Routing Linux SYSLOG, UNIX SYSLOG, and application log file data to IBM Operations Manager for z/VM

This document can be found on the web at www.ibm.com/support/techdocs
Search for author's name under the category of "White Papers".

Version Date: October 2018

IBM Advanced Technical Support

Tracy Dean
Offering Manager z/VM Tools
tld1@us.ibm.com

Special Notices

This document reflects the IBM Advanced Technical Skills organization's understanding of the Operations Manager for z/VM configuration statement DEFIPCS. It was produced and reviewed by the members of the IBM Advanced Technical Skills 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 Skills organization. These opinions are based on the authors' experiences. If you have questions about the contents of this document, please contact the author at tld1@us.ibm.com.

Trademarks

The following are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

IBM, the IBM logo, DB2, WebSphere, z/OS, z/VM, AIX.

A full list of U.S. trademarks owned by IBM may be found at http://www.ibm.com/legal/copytrade.shtml.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

Other company, product and service names may be trademarks or service marks of others.

Introduction

IBM Operations Manager for z/VM provides the DEFIPCS statement to establish an IP connection to Operations Manager. Among other things, this configuration statement allows Operations Manager to collect and process syslog data or file input from local or remote systems. Many customers centralize log management to Operations Manager for z/VM to meet log management best practices. Data written to syslog or application log files can be processed by Operations Manager in the same manner as SECUSER or OBSERVER data from a monitored console. This means that rules can be written to automatically take an action if a specified message appears in the log data. In addition, once this data is collected by Operations Manager, it can be viewed using the VIEWCON command, the same way console data is viewed.

This whitepaper will document the steps required to route local and remote syslog data as well as application log file data to Operations Manager and to configure Operations Manager to receive it. Examples using Linux syslogd, syslog-ng, rsyslog, and UNIX syslogd are provided. AIX will be used for the UNIX syslogd example. Application log data is discussed in a separate section of this document and uses the rsyslog technology. In general, these steps include:

- Configuring Operations Manager to collect syslog data
- Updating the TCP/IP server on z/VM
- Configuring the Linux guest or AIX system for log routing (syslogd, syslog-ng, or rsyslog)

These three steps can be done in any order.

Once all the steps are complete, we will discuss how to test it. Additional consideration of defining multiple syslog routes will be discussed as well.

Configuring Operations Manager to collect log data

Specifying the DEFIPCS configuration statement

In your Operations Manager configuration file (COMMON CONFIG, by default), add the following statement:

```
DEFIPCS NAME LNXSYSLG +
  USER TCPIP +
  ADDR 000.000.000.000 +
  PORT 514 +
  APPL GOMRSYL +
  PARM 'LXSYSLOG03330417UTF8'
```

DEFIPCS Operands:

NAME LNXSYSLG: specifies a unique name for this IP connection definition in Operations Manager. It can be any 8 character name you choose.

USER TCPIP: specifies the name of the z/VM userid providing TCP/IP communications. The userid is TCPIP on this system, which is quite common in z/VM environments.

ADDR 000.000.000.000: Specifying all zeros allows Operations Manager to listen on all available IP addresses for USER, including localhost. If Operations Manager should listen to a single IP address and USER has multiple IP addresses, specify the specific IP address for this operand value.

PORT 00514: specifies the IP port on which Operations Manager listens for the remote syslog input. Leading zeros are required to make the value 5 digits. This example specifies 00514 which is the Linux default port for routing syslog data.

APPL GOMRSYL: specifies the name of the Operations Manager processing program. GOMRSYL must be specified as this is the processing program for syslog input.

PARM 'LXSYSLOG03330417UTF8': This statement provides parameters to the GOMRSYL program identified in the APPL statement. The parameter string is parsed positionally as follows:

- Position 01-08 (LXSYSLOG): This is the name you want associated with the syslog data arriving on the IP address and port specified on ADDR and PORT. Do not specify the name of a real user ID on your system. Operations Manager will treat this like a user ID for rule processing and console viewing of the syslog data. This is the user value you specify in the VIEWCON command to view the syslog data. If you specify blanks, the internal program name is used.
- Position 09-16 (03330417): Filename of the ASCII to EBCDIC translation table. The filetype is TABLE. This file, 03330417 TABLE, may be located on any disk accessed by OPMGRM1. We recommend you place the file on the OPMGRM1 198 disk. (See instructions below.) Operations Manager provides three sample tables to choose from, including 03330417 used above. If you specify blanks, the internal ASCII to EBCDIC table is used.
- Position 17-20 (UTF8): Character encoding scheme (IS08 or UTF8). If you specify blanks, IS08 is used.

Giving Operations Manager access to translation table(s)

Operations Manager provides three sample translation tables for use with DEFIPCS. They are shipped with a filetype of SAMPTABL on the 2C2 disk of user ID 5697J10x (depending on your release of Operations Manager.) To make these files accessible to Operations Manager, copy them to the OPMGRM1 198 disk:

1. Logon to user ID 5697J10x
2. `acc 2c2 f`
3. `link opmgrm1 198 198 mr`
4. `acc 198 c`
5. `copyfile * samptabl f = table c`
6. Logoff user ID 5697J10x

Reload Operations Manager configuration files and re-access disks

From an authorized user ID, perform one of the following steps so that Operations Manager will re-access the 198 disk (to see the new translation tables) and reload its configuration file (to see the new DEFIPCS statement.) The following statements assume you are using the default name for your configuration file (OPMGRM1 CONFIG) and that it's on the OPMGRM1 198 disk.

1. If you are using the OMRELOAD EXEC (shipped as a sample with Operations Manager), then issue:
   ```
   omreload
   ```

2. If you are not using the OMRELOAD EXEC, then issue:
   ```
   gomcmd opmgrm1 cms cmd 'acc 198 e'
   gomcmd opmgrm1 config file 'opmgrm1 config e' clear all
   ```

Updating the TCP/IP server on z/VM

Authorizing Operations Manager to listen on the IP port

Assuming you have limited the ability for z/VM service machines to listen on IP ports, you must authorize the Operations Manager user ID OPMGRM1 (and any user ID running GOMMAIN) to listen on the port specified in the DEFIPCS statement.

Note: OPMGRM1 is the default name of the Operations Manager VM user ID. If another name is selected for your system, substitute names accordingly.

Add the following line to the file PROFILE TCPIP (on TCPMAINT's 198 disk on the authors' system):

```
514  UDP OPMGRM1             ; OPERATION MANAGER SYSLOG PORT
```

For this port change to take affect, recycle TCPIP. To dynamically activate these changes without restarting the TCPIP server, use the NETSTAT OBEY command. The following example was used on the authors' system:

```
netstat obey port 514 udp opmgrm1 noautolog
```

Note: In most cases, you should complete both steps above. Use NETSTAT OBEY to activate the change without restarting the TCPIP server, and update PROFILE TCPIP so that each time the TCPIP server is restarted, the authorization will be active.

Configuring the Linux Guest or AIX for syslog routing

In addition to configuring Operations Manager to receive syslog messages from a Linux or AIX server, the Linux or AIX server must be configured to route its syslog data to Operations Manager.

There are configuration and functional differences depending on each syslog implementation (AIX syslogd; Linux syslogd, syslog-ng, or rsyslog). An example for each is presented below. It is important to understand that other remote system platforms may also send their data to Operations Manager and/or other syslog daemons may be implemented. This document is simply selecting a subset as examples.

Linux Syslogd Example:

In this syslog example, the Linux syslog data is routed to the remote z/VM system with basic filtering to select the message types being forwarded. Syslogd uses port 514. In the prior examples above, both Operations Manager for z/VM and TCP/IP on z/VM were configured for port 514.

Update */etc/hosts*:

It is necessary to make the z/VM system where Operations Manager is running a known system to the Linux guest and define it as the loghost. To do this, add an entry in the /etc/hosts file for the appropriate Linux guest. Use the following format:

```
IP-address                    fully-qualified-domain-name hostname "loghost"
```

```
"loghost"
```

For example:

```
9.39.68.141                   dem1zvm.demopkg.ibm.com     dem1zvm   loghost
```

In the example above, the z/VM system hosting Operations Manager is DEMLZVM with an IP address of 9.39.68.141. (Substitute your system name and IP address, accordingly.) Figure 1 shows a copy of the Linux Server HASL106's */etc/hosts* file after the entry for DEMLZVM is added see red arrow.

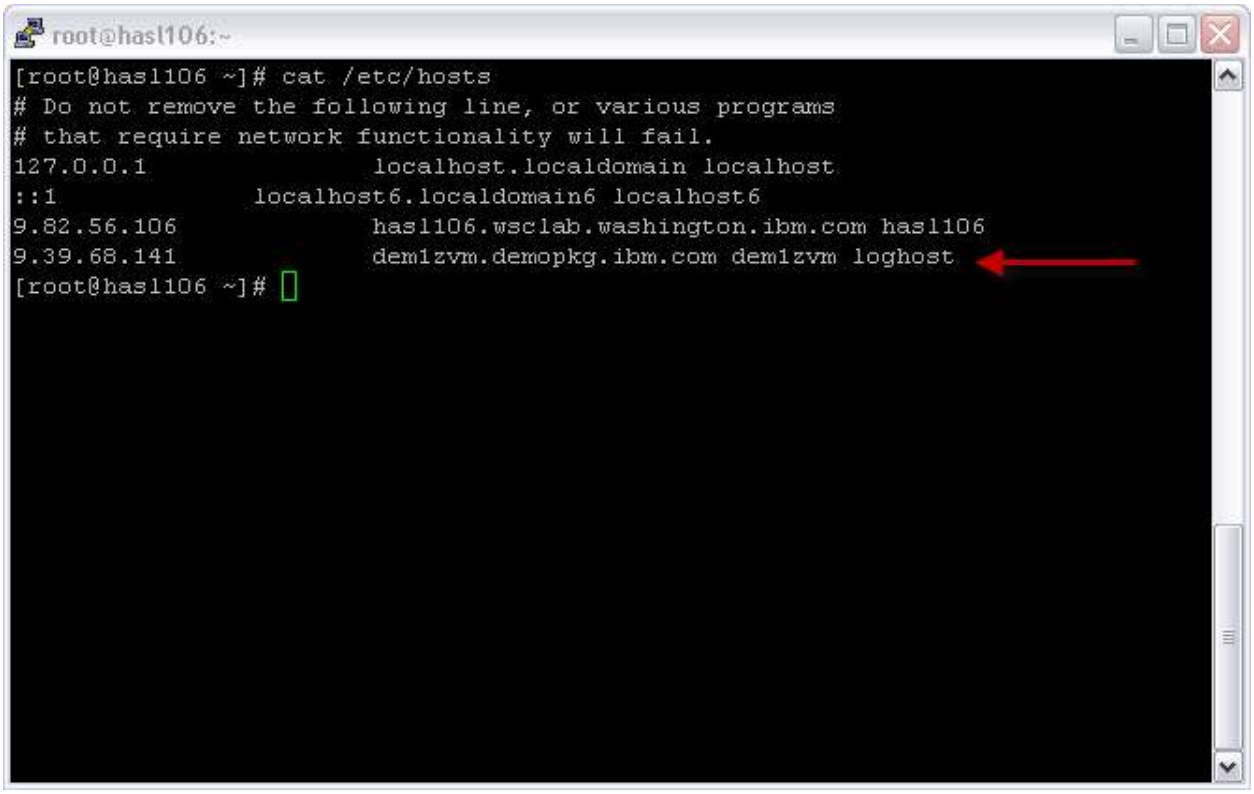

Figure 1

Now HASL106's */etc/hosts* file has the z/VM system DEMLZVM, where Operations Manager is installed, defined as the loghost.

Update */etc/syslog.conf*:

On the Linux guest, edit the */etc/syslog.conf* file to send SYSLOG messages to DEMLZVM as the "loghost". The entry is in the following format:

```
*.*                                             @loghost
```

The above syntax will send all messages to DEMLZVM. To send all informational or higher messages, the following syntax would be specified:

```
*.debug                                         @loghost
```

Figure 2 below shows HASL106's */etc/syslog.conf* file entry to route all SYSLOG messages to DEMLZVM, see the red arrow.

