Creating an Event Console with Automation for z/VM and Linux

Version Date: November 2018

Tracy Dean
Product Manager: IBM z/VM Tools

Special Notices

This document is a collaboration between IBM Advanced Technical Skills and the IBM Software Group organizations and reflects their understanding of IBM Operations Manager for z/VM. It was produced and reviewed by the members of the IBM Advanced Technical Support organization. This document is presented "As-Is" and IBM does not assume responsibility for the statements expressed herein. It reflects the opinions of the IBM Advanced Technical Support organization. These opinions are based on the authors' experiences. If you have questions about the contents of this document, please contact the author.

Trademarks

Introduction

One of the many growing pains for mainframe staff associated with introducing z/VM and Linux into an existing mainframe environment is the need to provide a traditional level of event management and automation typical of the z/OS systems that have existed for many years. With Linux being a distributed based solution, including Linux on IBM Z, event management and automation is often approached at the Linux virtual machine level only. While this may provide an acceptable level of Linux management, this approach is lacking in several areas:

- The z/VM hypervisor hosting all the Linux systems is omitted from this strategy.
- It is often helpful to have a single console including both z/VM and Linux events.
- Distributed only monitoring and automation events don't fit the existing operations environment of many mainframe customers and their operations staff.
- A 3270 console that mimics the traditional z/OS consoles provide a level of event management familiar to many operations staff.

This paper will introduce IBM Operations Manager for z/VM and demonstrate some of the ways this management tool can be used to provide event notification and automation in the structure and framework familiar to many z/OS system programmers and operations staff.

It is assumed that the reader has a working knowledge of Operations Manager for z/VM. Not all functions and features of Operations Manager for z/VM will be reviewed in detail. For more information on Operations Manager for z/VM, go to:
https://www.ibm.com/products/operations-manager-for-zvm

System Console

Many z/OS customers provide a centralized management console in their operations center. This is often the system console enhanced with products like IBM Tivoli NetView for z/OS to highlight messages, automate actions associated with known messages, and suppress messages. Highlighted and held messages are designed to grab the operator's attention. Suppressed messages are designed to take away the distraction of messages determined to be non-essential to operational support. Some operations staff is accustomed to this type of message monitoring and quickly adapt to the look and feel. While the z/VM OPERATOR user ID is similar to a systems console, it may not be appropriate to suppress messages on OPERATOR. Additionally, providing direct access to the OPERATOR console for those other than system support is often not desired. Creating a custom console for the operations staff with appropriate authorization, message attributes, and automation often provides the perfect console for operations staff in a manner that they find familiar.

Note: Throughout this document the following terms will be used to refer to a virtual machine running under z/VM:

- *Virtual Machine*
- *Guest*
- *User ID*

Collecting Console Messages across z/VM and Linux

Operations Manager for z/VM is defined as the secondary user of a z/VM virtual machine allowing collection of its console messages. Virtual machines of z/VM include the CMS guests (such as TCPIP, DIRMAINT, etc), as well as Linux, z/VSE, or z/OS guest consoles (if these operating systems are running as virtual machines under z/VM.)

With IBM Operations Manager for z/VM you can:

- Monitor and interact with a virtual machine console as if logged onto the virtual machine account.
- Define rules and monitors that specify what problems or types of issues to look for among your z/VM and Linux on IBM Z consoles.
- Define actions that specify what should be done in response to a specific situation that requires intervention.
- Monitor and manage spool usage and spool files.
- Create schedules that will execute a defined action at a specific time.
- Setup automatic sessions that perform specific tasks.

We will make use of these features and functions of Operations Manager for z/VM to create an event console for z/VM and Linux that is similar to what is often found in z/OS Operation Centers.

Create a CMS guest to be the z/VM and Linux Event Console:

As mentioned above, it is often not desirable to provide direct access to the OPERATOR user ID on z/VM for regular console monitoring, as it may have more authorization than desired for the operations staff. Additionally, it will not necessarily have the specific messages associated with guest consoles. For our event console, a new CMS user ID will be created to be the console on which all desired messages will be routed and viewed. The console for the z/VM and Linux messages will be a standard z/VM CMS guest user ID. This CMS user ID will only get the permissions appropriate for the operations staff (in our example privilege class G). The user ID will be named OPER8. There is nothing significant about the name. It can be any name desired that isn't already used on the z/VM system and meets the z/VM naming conventions. The CP directory entry for OPER8 on our system is shown below in Figure 1.

```
USER OPER8 ***** 32M 64M G
   ACCOUNT SYSTEM SYSPROG
   IPL CMS
   MACHINE ESA
   CONSOLE 0009 3215 T OPMGRM1
   SPOOL 000C 2540 READER A
   SPOOL 000D 2540 PUNCH A
   SPOOL 000E 1403 A
   LINK MAINT 0190 0190 RR
   LINK MAINT 019D 019D RR
   MDISK 0191 3390 553 10 DMZU00 MR
```
Figure 1

Operations Manager for z/VM will receive all console messages from all z/VM guests (CMS, Linux, etc) that define Operations Manager as the secondary user. Rules can then be written to invoke an action for a subset of those messages that are deemed critical. The rules will filter for only those messages desired for forwarding to alert the operations staff. The action associated with these messages will be to send the message to OPER8 and apply attributes like highlighting and/or holding the message, re-wording the message, etc. Notice in Figure 1 that the console of OPER8 specifies OPMGRM1 as a secondary user. OPMGRM1 is the user ID of the Operations Manager main service machine. This allows Operations Manager to monitor the console of OPER8, which is how attribute processing will be applied to the console messages. Operations staff can then use Operations Manager for viewing the console of OPER8.

