zVPS Alerts

Richard Smrcina
Velocity Software, Inc.
March, 2012

VELOCITY SOFTWARE

PROVEN PERFORMANCE

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
- **Advanced topics**

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

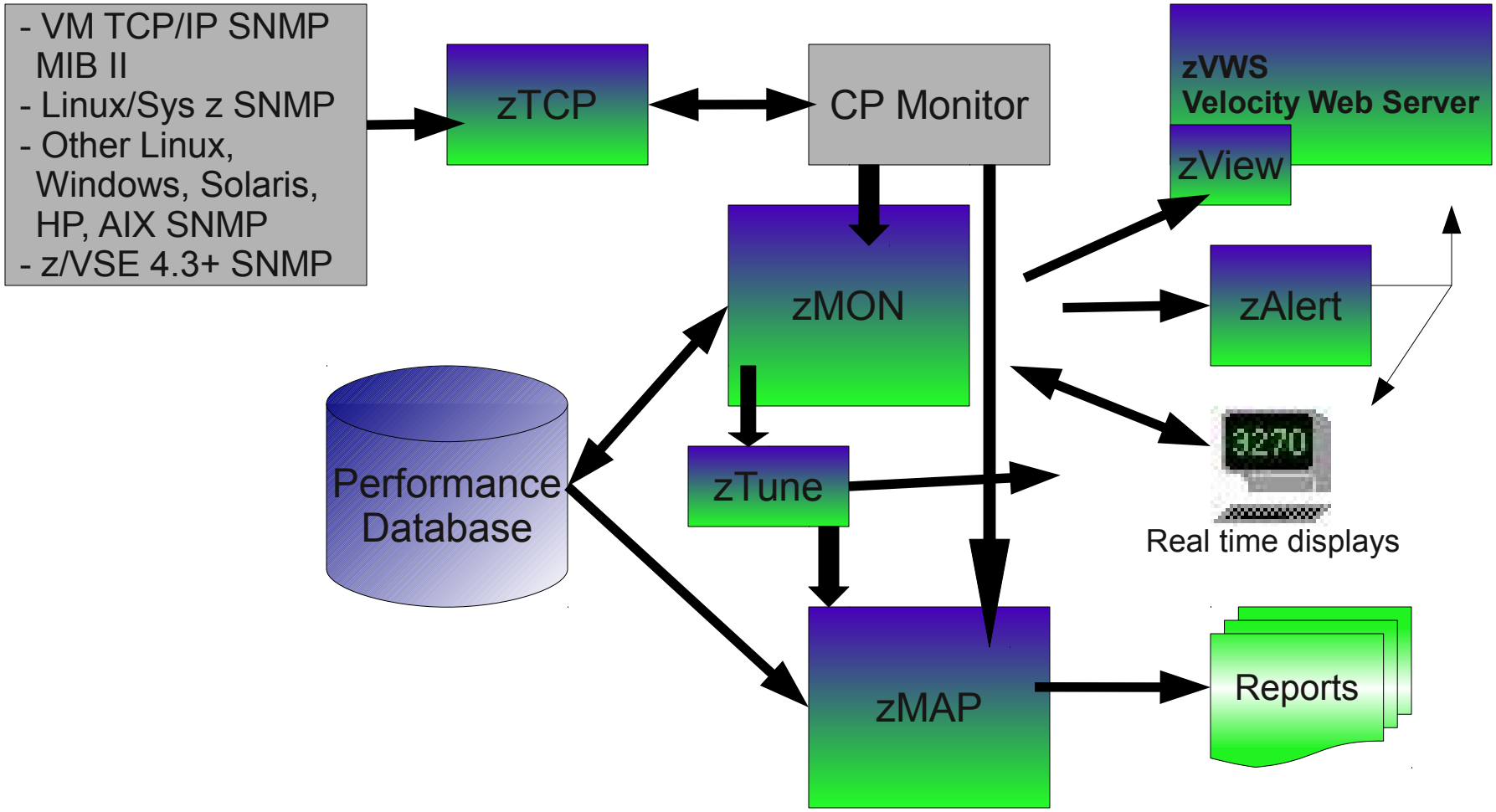
Overview

This presentation goes through the finer points of alert processing.

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

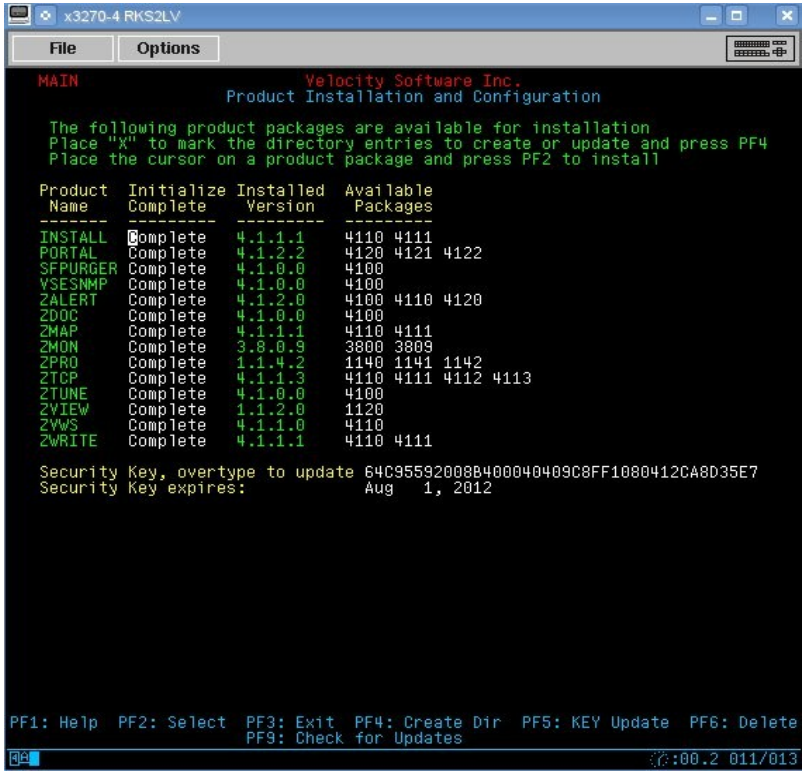
Alerts are no good if they need to be visually watched or monitored... a proactive mechanism to using alerts, notifications, are also covered.

Where do alerts fit?



Installing zAlert package

- **zAlert is part of the Velocity Performance Suite (zVPS)**
- **Installed via the Version 4 installer**
  - Creates a virtual machine (ZALERT)
    - By default SFSZVPS:ZALERT.
  - Sample alerts provided
  - More on the website



VELOCITY SOFTWARE

Installing zAlert package

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

Installing zAlert package

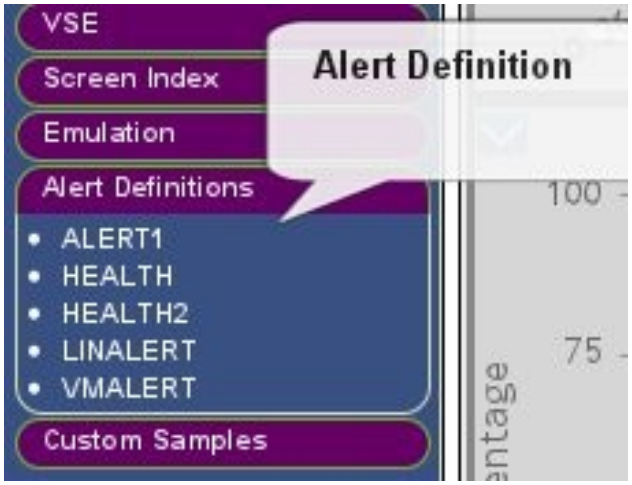
- **A notification can be any of**
  - Message on the alert console or zView
  - CP MSG to a user
  - Email to interested parties
    - Text message on a mobile device
  - SNMP trap sent to a management console
  - Combinations of the above

