EXTREME NETWORKS

Apply Config Template XIQ-SE workflow

Ludovico Stevens

Technical Marketing Engineering

July 2022



Apply Config Template XIQ-SE workflow



Workflow to apply an ASCII config template containing variables. Works with any device family The embedded variables can be of three types:

- \${variableName}: XIQ-SE Global or Site specific variables, in this preference order: local site, site
 parents, global
 - Selected emc_vars can be included by adding them to const_EXPORT_EMC_VARS; currently: deviceIP, serverIP, serverName
- \$<csvColumnKey>: Device specific variables extracted from supplied CSV file
- \$UD1, \$UD2, \$UD3, \$UD4: Device specific values extracted from device User Data 1-4 For the CSV variables, a CSV file must be provided with the following syntax:
- First row has column labels, which need to match the \$<csvColumnKey> variables, without the \$<>
- Subsequent rows contain data values for every device, one row per device.
- First column contains the device lookup, either the device IP or Serial Number

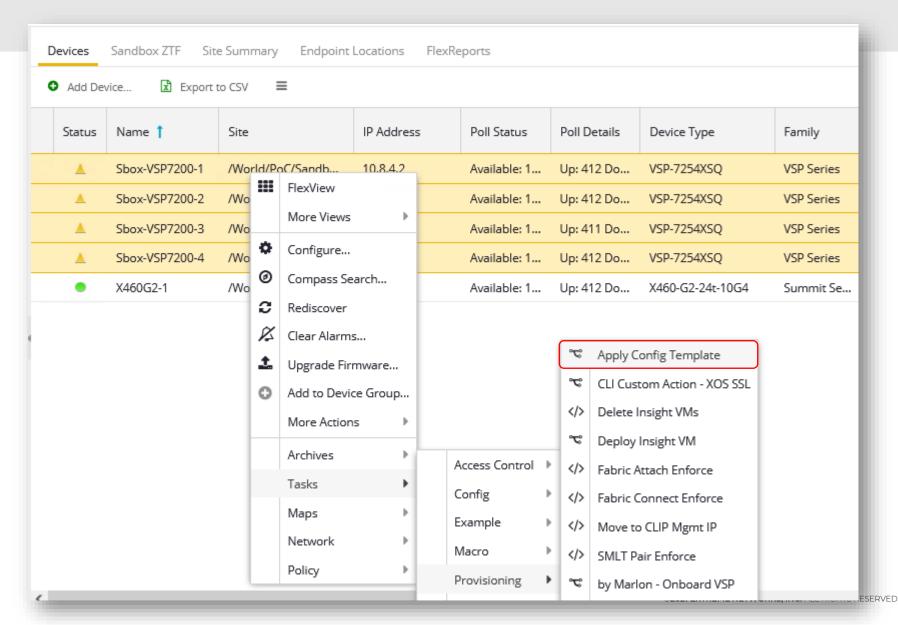
ASCII config template file and the CSV file must be placed on the XIQ-SE filesystem.

Commands which generate a confirmation prompt enter as example: "no spanning-tree mstp\ny"

Workflow manual execution



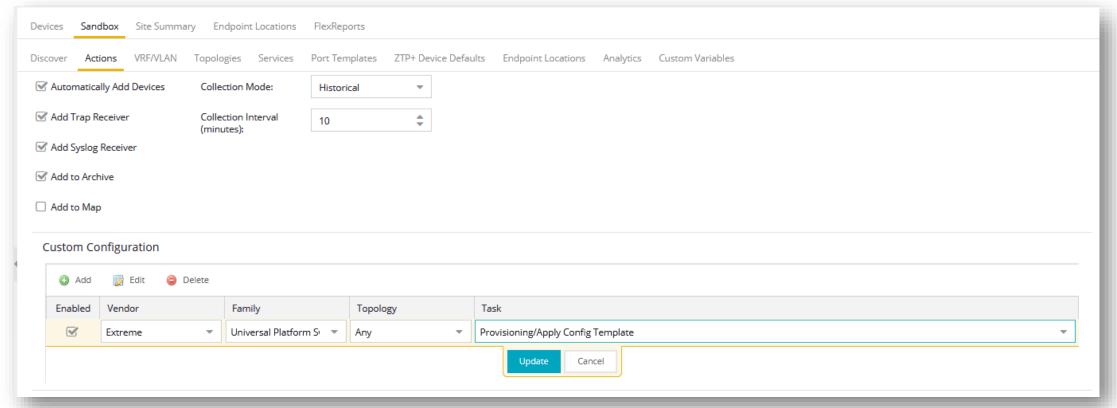
 Workflow can be manually run against 1 or many switches simultaneously



