

- Switch Port Configuration -

Cisco Operating Systems

Cisco offers two brands of network switches:

- **Catalyst** – Cisco’s flagship switching platform, with a large selection of models spanning access, distribution, and core layers.
- **Nexus** – high-end switches focused at datacenter environments.

Depending on the brand and model, Cisco supports one of three switch operating systems:

- **Catalyst OS (CatOS)** - interface based on *set* commands, that is almost entirely deprecated. CatOS will not be covered in this guide.
- **IOS** – interface that is nearly identical to the Cisco router IOS, except for switching-specific commands.
- **NX-OS** – interface supported exclusively on Nexus brand switches.

The following details the supported operating system for various Cisco platforms:

<u>CatOS</u>	<u>IOS</u>	<u>NX-OS</u>
• Catalyst 1200	• Catalyst 29xx	• Nexus 1000V
• Catalyst 2948	• Catalyst 35xx	• Nexus 3000
• Catalyst 4000	• Catalyst 36xx	• Nexus 4000
• Catalyst 4500	• Catalyst 37xx	• Nexus 5000
• Catalyst 5000	• Catalyst 38xx	• Nexus 7000
• Catalyst 5500	• Catalyst 4500	• Nexus 9000
• Catalyst 6000	• Catalyst 4900	
• Catalyst 6500	• Catalyst 6500	
	• Catalyst 6800	

The basic IOS interface is nearly identical between switches and routers, and is covered in *great* detail in other guides on this site:

- [Introduction to the Cisco IOS](#)
- [Advanced IOS Functions](#)

This guide will focus on switch-specific IOS functions. For a more comprehensive IOS review, consult the guides listed above.

* * *

All original material copyright © 2014 by Aaron Balchunas (aaron@routeralley.com), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

Switch Port Configuration

Traditionally, network connections on switches have been referred to as **ports**, while on routers they are referred to as **interfaces**. However, in the Cisco IOS, switch ports are referred to as interfaces as well.

Additionally, most Cisco switches are **modular**. Thus, configuration commands must reflect both the *module* and *interface* number, in the following format: *module/interface*

Some catalyst switches support being **stacked** – essentially, multiple physical switches connected together to form one logical switch. Configuration commands must reflect the *stack*, *module*, and *interface* number, in the following format: *stack/module/interface*

To enter interface configuration mode for the *third* Fast Ethernet interface off of the *second* module:

```
Switch(config)# interface FastEthernet 2/3
Switch(config-if)#
```

Note that most switches will number their modules and interfaces starting at *1*, while most routers will number their modules/interfaces starting at *0*.

The above command can be *shortened*, as long as the truncated command is not ambiguous:

```
Switch(config)# interface fa 2/3
```

If the interface above was Gigabit Ethernet instead of Fast Ethernet:

```
Switch(config)# interface GigabitEthernet 2/3
Switch(config)# interface gi 2/3
```

Note: On Nexus switches using the NX-OS, the speed of an Ethernet interface is not used to identify it – all interfaces are simply *Ethernet*:

```
NexusSwitch(config)# interface Ethernet 2/3
```

Multiple interfaces can be configured simultaneously:

```
Switch(config)# interface range gi2/3 , gi2/5 , gi2/7
Switch(config-if-range)#
```

The above command selects interfaces *gi2/3*, *gi2/5*, *gi2/7*. Please note that there is a space on either side of each comma.

* * *

All original material copyright © 2014 by Aaron Balchunas (aaron@routeralley.com), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

Switch Port Configuration (continued)

A contiguous range of interfaces can be specified:

```
Switch(config)# interface range gi3/10 - 15
```

The above command selects interfaces *gi3/10* through *gi3/15*. Please note the space on either side of the dash.

Macros can be created for groups of interfaces that are configured often:

```
Switch(config)# define interface-range MACRONAME gi3/10 - 15
Switch(config)# interface range macro MACRONAME
Switch(config-if-range)#
```

The first command creates a macro (or group) of interfaces called *MACRONAME*. The second command actually selects those interfaces for configuration.

Descriptions can be applied to an interface for documentation purposes:

```
Switch(config)# interface gi3/10
Switch(config-if)# description WEBSERVER
```

Spaces are allowed in a description:

```
Switch(config)# interface gi2/10
Switch(config-if)# description CHECK PRINTER IN PAYROLL
```

Descriptions can greatly assist in troubleshooting issues, as long as they are accurate. Descriptions on access-layer switches can be difficult to manage, as hosts may change interfaces often.

Scenarios where adding descriptions to interfaces is valuable:

- Switch or router uplinks
- WAN connections
- Datacenter hosts – such as servers or firewalls

Descriptions should be as detailed as possible. For example, including the circuit number on a WAN connection can accelerate resolution of an outage.