Viewing alerts

- **Terminal session**
  - Via ESAMON <alertfile>
    eg: ESAMON ALERT1

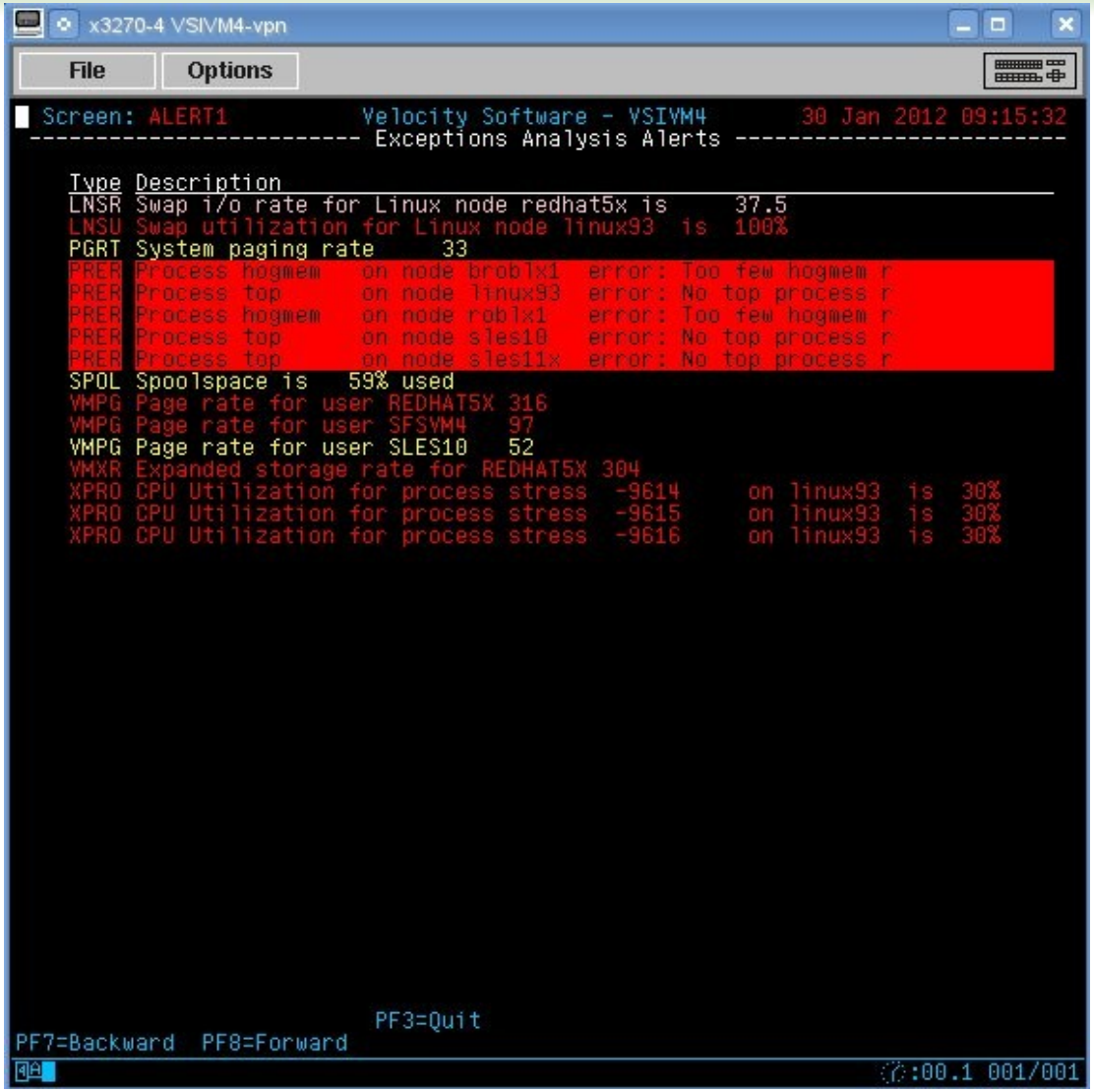
```
vmlink .dir sfszvps:zmon.code
vmlink .dir sfszvps:zvps.config
ESAMON ALERT1
```

- **zView**
  - Navigate to 'Alert Definitions' menu
  - Select alert file to display



Viewing alerts

- **Alert display on 3270**



Viewing alerts

- **Alert display on zView**

Exceptions Analysis Alerts - 12/01/30 at 09:16

Code	Alert Description
DVRT	I/O rate for volume 0153(DXC2L2) 21/sec
LNSR	Swap i/o rate for Linux node redhat5x is 18.7
LNSU	Swap utilization for Linux node linux93 is 100%
PGRT	System paging rate 29
PRER	Process hogmem on node broblx1 error: Too few hogmem r
PRER	Process top on node linux93 error: No top process r
PRER	Process hogmem on node roblx1 error: Too few hogmem r
PRER	Process top on node sles10 error: No top process r
PRER	Process top on node sles11x error: No top process r
SPOL	Spoolspace is 59% used
VMCP	User SUSELNX2 at 44.9% of processor
VMPG	Page rate for user ROBLX1 65
VMPG	Page rate for user SLES10 311
VMPG	Page rate for user SLES11 108
VMPG	Page rate for user SLES8 86
VMPG	Page rate for user SUSELNX2 143
VMXR	Expanded storage rate for REDHAT3 717
XACP	Processor utilization at 63.3%
XPRO	CPU Utilization for process stress -9614 on linux93 is 30%
XPRO	CPU Utilization for process stress -9615 on linux93 is 30%
XPRO	CPU Utilization for process stress -9616 on linux93 is 30%
XPRO	CPU Utilization for process smallstr-24385 on suselnx2 is 43%

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
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 CPU_TOTAL

