

BCMSN

# Mega Guide

## Prepare With Confidence

This PrepLogic Mega Guide was written by certified subject matter experts and published authors to provide you accurate, in-depth exam coverage. All exam objectives are covered in detail, giving you the knowledge and confidence you need to pass your exam.



**PrepLogic**

*Be Prepared. Be Confident. Get Certified.*

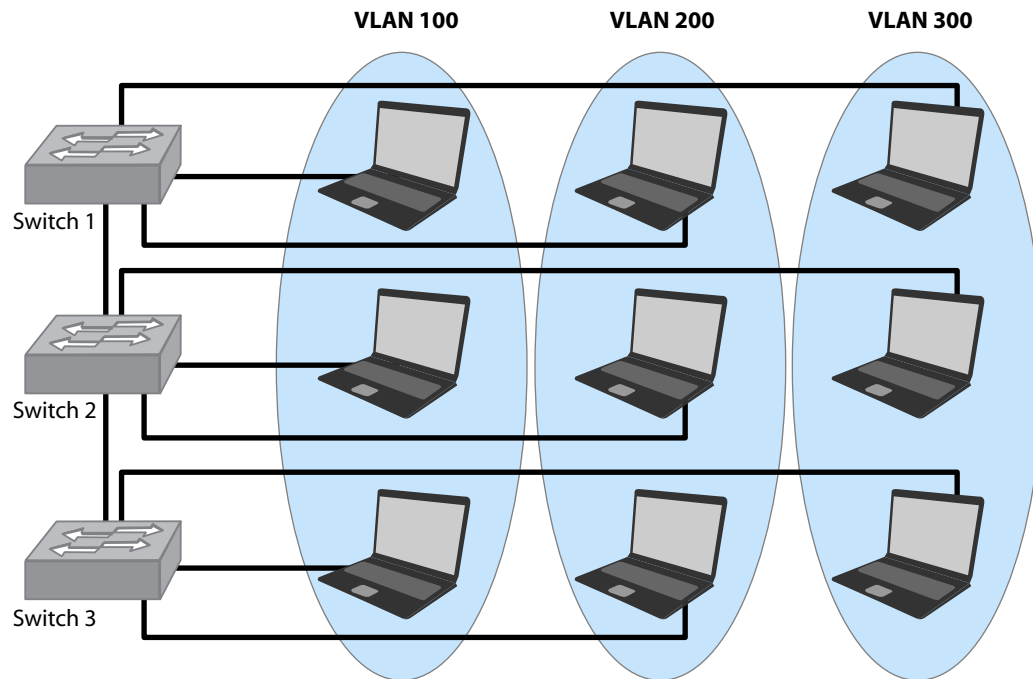


Andrew Froehlich - Author  
Gene Bagwell - Technical Editor

# Implementing VLANs

## VLANs in a Hierarchical Network

A virtual LAN (VLAN) is a logical network that allows a group of devices to act as if they are on the same physical network. All devices within a VLAN share the same unicast, broadcast and multicast domain. Networked devices can be connected to different switches in different locations and still be on the same VLAN. This allows for a great amount of flexibility as seen in *Figure 1*. Communication between VLANs must pass through a layer 3 device such as a router.



**Figure 1**

All devices within a particular VLAN are typically in the same IP subnet. For example, all PC's in VLAN 100 have an IP address within the 192.168.100.X/24 range while PC's in VLAN 200 have an IP address of 192.168.200.X/24. The IP subnet is configured either on a router interface or a Layer 3 switch VLAN interface. These VLAN interfaces are also known as switch virtual interfaces (SVI).

In a typical 3 tier Cisco architecture, the VLAN SVIs are configured at the distribution layer. That effectively creates the layer 3 gateways on the VLAN interfaces. Layer 2 physical interfaces are then configured to connect to the access layer switches. If multiple VLANs are required on a particular switch, a trunk is configured to switch multiple VLANs across the same physical connection.

## Configuring VLANs

The Cisco recommended method to create and verify a VLAN is as follows:

Syntax:

```
Switch# configure terminal  
Switch(config)# vlan vlan-id  
Switch(config-vlan)# name vlan-name  
Switch(config-vlan)# end  
Switch# show vlan {name vlan-name | id vlan-id}
```

So let's say that we want to create VLAN 200 on our switch and name it "Accounting". Here's how it would be configured:

Example:

```
Switch# configure terminal  
Switch(config)# vlan 200  
Switch(config-vlan)# name Accounting  
Switch(config-vlan)# end  
Switch# show vlan Accounting (or show vlan 200)
```

A second method that can be used on a Cisco switch is by entering the VLAN database in privileged EXEC mode:

Syntax:

```
Switch# vlan database  
Switch(vlan)# vlan vlan-id name vlan-name  
Switch(vlan)# exit  
Switch# show vlan {name vlan-name | id vlan-id}
```

So to create a VLAN 200 with the name Accounting using the VLAN database method, it would look like this:

```
Switch# vlan database  
Switch(vlan)# vlan 200 name Accounting  
Switch(vlan)# exit  
Switch# show vlan Accounting (or show vlan 200)
```

**Note:** You cannot add extended-range VLANs in VLAN database configuration mode. Normal-range vlans are 1-1005 with 1002-1005 being reserved for token-ring VLANs. Extended-range VLANs (1006-4094) are configured using the config-vlan method.

