

NEXUS 7000 Training

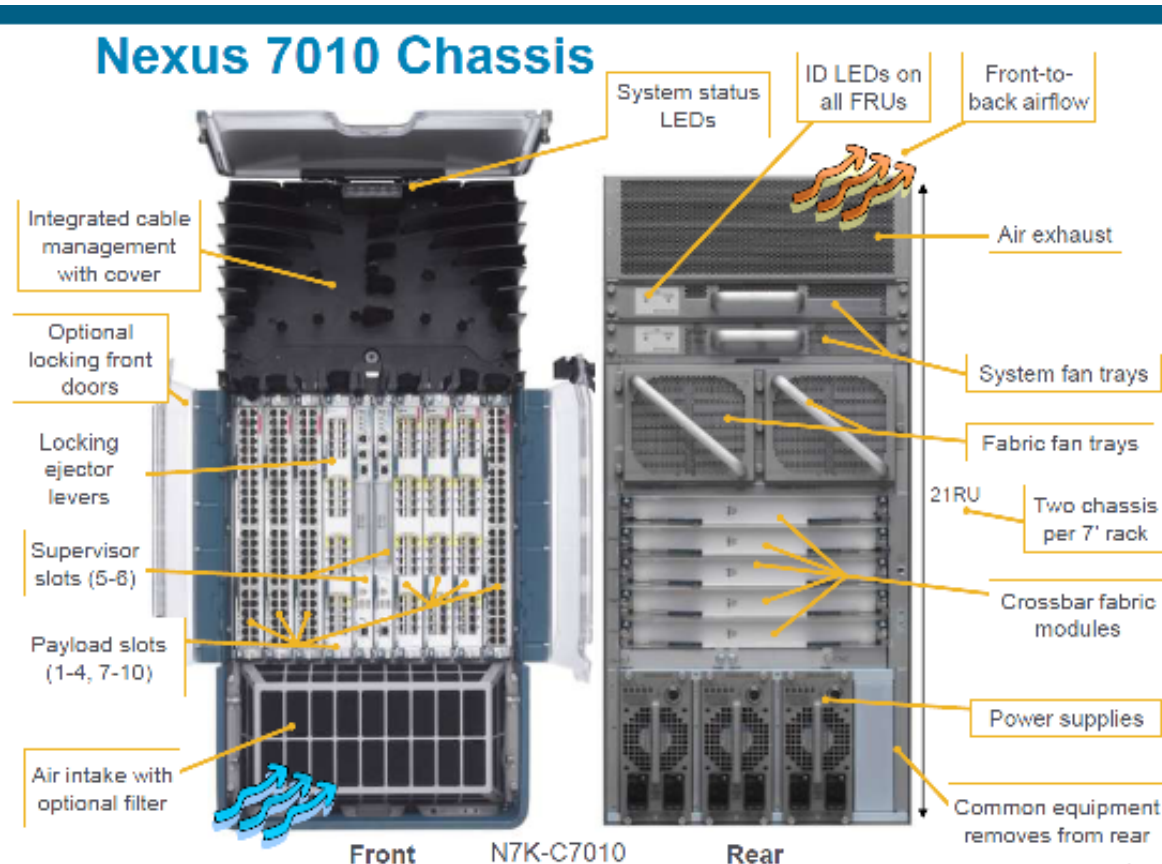
Scope and Purpose

The scope and purpose of this document is to familiarize the user with the Cisco Nexus 7010 Platform that is going to be deployed in Any City,USA. The purpose of this document is not to teach the user basic Cisco commands or the fundamentals of IOS/NXOS. It is assumed that the trainee has a good understanding of the IOS and all Cisco technologies. Covered will be basic hardware and software configuration differences.

