© 2018 SolarWinds Worldwide, LLC. All rights reserved.

This document may not be reproduced by any means nor modified, decompiled, disassembled, published or distributed, in whole or in part, or translated to any electronic medium or other means without the prior written consent of SolarWinds. All right, title, and interest in and to the software, services, and documentation are and shall remain the exclusive property of SolarWinds, its affiliates, and/or its respective licensors.

SOLARWINDS DISCLAIMS ALL WARRANTIES, CONDITIONS, OR OTHER TERMS, EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, ON THE DOCUMENTATION, INCLUDING WITHOUT LIMITATION NONINFRINGEMENT, ACCURACY, COMPLETENESS, OR USEFULNESS OF ANY INFORMATION CONTAINED HEREIN. IN NO EVENT SHALL SOLARWINDS, ITS SUPPLIERS, NOR ITS LICENSORS BE LIABLE FOR ANY DAMAGES, WHETHER ARISING IN TORT, CONTRACT OR ANY OTHER LEGAL THEORY, EVEN IF SOLARWINDS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The SolarWinds, SolarWinds & Design, Orion, and THWACK trademarks are the exclusive property of SolarWinds Worldwide, LLC or its affiliates, are registered with the U.S. Patent and Trademark Office, and may be registered or pending registration in other countries. All other SolarWinds trademarks, service marks, and logos may be common law marks or are registered or pending registration. All other trademarks mentioned herein are used for identification purposes only and are trademarks of (and may be registered trademarks) of their respective companies.
# Table of Contents

**Introduction to SolarWinds NCM** .................................................................................................................. 1  
Scheduled configuration backups ......................................................................................................................... 1 
Policy management ............................................................................................................................................. 1 
Inventory ............................................................................................................................................................ 1 
Configuration change history ............................................................................................................................. 1 
Configuration viewing, tracking, and comparison .............................................................................................. 2 
Device configuration change management .......................................................................................................... 2 
Real-time change detection .................................................................................................................................. 2 
Firmware upgrades ............................................................................................................................................... 2 
**Common tasks** ................................................................................................................................................. 3  
Schedule daily backups of config files .................................................................................................................. 3 
Change the community string on multiple nodes ............................................................................................... 3 
   Verify success by comparing configs ............................................................................................................... 4 
Block a MAC address on a wireless access point ............................................................................................... 4 
   Back up the running configuration .................................................................................................................... 4 
   Block a MAC address on a wireless access point ........................................................................................... 5 
   Verify success by comparing configs ............................................................................................................... 5 
Change an interface description .......................................................................................................................... 6 
   Back up the running configuration .................................................................................................................... 6 
   Update the interface descriptions ...................................................................................................................... 6 
   Verify success by checking the interface description ..................................................................................... 7 
Upgrade IOS and firmware ................................................................................................................................... 7 
Block all private addresses with an access control list .................................................................................... 7 
   Back up the running configuration .................................................................................................................... 7 
   Update the access control list for a group of nodes ......................................................................................... 7 
   Verify success by comparing configs ............................................................................................................... 8 
Customize the login banner of a device .................................................................................................................. 8 
   Modify the login banner .................................................................................................................................... 9
Verify success by comparing configs ................................................................. 9
Create a CRON expression ............................................................................... 10
Example ........................................................................................................ 10

Get started with NCM .................................................................................... 11
SNMP communication ..................................................................................... 11
Add nodes to NCM .......................................................................................... 11
  Add an individual node ................................................................................ 11
  Add nodes via Network Discovery .............................................................. 12
Select network devices for monitoring in NCM ................................................. 14
  Select the results of a network discovery .................................................. 14
Communication and logging options ............................................................... 15
Set node communication defaults .................................................................. 15
  Set node communication default parameters ........................................... 15
  Override default settings for a specific node ............................................ 15
Define and apply global device logins ............................................................. 15
Set communication limits ............................................................................. 16
Log events for a detailed record of SolarWinds NCM functions ....................... 16
  Enable verbose log information .............................................................. 16
  Enable session tracing .............................................................................. 16
  Enable SSL communication with SolarWinds NCM .................................. 17

Node management ......................................................................................... 18
Manage SolarWinds NCM nodes .................................................................... 18
  Find a specific node .................................................................................. 18
  Add or edit a node ................................................................................... 18
View node interface details ............................................................................ 18
  View the interface details of a specific node ............................................. 19
Manage end of support and end of sales (EOS) ................................................ 19
  Open the EOS lookup ............................................................................... 19
  Refresh EOS dates .................................................................................. 19
  Assign EOS dates ................................................................................... 20
Ignore devices in EOS Management ................................................................. 20
Export EOS information .................................................................................. 20
Filter EOS data .................................................................................................. 21
Delete EOS data .................................................................................................. 21
SolarWinds NCM macros ...................................................................................... 21
Supported macro types ....................................................................................... 22
Custom macros .................................................................................................... 22
Access custom macros ......................................................................................... 22
Create a custom macro ........................................................................................ 22
Edit a custom macro ............................................................................................ 23
Delete a custom macro ........................................................................................ 23
Find connected port for a host: SolarWinds NPM integration .......................... 23
Search for connected ports .................................................................................. 23
Unmanage and remanage nodes ........................................................................... 24
Access the management page .............................................................................. 25
Unmanage nodes .................................................................................................. 25
Remanage nodes ................................................................................................. 25

**Firmware vulnerability data** .............................................................................. 26
View firmware vulnerability details and update the remediation status ............. 26
View firmware vulnerability reports .................................................................... 27
Troubleshoot firmware vulnerability reports ...................................................... 28
Access firmware vulnerability settings ............................................................... 28
Manually download and add firmware vulnerability files .................................... 28

**Firmware upgrades** .......................................................................................... 30
Set up the storage location and firmware repository .......................................... 30
Integrity of firmware images ................................................................................ 31
Manage firmware upgrade templates ................................................................. 31
NCM default firmware upgrade templates ........................................................ 31
Add a firmware upgrade template ......................................................................... 32
Share firmware upgrade templates between Orion servers .................................. 33
Export a firmware upgrade template ................................................................. 33
Import a firmware upgrade template .............................................................. 33
Perform a firmware upgrade operation ............................................................ 34
Add a firmware upgrade operation ................................................................. 34
Upgrade Options .............................................................................................. 35
Determine image paths .................................................................................... 35
Config comparison .......................................................................................... 35
Operation statuses for firmware upgrades ...................................................... 35
Additional actions for firmware upgrade operations ....................................... 36
Cancel a firmware upgrade operation ............................................................. 36
Rollback a firmware upgrade operation ......................................................... 36
Re-execute a firmware upgrade operation ..................................................... 36
Review a firmware upgrade operation ............................................................ 36

Configuration files ......................................................................................... 37
Setup required for binary configs ................................................................. 37
Download configuration files in NCM ........................................................... 38
  Downloading from a single node ................................................................. 38
  Downloading from multiple nodes .............................................................. 38
Enable a new config type ................................................................................ 39
  Enable a new config type for a device ......................................................... 39
Enable the config and policy caches .............................................................. 39
Edit config files ............................................................................................... 40
  Upload the edited config file ........................................................................ 40
Troubleshooting: Confine recognition of the command prompt during download ......................................................................................................................... 40
Compare configuration files ............................................................................ 41
Exclude config content from comparison ....................................................... 41
  Access the Comparison Criteria page ......................................................... 42
Create a new regular expression ..................................................................... 42
Edit a regular expression ................................................................................ 43
Enable or disable a regular expression ............................................................ 43
Delete a regular expression ........................................................................................................ 43
Regular expression examples ........................................................................................................ 43
Execute a script on a node ........................................................................................................... 43
  View the results .......................................................................................................................... 44
Upload a config .......................................................................................................................... 44
  Upload a config to one or more nodes ...................................................................................... 45
Import config files ...................................................................................................................... 45
  Import a config file .................................................................................................................... 45
Establish baselines as a comparison point for config changes .................................................. 45
  What is a baseline? ...................................................................................................................... 45
  Set a baseline ............................................................................................................................. 46
  Remove a config file as a baseline ............................................................................................ 46
  Baseline your entire network ..................................................................................................... 46
  Clear all baselines to reset the comparison point for config changes ...................................... 47
Upload configuration changes .................................................................................................... 47
  Upload an entire configuration ................................................................................................ 47
Create a config archive for local or network storage ................................................................. 48
  Create a config archive for local storage .................................................................................. 48
  Create a config archive for network storage ............................................................................ 48
Real time change detection ......................................................................................................... 49
  Real time change detection requirements ............................................................................... 49
  Configure real time change detection ...................................................................................... 49
    Access real-time change detection settings .......................................................................... 50
    Step 1: Manually configure your devices to send syslog or trap messages .......................... 50
    Step 2: Configure alerts and filters triggered by syslog and trap messages ......................... 51
    Step 3: Config changes .......................................................................................................... 54
    Step 4: Config downloads and notifications ........................................................................... 54
    Step 5: Enter NCM SMTP server details ............................................................................... 54
    Step 6: Enable real time config change notifications ............................................................ 54
Limit real time notification download operations ....................................................................... 55
Run syslog and trap services as administrator .................................................................55
Search config files or node properties ............................................................................55
Delete config files from the database ............................................................................56
Automate config file purges .........................................................................................56

Device templates ...........................................................................................................58

Gather the information you need ................................................................................58
Best practices for device command templates ..............................................................60
Communication process diagrams ..............................................................................62
Create device templates in SolarWinds NCM ..................................................................63
Create a device template using the interactive wizard ..................................................64
Create a device template with the XML editor ..............................................................64

Edit a device template ........................................................................................................64
Edit a device template with the interactive wizard .....................................................64
Edit a device template with the XML editor .............................................................65
Confirm the device template is working ........................................................................66

Sample CLI device command templates .......................................................................66
Cisco IOS example ..........................................................................................................66
Nortel BayStack 380 example .......................................................................................67

SCP example of config upload and download .................................................................67
Create a menu-based command template ....................................................................68

Command template commands ...................................................................................71
Pre-command and command template variables ............................................................73
Pre-command device template entry example ...............................................................74
Preclude pseudoterminal setup with command template variables ...............................75
Declare a special command prompt with command template variables .......................75
Switch user context with command template variables ................................................75
Respond to post-login interaction requests with command template variables ............75

Troubleshoot device connections ..................................................................................76
F5 device template behavior .........................................................................................76
Device setup prerequisites .............................................................................................76
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Template details</td>
<td>76</td>
</tr>
<tr>
<td><strong>Command scripts</strong></td>
<td>78</td>
</tr>
<tr>
<td>Create an executable command script</td>
<td>78</td>
</tr>
<tr>
<td>Use variables in scripts</td>
<td>78</td>
</tr>
<tr>
<td>Troubleshoot script execution</td>
<td>78</td>
</tr>
<tr>
<td>Example variable script</td>
<td>79</td>
</tr>
<tr>
<td>Define script variables</td>
<td>79</td>
</tr>
<tr>
<td>Reference variables with variables</td>
<td>80</td>
</tr>
<tr>
<td><strong>Inventory</strong></td>
<td>81</td>
</tr>
<tr>
<td>Run an inventory scan</td>
<td>81</td>
</tr>
<tr>
<td>Run a complete inventory scan</td>
<td>81</td>
</tr>
<tr>
<td>Run an inventory scan on a single node</td>
<td>81</td>
</tr>
<tr>
<td>Schedule an inventory scan</td>
<td>81</td>
</tr>
<tr>
<td>Adjust inventory settings</td>
<td>82</td>
</tr>
<tr>
<td>View inventory status</td>
<td>82</td>
</tr>
<tr>
<td><strong>Ensure compliance to policy rules</strong></td>
<td>83</td>
</tr>
<tr>
<td>How rules, policies, and policy reports work together</td>
<td>83</td>
</tr>
<tr>
<td>Common uses for policy reports</td>
<td>83</td>
</tr>
<tr>
<td>Regulatory compliance</td>
<td>83</td>
</tr>
<tr>
<td>Compliance with internal standards and policies</td>
<td>84</td>
</tr>
<tr>
<td>Security</td>
<td>84</td>
</tr>
<tr>
<td>Tasks to create and run a policy report</td>
<td>84</td>
</tr>
<tr>
<td>Create and manage policy rules</td>
<td>84</td>
</tr>
<tr>
<td>Create a rule</td>
<td>85</td>
</tr>
<tr>
<td>Add and identify the rule</td>
<td>85</td>
</tr>
<tr>
<td>Define the search string using basic matching</td>
<td>85</td>
</tr>
<tr>
<td>Define search conditions using Advanced Config Search</td>
<td>86</td>
</tr>
<tr>
<td>(Optional) Define a remediation script</td>
<td>87</td>
</tr>
<tr>
<td>Test and save the rule</td>
<td>87</td>
</tr>
<tr>
<td>Examples of search string conditions</td>
<td>88</td>
</tr>
</tbody>
</table>
Edit a rule ........................................................................................................................................ 90
Delete a rule ..................................................................................................................................... 90
Customize alert level descriptions for rules ....................................................................................... 91
Learn more .......................................................................................................................................... 91
Regular expression pattern matching examples ..................................................................................... 91
Create and manage policies .................................................................................................................. 92
Create a policy ...................................................................................................................................... 92
Edit a policy .......................................................................................................................................... 93
Delete a policy ...................................................................................................................................... 93
Learn more .......................................................................................................................................... 93
Create and manage policy reports ........................................................................................................ 94
Create a policy report ........................................................................................................................... 94
Edit a policy report ............................................................................................................................... 94
Delete a policy report ........................................................................................................................... 94
Configure DISA STIG compliance reports ............................................................................................ 95
Create a DISA STIG dashboard ............................................................................................................. 95
Learn more .......................................................................................................................................... 95
Find and remediate policy violations .................................................................................................... 95
Update cached policy compliance information ....................................................................................... 96
  Automatically update cached policy compliance information ........................................................... 96
  Manually update cached policy compliance information ................................................................... 96
  Schedule a policy report job ................................................................................................................ 96
View a policy report ............................................................................................................................. 97
Verify that a policy violation was remediated ....................................................................................... 98
Import and export policy report definitions ........................................................................................... 98
Share policy reports between Orion servers ............................................................................................ 98
  Export a policy report definition as a file ............................................................................................ 98
  Import a policy report definition ......................................................................................................... 99
  Export a policy report definition to THWACK .................................................................................... 99
Jobs ..................................................................................................................................................... 100
<table>
<thead>
<tr>
<th>Action</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create or edit a job</td>
<td>100</td>
</tr>
<tr>
<td>Enable a job</td>
<td>101</td>
</tr>
<tr>
<td>Disable a job</td>
<td>101</td>
</tr>
<tr>
<td>Start a job</td>
<td>102</td>
</tr>
<tr>
<td>Stop a job</td>
<td>102</td>
</tr>
<tr>
<td>View job logs</td>
<td>102</td>
</tr>
<tr>
<td>Delete a job</td>
<td>102</td>
</tr>
</tbody>
</table>

**Approval system for device configuration changes** ....................................................... 103

- Set up config change approval ................. 103
- Turn on approval system                      104
  - Enable config change approval ............... 104
  - Disable config change approval .............. 104
- View, approve, and decline approval requests 104
- Create, edit, and delete NCM accounts ........ 105
  - Access account settings ...................... 105
  - Create a new account                      105
  - Edit an existing account                  105
  - Delete an account                         105
- Device access information                    106
  - Third Party Authentication ................. 106

**Access control list management** ................................................................. 107

- Manage ACLs on Cisco ASA and Nexus devices  107
  - Display the access control lists on a device 107
- Compare ACLs ........................................ 108
  - Compare ACLs on the same device............. 108
  - Compare ACLs on different devices ........... 108
- Display ACL rules .................................. 109
- Identify rules that have not been applied .... 109
- Investigate overlapping rules ................. 109
- Display information about objects or object groups 110
Overlapping ACL rules ........................................................................................................ 110
Fully shadowed rules ........................................................................................................ 111
Partially shadowed rules ............................................................................................... 111
Fully redundant rules .................................................................................................... 111
Partially redundant rules .............................................................................................. 112

Reports ............................................................................................................................... 113
Run a config change report once ..................................................................................... 113
Schedule a recurring config change report .................................................................... 113
Predefined NCM reports .................................................................................................. 114
  NCM Audit ....................................................................................................................... 114
  Brocade Inventory .......................................................................................................... 115
  Cisco Inventory ................................................................................................................ 115
  F5 Inventory .................................................................................................................... 116
  NCM Inventory ............................................................................................................... 117
  Node Details Report ....................................................................................................... 117
  Polling Status ................................................................................................................ 118
  Route Tables Inventory ................................................................................................. 119
  Security .......................................................................................................................... 119
  Windows Server Inventory ............................................................................................ 120

Servers for file transfers and email notifications .............................................................. 121
  Use the SolarWinds TFTP server .................................................................................... 121
  Use the SFTP & SCP server ......................................................................................... 121
  Use a third-party SCP server ........................................................................................ 122
  Configure email notifications ......................................................................................... 122
    SMTP server credentials used to send notifications for config change approvals, real time
    change detection, and running jobs ................................................................................ 122
    Email notification defaults used in new scheduled jobs ............................................... 122

Config change templates .................................................................................................. 124
  The basics of config change templates .......................................................................... 124
  Preparation and use of a template .................................................................................. 124
<table>
<thead>
<tr>
<th>Entity/Relationships</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCM.ARPTables entity relationships</td>
<td>145</td>
</tr>
<tr>
<td>NCM.BridgePorts</td>
<td>145</td>
</tr>
<tr>
<td>NCM.BridgePorts entity relationships</td>
<td>147</td>
</tr>
<tr>
<td>NCM.CatalystCards</td>
<td>147</td>
</tr>
<tr>
<td>NCM.CatalystCards entity relationships</td>
<td>149</td>
</tr>
<tr>
<td>NCM.CiscoCards</td>
<td>150</td>
</tr>
<tr>
<td>NCM.CiscoCards entity relationships</td>
<td>151</td>
</tr>
<tr>
<td>NCM.CiscoCdp</td>
<td>151</td>
</tr>
<tr>
<td>NCM.CiscoCdp entity relationships</td>
<td>152</td>
</tr>
<tr>
<td>NCM.CiscoChassis</td>
<td>153</td>
</tr>
<tr>
<td>NCM.CiscoChassis entity relationships</td>
<td>154</td>
</tr>
<tr>
<td>NCM.CiscoFlash</td>
<td>154</td>
</tr>
<tr>
<td>NCM.CiscoFlash entity relationships</td>
<td>158</td>
</tr>
<tr>
<td>NCM.CiscoFlashFiles</td>
<td>158</td>
</tr>
<tr>
<td>NCM.CiscoFlashFiles entity relationships</td>
<td>159</td>
</tr>
<tr>
<td>NCM.CiscoImageMIB</td>
<td>159</td>
</tr>
<tr>
<td>NCM.CiscoImageMIB entity relationships</td>
<td>159</td>
</tr>
<tr>
<td>NCM.CiscoMemoryPools</td>
<td>160</td>
</tr>
<tr>
<td>NCM.CiscoMemoryPools entity relationships</td>
<td>160</td>
</tr>
<tr>
<td>NCM.EntityLogical</td>
<td>160</td>
</tr>
<tr>
<td>NCM.EntityLogical entity relationships</td>
<td>161</td>
</tr>
<tr>
<td>NCM.EntityPhysical</td>
<td>162</td>
</tr>
<tr>
<td>NCM.EntityPhysical entity relationships</td>
<td>165</td>
</tr>
<tr>
<td>NCM. Interfaces</td>
<td>165</td>
</tr>
<tr>
<td>NCM. Interfaces entity relationships</td>
<td>169</td>
</tr>
<tr>
<td>NCM.IpAddresses</td>
<td>169</td>
</tr>
<tr>
<td>NCM.IpAddresses entity relationships</td>
<td>169</td>
</tr>
<tr>
<td>NCM.MacForwarding</td>
<td>170</td>
</tr>
<tr>
<td>NCM.MacForwarding entity relationships</td>
<td>170</td>
</tr>
<tr>
<td>NCM.Nodes</td>
<td>170</td>
</tr>
</tbody>
</table>
NCM.Nodes entity relationships ................................................................. 172
NCM.PortsTcp .................................................................................. 173
NCM.PortsTcp entity relationships ...................................................... 174
NCM.PortsUdp .................................................................................. 174
NCM.PortsUdp entity relationships ...................................................... 174
NCM.RouteTable ............................................................................... 174
NCM.RouteTable entity relationships ................................................ 178
NCM.VLANs .................................................................................... 178
NCM.VLANs entity relationships ........................................................ 179

EnergyWise devices ............................................................................ 180
What is EnergyWise? ........................................................................... 180
Manage and enable EnergyWise nodes .......................................... 180
Manage Power over Ethernet ports ................................................. 180

Integration with Engineer's Toolset .................................................. 182

References ...................................................................................... 183
Node and configuration archive variables ...................................... 183
Configuration archive variables ..................................................... 183
Introduction to SolarWinds NCM

SolarWinds Network Configuration Manager is a comprehensive, intuitive solution designed to streamline and automate network configuration management. NCM increases availability, saves time, improves security, and ensures policy adherence. NCM features automation capabilities that reduce the amount of time network engineers spend on mundane network tasks, allowing them to focus on business-critical network projects.

Use NCM to manage configurations on heterogeneous, multi-vendor networks. NCM provides a single point of management for routers, switches, firewalls, load balancers, and wireless access points. Supported vendors include Cisco, Cisco ASA, Cisco Nexus, Dell, Adtran, Arris, Aruba, Nortel, Nortel Alteon, Nortel BayStack, Extreme, Marconi, Radware, Netscreen, Motorola, HP, Netscalar, Juniper, Foundry, and F5.

Use NCM to automate and streamline the following tasks.

Scheduled configuration backups

Schedule configuration downloads, configuration uploads, device reboots, command script execution, and more. In addition, configuration backups are stored both in a relational database for archival history and as flat files in an intuitive folder structure for easy viewing.

Policy management

Use NCM policy reports to ensure device compliance with federal regulations and corporate standards. The Policy Reporting Manager includes policy reports such as SOX, HIPAA, CISP, and Cisco Security. You can also create custom reports.

Inventory

Use NCM to perform inventory scans on all of your nodes, node groups, or on a single node. The collected inventory statistics are displayed in the detail view of each device.

Configuration change history

Receive reports on what devices have had configuration changes over a specified time period. Configuration change reports can also compare current configurations with a baseline configuration, alerting you whenever a change is discovered.
Configuration viewing, tracking, and comparison

Use SolarWinds NCM to remotely view, track and make changes, and compare network device configurations without logging in to the physical SolarWinds NCM server. The Orion Web Console offers these functions to the users you select.

Device configuration change management

Set up a request and approval system for processing the workflow of device configuration changes.

Real-time change detection

The Real Time Change Detection feature provides notification through email whenever a change to any of your device configurations occurs.

Firmware upgrades

Upgrades the firmware on many devices at one time. Because security fixes are distributed as new firmware revisions, upgrading firmware helps network administrators correct security vulnerabilities.
Common tasks

The following topics present example scenarios to help demonstrate how you can use SolarWinds Network Configuration Manager in different network environments.

Schedule daily backups of config files

SolarWinds NCM installs an example job that downloads the configuration files nightly for all nodes in the database. You can modify the example for your specific needs, or you can create a new job.

1. Click My Dashboards > Configs > Jobs.
2. Click Create New Job.
3. Name the job, and select Download Configs from Devices from Job Type.
4. Select the Schedule Type. If you are creating an Advanced schedule, use the five fields to create a CRON expression.
5. Add a comment if this job relates to a business rule, and click Next.
6. Select the nodes to target with this job, and click Next.
7. Select an email notification option, and click Next. If you click Email Results, the default email notification and SMTP server settings are populated. These settings can be overridden in each job.
8. Select the configuration types you want to download, select the send config change notifications, and click Next.
9. Review the settings for the job, and click Finish.

Change the community string on multiple nodes

This procedure replaces the public read-only community string with a new read-only community string on several network nodes at the same time.

1. Click My Dashboards > Configs > Jobs.
2. Click Create New Job.
3. Name the job, and select Execute Command Script on Devices from Job Type.
4. Select the Schedule Type. If you are creating an Advanced schedule, use the five fields to create a CRON expression.
5. Add a comment if this job relates to a business rule, and click Next.
6. Select the nodes to target with this job, and click Next.
7. Select an email notification option, and click Next. If you click Email Results, the default email notification and SMTP server settings are populated. These settings can be overridden in each job.
8. Add the script to execute.
   - If you want to load a saved script, click Load Saved Script, and select a script.
   - If you want to load an existing script from disk, click Load Script from File, browse to the file, and click Open.
   - If you want to create a new script, enter your script in the text box. For example:
     ```
     config t
     no snmp-server community public RO
     snmp-server community 123@dm1n RO
     exit
     wr mem
     ```
     Where 123@dm1n is the new community string.

9. If you want to save a script, click Save Script, specify a location, and click Save.

10. Select Filter Results that match a pattern if you want to see in the script output only those lines that match a specific regular expression pattern.

11. Select Show Commands in Output to view what SolarWinds NCM sent to the targeted devices.

12. Click Next.

13. Review the settings for the job, and click Finish.

**Verify success by comparing configs**

1. Click My Dashboards > Configs > Configuration Management.
2. Select one or more nodes, and click Compare Node(s) Configs.
3. Select a config from each list.
4. Click Compare Selected. Changes to are highlighted in red, green, and yellow.

**Block a MAC address on a wireless access point**

If you discover a device utilizing unauthorized access through your wireless network, you can block the MAC address to prevent future access. The following procedure uses an access control list on a wireless access point to block a specific MAC address.

**Back up the running configuration**

1. Click My Dashboards > Configs > Configuration Management.
2. Select one or more nodes.
3. Click Download > Running.
Block a MAC address on a wireless access point

1. Click My Dashboards > Configs > Jobs.
2. Click Create New Job.
3. Name the job, and select Execute Command Script on Devices from Job Type.
4. Select the Schedule Type. If you are creating an Advanced schedule, use the five fields to create a CRON expression.
5. Add a comment if this job relates to a business rule, and click Next.
6. Select the nodes to target with this job, and click Next.
7. Select an email notification option, and click Next. If you click Email Results, the default email notification and SMTP server settings are populated. These settings can be overridden in each job.
8. Add the script to execute.
   - If you want to load a saved script, click Load Saved Script, and select a script.
   - If you want to load an existing script from disk, click Load Script from File, browse to the file, and click Open.
   - If you want to create a new script, enter your script in the text box. For example:

     ```plaintext
     ${EnterConfigMode}
     access-list 724 deny 000e.0ca1.a2b4 0000.0000.0000
     exit
     wr mem
     ```

     Where 724 is the name of the ACL, and 000E.0CA1.A2B4 is the MAC address to block. ${EnterConfigMode} is a variable that is equivalent to config terminal on Cisco devices.

9. If you want to save a script, click Save Script, specify a location, and click Save.
10. Select Filter Results that match a pattern if you want to see in the script output only those lines that match a specific regular expression pattern.
11. Select Show Commands in Output to view what SolarWinds NCM sent to the targeted devices.
12. Click Next.
13. Review the settings for the job, and click Finish.

Verify success by comparing configs

1. Click My Dashboards > Configs > Configuration Management.
2. Select one or more nodes, and click Compare Node(s) Configs.
3. Select a config from each list.
4. Click Compare Selected. Changes to are highlighted in red, green, and yellow.
Change an interface description

Updating interface descriptions with SolarWinds Network Configuration Manager saves time because you do not have to remember IP addresses or login credentials for the device you are updating.

Back up the running configuration

1. Click My Dashboards > Configs > Configuration Management.
2. Select one or more nodes.
3. Click Download > Running.

Update the interface descriptions

1. Click My Dashboards > Configs > Jobs.
2. Click Create New Job.
3. Name the job, and select Execute Command Script on Devices from Job Type.
4. Select the Schedule Type. If you are creating an Advanced schedule, use the five fields to create a CRON expression.
5. Add a comment if this job relates to a business rule, and click Next.
6. Select the nodes to target with this job, and click Next.
7. Select an email notification option, and click Next. If you click Email Results, the default email notification and SMTP server settings are populated. These settings can be overridden in each job.
8. Add the script to execute.
   - If you want to load a saved script, click Load Saved Script, and select a script.
   - If you want to load an existing script from disk, click Load Script from File, browse to the file, and click Open.
   - If you want to create a new script, enter your script in the text box. For example:
     
     ```
     config t
     interface Ethernet0
     no description
     description Link to Upstairs Lab
     exit
     exit
     wr mem
     ```
     
     Where Link to Upstairs Lab is the new description.
9. If you want to save a script, click Save Script to File, specify a location, and click Save.
10. Select Filter Results that match a pattern if you want to see in the script output only those lines that match a specific regular expression pattern.
11. Select Show Commands in Output to view what SolarWinds NCM sent to the targeted devices.
12. Click Next.

13. Review the settings for the job, and click Finish.

Verify success by checking the interface description

1. Click My Dashboards > Configs > Config Summary.
2. Click the updated node in the list.
3. Click the Configs tab.
4. Under Config List, select the running config, and click Edit Config.
5. Locate the interface definitions.
6. Verify that the new description has been applied to the interface you modified.

Upgrade IOS and firmware

You can use SolarWinds NCM to upgrade the firmware on many devices at one time. Because security fixes are distributed as new firmware revisions, upgrading firmware helps network administrators correct security vulnerabilities.

You can use the wizard under Firmware upgrades to simplify this process.

Block all private addresses with an access control list

Routers connected the Internet are normally configured to discard any traffic using private IP addresses. This isolation gives your private network a basic form of security as it is not usually possible for the outside world to establish a connection directly one of your network devices using these addresses. The following procedure updates the access control list to block all private IP addresses on several devices at the same time.

Back up the running configuration

1. Click My Dashboards > Configs > Configuration Management.
2. Select one or more nodes.
3. Click Download > Running.

Update the access control list for a group of nodes

1. Click My Dashboards > Configs > Jobs.
2. Click Create New Job.
3. Name the job, and select Execute Command Script on Devices from Job Type.
4. Select the Schedule Type. If you are creating an Advanced schedule, use the five fields to create a CRON expression.

5. Add a comment if this job relates to a business rule, and click Next.

6. Select the nodes to target with this job, and click Next.

7. Select an email notification option, and click Next. If you click Email Results, the default email notification and SMTP server settings are populated. These settings can be overridden in each job.

8. Add the script to execute.
   - If you want to load a saved script, click Load Saved Script, and select a script.
   - If you want to load an existing script from disk, click Load Script from File, browse to the file, and click Open.
   - If you want to create a new script, enter your script in the text box. For example:
     ```
     ${EnterConfigMode}
     access-list 102 deny ip 10.0.0.0 0.255.255.255 any log
     access-list 102 deny ip 172.16.1.0 0.15.255.255 any log
     access-list 102 deny ip 192.168.0.0 0.0.255.255 any log
     exit
     write memory
     
     Where 102 is the name of the ACL. ${EnterConfigMode} is a variable that is equivalent to `config terminal` on Cisco devices.
     ```

9. If you want to save a script, click Save Script, specify a location, and click Save.

10. Select Filter Results that match a pattern if you want to see in the script output only those lines that match a specific regular expression pattern.

11. Select Show Commands in Output to view what SolarWinds NCM sent to the targeted devices.

12. Click Next.

13. Review the settings for the job, and click Finish.

### Verify success by comparing configs

1. Click My Dashboards > Configs > Configuration Management.

2. Select one or more nodes, and click Compare Node(s) Configs.

3. Select a config from each list.

4. Click Compare Selected. Changes to are highlighted in red, green, and yellow.

### Customize the login banner of a device

You can easily change the login banner for a router, switch, or firewall using SolarWinds Network Configuration Manager. This customization can be rolled out to a single or multiple devices.
Modify the login banner

1. Click My Dashboards > Configs > Jobs.
2. Click Create New Job.
3. Name the job, and select Execute Command Script on Devices from Job Type.
4. Select the Schedule Type. If you are creating an Advanced schedule, use the five fields to create a CRON expression.
5. Add a comment if this job relates to a business rule, and click Next.
6. Select the nodes to target with this job, and click Next.
7. Select an email notification option, and click Next. If you click Email Results, the default email notification and SMTP server settings are populated. These settings can be overridden in each job.
8. Add the script to execute.
   - If you want to load a saved script, click Load Saved Script, and select a script.
   - If you want to load an existing script from disk, click Load Script from File, browse to the file, and click Open.
   - If you want to create a new script, enter your script in the text box. For example:

```
config t
  no banner login
  banner login ^Unauthorized use of these systems is punishable by law^ exit
  wr mem

Where Unauthorized use of these systems is punishable by law is the new banner.
```
9. If you want to save a script, click Save Script, specify a location, and click Save.
10. Select Filter Results that match a pattern if you want to see in the script output only those lines that match a specific regular expression pattern.
11. Select Show Commands in Output to view what SolarWinds NCM sent to the targeted devices.
12. Click Next.
13. Review the settings for the job, and click Finish.

Verify success by comparing configs

1. Click My Dashboards > Configs > Configuration Management.
2. Select one or more nodes, and click Compare Node(s) Configs.
3. Select a config from each list.
4. Click Compare Selected. Changes to are highlighted in red, green, and yellow.
Create a CRON expression

In Unix computing, the CRON daemon performs all scheduled jobs. Each job is defined in a command line syntax that looks like this:

\{Minutes: 0-59\} \{Hours: 0-23\} \{Day of Month: 1-31\} \{Month: 1-12\} \{Day of Week: 0-6 \(\text{Sunday to Saturday}\)\} \{Command to Execute\}

SolarWinds NCM adopts this model in the resource of the advanced schedule type. The five input fields correspond to the five elements of CRON job syntax. Additionally, SolarWinds NCM supports these CRON operators:

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Asterisk</td>
<td>The wildcard for the value all</td>
</tr>
<tr>
<td>-</td>
<td>Hyphen</td>
<td>Sets a range: #-#</td>
</tr>
<tr>
<td>,</td>
<td>Comma</td>
<td>Specifies terms out of series</td>
</tr>
</tbody>
</table>

Example

If want a job to run twice a week every month at midnight on Sunday and Thursday morning, the values for the fields would be:

\(0 0 * * 0, 4\) \{Command\}

This tells SolarWinds NCM to run the job at 0 minutes and 0 hours, on all valid days of month, in all valid months, on Sunday and Thursday.
Get started with NCM

A significant amount of time spent managing your network devices can be decreased using SolarWinds Network Configuration Manager. Before adding or discovering nodes to manage in NCM, make sure your devices are properly set up to communicate with NCM and to export network flows.

The following sections walk you through adding nodes to monitor and other common tasks.

SolarWinds NCM uses SNMP communication to collect inventory information.

SNMP communication

SolarWinds Network Configuration Manager uses SNMP communication to collect inventory information. Properly configure SNMP on all devices from which you want to collect detailed information.

Add nodes to NCM

Use any of the following methods to add nodes for configuration management in the SolarWinds Network Configuration Manager:

- Add an individual node.
- Add nodes with Network Discovery.
- Import a list of IP addresses or host names from a CSV file.

Add an individual node

Use this option to add a single node that you want to start managing in your Network Configuration Manager installation. If adding a large number of nodes, consider using Network Discovery.

1. Click Settings > Manage Nodes.
2. Click Add Node.
3. Type the hostname or IP Address of the node you are adding. Both IPv4 and IPv6 are supported.
4. Select a Polling Method:
   a. External Node — reports no status for node itself but is useful for monitoring a hosted application.
   b. Status Only: ICMP — reports status, response time, and packet loss only.
   c. Most Devices: SNMP and ICMP — default polling method typically used for routers, switches, and Linux/Unix servers. Make sure to specify a community string if using one other than "public".
5. Use the TEST option to verify that your settings are valid before continuing.
6. Select the Additional Monitoring Options, and click Next.
6. Select the interfaces, volumes, and interface charts to monitor for the added node, and click Next.

7. On the Change Properties tab, under Polling, edit the default polling settings if desired.
   - Node Status Polling: The number of seconds between status checks performed on the added node.
   - Collect Statistics Every: The period of time between updates made to displayed statistics for the added node

8. Under Custom Properties, add values if you defined custom properties for monitored nodes.
   - The Custom Properties area is empty if you did not define custom properties for monitored nodes.
   - SolarWinds NCM ignores custom properties that use words from the list of reserved words.

9. Click OK, Add Node when you have completed the properties configuration.

The added node appears in the Config Summary.

ℹ️ You can also use the More Actions menu on the Managed Nodes page to "Add Nodes to NCM" that you have selected.