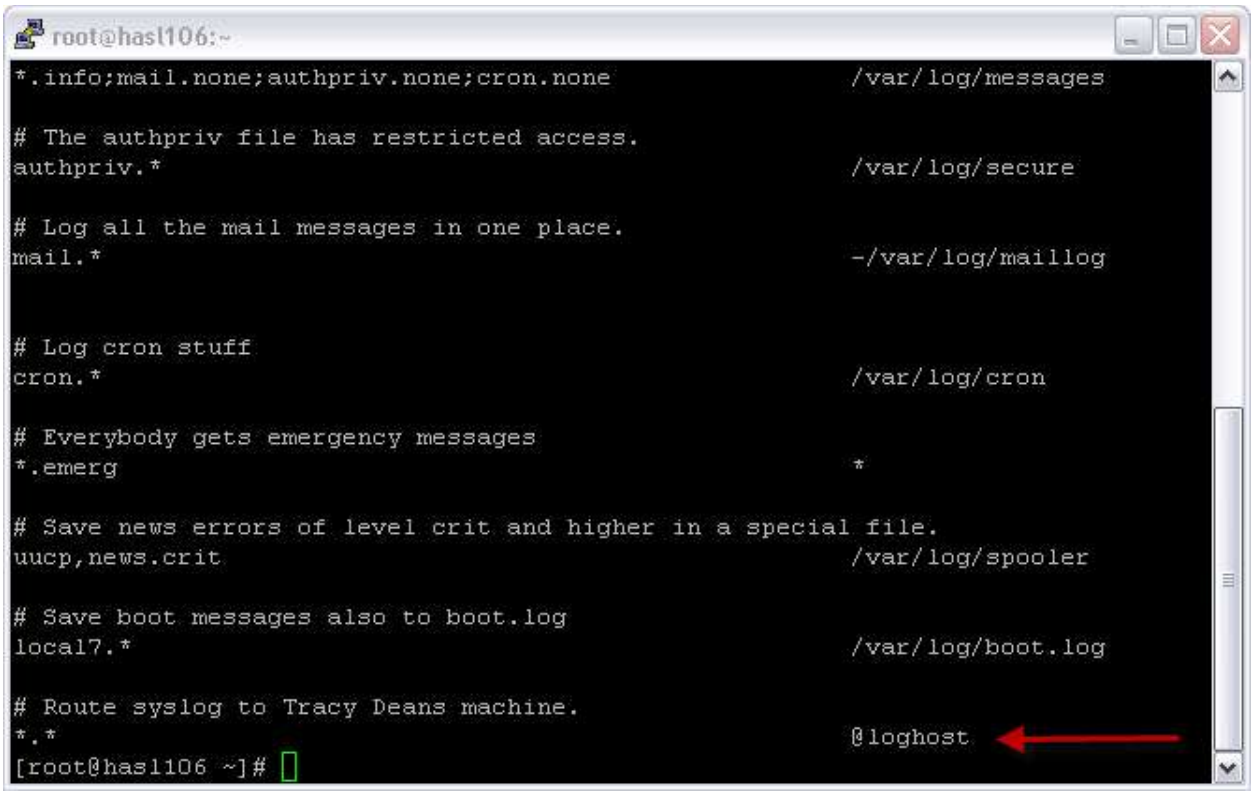

Figure 2

Restart SYSLOG:

Issue: */etc/init.d/syslog* restart

Syslog-ng Example:

The syslog-ng (syslog next generation) is an open source implementation of the syslog protocol. It enhances the traditional syslogd with more advanced filtering options and adds important features to syslogd, like protocol options and port flexibility. The configuration differences warrant a separate set of examples in this paper. Novell SUSE Enterprise Linux distributions often default syslog-ng as the syslog daemon.

Update /etc/syslog-ng/syslog-ng.conf:

The syslog-ng.conf message route is made up of three parts: source, destination, and filtering rules. Including destination in the route removes the need to update /etc/hosts shown in the syslogd example above.

The syntax for the destination statement is as follows:

```
destination <destname> { destdriver params; destdriver params; ... ; };
```

In the example below, the destination name is loghost, the protocol is udp, and the destination parms include the IP address and port of the z/VM system where Operations Manager resides. Please note the port number is not the default 514 used by syslogd. Instead it is port 515 (an available port selected). This is one benefit of using syslog-ng, port flexibility.

With port flexibility,syslog data from different Linux servers or guests can be sent to different ports on z/VM. Operations Manager can be set up with multiple listeners using DEFIPCS, giving them each a different name and port. This means, for example, that you can:

- Send all syslog data from all guests to one port, or
- Send syslog data from a group of guests to one port to be viewed by a specific user/console name, and send syslog data from a different group of guests to another port to be viewed using a different user/console name.

```
destination loghost { udp("9.39.68.141" port(515));};
```

The syntax for the log statement is as follows:

```
log { source S1; source S2; ... filter F1; filter F2; ... destination D1; destination D2; ... };
```

In the example below, the log statement identifies the three parts of the message route.

1) Source: The source parameter below points to the name src defined earlier in the syslog-ng.conf file. The default src provided in syslog-ng is used and includes internal syslog-ng messages as well as /dev/log.
2) Filter: The filter parameter is f_messages which routes messages not including facility(news, mail) and not filter(f_iptables).
3) Destination: The destination parameter is loghost referring to the destination statement named in this example above.

```
log { source(src); filter(f_messages); destination(loghost); };
```

With the destination and log statements and their parameters, we have completed the definitions necessary to route syslog data from a local Linux guest or a remote Linux guest or system to Operations Manager on z/VM.

Figure 3 below shows these two statements (highlighted in yellow) in the syslog-ng.conf file.

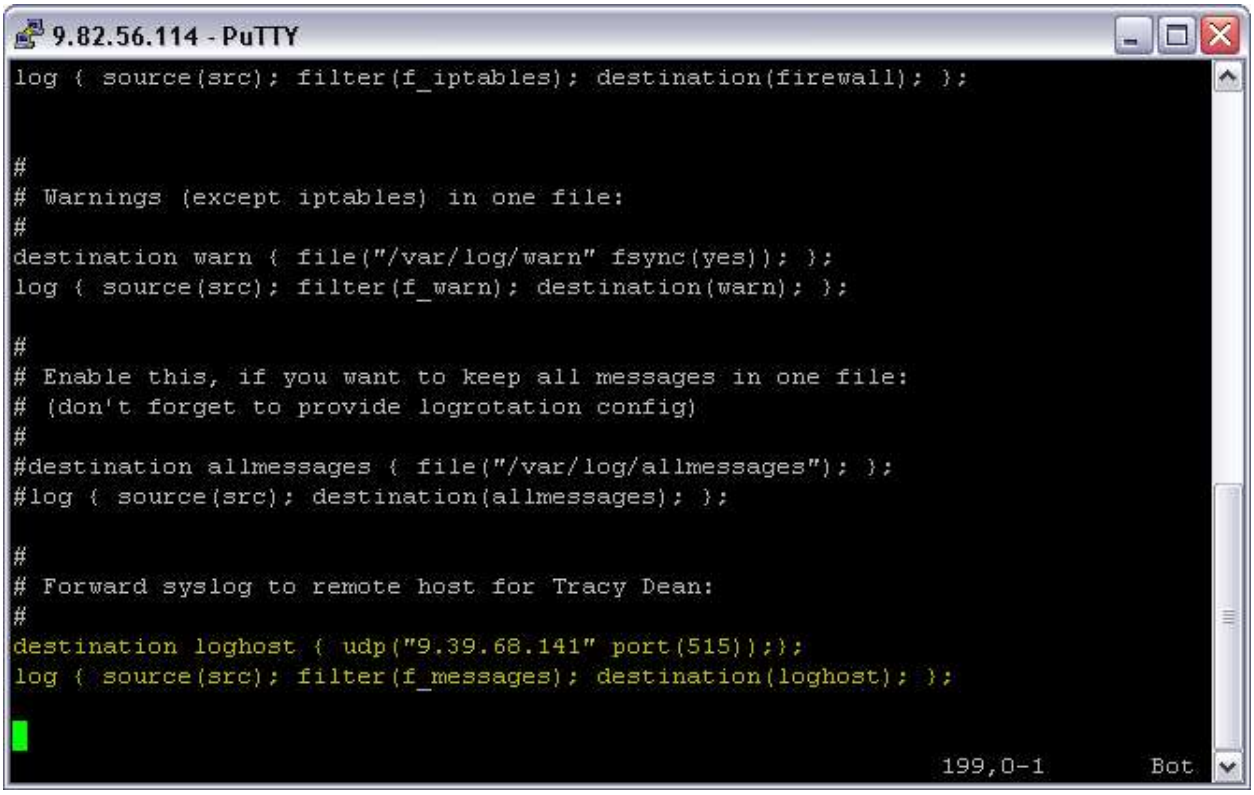

Figure 3

Restart SYSLOG:

Issue: */etc/init.d/syslog* restart

rsyslog Example:

Rsyslog is another open source program for forwarding messages via TCP. Rsyslog was introduced to compete with syslog-ng and address issues the author of rsyslog felt existed in the current syslog daemons, the original syslogd and syslog-ng solutions. The first Enterprise Linux (Linux for System z) to include rsyslog is RedHat 6. Rsyslog implements the basic syslog protocol, extends it with content-based filtering, rich filtering capabilities, flexible configuration options and adds important features such as using TCP for transport. Our demo implementation is still a UDP implementation, since Operations Manager requires UDP. To perform basic syslog forwarding requires minimal configuration with rich features that have become standard since syslog-ng.

Update /etc/rsyslog.conf:

The simplicity of rsyslog is immediately apparent with a basic routing of all messages to a remote loghost. In the example below (Figure 4), one statement routes all syslog messages to DEM1ZVM (9.39.68.141) on port 516 (see red arrow).

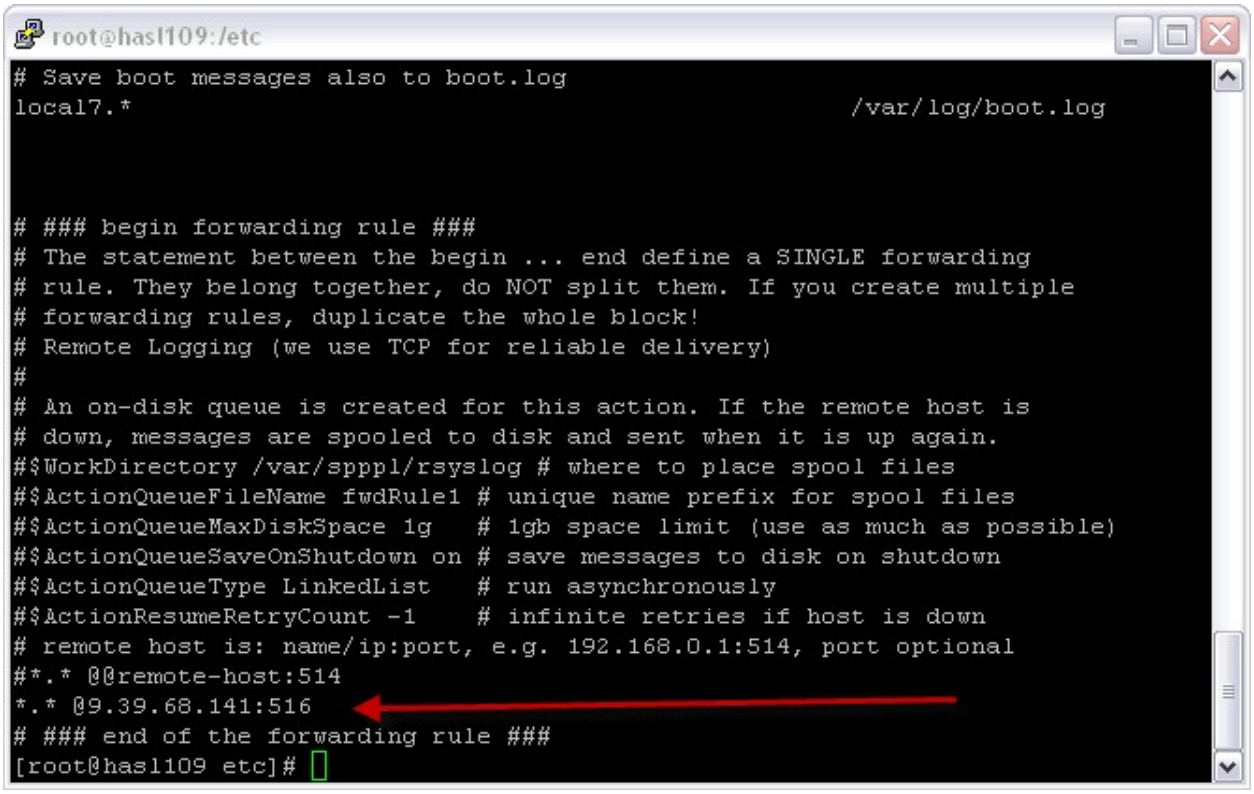

Figure 4

Note: As with syslog-ng, port flexibility is an option with rsyslog providing the same benefits described above in the syslog-ng example.

