CHAPTER 12

Basic Interface Configuration (ASAv)

This chapter includes tasks for starting your interface configuration for the ASAv, including configuring Ethernet settings, redundant interfaces, and VLAN subinterfaces.

This chapter includes the following sections:

- Information About Starting ASAv Interface Configuration, page 12-1
- Licensing Requirements for ASAv Interfaces, page 12-6
- Guidelines and Limitations, page 12-6
- Default Settings, page 12-7
- Starting Interface Configuration (ASAv), page 12-7
- Monitoring Interfaces, page 12-14
- Configuration Examples for ASAv Interfaces, page 12-14
- Where to Go Next, page 12-15
- Feature History for ASAv Interfaces, page 12-15

Information About Starting ASAv Interface Configuration

This section includes the following topics:

- ASAv Interfaces and Virtual NICs, page 12-1
- Interfaces in Transparent Mode, page 12-2
- Management Interface, page 12-2
- Redundant Interfaces, page 12-3
- Controlling Fragmentation with the Maximum Transmission Unit and TCP Maximum Segment Size, page 12-4

ASAv Interfaces and Virtual NICs

As a guest on a virtualized platform, the ASAv utilizes the network interfaces of the underlying physical platform. Each ASAv interface maps to a VMware virtual NIC (vNIC).

- ASAv Interfaces, page 12-2
- Supported vNICs, page 12-2

- ASAv Interface Concordance with vNICs, page 12-2

ASAv Interfaces

The ASAv includes the following Gigabit Ethernet interfaces:

- Management 0/0
- GigabitEthernet 0/0 through 0/8. Note that the GigabitEthernet 0/8 is used for the failover link when you deploy the ASAv as part of a failover pair.

Supported vNICs

VMware supports the following vNIC for ASAv interfaces:

- E1000—This vNIC is used by default.

ASAv Interface Concordance with vNICs

The vSphere Client Virtual Machine Properties screen (right-click the ASAv instance, and choose **Edit Settings**) shows each Network Adapter and the assigned network. However, that screen does not show the ASAv interface IDs (only Network Adapter IDs). See the following concordance of Network Adapter IDs and ASAv IDs:

Network Adapter ID	ASAv Interface ID
Network Adapter 1	Management0/0
Network Adapter 2	GigabitEthernet0/0
Network Adapter 3	GigabitEthernet0/1
Network Adapter 4	GigabitEthernet0/2
Network Adapter 5	GigabitEthernet0/3
Network Adapter 6	GigabitEthernet0/4
Network Adapter 7	GigabitEthernet0/5
Network Adapter 8	GigabitEthernet0/6
Network Adapter 9	GigabitEthernet0/7
Network Adapter 10	GigabitEthernet0/8

Interfaces in Transparent Mode

Interfaces in transparent mode belong to a “bridge group,” one bridge group for each network. You can have up to 8 bridge groups of 4 interfaces. For more information about bridge groups, see Bridge Groups in Transparent Mode, page 14-1.

Management Interface

- Management Interface Overview, page 12-3
- Using Any Interface for Management-Only Traffic, page 12-3

- Management Interface for Transparent Mode, page 12-3
- No Through Traffic Support, page 12-3

Management Interface Overview

You can manage the ASA by connecting to:

- Any through-traffic interface
- The dedicated Management 0/0 interface

You may need to configure management access to the interface according to Chapter 43, "Management Access."

Using Any Interface for Management-Only Traffic

You can use any interface as a dedicated management-only interface by configuring it for management traffic (see the **management-only** command).

Management Interface for Transparent Mode

In transparent firewall mode, in addition to the maximum allowed through-traffic interfaces, you can also use the Management 0/0 interface (either the physical interface or a subinterface) as a separate management interface. You cannot use any other interface types as management interfaces. The management interface is not part of a normal bridge group. Note that for operational purposes, it is part of a non-configurable bridge group.

Note In transparent firewall mode, the management interface updates the MAC address table in the same manner as a data interface; therefore you should not connect both a management and a data interface to the same switch unless you configure one of the switch ports as a routed port (by default Cisco Catalyst switches share a MAC address for all VLAN switch ports). Otherwise, if traffic arrives on the management interface from the physically-connected switch, then the ASA updates the MAC address table to use the *management* interface to access the switch, instead of the data interface. This action causes a temporary traffic interruption; the ASA will not re-update the MAC address table for packets from the switch to the data interface for at least 30 seconds for security reasons.

