USERnn` says error on `DIRMDEMO` ...

Verify the timestamp to make sure this was triggered by the message in step 15, not from step 6.

F3 to exit `VIEWCON`.

19. Issue the following command to send another message to `DIRMDEMO`:

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

Be sure to include the full text of the message.

20. To verify the message was sent, view the console of the service machine having the problem (`DIRMDEMO`):

```
viewcon DIRMDEMO
```

F3 to exit `VIEWCON`.

21. Now, view the console of `OPERDEMO` to see if your rule was triggered and the message forwarded to `OPERDEMO`.

```
viewcon OPERDEMO
```

This time you should **not** see another occurrence of the message you specified in step 19 above because the text you're looking for is not in the right columns. Remember, your rule is only looking in columns 1 through 65.

F3 to exit `VIEWCON`.

22. If you wish to continue with more advanced options for console rules, please proceed. (Examples will **not** be provided.) If you wish to stop now, please proceed with another lab.

23. Edit your configuration file again and create a new rule that looks for a message containing the word "abend", but only on the console of `DIRMDEMO`. (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.

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

```
EXEC SMTPNOTE <yourid> AT <nodeid> <userid> <errtype> <message text>
```

In this case, we'll just use:

```
EXEC SMTPNOTE <USERnn> AT TIVLP3 <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**.

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

26. Issue:

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

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

28. Verify that your user ID has received the email:

```
rdrlist
```

29. Issue:

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

30. View the console of **RACFDEMO**. 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 **RACFDEMO** and your rule should only apply to **DIRMDEMO**.

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 **OPERDEMO**) 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. Edit your configuration file:  
**x usernn config d**
2. Create a rule and corresponding action to highlight a message on **OPERDEMO**, or change it to red. 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.

For example:

```
*  
DEFRULE NAME( nnLAB4A ) , +  
  MATCH( *show*usern* ) , +  
  USER( OPERDEMO ) , +  
  ACTION( nnLAB4A1 )
```

3. Using the **INPUT** parameter on **DEFACTN**, create an action that highlights the message or changes it to red in the console view of **OPERDEMO**.

For example:

```
*  
DEFACTN NAME( nnLAB4A1 ) , +  
  INPUT( AHI )
```

Or:

```
*  
DEFACTN NAME( nnLAB4A1 ) , +  
  INPUT( CRE )
```

4. Save your changes and exit:  
**file**
5. Use F3 to tell Operations Manager to load your new rule and action.
6. To test your rule, issue the following. Your message should be displayed on OPERDEMO.  
**msg OPERDEMO please show this message for USERnn**

Be sure to replace *nn* with your user number.

7. View the console of OPERDEMO.  
**viewcon OPERDEMO**

You should see the message highlighted or in red since it matches your rule.

F3 to exit **VIEWCON**.

8. Now create a **new** rule that looks for the text “**extra noise\*USERnn**” on the console of user ID **OPERDEMO**. 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.

For example:

```
*
DEFRULE NAME(nnLAB4B),+
  MATCH(*extra noise*usernn*),+
  USER(OPERDEMO),+
  ACTION(nnLAB4B1)
```

9. Using the **INPUT** parameter on **DEFACTN**, create an action that **suppresses** the message from the console view of **OPERDEMO**.

For example:

```
*
DEFACTN NAME(nnLAB4B1),+
  INPUT(SUP)
```

10. Save your changes and exit:  
**file**
11. Use F3 to tell Operations Manager to load your new rule and action.
12. To test your rule, issue the following. Your message should be displayed on OPERDEMO.  
**msg OPERDEMO please show this message for USERnn**

Be sure to replace *nn* with your user number.

13. View the console of OPERDEMO.

```
viewcon OPERDEMO
```

You should see the message since it doesn't match your new rule. (And of course the message should be highlighted or in red since it matches your earlier rule.)

F3 to exit VIEWCON.

14. Now test it again, this time sending a message that matches your new rule:

```
msg OPERDEMO this message is just extra noise for USERnn
```

Be sure to replace *nn* with your user number.

15. View the console of OPERDEMO.