Restart SYSLOG:

Issue: */etc/init.d/service rsyslog restart*

AIX syslogd Example:

AIX syslogd is similar to the Linux syslogd configuration. However, it is important to pay attention to details. Small errors can result in an invalid valid configuration. Be sure to review the syslogd man pages and other documentation to correctly configure the /etc/syslog.conf file associated with your version of AIX. For example, in the AIX example below, the only valid whitespace in the routing statements is “tab”. Additionally, *.* type statements are invalid.

Update /etc/hosts

Update /etc/hosts and define the DEM1ZVM as the loghost for this AIX system (see Figure 5).

Figure 5

Update /etc/syslog.conf:

Update /etc/syslog.conf file. In the example below (Figure 6), log files for the debug and higher are defined across the syslog message types. Adding an additional entry to route all debug or higher messages to the loghost will route these message types to the loghost defined in the /etc/hosts file.

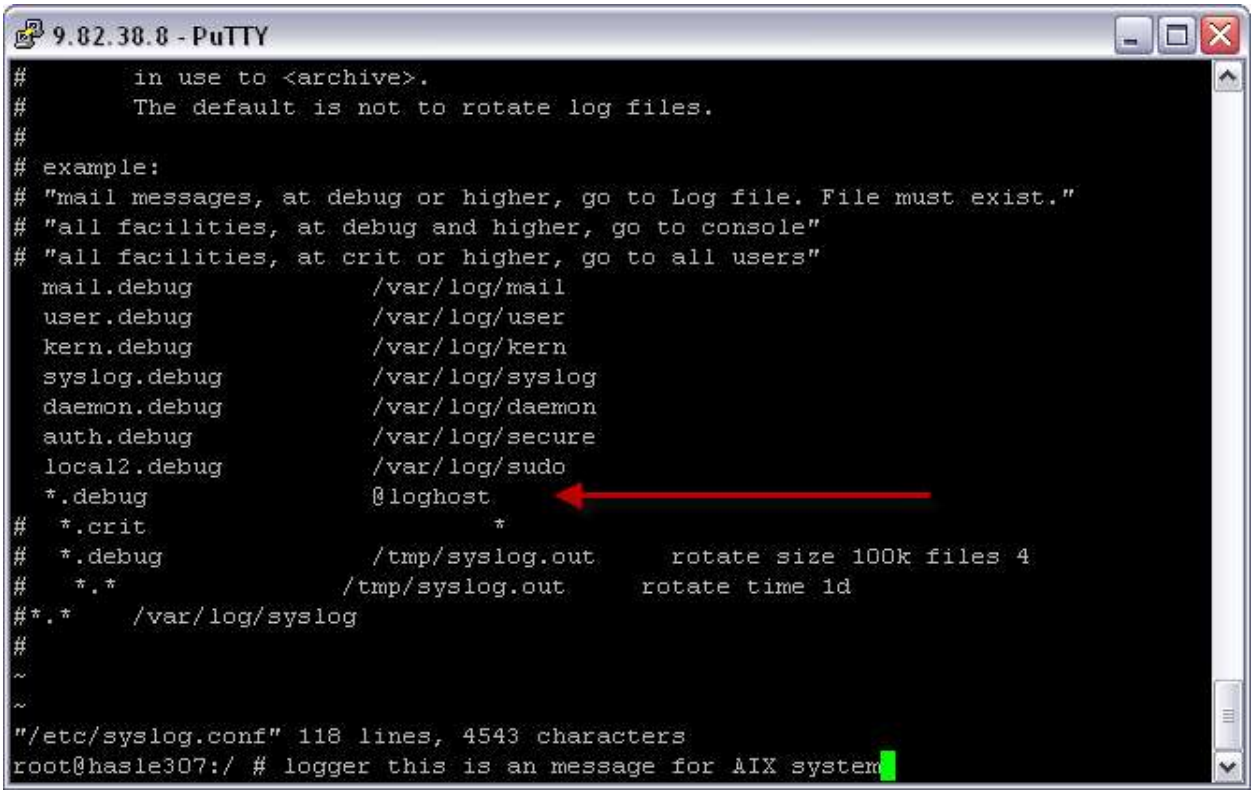

Figure 6

Restart SYSLOG:

Issue: *refresh –s syslogd*

Testing the syslog route to Operations Manager

A simple way to test your configuration changes in Operations Manager, TCP/IP, and Linux is to use the Linux “logger” command.

The logger command makes entries in the system log. It provides a shell command interface to the syslog(3) system log module. The syntax follows:

logger [-**isd**] [-**f** *file*] [-**p** *pri*] [-**t** *tag*] [-**u** *socket*] [*message ...*]

In all the configuration examples above, the message filters defined will match the message level issued by the logger command for each of the systems tested. Therefore, a simple logger message statement will determine if your configuration updates are correct.

Figure 7 shows an example of the logger command being issued:

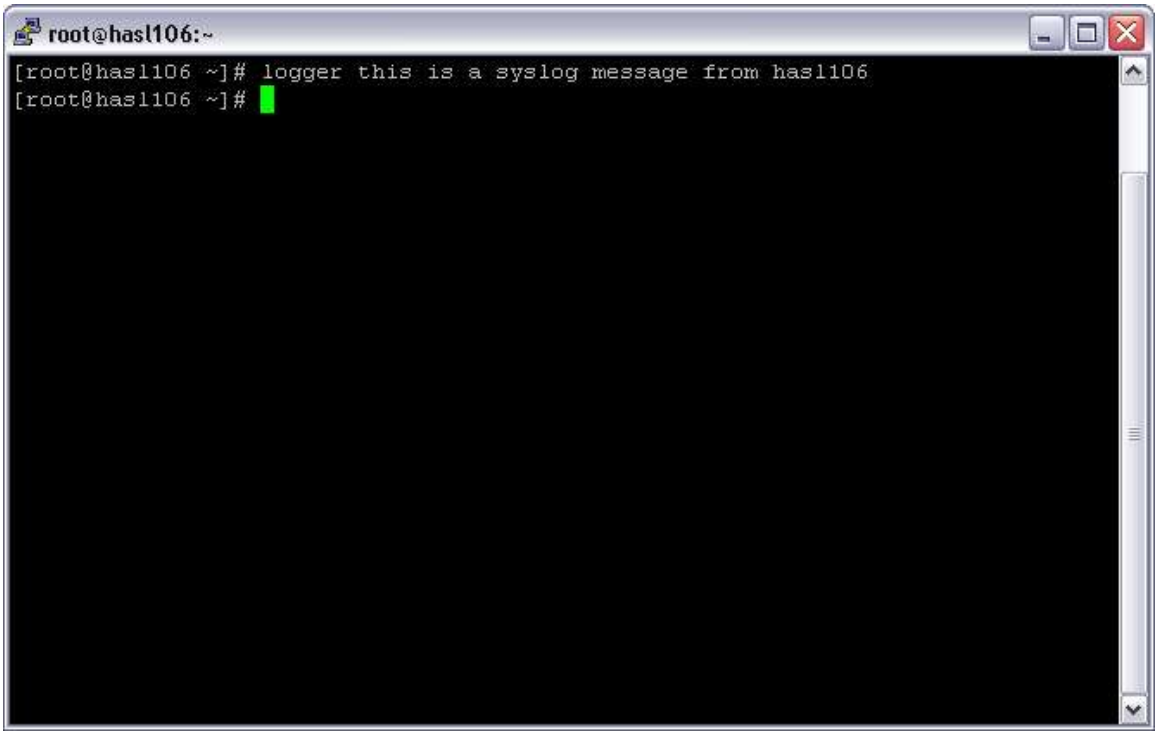

Figure 7

To view the syslog from Operations Manager, use the following command from an authorized user on z/VM:This assumes you are using the sample VIEWCON EXEC shipped with Operations Manager:

```
viewcon lxsyslog
```

If you are not using the sample VIEWCON EXEC shipped with Operations Manager, then issue:
```
gomcmd opmgrm1 viewcon user lxsyslog
```

Figure 8 shows the viewcon command being issued and Figure 9 shows the resulting screen.

Note: The user lxsyslog keyword refers to the name found in positions 1-8 of the PARM operand of the DEFIPCS statement.

Note: If necessary, to authorize a user to view this syslog data specifically, issue the following command (and add the AUTH statement to your Operations Manager configuration file):

```
gomcmd opmgrm1 auth user userid console lxsyslog
```

