zVPS Alerts

Richard Smrcina
Velocity Software, Inc.
Performance Workshop
June, 2016

VELOCITY SOFTWARE

PROVEN PERFORMANCE

Copyright 2012 Velocity Software, Inc. All Rights Reserved. Other products and company names mentioned herein may be trademarks of their respective companies.

Agenda

- Overview
- What are alerts?
  - Where do alerts fit
- Installing zAlert package
  - Viewing alerts
- Alert samples
- Defining your own alert
  - CPU Utilization
  - LPAR Utilization
- Notification
  - Email
  - SNMP trap
- Level Text
- Alert options
- Enable/Disable
- Limit
- Time
- Include/Exclude
- Multiple alerts
- External processing
- Integration with zOperator
- Operating zAlert

What are alerts?

- **An alert is an indication of an abnormal condition**
- **An abnormal condition can be**
  - Exceeding a certain threshold
  - An object in a state not conducive to proper operation
    - Volume offline
    - Virtual machine not logged on
    - Incorrect system settings

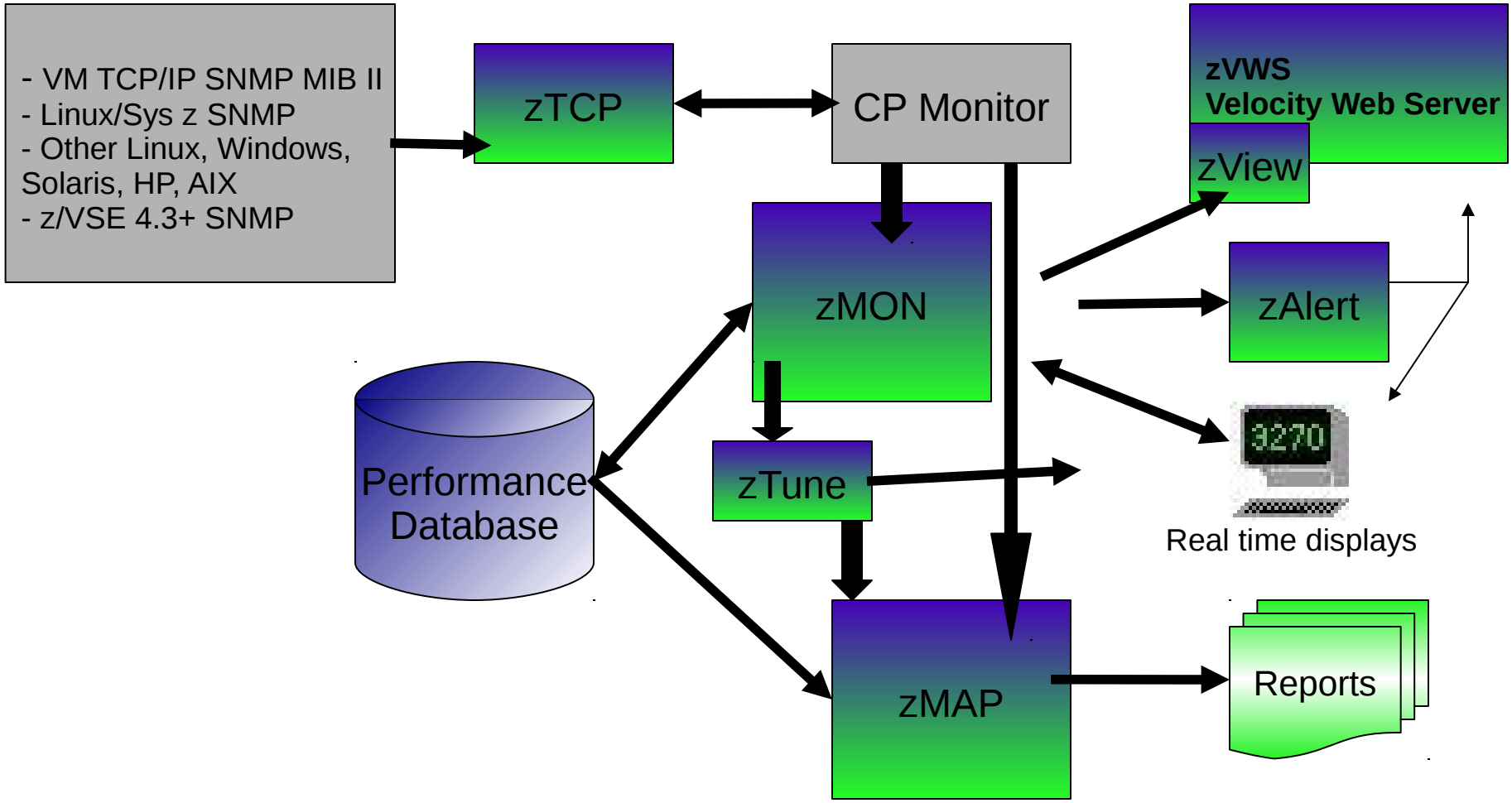
Overview

This presentation goes through the finer points of alert processing.

Where alerts come from, how they are used and specified in the product.

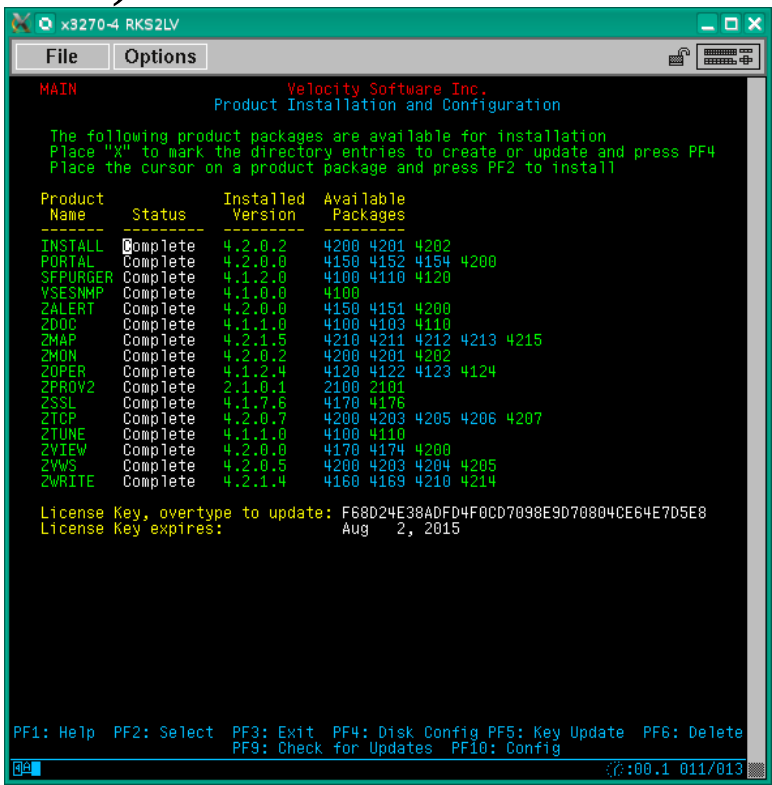
Alerts are no good if they need to be visually watched or monitored... notifications provide a proactive mechanism to using alerts.

Where do alerts fit?



Installing zAlert package

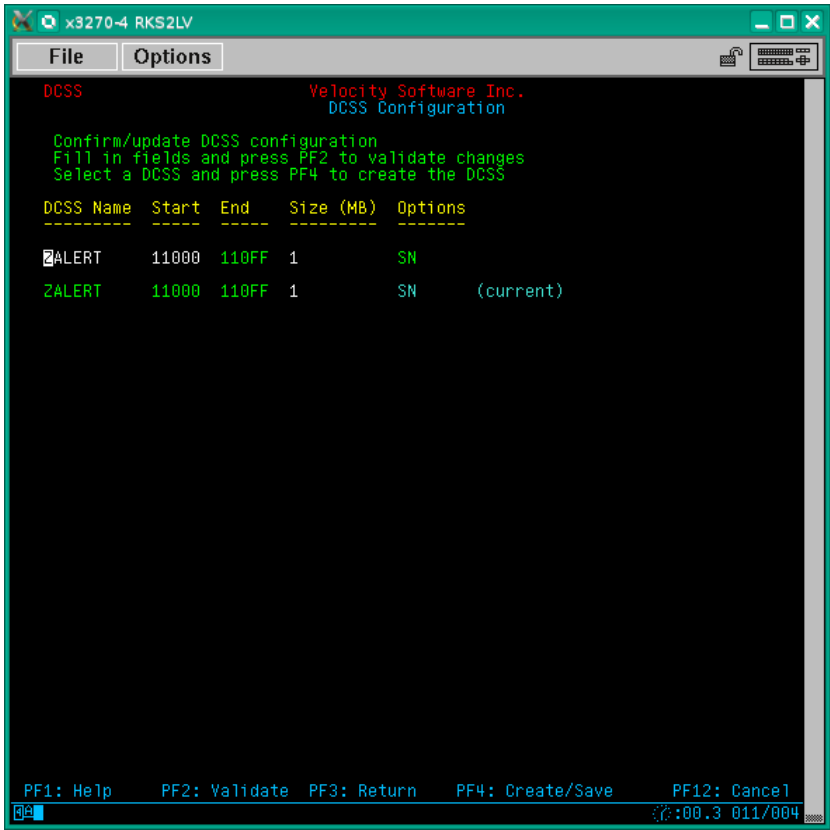
- **zAlert is part of the Velocity Performance Suite (zVPS)**
- **Installed via the zVPS installer**
  - Requires a virtual machine (ZALERT)
  - SFS filespace
    - By default SFSZVPS:ZALERT.
  - DCSS to keep alert messages
  - Sample alerts provided
  - More on the website



VELOCITY SOFTWARE

Installing zAlert package

- zAlert 4.2 separates the alert function (alert engine) from the display function
- Requires a DCSS for proper operation
- Alert messages stored in the DCSS
  - Message retrieval handled by a separate EXEC
- zAlert 4.1 can still be used as is, but is functionally stabilized



VELOCITY SOFTWARE

Installing zAlert package

- **zAlert directory entry**
- **ZALERT DCSS is unrestricted (4.2)**
  - NAMESAVE statement not required for ZALERT
- **ZMON DCSS is required**

```
USER ZALERT ALERTS 32M 32M G
   INCLUDE VSIPROF
   IPL CMS PARM FILEPOOL SFSZVPS:
   IUCV ALLOW
   NAMESAVE ZMON ZALERT
   XAUTOLOG ZSERVE ZVPS
```

Alert processing

- **Separate virtual machine is used to process alerts and send notifications**
- **The alert virtual machine wakes up every minute, reading the configured MONALERT file(s)**
- **Each of the defined extracts is executed**
  - Values returned from extracts is compared against user defined thresholds
  - Message displayed or action taken when thresholds are exceeded

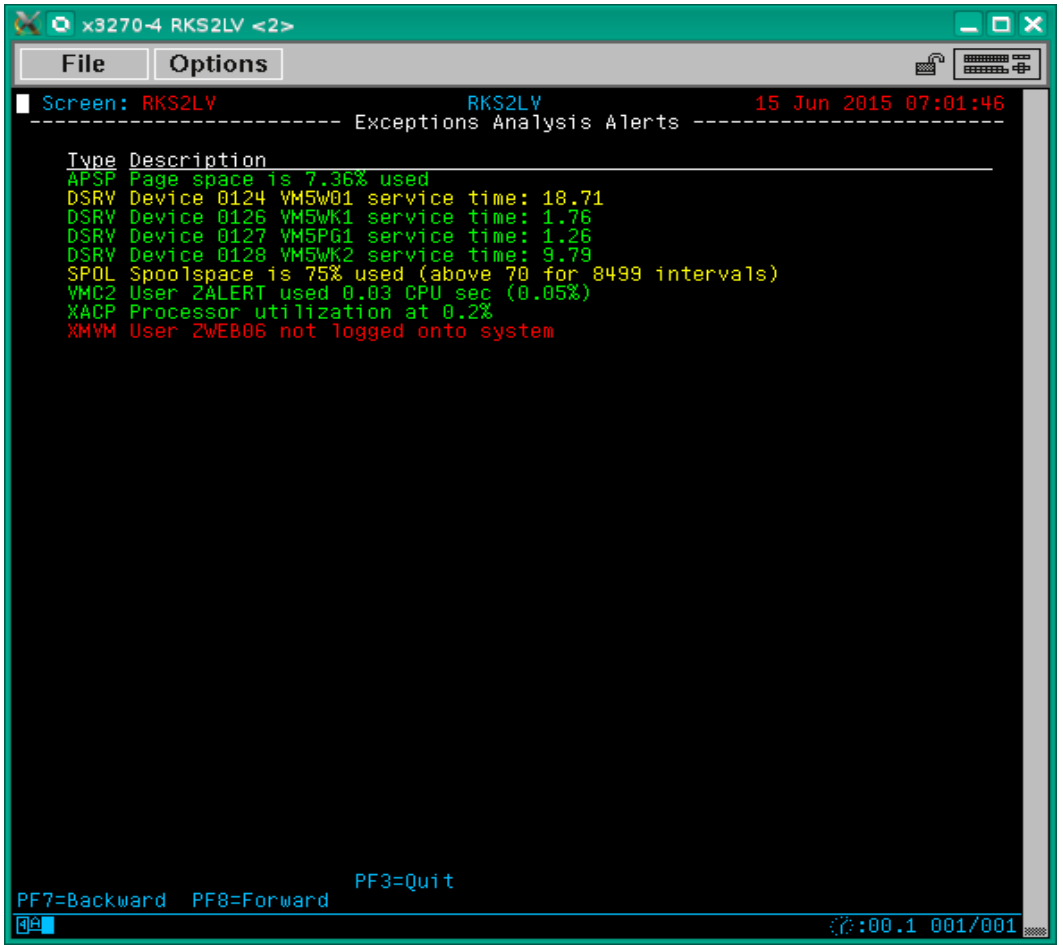
Installing zAlert package

- **A notification can be any of**
  - Message on the alert console, zView or zAlert CGI
  - CP MSG to a user
  - Email to interested parties
    - Text message on a mobile device
  - SNMP trap sent to a management console
  - Combinations of the above

Viewing alerts

- **Terminal session**
  - ZALERT <alertfile>

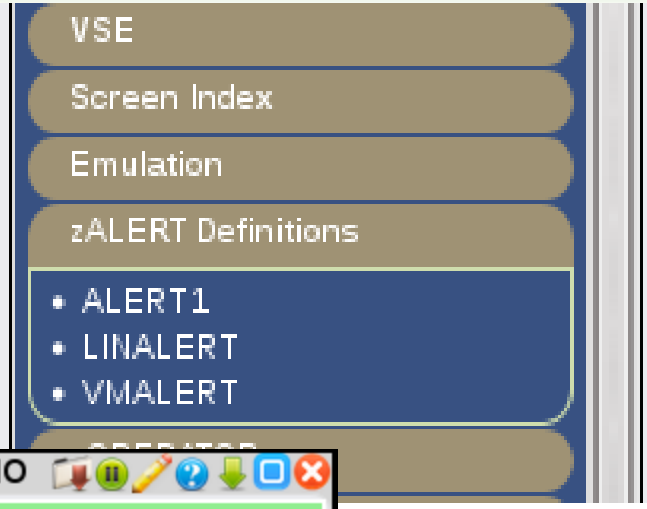
```
vmlink .dir sfszvps:zmon.code
ZALERT RKS2LV
```