If an interface *does not* have an accurate description, there are two methods of determining what is connected to it:

- Trace the physical cable to the host (always fun)
- Leverage the CAM table to identify a host by its MAC address

* * *

All original material copyright © 2014 by Aaron Balchunas (aaron@routeralley.com), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

Switch Port Configuration – Speed and Duplex

Some switch interfaces are **fixed** at a single speed. Often though, a single switch interface can support multiple speeds.

For example, a Gigabit Ethernet interface is often backwards-compatible with original and Fast Ethernet, and is referred to as a *10/100/1000* interface.

To manually specify the **speed** of an interface:

```
Switch(config)# interface gi3/10
Switch(config-if)# speed 10
Switch(config-if)# speed 100
Switch(config-if)# speed 1000
```

The **duplex** of an interface can be manually specified as well:

```
Switch(config)# interface gi3/10
Switch(config-if)# duplex half
Switch(config-if)# duplex full
```

Ethernet also has the ability to **autonegotiate** both the speed and duplex of an interface. Autonegotiation will attempt to use the *fastest* speed available, and will attempt to use *full-duplex* if both devices support it.

Only one command is required to force an interface to autonegotiate its speed and duplex:

```
Switch(config)# interface gi3/10
Switch(config-if)# speed auto
```

The configuration *must* be consistent on both sides of the connection. Both sides must be configured to autonegotiate, or both sides must be hardcoded with *identical* settings. Otherwise a **duplex mismatch** error can occur.

For example, if a workstation's interface is configured to autonegotiate, and the switch interface is hardcoded for 100Mbps and full-duplex, then a duplex mismatch will occur. The workstation's interface will sense the correct speed of 100Mbps, but will not detect the correct duplex and will default to *half-duplex*.

If the duplex is mismatched, collisions will occur. Because the full-duplex side of the connection does not utilize CSMA/CD, performance is *severely* degraded. These issues can be difficult to troubleshoot, as the network connection will still function, but will be excruciatingly slow.

* * *

All original material copyright © 2014 by Aaron Balchunas (aaron@routeralley.com), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

Switch Port Configuration – Speed and Duplex (continued)

When autonegotiation was first developed, manufacturers did not always adhere to the same standard. This resulted in frequent mismatch issues, and a sentiment of distrust towards autonegotiation.

Though modern network hardware has alleviated most of the incompatibility, many administrators are still skeptical of autonegotiation and choose to hardcode all connections. Another common practice is to hardcode server and datacenter connections, but to allow user devices to autonegotiate.

Gigabit Ethernet provided several enhancements to autonegotiation, such as hardware flow control. Most manufacturers **recommend autonegotiation** on Gigabit Ethernet interfaces as a best practice.

Switch Port Configuration - Shutdown

By default on most Cisco switches, all interfaces will be in an **administratively shutdown** state. An interface that is shutdown will not send or receive traffic.

To take an interface out of a shutdown state:

```
Switch(config)# interface gi3/10
Switch(config-if)# no shutdown
```

Intuitively, to place the interface into a shutdown state again:

```
Switch(config)# interface gi3/10
Switch(config-if)# shutdown
```

It is possible to shut or no shut multiple interface simultaneously, using the range command:

```
Switch(config)# interface range gi3/10 – 15
Switch(config-if-range)# no shut
```

Be careful – do not unintentionally shutdown the wrong interfaces when using the range command.

It is possible to apply any configuration to an interface that is shutdown. However, the interface (and corresponding configuration) will not become active until the interface is *no shut*.

* * *

All original material copyright © 2014 by Aaron Balchunas (aaron@routeralley.com), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

Switch Port Error Conditions

Cisco switches can detect **error conditions** on an interface. If the error is severe enough, the switch can automatically disable that interface. An interface that is disabled due to an error is considered to be in an **errdisable state**.

The following events can put a port into errdisable state:

arp-inspection	Error with dynamic ARP inspection, used to prevent ARP poisoning.
bpduguard	BPDU is received on an interface configured for STP Portfast and BPDU guard.
channel-misconfig	Error with an Etherchannel configuration
dhcp-rate-limit	Error with DHCP Snooping
dtp-flap	<i>Flapping</i> between trunk encapsulations (ISL or 802.1Q)
gbic-invalid	Invalid SFP or GBIC module
ilpower	Error with inline power
loopback	Interface is <i>looped</i> (received a keepalive packet that it sent out).
l2tpguard	Error with L2 tunneling
link-flap	Interface is flapping between an <i>up</i> or <i>down</i> state
mac-limit	Violation of MAC address limit
pagp-flap	Flapping of an Etherchannel during PaGP negotiation, usually due to misconfiguration
psecure-violation	Violation of port-security
rootguard	BPDU from a root bridge received on a non-designated port
security-violation	Violation of 802.1x security
storm-control	Broadcast storm detected
udld	Unidirectional link detected
unicast-flood	Violation of unicast flooding limit

(Reference: <http://www.cisco.com/c/en/us/support/docs/lan-switching/spanning-tree-protocol/69980-errdisable-recovery.html>)

