

ESAMON coding and usage

Richard Smrcina
Velocity Software, Inc.
Performance Workshop
June, 2017

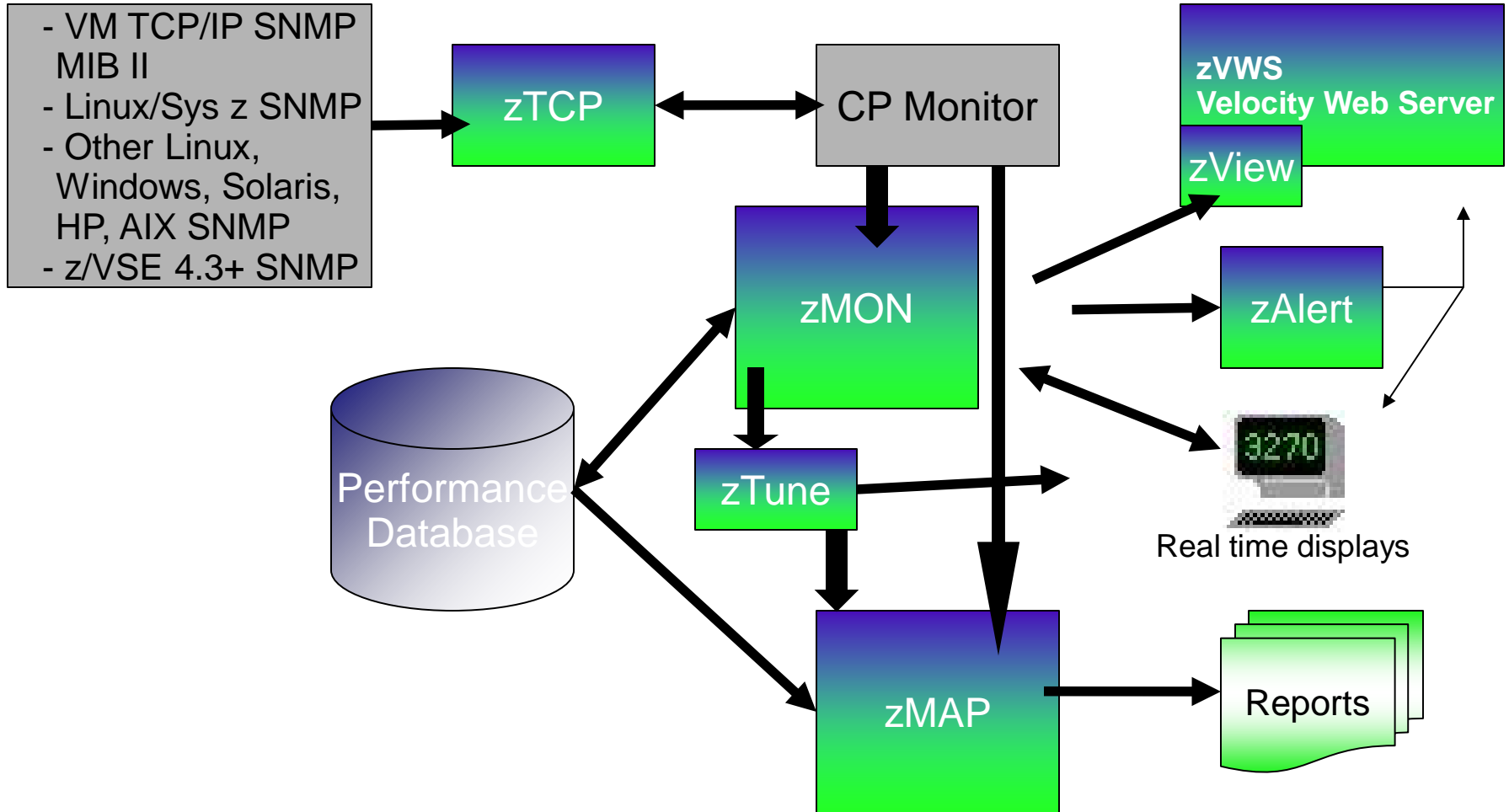
- **ESAMON**

- Predecessor to ZMON
- Module/EXEC
- Interface to the PDB
(Performance Database)

- **Interface to PDB**

- Provides an easy way to retrieve data via REXX
- A simple macro language

Map of Performance Products



- **The compendium of all data elements (PDR)**
 - Physicians Desk Reference
 - Program Data Reference
- **Just under 4000 items**
- **All of the data that can be viewed in zMon/zView**
- **PDR is arranged by data segment (or record)**
 - Multiple data fields in each segment
- **In some cases**
 - There are separate field names for ZMON and ZMAP

SYTASG: Auxiliary Storage (Global)

The following metrics are provided as a single observation for the entire system.