var  cpu_serial | 0   | serial
var  util       | 5 1 | sytprp.cpuutil

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

ALERT statement defines a specific alert

Four character code used when displaying alerts

Each alert requires a previously defined variable

- **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**

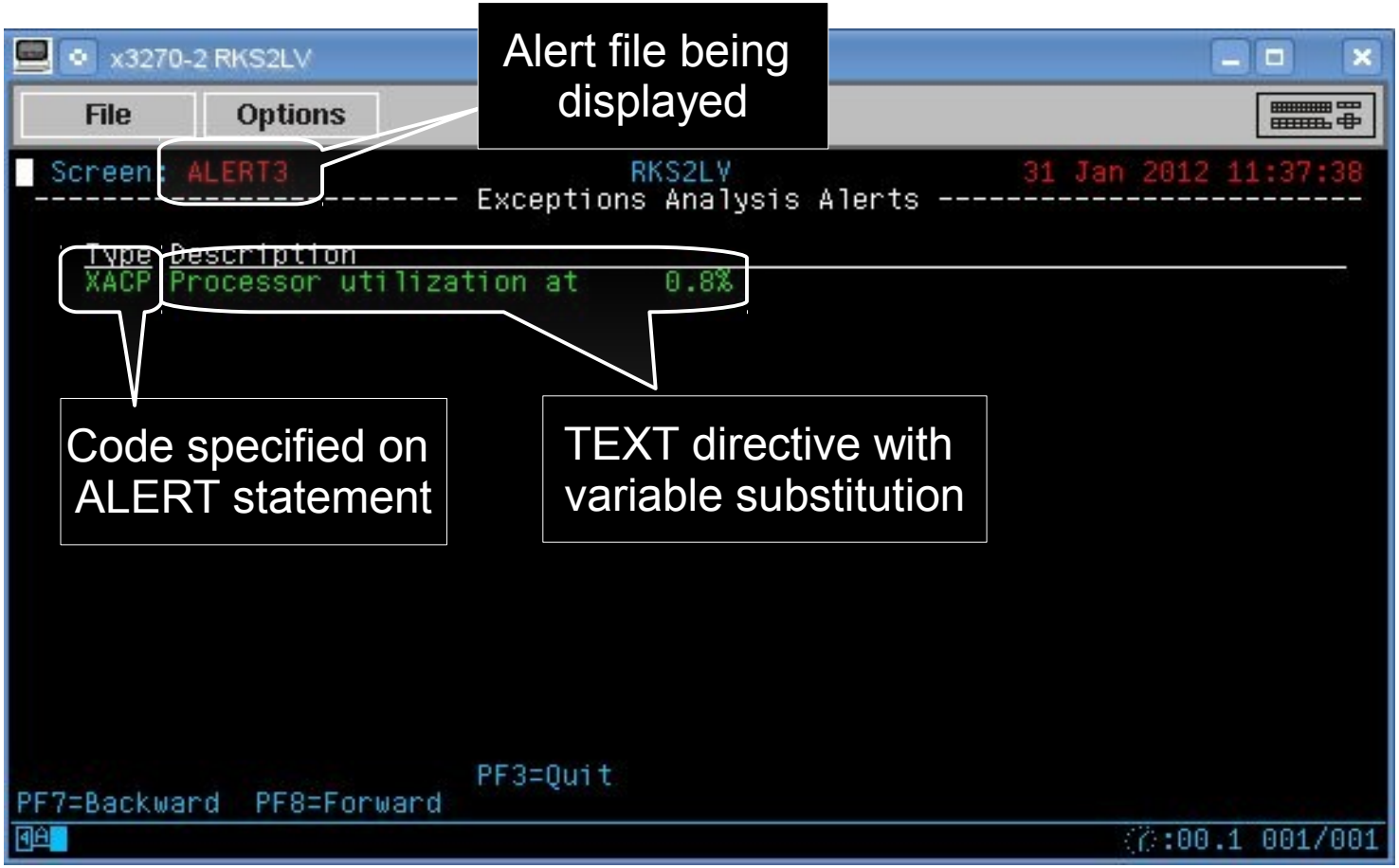
```
Extract
Parms CPU TOTAL

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

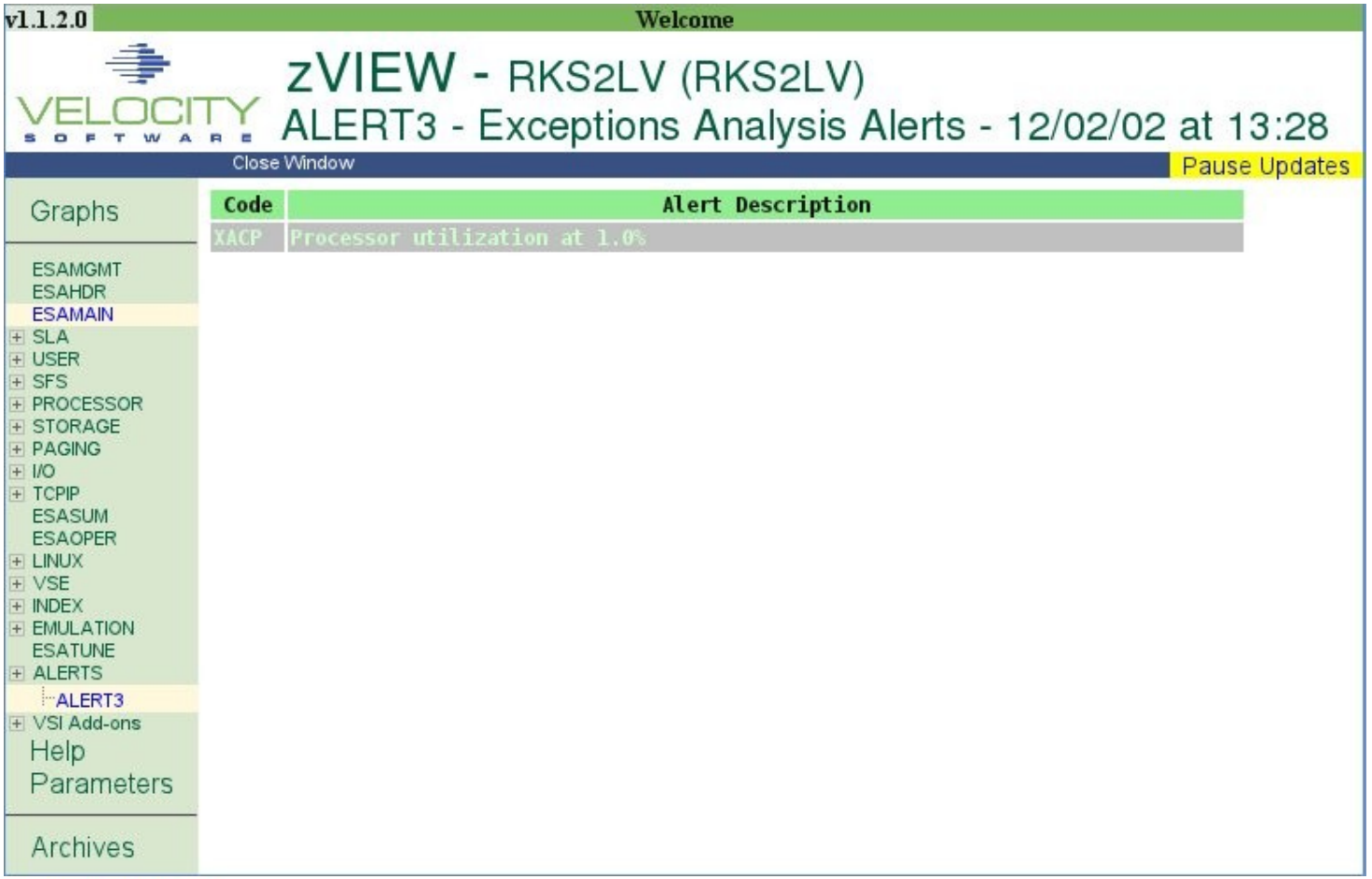
Alert result - 3270

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



Alert result - zView

- **Same data in zView**



Alert rules

- **Adjust the number and value of levels based on local requirements**
  - At least one LEVEL statement is necessary
  - LEVEL statements must appear in ascending value order
- **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**

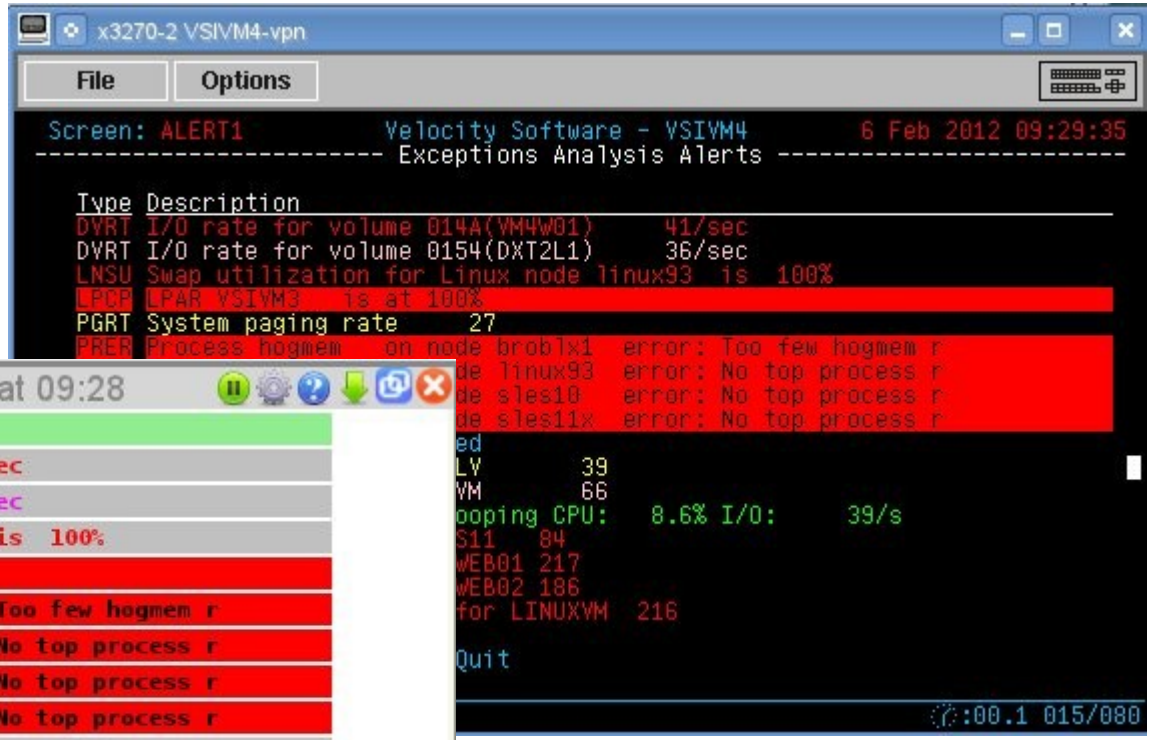
```
Extract
Parms LPAR *
Criteria sytcup.lcupname <> 'Totals:'
var  lparname    | 8   | sytcup.lcupname
var  lparutil    | 3 0 | sytcup.pctcpu