VELOCITY SOFTWARE

Viewing alerts

- **zView**
  - Select 'zAlert Definitions'
  - Select alert file to display



ALERT1 - Exceptions Analysis Alerts - 15/06/15 at 06:48 - DEMO

Code	Alert Description
CHEK	Spool Utilization is 73%
JHPU	JVM 'Server1' on lxora12 Heap Utilization 33.9%
LPCP	LPAR VSIVM4 is at 54%
ORPG	DB db01 on lxora12 PGA Utilization 72%
ORPG	DB db01 on s11s2ora PGA Utilization 62%
ORSW	DB orcl on oracle System IO Waits 1 Time 0.000
PGUT	Page space is 43% used
PRCK	roblx1 proc hogmem not found.
PRCK	sles11x proc top not found.
SFSS	Filespace DXTWRITE in SFSVM4 at 99%
SPOL	Spoolspace is 73% used
VMLP	User BLAKEMC may be looping; CPU 17%, loop count 4417
VMLP	User ROBLX1 may be looping; CPU 20%, loop count 482
VMOC	VM overcommit ratio is 3.2
XACP	Processor utilization at 52.8%

Viewing alerts

- **CGI**
  - Copy ZALERT.CGI from ZALERT top level directory to ZVWS.ROOT

http://<vm-host>/zalert.cgi
http://<vm-host>/zalert.cgi?file=<alertfile>

VELOCITY SOFTWARE

Exception Analysis Alerts

Velocity Software - VSIVM4

Alert File:ALERT1 15 Jun 2015
System:VSIVM4 06:52:43

Type	Description
CHEK	Spool Utilization is 73%
JHPU	JVM 'Server1' on lxora12 Heap Utilization 34.6%
JHPU	JVM 'Java Nr 7' on roblx1 Heap Utilization 39.0%
LPCP	LPAR VSIVM4 is at 55%
ORPG	DB db01 on lxora12 PGA Utilization 72%
ORPG	DB db01 on s11s2ora PGA Utilization 62%
ORSW	DB db01 on lxora12 System IO Waits 1 Time 0.000
PGUT	Page space is 43% used
PRCK	roblx1 proc hogmem not found.
PRCK	sles11x proc top not found.
SFSS	Filespace DXTWRITE in SFSVM4 at 99%
SPOL	Spoolspace is 73% used
TIDL	Test Idle for BLAKEMC is 8
VMLP	User BLAKEMC may be looping; CPU 17%, loop count 4421
VMLP	User ROBLX1 may be looping; CPU 20%, loop count 486
VMOC	VM overcommit ratio is 3.2
XACP	Processor utilization at 53.8%

Alert samples

- **Alert samples are shipped with the ZALERT package**
  - ALERT1 MONALERT is combined from the four previously provided sample files
  - Older sample files are shipped with the filetype MONSAMP
    - VMALERT, LINALERT, HEALTH and HEALTH2
  - Samples ship with alerts to check various conditions that can potentially occur
    - CPU/Spool/Page Utilization, I/O Rate, Paging Rate
    - Network node CPU utilization, I/O Rate, disk utilization, swap rate and utilization
- **Additional samples available on our web site**

Defining your own alerts

- **Coding an alert requires the use of data fields maintained by zVPS**
- **Data is extracted from the monitor**
- **Analyzed to determine if it exceeds a threshold**
- **For values greater than threshold**
  - Message issued
  - Optional action is taken
- **Alerts generally use the following statements**
  - EXTRACT
  - VAR
  - ALERT
  - LEVEL
  - TEXT

Defining your own alerts

- **Alert for CPU Utilization**

```
Extract
Parms CPU TOTAL

var  cpu_serial | 6    | serial
var  util       | 5 1  | sytprp.cpuutil

alert util xacp
level 00  green
level 20  yellow
level 40  pink
level 80  red
text Processor utilization at &util%
```

'Extract' is the beginning of an alert definition or set of alert definitions

Defining your own alerts

- **Alert for CPU Utilization**

```
Extract
Parms CPU TOTAL

var  cpu_serial | 6     | serial
var  util       | 5 1   | sytprp.cpuutil

alert util xacp
level 00  green
level 20  yellow
level 40  pink
level 80  red
text Processor utilization at &util%
```

PARMS determines the type of data to extract

Defining your own alerts

- **Alert for CPU Utilization**

```
Extract
Parms CPU TOTAL

var  cpu_serial | 6   | serial
var  util       | 5 1 | sytprp.cpuutil

alert util xacp
level 00  green
level 20  yellow
level 40  pink
level 80  red
text Processor utilization
```

Fields to extract -
names are described in the PDR
(Performance Data Reference)

Can be a single field or multiple fields involved in simple to complex math operations.

Defining your own alerts

- **Alert for CPU Utilization**

```
Extract
Parms CPU TOTAL

var  cpu_serial | 6   | serial
var  util       | 5 1 | sytprp.cpuutil

alert util xacp
level 00  green
level 20  yellow
level 40  pink
level 80  red
text Processor utilization at &util%
```

Variables defined for use in the following alerts

Size of each variable with optional decimal precision

Defining your own alerts

- **Alert for CPU Utilization**

```
Extract
Parms

var  cpu_serial | 6    | serial
var  util       | 5 1  | sytprp.cpuutil

alert util xacp
level 00  green
level 20  yellow
level 40  pink
level 80  re
text Process
```

ALERT statement defines a specific alert

Four character code used when displaying alerts

Each alert requires a previously defined variable

Defining your own alerts

- **Alert for CPU Utilization**

```
Extract
Parms CPU TOTAL

var                          serial
var   util          | 5 1 | sytprp.cpuutil

alert util xacp
level 00  green
level 20  yellow
level 40  pink
level 80  red
text Proces                          %
```

LEVEL statement controls the threshold values

Color of the alert text when this level is exceeded

Values tested against the alert variable

Defining your own alerts

- **Alert for CPU Utilization**

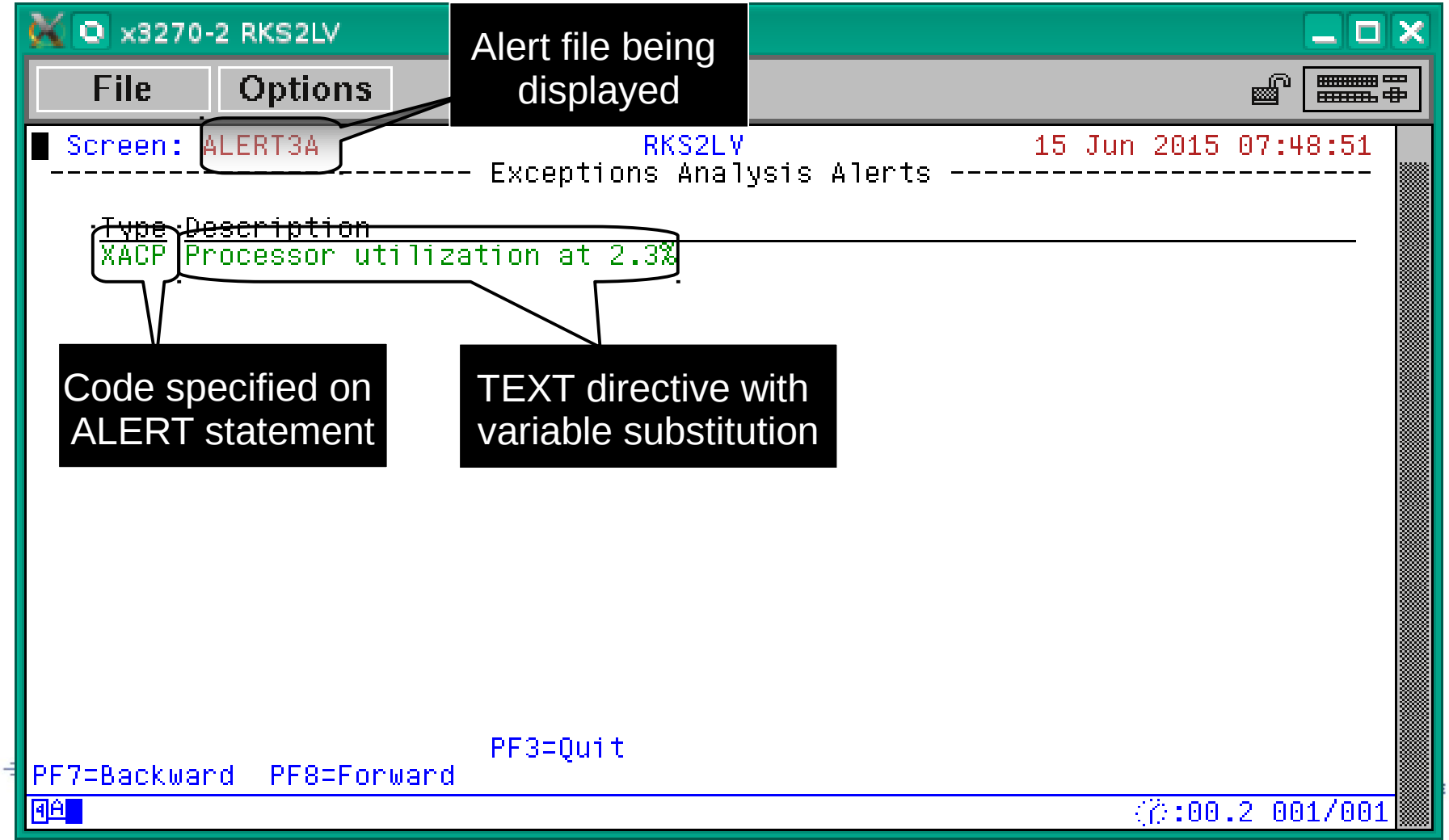
```
Extract
Parms CPU TOTAL

var  cpu_serial | 6   | serial
var  util       | 5 1 | sytprp.cpuutil

alert util xacp
level 00  green
level 20  yellow
level 40  pink
level 80  red
text Processor utilization at &util%
```

Message displayed on 3270 and zView alert screens

Alert variable substitution

Alert result

- **The 3270 screen based on the alert definition**



Alert rules

- **Adjust the number and value of levels based on local requirements**
  - At least one LEVEL statement is necessary
  - LEVEL statements are evaluated bottom to top
- **Standard 3270 colors are allowed**
  - Turquoise, Blue, Red, Yellow, Green, Pink, White
  - If no color is specified, the default is Green
  - Color modifiers are allowed
    - **REV**video – reverse video
    - **BLI**nk – blink the entire text
    - **UNDERLINE** – underline the entire text

Defining your own alert - LPAR

- **Alert for LPAR Utilization**

```
Extract
Parms LPAR *
Criteria sytcup.lcupname <> 'Totals:'
var  lparname    | 8   | sytcup.lcupname
var  lparutil    | 3 0 | sytcup.pctcpu

alert lparutil lpcp
level 90  red
level 95  red rev
text LPAR utilization of &lparname is &lparutil%
```

Defining your own alert - LPAR

- **Alert for LPAR Utilization**

```
Extract
Parms LPAR *
Criteria sytcup.lcupname <> 'Totals:'
var  lparname    | 8   | sytcup.lcupname
var  lparutil    | 3 0 | sytcup.pctcpu

alert lparutil lpcp
level 90  red
level 95  red rev
text LPAR utilization of &lparname is &lparutil%
```

Informs the extract to pull data for all LPARs

Additional data selection passed to the extract as WHILE criteria

Defining your own alert - LPAR

- **Alert for LPAR Utilization**

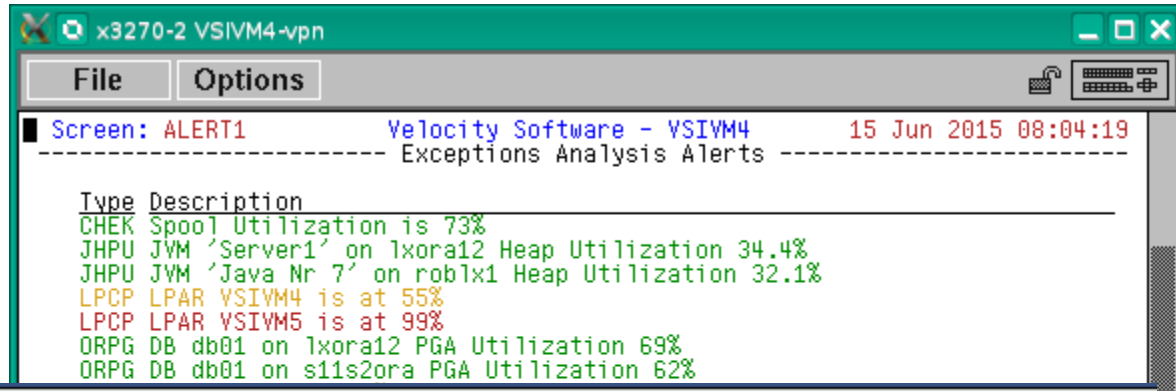
```
Extract
Parms LPAR *
Criteria sytcup.lcupname <> 'Totals:'
var  lparname    | 8    | sytcup.lcupname
var  lparutil    | 3 0 | sytcup.pctcpu

alert lparutil lpcp
level 90  red
level 95  red rev
text LPAR utilization of &lparname is &lparutil%
```

Text will be in reverse video (black text, red background)

Defining your own alert - LPAR

- **Alert for LPAR Utilization display**

3270 >



zView v

ALERT1

ALERT1 - Exceptions Analysis Alerts - 15/06/15 at 08:05 - DEMO

Code	Alert Description
CHEK	Spool Utilization is 73%
JHPU	JVM 'Server1' on lxora12 Heap Utilization 35.2%
JHPU	JVM 'Java Nr 7' on roblx1 Heap Utilization 34.9%
LPCP	LPAR VSIVM4 is at 58%
LPCP	LPAR VSIVM5 is at 100%
ORPG	DB db01 on lxora12 PGA Utilization 69%
ORPG	DB db01 on s11s2ora PGA Utilization 62%
PGUT	Page space is 43% used
PRCK	roblx1 proc hogmem not found.
PRCK	sles11x proc top not found.
SFSS	Filespace DXTWRITE in SFSVM4 at 99%
SPOL	Spoolspace is 73% used
TIDL	Test Idle for SFSZVPS4 is 13
VMLP	User BLAKEMC may be looping; CPU 17%, loop count 4494
VMLP	User ROBLX1 may be looping; CPU 20%, loop count 559
VMOC	VM overcommit ratio is 3.2
XACP	Processor utilization at 56.9%

Defining your own alert – Complex operations

- **Numerous fields can be combined using math operations**
  - Statements can be continued with a dash

```
extract
parms user *
criteria userdata.userid <> 'System:' & useact.vmdttime > 0
var    userid      | 8   | userdata.userid
var    cpuutil     | 3 1 | useact.vmdttime * 100 / RUNTIME
var    io_rate     | 6 0 | (useact.vmdvdsct+useact.vmdvosct-
                          +useact.vmdvcsct+useact.vmdvusct-
                          +useact.vmdvtsct)/runtime
var    page_rate   | 6   | (useact.vmdctpgr+useact.vmdctpgw)/RUNTIME
var    exp_store   | 8   | useact.vmdctxrd+useact.vmdctxwt
var    userprt     | 8   | useact.vmdctpgr
var    vmdttime    | 5 2 | useact.vmdttime
```

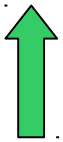
Defining your own alert – Second vdisk usage

- **Using two swap disks with different priority**
  - Second disk larger than the first
  - First disk fills, Linux uses the second disk
  - Alert when second disk is used

