EXTREME NETWORKS

Apply Config Template XIQ-SE workflow

Ludovico Stevens

Technical Marketing Engineering

July 2022

ADVANCE WITH US

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 four 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","deviceName","deviceSysOid","deviceType","family","serverIP","serverName","date"
- $<csvColumnKey>: Device specific variables extracted from supplied CSV file
- $[variableName]: Variables set from within the config template using `#eval variableName=()`
- $UD1, $UD2, $UD3, $UD4: Device specific values extracted from device User Data 1-4

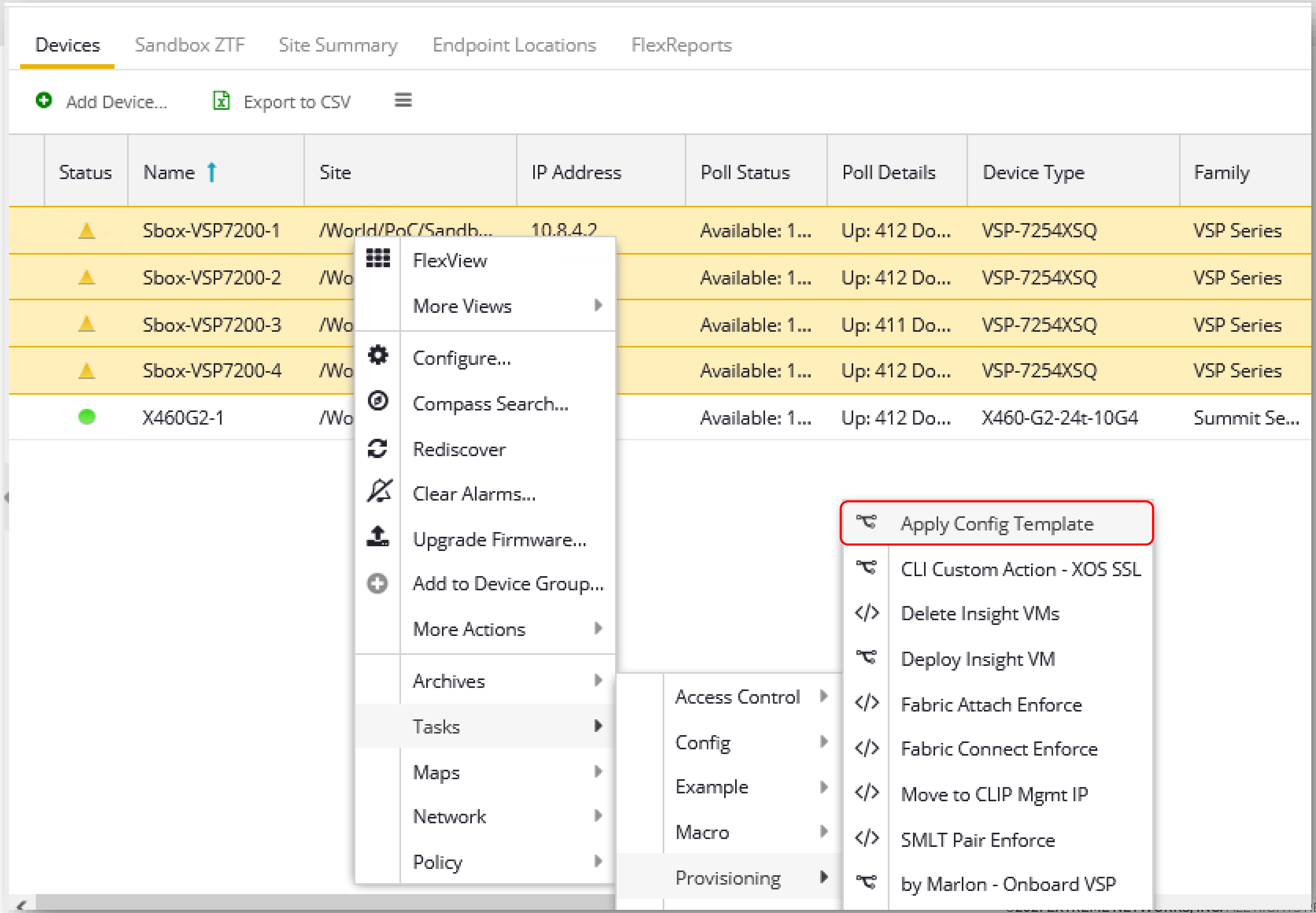
For the CSV variables, a CSV file must be provided with the following syntax:

- 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.

Workflow manual execution

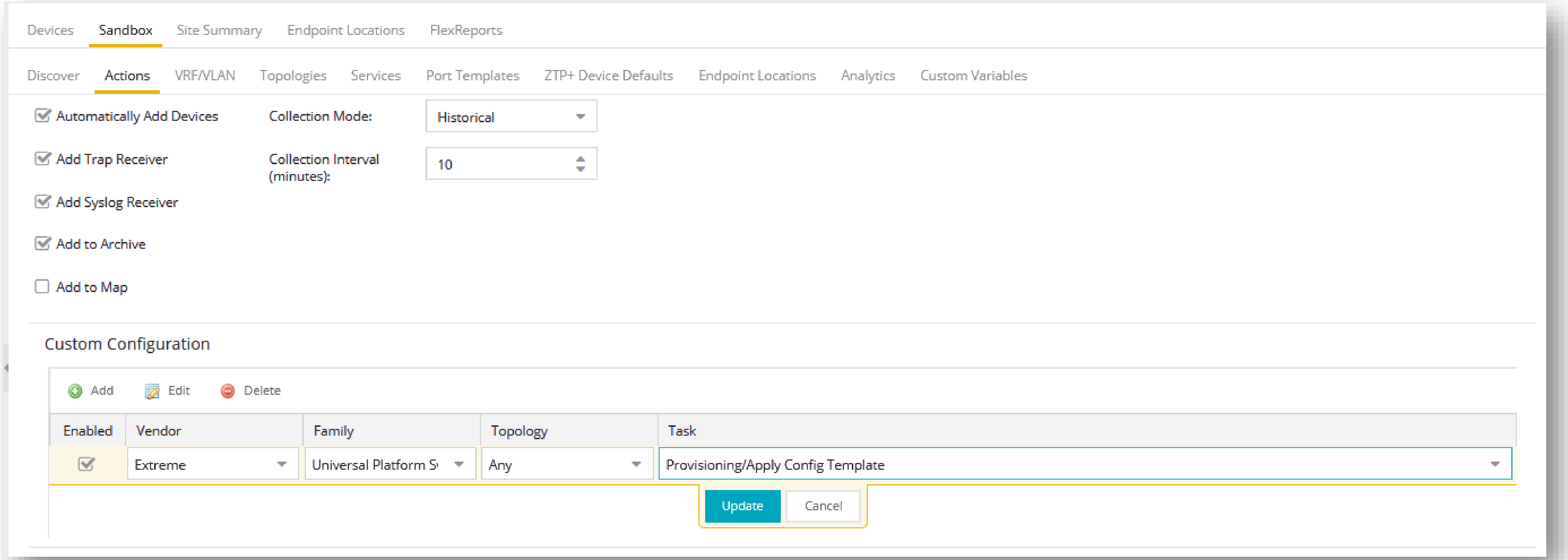
- Workflow can be manually run against 1 or many switches simultaneously



Status	Name	Site	IP Address	Poll Status	Poll Details	Device Type	Family
▲	Sbox-VSP7200-1	/World/PoC/Sandb...	10.8.4.2	Available: 1...	Up: 412 Do...	VSP-7254XSQ	VSP Series
▲	Sbox-VSP7200-2	/Wo		Available: 1...	Up: 412 Do...	VSP-7254XSQ	VSP Series
▲	Sbox-VSP7200-3	/Wo		Available: 1...	Up: 411 Do...	VSP-7254XSQ	VSP Series
▲	Sbox-VSP7200-4	/Wo		Available: 1...	Up: 412 Do...	VSP-7254XSQ	VSP Series
●	X460G2-1	/Wo		Available: 1...	Up: 412 Do...	X460-G2-24t-10G4	Summit Se...

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



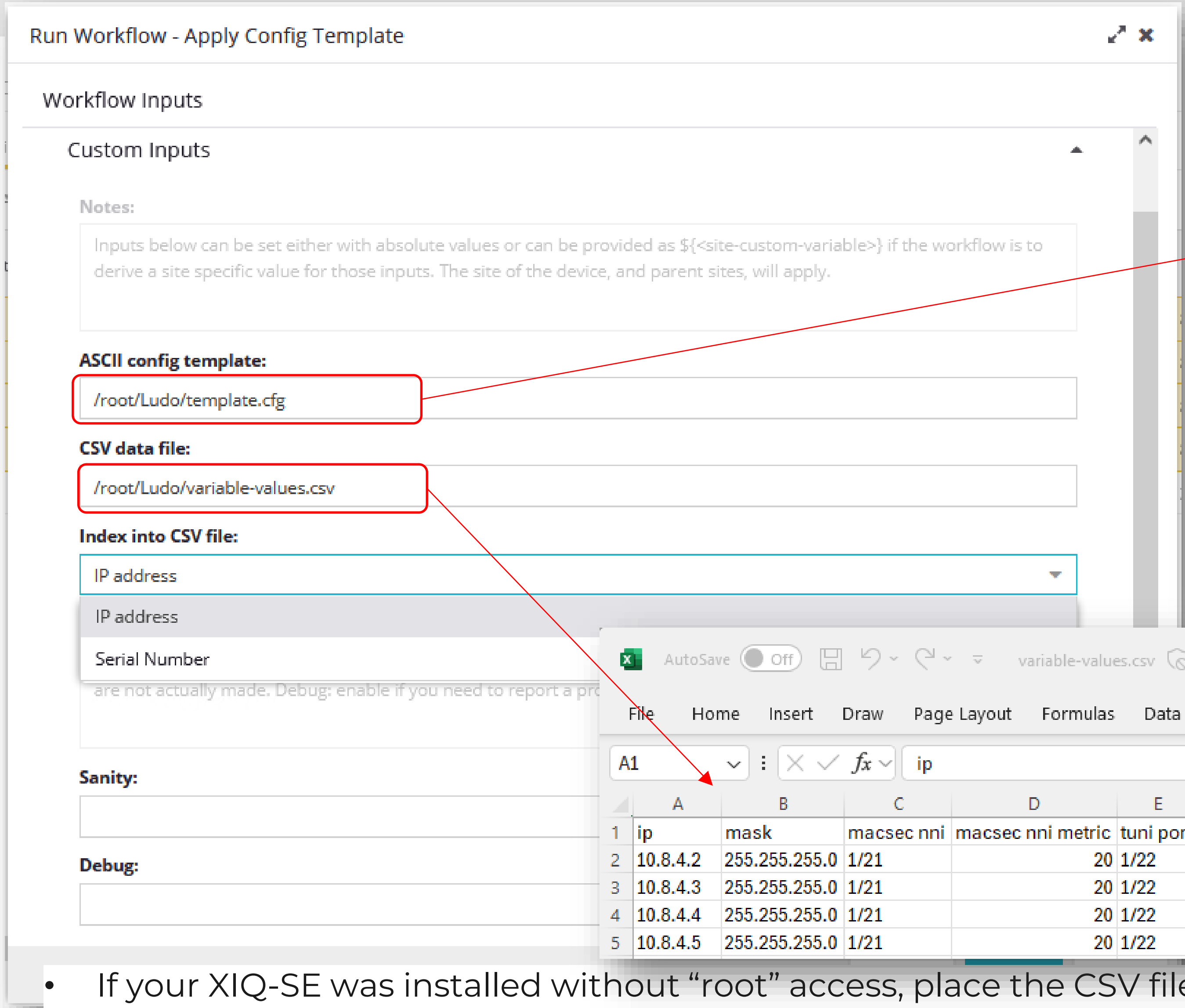
☑ Automatically Add Devices
☑ Add Trap Receiver
☑ Add Syslog Receiver
☑ Add to Archive
☐ Add to Map

Collection Mode: Historical
Collection Interval (minutes): 10

Custom Configuration

Enabled	Vendor	Family	Topology	Task
✓	Extreme	Universal Platform S	Any	Provisioning/Apply Config Template

Workflow inputs, with CSV values


Run Workflow - Apply Config Template

Workflow Inputs

Custom Inputs

Notes:

Inputs below can be set either with absolute values or can be provided as ${<site-custom-variable>} if the workflow is to derive a site specific value for those inputs. The site of the device, and parent sites, will apply.

ASCII config template: /root/Ludo/template.cfg

CSV data file: /root/Ludo/variable-values.csv

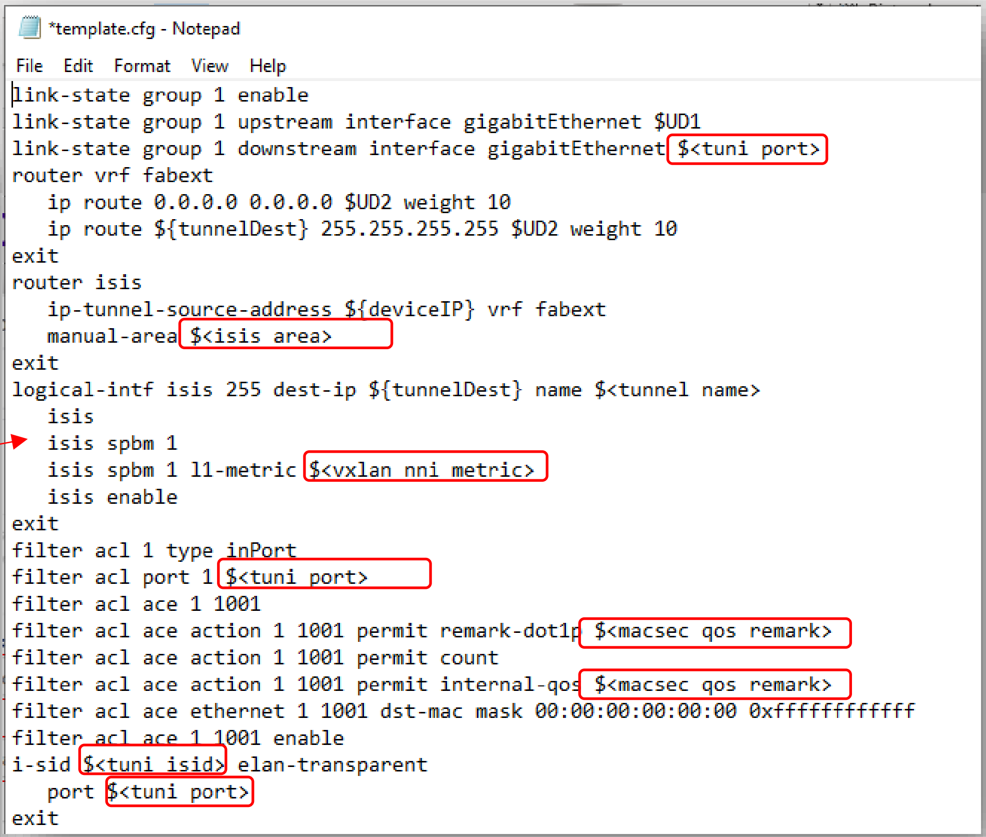
Index into CSV file: IP address

- IP address
- Serial Number

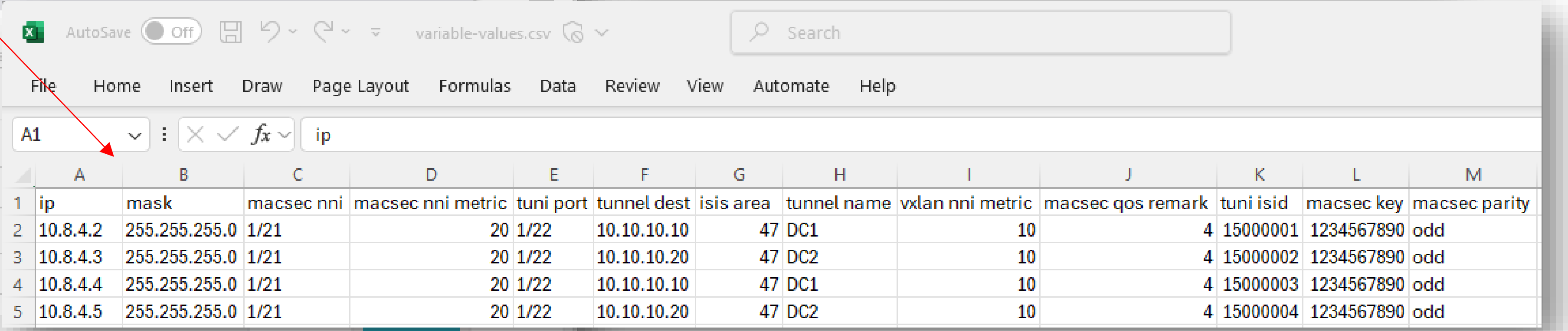
are not actually made. Debug: enable if you need to report a pro