zMON - SYTASG		zMAP - SYTASG		Description
SAMPLES	flt	=	flt 1	Observations
CAL90FUL	flt	=	flt 0	Times paging area was 90 percent full
CAL91FUL	flt	=	flt 1	Times spooling area was 90 percent full
CALSLTA1	flt	=	flt 1	Paging slots allocated
CALSLTI1	flt	=	flt 0	Paging slots in use
SALPRFAV	flt	=	flt 0	Preferred paging slots allocated
SALPRFIU	flt	=	flt 0	Preferred paging slots in use
CALSLTA2	flt	=	flt 0	Spooling slots allocated
CALSLTI2	flt	=	flt 0	Spooling slots in use

- **zMON Extracts**

- Front ended with REXX
- Runs in an environment called ESAMON
 - All commands issued as ADDRESS ESAMON
- Requires a filetype of ESAMON
- Invoked with the ESAMON command
- ESAMON macro name is the first parameter
- Parameters can be passed to the macro
- Most useful subcommands
 - EXTRACT
 - RETRIEVE

- **zMON Extracts**

The subcommand operands are as follows:

```
EXTRACT FIELDS field [...]
[ACCT name | *]
[APPLICATION name *]
[CHANNEL number | range]
[CLASS classname | * | TOPUSERS | KEYUSERS]
[CPU processor | * | ALL | TOTAL]
[CU number | range | class]
[DCSS name | *]
[DEVICE number | range | class]
[FIRST number]
[FOR | FROM source]
[FOR | FROM time [ON date]]
[LIMIT limit]
[LISTING name]
[LPAR name | *]
[MDISKS | SEEKS]
[MAXIMUM count field]
[MINIMUM count field]
[NODE] name of server
[ON date]
[SCREEN screen]
[SPACE name | *]
[SYSTEM serial]
[TO | UNTIL time [ON date]]
[USER userid | * | TOPUSERS | KEYUSERS]
[WHILE field relation value [logical ...] ]
```

Figure 3-5 EXTRACT Subcommand Format

- **Test Extract**
 - ESAMON 'SAY' macro

```
esamon say sytasg.calslta1 sytasg.calslti1
SYTASG.CALSLTA1 = 600840
SYTASG.CALSLTI1 = 100103
```

```
esamon say tcpsys.node ucdsys.realsize ucdsys.totcpu / node
linux*
TCPSYS.NODE = linux001
UCDSYS.REALSIZE = 247236
UCDSYS.TOTCPU = 0.1072391
TCPSYS.NODE = linux002
UCDSYS.REALSIZE = 1020144
UCDSYS.TOTCPU = 0.1336309
```


Extracting Data

- **First extract**

```
PRES1      ESAMON      A1  F 80  Trunc=80  Size=5  Line=0  Col=1  Alt=0
          |...+....1....+....2....+....3....+....4....+....5....+...
===== * * * Top of File * * *
===== /* */
===== 'extract from interval' ,
===== 'fields sytasg.calslta1 sytasg.calslti1'
=====
===== pct = (sytasg.calslti1/sytasg.calslta1)*100
===== pct = format(pct,3,2)
===== say sytasg.calslta1 sytasg.calslti1 pct
===== * * * End of File * * *
```

- **Breaking it down**

```
'extract from interval' ,  
  'fields sytasg.calslta1 sytasg.calslti1'  
  
pct = (sytasg.calslti1/sytasg.calslta1)*100  
pct = format(pct,3,2)  
say sytasg.calslta1 sytasg.calslti1 pct
```

The 'extract' command begins a request for data

- **Breaking it down**

```
'extract from interval',  
  'fields sytasg.calslta1 sytasg.calslti1'  
  
pct = (sytasg.calslti1/sytasg.calslta1)*100  
pct = format(pct,3,2)  
say sytasg.calslta1 sytasg.calslti1 pct
```

'from' has multiple uses, it defines a data source or a timeframe

'interval' is the current minute, if not specified interval is assumed

- **Breaking it down**

```
'extract from interval' ,  
  'fields sytasg.calslta1 sytasg.calslti1'  
  
pct = (sytasg.calslti1/sytasg.calslta1)*100  
pct = format(pct,3,2)  
say sytasg.calslta1 sytasg.calslti1 pct
```

'fields' defines the data elements to be returned

CALSLTA1	flt	=	flt 1	Paging slots allocated
CALSLTI1	flt	=	flt 0	Paging slots in use

- **Breaking it down**

```
'extract from interval' ,  
  'fields sytasg.calslta1 sytasg.calslti1'
```

```
pct = (sytasg.calslti1/sytasg.calslta1)*100  
pct = format(pct,3,2)  
say sytasg.calslta1 sytasg.calslti1 pct
```

- Operate on the values
- Data returned from an extract is stored in a REXX variable of the same name