ESAVDSK

VDISK Analysis

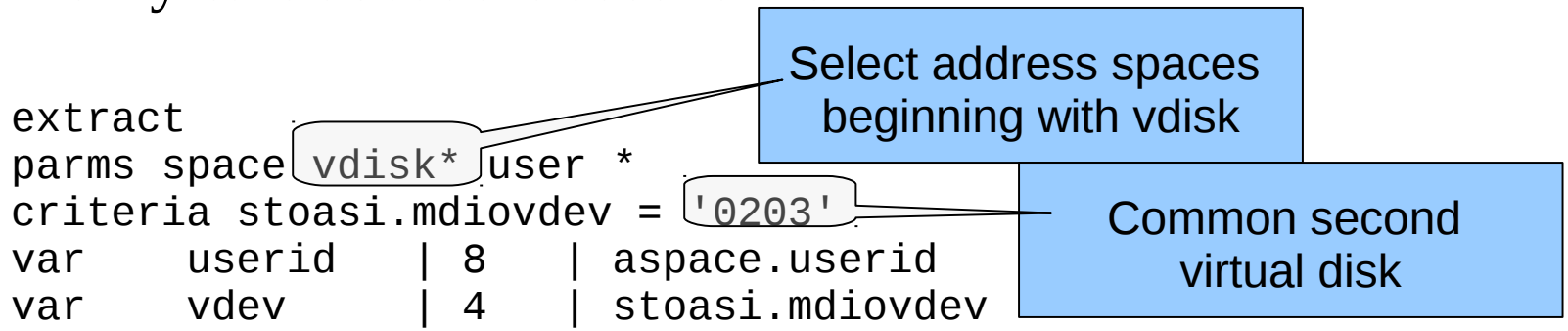
Time	Owner	Space Name	<--Size---> AddSpc Pages	VDSK Blks	<--pages--> Resi-dent	Lock-ed	Prv or Shr	VIO rate /min	Usr Lks	<AddSpce> Cre-ates	Del-etes	Sto-len	<----------pages/second----------> <--DASD---> Read	Write	Expanded PGIN	PGOUT	Storage Migr	DASD Page Slots	X-Store Blks
17:20:00	LINUX001	VDISK$LINUX001$0202$0048	16000	128K	0	0	Shr	0	1	0	0	0	0	0	0	0	0	12230	0
17:20:00	LINUX001	VDISK$LINUX001$0203$0049	16000	128K	0	0	Shr	0	1	0	0	0	0	0	0	0	0	160	0
17:20:00	LINUX002	VDISK$LINUX002$0202$002F	16000	128K	0	0	Shr	0	1	0	0	0	0	0	0	0	0	168	0
17:20:00	LINUX002	VDISK$LINUX002$0203$0030	16000	128K	0	0	Shr	0	1	0	0	0	0	0	0	0	0	160	0
17:20:00	ZPR001	VDISK$ZPR001$$$0192$0043	208	1664	0	0	Shr	0	1	0	0	0	0	0	0	0	0	12	0
17:20:00	ZPR002	VDISK$ZPR002$$$0192$0044	208	1664	0	0	Shr	0	1	0	0	0	0	0	0	0	0	3	0
17:20:00	ZPR003	VDISK$ZPR003$$$0192$0045	208	1664	0	0	Shr	0	1	0	0	0	0	0	0	0	0	3	0



Vdisk activity indicator

Defining your own alert – Second vdisk usage

- **Create an alert to show Vdisk activity**
  - Only care about the second disk




```
extract
parms space vdisk* user *
criteria stoasi.mdiovdev = '0203'
var    userid    | 8    | aspace.userid
var    vdev      | 4    | stoasi.mdiovdev
var    io_rate   | 6    | stoasi.qdiiocnt

alert io_rate lsvd
level 0 red
text Node &userid is using the second virtual disk
```

Notifications

- **A notification is a message sent to interested parties of an alert condition**
- **Sent in one or more of the following forms**
  - CP MSG/MSGNOH
  - Email
  - Text page (via email)
  - SNMP Trap

Notifications

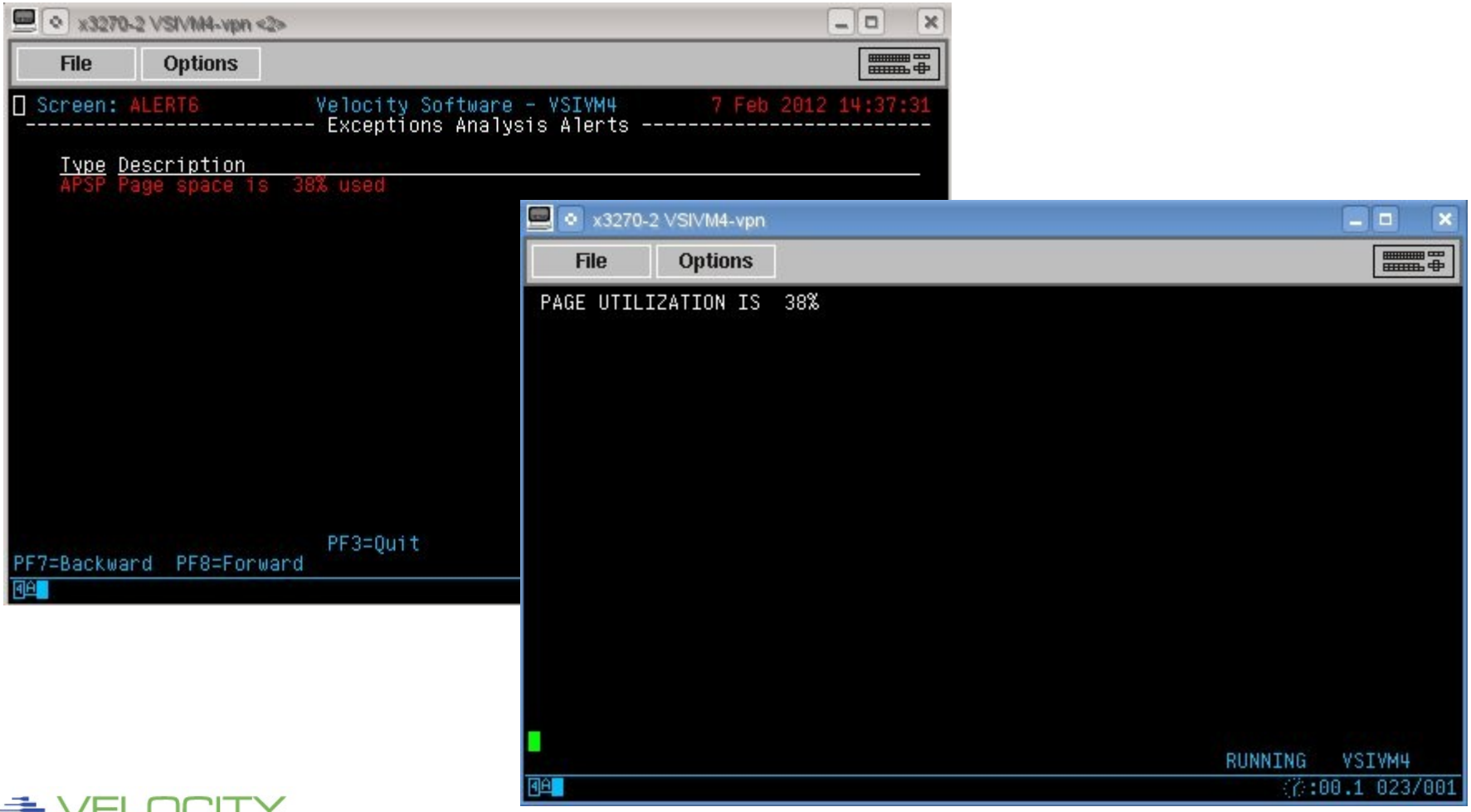
- **At it's simplest, a notification can take the form of a message to a CMS user**

```
Extract
var  page_use   | 3 0 | (sytasg.calslti1*100)/sytasg.calslta1

alert page_use apsp
level 30  yellow
level 35  red   ACTION CP MSGNOH ZVPS Page utilization is &page_use%
level 50  red rev
text Page space is &page_use% used
```

ACTION keyword on the LEVEL statement allows targeted messaging for a specific threshold

Notifications

- **Results of ACTION**



- **A REXX EXEC can be invoked to send an email**

```
Extract
var  spool_use  | 3 0 | (sytasg.calslti2*100)/sytasg.calslta2

alert spool_use spol
level 50 yellow
level 75 red
level 85 red rev ACTION CMS EMAIL RKS2LV SPOL &spool_use
text Spool area utilization &spool_use%
```

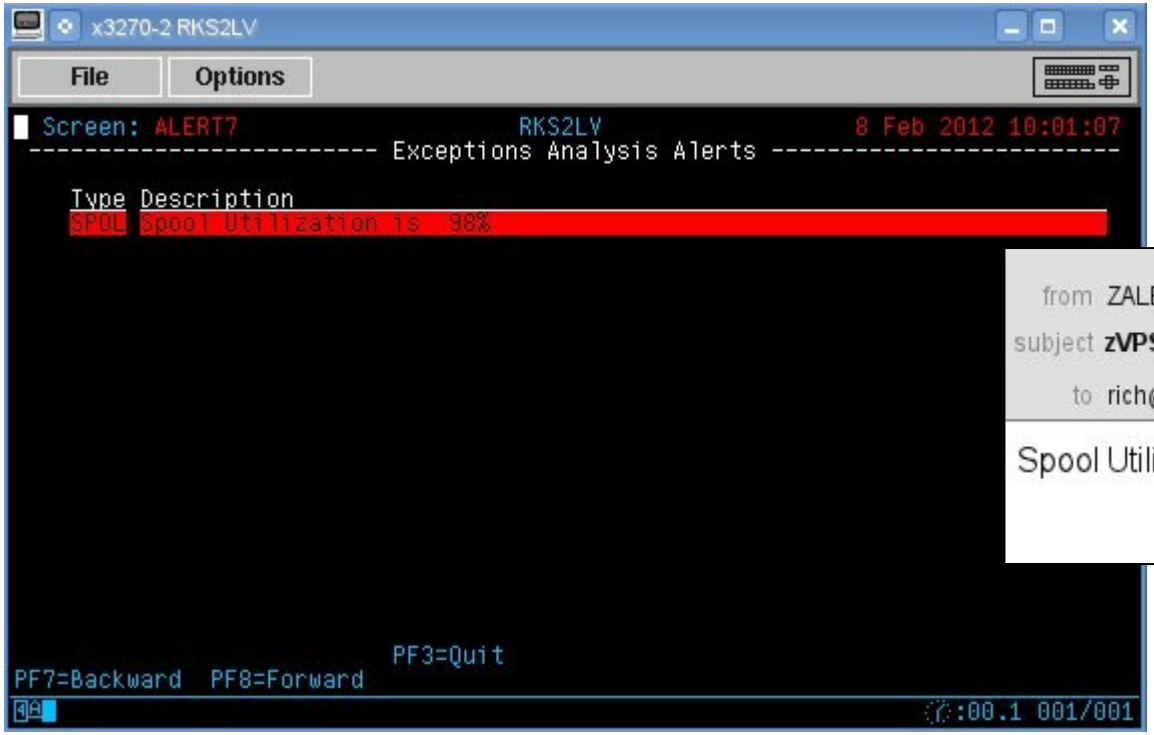
Executes a CMS command - EMAIL EXEC

- **A REXX EXEC can be invoked to send an email**

```
/* EMAIL: Sample EXEC to provide notify function */
parse arg node code value

Select
  When code='SPOL' then
    msg='Spool Utilization for 'node value'%'
  When code='XACP' then
    msg='CPU Utilization for 'node value'%'
  Otherwise
    exit
  End

Queue 'input Subject: zVPS 'code' alert'
Queue 'input 'msg
Queue 'COMMAND CMS SENDFILE ( NOTE'
'EXEC NOTE rich@velocitysoftware.com (NONOTEBOOK'
exit
```

Notifications

- **A REXX EXEC can be invoked to send an email**



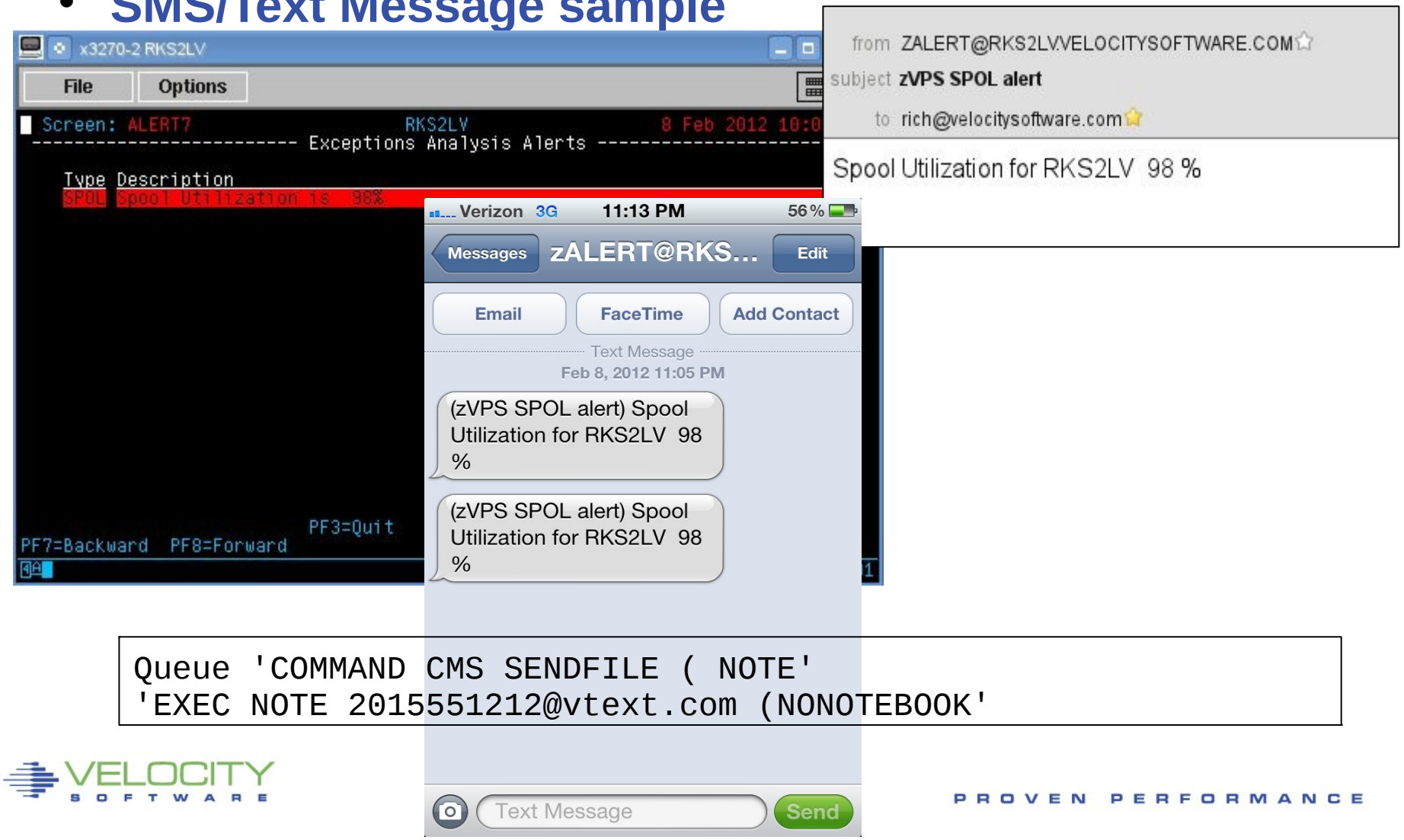
- **Cell phone text alerts**
  - Produced via an email message
  - Each carrier varies in their approach
  - List of Email to SMS gateways for most providers

http://www.tech-faq.com/how-to-send-text-messages-free.html

AT&T (formerly Cingular)	[10-digit-number]@txt.att.net
Sprint	[10-digit-number]@messaging.sprintpcs.com
T-Mobile	[10-digit-number]@tmomail.net
US Cellular	[10-digit-number]@email.uscc.net
Verizon	[10-digit-number]@vtext.com

Notifications

- **SMS/Text Message sample**




```
Queue 'COMMAND CMS SENDFILE ( NOTE'
'EXEC NOTE 2015551212@vtext.com (NONOTEBOOK'
```

- **SNMP Trap configuration**
  - Create/Modify SNMP TRAPDEST on the CONFIG disk

```
* following is default 1.3.6.1.4.1.15601
   192.168.5.64 velocity 2B06010401F971 ;
```