### Add nodes via Network Discovery

Network Discovery is particularly useful when you have multiple devices that you want to add for configuration management. At first, SolarWinds recommends scanning a small subnet or a few individual IP addresses.

1. Click Settings > Network Discovery.
2. Click Add New Discovery, and click Start.
3. To discover devices on your network within a specific range of IP addresses:
   a. In the IP Ranges section, click Add Range and provide a Start and End Address for the range.
      Scheduled discovery profiles should not use IP address ranges that include nodes with dynamically assigned IP addresses (DHCP).
   b. Repeat to add ranges, and then click Next when finished.
4. To discover devices connected to a specific router or on a specific subnet of your network:
   a. In the Subnets section, click Add > Subnet, and enter a Subnet IP Address. Repeat to add subnets.
   b. If you want to discover devices using a seed router, click Add > Seed Router, enter the IP address of the Router, and click Scan Router.

   - Repeat this step for each additional seed router you want to use.
   - Network Sonar reads the routing table of the designated router and offers to discover nodes on the Class A network (255.0.0.0 mask) containing the seed router and, if you are discovering devices for an SolarWinds NPM installation, the Class C networks (255.255.255.0 mask) containing all interfaces on the seed router, using the SNMP
version chosen previously on the SNMP Credentials page.

- Networks connected through the seed router are not automatically selected for discovery.
  c. When finished, click Next.

5. If you already know the IP addresses or hostnames of the devices you want to discover:
   a. In the IP Addresses section, click Add IP Address and enter the IPv4 or IPv6 addresses or hostnames of the devices. Type only one address or hostname per line.
   c. Click Validate to confirm that the provided IPv4 addresses and hostnames are assigned to SNMP-enabled devices.
   d. When finished, click Next.

6. If you want to import nodes by querying your Active Directory Domain Controllers for registered devices, click Add Active Directory Domain Controller to Query. Choose existing credentials or add the credentials to access the Active Directory server.

7. On the Agents tab, decide whether to poll for updates on these nodes, and click Next.

8. On the Virtualization tab, decide whether to poll for VMware, and click next.

9. On the Config Management tab, Add SSH or Telnet credentials, and click Next.

10. On the SNMP tab, specify SNMP credentials for discovery:
    a. If the devices on your network do not require community strings other than the default public and private strings provided by SolarWinds NCM, click Next.
    b. If any of your network devices require community strings other than public and private provided, or if you want to use an SNMPv3 credential, add the required SNMP credential for each community string:

        i. To speed up discovery, highlight the most commonly used community strings on your network, and then use the arrows to move them to the top of the list. NCM supports SNMPv3 with AES-256.

        a. Click Add New Credential.
        b. Select the SNMP Version, enter the required information, and click Add.
        c. Click Next.


12. Set polling and monitoring settings, and click Next.

13. On the Discovery Settings tab, edit the Details, Retries, and Timeouts.
    a. Name and Description: distinguishes the current discovery profile from other profiles you may use to discover other network areas. This Description displays next to the Name in the list of available network discovery configurations on the Network Sonar view.
    b. SNMP Timeout: If you are encountering numerous SNMP timeouts during Network Discovery, increase the value for this setting. A recommended value is at least more than double the time it takes a packet to travel the longest route between devices on your network.
b. Search Timeout: The amount of time Network Sonar Discovery waits to determine if a given IP address has a network device assigned to it.

c. SNMP Retries: The number of times Network Sonar Discovery retries a failed SNMP request, when a response is not received within the time set in SNMP Timeout.

d. Hop Count: If the Hop Count is greater than zero, Network Sonar Discovery searches for devices connected to any discovered device. Each connection to a discovered device counts as a hop.

e. Discovery Timeout: The amount of time, in minutes, Network Sonar Discovery is allowed to complete a network discovery. If the Discovery Timeout value is exceeded, the discovery is terminated.

14. The default Frequency for discovery is Once, which starts immediately. If you want the discovery to run on a regular schedule, select Hourly, Daily, or Advanced as the Frequency, and click Discover.

15. If you do not want to run the network discovery, select No, Don't Run Now, and click Save or Schedule, depending on the Frequency selection.

Select network devices for monitoring in NCM

The Network Sonar Results Wizard opens to guide you through selecting the devices from discovery results that you want to monitor. These results are available whenever a requested or scheduled SolarWinds Network Sonar Discovery operation completes, or whenever you choose to either Import All Results or Import New Results for a selected discovery profile.

The Results Wizard automatically adds non-server nodes as SolarWinds NCM nodes if there is at least one connection profile set up with Use for Auto Detect enabled.

Select the results of a network discovery

1. Click Settings > Network Discovery.
2. Select an existing discovery profile, and click Import New Results.
3. Select the device types you want to monitor, and click Next.
4. Select the interface types you want to monitor, and click Next.
5. Select the volume types you want to monitor, and click Next.
6. Select the ApplInsight applications to use with your environment, and click Next.
7. If there are any devices on the Import Preview that you do not want to import, select the device, and click Ignore.
   Selected nodes are added to the Discovery Ignore List.
8. Confirm that the network objects you want to monitor are selected on the Import Preview tab, and click Import.
9. After the import completes, click Finish.
10. Click My Dashboards > Home > Summary to see imported devices in the All Nodes resource.
Communication and logging options

See the following topics for information about setting communication options for nodes, applying global device logins, and enabling logging.

- Set node communication defaults
- Define and apply global device logins
- Set communication limits
- Log events for a detailed record of SolarWinds NCM functions
- Enable SSL communication with SolarWinds NCM

Set node communication defaults

A number of variables can be set globally and applied to all new nodes added to SolarWinds NCM. You can override the defaults when adding nodes, however.

Though SolarWinds NCM supports SNMPv3 with AES-256, the SNMPWalk utility only supports AES-128. If you need to troubleshoot SNMP related to your node, use SNMPWalk to test the SNMPv3 setting for AES-128, and set it back to AES-256 after you have resolved the SNMP-related issue.

Set node communication default parameters

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
4. Enter the appropriate values, and click Submit.

Override default settings for a specific node

1. Click Settings > Manage Nodes.
2. Use the Group By list to filter the nodes.
3. Select a node in the list, and click Edit Properties.
4. Scroll down to NCM Properties.
5. Edit login and communication information, and click Submit.

Define and apply global device logins

A connection profile is a global device login that you apply to one or more devices managed by SolarWinds NCM.

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
4. To create a new connection profile, click Create New.
5. To edit an existing profile, select a profile, and click Edit.

6. Enter values. See below for more information on the last option, Automatically Test This Profile Against Monitored Nodes That Allow It:
   - Make this profile available to SolarWinds NCM in communicating with nodes for which no profile is assigned.
   - If you have multiple connection profiles with this setting enabled, SolarWinds NCM cycles through them before performing the desired operation on a node without a specific connection profile applied. SolarWinds NCM permanently assigns the first connection profile that works to the node.
   - Though selecting this setting on your connection profiles helps SolarWinds NCM communicate with devices without assigned profiles, we recommend applying a specific connection profile to each managed node.

7. Click Submit.

Set communication limits
Define timeout values and retry numbers for ICMP, SNMP, Telnet, and SSH communication.

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
4. Enter timeout settings for each protocol, and click Submit.

Log events for a detailed record of SolarWinds NCM functions
Enable verbose log information
Select which log types to keep verbose log information for. This information aids in troubleshooting.

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
4. Select the log types.
5. Click Submit.

Enable session tracing
Enable session tracing to create a log file of each Telnet session to troubleshoot communication node by node.

1. Click Settings > All Settings.
2. Under Product Specific Settings, click CLI Settings.
3. Select Enable Session Tracing.
Enable SSL communication with SolarWinds NCM

SolarWinds NCM supports the use of Secure Sockets Layer certificates to enable secure communications with the Orion Web Console. Secure SSL communications are conducted over port 443.

An SSL certificate must be installed on your SolarWinds NCM server.

1. Log in as an administrator to your main SolarWinds NCM server.
3. Expand Services and Applications > Internet Information Services (IIS) Manager > Web Sites.
5. Click the Web Site tab.
6. Confirm that SSL port is set to 443, and click Apply.
7. Click Advanced.
8. If the Multiple SSL Identities for This Web Site field does not list the IP address for the Orion Web Console with SSL port 443, complete the following steps:
   a. Click Add, and select the IP address of the Orion Web Console.
      As it was set initially in the Configuration Wizard, this option is usually set to (All Unassigned). If the IP address of the Orion Web Console was not initially set to (All Unassigned), select the actual, configured IP address of the Orion Web Console.
   b. Type 443 as the TCP port, and click OK.
10. Click Edit in the Secure communications section.
11. Select Require Secure Channel (SSL), and click OK.
12. Click Apply, and click OK to exit.
Node management

SolarWinds Network Configuration Manager helps you manage, organize, and track changes to your network devices. These devices are known collectively as nodes and include switches, routers, firewalls, and Windows servers.

After you Add nodes to NCM you can begin to Manage SolarWinds NCM nodes and their device configurations. If you use SolarWinds Network Sonar Discovery to add nodes, you can then Select network devices for monitoring in NCM from the discovery results.

Manage SolarWinds NCM nodes

The NCM Manage Nodes view shows the status of managed nodes and enables you to select added nodes to determine when and how the devices will be managed. You can modify the node properties, work with custom properties, and initiate polling and discovery actions.

Click Settings > Manage Nodes.

To add or change columns displayed in the view, click the double-right angle icon at the far right of the table header and choose the data you want displayed.

Find a specific node

Enter an IP address or hostname in the Search field, and click Search.

Add or edit a node

1. Use the Group By list to filter the nodes.
2. Select one or more nodes you want to license in SolarWinds NCM, and click Edit Properties.
3. Scroll down to the Manage Node(s) with NCM list, and select Yes, No, or Never.

If you remove a node from SolarWinds NCM, all data associated with the node, such as configs and inventory data, are also removed.

View node interface details

The interface details view displays the following information about interfaces being monitored on NCM nodes.

- Interface/Port Status
- Interface Traffic
- Interface/Port Configuration
• Time data was last transmitted/received
• Cisco Discovery Protocol (CDP)
• User activity
• Real-time ARP cache

View the interface details of a specific node

1. Click the Interfaces tab.
2. Select a category from the Interface Details list.
3. Click Refresh to view the selected statistics.
4. Select Auto Refresh to update statistics every 30 seconds.
5. To change the Auto Refresh time interval:
   a. Click Interfaces under Node Details.
   b. Adjust the interval slider.
   c. Click OK.

Manage end of support and end of sales (EOS)

Use the End of Support & End of Sales Lookup to search vendor-published end of support and sales dates associated with your SolarWinds NCM devices. You can also search for specific nodes and assign EOS dates based on information related to device models. Automatic refresh of EOS data occurs once per day.

Open the EOS lookup

Click My Dashboards > Configs > End of Support.

NCM shows the End of Support and End of Sales Lookup data currently associated with your SolarWinds NCM devices.

SolarWinds neither verifies nor supports EOS data provided by EOS Lookup. Consult your vendor with any data-related issues or questions.

Refresh EOS dates

SolarWinds NCM maintains a database with EOS data for vendor device models. Based on a schedule, NCM matches EOS data with the machine type of your NCM devices. What you see in the table of the EOS resource is the result of the matching. To make sure you are looking at the latest matches, use Refresh Suggested Dates.

1. Click My Dashboards > Configs > End of Support.
2. Select the devices for which you want updated data, if available. Use the Group By list if necessary.
3. Click Refresh Suggested Dates.
Assign EOS dates

1. Click My Dashboards > Configs > End of Support.
2. Select the devices for which you want updated data, if available. Use the Group By list if necessary.
3. Click Assign Dates.
   You should see a list of your selected devices on the Assign EOS Data screen under Node Selected. If a device is missing, click Add More Nodes, select the nodes, and click OK.
4. Search the table for the model of your selected SolarWinds NCM nodes in the Choose Dates table. If you find it, select that row.
5. Search the table for the model of your selected SolarWinds NCM nodes in the Set EOS Dates table. If you find it, select that row.
   The dates listed for a model or series have indications in the Reliability column:
   - High indicates that the date(s) are unambiguous and such dates are automatically applied to the relevant devices.
   - Medium indicates that the date(s) remains ambiguous due to other incomplete or conflicting information.
   - Confirmed indicates that the date(s) were confirmed by a SolarWinds NCM user.
6. If you did not find the model of your selected SolarWinds NCM nodes but want to assign dates anyway, select Option 2 and define your own dates.
7. Add comments, and click Assign.

Ignore devices in EOS Management

1. Click My Dashboards > Configs > End of Support.
2. Select the devices for which you want updated data, if available.
3. Click Ignore Devices.
4. You should see a list of your selected devices on the Assign EOS Data screen under Node Selected. If a device is missing, click Add More Nodes, select the nodes, and then click OK.
5. Search the table for the model of your selected SolarWinds NCM nodes in the Choose Dates table. If you find it, select that row.
6. If you did not find the model of your selected SolarWinds NCM nodes, but you want to assign dates anyway, select Option 2 and define your own dates.
7. Add comments as needed.
8. Click Assign.

Export EOS information

1. Click My Dashboards > Configs > End of Support.
2. Select the devices for which you want updated data, if available. Use the Group By list if necessary.
3. Click Export, and select the format.
4. Select a location, and click OK.

Filter EOS data

By default, SolarWinds NCM presents data in the End of Support and Sales table for all nodes it manages. SolarWinds NCM collects data either on a daily schedule or when you choose to Refresh Suggested Dates. You can filter data in the table by column. Each filter you create is applied in the order it is listed above the table. For example, if you first set a filter for Name, SolarWinds NCM filters the Names column before it applies whatever filter comes next in the list of filters.

Setting filters is useful in the following scenarios:

- Displaying which devices reach End of Support or End of Sales at the same time (for example, in the Next 3 Months)
- Displaying EOS status for devices from the same vendor
- Displaying EOS status on devices in the same subnet

If you want to find the EOS status for a specific device, enter the device name or IP address in the Search window.

Delete EOS data

1. Click My Dashboards > Configs > End of Support.
2. Select the devices for which you want updated data, if available. Use the Group By list if necessary.
3. Click Delete EOS Data.
4. Click OK.

SolarWinds NCM macros

SolarWinds NCM macros are used by all SolarWinds NCM web and desktop applications and apply to all editable Node fields.

All SolarWinds NCM macros are enclosed in ${ }$. The macro for system name, for example, is ${SysName} . You can concatenate any number of macros in each editable node field. For example, the macros ${SysName}${Vendor} contain the system name and vendor in the field.

SolarWinds NCM can also define macros that point to other macros, and the macro parser can recursively parse the chain of macros applied in the node field.
Supported macro types

<table>
<thead>
<tr>
<th>MACRO</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes macros</td>
<td>Macros which point to another column in Nodes table. For example: ${SysName} points to the SysName column in the Nodes table. Node macros are unique to each Node.</td>
</tr>
<tr>
<td>Global macros</td>
<td>Macros defined on the application level and stored in the GlobalSettings table. As their name suggests, the value of these macros is the same for all the nodes. Several global macros, such as ${GlobalCommunityString}, are predefined. You can also create custom global macros.</td>
</tr>
<tr>
<td>Device template macros</td>
<td>Macros related to a device vendor and stored in device templates. For example: ${ConfigType}.</td>
</tr>
<tr>
<td>Menu-based macros</td>
<td>Macros defined to operate with menu-based devices. For example: ${DownArrow} simulates sending the Down Arrow key while connected to the device.</td>
</tr>
</tbody>
</table>

Custom macros

A custom macro is a global macro that you create to use in a script, job, or for a specific property that applies across all managed nodes.

Access custom macros

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
3. Under Advanced, click Manage Macros.

Create a custom macro

1. Click Add New.
2. Enter a name and value for the new macro.
   For example: To define a macro to track the provision date and location of devices, use ProvisionAustin1 as the name for devices in Austin, Texas provisioned on a certain date and use the city and date as the value:
   - Macro Name: ProvisionedAustin1
   - Macro Value: Austin 10/10/2015
3. Click Submit.
Edit a custom macro

1. Select the macro, and then click Edit.
2. Modify the value of the macro.
3. Click Submit.

Delete a custom macro

1. Select the macro.
2. Click Delete.
3. Click OK.

Find connected port for a host: SolarWinds NPM integration

A SolarWinds NCM integration with SolarWinds NPM allows you to search for currently connected ports on wired or wireless end hosts.

This feature requires that you manage both nodes and interfaces involved in the connection.

The information returned is based on the data available in the Orion database as of the last scheduled discovery of network nodes. It is specified at the top of the table in which search results are presented in this form: As of last discovery [MM/DD/YYYY] [HH:MM:SS] [AM/PM].

You can search for connected ports by these node properties:

- IP Address
- DNS Hostname
- MAC Address
- Port Description

Search for connected ports

1. Click My Dashboards > Configs > Config Summary.
2. Under Find Connected Port for End Host, select the Search By filter, and enter a value in the Find field.
3. Click Edit in the resource header to adjust the columns of data to include in your search results.
4. Click Find.

Results are presented as rows in a table, one for each connection within the reach of hop of the context node, with the following default columns:

<table>
<thead>
<tr>
<th>COLUMN</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>For Wired Devices</td>
<td></td>
</tr>
<tr>
<td>COLUMN</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Node</td>
<td>Vendor and model of context node</td>
</tr>
<tr>
<td>IP Address</td>
<td>Of the context node</td>
</tr>
<tr>
<td>MAC Address</td>
<td>Of the context node</td>
</tr>
<tr>
<td>Connected Via Interface</td>
<td>On the context node</td>
</tr>
<tr>
<td>To This Interface</td>
<td>On a connection point</td>
</tr>
<tr>
<td>On This Node</td>
<td>Vendor and model of device supporting the connection point</td>
</tr>
<tr>
<td>IP Address</td>
<td>Of a connection point</td>
</tr>
<tr>
<td>MAC Address</td>
<td>Of a connection point</td>
</tr>
</tbody>
</table>

**For Wireless Devices**

| Mapped Host Name                | Vendor and model of context node                                          |
| Mapped MAC Address              | Of context node                                                           |
| Mapped Device Type              | Of context node                                                           |
| Source Interface                | Of the wireless access point                                              |
| Controller Source IP Address    | Of the wireless access point                                              |
| Controller Description          | Of the wireless access point                                              |
| Controller Host Name Source SSID | Of the wireless access point                                              |
| Source Channel                  | Of the wireless access point                                              |
| Source Interface Alias          | Of the wireless access point                                              |
| Source Radio Type               | Of the wireless access point                                              |
| Source Host Name                | Of the wireless access point                                              |
| Source Device Type              | Of the wireless access point                                              |
| Source IP Address               | Of the wireless access point                                              |

**Unmanage and remanage nodes**

SolarWinds NCM does not perform job actions on unmanaged nodes. When performing maintenance on nodes, unmanage the nodes to discontinue processing jobs against them. Maintenance includes upgrading firmware, installing new hardware, or updating security.

Configuration data for unmanaged nodes will remain in the Orion platform database.
Access the management page

Click Settings > Manage Nodes.

Unmanage nodes

Select the nodes you want to unmanage in the nodes list, and click Unmanage.

Remanage nodes

Select the nodes you want to remanage in the nodes list, and click Remanage.
Firmware vulnerability data

SolarWinds NCM helps identify risks to network security by detecting potential vulnerabilities in the following device types:

- Cisco IOS
- Cisco Adaptive Security Appliance (ASA)
- Cisco Nexus
- Juniper

NCM imports the firmware vulnerability warnings provided by National Institute of Standards and Technology (NIST), and correlates vulnerability data with nodes that it currently manages. If NCM finds a match, the Firmware Vulnerabilities resource on the Config Summary page displays information about the vulnerability and the number of affected nodes.

- The firmware vulnerability feature is enabled by default. If necessary, you can enable or disable this feature or change other default settings.
- If you are on a closed network, you can manually import vulnerability data.

View firmware vulnerability details and update the remediation status

When a firmware vulnerability potentially affects one or more managed nodes, use the Vulnerability Summary page to get additional information and track the remediation status.

1. From the Firmware Vulnerabilities widget on the Config Summary dashboard, click a vulnerability's Entry ID.

   The Vulnerability Summary page displays a summary and the current state. You can click the URL to open the National Vulnerability Database web page for detailed information and links to related advisories and solutions.

2. Optionally, add a comment to record findings, plans, or completed actions.

3. Select the state that reflects the current remediation status:

<table>
<thead>
<tr>
<th>STATE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Potential vulnerability</td>
<td>The vulnerability has not yet been verified. (This is the default.)</td>
</tr>
<tr>
<td>Confirmed vulnerability</td>
<td>The vulnerability is confirmed but no remediation is planned.</td>
</tr>
<tr>
<td>Not applicable</td>
<td>The vulnerability does not apply to the selected nodes.</td>
</tr>
</tbody>
</table>

page 26
4. Apply the selected state to all nodes, or select specific nodes.
5. Click Submit.

View firmware vulnerability reports

Firmware vulnerability reports list vulnerabilities discovered in the last run of the vulnerability matching logic. That logic is based on data last downloaded from sources in Firmware Vulnerability Settings.

1. Click Reports > All Reports.
2. In the Group By list, select Report Category.
3. Click the NCM Security category.
4. Click the report name:
   - **Nodes for each Vulnerability** is organized by vulnerability. The associated nodes are listed below each vulnerability.
   - **Vulnerabilities for each Node** is organized by node. The associated vulnerabilities are listed below each node.
   - **Vulnerabilities for each Node - <stageName>** lists only the nodes and associated vulnerabilities in a specific remediation stage (for example, Confirmed or Remediation planned).

Each report includes the following information.

<table>
<thead>
<tr>
<th>FIELD</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caption/Entry ID</td>
<td>The Common Vulnerabilities and Exposures (CVE) identifier for a specific vulnerability.</td>
</tr>
<tr>
<td>IOS Version</td>
<td>The operating system software versions to which the CVE pertains.</td>
</tr>
<tr>
<td>IOS Image</td>
<td>The operating system software image to which the CVE pertains.</td>
</tr>
<tr>
<td>URL</td>
<td>The location of the CVE on the NIST website from which NCM obtained vulnerability data.</td>
</tr>
<tr>
<td>CVSS V2 Base Score</td>
<td>A score that reflects the severity of the vulnerability. This score is calculated using the Common Vulnerability Scoring System (CVSS). Use this information to prioritize remediation activities.</td>
</tr>
<tr>
<td><strong>FIELD</strong></td>
<td><strong>DESCRIPTION</strong></td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Severity</td>
<td>The severity of the vulnerability based on the CVSS score:</td>
</tr>
<tr>
<td></td>
<td>• Low (0-3.9)</td>
</tr>
<tr>
<td></td>
<td>• Medium (4.0-6.9)</td>
</tr>
<tr>
<td></td>
<td>• High (7.0-10.0)</td>
</tr>
<tr>
<td>State</td>
<td>The current status of remediation activities on the associated nodes.</td>
</tr>
<tr>
<td>Last State Change</td>
<td>The date on which the State last changed for the associated nodes.</td>
</tr>
</tbody>
</table>

**Troubleshoot firmware vulnerability reports**

If a node is not listed with others of its type in a vulnerability announcement, check for errors in the **Vulnerability Log** ($\{All\ Users\ Profile\}\Application Data\SolarWinds\Logs\Orion\NCM\VulnLib.log).

**Access firmware vulnerability settings**

SolarWinds NCM imports the **firmware vulnerability warnings** provided by National Institute of Standards and Technology (NIST) and correlates vulnerabilities with managed nodes. Use the firmware vulnerability settings to specify when the matching logic runs or change the default import locations or alert threshold.

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
5. Click Submit.

**Manually download and add firmware vulnerability files**

If SolarWinds NCM cannot automatically download firmware vulnerability data (for example, because you are on a closed network), you can import vulnerability data files from the National Institute of Standards and Technology (NIST) and then manually add them to your NCM server. NCM will use the information in these files to search for **vulnerabilities that could affect your devices**.

1. Download the firmware vulnerability .zip files from NIST.
2. Log in to your NCM server.
3. Verify the location of the vulnerability announcements folder. (The default location is C:\ProgramData\SolarWinds\NCM\Vuln\Xml.)
   a. Click Settings > All Settings.
   b. Under Product Specific Settings, click NCM Settings.
   c. Under Advanced, click Firmware Vulnerability Settings.
   d. Under Vulnerability Data Import Settings, verify the location of the folder with vulnerability announcements XML data.

4. Extract the contents of the .zip files to the location verified above.

5. Under Vulnerability Search Settings, click Run Now.
Firmware upgrades

You can use SolarWinds NCM to upgrade the firmware on many devices at one time. Because security fixes are distributed as new firmware revisions, upgrading firmware helps network administrators correct security vulnerabilities.

To ensure device security and compliance, use vulnerability reporting to identify NCM nodes at risk. Next, upgrade the firmware of those devices to address vulnerabilities.

See the following articles to get started:

- Set up the storage location and firmware repository
- Firmware upgrade templates
- Perform a firmware upgrade operation

Set up the storage location and firmware repository

Before you perform a firmware upgrade operation, you must complete the initial setup. First, you designate a storage location that will contain the firmware images. This feeds the firmware repository in NCM, from which you select images for devices when upgrading.

<table>
<thead>
<tr>
<th>STEP</th>
<th>INSTRUCTIONS</th>
</tr>
</thead>
</table>
| Set up the storage location | 1. Click Settings > All Settings.  
2. Under Product Specific Settings, click NCM Settings.  
3. In the Firmware Upgrade section, click Storage Location.  
4. Enter the path to the firmware upgrade storage location.  
   For basic installations with one polling engine, you can use a local path on the server where NCM is installed.  
   The storage location cannot be the same as the TFTP or SFTP/SCP root folders.  
   If you have more than one polling engine, use a shared network location. Make sure all polling engines have access to this location, and they are in the same domain.  
5. Enter credentials to access the storage location, and click Submit. |
Populate the firmware repository

Next, populate the storage location with firmware images through the file system. Now you can manage firmware upgrade images in NCM with the Firmware Repository:

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
3. In the Firmware Upgrade Storage section, click Firmware Repository.
4. Click Scan Firmware Repository to see the latest image files. By default, all files with the .bin extension are displayed.

If you select one or more entries and click Delete Firmware Images, the files are deleted from the folder you set as the storage location.

**Integrity of firmware images**

You can use the MD5 Hash column in the firmware repository to verify the integrity of your images. This value is calculated automatically when you add an image to the storage location. Compare the image hashes between the firmware repository and the website of the manufacturer of your device.

By default, NCM verifies the integrity of backed up and uploaded firmware images. You can toggle these settings under Upgrade Options in the Firmware Upgrade Operation wizard.

**Manage firmware upgrade templates**

Each firmware upgrade template defines a set of device-specific commands and options that SolarWinds NCM uses to upgrade the firmware on a device of that type. NCM provides a set of default firmware upgrade templates, and you can create new templates to enable firmware upgrades on other device types.

**NCM default firmware upgrade templates**

SolarWinds NCM provides firmware upgrade templates for the following device series:

- **Cisco ASA 5512 Multi Context Mode**
  
  Upgrades Cisco Adaptive Security Appliance (ASA) devices that have been partitioned into multiple contexts.

  To monitor a Cisco ASA device with multiple contexts, you must add each context as a node to be monitored by NCM. To do this, add the admin context by IP address. All other contexts are automatically discovered, and you can add each one as a node.

- **Cisco ASA 5512 Single Context Mode**
- **Cisco Catalyst 2960**
- **Cisco Catalyst 3560**
Cisco Catalyst 3750
Cisco Catalyst 3850
Cisco IOS Software 2800/2801
Cisco IOS Software 2921
Cisco Nexus 5548

This template performs an in-service software upgrade (ISSU) and then reboots the device. Rebooting the device ensures that the firmware upgrade is completed successfully. Use this template only for upgrades. The ISSU process does not support downgrades.

If a Cisco Nexus device has multiple virtual device contexts (VDCs), firmware updates apply to all VDCs on the physical device. You cannot upgrade the firmware for an individual VDC.

Cisco Nexus C7004 - ISSU

This template performs an ISSU and then reboots the device. Rebooting the device ensures that the firmware upgrade is completed successfully. Use this template only for upgrades. The ISSU process does not support downgrades.

Cisco Nexus C7004 - Traditional

This template performs a traditional upgrade or downgrade. Use this template when ISSU is not supported.

Add a firmware upgrade template

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
3. In the Firmware Upgrade section, click Firmware Upgrade Templates.
4. Perform one of the following actions:
   - To create a template based on a predefined NCM template, select the template and click Edit. Then click Duplicate & Edit Firmware Upgrade Template.
   - To create a new template from scratch, click Add.
5. Enter a name and description.
6. In the Collect Info Commands section, specify what commands to run to gather information from devices before they are upgraded with this template.
   To enable an action, select the checkbox and then enter the command to perform that action.
   For more information about the commands in each section, see the Firmware Upgrade Guide.
7. In the Upgrade Commands section:
a. Specify at least one firmware image type:

- For devices with only one image, enter any description (for example, "Main") in the Firmware Image Type box. Then click Add New.

<table>
<thead>
<tr>
<th>Firmware image(s)</th>
<th>Main</th>
<th>Not defined</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firmware Image Type</td>
<td>ADD NEW</td>
<td></td>
</tr>
</tbody>
</table>

- For Cisco Nexus devices, which have multiple images:
  a. Enter the first image type (for example, "System" or "Kickstart") in the Firmware Image Type box.
  b. Click Add New.
  c. Repeat to add other images.

<table>
<thead>
<tr>
<th>Firmware image(s)</th>
<th>System</th>
<th>Not defined</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kickstart</td>
<td>Not defined</td>
<td></td>
</tr>
<tr>
<td>Firmware Image Type</td>
<td>ADD NEW</td>
<td></td>
</tr>
</tbody>
</table>

b. (Optional) Associate a firmware image from the repository to this template.

If you do not associate a firmware image with the template, you can select one later when you **perform a firmware upgrade operation**.

c. Enter commands on how to upgrade the devices with this template.

8. In the Upgrade Options section, select the tasks to run before and after the upgrade with this template.

9. Click Save.

**Share firmware upgrade templates between Orion servers**

You can export a firmware upgrade template as an XML file, and then import it to another Orion server.

**Export a firmware upgrade template**

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
3. In the Firmware Upgrade section, click Firmware Upgrade Templates.
4. Select a template, and click Export As File.
   
   The XML file is saved in your Downloads directory. The file name is based on the template name.

**Import a firmware upgrade template**

You can import a firmware upgrade template that was previously exported as a file.
1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
3. In the Firmware Upgrade section, click Firmware Upgrade Templates.
4. Click Import.
5. Click Choose File, browse to the template, and select it.
6. Click Submit.

**Perform a firmware upgrade operation**

After you have set up your storage location and firmware repository, and reviewed or defined firmware upgrade templates, you can perform an upgrade operation.

**Add a firmware upgrade operation**

Create a firmware upgrade operation to upgrade the firmware on many devices at one time. The devices are upgraded one at a time in sequential order, which you can customize.

You must have the NCM Administrator and Orion Administrator roles to add and execute a firmware upgrade operation.

1. Click My Dashboards > Configs > Firmware Upgrades.
2. Click Add.
3. Enter an operation name, and select a firmware upgrade template. Click Next.
4. On the Select Firmware Image step, there are two options:
   - Use the firmware image preselected by the upgrade template. If an entry is listed under Select Firmware Image, click Next to use it.
   - Click Select New Image From Repository to use an image for all the nodes you select later. This option overrides the current settings.
5. Select the nodes you want to upgrade, and click Start Collecting Data. The data collection may take a few moments, so you are returned to the Firmware Upgrade Operations view.
6. When the data collection finishes, click Please Review to Continue in the Status/Details column of the operation.
7. The wizard resumes at the Upgrade Options step. Here you can apply specific upgrade options to each node. Review the Upgrade Options section before continuing.
8. After you verify the upgrade procedure, click Confirmed for each node and then click Next.

NCM displays a warning if there is not enough free space to continue the operation without removing the current firmware image. These upgrade options are automatically selected:

- Back up existing firmware image
- Delete existing firmware image
9. Select your email preference, and click Next.

10. On the Summary step, you can change the upgrade order, and decide whether to run the upgrade operation immediately or schedule it for later. Type YES to continue, and click Finish.

You are returned to the Firmware Upgrade Operations view. You can check the progress of the operation in the Status/Details column.

Upgrade Options

Determine image paths

**Path to upload image:** a required field under the selected firmware image. This is the path NCM uses to upload the new firmware image. NCM attempts to detect the existing image path and populate this field with the same drive where the existing image is located. For example, flash:.

**Back up existing firmware image:** an option under Upgrade Options with a required field to enter the image path. NCM attempts to automatically determine the path to the existing firmware image on the device. For example, flash:c2951-universalk9-mz.SPA.155-3.M1.bin.

If one or both of these fields are empty, refer to the documentation of your device for more information. You can also try to determine the path from the output of the `show version` command.

Config comparison

Select the following options to list a config comparison report in the log, or in an email notification if enabled, after the firmware upgrade operation is complete:

- Back up running and startup configs before upgrade
- Back up running and startup configs after upgrade

Operation statuses for firmware upgrades

Check the status of an operation in the Status/Details column of the Firmware Upgrades Operations view.

<table>
<thead>
<tr>
<th>STATUS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collecting data</td>
<td>NCM is executing the <code>collect info</code> commands for the upgrade operations.</td>
</tr>
<tr>
<td>Need review</td>
<td>All <code>collect info</code> commands have executed, and the operation is waiting for user review and confirmation of the upgrade options for each node.</td>
</tr>
<tr>
<td></td>
<td>If necessary, you can remove any node from the upgrade operation.</td>
</tr>
<tr>
<td>Queued</td>
<td>Another firmware upgrade operation is already in the &quot;Upgrading&quot; status, and NCM is waiting for it to finish before starting this operation.</td>
</tr>
<tr>
<td>Upgrading</td>
<td>NCM is currently executing the firmware upgrade scripts.</td>
</tr>
<tr>
<td>Complete</td>
<td>The firmware upgrade operation finished successfully.</td>
</tr>
</tbody>
</table>
### Additional actions for firmware upgrade operations

Based on the operation status of a firmware upgrade, you may need to cancel, rollback, or re-execute the operation.

#### Cancel a firmware upgrade operation

If you made a mistake when configuring an operation and it currently has the "Upgrading" status, you can cancel the active firmware upgrade operations.

> **If you cancel a currently running operation, you do not have the option to rollback.**

1. Click My Dashboards > Configs > Firmware Upgrades.
2. Select one or more entries, and click Cancel Selected.

#### Rollback a firmware upgrade operation

If there is a problem with the new firmware, you can restore each node in a firmware upgrade operation to its previous image, boot register, and configuration.

1. Click My Dashboards > Configs > Firmware Upgrades.
2. Select one or more entries, and click Rollback.

#### Re-execute a firmware upgrade operation

If some of the nodes in a firmware upgrade operation were not upgraded successfully, you can re-execute the operation on only the failed nodes.

1. Click My Dashboards > Configs > Firmware Upgrades.
2. Select one or more entries, and click Re-execute.

#### Review a firmware upgrade operation

If you started but did not complete or execute a firmware upgrade operation, you can review it and continue.

1. Click My Dashboards > Configs > Firmware Upgrades.
2. Select an entry, and click Review.
   
   You are returned to the Upgrade wizard.

---

<table>
<thead>
<tr>
<th><strong>STATUS</strong></th>
<th><strong>DESCRIPTION</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>The firmware upgrade operation failed.</td>
</tr>
<tr>
<td></td>
<td>See <a href="#">Additional actions for firmware upgrade operations</a> to review your options.</td>
</tr>
</tbody>
</table>
Configuration files

When nodes are added to NCM, you can download and edit the configuration files of those devices, compare configuration changes, and create an archive of config files for backup. NCM recognizes any configuration file with a minimum of 11 lines as valid for file transfer. You can change this default value if you plan to work with configuration files that have fewer lines.

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
3. In the Configs section, click Configs.
4. Under Config Transfer, use the slider bar to adjust the value for Config Min Length, or type the new value.
   This setting changes the default value for the minimum number of lines recognized by NCM as valid for download.

Use the Simultaneous Downloads/Uploads value to throttle the load that a big job could put on the network. By default, this value is set to 25 download/upload sessions.

If you plan to use NCM to back up device configurations that are in binary file format, complete the **Setup required for binary configs**. Editing or running comparisons on binary configuration files are not currently an option in NCM.

Setup required for binary configs

SolarWinds NCM stores binary configuration files that you download on a network share. If you will be managing device configurations that are in binary format, you must first specify the network share that you want NCM to use.