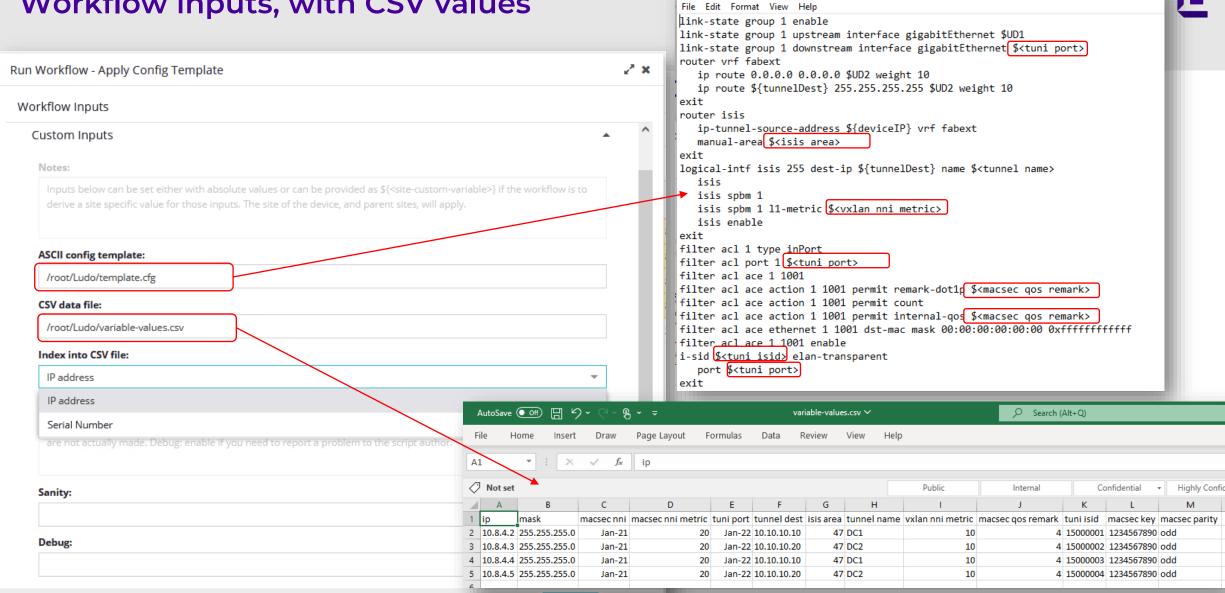
Workflow automatic execution during onboarding



- Workflow can be automatically run after ZTP+ onboarding, under XIQ-SE Site Actions
- In this case script will always run against 1 switch only, the onboarding switch



Workflow inputs, with CSV values



*template.cfg - Notepad

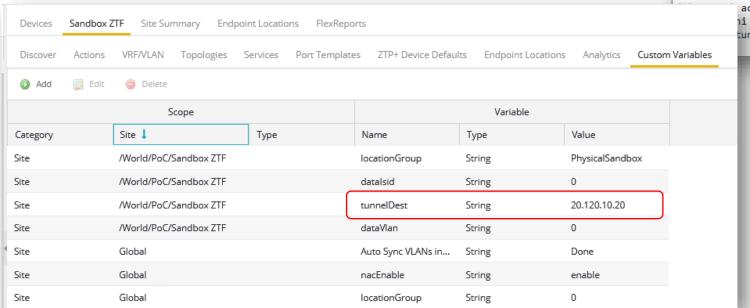
If your XIQ-SE was installed without "root" access, place the CSV file here instead: /usr/local/Extreme Networks/NetSight/appdata/logs/scripting/NetSight Server

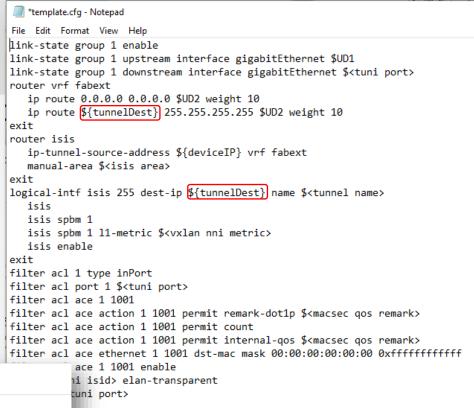


Workflow, site variables

E

- Template can also take \${var} variables
- Values for these variables are looked up in the Site Custom variables, in this preference order:
 - Site of device
 - Parent Site of device
 - Parent sites up to Root site
 - Global variable