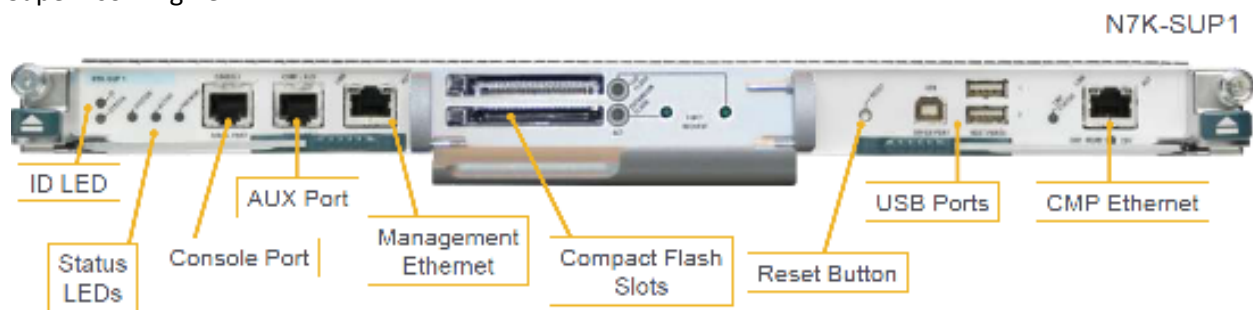
Also with this document are 18 labs that help you to get comfortable with NX-OS and the command structure. These labs are only available while the Nexus 7000s are in the lab.

Nexus 7010 Chassis Overview

Below are the components of the Nexus 7000 Chassis



Supervisor Engine:



The Management Ethernet interface is shared between all VDCs, so the management interfaces for all VDCs must be on the same subnet. Be aware of this as well when it comes to port-security as you need to allocate for multiple MAC addresses.

The CMP Ethernet interface is an always-on Connectivity Management Processor for lights-out management. This runs lightweight software that is completely independent of the NX-OS. This interface will allow you to reboot the entire chassis if the supervisors are unresponsive. This is very similar to ILO operation on servers.

Background:

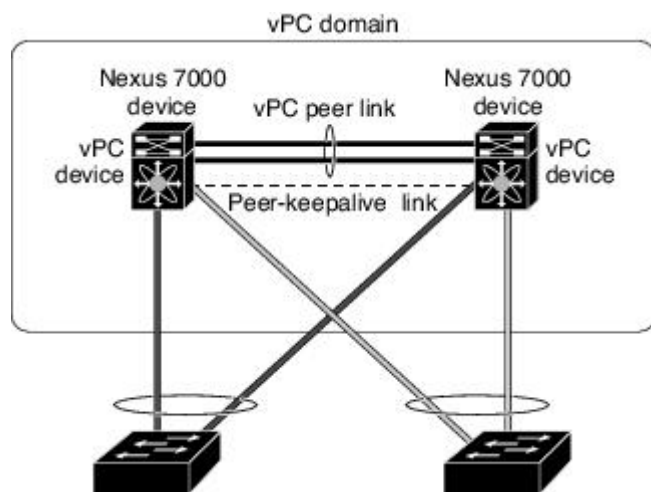
The current Core Switches installed in Any City, CA (Cisco 6509) are end-of-life and near end-of-support. The decision was made to deploy the next generation of Cisco switches, the Nexus 7000, at this location due to the highly scalable nature of the switch as well as this switch is designed for mission-critical data center operations. The fabric architecture easily scales and has built-in future support for 40-Gbps and 100-Gbps Ethernet switching.

The Nexus 7000 also runs a new version of operating code called NX-OS. NX-OS is a module operating system and designed for scalability, continuous operation, serviceability, and transport flexibility. The switches that are to be deployed in Any City have dual-supervisors installed for maximum uptime. It is the nature of this switch to be upgraded in real-time with lossless failover. In other words you can (still would do it in a maintenance window) upgrade the NX-OS code without taking down the network. Due to the modularity of NX-OS, you can upgrade the entire code or just specific portions of code. Another feature of this modularity is that if you do not need a feature, for instance OSPF, you do not need to enable it. This helps to reduce the overhead on the switch as well as prevent any bug/vulnerability in the OSPF processes from impacting normal operations.

One fast difference between traditional IOS and NX-OS are the interfaces. In IOS you have Ethernet, FastEthernet, GigabitEthernet, etc – with the NX-OS, everything is Ethernet. So instead of Gig1/0, you have Eth1/0 for the gigabit Ethernet port.