No Through Traffic Support

The Management 0/0 interface is always set to management-only; you cannot use this interface for through traffic support.

Redundant Interfaces

A logical redundant interface consists of a pair of physical interfaces: an active and a standby interface. When the active interface fails, the standby interface becomes active and starts passing traffic. You can configure a redundant interface to increase the ASA reliability. This feature is separate from device-level failover, but you can configure redundant interfaces as well as device-level failover if desired.

Redundant Interface MAC Address

The redundant interface uses the MAC address of the first physical interface that you add. If you change the order of the member interfaces in the configuration, then the MAC address changes to match the MAC address of the interface that is now listed first. Alternatively, you can assign a MAC address to the redundant interface, which is used regardless of the member interface MAC addresses (see Configuring the MAC Address, MTU, and TCP MSS, page 13-9 or the Configuring Multiple Contexts, page 7-15). When the active interface fails over to the standby, the same MAC address is maintained so that traffic is not disrupted.

Controlling Fragmentation with the Maximum Transmission Unit and TCP Maximum Segment Size

- MTU Overview, page 12-4
- Default MTU, page 12-4
- Path MTU Discovery, page 12-4
- Setting the MTU and Jumbo Frames, page 12-5
- TCP Maximum Segment Size Overview, page 12-5
- Default TCP MSS, page 12-5
- Setting the TCP MSS for VPN and Non-VPN Traffic, page 12-5
- Examples, page 12-5

MTU Overview

The maximum transmission unit (MTU) specifies the maximum frame payload size that the ASA can transmit on a given Ethernet interface. The MTU value is the frame size *without* Ethernet headers, FCS, or VLAN tagging. The Ethernet header is 14 bytes and the FCS is 4 bytes. When you set the MTU to 1500, the expected frame size is 1518 bytes including the headers. If you are using VLAN tagging (which adds an additional 4 bytes), then when you set the MTU to 1500, the expected frame size is 1522. Do not set the MTU value higher to accommodate these headers. For information about accommodating TCP headers for encapsulation, do not alter the MTU setting; instead change the TCP Maximum Segment Size (the TCP Maximum Segment Size Overview, page 12-5).

Note The ASA can receive frames larger than the configured MTU as long as there is room in memory. See Enabling Jumbo Frame Support, page 12-13 to increase memory for larger frames.

Default MTU

The default MTU on the ASA is 1500 bytes. This value does not include the 18 or more bytes for the Ethernet header, CRC, VLAN tagging, and so on.

Path MTU Discovery

The ASA supports Path MTU Discovery (as defined in RFC 1191), which lets all devices in a network path between two hosts coordinate the MTU so that they can standardize on the lowest MTU in the path.

Setting the MTU and Jumbo Frames

See Configuring the MAC Address, MTU, and TCP MSS, page 13-9.

See Enabling Jumbo Frame Support, page 12-13.

See the following guidelines:

- Matching MTUs on the traffic path—We recommend that you set the MTU on all ASA interfaces and other device interfaces along the traffic path to be the same. Matching MTUs prevents intermediate devices from fragmenting the packets.
- Accommodating jumbo frames—If you enable jumbo frames, you can set the MTU up to 9000 bytes.

TCP Maximum Segment Size Overview

The TCP maximum segment size (TCP MSS) is the size of the TCP payload *before* any TCP headers are added. UDP packets are not affected. The client and the server exchange TCP MSS values during the three-way handshake when establishing the connection.

You can set the TCP MSS on the ASA. If either endpoint of a connection requests a TCP MSS that is larger than the value set on the ASA, the ASA overwrites the TCP MSS in the request packet with the ASA maximum. If the host or server does not request a TCP MSS, then the ASA assumes the RFC 793-default value of 536 bytes, but does not modify the packet. You can also configure the minimum TCP MSS; if a host or server requests a very small TCP MSS, the ASA can adjust the value up. By default, the minimum TCP MSS is not enabled.

For example, you configure the default MTU of 1500 bytes. A host requests an MSS of 1700. If the ASA maximum TCP MSS is 1380, then the ASA changes the MSS value in the TCP request packet to 1380. The server then sends 1380-byte packets.

Default TCP MSS

By default, the maximum TCP MSS on the ASA is 1380 bytes. This default accommodates VPN connections where the headers can add up to 120 bytes; this value fits within the default MTU of 1500 bytes.