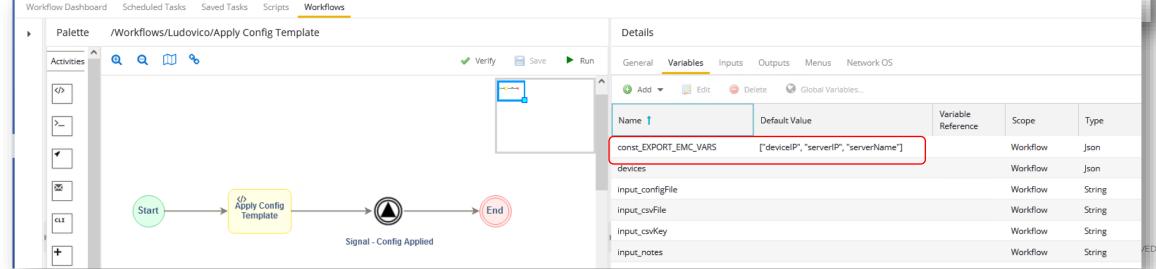




Workflow, emc_vars as variables

 Selected emc_vars can also be fed into the same \${var} space, by adding them to workflow variable const EXPORT EMC VARS

```
*template.cfg - Notepad
File Edit Format View Help
link-state group 1 enable
link-state group 1 upstream interface gigabitEthernet $UD1
link-state group 1 downstream interface gigabitEthernet $<tuni port>
router vrf fabext
  ip route 0.0.0.0 0.0.0.0 $UD2 weight 10
  ip route ${tunnelDest} 255.255.255.255 $UD2 weight 10
exit
router isis
  ip-tunnel-source-address ${deviceIP} vrf fabext
   manual-area $<isis area>
logical-intf isis 255 dest-ip ${tunnelDest} name $<tunnel name>
  isis
  isis spbm 1
  isis spbm 1 l1-metric $<vxlan nni metric>
  isis enable
exit
filter acl 1 type inPort
filter acl port 1 $<tuni port>
filter acl ace 1 1001
filter acl ace action 1 1001 permit remark-dot1p $<macsec gos remark>
filter acl ace action 1 1001 permit count
filter acl ace action 1 1001 permit internal-gos $<macsec gos remark>
filter acl ace ethernet 1 1001 dst-mac mask 00:00:00:00:00 0xfffffffffffff
filter acl ace 1 1001 enable
i-sid $<tuni isid> elan-transparent
```