1. Click Settings > All Settings in the menu bar.
2. In the Product Specific Settings section, select NCM Settings.
3. In the Configs section, select Binary Config Storage Settings.
4. Enter the path to the network share. For example: \MYLAB-SERVERNAME-01\binarystorage.
5. Enter the credentials required to access the network share.
6. To validate the connection before submitting your settings, click Validate.
7. When finished, click Submit.

The NCM server must be in the same domain as the network share system.
Download configuration files in NCM

Download configuration files to view the current configuration of your managed devices, compare current and previous configurations, or archive configuration files for backup. SolarWinds NCM can transfer files using direct and indirect transfers.

If a configuration includes the command prompt character, SolarWinds NCM stops downloading. SolarWinds NCM provides the `UseExactPromptMatch` device template command you can use to instruct NCM to act on the command prompt only if it appears on a separate line. For more details about using this command, see Troubleshooting: Confine recognition of the command prompt during download.

Downloading from a single node

1. Log in to the Orion Console using an account that has the WebDownloader, WebUploader, Engineer, or Administrator role.
2. Click My Dashboards > Configs > Config Summary.
3. Under NCM Node List, click a node in the list.
4. Click Configs on the left.
5. Under Download Config, select the config type, and click Download.

By default, NCM does not download a config with fewer than 11 lines. Adjust the Config Min Length setting if necessary. For more information, see Managing Configuration Files.

6. If you receive a connection error, click Fix Connection in Device Template, and follow the Device Template wizard.
7. Follow the prompts of the Device Template wizard:

Downloaded configuration files are stored on your server in an archive. The location is specified in NCM Settings > Configs > Config Archive Folder Locations.

Downloading from multiple nodes

1. Click My Dashboards > Configs > Jobs.
2. Click Create New Job.
3. Name the job, and then select Download Configs from Devices from Job Type.

By default, NCM does not download a config with fewer than 11 lines. Adjust the Config Min Length setting if necessary. For more information, see Managing Configuration Files.

4. Select the Schedule Type. If you are creating an Advanced schedule, use the five fields to create a CRON expression.
5. Add a comment if this job relates to a business rule, and click Next.
6. Select the nodes to target with this job, and click Next.
7. Select an email notification option, and click Next. If you click Email Results, the [default email notification and SMTP server settings](#) are populated. These settings can be overridden in each job.

8. Select the configuration types you want to download.

9. Set your notifications preference, and click Next.

10. Review the settings for the job, and click Finish.

**Enable a new config type**

You can create a custom type of configuration to download from relevant devices. To do this you must enable SolarWinds NCM to recognize the new type and modify the templates for devices from which you intend to download the new type of configuration.

All functions that operate on standard config types in SolarWinds NCM operate as well with custom config types, with the following exceptions:

- The Overall Baseline vs. Running Config Conflicts chart is limited to the specified standard config types.
- The Overall Running vs. Startup Config Conflicts chart is limited to the specified standard config types.

**Enable a new config type for a device**

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
3. Under Configs, click Configs.
4. Under Config Types, enter a name for the new config type.
5. Click Add New.
6. Click Submit.
7. See the [Communication process diagrams](#) to modify your device template with a statement.

**Enable the config and policy caches**

When the config and policy caches are enabled, NCM automatically updated cached configuration information at the specified time. By default, the config and policy caches are enabled, and the policy cache is generated at 11:55 PM.

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
5. Specify what time to generate the policy cache.
6. Click Submit.

**Edit config files**

To update access lists, modify community strings, or make other configuration changes, edit the configuration files you downloaded with SolarWinds NCM.

1. Click My Dashboards > Configs > Config Summary. Use an account that has the WebDownloader, WebUploader, Engineer, or Administrator role.
2. Click a node in the node list on the left.
3. Click Configs on the left.
4. Under Config List, select a config, and click Edit Config.
5. Edit the Config Title if necessary.
6. Select Edit Config Text, and make changes.
7. Add comments, and click Submit.
   The revision is saved in the Config List with the indication that it is an Edited config of its type.

**Upload the edited config file**

1. Under Upload Config, select the edited config file.
2. Click Upload.

**Troubleshooting: Confine recognition of the command prompt during download**

By default, SolarWinds NCM stops downloading a config file that contains the device command prompt in the body of the config.

Use the `UseExactPromptMatch` command to prevent SolarWinds NCM from recognizing the command prompt unless it appears on a separate line:

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
4. Select a device template.
5. Click Edit > Using XML Editor.
6. Add this line to the Template XML:
   
   `<Command Name="UseExactPromptMatch" Value="true"/>

7. Click Save.
Compare configuration files

SolarWinds NCM can compare configuration files between two nodes or different versions of configs on the same node. Use NCM to:

- Compare running and startup configs to detect changes that were made to the in-memory config but were not permanently saved to non-volatile memory (NVRAM). These changes will be lost if the device loses power or is rebooted.
- In a multi-vendor environment, compare configs from different vendors for reference purposes.
- Compare different versions of the same config to quickly see what changed.
- Compare the configs on different nodes to ensure they match.

To avoid flagging inconsequential changes (such as the last updated date), you can specify lines or blocks to exclude from comparisons.

1. Click My Dashboards > Configs > Configuration Management.
2. Select the nodes:
   - To compare configs from two different nodes, select both nodes.
   - To compare two configs from the same node, select a single node.
3. Click Compare Node(s) Configs, and then select Compare node(s) configs from the drop-down menu.
4. Select the configs you want to compare.
   - The Compare Node(s) Config window displays the configs next to each other.
5. Review the configs for changes (yellow), added lines (green), and missing lines (red).
6. Click Edit Config for either config if you need to make changes.
7. Click Set/Clear Baseline on either config to make it the baseline. NCM uses the baseline to alert you to future config changes.
8. If a config is obsolete and should be removed, click Delete Config.
9. Click Close.

Exclude config content from comparison

You can identify lines or blocks of text that you do not want SolarWinds NCM to evaluate when it compares configs. For example, you can exclude:

- Empty lines
- Lines that do not represent substantive changes, such as a Last Updated line
- Blocks of text that do not represent substantive changes, such as the certificate section of the config

Excluding this content speeds up the comparison process and makes reviewing the compared configs easier.
NCM uses regular expressions to identify the content that it should ignore during comparisons. You can create and manage regular expressions on the Comparison Criteria page. NCM uses these regular expressions in all comparison operations, including manual comparisons, config change reports, and scheduled jobs.

Access the Comparison Criteria page

From the Comparison Criteria page, you can create, edit, enable, disable, or delete a regular expression.

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
3. Under Configs, click Comparison Criteria.

Create a new regular expression

Create a regular expression that identifies the text that you want NCM to ignore when comparing configs.

1. On the Comparison Criteria page, click Add New.
2. Enter a descriptive title.
3. Identify the text to exclude:
   - To exclude a single line, enter a regular expression that identifies the beginning of the line.
   - To exclude a block of text:
     a. In the RegEx pattern box, enter the regular expression that identifies the beginning of the first line in the block.
     b. Select Is block.
     c. In the End RegEx pattern box, enter the regular expression that identifies the beginning of the last line in the block.

   The search for the pattern that identifies the end of the block is "lazy." If the pattern occurs multiple times, NCM finds the first occurrence of the pattern.

4. To activate this regular expression, select Enable NCM to ignore this pattern. Otherwise, leave it disabled for future editing.
5. Click OK.

When you create and enable a new regular expression, it applies to manual config comparisons immediately. However, you must download the config again before the new regular expression affects config change reports and charts (such as the Overall Configuration Changes Snapshot).
Edit a regular expression

1. Select a regular expression on the Comparison Criteria screen, and click Edit.
2. Update any values on the Edit RegEx Pattern dialog.
3. Click OK.

Enable or disable a regular expression

Select one or more regular expressions on the Comparison Criteria screen, and click Enable or Disable.

Delete a regular expression

Select one or more regular expressions on the Comparison Criteria screen, and click Delete.

Regular expression examples

The following are examples of regular expression patterns that you could use to exclude config lines or blocks from comparison.

<table>
<thead>
<tr>
<th>REGEX PATTERN</th>
<th>END REGEX PATTERN</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>^! Last</td>
<td></td>
<td>Ignores the !Last Configuration change line in Cisco configurations.</td>
</tr>
<tr>
<td>^ntp clock-period</td>
<td></td>
<td>Ignores the ntp clock-period line in Cisco configurations.</td>
</tr>
<tr>
<td>^wlccp ap username</td>
<td></td>
<td>Ignores the wlccp line in Cisco access point configurations.</td>
</tr>
<tr>
<td></td>
<td>cisco</td>
<td></td>
</tr>
<tr>
<td>certificate ca</td>
<td>quit</td>
<td>Ignores all lines in a certificate block that starts with certificate ca and ends with quit.</td>
</tr>
</tbody>
</table>

Execute a script on a node

This resource is available to users with the WebUploader, Engineer, or Administrator role.

The script you execute must be saved as a text file that can be uploaded from the client computer. For more information about creating command scripts, see Create an executable command script.

1. Click My Dashboards > Configs > Configuration Management.

   If you do not have administrator privileges you may not be able to see some nodes. Your account must at least have the WebDownloader role to download configs, and the WebUploader role to execute a script against nodes.
2. Select one or more nodes.
3. Click Execute Script.
4. Select a script to execute or click Load Script from File, and locate it in the file system.
   You can specify a delay in seconds inside a script. A delay is the time SolarWinds NCM waits before sending the next command. The following is a sample script that includes a delay:
   ```
   {Command 1}
   ${Delay:20}
   {Command 2}
   ```
   This feature is useful, for example, when uploading a flash image. Some time is required for the formatting of the flash to complete before then performing the image upload.
5. Select script options at the bottom if necessary.
6. Click Execute.

**View the results**

1. Click My Dashboards > Configs > Configuration Management.
2. Click Transfer Status.
3. In the Action column, locate the most recent entry labeled Execute Script.
4. Click Show Script Results in the Status/Details column.

If execution fails, see [Troubleshoot script execution](#). Alternatively, try executing the script by creating a job.

**Upload a config**

This resource allows you to upload a configuration file you have previously downloaded from this node to SolarWinds NCM.

This resource helps you easily correct unauthorized or incorrect changes made to a device configuration. You can also write the uploaded configuration to NVRAM, essentially making it the startup configuration for the device.

In the case of a multi-node upload/download operation, the Simultaneous Downloads/Uploads setting can be used as a throttle. By default it runs 25 sessions simultaneously.

For IPv6, you can rediscover devices that were previously discovered with the engine using IPv4. You can do inventories for devices already discovered with IPv4 or rediscovered with IPv6. Otherwise, new IPv6 addresses can be added to SolarWinds NCM, though IPv6 addresses cannot be communicated with through SNMP. You can execute scripts, upload, and download configuration files on IPv6 addresses. Telnet and SSH communications are supported.

- If you selected a binary configuration for upload, editing of the configuration is not an option from the Upload Config screen; instead you will see the Binary Config notation in the editing pane.
Upload a config to one or more nodes

1. Click My Dashboards > Configs > Configuration Management.
   If you do not have administrator privileges you may not be able to see some nodes. Your account must be a member of the WebUploader group to upload configs.

2. Select one or more nodes, and click Upload Config.

3. Expand the tree to locate the correct node.

4. Select the config you want to upload.

5. Click Advanced and select options, if desired.

6. Click Upload.

Import config files

You can import configuration files that you have already downloaded from your devices into SolarWinds NCM. NCM imports Configuration files that use the following file formats:

- SolarWinds NCM Archive: .Config
- SolarWinds Cisco Config Downloader: .CiscoConfig
- Text File: .txt
- Configuration File: .cfg
- Any file in ASCII text

Import a config file

1. Click My Dashboards > Configs > Configuration Management.

2. Select a node.

3. Under Nodes Details, click Configs.

4. Under Config List, click Import Config.

5. Click Choose File, and browse to the config file.

6. Click Submit.

Establish baselines as a comparison point for config changes

What is a baseline?
A baseline is a configuration file that is known to be good for a particular application. It is important to establish a baseline when making node configuration changes.

Use a baseline to compare future downloads against.
Set a baseline

1. Click My Dashboards > Configs > Configuration Management.
2. Click a node.
3. Click Configs on the left.
4. Under Config List, select the config, and click Set/Clear Baseline.

If you do not have administrator privileges you may not be able to see some nodes. Your account must have the WebUploader role to upload configs.

Resetting a baseline does not regenerate cache data, so there may be a delay before the chart displays the most current data. Downloading a config resets cache data, however.

When downloading new configuration files, select Compare to Last Baseline Config in the Download Config window to automatically compare the new configuration file to the baseline. If no baseline is found, the configuration is compared against the previously downloaded configuration file.

Remove a config file as a baseline

1. Click My Dashboards > Configs > Configuration Management.
2. Click a node.
3. Click Configs on the left.
4. Under Config List, select the current baseline.
5. Click Set/Clear Baseline.

If you do not have administrator privileges you may not be able to see some nodes. Your account must have the WebUploader role to upload configs.

Baseline your entire network

A full inventory scan can take anywhere from a few minutes to several hours to complete. The time period varies based on the number of nodes and the type of statistics you want to collect. For more information on how to establish what statistics are collected, see Adjust inventory settings.

1. Click My Dashboards > Configs > Jobs.
2. Click Create New Job.
3. Name the job and select Baseline Entire Network from Job Type.
4. Select the Schedule Type. If you are creating an Advanced schedule, use the five fields to create a CRON expression.
5. Add a comment if this job relates to a business rule, and click Next.
6. On the Choose Nodes tab, click Next.
7. Select an email notification option, and click Next. If you click Email Results, the default email notification and SMTP server settings are populated. These settings can be overridden in each job.
8. Select one of the following options:
Set the Network Baseline to the Last Config Downloaded from Each Node
Set the Network Baseline to a Specific Date

9. Click Next.
10. Review the settings for the job, and click Finish.

Clear all baselines to reset the comparison point for config changes

1. Click My Dashboards > Configs > Jobs.
2. Click Create New Job.
3. Name the job, and select Baseline Entire Network from Job Type.
4. Select the Schedule Type. If you are creating an Advanced schedule, use the five fields to create a CRON expression.
5. Add a comment if this job relates to a business rule, and click Next.
6. On the Choose Nodes tab, click Next.
7. Select an email notification option, and click Next. If you click Email Results, the default email notification and SMTP server settings are populated. These settings can be overridden in each job.
8. Click Clear All Baselines, and click Next.
9. Review the settings for the job, and click Finish.

Upload configuration changes

For a multi-node upload or download operation, the Simultaneous Downloads/Uploads setting can be used as a throttle. By default it is set to run 25 Sessions simultaneously. To change this setting, see Configuration files.

For IPv6, you can rediscover devices that were previously discovered with the engine using IPv4, and you can do inventories for devices already discovered with IPv4 or rediscovered with IPv6. Otherwise, new IPv6 addresses can be added to SolarWinds NCM, though IPv6 addresses cannot be communicated with through SNMP. You can execute scripts, upload, and download configuration files on IPv6 addresses, and Telnet and SSH communication is supported.

You can upload changes to a custom config type only to a single device, but through an indirect transfer protocol, such as Telnet or TFTP. As a result, the Write Config to NVRAM After Upload option is disabled.

Upload an entire configuration

1. Click My Dashboards > Configs > Configuration Management.
2. Select one or more nodes.
3. Click Upload.
4. Select a config, and edit it if necessary.
5. Click Advanced at the bottom.
6. Select one or more advanced options, and then click Upload.

Create a config archive for local or network storage

Create a config archive for local storage

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
3. Under Configs, click Configs.
4. Under Config Archive, select Save a Copy of Each Config to the Config Archive Directory, As It Is Downloaded.
5. If you are low on storage space and only need the last and current configs, select When Configs Are Edited, Only Retain the Last Version.
6. Type the path of the local directory to store the NCM Config Archive.
   By default, SolarWinds NCM sets the directory for the config archive as %PROGRAMDATA%\SolarWinds\NCM\Config-Archive. As preparation for a growing archive, SolarWinds recommends moving the config archive to a different location.
7. Type the template you want to use when naming the configuration files. For more information, see Configuration archive variables.
8. Click Submit.

Create a config archive for network storage

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
5. Enter credentials, and click Validate Credentials.
   If credentials fail to validate, check that the NCM-related account has Windows permissions for the network storage.
6. Click Settings > All Settings.
7. Under Product Specific Settings, click NCM Settings.
8. Under Configs, click Configs.
9. Under Config Archive, select Save a Copy of Each Config to the Config Archive Directory, As It Is Downloaded.
10. If you are low on storage space and only need the last and current configs, select When Configs Are Edited, Only Retain the Last Version.
11. Type the path of the network storage for the config archive.
12. Type the template you want to use when naming the configuration files. For more information, see Configuration archive variables.
13. Click Submit.

Real time change detection

The Real Time Change Detection feature provides notification through email whenever a change to any of your device configurations occurs. Unlike the Config Change Report, changes are detected only on the same configuration type. For example, if you download a startup configuration, make changes to it, and upload it as a running configuration, the change will be detected against the previous running configuration. A comparison is not made between running and startup configuration types.

- Real time change detection requirements
- Configure real time change detection
- Limit real time notification download operations
- Run syslog and trap services as administrator

Real time change detection requirements

- A Windows user account with administrative rights.
- Network devices configured to send Syslog or SNMP Trap messages when configurations change.
- The SolarWinds Syslog Service account must have read-write access to the Orion platform database. For example, if your SQL Server resides on the same server as SolarWinds NCM, consider using a local administrator account for the SolarWinds Syslog Service.
- The SolarWinds Trap Service account must have read-write access to the Orion platform database. For example, if your SQL Server resides on the same server as SolarWinds NCM, consider using a local administrator account for the SolarWinds Trap Service.
- The SolarWinds Syslog and Trap Services must be configured to run as administrator so that their scheduled jobs are processed correctly. For detailed steps, see Run syslog and trap services as administrator.
- Ensure the SNMP Trap Service is running. If the SNMP Trap Service is not listed as a running service in the service control manager (services.msc), you can enable Simple Network Management Protocol in the Management and Monitoring Tools through Add/Remove Windows Components in the Add/Remove Programs application.

Configure real time change detection

The SolarWinds Syslog and Trap Services must be configured to run as administrator so that scheduled jobs are processed correctly. For detailed steps, see Run syslog and trap services as administrator.
Cisco devices send trap messages when a user enters config mode, but not when the user exits. As a result, if you make changes to the configuration on your device, you will receive a trap message about those changes only when you enter config mode the next time. That is, usually not until another change to the configuration needs to be done. Due to this behavior, SolarWinds recommends that you use the syslog option for setting up real-time change detection.

Access real-time change detection settings

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
3. Under Real Time Change Detection, click Configure Real Time Change Detection. You must complete all six steps for Real Time Change Detection (RTCD) to operate correctly.

Step 1: Manually configure your devices to send syslog or trap messages

1. Click My Dashboards > Configs > Configuration Management.
2. Select the node(s), and then click Execute Script.
3. Paste in the commands from the following examples, changing the IP address to match your device:

   **Syslog (IOS)**
   ```
   config terminal
   logging 10.199.3.43
   logging trap 6
   end
   ```

   **Syslog (CatOS)**
   ```
   set logging server 192.168.0.30
   set logging server facility local4
   set logging server severity 4
   set logging server enable
   ```

   **Traps (IOS)**
   ```
   snmp-server host 10.110.68.33 public config
   snmp-server enable traps config
   ```

   **Traps (CatOS)**
   ```
   set snmp trap 10.110.68.33 public config
   set snmp trap enable config
   ```

4. Click Execute.
5. Click Transfer Status.
6. In the Action column, locate the most recent entry labeled Execute Script.
7. Click Show Script Results in the Status/Details column.

For more information, refer to the documentation for each network device.

Remove device configurations by running a command with `no` in front of it. For example, `no set logging server ip_address` removes that target from the remote logging stream.
Step 2: Configure alerts and filters triggered by syslog and trap messages

Cisco devices that send change notifications using syslog messages

1. Start Syslog Viewer in the SolarWinds Orion program folder.
2. Click View > Alerts/Filter Rules.
4. Click OK.

Non-Cisco devices that send change notifications using syslog messages

1. Start Syslog Viewer in the SolarWinds Orion program folder.
2. Click View > Alerts/Filter Rules.
3. Click Add New Rule.
4. Provide the information under General and DNS Hostname.
5. Click Message.
6. In the Message Type Pattern field, type the pattern to find.
   The message pattern varies by device type. For example, when a change is made to a Cisco router, a syslog message containing SYS-5-CONFIG_I: is sent. For more information about what messages are sent, see the documentation of your device.
7. Click Alert Actions, and click Add New Action.
8. Select Execute an External Program, and click OK.
9. Type the following in the Program to Execute field:

   Path\Orion\SolarWinds.NCM.RTNForwarder.exe ${IP},RealtimeNotification,${DateTime},${Message}

   Where:
   Path
   The location of the Orion folder. If the path contains spaces, enclose the path section of the statement in quotation marks: "Path to executable".

   ${IP}
   The IP address of the triggering device.

   RealtimeNotification
   This text is displayed as the user name value. Currently, there is no means to parse the message text for the user name. The text is required to include the Message variable.

   ${DateTime}
   The current date and time. This is equivalent to the Windows Control Panel defined Short Date and Short Time format.

   ${Message}
The Syslog message in the real time detection notification. If your Syslog message contains the user making the change, the user name is included through the use of this variable.

1. You must include the commas and, if including Message, you must include placeholder text in the second comma delimited location and the DateTime variable.

10. Click OK.
11. Ensure the new rule is selected in the Alerts/Filter Rules tab of the Syslog Server Settings window, and click OK.

Devices that send change notifications using SolarWinds Kiwi Syslog Server

1. Start the Kiwi Syslog Server Console in the SolarWinds Syslog Server Console program folder.
2. Click File > Setup.
3. Click Filter, and right-click New Filter to rename it.
4. Select Field > Message Text and Filter Type > Simple, and type the message to include with a syslog notification.
5. Right-click Actions, and rename New Action.
6. In the Program File Name field, type Path\Orion\SolarWinds.NCM.RTNForwarder.exe.
   Where:
   Path
   The location of the Orion folder. If the path contains spaces, enclose the path section of the statement in quotation marks: "Path to executable".
7. Add the string %MsgIPAddr,RTN,%MsgText to Command Line Options
8. Click Apply/OK.
9. Ensure the correct filter and action are selected in Rules lists, and click OK.

Devices that send change notifications using SNMP trap messages

1. Start the Trap Viewer in the SolarWinds Orion program folder.

SolarWinds does not include a predefined rule with filters for trap messages since we strongly recommend using the syslog option instead. However, if you want to use trap messages for Real Time Change Detection, continue with these steps.
2. Click View > Alerts/Filter Rules.
3. Click Add Rule.
4. Provide information on the General and DNS Hostname tabs.
5. Click Conditions, and click Add a Condition.
6. Click SNMPv2-MIB:snmpTrapOID, and then browse to the MIB that contains the trap message.
   For example, browse to CISCO-CONFIG-MAN-MIB:ccmHistoryEventConfigDestination (1.3.6.1.4.1.9.9.43.1.1.6.1.5).
7. Click the asterisk, and type the message pattern to match. 
   For example, when a change is made to the running config the HistoryEventMedium is 3. Changes to 
   the startup config are designated by the integer 4.
8. If you need to match on more than one condition, click Browse next to your last condition, and then 
   click the correct conjunction: And or Or.
   Repeat Steps f through g for as many conditions as you need to match. For example, along with the 
   change history event value, consider matching the command source CISCO-CONFIG_MAN_ 
   MIB:ccmHistoryEventCommandSource (1.3.6.1.4.1.9.9.43.1.1.6.1.3) and select 1 (command line) or 2 
   (snmp) as the value. For more information about what messages are sent from your devices, see the 
   documentation of your device.
9. Click Alert Actions.
10. Click Add Action.
11. Select Execute an External Program, and click OK.
12. Type the following in the Program to execute field:
    "Path\Orion\SolarWinds.NCM.RTNforwarder.exe" ${IP}.
    Where:
    Path
    The location of the Orion folder. If the path contains spaces, enclose the path section of the 
    statement in quotation marks: "Path to executable".
    ${IP}
    The IP address of the triggering device.
13. Click OK.
14. Ensure the new rule is selected in the Alerts/Filter Rules tab of the Trap Server Settings window, and 
    click OK.

Devices that send change notifications to a system other than SolarWinds NCM

1. Start your third-party Syslog or SNMP Trap receiver.
2. Set up an alert that executes an external program.
3. Type the following in the Program to execute field:
   "Path\Orion\SolarWinds.NCM.RTNforwarder.exe" ${IP}.
   Where:
   Path
   The location of the Orion folder. If the path contains spaces, enclose the path section of the 
   statement in quotation marks: "Path to executable".
   ${IP}
   The IP address of the triggering device.
4. Save the alert, and make sure it is enabled.
Step 3: Config changes

1. Click Config Changes.
2. Select Enable These Account Credentials to Access All NCM-managed Devices if you want to allow them to access all network devices managed in NCM.

If the check box is disabled, then Device Login and User Account Credentials is set to Global – Device Level on the Security resource, located at Settings > NCM Settings > Security. Click Security to change that setting if necessary.

3. Select Include Syslog/Trap Message in NCM Email Notification if desired.
4. Click Submit.

Step 4: Config downloads and notifications

1. Click Config Downloads and Notification Settings.
2. Under Previously Downloaded Config File, make a selection from Monitor This File Type.
3. Under Baseline Config File, select the config file type against which you want to compare differences with the file downloaded as part of the RTCD operation.
4. Select the relevant Email Notification Options
5. Enter the Sender Name, Subject, and To address information to be used in sending out RTCD email notifications. Reply Address is optional.
6. Click Submit.

Step 5: Enter NCM SMTP server details

The email server settings you enter here will be used to send notifications regarding RTCD, config change approvals, and running jobs. For information on config change approvals, see Approval system for device configuration changes.

1. Click NCM SMTP Server.
2. Enter the fully qualified domain name (FQDN) or IP address of the mail server.
3. Enter the port number on which the mail server handles messages.
4. Select None or Password as the Authentication method.
5. Enter a user name and password.
6. Click Submit.

Step 6: Enable real time config change notifications

1. Click Enable.
2. Click Submit.
Limit real time notification download operations

The config downloads may overwhelm resources on your SolarWinds NCM server, in response to syslog and trap information flowing in from your network devices.

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
3. Under Configs, click Configs.
4. Adjust the number of Simultaneous Download/Uploads. The default number of concurrent sessions is 10.
5. Click Submit.

Run syslog and trap services as administrator

Although the SolarWinds Syslog Service and Trap Service are installed and launched with administrator access, you must manually grant them administrator access to run correctly.

1. Navigate to the SolarWinds installation folder. The default is C:\Program Files\SolarWinds\Orion\.
2. Right-click SyslogService.exe and click Run As.
3. Click The Following User, and select Administrator.
4. Click OK.
5. Right-click SWTrapService.exe and click Run As.
6. Click The Following User, and select Administrator.
7. Click OK.

Search config files or node properties

Use the Advanced Search function to search for a text string within configurations and node properties. Use this functionality to find a specific node or config, or all nodes or configs that share a common characteristic. For example, you can search for all configs that have TACACS security enabled, find an IP address that has been assigned to an interface, or view which configs are managed by a vendor.

Complete the following procedure to search for specific strings of text in the properties of managed nodes or in configuration files stored in the Orion platform database.

- When a config is initially added to the database, a search might not find it for up to 10 minutes.

1. Click My Dashboards > Configs > Config Summary.
2. Under Search NCM, click Advanced Search.
3. Enter the search string.
4. Select a search target from the drop-down menu, and then specify which node properties or config files to search:

<table>
<thead>
<tr>
<th>Target</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>Searches all properties on all nodes, and the specified config files on all nodes</td>
</tr>
<tr>
<td>Nodes</td>
<td>Searches the specified properties on all nodes</td>
</tr>
<tr>
<td>Configs from All Nodes</td>
<td>Searches the specified configs on all nodes</td>
</tr>
<tr>
<td>Configs from Selected Nodes</td>
<td>Searches the specified config files on the selected nodes</td>
</tr>
</tbody>
</table>

5. Click Search.

6. To search within the returned results, select Search in Results, type a new string pattern, and click Search again.

7. Click a link in the search results to open the node or config details view.

**Delete config files from the database**

1. Click My Dashboards > Configs > Configuration Management.
2. Use the Group By list to find the node, and click it.
3. Under the node list in the center, click + to expand the list of configs under the targeted node.
4. Under the Suggested Action column, click Delete beside the relevant config.

**Automate config file purges**

If you do not need to keep historical configuration files and want to improve database performance, you can automate the removal of unnecessary configuration files. If your database is not stored on a high performance SQL Server, or is running on a locally installed instance of SQL Server Express, ensure you regularly purge unused config history.

1. Click My Dashboards > Configs > Jobs.
2. Click Create New Job.
3. Name the job, and then select Purge Old Configs from Database from Job Type.
4. Select the Schedule Type. If you are creating an Advanced schedule, use the five fields to create a CRON expression.
5. Add a comment if this job relates to a business rule, and click Next.
6. Select the nodes to target with this job, and click Next.
7. Select a Save option.
10. Select an email notification option, and click Next. If you click Email Results, the default email notification and SMTP server settings are populated. These settings can be overridden in each job.

11. Select a Purge Configs option, and then click Next.

12. Review the settings for the job, and click Finish.
Device templates

Device templates contain device-specific commands that enable SolarWinds NCM to perform actions on a type of device. For example, a device template could enable NCM to execute scripts and download configuration files from Cisco Pix Firewall 535 devices.

NCM provides a set of device templates to perform actions on common device types. You cannot modify the default device templates, but you can create a custom device template based on a default template.

Create custom device templates to add support for additional device types. To prepare for creating a custom device template, follow these suggestions.

**Before you begin:**

- First, gather the information you will need for the device template.
- For additional guidance, review the "Best practices for device command templates".
- Review the Communication Process diagrams.

> If using the F5 templates to manage BIG-IP F5 device configurations, see the "F5 device template behavior" in the reference section for important details.

When you are ready, you can [create a device template](#) using either the wizard or the XML editor.

### Gather the information you need

Before attempting to modify or create a new template, answer all of the following questions about your device:

<table>
<thead>
<tr>
<th>QUESTION</th>
<th>NOTE</th>
<th>EXAMPLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>What are the Machine Type and System OID values displayed in the Device Details tab of the node properties?</td>
<td>Use this information to save the device template with a unique name that SolarWinds NCM recognizes, ensuring its use when connecting to the device.</td>
<td></td>
</tr>
<tr>
<td>What command is used to disable pagination?</td>
<td>This command is the value used in the template <code>RESET</code> command.</td>
<td><code>terminal pager 0</code></td>
</tr>
<tr>
<td>What command is used to reboot the device?</td>
<td>This command is the value used in the template <code>Reboot</code> command.</td>
<td><code>reload noconfirm</code></td>
</tr>
<tr>
<td><strong>QUESTION</strong></td>
<td><strong>NOTE</strong></td>
<td><strong>EXAMPLE</strong></td>
</tr>
<tr>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
<td>---------------------------------------------------------</td>
</tr>
<tr>
<td>What command is used to enter configuration mode?</td>
<td>This command is the value used in the template EnterConfigMode command.</td>
<td><code>config terminal</code></td>
</tr>
<tr>
<td>What command is used to exit configuration mode?</td>
<td>This command is the value used in the template ExitConfigMode command.</td>
<td><code>quit</code></td>
</tr>
<tr>
<td>What command is used to specify the startup configuration?</td>
<td>This command is the value used in the template Startup command.</td>
<td><code>startup</code></td>
</tr>
<tr>
<td>What command is used to specify the running configuration?</td>
<td>This command is the value used in the template Running command.</td>
<td><code>running</code></td>
</tr>
<tr>
<td>What command sequence is used to directly download the configuration using Telnet or SSH?</td>
<td>This command sequence is the value used in the template DownloadConfig command*.</td>
<td><code>show ${ConfigType}</code></td>
</tr>
<tr>
<td>What command sequence is used to upload the configuration using Telnet or SSH?</td>
<td>This command sequence is the value used in the template UploadConfig command*.</td>
<td><code>${EnterConfigMode}${CRLF}${ConfigText}${CRLF}${ExitConfigMode}</code></td>
</tr>
<tr>
<td>What command sequence is used to download the configuration using SNMP, that is, indirect transfer?</td>
<td>This command sequence is the value used in the template DownloadConfigIndirect command*.</td>
<td><code>copy ${TransferProtocol}://${StorageAddress}/${StorageFilename}${ConfigType}${CRLF}${CRLF}</code></td>
</tr>
<tr>
<td>What command sequence is used to erase the configuration?</td>
<td>This command sequence is used in the template EraseConfig command.</td>
<td><code>write erase ${CRLF}Yes</code></td>
</tr>
<tr>
<td>QUESTION</td>
<td>NOTE</td>
<td>EXAMPLE</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>What command sequence is used to commit a configuration to memory?</td>
<td>This command sequence is used in the template <code>SaveConfig</code> command.</td>
<td><code>write memory</code></td>
</tr>
<tr>
<td>What command sequence is used to show the version information?</td>
<td>This command sequence is used in the template <code>Version</code> command.</td>
<td><code>show version</code></td>
</tr>
</tbody>
</table>

*For more information about variables that can be used in command scripts and device command templates, see [Pre-command and command template variables](#).*

### Best practices for device command templates

Review the following best practices before modifying device command templates.

- Review several device templates and familiarize yourself with the command syntax before creating a new template.
- Write down a list of all the commands you need to include in the new device template, including whether or not you have to press `Enter` after you type the command to ensure the device recognizes the command.
- Telnet to your device to find the pre-commands you need. A pre-command can be used for any device which requires input before prompting for credentials. A pre-command is used before logging in. For example, when you connect to a router and before you are asked for a password, you must press `Enter` to wake up the connection. Add the following line to the template:
  ```xml
  <Command Name="PreCommand" Value="\r\n"/>
  ```
- Create a new device template by modifying an existing device template.
- Before modifying a device template, make a copy of the original.
- If you have a device that indicates enable mode with any character other than the `#` character, add the following line to the template: `<Command Name="EnableIdentifier" Value="*"/>`, where `*` is the character used to indicate the enable privilege level.
- Ensure that you do not have two command templates with the same System OID.
- If the value for the Command Device Template field within the Node Details view is set to Auto Determine, SolarWinds NCM chooses the command template with the System OID value that is closest to the system OID of the device. For example, if the System OID for the device is `1.3.6.1.9.25.5.4`, then SolarWinds NCM starts the search for a template that includes `1.3.6.1.9.25.5.4` as the System OID. If no template is found, SolarWinds NCM looks for a template with `1.3.6.1.9.25.5`, and then `1.3.6.1.9.25`, and so on. To be safe, use the full System OID when building templates.
- To declare the ready prompt for your device, use the `VirtualPrompt` command to designate the

---

*For more information about variables that can be used in command scripts and device command templates, see [Pre-command and command template variables](#).*

---

*To declare the ready prompt for your device, use the `VirtualPrompt` command to designate the*
prompt: <Command Name="VirtualPrompt" Value="unc-dsf%"/>

where unc-dsf% is the prompt used by the device to designate it is ready for commands to be sent. You can use the Virtual Prompt to avoid an issue with special characters in banners. For example, to avoid SolarWinds NCM recognizing the # character as an enable prompt. Ensure you use the MenuBased command when using the VirtualPrompt command: <Command Name="MenuBased" Value="false"/> or <Command Name="MenuBased" Value="true"/>.

- Devices, such as VPN concentrators, may require a null value for the Reset command to function properly. If you receive an Out of Range error, change the value of the Reset command from 0 to blank (). For example, <Command Name="RESET" Value=""/>.

Not all commands are supported on all devices.
Communication process diagrams
Create device templates in SolarWinds NCM

Use either of the following methods to create or edit device templates in SolarWinds NCM:

- Use the interactive wizard to guide you through the options
- Use the XML editor
Create a device template using the interactive wizard

1. Click Settings > All Settings.
2. Under Node & Group Management, click Manage Device Templates.
4. Use the Level and Group By lists to organize the node list.
5. Select a node, and click Next. SolarWinds NCM pulls the OID value from the properties of the node you select.
6. Choose the scope of operation for this device template, and click Next.
7. Verify and adjust Connection Profile and Device Template settings.
8. Click Test, and click Next.
10. Set the config type to use in the download test, and click Perform Download Test.
11. Click Next.
12. Enter and adjust the Device Template and Assign Device Template settings, and click Finish.

Create a device template with the XML editor

1. Click Settings > All Settings.
2. Under Node & Group Management, click Manage Device Templates.
3. Click Add New > Using XML Editor.
4. Add information to the template, and modify the XML according to your device.
5. Click Save.