Routing messages to OPER8

Using the Operations Manager for z/VM configuration, rules can be defined to look for critical messages to be forwarded to OPER8 (filter stage), and have attributes applied to them for viewing by operations staff (attribute stage). The following example

- Filters for a specific console message for forwarding to OPER8
- Highlights and holds the message on the OPER8 console

FILTER STAGE:
The first stage of processing is to determine if the console message received is one appropriate for forwarding to OPER8. This will be referred to as the “filter stage”. Once a message meets the filter criteria via an Operations Manager rule, an action will be defined to send the message to OPER8. The message can be sent in its original or modified form. For example, the following rule named ABEND, selects all messages that have the word “abend” and do not have the word “remote” in the text.

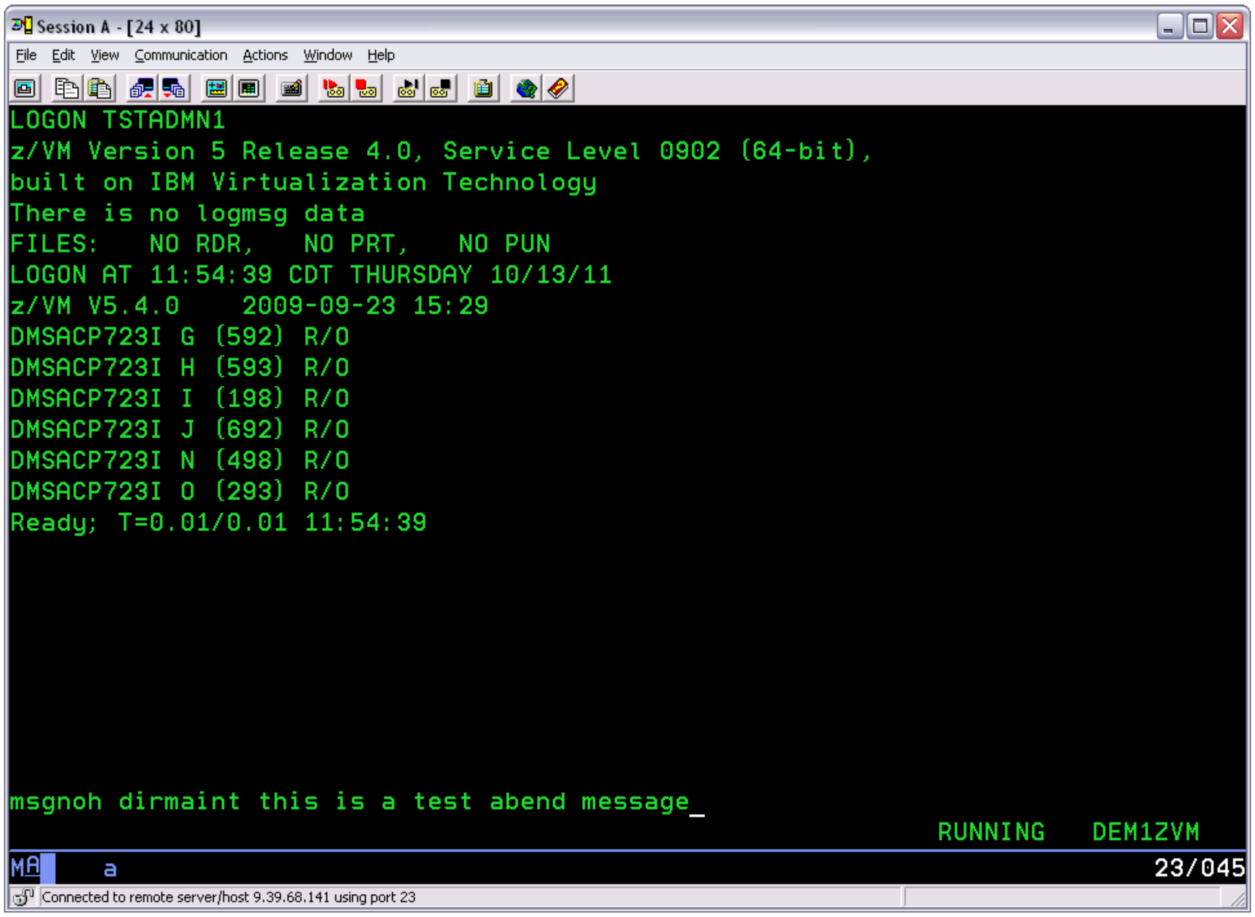
***Note:** The rule must exclude the messages on OPER8 to prevent recursively triggering the rule. Remember OPER8 is also monitored by Operations Manager for z/VM and is thus subject to its rule processing.*

Any message meeting this rule’s criteria will invoke the action named “MSGOPER8”. The action MSGOPER8 issues the CP MSGNOH command to OPER8. Included in the MSGNOH command are the values of the Operations Manager &U and &T substitution variables. The &U variable passes the user ID upon which the original message appeared. The &T variable passes the message text that matched the rule (i.e. the message that met the filter criteria). This filter stage is the first phase of processing this message text. If the message is to be modified, the &T variable will be replaced or amended with the new text.

```
*
DEFRULE NAME ABEND +
  MATCH '*abend*' +
  EXCLUDE '*remote*' +
  EXUSER OPER8 +
  ACTION MSGOPER8
*
```

```
DEFACTN NAME MSGOPER8 +
  COMMAND 'CP MSGNOH OPER8 &U : &T' +
  ENV LVM
*
```

