



# Support of KVM Hypervisor Using VCB Customization

---

The purpose of this chapter is to present a possible solution on how to achieve device and fault management support for the KVM Hypervisor in the Prime Network. This can be accomplished via VCB Customizations.

Topics include:

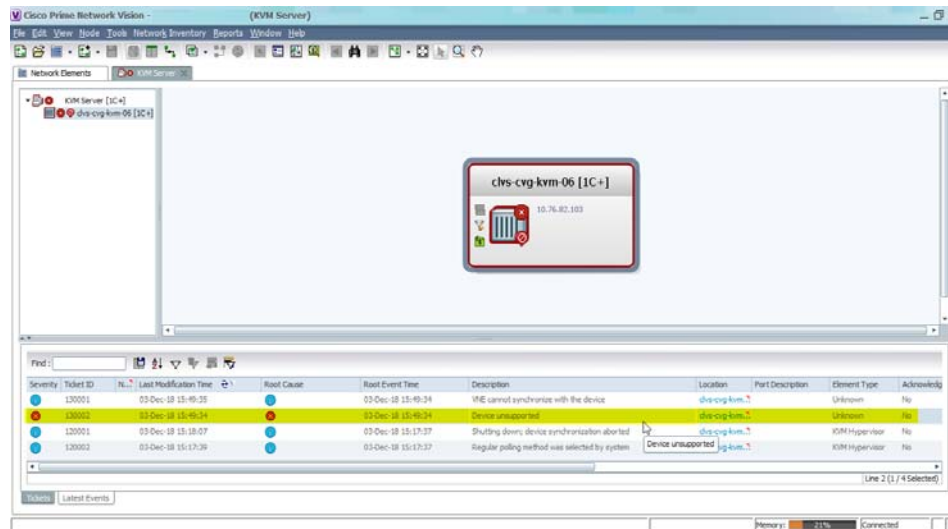
- [Problem Statement, page D-1](#)
- [Background, page D-1](#)
- [Solution, page D-2](#)
- [VCB Customization for Device Type Support, page D-2](#)
- [VCB Customization for Supporting Traps, page D-7](#)
- [Adding Soft Properties for KVM Hypervisor, page D-16](#)

## Problem Statement

- Currently Prime Network does not support KVM Hypervisor. Therefore, when the user models it, Prime Network discovers it as an 'Unsupported' VNE.
- In case Prime Network supports KVM Hypervisor via VCB, then there occurs a problem with unsupported traps. If KVM Hypervisor sends a trap (for example, VM suspend traps), then Prime Network will consider it an unsupported trap and will display it under 'Standard' traps in Prime Network Events Application.

## Background

Prime Network detects specific device type using its SysOID. Prime Network discovers the KVM Hypervisor as Unsupported because it does not have its SysOID and raises 'Device Unsupported' ticket.



To overcome this problem, user should configure the SysOID of KVM Hypervisor via VCB Customization.

Prime Network needs to know the `snmpTrapOID` of a trap to categorize it. Else, the trap will be shown as unsupported 'Standard' trap.

VCB allows configuration of end-to-end trap information in order to support it in Prime Network.

## Solution

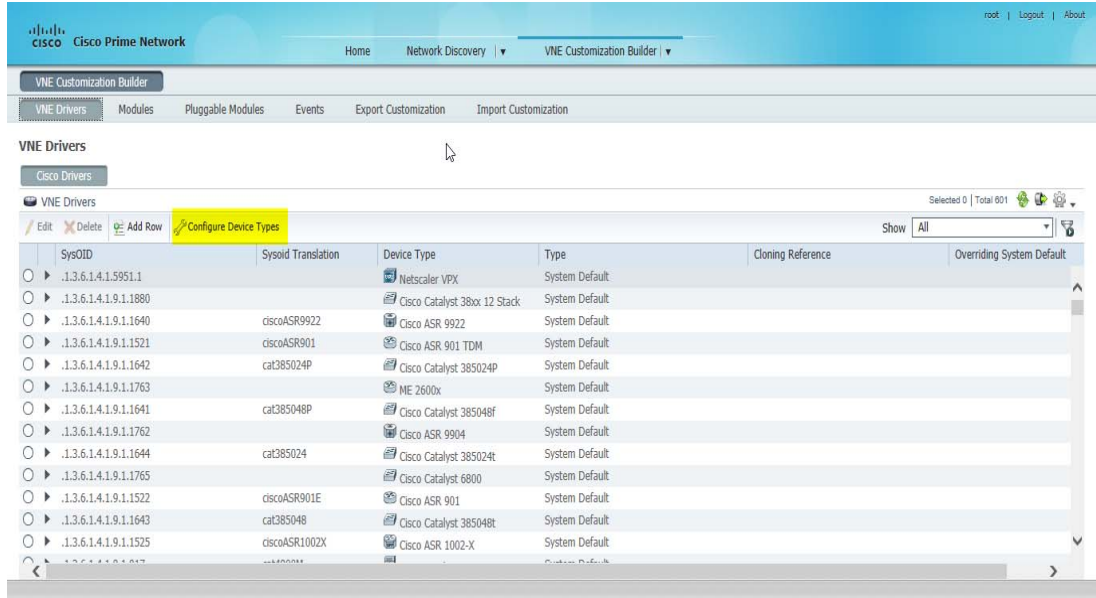
VNE Customization Builder (VCB) Tool of Prime Network provides solutions to overcome both the problems of unsupported device types and unsupported traps via simple customization. KVM Hypervisor will be supported using GenericUVNE template and modeled as a generic device. No virtualization specific inventory will be discovered. KVM-specific traps can then be supported using VCB Events customization.

Let us see about those customizations in detail.

## VCB Customization for Device Type Support

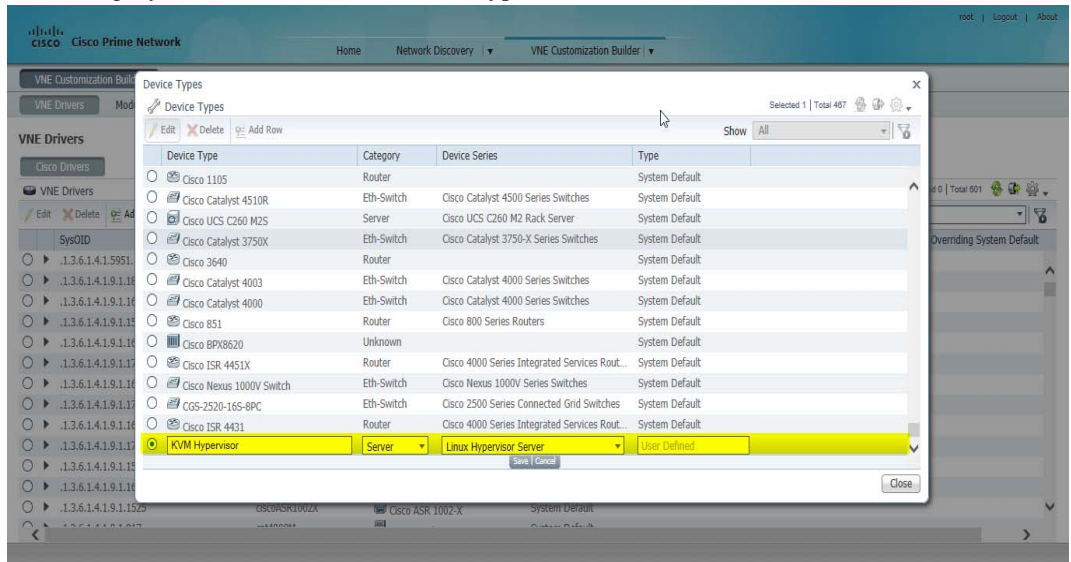
**Step 1** Login to VCB. Go to *VNE Customization Builder > VNE Drivers*

**Step 2** Click *Configure Device Types*.



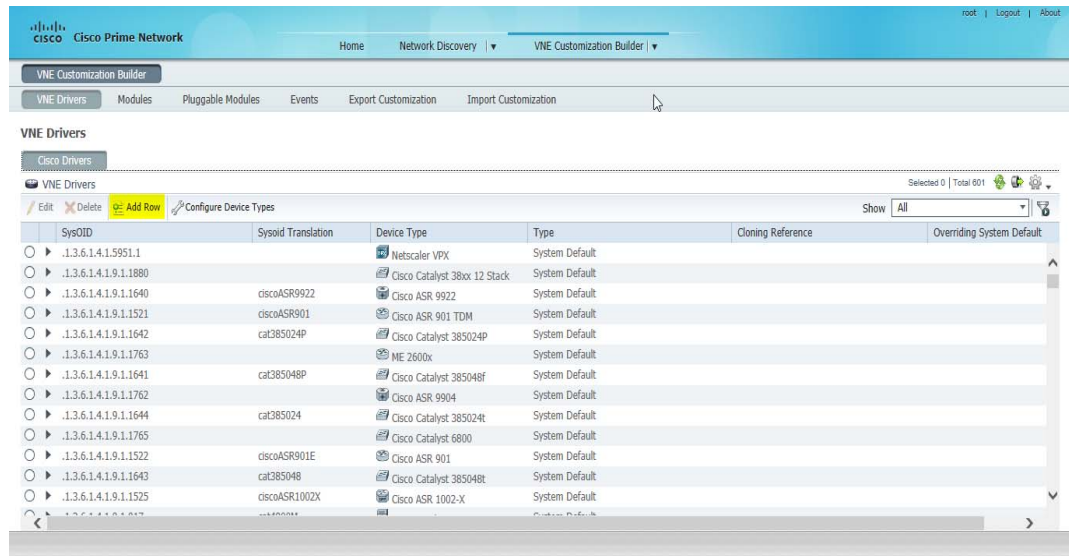
**Step 3** Click *Add Row*.

**Step 4** Choose Category as 'Server'. Define Device Type and Device Series. Click *Save*.



**Step 5** Newly added device type appears. Click *close*.

**Step 6** Click 'Add Row' in VNE Drivers.



**Step 7** Enter the *device SysOID* under SysOID.

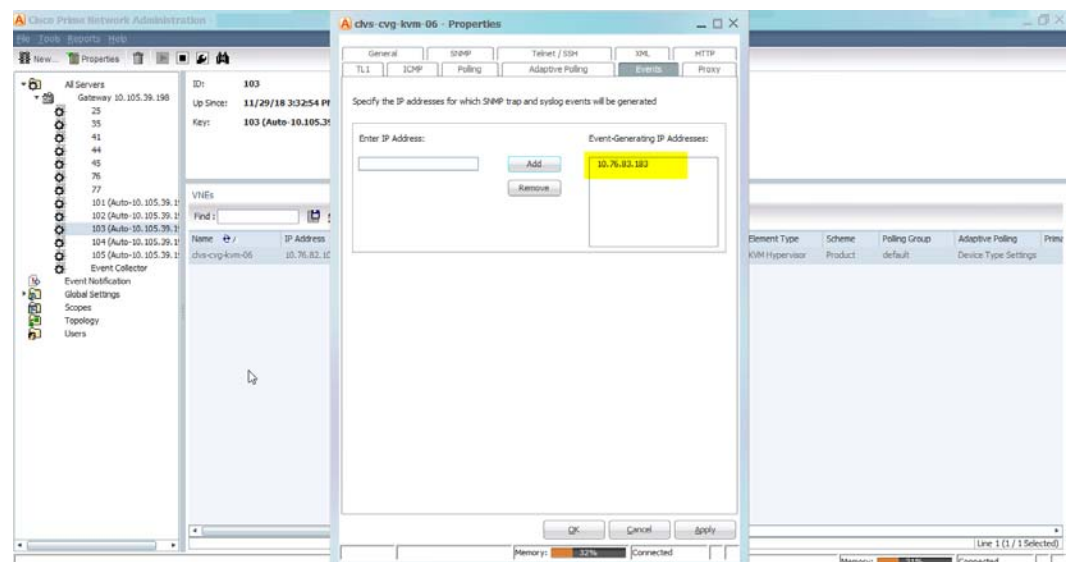
Steps to Get SysOID via PN:

- Model KVM Server. Choose 'Auto Detect' as Type and 'Product' as Scheme.