Edit a device template

If you need to modify an existing SolarWinds NCM device template, you can edit it using the interactive wizard or the XML editor.

Edit a device template with the interactive wizard

1. Click Settings > All Settings.
2. Under Node & Group Management, click Manage Device Templates.
3. Select a device template, click Copy, name the copy, and click Save.

<i>You must copy and rename SolarWinds-provided templates because they are not editable. SolarWinds recommends copying and renaming any template before making changes because you might have applied the existing template to nodes.</i>

4. Select the copied template, and click Edit > Using Interactive Wizard.
5. Use the Level and Group By lists to organize the node list.

6. Select a node, and then click Next.

   SolarWinds NCM pulls the OID value from the properties of the node you select.

7. Choose the scope of operation for this device template, and click Next.

8. Verify and adjust Connection Profile and Device Template settings.

9. Click Test, and click Next.

10. Verify and adjust Download Command and Config Types settings.

11. Set the config type to use in the download test, and click Perform Download Test.

12. Click Next.

13. Enter and adjust the Device Template and Assign Device Template settings, and click Finish.

Edit a device template with the XML editor

1. Click Settings > All Settings.

2. Under Node & Group Management, click Manage Device Templates.

3. Select a device template, click Copy, name the copy, and click Save.

   You must copy and rename SolarWinds-provided templates because they are not editable. SolarWinds recommends copying and renaming any template before making changes because you might have applied the existing template to nodes.

4. Select the copied template, and click Edit > Using XML Editor.

5. If you want to modify an existing command, make changes as needed.

   For example, a device shows version information when you type `show sys info`. The current device command template includes:

   ```xml
   <Command Name="Version" Value="show version"/>
   ```

   The value needs to be changed to `show sys info`.

   The updated command is:

   ```xml
   <Command Name="Version" Value="show sys info"/>
   ```

   You can set up cascading templates by creating a series that targets slightly different OIDs. For example:

   1.3.6.1.4.1.9 = Cisco (All)

   1.3.6.1.4.1.9.1.23 = Cisco 2507

   You can specify a specific device with a more exact OID, and SolarWinds NCM will try to find the closest match. In this example, if SolarWinds NCM is communicating with a device with a system OID of 1.3.6.1.4.1.9.1.25, it uses the Cisco (All) template, but if the system OID is 1.3.6.1.4.1.9.1.23, it uses the Cisco 2507 template.

6. Click Save.
Confirm the device template is working

1. Click My Dashboards > Configs > Config Summary.
2. Under NCM Node List, click a node to which you assigned the device template.
3. On the node details page, click Configs on the left.
4. Under Download Config, select a config type.
5. Click Download.
6. If the downloads succeeds, the device template is working as expected.

Sample CLI device command templates

Two examples of device command templates for CLI devices are provided below:

- **Cisco IOS example**
- **Nortel BayStack 380 example**

For a list of commands and their descriptions, see [Command template commands](#).

**Cisco IOS example**

**File Name**

Cisco IOS-1.3.6.1.4.1.9.ConfigMgmt-Commands

**Contents**

```xml
<Configuration-Management Device="Cisco Devices" SystemOID="1.3.6.1.4.1.9">
  <Commands>
    <Command Name="RESET" Value="terminal width 0${CRLF}terminal length 0"/>
    <Command Name="Reboot" Value="reload${CRLF}y${CRLF}"/>
    <Command Name="EnterConfigMode" Value="config terminal"/>
    <Command Name="ExitConfigMode" Value="end"/>
    <Command Name="Startup" Value="startup"/>
    <Command Name="Running" Value="running"/>
    <Command Name="DownloadConfig" Value="Show ${ConfigType}"/>
    <Command Name="UploadConfig" Value="${EnterConfigMode}${CRLF}${ConfigText}${CRLF} ${ExitConfigMode}"/>
    <Command Name="DownloadConfigIndirect" Value="copy ${ConfigType}${TransferProtocol}://${StorageAddress}/${StorageFilename}${CRLF}"/>
    <Command Name="UploadConfigIndirect" Value="copy ${TransferProtocol}://${StorageAddress}/${StorageFilename} ${ConfigType}${CRLF}"/>
  </Commands>
</Configuration-Management>
```
<Command Name="EraseConfig" Value="write erase${CRLF}Y"/>
<Command Name="SaveConfig" Value="write memory"/>
<Command Name="Version" Value="show version"/>
</Commands>
</Configuration-Management>

Nortel BayStack 380 example

File Name
Nortel Baystack380-1.3.6.1.4.1.45.3.45.ConfigMgmt-Commands

Contents

<!--SolarWinds Network Management Tools-->  
<!--Copyright 2008 SolarWinds.Net All rights reserved-->  
<Configuration-Management Device="Nortel BayStack 380 Devices">
SystemOID="1.3.6.1.4.1.45.3.45">
<Commands>
  <Command Name="RESET" Value="terminal length 0"/>
  <Command Name="Reboot" Value="reload${CRLF}Yes"/>
  <Command Name="EnterConfigMode" Value="config terminal"/>
  <Command Name="ExitConfigMode" Value="end"/>
  <Command Name="Startup" Value="configuration"/>
  <Command Name="Running" Value="running-config"/>
  <Command Name="DownloadConfig" Value="show ${ConfigType}"/>
  <Command Name="UploadConfig" Value="${EnterConfigMode}${CRLF}${ConfigText}${CRLF}${ExitConfigMode}"/>
  <Command Name="DownloadConfigIndirect" Value="copy ${ConfigType}${TransferProtocol}:${StorageAddress}/${StorageFilename} ${CRLF}"/>
  <Command Name="UploadConfigIndirect" Value="copy ${TransferProtocol}:${StorageAddress}/${StorageFilename} ${ConfigType}${CRLF}"/>
  <Command Name="Version" Value="show sys info"/>
  <Command Name="PreCommand" Value="${CTRL+Y}"/>
</Commands>
</Configuration-Management>

SCP example of config upload and download

File Name
Cisco IOS-1.3.6.1.4.1.9.ConfigMgmt-Commands
Create a menu-based command template

SolarWinds Network Configuration Manager supports the upload and download of configs on menu-based devices that do not have command line interfaces. SolarWinds NCM does not support execution of command scripts on exclusively menu-based devices, however.

All Telnet commands for menu-based devices should be described in the device command template XML file (*.ConfigMgmt-Commands). For more information about file contents, see Command template commands.

On some menu-based devices, such as Cisco SF300 LAN switches, menu item numbers can be used instead of arrow moves. For example, instead of assigning:

Value="${ENTER}${DownArrow}${DownArrow}${DownArrow}${DownArrow}${DownArrow}${DownArrow}"$

You could instead assign:

Value="1${ENTER}7${ENTER}"
Login user name and password have to be sent as pre-command values instead of from the NCM Node Details configuration.

To do this:

1. Clear the user name and password fields for the node in Login Information.
2. Set Enable to <No Enable Login>.
3. Use the following pre-commands:
   
   ```
   <Command Name="PreCommand" Value="username${DownArrow}" />
   <Command Name="PreCommand" Value="password${ENTER}" />
   ```

The following example provides the values declared for menu-driven indirect transfer:

```xml
<Commands>
  <Command Name="RESET" Value=""/>
  <Command Name="Reboot" Value=""/>
  <Command Name="EnterConfigMode" Value=""/>
  <Command Name="ExitConfigMode" Value=""/>
  <Command Name="Startup" Value=""/>
  <Command Name="Running" Value=""/>
  <Command Name="DownloadConfigIndirect" Value="${Downarrow}${Downarrow}${Downarrow}${Downarrow}${Downarrow} ${CRLF} ${StorageFilename} ${CRLF} ${StorageAddress} ${CRLF} ${DownArrow} ${CRLF}" Delay="300" RegEx="written"/>
  <Command Name="UploadConfig" Value=""/>
  <Command Name="EraseConfig" Value=""/>
  <Command Name="SaveConfig" Value=""/>
  <Command Name="Version" Value="${DownArrow}${Down Arrow}${Down Arrow}${Down Arrow}${Down Arrow} ${Down Arrow} ${CRLF}" RegEx="Event Log"/>
  <Command Name="PreCommand" Value="${CTRL+Y}" />
</Commands>
```

To create a menu-based device command template:

1. Manually Telnet to your device to discover the pre-commands you must send before the device presents the login screen. Pre-commands are used for any device which requires input before prompting for credentials. For example, when you connect to a router and before you are asked for password, you must press Enter to wake up the connection. Add the following line to the template:

   ```
   <Command Name="PreCommand" Value="${CRLF}" />
   ```

2. SolarWinds Network Configuration Manager also sends a Version command during the validate login action. To set this command value, complete the following:
   
   a. To determine this command, find the option in the menu which shows device version information. For example, if the System Information menu shows device version information and to access this menu item you press the down arrow key (↓) twice and then press Enter,
type the following line into the device command template:

```
<Command Name="Version" Value="$\{DownArrow\}$\{DownArrow\}$\{CRLF\}">
```

b. Find the string that is received when the command is complete. For example, if the command is complete when the device responds with System Characteristic, then you must add the following attribute to the command:

```
RegEx="System Characteristic"
```

c. Add a delay between keystrokes by adding the following attribute:

```
Delay="300"
```

d. The complete command line for the Version command is now:

```
<Command Name="Version" Value="$\{DownArrow\}$\{DownArrow\}$\{CRLF\}"
RegEx="System Characteristic" Delay="300" />
```

3. Access the configuration file menu, and then download a configuration manually. During this operation, note the keys you press to complete this process. For example, on a Nortel BayStack 552048T you would press the following keys to download a configuration:

a. Down arrow (↓) nine times – Highlights Configuration file menu item

b. Enter – Opens Configuration file menu

c. Enter – Opens file Download/Upload menu

d. ConfigName + Enter – Sets the name of configuration file

e. Down arrow (↓) + TFTP IP Address + Enter – Sets the TFTP server address

f. Down arrow (↓) + Space + Enter – Starts the downloading process

4. Translate all these command into SolarWinds NCM variables. In this example, the following commands are used:

```
$\{Downarrow\}$\{Downarrow\}$\{Downarrow\}$\{Downarrow\}$\{Downarrow\}$\{Downarrow\}$\{CRLF\}$\{CRLF\}$\{StorageFilename\}$\{CRLF\}$\{DownArrow\}$\{StorageAddress\}$\{CRLF\}$\{DownArrow\} $\{CRLF\}$
```

For a list of commands and their descriptions, see [Command template commands](#).

5. Find the string that is received when the command is complete. For example, the command is complete when the device responds with written. In this case, you must add the following attribute to the command: `RegEx="written"`.

6. Add a delay between keystrokes by adding the following attribute: `Delay="300"`. 
7. The complete download command is as follows:

```xml
<Command Name="DownloadConfigIndirect" Value="${Downarrow}${Downarrow}
${Downarrow}${Downarrow}${Downarrow}${Downarrow}${Downarrow}
${CRLF}${CRLF}${StorageFilename}${CRLF}${DownArrow}
${CRLF}${StorageAddress}${CRLF}${DownArrow} ${CRLF}"
Delay="300" RegEx="written"/>
```

## Command template commands

The following commands are used to modify and declare the behavior of device templates in SolarWinds NCM. These commands modify the interaction between SolarWinds NCM and your network devices. Ensure you fully understand what modifications will do before modifying a device in production using these commands.

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CustomPasswordPrompt</td>
<td>Allows you to specify the password prompt issued by a device in case NCM</td>
</tr>
<tr>
<td></td>
<td>is having trouble recognizing the device's prompt (for example, due to</td>
</tr>
<tr>
<td></td>
<td>unsupported characters).</td>
</tr>
<tr>
<td>DownloadConfig</td>
<td>Series of commands used to download a configuration from a device.</td>
</tr>
<tr>
<td>DownloadConfigIndirect</td>
<td>Series of commands used to download a configuration indirectly from a</td>
</tr>
<tr>
<td></td>
<td>device using TFTP.</td>
</tr>
<tr>
<td>DownloadConfigIndirectSCP</td>
<td>Series of commands used to download a configuration indirectly from a</td>
</tr>
<tr>
<td></td>
<td>device using SCP.</td>
</tr>
<tr>
<td>EnableCommand</td>
<td>Allows you to declare a custom enable command for those devices that do</td>
</tr>
<tr>
<td></td>
<td>not use Enable as the command.</td>
</tr>
<tr>
<td>EnableIdentifier</td>
<td>Only used when a device does not return the # character at the end of a</td>
</tr>
<tr>
<td></td>
<td>prompt to indicate enable mode. Declare the value displayed while in</td>
</tr>
<tr>
<td></td>
<td>enable mode for a device.</td>
</tr>
<tr>
<td>EnterCLI</td>
<td>Specifies the commands to send upon receiving the VirtualPrompt</td>
</tr>
<tr>
<td></td>
<td>command to enter the CLI mode of the menu driven device. Use pre-</td>
</tr>
<tr>
<td></td>
<td>command variables to declare the command values. For more information,</td>
</tr>
<tr>
<td></td>
<td>see Pre-command and command template variables.</td>
</tr>
<tr>
<td>EnterConfigMode</td>
<td>Series of commands used to enter the configuration mode of a device.</td>
</tr>
<tr>
<td>ExitConfigMode</td>
<td>Series of commands used to exit the configuration mode of a device.</td>
</tr>
<tr>
<td>IPAddress</td>
<td>The IP address of the server where SolarWinds NCM is installed.</td>
</tr>
<tr>
<td>MenuBased</td>
<td>Specifies whether the device is menu- or CLI-based. If a device is menu-</td>
</tr>
<tr>
<td></td>
<td>based and you can switch it to CLI from the menu, use the VirtualPrompt</td>
</tr>
<tr>
<td></td>
<td>and EnterCLI commands to do so. Valid values are true or false.</td>
</tr>
</tbody>
</table>

ADMINISTRATOR GUIDE: NETWORK CONFIGURATION MANAGER
<table>
<thead>
<tr>
<th><strong>PROPERTY</strong></th>
<th><strong>DESCRIPTION</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>MenuDrivenConfigStart</td>
<td>Allows you to declare a value after which the transmitted data is considered the config, requested from the menu-driven device. For example, in the Cisco VPN Concentrator device template, the declared value is &quot;#####&quot;. The information sent after the ten hash signs is saved as the requested configuration file.</td>
</tr>
<tr>
<td>More</td>
<td>Specifies the More prompt in the rare instance that this prompt is not recognized automatically. Do not specify this command unless you are experiencing issues with paging.</td>
</tr>
<tr>
<td>Precommand</td>
<td>Specifies the device requires a pre-command. For more information, see Pre-command and command template variables.</td>
</tr>
<tr>
<td>Reboot</td>
<td>Series of commands used to reboot the device.</td>
</tr>
<tr>
<td>RebootAt</td>
<td>Series of commands used to reboot a device at a specified time. Use the variables listed in the Configuration archive variables section to assign the date and time.</td>
</tr>
<tr>
<td>RESET</td>
<td>Series of commands used to set the length and pagination of the session.</td>
</tr>
<tr>
<td>Running</td>
<td>Value used to specify a running configuration type.</td>
</tr>
<tr>
<td>SaveConfig</td>
<td>Series of commands used to write the configuration to the devices memory.</td>
</tr>
<tr>
<td>Startup</td>
<td>Value used to specify a startup configuration type.</td>
</tr>
<tr>
<td>UploadConfig</td>
<td>Series of commands used to upload a configuration to a device</td>
</tr>
<tr>
<td>UploadConfigIndirect</td>
<td>Series of commands used to upload a configuration indirectly to a device using TFTP.</td>
</tr>
<tr>
<td>UploadConfigIndirectSCP</td>
<td>Series of commands used to upload a configuration indirectly to a device using SCP.</td>
</tr>
<tr>
<td>Version</td>
<td>Series of commands used to display the software version of the device.</td>
</tr>
<tr>
<td>VirtualEnablePrompt</td>
<td>Allows you to specify a regular expression and search for the defined value in the entirety of a device response. This command is often used with menu-based devices, allowing you to locate a specific phrase returned by the device. When specified, ensure you also declare the MenuBased command as true or false.</td>
</tr>
<tr>
<td>VirtualPrompt</td>
<td>Specifies the command prompt that will be sent when the device is ready for command input. Use this command along with the MenuBased command to specify the exact prompt SolarWinds NCM should wait to receive before sending commands.</td>
</tr>
</tbody>
</table>
**Pre-command and command template variables**

Pre-command variables are used in command scripts and device command templates. The Pre-command variables mimic keyboard strokes that are normally entered in the command interface. For more information on creating command scripts, see [Command scripts](#).

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>${ConfigType}</td>
<td>Value used to insert the type of configuration</td>
</tr>
<tr>
<td>${CR}</td>
<td>Carriage return</td>
</tr>
<tr>
<td>${CRLF} or ${[ENTER]}</td>
<td>Carriage return - linefeed combination</td>
</tr>
<tr>
<td>${CTRL+@}</td>
<td>CTRL + @</td>
</tr>
<tr>
<td>${CTRL+A}</td>
<td>CTRL + A</td>
</tr>
<tr>
<td>${CTRL+B}</td>
<td>CTRL + B</td>
</tr>
<tr>
<td>${CTRL+C}</td>
<td>CTRL + C</td>
</tr>
<tr>
<td>${CTRL+D}</td>
<td>CTRL + D</td>
</tr>
<tr>
<td>${CTRL+E}</td>
<td>CTRL + E</td>
</tr>
<tr>
<td>${CTRL+F}</td>
<td>CTRL + F</td>
</tr>
<tr>
<td>${CTRL+G}</td>
<td>CTRL + G</td>
</tr>
<tr>
<td>${CTRL+H}</td>
<td>CTRL + H</td>
</tr>
<tr>
<td>${CTRL+I}</td>
<td>CTRL + I</td>
</tr>
<tr>
<td>${CTRL+J}</td>
<td>CTRL + J</td>
</tr>
<tr>
<td>${CTRL+K}</td>
<td>CTRL + K</td>
</tr>
<tr>
<td>${CTRL+L}</td>
<td>CTRL + L</td>
</tr>
<tr>
<td>${CTRL+M}</td>
<td>CTRL + M</td>
</tr>
<tr>
<td>${CTRL+N}</td>
<td>CTRL + N</td>
</tr>
<tr>
<td>${CTRL+O}</td>
<td>CTRL + O</td>
</tr>
<tr>
<td>${CTRL+P}</td>
<td>CTRL + P</td>
</tr>
<tr>
<td>${CTRL+Q}</td>
<td>CTRL + Q</td>
</tr>
<tr>
<td>${CTRL+R}</td>
<td>CTRL + R</td>
</tr>
<tr>
<td>${CTRL+S}</td>
<td>CTRL + S</td>
</tr>
<tr>
<td><strong>PROPERTY</strong></td>
<td><strong>DESCRIPTION</strong></td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>${CTRL+T}</td>
<td>CTRL + T</td>
</tr>
<tr>
<td>${CTRL+U}</td>
<td>CTRL + U</td>
</tr>
<tr>
<td>${CTRL+V}</td>
<td>CTRL + V</td>
</tr>
<tr>
<td>${CTRL+W}</td>
<td>CTRL + W</td>
</tr>
<tr>
<td>${CTRL+X}</td>
<td>CTRL + X</td>
</tr>
<tr>
<td>${CTRL+Y}</td>
<td>CTRL + Y</td>
</tr>
<tr>
<td>${CTRL+Z}</td>
<td>CTRL + Z</td>
</tr>
<tr>
<td>${CTRL+[]}</td>
<td>CTRL + [ Equivalent to Esc key press</td>
</tr>
<tr>
<td>${CTRL+}</td>
<td>CTRL + \</td>
</tr>
<tr>
<td>${CTRL+]</td>
<td>CTRL + ]</td>
</tr>
<tr>
<td>${CTRL+CTRL}</td>
<td>CTRL + CTRL</td>
</tr>
<tr>
<td>${CTRL+_}</td>
<td>CTRL + _</td>
</tr>
<tr>
<td>${UPARROW}</td>
<td>Up Arrow</td>
</tr>
<tr>
<td>${DOWNARROW}</td>
<td>Down Arrow</td>
</tr>
<tr>
<td>${RIGHTARROW}</td>
<td>Right Arrow</td>
</tr>
<tr>
<td>${LEFTARROW}</td>
<td>Left Arrow</td>
</tr>
<tr>
<td>${StorageAddress}</td>
<td>Value used to insert the TFTP server IP address or hostname</td>
</tr>
<tr>
<td>${StorageFilename}</td>
<td>Value used to insert the name generated by SolarWinds NCM for the downloaded configuration file</td>
</tr>
<tr>
<td>${TransferProtocol}</td>
<td>Value used to insert the transfer protocol used during indirect transfer</td>
</tr>
</tbody>
</table>

### Pre-command device template entry example

The following line from a device command template specifies the pre-command, the delay, and the text that triggers the pre-command. Delay and trigger text (RegEx) are optional variables.

```
<Command Name="Precommand" Value="${CTRL+Y}" Delay="3" RegEx="password:"/>
```

Device command templates are located in the DeviceTypes folder of your installation folder. The default location is: \Program Files\SolarWinds\Configuration Management\DeviceTypes\.
Preclude pseudoterminal setup with command template variables

If your device does not support pseudoterminal device pairs, you can prevent Telnet from attempting to negotiate pseudoterminal setup by using the following command variable:

```xml
<Command Name="allocatePty" Value="false"/>
```

Specifies that the command script will be run with pseudoterminal mode disabled.

Declare a special command prompt with command template variables

If the command prompt is not > or #, or you need to specify more than one character to designate the command prompt, as in the case of banners using the # character, declare the command prompt using the following command variables:

```xml
<Command Name="MenuBased" Value="false"/>
```

Specifies that the template logic should run in CLI mode

```xml
<Command Name="VirtualPrompt" Value="CustomPrompt%"/>
```

Specifies the exact value of the command prompt designating the device is ready to receive commands.

Switch user context with command template variables

If you log in to a device and must switch user context to execute a command, resulting in a different command prompt, use the following command variables to switch context and recognize the new command prompt:

```xml
<Command Name="MenuBased" Value="false"/>
```

Specifies that the template logic should run in CLI mode

```xml
<Command Name="Reset" Value="appropriateSwitchContextCommands" RegEx="newPrompt"/>
```

Specifies the reset command to switch user context and the new command prompt to expect. Use pre-command variables to designate the switch context commands and specify the entire new prompt in the RegEx value.

Respond to post-login interaction requests with command template variables

If you log in to a device and perform an action and are then prompted for interaction, for example, you receive a Press any key prompt, use the following command variables in the command template to not time out:

```xml
<Command Name="PreCommand" Value="${CTRL+Y}"/>
```

Sent when the device does not respond for three seconds.
<Command Name="PreCommand" Value="{$CTRL+Y}" Delay="3"/>

Sent when the device does not respond for more than three seconds.

<Command Name="PreCommand" Value="{$CTRL+Y}" Delay="3" RegEx="Press Any Key"/>

Sent when the device does not respond for more than three seconds and the last received data was Press Any Key.

Troubleshoot device connections

When you experience problems connecting to a device, you may need to perform a session trace to troubleshoot the issue. A session trace shows all communication sent between SolarWinds NCM and the network device to which you are connecting. The session trace log contains error messages and commands sent that generated the error.

1. Click Settings > All Settings.
2. Under Product Settings, click CLI Settings.
3. Select Enable Session Tracing.
4. Perform the steps to recreate the issue you are troubleshooting.
5. Open the session trace file.
6. Apply the necessary changes to the device command template to resolve the issue. For more information, see Device templates.
7. Turn off Enable Session Tracing when the problem has been resolved.

F5 device template behavior

When using device templates for F5 devices, consider the following behavior.

Device setup prerequisites

- Set the Terminal Access user on the device to Advanced Shell. Do not use Traffic Management Shell (TMSH).
- Verify the device can reach the NCM server for configuration transfers using SCP.
- If you plan to upload binary user configuration set (UCS) files, the upload can take a long time. Increase the NCM timeouts to have a complete upload report.
  1. Click Settings > All Settings.
  2. Under Product Specific Settings, choose CLI settings.
  3. Set Telnet/SSH Connection Timeout: 360 seconds
  4. Telnet/SSH Prompt Timeout: 300 seconds

Template details

If you set up NCM to transfer configurations using SSH:
The template can download the current running config.
You cannot download the startup config.
You cannot upload the running or startup config.

If you set up NCM to transfer configurations using SCP:

- The template can download and upload a text single configuration file (SCF) as a running config.
- The template can download and upload a binary UCS as startup config.
- You cannot upload the running config to startup, or startup to running config.

Downloading or uploading configs using TFTP is not supported.
Command scripts

Several tasks can be automated with command scripts, for example:

- Downloading configuration files
- Uploading configuration files
- Uploading IOS images
- Updating logon banners
- Updating access control lists (ACLs)

With the correct use of variables, a single script can be executed on several different devices, without concern for syntax differences.

Scripts are delivered one line at a time to the target devices.

Create an executable command script

1. Click My Dashboards > Configs > Configuration Management.
2. Click the Script Management tab.
3. Click Add New Script.
4. Add the script information.
5. Click Save.
6. To execute the script, on the Config Management tab select one or more nodes and follow the steps to [Execute a script on a node](#).

Use variables in scripts

Variables are a powerful feature of the SolarWinds NCM scripting engine. Variables always begin with a dollar sign and curly brace (${}) and end with a curly brace (}).

Script variables substitute the right commands based on the device type. For example, the variable ${EnterConfigMode} parses as config terminal when communicating with Cisco IOS devices, but parses as configure when communicating with an HP Procurve Switch.

Troubleshoot script execution

If a script fails to execute on a SolarWinds NCM node, review the log for a possible cause. The log is located at C:\ProgramData\SolarWinds\NCM\Logging\NCMBusinessLayerPlugin.log.

Alternatively, execute the script by using [Jobs](#).
Example variable script

The following script contains commands with variables to remove the public read-only community string:

```
${EnterConfigMode}
no snmp-server community public RO
${ExitConfigMode}
${SaveConfig}
${Reboot}
```

Parsed for Cisco IOS devices:

```
config terminal
no snmp-server community public RO
end
write memory
reload${CRLF}y${CRLF}y
```

Parsed for a Dell PowerConnect Switch:

```
config
no snmp-server community public RO
end
copy running-config startup-config${CRLF}${CRLF}
reload${CRLF}y
```

The `${CRLF}` variable equals a carriage return line feed for all devices.

Define script variables

Script variables are defined in device command templates. Templates are located in the `\Configuration Management\DeviceTypes` folder. Each `.ConfigMgmt-Commands` file contains a System OID that is used to uniquely identify a device. A list of command names, and the corresponding commands to be sent to the device when the command name is called, are also included in the templates. These command names are the variables used when creating a script.

Consider the following line taken from the Cisco IOS device command template:

```
<Command Name="EnterConfigMode" Value="config terminal"/>
```

When a script is run on a Cisco IOS device, the variable `${EnterConfigMode}` parses as `config terminal`. New command names can be added and existing command names can be modified within these files.
Reference variables with variables

The script engine also allows you to reference variables with variables. For example, you can define a complex variable in the device template:

ShowInt = running | include interface

Then define another variable that includes the first:

Reveal = show ${ShowInt}

When you call ${Reveal}, it equals show running | include interface.
Inventory

The inventory engine of SolarWinds Network Configuration Manager compliments the product's configuration management functions. You can perform inventory scans on all of your nodes, node groups, or on a single node. You can also manage the inventory collection process. The collected inventory statistics are displayed in the detail view of each device.

Run an inventory scan

You can run a scan at any time to get the latest statistics on your full inventory of managed devices in NCM. You may also run a single-node scan.

Run a complete inventory scan

A full inventory scan can take anywhere from a few minutes to several hours to complete. The time period varies based on the number of nodes and the type of statistics you want to collect. For more information on how to establish what statistics are collected, see Adjust inventory settings.

1. Click My Dashboards > Confis > Configuration Management.
2. Use the Group By list and select No Grouping.
3. Select all nodes.
4. Click Update Inventory.

To run an inventory scan as a recurrent job, see Baseline your entire network

Run an inventory scan on a single node

1. Click My Dashboards > Confis > Configuration Management.
2. Use Search or the Group By list to find a node.
3. Select the node, and click Update Inventory.

Schedule an inventory scan

1. Click My Dashboards > Confis > Jobs.
2. Click Create New Job.
3. Name the job, and then select Update Inventory from Job Type.
4. Select the Schedule Type. If you are creating an Advanced schedule, use the five fields to create a CRON expression.
5. Add a comment if this job relates to a business rule, and click Next.
6. Select the nodes to target with this job, and click Next.

7. Select an email notification option, and click Next. If you click Email Results, the default email notification and SMTP server settings are populated. These settings can be overridden in each job.

8. Select the information types to include in the Inventory job, and click Next.

9. Review the settings for the job, and click Finish.

Adjust inventory settings

This setting gives you some control over the type of information collected by NCM for the inventory, as well as other settings that help you manage the inventory process. For example, it may be necessary to change the number of devices that NCM can inventory concurrently.

1. Click Settings > All Settings.

2. Under Product Specific Settings, click NCM Settings.

3. Under NCM Node Management, click Node Inventory.

4. Select the information types to collect.

5. Adjust the number of devices that should be inventoried concurrently. The default is five.

6. If you have VLANs extended across network trunks, and you want to inventory the relevant devices, select Extend VLANs inventory.

SolarWinds recommends against this situation due to the slow performance of the inventory process.

7. If the inventory process causes the SolarWinds NCM server to hang or if the process takes too long, select Disable Inventory Lookup.

Determine what caused the problem, for example, inadequate server capacity, before re-enabling Inventory Lookup.

8. Click Submit.

View inventory status

1. Click My Dashboards > Configs > Configuration Management.

2. Click the Inventory Status tab.

3. Click a column to change the sort order.
Ensure compliance to policy rules

Use policy reports to verify that device configurations comply with internal policies and external regulations. Each policy report enforces one or more rules. When the policy report runs, SolarWinds NCM scans the specified configuration files and reports any rule violations. Policy reports can also include remediation scripts to bring the configuration file into compliance.

SolarWinds provides a set of example policy reports that you can modify to meet your needs. You can also create custom policy reports.

How rules, policies, and policy reports work together

Each policy report includes one or more policies, and each policy includes one or more rules.

- A rule defines a condition that must or must not exist. Rules can also contain remediation scripts to be run if the rule is violated.
  
  For example, a rule could specify that devices must have banners that include copyright information. And the rule could contain a script to add the copyright information if it is missing.

- A policy groups related rules, and specifies which nodes and config types the rules apply to.

- A policy report groups related policies. When the report runs, it scans the configs specified in the policies and reports any rule violations.

Common uses for policy reports

Regulatory compliance

Use policy reports to ensure that you are in compliance with federal regulations and other industry standards, including:

- Sarbanes-Oxley Act (SOX)
- Health Insurance Portability and Accountability (HIPAA)
- Computer Inventory of Survey Plans (CISP)
- Payment Card Industry (PCI) data security policies

Compliance with internal standards and policies

Standardization is a vital part of keeping the network running smoothly. Use policy reports to locate device configurations that do not comply with your organization's standards or policies. For example:

- Enforce interface naming guidelines.
- Enforce Quality of Service (QoS) traffic shaping policies.
- Verify that the correct banner is present.
- Change the copyright date when the year changes.

Security

To proactively protect against hackers, malware, and other security threats, use policy reports to ensure that network device configurations comply with your organization's security policies. For example:

- Ensure that the default password has been reset on all devices.
- For SNMP-enabled devices, verify that the default public community string is not present.
- Enforce standards for password length.
- Search access control lists (ACLs) for rules that must or must not be present.

Tasks to create and run a policy report

To create custom policy reports, complete the following tasks:

1. Create the rules that the policy report will enforce.
2. Create policies to group related rules and to define which device configs will be checked for compliance.
3. Create the policy report to group related policies.

You can then run the policy report to find and remediate policy violations. You can also share policy reports between Orion servers or with other SolarWinds users.

Create and manage policy rules

Rules are used to ensure that device configurations comply with policies. Each rule defines a search string that either must or must not be present in a configuration. For example, a rule can specify that configurations cannot include the read-only community string "public."

Each rule applies to a specific set of device configurations, which are selected at the policy level.

To learn how rules, policies, and policy reports work together, see Policy compliance.
Create a rule

Complete the following tasks to create a rule:

1. **Add and identify the rule.**
2. Do one of the following:
   - **Define the search string using basic matching.**
     Use basic matching if you want to search the entire config for one string.
   - **Define search conditions using Advanced Config Search.**
     Use Advanced Config Search if you need to build conditions using multiple strings, or limit the search to a block of the config.
3. **(Optional) Define a remediation script.**
4. **Test and save the rule.**

Add and identify the rule

1. Click My Dashboards > Configs > Compliance.
2. Click Manage Policy Reports.
3. Click the Manage Rules tab.
4. Click Add New Rule.
5. Name the rule, and enter a description.
6. Select the Alert Level that indicates the severity of violating this rule.
   - You can customize alert level descriptions.
7. Select an existing folder to store the rule in, or select New folder and enter a folder name.

Define the search string using basic matching

Use basic string matching to search the entire config for one string. You can enter a simple string or a regular expression.

1. Specify whether the rule triggers an alert when the string **is** found or when the string is **not** found.
2. Specify whether the string should be evaluated as a regular expression or a simple string.
   - A simple string can contain the variables ? and *.
3. Enter the string to search for.

### Notes

- NCM uses the Microsoft .NET RegEx engine to evaluate regular expressions. See [Regular expression pattern matching examples](#) for some examples of use in NCM. For detailed information about the syntax, see Microsoft's Regular Expression Language Quick Reference.
Define search conditions using Advanced Config Search

Choose Advanced Config Search if you want to:

- Define a search string that includes conditions (multiple lines connected by **AND** or **OR** operators).
- Specify a block of the config to search, instead of searching the entire config.

> See [Examples of search string conditions](#) for information about defining conditions.

1. Specify whether the rule triggers an alert when the string **is** found or when the string is **not** found.

   - If the search string includes lines that must be included and lines that must not be included, choose the alerting option that is appropriate for the first line. See *Example 2*.

2. Select Advanced Config Search.

   The Advanced Config Search and Search Config File/Block sections are displayed.

3. In the Advanced Config Search section, define the first search string.

   **Must/Must Not Contain**

   Specify whether the string should be included in the config.

   **String Type**

   Specify whether the string should be evaluated as a regular expression or a simple string.

   - A simple string can contain the variables `?` and `*`.

   **String**

   Enter the string to search for.

   > NCM uses the Microsoft .NET RegEx engine to evaluate regular expressions. See [Regular expression pattern matching examples](#) for some examples of use in NCM. For detailed information about the syntax, see Microsoft's Regular Expression Language Quick Reference.

4. To build conditions into your search:
   a. Click Add Another String.
   b. Enter the string and specify the relationship between the lines. (See *Examples of search string conditions*.)

   **And/Or**

   Select an operator to define the relationships between strings.

   **Parens**

   Optionally, use parentheses to group strings into conditional relationships.

   c. Repeat to add as many conditions as you need to define the rule.

5. To search only part of the config:
   a. Under Search Config File/Block, select Config block.
   b. Enter strings or regular expressions to define the beginning and end of the block.
   c. Specify the string type.
(Optional) Define a remediation script

Each rule can include a script to add or remove lines of a configuration to remediate the policy violation.

1. Enter or load a script that makes the required changes to the config.
   The remediation script must include CLI statements that run on the relevant devices. When executed, the script runs through the default communication protocol, such as Telnet or SSH.
   The script should perform the following tasks in order:
   a. Put the device into configuration mode, if needed.
   b. Issue a series of config commands.
   c. Exit config mode.

2. Select a Remediation Script Type. Your selection determines how NCM executes commands against targeted devices to remediate a policy rule violation.
   - CLI allows NCM to use the commands in the script to change the config.
   - Config Change Template launches the Config Change Template wizard to guide you through executing the script.

3. For a CLI script:
   - If you want NCM to execute the script automatically when a policy violation is found, select Automatically execute this script.
   Otherwise, you can choose to execute the script when you view the policy report that lists the rule violation.
     Be sure to test the script thoroughly before you choose to execute it automatically.
   - If you want NCM to put the device in config mode before executing the script, select Execute Script in Config Mode.

Test and save the rule

Test the rule to verify that it finds policy violations as expected.

If the rule includes multiple lines containing regular expressions, SolarWinds recommends testing each regular expression individually before you test the rule as a whole. You can:

- Use an online RegEx tester to test each expression.
- Include only one of the regular expressions in a rule, and use NCM to test that rule against your configs.