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)

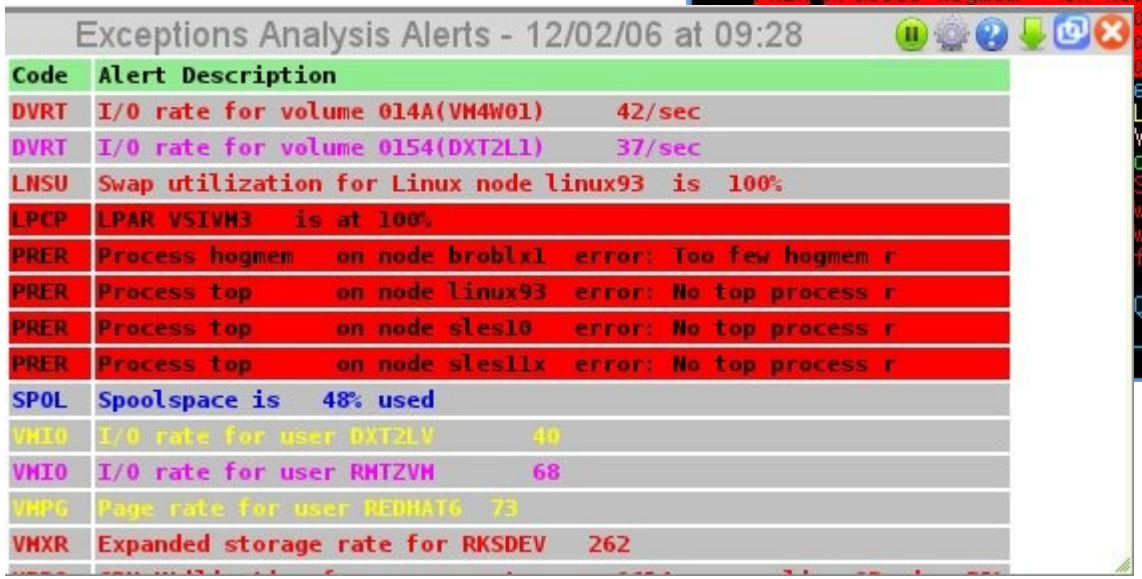
Defining your own alert - LPAR

- **Alert for LPAR Utilization display**

3270 >



zView v



Defining your own alert – Complex operations

- **Numerous fields can be combined using math operations**

```
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
```

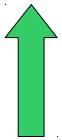
Defining your own alert – Second vdisk usage

- **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
```

Select address spaces beginning with vdisk

Common 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

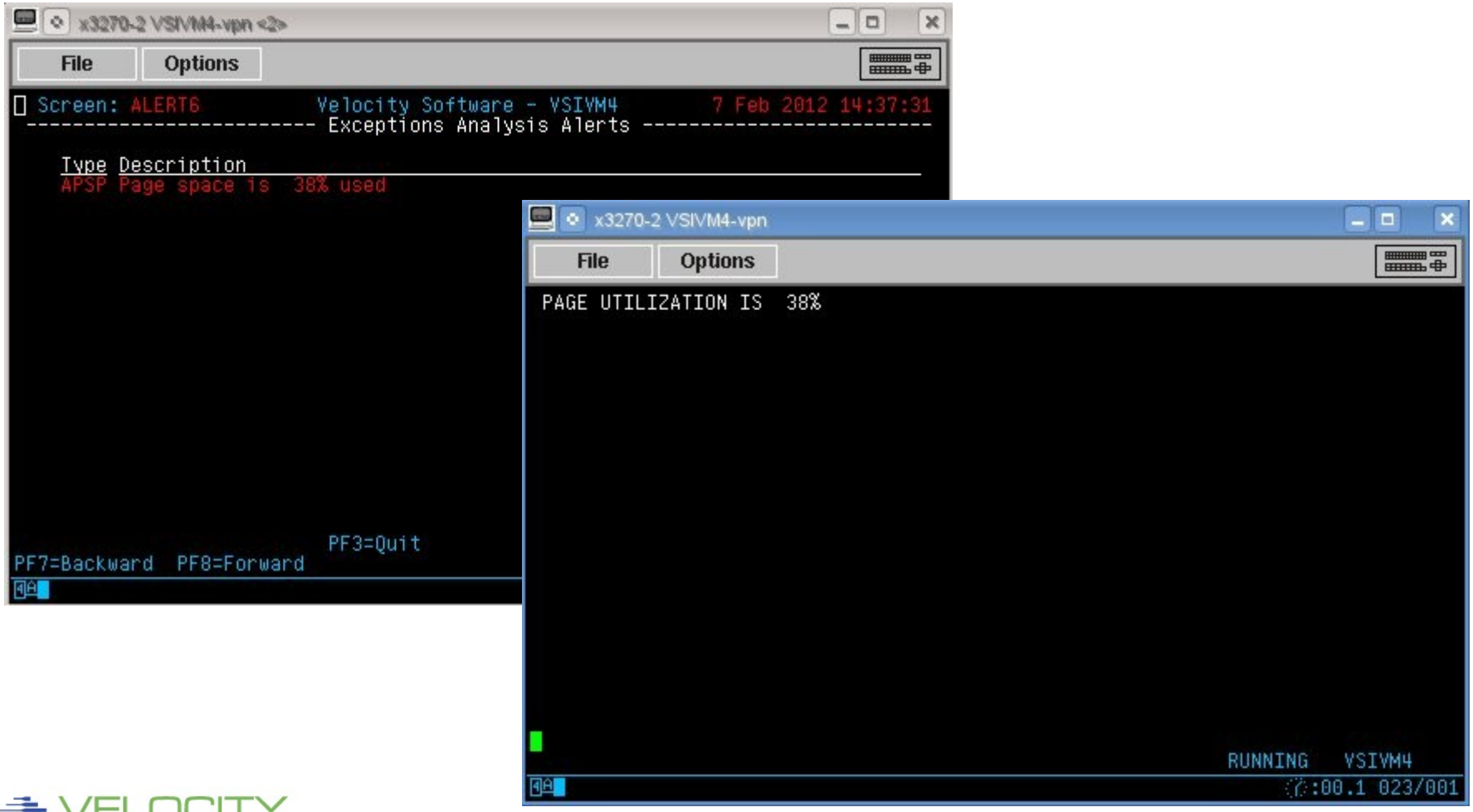
- **At it's simplest a notification can take the form of a message to a CMS user**