Setting the TCP MSS for VPN and Non-VPN Traffic

See Configuring the MAC Address, MTU, and TCP MSS, page 13-9.

See the following guidelines:

- Non-VPN traffic—If you do not use VPN and do not need extra space for headers, then you should disable the TCP MSS limit and accept the value established between connection endpoints. Because connection endpoints typically derive the TCP MSS from the MTU, non-VPN packets usually fit this TCP MSS.
- VPN traffic—Set the maximum TCP MSS to the MTU - 120. For example, if you use jumbo frames and set the MTU to a higher value, then you need to set the TCP MSS to accommodate the new MTU.

Examples

The following example enables jumbo frames, increases the MTU on all interfaces, and disables the TCP MSS for non-VPN traffic (by setting the TCP MSS to 0, which means there is no limit):

```
jumbo frame-reservation
mtu inside 9000
mtu outside 9000
sysopt connection tcpmss 0
```

The following example enables jumbo frames, increases the MTU on all interfaces, and changes the TCP MSS for VPN traffic to 8880 (the MTU minus 120):

```
jumbo frame-reservation
mtu inside 9000
mtu outside 9000
sysopt connection tcpmss 8880
```

Licensing Requirements for ASAv Interfaces

Model	License Requirement
ASAv with 1 Virtual CPU	VLANs[1]: Standard and Premium License: 50 Interfaces of all types[2]: Standard and Premium License: 716
ASAv with 4 Virtual CPUs	VLANs[1]: Standard and Premium License: 200 Interfaces of all types[2]: Standard and Premium License: 1316

[1] For an interface to count against the VLAN limit, you must assign a VLAN to it. For example:
interface gigabitethernet 0/0.100
vlan 100

[2] The maximum number of combined interfaces; for example, VLANs, physical, redundant, and bridge group interfaces. Every **interface** command defined in the configuration counts against this limit.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

Firewall Mode Guidelines

- For transparent mode, you can configure up to 8 bridge groups.
- Each bridge group can include up to 4 interfaces.

Failover Guidelines

- When you use a redundant interface as a failover link, it must be pre-configured on both units in the failover pair; you cannot configure it on the primary unit and expect it to replicate to the secondary unit because *the failover link itself is required for replication*.
- If you use a redundant interface for the state link, no special configuration is required; the configuration can replicate from the primary unit as normal.

- You can monitor redundant interfaces for failover using the **monitor-interface** command; be sure to reference the logical redundant interface name. When an active member interface fails over to a standby interface, this activity does not cause the redundant interface to appear to be failed when being monitored for device-level failover. Only when all physical interfaces fail does the redundant interface appear to be failed.
- You cannot share a failover or state interface with a data interface.

Redundant Interface Guidelines

- You can configure up to 8 redundant interface pairs.
- All ASA configuration refers to the logical redundant interface instead of the member physical interfaces.
- If you shut down the active interface, then the standby interface becomes active.
- You cannot set a redundant interface as management-only.
- For failover guidelines, see Failover Guidelines, page 12-6.

Default Settings

This section lists default settings for interfaces if you do not have a factory default configuration. For information about the factory default configurations, see Factory Default Configurations, page 4-18.

Default State of Interfaces

- Physical interfaces—Disabled.
- Redundant Interfaces—Enabled. However, for traffic to pass through the redundant interface, the member physical interfaces must also be enabled.
- Subinterfaces—Enabled. However, for traffic to pass through the subinterface, the physical interface must also be enabled.

Default Speed and Duplex

- By default, the speed and duplex for interfaces are set to auto-negotiate.

Default MAC Addresses

By default, the physical interface uses the burned-in MAC address, and all subinterfaces of a physical interface use the same burned-in MAC address.

Default vNIC

All interfaces use the E1000 emulation.

Starting Interface Configuration (ASAv)

This section includes the following topics:

- Task Flow for Starting Interface Configuration, page 12-8
- Enabling the Physical Interface and Configuring Ethernet Parameters, page 12-8
- Enabling the Physical Interface and Configuring Ethernet Parameters, page 12-8
- Configuring a Redundant Interface, page 12-10

- Configuring VLAN Subinterfaces and 802.1Q Trunking, page 12-12
- Enabling Jumbo Frame Support, page 12-13

Task Flow for Starting Interface Configuration

To start configuring interfaces, perform the following steps:

Step 1 Enable the physical interface, and optionally change Ethernet parameters. See Enabling the Physical Interface and Configuring Ethernet Parameters, page 12-8.