1. Click Test to validate the rule against a device configuration.
2. Select a config to test the rule against.
3. Click Test Rule Against Selected Config.
   If the config violates the rule, the Results panel indicates which search strings violated the rule and whether the violation occurred because the string was found or not found.
If the violation occurred because the string was found, you can click the arrow to display the line number in the config file.

4. Click Select Different Config to test the rule against another config.

SolarWinds recommends testing the rule against at least two configurations: one known to comply with the rule, and another known to violate it.

5. When you have finished testing, click Close.

6. Click Submit to save the rule in the selected folder.

Examples of search string conditions

The following sections show how to define some simple conditions, and how NCM evaluates each example.

Example 1: Config must contain at least one of multiple strings

In this example, a config must contain at least one of the following strings:

Must contain string1
OR must contain string2
OR must contain string3

NCM reports a rule violation if the config does not contain any of the strings.
Example 2: A config must contain one string and must not contain another

In this example, the first string is required, but the second cannot be included.

Must contain string1
AND must not contain string2

NCM reports a rule violation if:

- A config does not contain string1.
- A config contains both string1 and string2.

Example 3: Config must contain at least one string in a group and a string outside the group

This example uses parentheses to group the first two strings. A config must contain at least one of the strings in the group, as well as the third string.

(Must contain string1
OR must contain string2)
AND must contain string3
NCM reports a rule violation if:

- A config contains either string1 or string2, but does not contain string3.
- A config contains string3, but does not contain either string1 or string2.
- A config does not contain any of the strings.

![Advanced Config Search](image)

**Edit a rule**

Edit a rule to reflect policy changes or to refine the search string or remediation script. For example, if policy reports return false positives on violations, edit the rule to improve the search string.

1. Click My Dashboards > Configs > Compliance.
2. Click Manage Policy Reports.
3. Click the Manage Rules tab.
4. Select a rule, and click Edit.
5. Edit values, and click Submit.

**Delete a rule**

1. Click My Dashboards > Configs > Compliance.
2. Click Manage Policy Reports.
3. Click the Manage Rules tab.
4. Select a rule, and click Delete.
5. Click Yes.
Customize alert level descriptions for rules

NCM provides three levels to specify the severity of a rule violation. By default, these levels have the following descriptions.

<table>
<thead>
<tr>
<th>LEVEL</th>
<th>DEFAULT DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Informational</td>
</tr>
<tr>
<td>2</td>
<td>Warning</td>
</tr>
<tr>
<td>3</td>
<td>Critical</td>
</tr>
</tbody>
</table>

You can customize the description of each level.

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
4. In the text field for the level that you are changing, replace the word describing that level.
5. Click Submit.

Learn more

After you define rules, you can create a policy and then create a policy report.

Regular expression pattern matching examples

The following examples illustrate some of the uses of Regular Expression patterns in SolarWinds NCM:

- **snmp-server community public**
  - Finds any line that includes the text `snmp-server community public`. There can be text before and/or after the string on the same line.

- **service tcp-keepalives-in.*\n(.*\n)*.*service tcp-keepalives-out**
  - Finds the first line `service tcp-keepalives-in` and then looks for `service tcp-keepalives-out` on any line after that. The regular expression string `.\n(.*\n)*.*` is used to search any number of lines between strings.

- **access-list 105 deny.*tcp any any eq 139 log**
  - Finds the line with `access-list 105 deny`, followed by any number of characters of any type, followed by `tcp any any eq 139 log` on the same line. The regular expression string `.*` finds any character, and any number of characters on the same line. So, this could be used to find spaces, tabs, numbers, letters, or special characters.

- **ntp clock-period \d***
Finds any line that includes `ntp clock-period` followed by any number. The regular expression string `\d*` will find any number at any length, such as 3, 48, or 2394887.

```
user \x2a
```

Finds any line that includes `user *`. The regular expression string `\x` followed by a hexadecimal value specifies an individual character. In this example, `\x2a` represents the asterisk character, which has a hexadecimal value of 2a.

```
\baccess-list\.*\bjoe\b
```

Finds any line that includes `access-list` and `joe`. The regular expression `\b` specifies a word boundary, so the pattern will not find (for example) `access-lists` or `joel`. The `.` matches any character except line breaks, and the `*` matches 0 or more of the preceding character. Together, the `.*` allow any number of characters between the two strings, but not line breaks.

Create and manage policies

A policy is a collection of one or more rules used to ensure policy compliance. For example, you could create a password policy that includes all of your organization’s password rules. Each policy also identifies which nodes and config types will be checked for compliance to the associated rules.

- You can include the same rule in multiple policies.
- To learn how rules, policies, and policy reports work together, see Policy compliance.

Create a policy

1. Click My Dashboards > Configs > Compliance.
2. Click Manage Policy Reports.
3. Click the Manage Policies tab, and click Add New Policy.
4. Name the policy, and enter a description.
5. Select an existing folder to store the policy in, or select New folder and enter a folder name.
6. Select the nodes that must comply to this policy.
   - Select only nodes that you want to report on.
   - You can use custom properties to identify specific nodes.
7. Select the type of config to search for policy violations.
8. Under All Policy Rules, select the rules to include and click Add. If the rule you need is not listed, you can create a rule.
9. Click Submit.
Edit a policy

1. Click My Dashboards > Configs > Compliance.
2. Click Manage Policy Reports.
3. Click the Manage Policies tab.
4. Select a policy, and click Edit.
5. Edit values, and click Submit.

Delete a policy

1. Click My Dashboards > Configs > Compliance.
2. Click Manage Policy Reports.
3. Click the Manage Policies tab.
4. Select a policy, and click Delete.
5. Click Yes.

Learn more

After you create your policies, you can [create a policy report](#).
Create and manage policy reports

Policy reports provide a way to group policies, either by the devices that they will be executed against or by the type of rules they enforce. When a policy report runs, NCM checks device configurations to determine if they comply with the policies.

- You can include the same policy in multiple policy reports.
- To learn how rules, policies, and policy reports work together, see Policy compliance.

Create a policy report

1. Click My Dashboards > Configs > Compliance.
2. Click Manage Policy Reports.
4. Name the report, and enter a description.
5. Select settings for Save in Folder and Display Settings.
6. Under All Policies, select the policies to include and click Add. If the policy you need is not listed, you can Create a policy.
7. Click Submit.

Edit a policy report

1. Click My Dashboards > Configs > Compliance.
2. Click Manage Policy Reports.
3. Select a report, and click Edit.
4. Edit values, and click Submit.

Delete a policy report

1. Click My Dashboards > Configs > Compliance.
2. Click Manage Policy Reports.
3. Select a report, and click Delete.
4. Click Yes.
Configure DISA STIG compliance reports

SolarWinds NCM provides default rules, policies, and policy reports you can use to determine if your devices comply with DISA standards. The Defense Information Systems Agency Security Technical Implementation Guides (DISA STIG) reports are available in your NCM installation. To run the reports that specify the policies for your devices, make sure that you have first defined the node "Device_Type" in custom properties for your device type.

For example, for Cisco devices, specify the Device_Type and OS in custom properties with the following options:

**Device Types:**

PRTR, PL3S, IRTR, IL3S, L2SW, FW

**OS types supported:**

IOS, NX, XE, ASA

Likewise, for F5 devices, specify the Device_Type in custom properties.

Create a DISA STIG dashboard

You can set up a dashboard specifically to monitor compliance. To create a DISA STIG compliance dashboard, see the complete details in How to Create a DISA STIG Dashboard on THWACK.

Learn more

After you define policy reports, you can use the reports to find and remediate policy violations. You can also share policy reports between Orion servers or with other SolarWinds users.

Find and remediate policy violations

You can use policy reports to find device configurations that do not comply to policies. To view information about policy violations, you must:

1. Make sure the cached policy compliance information is up-to-date.
2. View a policy report to display the cached information.

After you have viewed a report and remediated one or more violations, you can verify that the violations were successfully remediated.
Update cached policy compliance information

NCM runs policy reports against the selected nodes to locate any policy violations, and then caches the results so that the data can be accessed quickly. You can update cached policy compliance information in any of the following ways:

- Enable the policy cache to **automatically update information each day**. (This option is enabled by default.)
- **Manually** update the policy cache.
- **Schedule a policy report job** to update cached information for that report and send emails about violations.

Automatically update cached policy compliance information

When the policy cache is enabled, NCM automatically runs policy reports at the specified time to check for policy violations. By default, the policy cache is updated daily at 11:55 PM.

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
5. Specify what time to generate the policy cache.
6. Click Submit.

Manually update cached policy compliance information

After you modify policy rules or download updated configuration files, you can manually update cached policy compliance information to reflect the changes.

1. Click My Dashboards > Configs > Compliance.
2. To manually update the cached information:
   - To update all reports, click Update All.
   - To update one or more reports, select the reports and click Update Selected.

Schedule a policy report job

You can configure a policy report job to send emails each time the job runs or only if it finds policy violations. When the job runs, it updates the cached policy compliance information so that the report provides a snapshot of current policy compliance.

1. Click My Dashboards > Configs > Jobs.
2. Click Create New Job.
3. Name the job, and select Generate a Policy Report from Job Type.
4. Select the Schedule Type. If you are creating an Advanced schedule, use the five fields to **create a CRON expression**.
5. Add a comment if this job relates to a business rule, and click Next.
6. On the Choose Nodes tab, click Next.

7. Select an email notification option, and click Next. If you click Email Results, the default email notification and SMTP server settings are populated. These settings can be overridden in each job.

8. Select the policy report to generate as part of the job.

9. If you want to suppress notifications when no violations are found, select Send Notification Only When There Are Policy Violations.

10. Click Next.

11. Review the settings for the job, and click Finish.

View a policy report

When you view a report, it displays the latest cached policy compliance information. Use this information to investigate and remediate policy violations.

If compliance information for a report is not current, you can manually update the information.

1. Click My Dashboards > Configs > Config Summary.
   The Policy Violations resource lists the policy violations found the last time each report ran.

2. Click a report name to open the Report Details page.
   The upper left corner shows the Last Updated date and time.
   Icons indicate which rules were violated on each node:
   - A green check mark indicates that the rule was not violated.
   - Any other icon indicates that the rule was violated, and reflects the severity of violating that rule.

3. Click a violation icon.
   The Violation Details dialog shows the rule that was violated and indicates if a remediation script is available.
   If the violation occurred because the string was found, you can click the arrow to display the line number in the config file.
4. (Optional) Click View Config to open the Config Details view in a different tab.

5. (Optional) If a remediation script is available, click a Management option to execute the script on this node or on all nodes in violation.

Verify that a policy violation was remediated

When you discover a policy violation, complete the following steps to resolve the issue and verify that the remediation was successful.

1. Update the configuration file to resolve the policy violation, either by editing the file or by running a remediation script.
2. Download the updated configuration file.
3. Update the cached policy compliance information for the report that detected the policy violation.
4. View the report to verify that the policy violation is not found.

Import and export policy report definitions

If you have multiple Orion servers, you can share policy report definitions between them. You can also share your policy report definitions with other SolarWinds users by exporting them to THWACK.

Share policy reports between Orion servers

You can export a policy report definition as an XML file, and then import it to another Orion server. The exported file also includes the underlying policy and rule definitions.

Export a policy report definition as a file

1. Click My Dashboards > Configs > Compliance.
2. Click Manage Policy Reports.

   The XML file is saved in your \Downloads directory. The file name is based on the report name.

Import a policy report definition

1. Click My Dashboards > Configs > Compliance.
2. Click Manage Policy Reports.
3. Click Import.
4. Click Choose File, and select a policy report previously exported as a file.
5. Click Submit.

   The policy report and underlying policies and rules are created on the Orion server. By default, the policy report is not assigned to a folder. You can edit the report to assign it to a folder.

Export a policy report definition to THWACK

1. Click My Dashboards > Configs > Compliance.
2. Click Manage Policy Reports.
3. Select a report, and click Export to THWACK.
4. If prompted, enter your THWACK user name and password, and click Log In.
Jobs

SolarWinds Network Configuration Manager provides job scheduling to automate the management of network devices and configuration files.

You can schedule numerous operations, including configuration file uploads and downloads, node reboots, and command script execution.

Orion Platform Administrator, NCM Administrator and NCM Engineer roles have full access to all jobs in the job list. Other assigned SolarWinds NCM roles can access and manage only the jobs they create, but not others.

Create or edit a job

When processing an active job, SolarWinds NCM uses credential settings of the user who last edited the job. For example, if the user has user-level login credentials set, NCM uses the connection profile associated with the device.

1. Click My Dashboards > Configs > Jobs.
2. If you are creating a new job, click Create New Job.
3. If you want to edit an existing job, select a job, and click Edit.
4. Name the job, but do not use special characters in the name.
5. Select a Job Type.
6. Select the Schedule Type. If you are creating an Advanced schedule, use the five fields to create a CRON expression.
7. Add a comment if this job relates to a business rule, and click Next.
8. Select the nodes to target with this job, and click Next.
9. Select one or more notification options, and click Next.

<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save Job Log</td>
<td>Select this option to save the results of the job to the History column of the database, and viewable from the Web Console as a glance of completion for example. This file is not configurable.</td>
</tr>
<tr>
<td>Save Results to File</td>
<td>You can use this option to keep a job history, such as an Execute Command script or Download Config. You can edit the path and file name of the saved file, as long as NCM has access to the path with read and write privileges. For more information, see Writing job results and other files to a Network Share. The path must exist. NCM can create the file but not the folders in the path. Supported file extensions include .txt, .csv, .xls, .doc, and .htm.</td>
</tr>
<tr>
<td>Email Results</td>
<td>If you click Email Results, the default email notification and SMTP server settings are populated. These settings can be overridden in each job.</td>
</tr>
</tbody>
</table>

10. Add details based on the specific job, and click Next.

11. Review the settings for the job, and click Finish.

**Enable a job**

The enable operation applies to jobs that run according to a schedule. You must enable a job in SolarWinds NCM before you can start it. A job that is not enabled will not start.

1. Click My Dashboards > Configs > Jobs.
2. Select a disabled job in the list.
3. Click Enable.

**Disable a job**

The disable operation applies to jobs that run according to a schedule. You can delete any job in SolarWinds NCM that you no longer use.

Disable a job to suspend it, but not delete it.

1. Click My Dashboards > Configs > Jobs.
2. Select a scheduled job in the list.
3. Click Disable.
Start a job

Though using a schedule is the most efficient way to manage jobs, you can manually start a job if necessary.

You can start any job that is enabled.

1. Click My Dashboards > Configs > Jobs.
2. Select a job in the list.
3. If the job is disabled, click Enable.
4. Click Start Job.

Stop a job

Though using a schedule is the most efficient way to manage jobs, you can manually stop a job if necessary.

A job currently running shows the status: running.

1. Click My Dashboards > Configs > Jobs.
2. Select a running job in the list.
3. Click Stop Job.

View job logs

View the job log to verify that a job was run as scheduled, or to view the history of the job.

1. Click My Dashboards > Configs > Jobs.
2. Select the job in the list.
3. Sort by the Last Date Run column.
4. Click the page icon in the History column to view the log.

Delete a job

Permanently remove a job, rather than temporarily disable it.

1. Click My Dashboards > Configs > Jobs.
2. Select a disabled job in the list.
3. Click Delete.
Approval system for device configuration changes

SolarWinds Network Configuration Manager enables you to define a semi-automated approval process for making configuration changes on network devices.

SolarWinds NCM uses roles to determine which Orion accounts are able to perform the tasks of changing device configurations (WebUploader), approving those changes (Administrator), and changing the roles of Orion accounts (Engineer, Administrator).

SolarWinds NCM uses email to relay config change approval requests. As part of setting up the config change approval system, you must provide SMTP information.

Set up config change approval

The setup wizard guides you through the process of specifying an email server for notifications, addresses of change approvers, and accounts of team members who manage device configurations.

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
4. Select an approval mode, and click Submit.
5. Enter SMTP settings, and click Submit.
6. Enter Admin Email settings.
   - If you are using one-level approval, the addresses in the To field receive notifications of pending device config changes.
   - The address in the From field is shown as the sending address for config change approval requests.
   - The Subject field appears in the subject line of config change approval requests.
   - Add an additional message regarding to inform approvers of the standard action to take.
7. Click Submit.
8. If you selected one-level approval, click Manage Users to adjust the NCM role for accounts that will be submitting upload requests, and follow the steps in Create, edit, and delete NCM accounts. Otherwise, click Finish.
9. If you selected a form of two-level approval, select a user and enter a valid address for Approval Level 1 and Approval Level 2, and click Finish.
   - For each user in the list that you want to have a WebUploader role but neither the Engineer or Administrator role, select the user name, and click Edit.
10. When you are finished managing users and adding level 1 and level 2 addresses, click Submit.
When you enable the change approval system, SolarWinds NCM prevents the system from executing device uploads until an NCM administrator has approved the scheduled job.

**Turn on approval system**

After setting up the configuration change approval system in NCM, you need to enable the system to send approval notifications and process acknowledgments. The approval system can be disabled at any time.

**Enable config change approval**

When you enable the change approval system, SolarWinds NCM prevents the system from executing device uploads until an NCM administrator has approved the scheduled job.

> The default setting is to allow any NCM user with the WebUploader role or higher to make and push a device configuration change with no approval necessary. Enabling the Change Approval System without changing the default setting is the same as not enabling the system at all. Review the default settings and user roles and make adjustments as necessary before enabling the approval system.

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
3. Under Config Management Change Approval, click Enable Approval System.
4. Click Enable.

**Disable config change approval**

When you disable the system, SolarWinds NCM processes device configuration changes normally, either as scheduled or immediately, depending on the actions of the relevant team member with the WebUploader account privileges.

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
3. Under Config Management Change Approval, click Disable Approval System.

**View, approve, and decline approval requests**

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
3. Under Config Management Change Approval, click Pending Approval.
4. Select a request in the list, and click Approve or Decline.
Create, edit, and delete NCM accounts

Only a user with Administrator privileges can create a new account and prepare it for use with NCM.

Access account settings
1. Click Settings > All Settings.
2. Under User Accounts, click Manage Accounts.

Create a new account
1. Click Add New Account.
2. Select Orion Individual Account, and click Next.
3. Enter credentials for the new account, and click Next.
4. Define settings for the Orion individual account.
   - Decide if this account should have Administrator Rights and Node Management Rights.
   - Accept the current defaults under Default Menu Bar and Views and Orion General Settings. The account holder will be able to customize these settings.
   - Assign the NCM role in the Network Configuration Manager Settings. The account must at least have the WebUploader for the user make and upload configuration changes to network devices.
5. Click Submit.

Edit an existing account
1. Select an account, and click Edit.
2. Edit settings for the Orion individual account.
   - Decide if this account should have Administrator Rights and Node Management Rights.
   - Assign the NCM role in the Network Configuration Manager Settings. The account must at least have the WebUploader for the user make and upload configuration changes to network devices.
3. Click Submit.

Delete an account
1. Select an account.
2. Click Delete.
Device access information

A SolarWinds NCM user logs on directly at the network device with unencrypted credentials and can perform actions the NCM role gives permission for.

If the network administrator wants to use the same credentials for SolarWinds NCM to log on to all network devices, the NCM software provides a Global Login and an option to enable global login settings on all devices. See Set node communication defaults.

Third Party Authentication

If a network administrator sets up third-party authentication, such as a Diameter, RADIUS, or TACACS server, the admin should create valid accounts and permissions in the authentication server database for SolarWinds NCM users.

Only NCM interacts with network devices, not the authentication server. Though the network device must handle interaction with Diameter, RADIUS, TACACS, or any other authentication server, special logic in the relevant NCM component (SWTelnet9) handles the RADIUS authentication prompt, since devices connected to the RADIUS server may have a slightly different login flow.
Access control list management

Use NCM to manage access control list (ACL) rules for Cisco ASA firewalls and Cisco Nexus devices. You can:

- **View and compare Cisco ASA ACL rules**
  You can compare ACLs (or different versions of the same ACL), and view and manage the rules within an ACL. NCM automatically detects redundant and shadowed rules.

- **Learn about overlapping (shadowed and redundant) ACL rules**
  Eliminating shadowed and redundant rules reduces the size of the rule set, making it easier to manage, and helps you ensure that the rules achieve the intended results.

Manage ACLs on Cisco ASA and Nexus devices

Use SolarWinds NCM to help you manage the access control lists (ACLs) for your Cisco ASA and Cisco Nexus devices. Find rules that are not being applied as intended, and identify unnecessary or redundant rules that can be removed. Streamlining ACLs makes them easier to manage and saves CPU and memory on your devices.

- **Display** the ACLs on a device.
- **Compare** ACLs on the same device or different devices.
- **Display the rules** in an ACL.
- **Identify rules that have not been applied.**
- **Investigate overlapping rules** in an ACL.
- **Display information about objects or objects groups** in a rule.

Display the access control lists on a device

Before you can complete other management tasks, you must display a list of the ACLs on a device.

2. Double-click the name of a Cisco ASA or Cisco Nexus device.
   The Node Details page opens.
3. From the menu on the left, choose Access Lists.
   The Access Lists page lists the ACLs configured for that device. If an ACL has changed, click the arrow to display a list of previous versions.

   A warning icon  
   indicates that the ACL contains overlapping rules. You can display the ACL rules.
to find out which rules overlap.

![Displays previous versions of an ACL]

Indicates that overlapping rules were detected

### Compare ACLs

Use NCM to quickly locate the differences between ACLs or ACL versions. For example, you can compare two versions of the same ACL to determine what changed, or to verify that changes were implemented correctly. You can compare ACLs on different nodes to verify that the same rules are being applied on both devices.

#### Compare ACLs on the same device

1. **Display the list of ACLs** on a device.
2. Select the ACLs or ACL versions to compare.
   - To compare the current version to a previous version, expand the node to list previous versions.
3. At the top of the page, click Compare ACL.
   - The rules from both ACLs are displayed beside each other. The line number of rules with differences are highlighted.

#### Compare ACLs on different devices

To compare ACLs on different devices, first select two ACLs on the same device, and then change one of the ACLs being compared.

1. **Display the list of ACLs** on a device.
2. Select the ACL that you want to compare, and any other ACL on that node. (You will change the second selection later.)
3. At the top of the page, click Compare ACL.
   - The rules from both ACLs are displayed beside each other. The line number of rules with differences are highlighted.
4. Change the second ACL to an ACL on a different device:
   a. Near the top of the page, click Change ACLs Compared.
   b. Select the node, interface, name, and version of the ACL you want to compare.
   c. Click Change.

Display ACL rules

When you display ACL rules, also known as Access Control Entries (ACEs), SolarWinds NCM identifies overlapping rules, which might require additional investigation.

1. Display the list of ACLs on a device.
2. Click an ACL name.
   The rules (or ACEs) are listed on the Rules of This Access List page. The right column shows the number of hits, and a warning icon 🔄 indicates that the rule overlaps another rule.

   Use the search and filter options to find a specific rule quickly. Or use filters to display all rules that meet certain criteria.
   Click Edit Filter Properties to change the options available for filtering.

Identify rules that have not been applied

1. Display ACL rules.
   On the right side of each line, the ACL browser displays the hit count for the rule. The hit count indicates the number of times the rule has been applied.
   By default, rules are sorted by line number.
2. To sort the list by hit count, click the down-arrow on the sort menu and choose Hit Count.
   Rules that have never been applied (0 hits) are at the top of the list.

   Rules with 0 hits

Investigate overlapping rules

1. Display ACL rules.
2. Point to the warning icon 🔄 that identifies a rule that overlaps another rule.
   A message describes the issue.
3. Click Show the details.

A dialog box displays the preceding rules that shadow the rule or make it redundant.

Display information about objects or object groups

1. Display ACL rules.
2. If an object or object group is listed as the source or destination of a rule, click the name of an object or object group.
   Information about the current version of the object or object group is shown on the right.
3. To view information about a previous version of the object or object group, select a version from the drop-down menu.
4. To compare another version to the currently selected version, click Compare Diff and select the version.

Overlapping ACL rules

Cisco ASA and Nexus devices evaluate rules in order, from top to bottom. Overlapping rules occur when some or all of the traffic that would have been processed by one rule has already been processed by a previous rule. When you view ACL rules for a Cisco ASA or Nexus device, SolarWinds NCM displays a warning icon to identify overlapping rules.

Finding and eliminating overlapping rules reduces the size of the rule set, making it easier to manage, and also helps you ensure that the rules achieve the intended results.

NCM detects four types of overlapping rules on Cisco ASA and Nexus devices:

- Fully shadowed
- Partially shadowed
- Fully redundant
- Partially redundant

i When detecting overlapping rules, NCM supports both contiguous and discontiguous masks.
Fully shadowed rules

A fully shadowed rule is detected when:

- The criteria for one rule matches all of the traffic covered by a second rule.
- The two rules apply different actions.

The second rule is fully shadowed by the first. The rules conflict, but the shadowed rule is never applied to any traffic because it comes later in the access list. For example:

```
access-list acl1 permit ip 1.0.0.0 255.0.0.0 any
access-list acl1 deny ip 1.229.0.0 255.255.0.0 any
```

Partially shadowed rules

A partially shadowed rule is detected when:

- The criteria for one rule matches some of the traffic covered by a second rule.
- The two rules apply different actions.

The second rule is partially shadowed by the first. It is applied to only some of the intended traffic. For example:

```
access-list acl102 permit ip 10.0.0.0 255.255.255.0 any
access-list acl102 deny ip any any
```

*In some cases, a partially shadowed rule might be intentional. For example, you might want to permit traffic from specific IP addresses, but deny all others.*

Fully redundant rules

A fully redundant rule is detected when:

- The criteria for one rule matches all of the traffic covered by a second rule.
- The two rules apply the same action.

The second rule is fully redundant because of the first. It is never applied to any traffic. For example:

```
access-list acl3 permit ip 1.0.0.0 255.0.0.0 any
access-list acl3 permit ip 1.229.0.0 255.255.0.0 any
```
Partially redundant rules

A partially redundant rule is detected when:

- The criteria for one rule matches **some** of the traffic covered by a second rule.
- The two rules apply the **same** action.

The second rule is partially redundant because of the first. It is applied to only some of the intended traffic. For example:

```plaintext
access-list acl4 deny ip 192.168.169.0 255.255.255.0 any
access-list acl4 deny tcp any any range echo acl
```
Reports

The Orion platform database accumulates much information that can be presented in a variety of formats. Click Reports > All Reports to access all reports.

Run a config change report once

1. Click My Dashboards > Configs > Jobs.
2. Click Create New Job.
3. Name the job, and select Generate a Config Change Report from Job Type.
4. Select Basic as the Schedule Type.
5. Select Once, and then enter a day and time that is at least 15 minutes from the current NCM server time.
6. Add a comment if this job relates to a business rule, and click Next.
7. Select the nodes to target with this job, and click Next.
8. Select an email notification option, and click Next. If you click Email Results, the default email notification and SMTP server settings are populated. These settings can be overridden in each job.
9. Select the type of config change report to generate, and click Next.
10. Select Only Send Devices That Had Changes if desired, and click Next.
11. Review the settings for the job, and click Finish.

To create a config change report to run recurrently, see Schedule a recurring config change report.

When processing an active job, SolarWinds NCM uses credential settings of the user who last edited the job. For example, if the user has user-level login credentials, SolarWinds NCM uses the connection profile associated with the device.

Schedule a recurring config change report

1. Click My Dashboards > Configs > Jobs.
2. Click Create New Job.
3. Name the job, and select Generate a Config Change Report from Job Type.
4. Select the Schedule Type. If you are creating an Advanced schedule, use the five fields to create a CRON expression.
5. Add a comment if this job relates to a business rule, and click Next.
6. Select the nodes to target with this job, and click Next.
7. Select an email notification option, and click Next. If you click Email Results, the `default email notification and SMTP server settings` are populated. These settings can be overridden in each job.

8. Select the type of config change report to generate, and click Next.

9. Review the settings for the job, and click Finish.

When processing an active job, NCM uses credential settings of the user who last edited the job. For example, if the user has user level login credentials set, NCM uses the connection profile associated with the device. For details on managing a connection profile, see [Define and apply global device logins](#).

### Predefined NCM reports

The following reports are predefined and included with SolarWinds Network Configuration Manager.

<table>
<thead>
<tr>
<th>Report</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCM Audit</td>
<td>Displays results of audit checks of performance, activity, configuration transfers, real-time change notifications, and vulnerability state changes</td>
</tr>
<tr>
<td>Brocade Inventory</td>
<td>Displays information about the physical entities and agent config modules running on the chassis of Brocade devices</td>
</tr>
<tr>
<td>Cisco Inventory</td>
<td>Displays information about the physical properties of the system</td>
</tr>
<tr>
<td>F5 Inventory</td>
<td>Display information about the physical entities of the discovered F5 systems and LTM and GTM object activity details</td>
</tr>
<tr>
<td>NCM Inventory</td>
<td>Displays Information about physical entities and network services configured on NCM discovered devices.</td>
</tr>
<tr>
<td>Node Details</td>
<td>Displays details such as status, availability, response times, events, and alerts for each node under NCM management</td>
</tr>
<tr>
<td>Polling Status</td>
<td>Displays information about polling and rediscovery cycles</td>
</tr>
<tr>
<td>Route Tables Inventory</td>
<td>Displays routing tables for relevant devices</td>
</tr>
<tr>
<td>Security</td>
<td>Displays information about logins, vulnerabilities, and security settings such as the SNMP community strings used by devices</td>
</tr>
<tr>
<td>Windows Server Inventory</td>
<td>Displays information about the software and services installed on Windows nodes</td>
</tr>
<tr>
<td>STIG compliance</td>
<td>Displays results from a compliance check with DISA STIG</td>
</tr>
</tbody>
</table>

### NCM Audit

**Config Transfer Audit**

Displays information on config transfers.
Real Time Change Detection Audit
Displays change notification messages on nodes within the network.

User Activity Tracking Report
Displays relevant information on node activity within the network.

Vulnerability State Change Audit
Displays information about vulnerability state changes.

Brocade Inventory

Brocade Agent Config Module
Displays information on Brocade agent config modules running in chassis.

Brocade Chassis Serial Number
Displays serial number for Brocade chassis.

Brocade Chassis Unit Serial Number
Displays serial number for units within Brocade chassis.

Cisco Inventory

Cisco 3750 Stack - Physical Entity
Displays information about physical entities within each device.

Cisco Bootload Images
Displays information related to bootloader images within each device.

Cisco Card Data
Displays information about Cisco card data within each device.

Cisco Catalyst Cards
Displays the card in each Catalyst switch.

Cisco Chassis IDs
Displays the chassis ID and serial number for Cisco devices.

Cisco Discovery Protocol
Displays the CDP discovered from each device.

Cisco Fan Tray Status
Lists the operational status information for a device's fans or fan trays.

Cisco Flash File System
Displays the list of files in the flash memory of Cisco devices.
Cisco Flash Memory
Displays the amount of flash memory in Cisco devices.

Cisco FRU Power Status
Lists the power-related administrative status and operational status of the manageable components in the system.

Cisco FRU Power Supply Groups in the system
Displays the redundancy mode and the operational status of the power supply groups in the system.

Cisco IOS Image Details
Displays details of IOS in each Cisco device.

Cisco Memory Pools
Displays memory pools in Cisco devices.

Cisco VLANs
Displays the VLANs to which devices belong.

Memory in Cisco Devices
Displays memory pools in Cisco devices.

Old Cisco Cards
Displays the cards in Cisco devices.

ROM Bootstrap for Cisco Devices
Displays the ROM bootstrap version for selected devices.

F5 Inventory

GTM Object Activity
Displays details about Global Traffic Manager object activity that indicates usage rates for the relevant devices.

Capacity Planning
Displays capacity details such as CPU usage, memory, and throughput relevant to the F5 devices in your network.

F5 Device Inventory
Displays information about the physical entities, software, and state of each F5 device discovered by NCM in your network.

LTM Node Inventory
Displays status and configuration information about the Local Traffic Manager node.
LTM Object Activity
Displays activity details that indicate usage rates for Local Traffic Manager objects in your NCM-managed network.

NCM Inventory
ARP Tables
Displays ARP tables from each device.

Interfaces
Displays details about each interface.

IP Addresses on Each Interface
Displays the IP addresses assigned to each device.

Juniper Physical Entities
Displays information about Juniper physical entities within each device.

Last Status Change for Each Interface
Displays the time the status last changed on each interface.

Logical Entities
Displays the logical entities within each device.

Physical Entities (Serial Number)
Displays information about physical entities within each device.

Physical Entities (Serial Number) v2
Displays information about physical entities within each device.

Switch Ports
Displays spanning tree and VLAN information on each switch port.

TCP Services
Displays the TCP services on each device.

UDP Services
Displays the UDP services on each device.

Node Details Report
All Nodes
Displays all nodes managed by NCM.
Backup Status of Running Config
Displays the timestamp for when the current Running config was backed-up or an indication that it has never been backed up.

Backup Status of Startup Config
Displays the timestamp for when the current Startup config was backed-up or an indication that it has never been backed up.

Connection Profiles
Displays the Connection Profile used by each device.

Current IOS Image and Version
Displays the current IOS image and version.

End of Support Devices
Displays devices that are at the end of their support.

Last Boot Time for Each Device
Displays timestamp for when the device was last booted.

Last Inventory of Each Device
Displays timestamp for when the device was last inventoried.

Overall Baseline Vs. Running Config Conflicts
Displays where the Running config differs from the Baseline config.

Overall Configuration Changes Snapshot
Displays all changes to the Running, Baseline, and Startup configs.

Overall Devices Backed Up Vs. Not Backed Up
Displays whether nodes have been backed up.

Overall Devices Inventoried Vs. Not Inventoried
Displays whether nodes have been inventoried.

Overall Running Vs. Startup Config Conflicts
Displays where the Running config differs from the Startup config.

System Information of Each Device
Displays the System Description, Location and Contact Information discovered from each device.

Polling Status

Devices that do not respond to SNMP
Displays information on Brocade agent config modules running in chassis.
Down Nodes
Displays serial number for Brocade chassis.

Route Tables Inventory
Route Tables
Displays route tables for relevant devices.

Security
Community Strings for Each Node
Displays the SNMP community string used by each device.

Login Failure Report
Displays the devices that cannot be logged into and the error information.

Login Information for Each Device
Displays the username, enable level, and community strings for each device.

Login Status
Displays any error messages relating to login information for each device.

Nodes for Each Vulnerability
Displays nodes by vulnerability.

Vulnerabilities for Each Node
Lists vulnerabilities for each device.

Vulnerabilities for Each Node - Confirmed
Lists confirmed vulnerabilities for each device.

Vulnerabilities for Each Node - Not Applicable
Lists not applicable vulnerabilities for each device.

Vulnerabilities for Each Node - Potential
Lists potential vulnerabilities for each device.

Vulnerabilities for Each Node - Remediated
Lists remediated vulnerabilities for each device.

Vulnerabilities for Each Node - Remediation Planned
Lists remediation planned vulnerabilities for each device.

Vulnerabilities for Each Node - Waiver
Lists waiver vulnerabilities for each device.
Windows Server Inventory

**Installed Services**
- Displays installed services for Windows nodes.

**Installed Software**
- Displays installed software for Windows nodes.

**Windows Accounts**
- Displays Windows accounts for relevant nodes.
Servers for file transfers and email notifications

Use the following topics to configure servers for file transfers and email notifications regarding real-time config detection.

Use the SolarWinds TFTP server

SolarWinds NCM uses TFTP for file transfers.

If you have NCM and NPM integrated but installed on separate servers, then follow these steps on the NPM server. The settings will automatically apply to NCM.

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
4. Enter the TFTP Server settings, and click Validate.
5. Click Submit.

Use the SFTP & SCP server

The SFTP & SCP Server runs as a service, but some basic configuration may be necessary to ensure it behaves in a way that works best in your environment.

2. Click File > Configure.
3. Type or browse to a Root Directory.
4. Select the protocols you want the server to support from the Allowed Protocols list.
5. Select options in the Permitted File Transfer Operations section.
6. Click the TCP/IP Settings tab, and type the TCP Port.
7. If you want to specify the IP address configuration, select Use Custom IP Address Binding, and select the IP address you want to use.
8. If you want to enable user authentication on the server, complete the following:
   a. Click the Users tab.
   b. Click New User.
   c. Type the user name and password, and click Apply Changes.
   d. If you want to remove a user, select the user, and click Remove.
9. Click the Startup & System Tray tab, and select the desired options.
10. Click OK.

Use a third-party SCP server

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
4. Select Use Third-Party SCP Server.
5. Select Allow Me to Specify the IP Address of the SCP Server to prevent NCM from resolving the SCP server based on its own host and disabling other entries.
6. Enter a valid user name and password.
7. Specify a config transfer directory, and click Validate to verify that the SCP Root Directory is set to be the same as the third-party server.
8. Click Submit.