```
viewcon OPERDEMO
```

You should **not** see the message you sent in step 8 above because your rule and action suppressed it.

F3 to exit VIEWCON.

16. If you wish to continue with more advance options for how messages are displayed, please proceed. (Examples will **not** be provided.) If you wish to stop now, please proceed with another lab.

17. Create a new rule and corresponding action to hold a message and make it pink on OPERDEMO. 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 pressing 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 spool space and suspend rules/monitors**

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 **OPERDEMO**) 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 5:**

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. Edit your configuration file:  
**x usernn config d**
3. Create a spool monitor (using **DEFSMON**) that checks the percent full of the spool. Specify a range that includes the current percent full. **Note:** both the low and high numbers for the **USAGE** parameter must be 3 digits.

Assign a new action to this spool monitor.

For example:

```
*  
DEFSMON NAME( nnLAB5 ), +  
  USAGE( xxx-100 ), +  
  ACTION( nnLAB5A )
```

Replace **xxx** with a number lower than the current percent full of the spool. If you don't specify the **INTERVAL** parameter, then by default Operations Manager will check every 1 minute.

4. Create an action (use the name you gave it in step 3) that sends a message to the user ID **OPERDEMO** 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 DEFSSMON.
- Your user name/number in the text of your message to make it unique.

For example:

```
*  
DEFACTN NAME(nnLAB5A),+  
  COMMAND(CP MSGNOH OPERDEMO USERnn says spool is &4% full.),+  
  ENV(SVM)
```

5. Save your changes and exit:  
**file**

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

7. View the console of **OPERDEMO**.  
**viewcon OPERDEMO**

You should see your message from step 4 above. Remember that this can be delayed up to 30 seconds based on how often Operations Manager checks the monitors.

F3 to exit **VIEWCON**.

8. 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 **OPERDEMO**:  
**gomcmd opmgrml suspend spool(<your spool monitor name>)**

Replace **<your spool monitor name>** with the name you assigned to your spool monitor in the DEFSSMON statement in step 3.

9. For more advanced functions, update your spool monitor to use the LIMIT operand to restrict your messages on **OPERDEMO** to a maximum of 3 per hour. Remember to reload the configuration file once you have updated your spool monitor.

10. View the console of **OPERDEMO** to confirm the message is displayed only 3 times.

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



## **Lab 6: 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 `OPERDEMO`) 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 `DEFFMON`, `DEFACTN`, and `SUSPEND` commands in the Admin Guide for syntax and details about the required commands for this lab.

### **Steps for Lab 6:**

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.

- c. F3 to exit `VIEWCON`.

2. Edit your configuration file:  
**x usernn config d**

3. 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.)  
**Note:** both the low and high numbers for the USAGE parameter must be 3 digits.

Assign a new action to this page monitor.

For example:

```
*  
DEFPMON NAME(nnLAB6),+  
  USAGE(xxx-100),+  
  ACTION(nnLAB6A)
```

Replace xxx with a number lower than the current percent full of page space. If you don't specify the INTERVAL parameter, then by default Operations Manager will check every 1 minute.

4. Create an action (use the name you gave it in step 3) that sends a message to the user ID **OPERDEMO** 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.

For example:

```
*  
DEFACTN NAME(nnLAB6A),+  
  COMMAND(CP MSGNOH OPERDEMO USERnn says page space is &4% full.),+  
  ENV(SVM)
```

5. Save your changes and exit:  
**file**

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

7. View the console of **OPERDEMO**.  
**viewcon OPERDEMO**

You should see your message from step 4 above. Remember that this can be delayed up to 30 seconds based on how often Operations Manager checks the monitors.

F3 to exit **VIEWCON**.

8. 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 **OPERDEMO**:  
**gomcmd opmgrml suspend page(<your page monitor name>)**

Replace **<your page monitor name>** with the name you assigned to your page space monitor in the DEFPMON statement in step 3.

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

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

## **Answers:**

You can find sample solutions to the labs in the file `USER21 CONFIG D`. If you'd like a copy of this file, email Tracy Dean ([tld1@us.ibm.com](mailto:tld1@us.ibm.com)) or bring a flash drive to the instructor and he/she can copy it to your flash drive.