  - Use the ALERT/TRAP option on the LEVEL command

```
Extract
var  spool_use  | 3 0 | (sytasg.calslti2*100)/sytasg.calslta2

alert spool_use spol
level 50 yellow
level 75 red
level 85 red rev TRAP VM (RKS2LV) Spool Utilization is &spool_use%
text Spool area utilization &spool_use%
```

- **SNMP Trap configuration**
  - Enterprise management consoles
    - NetCool, HP OpenView, CA-Unicenter TNG
  - Trap string can be generated in any required format for proper handling
  - Using a special code as the first token of the alert, trap payload is set specifically for management consoles

```
Extract
var  spool_use  | 3 0 | (sytasg.calslti2*100)/sytasg.calslta2

alert spool_use spol
level 50 yellow
level 75 red
level 85 red rev TRAP SPL002 VM (RKS2LV) Spool Utilization is &spool_use%
text Spool area utilization &spool_use%
```

Notifications

- **SNMP Trap result**

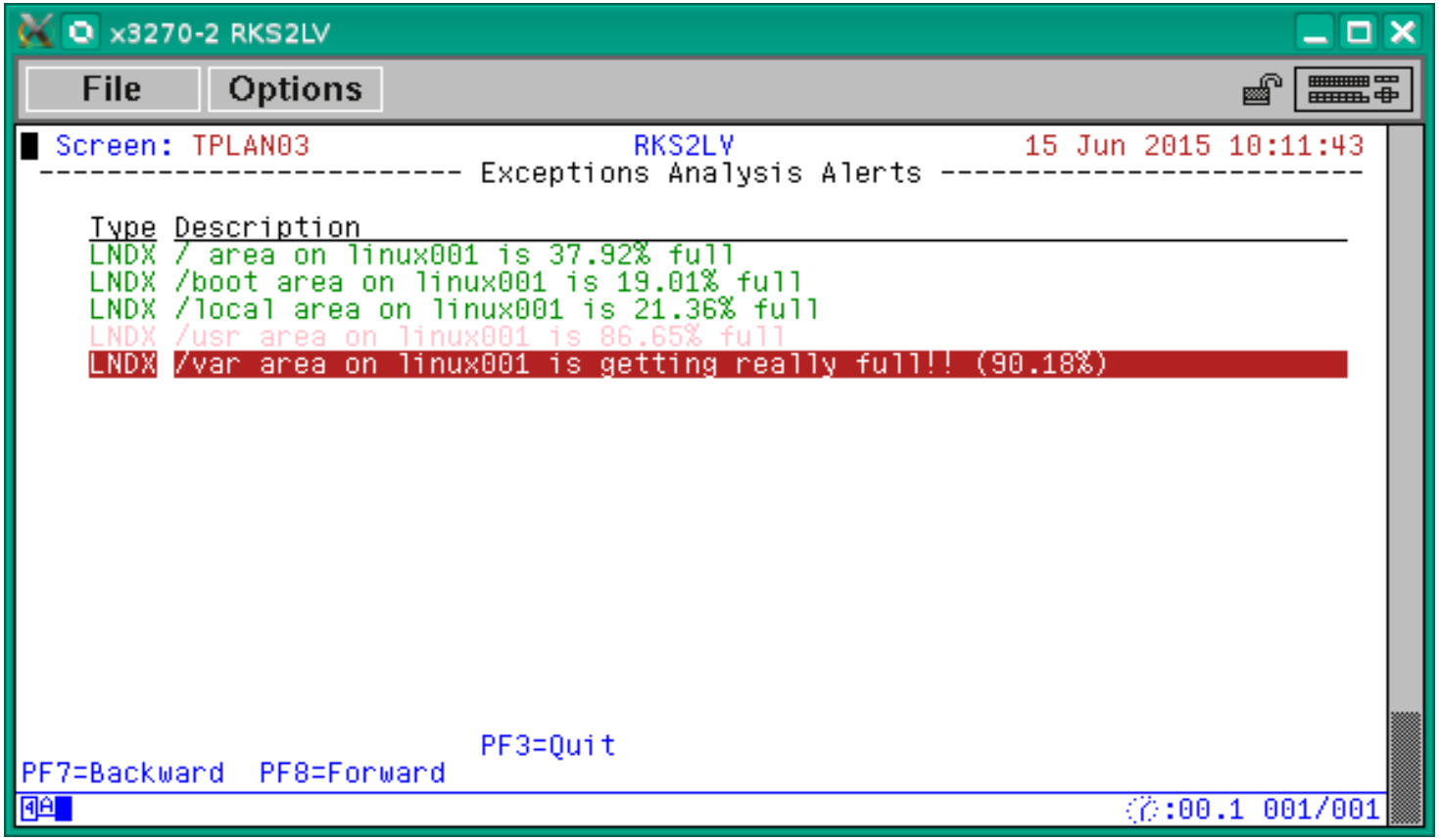
```
Received 97 bytes from UDP: [192.168.5.48]:1114
0000: 30 5F 02 01  00 04 08 76  65 6C 6F 63  69 74 79 A4    0_.....velocity.
0016: 50 06 07 2B  06 01 04 01  F9 71 40 04  C0 A8 05 30    P..+.....q@....0
0032: 02 01 06 02  01 00 43 04  00 00 00 0C  30 33 30 31    ......C.....0301
0048: 06 07 2B 06  01 04 01 F9  71 04 26 56  4D 20 28 52    ..+.....q.&VM (R
0064: 4B 53 32 4C  56 29 20 53  50 4F 4F 4C  20 55 54 49    KS2LV) SPOOL UTI
0080: 4C 49 5A 41  54 49 4F 4E  20 49 53 20  20 39 36 25    LIZATION IS  96%
0096: 20

2012-02-16 13:42:58 192.168.5.48(via UDP: [192.168.5.48]:1114) TRAP, SNMP v1, community
velocity
        VELOCITY-MIB::velocity Enterprise Specific Trap (0) Uptime: 0:00:00.12
        VELOCITY-MIB::velocity = STRING: "VM (RKS2LV) SPOOL UTILIZATION IS  96%"
```

Level Text

- **The LTEXT directive allows alternate text to be displayed for different alert levels**
  - LTEXT is tied to the level immediately preceding

```
extract
parms node *
criteria hstmem.used > 0
var    node    | 8   | tcpip.node
var    memused | 6 2 | (hstmem.used/hstmem.size)*100
var    desc    | 16  | hstmem.desc
function diskpct | 6 2 | &node &memused &desc

alert diskpct lndx
level  5  green
level 50  yellow
level 80  pink
level 90  red rev
ltext &desc area on &node is getting really full!! (&diskpct%)
text &desc area on &node is &diskpct% full
```

Level Text


x3270-2 RKS2LV
File
Options
Screen: TPLAN03 RKS2LV 15 Jun 2015 10:11:43
Exceptions Analysis Alerts
Type Description
LNDX / area on linux001 is 37.92% full
LNDX /boot area on linux001 is 19.01% full
LNDX /local area on linux001 is 21.36% full
LNDX /usr area on linux001 is 86.65% full
LNDX /var area on linux001 is getting really full!! (90.18%)
PF3=Quit
PF7=Backward PF8=Forward
:00.1 001/001

Alert Options

- **Options add additional function at the alert level**
  - Priority
  - Log
  - Count
  - Separated on alert directive with a vertical bar

```
ALERT DISKPCT LNDX | <options>
```

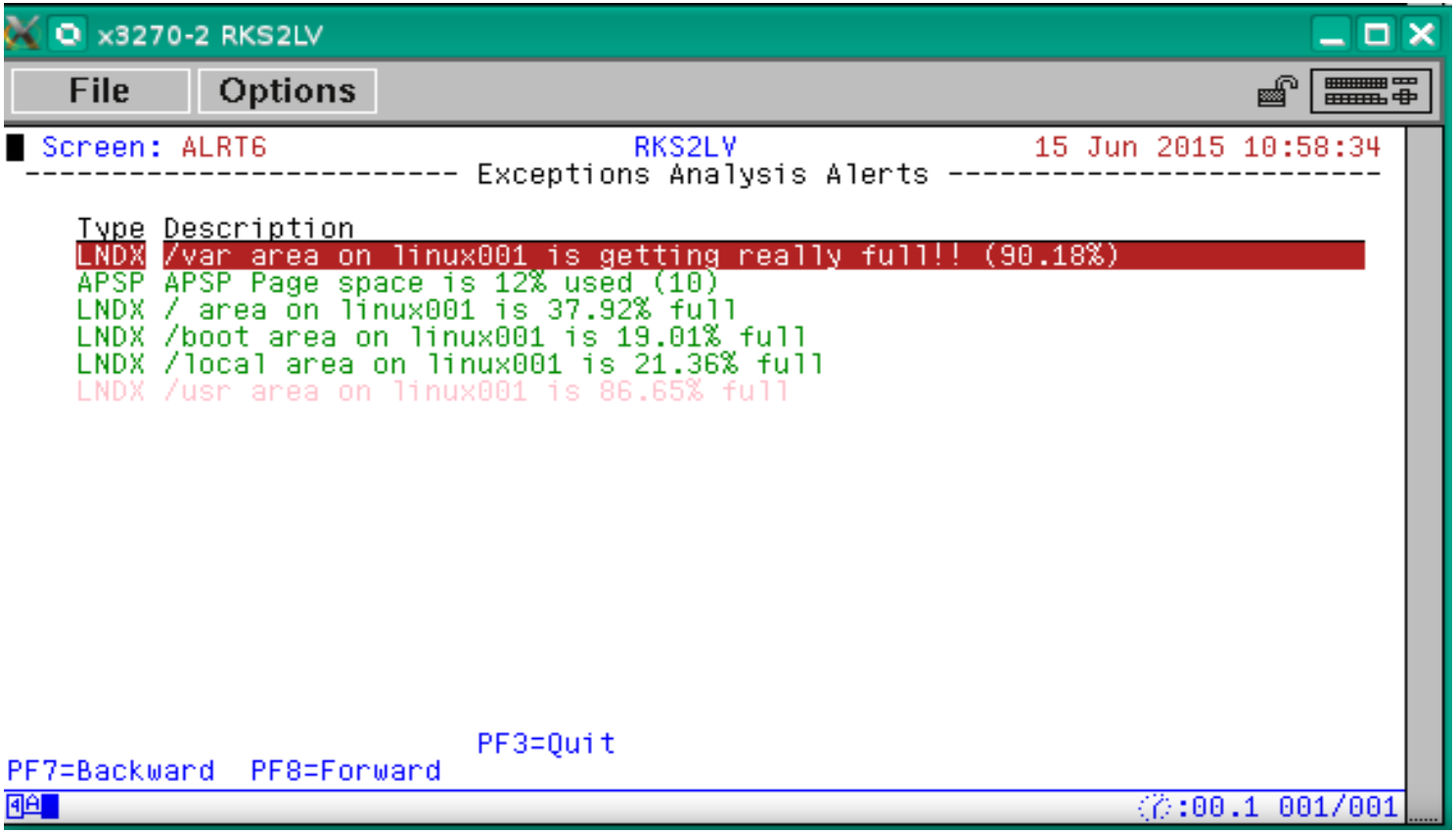
- **Pri – Prioritize the display of alert messages**
- **Log - Writes alert text displays to a file**
- **Count - Tally the number of times an alert appears**

Priority

- **Priority alters the display sequence of an alert or level**
  - Priority is a numeric value 1-9, default is 3

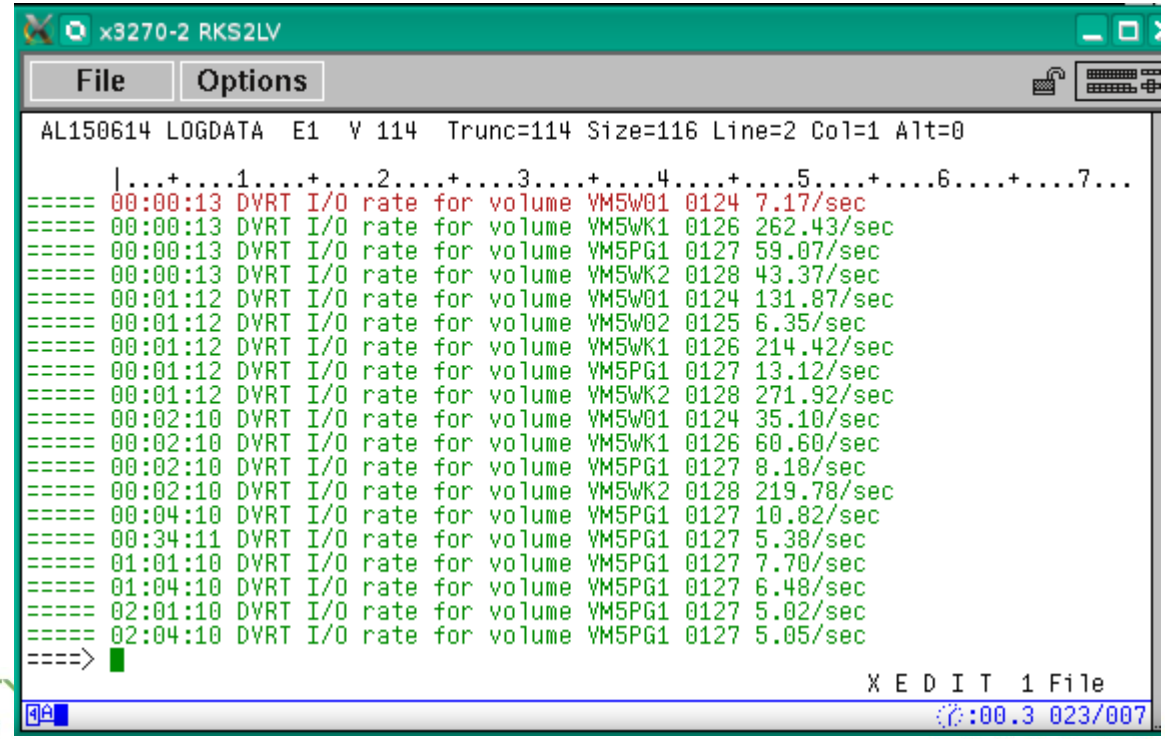
```
extract
parms node *
criteria hstmem.used > 0
var    node    | 8   | tcpip.node
var    memused | 6 2 | (hstmem.used/hstmem.size)*100
var    desc    | 16  | hstmem.desc
function diskpct  | 6 2 | &node &memused &desc

alert diskpct lndx | pri 2
level  5  green
level 50  yellow
level 80  pink
level 90  red rev pri 6
ltext &desc area on &node is getting really full!! (&diskpct%)
text &desc area on &node is &diskpct% full
```

Priority


x3270-2 RKS2LV
File
Options
Screen: ALRT6 RKS2LV 15 Jun 2015 10:58:34
Exceptions Analysis Alerts
Type Description
LNDX /var area on linux001 is getting really full!! (90.18%)
APSP APSP Page space is 12% used (10)
LNDX / area on linux001 is 37.92% full
LNDX /boot area on linux001 is 19.01% full
LNDX /local area on linux001 is 21.36% full
LNDX /usr area on linux001 is 86.65% full
PF3=Quit
PF7=Backward PF8=Forward
:00.1 001/001