```
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**



Notifications

- **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%
```

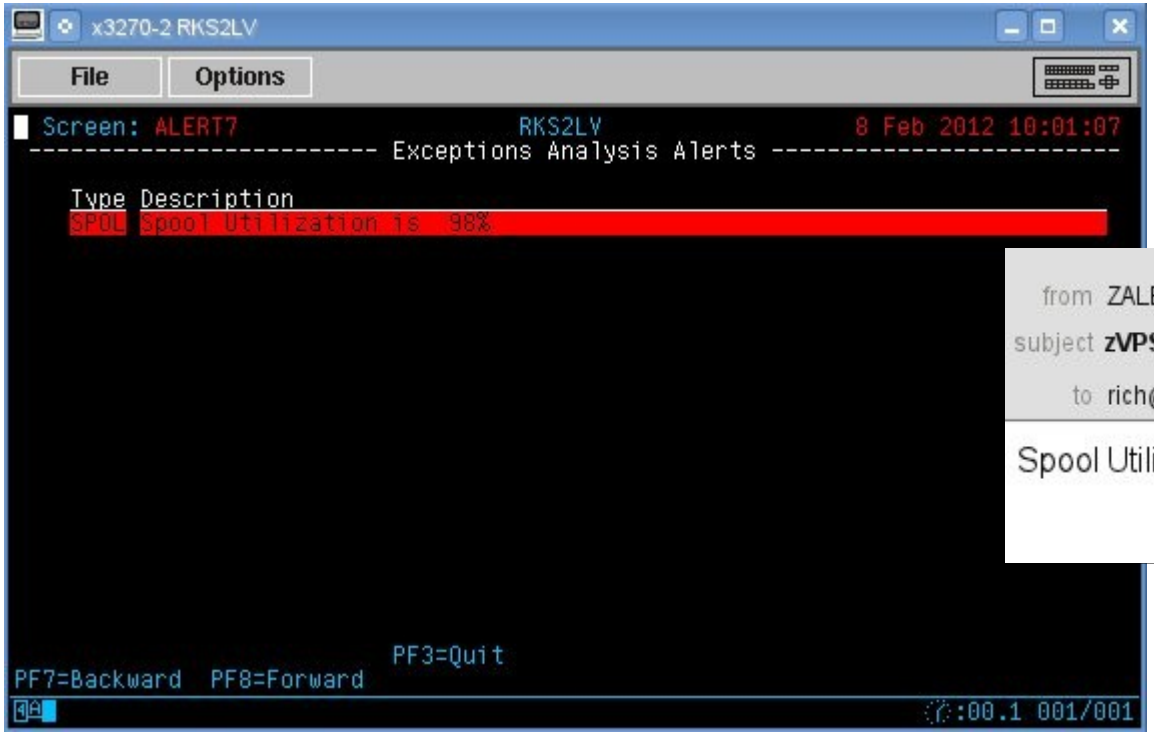
Executes a CMS command - EMAIL EXEC

- **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**



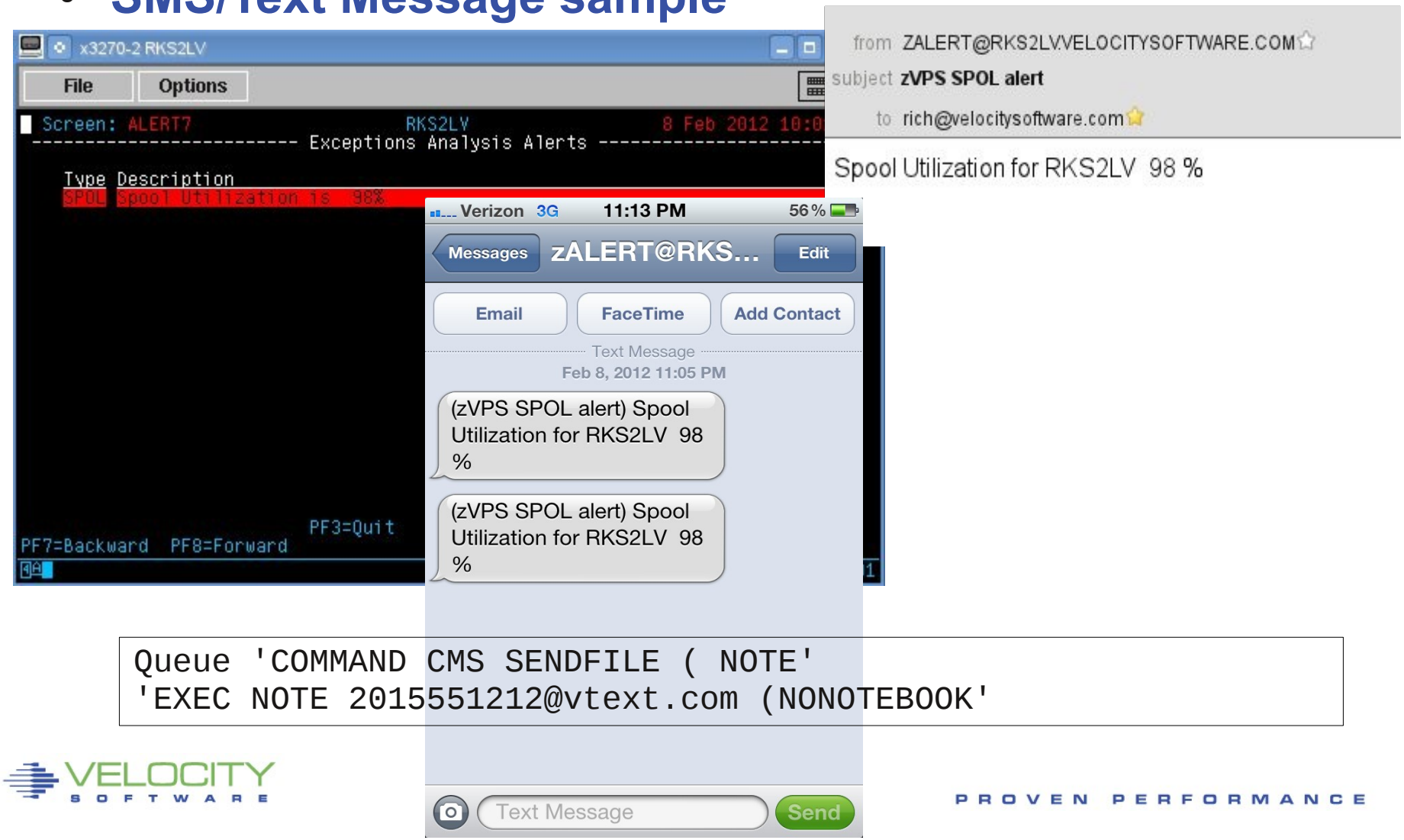
- **Cell phone text alerts**
  - Produced via an email message
  - Each carrier varies in their approach
  - List of Email to SMS gateways for most providers

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 directive 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 ALERT 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 ALERT 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%"
```

Advanced topics

- Limit
- Time
- Include/Exclude
- Multiple alerts
- External processing

Advanced topics - 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 2 5 | serial
level 70  yellow
level 80  red
level 90  red rev ACTION CP MSG ZVPS &date &time Spool Util is ,
          &spool_use%
text Spool Utilization is &spool_use%
```

- **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 2 5 | serial
level 70  yellow
level 80  red
level 90  red rev ACTION CP MSG ZVPS &date &time Spool Util is ,
          &spool_use%
text Spool Utilization is &spool_use%
```