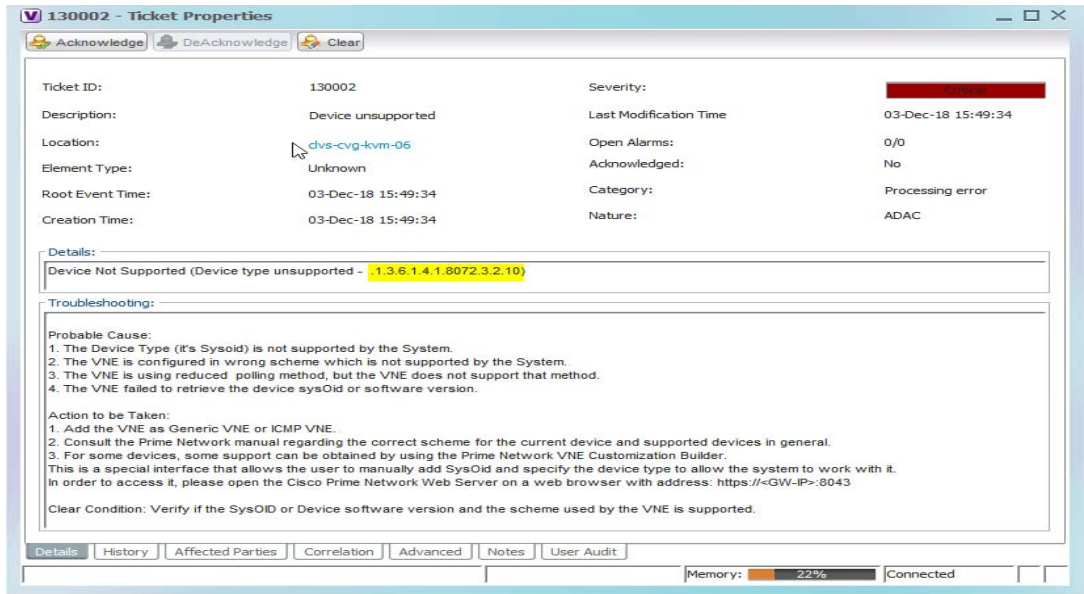


**Note**

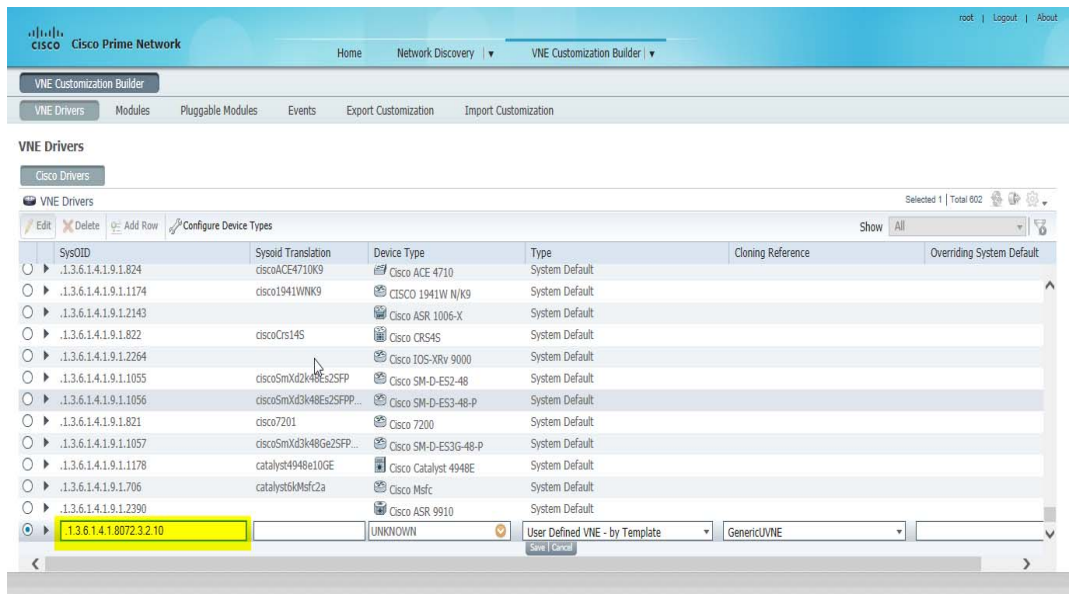
In case if multiple IPs are configured in KVM Server, identify the IP address that is used to send traps and add that IP address under Events Tab & Restart VNE. This is required only if the IP used to send traps is different from the management/VNE IP.



- Add the VNE to Map. Open the properties for Device Unsupported Ticket to get the SysOID of the device.
- Copy the SysOID from the details section.



d. Now Paste the *SysOID* here.



**Step 8** Choose *Server > KVM > KVM Hypervisor* as Device Type.

## VCB Customization for Device Type Support

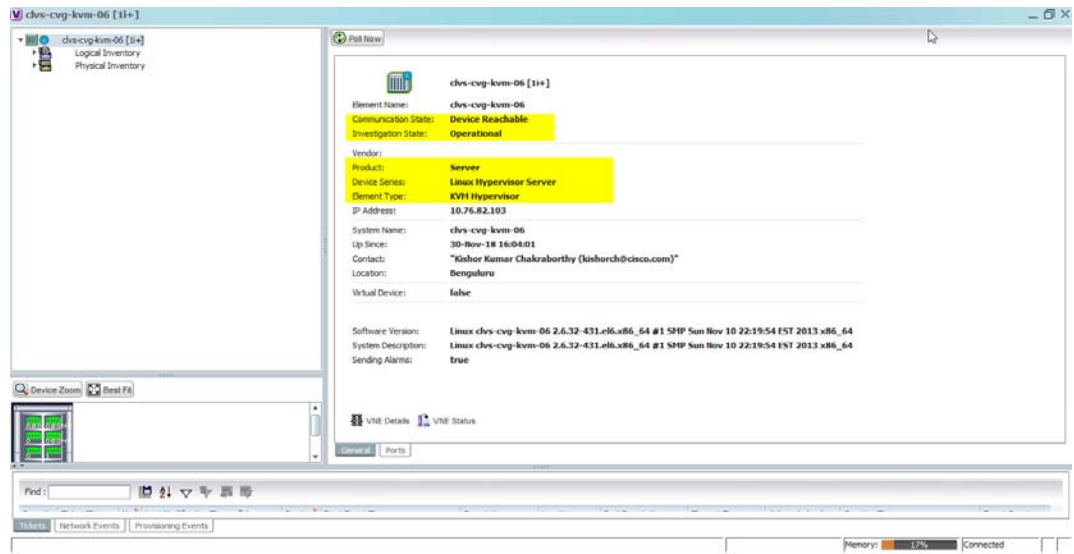
The screenshot shows the Cisco Prime Network VNE Customization Builder interface. The 'VNE Drivers' section is active, displaying a table of drivers. A search box is open, showing 'Search Results' for the term 'Server'. The search results list several categories: Access Server, Carrier Routing Server, Server (highlighted), Server Farm, Virtual Server, and WWW Server. The table below shows various SysOIDs and their corresponding Device Types.

SysOID	Sysoid Translation	Device Type
.1.3.6.1.4.1.9.1.824	discoACE4710K9	Cisco ACE 4710
.1.3.6.1.4.1.9.1.1174	disco1941WNNK9	CISCO 1941W N/K9
.1.3.6.1.4.1.9.1.2143		Cisco ASR 1006-X
.1.3.6.1.4.1.9.1.822	discoCrs145	Cisco CRS45
.1.3.6.1.4.1.9.1.2264		Cisco IOS-XRv 9000
.1.3.6.1.4.1.9.1.1055	discoSmXd3k48Es2SFP	Cisco SM-D-ES2-48
.1.3.6.1.4.1.9.1.1056	discoSmXd3k48Es2SFP...	Cisco SM-D-ES3-48-P
.1.3.6.1.4.1.9.1.821	disco7201	Cisco 7200
.1.3.6.1.4.1.9.1.1057	discoSmXd3k48Ge2SFP...	Cisco SM-D-ES3G-48-P
.1.3.6.1.4.1.9.1.1178	catalyst4948e10GE	Cisco Catalyst 4948E
.1.3.6.1.4.1.9.1.706	catalyst6kMsf2a	Cisco Msf
.1.3.6.1.4.1.9.1.2390		Cisco ASR 9910

The screenshot shows the Cisco Prime Network VNE Customization Builder interface. The 'VNE Drivers' section is active, displaying a table of drivers. The newly added SysOID is highlighted in yellow. The table below shows various SysOIDs and their corresponding Device Types and Types.

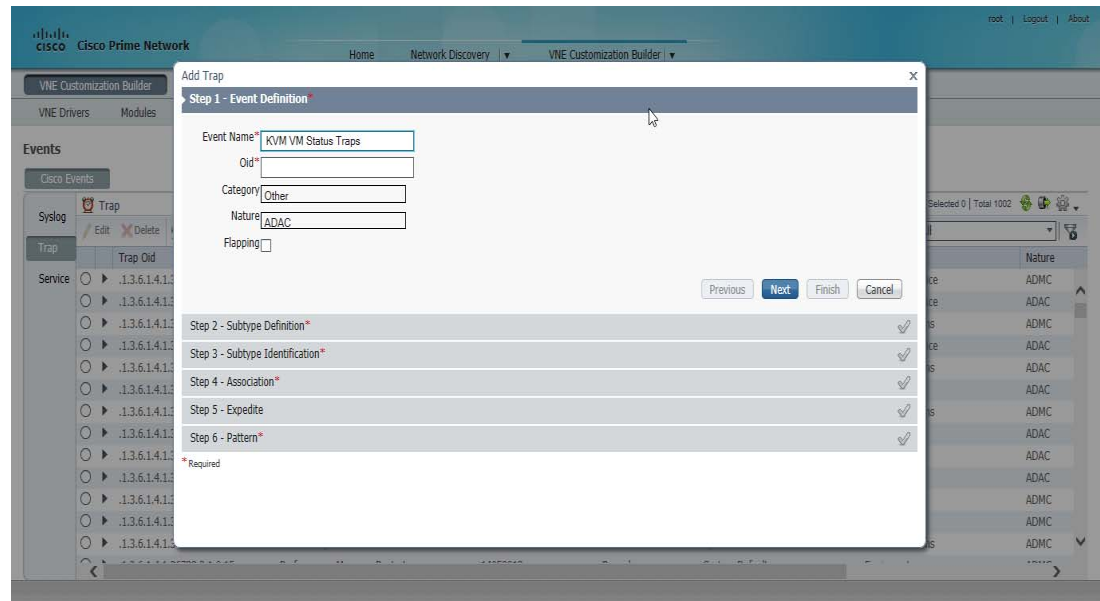
SysOID	Sysoid Translation	Device Type	Type	Cloning Reference	Overriding System Default
.1.3.6.1.4.1.9.1.824	discoACE4710K9	Cisco ACE 4710	System Default		
.1.3.6.1.4.1.9.1.1174	disco1941WNNK9	CISCO 1941W N/K9	System Default		
.1.3.6.1.4.1.9.1.2143		Cisco ASR 1006-X	System Default		
.1.3.6.1.4.1.9.1.822	discoCrs145	Cisco CRS45	System Default		
.1.3.6.1.4.1.9.1.2264		Cisco IOS-XRv 9000	System Default		
.1.3.6.1.4.1.9.1.1055	discoSmXd2k48Es2SFP	Cisco SM-D-ES2-48	System Default		
.1.3.6.1.4.1.9.1.1056	discoSmXd3k48Es2SFP...	Cisco SM-D-ES3-48-P	System Default		
.1.3.6.1.4.1.9.1.821	disco7201	Cisco 7200	System Default		
.1.3.6.1.4.1.9.1.1057	discoSmXd3k48Ge2SFP...	Cisco SM-D-ES3G-48-P	System Default		
.1.3.6.1.4.1.9.1.1178	catalyst4948e10GE	Cisco Catalyst 4948E	System Default		
.1.3.6.1.4.1.9.1.706	catalyst6kMsf2a	Cisco Msf	System Default		
.1.3.6.1.4.1.9.1.2390		Cisco ASR 9910	System Default		
.1.3.6.1.4.1.8072.3.2.10		KVM Hypervisor	User Defined VNE - by Template	GenericVNE	

- Step 9** Click *Save*. Newly added SysOID details appears.
- Step 10** Restart VNE.
- Step 11** Device is supported and VNE enters into operational state.