The user must have receive and transmit permissions configured in the third-party SCP server.

Configure email notifications

SMTP server credentials used to send notifications for config change approvals, real time change detection, and running jobs

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
3. Under Manage Notifications, click SMTP Server.
4. Enter the fully qualified domain name (FQDN) of the server in Email Server Address.
5. Enter the Port Number on which the server handles messages.
6. Select an Authentication type.
7. If you selected Password as your Authentication type, enter a user name and password that the server accepts.
8. Click Submit.

Email notification defaults used in new scheduled jobs

1. Click Settings > All Settings.
2. Under Product Specific Settings, click NCM Settings.
3. Under Manage Notifications, click Email Notification.
4. Enter in Sender Name the name you want to appear in a job completion notification.
5. Enter a Reply Address if desired.
6. Enter the Subject to be used to send a job completion notification.
7. Enter recipients in the To field.
8. Click Submit.
Config change templates

SolarWinds Network Configuration Manager allows you to create, use, and manage config change templates that streamline making recurrent and complex configuration changes.

Generate and execute accurate sets of CLI commands with a single config change template. Perform a specific task on different machine types in your network.

The basics of config change templates

A change config template enables you to accomplish a specific device configuration task for a set of NCM-managed nodes. Template describes the run-time wizard through which the user selects the NCM nodes or interfaces targeted for the change. The script behind the wizard articulates the logic of the configuration change itself.

An example of configuration change workflow is an IT manager who creates the script for a template, and other team members who use the template's wizard to perform the specific configuration changes on a set of NCM-managed nodes.

The framework for creating config change templates depends on the SolarWinds Information Service (SWIS). SWIS is an API that is installed with NCM and that interacts with inventory data in the Orion platform database. Any device that has not been inventoried in NCM cannot be targeted with a config change template. Each object in a device inventory is a SWIS entity that can be referenced in specific ways within scripts.

Examples of routine changes you can expedite with config change templates include:

- Changing VLAN membership by device port
- Configuring device interfaces based on description
- Enabling IPSLA for VOIP implementations
- Managing NetFlow collection at the source devices

Preparation and use of a template

Every change config template executes changes based on NCM device inventory objects. Those objects are the database entities that the SolarWinds Information System accesses in managing the communication of NCM with its database. As a result, performing an NCM device inventory and updating device inventories are the prerequisites for creating and running a config change template.

See SolarWinds SolarWinds Information Service data entities for all the NCM device entities and their properties that you can use in your work with config change templates.

Two types of NCM users work with config change templates:
Template Creator:

This user creates the script for a config change template. The user must know the basics of writing a script that uses commands, variables, and logical structures.

When executed, a config change template displays a wizard that uses input values to generate CLI commands that accomplish a specific config change on targeted NCM devices. Based on input, the template's run-time wizard generates a different set of commands for each type of device that you specify as a target for config changes.

You must have the NCM role of Administrator or Engineer to create or edit a change config template.

Template User

This user enters values based on a template's run-time wizard input prompts, reviews the CLI commands that the template outputs for each type of targeted device, and tells NCM to execute the commands against targeted NCM devices. The user makes specific config changes if necessary.

You must have the NCM role of Administrator, Engineer, or WebUploader to use a change config template.

Parts of a config change template

Every config change template includes two parts: parameters and commands.

Parameters

Parameters tell a user about the template's purpose through its descriptions. Parameters prompt the user for the values, such as the specific node(s) on which to make the template's specific config change.

Commands

Commands declare the input type for a variable through arguments. Commands include arguments and logical operations needed to produce a set of CLI commands and execute those commands against each NCM node targeted for a specific config change.

A template creator develops a script for a template by defining the parameters that tell a user about the template and associating a description or label with a variable in the template's user interface.

Config change template details

A config change template named Change VLAN Membership on Ports Cisco IOS installs with SolarWinds NCM. Its purpose is to change VLAN membership on Cisco IOS device ports.

The following sections explain the specific components of a config change template by demonstrating how to use the Change VLAN Membership on Ports Cisco IOS template to make VLAN membership config changes on hypothetical Cisco device interfaces.

Viewed as parsable code, a config change template consists of two parts: parameters and commands. For more information, see Parts of a Config Change Template.
Example

This section assumes that you know how to make VLAN membership changes to device interfaces from the Cisco IOS command line. This section also assumes that you are familiar with using variables, data arrays, foreach loops, if/else conditional statements, and logical operators in creating system administration scripts.

The following sections show the reference template broken up into parameter, command, and output sections.

Parameters

These are the parameters for the Change VLAN Membership on Ports Cisco IOS template. Notice that the parameters already have values associated with them, which are either a string or a variable.

```*/
.CHANGE_TEMPLATE_DESCRIPTION
   This change template configures VLAN membership on
   Cisco IOS devices. The template was verified on
   Cisco 2950 Catalyst Switch running IOS software
   version 12.1(12c).
.CHANGE_TEMPLATE_TAGS
   Cisco, IOS, VLAN Membership
.PLATFORM_DESCRIPTION
   Cisco IOS

.PARAMETER_LABEL @ContextNode
   NCM Node
.PARAMETER_DESCRIPTION @ContextNode
   The node the template will operate on. All
   templates require this by default. The target
   node is selected during the first part of the
   wizard so it will not be available for selection
   when defining values of variables.

.PARAMETER_LABEL @TargetPorts
   Select Port(s)
.PARAMETER_DESCRIPTION @TargetPorts
   Select the port(s) for which you would like to
   change VLAN membership.

.PARAMETER_LABEL @VlansToRemove
   VLAN(s) to remove
.PARAMETER_DESCRIPTION @VlansToRemove
   Select the VLAN(s) to remove. Selecting VLANs
   irrelevant to interfaces results in no actions
   taken for those interfaces.
```
Commands

There is one instance of the script command and multiple instances of the CLI{  } command, and all variables have declarations.

```plaintext
script ConfigureVLANmembershipCiscoIOS (
    NCM.Nodes @ContextNode,
    NCM.Interfaces[] @TargetPorts,
    NCM.VLANs[] @VlansToRemove,
    NCM.VLANs @VlanToAssign )
{
    // Enter configuration mode
    CLI
    { configure terminal }

    // Loop through selected ports
    foreach (@portItem in @TargetPorts)
    {
        CLI
        { interface @portItem.InterfaceDescription }

        // Loop through list of vlans to remove
        foreach (@vlanRemove in @VlansToRemove)
        {
            CLI
            { no switchport access vlan @vlanRemove.VLANID }
        }

        CLI
        { switchport access vlan @VlanToAssign.VLANID }
        CLI
        { exit }
    }

    // Exit configuration mode
    CLI
    { exit }
}
```
Output Commands

These are the commands that NCM executes after logging on to the NCM device(s) selected as the target for this config change template. We are changing VLAN membership on one interface of two different Cisco switches.

bgp-2651-03

```
configure terminal
interface FastEthernet0/0
no switchport access vlan 1004
switchport access vlan 1002
exit
exit
```

cur-3725

```
configure terminal
interface FastEthernet0/1
no switchport access vlan 1004
switchport access vlan 1002
exit
exit
```

Set up the parameters

The parameters of a script define and label the variables for which a user of the template must provide values when the template is executed.

The script of every config change template includes at least five parameters. Only one, PARAMETER_LABEL, can recur in a single template and each instance requires user input to determine the value of a specific variable.

Required Parameters

CHANGE_TEMPLATE_DESCRIPTION

This parameter appears at the top of the script and briefly explains the purpose of the template. It does not have any associated variable(s) and is not exposed in the run-time wizard.

CHANGE_TEMPLATE_TAGS

This parameter holds the tags that NCM uses to provide grouping options in the Config Change Template resource. It does not have any associated variable(s) and is not exposed in the run-time wizard.

PLATFORM DESCRIPTION

This parameter defines the type of NCM device for which the template is designed.

PARAMETER_LABEL @<variable_name>
Each instance of this parameter in a config change template is associated with a specific variable. The template's run-time wizard requires the user to provide the value for each parameter variable.

By providing the input parameters for executing a template, \texttt{PARAMETER\_LABEL} delimits the data that a template can use. Think of \texttt{PARAMETER\_LABEL} as simultaneously making a variable available for user input and providing the metadata so that the user knows what the variable is holding a place for.

For example, \texttt{PARAMETER\_LABEL} is used in every template with \texttt{@ContextNode}. The user sets the value by selecting the NCM node(s) that will be targeted for config change. An instance of the parameter appears in a script as follows:

\begin{verbatim}
PARAMETER\_LABEL @ContextNode NCM Node
\end{verbatim}

In this case, \texttt{NCM Node} is the actual label that appears under the field where the NCM nodes are selected in the template's run-time wizard.

A config change template may have as many instances of \texttt{PARAMETER\_LABEL} as needed to support the user input needed for the template.

\textbf{PARAMETER\_DESCRIPTION}

This parameter holds the explanatory text for an input field and always appears after \texttt{PARAMETER\_LABEL}.

For example, the \texttt{PARAMETER\_DESCRIPTION} for the input field labeled \texttt{NCM Node} might be something like:

The NCM nodes the template will operate on. Target nodes are selected during the first part of the wizard and cannot be changed when defining values of variables.

\section*{Optional Parameters}

\textbf{PARAMETER\_DISPLAY\_TYPE}

This parameter creates a list of options. The format for using this parameter is as follows. The vertical bar character (\texttt{|}) divides the items in the list.

\begin{verbatim}
PARAMETER\_DISPLAY\_TYPE @VariableName
Listbox:1=String1|2=String2|3=String3
\end{verbatim}

\section*{Basic commands}

There are two commands in a config change template: \texttt{script} and \texttt{CLI}.

\subsection*{Script Command}

The \texttt{script[]} command declares the input type of every variable that the template uses. The \texttt{script} command declares the input type for every variable introduced in setting up the template parameters. The form of the script command is:
The data_type can be swis.entity (for example, NCM.Nodes), int (integer), or string.

Cisco Example

```
script ConfigureVLANmembershipCiscoIOS (
    NCM.Nodes @ContextNode,
    NCM.Interfaces[] @TargetPorts,
    NCM.VLANs[] @VlansToRemove,
    NCM.VLANs[] @VlanToAssign
)
```

Four variables are introduced in the parameter section of the template with an instance of PARAMETER_LABEL given a specific SolarWinds Information Service entity data type:

- @ContextNode is determined with data from the NCM.Nodes entity in the database.
- @TargetPorts is determined with data from the NCM.Interfaces entity.
- @VlansToRemove and @VlanToAssign are determined with data from the NCM.VLANs entity.

Any variable that references an NCM object that NCM knows through device inventory must take a SolarWinds Information Service entity as its data type. In this case, the four variables work with data that NCM captured and stored in the database through the device inventory process. If you attempt to assign a string instead of a SWIS entity in such cases, NCM will fail to correctly parse your script.

**CLI command**

CLI{ } defines a specific CLI command that NCM issues on a target device when the config change template is executed by a user.Script command. Its purpose is to create a command line statement that NCM can execute directly on the command line of NCM nodes targeted for the template's config change(s).

The config change template creator creates a CLI command by including its arguments wrapped by curly brackets { }. At run time, NCM parses any variables contained within CLI { }. Often a CLI command is as simple as the command you would type directly on the command line of an NCM device.

Cisco Example

To enter config mode on Cisco IOS devices, type configure terminal. In your config change template script, add the command as follows:

```
CLI
{
    configure terminal
}
```
NCM parses the argument of the CLI \{\} command by passing through the string itself (configure terminal) as a command to execute against each targeted NCM node at template run time:

Cisco Example with Variables

```python
script ConfigureVLANmembershipCiscoIOS (  
    NCM.Nodes @ContextNode,  
    NCM.Interfaces[] @TargetPorts,  
    NCM.VLANs[] @VlansToRemove,  
    NCM.VLANs[] @VlanToAssign  
)  
{  
    CLI  
    {  
        vlan database vlan @vlaniddescription @vlandesc exit  
    }  
}
```

This example shows a CLI statement with variables to specify VLAN properties while using the `vlan database` command line editor. For purposes of demonstration, we assume that:

- `@vlanid = 1`
- `@vlandesc = Local-Office`

At run time, NCM parses the CLI \{\} command as:

`vlan database vlan 1 description Local-Office exit`

If this were all that is included in the Change VLAN Membership on Ports Cisco IOS template, then the config change result would be to set the description of `vlan 1` to `Local-Office` on all NCM nodes selected as targets.

This config change template, however, actually changes the VLANs associated with targeted NCM node ports. For that we need to introduce advanced CLI \{\} command logic.

Advanced commands

The scripting framework for change config templates allows you to create CLI \{\} command arguments that include foreach loops, if/else conditional operations, and functions for manipulating string patterns.

Foreach Loops

A foreach statement iterates through an array of items based on a SolarWinds Information Service (SWIS) entity data type. Foreach statements use the following pattern:

```
foreach (@ItemVariable in @EntityArrayVariable)
```

A primary purpose of a foreach loop is to allow the template user to select multiple NCM objects for config change. The loop instructs NCM to perform the same config change on all items in scope as determined by the SWIS entity in the database and delimited at run time by the template user's selections in the template wizard.
Cisco Example

```csharp
foreach (@portItem in @TargetPorts)
{
    CLI
    {
        interface @portItem.InterfaceDescription
    }
}
```

The `foreach` statement creates a set that contains two related variables: `@portItem` and `@TargetPorts`.

The `@TargetPorts` variable holds an array of objects with the data type of an SWIS entity called `NCM.Interfaces[]`. The array will be a set of interfaces on NCM nodes.

The `@TargetPorts` variable is associated with the `PARAMETER_LABEL` Select port(s) and the template user selects one or more ports at run time. The template user determines the set of interfaces to fill the array `NCM.Interfaces[]`, and the template will perform VLAN membership config changes on each interface in that array.

`@portItem` is a dynamic variable that the loop uses during its iterating to hold the value of the current interface from the array represented by `@TargetPorts`.

The `foreach` loop format is fixed and NCM expects it to include the dynamic variable.

The user interacts with this template wizard screen:

Click Select Interfaces List to load a tree that displays available interfaces and NCM nodes previously selected in the wizard.

**Conditional Statements**

Conditional logic in a config change template script uses an if/else pattern to define two branches of possible action, enclosing specific conditions within parentheses. Within each branch of the conditional pattern are `CLI{ }` commands to execute if that branch meets the specific conditions.

Here is the basic structure:
if (condition is true)
CLI
{
  execute commands
}
else
CLI
{
  Execute other commands
}

The else section is optional. If you omit it, and the if condition is false, NCM excludes the relevant `CLI{ }` commands from the template output.

Operators

Use any of the following operators to specify a parenthetical condition. Use single quotes around string values.

<table>
<thead>
<tr>
<th>OPERATOR</th>
<th>CONDITION</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>==</code></td>
<td>Is Equal To</td>
</tr>
<tr>
<td><code>&gt;</code></td>
<td>Is Greater Than</td>
</tr>
<tr>
<td><code>&gt;=</code></td>
<td>Is Greater Than or Equal To</td>
</tr>
<tr>
<td><code>&lt;</code></td>
<td>Is Less Than</td>
</tr>
<tr>
<td><code>&lt;=</code></td>
<td>Is Less Than or Equal To</td>
</tr>
<tr>
<td><code>!=</code></td>
<td>Is Not Equal To</td>
</tr>
<tr>
<td><code>Contains</code></td>
<td>'string'</td>
</tr>
<tr>
<td><code>containsExact</code></td>
<td>'case sensitive string'</td>
</tr>
<tr>
<td><code>startsWith</code></td>
<td>'string'</td>
</tr>
<tr>
<td><code>startsWithExact</code></td>
<td>'case sensitive string'</td>
</tr>
<tr>
<td><code>endsWith</code></td>
<td>'string'</td>
</tr>
<tr>
<td><code>endsWithExact</code></td>
<td>'case sensitive string'</td>
</tr>
</tbody>
</table>

Cisco Example

Add conditional logic in the foreach loop to prevent errors that may occur if the user accidentally selects an incorrect interface (for example, the loopback address).
foreach @portItem in @TargetPorts
{
    if (@PortItem.InterfaceDescription != 'Loop0')
    {
        CLI
        {
            interface @portItem.InterfaceDescription
        }
    }
}

If the template encounters the loopback interface, it does nothing and passes on to the next interface. This code prevents damage from template user error.

Manipulating Strings

Five functions for manipulating strings constitute a final scripting resource that you can use most readily for managing ACL config changes for network firewalls, in which a config change template needs to iterate through a predictably variable set of IP addresses, for example.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Declaration</th>
<th>Variable Definitions</th>
</tr>
</thead>
</table>
| Substring  | Specify a starting point within a string and the length from the starting point that you want to capture for manipulation. | string Substring (string str, int startIndex, int length) | • str is the full string from which the substring comes  
• startIndex marks the position where the substring begins  
• length is the number of characters that the substring includes |
<p>| StrLength  | Return the length of a string.                                             | In StrLength (string str)    | • str is the user-input string whose length is used as the integer value |</p>
<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>DESCRIPTION</th>
<th>DECLARATION</th>
<th>VARIABLE DEFINITIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>IndexOf</td>
<td>Find the number of characters in a string.</td>
<td><code>int IndexOf (string str, string search)</code></td>
<td>• <code>str</code> is a string to search on&lt;br&gt;• <code>search</code> is a user-input string NCM uses to find the numerical value of the string being searched</td>
</tr>
<tr>
<td>SetOctet</td>
<td>Replace an octet within an IP address.</td>
<td><code>string SetOctet (string ipAddress, int octetPosition, string octet)</code></td>
<td>• <code>ipAddress</code> is the IP address&lt;br&gt;• <code>octetPosition</code> marks the position where the target octet begins&lt;br&gt;• <code>octet</code> is the new value of the target octet</td>
</tr>
<tr>
<td>GetOctet</td>
<td>Retrieve an octet from a user-specified IP address and octet position.</td>
<td><code>string GetOctet (string ipAddress, int octetPosition)</code></td>
<td>• <code>ipAddress</code> is a user-input IP address&lt;br&gt;• <code>octetPosition</code> is the user-input value for the place where the function finds the beginning of the octet to get</td>
</tr>
</tbody>
</table>

Example 1: Manipulating a String

```c
script IPshuffle(string @str, string @search )
{
    int @length = strlength(@str)
    int @startIndex = indexof(@str,@search)
    int @substringLength = @length - @startIndex
    string @res = substring(@strA, @startIndex, @substringLength)

    CLI
    {
        @res
```
The user enters ABCDEF for the @str variable and CD for the @search variable in the template's wizard. Based on those values, the script does the following:

1. Uses ABCDEF in the strlen function to give a value of 6 to a variable called @length.
2. Uses CD as the substring of ABCDEF to set a value 2 for the variable called @startIndex.
3. Subtracts 2 (@startIndex) from 6 (@length) to determine the value of @substringLength as 4.
4. Takes the original string ABCDEF and calculates a result (@res) using @startIndex to count in two positions and @substringLength to count four positions from the start index.
5. Outputs CDEF as the result.

Example 2: Changing an Access Control List

This example creates a block of Access Control List (ACL) instructions that predictably vary the value of a specific octet within an IP address. The instructions conform to the pattern 10.10.@id.10, where the value of @id is determined by user input.

The user enters 10.10.10.10 as the value of @ipaddress in the config change template's run-time wizard. The user enters 1,22, and 222 for the @indexes variable declared in the script command.
The script uses the SetOctet function to determine the value of an @ipnew variable. SetOctet is defined to take the user-input IP address and create a new IP address by iteratively replacing the third octet with user-input values. For each new IP address, the script produces a command to create outgoing UDP transmission access through port 2055:

- Allow 10.10.1.10 out
- Allow 10.10.1.10 UDP 2055 OUT
- Allow 10.10.22.10 out
- Allow 10.10.22.10 UDP 2055 OUT

Example 3: Managing an Access Control List for Multiple Routers

In this example, a config change template generates a block of ACL instructions for a router in a store. We create an ACL block of instructions for this device that varies based on a portion of the device's IP address.

If the store has four routers, 10.1.1.1, 10.1.4.1, 10.1.6.1, and 10.1.10.1, the template script generates an ACL block that appears this way on the selected router (10.1.1.1):

Allow 10.1.2.0/24 out
Allow 10.1.2.4 UDP 2055 OUT
Allow 10.1.4.0/24 out
Allow 10.1.4.4 UDP 2055 OUT
Allow 10.1.6.0/24 out
Allow 10.1.6.4 UDP 2055 OUT
Allow 10.1.10.0/24 out
Allow 10.1.10.4 UDP 2055 OUT

Here is the script that produces the output:

```csharp
script OpenACLs(NCM.Nodes @ContextNode, string[] @IpRouters)
{
    foreach(@ipRouter in @IpRouters)
    {
        string @octet = getoctet(@ipRouter,3)
        string @ipnew = setoctet(@ContextNode, 3,@octet)
        CLI
        {
            Allow @ipnew out
            Allow @ipnew UDP 2055 OUT
        }
    }
}
```
This script does the following:

- Uses a foreach loop to go through a user-input series of router IP addresses.
- Uses the `GetOctet` function to focus the third octet of the current router IP address.
- Uses the `SetOctet` function to create a new IP address as a value for `@ipnew`.
- Creates a CLI `{ }` command that will execute Allow operations for each of the selected routers.

The result is a set of Allow commands that open access in the ACL so that the router 10.1.1.1 can send OUT traffic via UDP on port 2055 to 10.1.4.1, 10.1.6.1, and 10.1.10.1.

Here are the parameters for this config change template. The template user selects the router on which to make ACL changes and inputs the target router IP address through this template:

```plaintext
.PARAMETER_LABEL @ACLRouter
  Router for ACL Change
.PARAMETER_DESCRIPTION @ACLRouters
  Select a Router
.PARAMETER_LABEL @ipRouters
  Target Routers
.PARAMETER_DESCRIPTION @ipRouters
  Add Routers to Target with ACL Allowances
```

## Enable NetFlow on CiscoASA example

A config change template named Enable NetFlow on CiscoASA installs with SolarWinds NCM. It configures a CiscoASA for NetFlow export.

Here are the commands that this template executes on the command line of the targeted devices selected in the template's run-time setup wizard. For this example, we are including values as if the user entered them in the wizard interface.

```plaintext
configure terminal
flow-export destination inside 10.10.18.157 2055
flow-export template timeout-rate 1
flow-export delay flow-create 60
logging flow-export syslogs disable
access-list netflow-export extended permit ip any
class-map netflow-export-class
match access-list netflow-export
policy-map netflow-policy
class netflow-export-class
flow-export event-type all destination 10.10.18.157
service-policy netflow-policy global
flow-export enable
exit
end
```
You could execute this set of CLI commands on your target device and the result would be config changes in the status of NetFlow data processing by the device.

The config change template that produces this output of CLI commands is:

```c
/*
.CHANGE_TEMPLATE_DESCRIPTION
   This change template configures your Cisco ASA for NetFlow export. This was verified on an ASA 5505 running ASA software version 8.2(1)12.
.CHANGE_TEMPLATE_TAGS
   Cisco, ASA, NetFlow
.PLATFORM_DESCRIPTION
   Cisco ASA

.PARAMETER_LABEL @ContextNode
   NCM Node
.PARAMETER_DESCRIPTION @ContextNode
   The node the template will operate on. All templates require this by default. The target node is selected during the first part of the wizard so it will not be available for selection when defining values of variables.

.PARAMETER_LABEL @NetFlowCollectorIPAddress
   NetFlow Collector IP Address
.PARAMETER_DESCRIPTION @NetFlowCollectorIPAddress
   Enter the IP address of the server running the NetFlow traffic analysis solution (for example: SolarWinds NetFlow Traffic Analyzer).

.PARAMETER_LABEL @NetFlowExportPort
   NetFlow Export Port
.PARAMETER_DESCRIPTION @NetFlowExportPort
   Enter the NetFlow export port. The default for SolarWinds NTA is 2055.
*/

script EnableNetflowOnCiscoASA (  
   NCM.Nodes @ContextNode,  
   string @NetFlowCollectorIPAddress,  
   int @NetFlowExportPort  
)  
{
   // Enter configuration mode and generate NetFlow commands
   CLI
   
   
}```
```bash
configure terminal
flow-export destination inside @NetFlowCollectorIPAddress
@NetFlowExportPort
flow-export template timeout-rate 1
flow-export delay flow-create 60
logging flow-export-syslogs disable
access-list netflow-export extended permit ip any any
class-map netflow-export-class
match access-list netflow-export
policy-map netflow-policy
class netflow-export-class
flow-export event-type all destination @NetFlowCollectorIPAddress
service-policy netflow-policy global
flow-export enable
exit
```}

Parameters

The parameters defined at the beginning of this script create an interface in which the user types the IP address and port of the NetFlow receiver.

```
.PARAMETER_LABEL @NetFlowCollectorIPAddress
NetFlow Collector IP Address
.PARAMETER_DESCRIPTION @NetFlowCollectorIPAddress
Enter the IP address of the server running the NetFlow traffic analysis solution (e.g. SolarWinds NetFlow Traffic Analyzer--NTA).

.PARAMETER_LABEL @NetFlowExportPort
NetFlow Export Port
.PARAMETER_DESCRIPTION @NetFlowExportPort
Enter the NetFlow export port (default for SolarWinds NTA is 2055).
```

The first line defines the parameter or variable name (in this case, `@NetFlowCollectorIPAddress`) for which the user enters a value in the wizard interface text box at run time. The second line defines the label (in this case, NetFlow Collector IP Address) that appears in the wizard interface to prompt the user to enter the IP address. The third and fourth lines define the description that appears below the wizard interface text box.

The parameters for NetFlow Export Port (in lines 5-12) function exactly the same way as the first four. The parameter variables, labels, and descriptions guide the config change template's run-time execution by receiving specific user input.
Command Declarations (script)

The script declarations include all the variables for which the template prompts the user to provide input. In this case, three variables and their data types are declared:

```java
script EnableNetflowOnCiscoASA (  
    NCM.Nodes @ContextNode,  
    string @NetFlowCollectorIPAddress,  
    int @NetFlowExportPort  
) {

NCM.Nodes is applied to the @ContextNode variable. NCM.Nodes refers to the Nodes entity in the SolarWinds Information Service (SWIS). In the interface wizard, the user enters a string value for the NetFlow Collector IP Address and an integer value for the NetFlow Export Port on the device.

For a complete list of entities and properties, see SolarWinds Information Service data entities.

CLI Commands

The majority of config change template code outputs original CLI commands with only a few parsed variables. Any time a variable is referenced, a value is used in its place. For example, since the user typed 10.10.18.157 as the IP address and 2055 as the collector port number, NetFlowCollectorIPAddress is replaced with 10.10.18.157 and @NetFlowExportPort is replaced with 2055 when the script runs.

```bash
flow-export destination inside @NetFlowCollectorIPAddress @NetFlowExportPort
```

The previous line of code generates the following output:

```bash
flow-export destination inside 10.10.18.157 2055
```

Execute a config change template

1. Click My Dashboards > Configs > Config Change Templates.
2. Select a template in list.
3. Click Define Variables & Run.
4. Select the target nodes.
5. Click Next.
6. Enter values in the input fields.

   Input fields for a change configuration template are defined and managed through the Edit Config Template resource. For example, in a template that enables NetFlow data exporting for a set of Cisco devices, you may be asked to enter the IP address of the relevant NetFlow collector and the port on which your target device exports flow data.

7. Click Next.

8. If you want to see the CLI commands for a device, click Show Commands in New Window.

9. Click Execute.

Create a config change template

1. Click My Dashboards > Confis > Config Change Templates.
2. Click Create New Config Change Template.
3. Name the template, enter a description, and add tags.
4. Create your Config Change Template.
5. Click Validate to check syntax.
6. Click Submit to save the template, or Execute to save and run it.

   If you choose the execute the template, SolarWinds NCM validates the syntax of the template. If validation succeeds, SolarWinds NCM saves a copy of the template and loads the relevant interface for user input.

   If validation fails, SolarWinds NCM displays an error that guides you to make changes.

Import a config change template

1. Click My Dashboards > Confis > Config Change Templates.
2. Click Import.
3. Click Choose File to find the file on your computer. Config change template files have a .ncm-template file extension.
4. Click Submit.
5. Change the name, edit the description, and add tags, if necessary.
6. Modify the logic of your Config Change Template.
7. Click Validate to check syntax.
8. Click Submit.
Tag a config change template
1. Click My Dashboards > Configs > Config Change Templates.
2. Select a template in the list.
3. Click Tags.
4. Edit tags: Add Existing, Add New, or Remove.
5. Click Submit.

Edit a config change template
1. Click My Dashboards > Configs > Config Change Templates.
2. Select a template in the list.
3. Click Advanced Modify.
4. Change the name, edit the description, and add tags, if necessary.
5. Modify the logic of your Config Change Template.
6. Click Validate to check syntax.
7. Click Submit.

Export a config change template to THWACK
1. Click My Dashboards > Configs > Config Change Templates.
2. Select a template in list.
3. Click Export to THWACK.
4. If prompted, enter your user name and password, and click Log In.

Export a config change template as a file
1. Click My Dashboards > Configs > Config Change Templates.
2. Select a template in the list.
3. Click Export as File. Verify that a pop-up blocker does not prevent the file from being downloaded.
4. Download the file to a local folder.
5. Click Return to Config Change Templates.

Delete a config change template
1. Click My Dashboards > Configs > Config Change Templates.
2. Select a template in the list.
3. Click Delete.

**SolarWinds Information Service data entities**

The following tables document all the SWIS entities and properties that you can use in developing config change templates.

### NCM.ArpTables

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Datatype</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InterfaceIndex</td>
<td>System.Int32</td>
<td>The interface on which this entry's equivalence is effective. The interface identified by a particular value of this index is the same interface as identified by the same value of RFC 1573's ifIndex.</td>
</tr>
<tr>
<td>InterfaceID</td>
<td>System.String</td>
<td>A unique GUID ID from ncm.Interfaces table.</td>
</tr>
<tr>
<td>MAC</td>
<td>System.String</td>
<td>The media dependent `physical' address.</td>
</tr>
<tr>
<td>IPAddress</td>
<td>System.String</td>
<td>The IP address corresponding to the media dependent physical address.</td>
</tr>
<tr>
<td>IPSort</td>
<td>System.Double</td>
<td>A list of IP addresses sorted with octet markers (dots) omitted.</td>
</tr>
<tr>
<td>Source</td>
<td>System.String</td>
<td>The type of IP address associated with an ARP operation and media dependent address.</td>
</tr>
</tbody>
</table>

Possible Values:
- Other (1)
- Invalid (2)
- Dynamic (3)
- Static (4)

Setting this object to the value invalid (2) has the effect of invalidating the corresponding entry in the ipNetToMediaType. That is, it effectively disassociates the interface identified with said entry from the mapping identified with said entry. It is an implementation specific matter as to whether the agent removes an Invalidated entry from the table. Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries not currently in use. Proper interpretation of such entries requires examination of the relevant ipNetToMediaType object.
<table>
<thead>
<tr>
<th>PROPERTY NAME</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDNSLookup</td>
<td>System.String</td>
<td>Result of DNS lookup on IPAddress.</td>
</tr>
<tr>
<td>NodeID</td>
<td>System.String</td>
<td>A SWIS-generated unique identifier of a network node in the current inventory. (Instances of this property recur in this table according to the number of interfaces for which ARP data is reported.)</td>
</tr>
<tr>
<td>LastDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM last discovered the device during inventory.</td>
</tr>
<tr>
<td>FirstDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM first discovered the device during inventory.</td>
</tr>
</tbody>
</table>

**NCM.ARPTables entity relationships**

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ENTITY</th>
<th>JOINED DATA ENTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>NCM.Nodes</td>
<td>NCM.NodeHostsArpTables (System.Hosting)</td>
</tr>
</tbody>
</table>

**NCM.BridgePorts**

<table>
<thead>
<tr>
<th>PROPERTY NAME</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>InterfaceIndex</td>
<td>System.Int32</td>
<td>The value of the instance of the ifIndex object, defined in MIB-II, for the interface corresponding to this port.</td>
</tr>
<tr>
<td>Port</td>
<td>System.Int32</td>
<td>The port number of the port for which this entry contains bridge management information.</td>
</tr>
<tr>
<td>SpanningTreeEnabled</td>
<td>System.String</td>
<td>The enabled/disabled status of the port.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Possible Values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enabled (1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Disabled (2)</td>
</tr>
<tr>
<td>SpanningTreeState</td>
<td>System.String</td>
<td>The port's current state as defined by application of the Spanning Tree Protocol. This state controls what action a port takes on reception of a frame. If the bridge has detected a port that is malfunctioning it will place that port into the broken(6) state. For ports which are disabled (see dot1dStpPortEnable), this object will have a value of disabled (1).</td>
</tr>
<tr>
<td>Property Name</td>
<td>Datatype</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>VlanType</td>
<td>System.String</td>
<td>The type of VLAN membership assigned to this port. A port with static VLAN membership is assigned to a single VLAN directly. A port with dynamic membership is assigned a single VLAN based on content of packets received on the port and through VQP queries to VMPS. A port with multiple VLAN membership may be assigned to one or more VLANs directly. A static or dynamic port membership is specified by the value of vmVlan. A multiVlan port membership is specified by the value of vmVlans. Possible Values: Static(1) Dynamic(2) MultiVlan(3)</td>
</tr>
<tr>
<td>VLANID</td>
<td>System.Int32</td>
<td>The VLAN id of the VLAN the port is assigned to when vmVlanType is set to static or dynamic. This object is not instantiated if not applicable. The value may be 0 if the port is not assigned to a VLAN.</td>
</tr>
<tr>
<td>PortStatus</td>
<td>System.String</td>
<td>An indication of the current VLAN status of the port. A status of inactive(1) indicates that a dynamic port does not yet have a VLAN assigned, or a port is assigned to a VLAN that is currently not active. A status of active(2) indicates that the currently assigned VLAN is active. A status of shutdown(3) indicates that the port has been disabled as a result of VQP shutdown response. Possible Values: inactive(1) active(2) shutdown(3)</td>
</tr>
<tr>
<td>NodeID</td>
<td>System.String</td>
<td>A SWIS-generated unique identifier of a network node in the current inventory. (Instances of this property recur in this table according to the number of interfaces for which bridge port data is reported.)</td>
</tr>
<tr>
<td>LastDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM last discovered the device during inventory.</td>
</tr>
<tr>
<td>FirstDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM first discovered the device during inventory.</td>
</tr>
</tbody>
</table>
NCM.BridgePorts entity relationships

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ENTITY</th>
<th>JOINED DATA ENTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>NCM.Nodes</td>
<td>NCM.NodeHostsBridgePorts (System.Hosting)</td>
</tr>
</tbody>
</table>

