Configuring Flexible NetFlow

- Prerequisites for Flexible NetFlow, on page 1
- Restrictions for Flexible NetFlow, on page 2
- Information About Flexible Netflow, on page 4
- How to Configure Flexible Netflow, on page 18
- Monitoring Flexible NetFlow, on page 30
- Configuration Examples for Flexible NetFlow, on page 31
- Feature Information for Flexible NetFlow, on page 34

Prerequisites for Flexible NetFlow

- You are familiar with the Flexible NetFlow key fields as they are defined in the following commands:
  - **match flow**
  - **match interface**
  - **match** {**ipv4** | **ipv6**}
  - **match routing**
  - **match transport**
- You are familiar with the Flexible NetFlow nonkey fields as they are defined in the following commands:
  - **collect counter**
  - **collect flow**
  - **collect interface**
  - **collect**{**ipv4** | **ipv6**}
  - **collect routing**
  - **collect timestamp sys-uptime**
  - **collect transport**
- The networking device must be running a Cisco release that supports Flexible NetFlow.

- Flexible NetFlow configurations applied on one of the Layer 3 port-channel member port will also be applied across all other member ports within the same device. For devices configured either as part of a stack or with StackWise Virtual, Cisco recommends to configure the Flexible NetFlow on the port-channel interface.

IPv4 Traffic

- The networking device must be configured for IPv4 routing.
- One of the following must be enabled on your device and on any interfaces on which you want to enable Flexible NetFlow: Cisco Express Forwarding or distributed Cisco Express Forwarding.

IPv6 Traffic

- The networking device must be configured for IPv6 routing.
- One of the following must be enabled on your device and on any interfaces on which you want to enable Flexible NetFlow: Cisco Express Forwarding IPv6 or distributed Cisco Express Forwarding.

Restrictions for Flexible NetFlow

The following are restrictions for Flexible NetFlow:

- Flexible NetFlow is not supported on the Layer 2 port-channel interface, but is supported on the Layer 2 port-channel member ports.
- Traditional NetFlow accounting is not supported.
- Flexible NetFlow Version 9 and Version 10 export formats are supported. However, if you have not configured the export protocol, Version 9 export format is applied by default.
- Flexible NetFlow and NBAR cannot be configured together at the same time on the same interface.
- Layer 2, IPv4, and IPv6 traffic types are supported. Multiple flow monitors of different traffic types can be applied for a given interface and direction. Multiple flow monitors of same traffic type cannot be applied for a given interface and direction.
- Layer 2, VLAN, Layer 3 and SVI interfaces are supported, but the device does not support tunnels.
- The following NetFlow table sizes are supported:

Trim Level	Ingress NetFlow Table	Egress NetFlow Table
Network Essentials	32 K	32 K
Network Advantage	32 K	32 K

- Depending on the switch type, a switch will have one or two forwarding ASICs. The capacities listed in the above table are on a per-Core/per-ASIC basis.
- The switch can support either one or two cores. Each Overflow TCAM can support 256 ingress and 256 egress entries per core.

- The NetFlow tables are on separate compartments and cannot be combined. Depending on which core processed the packet, the flows will be created in the table in the corresponding core.
- NetFlow hardware implementation supports four hardware samplers. You can select a sampler rate from 1 out of 2 to 1 out of 1024. Both — random and deterministic — sampling modes are supported.
- NetFlow hardware uses hash tables internally. Hash collisions can occur in the hardware. Therefore, in spite of the internal overflow Content Addressable Memory (CAM), the actual NetFlow table utilization could be about 80 percent.
- Depending on the fields that are used for the flow, a single flow could take two consecutive entries. IPv6 and datalink flows also take two entries. In these situations, the effective usage of NetFlow entries is half the table size, which is separate from the above hash collision limitation.
- The device supports up to 15 flow monitors.
- The NetFlow software implementation supports distributed NetFlow export, so the flows are exported from the same device in which the flow was created.
- Ingress flows are present in the ASIC that first received the packets for the flow. Egress flows are present in the ASIC from which the packets actually left the device set up.
- The reported value for the bytes count field (called “bytes long”) is Layer-2-packet-size—18 bytes. For classic Ethernet traffic (802.3), this will be accurate. For all other Ethernet types, this field will not be accurate. Use the "bytes layer2” field, which always reports the accurate Layer 2 packet size. For information about supported Flexible NetFlow fields, see 'Supported Flexible NetFlow Fields' topic.
- Configuration of IPFIX exporter on an AVC flow monitor is not supported.
- Flexible NetFlow export is not supported on the Ethernet management port, GigabitEthernet 0/0.
- When a flow record has only Source Group Tag (SGT) and Destination Group Tag (DGT) fields (or only either of the two) and if both the values are not applicable, then a flow will still be created with zero values for SGT and DGT. The flow records are expected to include source and destination IP addresses, along with SGT and DGT fields.
- On non-Cisco TrustSec interfaces, an SGT value of zero implies that there is no command header. On Cisco TrustSec interfaces, an SGT value of zero implies an unknown tag.
- When a quality of service (QoS) marked packet is received on an interface which has NetFlow configured in the ingress direction, the QoS value of the packet is captured by the NetFlow collector. However, when the packet is received on an interface which has NetFlow configured in the egress direction and the QoS value has been rewritten on ingress by the switch, the new QoS value of the packet is not captured by the collector.
- For an IPv6 flow monitor, Source Group Tag (SGT) and Destination Group Tag (DGT) fields cannot co-exist with MAC address fields.
- NetFlow records do not support MultiProtocol Label Switching-enabled (MPLS-enabled) interfaces.
- Data capture based on MPLS label inside the MPLS network is not supported. Capture of IP header fields of an MPLS tagged packet is not supported.
- Egress flow monitors do not capture flows that are egressing out in EoMPLS mode or in L3VPN Per-Prefix mode.
- Flow exporter exports the flow data only after the template data time out period ends. Configuration changes such as VPN ID modification or VRF deletion will take effect after the time out period ends.

- A flow monitor cannot be shared across Layer 3 physical interfaces and logical interfaces (such as, Layer 3 port-channel interface, Layer 3 port-channel member, and switch virtual interface [SVI]), but a flow monitor can be shared across logical interfaces or Layer 3 physical interfaces.

Information About Flexible Netflow

The following sections provide information about Flexible Netflow.

Flexible Netflow Overview

Flexible NetFlow uses flows to provide statistics for accounting, network monitoring, and network planning.

A flow is a unidirectional stream of packets that arrives on a source interface and has the same values for the keys. A key is an identified value for a field within the packet. You create a flow using a flow record to define the unique keys for your flow.

The device supports the Flexible NetFlow feature that enables enhanced network anomalies and security detection. Flexible NetFlow allows you to define an optimal flow record for a particular application by selecting the keys from a large collection of predefined fields.

All key values must match for the packet to count in a given flow. A flow might gather other fields of interest, depending on the export record version that you configure. Flows are stored in the Flexible NetFlow cache.

You can export the data that Flexible NetFlow gathers for your flow by using an exporter and export this data to a remote system such as a Flexible NetFlow collector. The Flexible NetFlow collector can use an IPv4 address.

You define the size of the data that you want to collect for a flow using a monitor. The monitor combines the flow record and exporter with the Flexible NetFlow cache information.