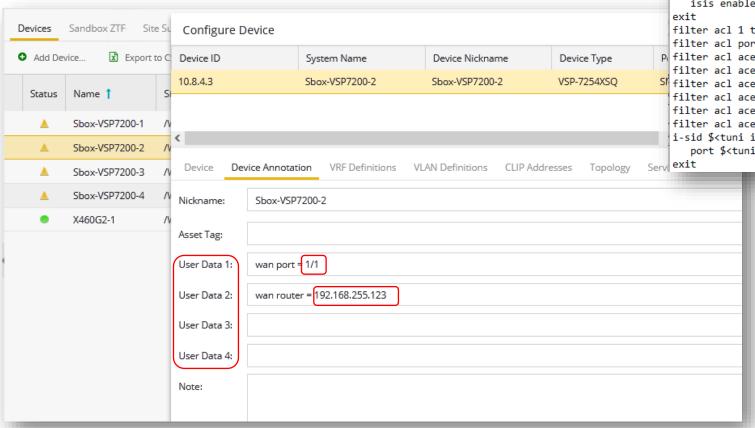




Workflow, UserData1-4 variables

E

- Values unique to each device can also be fetched from the device UserData1-4 fields
- Only value after "=" is used



```
*template.cfg - Notepad
  File Edit Format View Help
  link-state group 1 enable
  link-state group 1 upstream interface gigabitEthernet $UD1
  link-state group 1 downstream interface gigabitEthernet $<tuni port>
  router vrf fabext
    ip route 0.0.0.0 0.0.0.0 $UD2 weight 10
    ip route ${tunnelDest} 255.255.255.255 $UD2 weight 10
  exit
  router isis
    ip-tunnel-source-address ${deviceIP} vrf fabext
     manual-area $<isis area>
  logical-intf isis 255 dest-ip ${tunnelDest} name $<tunnel name>
    isis spbm 1
    isis spbm 1 l1-metric $<vxlan nni metric>
    isis enable
  filter acl 1 type inPort
  filter acl port 1 $<tuni port>
p filter acl ace 1 1001
  filter acl ace action 1 1001 permit remark-dot1p $<macsec gos remark>
filter acl ace action 1 1001 permit count
  filter acl ace action 1 1001 permit internal-gos $<macsec gos remark>
  filter acl ace ethernet 1 1001 dst-mac mask 00:00:00:00:00 0xfffffffffffff
  filter acl ace 1 1001 enable
  i-sid $<tuni isid> elan-transparent
     port $<tuni port>
```

Comparison of template variables



- Site variables \${var}: Useful to apply same values to all devices in same XIQ-SE Site. Or to apply same values to all devices in same sub-Sites
- CSV variables \$<var>: Useful to provide device specific values
- UserData variables \$UD1-4: Useful to provide device specific values, but for values obtained dynamically from the device itself (by another workflow or activity) and then make these available in this Apply Config Template workflow
- Emc_vars \${deviceIP}: Useful to feed some of these values into the same space as Site variables

Cisco velocity type statements: #if/#elseif/#else/#end



- The template file can also include #if/#elseif/#else/#end statement blocks
- To match Cisco velocity type statements
- The conditional string, inside "(" ")" will be evaluated using Python's eval() function, so any valid Python expression may be used
- Any of CSV variables [\$<var>], Site variables [\${var}] or UserData variables [\$UD1-4] can be inserted inside the conditional string, but they will always be evaluated as String values.
- For instance, to evaluate an integer value, insert the variable inside Python's int() method like this:

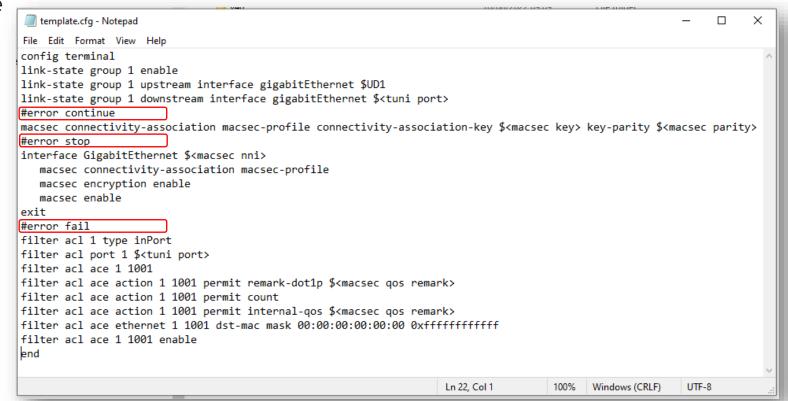
```
• #if (int($<myvar>) > 10)
```

```
template2.cfg - Notepad
File Edit Format View Help
config terminal
interface GigabitEthernet $<myport>
#if($<myport> == "1/1")
    name "first port"
#elseif($<myport> == "1/24")
    name "middle port"
#else
    name "last port"
#end
exit
```

Error mode: #error fail|stop|continue



- The template file can also include #error fail|stop|continue statement
- Determines the behaviour if a command in the template errors when executed on the switch
 - fail: workflow aborts immediately with an error
 - stop: no further commands from the template are executed, the workflow continues and does not fail
 - continue: execution of template commands continues even if commands error
- The default behaviour is fail



Commands which require Y/N confirmation prompt on device



- Some commands require "y" confirmation on certain devices (VOSS/ERS)
- To push such commands via the template, append "\ny" to those commands, as shown
- On EXOS the workflow will automatically "disable cli prompting" so this is not necessary

```
template.txt - Notepad

File Edit Format View Help

config term

interface gigabitEthernet 1/1-1/48

no spanning-tree mstp\ny
exit
end
```

