



Cisco Multi NetFlow Collector User Guide

Release 6.0

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Customer Order Number:
Text Part Number: OL-12885-01

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

CCVP, the Cisco logo, and Welcome to the Human Network are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn is a service mark of Cisco Systems, Inc.; and Access Registrar, Aironet, BPX, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Enterprise/Solver, EtherChannel, EtherFast, EtherSwitch, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, iQuick Study, LightStream, Linksys, MeetingPlace, MGX, Networkers, Networking Academy, Network Registrar, PIX, ProConnect, ScriptShare, SMARTnet, StackWise, The Fastest Way to Increase Your Internet Quotient, and TransPath are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0710R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Cisco Multi NetFlow Collector User Guide

© 2007 Cisco Systems, Inc. All rights reserved.



CONTENTS

About This Guide vii

Objective vii

Audience vii

How This Guide Is Organized viii

Command Syntax Conventions viii

Obtaining Documentation, Obtaining Support, and Security Guidelines ix

CHAPTER 1

Overview 1-1

What Are NetFlow Services? 1-1

NetFlow Services Device and IOS Release Support 1-2

NetFlow Data Export 1-2

How and When Flow Statistics Are Exported 1-2

NetFlow Data Export Formats 1-3

What Is Cisco NetFlow Collector? 1-4

What Is Cisco Multi NetFlow Collector? 1-5

CHAPTER 2

Using Multi NetFlow Collector 2-1

Starting and Stopping MNFC 2-1

The Main MNFC Components 2-1

The cscomnfc Script 2-2

The mnfc Script 2-3

Starting and Stopping the MNFC Storage Manager 2-3

Starting and Stopping the MNFC Database 2-3

Starting the Cisco Multi NetFlow Collector User Interface 2-3

Using the Cisco Multi NetFlow Collector User Interface 2-4

The MNFC Login Window 2-4

Navigation 2-5

Working with MNFC Configuration 2-6

Deleting Database Tables 2-7

Collectors 2-7

Adding a Collector 2-8

Inter-tier Communications 2-9

Aggregators	2-9
Adding an Aggregator	2-10
Metadata Transfer for Aggregators	2-11
Records Retention and Data Latency in Primary Table	2-11
Summarizations	2-12
Adding a Summarization	2-12
Storage Option <i>Top N Values</i>	2-14
Datasources	2-14
Correlators	2-15
Adding a Correlator	2-16
Timestamps on Application Data	2-16
Configuring Indexing	2-17
Default Indexing	2-17
Adding an Index	2-17
Working with MNFC Report	2-18
Report Fundamentals	2-19
Supported Report Types	2-20
Report Specs Browser	2-20
Defining Report Specification	2-21
Functionality Common for all Report Specifications	2-21
Time Coverage Parameter in Report Specifications	2-22
Setting the Report's Target Table	2-22
Determining the Latest Available Timestamp	2-23
Defining a Trending Report	2-24
Defining a PE-PE Report	2-25
Configuring the PE Devices	2-25
Configuring the 1st Tier NFC	2-25
Configuring MNFC	2-27
Reporting	2-28
Defining a CE-CE Report	2-29
Configuring PE Devices	2-29
Configuring 1st Tier NFCs	2-30
MNFC Configuration for CE-CE Reporting: Collector and Aggregator	2-33
MNFC Configuration for CE-CE Reporting: VPN MIB Collector	2-34
MNFC Configuration for CE-CE Reporting: CLI Collector	2-34
MNFC Configuration for CE-CE Reporting: Correlators	2-35
Specifying a CE-CE Report	2-39
Defining a VPN Traffic Summary Report	2-39
Configuring NetFlow export on devices	2-39
Configuring 1st Tier NFCs	2-39

	VPN Traffic Summary on MNFC	2-40
	Executing Reports in Background	2-40
	Interactively Executing Reports	2-40
	Scheduled Reports	2-40
	Report Results Browser	2-41
	Viewing Report Results in Tabular Format	2-41
	Viewing Report Results as Graphs	2-42
	Saving, Exporting and Printing Reports	2-42
	Retention of Report Results	2-42
	Report Results Retention Rules Form	2-42
	Working with MNFC Status and Control	2-43
	Controlling MNFC Processes	2-43
	Monitoring MNFC Status	2-43
	Metadata Transfer	2-43
	File Transport	2-44
	Data Upload Status	2-44
	Monitoring Log Files	2-44
CHAPTER 3	Multi NetFlow Collector Advanced Features	3-1
	Storage Manager	3-1
	Process Watcher	3-2
	Process Watcher Configuration	3-2
	Transport	3-3
CHAPTER 4	Multi NetFlow Collector Logging	4-1
	Configuration	4-1
APPENDIX A	Troubleshooting the Multi NetFlow Collector	A-1
	Solving Multi NetFlow Collector Problems	A-1
APPENDIX B	Database Fragmentation Profiles	B-1
APPENDIX C	MNFC Aggregated File Transfer and Internal Tables	C-1
	Aggregated File Transfer and Uploading Sequence	C-1
	MNFC Internal Tables	C-2
INDEX		



About This Guide

Objective

The *Cisco Multi NetFlow Collector User Guide* describes the Cisco Multi NetFlow Collector application, which is used with the NetFlow services data export feature on Cisco routers and Catalyst switches. This document also describes the system requirements that must be met to install the Cisco Multi NetFlow Collector product, as well as, how to install, start, and configure Cisco Multi NetFlow Collector.

NetFlow services consist of high-performance IP switching features that capture a rich set of traffic statistics exported from routers and switches while they perform their switching function. Cisco NetFlow Collector provides fast, scalable, and economical data collection from multiple export devices exporting NetFlow data records.

Cisco NetFlow Collector, Release 6.0 introduces a tiered netflow collection architecture that provides increased scalability and performance. The role of the first tier (Tier 1) maps to the NFC functionality of Cisco NetFlow Collector 5.0.3 with the addition of new features described in *Release Notes for Cisco NetFlow Collector, Release 6.0*.

Cisco NetFlow Collector, Release 6.0 supports new Cisco NetFlow Collector Tier 2 functionality, also referred to as Multi NetFlow Collector. The Multi NetFlow Collector runs on separate server hardware and provides an aggregation layer that correlates data from several Tier 1 instances.

Prior to reading this manual, you should read the *Release Notes for Cisco Multi NetFlow Collector, Release 6.0* document. These release notes provide information about known software and documentation problems and any last minute information about the Multi NetFlow Collector software not available when this guide was produced.

Audience

This guide is intended primarily for individuals with network and system administration skills. You should have a basic understanding of network design, operation, and terminology, as well as familiarity with your own network configurations. You also must have a basic familiarity with Web browsers, Red Hat Enterprise Linux, or Sun Microsystems's Solaris Operating System.

How This Guide Is Organized

This guide is organized as follows:

[Chapter 1, “Overview”](#) describes the Cisco Multi NetFlow Collector application.

[Chapter 2, “Using Multi NetFlow Collector”](#) describes how to use the Cisco Multi NetFlow Collector User Interface.

[Chapter 3, “Multi NetFlow Collector Advanced Features”](#) describes Cisco Multi NetFlow Collector advanced features.

[Chapter 4, “Multi NetFlow Collector Logging”](#) describes the Cisco Multi NetFlow Collector logging feature.

[Appendix A, “Troubleshooting the Multi NetFlow Collector”](#) contains troubleshooting information in case you encounter problems while using the Cisco Multi NetFlow Collector.

[Appendix B, “Database Fragmentation Profiles”](#) describes the Cisco Multi NetFlow Collector database fragmentation profiles.

[Appendix C, “MNFC Aggregated File Transfer and Internal Tables”](#) describes the Cisco Multi NetFlow Collector aggregated file transfer and internal tables.

An Index is also provided.

Command Syntax Conventions

[Table 1](#) describes the syntax used with the commands in this document.

Table 1 Command Syntax Guide

Convention	Description
boldface	Commands and keywords.
<i>italic</i>	Command input that is supplied by you.
[]	Keywords or arguments that appear within square brackets are optional.
{ x x x }	A choice of keywords (represented by x) appears in braces separated by vertical bars. You must select one.
^ or Ctrl	Represent the key labeled <i>Control</i> . For example, when you read ^D or <i>Ctrl-D</i> , you should hold down the Control key while you press the D key.
screen font	Examples of information displayed on the screen.
boldface screen font	Examples of information that you must enter.
< >	Nonprinting characters, such as passwords, appear in angled brackets.
[]	Default responses to system prompts appear in square brackets.

Obtaining Documentation, Obtaining Support, and Security Guidelines

For information on obtaining documentation, obtaining support, providing documentation feedback, security guidelines, and also recommended aliases and general Cisco documents, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

<http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html>



CHAPTER 1

Overview

This chapter describes the Cisco NetFlow Collector (NFC) and Multi NetFlow Collector (MNFC) applications, that are used with the NetFlow services data export feature on Cisco routers and Catalyst switches.

This chapter includes the following sections:

- [What Are NetFlow Services?](#)
- [What Is Cisco NetFlow Collector?](#)
- [What Is Cisco Multi NetFlow Collector?](#)

What Are NetFlow Services?

NetFlow services consist of high-performance IP switching features that capture a rich set of traffic statistics exported from routers and switches while they perform their switching functions. The exported NetFlow data consists of traffic flows, which are unidirectional sequences of packets between a particular source device and destination device that share the same protocol and transport-layer information. The captured traffic statistics can be used for a wide variety of purposes, such as network analysis and planning, network management, accounting, billing, and data mining.

Because of their unidirectional nature, flows from a client to a server are differentiated from flows from the server to the client. Flows are also differentiated on the basis of protocol. For example, Hypertext Transfer Protocol (HTTP) Web packets from a particular source host to a particular destination host constitute a separate flow from File Transfer Protocol (FTP) file transfer packets between the same pair of hosts.

Routers and switches identify flows by looking for the following fields within IP packets:

- Source IP address
- Destination IP address
- Source port number
- Destination port number
- Protocol type
- Type of service (ToS)
- Input interface

Catalyst 5000 series switches can identify flows by looking at a subset of these fields. For example, they can identify flows by source and destination address only.

**Note**

For Catalyst 5000 series switches, the analog to NetFlow services is integrated Multilayer Switching (MLS) management. Included are products, utilities, and partner applications designed to gather flow statistics, export the statistics, and collect and perform data reduction on the exported statistics. MLS management then forwards them to consumer applications for traffic monitoring, planning, and accounting.

NetFlow Services Device and IOS Release Support

You can find the most up-to-date information available to help you determine the compatibility among different Cisco hardware platforms, Cisco IOS software releases, and supported NetFlow data export versions at the following URL:

<http://tools.cisco.com/ITDIT/CFN/Dispatch?SearchText=Netflow&act=featSelect&rnFeatId=null&featStartsWith=&task=TextSearch&altrole=>

**Note**

Except for descriptions requiring references to specific router or switch platforms, the remainder of this chapter and the remaining chapters of this guide use the term export device instead of the terms router and switch.

NetFlow Data Export

NetFlow data export makes NetFlow traffic statistics available for purposes of network planning, billing, and so on. An export device configured for NetFlow data export maintains a flow cache used to capture flow-based traffic statistics. Traffic statistics for each active flow are maintained in the cache and are updated when packets within each flow are switched. Periodically, summary traffic statistics for all expired flows are exported from the export device by means of User Datagram Protocol (UDP) and Stream Control Transmission Protocol (SCTP) datagrams, which NetFlow Collector receives and processes.

How and When Flow Statistics Are Exported

NetFlow data exported from the export device contains NetFlow statistics for the flow cache entries that have expired since the last export. Flow cache entries expire and are flushed from the cache when one of the following conditions occurs:

- The transport protocol indicates that the connection is completed (TCP FIN) plus a small delay to allow for the completion of the FIN acknowledgment handshaking.
- Traffic inactivity expires.

For flows that remain continuously active, flow cache entries expire after a specified period of time, for example every 30 minutes, to ensure periodic reporting of active flows.

NetFlow data export packets are sent to a user-specified destination, such as the workstation running NetFlow Collector, either when the number of recently expired flows reaches a predetermined maximum, or every second-whichever occurs first. For:

- Version 1 datagrams, up to 24 flows can be sent in a single UDP datagram of approximately 1200 bytes.
- Version 5 datagrams, up to 30 flows can be sent in a single UDP datagram of approximately 1500 bytes.
- Version 7 datagrams, up to 27 flows can be sent in a single UDP datagram of approximately 1500 bytes.
- Version 8 datagrams, the number of flows sent in a single UDP datagram varies by aggregation scheme.
- Version 9 datagrams, the number of flows is variable, and depends on the number and size of fields defined in one or more templates.

See [Appendix B, “NetFlow Export Datagram Formats,”](#) in the *Cisco NetFlow Collector User Guide* for details on all versions of the NetFlow data export format.

NetFlow Data Export Formats

NetFlow exports flow information in UDP datagrams in one of five formats: Version 1 (V1), Version 5 (V5), Version 7 (V7), Version 8 (V8), or Version 9 (V9).

Version 1 is the original format supported in the initial NetFlow releases. Version 5 is an enhancement that adds Border Gateway Protocol (BGP) autonomous system information and flow sequence numbers. Version 7 is an enhancement that exclusively supports Cisco Catalyst 5000 series switches equipped with a NetFlow feature card (NFFC). V7 is not compatible with Cisco routers. Version 8 is an enhancement that adds router-based aggregation schemes. Version 9 is an enhancement to support different technologies such as Multicast, Internet Protocol Security (IPSec), and Multi Protocol Label Switching (MPLS). NetFlow Collector Release 5.0 can collect, filter, and aggregate Version 9 data in the same way it does for NetFlow Data Export Versions 1 through 8.