## VCB Customization for Supporting Traps

- Step 1** Login to VCB. Go to *VNE Customization Builder > Events > Traps*.
- Step 2** Click *Add Row*.
- Step 3** Enter *Event Name*.

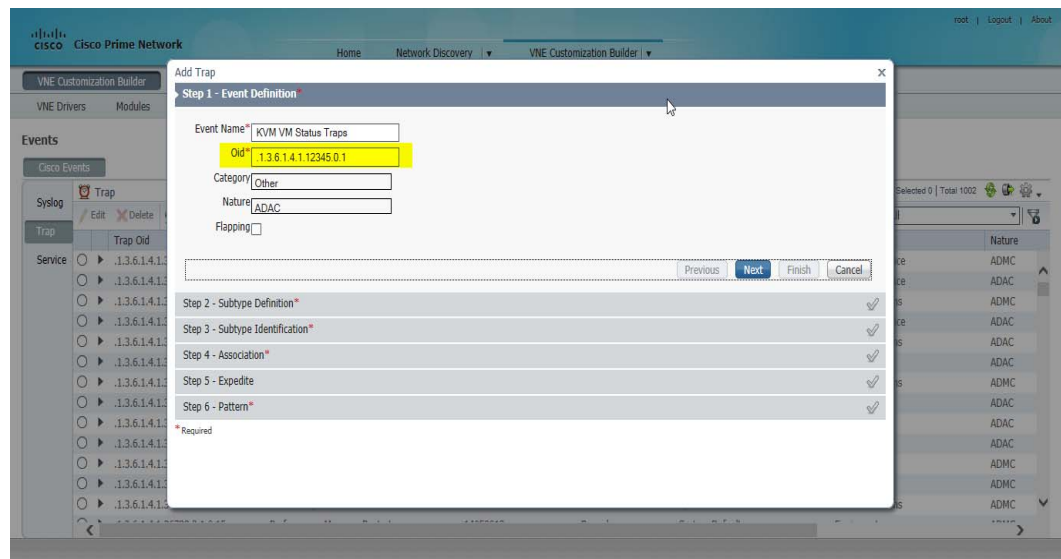


- Step 4** Get SNMPTrapOID value from received unsupported traps and map it to Oid:
  - a. Go to *Standard* Tab of Event Vision
  - b. Double click on any of the received unsupported Traps to see Event Properties.

- c. Go to *Trap* Tab.
- d. Right Click on the snmpTrapOID row and click Properties.
- e. Copy the *SNMP value*.

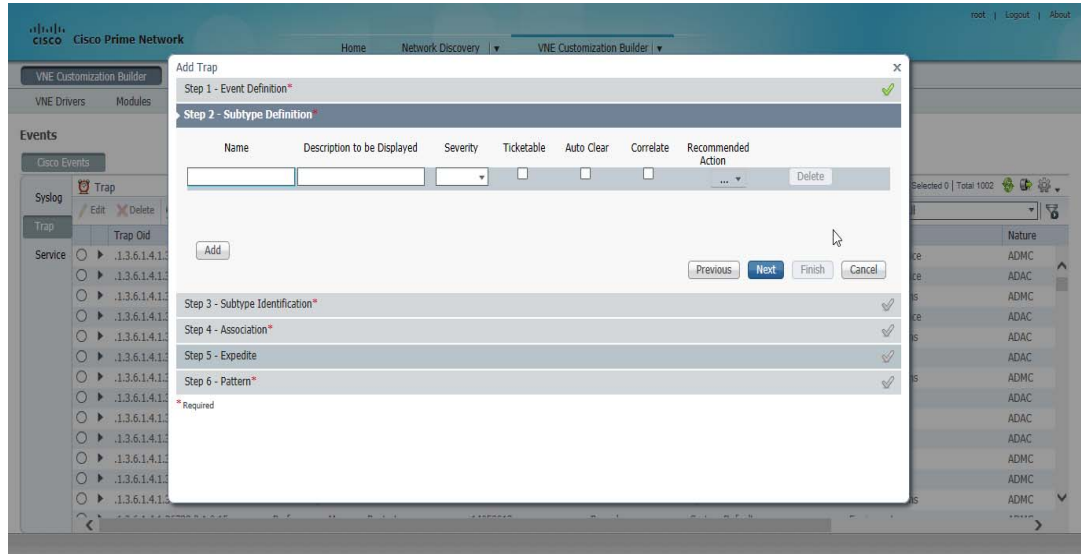


- f. Paste the copied *SNMP Value* in 'Oid' text box. Then click 'Next' button.





**Step 5** Enter the Events details such as description, severity, ticketable or not in Step 2.



a. Determine the event types from libvirt MIB using *libvirtGuestState* property.

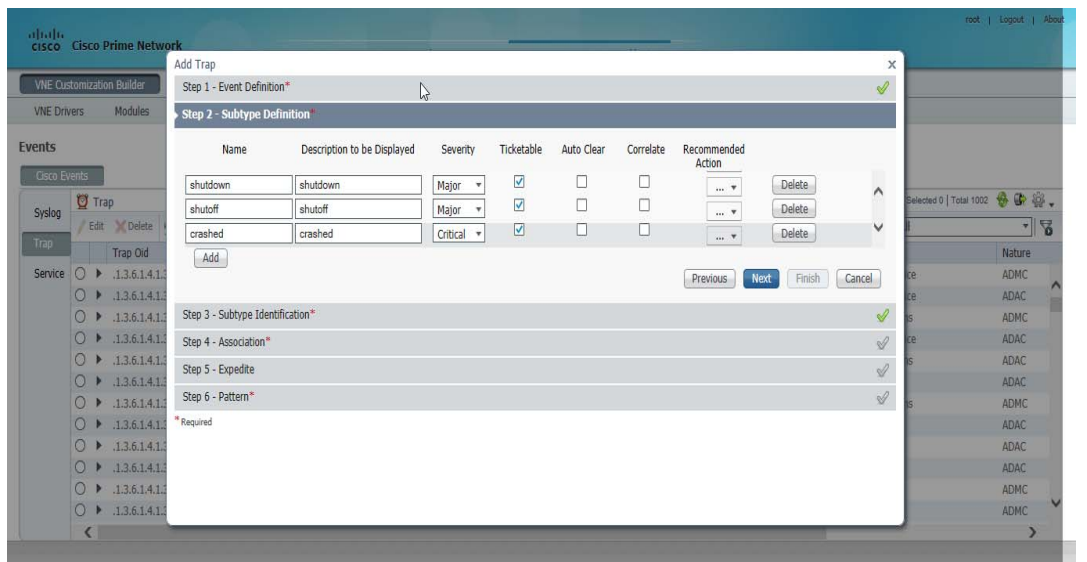
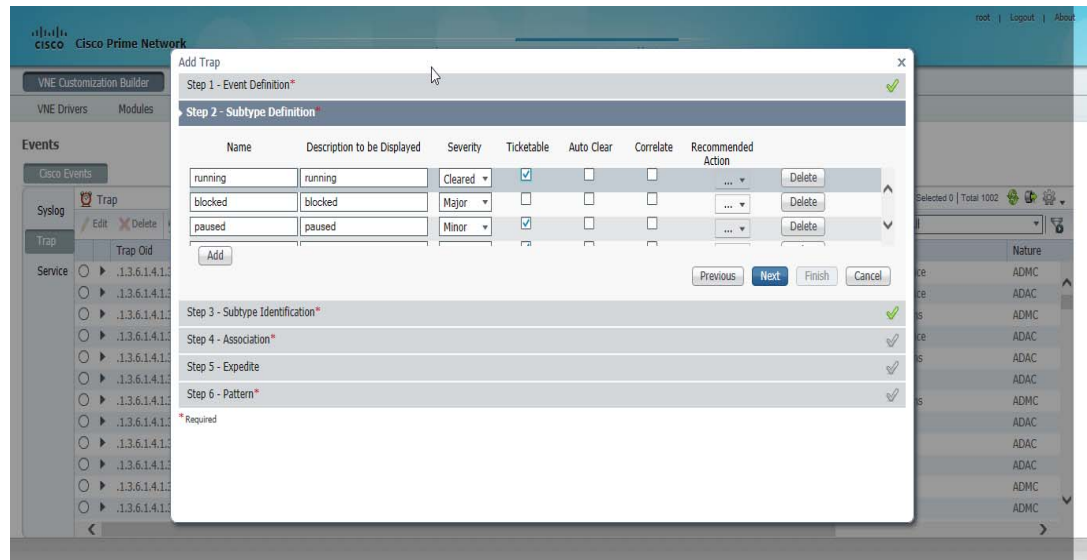
```
libvirtGuestUUID OBJECT-TYPE
SYNTAX      UUID
MAX-ACCESS not-accessible
STATUS      current
DESCRIPTION
    "The UUID of the virtual guest."
 ::= { libvirtGuestEntry 1 }

libvirtGuestName OBJECT-TYPE
SYNTAX      OCTET STRING
MAX-ACCESS read-only
STATUS      current
DESCRIPTION
    "Name of active virtual guest."
 ::= { libvirtGuestEntry 2 }

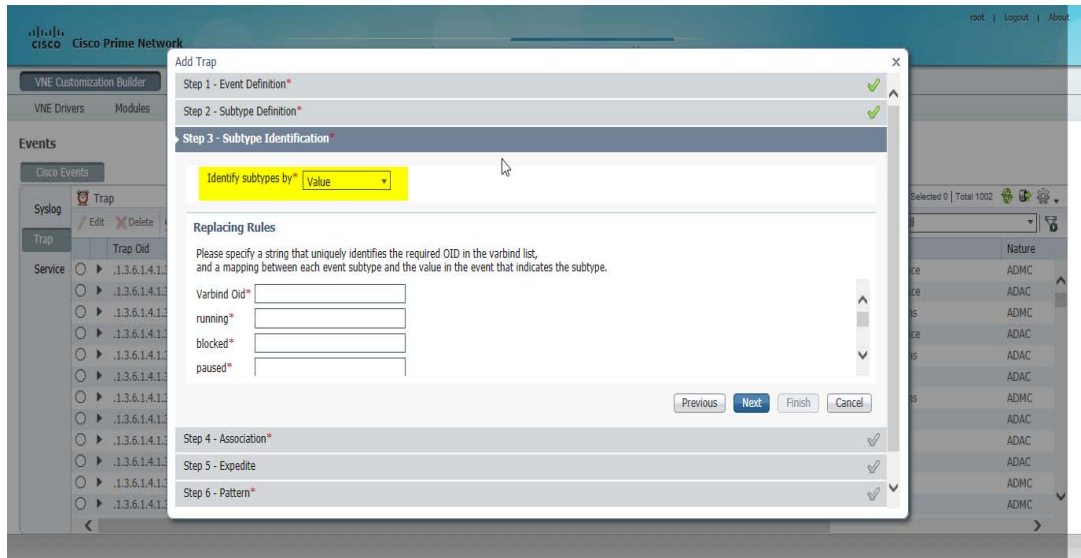
libvirtGuestState OBJECT-TYPE
SYNTAX      INTEGER {
    running(1),
    blocked(2),
    paused(3),
    shutdown(4),
    shutoff(5),
    crashed(6)
}
MAX-ACCESS read-create
STATUS      current
DESCRIPTION
    "Current state of the active guest. This column is writeable, and only
    these state transitions are allowed to be sed by SNMPSET request:
        running -> paused
        paused -> running
        running -> shutdown
    A new row can be created with this column set to 'running' or 'paused'
    or with no value of this column at all ('running' is then assumed).
    The state transition is not instant, e.g. it can take a while until a
    virtual machine is shut down and the agent may show the machine as
    'running' for some time."
```