Another feature of these switches is Virtual Device Contexts (VDC). VDCs allow a switch to be logically segmented into up to 4 virtual switches. The default VDC (VDC 1) is where all interfaces are initially located and then can be assigned into a VDC from there. An interface can only exist in one VDC and managed from that same VDC. You cannot manage an interface if it does not belong to that VDC.

Virtual Port Channels, or vPCs, are another feature of the Nexus 7000 switch. A vPC allows links that are physically connected to two different Nexus 7000 switches to appear as a single port-channel by a another device. The other device can be a switch, a server, or any other networking device that supports port channels. By utilizing vPCs for connections to other switches, spanning-tree topologies can be greatly simplified as there should be no uplink ports in BLOCKING mode.



Systems Management

Checking resources on a module

To check the resources of an individual module, from the main VDC:

Attach to the module (here module 3):

```
CoreSwitch1.VDC1.RWC# attach module 3  
Attaching to module 3 ...  
To exit type 'exit', to abort type '$.'
```

Execute the following command:

```
module-3# show system resources  
Load average: 1 minute: 0.20 5 minutes: 0.14 15 minutes: 0.12  
Processes : 66 total, 1 running  
CPU states : 0.0% user, 0.0% kernel, 100.0% idle  
Memory usage: 1036876K total, 3396172K used, 1835008K free  
291392K buffers, 578124K cache
```

Then exit the module to return to the VDC

```
module-3# exit  
rlogin: connection closed.  
CoreSwitch1.VDC1.RWC#
```

Saving your configuration

The NX-OS does not support write mem, you must use the command copy running-config startup-config:

```
CoreSwitch1.VDC1.RWC# copy running-config startup-config  
[#####] 100%  
CoreSwitch1.VDC1.RWC#
```

I suggest creating an alias for **wr** if you are in the habit of using those commands.

```
CoreSwitch1.VDC1.RWC(config)# cli alias name wr copy runn start
```

Transceiver Identification

To find out what type of transceiver is installed in a SFP port, execute:

```
CoreSwitch1.VDC1.RWC# sh interface ethernet 7/1 transceiver  
Ethernet7/1  
sfp is present  
type is 10Gbase-SR  
name is CISCO-SUMITOMO  
part number is SPP5100SR-C1  
revision is A  
serial number is SPC141301JG  
nominal bitrate is 10300 MBit/sec  
Link length supported for 50/125 um fiber is 80 m  
Link length supported for 62.5/125 um fiber is 20 m  
cisco id is --  
cisco extended id number is 4
```

```
CoreSwitch1.VDC1.RWC#
```

This command also works for ranges, so instead of 7/1 you can use 7/1-32 and get a list of all the transceivers installed in that card.

Line card identification

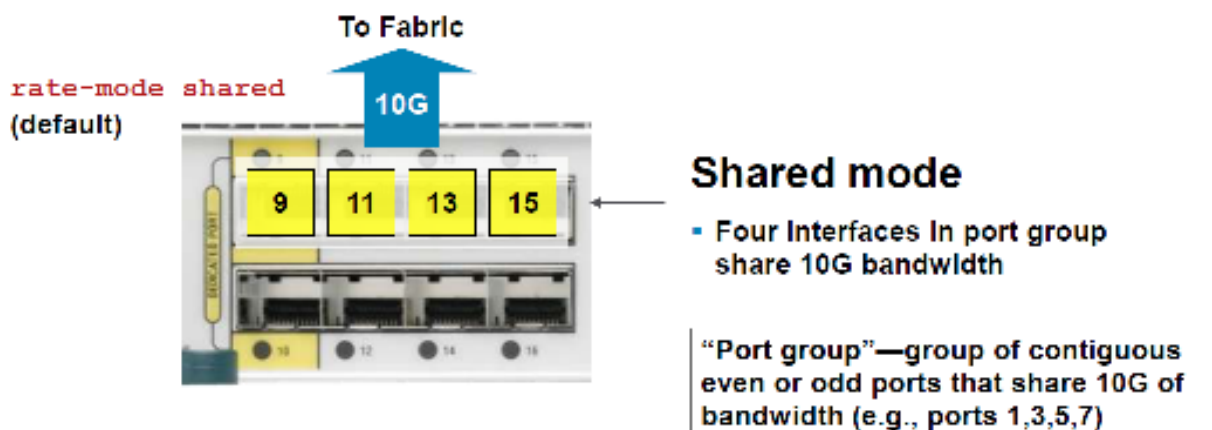
With the Nexus 7000 switches there is a new feature on the cards in the shape of a ID LED. This ID LED is a blue LED that you enable from the CLI to either identify a line-card or a switch. To turn this ID LED on and off, use the *locator-led* command to enable the LED and the *no locator-led* to disable.