Versions 2, 3, 4, and 6 are not supported by NetFlow Collector. For more information on the distinctions among the NetFlow data export formats, see [Appendix B, “NetFlow Export Datagram Formats,”](#) in the *Cisco NetFlow Collector User Guide*.

The following types of information are part of the detailed traffic statistics:

- Source and destination IP addresses
- Next hop address
- Input and output interface numbers
- Number of packets in the flow
- Total bytes (octets) in the flow
- First and last time stamps of packets that were switched as part of this flow
- Source and destination port numbers
- Protocol
- Type of service (ToS)
- Source and destination autonomous system (AS) numbers, either origin or peer (present in V5 and select V8 datagrams)
- Source and destination prefix mask bits (present in V5, V7, and V8 datagrams)
- Shortcut router IP address (present in V7 on Cisco Catalyst 5000 series switches only).

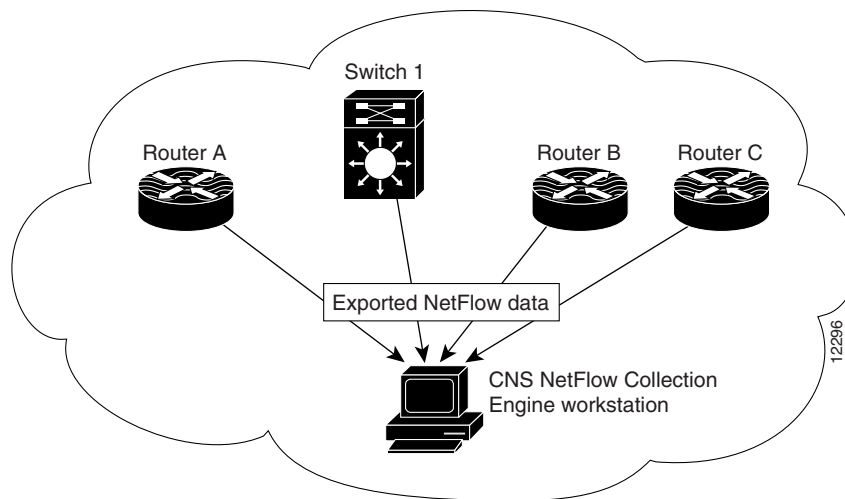
**Caution**

Throughout this publication there are numerous examples of NetFlow Collector input commands and output results. Included are examples of IP addresses. Be aware that IP address examples are not usable IP addresses. The examples do not represent real-life configurations.

What Is Cisco NetFlow Collector?

The Cisco NetFlow Collector application provides fast, scalable, and economical data collection from multiple export devices exporting NetFlow data records. [Figure 1-1](#) shows an example of a typical NetFlow data export scheme. In it, various export devices send export data to user-specified NetFlow Collector UDP and SCTP ports.

Figure 1-1 *NetFlow Collector Overview*



Each of the export devices in this example is configured for NetFlow data export. Part of the configuration information for each export device includes the IP address and the UDP or SCTP port number (a logical port designator) that identify NetFlow Collector as the receiver of flows from this export device. The port number is a user-configurable designator: you can configure NetFlow Collector to listen for flows on a number of different ports, and then configure your export devices so that each device exports flows to a dedicated port, or have a number of devices export flows to the same, shared port.

After you configure and start Cisco NetFlow Collector, it listens to the user-specified UDP and SCTP ports for exported flows from the export devices you have configured for NetFlow data export.

Cisco NetFlow Collector performs the following functions:

- NetFlow data collection from multiple export devices
- Reduction in data volume through filtering and aggregation
- Hierarchical data storage (helps client applications retrieve data)
- File system space management

Cisco NetFlow Collector collects and summarizes (aggregates) data into data files based on user-defined criteria specified in a NetFlow Collector *aggregator*. An *aggregator* is an aggregation task defined by a set of user-configurable attributes that specify how NetFlow Collector summarizes the traffic flows that are received. Three important aggregator attributes are:

- Aggregation schemes – defines the subset of data of interest in a traffic flow, as well as which statistics are kept
- Filter – criteria for accepting or rejecting flows that are aggregated (summarized)
- Port – UDP or SCTP destination port configured on the export device.

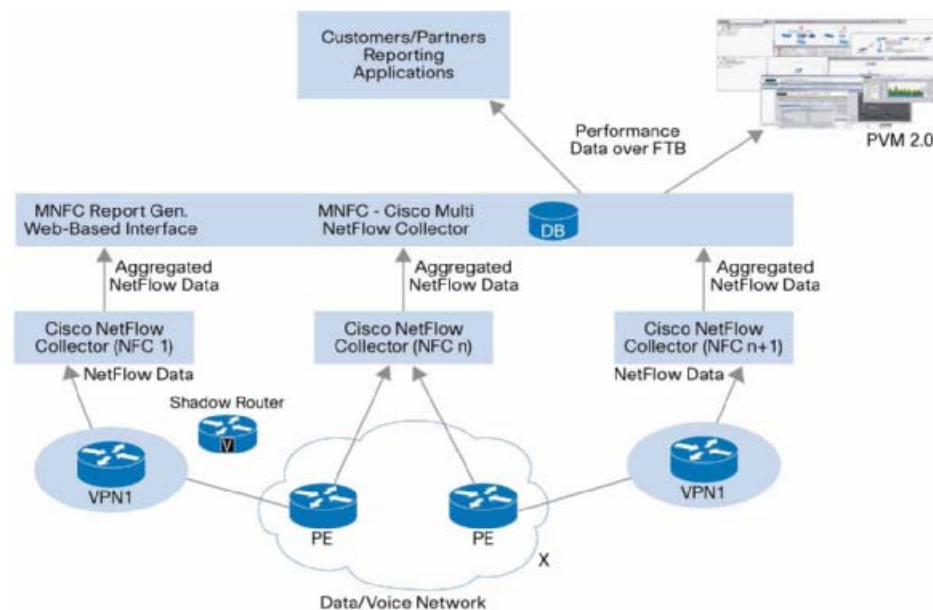
Cisco NetFlow Collector provides a set of predefined aggregation schemes to help you collect NetFlow export data and summarize the data (that is, aggregate the flows). You can choose one or more of these aggregation schemes to customize NetFlow Collector for your operating context. Moreover, starting in Release 5.0 you can modify any of the predefined aggregation schemes or define your own aggregation schemes based on them. You can also use filters with aggregation schemes to include or exclude certain types of NetFlow data.

For more information about threads, aggregation schemes, and filters, see [Chapter 4, “Customizing the CNS NetFlow Collection Engine,”](#) in the *Cisco NetFlow Collector User Guide*.

What Is Cisco Multi NetFlow Collector?

The Cisco Multi NetFlow Collector is the second-tier application of the NetFlow architecture. MNFC imports the data files resident in multiple NFCs and performs network-level correlation and provides a central view for all distributed Cisco NFC implementations in the network. [Figure 1-2](#) shows an example of a typical NetFlow data export scheme. .

Figure 1-2 Multi NetFlow Collector Overview



Cisco MNFC supports only Cisco NFC Release 6. It does not support previous NFC releases. Cisco MNFC and NFC must run on separate servers. [Table 1-1](#) describes for MNFC features.

Table 1-1 *Cisco Multi NetFlow Collector Features*

Feature	Benefit
NF-Egress Packets lost and site in-out traffic summary	Monitors packets lost from IP-IP flows. You can use this feature to monitor the point of failure of each link in the network.
PE-PE, PE-CE, CE-PE, and CE-CE data collection	Provides traffic statistics between two IP networks.
Correlation traffic summary for VPN/VRF and VPN/non-VPN	Provides a view of traffic statistics for each VPN based on each VRF. You can classify and report site-to-site and non-VPN/VPN traffic summaries.
Embedded data	Provides centralized storage of all data files from multiple distributed Cisco NFC implementations for the longer period of the trending report.
Report generator	Imports the data files resident in multiple Cisco NFCs to its server to perform network-level correlation, a central view of end-to-end traffic summaries, and classification information.



CHAPTER 2

Using Multi NetFlow Collector

This chapter contains information on common Multi NetFlow Collector (MNFC) tasks:

- [Starting and Stopping MNFC, page 2-1](#)
- [Starting the Cisco Multi NetFlow Collector User Interface, page 2-3](#)
- [Using the Cisco Multi NetFlow Collector User Interface, page 2-4](#)
- [Working with MNFC Configuration, page 2-6](#)
- [Working with MNFC Report, page 2-18](#)
- [Working with MNFC Status and Control, page 2-43](#)

Starting and Stopping MNFC

The Multi NetFlow Collector server contains number of processes which are expected to remain up and running without interruption 24 hours a day unless there is a need to perform system maintenance. MNFC also needs its database server (Informix IDS 9.40 relational database) to remain operational.

The Main MNFC Components

The Multi NetFlow Collector runs on the Java platform and includes the following processes executed in separate instances of JVM:

- Storage Manager (or **nfcdb**) executed with userid **informix**
- Process Watcher
- Concentrator
- Web GUI process (as Tomcat by Apache with web application **mnfc**)
- Report Daemon
- VPN MIB Collector
- CLI Collector

Storage Manager is the only process executed with user id **informix**. All other processes are executed with user id **nfcuser**. See [Table 2-1](#) for information about MNFC component processes.

Table 2-1 *MNFC Component Processes*

Short Name	Full Name	Purpose	UNIX User ID*	Default Presence**	Log Properties file in config directory
nfcdb	Storage Manager	Run-time maintenance of nfc_db database in IDS	informix	Mandatory	mnfcdb-log4j
pw	Process Watcher	Running of all processes except Storage Manager	nfcuser	Mandatory	mnfcpw-log4j
concentrator	Concentrator	MNFC back-end server	nfcuser	Mandatory	mnfc-log4j
web	Web GUI	Tomcat with mnfc webapp, MNFC front-end server	mfcuser	Mandatory	mnfcweb-log4j
rd	Report Daemon	Execution of reports	nfcuser	Mandatory	mnfcrd-log4j
vpnmibcltr	VPN MIB Collector	SNMP-based collection of CE-CE data	nfcuser	Optional	vpnmibcltr-log4j
clicollector	CLI Collector	CLI Collector	nfcuser	Optional	clic-log4j

* Instead of nfcuser, another id can be designated by the operator executing MNFC installation.

** With exception of Storage Manager and Process Watcher the presence flag is configurable via the Process Watcher's controls.

The cscomnfc Script

The script **cscomnfc** is intended as a single invocation point for starting up or shutting down all MNFC processes, including those executed using user ids **informix** and **nfcuser**. Run this script as the user root. When running this script, you must use either the **start** or **stop** parameter.

The location of the **cscomnfc** script is under **/opt/CSCOmnc/bin** or **\$MNFC_DIR/bin**.

The mnfc Script

The script **mnfc** is intended for starting and stopping of MNFC processes executed by Process Watcher using user id **nfcuser**. The script accepts mandatory verb parameter **start**, **stop**, **shutdown**, and **status**. If you use **start** or **stop**, then a second parameter is needed. Use the short name for the MNFC process. For a listing of MNFC processes, see [Table 2-1 on page 2-2](#).

When starting and stopping all MNFC processes for the id **nfcuser**, you must use the following:

- For starting: **mnfc start all**
- For stopping: **mnfc shutdown**

The location of the **mnfc** script is under **/opt/CSCOmnc/bin** or **\$MNFC_DIR/bin**.

Starting and Stopping the MNFC Storage Manager

To start the MNFC storage manager, as user **informix** enter the shell command **startnfcdb.sh**.

To stop the MNFC storage manager, as user **informix** enter the shell command **stopProcess.sh nfcdb**.

The location of the **startnfcdb.sh** and **stopProcess.sh** script is under **/opt/CSCOmnc/bin** or **\$MNFC_DIR/bin**.

Starting and Stopping the MNFC Database

Before the Informix database is shutdown or restarted, the Multi NetFlow Collector application must be shutdown.

For information on operating the Informix server, refer to IDS 9.40 documentation by IBM.

Starting the Cisco Multi NetFlow Collector User Interface

To start the Cisco Multi NetFlow Collector User Interface, do the following.



Note

The Cisco Multi NetFlow Collector User Interface requires JRE 1.5 or higher. You can download a plug-in for Java 1.5 or higher from java.sun.com, section **Downloads**, **J2SE** folder; and install it on the platform on which the browser will run.

Step 1 To run Cisco Multi NetFlow Collector, log in as the user specified during installation.

Step 2 Enter the following command:

```
/opt/CSCOnfc/bin/nfcollector start all
```

Step 3 From a web browser enter:
`//<nfc-hostname>:8080/mnfc`

**Note**

The web-based UI only works with the collector located on the same machine. To access a different instance of Cisco NetFlow Collector you must start that collector's web server and access it through the corresponding URL.

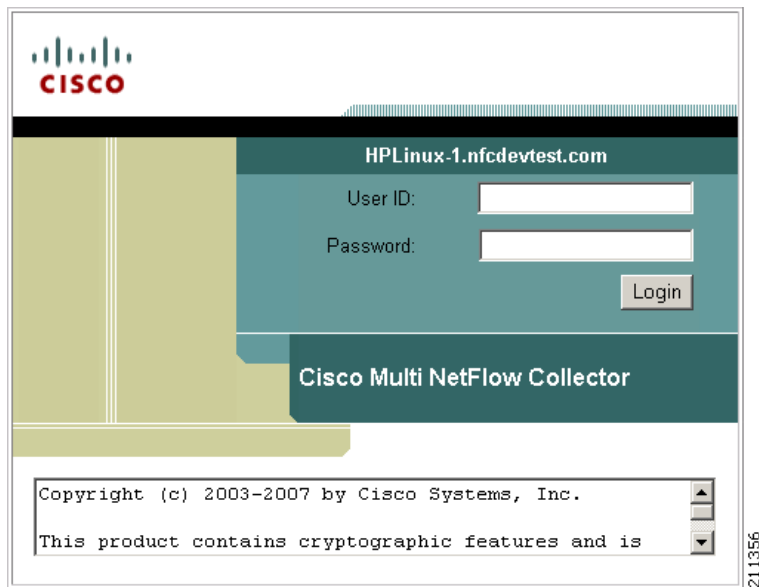
Using the Cisco Multi NetFlow Collector User Interface

The following sections describe using the Cisco Multi NetFlow Collector User Interface.