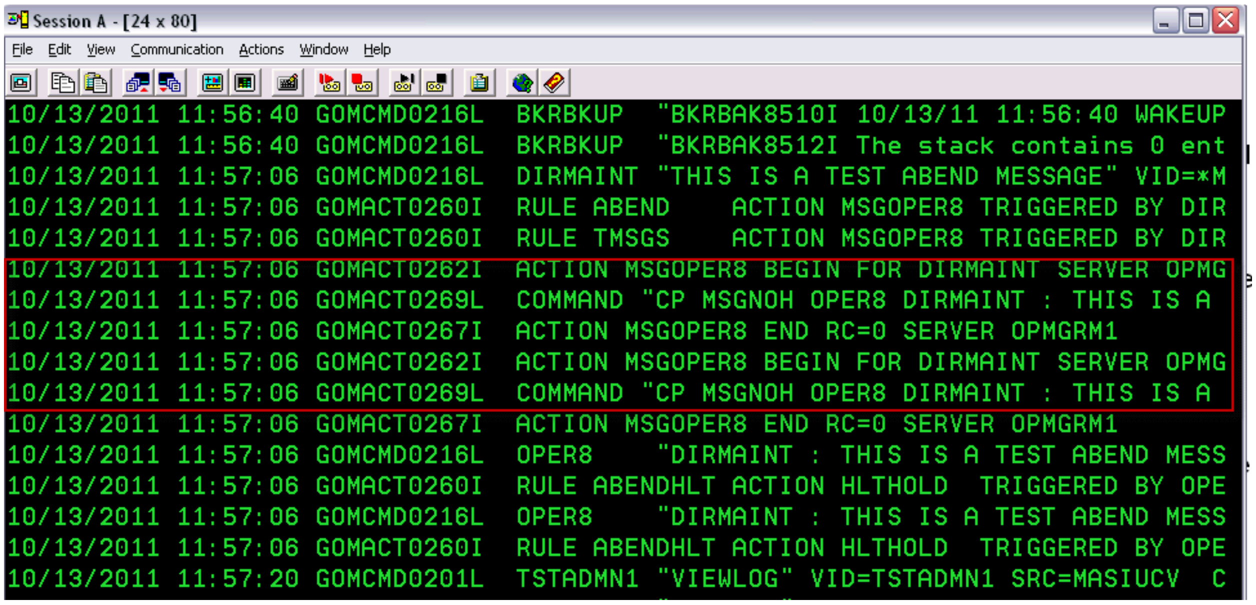
The affects of the filter stage are shown in Figures 2 and 3 below. Figure 2 shows a test abend message, “this is a test abend message”, being sent to the DIRMAINT console which is monitored by Operations Manager for z/VM.



```
LOGON TSTADMN1
z/VM Version 5 Release 4.0, Service Level 0902 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 11:54:39 CDT THURSDAY 10/13/11
z/VM V5.4.0    2009-09-23 15:29
DMSACP723I G (592) R/O
DMSACP723I H (593) R/O
DMSACP723I I (198) R/O
DMSACP723I J (692) R/O
DMSACP723I N (498) R/O
DMSACP723I O (293) R/O
Ready; T=0.01/0.01 11:54:39









msgnoh dirmaint this is a test abend message_
                                                            RUNNING   DEM1ZVM
MA    a                                                              23/045
Connected to remote server/host 9.39.68.141 using port 23
```

Figure 2

Figure 3 is an Operations Manager VIEWLOG display showing action MSGOPER8 being invoked and successfully forwarding the message (prepended with the DIRMAINT user ID) to console OPER8. See the area inside the red outline box.



```
10/13/2011 11:56:40 GOMCMD0216L  BKRBKUP  "BKRBAK8510I 10/13/11 11:56:40 WAKEUP
10/13/2011 11:56:40 GOMCMD0216L  BKRBKUP  "BKRBAK8512I The stack contains 0 ent
10/13/2011 11:57:06 GOMCMD0216L  DIRMAINT "THIS IS A TEST ABEND MESSAGE" VID=*M
10/13/2011 11:57:06 GOMACT0260I  RULE ABEND    ACTION MSGOPER8 TRIGGERED BY DIR
10/13/2011 11:57:06 GOMACT0260I  RULE TMSGS    ACTION MSGOPER8 TRIGGERED BY DIR
10/13/2011 11:57:06 GOMACT0262I  ACTION MSGOPER8 BEGIN FOR DIRMAINT SERVER OPMG
10/13/2011 11:57:06 GOMACT0269L  COMMAND "CP MSGNOH OPER8 DIRMAINT : THIS IS A
10/13/2011 11:57:06 GOMACT0267I  ACTION MSGOPER8 END RC=0 SERVER OPMGRM1
10/13/2011 11:57:06 GOMACT0262I  ACTION MSGOPER8 BEGIN FOR DIRMAINT SERVER OPMG
10/13/2011 11:57:06 GOMACT0269L  COMMAND "CP MSGNOH OPER8 DIRMAINT : THIS IS A
10/13/2011 11:57:06 GOMACT0267I  ACTION MSGOPER8 END RC=0 SERVER OPMGRM1
10/13/2011 11:57:06 GOMCMD0216L  OPER8    "DIRMAINT : THIS IS A TEST ABEND MESS
10/13/2011 11:57:06 GOMACT0260I  RULE ABENDHLT ACTION HLTHOLD  TRIGGERED BY OPE
10/13/2011 11:57:06 GOMCMD0216L  OPER8    "DIRMAINT : THIS IS A TEST ABEND MESS
10/13/2011 11:57:06 GOMACT0260I  RULE ABENDHLT ACTION HLTHOLD  TRIGGERED BY OPE
10/13/2011 11:57:20 GOMCMD0201L  TSTADMN1 "VIEWLOG" VID=TSTADMN1 SRC=MASIUCV  C
```