NCM.CatalystCards

<table>
<thead>
<tr>
<th>PROPERTY NAME</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CardIndex</td>
<td>System.Int32</td>
<td>A unique value for each module within the chassis.</td>
</tr>
</tbody>
</table>
Possible Values:
(notdefined(0), version1(1), version2(2), version3(3), version4(4), version5(5), version6(6), version7(7), version8(8), version9(9), version10(10), vi2(11), vi4(12), vi30(13), s1b(14), sa2(15), as1(16), new8as1(17), lsa(18), fxs2(19), fxo2(20), em2(21), fxs4(22), fxo4(23), em4(24), sab(25), e1vi(26), am12(27), am6(28), ndec(29), newsa2(30), aux(31), console(32), sic-wan(33), sic-1fe(34), sic-1sa(35), sic-3as(36), sic-1e1(37), sic-1t1(38), sic-1bu(39), sic-2bu(40), sic-1bs(41), sic-2bs(42), sic-1am(43), sic-2am(44), sic-1em(45), sic-2em(46), sic-1fxs(47), sic-2fxs(48), sic-1fxo(49), sic-2fxo(50), fcm6(51), sa8(52), t1153, t1254, t1455, t1vi56, fcm4(57), fcm2(58), rtb21ce(59), ame6(60), ame12(61), wsx5162(62), e11f(65), e12f(66), e14f(67), t11f(68), t12f(69), t14f(70), e11f17(71), t11f17(72), rtb21ct(73), atmads1(74), atmads2(75), atm155m(76), ase8(77), ase16(78), sae4(79), sae2(80), wsx5012a(81), wsx5167(82), wsx5168(83), wsx5305(85), wsx5550(87), wsfs541(88), atmshdsl1(90), atmshdsl2(91), atmshdsl4(92), atm25m(93), atm3e(94), atm3t(95), xdsl-fe(96), xdsl-ads(97), xdsl-gshds(98), xdsl-bri(99), xdsl-scc(100), ge1(101), pos155m(102), cpos(103), fe1op(104), sae8(105), atm155m-mm(106), atm155m-sm(107), atm155m-sml(108), fe1op-sfx(109), fe1op-mfx(110), cpos-t(111), ge1-op(112), ge2-op(113), ge2(114), fix-1wan(115), fix-1asae(116), cavium(117), sic-1eth(118), atm1ADSL(119), atm2ADSL(120), fix-e11(121), fix-t11(122), e18-75(123), e18-120(124), t18(125), sic-1vfxs(126), sic-1vfo(127), sic-2vfxs(128), sic-2vftx(129), xdsl-fec-new(130), xdsl-sa(131), bs4(132), ima8e175(133), ima-8e175(134), ima-4e175(135), ima-4e175(136), ima-175(137), ima-4t1(138), sic-1t1f(139), sic-1e1f(140), fe4(149), atm1shdsl4wire(151), atmima4shds(152), ls4(153), ls8(154), ls16(155), sic-adls2plus-isd(156), sic-adls2plus-isd(157), ft3(158), ce32(159), bsv2(160), bsv4(161), rpu(162), erpu(163), ssl(164), ns(165), wsx6ksup12ge(200), wsx6408gbic(201), wsx6224mmmt(202), wsx6248rj45(203), wsx6248tel(204), wsx6302sm(206), wsf6kmsfc(207), wsx6024fmlmt(208), wsx6101oc12mm(209), wsx6101oc12mhf(210), wsx6416gent(211), wsx61822pa(212), osm2oc12AtmMM(213), osm2oc12AtmSL(214), osm4oc12PosMM(216), osmm4oc12PosSL(217), osm4oc12PosSL(218), wsx6ksup1a2ge(219), fe18-75(220), fe18-120(221), ft8(222), cf-card(223), bsv2-v2(224), e1vi1-v2(225), e1v1v2(226), t1v1v2(227), t1v2v2(228), osm229, sd707(230), dm-epri(231), dm-trpri(232), erpu-h(233), wsf6kmsfc2(234), wsx6324mmmt(235), wsx6348rj(236), wsx6ksup22ge(237), wsx6324sm(238), wsx6516gbic(239), osm4geWanGbic(240), osm1oc48PosSS(241), osm1oc48PosSL(242), osm1oc48PosSL(243), wsx6381ids(244), wsc65050(245), osm16oc3PosMM(246), osm16oc3PosSL(247), osm16oc3PosSL(248), osm2oc12PosMM(249), osm2oc12PosSL(250), osm2oc12PosSL(251), wsx650210ge(252), osm8oc3PosMM(253), osm8oc3PosSL(254), osm8oc3PosSL(255), wsx6548rj45(258), wsx6524mmmt(259), wsx6066lbgac(260), wsx6516gbtc(261), osm2oc48oneDptSS(265), osm2oc48oneDptSL(266), osm2oc48oneDptSL(267), osm2oc48oneDptSSDual(268), osm2oc48oneDptSSDual(269), osm2oc48oneDptSSDual(270), wsx6816gbic(271), osm4hosc12T3MM.
<table>
<thead>
<tr>
<th>PROPERTY NAME</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CardName</td>
<td>System.String</td>
<td>A descriptive string used by the network administrator to name the module.</td>
</tr>
<tr>
<td>ModuleModel</td>
<td>System.String</td>
<td>The manufacturer's model number for the module.</td>
</tr>
<tr>
<td>CardSerial</td>
<td>System.String</td>
<td>The serial number of the module. This MIB object will return the module serial number for any module that either a numeric or an alphanumeric serial number is being used.</td>
</tr>
<tr>
<td>HWVersion</td>
<td>System.String</td>
<td>The hardware version of the module.</td>
</tr>
<tr>
<td>FWVersion</td>
<td>System.String</td>
<td>The firmware version of the module.</td>
</tr>
<tr>
<td>SWVersion</td>
<td>System.String</td>
<td>The software version of the module.</td>
</tr>
<tr>
<td>Slot</td>
<td>System.Int32</td>
<td>This value is determined by the chassis slot number where the module is located. Valid entries are 1 to the value of chassisNumSlots.</td>
</tr>
<tr>
<td>Parent</td>
<td>System.Int32</td>
<td>The value of the instance of the entPhysicalIndex object, defined in ENTITY-MIB, for the entity physical index corresponding to this module.</td>
</tr>
<tr>
<td>OperStatus</td>
<td>System.String</td>
<td>The operational status of the module. If the status is not ok, the value of moduleTestResult gives more detailed information about the module's failure condition(s). Possible Values: other(1), ok(2), minorFault(3), majorFault(4)</td>
</tr>
<tr>
<td>SlotsOnCard</td>
<td>System.Int32</td>
<td>The number of ports supported by the module.</td>
</tr>
<tr>
<td>NodeID</td>
<td>System.String</td>
<td>A SWIS-generated unique identifier of a network node in the current inventory. (Instances of this property recur in this table according to the number of interfaces for which card data is reported.)</td>
</tr>
<tr>
<td>LastDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM last discovered the device during inventory.</td>
</tr>
<tr>
<td>FirstDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM first discovered the device during inventory.</td>
</tr>
</tbody>
</table>

**NCM.CatalystCards entity relationships**

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ENTITY</th>
<th>JOINED DATA ENTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>NCM.Nodes</td>
<td>NCM.NodeHostsCatalystCards (System.Hosting)</td>
</tr>
<tr>
<td><strong>PROPERTY NAME</strong></td>
<td><strong>DATATYPE</strong></td>
<td><strong>DESCRIPTION</strong></td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------</td>
<td>----------------</td>
</tr>
<tr>
<td>CardIndex</td>
<td>System.Int32</td>
<td>Index into cardTable (not physical chassis slot number).</td>
</tr>
<tr>
<td>CardType</td>
<td>System.Int32</td>
<td>Functional type of this card. (integer value)</td>
</tr>
<tr>
<td>CardName</td>
<td>System.String</td>
<td>Functional type of this card. (Parsed from type name value).</td>
</tr>
<tr>
<td>CardDescr</td>
<td>System.String</td>
<td>Text description of this card.</td>
</tr>
<tr>
<td>CardSerial</td>
<td>System.String</td>
<td>The serial number of this card, or zero if unavailable.</td>
</tr>
<tr>
<td>HWVersion</td>
<td>System.String</td>
<td>Hardware revision level of this card, or an empty string if unavailable.</td>
</tr>
<tr>
<td>SWVersion</td>
<td>System.String</td>
<td>Version of the firmware or microcode installed on this card, or an empty string if unavailable.</td>
</tr>
<tr>
<td>Slot</td>
<td>System.Int32</td>
<td>Number of slots on this card, or 0 if no slots or not applicable, or -1 if not determinable.</td>
</tr>
<tr>
<td>Parent</td>
<td>System.Int32</td>
<td>(cardIndex of the parent card which directly contains this card, or 0 if contained by the chassis, or -1 if not applicable nor determinable.</td>
</tr>
</tbody>
</table>
| OperStatus        | System.String| The operational status of the card. cardOperStatus is up when a card is recognized by the device and is enabled for operation. cardOperStatus is down if the card is not recognized by the device, or if it is not enabled for operation. cardOperStatus is standby if the card is enabled and acting as a standby slave. Possible Values:  

  - not-specified(1)  
  - up (2)  
  - down (3)  
  - standby (4)  

  - standbyMaster (5)  
  - activeMaster (6)  
  - outOfService (7)  
  - masterBooting(8)  
  - activeMasterBooting(9)  
  - standbyMasterBooting(10)  
  - slaveBooting(11) |
<p>| SlotsOnCard       | System.Int32 | Number of slots on this card, or 0 if no slots or not applicable, or -1 if not determinable. |</p>
<table>
<thead>
<tr>
<th><strong>PROPERTY NAME</strong></th>
<th><strong>DATATYPE</strong></th>
<th><strong>DESCRIPTION</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>NodeID</td>
<td>System.String</td>
<td>A SWIS-generated unique identifier of a network node in the current inventory. (Instances of this property recur in this table according to the number of interfaces for which card data is reported.)</td>
</tr>
<tr>
<td>LastDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM last discovered the device during inventory.</td>
</tr>
<tr>
<td>FirstDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM first discovered the device during inventory.</td>
</tr>
</tbody>
</table>

**NCM.CiscoCards entity relationships**

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ENTITY</th>
<th>JOINED DATA ENTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>NCM.Nodes</td>
<td>NCM.NodeHostsCiscoCards (System.Hosting)</td>
</tr>
</tbody>
</table>

**NCM.CiscoCdp**

<table>
<thead>
<tr>
<th><strong>PROPERTY NAME</strong></th>
<th><strong>DATATYPE</strong></th>
<th><strong>DESCRIPTION</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>ifIndex</td>
<td>System.Int32</td>
<td>An indication of the type of address contained in the corresponding instance of cdpCacheAddress (parse just ifIndex from value for example 1,2,3)</td>
</tr>
<tr>
<td>CDPIndex</td>
<td>System.String</td>
<td>An indication of the type of address contained in the corresponding instance of cdpCacheAddress (full value. For example 1.6, 2.108, 2.3)</td>
</tr>
<tr>
<td>RemoteDevice</td>
<td>System.String</td>
<td>The Device-ID string as reported in the most recent CDP message. The zero-length string indicates no Device-ID field (TLV) was reported in the most recent CDP message.</td>
</tr>
<tr>
<td>RemoteIPAddress</td>
<td>System.String</td>
<td>The (first) network-layer address of the device's SNMP-agent as reported in the most recent CDP message. For example, if the corresponding instance of cacheAddressType had the value 'ip(1)' , then this object would be an IP address.</td>
</tr>
<tr>
<td>RemoteVersion</td>
<td>System.String</td>
<td>The Version string as reported in the most recent CDP message. The zero-length string indicates no Version field (TLV) was reported in the most recent CDP message.</td>
</tr>
<tr>
<td>RemotePort</td>
<td>System.String</td>
<td>The Port-ID string as reported in the most recent CDP message. This will typically be the value of the ifName object (e.g., 'Ethernet0'). The zero-length string indicates no Port-ID field (TLV) was reported in the most recent CDP message</td>
</tr>
<tr>
<td>PROPERTY NAME</td>
<td>DATATYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RemoteCapability</td>
<td>System.String</td>
<td>The Device's Functional Capabilities as reported in the most recent CDP message. For latest set of specific values, see the latest version of the CDP specification. The zero-length string indicates no Capabilities field (TLV) was reported in the most recent CDP message.</td>
</tr>
<tr>
<td>RemotePlatform</td>
<td>System.String</td>
<td>The Device's Hardware Platform as reported in the most recent CDP message. The zero-length string indicates that no Platform field (TLV) was reported in the most recent CDP message.</td>
</tr>
<tr>
<td>RemoteDuplex</td>
<td>System.String</td>
<td>The remote device's interface's duplex mode, as reported in the most recent CDP message. The value unknown(1) indicates no duplex mode field (TLV) was reported in the most recent CDP message. Possible Values: unknown(1) halfduplex(2) fullduplex(3)</td>
</tr>
<tr>
<td>RemoteNativeVLAN</td>
<td>System.Int32</td>
<td>The remote device's interface's native VLAN, as reported in the most recent CDP message. The value 0 indicates no native VLAN field (TLV) was reported in the most recent CDP message.</td>
</tr>
<tr>
<td>NodeID</td>
<td>System.String</td>
<td>A SWIS-generated unique identifier of a network node in the current inventory. (Instances of this property recur in this table according to the number of remote devices for which remote device data is reported.)</td>
</tr>
<tr>
<td>LastDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM last discovered the device during inventory.</td>
</tr>
<tr>
<td>FirstDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM first discovered the device during inventory.</td>
</tr>
</tbody>
</table>

**NCM.CiscoCdp entity relationships**

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ENTITY</th>
<th>JOINED DATA ENTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>NCM.Nodes</td>
<td>NCM.NodeHostsCiscoCdp (System.Hosting)</td>
</tr>
<tr>
<td>PROPERTY_NAME</td>
<td>DATATYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>chassisType</td>
<td>System.Int32</td>
<td>Chassis type (integer value). Possible Values: unknown(1), multibus(2), agsplus(3)</td>
</tr>
<tr>
<td>chassisName</td>
<td>System.String</td>
<td>Chassis type (parsed string value). Possible Values: unknown(1), multibus(2), agsplus(3)</td>
</tr>
<tr>
<td>chassisVersion</td>
<td>System.String</td>
<td>Chassis hardware revision level, or an empty string if unavailable.</td>
</tr>
<tr>
<td>chassisID</td>
<td>System.String</td>
<td>Unique ID string. Defaults to chassis serial number if available, otherwise empty. Can also be set with 'snmp-server chassis-id'.</td>
</tr>
<tr>
<td>chassisSerialNumberString</td>
<td>System.String</td>
<td>The serial number of the chassis. This MIB object will return the chassis serial number for any chassis that either a numeric or an alphanumeric serial number is being used.</td>
</tr>
<tr>
<td>RomSysVersion</td>
<td>System.String</td>
<td>ROM system software version or an empty string if unavailable.</td>
</tr>
<tr>
<td>processorRAM</td>
<td>System.Int32</td>
<td>Bytes of RAM available to CPU.</td>
</tr>
<tr>
<td>nvRAMSize</td>
<td>System.Int32</td>
<td>Bytes of nonvolatile configuration memory.</td>
</tr>
<tr>
<td>nvRAMUsed</td>
<td>System.Int32</td>
<td>Bytes of non-volatile configuration memory in use.</td>
</tr>
<tr>
<td>chassisSlots</td>
<td>System.Int32</td>
<td>Number of slots in this chassis, or -1 of neither applicable nor determinable.</td>
</tr>
<tr>
<td>romID</td>
<td>System.String</td>
<td>This variable contains a printable octet string which contains the System Bootstrap description and version identification.</td>
</tr>
<tr>
<td>whyReload</td>
<td>System.String</td>
<td>This variable contains a printable octet string which contains the reason why the system was last restarted.</td>
</tr>
</tbody>
</table>
freeMem | System.Int32 | Return the amount of free memory in bytes.
Note: This MIB object is obsolete as of IOS release 11.1. IOS release 11.1 introduced the CISCO-MEMORY-POOL-MIB which better instruments all of the memory pools.

NodeID | System.String | A SWIS-generated unique identifier of a network node in the current inventory. (Instances of this property recur in this table according to the number of nodes for which data is reported.)

LastDiscovery | System.DateTime | A SWIS-generated date and time marker for when NCM last discovered the device during inventory.

FirstDiscovery | System.DateTime | A SWIS-generated date and time marker for when NCM first discovered the device during inventory.

NCM.CiscoChassis entity relationships

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ENTITY</th>
<th>JOINED DATA ENTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>NCM.Nodes</td>
<td>NCM.NodeHostsCiscoChassis (System.Hosting)</td>
</tr>
</tbody>
</table>

NCM.CiscoFlash

<table>
<thead>
<tr>
<th>PROPERTY NAME</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>FlashSize</td>
<td>System.Int32</td>
<td>Total size of the Flash device. For a removable device, the size will be zero if the device has been removed.</td>
</tr>
<tr>
<td>Name</td>
<td>System.String</td>
<td>Flash device name. This name is used to refer to the device within the system. Flash operations get directed to a device based on this name. The system has a concept of a default device. This would be the primary or most used device in case of multiple devices. The system directs an operation to the default device whenever a device name is not specified. The device name is therefore mandatory except when the operation is being done on the default device, or the system supports only a single Flash device. The device name will always be available for a removable device, even when the device has been removed.</td>
</tr>
<tr>
<td>Property Name</td>
<td>Datatype</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>FlashDescription</td>
<td>System.String</td>
<td>Description of a Flash device. The description is meant to explain what the Flash device and its purpose is. Current values are: System flash - for the primary Flash used to store full system images. Boot flash: for the secondary Flash used to store bootstrap images. The ciscoFlashDeviceDescr, CiscoFlashDeviceController (if applicable), and ciscoFlashPhyEntIndex objects are expected to collectively give all information about a Flash device. The device description will always be available for a removable device, even when the device has been removed.</td>
</tr>
<tr>
<td>PartitionCount</td>
<td>System.Int32</td>
<td>Flash device partitions actually present. Number of partitions cannot exceed the minimum of ciscoFlashDeviceMaxPartitions and (ciscoFlashDeviceSize / ciscoFlashDeviceMinPartitionSize). Will be equal to at least 1, the case where the partition spans the entire device (actually no partitioning). A partition will contain one or more minimum partition units (where a minimum partition unit is defined by ciscoFlashDeviceMinPartitionSize).</td>
</tr>
<tr>
<td>MinPartitionSize</td>
<td>System.Int32</td>
<td>This object will give the minimum partition size supported for this device. For systems that execute code directly out of Flash, the minimum partition size needs to be the bank size. (Bank size is equal to the size of a chip multiplied by the width of the device. In most cases, the device width is 4 bytes, and so the bank size would be four times the size of a chip). This has to be so because all programming commands affect the operation of an entire chip (in our case, an entire bank because all operations are done on the entire width of the device) even though the actual command may be localized to a small portion of each chip. So when executing code out of Flash, one needs to be able to write and erase some portion of Flash without affecting the code execution. For systems that execute code out of DRAM or ROM, it is possible to partition Flash with a finer granularity (for eg., at erase sector boundaries) if the system code supports such granularity. This object will let a management entity know the minimum partition size as defined by the system. If the system does not support partitioning, the value will be equal to the device size in ciscoFlashDeviceSize. The maximum number of partitions that could be configured will be equal to the minimum of ciscoFlashDeviceMaxPartitions and (ciscoFlashDeviceSize / CiscoFlashDeviceMinPartitionSize)</td>
</tr>
<tr>
<td>Property Name</td>
<td>Datatype</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Controller</td>
<td>System.String</td>
<td>Flash device controller. The h/w card that actually controls Flash read/write/erase. Relevant for the AGS+ systems where Flash may be controlled by the MC+, STR or the ENVM cards, cards that may not actually contain the Flash chips. For systems that have removable PCMCIA flash cards that are controlled by a PCMCIA controller chip, this object may contain a description of that controller chip. Where irrelevant (Flash is a direct memory mapped device accessed directly by the main processor), this object will have an empty (NULL) string.</td>
</tr>
<tr>
<td>WriteProtectJumper</td>
<td>System.String</td>
<td>This object gives the state of a jumper (if present and can be determined) that controls the programming voltage called Vpp to the Flash device. Vpp is required for programming (erasing and writing) Flash. For certain older technology chips it is also required for identifying the chips (which in turn is required to identify which programming algorithms to use; different chips require different algorithms and commands). The purpose of the jumper, on systems where it is available, is to write protect a Flash device. On most of the newer remote access routers, this jumper is unavailable since users are not expected to visit remote sites just to install and remove the jumpers when upgrading software in the Flash device. The unknown(3) value will be returned for such systems and can be interpreted to mean that a programming jumper is not present or not required on those systems. On systems where the programming jumper state can be read back through a hardware register, the installed (1) or notInstalled (2) value will be returned. This object is expected to be used in conjunction with the ciscoFlashPartitionStatus object whenever that object has the readOnly(1) value. In such a case, this object will indicate whether the programming jumper is a possible reason for the readOnly state. Possible Values: installed(1) notInstalled(2) unknown(3)</td>
</tr>
<tr>
<td>Property Name</td>
<td>Datatype</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MaxPartitions</td>
<td>System.Int32</td>
<td>Max number of partitions supported by the system for this Flash device. Default will be 1, which actually means that partitioning is not supported. Note that this value will be defined by system limitations, not by the flash device itself (for eg., the system may impose a limit of 2 partitions even though the device may be large enough to be partitioned into 4 based on the smallest partition unit supported). On systems that execute code out of Flash, partitioning is a way of creating multiple file systems in the Flash device so that writing into or erasing of one file system can be done while executing code residing in another file system. For systems executing code out of DRAM, partitioning gives a way of sub-dividing a large Flash device for easier management of files.</td>
</tr>
<tr>
<td>Initialized</td>
<td>System.DateTime</td>
<td>System time at which device was initialized. For fixed devices, this will be the system time at boot up. For removable devices, it will be the time at which the device was inserted, which may be boot up time, or a later time (if device was inserted later). If a device (fixed or removable) was repartitioned, it will be the time of repartitioning. The purpose of this object is to help a management station determine if a removable device has been changed. The application should retrieve this object prior to any operation and compare with the previously retrieved value. Note that this time will not be real time but a running time maintained by the system. This running time starts from zero when the system boots up. For a removable device that has been removed, this value will be zero.</td>
</tr>
<tr>
<td>Removable</td>
<td>System.String</td>
<td>Whether Flash device is removable. Generally, only PCMCIA Flash cards will be treated as removable. Socketed Flash chips and Flash SIMM modules will not be treated as removable. Simply put, only those Flash devices that can be inserted or removed without opening the hardware casing will be considered removable. Further, removable Flash devices are expected to have the necessary hardware support: 1) on-line removal and insertion; 2) interrupt generation on removal or insertion.</td>
</tr>
<tr>
<td>NodeID</td>
<td>System.String</td>
<td>A SWIS-generated unique identifier of a network node in the current inventory. (Instances of this property recur in this table according to the number of Cisco devices for which data is reported.)</td>
</tr>
<tr>
<td>LastDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM last discovered the device during inventory.</td>
</tr>
</tbody>
</table>
**PROPERTY NAME** | **DATATYPE** | **DESCRIPTION**
--- | --- | ---
FirstDiscovery | System.DateTime | A SWIS-generated date and time marker for when NCM first discovered the device during inventory.

**NCM.CiscoFlash entity relationships**

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ENTITY</th>
<th>JOINED DATA ENTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>NCM.Nodes</td>
<td>NCM.NodeHostsCiscoFlash (System.Hosting)</td>
</tr>
</tbody>
</table>

**NCM.CiscoFlashFiles**

<table>
<thead>
<tr>
<th>PROPERTY NAME</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>FlashFileName</td>
<td>System.String</td>
<td>Flash file name as specified by the user copying in the file. The name should not include the colon (:) character as it is a special separator character used to delineate the device name, partition name, and the file name.</td>
</tr>
<tr>
<td>FlashFileSize</td>
<td>System.Int32</td>
<td>Size of the file in bytes. Note that this size does not include the size of the filesystem file header. File size will always be non-zero.</td>
</tr>
<tr>
<td>FlashCheckSum</td>
<td>System.String</td>
<td>File checksum stored in the file header. This checksum is computed and stored when the file is written into Flash. It serves to validate the data written into Flash. Whereas the system will generate and store the checksum internally in hexadecimal form, this object will provide the checksum in a string form. The checksum will be available for all valid and invalid-checksum files.</td>
</tr>
</tbody>
</table>
| FlashFileStatus | System.String | Status of a file. A file could be explicitly deleted if the file system supports such a user command facility. Alternately, an existing good file would be automatically deleted if another good file with the same name were copied in. Note that deleted files continue to occupy prime Flash real estate. A file is marked as having an invalid checksum if any checksum mismatch was detected while writing or reading the file. Incomplete files (files truncated either because of lack of free space or a network download failure) are also written with a bad checksum and marked as invalid. Possible Values: 
- deleted(1) 
- invalidChecksum(2) 
- valid(3) |
<table>
<thead>
<tr>
<th><strong>PROPERTY NAME</strong></th>
<th><strong>DATATYPE</strong></th>
<th><strong>DESCRIPTION</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>NodeID</td>
<td>System.String</td>
<td>A SWIS-generated unique identifier of a network node in the current inventory. (Instances of this property recur in this table according to the number of Cisco devices for which data is reported.)</td>
</tr>
<tr>
<td>LastDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM last discovered the device during inventory.</td>
</tr>
<tr>
<td>FirstDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM first discovered the device during inventory.</td>
</tr>
</tbody>
</table>

**NCM.CiscoFlashFiles entity relationships**

<table>
<thead>
<tr>
<th><strong>TYPE</strong></th>
<th><strong>ENTITY</strong></th>
<th><strong>JOINED DATA ENTITY</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>NCM.Nodes</td>
<td>NCM.NodeHostsCiscoFlashFiles (System.Hosting)</td>
</tr>
</tbody>
</table>

**NCM.CiscoImageMIB**

<table>
<thead>
<tr>
<th><strong>PROPERTY NAME</strong></th>
<th><strong>DATATYPE</strong></th>
<th><strong>DESCRIPTION</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>System.String</td>
<td>The string of this entry.</td>
</tr>
<tr>
<td>Value</td>
<td>System.String</td>
<td>The string of this entry.</td>
</tr>
<tr>
<td>NodeID</td>
<td>System.String</td>
<td>A SWIS-generated unique identifier of a network node in the current inventory. (Instances of this property recur in this table according to the number of Cisco devices for which data is reported.)</td>
</tr>
<tr>
<td>LastDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM last discovered the device during inventory.</td>
</tr>
<tr>
<td>FirstDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM first discovered the device during inventory.</td>
</tr>
</tbody>
</table>

**NCM.CiscoImageMIB entity relationships**

<table>
<thead>
<tr>
<th><strong>TYPE</strong></th>
<th><strong>ENTITY</strong></th>
<th><strong>JOINED DATA ENTITY</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>NCM.Nodes</td>
<td>NCM.NodeHostsCiscoImageMIB (System.Hosting)</td>
</tr>
</tbody>
</table>
### NCM.CiscoMemoryPools

<table>
<thead>
<tr>
<th>PROPERTY NAME</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>PoolName</td>
<td>System.String</td>
<td>A textual name assigned to the memory pool. This object is suitable for output to a human operator, and may also be used to distinguish among the various pool types, especially among dynamic pools.</td>
</tr>
<tr>
<td>PoolUsed</td>
<td>System.Int32</td>
<td>Indicates the number of bytes from the memory pool that are currently in use by applications on the managed device.</td>
</tr>
<tr>
<td>PoolFree</td>
<td>System.Int32</td>
<td>Indicates the number of bytes from the memory pool that are currently unused on the managed device. Note that the sum of ciscoMemoryPoolUsed and ciscoMemoryPoolFree is the total amount of memory in the pool.</td>
</tr>
<tr>
<td>PoolLargestFree</td>
<td>System.Int32</td>
<td>Indicates the largest number of contiguous bytes from the memory pool that are currently unused on the managed device.</td>
</tr>
<tr>
<td>NodeID</td>
<td>System.String</td>
<td>A SWIS-generated unique identifier of a network node in the current inventory. (Instances of this property recur in this table according to the number of Cisco devices for which data is reported.)</td>
</tr>
<tr>
<td>LastDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM last discovered the device during inventory.</td>
</tr>
<tr>
<td>FirstDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM first discovered the device during inventory.</td>
</tr>
</tbody>
</table>

### NCM.CiscoMemoryPools entity relationships

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ENTITY</th>
<th>JOINED DATA ENTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>NCM.Nodes</td>
<td>NCM.NodeHostsCiscoMemoryPools (System.Hosting)</td>
</tr>
</tbody>
</table>

### NCM.EntityLogical

<table>
<thead>
<tr>
<th>PROPERTY NAME</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>System.String</td>
<td>A textual description of the logical entity. This object should contain a string which identifies the manufacturer's name for the logical entity, and should be set to a distinct value for each version of the logical entity.</td>
</tr>
<tr>
<td>TDomain</td>
<td>System.String</td>
<td>Indicates the kind of transport service by which the logical entity receives network management traffic. Possible values for this object are presently found in the Transport Mappings for SNMPv2 document (RFC 1906 [RFC1906]).</td>
</tr>
<tr>
<td>Property Name</td>
<td>Datatype</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Type</td>
<td>System.String</td>
<td>An indication of the type of logical entity. This will typically be the OBJECT-IDENTIFIER name of the node in the SMI's naming hierarchy which represents the major MIB module, or the majority of the MIB modules, supported by the logical entity. For example: a logical entity of a regular host/router &gt; mib-2 a logical entity of a 802.1d bridge -&gt; dot1dBridge a logical entity of a 802.3 repeater -&gt; snmpDot3RptrMgmt If an appropriate node in the naming hierarchy of SMI cannot be identified, the value 'mib-2' should be used.</td>
</tr>
<tr>
<td>Community</td>
<td>System.String</td>
<td>An SNMPv1 or SNMPv2C community-string which can be used to access detailed management information for this logical entity. The agent should allow read access with this community string (to an appropriate subset of all managed objects) and may also return a community string based on the privileges of the request used to read this object. Note that an agent may return a community string with read-only privileges, even if this object is accessed with a read-write community string. However, the agent must take care not to return a community string which allows more privileges than the community string used to access this object.</td>
</tr>
<tr>
<td>TAddress</td>
<td>System.String</td>
<td>The transport service address by which the logical entity receives network management traffic, formatted according to the corresponding value of entLogicalTDomain. For snmpUDPDomain, a TAddress is 6 octets long, the initial 4 octets containing the IP-address in network-byte order and the last 2 containing the UDP port in network-byte order. Consult 'Transport Mappings for Version 2 of the Simple Network Management Protocol' (RFC 1906 [RFC1906]) for further information on snmpUDPDomain.</td>
</tr>
<tr>
<td>NodeID</td>
<td>System.String</td>
<td>A SWIS-generated unique identifier of a network node in the current inventory. (Instances of this property recur in this table according to the number of Cisco devices for which data is reported.)</td>
</tr>
<tr>
<td>LastDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM last discovered the device during inventory.</td>
</tr>
<tr>
<td>FirstDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM first discovered the device during inventory.</td>
</tr>
</tbody>
</table>

**NCM.EntityLogical entity relationships**

<table>
<thead>
<tr>
<th>Type</th>
<th>Entity</th>
<th>Joined Data Entity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>NCM.Nodes</td>
<td>NCM.NodeHostsEntityLogical (System.Hosting)</td>
</tr>
<tr>
<td>PROPERTY NAME</td>
<td>DATATYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EntityName</td>
<td>System.String</td>
<td>The textual name of the physical entity. The value of this object should be the name of the component as assigned by the local device and should be suitable for use in commands entered at the device's <code>console</code>. This might be a text name, such as <code>console</code> or a simple component number (e.g. port or module number), such as <code>1</code>, depending on the physical component naming syntax of the device. If there is no local name, or this object is otherwise not applicable, then this object contains a zero-length string. Note: The value of entPhysicalName for two physical entities will be the same in the event that the console interface does not distinguish between them, e.g., slot-1 and the card in slot-1.</td>
</tr>
<tr>
<td>EntityDescription</td>
<td>System.String</td>
<td>A textual description of physical entity. This object should contain a string which identifies the manufacturer's name for the physical entity, and should be set to a distinct value for each version or model of the physical entity.</td>
</tr>
<tr>
<td>EntityType</td>
<td>System.String</td>
<td>An indication of the vendor-specific hardware type of the physical entity. Note that this is different from the definition of MIB-II's sysObjectID. An agent should set this object to a enterprise-specific registration identifier value indicating the specific equipment type in detail. The associated instance of entPhysicalClass is used to indicate the general type of hardware device. If no vendor-specific registration identifier exists for this physical entity, or the value is unknown by this agent then the value {0} is returned.</td>
</tr>
<tr>
<td><strong>Property Name</strong></td>
<td><strong>Datatype</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ContainedIn</td>
<td>System.String</td>
<td>The value of entPhysicalIndex for the physical entity which 'contains' this physical entity. A value of zero indicates this physical entity is not contained in any other physical entity. Note that the set of 'containment' relationships define a strict hierarchy; that is, recursion is not allowed. In the event a physical entity is contained by more than one physical entity (e.g., double-wide modules), this object should identify the containing entity with the lowest value of entPhysicalIndex.</td>
</tr>
<tr>
<td>EntityClass</td>
<td>System.String</td>
<td>An indication of the general hardware type of the physical entity. An agent should set this object to the standard enumeration value which most accurately indicates the general class of the physical entity or the primary class if there is more than one. If no appropriate standard registration identifier exists for this physical entity, then the value 'other(1)' is returned. If the value is unknown by this agent, then the value 'unknown(2)' is returned.</td>
</tr>
<tr>
<td>Position</td>
<td>System.Int32</td>
<td>An indication of the relative position of this 'child' component among all its 'sibling' components. Sibling components are defined as entPhysicalEntries which share the same instance values of each of the entPhysicalContainedIn and entPhysicalClass objects.</td>
</tr>
<tr>
<td>HardwareRevision</td>
<td>System.String</td>
<td>The vendor-specific hardware revision string for the physical entity. The preferred value is the hardware revision identifier actually printed on the component itself (if present). Note that if revision information is stored internally in a non-printable (e.g., binary) format, then the agent must convert such information to a printable format, in an implementation-specific manner. If no specific hardware revision string is associated with the physical component, or this information is unknown to the agent, then this object will contain a zero-length string.</td>
</tr>
<tr>
<td><strong>PROPERTY NAME</strong></td>
<td><strong>DATATYPE</strong></td>
<td><strong>DESCRIPTION</strong></td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>FirmwareRevision</td>
<td>System.String</td>
<td>The vendor-specific firmware revision string for the physical entity. Note that if revision information is stored internally in a non-printable (e.g., binary) format, then the agent must convert such information to a printable format, in an implementation-specific manner. If no specific firmware programs are associated with the physical component, or this information is unknown to the agent, then this object will contain a zero-length string.</td>
</tr>
<tr>
<td>SoftwareRevision</td>
<td>System.String</td>
<td>The vendor-specific software revision string for the physical entity. Note that if revision information is stored internally in a non-printable (e.g., binary) format, then the agent must convert such information to a printable format, in an implementation-specific manner. If no specific software programs are associated with the physical component, or this information is unknown to the agent, then this object will contain a zero-length string.</td>
</tr>
<tr>
<td>Serial</td>
<td>System.String</td>
<td>The vendor-specific serial number string for the physical entity. The preferred value is the serial number string actually printed on the component itself (if present).</td>
</tr>
<tr>
<td>Manufacturer</td>
<td>System.String</td>
<td>The name of the manufacturer of this physical component. The preferred value is the manufacturer name string actually printed on the component itself (if present).</td>
</tr>
<tr>
<td>Model</td>
<td>System.String</td>
<td>The vendor-specific model name identifier string associated with this physical component. The preferred value is the customer-visible part number, which may be printed on the component itself. If the model name string associated with the physical component is unknown to the agent, then this object will contain a zero-length string.</td>
</tr>
<tr>
<td>Alias</td>
<td>System.String</td>
<td>This object is an 'alias' name for the physical entity as specified by a network manager, and provides a non-volatile 'handle' for the physical entity.</td>
</tr>
<tr>
<td>AssetID</td>
<td>System.String</td>
<td>This object is a user-assigned asset tracking identifier for the physical entity as specified by a network manager, and provides non-volatile storage of this information.</td>
</tr>
<tr>
<td>Property Name</td>
<td>Datatype</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>FieldReplaceable</td>
<td>System.String</td>
<td>This object indicates whether or not this physical entity is considered a 'field replaceable unit' by the vendor. If this object contains the value 'true(1)' then this entPhysicalEntry identifies a field replaceable unit. For all entPhysicalEntries which represent components that are permanently contained within a field replaceable unit, the value 'false(2)' should be returned for this object.</td>
</tr>
<tr>
<td>NodeID</td>
<td>System.String</td>
<td>A SWIS-generated unique identifier of a network node in the current inventory. (Instances of this property recur in this table according to the number of Cisco devices for which data is reported.)</td>
</tr>
<tr>
<td>LastDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM last discovered the device during inventory.</td>
</tr>
<tr>
<td>FirstDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM first discovered the device during inventory.</td>
</tr>
</tbody>
</table>

NCM.EntityPhysical entity relationships

<table>
<thead>
<tr>
<th>Type</th>
<th>Entity</th>
<th>Joined Data Entity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>NCM.Nodes</td>
<td>NCM.NodeHostsEntityPhysical (System.Hosting)</td>
</tr>
</tbody>
</table>