The MNFC Login Window

When starting the Cisco Multi NetFlow Collector, the first window that appears is the MNFC login window, as shown in [Figure 2-1](#). For security purposes, to use the web-based UI you must authenticate yourself with a user ID and password.

Figure 2-1 Cisco Multi NetFlow Collector User Interface Login Window



To log in to Cisco Multi NetFlow Collector, do the following:

Step 1 From the Login window, enter your User ID and Password.

Step 2 Click **Login**.

The Cisco NetFlow Collector Main window appears. From this window, you can select from the following tabs:

- Configuration

- Reports
- Status

Refer to the following sections for information on these functions.

Navigation

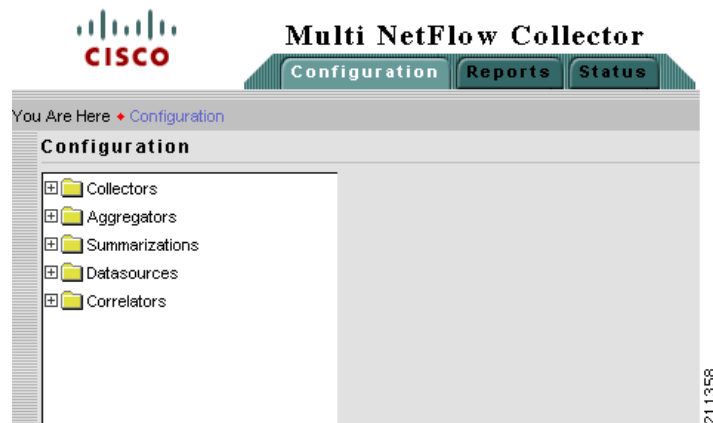
You can move around the MNFC web-based User Interface (UI) from two levels. Across the top of all MNFC windows are the MNFC UI navigation tabs. These tabs are the first level of navigation in to the MNFC UI, as shown in [Figure 2-2](#). From here you can select the **Configuration**, **Reports**, and **Status** tabs. The toolbar at the far right includes links to **Logout**, **Help**, and **About** windows.

Figure 2-2 MNFC UI Navigation Tabs



Each section of the MNFC User Interface has a navigation tree on the left-hand side, as shown in [Figure 2-3](#). This second level of navigation lets you focus in on a specific aspect of collector configuration, reporting, or status.

Figure 2-3 MNFC UI Navigation Tree

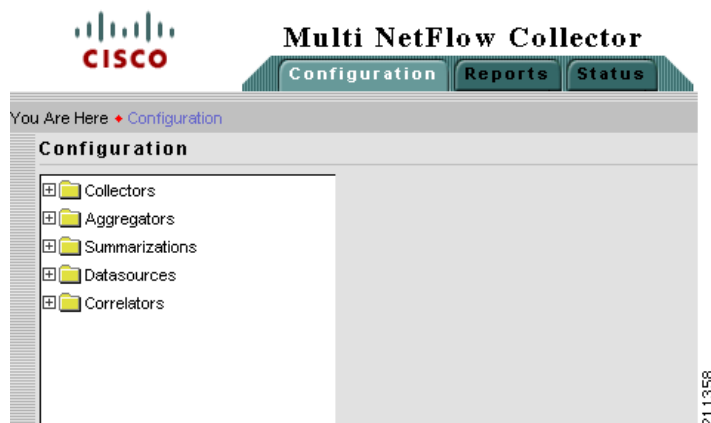


Working with MNFC Configuration

From the Configuration window you can perform tasks including specify global parameters; define fields, key builders, value builders and aggregators; and create filters.

From the Cisco Multi NetFlow Collector **Main** window, click the **Configuration** tab. The Configuration window appears, as shown in [Figure 2-4](#).

Figure 2-4 MNFC Configuration Window



From this window you can access or configure the following:

- Collectors—1st tier NetFlow Collector servers
- Aggregators—instances of NetFlow data aggregations driven by collectors
- Summarizations—time-based aggregations of information supplied by aggregators
- Datasources—data tables with non-NetFlow information domains (SNMP, user files, etc.)
- Correlators—products of joining NetFlow data with auxiliary data

With exception of Collectors, the logical entities share the following traits:

- The underlying storage for every configured entity is a table in the database, with table name equal to the unique ID of the configured instance.
- The ID should conform to constraints placed by SQL syntax standard on table names: It must contain only letters, digits and underscore symbols and must not exceed 18 characters in length.
- The IDs must be unique within the MNFC system. You can not have an instance of a Correlator with the same ID name as an Aggregator.

In addition:

- The database table-based configuration entities with exception of Aggregators should have indexes in accordance to their intended use. You should configure indexes manually.
- The database table-based configuration entities with optional exception of Datasources contain timestamp column. Every record has a field with semantic meaning of *time of record creation*. Information in Datasource can be either static or timestamped.

Deleting Database Tables

Entities such as aggregators, summarizations, and correlators have underlying tables created in the Informix database under names that coincide with the IDs of the MNFC configuration entities themselves. If you delete one of these entities from within the MNFC UI, the database table is also deleted.

Depending on the activity taking place around the database table in question, the database table might not be dropped for several minutes. Problems can arise if the database table for the deleted configuration entity is not dropped immediately, especially if you attempt to immediately create a new entity with same ID. If this happens, an Exception will be registered in Concentrator process and you will see an Error popup.

To verify if there is a delay in deleting a database table use shell access to the MNFC server to run database queries to determine whether the table designated for dropping is indeed the source of the problem.

For example, if you delete a Correlator named **corr1** and then see a failure when attempting to recreate it, use the following query:

```
select count(*) from corr1
```

This query can be used for periodic pollings of the table status. The recommended course of action is to wait for up to 10 minutes before attempting the query

Once the table in question is not seen, you can resume creation of the new configuration entity. If the table does not disappear after 10 minutes, you can manually drop the table. If you are still not able to create the new entity, contact Cisco's customer support.

Collectors

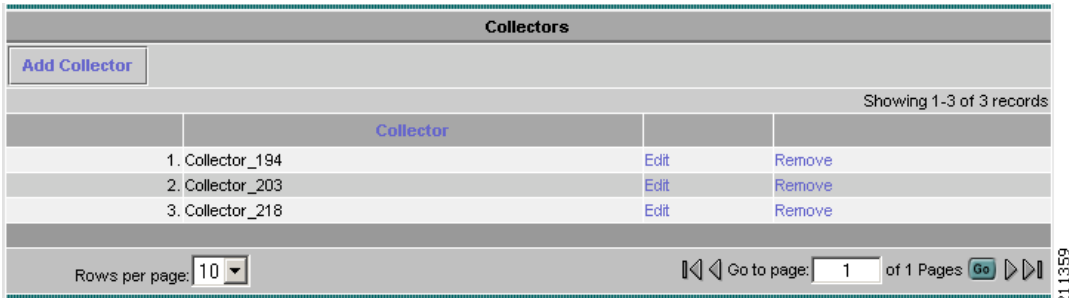
Collectors are 1st tier NFC servers used to collect and aggregate the NetFlow data from managed network devices. The aggregation done by the collectors is executed according to the aggregation schemes defined by the NFC's 1st tier configuration over periods of typically 1 to 5 minutes. Some aggregation periods can take up to 15 minutes. Each separate instance of the aggregation on the 1st tier NFC host is defined as 1st tier aggregator and is described in detail in *Cisco NetFlow Collector User Guide*.

In order to configure the transfer of data aggregated by the 1st tier NFC into MNFC (as 2nd tier NetFlow Concentrator server) and insertion of the aggregated records in database the following should be configured on MNFC server:

- The identity of 1st tier NFC server must be defined in MNFC as Collector
- The 2nd tier Aggregator must be defined to match 1st tier aggregator(s) on collector NFC(s)

Click on the **Collectors** folder of the MNFC UI navigation tree to display a table of all existing collectors, as shown in [Figure 2-5](#).

Figure 2-5 Collectors Window



The screenshot shows the 'Collectors' window. At the top is a header bar with the title 'Collectors'. Below the header is a button labeled 'Add Collector'. Underneath is a table with the following data:

	Collector		
1.	Collector_194	Edit	Remove
2.	Collector_203	Edit	Remove
3.	Collector_218	Edit	Remove

Below the table, there is a 'Rows per page' dropdown set to '10' and a pagination control showing 'Go to page: 1 of 1 Pages' with a 'Go' button. The text 'Showing 1-3 of 3 records' is also present.

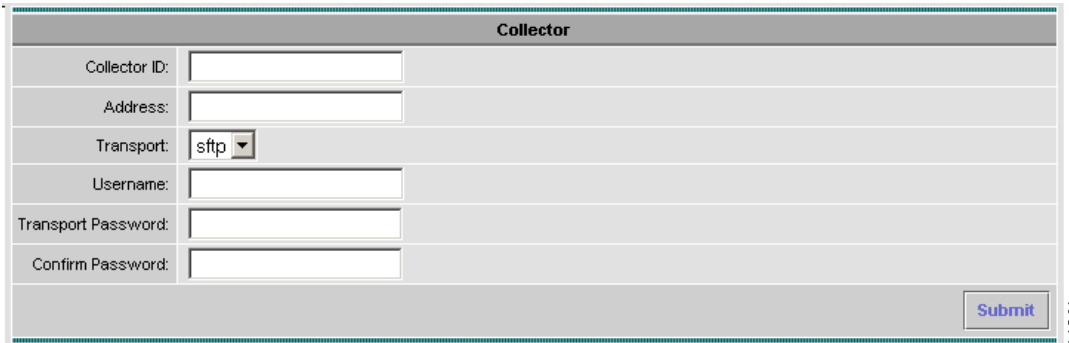
If a 1st tier NFC from which data is to be collected is located in a different time zone than that of the MNFC server, the following configuration must be done at that NFC in order for MNFC to correctly process input from the NFC:

- Step 1** Log in to the web UI for the 1st tier NFC from which data is to be collected.
- Step 2** Navigate **Configuration > Global**.
- Step 3** Click the checkbox **Include GMT offset in Filenames**:
- Step 4** Click **Submit**.

Adding a Collector

From the Collectors window, click on **Add Collector** to bring up the Add Collector window to define a new collector. See [Figure 2-6](#).

Figure 2-6 Add Collectors Window



The screenshot shows the 'Add Collectors' window. It has a header bar with the title 'Collector'. Below the header are several input fields:

- Collector ID: [text input]
- Address: [text input]
- Transport: [dropdown menu with 'sftp' selected]
- Username: [text input]
- Transport Password: [text input]
- Confirm Password: [text input]

At the bottom right of the form is a 'Submit' button.

To configure a Collector (1st tier NFC server) in MNFC, define the following fields:

- Collector ID** identifies this collector instance will be referenced by 2nd tier Aggregators.
- Address** identifies the 1st tier NFC on the network. You can enter either an IP address or a NIS-supported host name.

- **Transport** defines the protocol to be used to retrieve the aggregated data files being output of 1st tier Aggregators and input for primary tables' insertion. See the section [Aggregators](#). MNFC currently supports the FTP and SFTP protocols.
- **Username** and **Transport Password** are credentials to be used in transport connection to the specified 1st tier server; the default NFC user's **nfcuser** account is convenient choice

All Collector attributes except the ID field can be modified later at any time using Modify Collector.



Note

Defining data using **Add/Modify Collector** modifies only the internal operation of the MNFC configuration. Changes are stored in an XLM configuration file and do not include contacting the 1st tier server defined as a Collector.

Fill in the fields and click **Submit** to complete the operation.

Inter-tier Communications

The actual communication between MNFC and NFC configured as Collector takes place only during and after the definition of Aggregator entities (described below).

MNFC (as a Concentrator) and Collector NFC communicate on two separate channels:

- Metadata Transfer done over the application's RMI/XML connection and used to retrieve the definitions of Aggregators, Aggregation Schemes, etc.
- File Transport used periodically to retrieve the files with aggregated data from Aggregators defined in MNFC as 2nd tier aggregators.

Aggregators

An aggregator is defined on MNFC as a destination and repository for NetFlow records aggregated by one or more 1st tier collector NFCs. The schema (set of key and value fields) or granularity (aggregation period) of the records are not configurable on MNFC level but are defined by collecting 1st tier NFC server. If more than one 1st tier NFC is used to drive the same 2nd tier aggregator then the following properties of the 1st tier aggregators should match:

- Aggregator name (id)
- Aggregation scheme (key/value fields must be configured identically and in the same order)
- Aggregation period
- Output data files' sorting option

The records from all 1st tier aggregators named 'X' are transported into MNFC as a 2nd tier NFC server and inserted into the primary database table under same name 'X'. (If the name of the aggregator on 1st tier does not conform with the DDL conventions then straightforward conversion rule is applied).

The value of retention period in MNFC can be configured by picking value from set of available choices. Which values will be available as valid retention span choices depends on size and configuration of the disk partitions used by MNFC for the primary tables.

The **Sort Output** option on the 1st tier NFC server must be used for Aggregators that will be defined by the Summarization instance using the storage method **Top N Values**. The **Output Format** option should always be **default**. See the *Cisco NetFlow Collector User Guide* for details on Aggregator specifications.

Click on the **Aggregators** folder of the MNFC UI navigation tree to display a table of all existing aggregators, as shown in [Figure 2-7](#).