Starting with the Cisco IOS XE 16.12.1 release, Source Group Tag (SGT) and Destination Group Tag (DGT) fields over Flexible NetFlow are supported for IPv6 traffic.

Original NetFlow and Benefits of Flexible NetFlow

Flexible NetFlow allows the flow to be user defined. The benefits of Flexible NetFlow include:

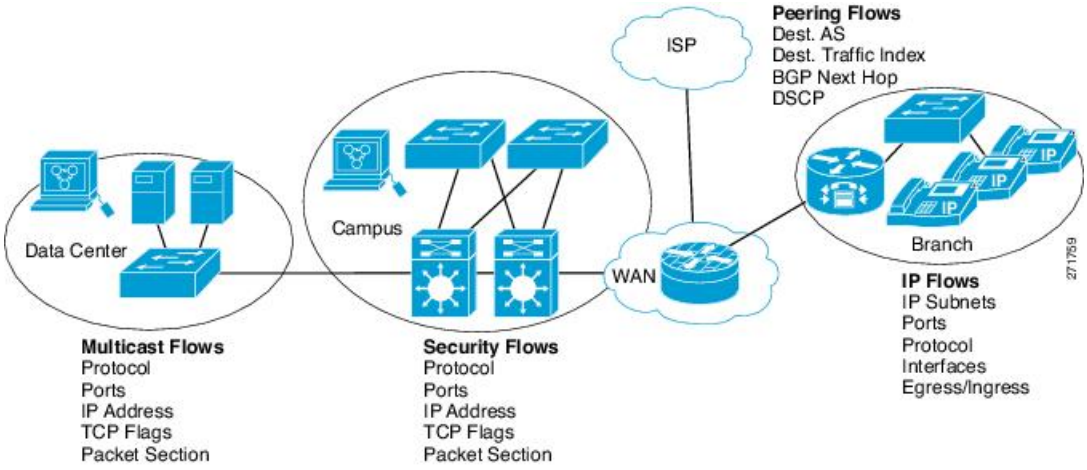
- High-capacity flow recognition, including scalability and aggregation of flow information.
- Enhanced flow infrastructure for security monitoring and dDoS detection and identification.
- New information from packets to adapt flow information to a particular service or operation in the network. The flow information available will be customizable by Flexible NetFlow users.
- Extensive use of Cisco’s flexible and extensible NetFlow Version 9.
- A comprehensive IP accounting feature that can be used to replace many accounting features, such as IP accounting, Border Gateway Protocol (BGP) Policy Accounting, and persistent caches.

Flexible NetFlow allows you to understand network behavior with more efficiency, with specific flow information tailored for various services used in the network. The following are some example applications for a Flexible NetFlow feature:

- Flexible NetFlow enhances Cisco NetFlow as a security monitoring tool. For instance, new flow keys can be defined for packet length or MAC address, allowing users to search for a specific type of attack in the network.
- Flexible NetFlow allows you to quickly identify how much application traffic is being sent between hosts by specifically tracking TCP or UDP applications by the class of service (CoS) in the packets.
- The accounting of traffic entering a Multiprotocol Label Switching (MPLS) or IP core network and its destination for each next hop per class of service. This capability allows the building of an edge-to-edge traffic matrix.

The figure below is an example of how Flexible NetFlow might be deployed in a network.

Figure 1: Typical Deployment for Flexible NetFlow



Flexible NetFlow Components

Flexible NetFlow consists of components that can be used together in several variations to perform traffic analysis and data export. The user-defined flow records and the component structure of Flexible NetFlow facilitates the creation of various configurations for traffic analysis and data export on a networking device with a minimum number of configuration commands. Each flow monitor can have a unique combination of flow record, flow exporter, and cache type. If you change a parameter such as the destination IP address for a flow exporter, it is automatically changed for all the flow monitors that use the flow exporter. The same flow monitor can be used in conjunction with different flow samplers to sample the same type of network traffic at different rates on different interfaces. The following sections provide more information on Flexible NetFlow components:

Flow Records

In Flexible NetFlow a combination of key and nonkey fields is called a record. Flexible NetFlow records are assigned to Flexible NetFlow flow monitors to define the cache that is used for storing flow data. Flexible NetFlow includes several predefined records that can help you get started using Flexible NetFlow.

A flow record defines the keys that Flexible NetFlow uses to identify packets in the flow, as well as other fields of interest that Flexible NetFlow gathers for the flow. You can define a flow record with any combination of keys and fields of interest. The device supports a rich set of keys. A flow record also defines the types of

counters gathered per flow. You can configure 64-bit packet or byte counters. The device enables the following match fields as the defaults when you create a flow record:

- **match datalink**—Layer 2 attributes
- **match flow direction**—Specifies a match to the fields identifying the direction of flow.
- **match interface**—Interface attributes
- **match ipv4**—IPv4 attributes
- **match ipv6**—IPv6 attributes
- **match transport**—Transport layer fields
- **match flow cts**—Cisco TrustSec fields

NetFlow Predefined Records

Flexible NetFlow includes several predefined records that you can use to start monitoring traffic in your network. The predefined records are available to help you quickly deploy Flexible NetFlow and are easier to use than user-defined flow records. You can choose from a list of already defined records that may meet the needs for network monitoring. As Flexible NetFlow evolves, popular user-defined flow records will be made available as predefined records to make them easier to implement.

Note Predefined records are not supported for regular Flexible NetFlow on Cisco Catalyst 9000 Series Switches.

User-Defined Records

Flexible NetFlow enables you to define your own records for a Flexible NetFlow flow monitor cache by specifying the key and nonkey fields to customize the data collection to your specific requirements. When you define your own records for a Flexible NetFlow flow monitor cache, they are referred to as *user-defined records*. The values in nonkey fields are added to flows to provide additional information about the traffic in the flows. A change in the value of a nonkey field does not create a new flow. In most cases the values for nonkey fields are taken from only the first packet in the flow. Flexible NetFlow enables you to capture counter values such as the number of bytes and packets in a flow as nonkey fields.

You can create user-defined records for applications such as QoS and bandwidth monitoring, application and end user traffic profiling, and security monitoring for dDoS attacks. Flexible NetFlow also includes several predefined records that emulate original NetFlow. Flexible NetFlow user-defined records provide the capability to monitor a contiguous section of a packet of a user-configurable size, and use it in a flow record as a key or a nonkey field along with other fields and attributes of the packet. The section may include any Layer 3 data from the packet. The packet section fields allow the user to monitor any packet fields that are not covered by the Flexible NetFlow predefined keys. The ability to analyze packet fields, enables more detailed traffic monitoring, facilitates the investigation of dDoS attacks, and enables implementation of other security applications such as URL monitoring.

Flexible NetFlow provides predefined types of packet sections of a user-configurable size. The following Flexible NetFlow commands (used in Flexible NetFlow flow record configuration mode) can be used to configure the predefined types of packet sections:

- **collect ipv4 section header size** *bytes* --Starts capturing the number of bytes specified by the *bytes*argument from the beginning of the IPv4 header of each packet.

- **collect ipv4 section payload size** *bytes* --Starts capturing bytes immediately after the IPv4 header from each packet. The number of bytes captured is specified by the *bytes* argument.
- **collect ipv6 section header size** *bytes* --Starts capturing the number of bytes specified by the *bytes*argument from the beginning of the IPv6 header of each packet.