Sanity:

Debug:


*template.cfg - Notepad

```
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>
exit
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 qos remark>
filter acl ace action 1 1001 permit count
filter acl ace action 1 1001 permit internal-qos $<macsec qos remark>
filter acl ace ethernet 1 1001 dst-mac mask 00:00:00:00:00:00 0xffffffffffff
filter acl ace 1 1001 enable
i-sid $<tuni isid> elan-transparent
   port $<tuni port>
exit
```

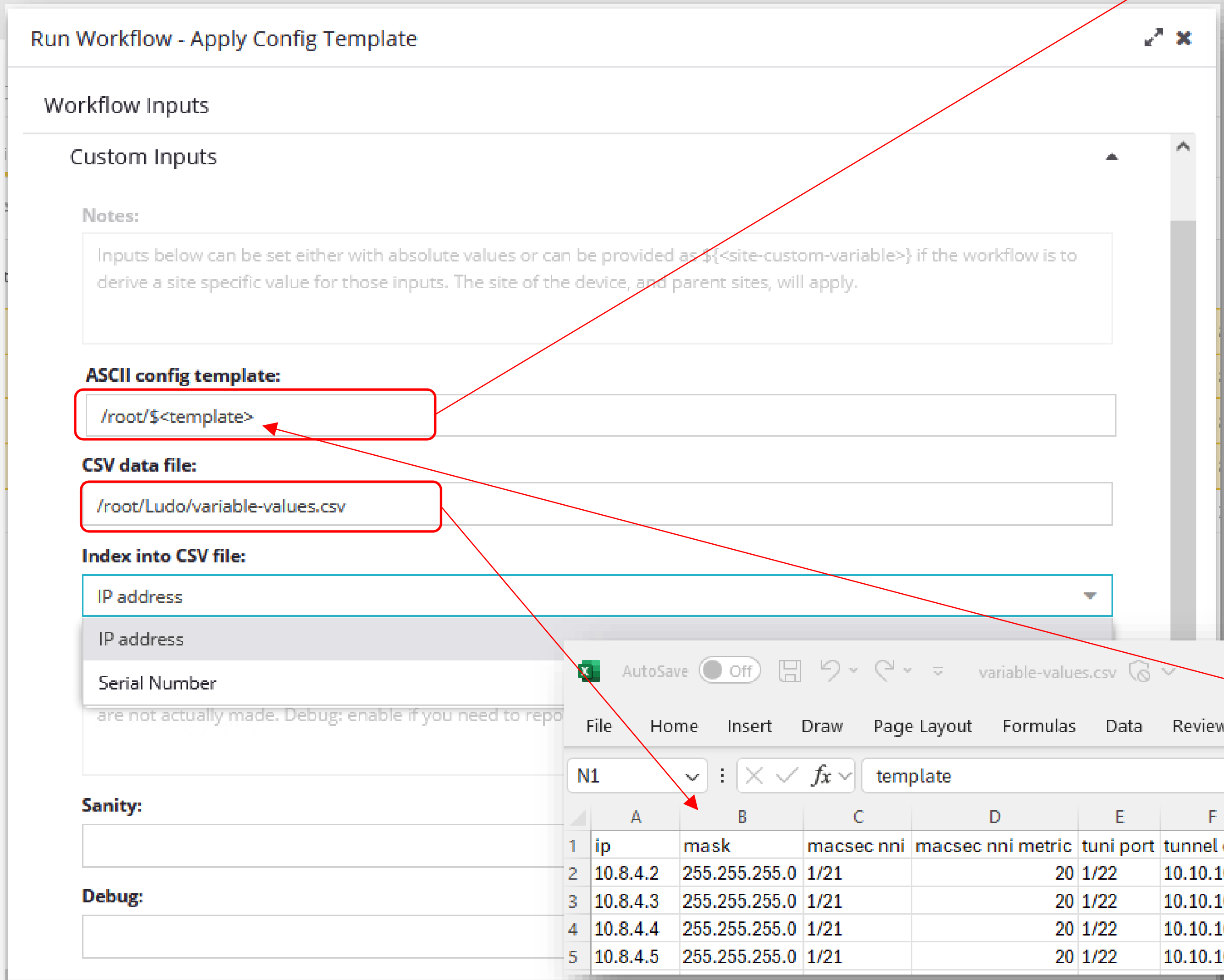

variable-values.csv

A1 | ip

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	ip	mask	macsec nni	macsec nni metric	tuni port	tunnel dest	isis area	tunnel name	vxlan nni metric	macsec qos remark	tuni isid	macsec key	macsec parity
2	10.8.4.2	255.255.255.0	1/21	20	1/22	10.10.10.10	47	DC1	10	4	15000001	1234567890	odd
3	10.8.4.3	255.255.255.0	1/21	20	1/22	10.10.10.20	47	DC2	10	4	15000002	1234567890	odd
4	10.8.4.4	255.255.255.0	1/21	20	1/22	10.10.10.10	47	DC1	10	4	15000003	1234567890	odd
5	10.8.4.5	255.255.255.0	1/21	20	1/22	10.10.10.20	47	DC2	10	4	15000004	1234567890	odd

- 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 inputs, with CSV values



Run Workflow - Apply Config Template

Workflow Inputs

Custom Inputs

Notes:

Inputs below can be set either with absolute values or can be provided as ${<site-custom-variable>} if the workflow is to derive a site specific value for those inputs. The site of the device, and parent sites, will apply.

ASCII config template: /root/$<template>

CSV data file: /root/Ludo/variable-values.csv

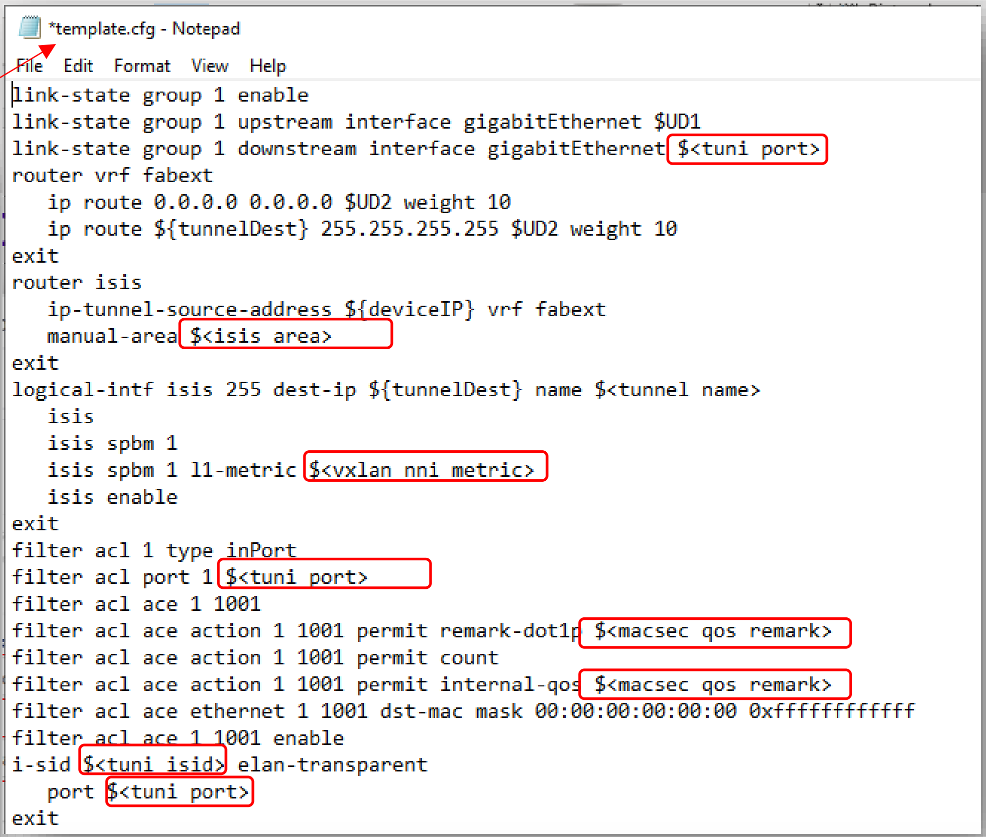
Index into CSV file: IP address

- IP address
- Serial Number

are not actually made. Debug: enable if you need to repo

Sanity:

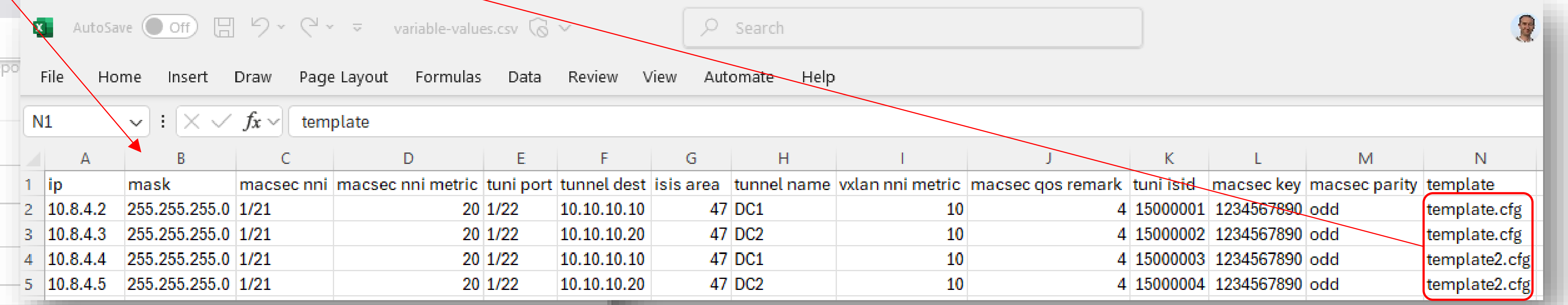
Debug:



*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>
exit
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 qos remark>
filter acl ace action 1 1001 permit count
filter acl ace action 1 1001 permit internal-qos $<macsec qos remark>
filter acl ace ethernet 1 1001 dst-mac mask 00:00:00:00:00 0xffffffffffff
filter acl ace 1 1001 enable
i-sid $<tuni isid> elan-transparent
   port $<tuni port>
exit
```



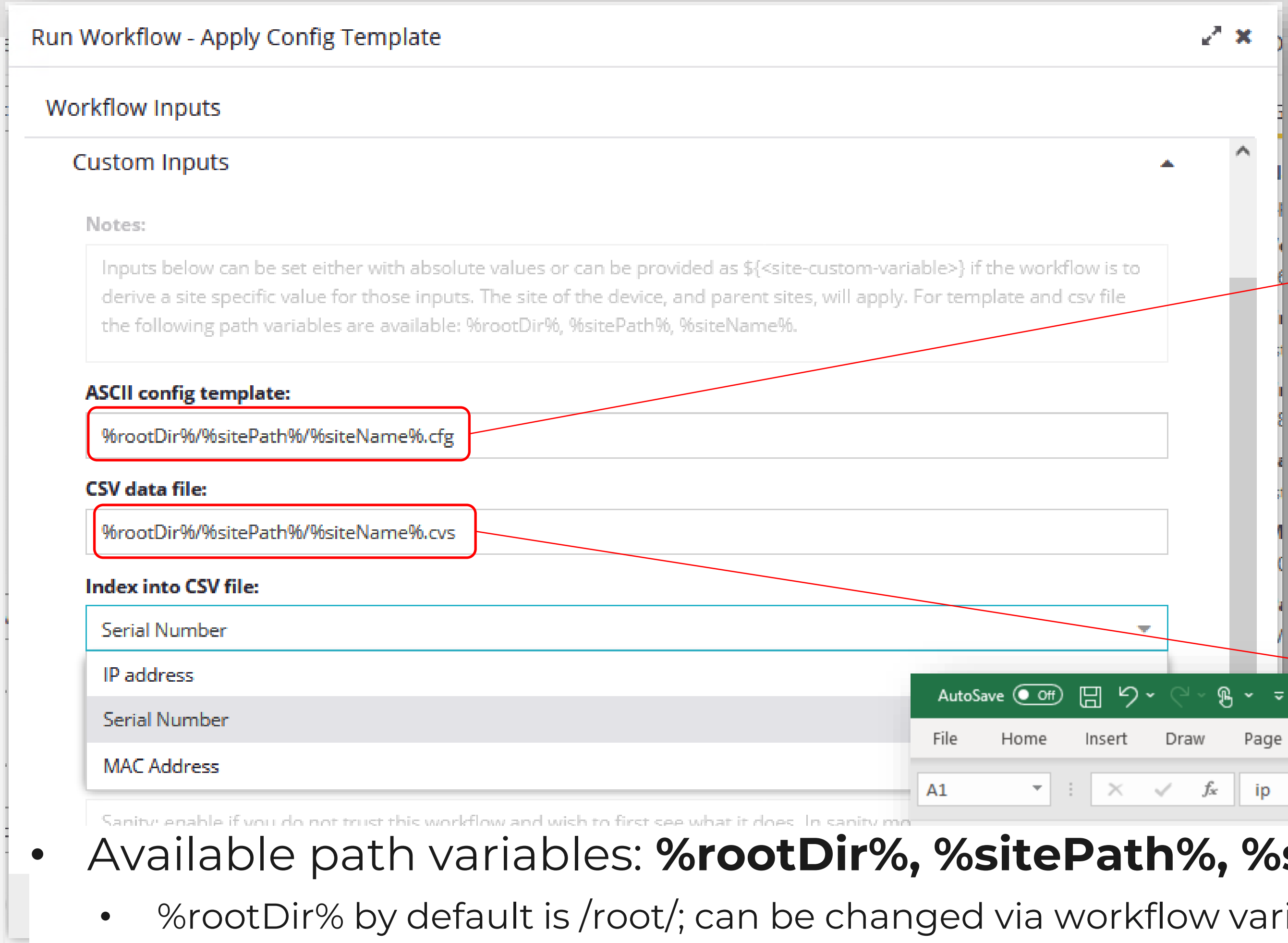
variable-values.csv

N1 | template

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	ip	mask	macsec nni	macsec nni metric	tuni port	tunnel dest	isis area	tunnel name	vxlan nni metric	macsec qos remark	tuni isid	macsec key	macsec parity	template
2	10.8.4.2	255.255.255.0	1/21	20	1/22	10.10.10.10	47	DC1	10	4	15000001	1234567890	odd	template.cfg
3	10.8.4.3	255.255.255.0	1/21	20	1/22	10.10.10.20	47	DC2	10	4	15000002	1234567890	odd	template.cfg
4	10.8.4.4	255.255.255.0	1/21	20	1/22	10.10.10.10	47	DC1	10	4	15000003	1234567890	odd	template2.cfg
5	10.8.4.5	255.255.255.0	1/21	20	1/22	10.10.10.20	47	DC2	10	4	15000004	1234567890	odd	template2.cfg

- Note that the CSV can also be used to specify the template file to use (if you have more than 1 template)

Workflow, path variables



Run Workflow - Apply Config Template

Workflow Inputs

Custom Inputs

Notes:

Inputs below can be set either with absolute values or can be provided as ${<site-custom-variable>} if the workflow is to derive a site specific value for those inputs. The site of the device, and parent sites, will apply. For template and csv file the following path variables are available: %rootDir%, %sitePath%, %siteName%.

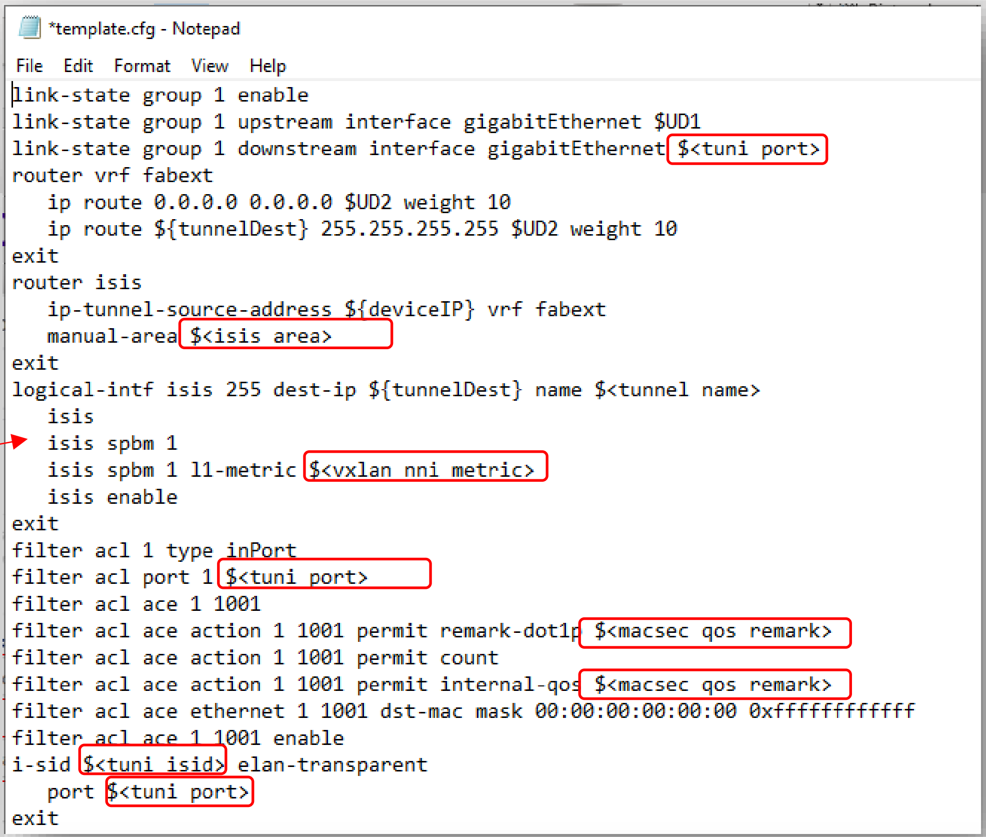
ASCII config template: %rootDir%/%sitePath%/%siteName%.cfg

CSV data file: %rootDir%/%sitePath%/%siteName%.cvs

Index into CSV file: Serial Number

- IP address
- Serial Number
- MAC Address

Sanity: enable if you do not trust this workflow and wish to first see what it does. In sanity mo



*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>
exit
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 qos remark>
filter acl ace action 1 1001 permit count
filter acl ace action 1 1001 permit internal-qos $<macsec qos remark>
filter acl ace ethernet 1 1001 dst-mac mask 00:00:00:00:00:00 0xffffffffffff
filter acl ace 1 1001 enable
i-sid $<tuni isid> elan-transparent
  port $<tuni port>
exit
```

AutoSave Off
variable-values.csv
Search (Alt+Q)
File Home Insert Draw Page Layout Formulas Data Review View Help
A1

ip

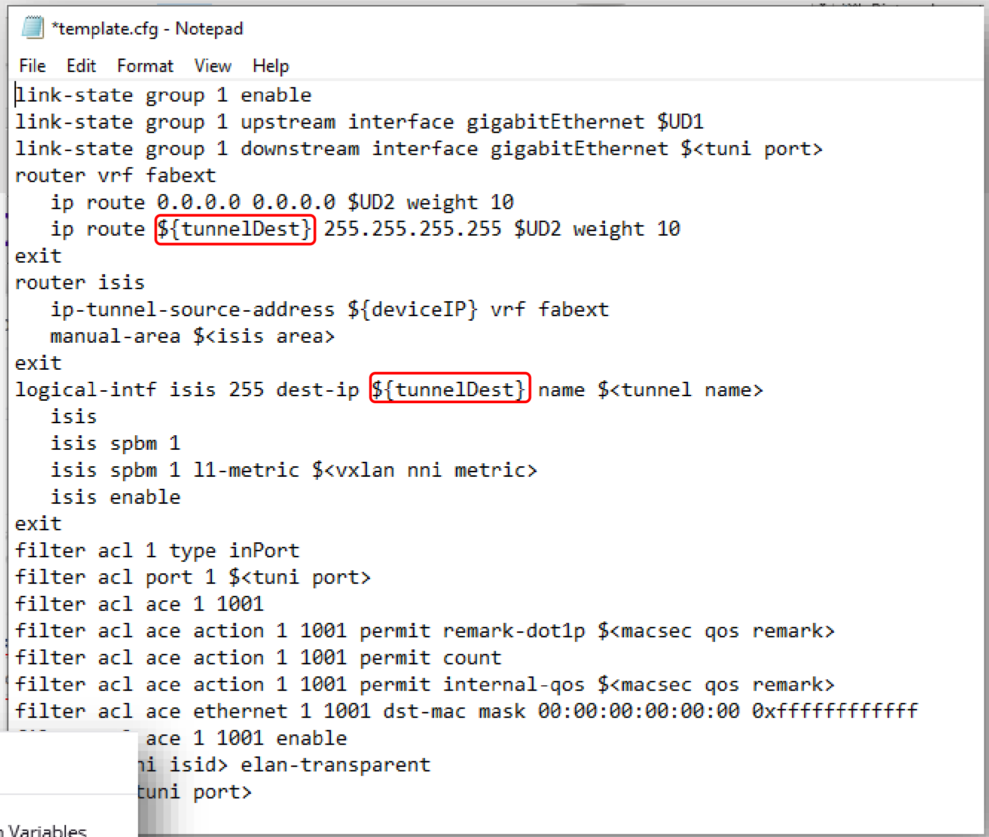
nly Confid

parity

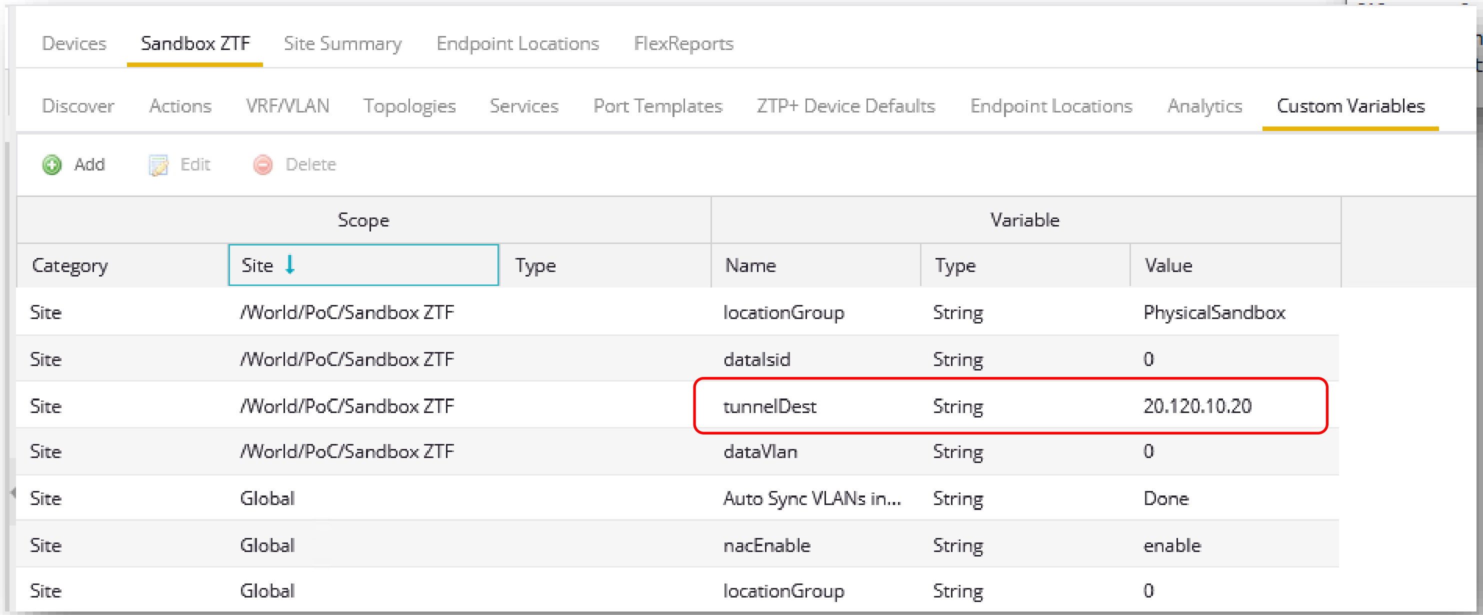
- Available path variables: **%rootDir%, %sitePath%, %siteName%**
  - %rootDir% by default is /root/; can be changed via workflow variable const_ROOT_PATH_VAR
  - %sitePath% and %siteName% are set based on site path of device; e.g. if device is in "/World/CTC-Reading/VSP Sandbox" then %sitePath% = "World/CTC-Reading" and %siteName% = "VSP Sandbox"
- Can use these to have different CSV & template files per site

Workflow, site variables

- 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



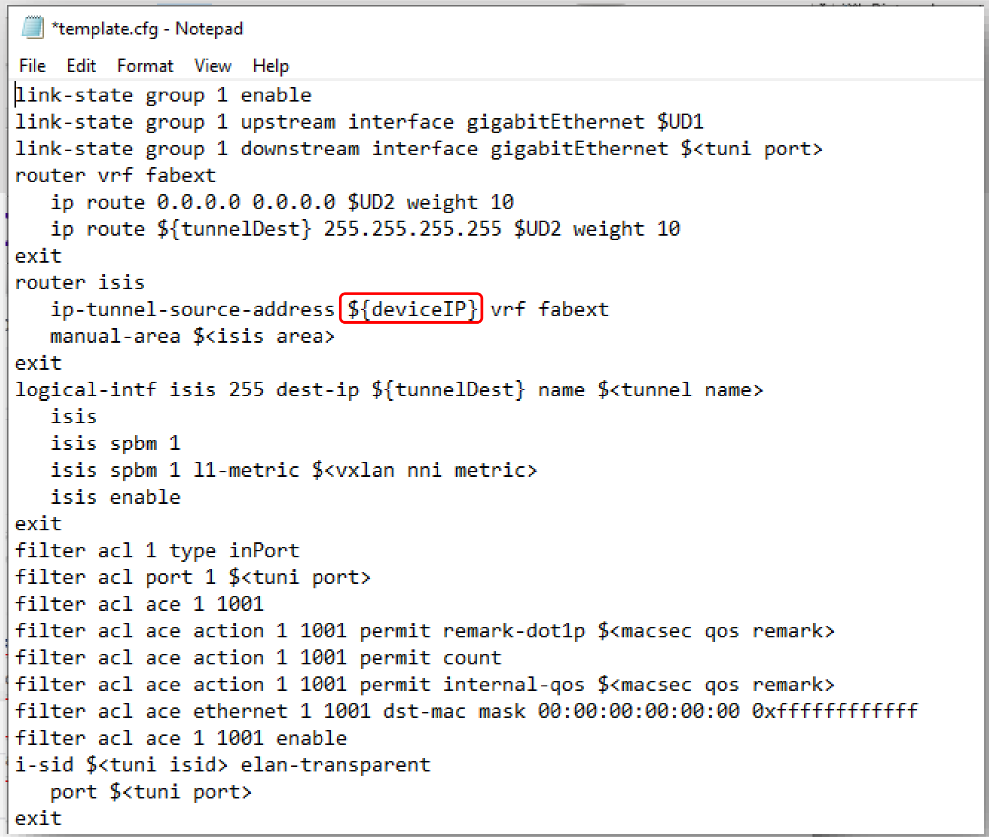
```
link-state group 1 enable
link-state group 1 upstream interface gigabitEthernet $UD1
link-state group 1 downstream interface gigabitEthernet $<tuni port>
router vrf fabext
   ip route 0.0.0.0 0.0.0.0 $UD2 weight 10
   ip route ${tunnelDest} 255.255.255.255 $UD2 weight 10
exit
router isis
   ip-tunnel-source-address ${deviceIP} vrf fabext
   manual-area $<isis area>
exit
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 qos remark>
filter acl ace action 1 1001 permit count
filter acl ace action 1 1001 permit internal-qos $<macsec qos remark>
filter acl ace ethernet 1 1001 dst-mac mask 00:00:00:00:00:00 0xffffffffffff
          ace 1 1001 enable
         i isid> elan-transparent
         tuni port>
```



