

IS-IS Instance Per VRF for IP

- Feature history for IS-IS instance per VRF for IP, on page 1
- IS-IS Instance per VRF for IP, on page 1
- Prerequisites for IS-IS Instance per VRF for IP, on page 2
- Restrictions for IS-IS Instance per VRF for IP, on page 2
- Configure IS-IS Instance per VRF for IP, on page 3
- Configuration examples, on page 6

Feature history for IS-IS instance per VRF for IP

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature name and description	Supported platform
Cisco IOS XE 26.2.1ea	IS-IS instance per VRF for IP: This feature support has been introduced.	Cisco C9550 Series Smart Switches
Cisco IOS XE 17.18.1	IS-IS instance per VRF for IP: VRF functionality enables ISPs to segregate routing protocol data and distribute it to the correct routing table and network devices.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

IS-IS Instance per VRF for IP

VRF functionality enables ISPs to segregate routing protocol data and distribute it to the correct routing table and network devices. A single router with VRF functionality decreases costs when compared to employing individual routers for routing information.

VRF-Aware IS-IS

You can configure IS-IS to be VRF-aware. A VRF has an IP routing table, a Cisco Express Forwarding (CEF) table that is derived, interfaces that use the forwarding table, and rules and routing protocol parameters that control the information in the routing table.

IS-IS Instance per VRF for IP operation

ISPs can use a single router to create multiple VRF-aware IS-IS instances, eliminating the need for duplicate hardware. IS-IS can be enabled to be VRF-aware, allowing ISPs to separate customer data with multiple instances and to propagate information to appropriate service providers.

For example, an ISP can create three VRFs, VRF First, VRF Second, and VRF Third, one for each customer. VRF-aware IS-IS instances are created and associated with each VRF: tagFIRST, tagSECOND, and tagTHIRD. Each instance will possess its own routing process, IS-IS database, routing table, and will calculate its shortest path first (SPF) tree.

Prerequisites for IS-IS Instance per VRF for IP

- Ensure IS-IS is running on your network.
- The VRF configuration is a prerequisite to associating an IS-IS instance with that specific VRF. However, the VRF configuration is independent of associating it with IS-IS or any other routing protocol. An IS-IS instance becomes VRF-aware after it is associated with a specific VRF.

Restrictions for IS-IS Instance per VRF for IP

- IS-IS VRF support is supported only for IPv4.
- IS-IS instances running Connectionless Network Services (CLNS) must have the same system ID.
- An IS-IS instance that is running CLNS or IPv6 cannot be associated with a VRF.
- You can configure only one IS-IS instance to run both CLNS and IP.
- IS-IS instances within the same VRF must have unique system IDs. IS-IS instances located in separate VRFs can have the same system ID.
- You can associate an IS-IS instance with only one VRF.
- You can configure the **passive-interface default** command only on one IS-IS instance per VRF.
- Redistribution is allowed only within the same VRF.
- You can enable only one IS-IS instance per interface.
- An interface can belong to an IS-IS instance only if they are associated with the same VRF.
- When using LDP, avoid using the **route-target** command during VRF configuration. The router will use BGP for Multiprotocol Label Switching (MPLS) labels.

Configure IS-IS Instance per VRF for IP

This section provides configuration information about IS-IS Instance per VRF for IP.

Create a VRF

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Device(config)# ip vrf vrfFirst`	Configures a VRF routing table, and enters VRF configuration mode.
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 1:1`	Creates routing and forwarding tables for a VRF.
Step 5	**end** **Example:** `Device(config-vrf)# end`	Exits VRF configuration mode and returns to privileged EXEC mode.

Attach an interface to the VRF

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface HundredGigabitethernet 1/0/1`	Configures an interface type and enters interface configuration mode.
Step 4	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding vrfFirst`	Associates a VPN routing and forwarding instance (VRF) with an interface or subinterface.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Create a VRF-Aware IS-IS instance in interface configuration mode

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface HundredGigabitethernet 1/0/1`	Configures an interface type and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [*secondary*] **Example:**	Sets a primary or secondary IP address for an interface.

	Command or Action	Purpose
	`Device(config-if)# ip address 172.16.11.1 255.255.255.255`	
Step 5	**ip router isis** *process-tag* **Example:** `Device(config-if)# ip router isis vrfFirst`	Configures an IS-IS routing process for IP on an interface and attaches a tag to the routing process. **Note** The interface-mode **ip router isis** command will overwrite the existing configuration on the interface only if it is changing the ownership to a different instance within the same VRF. The configuration is rejected if the change attempt involves two instances associated with different VRFs.
Step 6	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Restarts a disabled interface.
Step 7	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Create a VRF-Aware IS-IS instance in router configuration mode

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router isis** *process-tag* **Example:** `Device(config)# router isis tagFirst`	Enables the IS-IS routing protocol, specifies an IS-IS process, and enters router configuration mode.