The *bytes* values are the sizes in bytes of these fields in the flow record. If the corresponding fragment of the packet is smaller than the requested section size, Flexible NetFlow will fill the rest of the section field in the flow record with zeros. If the packet type does not match the requested section type, Flexible NetFlow will fill the entire section field in the flow record with zeros.

Flexible NetFlow adds a new Version 9 export format field type for the header and packet section types. Flexible NetFlow will communicate to the NetFlow collector the configured section sizes in the corresponding Version 9 export template fields. The payload sections will have a corresponding length field that can be used to collect the actual size of the collected section.

Flexible NetFlow Match Parameters

The following table describes Flexible NetFlow match parameters. You must configure at least one of the following match parameters for the flow records.

Table 1: Match Parameters

Command	Purpose
match datalink {**dot1q** | **ethertype** | **mac** | **vlan** }	Specifies a match to datalink or Layer 2 fields. The following command options are available: • **dot1q**—Matches to the dot1q field. • **ethertype**—Matches to the ethertype of the packet. • **mac**—Matches the source or destination MAC fields. • **vlan**—Matches to the VLAN that the packet is located on (input or output).
match flow direction	Specifies a match to the flow identifying fields.
match interface {**input** | **output**}	Specifies a match to the interface fields. The following command options are available: • **input**—Matches to the input interface. • **output**—Matches to the output interface.

Command	Purpose
match ipv4 {**destination** | **protocol** | **source** | **tos** | **ttl** | **version**}	Specifies a match to the IPv4 fields. The following command options are available: • **destination**—Matches to the IPv4 destination address-based fields. • **protocol**—Matches to the IPv4 protocols. • **source**—Matches to the IPv4 source address based fields. • **tos**—Matches to the IPv4 Type of Service fields. • **ttl**—Matches to the IPv4 Time To Live fields. • **version**—Matches to the IP version from the IPv4 header.
match ipv6 {**destination** | **hop-limit** | **protocol** | **source** | **traffic-class** | **version** }	Specifies a match to the IPv6 fields. The following command options are available: • **destination**—Matches to the IPv6 destination address-based fields. • **hop-limit**—Matches to the IPv6 hop limit fields. • **protocol**—Matches to the IPv6 payload protocol fields. • **source**—Matches to the IPv6 source address based fields. • **traffic-class**—Matches to the IPv6 traffic class. • **version**—Matches to the IP version from the IPv6 header.
match transport {**destination-port** | **igmp** | **icmp** | **source-port**}	Specifies a match to the Transport Layer fields. The following command options are available: • **destination-port**—Matches to the transport destination port. • **icmp**—Matches to ICMP fields, including ICMP IPv4 and IPv6 fields. • **igmp**—Matches to IGMP fields. • **source-port**—Matches to the transport source port.

Command	Purpose
match flow cts {source | destination} group-tag	Specifies a match to the CTS fields support in FNF record. The following command options are available: • **source** —Matches to the source of CTS entering the domain. • **destination** —Matches to the destination of the CTS leaving the domain.

Flexible NetFlow Collect Parameters

The following table describes the Flexible NetFlow collect parameters.

Table 2: Collect Parameters

Command	Purpose
collect counter { **bytes** { **layer2** { **long** } | **long** } | **packets** { **long** } }	Collects the counter fields total bytes and total packets.
collect interface {**input** | **output**}	Collects the fields from the input or output interface.
collect timestamp absolute {**first** | **last**}	Collects the fields for the absolute time the first packet was seen or the absolute time the most recent packet was last seen (in milliseconds).
collect transport tcp flags	Collects the following transport TCP flags: • **ack**—TCP acknowledgement flag • **cwr**—TCP congestion window reduced flag • **ece**—TCP ECN echo flag • **fin**—TCP finish flag • **psh**—TCP push flag • **rst**—TCP reset flag • **syn**—TCP synchronize flag • **urg**—TCP urgent flag **Note** On the device, you cannot specify which TCP flag to collect. You can only specify to collect transport TCP flags. All TCP flags will be collected with this command.
collect counter bytes	Configures the number of bytes seen in a flow as a nonkey field and enables collecting the total number of bytes from the flow.

Command	Purpose
collect counter packets	Configures the number of packets seen in a flow as a nonkey field and enables collecting the total number of packets from the flow.
collect flow sampler	Configures a flow sampler ID as a non-key field for the record. This field contains the ID of the flow sampler used to monitor the flow.
collect ipv4 destination	Configures the IPv4 destination as a non-key field for a flow record.
collect ipv4 source	Configures the IPv4 source as a non-key field for a flow record
collect ipv6 destination	Configures the IPv6 destination as a non-key field for a flow record.
collect ipv6 source	Configures the IPv6 source as a non-key field for a flow record
collect routing next-hop address	Configures the next-hop address value as a non-key field and enable collecting information regarding the next hop from the flows

Flow Exporters

Flow exporters export the data in the flow monitor cache to a remote system, such as a server running NetFlow collector, for analysis and storage. Flow exporters are created as separate entities in the configuration. Flow exporters are assigned to flow monitors to provide data export capability for the flow monitors. You can create several flow exporters and assign them to one or more flow monitors to provide several export destinations. You can create one flow exporter and apply it to several flow monitors.

NetFlow Data Export Format Version 9

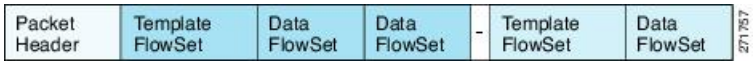
The basic output of NetFlow is a flow record. Several different formats for flow records have evolved as NetFlow has matured. The most recent evolution of the NetFlow export format is known as Version 9. The distinguishing feature of the NetFlow Version 9 export format is that it is template-based. Templates provide an extensible design to the record format, a feature that should allow future enhancements to NetFlow services without requiring concurrent changes to the basic flow-record format. Using templates provides several key benefits:

- Third-party business partners who produce applications that provide collector or display services for NetFlow do not have to recompile their applications each time a new NetFlow feature is added. Instead, they should be able to use an external data file that documents the known template formats.
- New features can be added to NetFlow quickly without breaking current implementations.
- NetFlow is “future-proofed” against new or developing protocols because the Version 9 format can be adapted to provide support for them.

The Version 9 export format consists of a packet header followed by one or more template flow or data flow sets. A template flow set provides a description of the fields that will be present in future data flow sets. These