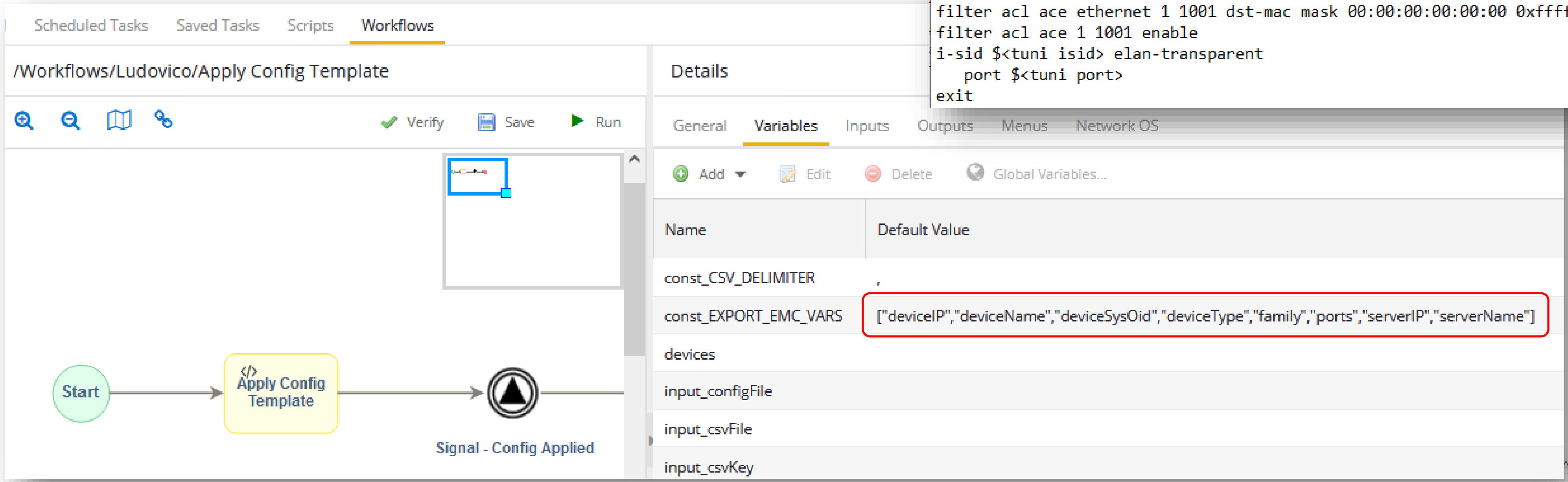
Scope			Variable		
Category	Site	Type	Name	Type	Value
Site	/World/PoC/Sandbox ZTF		locationGroup	String	PhysicalSandbox
Site	/World/PoC/Sandbox ZTF		dataIsid	String	0
Site	/World/PoC/Sandbox ZTF		tunnelDest	String	20.120.10.20
Site	/World/PoC/Sandbox ZTF		dataVlan	String	0
Site	Global		Auto Sync VLANs in...	String	Done
Site	Global		nacEnable	String	enable
Site	Global		locationGroup	String	0

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

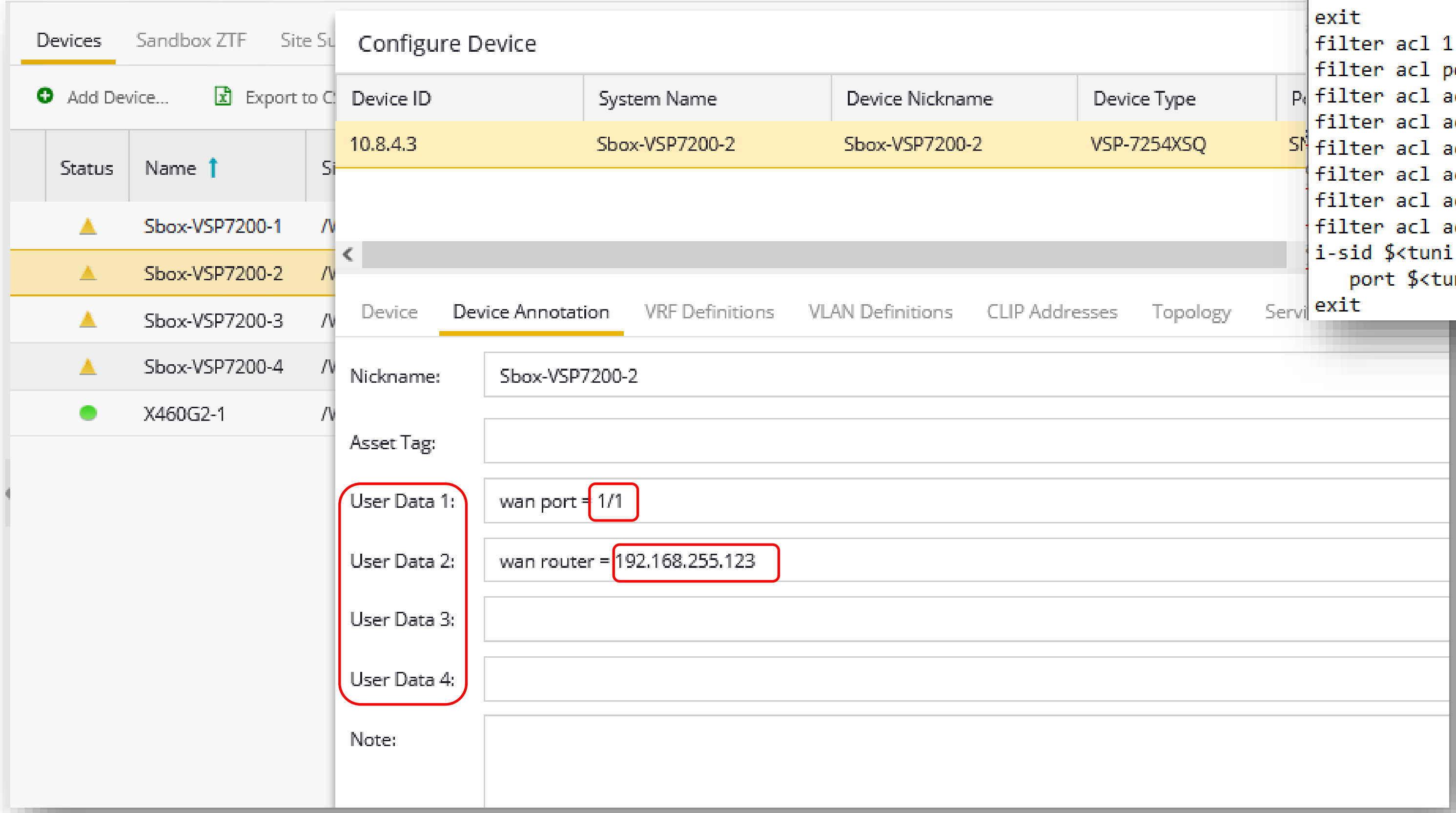
```
link-state group 1 enable
link-state group 1 upstream interface gigabitEthernet $UD1
link-state group 1 downstream interface gigabitEthernet $<tuni port>
router vrf fabext
   ip route 0.0.0.0 0.0.0.0 $UD2 weight 10
   ip route ${tunnelDest} 255.255.255.255 $UD2 weight 10
exit
router isis
   ip-tunnel-source-address ${deviceIP} vrf fabext
   manual-area $<isis area>
exit
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 qos remark>
filter acl ace action 1 1001 permit count
filter acl ace action 1 1001 permit internal-qos $<macsec qos remark>
filter acl ace ethernet 1 1001 dst-mac mask 00:00:00:00:00:00 0xffffffffffff
filter acl ace 1 1001 enable
i-sid $<tuni isid> elan-transparent
   port $<tuni port>
exit
```



Name	Default Value
const_CSV_DELIMITER	,
const_EXPORT_EMC_VARS	["deviceIP","deviceName","deviceSysOid","deviceType","family","ports","serverIP","serverName"]
devices	
input_configFile	
input_csvFile	
input_csvKey	

Workflow, UserData1-4 variables

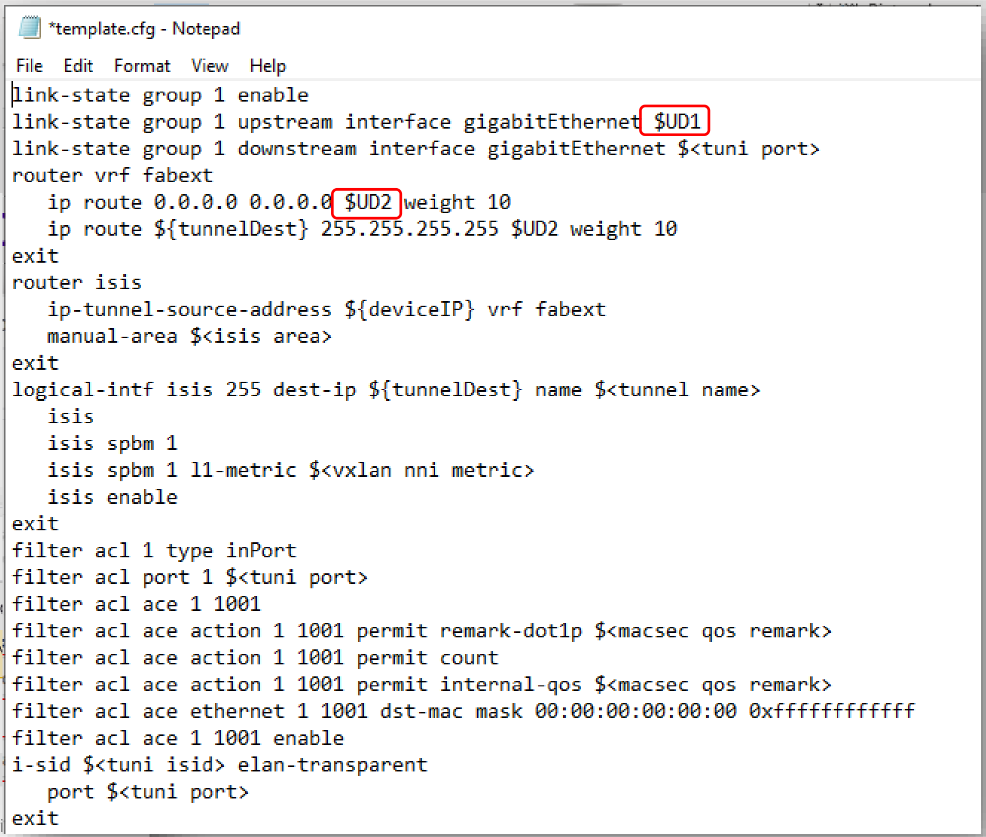
- Values unique to each device can also be fetched from the device UserData1-4 fields
- Only value after "=" is used



Status	Name
▲	Sbox-VSP7200-1
▲	Sbox-VSP7200-2
▲	Sbox-VSP7200-3
▲	Sbox-VSP7200-4
●	X460G2-1

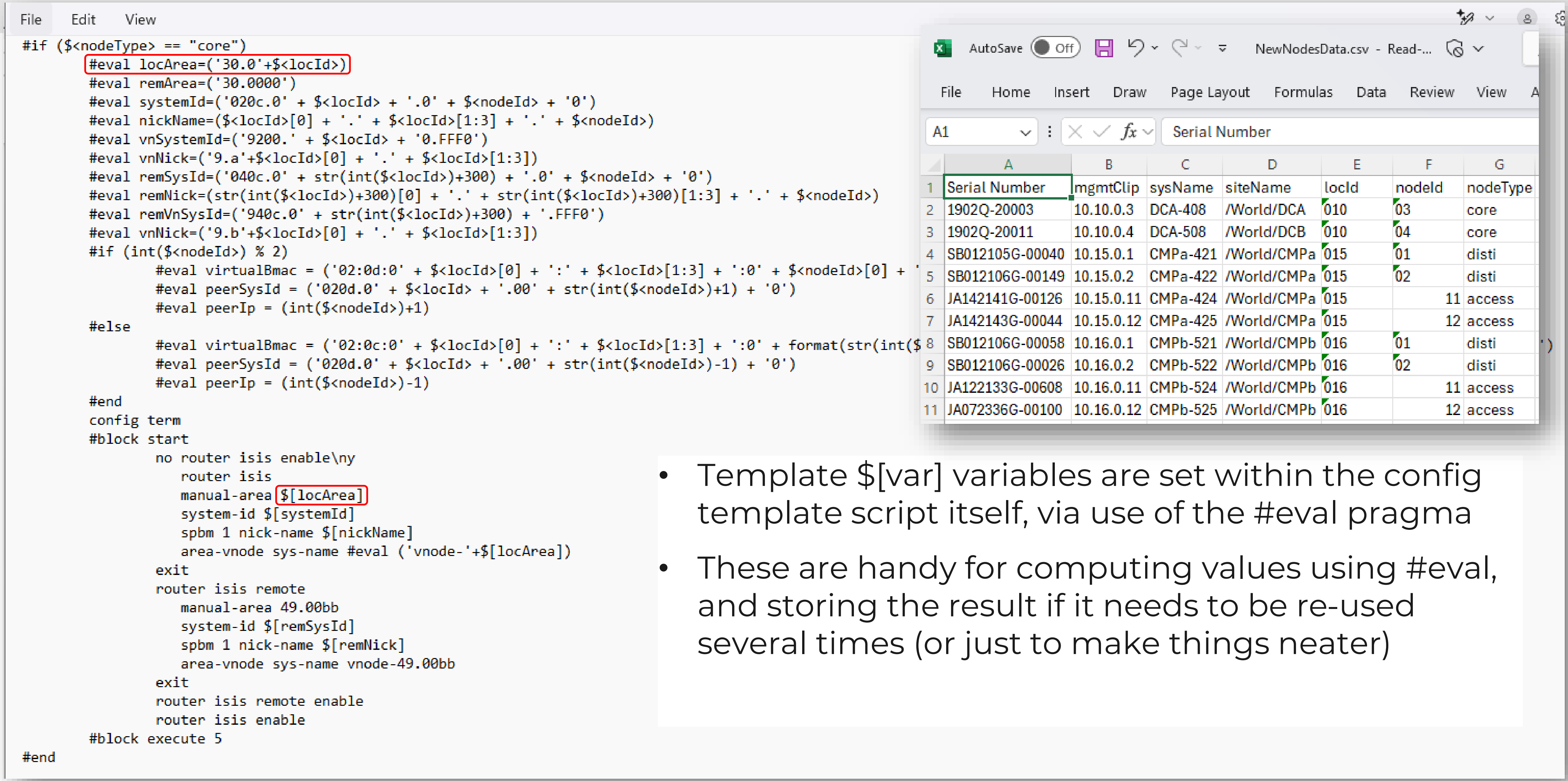
Device ID	System Name	Device Nickname	Device Type
10.8.4.3	Sbox-VSP7200-2	Sbox-VSP7200-2	VSP-7254XSQ

Nickname:	Sbox-VSP7200-2
Asset Tag:	
User Data 1:	wan port = 1/1
User Data 2:	wan router = 192.168.255.123
User Data 3:	
User Data 4:	
Note:	

