

Application Note



Application Note

Setting Group Thresholds Using Configuration Management

2/22/2019

Pete Bartz - Entuity



Table of Contents

1	Overview	3
2	Identify the thresholds that are needed	4
3	Use Case 1 – Show the current threshold value for CPU.....	6
4	Use Case 2 – Show the current threshold value for a dynamic threshold name.....	8
5	Use Case 3 – Set the threshold value for CPU.....	10
6	Use Case 4 – Set the threshold value based on dynamic parameters.....	13
7	Running the new tasks in ENA.....	17
7.1	Show CPU for a device	17
7.2	Set Device Average CPU Critical threshold to 63% for all devices in a view	19
7.3	Set a threshold for devices in a view based on parameters for the name and value.....	22

1 Overview





This Application Note describes a method that can be used to set device-based thresholds for a group of devices (view based) using an ENA Configuration Management Task.

This task will examine a selected device, device set or device in a view and set configured threshold values.

Note that this example applies to device level thresholds and the scripts can be adapted to include port-based thresholds if needed.

For additional detail on the default and custom thresholds that have been set, users have access to 2 standard reports in ENA.

[reports](#) > [Administrative](#)

Report	Schedule	History	Description
Custom Threshold Settings			Threshold settings that have been manually overridden
Default Server Threshold Settings			Threshold settings used as server level defaults and view level overrides

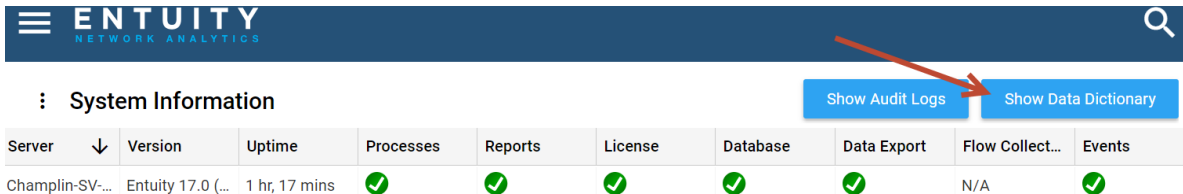
These can provide valuable details to support the changes you may want to make for groups of devices.

2 Identify the thresholds that are needed

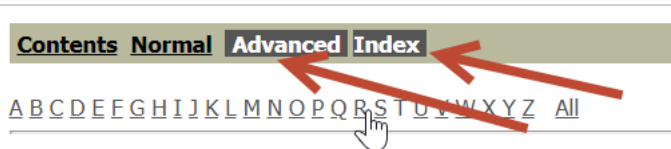
The first step in this process is to identify the thresholds that are needed to be set for a given group of devices. Once these are identified, you will need to identify the specific Threshold Name from the ENA data model for each one.

You can easily generate a list of threshold names that are configured on your server.

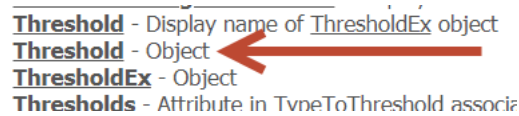
- a. From the ENA UI, go to the System Information UI and access “Show Data Dictionary”.



Select the Index and Advanced options

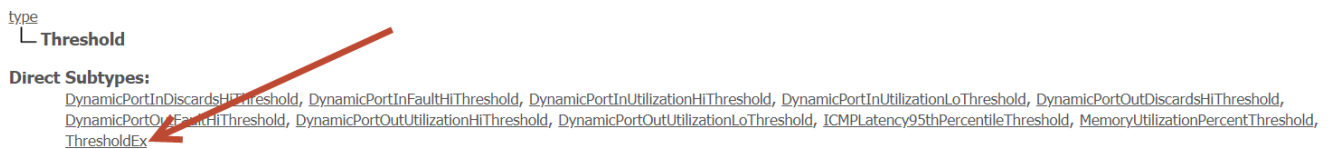


Scroll to and select the Threshold object



Copy and paste the Direct Subtypes into a new document. You can delete the ThresholdEx type from the document. Then from the Threshold object, select the ThresholdEx Subtype.

SwObject Type Threshold



Copy and paste all of the Direct Subtypes into the new document.

Direct Subtypes:

AbsoluteMemoryThreshold, AngularVelocitySensorHiThreshold, AngularVelocitySensorLoThreshold, APICHighCPUThreshold, APICHighMemoryThreshold, AtmVccInUtilizationHiThreshold, AtmVccInUtilizationLoThreshold, AtmVccOutUtilizationHiThreshold, AtmVccOutUtilizationLoThreshold, AwapHostCountHiThreshold, AwapHostCountLoThreshold, BackplaneUtilizationBusAThreshold, BackplaneUtilizationBusBThreshold, BackplaneUtilizationBusCThreshold, BackplaneUtilizationSystemBusThreshold, BECNThreshold, BroadcastTrafficThreshold, CPUUtilizationThreshold, CurrentSensorHiThreshold, CurrentSensorLoThreshold, DEThreshold, EPGLowHealthThreshold, FabricNodeLowHealthThreshold, FanTrayLowHealthThreshold, FECNThreshold, FrequencySensorHiThreshold, FrequencySensorLoThreshold, FWAcceptedPacketRateHighThreshold, FWCurrentConnectionsHighThreshold, FWDroppedPacketRateHighThreshold, FWLoggedPacketRateHighThreshold, FWRejectedPacketRateHighThreshold, HighMacAddressCountThreshold, HighVPNTotalHitsRateThreshold, HostResourcesLowDiskThreshold, HostResourcesPackagesThreshold, ICMPLatencyFallingThreshold, ICMPLatencyRisingThreshold, ICMPLatencyThreshold, ICMPLatencyTrendRisingThreshold, ICMPRedirectsThreshold, ICMP TTLThreshold, InboundPVCUtilizationThreshold, IPNoRoutesThreshold, IPSLAHighICPIFThreshold, IPSLAHighLatencyThreshold, IPSLALowMOSThreshold, IbcConnectionLimitThreshold, IbcCurrentSessionsThreshold, IbcDroppedPktsThreshold, IbcInErrRateThreshold, IbcLicenseDeniedThreshold, IbcLpCriticalAvailThreshold, IbcLpCriticalServiceAvailThreshold, IbcLpLowAvailThreshold, IbcLpLowServiceAvailThreshold, IbcMaxSessionsThreshold, IbcMemoryErrorThreshold, IbcNoHandlerDeniedThreshold, IbcOutErrRateThreshold, IbcSynDeniedThreshold, IbcTotalErrorsThreshold, LineCardLowHealthThreshold, ManagedHostActiveSessionsHighThreshold, ManagedHostMessagesReceivedThreshold, NewAndChangedMacInhibitThreshold, opticalInputPowerHighThreshold, opticalInputPowerLowThreshold, opticalOutputPowerHighThreshold, opticalOutputPowerLowThreshold, OutboundPVCUtilizationThreshold, PodLowHealthThreshold, PortInDiscardsHiThreshold, PortInFaultHiThreshold, PortInUtilizationHiThreshold, PortInUtilizationLoThreshold, PortOutDiscardsHiThreshold, PortOutFaultHiThreshold, PortOutUtilizationHiThreshold, PortOutUtilizationLoThreshold, PortPacketMinRateForDiscardsThreshold, PortPacketMinRateForFaultThreshold, PowerSensorHiThreshold, PowerSensorLoThreshold, PSULowHealthThreshold, QOSClassBitRateHiThreshold, QOSClassDropBitRateHiThreshold, QOSClassDropPktRateNoBufHiThreshold, QOSQueueDropBitRateHiThreshold, RelativeDeviceCPUThreshold, RelativeDeviceCriticalCPUThreshold, RelativeDeviceCriticalMemoryThreshold, RelativeDeviceMemoryThreshold, RelativeHumiditySensorHiThreshold, RelativeHumiditySensorLoThreshold, ServiceInTrafficHiThreshold, ServiceInTrafficLoThreshold, ServiceOutTrafficHiThreshold, ServiceOutTrafficLoThreshold, SlsSPCurrentHighThreshold, SNMPResponseTimeThreshold, SupervisorCardLowHealthThreshold, TemperatureSensorHiThreshold, TemperatureSensorLoThreshold, TenantLowHealthThreshold, ud_ahRadioTxPowerLowThreshold, ud_ifInTrafficThresholdLowThreshold, ud_rPDU2DeviceStatusPowerCriticalThreshold, ud_rPDU2DeviceStatusPowerHighThreshold, VDiskHiDiskUsageThreshold, ViewDeviceReachabilityThreshold, VMHiCPUThreshold, VMHiGuestMemoryThreshold, VMHiMaxDiskUsageThreshold, VoltageSensorHiThreshold, VoltageSensorLoThreshold, VolumeFlowSensorHiThreshold, VolumeFlowSensorLoThreshold, VpnActiveTunnelsHighThreshold, VpnHighClusterCurrentUsersThreshold, VpnHighCurrentUsersThreshold, VpnLoadAvgHiThreshold, VpnNetPortUtilHiThreshold, VpnTunnelUsageHiThreshold, vServerCurrentSessionsHighThreshold, vServerSessionsRateHighThreshold, vServiceCurrentSessionsHighThreshold, vServiceFailedSessionsRateHighThreshold, vServiceSessionsRateHighThreshold, WanPortInDiscardsHiThreshold, WanPortInErrorsHiThreshold, WanPortInUtilizationHiThreshold, WanPortInUtilizationLoThreshold, WanPortOutDiscardsHiThreshold, WanPortOutErrorsHiThreshold, WanPortOutUtilizationHiThreshold, WanPortOutUtilizationLoThreshold, WcApHostCountHiThreshold, WcApHostCountLoThreshold, WcBaseAerialIfChannelChangeFreqThreshold, WcBaseAerialIfHostCountHiThreshold, WcBaseAerialIfHostCountLoThreshold, WcBaseAerialIfPowerChangeFreqThreshold, wcCurrentNumberOfAPsConnectedHighThreshold, WpHostCountHiThreshold, WpHostCountLoThreshold