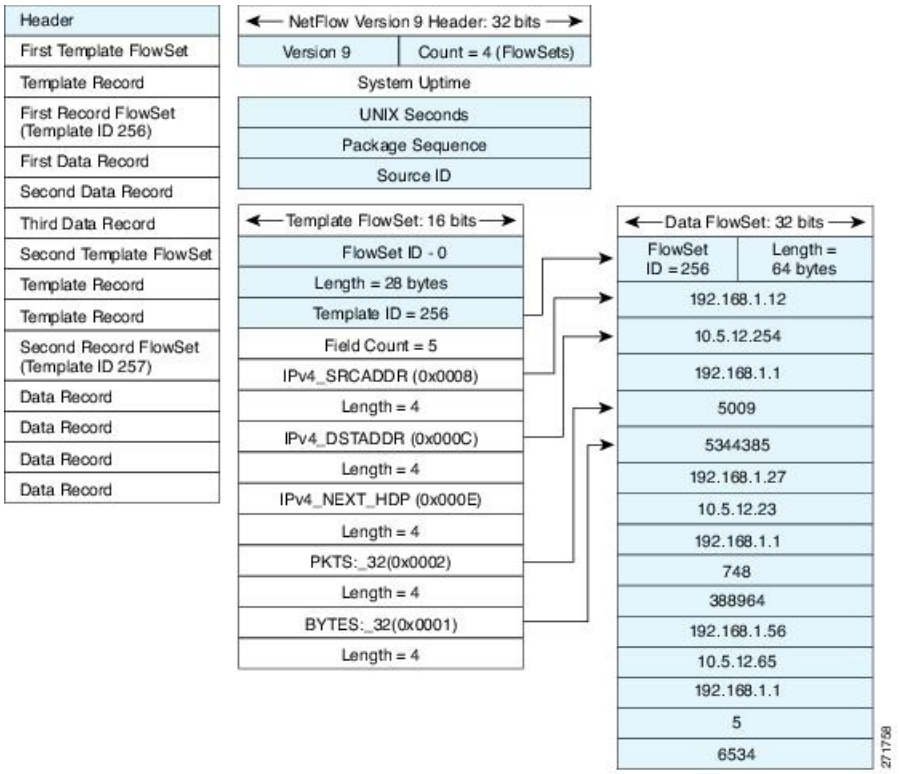
data flow sets may occur later within the same export packet or in subsequent export packets. Template flow and data flow sets can be intermingled within a single export packet, as illustrated in the figure below.

Figure 2: Version 9 Export Packet



NetFlow Version 9 will periodically export the template data so the NetFlow collector will understand what data is to be sent and also export the data flow set for the template. The key advantage to Flexible NetFlow is that the user configures a flow record, which is effectively converted to a Version 9 template and then forwarded to the collector. The figure below is a detailed example of the NetFlow Version 9 export format, including the header, template flow, and data flow sets.

Figure 3: Detailed Example of the NetFlow Version 9 Export Format



For more information on the Version 9 export format, refer to the white paper titled Cisco IOS NetFlow Version 9 Flow-Record Format, available at this URL: http://www.cisco.com/en/US/tech/tk648/tk362/technologies_white_paper09186a00800a3db9.shtml.

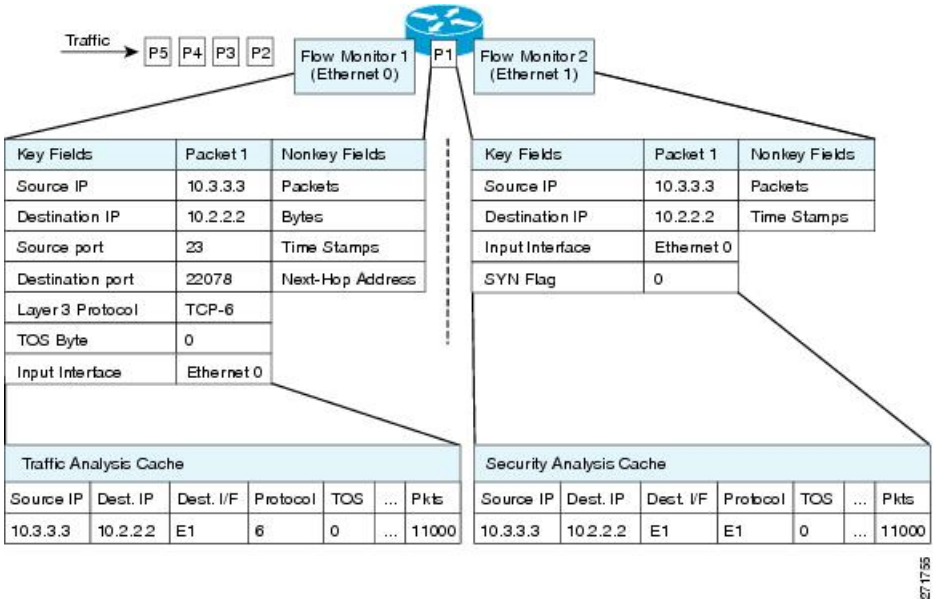
Flow Monitors

Flow monitors are the Flexible NetFlow component that is applied to interfaces to perform network traffic monitoring.

Flow data is collected from the network traffic and added to the flow monitor cache during the monitoring process based on the key and nonkey fields in the flow record.

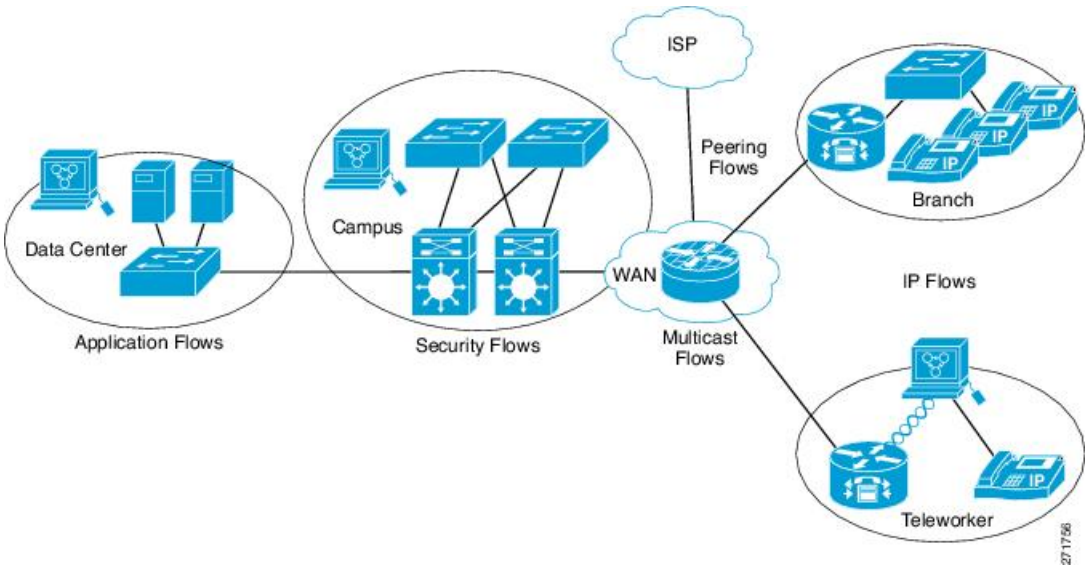
Flexible NetFlow can be used to perform different types of analysis on the same traffic. In the figure below, packet 1 is analyzed using a record designed for standard traffic analysis on the input interface and a record designed for security analysis on the output interface.

Figure 4: Example of Using Two Flow Monitors to Analyze the Same Traffic



The figure below shows a more complex example of how you can apply different types of flow monitors with custom records.

Figure 5: Complex Example of Using Multiple Types of Flow Monitors with Custom Records



Normal

The default cache type is “normal”. In this mode, the entries in the cache are aged out according to the timeout active and timeout inactive settings. When a cache entry is aged out, it is removed from the cache and exported via any exporters configured.

Flow Samplers

Flow samplers are created as separate components in a router’s configuration. Flow samplers are used to reduce the load on the device that is running Flexible NetFlow by limiting the number of packets that are selected for analysis.