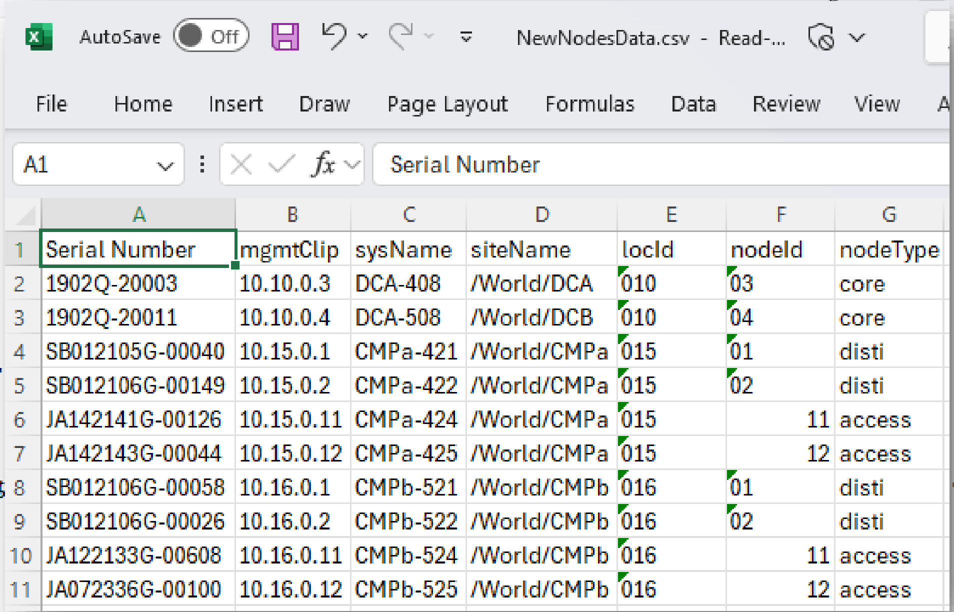


```
link-state group 1 enable
link-state group 1 upstream interface gigabitEthernet $UD1
link-state group 1 downstream interface gigabitEthernet $<tuni port>
router vrf fabext
   ip route 0.0.0.0 0.0.0.0 $UD2 weight 10
   ip route ${tunnelDest} 255.255.255.255 $UD2 weight 10
exit
router isis
   ip-tunnel-source-address ${deviceIP} vrf fabext
   manual-area $<isis area>
exit
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 qos remark>
filter acl ace action 1 1001 permit count
filter acl ace action 1 1001 permit internal-qos $<macsec qos remark>
filter acl ace ethernet 1 1001 dst-mac mask 00:00:00:00:00:00 0xffffffffffff
filter acl ace 1 1001 enable
i-sid $<tuni isid> elan-transparent
   port $<tuni port>
exit
```

Workflow, Template eval variables



```
#if ($<nodeType> == "core")
        #eval locArea=('30.0'+$<locId>)
        #eval remArea=('30.0000')
        #eval systemId=('020c.0' + $<locId> + '.0' + $<nodeId> + '0')
        #eval nickName=($<locId>[0] + '.' + $<locId>[1:3] + '.' + $<nodeId>)
        #eval vnSystemId=('9200.' + $<locId> + '0.FFF0')
        #eval vnNick=('9.a'+$<locId>[0] + '.' + $<locId>[1:3])
        #eval remSysId=('040c.0' + str(int($<locId>)+300) + '.0' + $<nodeId> + '0')
        #eval remNick=(str(int($<locId>)+300)[0] + '.' + str(int($<locId>)+300)[1:3] + '.' + $<nodeId>)
        #eval remVnSysId=('940c.0' + str(int($<locId>)+300) + '.FFF0')
        #eval vnNick=('9.b'+$<locId>[0] + '.' + $<locId>[1:3])
        #if (int($<nodeId>) % 2)
                #eval virtualBmac = ('02:0d:0' + $<locId>[0] + ':' + $<locId>[1:3] + ':0' + $<nodeId>[0] + '
                #eval peerSysId = ('020d.0' + $<locId> + '.00' + str(int($<nodeId>)+1) + '0')
                #eval peerIp = (int($<nodeId>)+1)
        #else
                #eval virtualBmac = ('02:0c:0' + $<locId>[0] + ':' + $<locId>[1:3] + ':0' + format(str(int($
                #eval peerSysId = ('020d.0' + $<locId> + '.00' + str(int($<nodeId>)-1) + '0')
                #eval peerIp = (int($<nodeId>)-1)
        #end
        config term
        #block start
                no router isis enable\ny
                   router isis
                   manual-area $[locArea]
                   system-id $[systemId]
                   spbm 1 nick-name $[nickName]
                   area-vnode sys-name #eval ('vnode-'+$[locArea])
                exit
                router isis remote
                   manual-area 49.00bb
                   system-id $[remSysId]
                   spbm 1 nick-name $[remNick]
                   area-vnode sys-name vnode-49.00bb
                exit
                router isis remote enable
                router isis enable
        #block execute 5
#end
```



	A	B	C	D	E	F	G
1	Serial Number	mgmtClip	sysName	siteName	locId	nodeId	nodeType
2	1902Q-20003	10.10.0.3	DCA-408	/World/DCA	010	03	core
3	1902Q-20011	10.10.0.4	DCA-508	/World/DCB	010	04	core
4	SB012105G-00040	10.15.0.1	CMPa-421	/World/CMPa	015	01	disti
5	SB012106G-00149	10.15.0.2	CMPa-422	/World/CMPa	015	02	disti
6	JA142141G-00126	10.15.0.11	CMPa-424	/World/CMPa	015	11	access
7	JA142143G-00044	10.15.0.12	CMPa-425	/World/CMPa	015	12	access
8	SB012106G-00058	10.16.0.1	CMPb-521	/World/CMPb	016	01	disti
9	SB012106G-00026	10.16.0.2	CMPb-522	/World/CMPb	016	02	disti
10	JA122133G-00608	10.16.0.11	CMPb-524	/World/CMPb	016	11	access
11	JA072336G-00100	10.16.0.12	CMPb-525	/World/CMPb	016	12	access

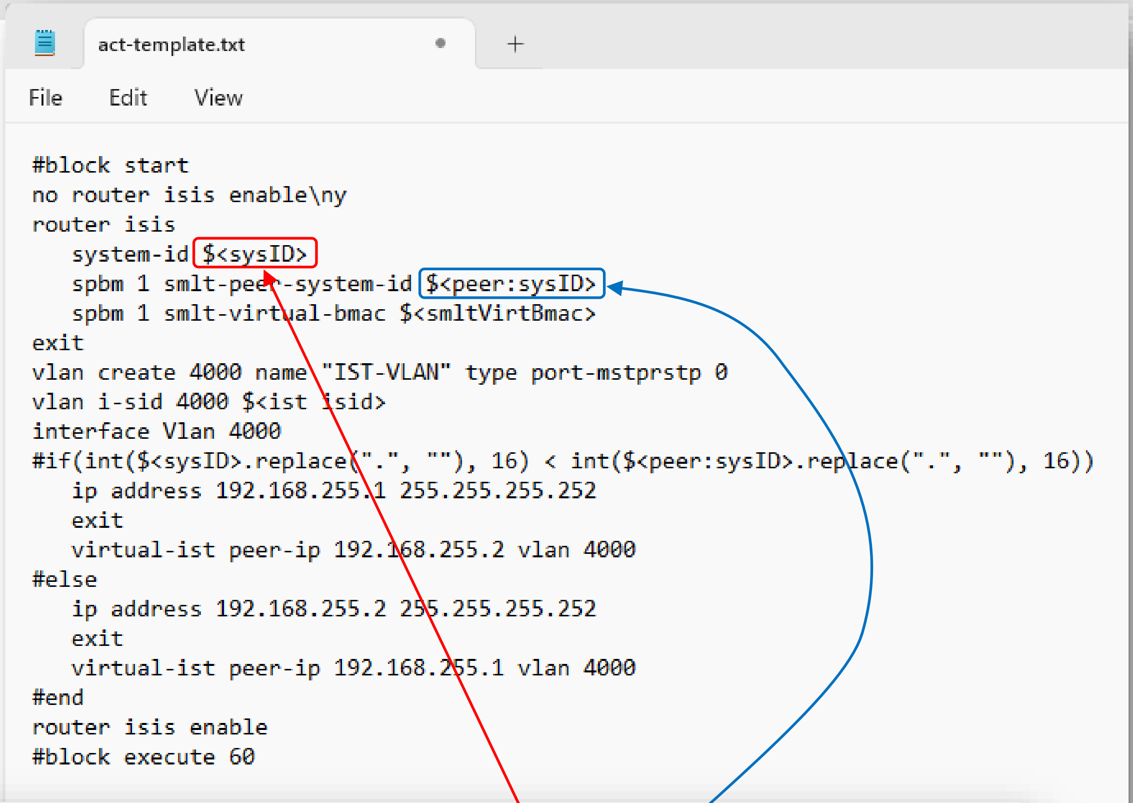
- Template $[var] variables are set within the config template script itself, via use of the #eval pragma
- These are handy for computing values using #eval, and storing the result if it needs to be re-used several times (or just to make things neater)

Comparison of template variables

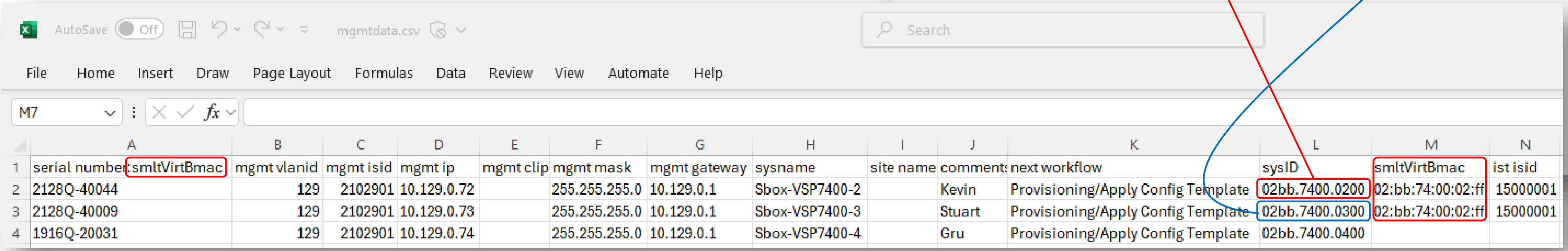
- Site variables **${var}**: Useful to apply same values to all devices in same XIQ-SE Site. Or to apply same values to all devices in same sub-Sites
- CSV variables **$<var>**: Useful to provide device specific values
- UserData variables **$UD1-4**: Useful to provide device specific values, but for values obtained dynamically from the device itself (by another upstream workflow or activity) and then make these available in this Apply Config Template workflow
- **Emc_vars** ${deviceIP}: Useful to feed some of these values into the same space as Site variables
- Template eval variables **$[var]**: Useful to compute new values within the template file and be able to store and re-use these values via a variable

CSV “peer:”values

- In the example shown, assume that the workflow is run against switch with S/N 2128Q-40044
- Variable $<sysID> will thus resolve to 02bb.7400.0200
- Cell A1 in the CSV, normally is just a label of the lookup index to the CSV values (serial number here). But it can be augmented with “:<other CSV variable>” (“:smltVirtBmac” here)
- When the CSV is parsed it will now not only return all values for the lookup serial number (2128Q-40044) but will also inspect the values of smltVirtBmac and if another entry in the CSV has the same value for smltVirtBmac then all CSV variables for that other entry will also become available in the config template as $<peer:var>. Hence $<peer:sysID> will resolve to 02bb.7400.0300
- The expectation is that only 2 entries in the CSV file will have the same value for the selected smltVirtBmac column; if more than 2 are found CSV parsing will halt with an error



```
#block start
no router isis enable\ny
router isis
   system-id $<sysID>
   spbm 1 smlt-peer-system-id $<peer:sysID>
   spbm 1 smlt-virtual-bmac $<smltVirtBmac>
exit
vlan create 4000 name "IST-VLAN" type port-mstprstp 0
vlan i-sid 4000 $<ist isid>
interface Vlan 4000
#if(int($<sysID>.replace(".", ""), 16) < int($<peer:sysID>.replace(".", ""), 16))
   ip address 192.168.255.1 255.255.255.252
   exit
   virtual-ist peer-ip 192.168.255.2 vlan 4000
#else
   ip address 192.168.255.2 255.255.255.252
   exit
   virtual-ist peer-ip 192.168.255.1 vlan 4000
#end
router isis enable
#block execute 60
```

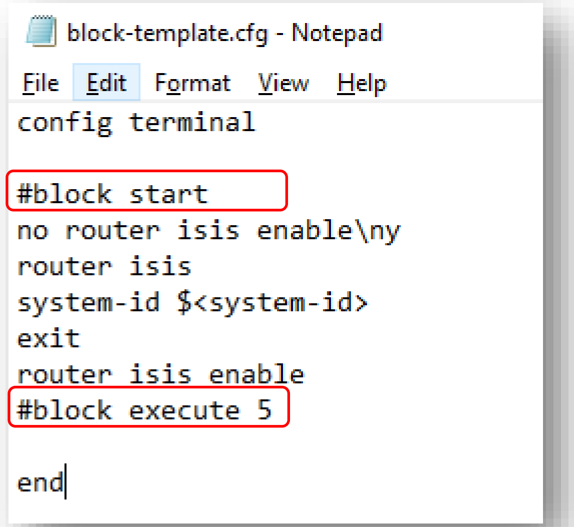


	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	serial number:smltVirtBmac	mgmt vlanid	mgmt isid	mgmt ip	mgmt clip	mgmt mask	mgmt gateway	sysname	site name	comments	next workflow	sysID	smltVirtBmac	ist isid
2	2128Q-40044	129	2102901	10.129.0.72		255.255.255.0	10.129.0.1	Sbox-VSP7400-2		Kevin	Provisioning/Apply Config Template	02bb.7400.0200	02:bb:74:00:02:ff	15000001
3	2128Q-40009	129	2102901	10.129.0.73		255.255.255.0	10.129.0.1	Sbox-VSP7400-3		Stuart	Provisioning/Apply Config Template	02bb.7400.0300	02:bb:74:00:02:ff	15000001
4	1916Q-20031	129	2102901	10.129.0.74		255.255.255.0	10.129.0.1	Sbox-VSP7400-4		Gru	Provisioning/Apply Config Template	02bb.7400.0400		

Commands which need to be sourced locally on the switch

- There may be situations where a bunch of commands need to be executed directly by the switch itself, like for configuration that would otherwise make the switch temporarily unreachable to XIQ-SE, e.g. shutting down ISIS, in order to change the fabric IDs
  - The commands are packed into a text file, which is TFTP downloaded to the switch and executed locally using the source command
  - On successful completion the file is removed from the switch, and the TFTP boot flag is disabled, if it was disabled to start with
- Of course this is risky; the commands need to be validated to ensure that XIQ-SE connectivity will be restored at the end of the block sequence
- Note this functionality will only work with VOSS/FabricEngine, EXOS/SwitchEngine and BOSS/ERS
- Note that this will not work with VOSS/FabricEngine "no ssh", as that command will kill the SSH session right away and nothing will get sourced locally
- Other velocity type statements (#if/#elseif/#else/#end/#error..) cannot be used inside the #block statements; but variables can be used
- Syntax:

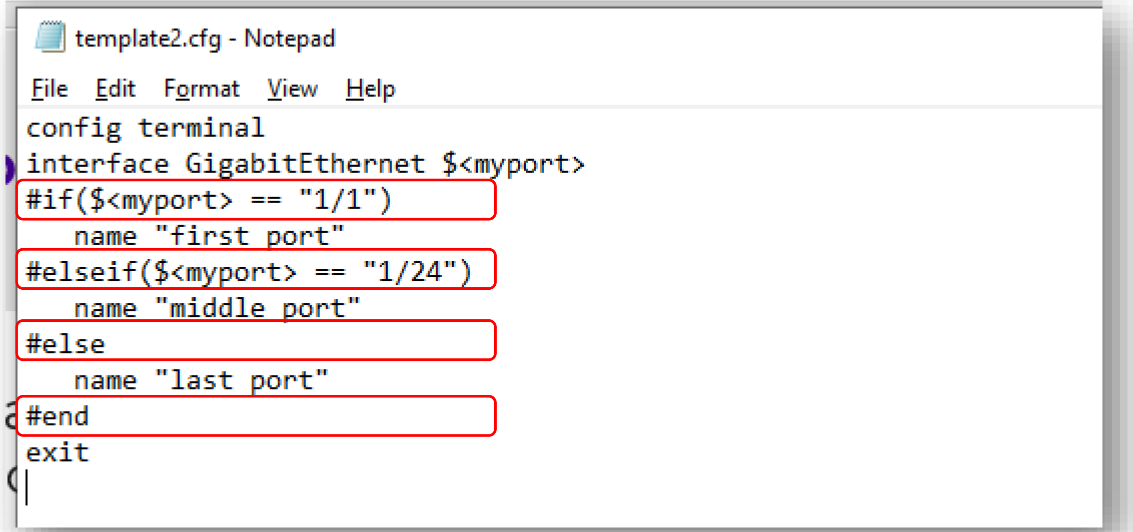
```
#block start [n]  : Mark the beginning of a block of commands which will need to
be sourced locally on the switch.

  [n] = Optional number of seconds to sleep after block execution

#block execute [n]: Mark the end of block of commands which are to sourced
locally on the switch. If this directive is not seen, and the "#block start" was
seen, all commands from the start of block section to the last command in the
template file will be sourced locally on the switch.

  [n] = Optional number of seconds to sleep after block execution
```



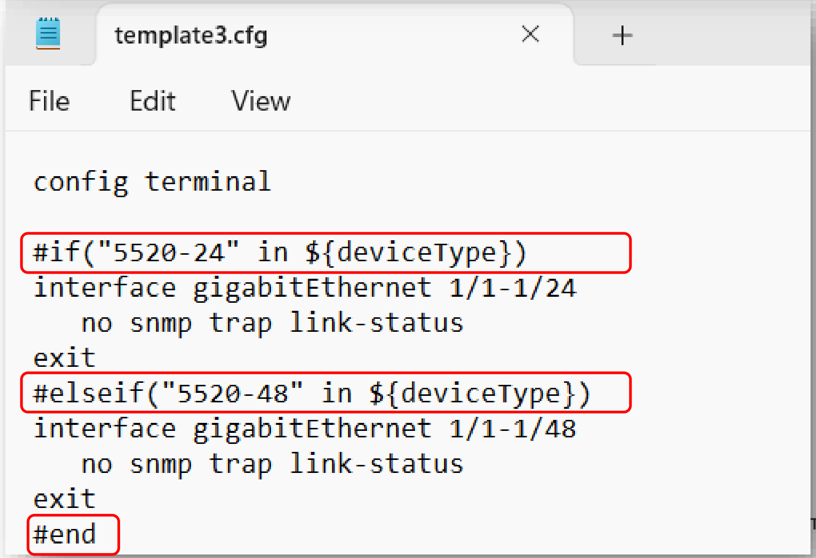
```
config terminal

#block start
no router isis enable\ny
router isis
system-id $<system-id>
exit
router isis enable
#block execute 5

end
```

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}, UserData variables $UD1-4 or template variables $[var] can be inserted inside the conditional string, but they will always be evaluated as String values.
- For instance, to evaluate an integer value, insert the variable inside Python’s int() method like this:
  - `#if ( int($<myvar>) > 10)`

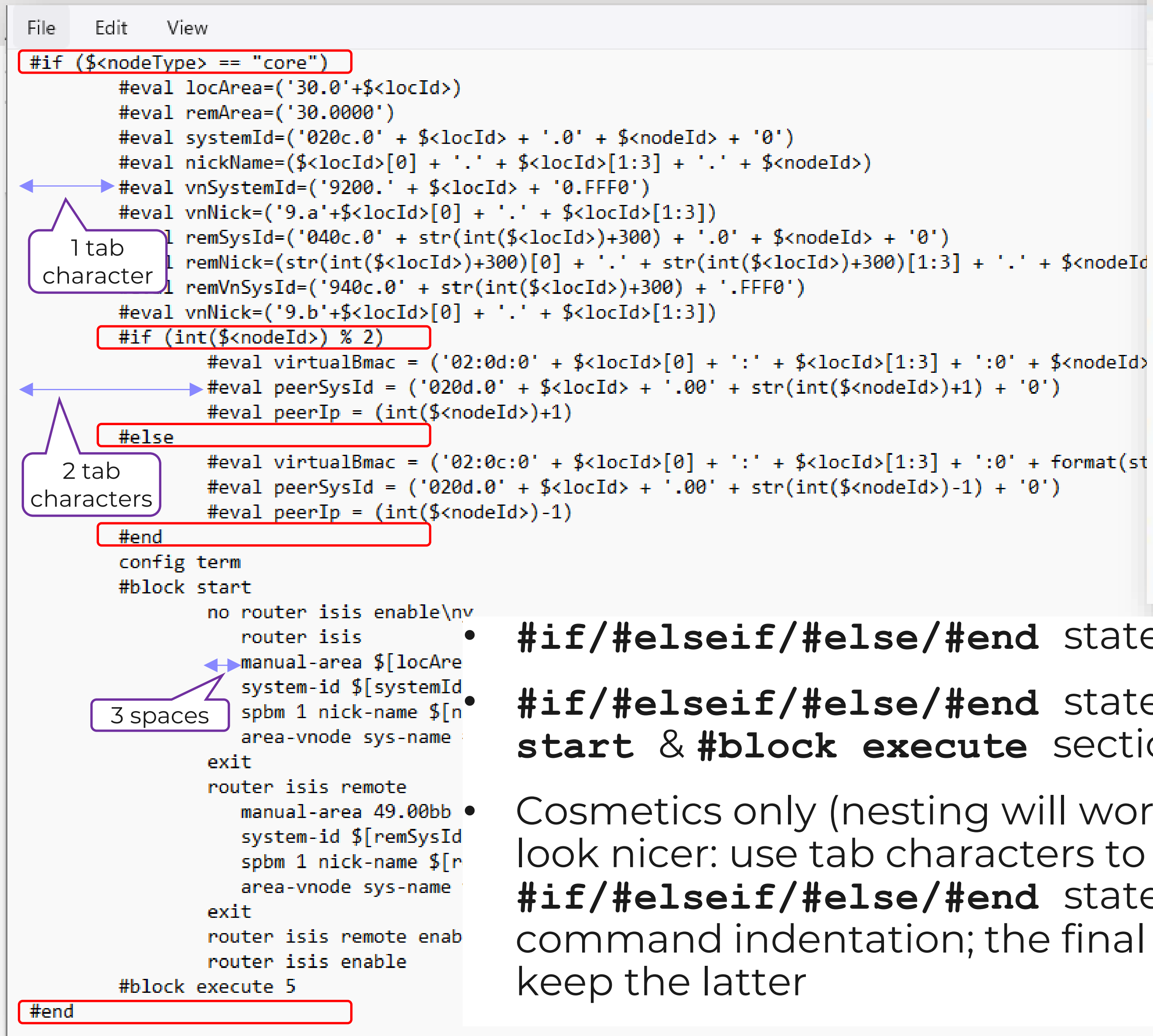


```
config terminal
interface GigabitEthernet $<myport>
#if($<myport> == "1/1")
   name "first port"
#elseif($<myport> == "1/24")
   name "middle port"
#else
   name "last port"
#end
exit
```

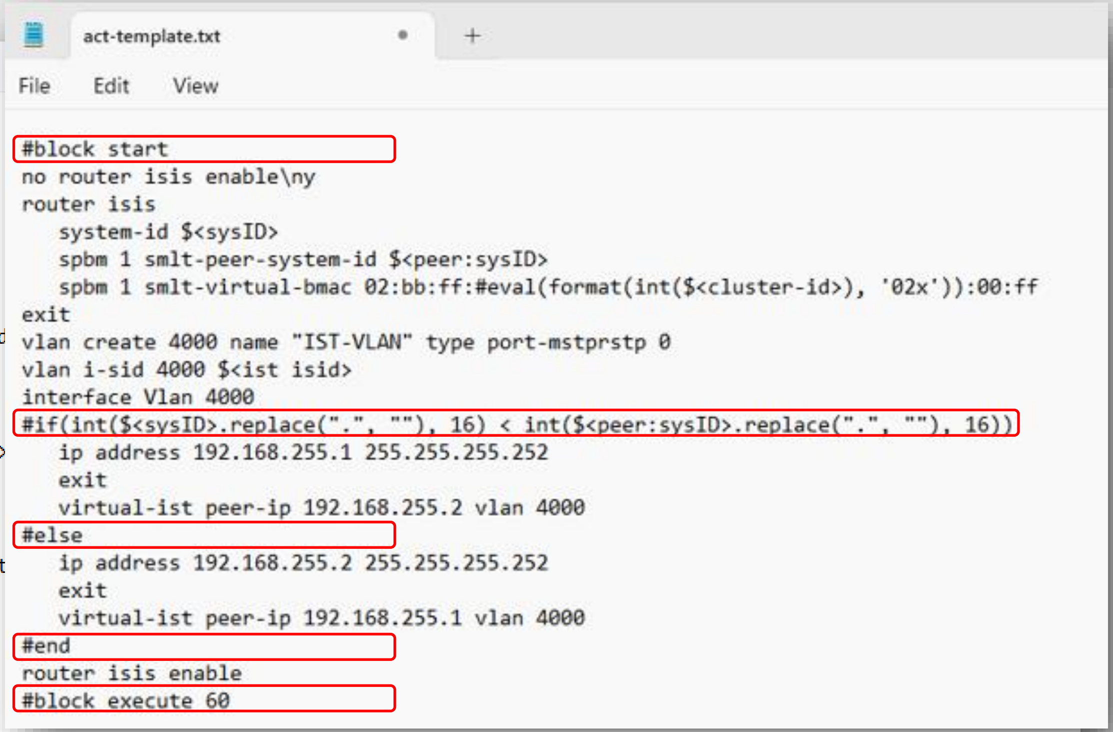


```
config terminal

#if("5520-24" in ${deviceType})
interface gigabitEthernet 1/1-1/24
   no snmp trap link-status
exit
#elseif("5520-48" in ${deviceType})
interface gigabitEthernet 1/1-1/48
   no snmp trap link-status
exit
#end
```

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



```
#if ($<nodeType> == "core")
        #eval locArea=('30.0'+$<locId>)
        #eval remArea=('30.0000')
        #eval systemId=('020c.0' + $<locId> + '.0' + $<nodeId> + '0')
        #eval nickName=($<locId>[0] + '.' + $<locId>[1:3] + '.' + $<nodeId>)
        #eval vnSystemId=('9200.' + $<locId> + '0.FFF0')
        #eval vnNick=('9.a'+$<locId>[0] + '.' + $<locId>[1:3])
            l remSysId=('040c.0' + str(int($<locId>)+300) + '.0' + $<nodeId> + '0')
            l remNick=(str(int($<locId>)+300)[0] + '.' + str(int($<locId>)+300)[1:3] + '.' + $<nodeId
            l remVnSysId=('940c.0' + str(int($<locId>)+300) + '.FFF0')
        #eval vnNick=('9.b'+$<locId>[0] + '.' + $<locId>[1:3])
        #if (int($<nodeId>) % 2)
                #eval virtualBmac = ('02:0d:0' + $<locId>[0] + ':' + $<locId>[1:3] + ':0' + $<nodeId>
                #eval peerSysId = ('020d.0' + $<locId> + '.00' + str(int($<nodeId>)+1) + '0')
                #eval peerIp = (int($<nodeId>)+1)
        #else
                #eval virtualBmac = ('02:0c:0' + $<locId>[0] + ':' + $<locId>[1:3] + ':0' + format(st
                #eval peerSysId = ('020d.0' + $<locId> + '.00' + str(int($<nodeId>)-1) + '0')
                #eval peerIp = (int($<nodeId>)-1)
        #end
        config term
        #block start
                no router isis enable\ny
                   router isis
                   manual-area $[locAre
                   system-id $[systemId
                   spbm 1 nick-name $[n
                   area-vnode sys-name
                exit
                router isis remote
                   manual-area 49.00bb
                   system-id $[remSysId
                   spbm 1 nick-name $[r
                   area-vnode sys-name
                exit
                router isis remote enab
                router isis enable
        #block execute 5
#end
```

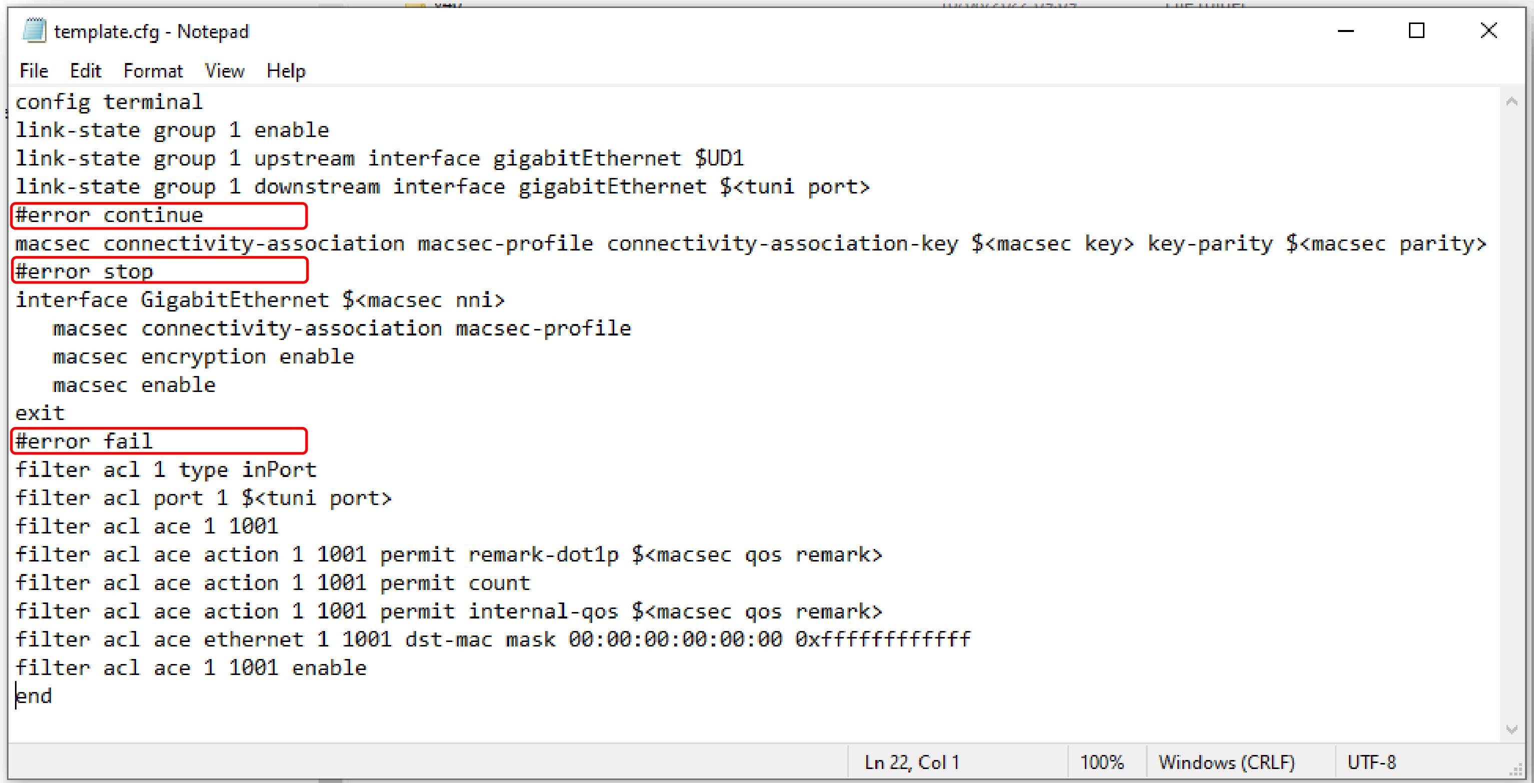


```
#block start
no router isis enable\ny
router isis
   system-id $<sysID>
   spbm 1 smlt-peer-system-id $<peer:sysID>
   spbm 1 smlt-virtual-bmac 02:bb:ff:#eval(format(int($<cluster-id>), '02x')):00:ff
exit
vlan create 4000 name "IST-VLAN" type port-mstprstp 0
vlan i-sid 4000 $<ist isid>
interface Vlan 4000
#if(int($<sysID>.replace(".", ""), 16) < int($<peer:sysID>.replace(".", ""), 16))
   ip address 192.168.255.1 255.255.255.252
   exit
   virtual-ist peer-ip 192.168.255.2 vlan 4000
#else
   ip address 192.168.255.2 255.255.255.252
   exit
   virtual-ist peer-ip 192.168.255.1 vlan 4000
#end
router isis enable
#block execute 60
```