Commands which prompt for interactive input on device



- Some commands which set passwords, prompt the user interactively to enter such passwords, sometimes twice
- To push such commands via the template, append the data (passwords) in the same sequence as they would be requested, separated by "//"
- The following example shows how this can be done for "config" which then prompts for:
 - Configuring from terminal or network [terminal]?
 To which, "term" will be fed
- The example also shows how to create an SNMPv3 privAuth user which will ask for both the priv password and auth password twice each

```
template.txt - Notepad

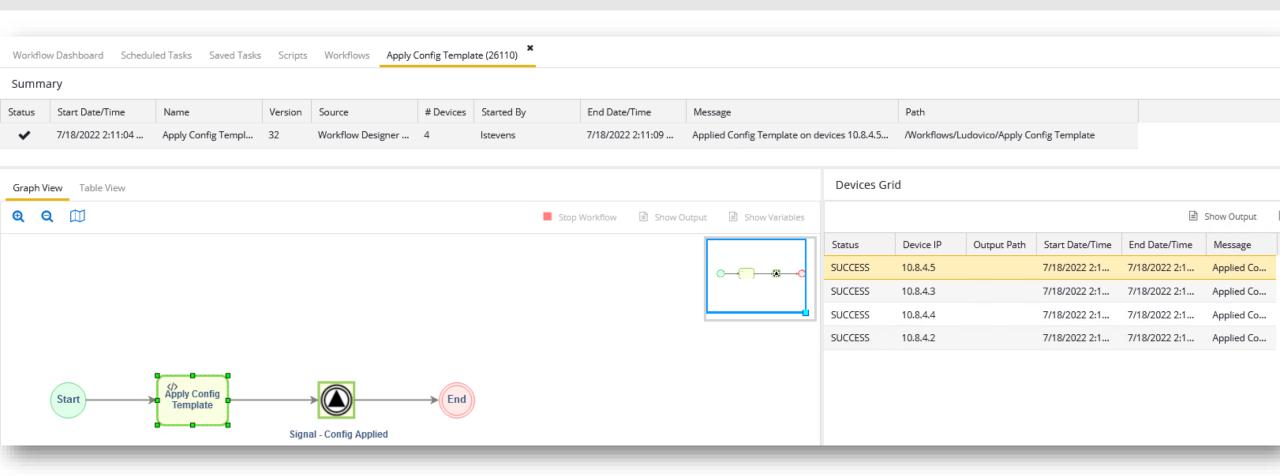
File Edit Format View Help

config // term

snmp-server user snmpuser1 group "snmprw" sha aes // snmpauthcred // snmpauthcred // snmpprivcred // snmpprivcred end
```

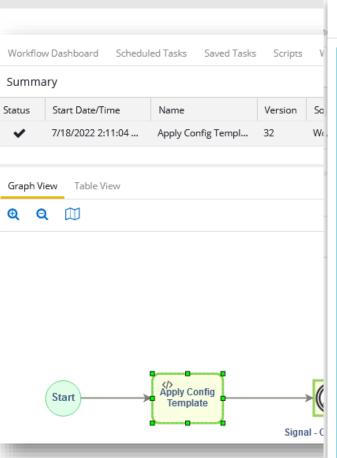
Workflow execution

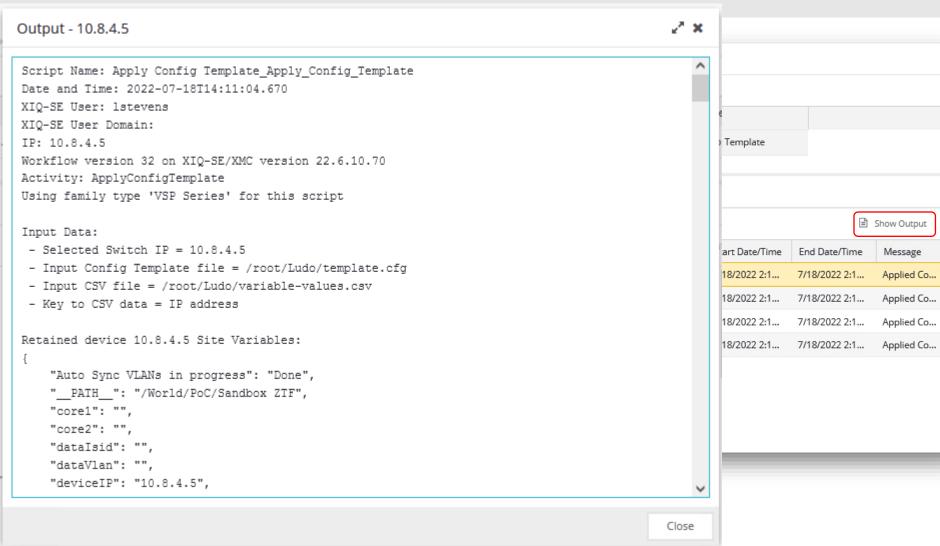




Workflow execution







Workflow Event signal