Figure 3

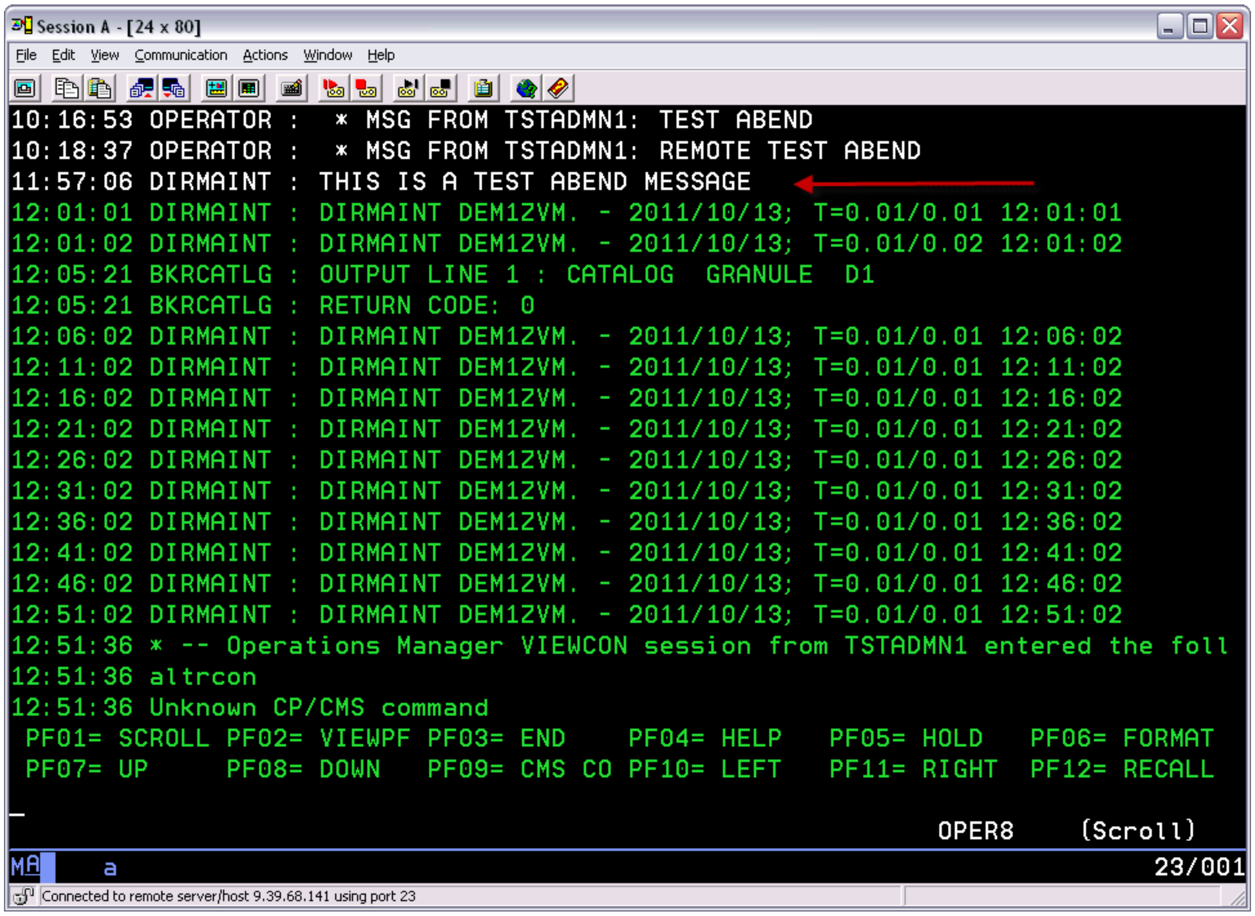
While this filter example is simplistic to make it easier to demonstrate in this document, it is important to note advanced wildcard and character position comparisons are available in Operations Manager. See the Operations Manager for z/VM Administration Guide for complete details about the DEFRULE statement.

ATTRIBUTE STAGE:
The second phase of processing is to apply input actions to the messages forwarded to OPER8. This is the attribute stage. These input actions will help to draw attention to the operations staff indicating the severity of the alert. This attribute stage is possible because OPER8 is a monitored console of Operations Manager for z/VM. The attributes are only visible when viewing the OPER8 console using the Operations Manager VIEWCON command.

In the example below, the rule ABENDHLT monitors the console of OPER8 for messages with *abend* and invokes the action HLTHOLD. Action HLTHOLD applies input actions AHI and HLD to highlight and hold the message. Specifically AHI sets the display attribute to high intensity and HLD holds the message so it doesn't automatically scroll off the console view on the OPER8 console until it's specifically released. Other messages will continue to scroll off the screen The USER keyword of DEFRULE limits these attributes to apply only to the OPER8 console.

```
DEFRULE NAME ABENDHLT +
  MATCH '*abend*' +
  USER OPER8 +
  ACTION HLTHOLD
*
DEFACTN NAME HLTHOLD +
  INPUT 'AHI,HLD'
*
```

The affects of the attribute stage are shown in Figure 4 below. Notice the message is intensified with the white highlight, and is held although additional messages continue to roll off the screen. Operations staff can remove the message by placing the cursor on the message and enter the ALTRCON command (which is assigned to PF5 by default unless otherwise customized in VIEWCON PROFILE).