b. Configure all the VM status in Step 2 and Click *Next*.

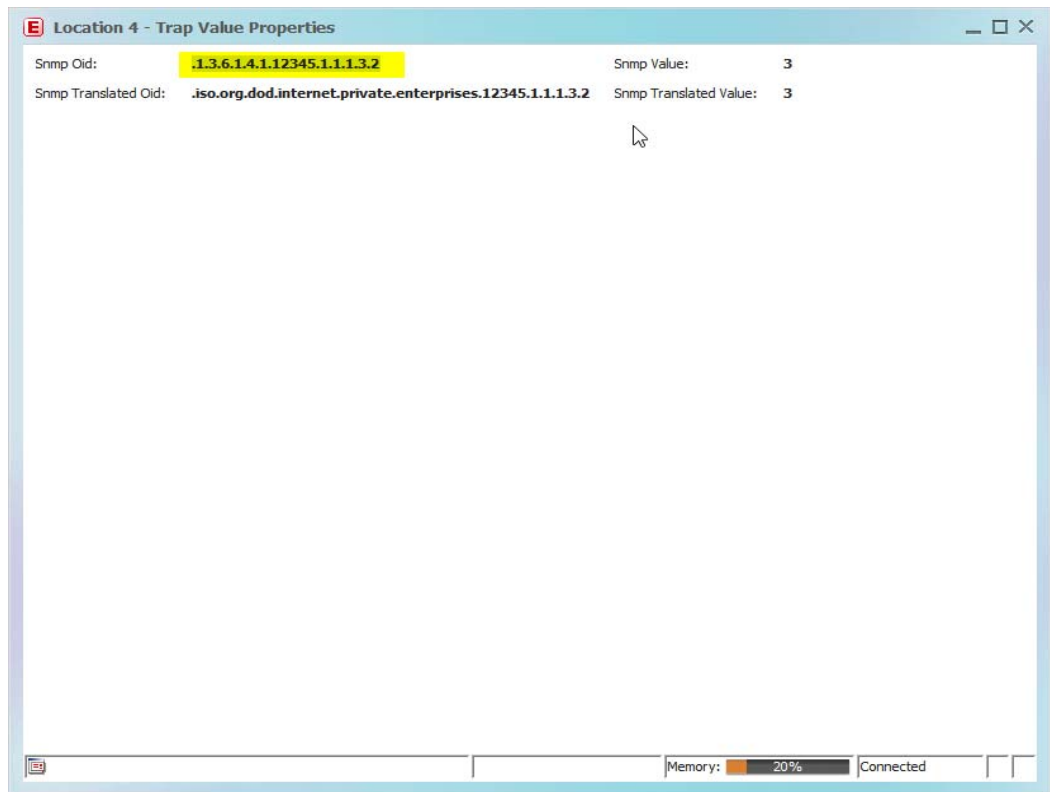
VCB Customization for Supporting Traps

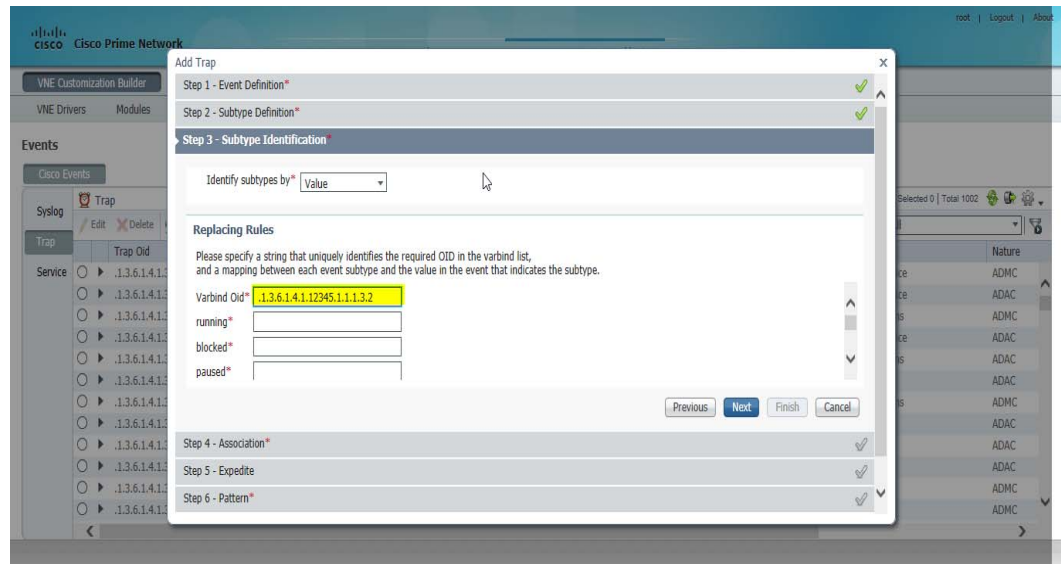


**Step 6** Select 'value' for Identify subtypes by following the below steps:

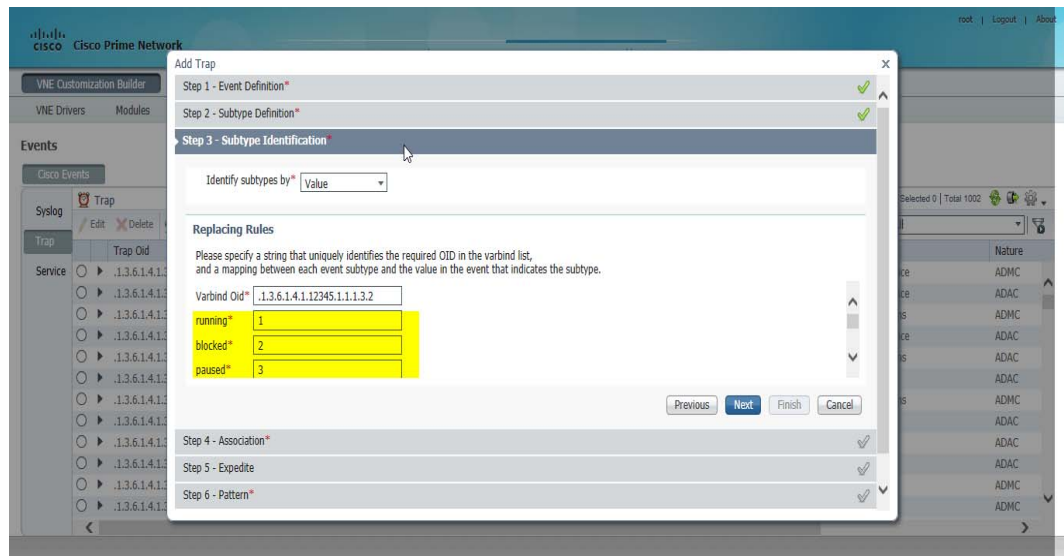


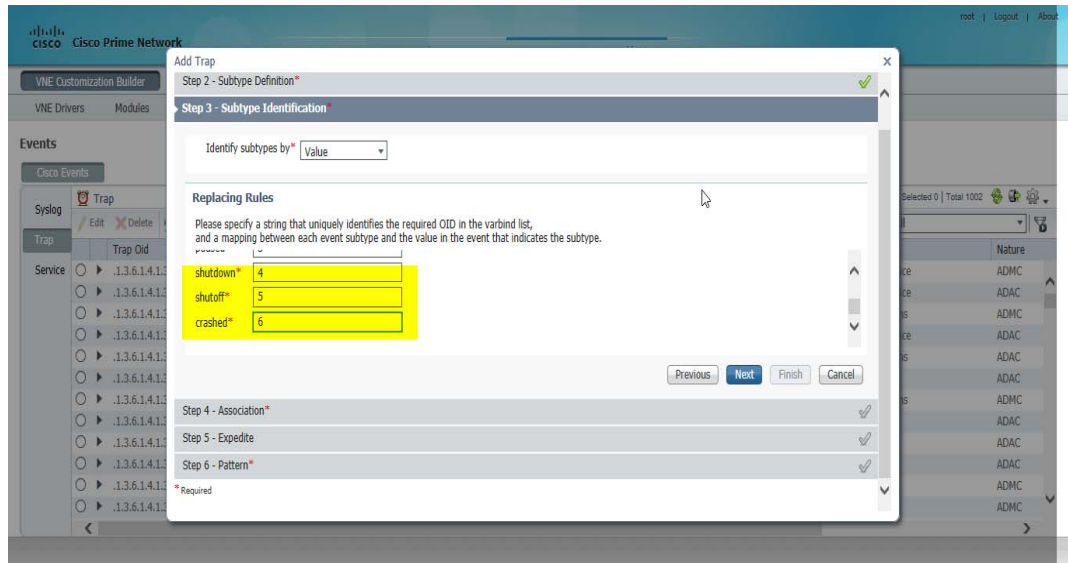
- a. Get the VM status Varbind Oid and map here. Go to Standard traps of Event Vision. Double click on any of the received unsupported traps. Go to Trap tab.
- b. Right Click on '.1.3.6.1.4.1.12345.1.1.1.3.2' OID which maps to VM status and select *properties*.
- c. Copy the *Snmp Oid* and map it to *Varbind Oid*.