Physical interfaces are disabled by default.

Step 2 (Optional) Configure redundant interface pairs. See Configuring a Redundant Interface, page 12-10.

A logical redundant interface pairs an active and a standby physical interface. When the active interface fails, the standby interface becomes active and starts passing traffic.

Step 3 (Optional) Configure VLAN subinterfaces. See Configuring VLAN Subinterfaces and 802.1Q Trunking, page 12-12.

Step 4 (Optional) Enable jumbo frame support according to the Enabling Jumbo Frame Support, page 12-13.

Enabling the Physical Interface and Configuring Ethernet Parameters

This section describes how to:

- Enable the physical interface
- Set a specific speed and duplex
- Enable pause frames for flow control

Detailed Steps

	Command	Purpose

Step 1	`interface` *`physical_interface`*
	Example:
	`ciscoasa(config)# interface gigabitethernet 0/0`
		Specifies the interface that you want to configure.
		where the *physical_interface* ID includes the type, slot, and port number as *type*[*slot*/]*port*.
		The physical interface types include the following:
		• **gigabitethernet**
		• **management**
		Enter the type followed by *slot*/*port*, for example, **gigabitethernet0/1**. A space is optional between the type and the slot/port.

Step 2	(Optional) `speed {auto	10	100	1000}` **Example:** `ciscoasa(config-if)# speed 100`	Sets the speed. The default setting is **auto**.

	Command	Purpose

Step 3	(Optional)
	duplex {**auto** | **full** | **half**}
	Example:
	`ciscoasa(config-if)# duplex full`
		Sets the duplex. The **auto** setting is the default.

Step 4	(Optional) **flowcontrol send on** [*low_water high_water pause_time*] [**noconfirm**] **Example:** `ciscoasa(config-if)# flowcontrol send on 95 200 10000`	Enables pause (XOFF) frames for flow control. If you have a traffic burst, dropped packets can occur if the burst exceeds the buffering capacity of the FIFO buffer on the NIC and the receive ring buffers. Enabling pause frames for flow control can alleviate this issue. Pause (XOFF) and XON frames are generated automatically by the NIC hardware based on the FIFO buffer usage. A pause frame is sent when the buffer usage exceeds the high-water mark. The default *high_water* value is 24 KB; you can set it between 0 and 47 KB. After a pause is sent, an XON frame can be sent when the buffer usage is reduced below the low-water mark. By default, the *low_water* value is 16 KB; you can set it between 0 and 47 KB. The link partner can resume traffic after receiving an XON, or after the XOFF expires, as controlled by the timer value in the pause frame. The default *pause_time* value is 26624; you can set it between 0 and 65535. If the buffer usage is consistently above the high-water mark, pause frames are sent repeatedly, controlled by the pause refresh threshold value. When you use this command, you see the following warning: `Changing flow-control parameters will reset the interface. Packets may be lost during the reset. Proceed with flow-control changes?` To change the parameters without being prompted, use the **noconfirm** keyword. **Note** Only flow control frames defined in 802.3x are supported. Priority-based flow control is not supported.
Step 5	**no shutdown** **Example:** `ciscoasa(config-if)# no shutdown`	Enables the interface. To disable the interface, enter the **shutdown** command. If you enter the **shutdown** command, you also shut down all subinterfaces. If you shut down an interface in the system execution space, then that interface is shut down in all contexts that share it.

What to Do Next

Optional Tasks:

- Configure redundant interface pairs. See Configuring a Redundant Interface, page 12-10.
- Configure VLAN subinterfaces. See Configuring VLAN Subinterfaces and 802.1Q Trunking, page 12-12.
- Configure jumbo frame support. See Enabling Jumbo Frame Support, page 12-13.

Required Tasks:

- Complete the interface configuration. See Chapter 13, “Routed Mode Interfaces,” or Chapter 14, “Transparent Mode Interfaces.”

Configuring a Redundant Interface

A logical redundant interface consists of a pair of physical interfaces: an active and a standby interface. When the active interface fails, the standby interface becomes active and starts passing traffic. You can configure a redundant interface to increase the ASA reliability. This feature is separate from device-level failover, but you can configure redundant interfaces as well as failover if desired.