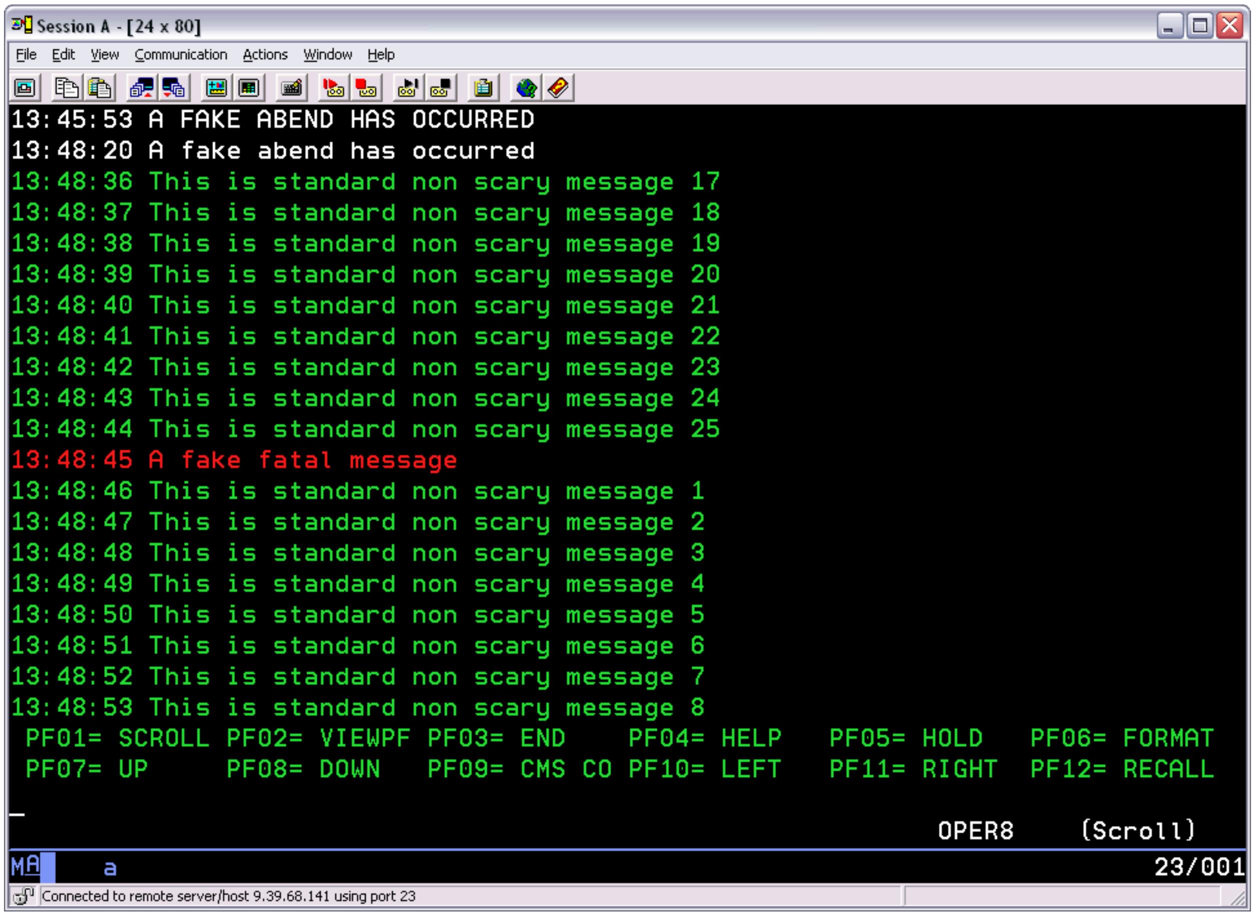


```
10:16:53 OPERATOR :  * MSG FROM TSTADMN1: TEST ABEND
10:18:37 OPERATOR :  * MSG FROM TSTADMN1: REMOTE TEST ABEND
11:57:06 DIRMAINT : THIS IS A TEST ABEND MESSAGE
12:01:01 DIRMAINT : DIRMAINT DEM1ZVM. - 2011/10/13; T=0.01/0.01 12:01:01
12:01:02 DIRMAINT : DIRMAINT DEM1ZVM. - 2011/10/13; T=0.01/0.02 12:01:02
12:05:21 BKRCATLG : OUTPUT LINE 1 : CATALOG  GRANULE  D1
12:05:21 BKRCATLG : RETURN CODE:  0
12:06:02 DIRMAINT : DIRMAINT DEM1ZVM. - 2011/10/13; T=0.01/0.01 12:06:02
12:11:02 DIRMAINT : DIRMAINT DEM1ZVM. - 2011/10/13; T=0.01/0.01 12:11:02
12:16:02 DIRMAINT : DIRMAINT DEM1ZVM. - 2011/10/13; T=0.01/0.01 12:16:02
12:21:02 DIRMAINT : DIRMAINT DEM1ZVM. - 2011/10/13; T=0.01/0.01 12:21:02
12:26:02 DIRMAINT : DIRMAINT DEM1ZVM. - 2011/10/13; T=0.01/0.01 12:26:02
12:31:02 DIRMAINT : DIRMAINT DEM1ZVM. - 2011/10/13; T=0.01/0.01 12:31:02
12:36:02 DIRMAINT : DIRMAINT DEM1ZVM. - 2011/10/13; T=0.01/0.01 12:36:02
12:41:02 DIRMAINT : DIRMAINT DEM1ZVM. - 2011/10/13; T=0.01/0.01 12:41:02
12:46:02 DIRMAINT : DIRMAINT DEM1ZVM. - 2011/10/13; T=0.01/0.01 12:46:02
12:51:02 DIRMAINT : DIRMAINT DEM1ZVM. - 2011/10/13; T=0.01/0.01 12:51:02
12:51:36 * -- Operations Manager VIEWCON session from TSTADMN1 entered the foll
12:51:36 altrcon
12:51:36 Unknown CP/CMS command
 PF01= SCROLL PF02= VIEWPF PF03= END    PF04= HELP   PF05= HOLD   PF06= FORMAT
 PF07= UP     PF08= DOWN   PF09= CMS CO PF10= LEFT   PF11= RIGHT  PF12= RECALL
_
                                                     OPER8     (Scroll)
MA     a                                                                23/001
Connected to remote server/host 9.39.68.141 using port 23
```

Figure 4

See the Operations Manager for z/VM Administration Guide for a detailed list of input actions available on the DEFACTN statement. Some well-known input actions are: audible alarm, blinking, high intensity, reverse video, underline, colors (e.g.: blue, cyan, green, pink, red, white, and yellow), hold, and suppress. See Figure 5 for an example of several console messages with different input actions.