Key field

Number of intervals to delay executing ACTION

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

Continuation IS allowed

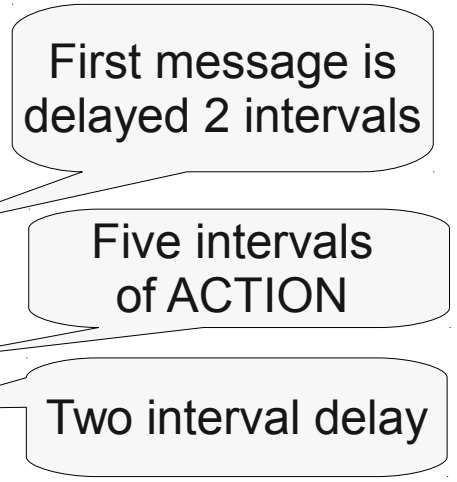
Advanced topics - Limit

- **This LIMIT directive:**

```
limit 2 5 | serial
```

- **Will delay ACTION for 2 intervals**
- **Execute ACTION for 5 intervals**
- **Repeat**
- **For example, when started at 11:52**

```
11:54:29  * MSG FROM ZALERT  : 10 Feb 2012 11:54 SPOOL UTIL IS  95%
11:55:29  * MSG FROM ZALERT  : 10 Feb 2012 11:55 SPOOL UTIL IS  95%
11:56:29  * MSG FROM ZALERT  : 10 Feb 2012 11:56 SPOOL UTIL IS  95%
11:57:29  * MSG FROM ZALERT  : 10 Feb 2012 11:57 SPOOL UTIL IS  95%
11:58:29  * MSG FROM ZALERT  : 10 Feb 2012 11:58 SPOOL UTIL IS  95%
12:01:30  * MSG FROM ZALERT  : 10 Feb 2012 12:01 SPOOL UTIL IS  95%
12:02:30  * MSG FROM ZALERT  : 10 Feb 2012 12:02 SPOOL UTIL IS  95%
12:03:30  * MSG FROM ZALERT  : 10 Feb 2012 12:03 SPOOL UTIL IS  95%
12:04:30  * MSG FROM ZALERT  : 10 Feb 2012 12:04 SPOOL UTIL IS  95%
12:05:30  * MSG FROM ZALERT  : 10 Feb 2012 12:05 SPOOL UTIL IS  95%
12:08:31  * MSG FROM ZALERT  : 10 Feb 2012 12:08 SPOOL UTIL IS  95%
```



Advanced topics - 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 &date &time Spool Util is ,
          &spool_use%
text Spool Utilization is &spool_use%
```

Multiple times can be specified

Timeframe for alert to be active

Advanced topics – Include/Exclude

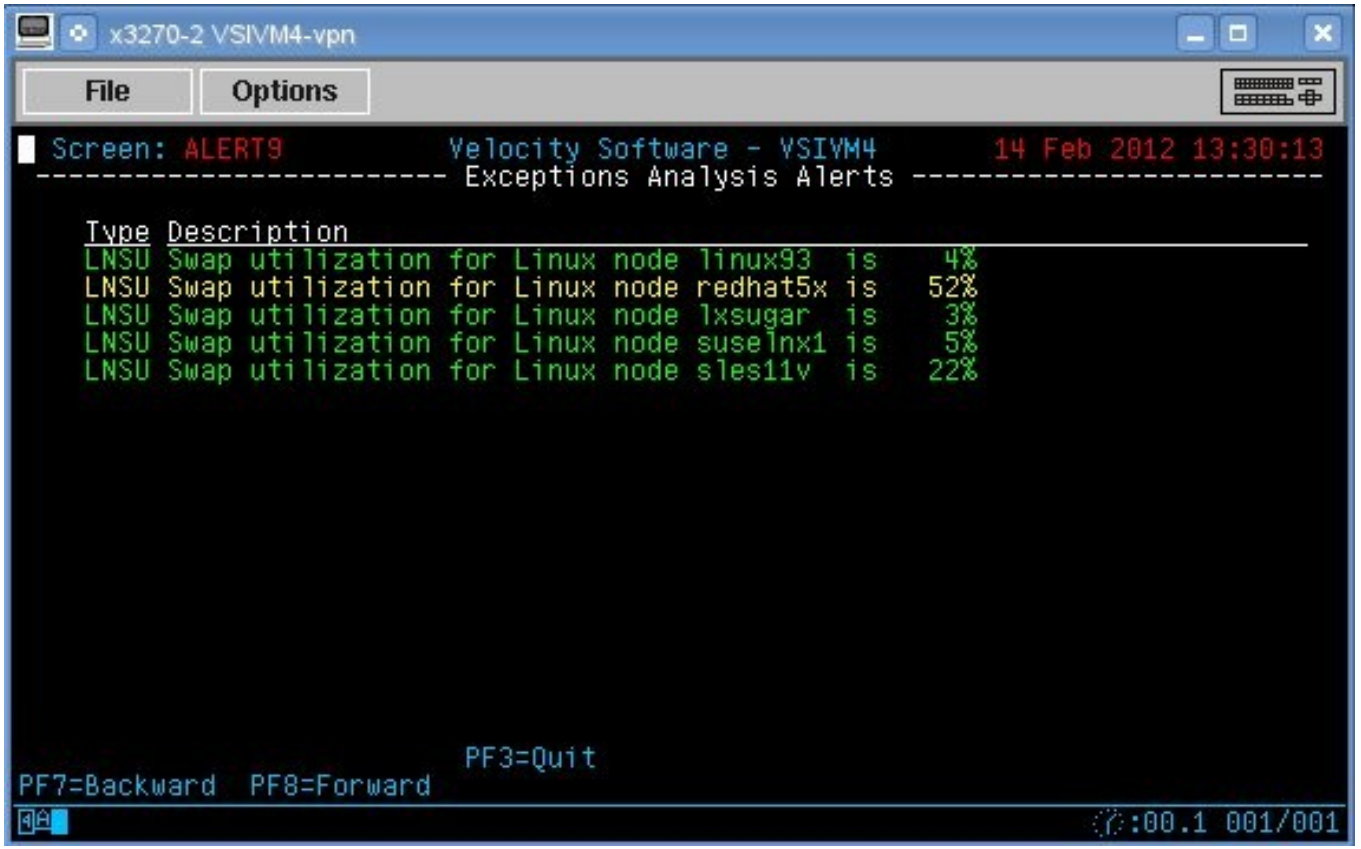
- **User or Node can be specified in an extract**
- **A subset can be selected with wildcards**
- **Given the following alert definition:**

All defined nodes are made available