Figure 2-7 *Aggregators Window*

Aggregators			
Add Aggregator			
Showing 1-4 of 4 records			
	Aggregator		
1.	PacketSection	Edit	Remove
2.	SCTP	Edit	Remove
3.	FCS	Edit	Remove
4.	cematrix	Edit	Remove
Rows per page: <input type="text" value="10"/> Go to page: <input type="text" value="1"/> of 1 Pages Go			

211362

Adding an Aggregator

From the Aggregators window, click on **Add Aggregator** to bring up the Add Aggregator window to define a new aggregator. See [Figure 2-8](#).

Figure 2-8 *Add Aggregators Window*

Aggregator ID:

State:

Retention Period:

Available Collectors

Collector_194
Collector_203
Collector_218

Selected Collectors

> Add >>

<< Remove <

Submit

211363

To configure an Aggregator, identify one or more 1st tier NFCs defined with in MNFC as Collectors and make sure the Aggregators defined there have matching definitions. These Collectors' IDs should be selected on Aggregator form.

Fill in the fields and click **Submit** to complete the operation. Once you click Submit, MNFC contacts the selected Collector(s) and engages in a Metadata Transfer operation.

To modify or remove an existing aggregator, click **Edit** for the aggregator which you wish to modify or remove from the list of aggregators displayed in the Aggregator window. The **Modify Aggregator** window displays,



Note

Multi NetFlow Collector Release 6 does not support manipulation of Aggregator's status.

Metadata Transfer for Aggregators

MNFC starts the metadata transfer once you click Submit from the Add Aggregator window. The transfer is carried out over MNFC's RMI/XML communication (proprietary to Cisco) and its results can be monitored by navigating **Status > Metadata Transfer**.

Records Retention and Data Latency in Primary Table

Repositories for periodic data in the MNFC primary table are backed by tables in the MNFC database. In most cases these tables will be fragmented and will use all **nfc_data** dbspaces so that the whole retention time span will be divided into number of fragments, each used for a fixed period set to one of the following retention intervals: 10, 15, 20, 30, or 60 minutes .



Note

In Multi NetFlow Collector Release 6, record retention is optimized for performance and can not be changed once defined.

Table 2-2 Latency and Retention in Primary Table

Option	Latency Interval	24 Hour Retention Span	48 Hour Retention Span	96 Hour Retention Span	192 Hour Retention Span
1	10 minutes	4 hours	8 hours	16 hours	32 hours
2	15 minutes	6 hours	12 hours	1 day	2 days
3	20 minutes	8 hours	16 hours	32 hours	2 days 16 hours
4	30 minutes	12 hours	1 day	2 days	4 days
5	60 minutes	1 day	2 days	4 days	8 days

Summarizations

Summarizations are entities defined on MNFC as aggregations of the NetFlow data from primary tables over longer time periods. While the aggregation period on 1st tier NFC servers (thus defining the granularity of data in primary table) typically does not exceed 15 minutes, the minimal period of summarization accordingly equals 15 minutes. The summarization can be seen as a continuation of data aggregation done over periods of 30 minutes, 1 hour, or more. It is possible to define multiple summarizations of the aggregator. Each of these summarizations will be kept in separate database table. For every summarization must be configured maximal retention period, all available values of which are expressed as multiples of summarization period, exact numbers depending on configuration of the database.

Click on the **Summarizations** folder of the MNFC UI navigation tree to display a table of all existing summarizations, as shown in [Figure 2-9](#).

Figure 2-9 *Summarizations Window*

Summarizations			
Add Summarization			
Showing 1-3 of 3 records			
	Summarization		
1.	cematrix_60m	Edit	Remove
2.	Sum_FCS_N	Edit	Remove
3.	Sum	Edit	Remove
Rows per page: 10 Go to page: 1 of 1 Pages Go			

211364

Adding a Summarization

From the **Summarizations** window, click on **Add Summarization** to bring up the Add Summarization window to define a new aggregator. See [Figure 2-10](#).

Figure 2-10 Add Summarization Window

The screenshot displays the 'Multi NetFlow Collector' web application. The left sidebar shows a navigation tree with 'Summarizations' selected. The main panel is titled 'Summarization' and contains the following configuration options:

- Summarization ID:** A text input field.
- State:** A dropdown menu set to 'active'.
- Period Duration (Frequency):** A dropdown menu set to '15min'.
- Retention: x Periods:** A dropdown menu set to '24'.
- Aggregator:** A dropdown menu set to 'PacketSection' with a 'Set Scheme' button next to it.
- Key Fields:** Two lists: 'Available Key Fields' (empty) and 'Selected Key Fields' (containing 'timestamp').
- Function:** A dropdown menu set to 'sum'.
- Value Fields:** Two lists: 'Available Value Fields' (empty) and 'Selected Value Fields' (empty).
- Retention: Top N values / period:** A text input field.
- Ordered By:** A dropdown menu.

Below the configuration section is an 'Indexing' section showing 'No records' and a table with the header 'Index Name'. The bottom of the page includes pagination controls: 'Rows per page: 10', 'Go to page: 1 of 1 Pages', and 'Submit'/'Discard' buttons.

To specify a new Summarization instance, define the following:

- Unique ID conforming to common constraints placed on IDs in MNFC
- The aggregation period— frequency of data collation
- The retention period—as multiple of summarization periods
- The source of records to be summarized
- Exact configuration of the keyset used for aggregation—in GROUP By
- Set of values to be aggregated (summarized fields)
- The aggregation function to be used over the aggregated values; default being sum(). Note that for some values there is an inherent aggregation function that will be used no matter what the field setting. For example, for the **starttime** function, **min()** will be always invoked.

211365

Optionally you can define a limited number of records to be derived and retained for every period, with the ordering done by values. For instance, to determine the topmost records one of the summarized values must be chosen for ordering criterion.

Fill in the fields and click **Submit** to complete the operation.

Storage Option *Top N Values*

When defining the Summarization entity, you have the option to restrict the scope of the records retained in the database table for every summarization period. You can select one value field and specify that for every period the summarized records are ordered, only those having value belonging to the topmost *N* per device will be retained. The number of records retained for a given period can be greater than *N* multiplied by number of active devices because it is possible to have multiple records with the same value fitting in to the Top *N* subset.

Defining the Top *N* saves space for the chosen summarization scheme as it makes saving records for small values of a chosen field unnecessary. The drawback to defining the Top *N* is that once defined, you can not specify a different value or different keyset. Therefore it is recommended to use the summarizations with Top *N* Values method only when you have a definite use case for tracking the top values in a particular field record for a certain keyset.

Note the following in MNFC Release 6.0:

- The summarization scheme with Top *N* Value storage method does not support reduction of the field set from the source Aggregator instance; all keys in the source aggregation scheme should be selected for summarization.
- Aggregations with multiple value fields are not supported. You can not derive summaries for Top *N* Values in say octets column while the records contain also packets count, etc.
- Only one summarization scheme with Top *N* Values storage method can be defined per the same instance of an Aggregator. It still can share the same Aggregator instance with one or more Summarizations defined with unbounded storage.

Due to these restrictions, when specifying the used keyset and value for the tracking of records for Top *N* Value, you must define the dedicated aggregation scheme and aggregator instance for 1st tier NFC collectors with the exact keyset and the only value field.



Note

Summarizations with the Top *N* values storage method always require that the Aggregators on all 1st tier Collectors to be configured with the **Sort Output** option.

Datasources

Datasources are logical entities in MNFC that define database tables with non-NetFlow data populated either by auxiliary data collectors (SNMP, CLI-based) or by means external to MNFC. Their main purpose is to define tables used by Correlators in combination with NetFlow tables.

This release contains built-in definitions of Datasources activated by starting of non-NetFlow collector processes (VPN MIB Collector and CLI Collector) used for the CE-CE reporting feature.

Click on the **Datasource** folder of the MNFC UI navigation tree to display a table of all existing datasources, as shown in [Figure 2-11](#).

Figure 2-11 Datasource Window

Datasources			
Add Datasource			
Showing 1-3 of 3 records			
	Datasource		
1.	ifindex_to_vrfname	Edit	Remove
2.	egress_vrf_routes	Edit	Remove
3.	ingress_vrf_routes	Edit	Remove
Rows per page: <input type="text" value="10"/>			
Go to page: <input type="text" value="1"/> of 1 Pages Go			

211366

Correlators

Correlators are tables with periodic information collected from NetFlow information with additional information coming from one of the Datasources. The main source can be NetFlow records from the primary or summarized table or another Correlator. The maximal retention period is configured as a multiple of the correlation periods from the available restricted set of values.



Note

Be sure that the Datasource table to be used when defining the Correlator is present in the database before you begin defining the Correlator.

Click on the **Correlators** folder of the MNFC UI navigation tree to display a table of all existing correlators, as shown in [Figure 2-12](#).

Figure 2-12 Correlators Window


Correlators			
Add Correlator			
Showing 1-3 of 3 records			
	Correlator		
1.	cematrix_vrf	Edit	Remove
2.	cematrix_label	Edit	Remove
3.	cematrix_hourly	Edit	Remove
Rows per page: <input type="text" value="10"/>			
Go to page: <input type="text" value="1"/> of 1 Pages Go			

211369

Adding a Correlator

From the **Correlators** window, click on **Add Correlator** to bring up the Add Correlator window to define a new correlator. See [Figure 2-13](#).

Figure 2-13 **Add Correlator Window**



Multi NetFlow Collector

Configuration
Reports
Status

[Logout](#) | [Help](#) | [About](#)

You Are Here > Configuration > Specify Correlator
User ID: nfcuser

Correlator

- Collectors
- Aggregators
- Summarizations
- Datasources
- Correlators
 - cematrix_hourly**
 - cematrix_label
 - cematrix_vrf

Correlator	
Correlator ID:	cematrix_hourly
State:	active
Frequency:	hourly
Retention: x Periods:	24
Aggregator & Datasource:	cematrix_label join egress_vrf_routes
Key Fields:	timestamp customer_name ingress_ce ingress_vrfname ingress_pe_addr egress_pe egress_ce egress_vrfname
Value Fields:	pkts octets

Condition			
Add Join			
Showing 1-2 of 2 records			
Field			
1. out_label		Edit	Remove
2. egress_pe		Edit	Remove
Rows per page: 10		Go to page: 1 of 1 Pages Go	

Indexing	
Add Index	
Showing 0-0 of 0 records	
Index Name	
No records.	
Rows per page: 10	
Go to page: 1 of 1 Pages Go	

Submit
Discard

Fill in the fields and click **Submit** to complete the operation.

Timestamps on Application Data

All application data, such as NetFlow records and other statistics collected from monitored network, is kept by MNFC in database tables as timestamped records. Every table with dynamic data which is not constant in time has a column of DATETIME YEAR TO SECOND type named **timestamp** by default.

The following functionality is reliant on having timestamps on data records:

- Retention of records in database tables
- Calculation of Summarizations and Correlated tables
- Reporting — for more on the Reporting Time Coverage concept see the [“Defining Report Specification” section on page 2-21](#).

The timestamp records the time when the measurement was taken. For example:

- In a primary table — the time when the record was generated by Collector.
- In a derived table — the beginning of summarization/correlation period. For example, if the hourly summary has timestamp value of 9:00am then the summarized records were taken from range from 09:00:00 to 09:59:59.

**Note**

There is a difference between 1st tier Nfc and MNFC timestamps. The NFC collector generates the measurement at the end of the aggregation period and outputs the measurements into a file with the current time value. These records are then uploaded by MNFC and given the same timestamp as a reference to the statistical record creation time, while the preceding aggregation interval can be in fact variable or unknown.

Configuring Indexing

Entities such as summarizations and correlators have underlying tables created in the Informix database. These tables require indexing for optimal performance. By default, the MNFC application performs minimal indexing. Additionally, you can define indices for aggregators, summarizations, and correlators using the MNFC UI.

In the case of datasources, they are not managed by MNFC. However, you can also define indices for datasources using the MNFC UI. Datasources supporting PE-PE and CE-CE reports have required indices.

Default Indexing

The following indices are created for fragmented tables driven by MNFC for aggregators (primary tables), summarizations, and correlators (derived tables):

- By timestamp column
- By spacen (integer number X signifying allocation of data fragment to **dbspace nfc_dataX**)

Adding an Index

From the Cisco Multi NetFlow Collector **Main** window, click the **Configuration** tab. Navigate to **Summarizations**, **Datasources**, or **Correlators**. The Indexing area is near the bottom of the window. See [Figure 2-14](#).

Figure 2-14 Indexing Window

Click **Add Index** to open the Add Index window. See [Figure 2-15](#).

Figure 2-15 Add Index Window

Define the following:

- Index ID—generic constraints of uniqueness and SQL syntax apply
- Key selection—pick the required keys and arrange them in desired order. If the query for which you are creating the index uses certain keys in particular order, then be sure that the index uses the keys in same order.