Note: Throughout this document, screen captures of OPER8's console will include "dummy" messages to simplify examples. To show a busy screen in these examples, these additional messages have been generated to show differentiation of highlighting and simulate console flow.



```
13:45:53 A FAKE ABEND HAS OCCURRED
13:48:20 A fake abend has occurred
13:48:36 This is standard non scary message 17
13:48:37 This is standard non scary message 18
13:48:38 This is standard non scary message 19
13:48:39 This is standard non scary message 20
13:48:40 This is standard non scary message 21
13:48:41 This is standard non scary message 22
13:48:42 This is standard non scary message 23
13:48:43 This is standard non scary message 24
13:48:44 This is standard non scary message 25
13:48:45 A fake fatal message
13:48:46 This is standard non scary message 1
13:48:47 This is standard non scary message 2
13:48:48 This is standard non scary message 3
13:48:49 This is standard non scary message 4
13:48:50 This is standard non scary message 5
13:48:51 This is standard non scary message 6
13:48:52 This is standard non scary message 7
13:48:53 This is standard non scary message 8
 PF01= SCROLL PF02= VIEWPF PF03= END     PF04= HELP    PF05= HOLD    PF06= FORMAT
 PF07= UP      PF08= DOWN   PF09= CMS CO PF10= LEFT    PF11= RIGHT   PF12= RECALL
_
                                                             OPER8     (Scroll)
MA     a                                                                 23/001
Connected to remote server/host 9.39.68.141 using port 23
```

Figure 5

Note: It is important to understand that all messages collected by Operations Manager are saved in a log file. Any customization of messages and suppression of messages on a specific console (such as OPER8) are not applied to log file.

Manual Responses to Console Messages

- The operations staff can respond to the console messages on OPER8 based on OPER8's privilege class authorization when appropriate.
- Operations Manager for z/VM can be used by authorized users to access virtual machine consoles allowing the operations staff to respond to OPER8 messages based on their Operations Manager for z/VM authorizations. This applies to all monitored consoles including Linux on IBM Z.

Automated Responses to Console Messages

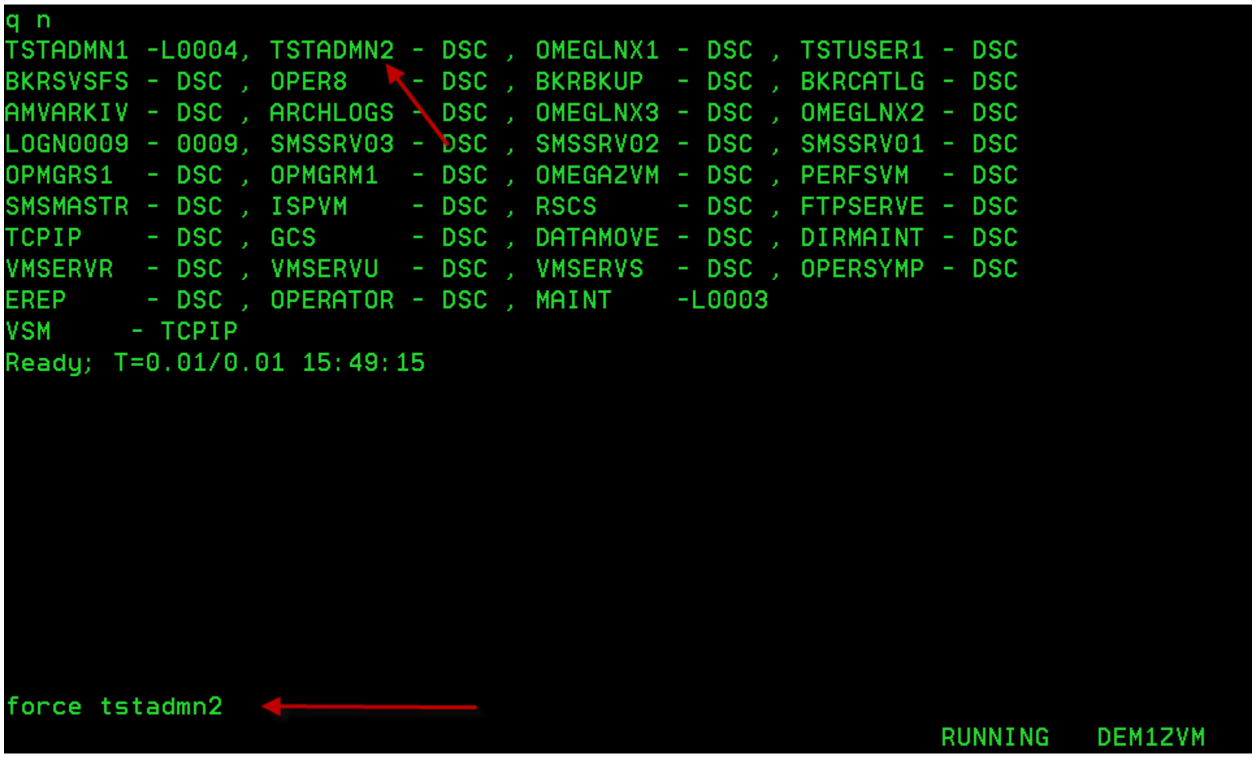
The same DEFRULE and DEFACTN statements used to filter messages and apply input actions also provide parameters to allow for automated responses triggered by console messages. The standard benefits of automation, including immediate response, removal of human error, standardization of responses, etc., are realized for the z/VM system as well as the guest virtual machines whose consoles are being monitored. This includes the Linux guest consoles as well.