```
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%
```

Advanced topics – Include/Exclude

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



Advanced topics – Include/Exclude

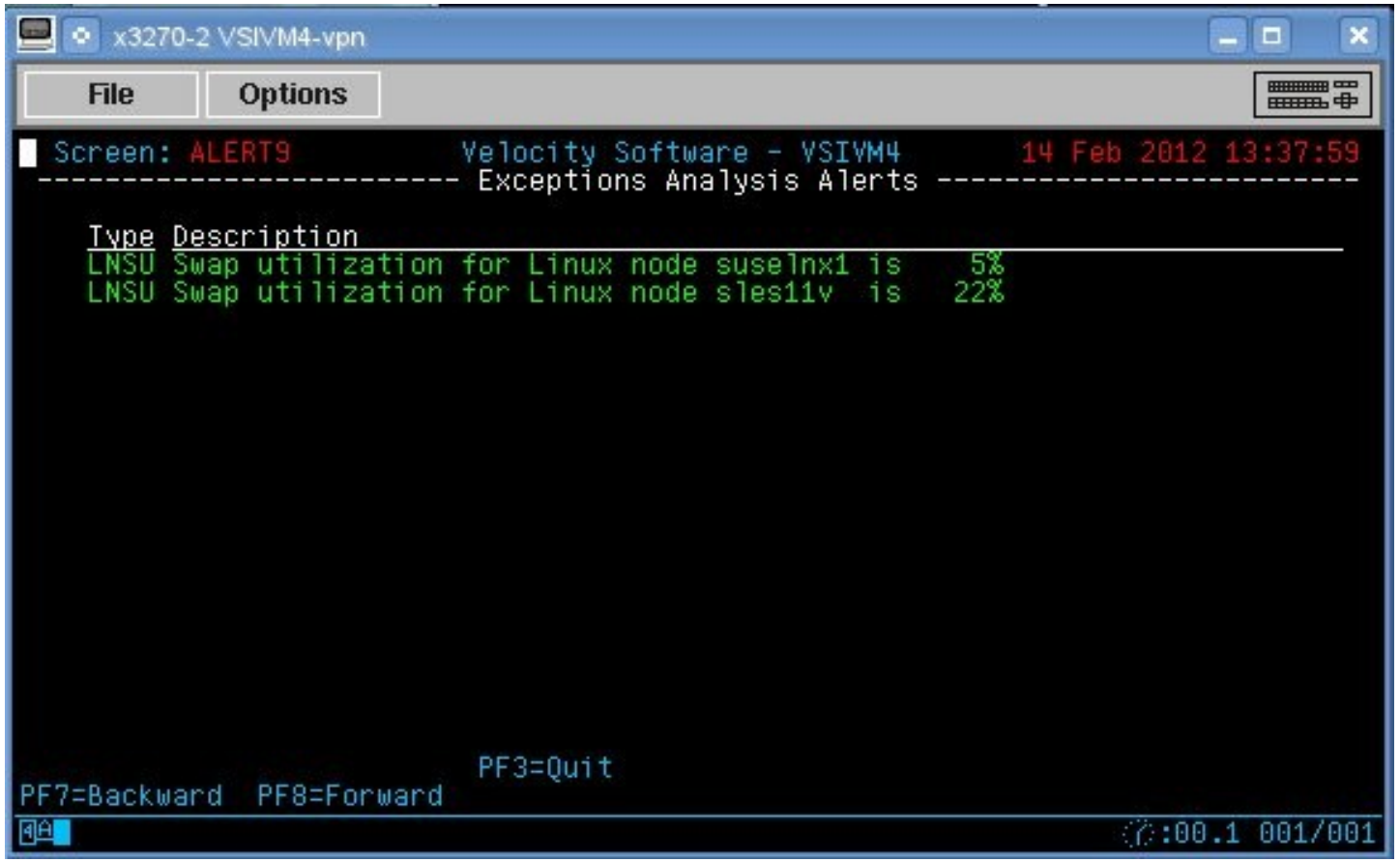
- **The alert can be tailored to show a subset by adjusting the wildcard**

```
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%
```

Only show nodes beginning with 's'

Advanced topics – Include/Exclude

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



Advanced topics – 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-
```

Advanced topics – 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-
```

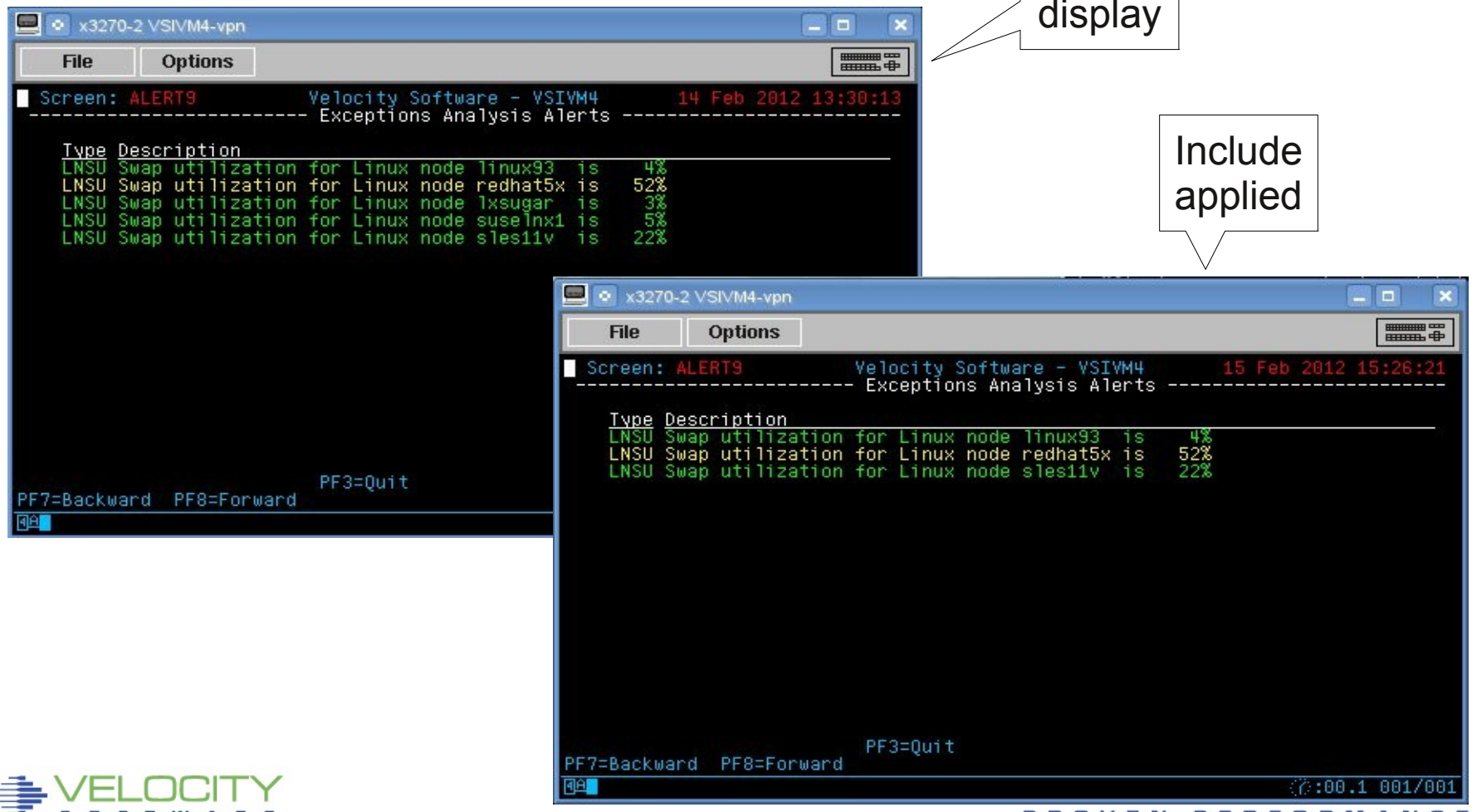
Variable used for matching

Include/Exclude file name must match the alert file name

List name applied to alert

Advanced topics – Include/Exclude

- **Results of Include file**



Advanced topics – 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%
```

Advanced topics – 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

Advanced topics – 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
```

Advanced topics – 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
```