Now that we know how to build a VLAN, we can configure specific switch interfaces so end devices can become part of the VLAN. By default, all switch ports are placed in VLAN 1. There are multiple ways to assign VLAN membership to a port. The first and most widely used method is static assignment. Below is the syntax to statically assign a port to a VLAN and verify the configuration:

Syntax:

```
Switch# configure terminal
Switch(config)# interface interface-id
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan vlan-id
Switch(config-if)# end
Switch# show interfaces interface-id switchport
```

Using our example above, let's statically assign switch port fa0/10 to VLAN 200:

Example:

```
Switch# configure terminal
Switch(config)# interface fastethernet0/10
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 200
Switch(config-if)# end
```

## VLAN Trunking

When connecting multiple switches together, a trunk link can be used to transport traffic from multiple VLANs across a single point-to-point link. This is called a trunk link. There are two ways to configure a trunk on Cisco devices. Cisco has their own proprietary trunking protocol called Inter-switch link (ISL). The other method is 802.1q which is a non-proprietary IEEE standard.

## Configuring ISL Trunks

The Inter-switch link protocol is a method of tagging a switch frame with the VLAN identification number so connected switches can identify the VLAN numbering correctly and to transfer end device communication from one switch to another on the same VLAN. ISL prepends a 26 byte header, including a 10 bit VLAN ID, to the frame header. It also appends a 4 byte CRC to the end of the frame. This effectively "encapsulates" the VLAN tagged frame. There are three steps to configuring an ISL trunk link on each side of the switch. The steps are:

1. Set the trunk encapsulation
2. Set the trunk mode
3. Set the default VLAN

The trunk encapsulation method must be the same on both sides of the trunk. A trunk port can be configured to negotiate the encapsulation method using the negotiation command.

The mode can be hard-coded to trunk or use the dynamic desirable/ dynamic auto methods to dynamically transition to a trunking state. If both sides of the link are set to dynamic desirable, a trunk will form. If both sides are set to auto, the trunk will not form. If one side of the trunk is auto and the other is desirable, the trunk will form. And if one side is set to trunk while the other is either desirable or auto, the trunk will form. The default mode for switch interfaces is dynamic desirable.

The default VLAN command is optional but is a good idea to do. It dictates what VLAN is used if the interface stops trunking. This is important if your trunk links are set to dynamic desirable or dynamic auto.

Figure 2 will serve as the example to configuring an ISL trunk between 2 switches. VLANs 100 and 200 are configured on both of the switches and we want the VLANs to be trunked using ISL for seamless communication of end devices on the same VLAN.

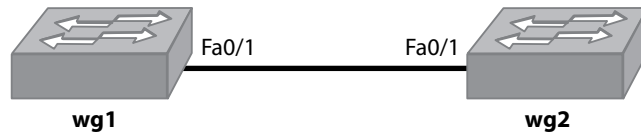


Figure 2

Syntax:

```
Switch# configure terminal  
Switch(config)# interface interface-id  
Switch(config-if)# switchport trunk encapsulation {isl | dot1q | negotiate}  
Switch(config-if)# switchport mode {dynamic {auto | desirable} | trunk}  
Switch(config-if)# switchport access vlan vlan-id
```

The mode can be hard-coded to trunk or use the dynamic desirable/dynamic auto methods to dynamically transition to a trunking state. If both sides of the link are set to dynamic desirable, a trunk will form. If both sides are set to auto, the trunk will not form. If one side of the trunk is auto and the other is desirable, the trunk will form. And if one side is set to trunk while the other is either desirable or auto, the trunk will form. The default mode for switch interfaces is dynamic desirable.

The default VLAN command is optional but is a good idea to do. It dictates what VLAN is used if the interface stops trunking. This is important if your trunk links are set to dynamic desirable or dynamic auto:

Example:

Because we know we want an ISL trunk link between wg1 and wg2 in Figure 2, the encapsulation will be set to ISL and the mode will be hard-coded to trunk on both sides. Even though the port will be hard-coded the default VLAN will be set to 100.

Switch wg1:

```
wg1# configure terminal  
wg1(config)# interface fa0/1  
wg1(config-if)#switchport trunk encapsulation isl  
wg1(config-if)#switchport mode trunk  
wg1(config-if)#switchport access vlan 100
```

Switch wg2:

```
wg2# configure terminal
wg2(config)# interface fa0/1
wg2(config-if)#switchport trunk encapsulation isl
wg2(config-if)#switchport mode trunk
wg2(config-if)#switchport access vlan 100
```

To verify our trunk we can do a show interfaces trunk:

```
wg1#show interfaces trunk

Port    Mode    Encapsulation    Status    Native vlan
Fa0/1   on      isl               trunking   1

Port    Vlans allowed on trunk
Fa0/1   1-4094

Port    Vlans allowed and active in management domain
Fa0/1   1,100,200

Port    Vlans in spanning tree forwarding state and not pruned
Fa0/1   1,100,200
```

The output shows that the trunk status is turning with ISL encapsulation. All VLANs that are configured on the switches are allowed to traverse the trunk but only VLAN 1, 100 and 200 are currently active on the switch.

To limit the VLANs that are used on the trunk, the **switchport trunk allowed vlan** *vlan-id* command could be used. In our example, we want to limit the VLANs trunked to 100 and 200.

```
wg1(config-if)#switchport trunk allowed vlan 100,200
```

Now, looking at the show interfaces trunk command, we see that only the two VLANs are allowed.

```
wg1#show interfaces trunk

Port    Mode    Encapsulation    Status    Native vlan
Fa0/1   on      isl               trunking   1

Port    Vlans allowed on trunk
Fa0/1   100,200

Port    Vlans allowed and active in management domain
Fa0/1   100,200

Port    Vlans in spanning tree forwarding state and not pruned
Fa0/1   100,200
```