Centralizing automation at the z/VM hypervisor level greatly enhances standardization and reduces duplication of automation efforts across virtual machines.

Automation Example:

For an example of automation, Operations Manager for z/VM will be used to manage the availability of a critical service virtual machine. This same capability may be used for any supported virtual machine including a Linux guest. In this scenario, the CMS virtual machine, TSTADMN2, should be running disconnected on the z/VM system at all times. If this machine were to be logged off by accident, or by a failure, Operations Manager for z/VM will XAUTOLOG this virtual machine to make it operational and available immediately.

From the MAINT ID on this z/VM system, the QUERY NAMES command is issued to show that TSTADMN2 is running in DSC mode. See Figure 6 below (top arrow).



```
q n
TSTADMN1 -L0004, TSTADMN2 - DSC , OMEGLNX1 - DSC , TSTUSER1 - DSC
BKRSVSFS - DSC , OPER8    - DSC , BKRBKUP  - DSC , BKRCATLG - DSC
AMVARKIV - DSC , ARCHLOGS - DSC , OMEGLNX3 - DSC , OMEGLNX2 - DSC
LOGN0009 - 0009, SMSSRV03 - DSC , SMSSRV02 - DSC , SMSSRV01 - DSC
OPMGRS1  - DSC , OPMGRM1  - DSC , OMEGAZVM - DSC , PERFSVM  - DSC
SMSMASTR - DSC , ISPVM    - DSC , RSCS     - DSC , FTPSERVE - DSC
TCPIP    - DSC , GCS      - DSC , DATAMOVE - DSC , DIRMAINT - DSC
VMSERVR  - DSC , VMSERVU  - DSC , VMSERVS  - DSC , OPERSYMP - DSC
EREP     - DSC , OPERATOR - DSC , MAINT    -L0003
VSM     - TCPIP
Ready; T=0.01/0.01 15:49:15









force tstadmn2
                                                          RUNNING   DEM1ZVM
```

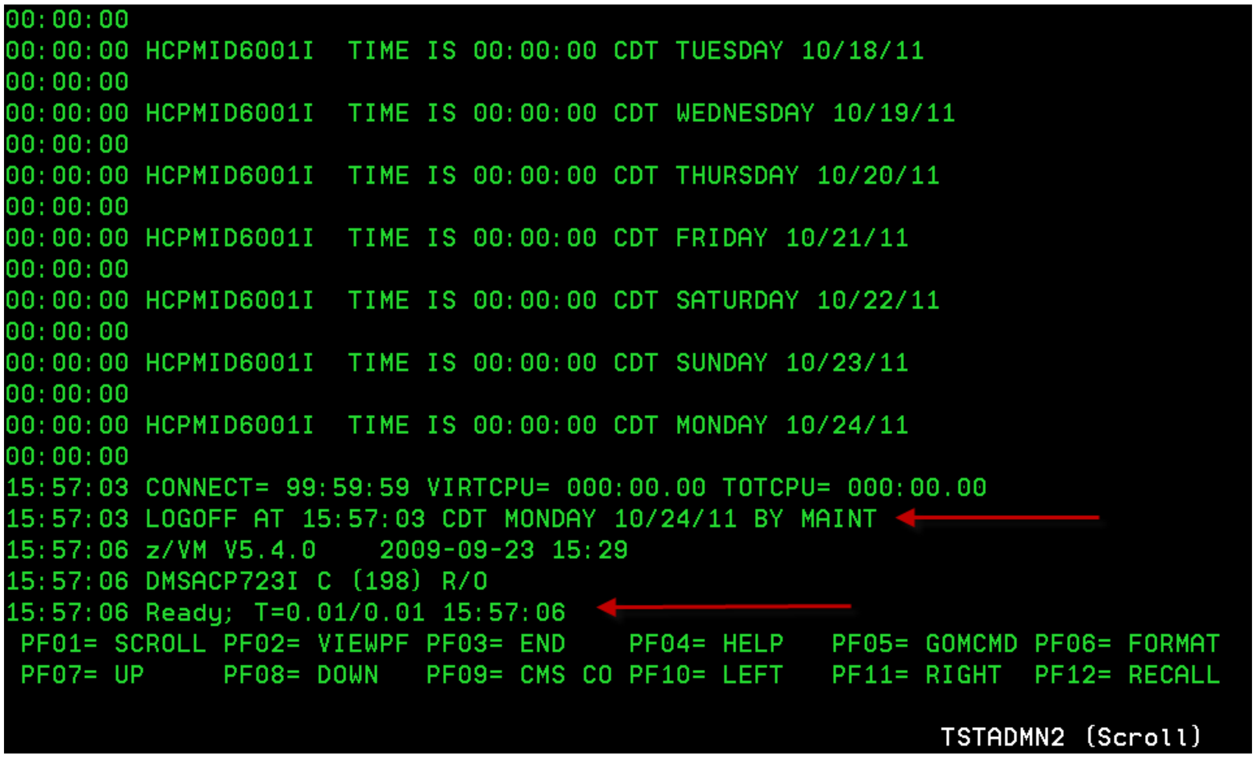
Figure 6

Also shown above in Figure 6, the FORCE TSTADMN2 command being issued to logoff user ID TSTADMN2 (bottom arrow).