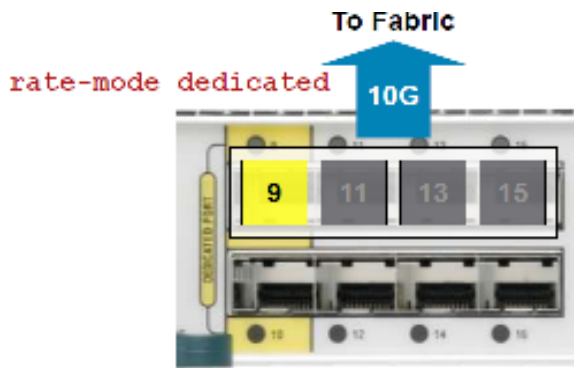
```
N7K2# locator-led module 4
N7K2#
N7K2# no locator-led module 4
N7K2#
N7K2#
```

10Gb Rate Modes

There are two rate-modes available for the 10GB Ethernet module, shared or dedication. In Shared mode, a 4 port-group of interfaces will share a single 10Gb ASIC. A port-group is a group of contiguous even or odd ports that share a 10G of bandwidth (ports 1,3,5,7). In Dedicated mode only the first port in the group will be available and the other three ports will be disabled. The first port of group is identifiable on the card via the yellow labeling.

Note: A port-group can only be assigned to a VDC as a group, you cannot assign Ethernet 7/1 to VDC-1 and Ethernet 7/3 to VDC-2. The port-group must be assigned to the same VDC.





Dedicated mode

- First Interface In port group gets 10G bandwidth
- Other three Interfaces In port group disabled

To configure these modes you use the rate-mode command:

```
CoreSwitch1.VDC1.RWC(config-if) # rate-mode ?
    dedicated  Rate mode is dedicated
    shared     Rate mode is shared
```

System Switchover or Processor failover

To switchover to the redundant supervisor, issue the command:

```
CoreSwitch1.VDC1.RWC# system switchover
CoreSwitch1.VDC1.RWC# writing reset reason 7,
20
NX7 SUP Ver 3.17.0
Serial Port Parameters from CMOS
```

Once you issue that command, the supervisor that was active will automatically reboot. The system is not down as the redundant supervisor has taken over prior to the reboot.

VDC

To see what VDCs are on the system:

```
CoreSwitch1.VDC1.RWC# sh vdc
```

vdc_id	vdc_name	state	mac
1	CoreSwitch1.VDC1.RWC	active	00:26:98:07:95:c1

```
CoreSwitch1.VDC1.RWC#
```

To create a VDC

```
CoreSwitch1.VDC1.RWC(config)# vdc CoreSwitch1.VDC2
Note: Creating VDC, one moment please ...
2010 Jun 10 01:59:02 CoreSwitch1 %$ VDC-1 %$ %VDC_MGR-2-VDC_ONLINE: vdc 2 has come
nline

CoreSwitch1.VDC1.RWC(config-vdc)# 2010 Jun 10 02:02:56 CoreSwitch1 %$ VDC-1 %$
%VDC_MGR-2-VDC_ONLINE: vdc 2 has come online
```