Flow sampling exchanges monitoring accuracy for router performance. When you apply a sampler to a flow monitor, the overhead load on the router of running the flow monitor is reduced because the number of packets that the flow monitor must analyze is reduced. The reduction in the number of packets that are analyzed by the flow monitor causes a corresponding reduction in the accuracy of the information stored in the flow monitor’s cache.

Samplers are combined with flow monitors when they are applied to an interface with the **ip flow monitor** command.

Supported Flexible NetFlow Fields

The following tables provide a consolidated list of supported fields in Flexible NetFlow (FNF) for various traffic types and traffic direction.

Note If the packet has a VLAN field, then that length is not accounted for.

Field	Layer 2 In	Layer 2 Out	IPv4 In	IP v4 Out	IPv6 In	IPv6 Out	Notes
Key or Collect Fields							
Interface input	Yes	—	Yes	—	Yes	—	If you apply a flow monitor in the input direction: • Use the **match** keyword and use the input interface as a key field. • Use the **collect** keyword and use the output interface as a collect field. This field will be present in the exported records but with a value of 0.

Field	Layer 2 In	Layer 2 Out	IPv4 In	IP v4 Out	IPv6 In	IPv6 Out	Notes
Interface output	—	Yes	—	Yes	—	Yes	If you apply a flow monitor in the output direction: • Use the **match** keyword and use the output interface as a key field. • Use the **collect** keyword and use the input interface as a collect field. This field will be present in the exported records but with a value of 0.

Field	Layer 2 In	Layer 2 Out	IPv4 In	IP v4 Out	IPv6 In	IPv6 Out	Notes
Key Fields							
Flow direction	Yes	Yes	Yes	Yes	Yes	Yes	
Ethertype	Yes	Yes	—	—	—	—	
VLAN input	Yes	—	Yes	—	Yes	—	Supported only for a switch port.
VLAN output	—	Yes	—	Yes	—	Yes	Supported only for a switch port.
dot1q VLAN input	Yes	—	Yes	—	Yes	—	Supported only for a switch port.
dot1q VLAN output	—	Yes	—	Yes	—	Yes	Supported only for a switch port.
dot1q priority	Yes	Yes	Yes	Yes	Yes	Yes	Supported only for a switch port.
MAC source address input	Yes	Yes	Yes	Yes	Yes	Yes	

Field	Layer 2 In	Layer 2 Out	IPv4 In	IP v4 Out	IPv6 In	IPv6 Out	Notes
MAC source address output	—	—	—	—	—	—	
MAC destination address input	Yes	—	Yes	—	Yes	—	
MAC destination address output	—	Yes	—	Yes	—	Yes	
IPv4 version	—	—	Yes	Yes	Yes	Yes	
IPv4 TOS	—	—	Yes	Yes	Yes	Yes	
IPv4 protocol	—	—	Yes	Yes	Yes	Yes	Must use if any of src/dest port, ICMP code/type, IGMP type or TCP flags are used.
IPv4 TTL	—	—	Yes	Yes	Yes	Yes	
IPv4 TTL	—	—	Yes	Yes	Yes	Yes	Same as IPv4 TTL.
IPv4 protocol	—	—	Yes	Yes	Yes	Yes	Same as IPv4 protocol. Must use if any of src/dest port, ICMP code/type, IGMP type or TCP flags are used.
IPv4 source address	—	—	Yes	Yes	—	—	

Field	Layer 2 In	Layer 2 Out	IPv4 In	IP v4 Out	IPv6 In	IPv6 Out	Notes
IPv4 destination address	—	—	Yes	Yes	—	—	
ICMP IPv4 type	—	—	Yes	Yes	—	—	
ICMP IPv4 code	—	—	Yes	Yes	—	—	
IGMP type	—	—	Yes	Yes	—	—	

Field	Layer 2 In	Layer 2 Out	IPv4 In	IP v4 Out	IPv6 In	IPv6 Out	Notes
Key Fields continued							
IPv6 version	—	—	Yes	Yes	Yes	Yes	Same as IP version.
IPv6 protocol	—	—	Yes	Yes	Yes	Yes	Same as IP protocol. Must use if any of src/dest port, ICMP code/type, IGMP type or TCP flags are used.
IPv6 source address	—	—	—	—	Yes	Yes	
IPv6 destination address	—	—	—	—	Yes	Yes	
IPv6 traffic-class	—	—	Yes	Yes	Yes	Yes	Same as IP TOS.
IPv6 hop-limit	—	—	Yes	Yes	Yes	Yes	Same as IP TTL.
ICMP IPv6 type	—	—	—	—	Yes	Yes	
ICMP IPv6 code	—	—	—	—	Yes	Yes	
source-port	—	—	Yes	Yes	Yes	Yes	

Field	Layer 2 In	Layer 2 Out	IPv4 In	IP v4 Out	IPv6 In	IPv6 Out	Notes
dest-port	—	—	Yes	Yes	Yes	Yes	

Field	Layer 2 In	Layer 2 Out	IPv4 In	IP v4 Out	IPv6 In	IPv6 Out	Notes
Collect Fields							
Bytes long	Yes	Yes	Yes	Yes	Yes	Yes	Packet size = (Ethernet frame size including FCS - 18 bytes) **Recommended:** Avoid this field and use Bytes layer2 long.
Packets long	Yes	Yes	Yes	Yes	Yes	Yes	
Timestamp absolute first	Yes	Yes	Yes	Yes	Yes	Yes	
Timestamp absolute last	Yes	Yes	Yes	Yes	Yes	Yes	
TCP flags	Yes	Yes	Yes	Yes	Yes	Yes	Collects all flags.
Bytes layer2 long	Yes	Yes	Yes	Yes	Yes	Yes	

Default Settings

The following table lists the Flexible NetFlow default settings for the device.

Table 3: Default Flexible NetFlow Settings

Setting	Default
Flow active timeout	1800 seconds
Flow timeout inactive	15 seconds

Flexible NetFlow—Ingress VRF Support Overview

The Flexible NetFlow—Ingress VRF Support feature enables collecting the virtual routing and forwarding (VRF) ID from incoming packets on a device by applying an input flow monitor having a flow record that collects the VRF ID as a key field.

Ingress Egress Flexible NetFlow on MPLS Overview

- Ingress Flexible Netflow on MPLS (IP level): This feature allows the capture of Internet Protocol (IP) flow information for packets undergoing MPLS label imposition that are entering the MPLS network. These packets arrive on a router as IP packets and are transmitted as MPLS packets. This feature can be enabled by configuring an ingress flow monitor for IPv4 and IPv6 traffic at the CE facing side of the PE node.
- Egress Flexible NetFlow on MPLS (IP level): This feature allows the capture of Internet Protocol (IP) flow information for packets undergoing MPLS label imposition that are exiting the MPLS network. These packets arrive on a router as MPLS packets and are transmitted as IP packets. The feature can be enabled by configuring an egress flow monitor for IPv4 and IPv6 traffic at the CE facing side of PE node.

Configuring VPN ID in Flexible NetFlow

Multiple VPNs from the same private network can use same private source and destination IPs in the data traffic. This can make it difficult to identify the IP address to which the data belongs. A VPN-ID can be used to solve this issue. A VPN-ID is a global unique Virtual Private Network Identifier. It is used to identity a VPN across autonomous Systems (AS). If VPN-ID is exported in NetFlow exported packets – the collector in another AS will be able to associate and segregate the flows based on the VPN to which the data belongs. VPN-ID is a system level property similar to VRF-ID and it can be exported in a similar fashion.