* * *

All original material copyright © 2014 by Aaron Balchunas (aaron@routeralley.com), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

Switch Port Error Conditions (continued)

All errdisable conditions are enabled by default. To disable a specific errdisable condition:

```
Switch(config)# no errdisable detect cause link-flap
```

It is also possible to disable *all* errdisable conditions:

```
Switch(config)# no errdisable detect cause all
```

To enable errdisable conditions again, either individually or collectively:

```
Switch(config)# errdisable detect cause udd
Switch(config)# errdisable detect cause all
```

An interface can be *manually* recovered from an errdisable state by performing a *shutdown* and then *no shutdown*:

```
Switch(config)# interface gi3/10
Switch(config-if)# shut
Switch(config-if)# no shut
```

However, if the error condition still exists, the interface will be placed back into an errdisable state again. Thus, an interface should not be recovered until the root cause is resolved.

Interfaces can also *automatically* recover from an error condition. First, the errdisable conditions that should auto-recover must be individually or collectively identified:

```
Switch(config)# errdisable recovery cause udd
Switch(config)# errdisable recovery cause all
```

By default, an interface will recover from an errdisable state in **300 seconds**. This timer can be adjusted:

```
Switch(config)# errdisable recovery interval 250
```

To view which errdisable conditions are currently enabled for recovery:

```
Switch# show errdisable recovery
```

By default, automatic errdisable recovery is **disabled for all conditions**.

Note that all errdisable commands are applied *globally*, and thus apply to all interfaces on the switch.

* * *

All original material copyright © 2014 by Aaron Balchunas (aaron@routeralley.com), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

Troubleshooting Switch Ports

Cisco switches offer several *show* commands for monitoring and troubleshooting switch ports.

The *show interface* is invaluable, and provides the following information:

- Current interface status – *up, down, administratively down, errdisable*
- Speed and duplex
- Current input/output rates
- Historical packet input/output and error statistics

To view the details of a specific interface, such as GigabitEthernet 3/10:

Switch# *show interface gi3/10*

```
GigabitEthernet3/10 is up, line protocol is up (connected)
Hardware is Gigabit Ethernet Port, address is 2c54.2db2.dab1
MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
Keepalive set (10 sec)
Full-duplex, 100Mb/s, link type is auto, media type is 10/100/1000-TX
input flow-control is on, output flow-control is on
Auto-MDIX on (operational: on)
ARP type: ARPA, ARP Timeout 04:00:00
Last input 00:00:25, output never, output hang never
Last clearing of "show interface" counters 1y9w
Input queue: 0/2000/0/0 (size/max/drops/flush); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 6000 bits/sec, 8 packets/sec
 190197453 packets input, 22130686309 bytes, 0 no buffer
  Received 1660069 broadcasts (1295453 multicasts)
  0 runts, 110243 giants, 0 throttles
 110243 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input packets with dribble condition detected
 518261888 packets output, 352807215573 bytes, 0 underruns
  0 output errors, 0 collisions, 0 interface resets
  0 unknown protocol drops
  0 babbles, 0 late collision, 0 deferred
  0 lost carrier, 0 no carrier
  0 output buffer failures, 0 output buffers swapped out
```

To view details of *all* interfaces:

Switch# *show interface*

All original material copyright © 2014 by Aaron Balchunas (aaron@routeralley.com), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

Troubleshooting Switch Ports (continued)

The *show interface* command provides an occasionally overwhelming amount of detailed information.

The *show interface status* command provides a high-level summary of all switch interfaces:

Switch# *show interface status*

Port	Name	Status	Vlan	Duplex	Speed	Type
Gi2/1		connected	15	a-full	a-100	10/100/1000-TX
Gi2/2		notconnect	15	auto	auto	10/100/1000-TX
Gi2/3		connected	15	a-full	a-100	10/100/1000-TX
Gi2/4		connected	15	a-full	a-1000	10/100/1000-TX
Gi2/5		connected	15	a-half	a-10	10/100/1000-TX
Gi2/6		notconnect	15	auto	auto	10/100/1000-TX
Gi2/7		connected	15	a-full	a-100	10/100/1000-TX
Gi2/8		connected	15	a-full	a-100	10/100/1000-TX
Gi2/9		connected	15	a-full	a-100	10/100/1000-TX
Gi2/10		notconnect	15	auto	auto	10/100/1000-TX
Gi2/11		notconnect	9	auto	auto	10/100/1000-TX

<snipped for brevity>

To view which interfaces are currently in an errdisable state, including the reason for the errdisable condition:

Switch# *show interface status err-disabled*

Port	Name	Status	Reason
Gi3/1		err-disabled	rootguard
Gi3/3		err-disabled	ilpower
Gi3/8		err-disabled	security-violation
Gi3/9		err-disabled	grue-eats-your-face

Watch out for the grue.

* * *

All original material copyright © 2014 by Aaron Balchunas (aaron@routeralley.com), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.