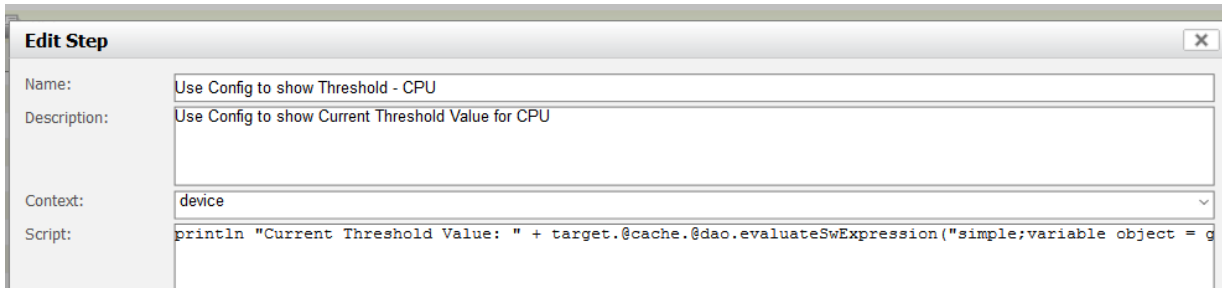
In your document, select a find and replace option to replace all of the “ , ” (comma space) characters with a new line. This will create a list of Threshold Names (1 per line) and this can be copied into an Excel spreadsheet.

The threshold names that are device based can be used for the configuration scripts defined in steps 4 and 6 below.

3 Use Case 1 – Show the current threshold value for CPU.

In this example we will create a Task to show the Device Average CPU Critical threshold for a device(s).

- a. From the Configuration Management UI, add a new step.



Set the name: Use Config to show Threshold – CPU

Set the Description: Use Config to show Current Threshold Value for CPU

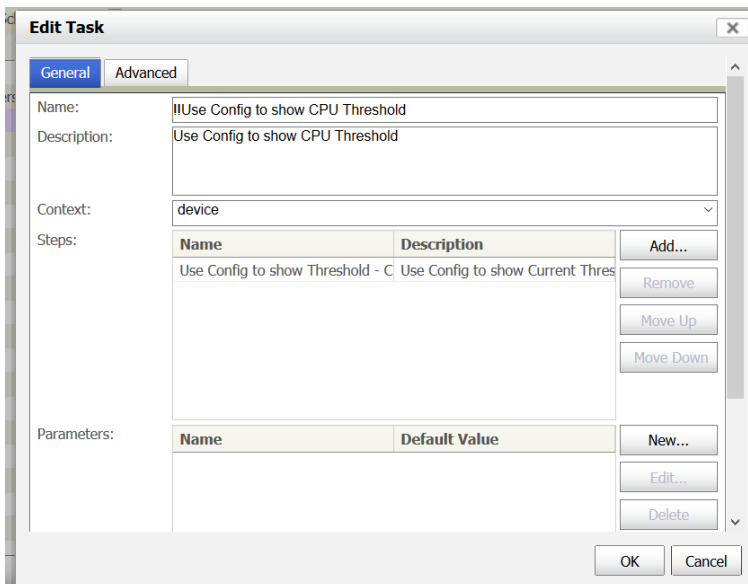
Set the Context: Device

Set the following into the Script (Note to make sure this is standard text and is a single line with no page breaks):

```
println "Current Threshold Value: " + target.@cache.@dao.evaluateSwExpression("simple;variable
object = getObject(${target.id}); eval(object, object,
get_threshold(\"RelativeDeviceCriticalCPUThreshold\"))")
```

Select OK to save the step.

- b. From the Configuration Management UI, add a new task.