- **`#if/#elseif/#else/#end`** statement blocks can be nested, without any limit
- **`#if/#elseif/#else/#end`** statement blocks can also be used inside **`#block start`** & **`#block execute`** sections, as long as they are fully contained inside
- Cosmetics only (nesting will work regardless), so that the config applied will look nicer: use tab characters to apply indentation to **`#if/#elseif/#else/#end`** statement blocks and use spaces for switch command indentation; the final config pushed will suppress the former and keep the latter

Error mode: #error fail|stop|continue

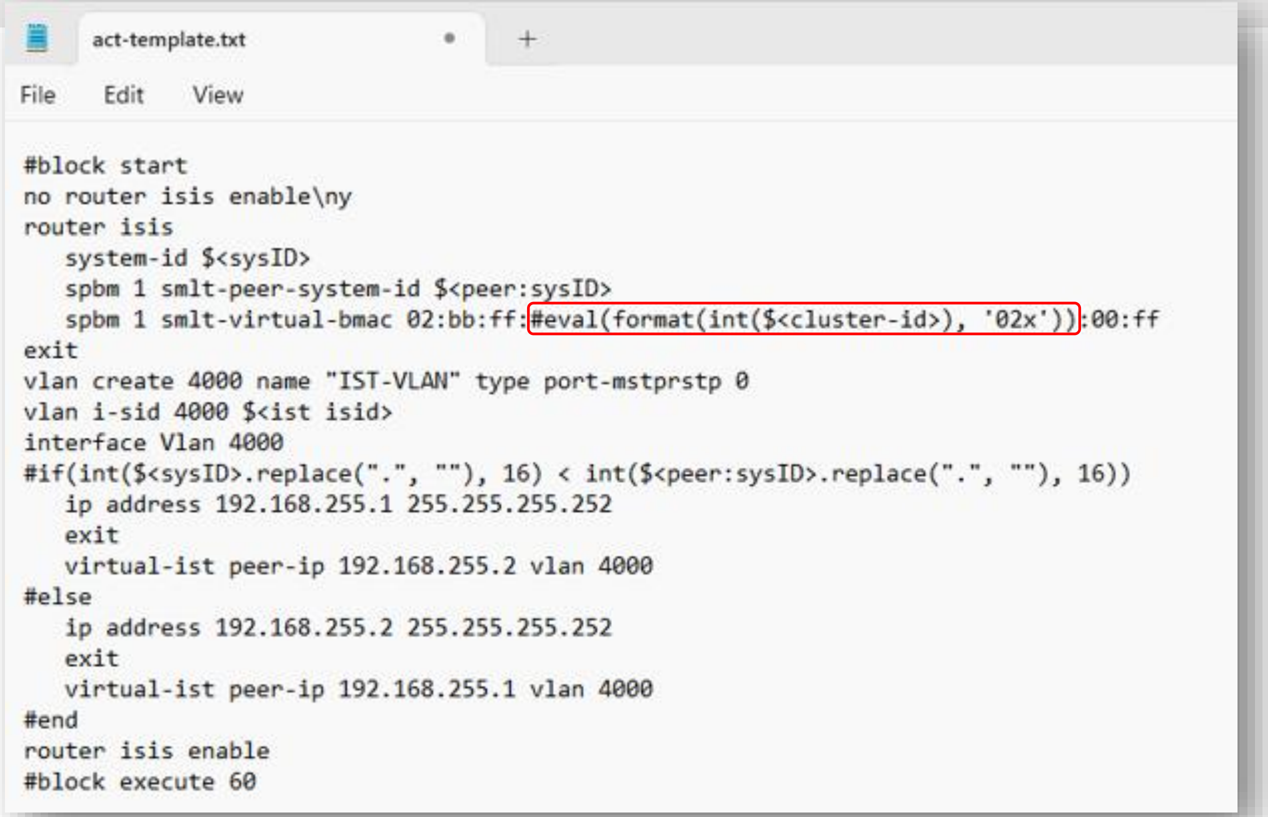
- The template file can also include **`#error fail|stop|continue`** statement
- Determines the behaviour if a command in the template errors when executed on the switch
  - **fail**: workflow aborts immediately with an error
  - **stop**: no further commands from the template are executed, the workflow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



```
config terminal
link-state group 1 enable
link-state group 1 upstream interface gigabitEthernet $UD1
link-state group 1 downstream interface gigabitEthernet $<tuni port>
#error continue
macsec connectivity-association macsec-profile connectivity-association-key $<macsec key> key-parity $<macsec parity>
#error stop
interface GigabitEthernet $<macsec nni>
   macsec connectivity-association macsec-profile
   macsec encryption enable
   macsec enable
exit
#error fail
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 qos remark>
filter acl ace action 1 1001 permit count
filter acl ace action 1 1001 permit internal-qos $<macsec qos remark>
filter acl ace ethernet 1 1001 dst-mac mask 00:00:00:00:00:00 0xffffffffffff
filter acl ace 1 1001 enable
end
```

Embedded eval: #eval

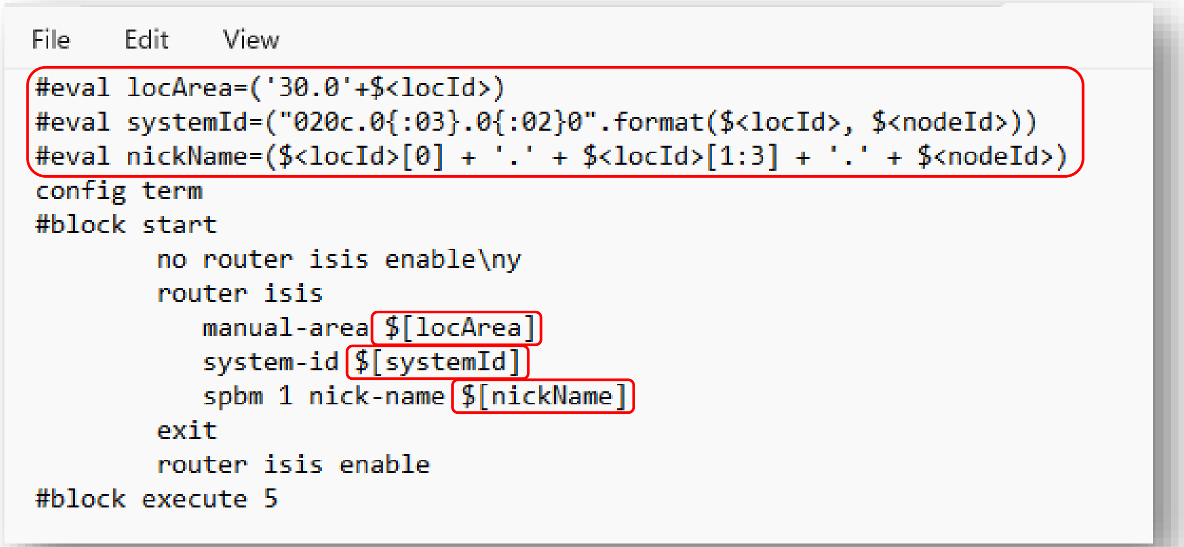
- Any config line can contain an eval statement:
  `#eval ()`
- The string, inside "(" ")" will be evaluated using Python's eval() function and the result converted to string (str)
- In the example the value of $<cluster-id> is converted to a 2-digit hex number



```
#block start
no router isis enable\ny
router isis
   system-id $<sysID>
   spbm 1 smlt-peer-system-id $<peer:sysID>
   spbm 1 smlt-virtual-bmac 02:bb:ff:#eval(format(int($<cluster-id>), '02x')):00:ff
exit
vlan create 4000 name "IST-VLAN" type port-mstprstp 0
vlan i-sid 4000 $<ist isid>
interface Vlan 4000
#if(int($<sysID>.replace(".", ""), 16) < int($<peer:sysID>.replace(".", ""), 16))
   ip address 192.168.255.1 255.255.255.252
   exit
   virtual-ist peer-ip 192.168.255.2 vlan 4000
#else
   ip address 192.168.255.2 255.255.255.252
   exit
   virtual-ist peer-ip 192.168.255.1 vlan 4000
#end
router isis enable
#block execute 60
```

Template eval variables: #eval <varname>=()

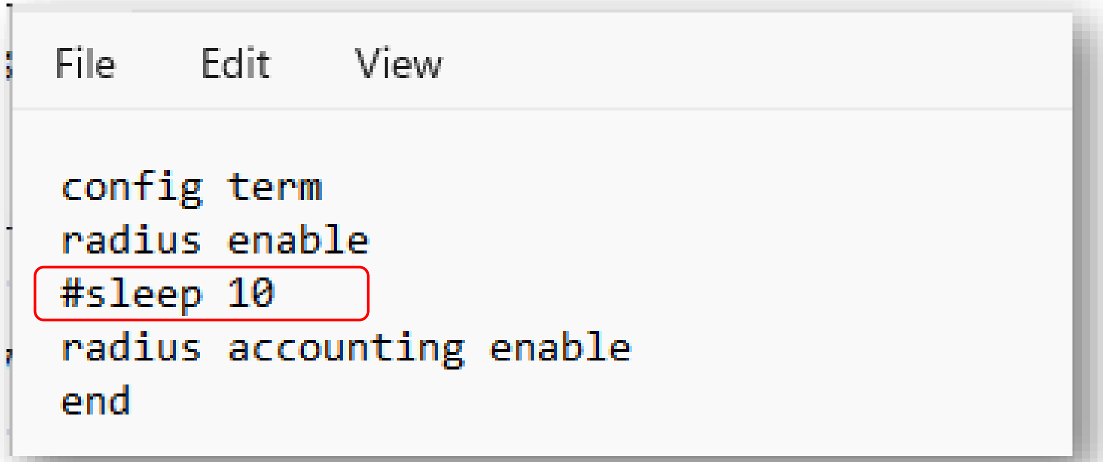
- Uses an entire line in the template file: **`#eval <varname>=()`**
- This syntax, unlike embedded eval from previous slide, cannot be embedded in commands to be sent to the switch
- The string, inside "(" ")" will be evaluated using Python's eval() function and the result converted to string (str)
- In the example, fabric ids are evaluated as template variables, then used in commands further down in the template



```
#eval locArea=('30.0'+$<locId>)
#eval systemId=("020c.0{:03}.0{:02}0".format($<locId>, $<nodeId>))
#eval nickName=($<locId>[0] + '.' + $<locId>[1:3] + '.' + $<nodeId>)
config term
#block start
        no router isis enable\ny
        router isis
           manual-area $[locArea]
           system-id $[systemId]
           spbm 1 nick-name $[nickName]
        exit
        router isis enable
#block execute 5
```

Sleep statement: #sleep

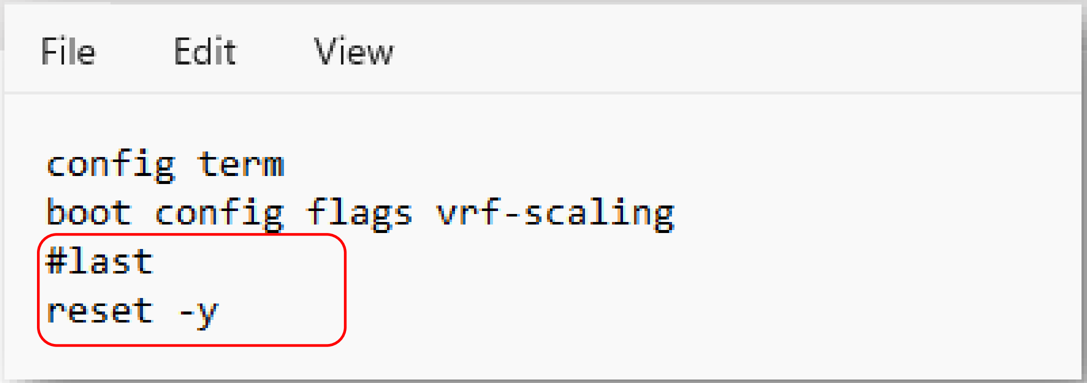
- Introduce a delay while sourcing the template file to the switch:
  `#sleep <seconds>`
- Some CLI commands will error if executed too rapidly after the previous command; e.g. on VOSS you have to wait several seconds before enabling RADIUS accounting after having enabled RADIUS…
- The **`#sleep`** statement cannot be used inside a **`#block start`** & **`#block execute`** section
- The <seconds> value can also be supplied by a CSV variable, site variable, template variable, etc..