To assigned interfaces to a VDC:

```
CoreSwitch1.VDC1.RWC(config-vdc) # allocate interface ethernet 7/1, ethernet 7/3, ethernet 7/5, e7/7
```

Moving ports will cause all config associated to them in source vdc to be removed. Are you sure you want to move the ports (y/n)? [yes] y

```
CoreSwitch1.VDC1.RWC(config-vdc) #
```

To see what ports are assigned to a given VDC:

```
CoreSwitch1.VDC1.RWC# sh vdc coreSwitch1.VDC2.RWC membership
```

```
vdc_id: 2 vdc_name: CoreSwitch1.VDC2.RWC interfaces:
      Ethernet7/1          Ethernet7/3          Ethernet7/5
      Ethernet7/7
```

```
CoreSwitch1.VDC1.RWC#
```

To switch and from a VDC use the commands switchto and switchback:

```
CoreSwitch1.VDC1.RWC# switchto vdc CoreSwitch1.VDC2
Cisco Nexus Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Copyright (c) 2002-2010, Cisco Systems, Inc. All rights reserved.
The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under
license. Certain components of this software are licensed under
the GNU General Public License (GPL) version 2.0 or the GNU
Lesser General Public License (LGPL) Version 2.1. A copy of each
such license is available at
http://www.opensource.org/licenses/gpl-2.0.php and
http://www.opensource.org/licenses/lgpl-2.1.php
CoreSwitch1.VDC1.RWC#
CoreSwitch1.VDC1.RWC-CoreSwitch1.VDC2# switchback
```

VLANS

To create VLANs on the NX-OS, you first need to enable the feature. If you do not enable the feature you will not be able to create vlans:

```
CoreSwitch1.VDC1.RWC(config) # interface vlan 100
```

```
Invalid command (interface name) at '^' marker.
```

```
CoreSwitch1.VDC1.RWC(config) # feature interface-vlan
```

```
CoreSwitch1.VDC1.RWC(config) # int vlan 100
```

```
CoreSwitch1.VDC1.RWC(config-if) #
```

Other Layer 2 features that you might need to enable are LACP and UDLD.

First Hop Redundancy Protocols

The Nexus 7000 supports HSRP, VRRP, and GLBP. As with VLANs, these protocols are features and you will need to enable the one that you want to use. A new feature with HSRP is the support for vPC. This new support will allow vPC traffic through both the active and standby HSRP router. If you are running vPC between your Nexus 7000 pair it is recommended that you run HSRP if you are going to use a FHRP.

As with almost any other protocol on the Nexus, these will be need to be enabled via the *feature* command:

```
CoreSwitch1.RWC(config)# feature hsrp
CoreSwitch1.RWC(config)# feature vrrp
CoreSwitch1.RWC(config)# feature glbp
```

To configure HSRP on an interface:

```
CoreSwitch1.RWC(config-if)# hsrp 1
CoreSwitch1.RWC(config-if-hsrp)# ip 1.1.1.2
CoreSwitch1.RWC(config-if-hsrp)# preempt
CoreSwitch1.RWC(config-if-hsrp)# priority 110
```

Routing Protocols

The Nexus 7000 supports OSPFv2, OSPFv3, EIGRP, IS-IS, BGP, and RIPv2 (RIPv1 is NOT supported). Again, each of these protocols is a feature and will need to be enabled on a PER VDC basis. To enable these features, use the *feature* command and the appropriate protocol:

```
CoreSwitch1.RWC(config)# feature eigrp
CoreSwitch1.RWC(config)# feature ospf
CoreSwitch1.RWC(config)# feature ospfv3
CoreSwitch1.RWC(config)# feature rip
CoreSwitch1.RWC(config)# feature bgp
CoreSwitch1.RWC(config)# feature isis
```

Once the routing protocol is enabled, to configure EIGRP and place an interface in EIGRP:

```
CoreSwitch1.RWC(config-if)# router eigrp 10
CoreSwitch1.RWC(config-router)# int e3/1
CoreSwitch1.RWC(config-if)# no switchport
CoreSwitch1.RWC(config-if)# ip add 1.1.1.1 255.255.255.0
CoreSwitch1.RWC(config-if)# ip router eigrp 10
```