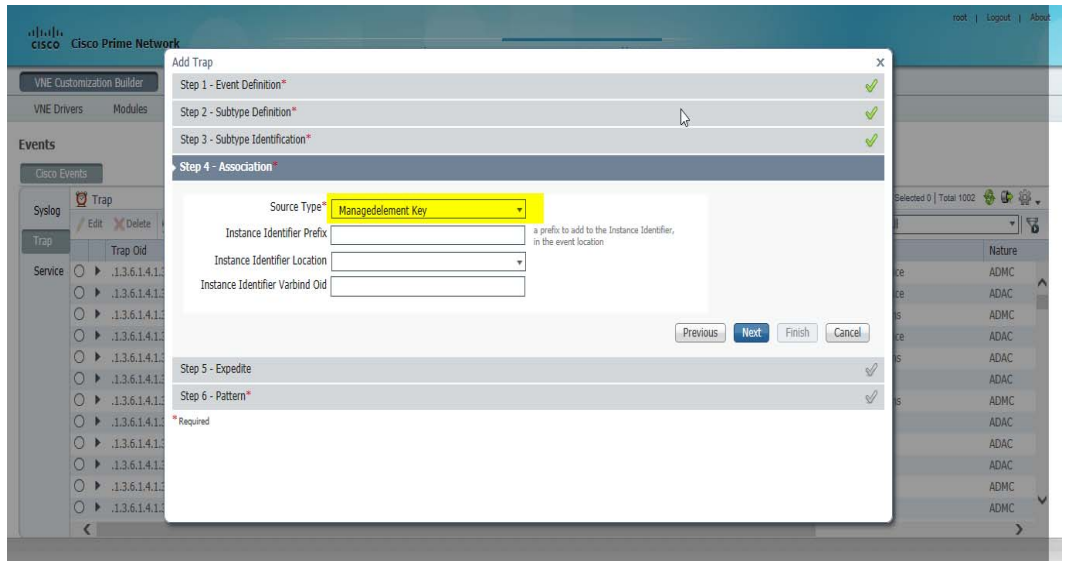
- d. Map the *SnmP Value* which will uniquely identify each of the VM status and Click *Next*.



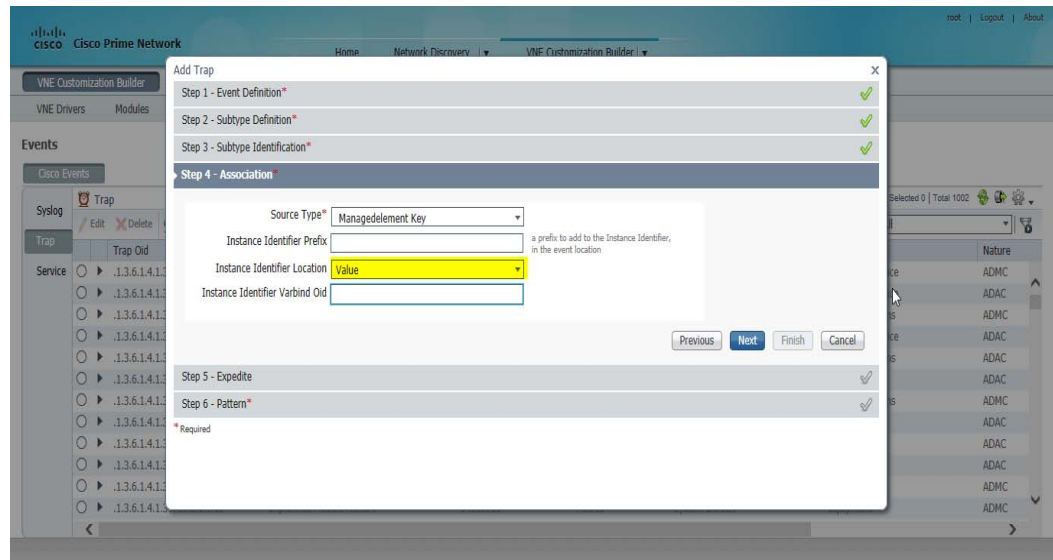


**Step 7** Map the Location details in Association Step.

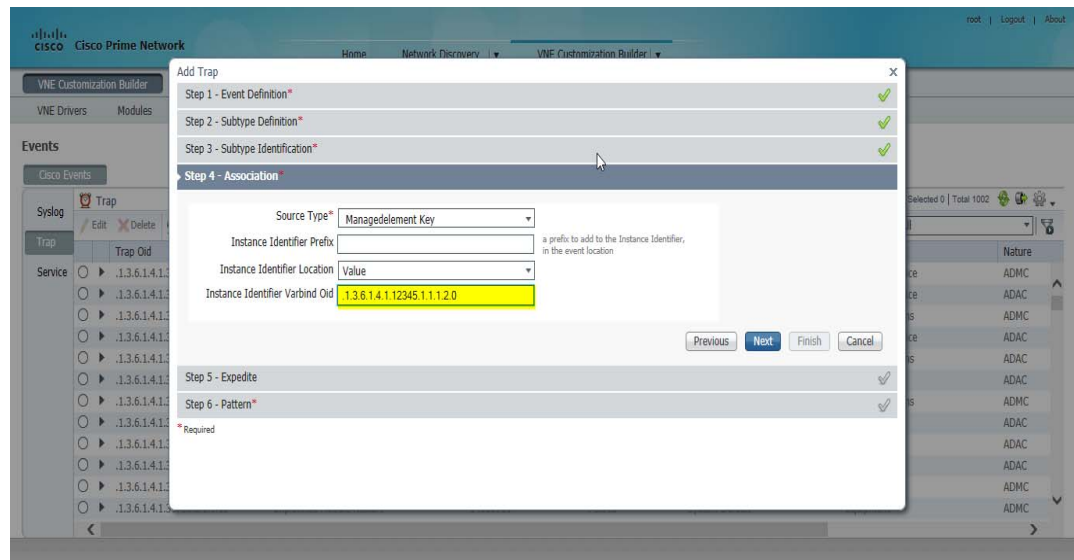
- a. Choose 'Managedelement Key' as Source Type.



- b. Choose 'Value' as Instance Identifier Location.

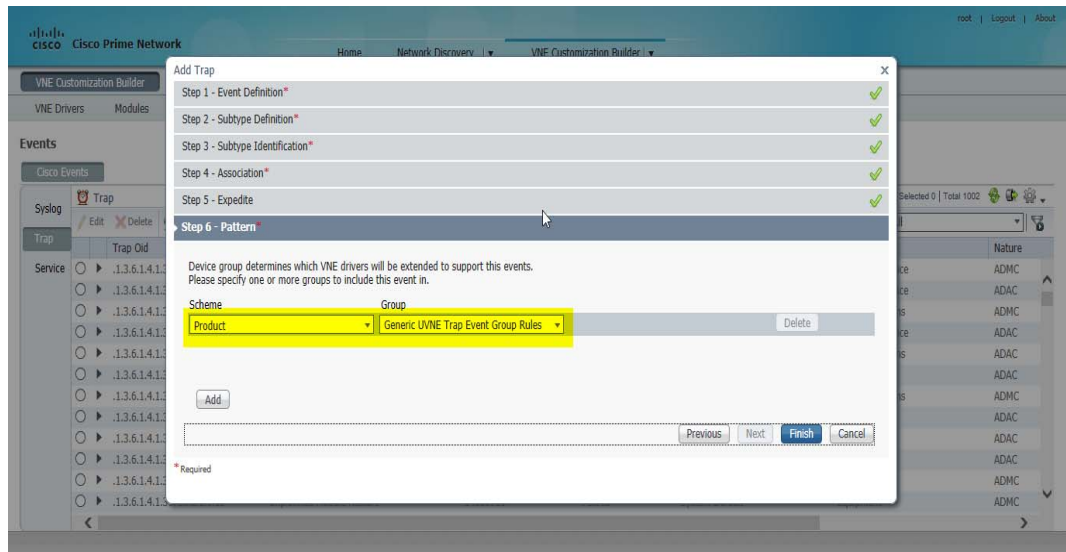


- c. Get the Location Varbind Oid from received traps and map here. Go to Standard traps of Event Vision. Double click on any of the received unsupported traps. Go to Trap tab.
- d. Right Click on `.1.3.6.1.4.1.12345.1.1.1.2.0` Oid which maps to location and click select properties.
- e. Copy the Snmp Oid and map it to Instance Identifier Varbind Oid and Click 'Next'.



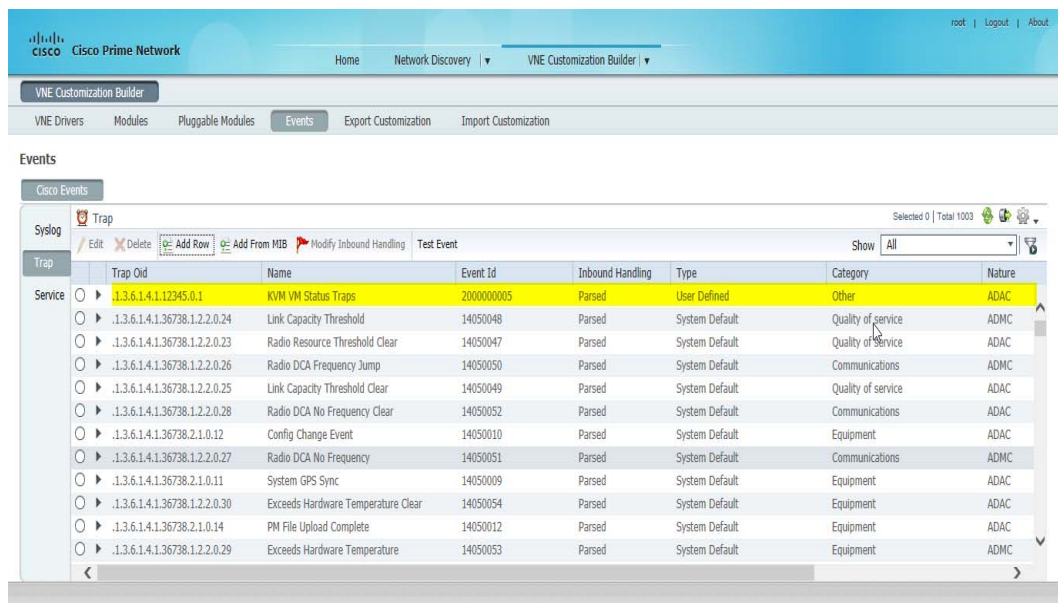
**Step 8** Click 'Next' on Expedite Step.