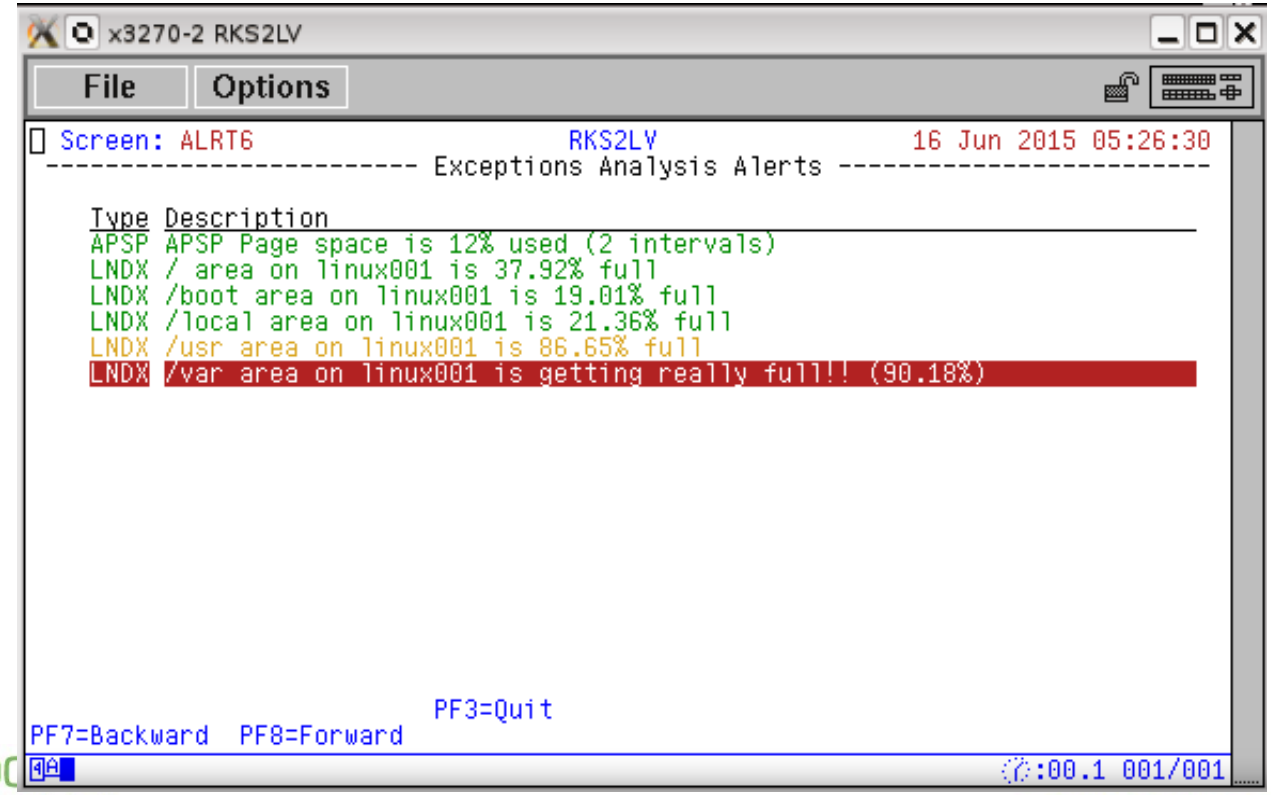
Log

```
alert io_rate dvrt | log
level  5  green
level 10  blue
level 20  turquoise
level 30  pink
level 40  red rev
text I/O rate for volume &volser &rdev &io_rate/sec
```



Count

```
alert page_use apsp | count
level 10  green
level 30  yellow
level 50  red
text &code Page space is &page_use% used (&tcount intervals)
```



Alert Options

- **Log writes out the currently displayable text message**
  - TEXT or LTEXT
- **Count takes an optional key value**
  - Used when an alert can return multiple values
  - Eg: user, node, device
  - Specify variable that contains the key value after COUNT keyword

```
alert usercpu vmcp | count &userid
```

- **Multiple options can be specified**

Enable/Disable

- **Disable removes an alert from evaluation (4.2)**
  - Used in a maintenance situation when unwanted alerts or false alerts may be triggered
- **Enable is the opposite of disable**
  - The default and need not be specified
  - Provided for consistency

```
alert page_use apsp
disable
level 10  green
level 30  yellow
level 50  red
text Page space is &page_use% used
```

Limit

- **The LIMIT directive delays an ACTION for the specified number of intervals**

```
extract
var serial       | 6    | system.serial
var spool_use    | 3 0  | (sytasg.calslti2*100)/sytasg.calslta2

alert spool_use spol
limit 5 1 | &serial
level 70  yellow
level 80  red
level 90  red rev ACTION CP MSG ZVPS Spool Util is &spool_use%
text Spool Utilization is &spool_use%
```

Limit

- **The LIMIT directive delays an ACTION for the specified number of intervals**

```
extract
var serial       | 6   | system.serial
var spool_use    | 3 0 | (sytasg.calslti2*100)/sytasg.calslta2

alert spool use spol
limit 5 1 | &serial
level 70  yellow
level 80  red
level 90  red rev ACTION CP MSG ZVPS Spool Util is &spool_use%
text Spool Utilization is &spool_use%
```

Key field

After the delay, number of intervals TO execute ACTION (default is 1)

Number of intervals to delay executing ACTION

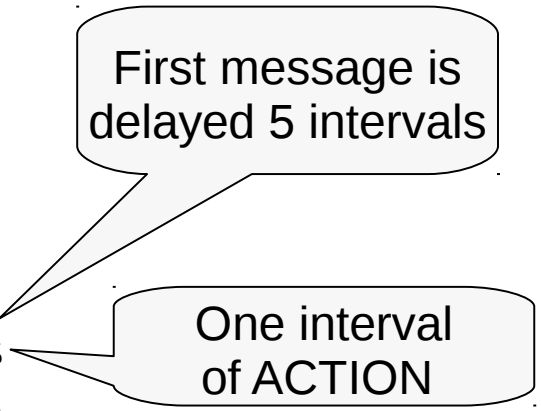
Limit

- **This LIMIT directive:**

```
Limit 5 1 | &serial
```

- **Will delay ACTION for 5 intervals**
- **Execute ACTION for 1 intervals**
- **Repeat**
- **For example, when started at 11:52**

```
11:58:29  * MSG FROM ZALERT  : 10 Feb 2012 11:58 SPOOL UTIL IS  95%
12:04:30  * MSG FROM ZALERT  : 10 Feb 2012 12:04 SPOOL UTIL IS  95%
12:10:31  * MSG FROM ZALERT  : 10 Feb 2012 12:10 SPOOL UTIL IS  95%
```



- **LIMIT initial action setting**

```
extract
var serial        | 6    | system.serial
var spool_use     | 3 0  | (sytasg.calslti2*100)/sytasg.calslta2

alert spool_use spol
limit 5 1 1 | &serial
level 70  yellow
level 80  red
level 90  red rev ACTION CP MSG ZVPS Spool Util is &spool_use%
text Spool Utilization is &spool_use%
```

Number of intervals the action is taken before the delay

- **LIMIT escalation**

```
extract
var serial       | 6    | system.serial
var spool_use    | 3 0  | (sytasg.calslti2*100)/sytasg.calslta2

alert spool_use spol
limit 10:5 1 | &serial
level 70  yellow
level 80  red
level 90  red rev ACTION CP MSG ZVPS Spool Util is &spool_use%
text Spool Utilization is &spool_use%
```

Delay 10 intervals, then 5

Time

- **A time based alert defines one or more periods of the day that an alert is active**
- **The display of the alert is discontinued and any actions not executed**

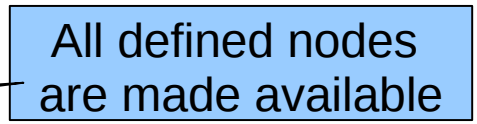
```
extract
var serial       | 6    | system.serial
var spool_use    | 3 0  | (sytasg.calslti2*100)/sytasg.calslta2

alert spool_use spol
time 07:00 to 11:00 | 13:00 to 17:00
level 70  yellow
level 80  red
level 90  red rev ACTION CP MSG ZVPS Spool Util is &spool_use%
text Spool Utilization is &spool_use%
```

Multiple times can be specified

Timeframe for alert to be active

Include/Exclude

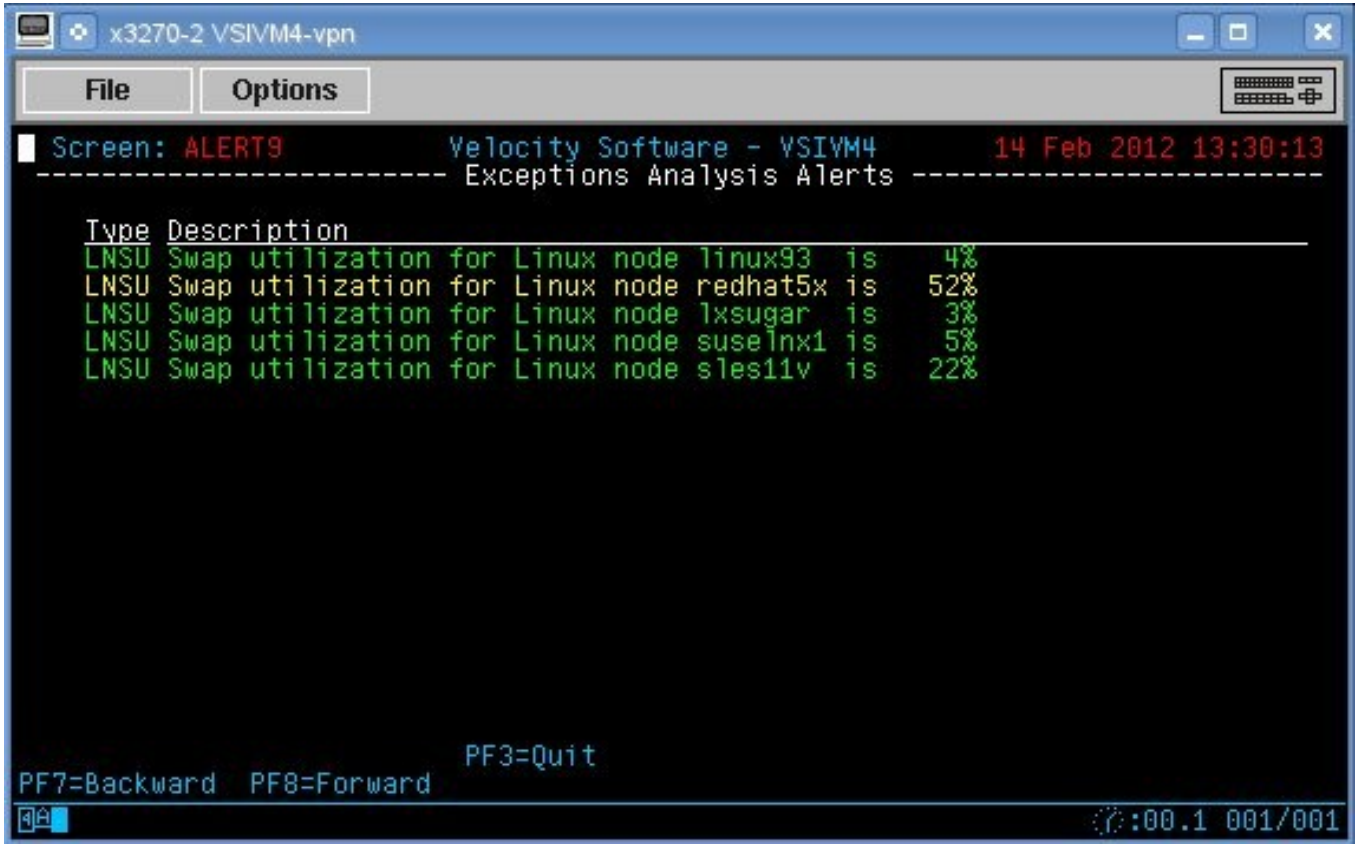
- **User or Node can be specified in an extract**
- **A subset can be selected with wildcards**
- **Given the following alert definition:**




```
extract
parms node *
criteria ucdsys.swappct > 0
var    node     | 8   | tcpip.node
var    swapused | 4 0 | ucdsys.swappct

alert swapused lnsu
level 01  green
level 50  yellow
level 80  pink
level 90  red rev
text Swap utilization for Linux node &node is &swapused%
```

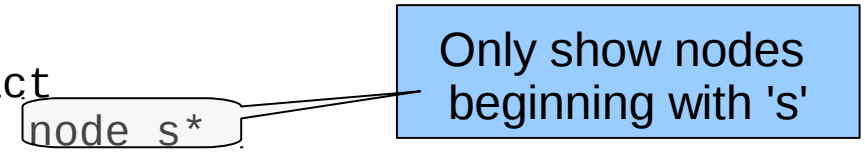
Include/Exclude

- **All nodes with at least 1% swap utilization are displayed**



Include/Exclude

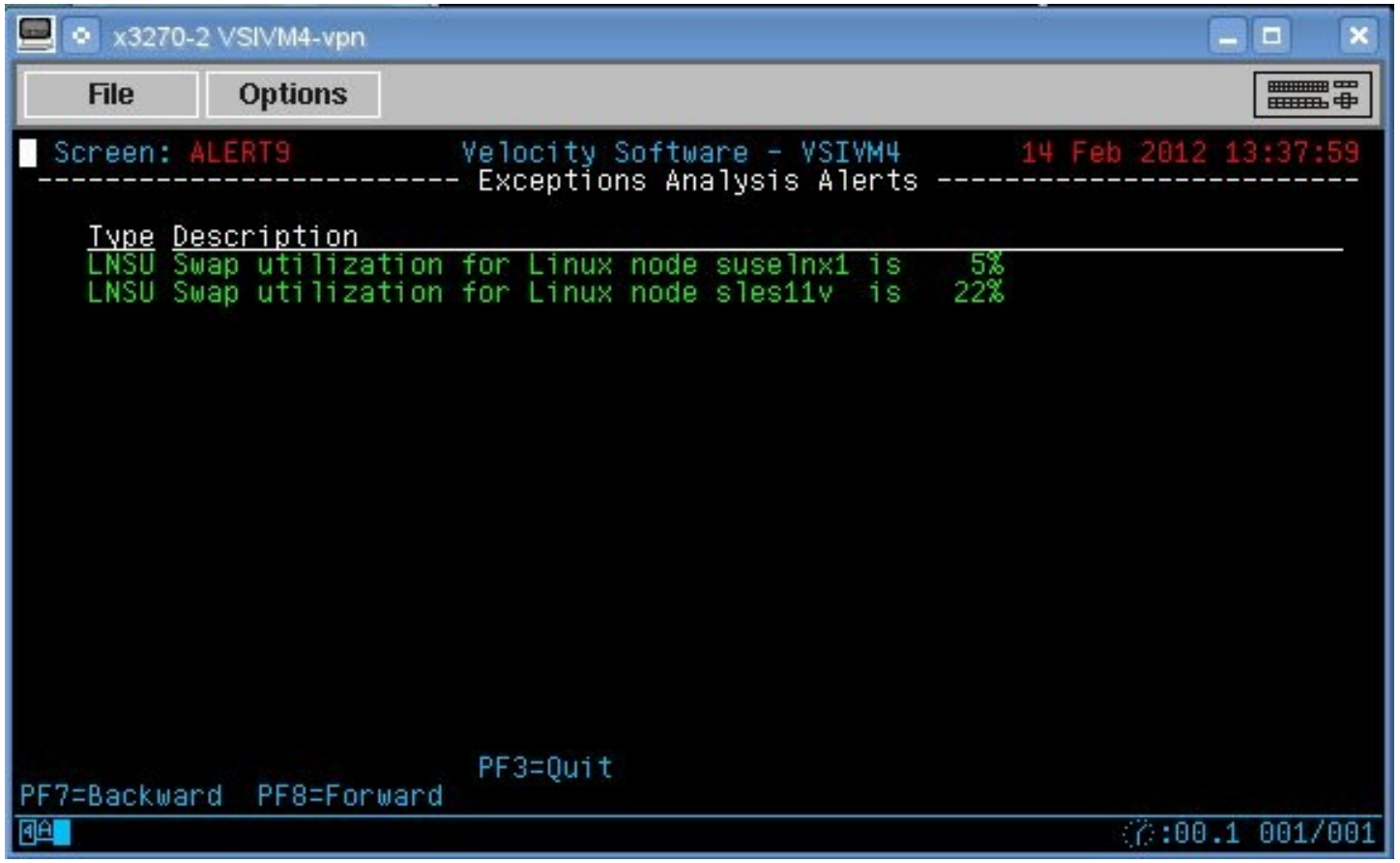
- **The alert can be tailored to show a subset by adjusting the wildcard**




```
extract
parms node s*
criteria ucdsys.swappct > 0
var    node     | 8   | tcpip.node
var    swapused | 4 0 | ucdsys.swappct

alert swapused lnsu
level 01  green
level 50  yellow
level 80  pink
level 90  red rev
text Swap utilization for Linux node &node is &swapused%
```