Set the name: !!Use Config to show CPU Threshold

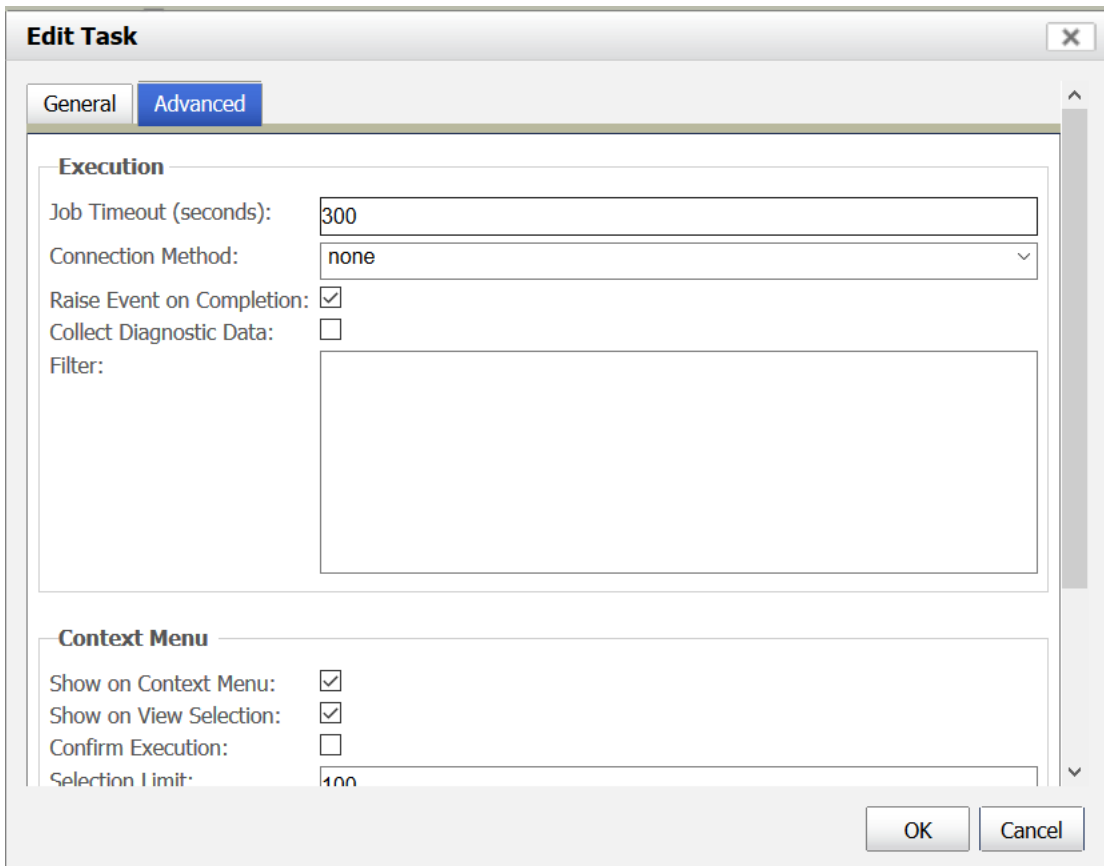
Note that I added “!!” at the front to help force the task to the top of the list. You can name this appropriately for your environment.

Set the Description: Use Config to show CPU Threshold

Set the Context: Device

Add a new step to include the new “Use Config to show Threshold – CPU” step created above.

Select the Advanced tab and set the parameters consistent with the screen below.



Edit Task

General **Advanced**

Execution

Job Timeout (seconds): 300

Connection Method: none

Raise Event on Completion:

Collect Diagnostic Data:

Filter:

Context Menu

Show on Context Menu:

Show on View Selection:

Confirm Execution:

Selection Limit: 100

OK Cancel

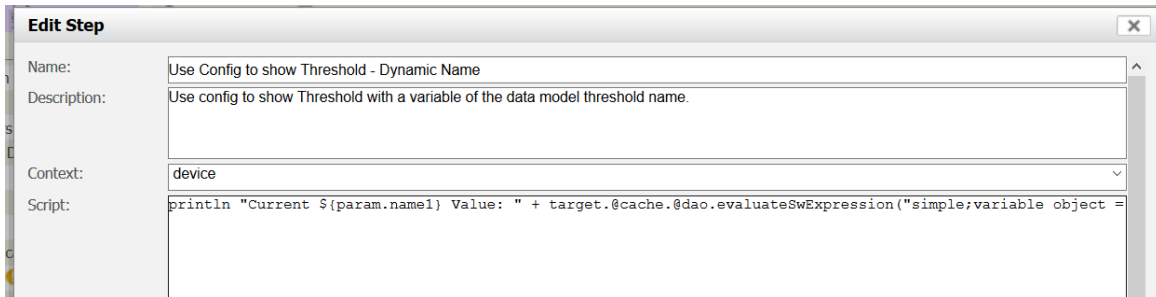
Select OK to save the task.

This task is now available to run. Note that from a device, you can right click and see the new configuration task. If it doesn’t show up, reload your browser screen.

4 Use Case 2 – Show the current threshold value for a dynamic threshold name

In this example we will create a Task to show the threshold value for a threshold name that is passed as a parameter in the task.

- a. From the Configuration Management UI, add a new step.



Edit Step

Name: Use Config to show Threshold - Dynamic Name

Description: Use config to show Threshold with a variable of the data model threshold name.

Context: device

Script: `println "Current ${param.name1} Value: " + target.@cache.@dao.evaluateSwExpression("simple;variable object =`

Set the name: Use Config to show Threshold - Dynamic Name

Set the Description: Use config to show Threshold with a variable of the data model threshold name.

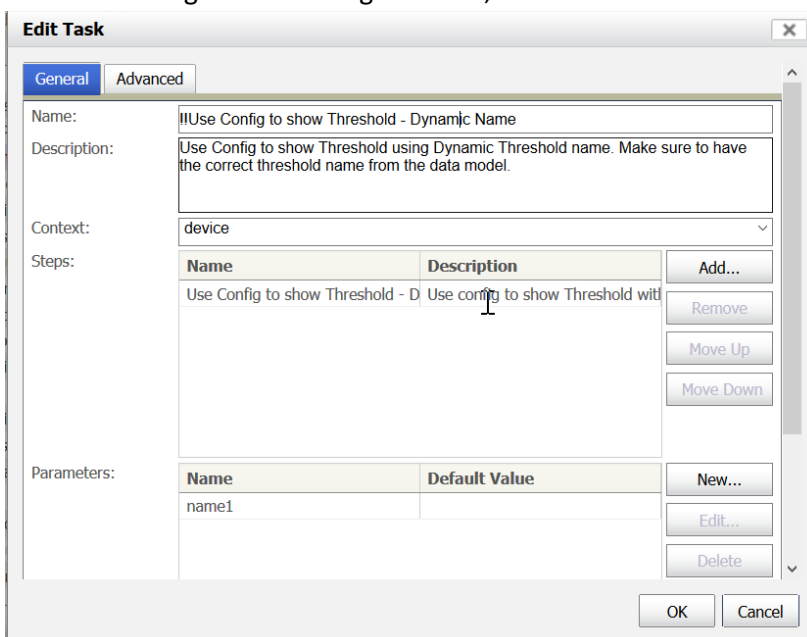
Set the Context: Device

Set the following into the Script (Note to make sure this is standard text and is a single line with no page breaks):

```
println "Current ${param.name1} Value: " +
target.@cache.@dao.evaluateSwExpression("simple;variable object = getObject(${target.id});
eval(object, object, get_threshold(\""+param.name1+"\"))")
```

Select OK to save the step.

- b. From the Configuration Management UI, add a new task.



Edit Task

General | Advanced

Name: Use Config to show Threshold - Dynamic Name

Description: Use Config to show Threshold using Dynamic Threshold name. Make sure to have the correct threshold name from the data model.