**Step 9** Choose 'Product' as Scheme and 'Generic UVNE Trap Event Group Rules' as Group.



**Step 10** Click 'Finish'.

**Step 11** Newly added trap oid appears.



**Step 12** Generate VM Suspend traps:

```
[root@clvs-cvg-kvm-06 ~]# virsh suspend cvg-vm07-lnx
Domain cvg-vm07-lnx suspended
```

**Step 13** PN receives suspended traps as V2 Trap.

Severity	Event ID	Time	Description	Location	Element Type	Alarm ID	Ticket ID	Causing Event ID	Duplication Count	Reduction Count
Warning	13572	04-Dec-18 12:54:43	paused	dis-cvg-kvm-06: cvg-vm074ns	KVM Hypervisor	170009	170009		1	1
Warning	13615	04-Dec-18 12:54:43	paused	dis-cvg-kvm-06: cvg-vm074ns	KVM Hypervisor	170009	170009		1	1
Warning	13486	04-Dec-18 12:54:43	paused	dis-cvg-kvm-06: cvg-vm074ns	KVM Hypervisor	170009	170009		1	1
Warning	13529	04-Dec-18 12:54:43	paused	dis-cvg-kvm-06: cvg-vm074ns	KVM Hypervisor	170009	170009		1	1
Warning	13400	04-Dec-18 12:54:43	paused	dis-cvg-kvm-06: cvg-vm074ns	KVM Hypervisor	170009	170009		1	1
Warning	13357	04-Dec-18 12:54:43	paused	dis-cvg-kvm-06: cvg-vm074ns	KVM Hypervisor	170009	170009		1	1
Warning	13314	04-Dec-18 12:54:43	paused	dis-cvg-kvm-06: cvg-vm074ns	KVM Hypervisor	170009	170009		1	1
Warning	13443	04-Dec-18 12:54:43	paused	dis-cvg-kvm-06: cvg-vm074ns	KVM Hypervisor	170009	170009		1	1
Info	13271	04-Dec-18 12:51:56	running	dis-cvg-kvm-06: cvg-vm074ns	KVM Hypervisor	170008	170008		1	1
Info	13228	04-Dec-18 12:51:56	running	dis-cvg-kvm-06: cvg-vm074ns	KVM Hypervisor	170007	170007		1	1
Info	13142	04-Dec-18 12:51:56	running	dis-cvg-kvm-06: cvg-vm074ns	KVM Hypervisor	170005	170005		1	1
Info	13185	04-Dec-18 12:51:56	running	dis-cvg-kvm-06: cvg-vm074ns	KVM Hypervisor	170006	170006		1	1
Info	12970	04-Dec-18 12:51:56	running	dis-cvg-kvm-06: cvg-vm074ns	KVM Hypervisor	170001	170001		1	1
Info	13099	04-Dec-18 12:51:56	running	dis-cvg-kvm-06: cvg-vm074ns	KVM Hypervisor	170004	170004		1	1
Info	13058	04-Dec-18 12:51:56	running	dis-cvg-kvm-06: cvg-vm074ns	KVM Hypervisor	170003	170003		1	1
Info	13013	04-Dec-18 12:51:56	running	dis-cvg-kvm-06: cvg-vm074ns	KVM Hypervisor	170002	170002		1	1

## Adding Soft Properties for KVM Hypervisor

Cisco Prime Network enables user to add required soft properties for KVM Hypervisor. Prime Network has provision to display soft properties in both scalar (using snmp or Telnet/SSH) and table format (using snmp).

This feature can be used to enhance the Generic UVNE Model with device specific inventory.

Let us see examples for adding Soft Properties for KVM Hypervisor using above stated methods.

### Scalar Type

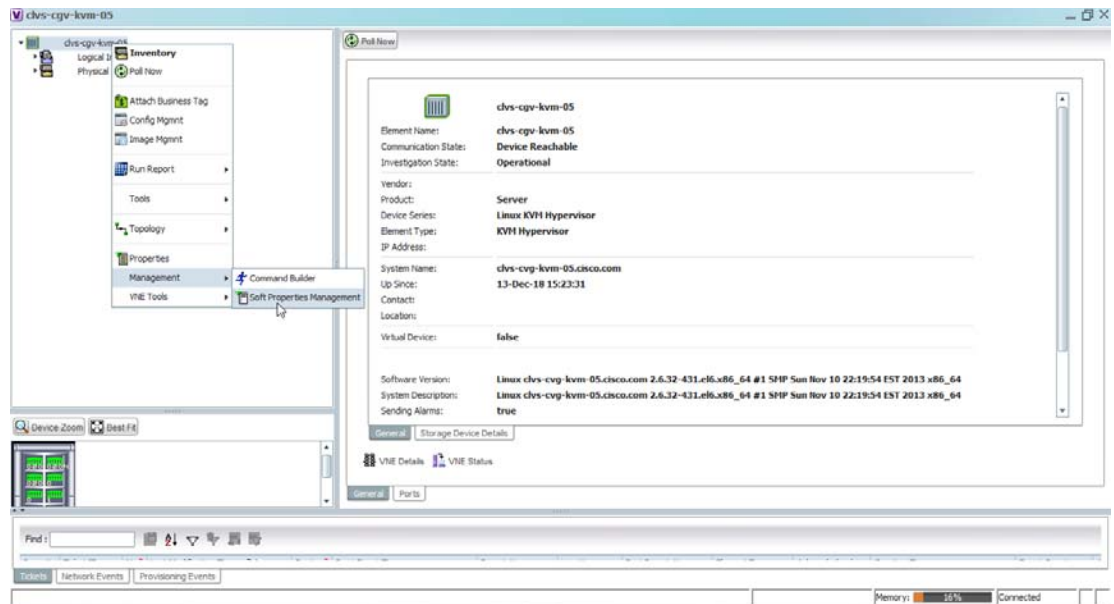
This topic lists the steps to add soft properties in scalar format using snmp or Telnet/SSH.

#### Using SNMP

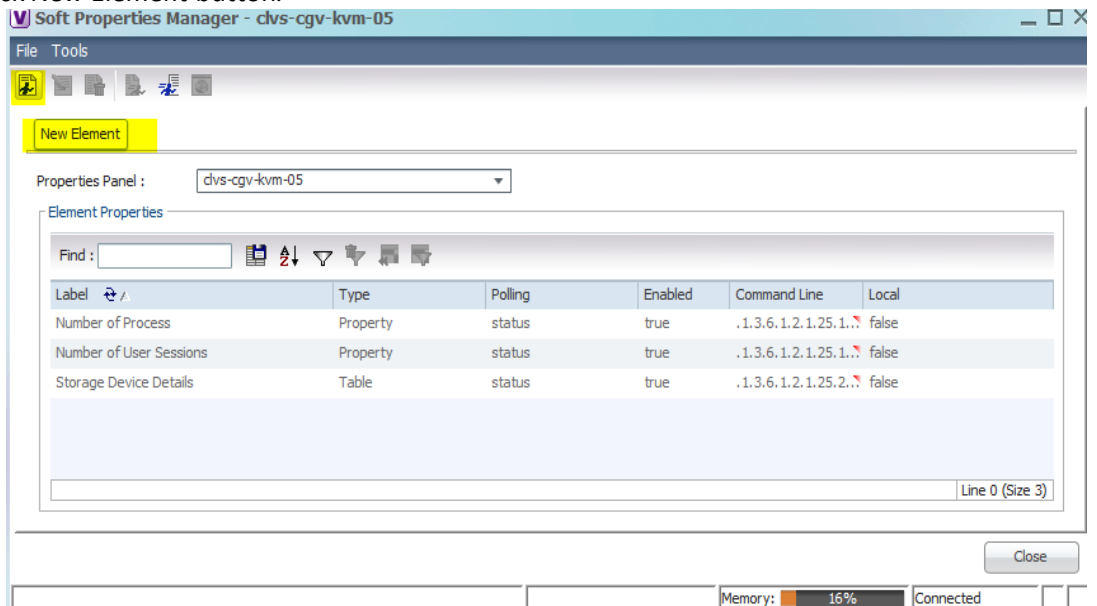
Let us consider host resource mibs *hrSystemNumUsers* and *hrSystemProcesses* as required scalar soft properties for KVM Hypervisor.

**Step 1** Right click *Managed Element > Management > Soft Properties Management*.





**Step 2** Click *New Element* button.



**Step 3** In *General* tab, enter *Name* and *Label*. Select '*property*' as *Type* and '*status*' as *Polling Rate*.

General Parsing TCA Alarms

Name : NumberofProcess

Label : Number of Process

Description :

Type : Property

Polling Rate : status

Enabled :

OK Cancel Debug

**Step 4** In Parsing Tab, select, use *SNMP get (OID)*. Enter *OID* for *hrSystemProcesses*, and click *OK*.

General Parsing TCA Alarms

Use SNMP get(OID)

Use Telnet/SSH

.1.3.6.1.2.1.25.1.6.0

Index	Operation
Empty	

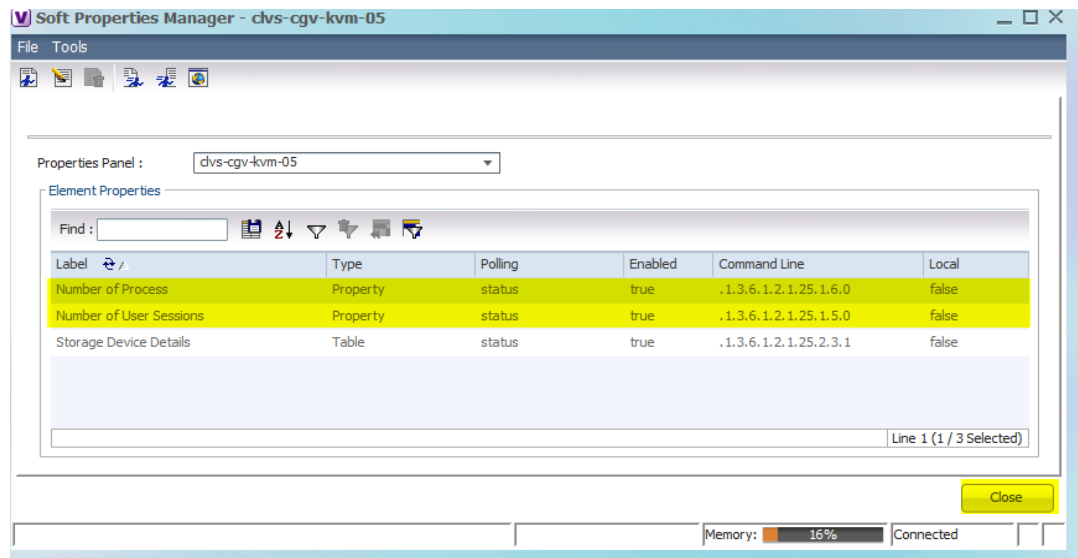
↑ Re-Order ↓

Add Edit Delete Test

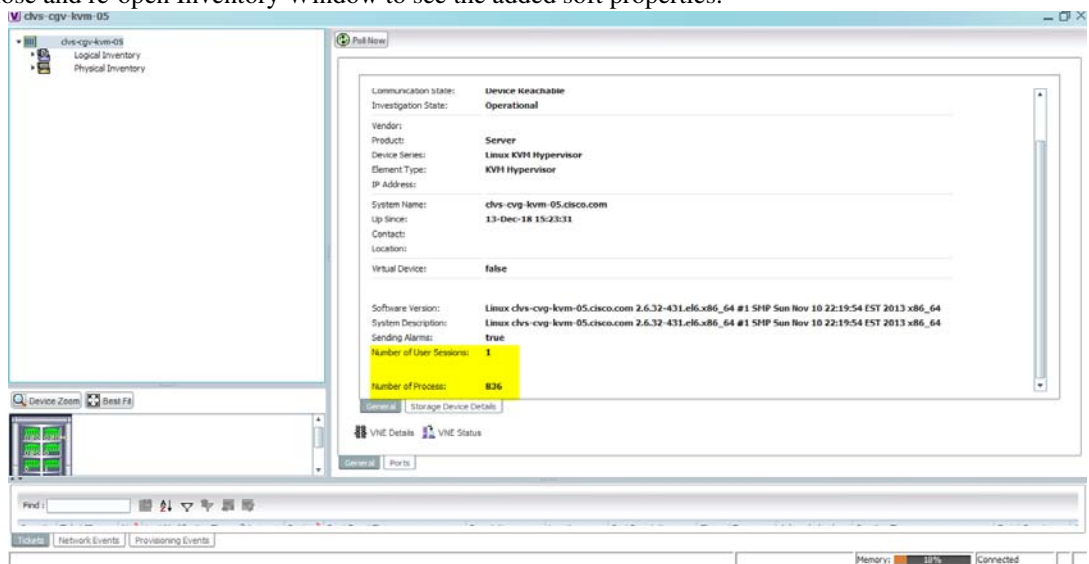
OK Cancel Debug

**Step 5** Repeat Steps 2, 3 and 4 to add *hrSystemNumUsers* soft property.

**Step 6** Click *Close*, if done with addition of all required soft properties.



**Step 7** Close and re-open Inventory Window to see the added soft properties.



## Using Telnet/SSH

### Prerequisites

To enable Telnet/SSH protocol for KVM Hypervisor Server, update the device avm registry with below configuration using runreg tool command and restart the VNE.

```
runRegTool.sh -gs localhost set 127.0.0.1 <DeviceAVM>/agents/da/<VNE
Name>/ips/<VNEIP>/protocols/telnet/connection/class
"com.sheer.net.protocols.telnet.SocketConnection"
```

```
runRegTool.sh -gs localhost set 127.0.0.1 <DeviceAVM>/agents/da/<VNE
Name>/ips/<VNEIP>/protocols/telnet/connection/implicitly-ask-for-pty "true"
```

```
runRegTool.sh -gs localhost add 127.0.0.1 <DeviceAVM>/agents/da/<VNE
Name>/ips/<VNEIP>/protocols/telnet/connection/transport

runRegTool.sh -gs localhost set 127.0.0.1 <DeviceAVM>/agents/da/<VNE
Name>/ips/<VNEIP>/protocols/telnet/connection/transport/pty-support enable

runRegTool.sh -gs localhost set 127.0.0.1 <DeviceAVM>/agents/da/<VNE
Name>/ips/<VNEIP>/protocols/telnet/connection/transport/pty-type xterm
```