To place the interface in PASSIVE mode:

```
CoreSwitch1.RWC(config-if)# ip passive-interface eigrp 10
```

To enabled Interface Authentication (MD5)

```
CoreSwitch1.RWC(config)# key chain EIGRP-KEY
CoreSwitch1.RWC(config-keychain)# key 1
CoreSwitch1.RWC(config-keychain-key)# key-string ThisIsTheSecretKey
CoreSwitch1.RWC(config)# interface e3/1
CoreSwitch1.RWC(config-if)# ip authentication mode eigrp 10 md5
CoreSwitch1.RWC(config-if)# ip authentication key-chain eigrp 10 EIGRP-KEY
```

To enable Distribution lists:

```
CoreSwitch1.RWC(config-if)# ip distribute-list eigrp 10 prefix-list EIGRP-DENY out
```

To configure route summarization under an interface

```
CoreSwitch1.RWC(config-if)# ip summary-address eigrp 10 10.0.0.0/8
```


Layer 3 Protocol High Availability

In a dual-supervisor configuration you should enable non-stop forwarding (NSF) with your routing protocols. This will provide the capability for the switch to maintain an IP forwarding table during a supervisor switchover.

This allows the Cisco Nexus 7000 to maximize the benefits of the separation between control plane and data plane forwarding in the architecture. During the restart of a graceful-operations-capable protocol, that routing protocol will utilize the relevant mechanism to signal to its neighbors that a planned restart is being executed. This notification will allow the neighbors to continue to forward traffic to the Nexus as if operations were normal. This allows the restarting service to remove itself from the network control plane in a non-disruptive fashion. The Nexus will continue to forward traffic based on the last established routing and forwarding information base (RIB/FIB), thereby allowing uninterrupted, continuous data delivery during the restart. Once the restart is complete; it will reestablish the neighbors and rebuild the adjacencies in a non-disruptive fashion. The only risk to packets here is there are any other network state changes (reconvergences) during the switchover and recovery process.

Multicast

The Nexus 7000 supports MBGP, MSDP, PIM Sparse, PIM6, and IGMP. PIM Dense mode is no longer supported in NX-OS. To utilize PIM or MSDP, you will need to enable the *feature*.

```
CoreSwitch1.RWC(config)# feature pim  
CoreSwitch1.RWC(config)# feature pim6  
CoreSwitch1.RWC(config)# feature msdp
```

QoS

QoS on the Nexus 7000 is very similar to Cisco IOS QoS layout. There are differences, but they are based on the line cards, in a per-card basis.

The 32-Port 10 G module supports:

Queues Per port:

- Ingress: 8 queues and 2 thresholds (RX: 8q2t)
- Egress: 1 strict priority, 7 deficit weighted round robin (DWRR), and 4 thresholds (TX: 1p7q4t)

Port Buffers:

- 1 MB + 65 MB ingress per port for **dedicated mode**
- 80 MB egress per port for **dedicated mode**
- 1 MB per port + 65 MB shared per ingress port group in **shared mode**
- 80 MB egress per 4-port group in **shared mode**

The 48-Port 10/100/1000 module supports:

Queues Per Port:

- Ingress: 2 queues and 4 thresholds (RX: 2q4t)
- Egress: 1 strict priority, 3 DWRR, and 4 thresholds (1p3q4t)

Port Buffers:

- 7.56 MB ingress per port
- 6.15 MB egress per port

QoS in the NX-OS is configured the same as IOS.

Security

All the normal security features are supported in the NX-OS. Features such as uRPF, IPv4 and IPv6 packet sanity checks, DHCP snooping, dynamic ARP inspection, port-security, ACL, etc, are all supported. By default CoPP is configured in the system in order to protect the control plane, the default configuration is sufficient the majority of the time. It can be edited if the need arises just like any other CoPP policy.