Context: device

Name	Description
Use Config to show Threshold - D	Use config to show Threshold with

Parameters:

Name	Default Value
name1	

OK Cancel

Set the name: !!Use Config to show Threshold - Dynamic Name

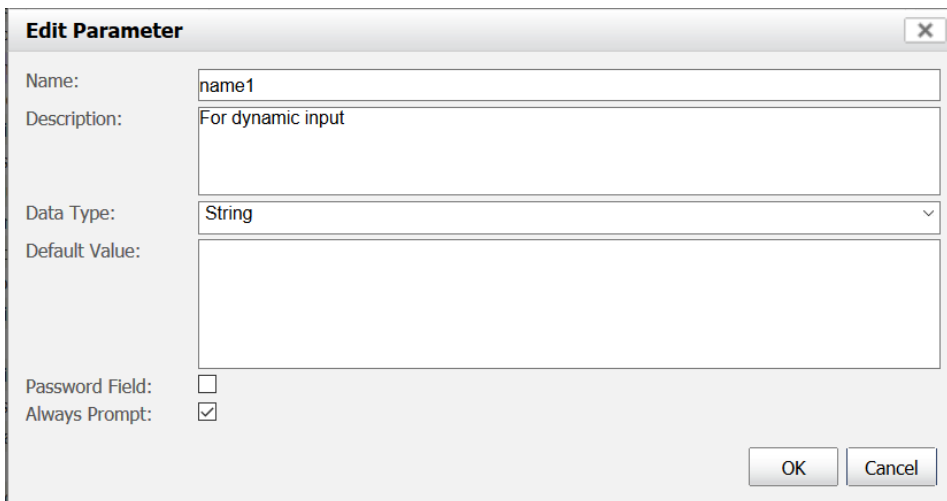
Note that I added “!!” at the front to help force the task to the top of the list. You can name this appropriately for your environment.

Set the Description: Use Config to show Threshold using Dynamic Threshold name. Make sure to have the correct threshold name from the data model.

Set the Context: Device

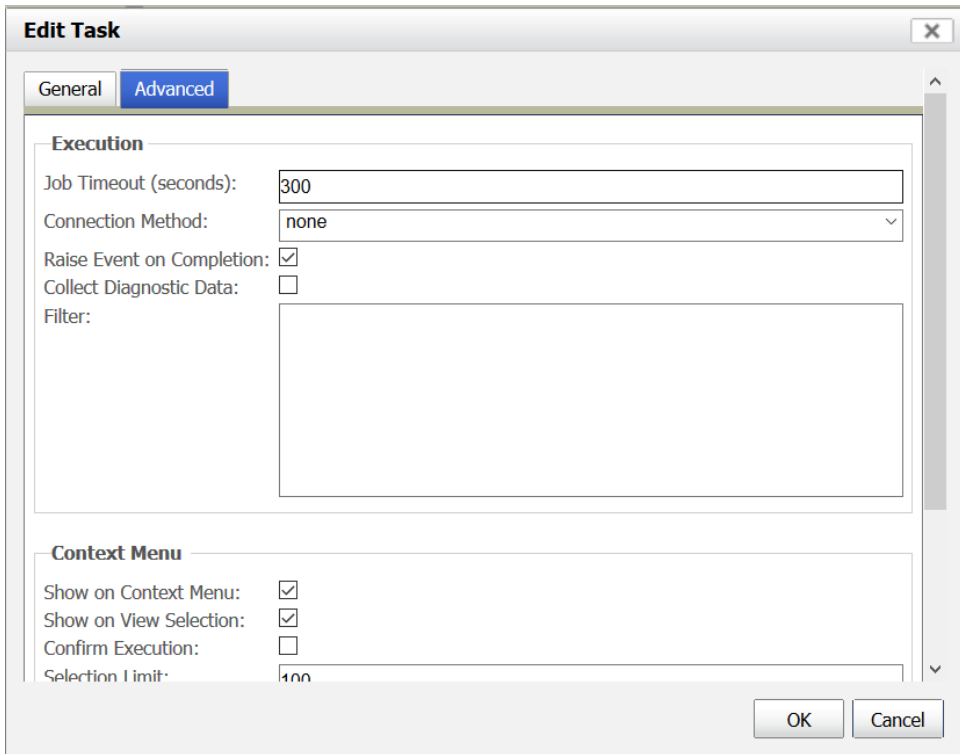
Add a new step to include the new “Use Config to show Threshold - Dynamic Name” step created above.

Add a new Parameter as defined below.



Select OK.

Select the Advanced tab and set the parameters consistent with the screen below.



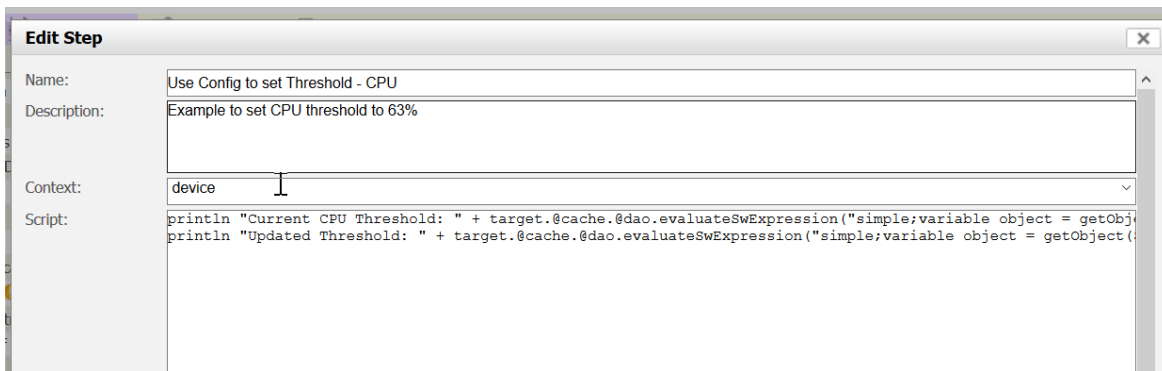
Select OK to save the task.

This task is now available to run. Note that from a device, you can right click and see the new configuration task. If it doesn't show up, reload your browser screen.

5 Use Case 3 – Set the threshold value for CPU.

In this example we will create a Task to set the Device Average CPU Critical threshold to 63% for a device(s).

- a. From the Configuration Management UI, add a new step.