For instance, if vne is modelled under *avm666* with VNE name *-clvs-cvg-kvm-06* and device IP *-10.76.82.103*, then update *avm666.xml* file with the below commands:

```
runRegTool.sh -gs localhost set 127.0.0.1
avm666/agents/da/clvs-cvg-kvm-06/ips/10.76.82.103/protocols/telnet/connection/class
"com.sheer.net.protocols.telnet.SocketConnection"

runRegTool.sh -gs localhost set 127.0.0.1
avm666/agents/da/clvs-cvg-kvm-06/ips/10.76.82.103/protocols/telnet/connection/explicitly-a
sk-for-pty "true"

runRegTool.sh -gs localhost add 127.0.0.1
avm666/agents/da/clvs-cvg-kvm-06/ips/10.76.82.103/protocols/telnet/connection/transport

runRegTool.sh -gs localhost set 127.0.0.1
avm666/agents/da/clvs-cvg-kvm-06/ips/10.76.82.103/protocols/telnet/connection/transport/pt
y-support enable

runRegTool.sh -gs localhost set 127.0.0.1
avm666/agents/da/clvs-cvg-kvm-06/ips/10.76.82.103/protocols/telnet/connection/transport/pt
y-type xterm
```

After running the above commands, the corresponding registry should have the following entries:

```
<key name="telnet">
  <entry name="port">22</entry>
  <entry name="protocoltype">2</entry>
  <key name="connection">
    <entry name="explicitly-ask-for-pty">>true</entry>
    <entry name="class">com.sheer.net.protocols.ssh.Ssh2Connection</entry>
  </key>
  <key name="algorithms"></key>
  <key name="transport">
    <entry name="pty-support">enable</entry>
    <entry name="pty-type">xterm</entry>
  </key>
```

### Procedure

Let us consider *Libvirt Version Info* as required scalar soft property to be displayed for KVM Hypervisor using Telnet/SSH:

- 
- Step 1** Right-click *Managed Element > Management > Soft Properties Management*.
  - Step 2** Click *New Element* button.
  - Step 3** In General tab, enter *Name* and *Label*. Select '*property*' as Type and '*System*' as Polling Rate.

**Edit Soft Property**

General Parsing TCA Alarms

Name : LibvirtVersionInfo  
 Label : Libvirt Version Info  
 Description :  
 Type : Property  
 Polling Rate : system  
 Enabled :

OK Cancel Debug

**Step 4** In Parsing tab, select *Use Telnet/SSH*. Enter the Telnet command and click *OK*.

**Edit Soft Property**

General Parsing TCA Alarms

Use SNMP get(OID)  
 Use Telnet/SSH

virsh -v

Index	Operation
0	Header And Footer

Line 0 (Size 1)

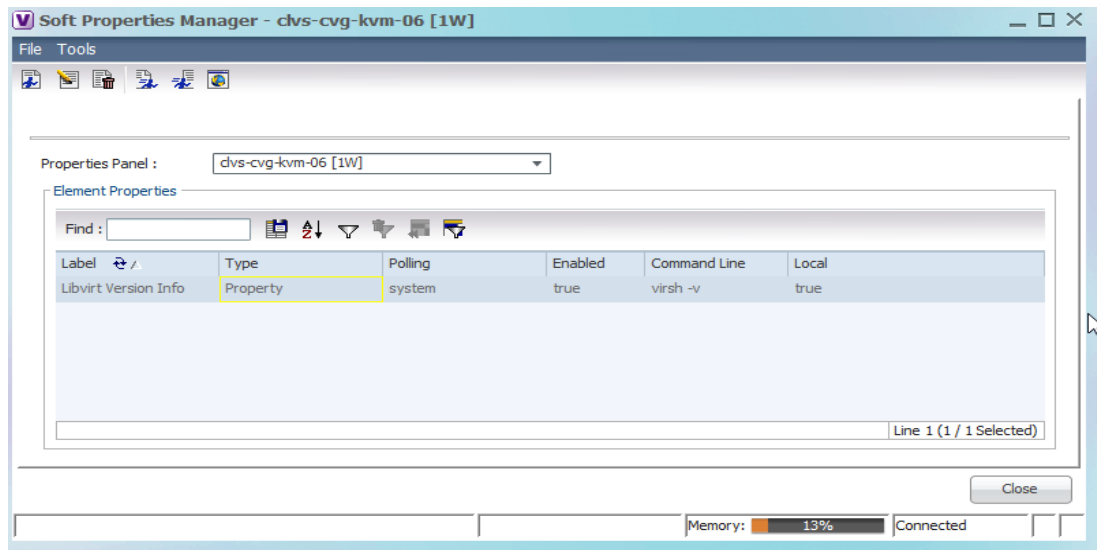
Re-Order

Add Edit Delete Test

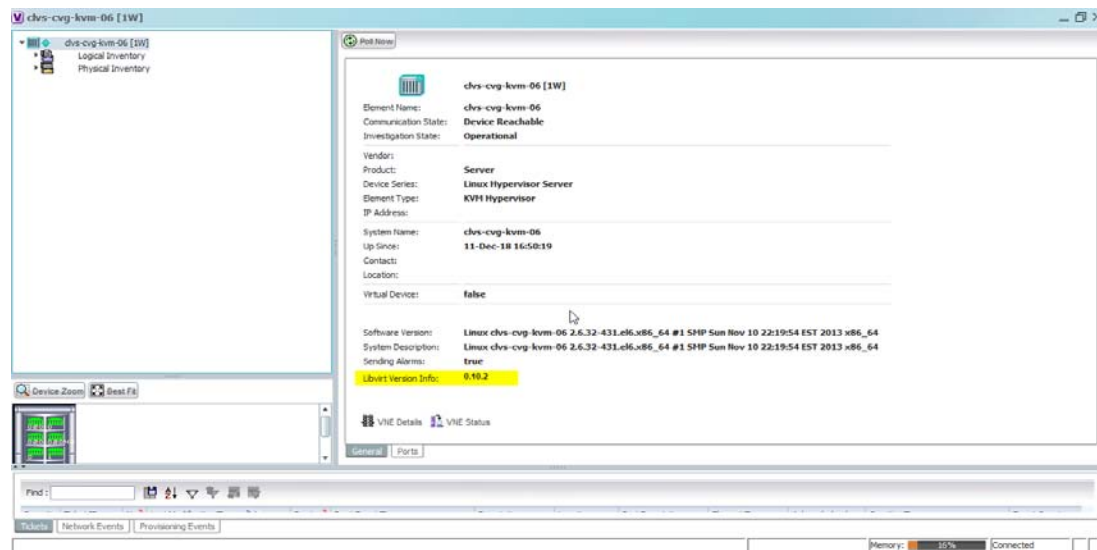
OK Cancel Debug

## Adding Soft Properties for KVM Hypervisor

**Step 5** Click *Close*, if done with addition of all required soft properties.



**Step 6** Close and re-open Inventory Window to see the added soft properties.



## Table Type

This topic lists steps to add soft properties in table format using snmp.

### Add 'Storage Device Details' Soft Properties for KVM Hypervisor

**Step 1** Right click *Managed Element > Management > Soft Properties Management*.

**Step 2** Click *New Element* button.

**Step 3** In General tab, enter *Name* and *Label*. Select 'Table' as Type and 'status' as Polling Rate.

The screenshot shows a configuration window with two tabs: 'General' and 'Parsing'. The 'General' tab is active. It contains the following fields:

- Name : StorageDeviceDetails
- Label : Storage Device Details
- Description :
- Type : Table (highlighted in yellow)
- Polling Rate : status (highlighted in yellow)
- Enabled :

At the bottom of the window are buttons for 'OK', 'Cancel', and 'Debug'.

**Step 4** In Parsing tab, select *Use SNMP get (OID)*. Enter Parent *OID*. In this case, *hrStorageEntry* OID.

The screenshot shows the 'Parsing' tab of the configuration window. It contains the following elements:

- Radio buttons:  Use SNMP get(OID) (highlighted in yellow),  Use Telnet/SSH
- Text field: .1.3.6.1.2.1.25.2.3.1 (highlighted in yellow)
- Table with columns 'Title' and 'OID'. The table is currently empty.
- Buttons: 'Add', 'Edit', 'Delete', 'Re-Order' (with up/down arrows), 'OK', 'Cancel', 'Debug'.

**Step 5** Click *Add*.

**Step 6** Enter *Column Title* and sub oid as *Column Data*. In this case, Enter 'Name' as column title and '3' as Column Data. Click *OK*.

The screenshot shows a dialog box titled "Add Edit Column Controller". It has two input fields: "Column Title" with the value "Name" and "Column Data" with the value "3". At the bottom right, there are "OK" and "Cancel" buttons.

**Step 7** Repeat Steps 5 and 6 to add required Storage Soft Properties.

**Step 8** Click *OK*, if all the required soft properties added.

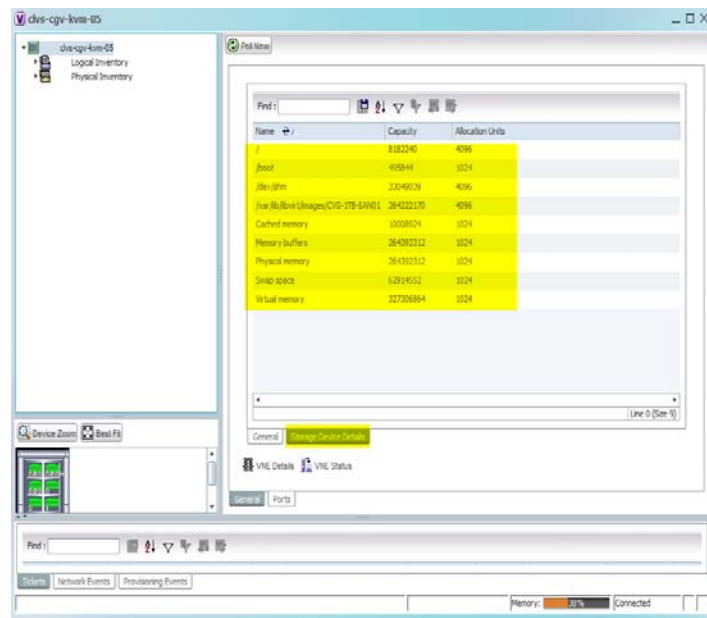
The screenshot shows the "Parsing" tab of a configuration window. It has two radio buttons: "Use SNMP get(OID)" (selected) and "Use Telnet/SSH". Below them is a text field containing ".1.3.6.1.2.1.25.2.3.1". A table lists properties:

Title	OID
Name	3
Capacity	5
Allocation Units	4

The "Name" row is highlighted in yellow. Below the table is a text field containing "Line 0 (Size 3)". At the bottom of the table area are "Add", "Edit", and "Delete" buttons. At the bottom of the window are "OK", "Cancel", and "Debug" buttons.