Components of the VPN ID

Each VPN ID consists of the following elements:

- An Organizational Unique Identifier (OUI), a three-octet hex number. The IEEE Registration Authority assigns OUIs to any company that manufactures components under the ISO/IEC 8802 standard. The OUI is used to generate universal LAN MAC addresses and protocol identifiers for use in local and metropolitan area network applications. For example, an OUI for Cisco Systems is 00-03-6B (hex).
- A VPN index, a four-octet hex number, which identifies the VPN within the company.

You can configure the VPN ID by using the **vpn id** command in VRF definition configuration mode. Specify the VPN ID in the following format:

vpn id oui:vpn-index

Once the VPN ID has been configured you can use the **option vrf-attributes** command in flow-exporter configuration mode to configure the VPN ID.

How to Configure Flexible Netflow

To configure Flexible Netflow, follow these general steps:

1. Create a flow record by specifying keys and non-key fields to the flow.

2. Create an optional flow exporter by specifying the protocol and transport destination port, destination, and other parameters.
3. Create a flow monitor based on the flow record and flow exporter.
4. Create an optional sampler.
5. Apply the flow monitor to a Layer 2 port, Layer 3 port, or VLAN.

Creating a Flow Record

Perform this task to configure a customized flow record.

Customized flow records are used to analyze traffic data for a specific purpose. A customized flow record must have at least one **match** criterion for use as the key field and typically has at least one **collect** criterion for use as a nonkey field.

There are hundreds of possible permutations of customized flow records. This task shows the steps that are used to create one of the possible permutations. Modify the steps in this task as appropriate to create a customized flow record for your requirements.

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	**flow record** *record-name* **Example:** `Device(config)# flow record FLOW-RECORD-1`	Creates a flow record and enters Flexible NetFlow flow record configuration mode. • This command also allows you to modify an existing flow record.
Step 4	**description** *description* **Example:** `Device(config-flow-record)# description Used for basic traffic analysis`	(Optional) Creates a description for the flow record.
Step 5	**match** {**ip** | **ipv6**} {**destination** | **source**} **address** **Example:**	**Note** This example configures the IPv4 destination address as a key field for the record.

	Command or Action	Purpose
	`Device(config-flow-record)# match ipv4 destination address`	
Step 6	Repeat Step 5 as required to configure additional key fields for the record.	—

Step 7	**match flow cts** {**source** | **destination**} **group-tag**
	Example:
	`Device(config-flow-record)# match flow cts source group-tag`
	`Device(config-flow-record)# match flow cts destination group-tag`
		Note This example configures the CTS source group tag and destination group tag as a key field for the record. For information about the other key fields available for the **match ipv4**/**ipv6** command, and the other **match** commands that are available to configure key fields.
		Note
		• Egress:
		• If either propagate SGT or CTS is disabled on the egress interface, then SGT will be zero.
		• In an outgoing packet, if SGACL configuration that corresponds to the (SGT, DGT) exists, DGT will be non-zero.
		• If SGACL is disabled on the egress port/VLAN or if global SGACL enforcement is disabled, then DGT will be zero
		• Ingress:
		• In an incoming packet, if a header is present, SGT will reflect the same value as the header. If no value is present, it will show zero.
		• The DGT value will not depend on the ingress port SGACL configuration.

Step 8	**end** **Example:** `Device(config-flow-record)# end`	Exits Flexible NetFlow flow record configuration mode and returns to privileged EXEC mode.
Step 9	**show flow record** *record-name* **Example:**	(Optional) Displays the current status of the specified flow record.

	Command or Action	Purpose
	`Device# show flow record FLOW_RECORD-1`	
Step 10	**show running-config flow record** *record-name* **Example:** `Device# show running-config flow record FLOW_RECORD-1`	(Optional) Displays the configuration of the specified flow record.

Creating a Flow Exporter

You can create a flow export to define the export parameters for a flow.

Note Each flow exporter supports only one destination. If you want to export the data to multiple destinations, you must configure multiple flow exporters and assign them to the flow monitor.

You can export to a destination using IPv4 address.

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(config)# configure terminal`	Enters global configuration mode.
Step 3	**flow exporter** *name* **Example:** `Device(config)# flow exporter ExportTest`	Creates a flow exporter and enters flow exporter configuration mode.
Step 4	**description** *string* **Example:** `Device(config-flow-exporter)# description ExportV9`	(Optional) Describes this flow record as a maximum 63-character string.

	Command or Action	Purpose
Step 5	**destination** {*ipv4-address*} **Example:** `Device(config-flow-exporter)# destination 192.0.2.1 (IPv4 destination)`	Sets the IPv4 destination address or hostname for this exporter.
Step 6	**dscp** *value* **Example:** `Device(config-flow-exporter)# dscp 0`	(Optional) Specifies the differentiated services codepoint value. The range is from 0 to 63. The default is 0.
Step 7	**source** *interface type interface number* **Example:** `Device(config-flow-exporter)# source gigabitEthernet1/0/1`	(Optional) Specifies the interface to use to reach the NetFlow collector at the configured destination. **Note** Flow Exporter does not support unnumbered IP interface as source interface. The following interfaces can be configured as source: • **Auto Template**—Auto-Template interface • **Capwap**—CAPWAP tunnel interface • **GigabitEthernet**—Gigabit Ethernet IEEE 802 • **GroupVI**—Group virtual interface • **Internal Interface**—Internal interface • **Loopback**—Loopback interface • **Null**—Null interface • **Port-channel**—Ethernet Channel of interface • **TenGigabitEthernet**—10-Gigabit Ethernet • **Tunnel**—Tunnel interface • **Vlan**—Catalyst VLANs
Step 8	**transport udp** *number* **Example:**	(Optional) Specifies the UDP port to use to reach the NetFlow collector.

	Command or Action	Purpose
	`Device(config-flow-exporter)# transport udp 200`	
Step 9	**ttl** *seconds* **Example:** `Device(config-flow-exporter)# ttl 210`	(Optional) Configures the time-to-live (TTL) value for datagrams sent by the exporter. The range is from 1 to 255 seconds. The default is 255.
Step 10	**export-protocol** {**netflow-v9**} **Example:** `Device(config-flow-exporter)# export-protocol netflow-v9`	Specifies the version of the NetFlow export protocol used by the exporter.
Step 11	**end** **Example:** `Device(config-flow-record)# end`	Returns to privileged EXEC mode.
Step 12	**show flow exporter** [**name** *record-name*] **Example:** `Device# show flow exporter ExportTest`	(Optional) Displays information about NetFlow flow exporters.
Step 13	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

What to do next

Define a flow monitor based on the flow record and flow exporter.

Creating a Customized Flow Monitor

Perform this required task to create a customized flow monitor.

Each flow monitor has a separate cache assigned to it. Each flow monitor requires a record to define the contents and layout of its cache entries. These record formats can be one of the predefined formats or a user-defined format. An advanced user can create a customized format using the **flow record** command.

Before you begin

If you want to use a customized record instead of using one of the Flexible NetFlow predefined records, you must create the customized record before you can perform this task. If you want to add a flow exporter to the flow monitor for data export, you must create the exporter before you can complete this task.

