



Application Note



Application Note

Populate an attribute using a CLI request via Configuration Management

23rd April 2019

John Diamond



Table of Contents

1	Overview	3
2	Creating a new object attribute	3
3	Device Task and Step definitions	4
4	Using the Configuration Management Task	8

1 Overview

Entuity usually populates the attributes of managed devices based on SNMP polling. Some devices including virtual servers and cloud devices are polled using a variety of non-SNMP APIs. This application note describes how a device that is already under management can have additional attributes added and populated using command line interface (CLI) interactions that can be automated using the Configuration Management capabilities of Entuity.

2 Creating a new object attribute

In order that a Configuration Management Task can be used to write data into an attribute, that attribute must be an Object Attribute (not created via the User Defined Polling facility) and must be configured to be editable. This is achieved using a configuration file and a reconfiguration of the Entuity system. This example will define an Object Attribute called “hardwareDetails” that can hold an arbitrary length text string. A test file called “sw_device_hardware.cfg” is created in the “etc” directory of the Entuity installation directory containing the following:

```
[Type DeviceEx]
Attributes+=,~hardwareDetails

[Attribute hardwareDetails]
DataType=Text
Description=Hardware Details
ClientData=displayName=Hardware Details\nreportableLevel=0

[Collector collectHardwareDetails]
Attribute=hardwareDetails
Description=Hardware Details
Method=simple;" "
Filter=simple;true
Priority=1
```

Note that the “~” before the attribute name in the second line of the file signifies that the attribute is allowed to be manually edited.

For this to be included into the system configuration, the following line is appended to the “sw_site_specific.cfg” file which is also in the “etc” directory:

```
!sw_device_hardware.cfg
```

To reconfigure the system to include this new attribute you’ll need to stop the Entuity server then run “configure defaults” from the “install” directory of the Entuity installation directory. Then restart the Entuity server.

Once the system is accessible again, you’ll see a new attribute described as “Hardware Details” in the Attributes dashboard when a device is selected.

Name	Value	Last Updated	Stream
EIGRP Peer Remote IP Addresses			EIGRP Peering Changes
Event Summary	Device Average CPU Utilization Cleared, source: C	11:02, 23 Apr 2019	Events
File Hits Rate	--		Device VPN Status
Firmware Version	11.0(10c)	13:01, 23 Apr 2019	
Hardware Details		13:01, 23 Apr 2019	
Hardware Version	L	13:01, 23 Apr 2019	
ICMP Reachable Known%	100%	13:08, 23 Apr 2019	Device Reachable Time without Root Cause An...

3 Device Task and Step definitions

The Configuration Management Step for gathering the hardware details for the device should be created first so that it can be incorporated in the corresponding Task. Here’s how such a Step would be created:

1. Administration -> Configuration Management -> Steps
2. Click “New” then fill in the Name, Description, Context and Script fields as follows:

New Step ✕

Name:

Description:

Context:

Script:

```
println GroovySystem.version;

expect.with
{
    if(vendor.equals("9")) // Only this section for Cisco devices
    {
        sendln "show hardware";
        // Buffer to hold show's output.
        StringBuffer buffer = new StringBuffer();
        expect(prompt, {buffer <<= matched})
        // Create an ArrayList with each line of text held separately
        lines = buffer.toString().split("\n").toList();
        if (lines.size() > 3) {
            lines.remove(0); // Remove first line
            lines.remove(lines.size()-1); // Remove last line
            lines.remove(lines.size()-1); // Remove last line
        }
    }
}
```