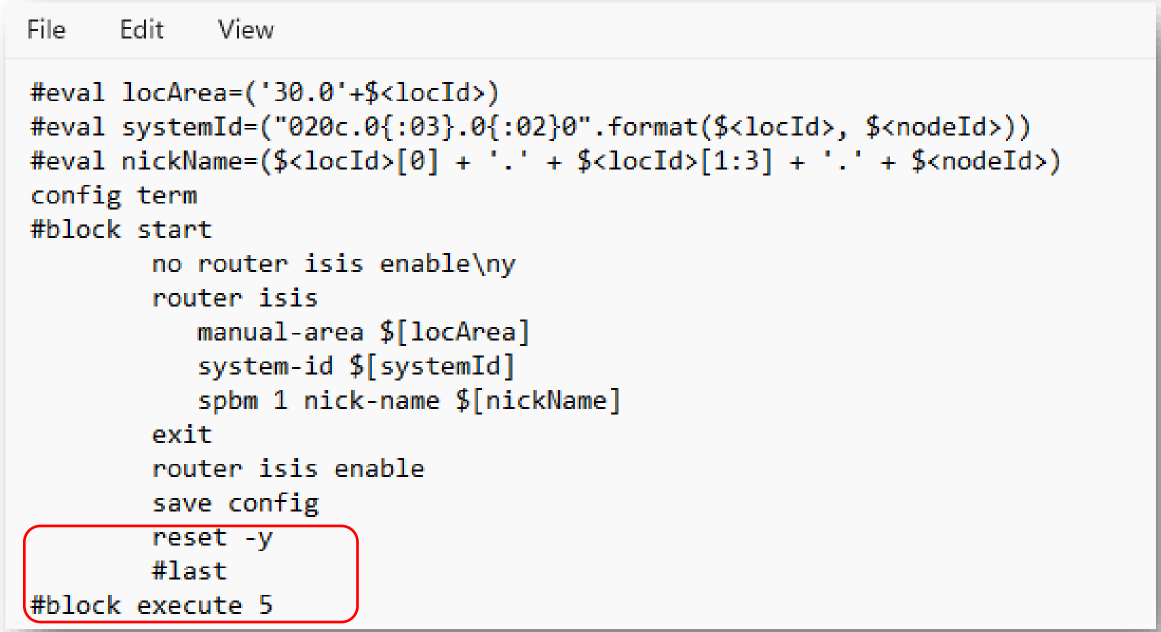
```
config term
radius enable
#sleep 10
radius accounting enable
end
```

Last statement: #last

- Do not wait for a prompt to come back on the very last command in the template: **`#last`**
- For example, if the template is to reboot the switch once done
  - Otherwise, if the last command in the template does not receive a prompt, the workflow will timeout and report a failure
- The **`#last`** statement must always appear on the penultimate non-empty line of the template file
  - If it is placed on any other line it will simply be ignored
- Will also work with **`#block start`** & **`#block execute`** sections, but must again be placed in line immediately preceding **`#block execute`** or preceding last line of block section if trailing **`#block execute`** is omitted



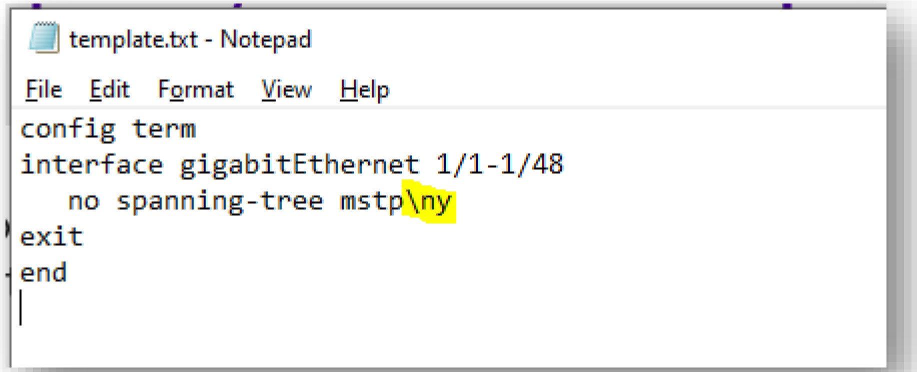
```
config term
boot config flags vrf-scaling
#last
reset -y
```



```
#eval locArea=('30.0'+$<locId>)
#eval systemId=("020c.0{:03}.0{:02}0".format($<locId>, $<nodeId>))
#eval nickName=($<locId>[0] + '.' + $<locId>[1:3] + '.' + $<nodeId>)
config term
#block start
        no router isis enable\ny
        router isis
           manual-area $[locArea]
           system-id $[systemId]
           spbm 1 nick-name $[nickName]
        exit
        router isis enable
        save config
        reset -y
        #last
#block execute 5
```

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
- On VOSS some commands (like “reset”) offer a “-y” switch which can bypass the y/n confirmation prompt; use that if available, else use “\ny”

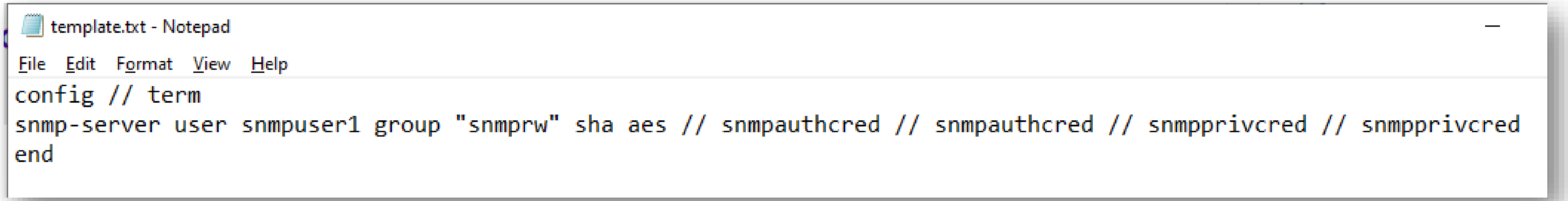


```
config term
interface gigabitEthernet 1/1-1/48
   no spanning-tree mstp\ny
exit
end
```

Commands which prompt for interactive input on device

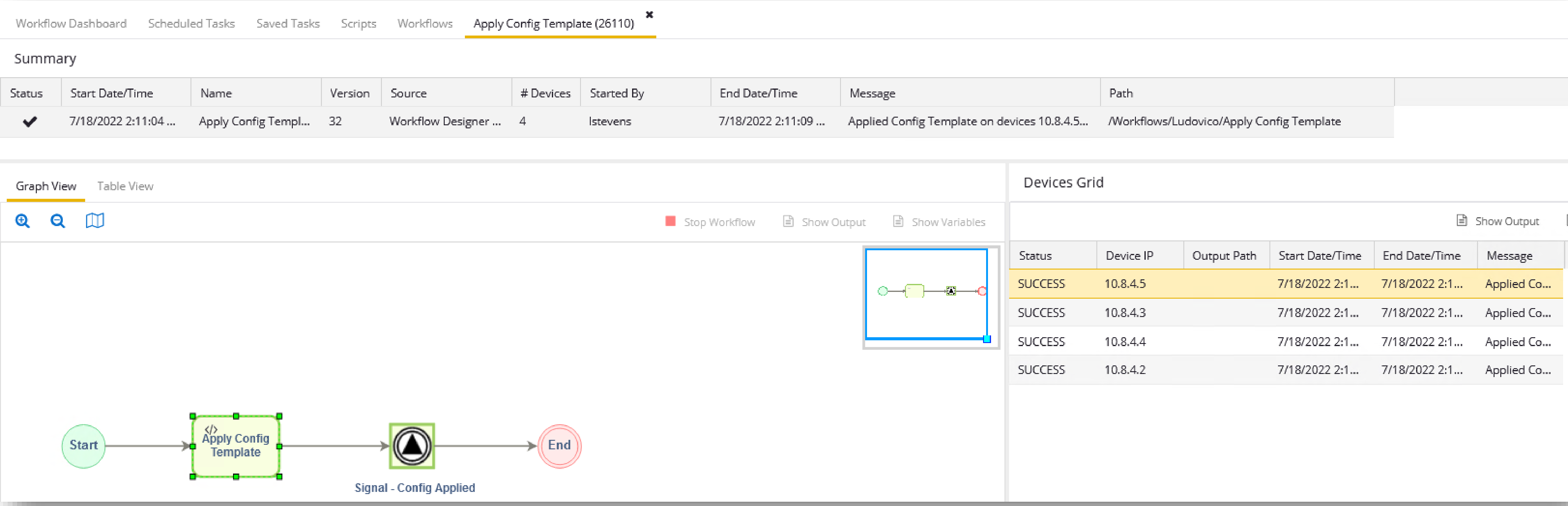
- Some commands which set passwords, prompt the user interactively to enter such passwords, sometimes twice
- To push such commands via the template, append the data (passwords) in the same sequence as they would be requested, separated by "//"
- The following example shows how this can be done for "config" 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



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

Workflow execution



Workflow Dashboard | Scheduled Tasks | Saved Tasks | Scripts | Workflows | Apply Config Template (26110)

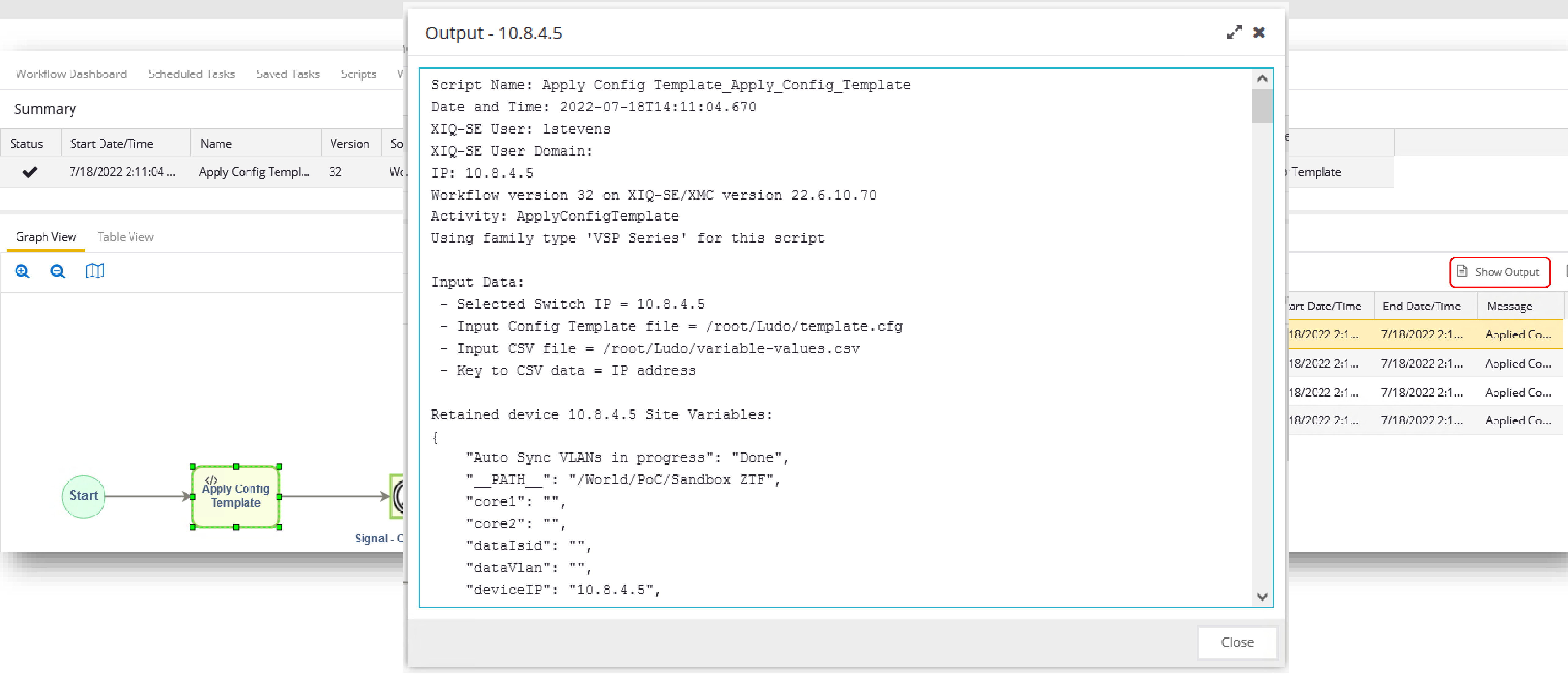
Summary

Status	Start Date/Time	Name	Version	Source	# Devices	Started By	End Date/Time	Message	Path
✓	7/18/2022 2:11:04 ...	Apply Config Templ...	32	Workflow Designer ...	4	lstevens	7/18/2022 2:11:09 ...	Applied Config Template on devices 10.8.4.5...	/Workflows/Ludovico/Apply Config Template

Devices Grid

Status	Device IP	Output Path	Start Date/Time	End Date/Time	Message
SUCCESS	10.8.4.5		7/18/2022 2:1...	7/18/2022 2:1...	Applied Co...
SUCCESS	10.8.4.3		7/18/2022 2:1...	7/18/2022 2:1...	Applied Co...
SUCCESS	10.8.4.4		7/18/2022 2:1...	7/18/2022 2:1...	Applied Co...
SUCCESS	10.8.4.2		7/18/2022 2:1...	7/18/2022 2:1...	Applied Co...

Workflow execution



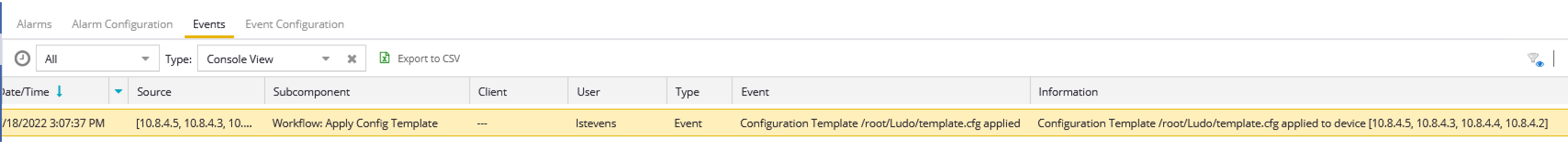
Status	Start Date/Time	Name	Version	So
✓	7/18/2022 2:11:04 ...	Apply Config Templ...	32	Wo

art Date/Time	End Date/Time	Message
18/2022 2:1...	7/18/2022 2:1...	Applied Co...
18/2022 2:1...	7/18/2022 2:1...	Applied Co...
18/2022 2:1...	7/18/2022 2:1...	Applied Co...
18/2022 2:1...	7/18/2022 2:1...	Applied Co...

Output - 10.8.4.5

```
Script Name: Apply Config Template_Apply_Config_Template
Date and Time: 2022-07-18T14:11:04.670
XIQ-SE User: lstevens
XIQ-SE User Domain:
IP: 10.8.4.5
Workflow version 32 on XIQ-SE/XMC version 22.6.10.70
Activity: ApplyConfigTemplate
Using family type 'VSP Series' for this script

Input Data:
 - Selected Switch IP = 10.8.4.5
 - Input Config Template file = /root/Ludo/template.cfg
 - Input CSV file = /root/Ludo/variable-values.csv
 - Key to CSV data = IP address

Retained device 10.8.4.5 Site Variables:
{
    "Auto Sync VLANs in progress": "Done",
    "__PATH__": "/World/PoC/Sandbox ZTF",
    "core1": "",
    "core2": "",
    "dataIsid": "",
    "dataVlan": "",
    "deviceIP": "10.8.4.5",
```

Workflow Event signal



Alarms | Alarm Configuration | Events | Event Configuration

All | Type: Console View | Export to CSV

)ate/Time	Source	Subcomponent	Client	User	Type	Event	Information
/18/2022 3:07:37 PM	[10.8.4.5, 10.8.4.3, 10....	Workflow: Apply Config Template	---	lstevens	Event	Configuration Template /root/Ludo/template.cfg applied	Configuration Template /root/Ludo/template.cfg applied to device [10.8.4.5, 10.8.4.3, 10.8.4.4, 10.8.4.2]

ADVANCE WITH US™