Include/Exclude

- **The display shows nodes matching the wildcard**



Include/Exclude

- **If an alert is required to show nodes that don't fit into a wildcard**
  - An include or exclude must be used

```
extract
parms node *
criteria ucdsys.swappct > 0
var    node      | 8   | tcpip.node
var    swapused | 4 0 | ucdsys.swappct

alert swapused lnsu
include node sub1
level 01  green
level 50  yellow
level 80  pink
level 90  red rev
text Swap utilization for Linux node &node is &swapused%
```

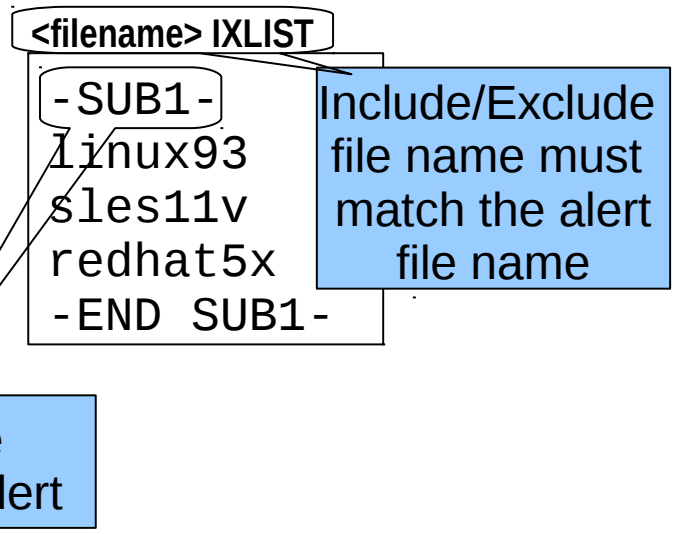
<filename> IXLIST

```
-SUB1-
linux93
sles11v
redhat5x
-END SUB1-
```

Include/Exclude

- **If an alert is required to show nodes that don't fit into a wildcard**
  - An include or exclude must be used

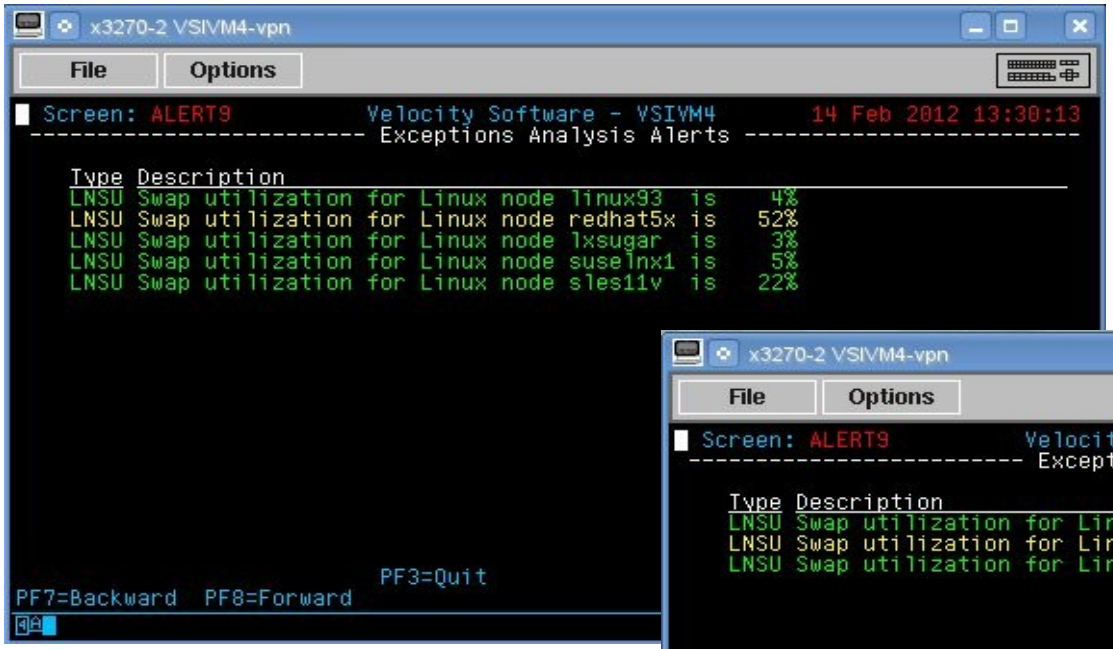
```
extract
parms node *
criteria ucdsys.swappct > 0
var    node     | 8    | tcpip.node
var     swapused | 4 0 | ucdsys.swappct

alert swapused lnsu
include node sub1
level 01  green
level 50  yellow
level 80  pink
level 90  red rev
text Swap utilization for Linux node &node is &swapused%
```



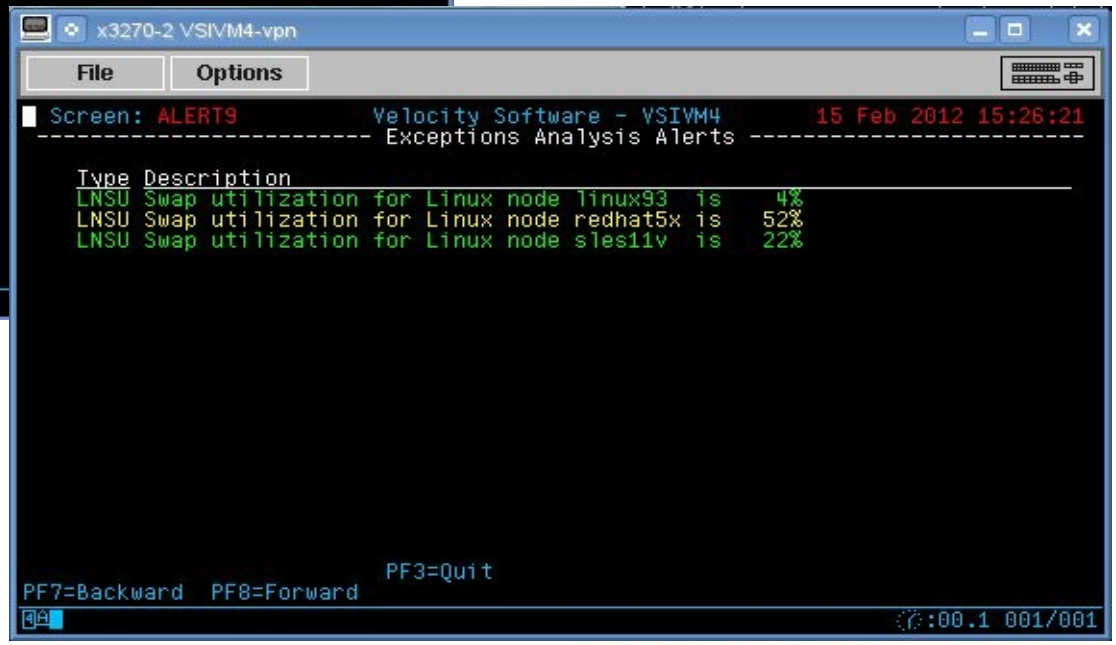
Include/Exclude

- **Results of Include file**

Original display



Include applied



Multiple alerts

- **One extract can supply data for multiple alerts**

```
extract
parms node *
criteria ucdsys.swappct > 0
var    node     | 8   | tcpip.node
var    swaprate | 6 1 | ucdsys.swaprate
var    swapused | 4 0 | ucdsys.swappct

alert swaprate lnsr
level 02  green
level 10  yellow
level 30  pink
level 50  red rev
text Swap i/o rate for Linux node &node is &swaprate

alert swapused lnsu
level 20  green
level 50  yellow
level 80  pink
level 90  red rev
text Swap utilization for Linux node &node is &swapused%
```

External Processing

- **An alert can call an external process**
  - Function
  - Stage
- **Function is a REXX EXEC that processes already extracted data**
  - Called for each record returned from an extract
  - Returns a single value
- **Stage is an EXEC that is called as a pipeline stage**
  - Must have a filetype of REXX
  - Can independently run it's own extract
  - Returns a single value or plugs the result into defined alert variables

External Processing

- **Function is specified in place of 'var'**

```
extract
parms node *
criteria hstmem.used > 0
var    node    | 8   | tcpip.node
var    memused | 6 2 | (hstmem.used/hstmem.size)*100
var    desc    | 16  | hstmem.desc
function diskpct  | 6 0 | &node &memused &desc

alert diskpct lndx
text Filesystem &desc on &node is at &diskpct%
level 20  green
level 50  yellow
level 80  pink
level 90  red rev
```

- **Function is specified in place of 'var'**

```
extract
parms node *
criteria hstmem.used > 0
var     node     | 8   | tcpip.node
var     memused  | 6 2 | (hstmem.used/hstmem.size)*100
var     desc     | 16  | hstmem.desc
function diskpct  | 6 0 | &node &memused &desc

alert diskpct lndx
text Filesystem &desc on &node is at &diskpct%
level 20  green
level 50  yellow
level 80  pink
level 90  red rev
```

Size of returned value

Parameters passed as exec args

Function definition is the exec called and the variable used in the alert

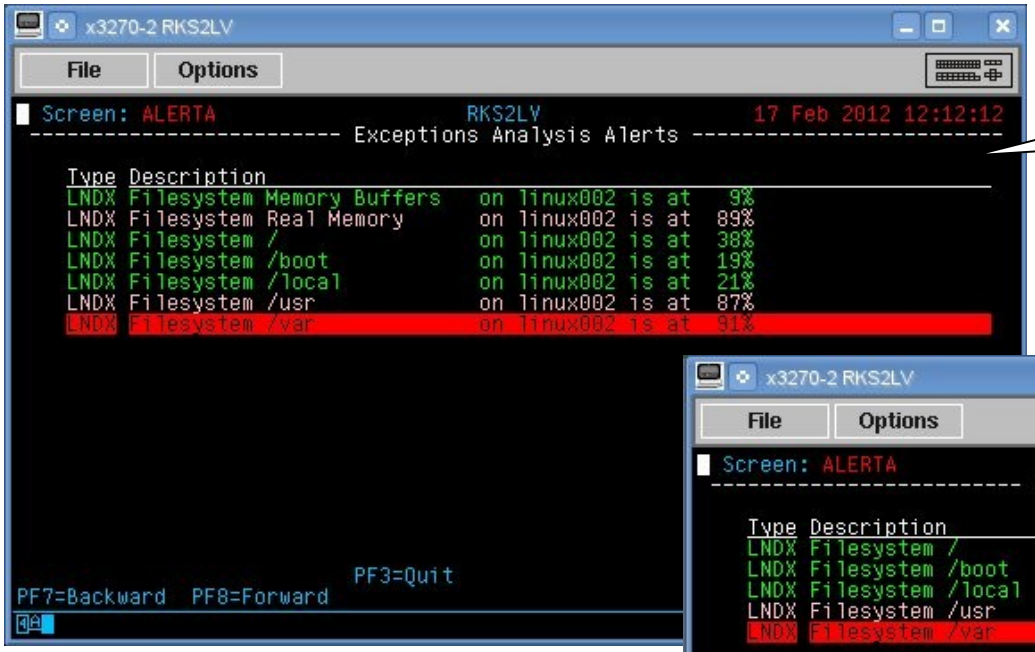
- **REXX exec called from the alert**

```
/* DISKPCT EXEC: Filter out memory or read-only filesystems */
parse arg node pct descr .

firstword = word(descr,1)
rptzero = 'Real Memory Swap Physical Virtual Cached'

if wordpos(descr,rptzero) > 0 then
  pct = 0

if left(descr,6) = '/media' then
  pct = 0

return pct
```