Set the name: Use Config to set Threshold – CPU

Set the Description: Example to set CPU threshold to 63%

Set the Context: Device

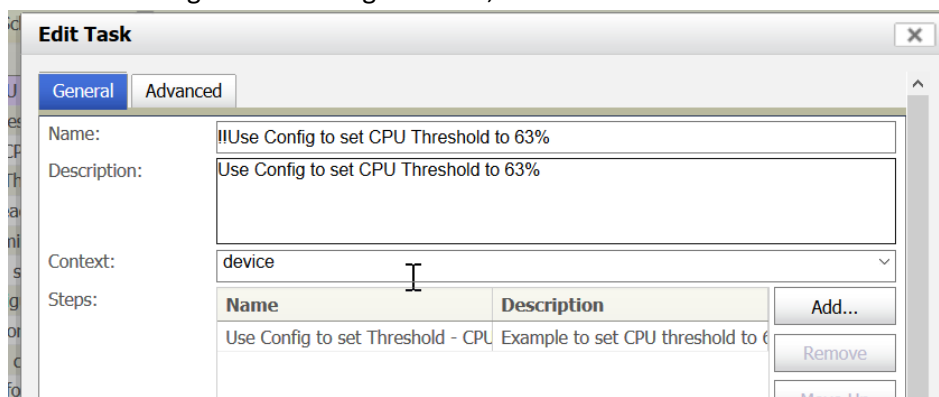
Set the following into the Script (Note to make sure this is standard text and is a single line with no page breaks):

```
println "Current CPU Threshold: " + target.@cache.@dao.evaluateSwExpression("simple;variable object = getObject(${target.id}); eval(object, object, replace_individual_threshold(\"RelativeDeviceCriticalCPUThreshold\",{63.0,\"enabled\"}))")
```

```
println "Updated Threshold: " + target.@cache.@dao.evaluateSwExpression("simple;variable object = getObject(${target.id}); eval(object, object, get_threshold(\"RelativeDeviceCriticalCPUThreshold\"))")
```

Select OK to save the step. Note that **RelativeDeviceCriticalCPUThreshold** is the threshold name that is being set. This is consistent with the data model and the document created in step 2. The value to be used in this example is **63.0** and can be changed as appropriate.

- b. From the Configuration Management UI, add a new task.



Set the name: !!Use Config to set CPU Threshold to 63%

Note that I added “!!” at the front to help force the task to the top of the list. You can name this appropriately for your environment.

Set the Description: Use Config to set CPU Threshold to 63%

Set the Context: Device

Add a new step to include the new “Use Config to set Threshold – CPU” step create above.

Select the Advanced tab and set the parameters consistent with the screen below.

Edit Task
X

General
Advanced

Execution

Job Timeout (seconds):

Connection Method:

Raise Event on Completion:

Collect Diagnostic Data:

Filter:

Context Menu

Show on Context Menu:

Show on View Selection:

Confirm Execution:

Selection Limit:

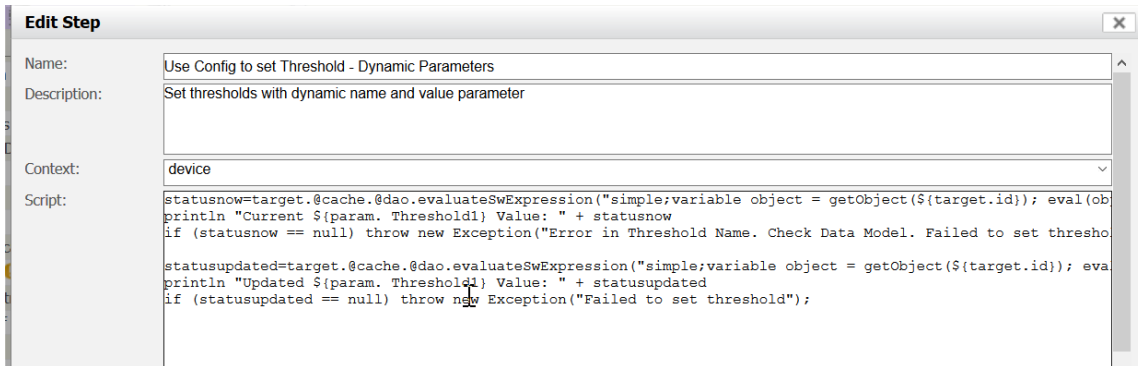
Select OK to save the task.

This task is now available to run. Note that from a device, you can right click and see the new configuration task. If it doesn't show up, reload your browser screen.

6 Use Case 4 – Set the threshold value based on dynamic parameters

In this example we will create a Task to set the threshold value for a threshold name and value that are passed as parameters in the task.

- a. From the Configuration Management UI, add a new step.



Set the name: Use Config to set Threshold - Dynamic Parameters

Set the Description: Set thresholds with dynamic name and value parameter

Set the Context: Device

Set the following into the Script (Note to make sure this is standard text and is a single line with no page breaks):

```
statusnow=target.@cache.@dao.evaluateSwExpression("simple;variable object =
getObject({target.id}); eval(object, object,
replace_individual_threshold(\""+param.Threshold1+"\",{"+param.Value1+",\"enabled\"}))")
println "Current ${param. Threshold1} Value: " + statusnow
if (statusnow == null) throw new Exception("Error in Threshold Name. Check Data Model. Failed to set
threshold");
```

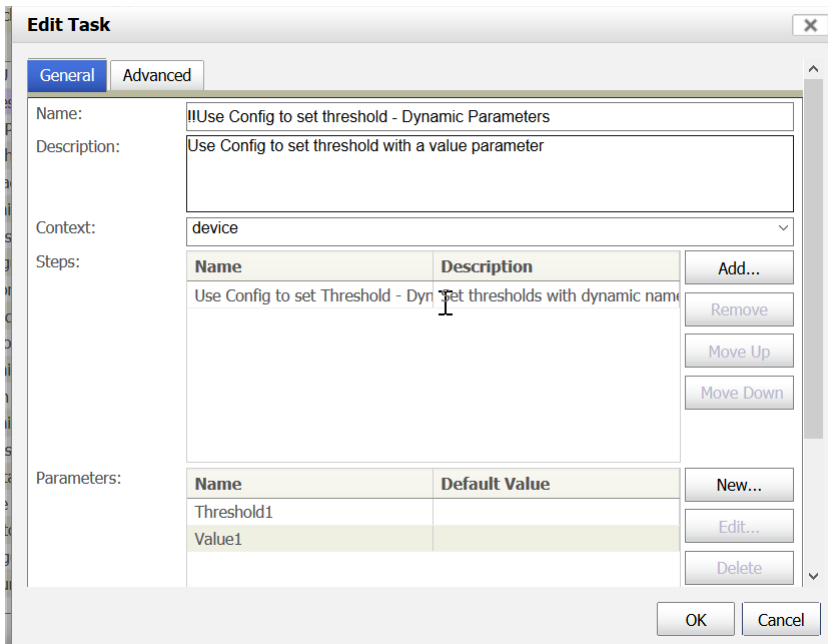
```
statusupdated=target.@cache.@dao.evaluateSwExpression("simple;variable object =
getObject({target.id}); eval(object, object, get_threshold(\""+param. Threshold1+"\"))")
println "Updated ${param. Threshold1} Value: " + statusupdated
if (statusupdated == null) throw new Exception("Failed to set threshold");
```

Note that the parameters are passed in the task below and need to match exactly by name in the script.

Note that this example sets 1 threshold. The step can be updated to include more than 1 parameter and threshold. Duplicate the script and change both Threshold1 and Value1 to another numeric value (ie Threshold2). Also add additional parameters in the task below to match the updated script.

Select OK to save the step.

b. From the Configuration Management UI, add a new task.