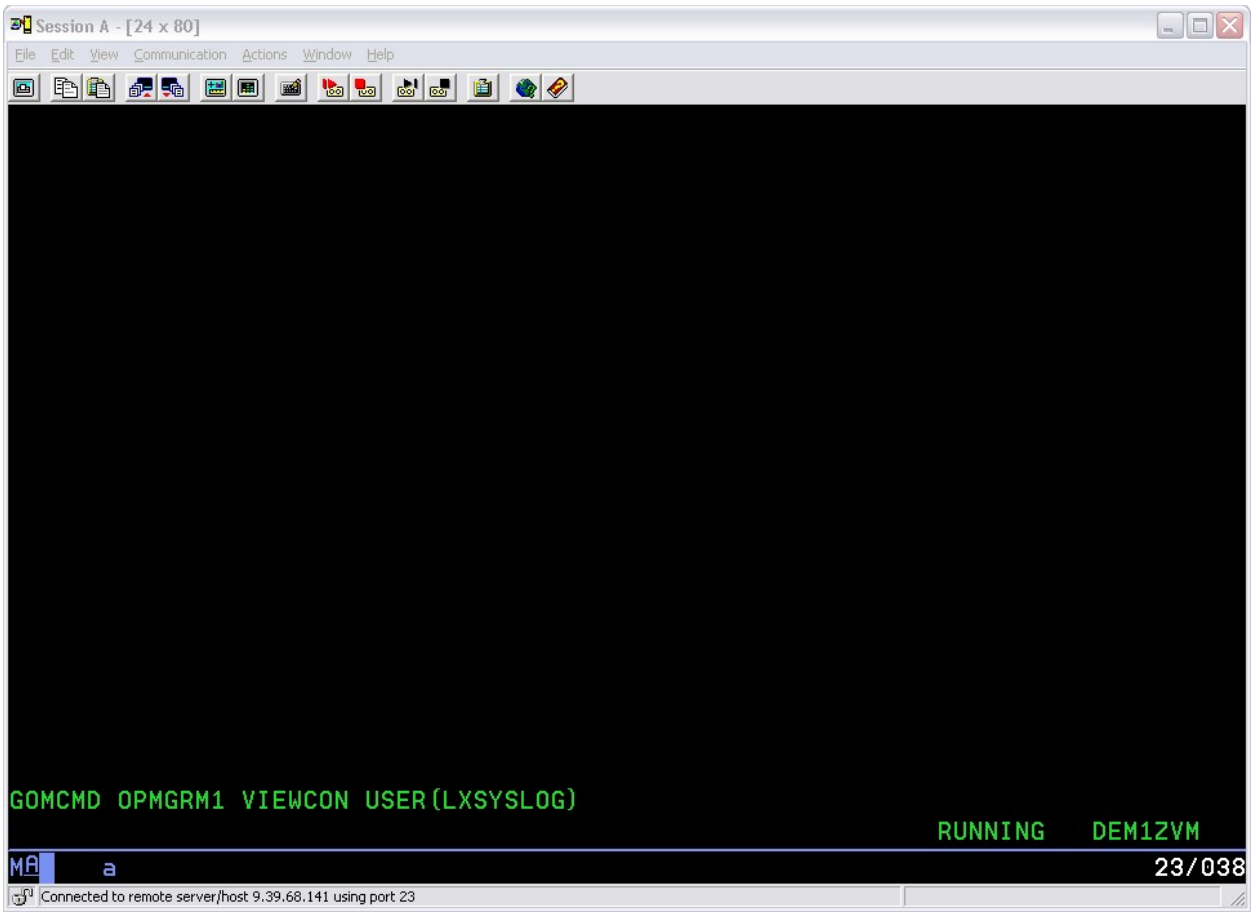

Figure 8

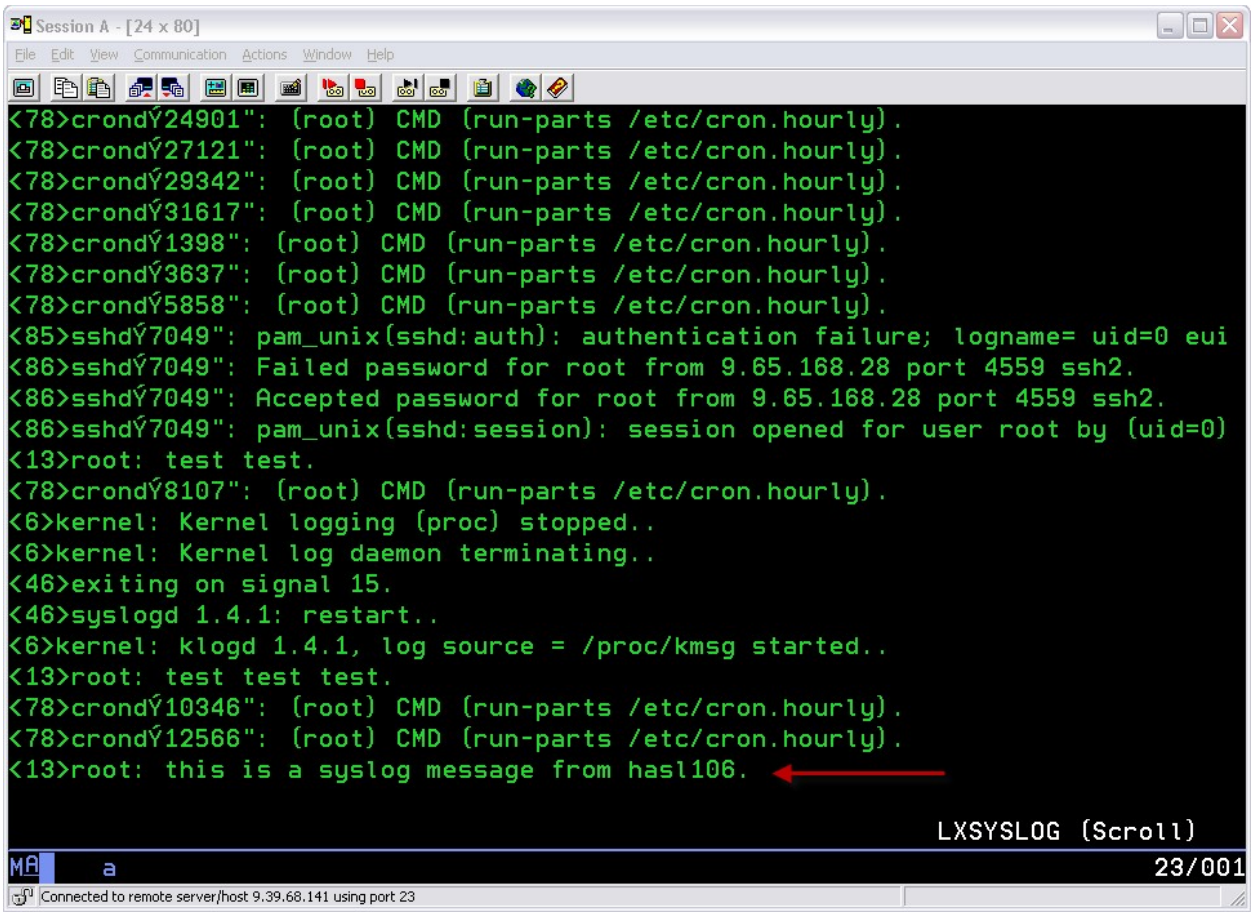

Figure 9

Note: The red arrow points to the logger message arriving from HASL106.

To have Operations Manager send its entire log for today (including the syslog data) to your z/VM reader, use one of the following commands:

```
gomcmd opmgrm1 viewlog date 'yyyy/mm/dd'
viewlog date 'yyyy/mm/dd'
```

Specify today's date for yyyy/mm/dd.

Figure 10 below shows this command issued and the resulting reader file sent message.

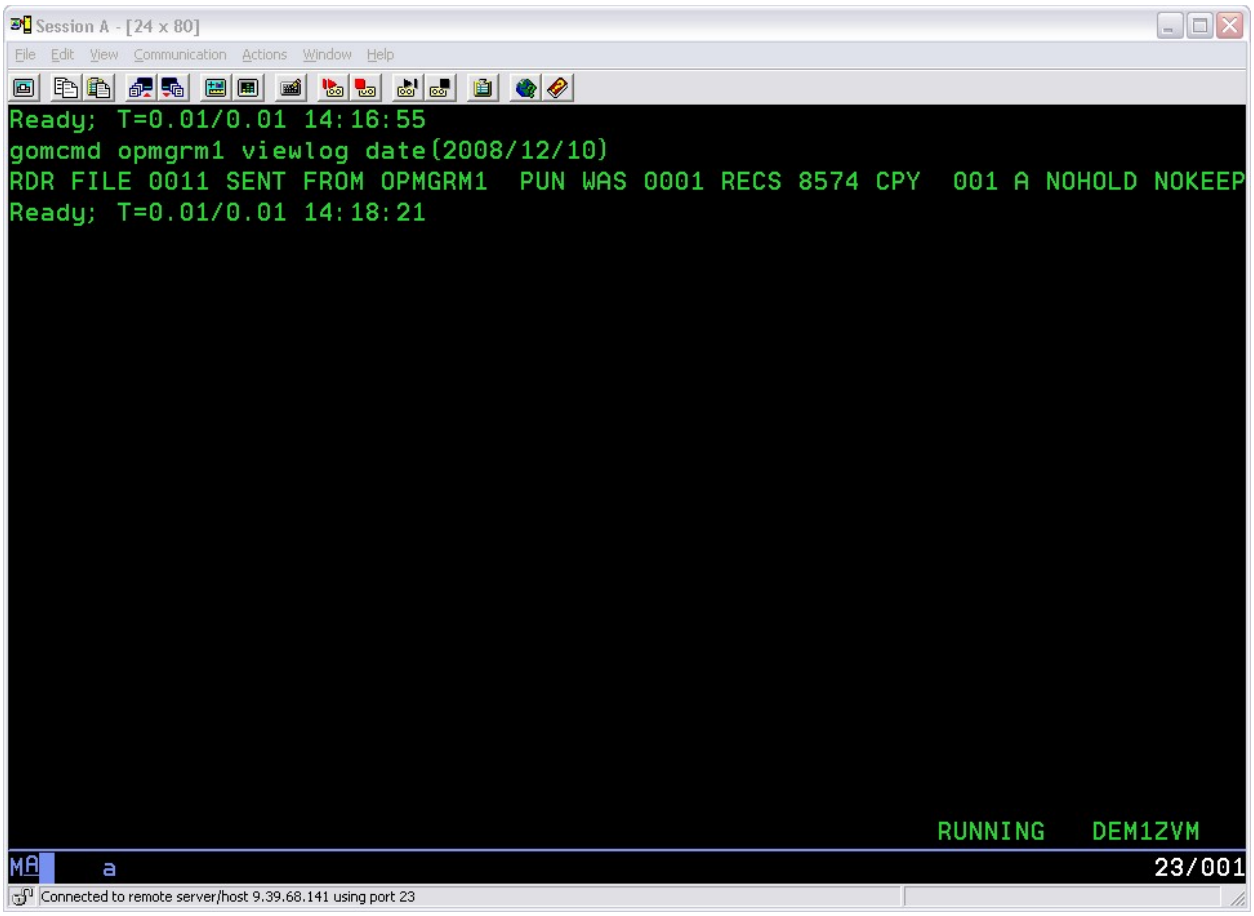

Figure 10

Multiple Syslog Routes

Many customers today have realized that Operations Manager for z/VM can be a repository, to allow customers to meet the best practices of centralized log management. Adding syslog routes to Operations Manager for z/VM will complement and not interrupt any existing or future routes customers may need for local Linux or UNIX management strategies. The following two examples are simple scenarios to help demonstrate this point.

SYSLOG-NG Example:

Figure 11 below shows /etc/syslog-ng/syslog-ng.conf file updated to route all f_messages to three different destinations. While three destinations may not be typical, it demonstrates the flexibility available to syslog configurations. The three red arrows in Figure 11 point to three distinct destination definitions: two remote, and one local. The destinations labels are loghost, loghost2, and d_file. These labels are then added to the log statement that combines the source, the filter, and the destinations of this specific route.

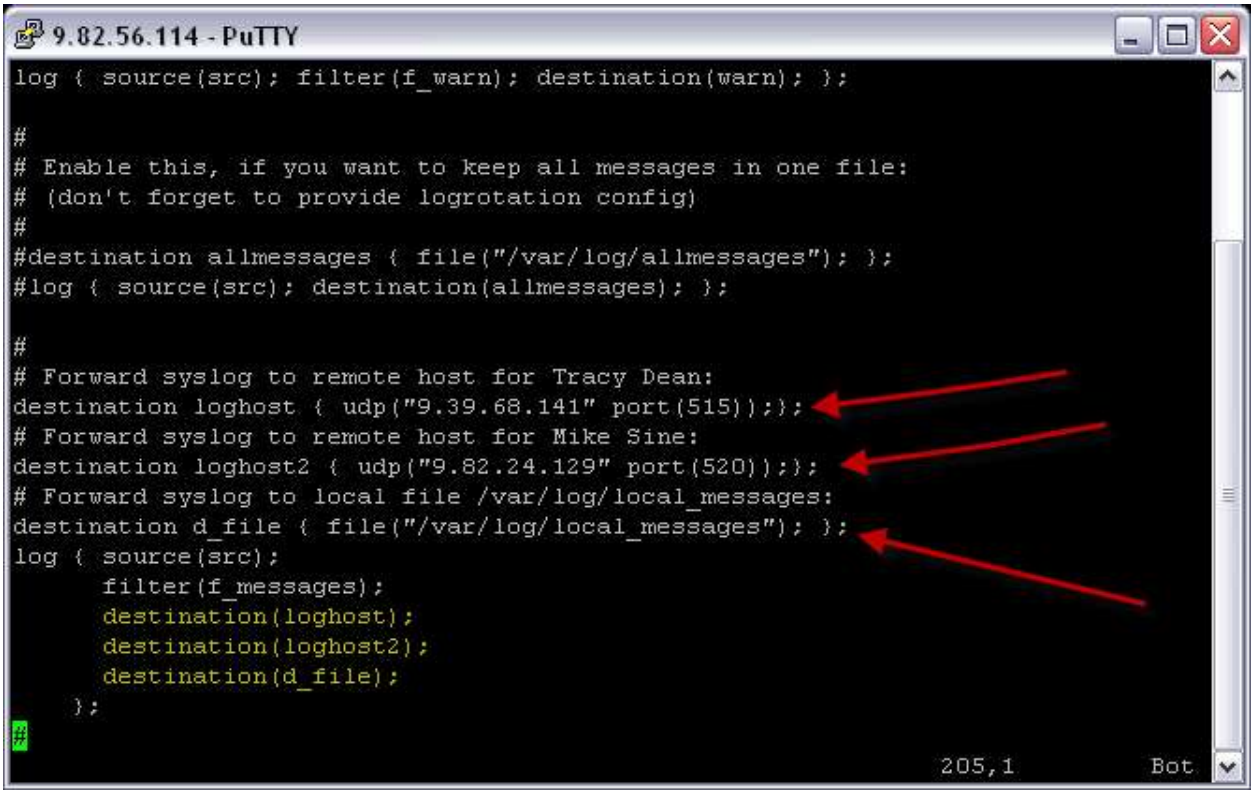

Figure 11

With this syslog-ng configuration, any f_messages will be routed to the three defined destinations. In Figure 12 below, we again issue the logger command to test these routes.