This section describes how to configure redundant interfaces and includes the following topics:

- Configuring a Redundant Interface, page 12-10
- Changing the Active Interface, page 12-11

Configuring a Redundant Interface

This section describes how to create a redundant interface. By default, redundant interfaces are enabled.

Guidelines and Limitations

- You can configure up to 8 redundant interface pairs.
- Redundant interface delay values are configurable, but by default the ASA inherits the default delay values based on the physical type of its member interfaces.
- See also the Redundant Interface Guidelines, page 12-7.

Prerequisites

- Both member interfaces must be of the same physical type. For example, both must be GigabitEthernet.
- You cannot add a physical interface to the redundant interface if you configured a name for it. You must first remove the name using the **no nameif** command.

Caution If you are using a physical interface already in your configuration, removing the name will clear any configuration that refers to the interface.

Detailed Steps

	Command	Purpose
Step 1	**interface redundant** *number* **Example:** `ciscoasa(config)# interface redundant 1`	Adds the logical redundant interface, where the *number* argument is an integer between 1 and 8. **Note** You need to add at least one member interface to the redundant interface before you can configure logical parameters for it such as a name.
Step 2	**member-interface** *physical_interface* **Example:** `ciscoasa(config-if)# member-interface gigabitethernet 0/0`	Adds the first member interface to the redundant interface. After you add the interface, any configuration for it (such as an IP address) is removed.
Step 3	**member-interface** *physical_interface* **Example:** `ciscoasa(config-if)# member-interface gigabitethernet 0/1`	Adds the second member interface to the redundant interface. Make sure the second interface is the same physical type as the first interface. To remove a member interface, enter the **no member-interface** *physical_interface* command. You cannot remove both member interfaces from the redundant interface; the redundant interface requires at least one member interface.

Examples

The following example creates two redundant interfaces:

```
ciscoasa(config)# interface redundant 1
ciscoasa(config-if)# member-interface gigabitethernet 0/0
ciscoasa(config-if)# member-interface gigabitethernet 0/1
ciscoasa(config-if)# interface redundant 2
ciscoasa(config-if)# member-interface gigabitethernet 0/2
ciscoasa(config-if)# member-interface gigabitethernet 0/3
```

What to Do Next

Optional Task:

- Configure VLAN subinterfaces. See Configuring VLAN Subinterfaces and 802.1Q Trunking, page 12-12.
- Configure jumbo frame support. See Enabling Jumbo Frame Support, page 12-13.

Required Tasks:

- Complete the interface configuration. See Chapter 13, “Routed Mode Interfaces,” or Chapter 14, “Transparent Mode Interfaces.”

Changing the Active Interface

By default, the active interface is the first interface listed in the configuration, if it is available. To view which interface is active, enter the following command:

```
ciscoasa# show interface redundantnumber detail | grep Member
```

For example:

```
ciscoasa# show interface redundant1 detail | grep Member
      Members GigabitEthernet0/3(Active), GigabitEthernet0/2
```

To change the active interface, enter the following command:

```
ciscoasa# redundant-interface redundantnumber active-member physical_interface
```

where the **redundant***number* argument is the redundant interface ID, such as **redundant1**.

The *physical_interface* is the member interface ID that you want to be active.

Configuring VLAN Subinterfaces and 802.1Q Trunking

Subinterfaces let you divide a physical or redundant interface into multiple logical interfaces that are tagged with different VLAN IDs. An interface with one or more VLAN subinterfaces is automatically configured as an 802.1Q trunk. Because VLANs allow you to keep traffic separate on a given physical interface, you can increase the number of interfaces available to your network without adding additional physical interfaces or ASAs.

Guidelines and Limitations

- Maximum subinterfaces—To determine how many VLAN subinterfaces are allowed for your model, see Licensing Requirements for ASAv Interfaces, page 12-6.
- Preventing untagged packets on the physical interface—If you use subinterfaces, you typically do not also want the physical interface to pass traffic, because the physical interface passes untagged packets. This property is also true for the active physical interface in a redundant interface pair. Because the physical or redundant interface must be enabled for the subinterface to pass traffic, ensure that the physical or redundant interface does not pass traffic by leaving out the **nameif** command. If you want to let the physical or redundant interface pass untagged packets, you can configure the **nameif** command as usual. See Chapter 13, “Routed Mode Interfaces,” or Chapter 14, “Transparent Mode Interfaces,” for more information about completing the interface configuration.