Set the name: !!Use Config to set threshold - Dynamic Parameters

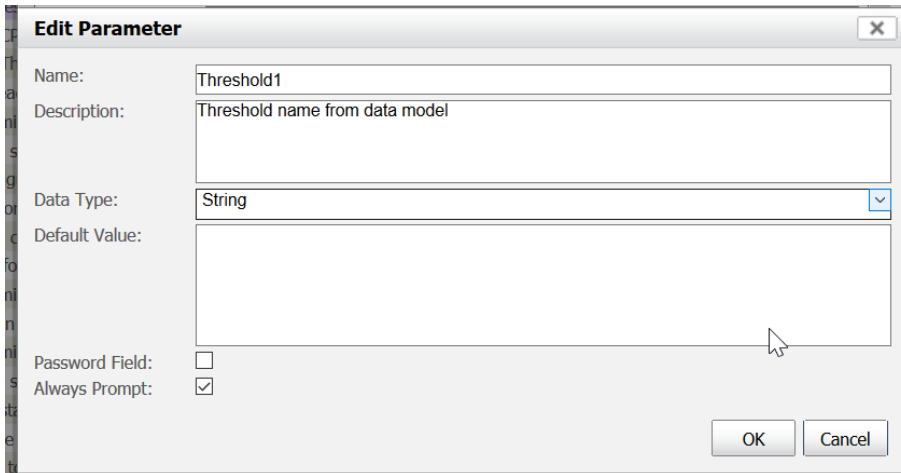
Note that I added “!!” at the front to help force the task to the top of the list. You can name this appropriately for your environment.

Set the Description: Use Config to set threshold with a value parameter. Make sure to have the correct threshold name from the data model.

Set the Context: Device

Add a new step to include the new “Use Config to set Threshold - Dynamic Parameter” step created above.

Add a new Parameter as defined below.



Edit Parameter

Name:

Description:

Data Type:

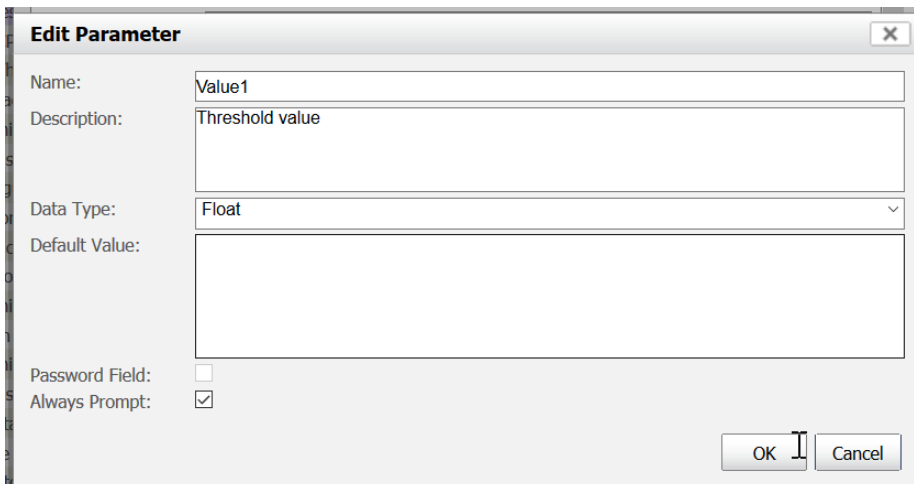
Default Value:

Password Field:

Always Prompt:

Select OK.

Add a second parameter as defined below.



Edit Parameter

Name:

Description:

Data Type:

Default Value:

Password Field:

Always Prompt:

Select OK.

Select the Advanced tab and set the parameters consistent with the screen below.

Edit Task
✕

General
Advanced

Execution

Job Timeout (seconds):

Connection Method:

Raise Event on Completion:

Collect Diagnostic Data:

Filter:

Context Menu

Show on Context Menu:

Show on View Selection:

Confirm Execution:

Selection Limit:

Select OK to save the task.

This task is now available to run. Note that from a device, you can right click and see the new configuration task. If it doesn't show up, reload your browser screen.

7 Running the new tasks in ENA

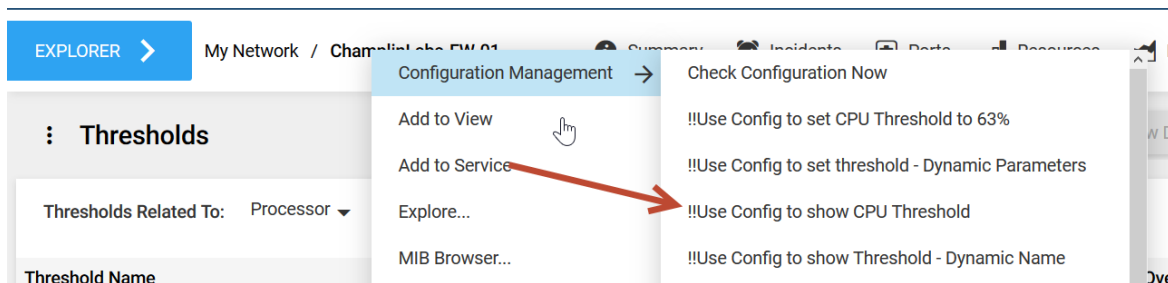
Now that the new tasks are created, they can be run against a single device, set of devices or view. Below are examples based on some of the tasks defined above.

7.1 Show CPU for a device

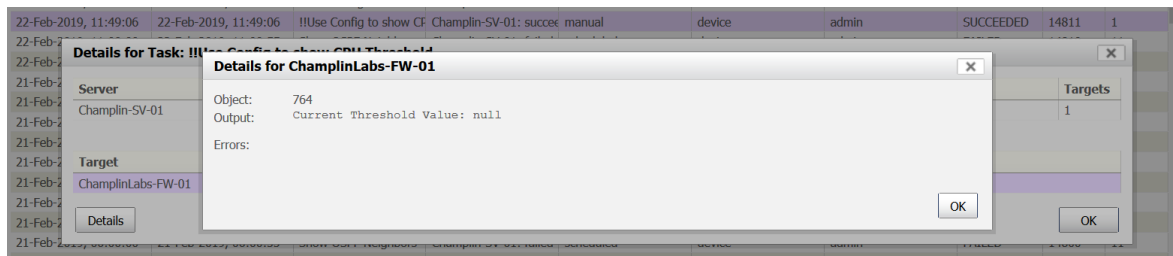
In this example, I am included a number of cases. In Use case 1, the threshold is not enabled.

Thresholds Related To: Processor ▼				
Threshold Name	Group	Enabled	Value	Overridden
Device Average CPU Utilization High	Device CPU	No	25	Yes
Device Average CPU Utilization Critical	Device CPU	No	99	Yes
Processor Utilization High	Processor	Yes	80	No

From either the bread crumbs or the Explorer, right-click on the device and select the “!!Use Config to show CPU Threshold below.



Check the Configuration Management History and examine the details of the target device.



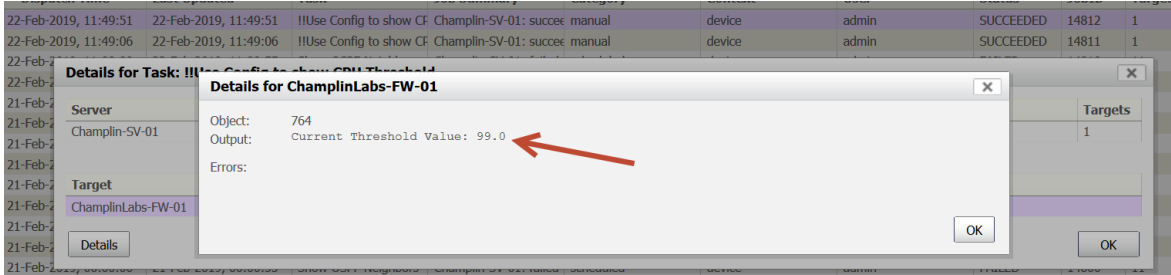
Note that in this example the output shows a null value. This is because the threshold is not enabled.