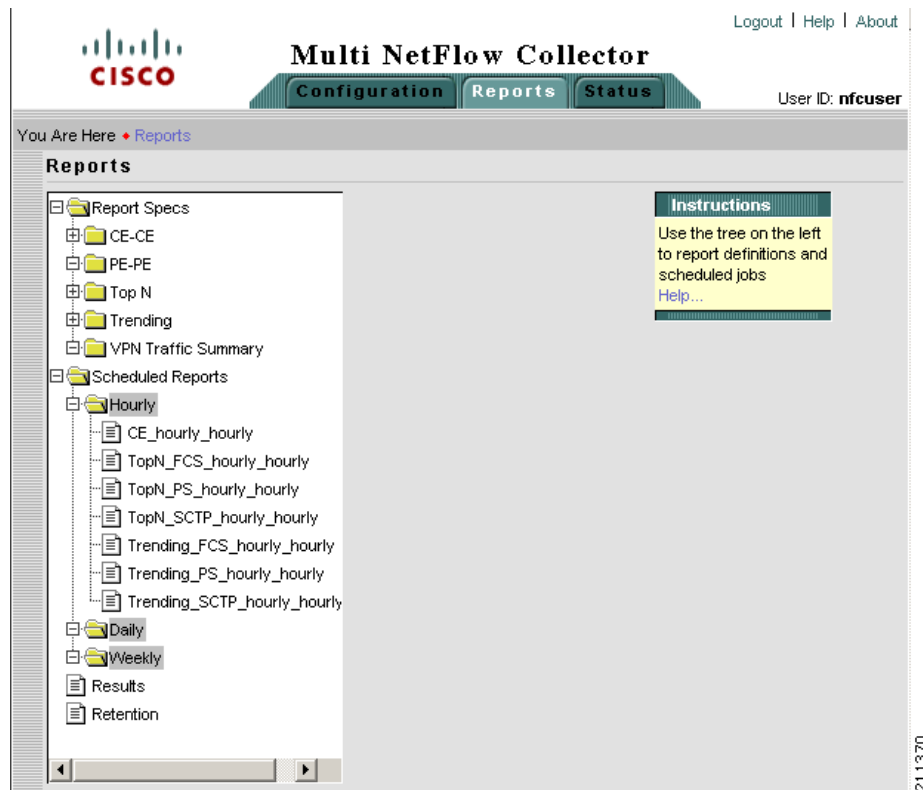
Fill in the fields and click **Submit** to complete the operation. The Indexing window displays the defined index(ices) in tabular form.

Working with MNFC Report

Cisco Multi NetFlow Collector reports are in effect a summary of the Multi NetFlow Collector's aggregated output. NetFlow data is first aggregated into Multi NetFlow Collector output files by the collector, and then the data in those files is further aggregated to generate a report. Reports are either custom (run immediately) or scheduled.

From the Cisco Multi NetFlow Collector **Main** window, click the **Reports** tab. The Reports window appears, as shown in [Figure 2-16](#).

Figure 2-16 MNFC Reports Window



Report Fundamentals

The components to the report processing within MNFC include:

- **Report type.** Report type is a template for a report specification. Report types define the basic structure of a report, and currently are predefined within MNFC Release 6.
- **Report specification.** Report specification is a named persistent definition of a query or queries, which produces a report result when invoked. Every report specification has an ID, a reference to its target table, and a time-coverage parameter associated with it. Report specifications are stored as XML documents on the MNFC server under the **\$MNFC_DIR/Reports/<Report-Type>** directories.

Report specifications can be run one or more times; both manually or using a scheduler.

Although the report types are static, the report specifications that are generated from them are not. You can create report specifications to address your specific goals.

- **Report result.** This is a result instance of a single invocation of a report specification. Report results are stored as XML documents, with name and contents identifying the report spec ID and the time of invocation, along with the results. Report Results conceptually are tables, but also can be seen in graphical form.

XML files with Report Results are stored on the MNFC server under the **\$MNFC_DIR/Reports/<Report-Type>/<Report-Spec>** directories and are subject to Report Results' Retention Rule.

Supported Report Types

MNFC release 6.0 supports the following report types:

- Top *N* reports
- Trending reports
- PE-PE reports
- CE-CE reports
- VPN Traffic Summary report

Report Specs Browser

From the Reports screen, you can view all Report Specifications (with their Report Types) currently existing on the MNFC server as shown in [Figure 2-17](#).

Figure 2-17 MNFC Report Specs Window

Report Spec	Report Type	Launch	Schedule	Browse Results	Edit ...	Remove
1. CE_hourly	CE-CE	Launch	Schedule	Browse Results	Edit ...	Remove
2. TopN_SCTP_hourly	Top N	Launch	Schedule	Browse Results	Edit ...	Remove
3. Trending_PS_hourly	Trending	Launch	Schedule	Browse Results	Edit ...	Remove
4. TopN_PS_hourly	Top N	Launch	Schedule	Browse Results	Edit ...	Remove
5. Trending_FCS_hourly	Trending	Launch	Schedule	Browse Results	Edit ...	Remove
6. Trending_SCTP_hourly	Trending	Launch	Schedule	Browse Results	Edit ...	Remove
7. TopN_FCS_hourly	Top N	Launch	Schedule	Browse Results	Edit ...	Remove

Showing 1-7 of 7 records

Rows per page: 10 Go to page: 1 of 1 Pages Go

.Actions available for displayed Report Specs include:

- **Launch:** The immediate start of the report execution in background mode. Once launched, MNFC displays the informational popup **Report Launched**. The execution status and the run's results can be followed using the Report Results Browser screen.
- **Schedule:** Provided that the Report Spec instance is escheatable, this option prompts you to **Add Scheduled Report** for the default scheduling period.
 - Only Report Specs with relative time coverage are schedulable, while the spec with absolute coverage can not be referenced by the scheduled job.
 - Usually the default scheduling period for report is **Daily**; this can be changed using advanced MNFC configuration methods.
- **Browse Results:** Opens the Report Results Browser screen with the pre-selected values of Report Spec ID and Report Type.
- **Edit:** Allows you to modify the spec. The type and exact configuration of the Spec Editor screen depends on the Report Type of the selected specification instance.
- **Remove:** Deletes the Report Specification instance from the MNFC server along with all Report Results for this Report Spec.

A drop-down selector **Specify New Report** above the report specs browser table is used for creation of the new instances of Report Specifications.

Defining Report Specification

Report Specification is a logical entity used to define an instance of report of specified type. Once a Report Spec is defined, the report can be generated one or more times.

If you are using the relative method of reporting time coverage definition, the same Report Specification can be used for repeated launch of multiple report invocations.

The **Specify <Report-Type> Report** screen is a Report Spec editor. The exact appearance of this screen depends on the Report Type. You can access this screen by doing one of the following:

- For a new instance of Report Spec select **Add New ... selector** at the top of the Report Spec Browser screen.
- For an existing instance of Report Spec select the **Edit ...** hyperlink from the respective row on the Report Spec Browser table.
- For an existing instance of Report Spec referred by the current Scheduled Report Job select the **Report Spec ID** hyperlink on Scheduled Job specification screen.

Functionality Common for all Report Specifications

Despite having dedicated spec editor screens that are dependant on the chosen Report Type, all report specifications share the following common traits and functions:

- Setting of the target table: the report runs as query/queries over specified a MNFC repository in the server's database. It can be a table with primary, summarized or correlated records. The specified report will have a logical source set as Aggregator, Summarization or Correlator respectively.
- Provided the target table is selected, the report spec must have definitions of keys and values to be used in the report.
- Definition of the timing conditions on records selected from the target table, called query's **Time Coverage**, so that the report takes only records with timestamp value falling between specified beginning and end of the specified interval.
- A **Generate** button for starting of immediate report execution in interactive mode.
- A **Launch** button for starting of report execution in background.
- A **Browse Results** button that takes you directly to the Report Results Browser screen.
- A **Schedule Hourly** button that takes you directly to the Add Scheduled Job screen.
- A **Save** button for storing of the Report Specification instance in permanent repository.
- A **Discard** button for cancellation of the recent changes.

Time Coverage Parameter in Report Specifications

The following screen displays a generic Report Specification used to set the **Time Coverage** parameter. This parameter defines the start and end times for which records are collected in to the reporting SQL query from the database table. See [Figure 2-18](#).

Figure 2-18 MNFC Parameter Window

The screenshot shows a window titled "Specify CE-CE Report". It contains a "Report Id:" field with the value "CE_hourly". Below this is the "Time Coverage:" section. It has two radio buttons: "Absolute" and "Relative". The "Absolute" radio button is selected. Under "Absolute", there are two columns: "Start" and "End". Each column has a "Date (dd MMM yyyy):" field and a "Time (hh:mm:ss):" field. For "Start", the date is "23 May 2007" and the time is "17:56:10". For "End", the date is "23 May 2007" and the time is "18:55:09". The "Relative" radio button is also present but not selected. Under "Relative", there are two columns: "Time delta (exec time - delta)" and "Time span (start time + span)". Each column has "Days:", "Hours:", and "Minutes:" fields. For "Time delta", the values are Days: 0, Hours: 2, Minutes: 0. For "Time span", the values are Days: 0, Hours: 0, Minutes: 59. A vertical text "211372" is visible on the right side of the window.

Two methods are available for defining of the starting and end times for selected records:

- **Absolute:** Both the start and end times are set explicitly using the Date and Time fields.
- **Relative:** The start of the reporting time window is defined as preceding the execution time of the report by specified Time delta value, while length of the specified interval is set as Time Span. For example, if the report defined by present Report Spec is instantiated to run on midnight then the start time would fall on 11:00pm and will cover records with timestamp not exceeding 23:59pm. This feature is useful for defining reports that will be placed on the Scheduler for periodic execution.

Setting the Report's Target Table

The field **Record Type** contains a pull-down menu of valid report targets. These timestamped database tables driven by configurable MNFC entities include

- Aggregators (primary tables)
- Summarizations
- Correlators

The Top N report provides the same functionality as the Custom Reports in 1st tier NFC. See [Figure 2-19](#).

Figure 2-19 Specify Top N Report Window

Multi NetFlow Collector

Configuration Reports Status

User ID: nfcuser

Are Here > Reports > Specify Top N Report

Specify Top N Report

Report Id: ReportConfig-20070523-1857

Time Coverage:

Start **End**

☐ Absolute Date (dd MMM yyyy): 23 May 2007 Time (hh:mm:ss): 17:57:35

☐ Relative Date (dd MMM yyyy): 23 May 2007 Time (hh:mm:ss): 18:56:34

Time delta (exec time - delta) Time span (start time + span)

☐ Relative Days: 0 Hours: 2 Minutes: 0 Days: 0 Hours: 0 Minutes: 59

Record Type: PacketSection

Devices:

☐ Combine devices

☐ Separate devices

☐ Single device: 172.20.98.193

Key Fields:

Available Key Fields: srcaddr, dstaddr, payloadcapture, prot

Selected Key Fields:

Value Fields:

Available Value Fields: pkts

Selected Value Fields:

N (Maximum Rows): 20

Ordered By: ☐ Asc ☒ Desc

Include All: ☐

Generate Launch Browse Results... Schedule Hourly Save Discard

211373

Determining the Latest Available Timestamp

The **End Time** parameter for the executing report is always compared with the timestamp of the latest records available in the target table. If **End Time** falls later than the data records available in the target storage, the report will not run. The report remains waiting for a limited time-out period to accommodate cases of data showing up in target table only a few moments later than usual; however you should verify that the defined report will find the data it is targeted to run.

The following cases should be considered:

- For Aggregator, the data latency is defined implicitly by its specified retention span. See the [“Records Retention and Data Latency in Primary Table” section on page 2-11](#).
- For Summarization or Correlator, the data latency is explicitly determined by their period. Unless their calculation is lagging because the server is overloaded, then for the repository having period P, the data for the period starting at HH:MM will show up after (HH:MM + 3 * P) time. For example, for a 15 min summary the data with a timestamp of 6:00pm should appear as available after 6:45pm. For the time coverage of the report that starts at 7:00 pm its end time should be set no later than 6:14 or 5:59.

In case the processing in MNFC is backlogged, the data records will show up in their storages with delay so you can determine the **latest available** value by directly querying the database table `storage_directory`.

Defining a Trending Report

The Trending report in MNFC contains an additional or last key in the GROUP BY clause: the timestamp of the record. Reports of this type always query the specified target storage for full distribution of the specified values grouped by specified combination of device(s), keys, and the record time as follows:

- The device field can be reported as:
 - **Combined:** Represented in output by asterisk
 - **Separate:** Showing every value in the query results
 - Specified by the concrete value from pull-down menu
- One or more keys, each either reported separately (showing every value in the query results) or restricted to single specified concrete value.
- The time parameter can be reported either with maximal granularity available in the specified target table or the reported results can be bundled into specified time intervals. The available intervals are multiples of the storage’s granularity.

For example, if the table contains records at 5 minute intervals and the time coverage is specified for 1 hour, then the maximal granularity will result in maximum of 12 intervals for every combination of other fields (provided the data are found), and other available choices will be 10, 15, 20 or 30 minute granularity resulting in 6, 4, 3 or 2 intervals respectively.



Note

Currently at least one key must be included in the Report Spec. You can not omit all keys.

Trending always shows the specified values’ totals for the reported interval, not rates. For example, if you are viewing traffic figures for 5 minute intervals and then decrease the granularity to 15 minute, then the reported values are expected to grow by a factor of 3, because the intervals are three times longer and will contain approximately three times more of traffic.

Plotting the trending report results in a line graph, or graphs if multiple values are reported (described in the [“Viewing Report Results as Graphs” section on page 2-42](#)). Sometimes the plotting is impossible to execute due to having too many data points for every combination of other fields. When this occurs, displaying the results in tabular format as the only option available. If you wish to see the results in line graph format, specify a more restrictive key specification and increase the time granularity.

Defining a PE-PE Report

A PE-PE traffic summary provides the total byte count and packet count of traffic between any pair of Provider Edge devices (PE) for a given period. Optionally, this summary can be drilled down to report traffic in different classes based on IP precedence. The final report is available on MNFC, while certain configuration is required on the 1st tier NFC and the PE devices.

Configuring the PE Devices

On the PEs, NetFlow should be enabled on all CE-facing interfaces using the following command:

```
Router(config-if)# ip route-cache flow
```

The **BGP next hop** field is required for PE-PE reports. To enable **BGP next hop export**, use the following commands at the global level:

```
Router# ip flow-export version 9 bgp-nexthop
```

```
Router# ip flow-export destination <NFC host IP address> <NFC port number>
```

Configuring the 1st Tier NFC

The 1st tier NFC must contain an aggregator that has at least one key, **egress-pe**, and two values, **byte count** and **packet count**. Refer to the following link for defining aggregator and aggregation scheme:

http://www.cisco.com/en/US/products/sw/netmgmtsw/ps1964/products_user_guide_chapter09186a008085fd55.html

You also need a configuration file, `/opt/CSCOnfc/config/peList.conf`, that contains the IDs of all PE devices in the provider network. Either the IP address or host name can be used as the ID of a PE device. Comment lines in this file should start with a `#` character. A sample of this file looks as follows:

```
#complete list of PE IP addresses
0.0.0.2
0.0.0.4
0.0.0.10
host1
```

If you want the report to contain traffic classification, another key needs to be included to prepare the IP precedence information. The definition of this key builder is shown in [Figure 2-20](#).