	Command or Action	Purpose
Step 4	**vrf** *vrf-name* **Example:** `Device(config-router)# vrf vrfFirst`	Associates an IS-IS instance with a VRF. • It is presumed that the VRF named vrfFirst was previously created.
Step 5	**net** *network-entity-title* **Example:** `Device(config-router)# net 49.000b.0000.0001.0002.00`	Configures an IS-IS NET for a CLNS routing process.
Step 6	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode.

Configuration examples

Refer this section for configuration examples of IS-IS Instance per VRF for IP.

Example: Configure multiple VRF-Aware IS-IS instances

In this example, the VRF Second is created and an IS-IS instance is created explicitly by entering the **router isis** command on the router:

```
Device(config)# ip routing
Device(config)# ip vrf Second
Device(config-vrf)# rd 1:1
Device(config-if)# router isis tagSecond
Device(config-router)# vrf Second
Device(config-router)# net 49.000b.0000.0001.0002.00
```

The following sample output verifies information for the VRF-aware IS-IS instances that were created:

```
Device# show clns tagSecond neighbors

Tag tagSecond:
System Id      Interface                     SNPA                State  Holdtime  Type
Protocol
router-03      HundredGigabitEthernet1/0/1   00d0.2b7f.9502      Up     9         L2
IS-IS
router-03      FourtyGigabitEthernet1/0/1    DLCI 211            Up     27        L2
IS-IS
router-02      GigabitEthernet1/0/2          DLCI 131            Up     29        L2
IS-IS
router-11      HundredGigabitEthernet1/0/2   000e.d79d.7920      Up     7         L2
IS-IS
router-11      HundredGigabitEthernet1/0/3   000e.d79d.7921      Up     8         L2
IS-IS
router-11      GigabitEthernet1/0/3          DLCI 451            Up     24        L2
```

```
IS-IS
.
.
.

Device# show clns tagSecond protocol

IS-IS Device: tagSecond
  System Id: 0000.0001.0002.00  IS-Type: level-2-only
  Manual area address(es):
        49.000b
  Routing for area address(es):
        49.000b
  Interfaces supported by IS-IS:
        HundredGigabitEthernet2/0/1 - IP
        HundredGigabitEthernet2/0/2 - IP
        HundredGigabitEthernet2/0/3 - IP
        HundredGigabitEthernet2/0/4 - IP
  Redistributing:
    static
  Distance: 110
  RRR level: none
  Generate narrow metrics: level-1-2
  Accept narrow metrics:   level-1-2
  Generate wide metrics:   none
  Accept wide metrics:     none
```

Example: Create an IS-IS instance without a process tag

In this example, an IS-IS instance was created without the optional process tag. You can display information for an IS-IS instance created without the optional process tag by entering commands with "null" specified for the *process-tag* argument, such as **show clns protocol**.

```
Device(config)# router isis
Device(config-router)# vrf first
Device(config-router)# net 49.000b.0000.0001.ffff.00
Device(config-router)# is-type level-1
Device(config)# interface FourtyGigInterface 1/0/1
Device(config-if)# ip vrf forwarding first
Device(config-if)# ip address 172.16.2.1 255.255.255.0
Device(config-if)# ip router isis
Device(config-if)# no shutdown
```

Because the IS-IS instance is created without the optional process tag, its information is displayed when the **show clns protocol** command is entered with with "null" specified for the *process-tag* argument:

```
Device# show clns null protocol

IS-IS Device: <Null Tag>
  System Id: 0000.0001.FFFF.00  IS-Type: level-1
  Manual area address(es):
        49.000b
  Routing for area address(es):
        49.000b
  Interfaces supported by IS-IS:
        FourtyGigInterface 1/0/1
  Redistributing:
    static
  Distance: 110
  RRR level: none
  Generate narrow metrics: level-1-2
  Accept narrow metrics:   level-1-2
```

```
  Generate wide metrics:   none
  Accept wide metrics:     none
```

Example: Redistribute routes from an IS-IS instance

In this sample configuration, routes have been redistributed from the IS-IS instance "null" and from an OSPF process in VRF Blue into the IS-IS instance named tagBLUE.

To redistribute between two different IS-IS instances, they must be configured in the same VRF context.

```
Device(config)# router isis tagBLUE
Device(config-router)# redistribute isis null ip metric 10 route-map isisMAP1
Device(config-router)# redistribute ospf 1 vrf BLUE metric 1 metric-type external level-1-2
.
.
.
Device(config)# route-map isisMAP1 permit 10
Device(config-route-map)# match route-type level-2 level-1
Device(config-route-map)# set level level-2
```

Example: Change the interface ownership

In this sample configuration, FourtyGigInterface 1/0/1 was originally enabled for IS-IS IP routing for a 'null' instance without a process tag, located in vrfSecond.

Note Use of the **ip router isis** command in interface configuration mode will overwrite the prior configuration on that interface, but only if the new configuration is attempting to change the interface ownership to a different instance that is in the same VRF as the currently configured owner instance. The configuration will be rejected if the attempt is to change between two instances associated with different VRFs.

```
Device(config)# interface FourtyGigInterface 1/0/1
Device(config-if)# ip router isis tagSecond
%ISIS: Interface detached from null and to be attached to instance tagSecond.
```