OK Cancel

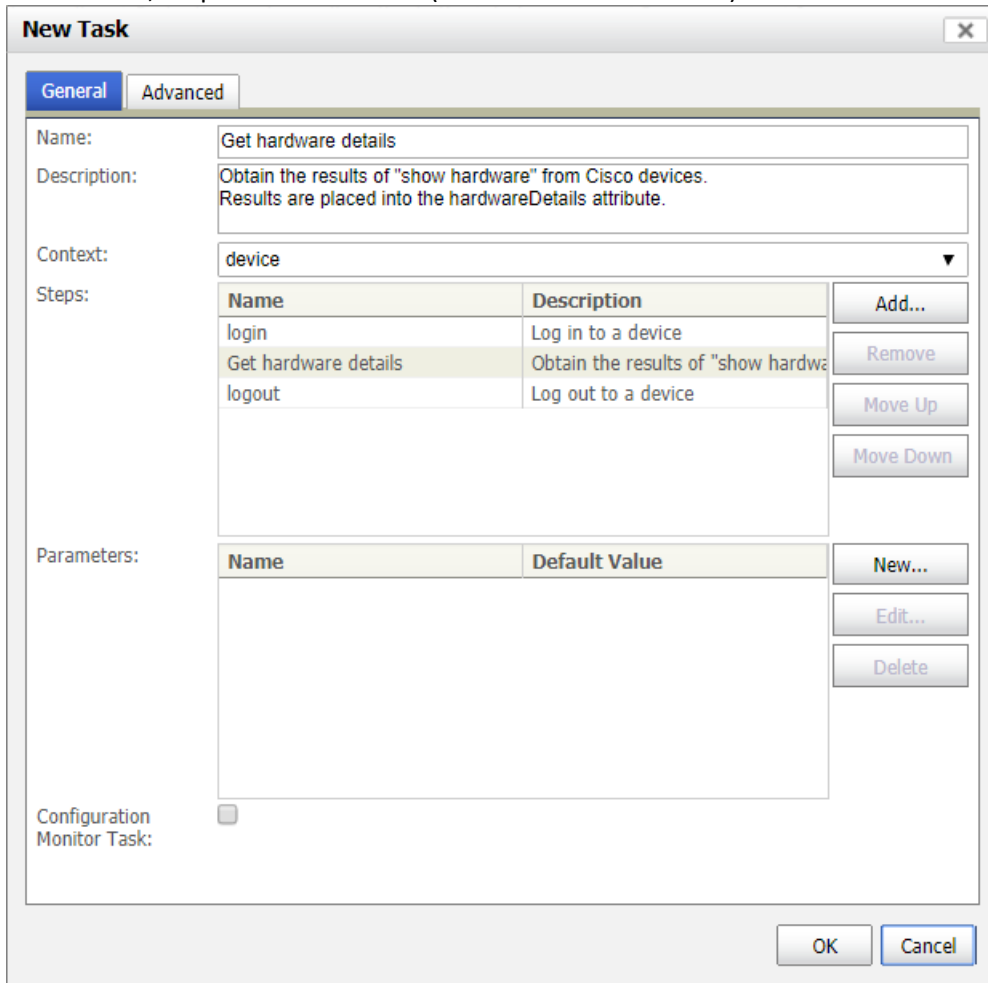
This is the text for the Script:

```
expect.with
{
  if(vendor.equals("9")) // Only this section for Cisco devices
  {
    sendIn "show hardware";
    // Buffer to hold show's output.
    StringBuffer buffer = new StringBuffer();
    expect(prompt, {buffer <<= matched})
    // Create an ArrayList with each line of text held separately
    lines = buffer.toString().split("\n").toList();
    if (lines.size() > 3) {
      lines.remove(0); // Remove first line
      lines.remove(lines.size()-1); // Remove last line
      lines.remove(lines.size()-1); // Remove last line
    }

    // Write the lines of text as a single string to the hardwareDetails object attribute
    def client = new com.entuity.jnirpc.JNIGenericRPC();
    def int userId = 3; // Must be valid userId with "Object Editing" tool permission
    def Object[] args = [userId, userId, Integer.parseInt(target.id), "hardwareDetails", lines.join("\n")];
    client.callRPCFunction("setObjectAttribute", args);

  }
  else
  {
    println "NO VALID METHOD FOR THIS DEVICE"
    throw new Exception("no valid method for this device")
  }
}
```

- Once the new Step has been saved select the Tasks tab and click “New” and fill in the Name, Description, Context, Steps and Parameters (Name and Default Value):



New Task

General | Advanced

Name: Get hardware details

Description: Obtain the results of "show hardware" from Cisco devices. Results are placed into the hardwareDetails attribute.

Context: device

Steps:

Name	Description
login	Log in to a device
Get hardware details	Obtain the results of "show hardware"
logout	Log out to a device

Parameters:

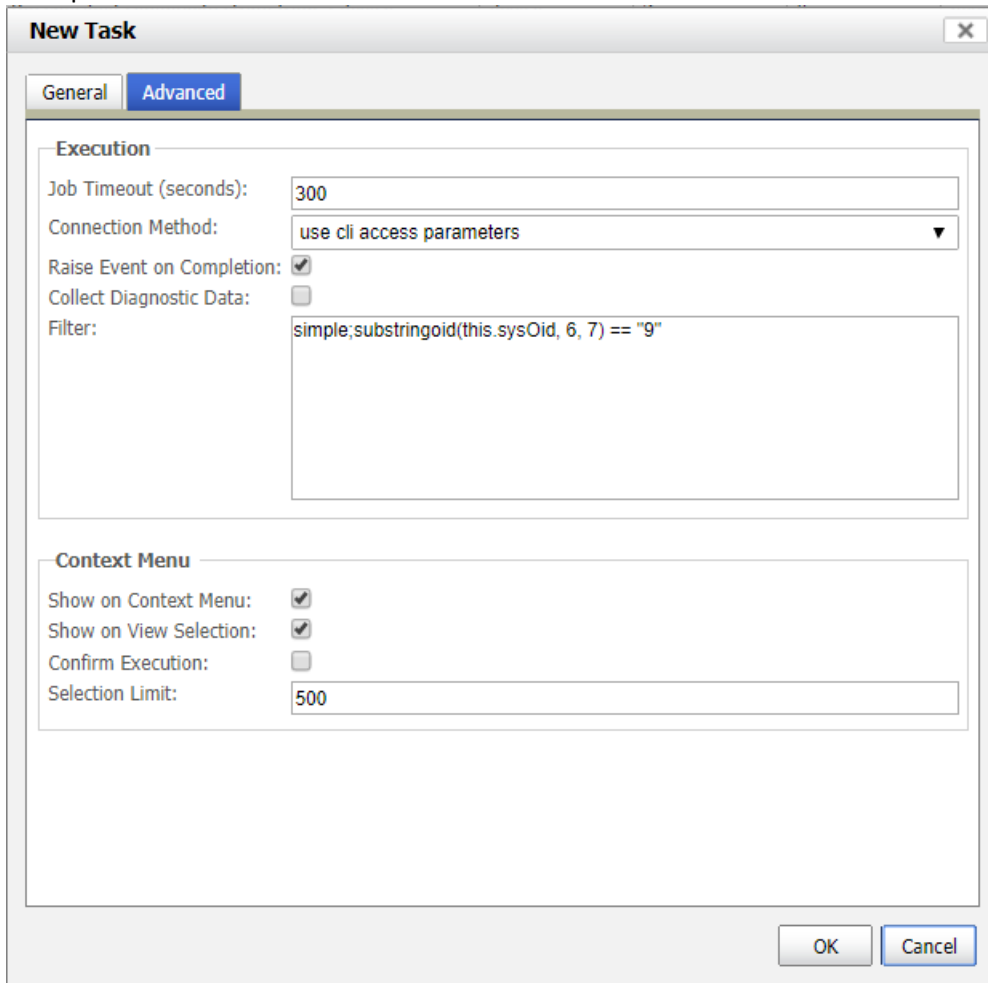
Name	Default Value
------	---------------

Configuration Monitor Task:

OK Cancel

This Task will log into the device, run the Step script defined earlier then log out.

- Select the Advanced tab then fill in the Filter and Selection Limit and check the Show on View Selection option:



New Task

General **Advanced**

Execution

Job Timeout (seconds): 300

Connection Method: use cli access parameters

Raise Event on Completion:

Collect Diagnostic Data:

Filter: simple;substringoid(this.sysOid, 6, 7) == "9"

Context Menu

Show on Context Menu:

Show on View Selection:

Confirm Execution:

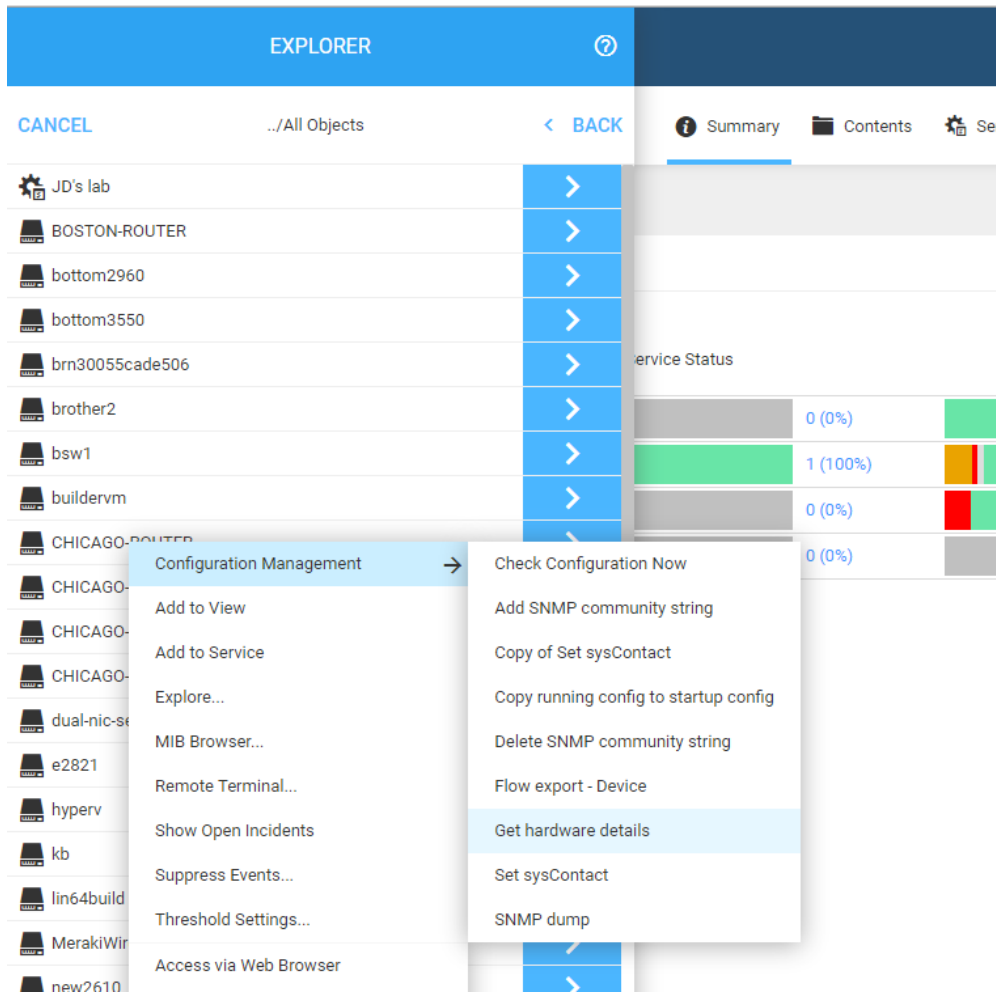
Selection Limit: 500

OK Cancel

The statement used for the Filter restricts the use of this Task to Cisco devices only. Enabling the Show on View Selection option caused an option to be added to the view context menu that allows the task to be executed on all qualifying devices which, in this case, means all the Cisco devices. The Selection Limit is the largest number of devices that can be selected for a single Configuration Management operation.

4 Using the Configuration Management Task

You can manually execute the Configuration Management task in the Explorer. Make sure you've configured at least one device for CLI access via the Inventory page settings. Right click either an individual device or a view and select the **Configuration Management -> Get hardware details** menu option. This corresponds to the new device level Task that's been created:



After a few seconds select the Advanced tab of the device in the Explorer and observe that the Hardware Details attribute has been populated from the output of the “show hardware” command that was issued to the device via its CLI:

The screenshot shows the Entuity Network Analytics interface. The breadcrumb trail is 'My Network / All Objects / CHICAGO-ROUTER'. The 'Attributes' section is active, displaying a table of device attributes. The 'Hardware Details' attribute is expanded, showing the following information:

Name	Value	Last Updated	Stream
EIGRP Peer Remote IP Addresses			EIGRP Peering Changes
Event Summary	Device Average CPU Utilization Cleared, source: CHICAGO-ROUTER, impacting: , details: 29.00%, high t	11:02, 23 Apr 2019	Events
File Hits Rate	--		Device VPN Status
Firmware Version	11.0(10c)	13:16, 23 Apr 2019	
Hardware Details	<p>Cisco Internetwork Operating System Software IOS (tm) 2500 Software (C2500-I-L), Version 12.3(22), RELEASE SOFTWARE (fc2) Technical Support: http://www.cisco.com/techsupport Copyright (c) 1986-2007 by cisco Systems, Inc. Compiled Wed 24-Jan-07 15:55 by ccai Image text-base: 0x03060CDC, data-base: 0x00001000</p> <p>ROM: System Bootstrap, Version 11.0(10c), SOFTWARE BOOTLDR: 3000 Bootstrap Software (IGS-BOOT-R), Version 11.0(10c), RELEASE SOFTWARE (fc1)</p> <p>CHICAGO-ROUTER uptime is 26 weeks, 1 day, 20 hours, 35 minutes System returned to ROM by power-on System restarted at 18:59:22 est Sun Feb 28 1993 System image file is "flash:c2500-I.123-22.bin"</p> <p>cisco 2500 (68030) processor (revision L) with 14336K/2048K bytes of memory. Processor board ID 01880688, with hardware revision 00000000 Bridging software X.25 software, Version 3.0.0. 2 Ethernet/IEEE 802.3 interface(s) 2 Serial network interface(s) 32K bytes of non-volatile configuration memory. 16384K bytes of processor board System flash (Read ONLY)</p> <p>Configuration register is 0x2102</p>	13:16, 23 Apr 2019	
Hardware Version	L	13:16, 23 Apr 2019	

The execution of this Configuration Management Task can be performed on a scheduled basis using the Configuration Management scheduler ... **Main Menu -> Administration -> Configuration Management -> Schedules**