Figure 2-20 Key Builder Window

Key Builder	
ID:	ip-precedence-key
Type:	Bit Field
Output Name:	ip-precedence
Field:	tos
Least Significant Bit:	5
Number of Bits:	3
Format:	Decimal
Allow Null Value:	<input type="checkbox"/>

For an example of an aggregation scheme with this additional key builder, see [Figure 2-21](#).

Figure 2-21 Aggregation Scheme Window

Aggregation Scheme ID: pematrix	
Key Fields:	
Available Key Fields srcaddr-key dstaddr-key src-mask-key dst-mask-key src-subnet-key dst-subnet-key masked-srcaddr-key masked-dstaddr-key srcport-key dstport-key	Selected Key Fields egress-pe-key ip-precedence-key
Value Fields:	
Available Value Fields flow-count-value inBytes inPackets outBytes outPackets siteInBytes siteInPackets siteOutBytes siteOutPackets flow-rate-value	Selected Value Fields packet-count-value byte-count-value

Configuring MNFC

On the MNFC server, you need to configure an aggregator to collect the data from the 1st tier NFC. See [Figure 2-22](#).

Figure 2-22 **Aggregator Window**

The screenshot shows the 'Aggregator' configuration window. At the top, the title bar reads 'Aggregator'. Below the title bar, there are three fields: 'Aggregator ID:' with the value 'pematrix', 'State:' with a dropdown menu set to 'active', and 'Retention Period:' with the value '32 hours'. Below these fields, there are two list boxes. The left list box is titled 'Available Collectors' and contains the items 'syates-lnx' and 'x336-3'. The right list box is titled 'Selected Collectors' and contains the item 'x336-2'. Between the two list boxes are two buttons: '> Add >>' and '<< Remove <'. Both list boxes have vertical scroll bars.

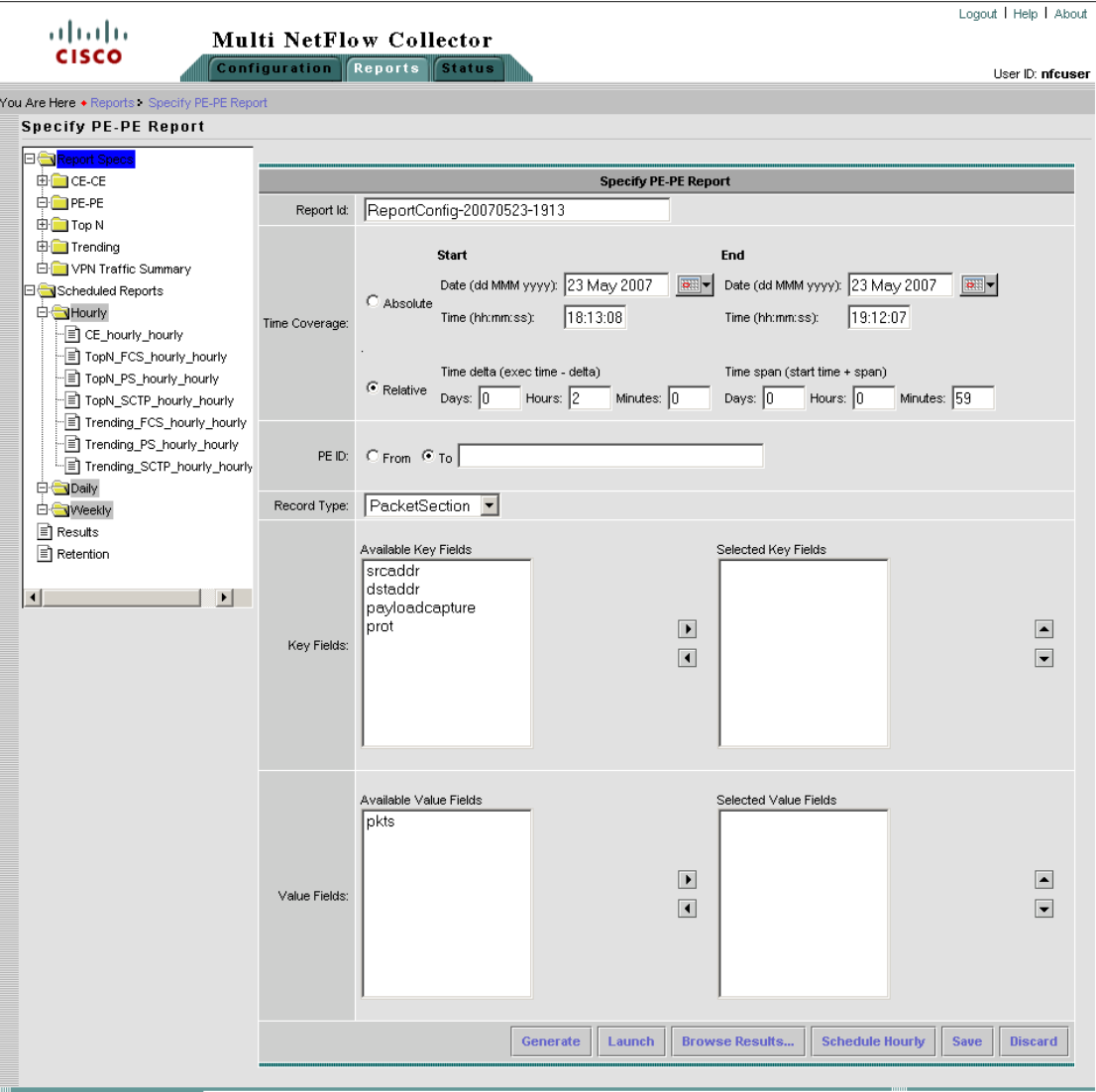
You can define the appropriate summarization on top of pematrix aggregator based on their report time window.

Reporting

From the Cisco Multi NetFlow Collector **Main** window, click the **Reports** tab. The Reports window appears,

You can specify a PE-PE Traffic Report by navigating **Reports > Report Specs > PE-PE** to access the Specify PE-PE Report window shown in [Figure 2-23](#).

Figure 2-23 PE-PE Report Window



Note that you can leave the **PE ID** field blank. If you leave this field blank, all values of PE IDs will be included in the report. If you do enter a value in the **PE ID** field, the report will only include records with an ingress or egress (from or to) the PE matching that value.

To run a report from a specific PE to all other PEs, enter the **PE ID** and select the **From** radio button in the **PE ID** row. Similarly, to run a report to a specific PE from all other PEs, enter the **PE ID** and select the **To** button.

Defining a CE-CE Report

A CE-CE Traffic Summary reports traffic volumes between any pair of customer edge (CE) routers for a certain customer. All Provider Edge (PE) routers in a provider network export traffic statistics (NetFlow Data Export or NDE) to a few hosts that run NetFlow Collection Engine (NFC) 6.0. These NFCs conduct aggregation of the NDE they receive. The MNFC then collects aggregation results from the 1st tier NFCs and produces the reports.

**Note**

This feature requires Cisco NetFlow Collector, Release 6.0.

Generating a CE-CE Traffic Summary requires specific configuration on the PE devices, 1st tier NFC, and MNFC.

Configuring PE Devices

On PEs, ingress NetFlow should be enabled on all CE-facing interfaces using the following command:

Router(config-if)# ip route-cache flow

Or, for 12.2(14)S, 12.2(15)T or later IOS release,

Router(config-if)# ip flow ingress

The **BGP next hop** field is required to generate a CE-CE Traffic Summary. To enable **BGP next hop** export (supported since IOS 12.0(26)S) and specify the NDE destination, use the following commands at the global level:

Router# ip flow-export version 9 bgp-nexthop

Router# ip flow-export destination <NFC host IP address> <NFC port number>

Configuring 1st Tier NFCs

On the 1st tier NFC, create an aggregator with aggregation scheme *cematrix*. See [Figure 2-24](#).

Figure 2-24 *Modify Aggregator Window*

Modify Aggregator	
Aggregator ID:	cematrix
Aggregation Scheme:	cematrix ▼
Aggregation Period (mins):	1
Port Number:	9991
Protocol:	udp ▼
State:	active ▼
Data Set Path:	\${NFC_DIR}/Data
Output Format:	default ▼
Compression:	<input type="checkbox"/>
Maximum Disk Usage (MBs):	0
Filter:	ingress-only ▼
Sort Output:	<input type="checkbox"/>
Threshold Directory:	\${NFC_DIR}/threshold-
Threshold Output Format:	mixed ▼

The filter **ingress-only** is only required when the egress Netflow is enabled on the PE devices. The filter condition requires a **Direction Key Builder**. See the following figure.

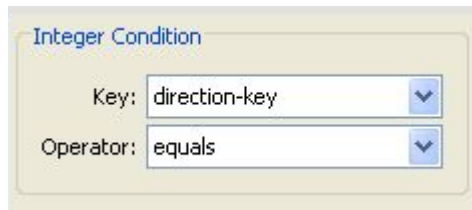
Key Builder	
ID: *	direction-key
Type: *	Integer ▼
Output Name:	direction
Field: *	DIRECTION ▼
Format: *	Decimal ▼
Allow Null Value:	<input checked="" type="checkbox"/>
<div>Modify Remove</div>	

To configure an ingress-only filter, do the following:

Navigate **Configuration > Add Filter**

In the **Add Filter** field, type **ingress-only** and click **Add Condition** at the bottom of the window.

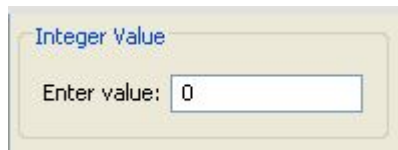
In the Integer Condition area of the window, select **direction-key** for the **Key** and **equals** for the **Operator**. See below.



The 'Integer Condition' dialog box contains two dropdown menus. The first is labeled 'Key:' and has 'direction-key' selected. The second is labeled 'Operator:' and has 'equals' selected.

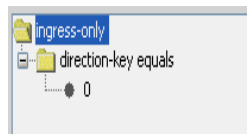
Click **Add value** at the bottom of the window.

In the Integer Value area of the window, enter **0**. See below.



The 'Integer Value' dialog box has a text input field labeled 'Enter value:' with the number '0' entered.

Click **Update filter** at the bottom of the window. The ingress-only filter is created, as shown in the following figure.



See [Figure 2-25](#) for a sample aggregation scheme definition window.

Figure 2-25 *Modify Aggregator Scheme Window*

You need to set the environment variable **MNFC_DIR** to **/opt/CSCOmncf**.

The **\$MNFC_DIR/config/peList.conf** file needs to include the loopback addresses or hostnames of all PEs exporting to this NFC. A sample of this configuration file looks as follows:

```
# This file is for the PE-PE traffic summary only
# It should contain a list of IDs for all PE devices in the provider network
# ID of PE device can be either host name or IP address
192.168.200.2
192.168.200.3
192.168.200.4
```

Another related requirement is as follows, in order to resolve the host name of egress PE device: For all PE devices exporting to this NFC server, the IP addresses of their loopback interfaces need to be resolvable with DNS lookup. Please verify this with utility program such as nslookup or host on the NFC host.

In addition, for the current release, users are required to create and maintain a file, **/opt/CSCOnfc/config/vpn.conf**, that contains VPN configuration. A sample file reads:

```
172.20.98.250, FastEthernet0/1.401, vpn1-branchB, CERouter-3, Cisco
172.20.98.250, FastEthernet0/1.601, vpn2-branchB, CERouter-4, IBM
172.20.98.248, FastEthernet2/1, vpn2-branchA, CERouter-2, IBM
172.20.98.246, FastEthernet0/1, vpn1-branchA, CERouter-1, Cisco
```

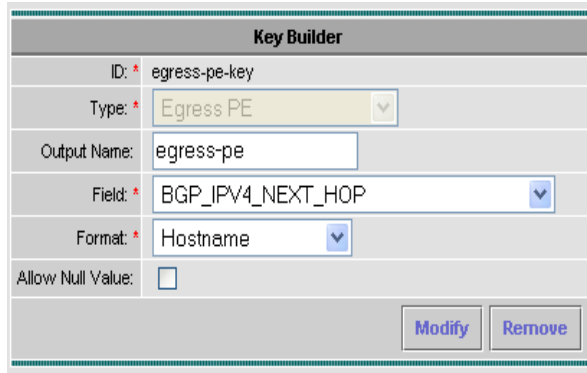
Each entry corresponds to one CE-facing interface one PE exporting NDE to the NFC server. The fields are separated by commas. Each row contains five columns: PE address (what is specified as flow export source on the PE), full interface name, site name, CE name, and customer name.

This file is shared with Site In/Out Traffic Summary feature. The value of site name can be empty string for CE-CE Traffic Summary.

MNFC Configuration for CE-CE Reporting: Collector and Aggregator

On the MNFC server you should first define the 1st tier NFCs from which you want to pull data. See [Figure 2-26](#). Make sure that the **egress-pe-key** is defined on the NFC as shown below.