In Use case 2 we will run the script for an enabled threshold as shown below:

Thresholds Related To: Processor ▾

Threshold Name	Group	Enabled	Value	Overridden
Device Average CPU Utilization High	Device CPU	No	25	Yes
Device Average CPU Utilization Critical	Device CPU	Yes	99	Yes
Processor Utilization High	Processor	Yes	80	No

The output of the script will show the current value.



The screenshot shows a task execution log with the following entries:

```

22-Feb-2019, 11:49:51 22-Feb-2019, 11:49:51 !!Use Config to show CF Champlin-SV-01: succee manual device admin SUCCEEDED 14812 1
22-Feb-2019, 11:49:06 22-Feb-2019, 11:49:06 !!Use Config to show CF Champlin-SV-01: succee manual device admin SUCCEEDED 14811 1
  
```

The 'Details for Task: !!Use Config to show CPU Threshold' dialog box is open, showing the following information:

- Server: Champlin-SV-01
- Target: ChamplinLabs-FW-01
- Object: 764
- Output: Current Threshold Value: 99.0
- Errors:

7.2 Set Device Average CPU Critical threshold to 63% for all devices in a view

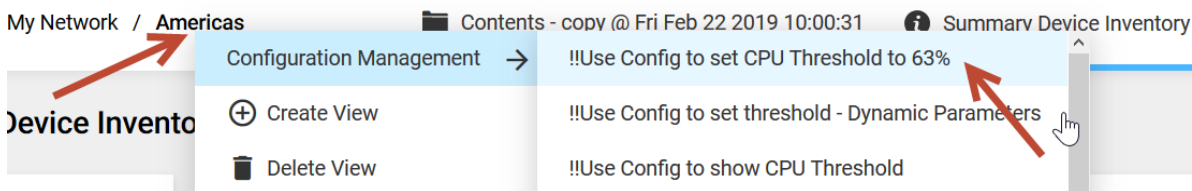
This example will set a predefined threshold to a predefined value using the task in step 5. If the CPU threshold is not enabled, the script will enable it and set the proper value.

The threshold is initially set as defined for at least one of the devices.

Thresholds Related To: Processor ▾

Threshold Name	Group	Enabled	Value	Overridden
Device Average CPU Utilization High	Device CPU	Yes	22	Yes
Device Average CPU Utilization Critical	Device CPU	No	67	Yes
Processor Utilization High	Processor	Yes	80	No

Right-click on a view and select the Configuration task.



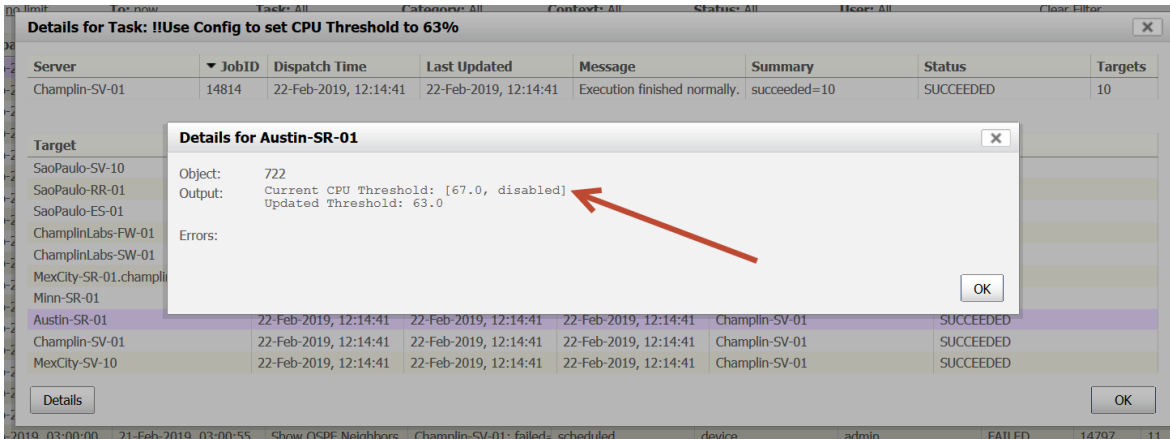
View the details of the task.

Details for Task: !!Use Config to set CPU Threshold to 63%

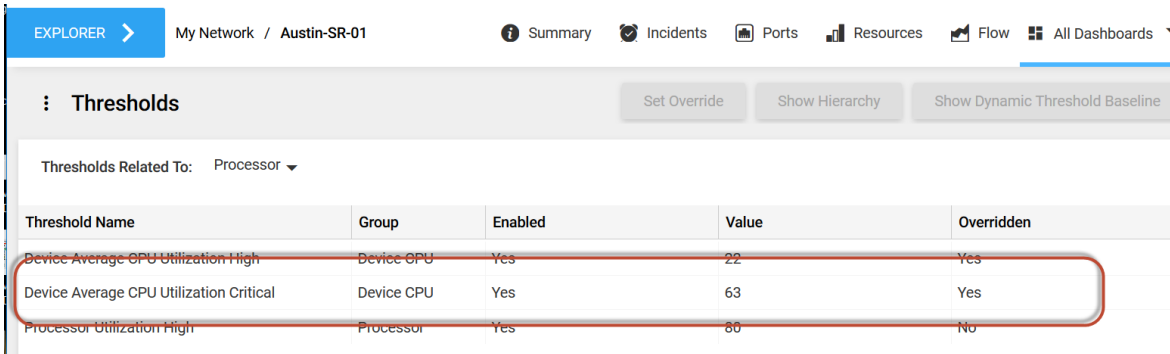
Server	JobID	Dispatch Time	Last Updated	Message	Summary	Status	Targets
Champlin-SV-01	14814	22-Feb-2019, 12:14:41	22-Feb-2019, 12:14:41	Execution finished normally.	succeeded=10	SUCCEEDED	10

Target	Started	Finished	Last Updated	Server	Status
SaoPaulo-SV-10	22-Feb-2019, 12:14:41	22-Feb-2019, 12:14:41	22-Feb-2019, 12:14:41	Champlin-SV-01	SUCCEEDED
SaoPaulo-RR-01	22-Feb-2019, 12:14:41	22-Feb-2019, 12:14:41	22-Feb-2019, 12:14:41	Champlin-SV-01	SUCCEEDED
SaoPaulo-ES-01	22-Feb-2019, 12:14:41	22-Feb-2019, 12:14:41	22-Feb-2019, 12:14:41	Champlin-SV-01	SUCCEEDED
ChamplinLabs-FW-01	22-Feb-2019, 12:14:41	22-Feb-2019, 12:14:41	22-Feb-2019, 12:14:41	Champlin-SV-01	SUCCEEDED
ChamplinLabs-SW-01	22-Feb-2019, 12:14:41	22-Feb-2019, 12:14:41	22-Feb-2019, 12:14:41	Champlin-SV-01	SUCCEEDED
MexCity-SR-01.champlin.local	22-Feb-2019, 12:14:41	22-Feb-2019, 12:14:41	22-Feb-2019, 12:14:41	Champlin-SV-01	SUCCEEDED
Minn-SR-01	22-Feb-2019, 12:14:41	22-Feb-2019, 12:14:41	22-Feb-2019, 12:14:41	Champlin-SV-01	SUCCEEDED
Austin-SR-01	22-Feb-2019, 12:14:41	22-Feb-2019, 12:14:41	22-Feb-2019, 12:14:41	Champlin-SV-01	SUCCEEDED
Champlin-SV-01	22-Feb-2019, 12:14:41	22-Feb-2019, 12:14:41	22-Feb-2019, 12:14:41	Champlin-SV-01	SUCCEEDED
MexCity-SV-10	22-Feb-2019, 12:14:41	22-Feb-2019, 12:14:41	22-Feb-2019, 12:14:41	Champlin-SV-01	SUCCEEDED

Select a device and view the output details. Note that the current CPU shows to be disabled but is updated correctly.