- **Invoke the EXEC**

```
esamon pres1  
600840 101506 16.89
```

The values are displayed based on the 'say' statement

- **Qualifiers can be used to target specific data**

CPU processor | * | ALL | TOTAL

Designates the number of an individual processor whose data is to be extracted. If an asterisk or ALL is specified then data is extracted for all CPUs. Specifying TOTAL causes the data extracted to be the sum for all processors. For example:

```
'extract from interval' ,  
  'cpu all' ,  
  'fields cpunumber sytprp.cpuutil'  
  
Do While rc = 0  
  say cpunumber sytprp.cpuutil  
  'retrieve'  
End
```

Extracting Data

- **Run it...**

```
esamon pres2  
0 11.26253  
1 11.98583
```

- **CPU 0...**

```
esamon pres3  
0 11.26253
```

- **CPU TOTAL**

```
esamon pres4  
96 23.24835
```


- **Filtering data**

```
'extract from interval' ,  
  'node *' ,  
  'fields tcpsys.node ucdsys.totcpu' ,  
  'while ucdsys.totcpu > 1'  
  
Do While rc = 0  
  say tcpsys.node ucdsys.totcpu  
  'retrieve'  
End
```

esamon pres5

BlakeNEW	1.053868
lxora12	4.077452
lxora12b	3.967409
mail	24.80019
NEWUSER1	1.075174
oracle	1.589446
REDHAT6X	1.663569
rhel7v	7.668565
sles12	1.533271
s11s2ora	4.640501
TDNORA1	1.260469
vpnbrc	2.216602
vpnbrz	4.687281
vpnz	2.981948
ZSXLBRO	1.628485

- **Combine the two**

```
'extract from interval' ,
'node lx*' ,
'while hstmem.used > 0' ,
'fields tcpip.node' ,
    'hstmem.allocun' ,
    'hstmem.used' ,
    'hstmem.size' ,
    'hstmem.desc'

say 'Node      AlUn      Used      Size      Free Pct' ,
'Description'
say '=====  
'
do while rc=0
  If left(hstmem.desc,1) <> '/' Then Do
    'retrieve'
    Iterate
    End
    ndnm = tcpip.node
    size = hstmem.size / 1024
    usd = hstmem.used / 1024
    line=left(ndnm,8) ,
        format(hstmem.allocun,6,0) ,
        format(usd,10,0) ,
        format(size,10,0)
    line=line format(size - usd,10,0) ,
        format((usd/size)*100,3,0) ,
        left(hstmem.desc,25)
    say line
    'retrieve'
end
```

Extracting Data

- Combine the two

```
'extract from interval' ,  
'node lx*' ,  
'while hstmem.used > 0' ,  
'fields tcpip.node' ,  
'hstmem.allocat
```

```
esamon dpct  
Node      AlUn      Used      Size      Free Pct Description  
=====  =====  =====  =====  =====  ===  =====  
say 'Node  lxdb2001      4096      1678      2308      630  73  /  
'Desc  lxdb2001      4096      976      2308      1332  42  /opt  
say '==== lxora12      4096      1589      2158      568  74  /  
'==== lxora12      4096      38      146      108  26  /boot  
do while lxora12      4096      6365      11087      4722  57  /opt  
If left lxora12      4096      10144      16113      5969  63  /opt/oracle/oradata  
'ret  lxora12      4096      1589      2158      568  74  /  
Iter  lxora12b     4096      38      146      108  26  /boot  
End   lxora12b     4096      6365      11087      4722  57  /opt  
ndnm = lxora12b     4096      10144      16113      5969  63  /opt/oracle/oradata  
sze =  lxora12b     4096      2310      2310      0  100  /  
usd =  lxora12b     4096      0      499      499  0  /dev  
line=1 lxora12b     4096      1226      2310      1084  53  /usr  
f  lxora12b     4096      1226      2310      1084  53  /usr  
f  lxora12b     4096      1226      2310      1084  53  /usr  
f  lxora12b     4096      1226      2310      1084  53  /usr  
line=1
```

```
format((usd/sze)*100,3,0) ,  
left(hstmem.desc,25)  
say line  
'retrieve'  
end
```

- **Selecting data from a historical timeframe**

```
/* */  
'extract from 10:00 on 17/06/09 to 10:10 on 17/06/09' ,  
  'cpu total' ,  
  'fields stoptime sytprp.cpuutil'
```

```
Do While rc = 0  
  say stoptime format(sytprp.cpuutil,3,1)  
  'retrieve'  
End
```

```
esamon pres2  
100100 38.4  
100200 28.0  
100300 26.8  
100400 25.7  
100500 27.2  
100600 76.1  
100700 28.2  
100800 27.5  
100900 25.9  
101000 25.2  
101100 26.4
```

- **Selecting data from a historical timeframe**

```
'extract from 06:00 on 17/06/01 to 06:10 on 17/06/01' ,  
'cpu total' ,  
'fields stoptime sytprp.cpuutil'
```

```
Do While rc = 0  
  say stoptime format(sytprp.cpuutil,3,1)  
  'retrieve'  
End
```

```
esamon pres2  
061500 34.8
```

Questions



Rich Smrcina
Velocity Software, Inc
rich@velocitysoftware.com