Figure 2-26 Key Builder Window

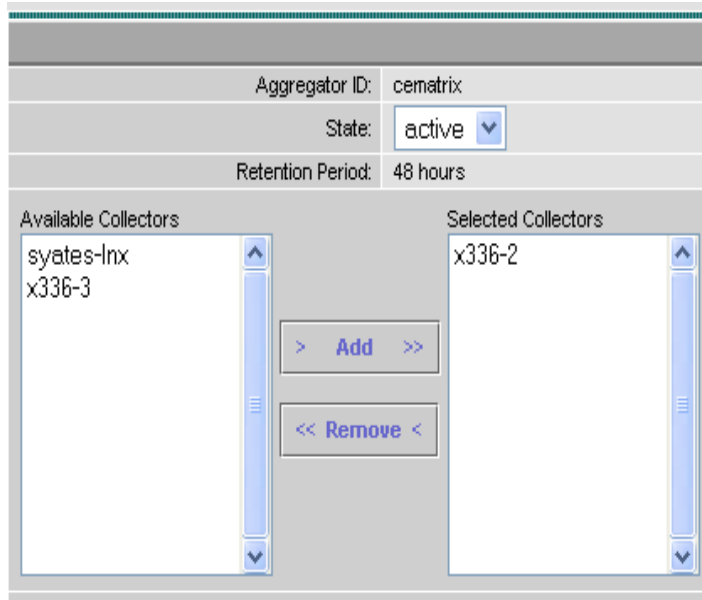


The Key Builder window is a form for defining a key. It contains the following fields and controls:

- ID:** * egress-pe-key
- Type:** * Egress PE (dropdown menu)
- Output Name:** egress-pe (text input)
- Field:** * BGP_IPV4_NEXT_HOP (dropdown menu)
- Format:** * Hostname (dropdown menu)
- Allow Null Value:** ☐
- Buttons:** Modify, Remove

Define an aggregator **cematrix** with the same name as the aggregator on NFC whose results will be pulled. See [Figure 27](#).

Figure 27 Aggregator Window



The Aggregator Window is a form for defining an aggregator. It contains the following fields and controls:

- Aggregator ID:** cematrix
- State:** active (dropdown menu)
- Retention Period:** 48 hours
- Available Collectors:** syates-lnx, x336-3
- Selected Collectors:** x336-2
- Buttons:** > Add >>, << Remove <

Next you need to prepare mapping data for VRF information and egress CE.

MNFC Configuration for CE-CE Reporting: VPN MIB Collector

First use the VPN MIB Collector to pull VRF information from PE devices. Check the file **\$MNFC_DIR/config/vpnmibcltr.xml** and make sure it contains a list of all PEs in your network:

```
<!-- List of PE to collect VPN MIB tables from -->
<pe-list>
  <pe id="172.20.98.250"/>
  <pe id="172.20.98.248"/>
  <pe id="172.20.98.246"/>
</pe-list>

<!-- SNMP ReadOnly community string. Default is "public" for all devices -->
<comm-string>public</comm-string> <!-- use this line to overwrite the default CS -->
<!-- <comm-string device="172.18.102.230">mystring</comm-string> -->
```

As **nfcuser**, start the VPN MIB Collector which will populate the **DataSource ifindex_to_vrfname** file. The command line signature is:

```
$MNFC_DIR/bin/mnfc start vpnmibcltr
```

See [Figure 2-28](#) for a sample datasource **ifindex_to_vrfname** window. The values in this window are automatically populated.

Figure 2-28 *Modify Datasource Window*

Datasource			
Datasource name: ifindex_to_vrfname			
Fields			
Add Field			
Showing 1-2 of 2 records			
	Field Name		
1.	ifindex	Edit	Remove
2.	vrfname	Edit	Remove
Rows per page: 10		Go to page: 1 of 1 Pages Go	
Indexing			
Add Index			
Showing 0-0 of 0 records			
	Index Name		
No records.			
Rows per page: 10		Go to page: 1 of 1 Pages Go	
			Submit

211375

MNFC Configuration for CE-CE Reporting: CLI Collector

The CLI collector collects necessary mapping information to resolve egress CE.

Prior to starting the CLI collector, make sure that the default PE router telnet password is reset in the file **\$MNFC_DIR/config/clc.properties** if it is not **cisco**:

```
defaultPassword=<password>
```

Also make sure that DNS lookup works for the PE addresses (configured as NDE source on devices). Then start the CLI collector using the command:

```
$MNFC_DIR/bin/mnfc start clicollector
```

Two data sources, **ingress_vrf_routes** and **egress_vrf_routes**, will be automatically populated. See [Figure 2-29](#).

Figure 2-29 *Modify Datasource Window*

The screenshot shows the 'Modify Datasource Window' for the datasource named 'ingress_vrf_routes'. The window is divided into two main sections: 'Fields' and 'Indexing'.

Fields Section:

- Buttons: Add Field
- Showing 1-4 of 4 records

Field Name		
1. ingress_pe_addr	Edit	Remove
2. ingress_vrfname	Edit	Remove
3. network	Edit	Remove
4. out_label	Edit	Remove

- Rows per page: 10
- Go to page: 1 of 1 Pages

Indexing Section:

- Buttons: Add Index
- Showing 0-0 of 0 records

Index Name
No records.

- Rows per page: 10
- Go to page: 1 of 1 Pages

At the bottom right, there is a 'Submit' button.

211376

MNFC Configuration for CE-CE Reporting: Correlators

You need three cascaded correlators to resolve all necessary fields for a CE-CE summary.

Correlator **cematrix_vrf** to resolve ingress vrf. See [Figure 2-30](#).

Figure 2-30 Correlator cematrix_vrf Window

The screenshot shows the 'Multi NetFlow Collector' interface with the 'Configuration' tab selected. The left sidebar shows a tree view with 'Correlators' expanded, highlighting 'cematrix_vrf'. The main area displays the configuration for 'cematrix_vrf'.

Correlator Configuration:

- Correlator ID: cematrix_vrf
- State: active
- Frequency: hourly
- Retention: x Periods: 24
- Aggregator & Datasource: cematrix_60m join ifindex_to_vrfname
- Key Fields: timestamp, dst_subnet, egress_pe, ingress_ce, customer_name, vrfdevice, vrfname
- Value Fields: octets, pkts

Condition Section:

Showing 1-2 of 2 records

Field		
1. device	Edit	Remove
2. input	Edit	Remove

Rows per page: 10 | Go to page: 1 of 1 Pages

Indexing Section:

Showing 0-0 of 0 records

No records.

Rows per page: 10 | Go to page: 1 of 1 Pages

Buttons: Submit, Discard

The two joins are

cematrix.device = ifindex_to_vrfname.vrfdevice

cematrix.input = ifindex_to_vrfname.ifindex

See the following figure.

This is a close-up of the 'Condition' section from the previous screenshot. It shows a table with two rows of conditions and their corresponding edit and remove buttons.

Field		
1. device	Edit	Remove
2. input	Edit	Remove

Showing 1-2 of 2 records

1. **cematrix_label** to resolve label:

The screenshot shows the Cisco Multi NetFlow Collector web interface. The top navigation bar includes 'Configuration', 'Reports', and 'Status'. The user is logged in as 'nfcuser'. The breadcrumb trail indicates 'You Are Here > Configuration > Specify Correlator'.

Correlator Configuration:

- Correlator ID:** cematrix_label
- State:** active
- Frequency:** hourly
- Retention: x Periods:** 24
- Aggregator & Datasource:** cematrix_vrf join ingress_vrf_routes
- Key Fields:** timestamp, customer_name, ingress_ce, ingress_vrfname, ingress_pe_addr, egress_pe, out_label
- Value Fields:** pkts, octets

Condition Section:

Showing 1-3 of 3 records

Field		
1. vrfname	Edit	Remove
2. dst_subnet	Edit	Remove
3. vrfdevice	Edit	Remove

Rows per page: 10 | Go to page: 1 of 1 Pages

Indexing Section:

Showing 0-0 of 0 records

No records.

Rows per page: 10 | Go to page: 1 of 1 Pages

Buttons: Submit, Discard

The three joins are:

Cematrix.vrfname=ingress_vrf_routes.ingress_vrfname


Cematrix.dst_subnet=ingress_vrf_routes.network

Cematrix.vrfdevice=ingress_vrf_routes.ingress_pe_addr

See the following figure.

Condition		
Add Join		
Showing 1-3 of 3 records		
Field		
1. vrfname	Edit	Remove
2. dst_subnet	Edit	Remove
3. vrfdevice	Edit	Remove

2. The last correlator resolves egress CE and egress VRF:



Multi NetFlow Collector

Configuration
Reports
Status

[Logout](#) | [Help](#) | [About](#)

You Are Here > Configuration > Specify Correlator
User ID: nfcuser

Correlator

- Collectors
- Aggregators
- Summarizations
- Datasources
- Correlators
 - cematrix_hourly**
 - cematrix_label
 - cematrix_vrf

Correlator	
Correlator ID:	cematrix_hourly
State:	active
Frequency:	hourly
Retention: x Periods	24
Aggregator & Datasource:	cematrix_label join egress_vrf_routes
Key Fields:	timestamp customer_name ingress_ce ingress_vrfname ingress_pe_addr egress_pe egress_ce egress_vrfname
Value Fields:	pkts octets

Condition			
Add Join			
Showing 1-2 of 2 records			
Field			
1. out_label		Edit	Remove
2. egress_pe		Edit	Remove
Rows per page: 10		Go to page: 1 of 1 Pages Go	

Indexing	
Add Index	
Showing 0-0 of 0 records	
Index Name	
No records.	
Rows per page: 10	
Go to page: 1 of 1 Pages Go	

[Submit](#)
[Discard](#)

The two joins are:

```
Cematrix_label.out_label = egress_vrf_routes.in_label
```

```
Cematrix_label.egress_pe = egress_vrf_routes.hostname
```

The last correlator creates an hourly CE-CE summary table.

See the following figure.

Condition			
<div>Add Join</div>			
Showing 1-2 of 2 records			
	Field		
1.	out_label	Edit	Remove
2.	egress_pe	Edit	Remove

The last correlator creates an hourly CE-CE summary table.

Specifying a CE-CE Report

You can run CE-CE reports on the hourly summary table. To launch the **SpecifyCECEReport** window, navigate **Reports > Report Specs > CE-CE**.

You can leave the **Customer Name** or **CE ID** field blank. When you leave these fields blank, the report will include all values of **Customer Name** and **CE IDs**. If you do enter a **Customer Name** value, the report will include only records with the customer name equal to the entered value. Similarly, if you type in a value in **CE ID** field, the report will only include records with an ingress or egress (from or to) CE matching that value.

To run a report from a specific CE to all other CEs of a certain customer, enter the **Customer Name** and **CE ID** and select the **From** button in the **CE ID** row. Similarly, to run a report to a specific CE from all other CEs of a customer, enter the **Customer Name** and **CE ID** and select the **To** button.

Defining a VPN Traffic Summary Report

The VPN Traffic Summary Report shows ingress and egress packet and byte counts over the specified reporting period for VPN and non-VPN traffic through PE-CE interfaces on which NetFlow is enabled. In addition, the difference between ingress and egress packet counts is shown. When NetFlow is configured on all PE-CE interfaces, the VPN packet difference can be interpreted as an approximation of VPN packet loss, because the majority of traffic that enters the network but does not leave consists of dropped packets.

Configuring NetFlow export on devices

To configure NetFlow export to generate a VPN Traffic Summary Report, do the following.

For every PE device, in addition to normal NetFlow destination device and port configurations, configure the ingress NetFlow, egress NetFlow, and VPN-aware egress NetFlow on every CE-facing interface on the PE device:

```
ip flow ingress
```

```
ip flow egress
```

```
mpls netflow egress
```

Configuring 1st Tier NFCs

On each first-tier NFC configured as a NetFlow destination, create an entry in the file **/opt/CSCOnfc/config/vpn.conf** for each CE-facing PE interface as documented above, for example:

```
172.20.98.250, FastEthernet0/1.401, vpn1-branchB, CERouter-3, Cisco
```

Create an aggregator with the pre-defined aggregation scheme **VPNTraffic** on the configured port. For this example, this aggregator is also named **VPNTraffic**.

VPN Traffic Summary on MNFC

Under the **Configuration** tab, create a collector for each first tier device.

Under the **Configuration** tab, create an aggregator for the VPNTraffic aggregator, and associate with each first tier collector.

Under the **Reports** tab, create a report spec of type VPN Traffic Summary. For **Record Type**, specify either the name of the first tier aggregator created above (VPNTraffic) to run a report against primary table data, or specify a summarization that was created separately for summarized VPNTraffic data.

At this point the report can be run, launched, scheduled, etc.

Executing Reports in Background

To schedule report execution in the background, you need to create and save a **Report Spec**.

Once the Report Spec instance is created, you can run the report with one of the following actions:

- By using the **Launch** button from Report Spec editor.
- By using the **Launch** hyperlink from Report Spec browser.
- By creating a Scheduled Report job on the Scheduler.

Executing a Report in background creates an instance of Report Results which will be retained on the MNFC server for configurable time; it will be available for repeated viewing.

The execution status of the Report executed in the background is possible to follow using the Report Results browser screen.

Interactively Executing Reports

To execute the Report interactively use the **Generate** action on the Report Spec editor screen. In addition, all **Drill Down** queries from Report Viewer are executed as interactive Reports.

During the execution of the Report in interactive mode the user is presented a **Report Progress** popup on the GUI, which in case of the success is transformed into Report Viewer window with results of the run. For interactive invocation the Report Results are not saved on the server and are lost when the Viewer window is closed.

Scheduled Reports

To schedule a report you should have a report spec with relative time coverage specified. The same instance of Report Specification can be used in multiple scheduled jobs.

To schedule a report, do the following:

-
- Step 1** Define the Report Spec that will be used for the scheduled report. Only a Report Spec with a **Relative** method of time coverage definition can be scheduled. A **Relative** time coverage defines the data start time by counting backwards from the scheduled launch time.
 - Step 2** Place the Report Spec on the scheduler by creating a **Scheduled Report Job**.
 - Step 3** From the **Scheduled Reports** window, select either **Hourly**, **Daily**, or **Weekly** for the job period.
 - Step 4** From the **Specify Job** window, specify the job's Report Spec ID.