NCM. Interfaces

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Datatype</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InterfaceID</td>
<td>System.String</td>
<td>[Swis]</td>
</tr>
<tr>
<td>InterfaceIndex</td>
<td>System.Int32</td>
<td>A unique value, greater than zero, for each interface. It is recommended that values are assigned contiguously starting from 1. The value for each interface sub-layer must remain constant at least from one reinitialization of the entity's network management system to the next reinitialization.</td>
</tr>
<tr>
<td>InterfaceDescription</td>
<td>System.String</td>
<td>A textual string containing information about the interface. This string should include the name of the manufacturer, the product name and the version of the interface hardware/software.</td>
</tr>
<tr>
<td>Property Name</td>
<td>Datatype</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>VLANID</td>
<td>System.Int32</td>
<td>The set of the device's member ports that belong to the VLAN. Each octet within the value of this object specifies a set of eight ports, with the first octet specifying ports 1 through 8, the second octet specifying ports 9 through 16, etc. Within each octet, the most significant bit represents the lowest numbered port, and the least significant bit represents the highest numbered port. Thus, each port of the VLAN is represented by a single bit within the value of this object. If that bit has a value of '1' then that port is included in the set of ports; the port is not included if its bit has a value of '0'. A port number is the value of dot1dBasePort for the port in the BRIDGE-MIB (RFC 1493).</td>
</tr>
<tr>
<td>PortStatus</td>
<td>System.String</td>
<td>An indication of the current VLAN status of the port. A status of inactive(1) indicates that a dynamic port does not yet have a VLAN assigned, or a port is assigned to a VLAN that is currently not active. A status of active(2) indicates that the currently assigned VLAN is active. A status of shutdown(3) indicates that the port has been disabled as a result of VQP shutdown response. Possible Values: inactive(1) active(2) shutdown(3)</td>
</tr>
<tr>
<td>VLANType</td>
<td>System.Int32</td>
<td>The type of this VLAN</td>
</tr>
<tr>
<td>InterfaceName</td>
<td>System.String</td>
<td>The textual name of the interface. The value of this object should be the name of the interface as assigned by the local device and should be suitable for use in commands entered at the device's <code>console</code>. This might be a text name, such as <code>le0</code> or a simple port number, such as <code>1</code>, depending on the interface naming syntax of the device. If several entries in the ifTable together represent a single interface as named by the device, then each will have the same value of ifName. If there is no local name, or this object is otherwise not applicable, then this object contains a 0-length string.</td>
</tr>
<tr>
<td>InterfaceAlias</td>
<td>System.String</td>
<td>This object is an 'alias' name for the interface as specified by a network manager, and provides a non-volatile 'handle' for the interface.</td>
</tr>
<tr>
<td><strong>PROPERTY NAME</strong></td>
<td><strong>DATATYPE</strong></td>
<td><strong>DESCRIPTION</strong></td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>InterfaceType</td>
<td>System.Int32</td>
<td>The type of interface. Additional values for ifType are assigned by the Internet Assigned Numbers Authority (IANA), through updating the syntax of the IANAifType textual convention.</td>
</tr>
<tr>
<td>InterfaceTypeName</td>
<td>System.String</td>
<td>The type of interface. Additional values for ifType are assigned by the Internet Assigned Numbers Authority (IANA), through updating the syntax of the IANAifType textual convention.</td>
</tr>
<tr>
<td>InterfaceTypeDescription</td>
<td>System.String</td>
<td>The type of interface. Additional values for ifType are assigned by the Internet Assigned Numbers Authority (IANA), through updating the syntax of the IANAifType textual convention.</td>
</tr>
<tr>
<td>InterfaceSpeed</td>
<td>System.Single</td>
<td>An estimate of the interface's current bandwidth in bits per second. For interfaces which do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth. If the bandwidth of the interface is greater than the maximum value reportable by this object then this object should report its maximum value (4, 294, 967, 295) and ifHighSpeed must be used to report the interface's speed. For a sub-layer which has no concept of bandwidth, this object should be zero.</td>
</tr>
<tr>
<td>MACAddress</td>
<td>System.String</td>
<td>The interface's address at its protocol sublayer. The interface's media specific MIB must define the bit and byte ordering and format of the value contained by this object. For interfaces which do not have such an address (e.g. a serial line), this object should contain an octet string of zero length.</td>
</tr>
<tr>
<td>AdminStatus</td>
<td>System.String</td>
<td>The desired state of the interface. The testing(3) state indicates that no operational packets can be passed. When a managed system initializes, all interfaces start with ifAdminStatus in the down(2) state. As a result of either explicit management action or per configuration information retained by the managed system, ifAdminStatus is then changed to either the up (1) or testing (3) states (or remains in the down (2) state).</td>
</tr>
<tr>
<td>PROPERTY NAME</td>
<td>DATATYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>OperStatus</td>
<td>System.String</td>
<td>The current operational state of the interface. The testing (3) state indicates that no operational packets can be passed. If ifAdminStatus is down (2) then ifOperStatus should be down (2). If ifAdminStatus is changed to up (1) then ifOperStatus should change to up (1) if the interface is ready to transmit and receive network traffic; it should change to dormant (5) if the interface is waiting for external actions (such as a serial line waiting for an incoming connection); it should remain in the down (2) state if and only if there is a fault that prevents it from going to the up (1) state.</td>
</tr>
<tr>
<td>InterfaceMTU</td>
<td>System.Int32</td>
<td>The size of the largest packet which can be sent/received on the interface, specified in octets. For interfaces that are used for transmitting network datagrams, this is the size of the largest network datagram that can be sent on the interface.</td>
</tr>
<tr>
<td>LastChange</td>
<td>System.DateTime</td>
<td>The value of sysUpTime at the time the interface entered its current operational state. If the current state was entered prior to the last re-initialization of the local network management subsystem, then this object contains a zero value.</td>
</tr>
<tr>
<td>PhysicalInterface</td>
<td>System.Char</td>
<td>This object has the value 'true(1)' if the interface sublayer has a physical connector and the value 'false(2)' otherwise.</td>
</tr>
<tr>
<td>Promiscuous</td>
<td>System.Char</td>
<td>This object has a value of false (2) if this interface only accepts packets/frames that are addressed to this station. This object has a value of true (1) when the station accepts all packets/frames transmitted on the media. The value true (1) is only legal on certain types of media. If legal, setting this object to a value of true (1) may require the interface to be reset before becoming effective.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The value of ifPromiscuousMode does not affect the reception of broadcast and multicast packets/frames by the interface.</td>
</tr>
<tr>
<td>NodeID</td>
<td>System.String</td>
<td>A SWIS-generated unique identifier of a network node in the current inventory. (Instances of this property recur in this table according to the number of Cisco devices for which data is reported.)</td>
</tr>
<tr>
<td>LastDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM last discovered the device during inventory.</td>
</tr>
<tr>
<td>PROPERTY NAME</td>
<td>DATATYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>FirstDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM first discovered the device during inventory.</td>
</tr>
</tbody>
</table>

**NCM.Interfaces entity relationships**

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ENTITY</th>
<th>JOINED DATA ENTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>NCM.Nodes</td>
<td>NCM.NodeHostsInterfaces (System.Hosting)</td>
</tr>
<tr>
<td></td>
<td>IpAddresses</td>
<td>NCM.InterfaceHostsIpAddresses (System.Hosting)</td>
</tr>
</tbody>
</table>

**NCM.IpAddresses**

<table>
<thead>
<tr>
<th>PROPERTY NAME</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>InterfaceIndex</td>
<td>System.Int32</td>
<td>The index value which uniquely identifies the interface to which this entry is applicable. The interface identified by a particular value of this index is the same interface as identified by the same value of RFC 1573's ifIndex.</td>
</tr>
<tr>
<td>IPAddress</td>
<td>System.String</td>
<td>The IP address to which this entry's addressing information pertains.</td>
</tr>
<tr>
<td>IPAddrIPSort</td>
<td>System.Double</td>
<td>Store IP address in double representation</td>
</tr>
<tr>
<td>SubnetMask</td>
<td>System.String</td>
<td>The subnet mask associated with the IP address of this entry. The value of the mask is an IP address with all the network bits set to 1 and all the hosts bits set to 0.</td>
</tr>
<tr>
<td>InterfaceID</td>
<td>System.String</td>
<td>InterfaceId from interfaces table</td>
</tr>
<tr>
<td>NodeID</td>
<td>System.String</td>
<td>A SWIS-generated unique identifier of a network node in the current inventory. (Instances of this property recur in this table according to the number of Cisco devices for which data is reported.)</td>
</tr>
<tr>
<td>LastDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM last discovered the device during inventory.</td>
</tr>
<tr>
<td>FirstDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM first discovered the device during inventory.</td>
</tr>
</tbody>
</table>

**NCM.IpAddresses entity relationships**

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ENTITY</th>
<th>JOINED DATA ENTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interfaces</td>
<td>NCM.Interfaces</td>
<td>NCM.InterfaceHostsIpAddresses (System.Hosting)</td>
</tr>
</tbody>
</table>
### NCM.MacForwarding

<table>
<thead>
<tr>
<th>PROPERTY NAME</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>System.Int32</td>
<td>Either the value '0', or the port number of the port on which a frame having a source address equal to the value of the corresponding instance of dot1dTpFdbAddress has been seen. A value of '0' indicates that the port number has not been learned but that the bridge does have some forwarding/filtering information about this address (e.g. in the dot1dStaticTable). Implementers are encouraged to assign the port value to this object whenever it is learned even for addresses for which the corresponding value of dot1dTpFdbStatus is not learned (3).</td>
</tr>
<tr>
<td>MAC</td>
<td>System.String</td>
<td>A unicast MAC address for which the bridge has forwarding and/or filtering information.</td>
</tr>
<tr>
<td>Source</td>
<td>System.String</td>
<td>The status of this entry. The meanings of the values are: other(1): none of the following. This would include the case where some other MIB object (not the corresponding instance of dot1dTpFdbPort, nor an entry in the dot1dStaticTable) is being used to determine if and how frames addressed to the value of the corresponding instance of dot1dTpFdbAddress are being forwarded.</td>
</tr>
<tr>
<td>NodeID</td>
<td>System.String</td>
<td>A SWIS-generated unique identifier of a network node in the current inventory. (Instances of this property recur in this table according to the number of Cisco devices for which data is reported.)</td>
</tr>
<tr>
<td>LastDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM last discovered the device during inventory.</td>
</tr>
<tr>
<td>FirstDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM first discovered the device during inventory.</td>
</tr>
</tbody>
</table>

### NCM.MacForwarding entity relationships

**TYPE ENTITY JOINED DATA ENTITY**

- Node    NCM.Nodes    NCM.NodeHostsMacForwarding (System.Hosting)

### NCM.Nodes

<table>
<thead>
<tr>
<th>PROPERTY NAME</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>AgentIP</td>
<td>System.String</td>
<td>Ip address of device entered by customer manually</td>
</tr>
<tr>
<td><strong>PROPERTY NAME</strong></td>
<td><strong>DATATYPE</strong></td>
<td><strong>DESCRIPTION</strong></td>
</tr>
<tr>
<td>------------------</td>
<td>---------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Status</td>
<td>System.Byte</td>
<td>NCM only specific status of device: Unknown = 0 (not polled yet) Up = 1 (based on ICMP pool) Down = 2 (based on ICMP pool) Warning = 3 (based on ICMP pool) MonitoringDisabled = 10 (NCM node monitoring is disabled by user) UnManaged = 9 (device is unmanaged in NCM)</td>
</tr>
<tr>
<td>Community</td>
<td>System.String</td>
<td>SNMP community string entered by user</td>
</tr>
<tr>
<td>ReverseDNS</td>
<td>System.String</td>
<td>DNS name of device</td>
</tr>
<tr>
<td>SysName</td>
<td>System.String</td>
<td>An administratively-assigned name for this managed node. By convention, this is the node's fully-qualified domain name.</td>
</tr>
<tr>
<td>SysDescr</td>
<td>System.String</td>
<td>A textual description of the entity. This value should include the full name and version identification of the system's hardware type, software operating-system, and networking software. It is mandatory that this only contains printable ASCII characters.</td>
</tr>
<tr>
<td>SysContact</td>
<td>System.String</td>
<td>The textual identification of the contact person for this managed node, together with information on how to contact this person</td>
</tr>
<tr>
<td>SysLocation</td>
<td>System.String</td>
<td>The physical location of this node (e.g., <code>telephone closet, 3rd floor</code>).</td>
</tr>
<tr>
<td>SystemOID</td>
<td>System.String</td>
<td>The vendor's authoritative identification of the network management subsystem contained in the entity. This value is allocated within the SMI enterprises subtree (1.3.6.1.4.1) and provides an easy and unambiguous means for determining <code>what kind of box</code> is being managed. For example, if vendor <code>Flintstones, Inc.</code> was assigned the subtree 1.3.6.1.4.1.4242, it could assign the identifier 1.3.6.1.4.1.4242.1.1 to its <code>Fred Router</code>.</td>
</tr>
<tr>
<td>Vendor</td>
<td>System.String</td>
<td>Vendor of device- determined based on SystemOID.</td>
</tr>
<tr>
<td>VendorIcon</td>
<td>System.String</td>
<td>Vendor icon of device- determined based on SystemOID.</td>
</tr>
<tr>
<td>MachineType</td>
<td>System.String</td>
<td>Machine Type - determined based on SystemOID.</td>
</tr>
<tr>
<td>LastBoot</td>
<td>System.DateTime</td>
<td>The time (in hundredths of a second) since the network management portion of the system was last re-initialized.</td>
</tr>
<tr>
<td>OSImage</td>
<td>System.String</td>
<td>Determined based on SysDescr</td>
</tr>
<tr>
<td>OSVersion</td>
<td>System.String</td>
<td>Determined based on SysDescr</td>
</tr>
<tr>
<td>SNMPLevel</td>
<td>System.Byte</td>
<td>SNMP version selected by user (1,2 or 3)</td>
</tr>
<tr>
<td><strong>PROPERTY NAME</strong></td>
<td><strong>DATATYPE</strong></td>
<td><strong>DESCRIPTION</strong></td>
</tr>
<tr>
<td>------------------</td>
<td>--------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>SNMPContext</td>
<td>System.String</td>
<td>SNMPv3 credentials entered by user</td>
</tr>
<tr>
<td>SNMPUsername</td>
<td>System.String</td>
<td>SNMPv3 credentials entered by user</td>
</tr>
<tr>
<td>SNMPAuthType</td>
<td>System.String</td>
<td>SNMPv3 credentials entered by user</td>
</tr>
<tr>
<td>SNMPAuthPass</td>
<td>System.String</td>
<td>SNMPv3 credentials entered by user</td>
</tr>
<tr>
<td>SNMPEncryptType</td>
<td>System.String</td>
<td>SNMPv3 credentials entered by user</td>
</tr>
<tr>
<td>SNMPEncryptPass</td>
<td>System.String</td>
<td>SNMPv3 credentials entered by user</td>
</tr>
<tr>
<td>SNMPStatus</td>
<td>System.String</td>
<td>status of SNMP connection to device (OK,No SNMP support, SNMP error description if any)</td>
</tr>
<tr>
<td>NodeID</td>
<td>System.String</td>
<td>A SWIS-generated unique identifier of a network node in the current inventory. (Instances of this property recur in this table according to the number of devices for which data is reported.)</td>
</tr>
<tr>
<td>LastDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM last discovered the device during inventory.</td>
</tr>
<tr>
<td>FirstDiscovery</td>
<td>System.DateTime</td>
<td>A SWIS-generated date and time marker for when NCM first discovered the device during inventory.</td>
</tr>
</tbody>
</table>

**NCM.Nodes entity relationships**

<table>
<thead>
<tr>
<th><strong>TYPE</strong></th>
<th><strong>ENTITY</strong></th>
<th><strong>JOINED DATA ENTITY</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Interfaces</td>
<td>NCM.Interfaces</td>
<td>NCM.NodeHostsInterfaces (System.Hosting)</td>
</tr>
<tr>
<td>MacForwarding</td>
<td>NCM.MacForwarding</td>
<td>NCM.NodeHostsMacForwarding (System.Hosting)</td>
</tr>
<tr>
<td>VLANs</td>
<td>NCM.VLANs</td>
<td>NCM.NodeHostsVLANs (System.Hosting)</td>
</tr>
<tr>
<td>BridgePorts</td>
<td>NCM.BridgePorts</td>
<td>NCM.NodeHostsBridgePorts (System.Hosting)</td>
</tr>
<tr>
<td>ArpTables</td>
<td>NCM.ArpTables</td>
<td>NCM.NodeHostsArpTables (System.Hosting)</td>
</tr>
<tr>
<td>CiscoCards</td>
<td>NCM.CiscoCards</td>
<td>NCM.NodeHostsCiscoCards (System.Hosting)</td>
</tr>
<tr>
<td>CiscoCdp</td>
<td>NCM.CiscoCdp</td>
<td>NCM.NodeHostsCiscoCdp (System.Hosting)</td>
</tr>
<tr>
<td>CiscoChassis</td>
<td>NCM.CiscoChassis</td>
<td>NCM.NodeHostsCiscoChassis (System.Hosting)</td>
</tr>
<tr>
<td>CiscoFlash</td>
<td>NCM.CiscoFlash</td>
<td>NCM.NodeHostsCiscoFlash (System.Hosting)</td>
</tr>
<tr>
<td>CiscoFlashFiles</td>
<td>NCM.CiscoFlashFiles</td>
<td>NCM.NodeHostsCiscoFlashFiles (System.Hosting)</td>
</tr>
<tr>
<td>CiscoImageMIB</td>
<td>NCM.CiscoImageMIB</td>
<td>NCM.NodeHostsCiscoImageMIB (System.Hosting)</td>
</tr>
<tr>
<td>CiscoMemoryPools</td>
<td>NCM.CiscoMemoryPools</td>
<td>NCM.NodeHostsCiscoMemoryPools (System.Hosting)</td>
</tr>
<tr>
<td>TYPE</td>
<td>ENTITY</td>
<td>JOINED DATA ENTITY</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>EntityLogical</td>
<td>NCM.EntityLogical</td>
<td>NCM.NodeHostsEntityLogical (System.Hosting)</td>
</tr>
<tr>
<td>EntityPhysical</td>
<td>NCM.EntityPhysical</td>
<td>NCM.NodeHostsEntityPhysical (System.Hosting)</td>
</tr>
<tr>
<td>PortsTcp</td>
<td>NCM.PortsTcp</td>
<td>NCM.NodeHostsPortsTcp (System.Hosting)</td>
</tr>
<tr>
<td>PortsUdp</td>
<td>NCM.PortsUdp</td>
<td>NCM.NodeHostsPortsUdp (System.Hosting)</td>
</tr>
<tr>
<td>RouteTable</td>
<td>NCM.RouteTable</td>
<td>NCM.NodeHostsRouteTable (System.Hosting)</td>
</tr>
</tbody>
</table>

**NCM.PortsTcp**

<table>
<thead>
<tr>
<th>PROPERTY NAME</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NodeID</td>
<td>System.String</td>
<td>The unique identifier of a network node subject to configuration actions.</td>
</tr>
<tr>
<td>LastDiscovery</td>
<td>System.DateTime</td>
<td>Date and time NCM last discovered the device during inventory.</td>
</tr>
<tr>
<td>FirstDiscovery</td>
<td>System.DateTime</td>
<td>Date and time NCM first discovered the device during inventory.</td>
</tr>
<tr>
<td>TCPLocalAddress</td>
<td>System.String</td>
<td>The local IP address for this TCP connection. In the case of a connection in the listen state which is willing to accept connections for any IP interface associated with the node, the value 0.0.0.0 is used.</td>
</tr>
<tr>
<td>TCPLocalPort</td>
<td>System.Int32</td>
<td>The local port number for this TCP connection.</td>
</tr>
<tr>
<td>TCPLocalPortName</td>
<td>System.String</td>
<td>Port description based on TCPLocalPort value</td>
</tr>
<tr>
<td>TCPRemoteAddress</td>
<td>System.String</td>
<td>The remote IP address for this TCP connection.</td>
</tr>
<tr>
<td>TCPRemotePort</td>
<td>System.Int32</td>
<td>The remote port number for this TCP connection.</td>
</tr>
<tr>
<td>TCPState</td>
<td>System.String</td>
<td>The state of this TCP connection. The only value which may be set by a management station is deleteTCB(12). Accordingly, it is appropriate for an agent to return a 'badValue' response if a management station attempts to set this object to any other value. If a management station sets this object to the value deleteTCB (12), then this has the effect of deleting the TCB (as defined in RFC 793) of the corresponding connection on the managed node, resulting in immediate termination of the connection. As an implementation-specific option, a RST segment may be sent from the managed node to the other TCP endpoint (note however that RST segments are not sent reliably).</td>
</tr>
<tr>
<td>PROPERTY NAME</td>
<td>DATATYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TCPRemotePortName</td>
<td>System.String</td>
<td>Port description based on TCPRemotePort value</td>
</tr>
</tbody>
</table>

### NCM.PortsTcp entity relationships

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ENTITY</th>
<th>JOINED DATA ENTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>NCM.Nodes</td>
<td>NCM.NodeHostsPortsTcp (System.Hosting)</td>
</tr>
</tbody>
</table>

### NCM.PortsUdp

<table>
<thead>
<tr>
<th>PROPERTY NAME</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NodeID</td>
<td>System.String</td>
<td>The unique identifier of a network node subject to configuration actions.</td>
</tr>
<tr>
<td>LastDiscovery</td>
<td>System.DateTime</td>
<td>Date and time NCM last discovered the device during inventory.</td>
</tr>
<tr>
<td>FirstDiscovery</td>
<td>System.DateTime</td>
<td>Date and time NCM first discovered the device during inventory.</td>
</tr>
<tr>
<td>UDPAddress</td>
<td>System.String</td>
<td>The local IP address for this UDP listener. In the case of a UDP listener which is willing to accept datagrams for any IP interface associated with the node, the value 0.0.0.0 is used.</td>
</tr>
<tr>
<td>UDPPort</td>
<td>System.Int32</td>
<td>The local port number for this UDP listener.</td>
</tr>
<tr>
<td>UDPPortName</td>
<td>System.String</td>
<td>Port description based on UDPPort value (like 161-SNMP)</td>
</tr>
</tbody>
</table>

### NCM.PortsUdp entity relationships

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ENTITY</th>
<th>JOINED DATA ENTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>NCM.Nodes</td>
<td>NCM.NodeHostsPortsUdp (System.Hosting)</td>
</tr>
</tbody>
</table>

### NCM.RouteTable

<table>
<thead>
<tr>
<th>PROPERTY NAME</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NodeID</td>
<td>System.String</td>
<td>The unique identifier of a network node subject to configuration actions.</td>
</tr>
<tr>
<td>LastDiscovery</td>
<td>System.DateTime</td>
<td>Date and time NCM last discovered the device during inventory.</td>
</tr>
<tr>
<td>FirstDiscovery</td>
<td>System.DateTime</td>
<td>Date and time NCM first discovered the device during inventory.</td>
</tr>
<tr>
<td>Property Name</td>
<td>Datatype</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>InterfaceIndex</td>
<td>System.Int32</td>
<td>The index value which uniquely identifies the local interface through which the next hop of this route should be reached. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex.</td>
</tr>
<tr>
<td>InterfaceID</td>
<td>System.String</td>
<td>Indicate the mask to be logical-ANDed with the destination address before being compared to the value in the ipRouteDest field. For those systems that do not support arbitrary subnet masks, an agent constructs the value of the ipRouteMask by determining whether the value of the correspondent ipRouteDest field belong to a class-A, B, or C network, and then using one of:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- mask network</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 255.0.0.0 class-A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 255.255.0.0 class-B</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 255.255.255.0 class-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the value of the ipRouteDest is 0.0.0.0 (a default route), then the mask value is also 0.0.0.0. It should be noted that all IP routing subsystems implicitly use this mechanism.</td>
</tr>
<tr>
<td>Destination</td>
<td>System.String</td>
<td>The type of route. Note that the values direct(3) and indirect(4) refer to the notion of direct and indirect routing in the IP architecture. Setting this object to the value invalid(2) has the effect of invalidating the corresponding entry in the ipRouteTable object. That is, it effectively disassociates the destination identified with said entry from the route identified with said entry. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries not currently in use. Proper interpretation of such entries requires examination of the relevant ipRouteType object.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Possible Values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- other(1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- invalid(2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- direct(3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- indirect(4)</td>
</tr>
<tr>
<td><strong>Property Name</strong></td>
<td><strong>Datatype</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>------------------</td>
<td>--------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Mask</td>
<td>System.String</td>
<td>The routing mechanism via which this route was learned. Inclusion of values for gateway routing protocols is not intended to imply that hosts should support those protocols. Possible Values: other(1) local(2) netmgmt(3) icmp(4) egp(5) ggp(6) hello(7) rip(8) is-is(9) es-is(10) ciscolgrp(11) bbnSpflgp(12) ospf(13) bgp(14)</td>
</tr>
<tr>
<td>NextHop</td>
<td>System.String</td>
<td>The number of seconds since this route was last updated or otherwise determined to be correct. Note that no semantics of 'too old' can be implied except through knowledge of the routing protocol by which the route was learned.</td>
</tr>
<tr>
<td>RouteType</td>
<td>System.String</td>
<td>The IP address of the next hop of this route. (In the case of a route bound to an interface which is realized via a broadcast media, the value of this field is the agent's IP address on that interface.)</td>
</tr>
<tr>
<td>RouteProtocol</td>
<td>System.String</td>
<td>The primary routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipRouteProto value. If this metric is not used, its value should be set to -1.</td>
</tr>
<tr>
<td>RouteAge</td>
<td>System.Int32</td>
<td>An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipRouteProto value. If this metric is not used, its value should be set to -1.</td>
</tr>
<tr>
<td>NextHopAS</td>
<td>System.Int32</td>
<td>An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipRouteProto value. If this metric is not used, its value should be set to -1.</td>
</tr>
<tr>
<td>Property Name</td>
<td>Datatype</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Metric1</td>
<td>System.Int32</td>
<td>An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipRouteProto value. If this metric is not used, its value should be set to -1.</td>
</tr>
<tr>
<td>Metric2</td>
<td>System.Int32</td>
<td>An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipRouteProto value. If this metric is not used, its value should be set to -1.</td>
</tr>
<tr>
<td>Metric3</td>
<td>System.Int32</td>
<td>The index value which uniquely identifies the local interface through which the next hop of this route should be reached. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex.</td>
</tr>
<tr>
<td>Metric4</td>
<td>System.Int32</td>
<td>Indicate the mask to be logical-ANDed with the destination address before being compared to the value in the ipRouteDest field. For those systems that do not support arbitrary subnet masks, an agent constructs the value of the ipRouteMask by determining whether the value of the correspondent ipRouteDest field belong to a class-A, B, or C network, and then using one of:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>mask network</td>
</tr>
<tr>
<td></td>
<td></td>
<td>255.0.0.0 class-A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>255.255.0.0 class-B</td>
</tr>
<tr>
<td></td>
<td></td>
<td>255.255.255.0 class-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the value of the ipRouteDest is 0.0.0.0 (a default route), then the mask value is also 0.0.0.0. It should be noted that all IP routing subsystems implicitly use this mechanism.</td>
</tr>
</tbody>
</table>
## Metric5

**Property Name:** Metric5  
**DataType:** System.Int32  
**Description:** The type of route. Note that the values direct(3) and indirect(4) refer to the notion of direct and indirect routing in the IP architecture.

Setting this object to the value invalid(2) has the effect of invalidating the corresponding entry in the ipRouteTable object. That is, it effectively disassociates the destination identified with said entry from the route identified with said entry. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries not currently in use. Proper interpretation of such entries requires examination of the relevant ipRouteType object.

Possible Values:
- other(1)
- invalid(2)
- direct(3)
- indirect(4)

### NCM.RouteTable entity relationships

<table>
<thead>
<tr>
<th>Type</th>
<th>Entity</th>
<th>Joined Data Entity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>NCM.Nodes</td>
<td>NCM.NodeHostsRouteTable (System.Hosting)</td>
</tr>
</tbody>
</table>

### NCM.VLANs

<table>
<thead>
<tr>
<th>Property Name</th>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NodeID</td>
<td>System.String</td>
<td>The unique identifier of a network node subject to configuration actions.</td>
</tr>
<tr>
<td>LastDiscovery</td>
<td>System.DateTime</td>
<td>Date and time NCM last discovered the device during inventory.</td>
</tr>
<tr>
<td>FirstDiscovery</td>
<td>System.DateTime</td>
<td>Date and time NCM first discovered the device during inventory.</td>
</tr>
<tr>
<td>PROPERTY NAME</td>
<td>DATATYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>VLANID</td>
<td>System.Int32</td>
<td>The set of the device's member ports that belong to the VLAN. Each octet within the value of this object specifies a set of eight ports, with the first octet specifying ports 1 through 8, the second octet specifying ports 9 through 16, etc. Within each octet, the most significant bit represents the lowest numbered port, and the least significant bit represents the highest numbered port. Thus, each port of the VLAN is represented by a single bit within the value of this object. If that bit has a value of '1' then that port is included in the set of ports; the port is not included if its bit has a value of '0'. A port number is the value of dot1dBasePort for the port in the BRIDGE-MIB (RFC 1493).</td>
</tr>
<tr>
<td>VLANName</td>
<td>System.String</td>
<td>The name of this VLAN. This name is used as the ELAN-name for an ATM LAN-Emulation segment of this VLAN.</td>
</tr>
<tr>
<td>VLANMTU</td>
<td>System.Int32</td>
<td>The MTU size on this VLAN, defined as the size of largest MAC-layer (information field portion of the) data frame which can be transmitted on the VLAN.</td>
</tr>
<tr>
<td>VLANType</td>
<td>System.Int32</td>
<td>The type of this VLAN</td>
</tr>
<tr>
<td>VLANState</td>
<td>System.Int32</td>
<td>The state of this VLAN. Possible Values: operational(1) suspended(2) mtuTooBigForDevice(3) mtuTooBigForTrunk(4)</td>
</tr>
</tbody>
</table>

NCM.VLANs entity relationships

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ENTITY</th>
<th>JOINED DATA ENTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>NCM.Nodes</td>
<td>NCM.NodeHostsVLANs (System.Hosting)</td>
</tr>
</tbody>
</table>
EnergyWise devices

This topic explores how you can use SolarWinds Network Configuration Manager in conjunction with SolarWinds Network Performance Monitor to enable and manage your Cisco EnergyWise devices.

What is EnergyWise?

EnergyWise is Cisco's response to the call to cut energy costs, address environmental concerns, and adhere to government directives around green technologies. By purchasing EnergyWise capable devices and enabling their energy-saving features, you can retain business critical systems in a fully powered state while allowing less critical power over ethernet (PoE) devices to power down or drop into standby during off hours.

EnergyWise gives you the ability to control your energy cost. SolarWinds NCM gives you the ability to remotely apply recurrence policies and schedule power usage, helping you use less power. And, SolarWinds NPM allows you to monitor your energy use and power levels. SolarWinds perfectly partners with Cisco and the EnergyWise technologies to help you save more and monitor your savings.

Manage and enable EnergyWise nodes

Cisco devices that support the EnergyWise technology can be enabled and their EnergyWise settings managed through the SolarWinds NCM integration with SolarWinds NPM.

Before completing the following procedure, EnergyWise nodes must be managed in both SolarWinds NCM and SolarWinds NPM.

For details on adding nodes, see Add nodes via Network Discovery.

1. Click Settings > All Settings.
2. Under Node & Group Management, click Manage Nodes.
3. Select the Cisco node for which you want EnergyWise enabled, and click More Actions > Manage EnergyWise.
4. Click Enable EnergyWise on these nodes.
5. Specify the values on the Manage EnergyWise Node page.
6. Click Execute Config Actions.

Manage Power over Ethernet ports

Power over Ethernet (PoE) devices are connected to your devices on an interface and are managed at the interface level. Before completing the following procedure, you must have installed the SolarWinds NCM integration for SolarWinds NPM on your SolarWinds NPM server and added your EnergyWise capable nodes to both SolarWinds NCM and SolarWinds NPM.
1. Click Settings > All Settings.
2. Under Node & Group Management, click Manage Nodes.
3. Expand the Cisco node containing the interface you want to configure.
4. Select the interface you want to enable EnergyWise, and click More Actions > Manage EnergyWise.
5. Click Enable EnergyWise on these nodes.
6. Specify the values on the Manage EnergyWise Interface page.
7. Click Execute Config Actions.
Integration with Engineer's Toolset

If Engineer's Toolset is installed on the same server as SolarWinds NCM, you can take advantage of the following integrations on the Node Details widget:

- Web browse to the selected node
- Telnet to the selected node
- Ping the selected node
- Run a trace route to the selected node
- Remote desktop to the selected node
References

The following sections contain supplemental reference material for SolarWinds NCM.

Node and configuration archive variables

SolarWinds Network Configuration Manager uses a variable system that is similar to that of SolarWinds Network Performance Monitor. Variables always begin with a dollar sign and a curly brace (${}), and always end with a curly brace ({}).

Variables can be used in most custom properties. They can also be used in any of the user editable system properties.

Variables can also be nested and recursive. That is, a single variable can refer to a node property that contains more variables that then contain even more variables. The following example demonstrates nested variables:

<table>
<thead>
<tr>
<th>NODE PROPERTY</th>
<th>VALUE OF PROPERTY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Rack ${Rack} on ${Floor} floor of ${Building} - ${SysLocation}</td>
</tr>
<tr>
<td>Building</td>
<td>Building C</td>
</tr>
<tr>
<td>SysLocation</td>
<td>Data Center A</td>
</tr>
<tr>
<td>Rack</td>
<td>15</td>
</tr>
<tr>
<td>Floor</td>
<td>Second</td>
</tr>
</tbody>
</table>

The database value of Location is Rack ${Rack} on ${Floor} floor of ${Building} - ${SysLocation}. The displayed value of Location is Rack 15 on Second floor of Building C - Data Center A.

Configuration archive variables

SolarWinds Network Configuration Manager stores all downloaded configurations in a database. It can also store a copy of them in the configuration archive directory. The directory structure can be specified using any of the previous variables.

Additional variables may also be used when specifying the configuration archive directory. Many of these variables use the localization settings for the current language and region.

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateTime</td>
<td>Local date and time in short date and local time format</td>
</tr>
<tr>
<td>Date</td>
<td>Date in short date format</td>
</tr>
<tr>
<td>PROPERTY</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>LongDate</td>
<td>Date in long date format</td>
</tr>
<tr>
<td>MediumDate</td>
<td>Date in medium date format</td>
</tr>
<tr>
<td>Time</td>
<td>Time in short time format</td>
</tr>
<tr>
<td>LongTime</td>
<td>Time in long time format</td>
</tr>
<tr>
<td>MediumTime</td>
<td>Time in medium time format</td>
</tr>
<tr>
<td>ShortTime</td>
<td>Time in short time format</td>
</tr>
<tr>
<td>DOW</td>
<td>Day of the week, spelled out</td>
</tr>
<tr>
<td>D</td>
<td>Day of the month</td>
</tr>
<tr>
<td>DD</td>
<td>Day of the month, with leading zero, if needed</td>
</tr>
<tr>
<td>ABREVIATETDOW</td>
<td>Day of the week in abbreviated format</td>
</tr>
<tr>
<td>LocalDow</td>
<td>Day of the week in the local language</td>
</tr>
<tr>
<td>Month</td>
<td>Number of the current month</td>
</tr>
<tr>
<td>M</td>
<td>Number of the current month</td>
</tr>
<tr>
<td>MM</td>
<td>Number of the current month, with leading zeros, if needed</td>
</tr>
<tr>
<td>MMM</td>
<td>Abbreviated name of the month</td>
</tr>
<tr>
<td>MMMM</td>
<td>Name of the month</td>
</tr>
<tr>
<td>LocalMonthName</td>
<td>Name of the month in the local language</td>
</tr>
<tr>
<td>DAYOFYEAR</td>
<td>Day number of the year</td>
</tr>
<tr>
<td>YYYY</td>
<td>4 digit year</td>
</tr>
<tr>
<td>YY</td>
<td>2 digit year</td>
</tr>
<tr>
<td>YEAR2</td>
<td>2 digit year</td>
</tr>
<tr>
<td>YEAR4</td>
<td>4 digit year</td>
</tr>
<tr>
<td>H</td>
<td>Hour</td>
</tr>
<tr>
<td>HH</td>
<td>2 digit hour, with leading zero, if needed</td>
</tr>
<tr>
<td>N</td>
<td>Minute</td>
</tr>
<tr>
<td>NN</td>
<td>2 digit minute, with leading zero, if needed</td>
</tr>
<tr>
<td>S</td>
<td>Seconds</td>
</tr>
<tr>
<td>SS</td>
<td>2 digit seconds, with leading zero, if needed</td>
</tr>
<tr>
<td><strong>PROPERTY</strong></td>
<td><strong>DESCRIPTION</strong></td>
</tr>
<tr>
<td>-------------</td>
<td>----------------</td>
</tr>
<tr>
<td>AMPM</td>
<td>AM or PM</td>
</tr>
<tr>
<td>CRLF</td>
<td>Carriage return - linefeed combination</td>
</tr>
<tr>
<td>ConfigType</td>
<td>Type of configuration: running, startup, etc.</td>
</tr>
<tr>
<td>Caption</td>
<td>Caption of the node, NodeCaption</td>
</tr>
</tbody>
</table>