Note You must use the **no ip flow monitor** command to remove a flow monitor from all of the interfaces to which you have applied it before you can modify the parameters for the **record** command on the flow monitor.

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	**flow monitor** *monitor-name* **Example:** `Device(config)# flow monitor FLOW-MONITOR-1`	Creates a flow monitor and enters Flexible NetFlow flow monitor configuration mode. • This command also allows you to modify an existing flow monitor.
Step 4	**description** *description* **Example:** `Device(config-flow-monitor)# description Used for basic ipv4 traffic analysis`	(Optional) Creates a description for the flow monitor.
Step 5	**record** {*record-name* | **netflow-original** | **netflow** {**ipv4** | **ipv6**} *record* [**peer**]} **Example:** `Device(config-flow-monitor)# record FLOW-RECORD-1`	Specifies the record for the flow monitor.
Step 6	**cache** {**timeout** {**active** | **inactive** | **update** | **rate-limit**} *seconds* | **type normal** } **Example:** `Device(config-flow-monitor)# cache type normal`	(Optional) Modifies the flow monitor cache parameters such as timeout values, and the cache type. Associates a flow cache with the specified flow monitor.

	Command or Action	Purpose
	`Device(config-flow-monitor)# cache timeout active`	
Step 7	Repeat Step 6 as required to finish modifying the cache parameters for this flow monitor.	—
Step 8	**statistics packet protocol** **Example:** `Device(config-flow-monitor)# statistics packet protocol`	(Optional) Enables the collection of protocol distribution statistics for Flexible NetFlow monitors.
Step 9	**statistics packet size** **Example:** `Device(config-flow-monitor)# statistics packet size`	(Optional) Enables the collection of size distribution statistics for Flexible NetFlow monitors.
Step 10	**exporter** *exporter-name* **Example:** `Device(config-flow-monitor)# exporter EXPORTER-1`	(Optional) Specifies the name of an exporter that was created previously.
Step 11	**end** **Example:** `Device(config-flow-monitor)# end`	Exits Flexible NetFlow flow monitor configuration mode and returns to privileged EXEC mode.
Step 12	**show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]] [**statistics**]] **Example:** `Device# show flow monitor FLOW-MONITOR-2 cache`	(Optional) Displays the status and statistics for a Flexible NetFlow flow monitor.
Step 13	**show running-config flow monitor** *monitor-name* **Example:** `Device# show running-config flow monitor FLOW_MONITOR-1`	(Optional) Displays the configuration of the specified flow monitor.
Step 14	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Creating a Flow Sampler

Perform this required task to configure and enable a flow sampler.

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	**sampler** *sampler-name* **Example:** `Device(config)# sampler SAMPLER-1`	Creates a sampler and enters sampler configuration mode. • This command also allows you to modify an existing sampler.
Step 4	**description** *description* **Example:** `Device(config-sampler)# description Sample at 50%`	(Optional) Creates a description for the flow sampler.
Step 5	**mode** {**random**} **1 out-of** *window-size* **Example:** `Device(config-sampler)# mode random 1 out-of 2`	Specifies the sampler mode and the flow sampler window size. • The range for the *window-size* argument is from 0 to 1024.
Step 6	**exit** **Example:** `Device(config-sampler)# exit`	Exits sampler configuration mode and returns to global configuration mode.
Step 7	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies an interface and enters interface configuration mode.
Step 8	{**ip** | **ipv6**} **flow monitor** *monitor-name* [[**sampler**] *sampler-name*] {**input** | **output**} **Example:**	Assigns the flow monitor and the flow sampler that you created to the interface to enable sampling.

	Command or Action	Purpose
	`Device(config-if)# ip flow monitor FLOW-MONITOR-1 sampler SAMPLER-1 input`	
Step 9	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 10	**show sampler sampler-name** **Example:** `Device# show sampler SAMPLER-1`	Displays the status and statistics of the flow sampler that you configured and enabled.

Applying a Flow to an Interface

You can apply a flow monitor and an optional sampler to an interface.

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(config)# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type* **Example:** `Device(config)# interface GigabitEthernet1/0/1`	Enters interface configuration mode and configures an interface. Flexible NetFlow is not supported on the L2 port-channel interface, but is supported on the L2 port-channel member ports. Flexible NetFlow is supported on the L3 port-channel interfaces and member ports but not on both at the same time.
Step 4	{**ip flow monitor** | **ipv6 flow monitor** | **datalink flow monitor**} *name* [**sampler** *name*] {**input** | **output**} **Example:** `Device(config-if)# ip flow monitor MonitorTest input`	Associates an IPv4, IPv6 and datalink flow monitor, and an optional sampler to the interface for input or output packets. **ip flow monitor** – Enables Flexible NetFlow to monitor IPv4 traffic. **ipv6 flow monitor** – Enables Flexible NetFlow to monitor IPv6 traffic.

	Command or Action	Purpose
		datalink flow monitor – Enables Flexible NetFlow to monitor non-IP traffic. **Note** You can associate multiple monitors to an interface in both input and output directions.
Step 5	**end** **Example:** `Device(config-flow-monitor)# end`	Returns to privileged EXEC mode.
Step 6	**show flow interface** [*interface-type number*] **Example:** `Device# show flow interface`	(Optional) Displays information about NetFlow on an interface.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring a Bridged NetFlow on a VLAN

You can apply a flow monitor and an optional sampler to a VLAN.

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(config)# configure terminal`	Enters global configuration mode.

Step 3	**vlan** [**configuration**] *vlan-id*
	Example:
	`Device(config)# vlan configuration 30`
	`Device(config-vlan-config)#`
		Enters VLAN or VLAN configuration mode.

	Command or Action	Purpose
Step 4	**ip flow monitor** *monitor name* [**sampler** *sampler name*] {**input** } **Example:** `Device(config-vlan-config)# ip flow monitor MonitorTest input`	Associates a flow monitor and an optional sampler to the VLAN for input packets.
Step 5	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring Layer 2 NetFlow

You can define Layer 2 keys in Flexible NetFlow records that you can use to capture flows in Layer 2 interfaces.

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(config)# configure terminal`	Enters global configuration mode.
Step 3	**flow record** *name* **Example:** `Device(config)# flow record L2_record Device(config-flow-record)#`	Enters flow record configuration mode.
Step 4	**match datalink** {**dot1q** |**ethertype** | **mac** | **vlan**} **Example:** `Device(config-flow-record)# match datalink ethertype`	Specifies the Layer 2 attribute as a key.
Step 5	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-flow-record)# end`	
Step 6	**show flow record** [*name*] **Example:** `Device# show flow record`	(Optional) Displays information about NetFlow on an interface.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Monitoring Flexible NetFlow

The commands in the following table can be used to monitor Flexible NetFlow.

Table 4: Flexible NetFlow Monitoring Commands

Command	Purpose
show flow exporter [**broker** | **export-ids** | **name** | *name* | **statistics** | **templates**]	Displays information about NetFlow flow exporters and statistics.
show flow exporter [**name** *exporter-name*]	Displays information about NetFlow flow exporters and statistics.
show flow interface	Displays information about NetFlow interfaces.
show flow monitor [**name** *exporter-name*]	Displays information about NetFlow flow monitors and statistics.
show flow monitor statistics	Displays the statistics for the flow monitor
show flow monitor cache format {**table** | **record** | **csv**}	Displays the contents of the cache for the flow monitor, in the format specified.
show flow record [**name** *record-name*]	Displays information about NetFlow flow records.
show sampler [**broker** | **name** | *name*]	Displays information about NetFlow samplers.