Detailed Steps

	Command	Purpose
Step 1	**interface** {*physical_interface* | **redundant** *number*}.*subinterface* **Example:** `ciscoasa(config)# interface gigabitethernet 0/1.100`	Specifies the new subinterface. See Enabling the Physical Interface and Configuring Ethernet Parameters, page 12-8 for a description of the physical interface ID. The **redundant** *number* argument is the redundant interface ID, such as **redundant 1**. The *subinterface* ID is an integer between 1 and 4294967293.
Step 2	**vlan** *vlan_id* **Example:** `ciscoasa(config-subif)# vlan 101`	Specifies the VLAN for the subinterface. The *vlan_id* is an integer between 1 and 4094. Some VLAN IDs might be reserved on connected switches, so check the switch documentation for more information. You can only assign a single VLAN to a subinterface, and you cannot assign the same VLAN to multiple subinterfaces. You cannot assign a VLAN to the physical interface. Each subinterface must have a VLAN ID before it can pass traffic. To change a VLAN ID, you do not need to remove the old VLAN ID with the **no** option; you can enter the **vlan** command with a different VLAN ID, and the ASA changes the old ID.

What to Do Next

Optional Task:

- Configure jumbo frame support. See Enabling Jumbo Frame Support, page 12-13.

Required Tasks:

- Complete the interface configuration. See Chapter 13, "Routed Mode Interfaces," or Chapter 14, "Transparent Mode Interfaces."

Enabling Jumbo Frame Support

A jumbo frame is an Ethernet packet larger than the standard maximum of 1518 bytes (including Layer 2 header and FCS), up to 9216 bytes. You can enable support for jumbo frames for all interfaces by increasing the amount of memory to process Ethernet frames. Assigning more memory for jumbo frames might limit the maximum use of other features, such as ACLs. See Controlling Fragmentation with the Maximum Transmission Unit and TCP Maximum Segment Size, page 12-4 for more information.

Prerequisites

- Changes in this setting require you to reload the ASA.
- Be sure to set the MTU for each interface that needs to transmit jumbo frames to a higher value than the default 1500; for example, set the value to 9000 using the **mtu** command. See Configuring the MAC Address, MTU, and TCP MSS, page 13-9.
- Be sure to adjust the TCP MSS, either to disable it for non-VPN traffic (using the **sysopt connection tcpmss 0** command), or to increase it in accord with the MTU according to the Configuring the MAC Address, MTU, and TCP MSS, page 13-9.

Detailed Steps

Command	Purpose
jumbo-frame reservation **Example:** `ciscoasa(config)# jumbo-frame reservation`	Enables jumbo frame support. To disable jumbo frames, use the **no** form of this command.

Examples

The following example enables jumbo frame reservation, saves the configuration, and reloads the ASA:

```
ciscoasa(config)# jumbo-frame reservation
WARNING: this command will take effect after the running-config is saved
and the system has been rebooted. Command accepted.

ciscoasa(config)# write memory
Building configuration...
Cryptochecksum: 718e3706 4edb11ea 69af58d0 0a6b7cb5

70291 bytes copied in 3.710 secs (23430 bytes/sec)
[OK]
ciscoasa(config)# reload
Proceed with reload? [confirm] Y
```

What to Do Next

Complete the interface configuration. See Chapter 13, "Routed Mode Interfaces," or Chapter 14, "Transparent Mode Interfaces."

Monitoring Interfaces

To monitor interfaces, enter one of the following commands:

Command	Purpose
show interface	Displays interface statistics.
show interface ip brief	Displays interface IP addresses and status.

Configuration Examples for ASAv Interfaces

This section includes the following topics:

- Physical Interface Parameters Example, page 12-15
- Subinterface Parameters Example, page 12-15

Physical Interface Parameters Example

The following example configures parameters for the physical interface:

```
interface gigabitethernet 0/1
    speed 1000
    duplex full
    no shutdown
```

Subinterface Parameters Example

The following example configures parameters for a subinterface:

```
interface gigabitethernet 0/1.1
    vlan 101
    no shutdown
```

Where to Go Next

Complete the interface configuration according to Chapter 13, “Routed Mode Interfaces,” or Chapter 14, “Transparent Mode Interfaces.”

Feature History for ASAv Interfaces

Table 12-1 lists the release history for this feature.

Table 12-1 Feature History for Interfaces

Feature Name	Releases	Feature Information
ASAv support	9.2(1)	The ASAv was introduced.