Using a user ID authorized to Operations Manager for z/VM, a view of the TSTADMN2 console will show the results of the FORCE TSTADMN2 command as well as the automation in Operations Manager to XAUTOLOG TSTADMN2, bringing it back up immediately. The Operations Manager for z/VM view of the TSTADMN2 console (Figure 7) shows:

- The results of the FORCE TSTADMN command issued by MAINT causing TSTADMN2 to logoff. This is indicated by the top red arrow.

- TSTADMN2 is immediately XAUTLOGged and already active with the READY prompt displayed. This is indicated by the bottom red arrow.



```
00:00:00
00:00:00 HCPMID6001I  TIME IS 00:00:00 CDT TUESDAY 10/18/11
00:00:00
00:00:00 HCPMID6001I  TIME IS 00:00:00 CDT WEDNESDAY 10/19/11
00:00:00
00:00:00 HCPMID6001I  TIME IS 00:00:00 CDT THURSDAY 10/20/11
00:00:00
00:00:00 HCPMID6001I  TIME IS 00:00:00 CDT FRIDAY 10/21/11
00:00:00
00:00:00 HCPMID6001I  TIME IS 00:00:00 CDT SATURDAY 10/22/11
00:00:00
00:00:00 HCPMID6001I  TIME IS 00:00:00 CDT SUNDAY 10/23/11
00:00:00
00:00:00 HCPMID6001I  TIME IS 00:00:00 CDT MONDAY 10/24/11
00:00:00
15:57:03 CONNECT= 99:59:59 VIRTCPU= 000:00.00 TOTCPU= 000:00.00
15:57:03 LOGOFF AT 15:57:03 CDT MONDAY 10/24/11 BY MAINT
15:57:06 z/VM V5.4.0    2009-09-23 15:29
15:57:06 DMSACP723I C (198) R/O
15:57:06 Ready; T=0.01/0.01 15:57:06
 PF01= SCROLL PF02= VIEWPF PF03= END    PF04= HELP   PF05= GOMCMD PF06= FORMAT
 PF07= UP     PF08= DOWN   PF09= CMS CO PF10= LEFT   PF11= RIGHT  PF12= RECALL

                                                           TSTADMN2 (Scroll)
```

Figure 7

The Operations Manager for z/VM configuration statements that make this possible are shown below:

```
DEFEMON NAME ADMIN2 +
  TYPE 1 +
  USER TSTADMN2 +
  ACTION AUTOLOG1
*
DEFACTN NAME AUTOLOG1 +
  DELAY 00:03
  COMMAND 'CP XAUTOLOG &3' +
  ENV SVM
```

The DEFEMON command defines an event monitor. Events from the CP system service *VMEVENT are captured and compared against defined event monitors. The TYPE keyword specifies the *VMEVENT Type. TYPE 1 indicates the LOGOFF event. The DEFEMON above comes true when user ID TSTADMN2 enters the logged off state. I.e. the user ID must change from another state into the logged off state, not already be in the logged off state. The action AUTOLOG1 is then invoked by Operations Manager. Action AUTOLOG1 waits for 3 seconds before issuing the actual XAUTOLOG command. Without this DELAY, the XAUTOLOG will almost always fail, indicating that the user is in LOGOFF PENDING status. By waiting 2 or 3 seconds, we allow the user ID to complete the logoff processing before autologging it.

You'll also notice that the user ID that triggered this event (TSTADMN2 in this example) is passed to the action routine using the &3 substitution variable. This allows you to use this same action for autologging any user ID that triggers an event.

Note: ENV SVM is used on the DEFACTN statement, indicating the XAUTOLOG will be done by an action processing server. This means the Operations Manager action processing servers must have authority to issue the XAUTOLOG command. By default, CP privilege class A or B provides this authority.

Using Operations Manager for z/VM, automated responses can be issued on service machines and on the originating console of the event. Therefore, automation can be performed on all virtual machines including Linux IBM Z. This centralized automation more easily allows combined z/VM and Linux automation steps, runs event management on the z/VM hypervisor versus duplication across all Linux guests, and reduces the amount of support staff needed to maintain the automation features.

Forwarding z/VM and Linux Console Messages to an Enterprise Event manager

Enterprise level event monitoring is an important piece of the overall systems management strategy of most advanced enterprises. Many events should be and will be handled by operations staff in the traditional realm of 3270 event monitoring. This document shows how z/VM and Linux systems can be included in this realm of management. This does not negate the need to include select events at an enterprise level. While this document is focused on providing a 3270 based operations console designed to mimic the typical z/OS console with automation enhancements, it is important to recognize the fact that many organizations use their z/OS automation to forward select events to an enterprise event manager like IBM Netcool/OMNIbus. It is important to understand that Operations Manager for z/VM's automation may also be used to forward select events to an enterprise event manager. It is outside the scope of this document to address integration to any specific event manager. The URL below links to another document that specifically addresses integration between Operations Manager for z/VM and IBM Netcool/OMNIbus. For those using other enterprise event managers, the document below may be used as an example and should prove helpful in integrating Operations Manager for z/VM with other vendor or in-house enterprise event solutions.

http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101492

Summary

One of the functions in IBM Operations Manager for z/VM is to receive console messages from virtual machines or guests under z/VM. Using the configuration commands provided with Operations Manager for z/VM, it is possible to set up an operations console similar to those often used by operations staff for z/OS. Operations Manager for z/VM provides message attributes that draw operator attention to the console messages. Creating an event console for z/VM and

Linux provides the operations staff, which may be new to z/VM and Linux, a console for event management that is familiar in style and function to what the staff is already accustomed to for z/OS. Automating actions to known events helps to reduce human error and provide a quicker resolution to events, providing increased up time and system support. This can be very helpful in easing the growing pains associated with the z/VM and Linux implementation.