If you examine the threshold for the device, you can see that it is enabled and the proper value is set.



All of the devices in the view have been changed. The results can also be viewed by running the Custom Thresholds report against the view.

Entuity Report
Custom Thresholds



Printed on: 22 Feb 2019 12:21:31 CST
 Description: Threshold settings that have been changed from their default values
 View: My Network/Americas
 Device: --AllDevices--

Americas Branch Sites	(Champlin-SV-01)
Americas Branch Sites: Service Low Outbound Aggregate Traffic Rate	1000000.0
Austin-SR-01	(Champlin-SV-01)
Austin-SR-01: Device Average CPU Utilization Critical	63.0
Austin-SR-01: Device Average CPU Utilization High	22.0
Champlin-SV-01	(Champlin-SV-01)
Champlin-SV-01: Device Average CPU Utilization Critical	63.0
ChamplinLabs-FW-01	(Champlin-SV-01)
ChamplinLabs-FW-01: Device Average CPU Utilization Critical	63.0
ChamplinLabs-FW-01: Device Average CPU Utilization High	Disabled
ChamplinLabs-SW-01	(Champlin-SV-01)
[g1] Sonicwall Uplink: ud_ifInTrafficThreshold Low Threshold	300000.0
[g2] EthernetInterface: ud_ifInFrameThreshold Low Threshold	6.0
ChamplinLabs-SW-01: Device Average CPU Utilization Critical	63.0
E:\ Label:Entuity_17_0_p Serial Number 3b2afb98	(Champlin-SV-01)

7.3 Set a threshold for devices in a view based on parameters for the name and value.

Note that this method can also be applied for the task in Use case 4 where you only need to show the thresholds. In this example we will use the task defined in Use case 6.

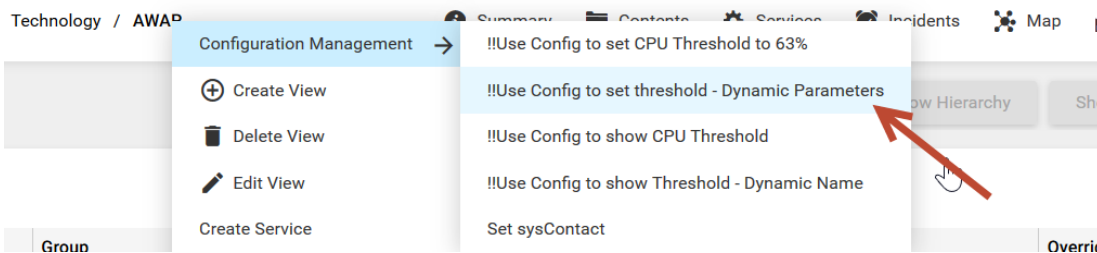
The threshold for one of my Wireless Access Points is shown below.

Threshold Name	Group	Enabled	Value	Overridden
AP Attached Host Count High	Autonomous AP	Yes	512	No
AP Attached Host Count Low	Autonomous AP	No	0	No

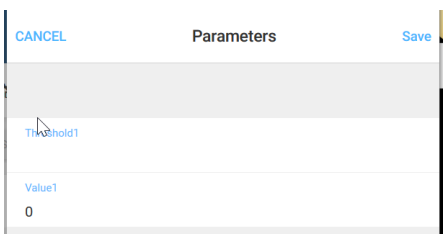
Using the document of threshold names created in step 2, we will change the “AP Attached Host Count High” threshold to 2.

The threshold name for this is: AwapHostCountHiThreshold

From the bread crumbs, right-click on the view and select the “!!Use Config to set threshold – Dynamic Parameters” task.

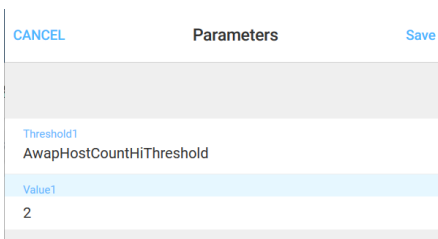


In the Parameters dialog:



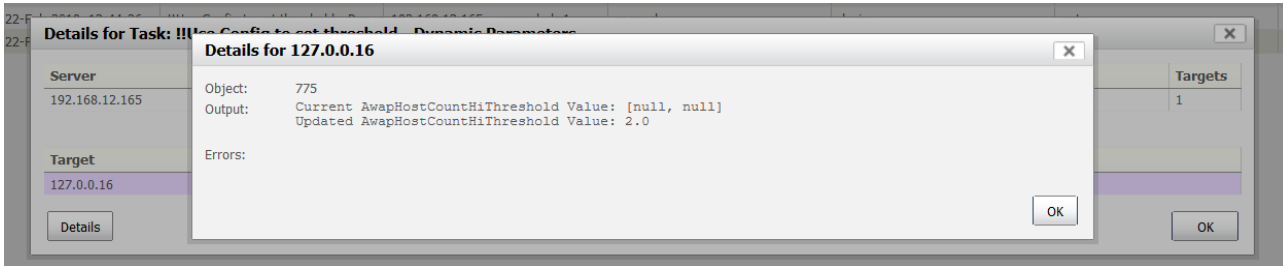
Set Threshold1 to: AwapHostCountHiThreshold

Set Value1 to: 2



Select the Save button which will run the task.

View the details of the completed task.



In this example the output is showing the current value to be null. This may be a flaw in the output statement for the script but even if this is the case, the threshold has been correctly updated to 2.

Thresholds Related To: Wireless ▾

Threshold Name	Group	Enabled	Value	Overridden
AP Attached Host Count High	Autonomous AP	Yes	2	Yes
AP Attached Host Count Low	Autonomous AP	No	0	No
AP Antenna Attached Host Count High	Autonomous AP Antenna	Yes	256	No

In most cases the current value should output correctly.

In my lab environment this change has also generated an incident to detect an Awap Host Count Abnormality.

!	#	Name	Source	Details	Last Updated	↑
2	1	Awap Host Count Abnormality	127.0.0.16	Count=15, High threshold=2.000...	12:57, 22 Feb 2019	
4	1	Device Reboot	127.0.0.16	Reboot at Thu Feb 21 12:35:31 ...	12:43, 22 Feb 2019	
5	1	Network Outage	127.0.0.16 [wifi1.5] wifi1.5	Port Unreachable: 127.0.0.16 [...	09:38, 21 Feb 2019	