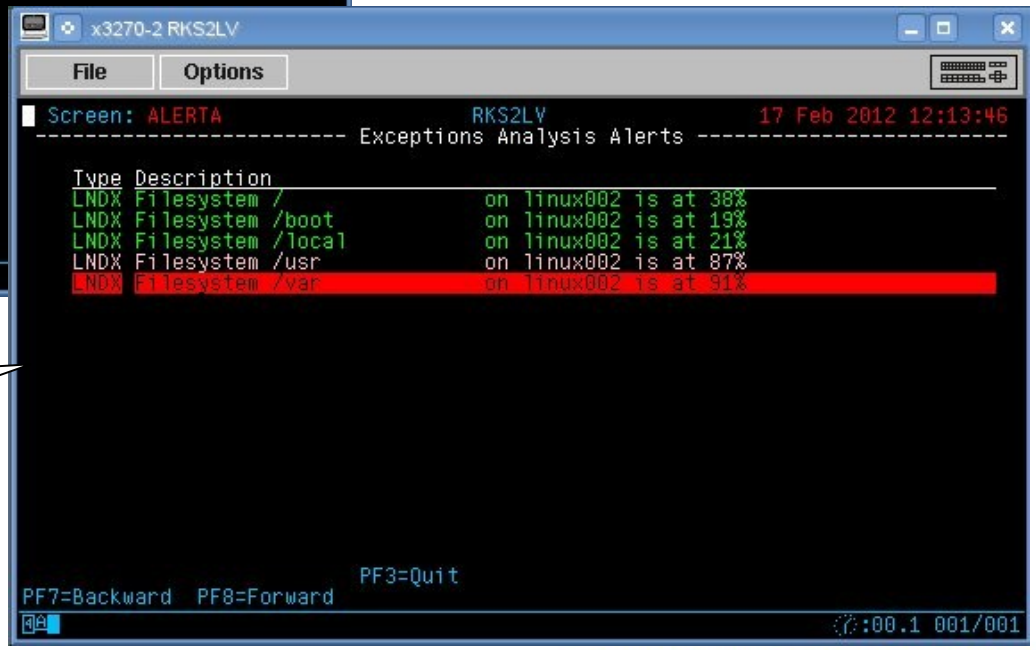
Parameters passed from alert

Value returned to the alert

- **Results of function call**



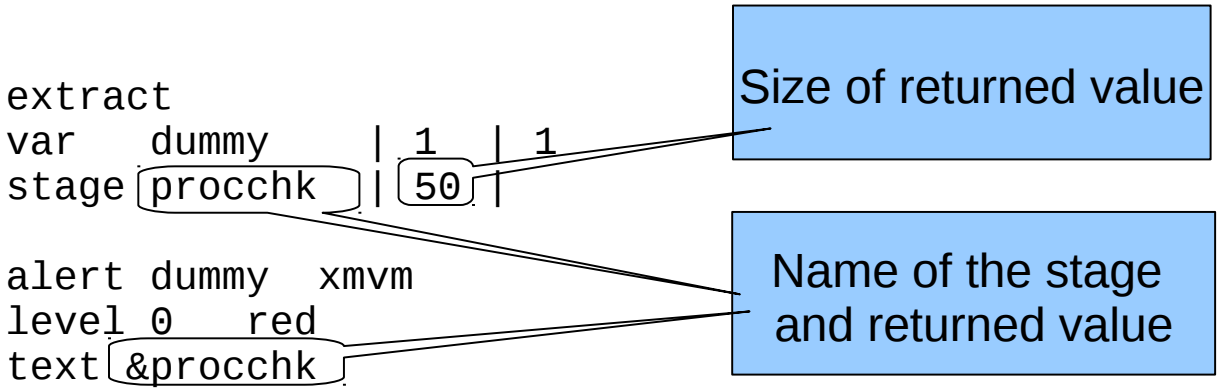
Original display



With DISKPCT EXEC

External Processing

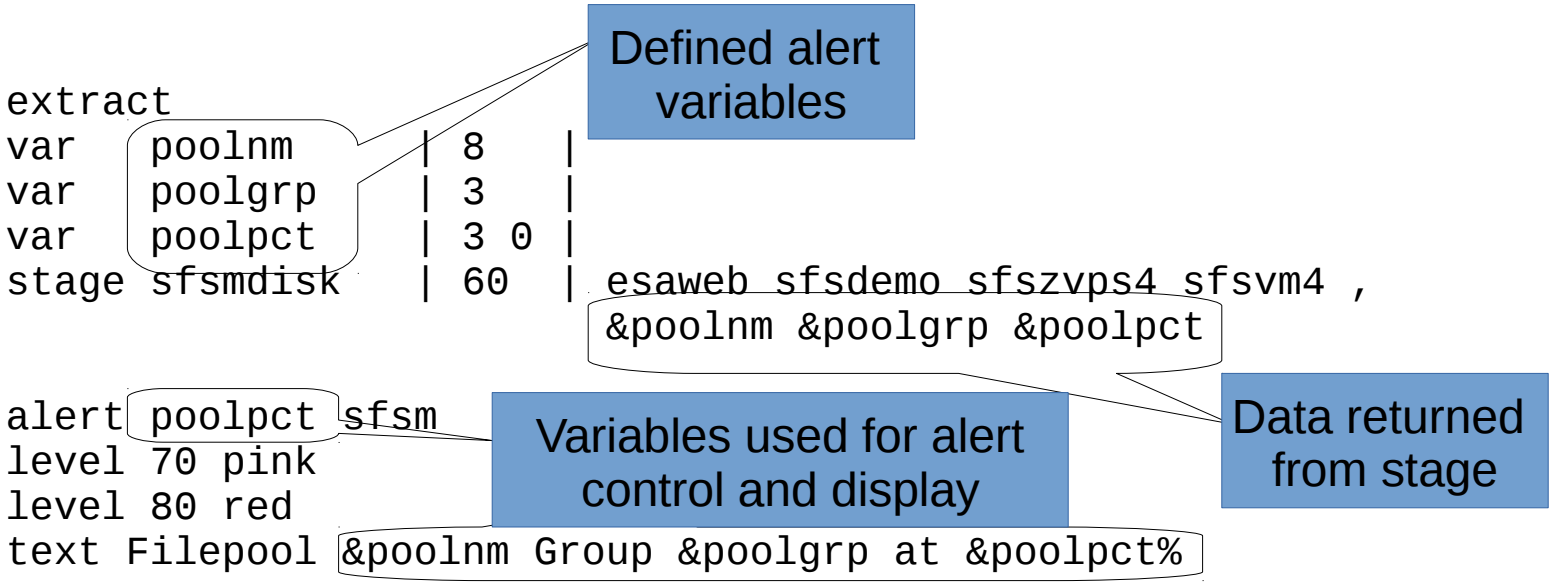
- **Stage is specified in place of 'var'**



- **Written as a pipe stage**
  - Using CALLPIPE to invoke pipes and return value(s)
  - Can execute zMON extracts

```
/*  Return msg stem to caller */
msg.0 = m
'CALLPIPE stem msg. | *:'
```

- **An alternative stage method returns values into alert variables**



More External Processing

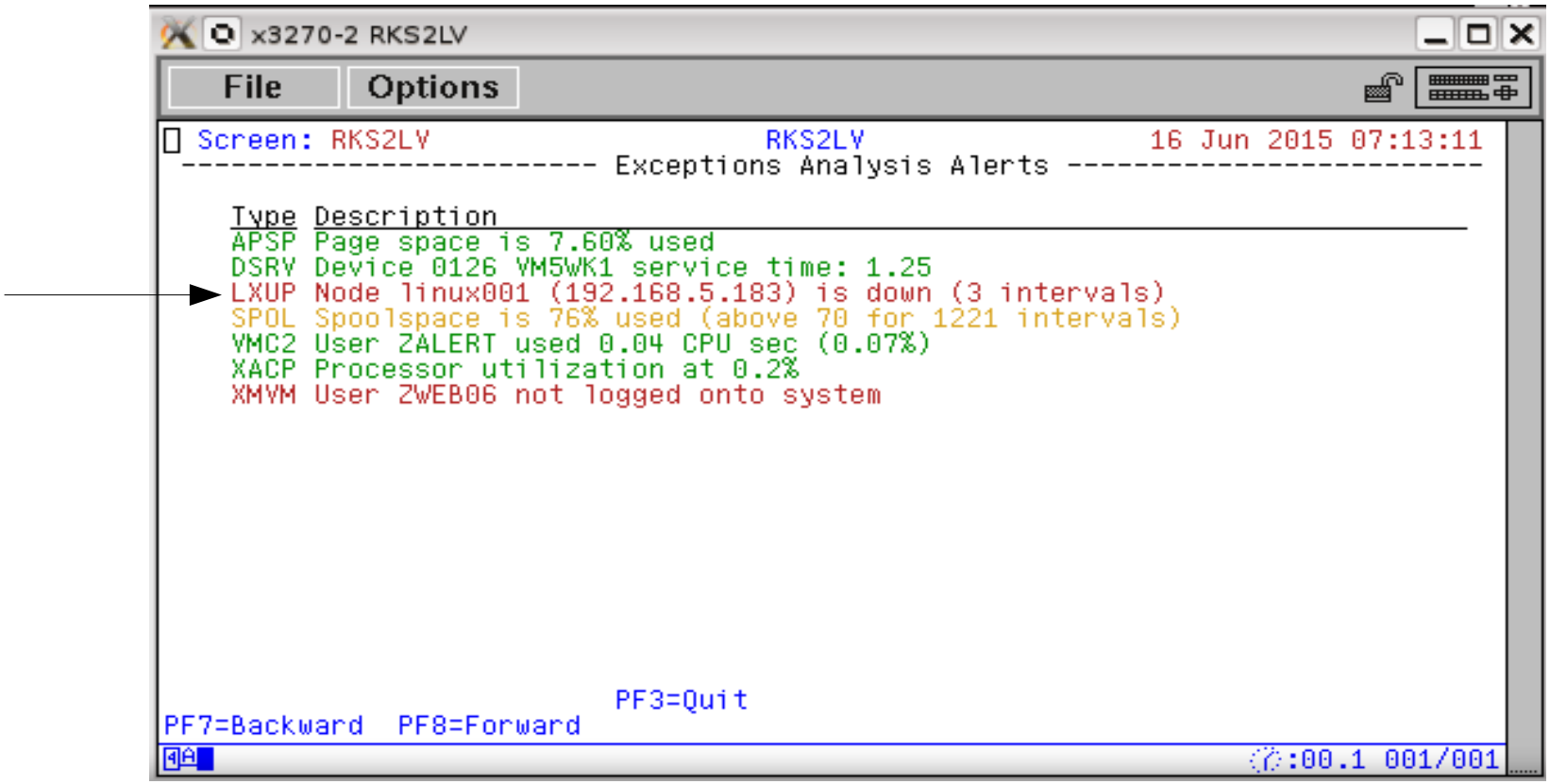
- **Check for 'node down'**

```
extract
parms node *
criteria hstsys.iplyy > 0
var  node       | 8  | tcpsys.node
var  ipaddr     | 15 | tcpsys.ipaddress
var  hsamp      | 1  | hstsys.samples

alert hsamp lxup | count &node
level =0 red
text Node &node (&ipaddr) is down (&tcount intervals)
```

- **No value in 'samples' indicates down**
- **Level allows additional comparison indicators for threshold evaluation (eg: < > = <= >= <>)**

More External Processing

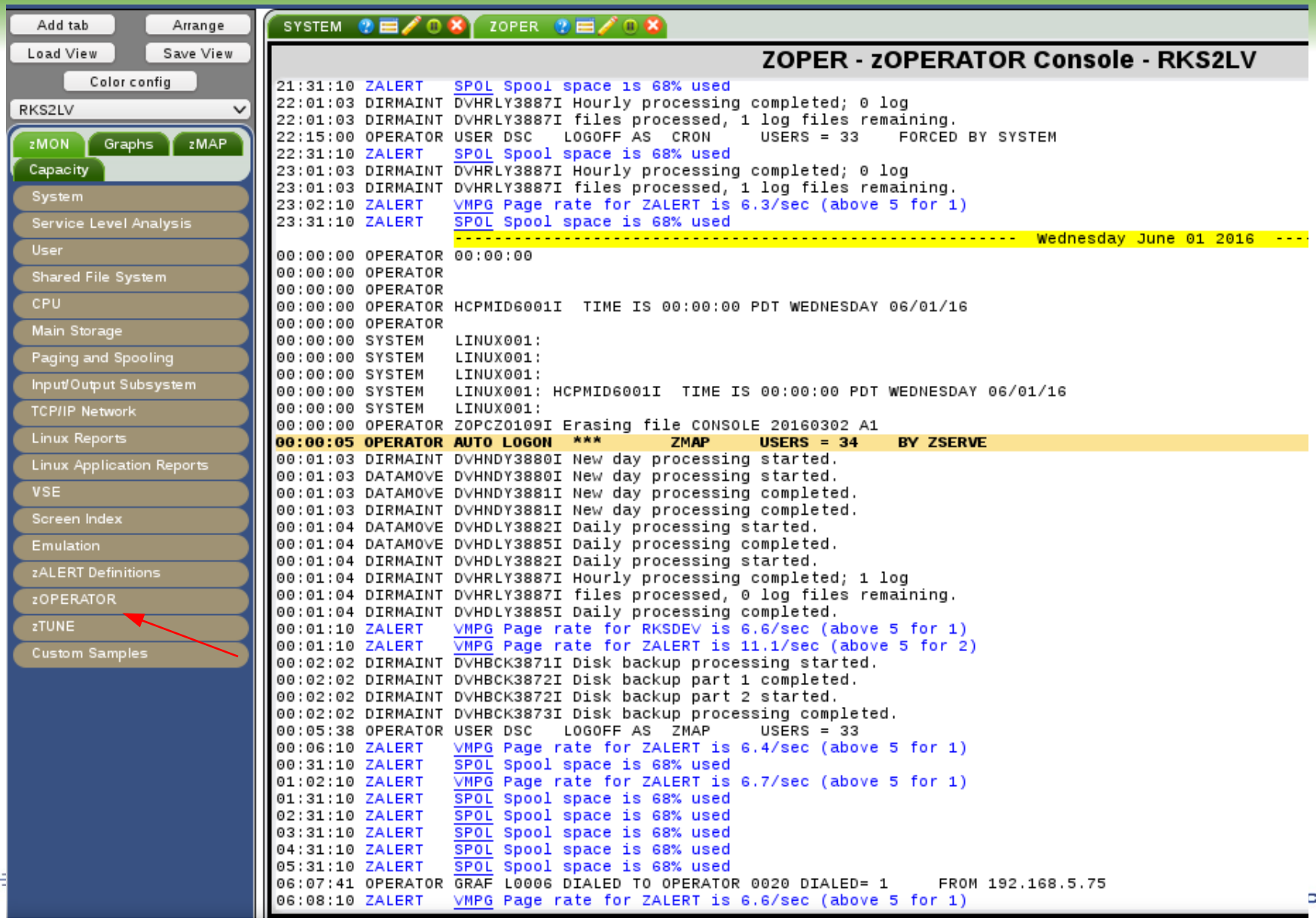
- **Results of 'node down'**



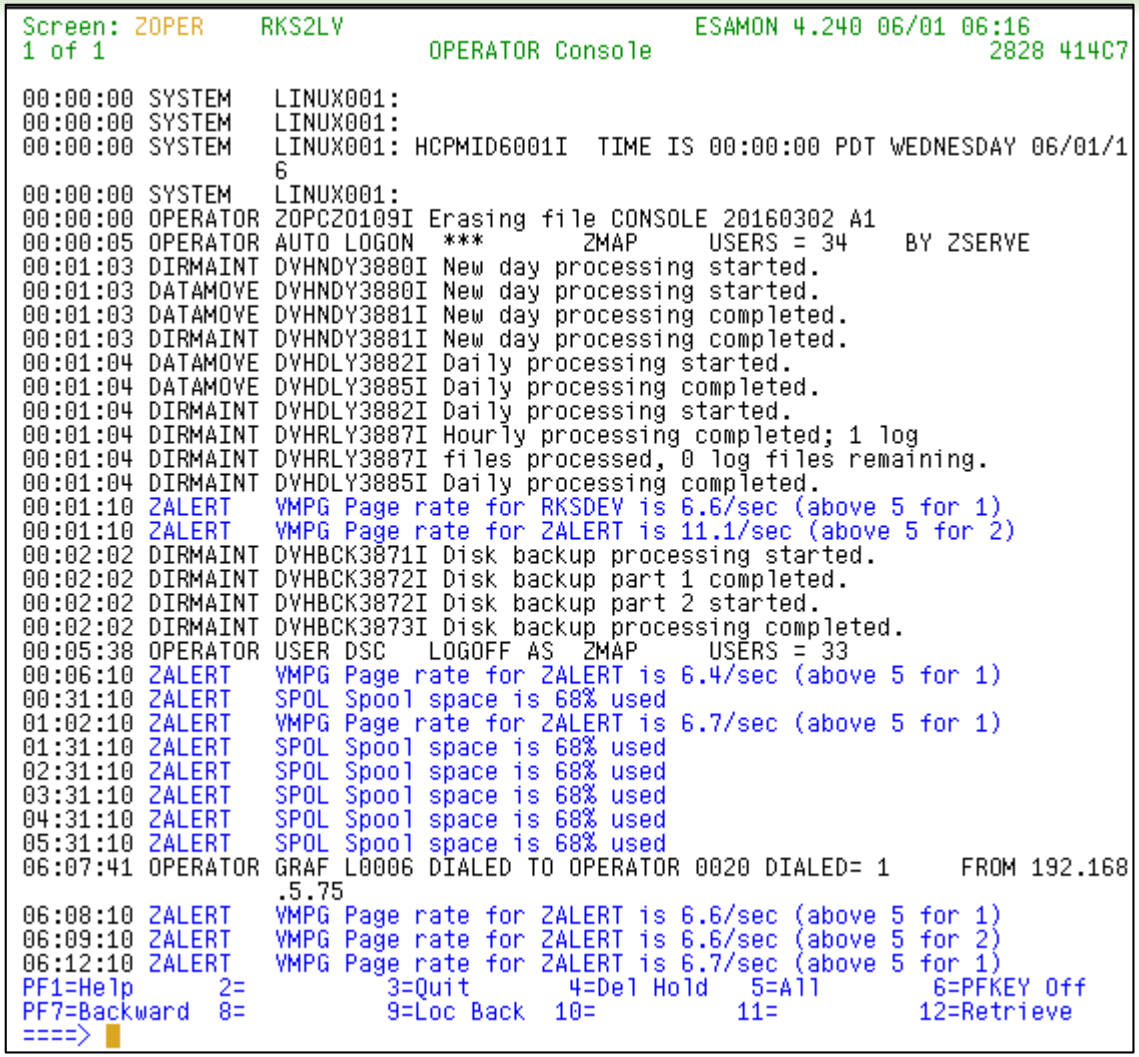
Integration with zOperator

- **zOperator enhances operational support**
- **Provides a traditional console display**
  - Message highlighting, suppression, actions
  - Logging, review, filtering
- **Allows operator console to be viewed via zView**
  - In addition to a 3270 dial interface