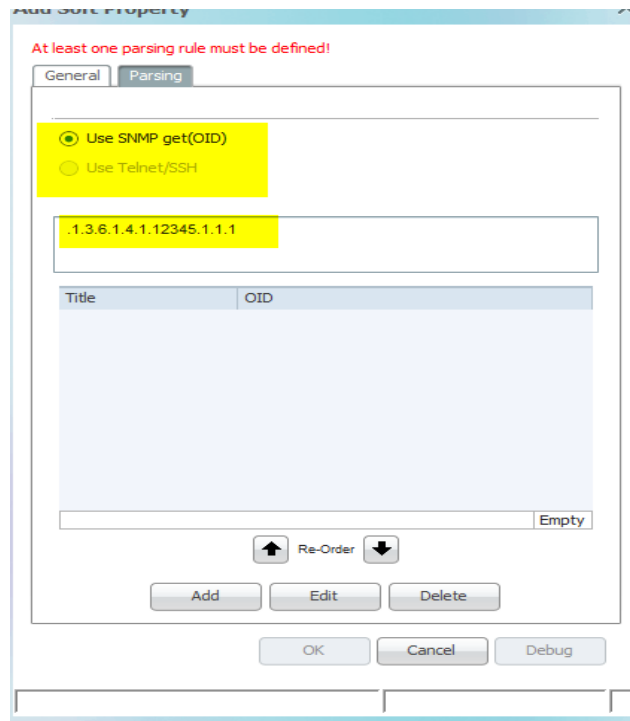
**Step 9** Close and Re-open Inventory Window to see the added Soft Properties.



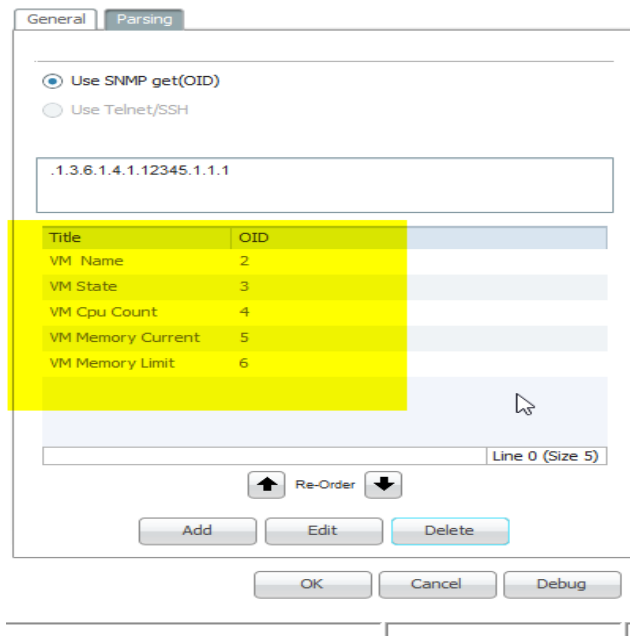


### Add 'VM Details' Soft Properties for KVM Hypervisor

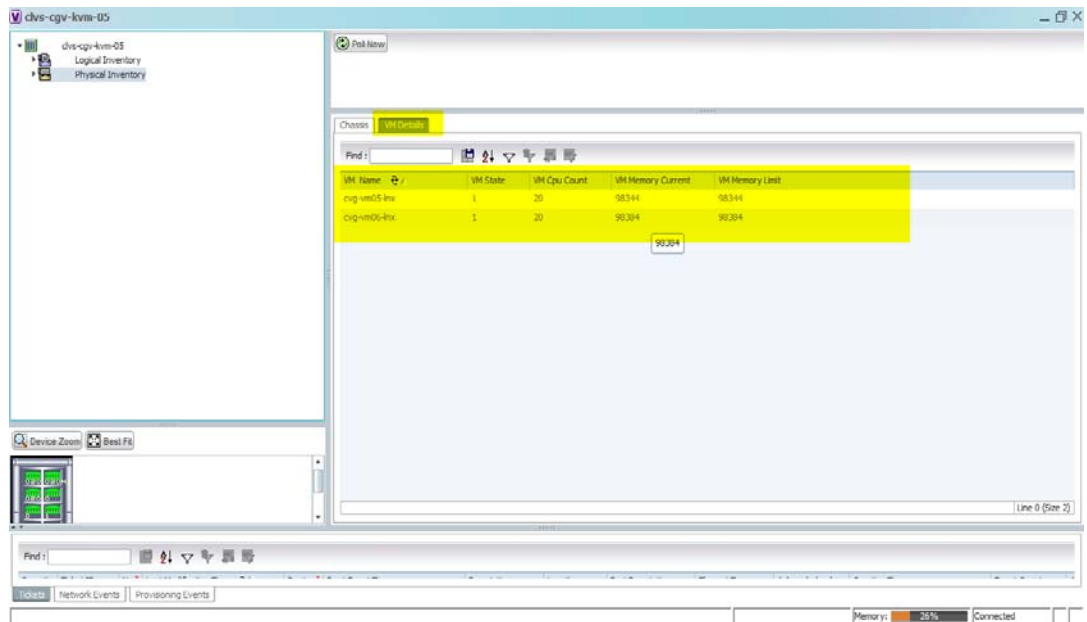
- Step 1** Right Click on *Physical Inventory* > *Management* > *Soft Properties Management*.
- Step 2** Click *New Element* button.
- Step 3** In *General* tab, enter *Name* and *Label*. Select 'Table' as Type and 'status' as Polling Rate.
- Step 4** In *Parsing* tab, select *Use SNMP get(OID)*. Enter Parent *OID*. In this case, *libvirtMIB OID*. Click *New Element* button.



**Step 5** Add all the required VM Soft Properties using *Add* option. Click *OK*.

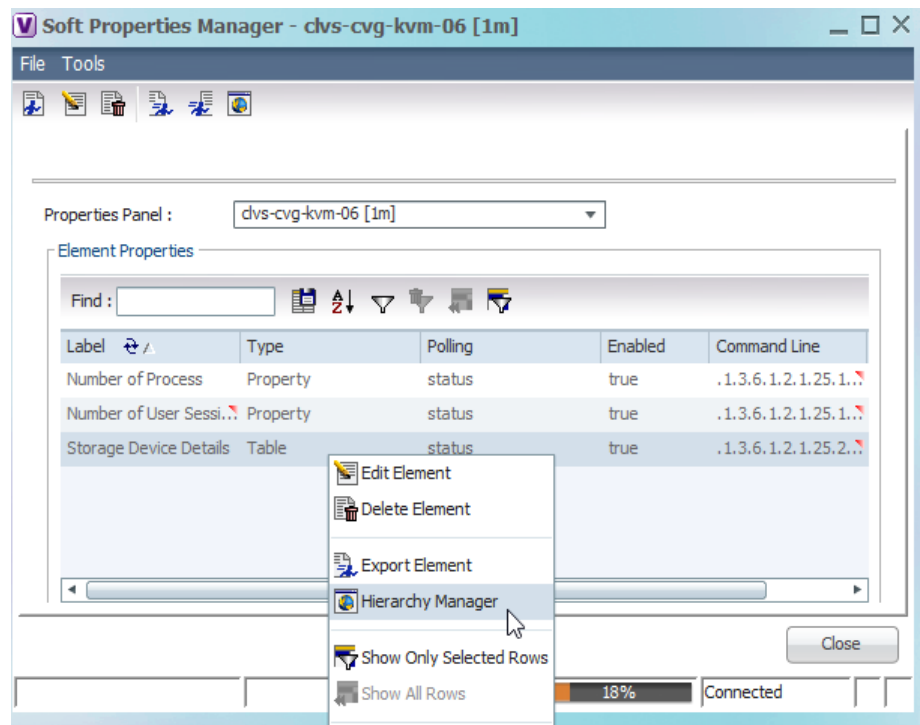


**Step 6** Close and Re-open Inventory Window to see the added VM Details Soft Properties.



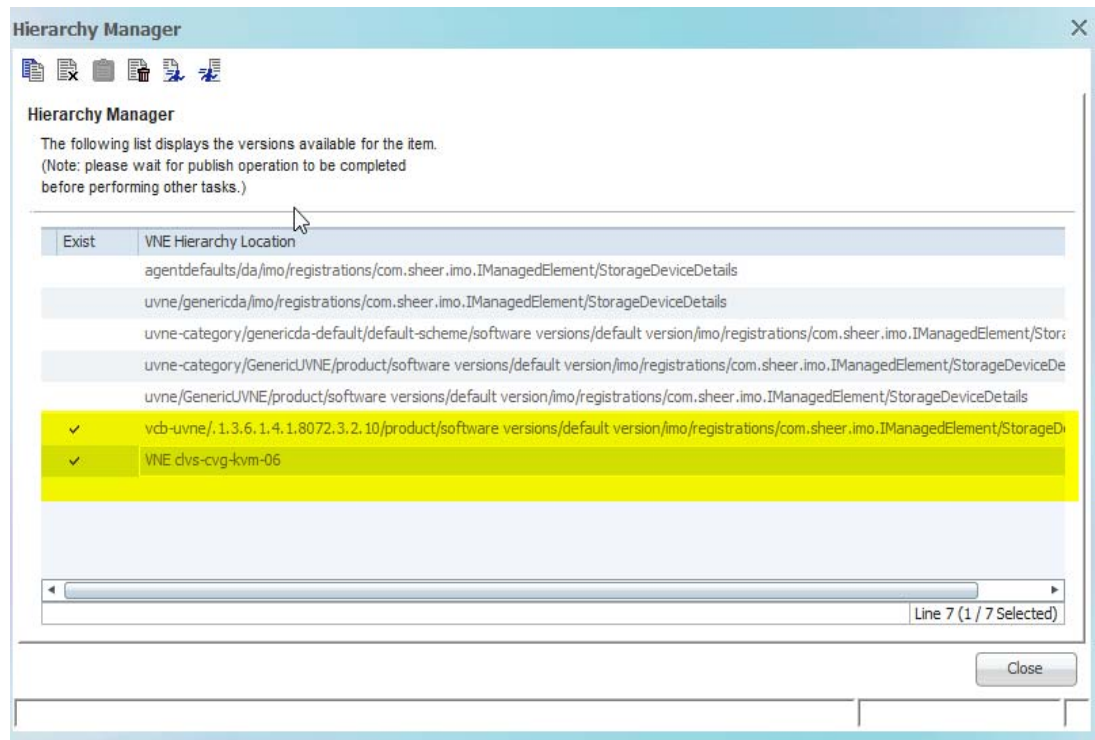
## Applying Soft Properties to all KVM Hypervisor VNEs

**Step 1** Right click added *Soft Properties > Hierarchy Manager*.



**Step 2** Copy the Existing Hierarchy using *Copy* option.

- Step 3** Select the New Hierarchy Location for which the Soft Property needs to be applied. Paste using *Paste* Icon.



- Step 4** Click *Close*.
- Step 5** Restart all affected VNE'S for which the soft properties are applied.