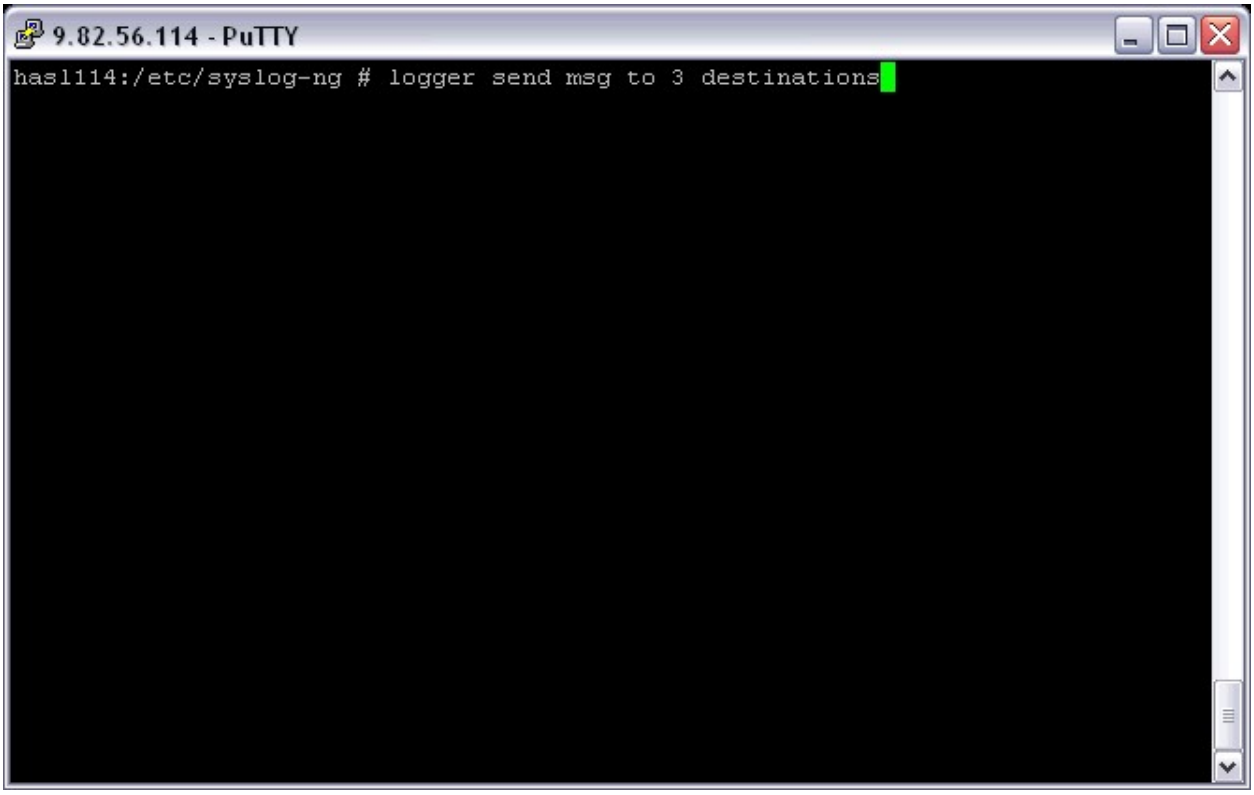

Figure 12

Figure 13 shows the route to loghost, which is an Operations Manager for z/VM listener:

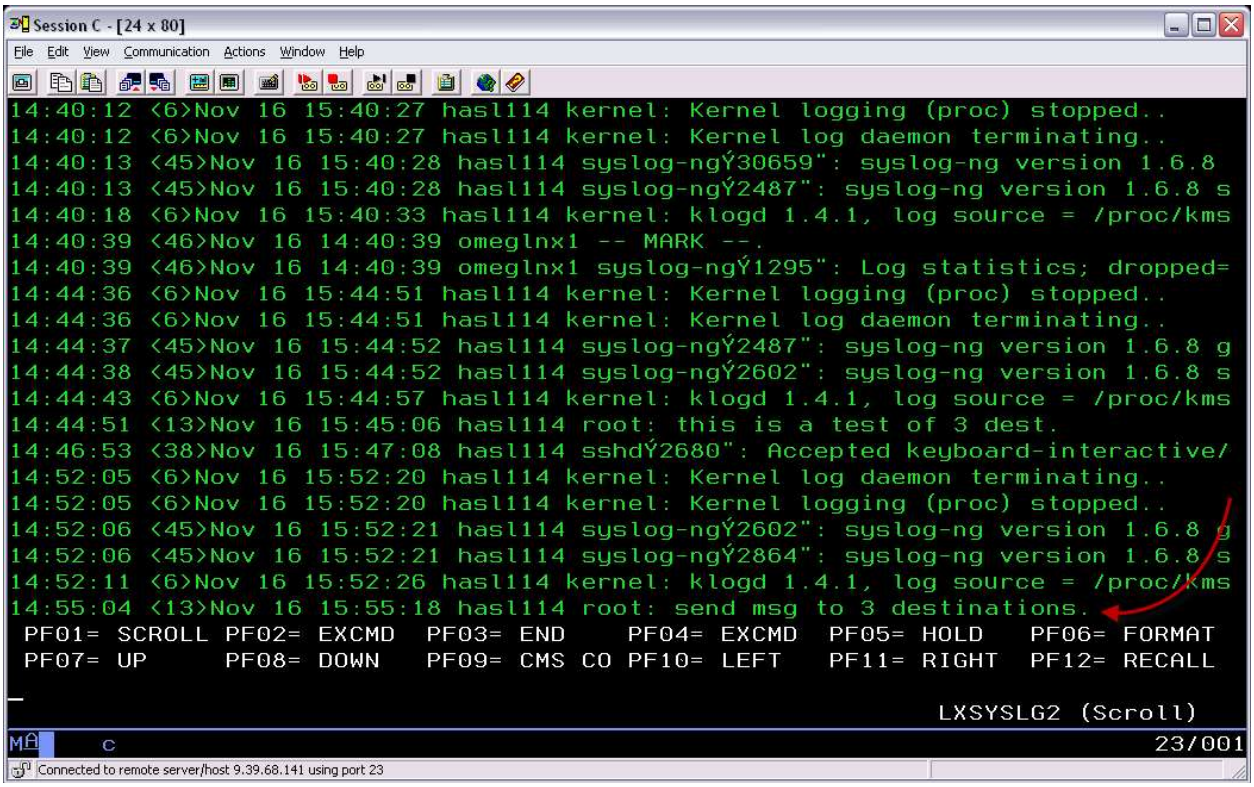

Figure 13

Figure 14 shows the route to loghost2, which is another Operations Manager for z/VM listener on another z/VM system:

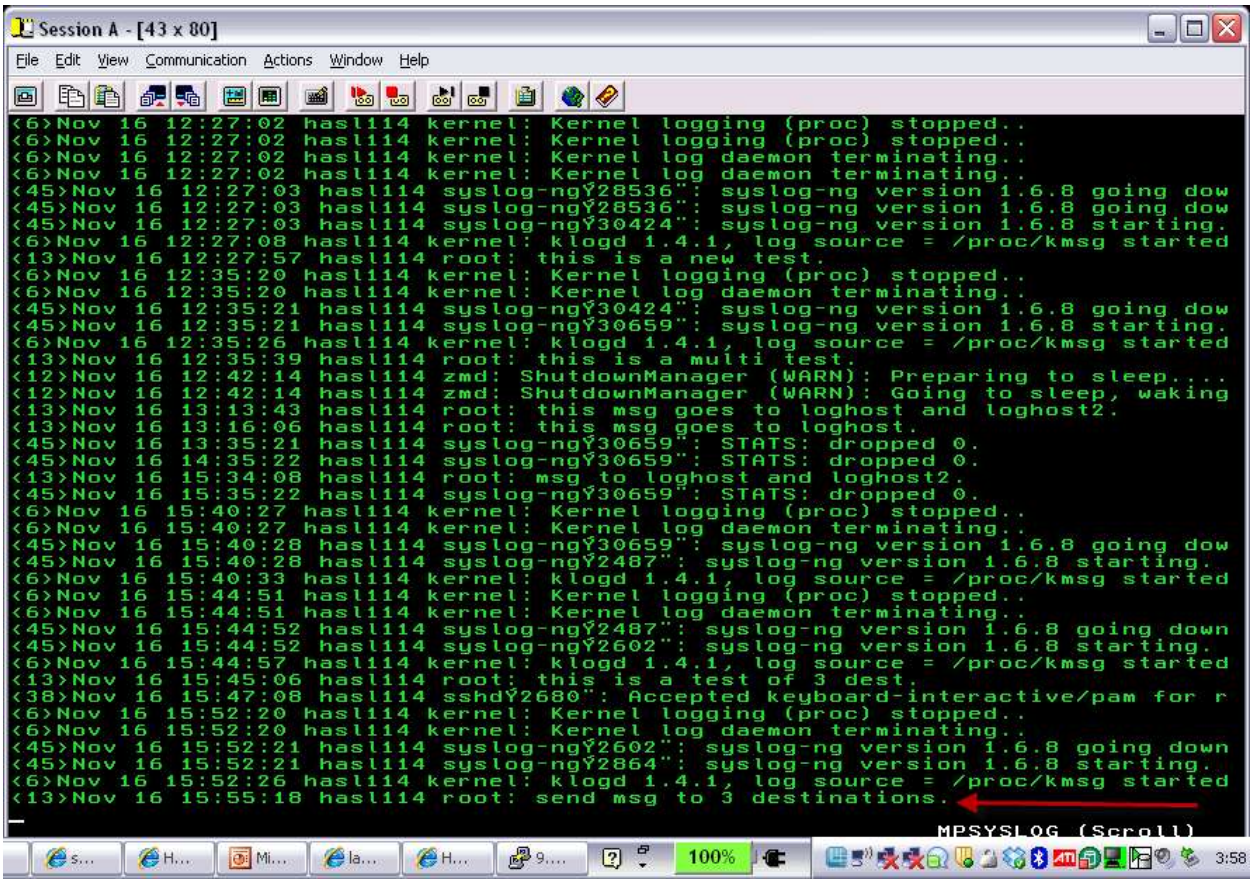

Figure 14

Figure 15 shows the route to d_file, a local file on the Linux guest:

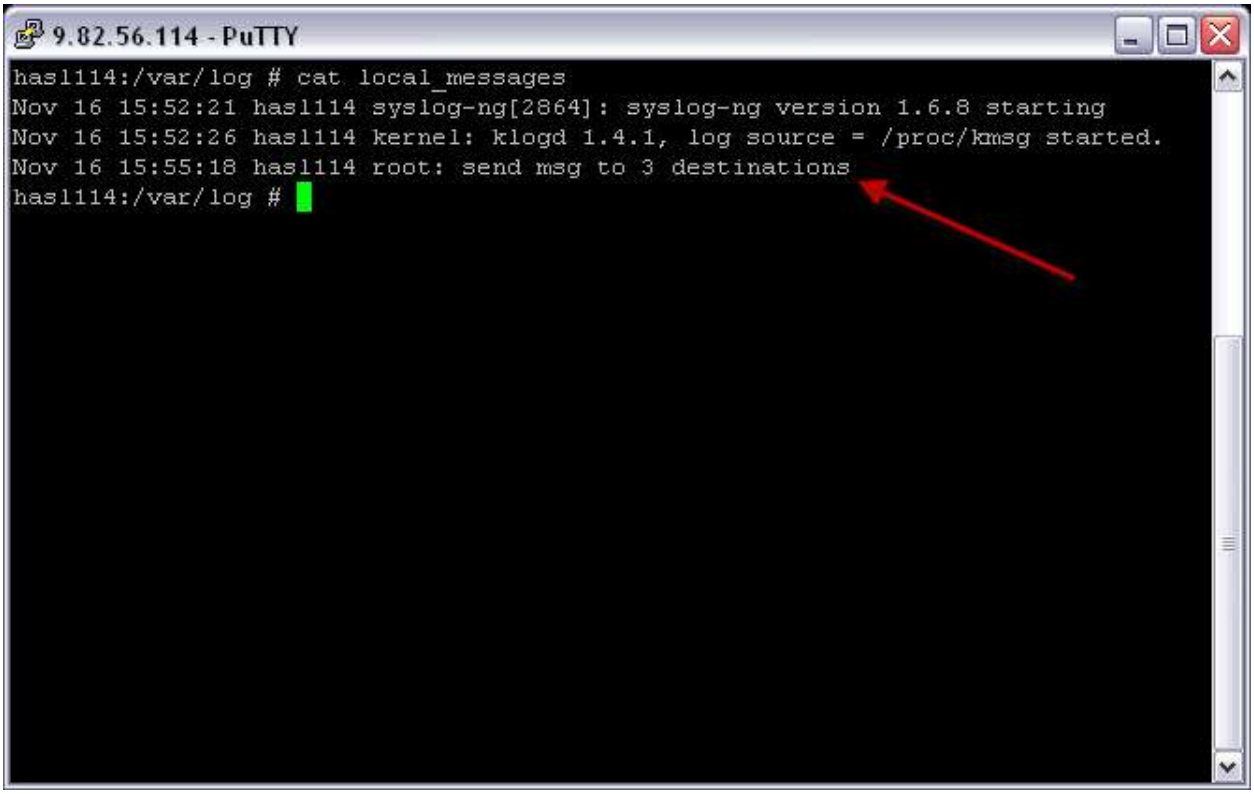

Figure 15

RSYSLOG Example:

Figure 16 below shows /etc/rsyslog.conf file updated to route all messages to two different destinations. While other routes also exist in the configuration file (ex: local7.* messages routing to /var/log/boot.log as seen at the tope of Figure 16), the two routes identified with the red arrows in Figure 16 are easy to demonstrate with a logger test message and will be highlighted in the screen captures below. The two red arrows below show one route to an Operations Manager for z/VM listener as well as a route to the file /var/log/local_messages.

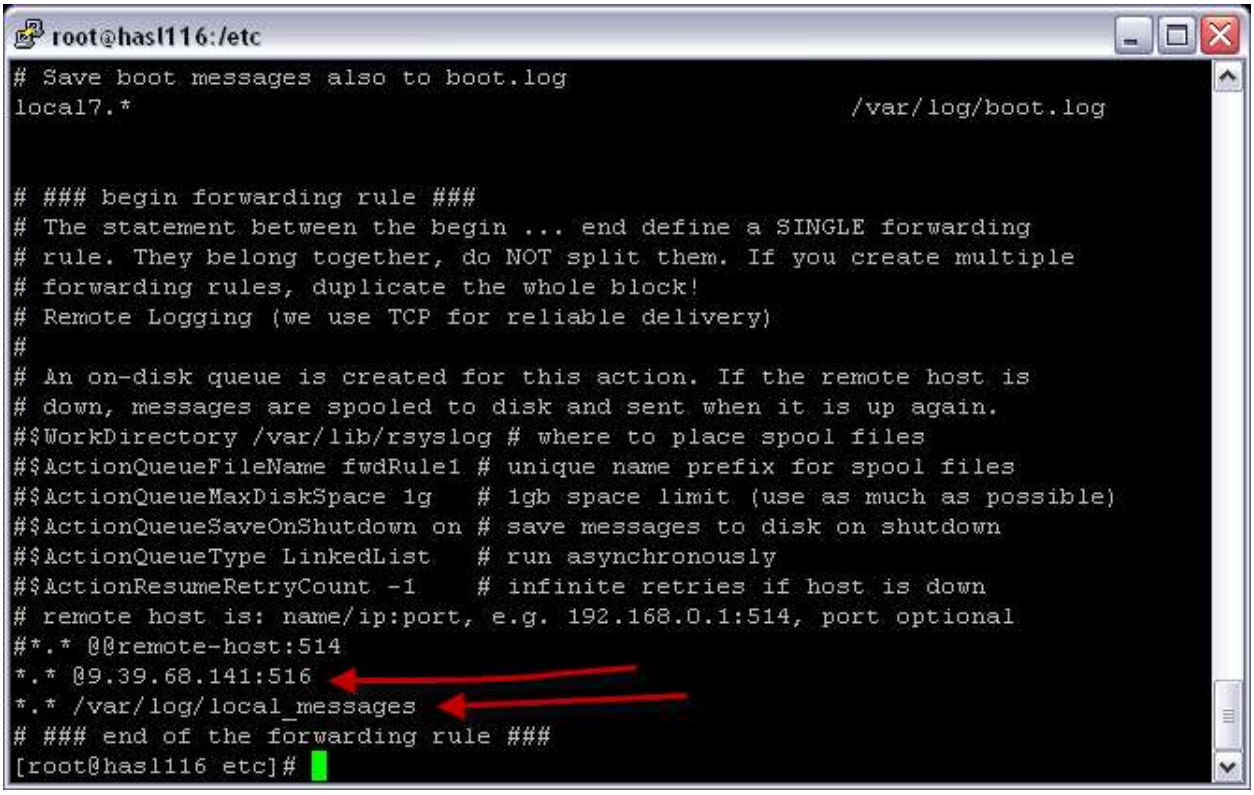

Figure 16

The logger test message again demonstrates these routes:

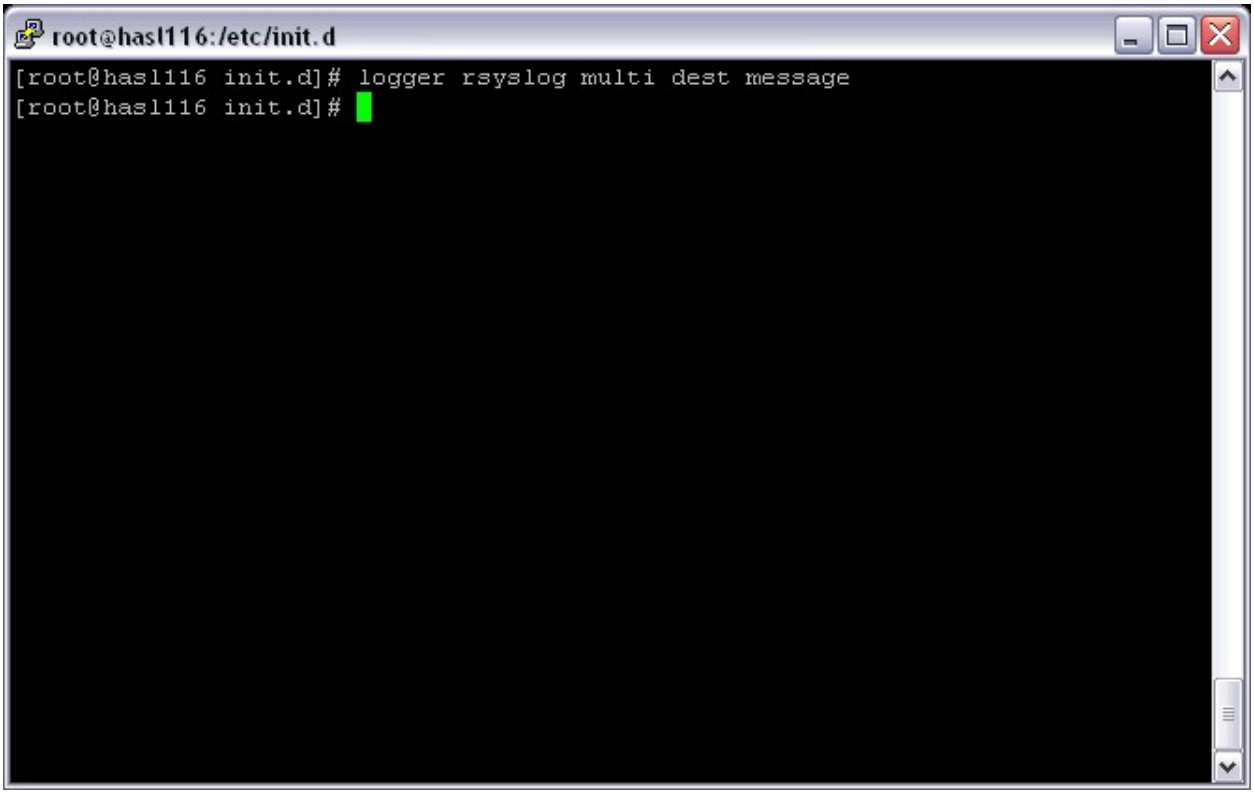

Figure 17

Figure 18 shows the route to Operations Manager for z/VM at port 516:

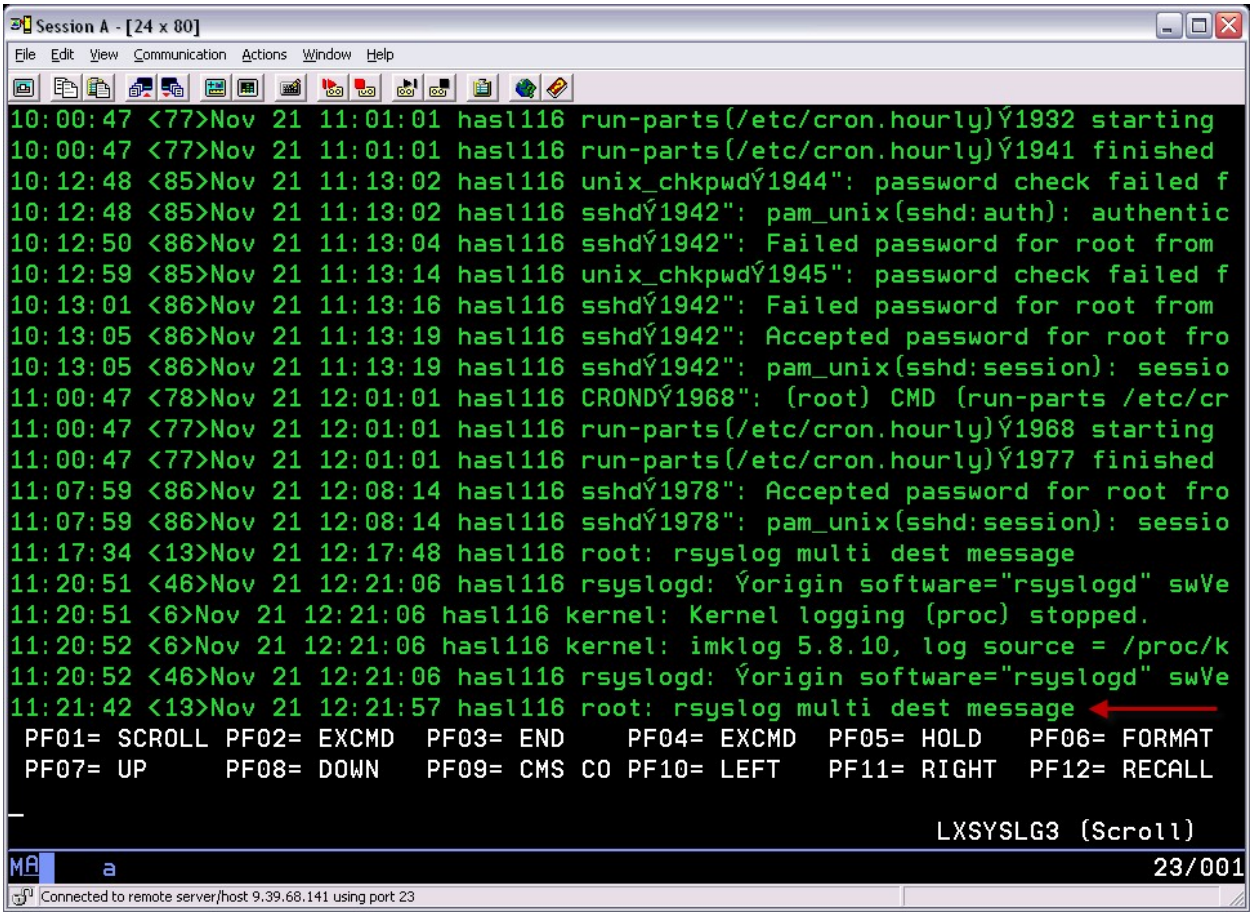

Figure 18

Figure 19 shows the route to /var/log/local_messages

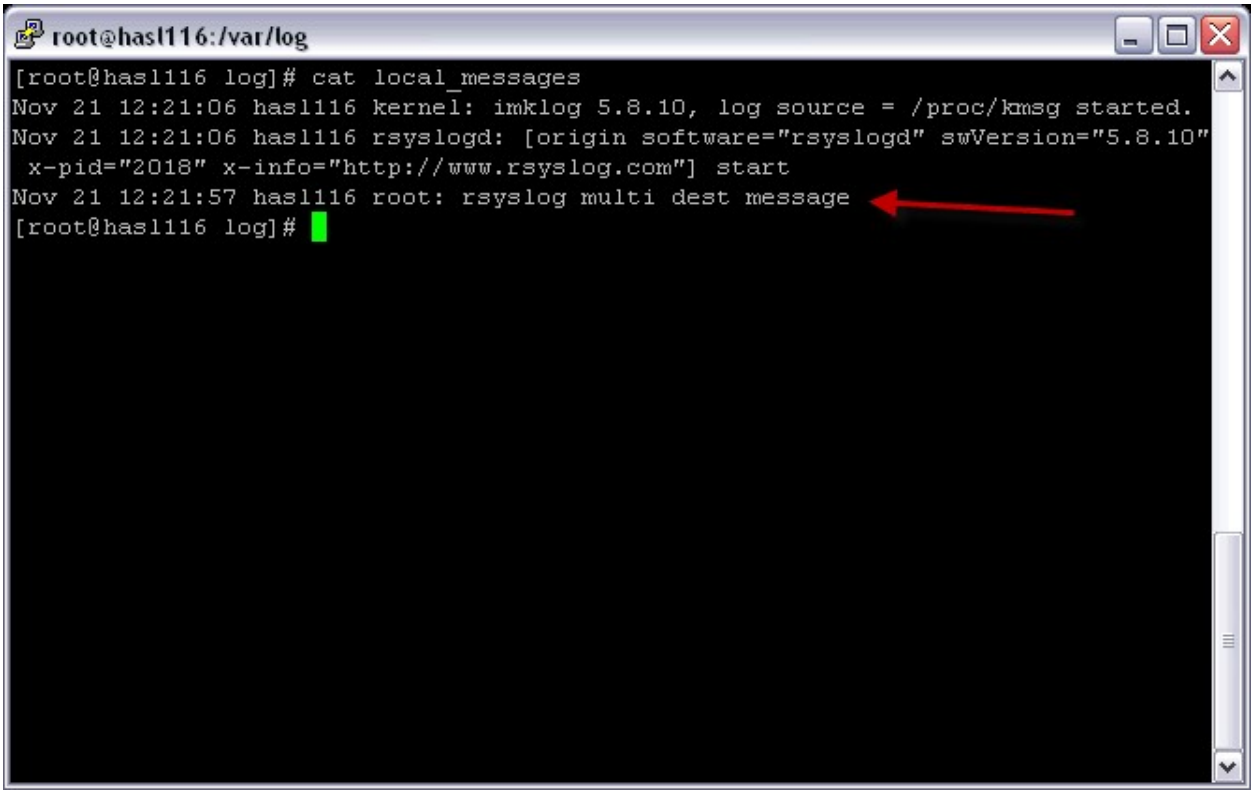

Figure 19

Routing Application and other log files

While syslog collection is very important, application log management may be even more important in many production virtual servers. Making use of rsyslog log file management functions, Linux application log files can be centrally managed along with the Linux syslog and console data via Operations Manager for z/VM. This section will demonstrate how rsyslog may be configured to transfer log files to Operations Manager for z/VM. While the example will be a default log file, this functional can be used for any application log file (or any log file) to be included in a centralized z/VM and Linux management infrastructure documented in this paper.

Note: The authors have found that syntax for log file management appears to have changed through the versions of rsyslog. The example in this paper is a flavor of rsyslog version 5 that came with RedHat 6.4 64 bit. Be sure to review the man pages associated with your version of rsyslog to confirm any examples you use are supported by your version of rsyslog.

Using rsyslog to monitor log files is made possible due to imfile: text file input module. The imfile module converts any standard text file's line(s) into a syslog message(s). A standard text file is a file consisting of printable characters with lines being delimited by LF. The file is read line-by-line and rsyslog's rules are applied to each line applying filters and actions. Blank lines are ignored. As new lines are added to the file, they are processed against the rsyslog rules.

New lines are processed on a polling interval, not immediately. A state file is used to keep track of the where rsyslogd left off when last reading lines from the file. File rotation is supported, but be sure to read the rules for file rotation associated with your version of rsyslog to understand what lines in the file will be missed or included as an active rotation takes place and rsyslogd is stopped and restarted.

Configuration Directives

The following configuration directives were taken directly from www.rsyslog.com at the time of this writing. For any syntax changes or updates, please refer back to this website. It is important to copy these directives into this document to understand the example provided based on the current directives, their syntax, and their current functionality.

```
$InputFileName /path/to/file
```

The file being monitored. So far, this must be an absolute name (no macros or templates)

```
$InputFileTag tag:
```

The tag to be used for messages that originate from this file. If you would like to see the colon after the tag, you need to specify it here (as shown above).

```
$InputFileStateFile /path/to/state/file
```

rsyslog must keep track of which parts of the monitored file it already processed. This is done in the state file. This file is always created in the rsyslog working directory (configurable via $WorkDirectory). Be careful to use unique names for different files being monitored. If there are duplicates, all sorts of "interesting" things may happen. rsyslog currently does not check if a name is specified multiple times.

```
$InputFileFacility facility
```

The syslog facility to be assigned to lines read. Can be specified in textual form (e.g. "local0", "local1", ...) or as numbers (e.g. 128 for "local0"). Textual form is suggested. Default is "local0".

```
$InputFileSeverity severity
```

The syslog severity to be assigned to lines read. Can be specified in textual form (e.g. "info", "warning", ...) or as numbers (e.g. 4 for "info"). Textual form is suggested. Default is "notice".

```
$InputRunFileMonitor
```

This activates the current monitor. It has no parameters. If you forget this directive, no file monitoring will take place.

```
$InputFilePollInterval seconds
```

This is a global setting. It specifies how often files are to be polled for new data. The time specified is in seconds. The default value is 10 seconds. Please note that future releases of imfile may support per-file polling intervals, but currently this is not the case. If multiple $InputFilePollInterval statements are present in rsyslog.conf, only the last one is used.

A short poll interval provides more rapid message forwarding, but requires more system resources. While it is possible to set the polling interval to 0 seconds, we strongly recommend against it. That will make rsyslogd become a CPU hog, taking up considerable resources. It is supported, however, for the few very unusual situations where this level may be needed. Even if you need quick response, 1 second should be sufficient. Please note that imfile keeps reading files as long as there is any data in them. So a "polling sleep" will only happen when nothing is left to be processed.

Can you clarify the last 2 sentences – does it mean it keeps reading as long as there is new data? What does "nothing left to be processed" mean? The file no longer exists or there is no new data? How does it know unless it polls and checks?

```
$InputFilePersistStateInterval lines
```

Specifies how often the state file shall be written when processing the input file. The default value is 0, which means a new state file is only written when the monitored file is closed (end of rsyslogd execution). Any other value n means that the state file is written every time n file lines have been processed. A value of 0 can be used to guard against message duplication due to fatal errors (like power fail). Note that this setting affects imfile performance, especially when set to a low value. Frequently writing the state file is very time consuming.

```
$InputFileReadMode mode
$InputFileMaxLinesAtOnce number
```

This is useful if multiple files need to be monitored. If mode (??) is set to 0, each file will be fully processed and then processing switches to the next file (this was the default in previous versions). If mode is set to ??, a maximum of [number] lines is processed in sequence for each file, and then the file is switched. This provides a kind of mutiplexing of multiple files and probably leads to a more natural distribution of events when multiple busy files are monitored. The default is ?? for mode and 10240 for number??.

```
$InputFileBindRuleset ruleset
```

Binds the listener to a specific *ruleset*.

Working Configuration Example

The rsyslog example above will be modified to include a log file from the same Linux system and write its lines as syslog entries to the same port already defined in the previous rsyslog example. Multiple ports can be used as documented elsewhere in this paper. This section is intended to show a simple configuration for log file collection. All documented functions describing integration to Operations Manager for z/VM on shared or separated ports apply to log file collection as they do for any syslog collection.