Configuration Examples for Flexible NetFlow

Example: Configuring a Flow

This example shows how to create a flow and apply it to an interface:

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.

Device(config)# flow export export1
Device(config-flow-exporter)# destination 10.0.101.254
Device(config-flow-exporter)# transport udp 2055
Device(config-flow-exporter)# exit
Device(config)# flow record record1
Device(config-flow-record)# match ipv4 source address
Device(config-flow-record)# match ipv4 destination address
Device(config-flow-record)# match ipv4 protocol
Device(config-flow-record)# match transport source-port
Device(config-flow-record)# match transport destination-port
Device(config-flow-record)# match flow cts source group-tag
Device(config-flow-record)# match flow cts destination group-tag
Device(config-flow-record)# collect counter byte long
Device(config-flow-record)# collect counter packet long
Device(config-flow-record)# collect timestamp absolute first
Device(config-flow-record)# collect timestamp absolute last
Device(config-flow-record)# exit
Device(config)# flow monitor monitor1
Device(config-flow-monitor)# record record1
Device(config-flow-monitor)# exporter export1
Device(config-flow-monitor)# exit
Device(config)# interface tenGigabitEthernet 1/0/1
Device(config-if)# ip flow monitor monitor1 input
Device(config-if)# end
```

Example: Monitoring IPv4 ingress traffic

This example shows how to monitor IPv4 ingress traffic (int g1/0/11 sends traffic to int g1/0/36 and int g3/0/11).

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# flow record  fr-1
Device(config-flow-record)# match ipv4 source address
Device(config-flow-record)# match ipv4 destination address
Device(config-flow-record)# match interface input
Device(config-flow-record)# collect counter bytes long
Device(config-flow-record)# collect counter packets long
Device(config-flow-record)# collect timestamp absolute first
Device(config-flow-record)# collect timestamp absolute last
Device(config-flow-record)# collect counter bytes layer2 long
Device(config-flow-record)# exit
```

```
Device(config)# flow exporter fe-ipfix6
Device(config-flow-exporter)# destination 2001:0:0:24::10
Device(config-flow-exporter)# source Vlan106
Device(config-flow-exporter)# transport udp 4739
Device(config-flow-exporter)# export-protocol ipfix
Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit

Device(config)# flow exporter fe-ipfix
Device(config-flow-exporter)# description IPFIX format collector 100.0.0.80
Device(config-flow-exporter)# destination 100.0.0.80
Device(config-flow-exporter)# dscp 30
Device(config-flow-exporter)# ttl 210
Device(config-flow-exporter)# transport udp 4739
Device(config-flow-exporter)# export-protocol ipfix
Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit

Device(config)# flow exporter fe-1
Device(config-flow-exporter)# destination 10.5.120.16
Device(config-flow-exporter)# source Vlan105
Device(config-flow-exporter)# dscp 32
Device(config-flow-exporter)# ttl 200
Device(config-flow-exporter)# transport udp 2055

Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit

Device(config)# flow monitor fm-1
Device(config-flow-monitor)# exporter fe-ipfix6
Device(config-flow-monitor)# exporter fe-ipfix
Device(config-flow-monitor)# exporter fe-1
Device(config-flow-monitor)# cache timeout inactive 60
Device(config-flow-monitor)#  cache timeout active 180
Device(config-flow-monitor)# record fr-1
Device(config-flow-monitor)# end

Device# show running-config interface g1/0/11
Device# show running-config interface g1/0/36
Device# show running-config interface g3/0/11
Device# show flow monitor fm-1 cache format table
```

Example: Monitoring IPv4 egress traffic

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# flow record  fr-1 out
Device(config-flow-record)# match ipv4 source address
Device(config-flow-record)# match ipv4 destination address
Device(config-flow-record)# match interface output
Device(config-flow-record)# collect counter bytes long
Device(config-flow-record)# collect counter packets long
Device(config-flow-record)# collect timestamp absolute first
Device(config-flow-record)# collect timestamp absolute last
Device(config-flow-record)# exit

Device(config)# flow exporter fe-1
Device(config-flow-exporter)# destination 10.5.120.16
```

```
Device(config-flow-exporter)# source Vlan105
Device(config-flow-exporter)# dscp 32
Device(config-flow-exporter)# ttl 200
Device(config-flow-exporter)# transport udp 2055
Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit

Device(config)# flow exporter fe-ipfix6
Device(config-flow-exporter)# destination 2001:0:0:24::10
Device(config-flow-exporter)# source Vlan106
Device(config-flow-exporter)# transport udp 4739
Device(config-flow-exporter)# export-protocol ipfix
Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit

Device(config)# flow exporter fe-ipfix
Device(config-flow-exporter)# description IPFIX format collector 100.0.0.80
Device(config-flow-exporter)# destination 100.0.0.80
Device(config-flow-exporter)# dscp 30
Device(config-flow-exporter)# ttl 210
Device(config-flow-exporter)# transport udp 4739
Device(config-flow-exporter)# export-protocol ipfix
Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit

Device(config)# flow monitor fm-1-output
Device(config-flow-monitor)# exporter fe-1
Device(config-flow-monitor)# exporter fe-ipfix6
Device(config-flow-monitor)# exporter fe-ipfix
Device(config-flow-monitor)# cache timeout inactive 50
Device(config-flow-monitor)#  cache timeout active 120
Device(config-flow-monitor)# record fr-1-out
Device(config-flow-monitor)# end

Device# show flow monitor fm-1-output cache format table
```

Example: Configuring Flexible NetFlow for Ingress VRF Support

The following example configures the collection of theVRF ID from incoming packets on a device by applying an input flow monitor having a flow record that collects the VRF ID as a key field.

```
Device> enable
Device# configure terminal
Device(config)# flow record rm_1
Device(config-flow-record)# match routing vrf input
Device(config-flow-record)# match ipv4 source address
Device(config-flow-record)# match ipv4 destination address
Device(config-flow-record)# collect interface input
Device(config-flow-record)# collect interface output
Device(config-flow-record)# collect counter packets
Device(config-flow-record)# exit

Device(config)# flow monitor mm_1
Device(config-flow-record)# record rm_1
Device(config-flow-record)# exit

Device(config)# interface GigabitEthernet 1/0/1
Device(config-if)# ip vrf forwarding green
Device(config-if)# ip address 172.16.2.2 255.255.255.252
```

```
Device(config-if)# ip flow monitor mm_1 input
Device(config-if)# end
```

Feature Information for Flexible NetFlow

This table provides release and related 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	Feature Information
Cisco IOS XE Everest 16.5.1a	Flexible NetFlow	This feature was introduced.
Cisco IOS XE Gibraltar 16.12.1	Support for SGT and DGT fields	Support was introduced for SGT and DGT fields over FNF for IPv6 traffic.
Cisco IOS XE Amsterdam 17.1.1	Ingress and Egress Flexible Netflow on MPLS	Support was introduced for Ingress and Egress Flexible Netflow on MPLS (IP level) .
Cisco IOS XE Amsterdam 17.2.1	VPN ID in Flexible Netflow	Support was introduced for configuring VPN ID in Flexible Netflow.