Integration with zOperator

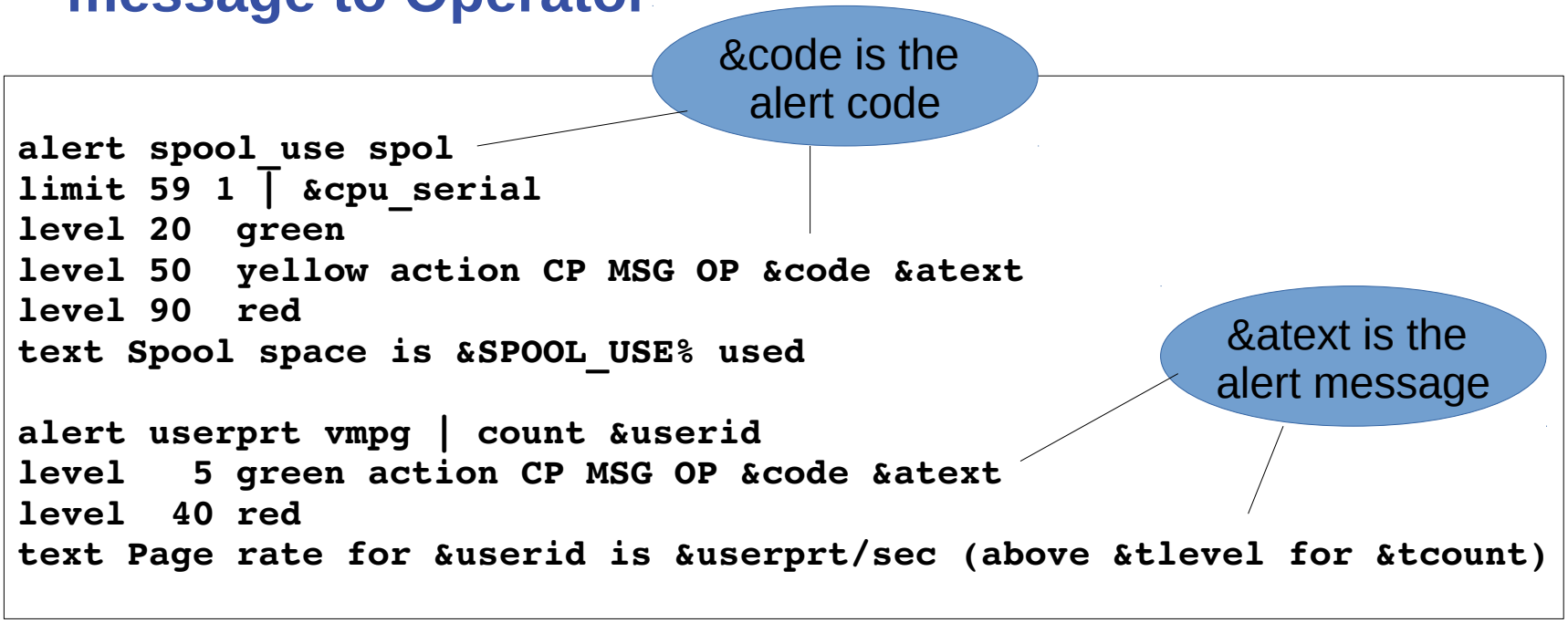


Integration with zOperator



Integration with zOperator

- **Use the ACTION feature of LEVEL to route a message to Operator**




```
alert spool_use spol
limit 59 1 | &cpu_serial
level 20  green
level 50  yellow action CP MSG OP &code &atext
level 90  red
text Spool space is &SPOOL_USE% used

alert userprt vmpg | count &userid
level   5 green action CP MSG OP &code &atext
level  40 red
text Page rate for &userid is &userprt/sec (above &tlevel for &tcount)
```

Integration with zOperator

- **Clickthru support provides a link on web based alert displays**
- **Clicking the link will open an appropriate display**

```
00:31:10 ZALERT    SPOL Spool space is 68% used
01:02:10 ZALERT    VMPG Page rate for ZALERT is 6.7/sec (above 5 for 1)
```

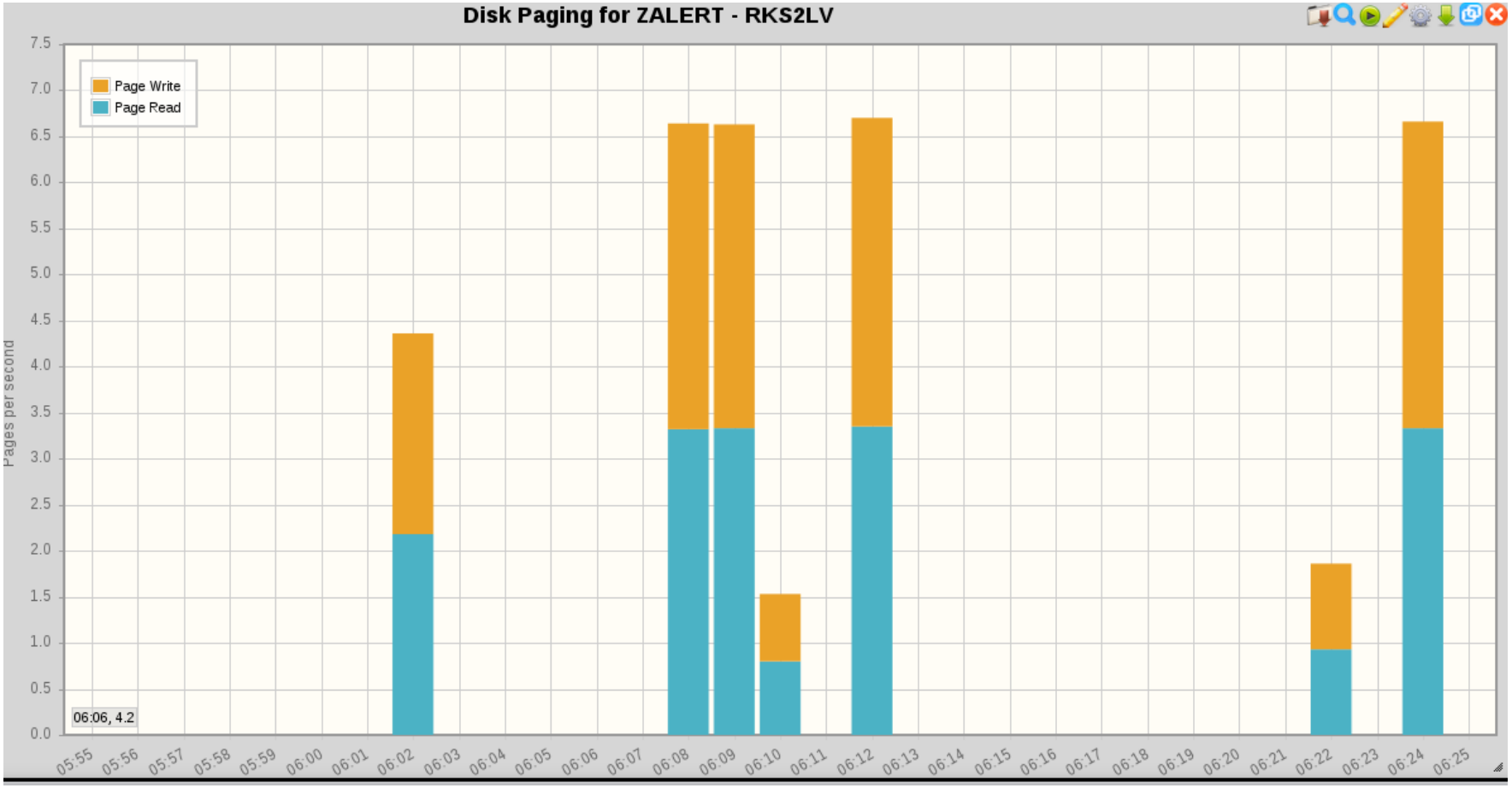
Click on SPOL...

ESAPSDV - Page And Spool Device Activity - RKS2LV

			<----------Paging---------->						<----------Spooling---------->						
	Dev		<---Slots--->				<--Rates-->		<---Slots--->				<--Rates-->		
Time	No.	Serial	Avail	Used	%Use	Max	Read	Write	Avail	Used	%Use	Max	Read	Write	Queued
06:32:00	0000	System	600840	121543	20.2	121543	5.1	1.8	56340	38003	67.5	38003	0.0	0.1	0
06:32:00	0124	VM5W01	0	0	.	0	0	0	56340	38003	67.5	38003	0.0	0.1	0
06:32:00	0127	VM5PG1	600840	121543	20.2	121543	5.1	1.8	0	0	.	0	0	0	0
06:31:00	0000	System	600840	121544	20.2	121544	0.2	0	56340	38001	67.4	38001	0	0	0
06:31:00	0124	VM5W01	0	0	.	0	0	0	56340	38001	67.4	38001	0	0	0
06:31:00	0127	VM5PG1	600840	121544	20.2	121544	0.2	0	0	0	.	0	0	0	0

Integration with zOperator

Click on VMPG...



Integration with zOperator

- **How does this work?**
- **zAlert config file contains CLICKTHRU statements**
  - They define the type of link to generate for web based displays
  - Screen type, screen name, and location of a qualifier

```
CLICKTHRU SPOL SCREEN=ESAPSDV
CLICKTHRU PGRT GRAPH=PAGERATE
CLICKTHRU VMPG GRAPH=USERPAGE USER=W4
CLICKTHRU DSRV GRAPH=IORESPDV USER=W2
CLICKTHRU VMIO SCREEN=ESAUSR3 USER=W5
CLICKTHRU LNSU GRAPH=LNXSWPTM NODE=W6
CLICKTHRU LNDX SCREEN=ESAHST2 NODE=W4
CLICKTHRU DVRT SCREEN=ESADSD2 DEVICE=W6
CLICKTHRU VMCP GRAPH=USERCPU USER=W2
CLICKTHRU XACP GRAPH=CPUUTIL
```

Integration with zOperator

- Create your own CLICKTHRU's
- … or modify ours

- For existing CONFIG files, the samples are contained in the file CLIKTHRU SAMPLE

- The alert engine virtual machine (ZALERT) should be XAUTOLOG-ged shortly after the monitor (ZSERVE)
- Config file (CONFIG ZALERT)

```
/*
/*  Configuration data for zALERT
/*

AUTHUSER   ZVPS
LOADEXT    YES
NTFYLOGS   30
LOGRETAIN  15
ALERTFILE RKS2LV ALRT6
```

- **Many ZALERT functions can be controlled via SMSG**
  - CMS
  - CP
  - QUERY
  - REREAD
  - RESTART
  - SET
  - STATUS
  - STOP

- **CMS – Returns the result of a CMS command**
- **CP – Returns the result of a CP command**

```
smsg zalert cms q search
 * MSG FROM ZALERT  : Processing command CMS Q SEARCH for ZVPS
 * MSG FROM ZALERT  : -       DIR   A    R/W   SFSZVPS:ZALERT.
 * MSG FROM ZALERT  : -       DIR   C    R/O   SFSZVPS:ZVPS.CONFIG
 * MSG FROM ZALERT  : -       DIR   D    R/O   SFSZVPS:ZMON.CODE
 * MSG FROM ZALERT  : MNT190  190   S    R/O
 * MSG FROM ZALERT  : MNT19E  19E   Y/S  R/O
```

- **REREAD – Rebuild internal structures from alert files**
  - LIMIT counts are preserved
- **RESTART – Terminate and start alert engine**

- **QUERY – See alert file status**

```
smsg zalert query rks2lv alerts
* MSG FROM ZALERT  : Processing command QUERY RKS2LV ALERTS for ZVPS
* MSG FROM ZALERT  : Alerts in RKS2LV:
* MSG FROM ZALERT  : AACP    DSRV    LNSR    SPOL    VMLP
* MSG FROM ZALERT  : APSP    DVRT    LNSU    VMCP-D  VMPG
* MSG FROM ZALERT  : DBSY    INQU    LXUP    VMC2    XACP
* MSG FROM ZALERT  : DEVQ    LNCP    PGAL    VMIO    XMVM
* MSG FROM ZALERT  : DRPN    LNDX    PGRT    VMLP    XPRO
```

- **QUERY – See include/exclude entries**

```
smsg zalert query rks2lv include vmc2
* MSG FROM ZALERT : Processing command QUERY RKS2LV INCLUDE VMC2 for ZVPS
* MSG FROM ZALERT : Include list for VMC2, include var USERID
* MSG FROM ZALERT : RKSDEV
* MSG FROM ZALERT : ZALERT
```

- ## Set – Enable/Disable an alert

```
smsg zalert set rks2lv disable vmcp
* MSG FROM ZALERT : Processing command SET RKS2LV DISABLE VMCP for ZVPS
* MSG FROM ZALERT : Alert VMCP in RKS2LV is disabled

smsg zalert set rks2lv enable vmcp
* MSG FROM ZALERT : Processing command SET RKS2LV ENABLE VMCP for ZVPS
* MSG FROM ZALERT : Alert VMCP in RKS2LV is enabled
```

- **Set – Alter include/exclude list**

```
smsg zalert query rks2lv include vmc2
 * MSG FROM ZALERT  : Processing command QUERY RKS2LV INCLUDE VMC2 for ZVPS
 * MSG FROM ZALERT  : Include list for VMC2, include var USERID
 * MSG FROM ZALERT  : RKSDEV
 * MSG FROM ZALERT  : ZALERT

smsg zalert set rks2lv include vmc2 add userid linux001
 * MSG FROM ZALERT  : Processing command SET RKS2LV INCLUDE VMC2 ADD USERID LINUX001 for ZVPS
 * MSG FROM ZALERT  : Alert VMC2 in RKS2LV, LINUX001 added to include list for USERID

smsg zalert query rks2lv include vmc2
 * MSG FROM ZALERT  : Processing command QUERY RKS2LV INCLUDE VMC2 for ZVPS
 * MSG FROM ZALERT  : Include list for VMC2, include var USERID
 * MSG FROM ZALERT  : RKSDEV
 * MSG FROM ZALERT  : ZALERT
 * MSG FROM ZALERT  : LINUX001
```

- **Set – Alter include/exclude list**

```
smsg zalert query rks2lv include vmc2
 * MSG FROM ZALERT  : Processing command QUERY RKS2LV INCLUDE VMC2 for ZVPS
 * MSG FROM ZALERT  : Include list for VMC2, include var USERID
 * MSG FROM ZALERT  : RKSDEV
 * MSG FROM ZALERT  : ZALERT
 * MSG FROM ZALERT  : LINUX001

smsg zalert set rks2lv include vmc2 rem userid linux001
 * MSG FROM ZALERT  : Processing command SET RKS2LV INCLUDE VMC2 REM USERID LINUX001 for ZVPS
 * MSG FROM ZALERT  : Alert VMC2 in RKS2LV, LINUX001 removed from include list for USERID

smsg zalert query rks2lv include vmc2
 * MSG FROM ZALERT  : Processing command QUERY RKS2LV INCLUDE VMC2 for ZVPS
 * MSG FROM ZALERT  : Include list for VMC2, include var USERID
 * MSG FROM ZALERT  : RKSDEV
 * MSG FROM ZALERT  : ZALERT
```

- **STATUS – Display alert engine status**

```
smsg zalert status
* MSG FROM ZALERT  : Processing command STATUS for ZVPS
* MSG FROM ZALERT  : zAlert Level 4.2.0.3
* MSG FROM ZALERT  : DCSS Cap: 8191 HWM: 82 Cur: 21   0.3%
* MSG FROM ZALERT  : Processing files RKS2LV VMALERT LINALERT
```

- **STOP – Terminate alert engine**

```
smsg zalert stop
* MSG FROM ZALERT : Processing command STOP for ZVPS
```

  - QUIT and END are synonyms for STOP

Summary

- Alerts provide the way to passively monitor your system
- Thresholds exceeded are displayed on one screen
- Notifications can be delivered for critical issues
- Management consoles fit this mechanism perfectly
- Many useful samples are provided

Questions

Rich Smrcina
Velocity Software, Inc
rich@velocitysoftware.com