Size of returned value

Parameters passed as exec args

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

Advanced topics – External Processing

- **REXX exec called from the alert**

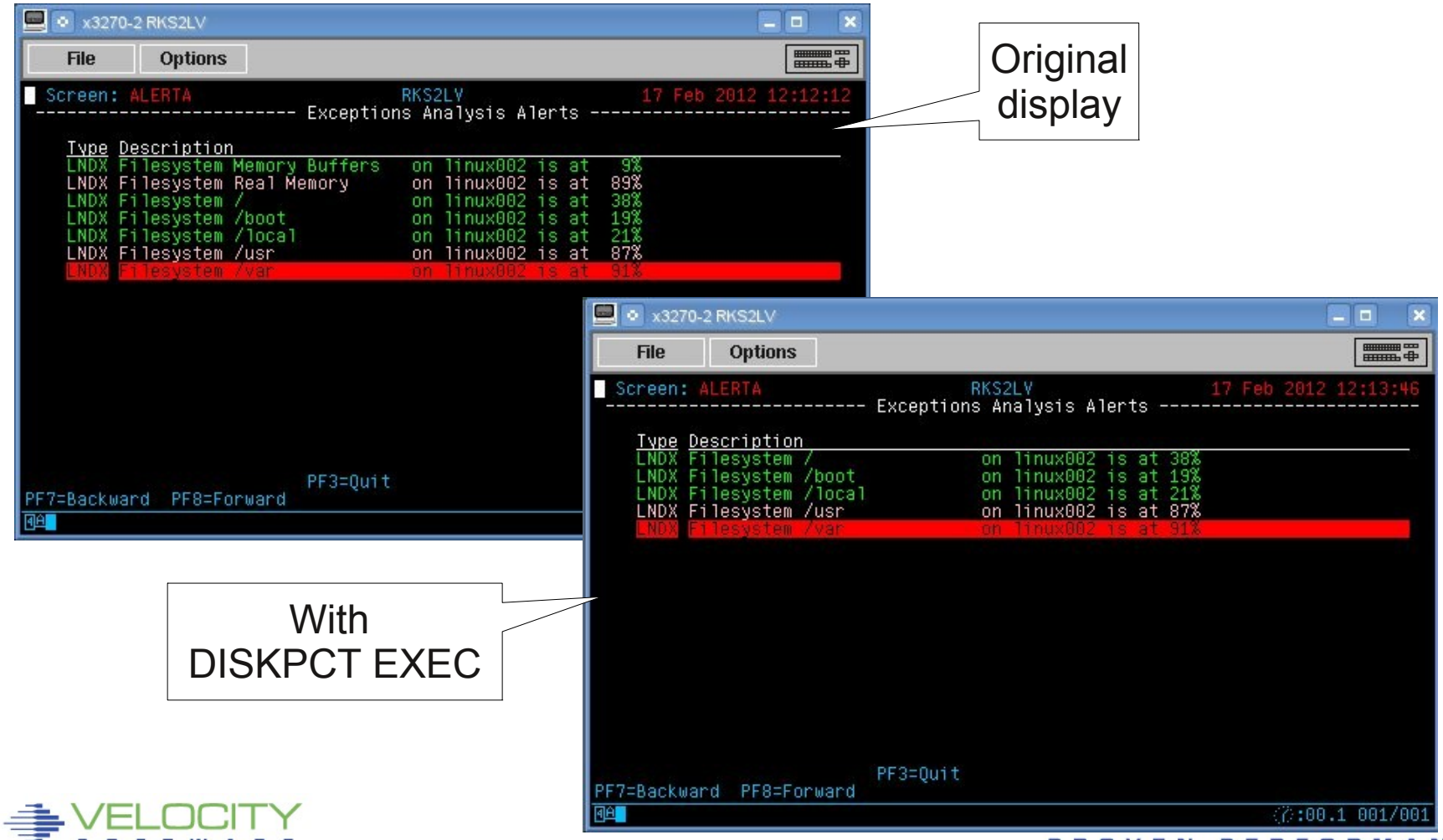
```
/* DISKPCT EXEC: Filter out memory or read-only filesystems */
parse arg node pct descr .

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

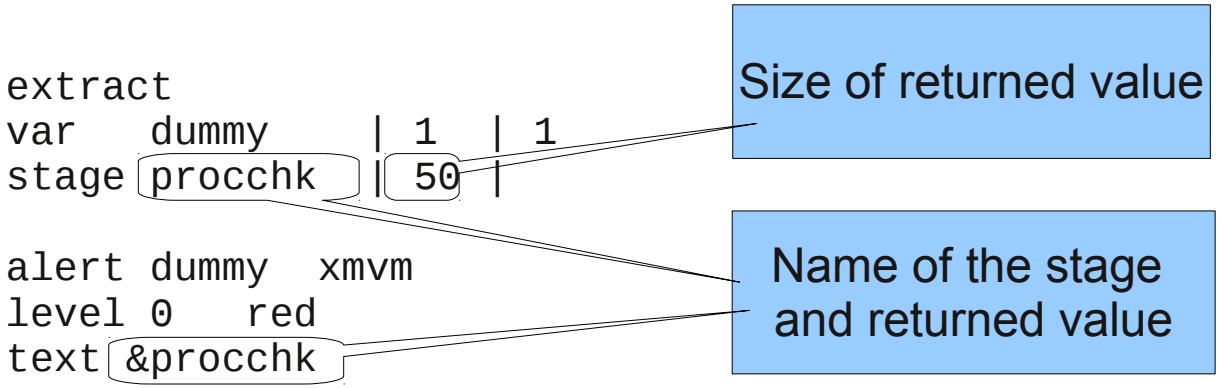
Advanced topics – External Processing

- **Results of function call**



Advanced topics – 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. | *:'
```

Advanced topics – More External Processing

- **Check for 'node down'**

```
extract
parms node *
criteria hstsys.iplyy > 0
var    nodenm   | 8  | tcpip.node
var    samples  | 1  | hstsys.samples
function nodedn | 2  | &nodenm &samples

alert nodedn lxup
level 0   red
text Node &nodenm is down
```

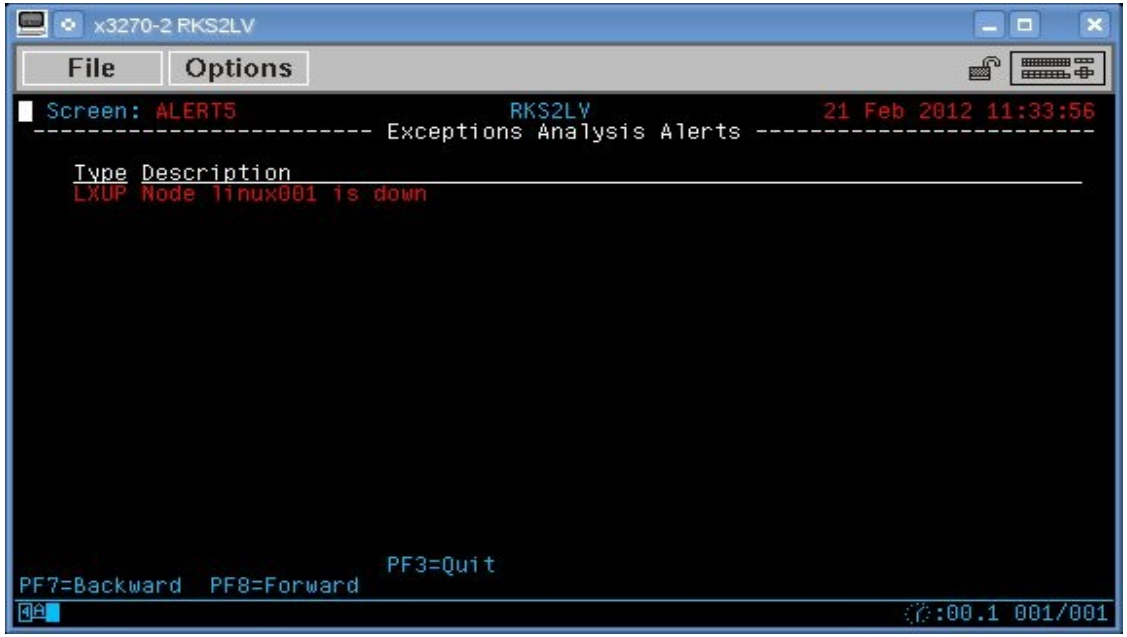
- **No value in 'samples' indicates down**
- **Level works with greater than only**
- **Function is required for further processing**

Advanced topics – More External Processing

```
extract
parms node *
criteria hstsys.iplyy > 0
var    nodenm   | 8 | tcpip.node
var    samples  | 1 | hstsys.samples
function nodedn | 2 | &nodenm &samples

alert nodedn lxup
level 0   red
text Node &nodenm is down
```

```
/* NODEDN: Alert function to check
   The interval samples for each
   passed node.  If samples is null,
   return 1 to the alert.  */
parse arg node samples

if samples >= 0 then
  return 0
else
  return 1
```

- Pass 'samples' from each node
- If >=0 return 0
- Otherwise return 1
- Level checks for >0



VELOCITY SOFTWARE

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