Ethanalzyer

Ethanalzyer is a NX-OS protocol analyzer tool based on Wireshark. The Ethanalzyer is a command-line version of Wireshark that captures and decodes packets, and can be used to troubleshoot the network and analyze control-plane traffic. The Ethanalzyer uses the same capture file syntax as tcpdump. More information on tcpdump can be found at http://www.tcpdump.org/tcpdump_man.html

To use Ethanalzyer, you must be in the default VDC context.

```
CoreSwitch1.RWC# ethanalzyer ?
  local  Start local capture of frames to Sup

CoreSwitch1.RWC# ethanalzyer local ?
  <CR>
  >      Redirect it to a file
  >>     Redirect it to a file in append mode
  interface  A live capture will start on following interface
  read       Filename to read the captured packet
  |          Pipe command output to filter

CoreSwitch1.RWC# ethanalzyer local interface ?
  inband  Inband/Outband interface
  mgmt    Management interface

CoreSwitch1.RWC# ethanalzyer local interface inband ?
  <CR>
  >      Redirect it to a file
  >>     Redirect it to a file in append mode
  capture-filter      Filter on ethanalzyer capture
  decode-internal     Include internal system header decoding
  detail              Display detailed protocol information
  display-filter       Display filter on frames captured
  limit-captured-frames Maximum number of frames to be captured (default is
                        10)
  limit-frame-size     Capture only a subset of a frame
  raw                  Hex/Ascii dump the packet with possibly one line
                        summary
  write                Filename to save capture to
  |                    Pipe command output to filter
```

Configuration Rollback

Checkpoints allow you to create a “snapshot” of the current configuration at a given point in time. This is a very useful feature for change management and should be used before a change, and use again if you need to roll-back a change.

To create a checkpoint:

```
CoreSwitch1.VDC1.RWC# checkpoint Initial  
.....Done  
CoreSwitch1.VDC1.RWC#
```

To view the checkpoint that you just created, or to see what has been created:

```
CoreSwitch1.VDC1.RWC# show checkpoint summary  
User Checkpoint Summary  
-----  
1) Initial:  
Created by admin  
Created at Thu, 00:55:02 10 Jun 2010  
Size is 11,139 bytes  
Description: None  
  
CoreSwitch1.VDC1.RWC#
```

You may also save the checkpoint to bootflash:

```
CoreSwitch1.VDC1.RWC# checkpoint file bootflash:ExampleCheckpoint  
Done  
CoreSwitch1.VDC1.RWC# dir  
11139 Jun 10 00:59:36 2010 ExampleCheckpoint
```

To rollback a checkpoint, depending on where you stored the checkpoint, the command syntax is:

```
CoreSwitch1.VDC1.RWC# rollback running-config checkpoint Initial  
or  
CoreSwitch1.VDC1.RWC# rollback running-config file bootflash:ExampleCheckpoint
```

Cisco IOS CLI	Cisco NX-OS CLI
Default User Prompt	
IOS>	NX-OS#
Entering Configuration Mode	
IOS# Configuration terminal	NX-OS# Configuration terminal
Saving the Running Config	
IOS# write memory or IOS# copy running-config startup-config	NX-OS# copy running-config startup-config
Erasing the startup config	
IOS# write erase	NX-OS# write erase
Installing a license	
Not application to IOS	NX-OS# install license bootflash:license_file.lic
Interface Naming Convention	
Interface Ethernet 1/1 Interface FastEthernet 1/1 Interface GigabitEthernet 1/1 Interface TenGigabitEthernet 1/1	Ethernet 1/1
Configuration software image boot variables	
boot system flash sup-bootdisk:ios.bin	boot kickstart bootflash:/kickstart.bin sup-1 boot system bootflash:/nx-os.bin sup-1 boot kickstart bootflash:/kickstart.bin sup-2 boot kickstart bootflash:/nx-os.bin sup-2
Enabling features	
Not Application on IOS	NX-OS(config)# feature <i>feature</i>
Enabling Telnet	
Enabled by default	NX-OS(config)#feature telnet
Configuring VTY timeout and session limit	
line vty 0 4 exec-timeout 15 0 login	line vty session-limit 5 exec-time 15