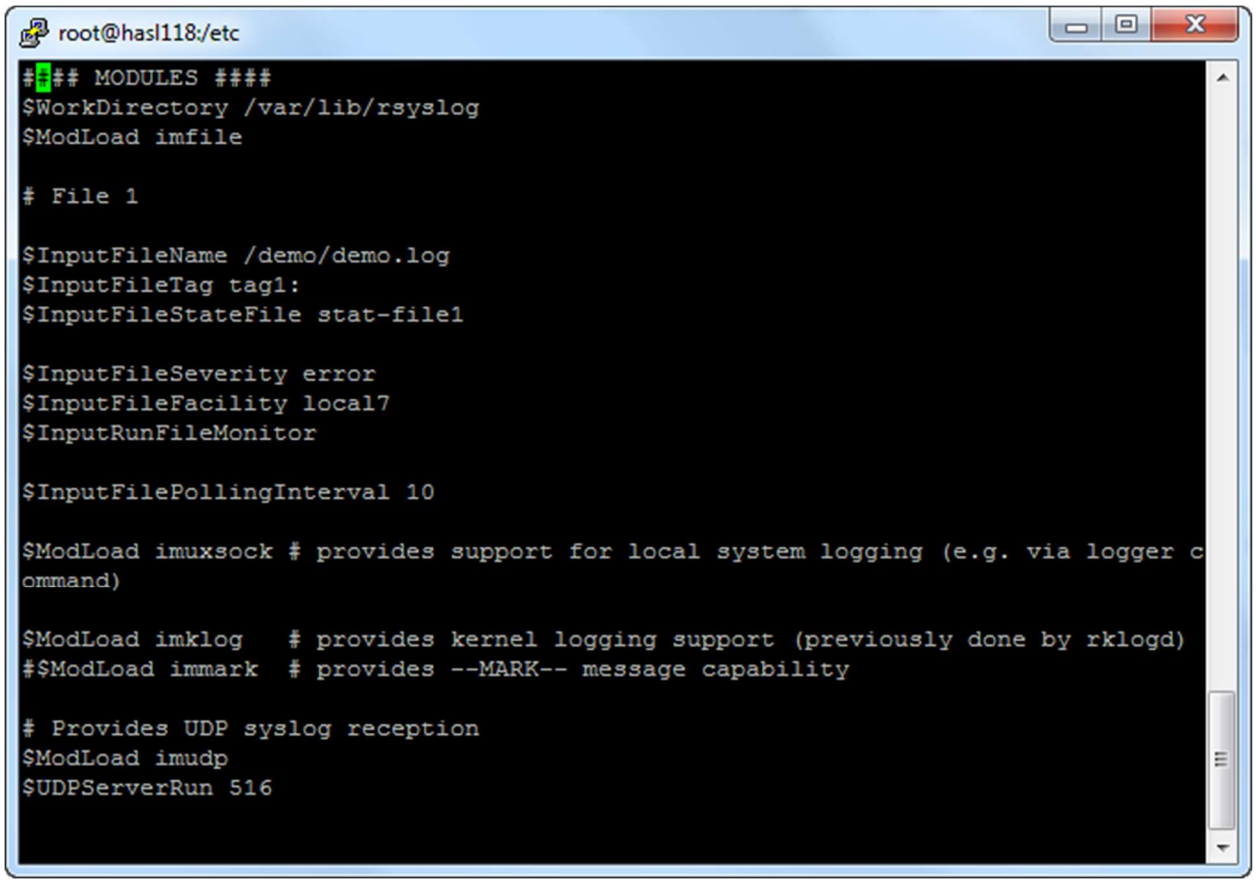

Figure 20

The authors have moved some of the configuration options around in this running configuration file, Figure 20, in order to more easily discuss what options were specified and how to work with this example.

$WorkDirectory: This directive identifies rsyslog's working directory. For log file management, this directive will identify where the state file will be located.

$ModLoad: In order for any of the log file directives to work, the imfile module must be loaded.

$InputFileName: Specifies the file to be monitored. For our example, the file being monitored is /demo/demo.log

$InputFileTag: This directive will add a tag to the beginning of each line from the file when it is converted to a syslog message. This can be very helpful in identifying where the messages originated if you are sharing loghost destination ports.

$InputFileStateFile: This directive specifies the name of the state file rsyslogd will use to identify where it left off the last time it read from the monitored file. Remember, this file will be created ?? (or do I have to create it) in the directory specified in the $WorkDirectory directive. The state file for this example will: /var/lib/rsyslog/stat-file1

$InputFileSeverity and $InputFileFacility: assign a syslog severity and facility to the lines read. The syslog rules for these values will be applied.

$InputRunFileMonitor: This starts the file monitor. Without this, file monitoring will not happen. To turn off file monitoring in this configuration, comment out this line.

$InputFilePollingInterval: The polling interval is set to 10 seconds. This is perfect for demonstrations, testing, and simple examples; however, this is one area that must be given consideration. It will be necessary to balance frequency of collection with the performance impact associated if the polling is too aggressive.

With this configuration activated, issue: *service rsyslog restart*

The file /demo/demo.log will now be monitored.

The lines added to this file will be given:

- a tag of tag1:
- a facility value of local7
- a severity value of error
- a state file of /var/lib/rsyslog/stat-file1

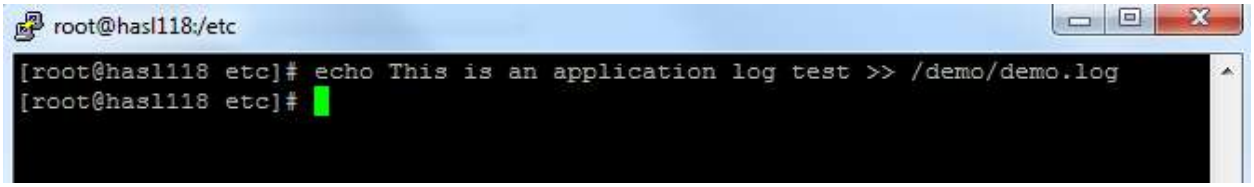

Figure 21

In Figure 21 above, we see a simple echo command appending text to the file /demo/demo.log. This text will now be converted to a syslog message with the tag tag1: prefixing the text.

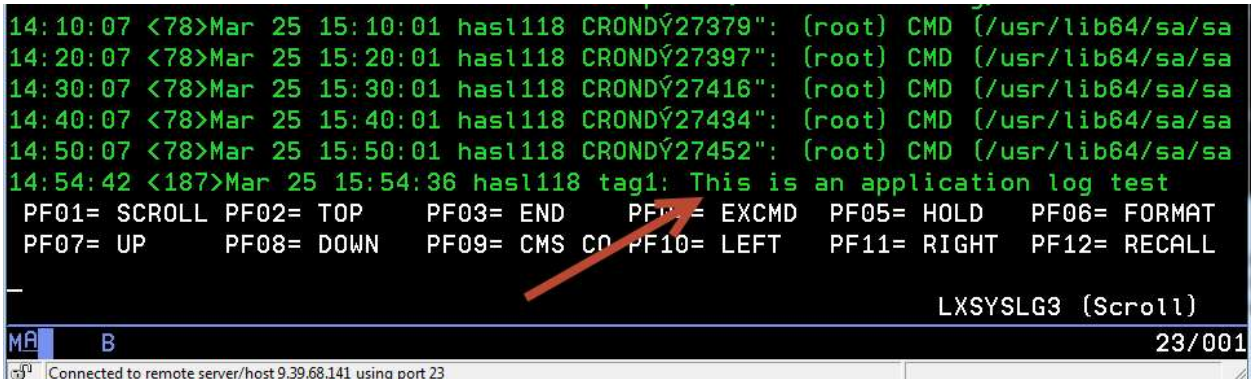

Figure 22

Figure 22 above shows the viewcon of user lxsyslg3 which is configured to monitor port 516 for syslog messages. Notice the text “tag1: This is an application log test”. Figure 23 below shows the latter part of the rsyslog.conf file similar to the configure shown earlier for rsyslog example.

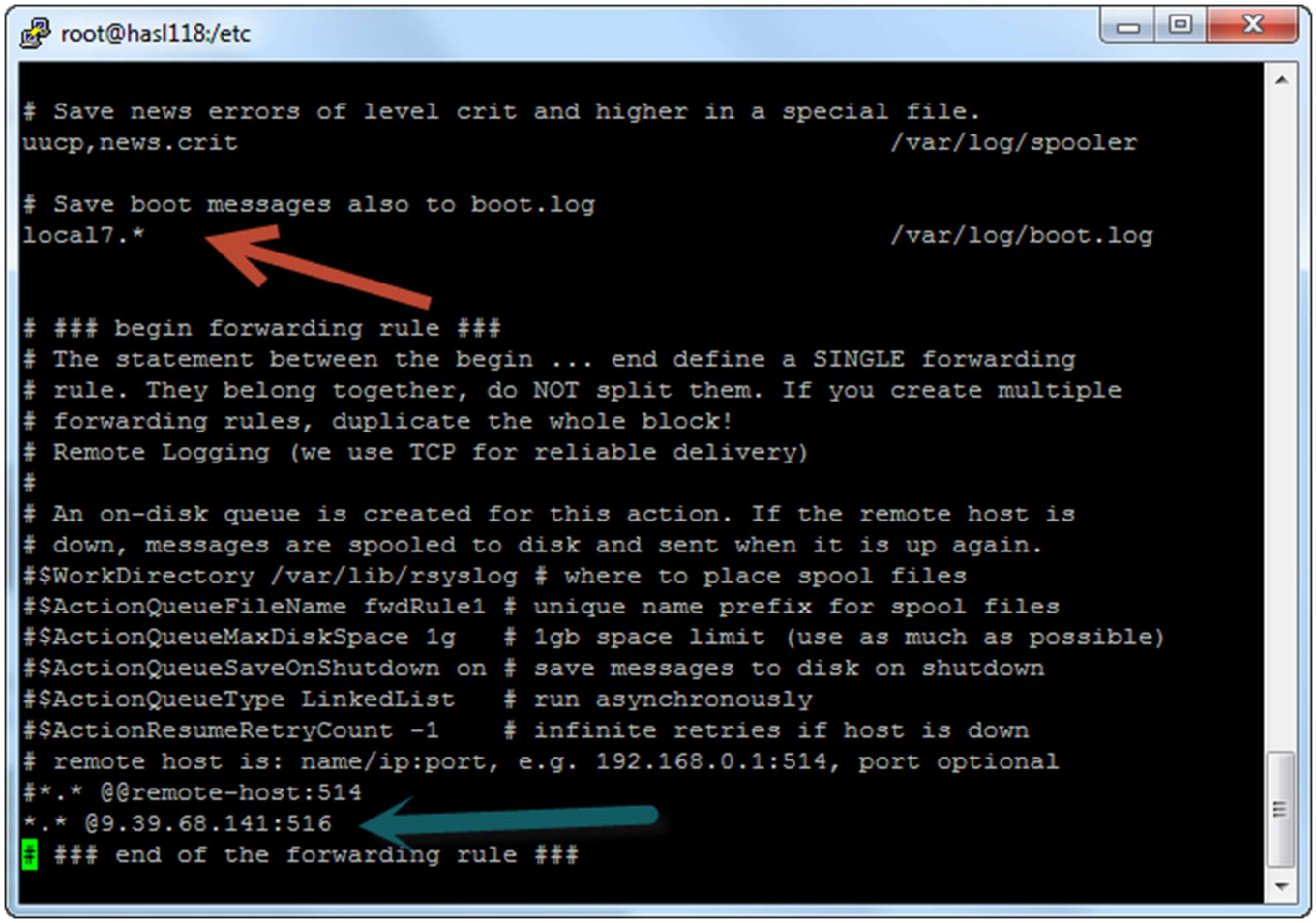

Figure 23

Note the red arrow shows that all local7 facility messages will be logged in /var/log/boot.log (see Figure 24). The blue arrow shows that all (*.*) messages will be routed to the loghost 9.39.68.141 on port 516. This is the Operations Manager for z/VM destination.

```
Starting certmonger: [  OK  ]
Mar 25 12:43:54 hasl118 tag1: this is a demo.log updated test2
Mar 25 12:52:14 hasl118 tag1: this is a demo.log updated test3
Mar 25 15:54:36 hasl118 tag1: This is an application log test
[root@hasl118 etc]#
```

Figure 24 (the bottom of /var/log/boot.log file)

Figure 21 – 24 show

- A line of text being written to /demo/demo.log
- That same text being sent as a syslog message to Operations Manager for z/VM
- The successful classification of this message with a facility of local7
- The message written to /var/log/boot.log
- The severity of message falling into the all encompassing severity of *.* routing to Operations Manager for z/VM.

Therefore, the text of /demo/demo.log is not just sent forward to the Operations Manager for z/VM loghost, it is coverted to a valid syslog message with directives applying attibutes to the

message that match additional rules in the rsyslog.conf file and those rules and actions apply to this message as well.

Summary

The Operations Manager for z/VM DEFIPCS configuration statement allows you to monitor local and remote files, application logs, and syslog data. Adding this configuration statement, authorizing Operations Manager to listen for syslog data, and the simple updates to Linux and UNIX systems (ex: AIX) to route their syslog data to Operations Manager allows you to use the VIEWCON command to monitor file, application logs, and syslog data in the same way that consoles are monitored. It also allows Operations Manager rule processing to apply to this data.

With these methods

- Centralized management of log messages to Operations Manager for z/VM helps customers meet log management best practices
- Local syslog management requirements are maintained