- Step 5** Depending on job period, specify the minute of hour, or hour and minute of day.
- Step 6** Click **Submit**.

Report Results Browser

In MNFC the reports executed in the background store their Report Results instances as XML files in MNFC server's file system in subdirectories under **\$MNFC_DIR/Reports**. These Report Results instances are retained for the specified retention periods and are available for repeated viewing. The Report Results browser presents tabular views of available Report Results instances for the specified combination of Report Type and Report Spec. See [Figure 2-31](#).

Figure 2-31 Browse Report Results Window

The table contains info on following properties for each Report Results instance:

- **Source:** The name of the report's target table
- **Executed At:** The time when the report instance was executed
- **Start :** The time coverage for the report instance
- **End:** The time coverage for the report instance
- **Status:** The report execution or completion; if the status is **Ready** the report results are available for viewing
- **N.records:** The total number of data records contained in the results file
- **View:** Hyperlink that launches the results viewer in a separate browser window
- **Delete:** Immediately deletes the results instance file. Note that once the results data is deleted using this property, it can no be recovered.

Viewing Report Results in Tabular Format

The Report Results Viewer window is opened:

- By selecting the **View** property from the Report Results window
- After successful interactive invocation substituting the Report Progress bar after it reaches 100%.

The functionality of Report Results Viewer is same as in 1st tier NFC.

Viewing Report Results as Graphs

Similar to 1st tier NFC, graphing options are available from the value columns on **Report Results Viewer** window. Select either the bar graph icon or pie icon near the column titles. Additionally, linear graphs can be built for **Plot** values which are available for report results from reports containing the type Trending Analysis.

Saving, Exporting and Printing Reports

Similar to the 1st tier NFC GUI, the following actions are available as icons on the **Report Results Viewer** window:

- **Save:** Preserves a copy of the report results on the client host
- **Export:** Saves the report results data in CSV format on the client host
- **Printing:** Prints the report results

Retention of Report Results

The results of reports executed by MNFC in background are retained in the server's file system as XML files. These report results files are kept for predefined times and are periodically purged. If you have access to MNFC's file system, you can permanently preserve the report results by locating the XML and copying it elsewhere. If you do not have access to the MNFC file system, you can save the report results by using the GUI and saving or printing the report results on to the client's workstation.

The Report Results XML files in MNFC file system are periodically purged according to configurable Report Results Retention Rules which can be accessed and altered using GUI as described in this section.

Report Results Retention Rules Form

You can display the View Report Results Retention Rules window by navigating **Reports > View Report Results Retention Rules**.

MNFC Report Results Retention logic supports rules of two types:

- Defined per Report Type, matching by equality
- Defined per Report Spec name, matching using regular expressions

Report Results Retention characteristics include:

- The allowed retention span for report results is specified in days.
- Rules defined per Report Type can not be deleted but can be redefined to have a different retention span.
- Rules defined as matching Report Spec names can be added and deleted.
- There is no limit on how many rules' instances of this type can be defined on the server.
- If more than one rule applies to particular Report Results' file, then the rule with the shortest defined span will take an effect.

Working with MNFC Status and Control

From the Cisco Multi NetFlow Collector **Main** window, click the **Status** tab. From the Status window, you can select:

- Control Processes
- Stats
- Logs

Controlling MNFC Processes

You can stop and start the Process Watcher from the Network Concentrator Processes window. Navigate **Status >Control Processes**. The Network Concentrator Processes window contains fields that can be used to establish a TCP connection to the Process Watcher running on the MNFC server. This is desirable when running MNFC using a non-standard port. For information on the Process Watcher, see the [“Process Watcher” section on page 3-2](#).

In case the connection to Process Watcher can not be established, you can use a Unix shell to access the MNFC server to make sure the Process Watcher is up and running.

Monitoring MNFC Status

From the **Status >Stats** window, you can view the following:

- Metadata Transfer
- Transport by Collector
- Transport by Record
- Upload

Metadata Transfer

Metadata transfer is action performed by the MNFC server to acquire definitions of Aggregators and Aggregation Schemes from 1st tier NFCs. The following conditions apply:

- You need to define of a new Aggregator instance using the MNFC Configuration UI. From the Cisco Multi NetFlow Collector **Main** window, click the **Configuration** tab.
- You need to reestablishing communication to the 1st tier NFC server as detected by FTP-based Transport, if the communication was down more than the specified communication time-out. The default is 15 minutes.

Metadata Transfer status shown as **Failed** can be caused by multiple reasons. If all Aggregators for the same Collector are shown as **Failed**, either the 1st tier NFC server is down or unreachable. You should try to open a UI session to the 1st tier NFC server in question. If at least one Aggregator instance for a Collector succeeds on a metadata transfer, then the inter-tier communication is established and the possible reason for failure is a definitions mismatch.



Note

Definitions of Aggregators with same Aggregator ID across multiple Collectors should always match .

File Transport

MNFC uses ftp/sftp-based transport to move aggregated data from Collectors (1st tier NFCs) to MNFC for uploading into database. You can monitor the file transport status using the following:

- **Transport by Collector:** Reports the last time for a successful ftp-based transaction to the particular Collector server even if no NetFlow data was transported. Transactions with status showing **Retrieved 0 files** reflect the periodic polling of the Collector hosts done by the File Transport.

Records shown on **Transport by Collector** contain estimated time difference between the MNFC and 1st tier NFC servers, labeled as **T.diff, sec**. A positive value indicates that the clock on the MNFC server is likely to be ahead of the 1st tier Collector. A negative value indicates that the 1st tier NFC is ahead of the MNFC server.

- **Transport by Record:** Reports the last successful delivery time of aggregated data for each particular Aggregator from every Collector.

Data Upload Status

The **Upload** window can be used to monitor the status of aggregated NetFlow files transported from 1st tier Collectors for uploading into database primary tables.

This window retains logical entities originated as individual aggregated files identified by the following attributes:

- **Record Type:** The Aggregator ID and name of the primary table.
- **Timestamp:** Taken from the aggregated file name and assigned as a timestamp for every individual record in database.
- **Device Name:** The name, address, or 1st tier Device Group ID.

For every such entry the following attributes are shown:

- Count: number of aggregated records
- Status

The following status values can be shown:

- **Submitted:** The file is converted into HPL format and the StorageManager was notified that the `/tmp/nfc` directory contains an HPL file (with records converted from original aggregated file) ready for uploading.
- **Uploaded:** The HPL file was uploaded into database.
- **An error message:** Can be shown here if the upload failed or in some cases when a already uploaded table fragment failed to attach to the primary storage.

Monitoring Log Files

MNFC log files can be reviewed using this window and include the following:

- Log files generated by MNFC processes. See the [“The Main MNFC Components” section on page 2-1](#).
- Log files named `<record-type>.log`, for example `CallRecord.log`, containing output from HPL sessions uploading records into primary database tables.



CHAPTER 3

Multi NetFlow Collector Advanced Features

This chapter describes the Cisco Multi NetFlow Collector (MNFC) advanced features and includes the following sections:

- [Storage Manager, page 3-1](#)
- [Process Watcher, page 3-2](#)
- [Transport, page 3-3](#)

Storage Manager

The MNFC Storage Manager is a resident process executed with the userid **informix**. The Storage Manager is responsible for maintaining the MNFC's persistent storage in the Informix database **nfc_db**. The Storage Manager should run 24 hours a day and has no UI or command interface.

The following functionality is allocated to the Storage Manager:

- Managing the fragmented storage: All tables with NetFlow data – both primary (for Aggregators) and derived (for Summarizations and Correlators) – are managed as circular buffers comprised of 24, 48, 96, or 192 fragments. The number of fragments is determined by the disk fragmentation done by the **configuredb.sh** script.
- Uploading of HPL files from the **/tmp/nfc** directory into fragmented database tables. In most cases the HPL files are uploaded into the primary storage, that is Aggregator NetFlow data is transported from 1st tier Collectors. The actual uploading is done by Informix HPL utility through the MNFC's script spawned by Storage Manager. Details on the uploading sequence are given in Appendix C, Aggregated File Transfer and Uploading Sequence.
- Calculating summarized and correlated information into fragmented database tables (so called derived storage such as for Summarizations and Correlators).

The Storage Manager maintains information in number of MNFC's own database tables, more information on which is given in Appendix C.

Process Watcher

The Process Watcher provides high availability to the MNFC system. It is responsible for starting, stopping, and restarting MNFC processes. It monitors MNFC processes and attempts to restart a process, up to the configured number of restarts, if the process dies unexpectedly with a non-zero return status.

The **mnfc** script starts the Process Watcher the first time a managed process is started. Once started, the Process Watcher remains running until **mnfc shutdown** is executed.

[Table 3-1](#) displays the **mnfc** command line arguments.

Table 3-1 *mnfc Command Line Arguments*

Argument	Description
start all	Starts all managed processes marked for autostart.
stop all	Stops all managed processes marked for autostart.
start <managed process id>	Starts the managed process with the corresponding ID attribute. The Process Watcher, and consequently all autostart processes, are started if necessary.
stop <managed process id>	Stops the managed process with the corresponding id attribute.
status	Lists all managed processes, whether or not they are running, and the process ID if one is stored in a file designated by the <pid-file> setting of a managed process.
shutdown	Gracefully stops all managed processes including the Process Watcher.

The Process Watcher can also be started with the **escomnfc** script. If you configure MNFC during installation to start at boot time, then system-dependent steps are taken to invoke the **escomnfc** script during system initialization and shutdown. This script invokes **mnfc start all** at system initialization and **mnfc shutdown** at shutdown.

Process Watcher Configuration

The configuration for the Process Watcher is stored in the file **MNFC_DIR/config/mnfcpw.xml**. XML element text values in this file may contain **\${MNFC_DIR}** and the Process Watcher will substitute that string with the **MNFC_DIR** environment variable. Each managed process must have an ID attribute and the child elements as described [Table 3-2](#).

Table 3-2 Configuration Elements

Child Element	Description	Required
commandline	What will be executed when the Process Watcher attempts to start the managed process.	Yes
stop-commandline	If defined, the Process Watcher will execute this command line when it attempts to stop the managed process. If omitted, the Process Watcher uses a system-dependent means of stopping the process.	No
autostart	A value of true tells the Process Watcher to automatically start the managed process when the Process Watcher starts or when the start all arguments are used with the nfc collector script. Default is false.	No
restart	A value of true tells the Process Watcher to monitor and restart the managed process if it dies unexpectedly with a non-zero exit status. Default is false.	No
restart-attempts	The number of times the Process Watcher should attempt to restart a process.	No
pid-file	If the managed process generates a file containing the process ID of the managed process, then the Process Watcher can include that information in its status output. Set this value to the path of that file.	No

For example, the MNFC concentrator process is configured with the following XML:

```
<managed-process id="concentrator">
  <commandline>${MNFC_DIR}/bin/startmnfc.sh</commandline>
  <autostart>true</autostart>
  <restart>true</restart>
  <restart-attempts>3</restart-attempts>
  <pid-file>${MNFC_DIR}/logs/mnfc.pid</pid-file>
</managed-process>
```

Transport

Transport pulls input data from Cisco NetFlow Collector servers, driven by MNFC aggregators. Refer to the [“Aggregators” section on page 2-9](#).

The following is a sample of the configuration file **\$MNFC_DIR/config/transport.properties**:

```
# transport.properties
#
# Copyright (c) 2005-2006 by Cisco Systems, Inc.
# All rights reserved.
#
# minutes of backlog data to retrieve from NFCs.
minsbacklog = 0
# Java class implementing IFTPSession
nfc.ftpClient = com.cisco.mnfc.transport.FtpClientSession
```

The value **minsbacklog** sets how many minutes worth of NFC data will be collected and processed before MNFC starts. The default value **0** results in no backlog data being processed, and MNFC only imports new NFC data files.

Note that you must add a line in the **transport.properties** file if the NFC install directory **/opt/CSCOnfc** is symbolically linked to an alternate location.

For example, if you have the following on the NFC server:

```
/opt/CSCOnfc ---> /localdata
```

then you need to add the following to the MNFC file **/opt/CSCOmncf/config/transport.properties**:

```
nfc.remotebasedir.<NFC IP address>=/localdata
```




CHAPTER 4

Multi NetFlow Collector Logging

Multi NetFlow Collector, Release 6.0 uses Log4J from the Apache Foundation to perform logging functions. In general, all logs can be tuned to provide the level and amount of logging desired.

Configuration

All logging configurations come from files stored in the **MNFC_DIR/config** directory. [Table 4-1](#) displays the log configuration file for each component of MNFC 6.0.

Table 4-1 Log Configuration Components

Component	Log Configuration File
MNFC Concentrator	MNFC_DIR/config/mnfc-log4j.properties
Process Watcher	MNFC_DIR/config/mnfcpw-log4j.properties
VPN MIB Collector	MNFC_DIR/config/vpnmibcltr-log4j.properties
CLI Collector	MNFC_DIR/config/clic-log4j.properties
Report Daemon	MNFC_DIR/config/mnfcrd-log4j.properties
Web-based UI	MNFC_DIR/config/mnfcweb-log4j.properties

Two settings stored in these configuration files are log filename and logging level. To customize the log filename, change the line with **log4j.appender...File=<default filename>** in the appropriate configuration file.

For example, to change the path to the MNFC Concentrator log, file you would change:

```
log4j.appender.mnfcLog.File=${MNFC_DIR}/logs/mnfc${MNFC_PROG}.log
```

to something like:

```
log4j.appender.mnfcLog.File=/tmp/mnfc.log
```

To customize the **logging level**, change the line with **log4j.logger...=INFO**, ... in the appropriate configuration file. Valid levels are FATAL, ERROR, WARN, and INFO.

For example, to change the logging level of MNFC Concentrator from INFO to ERROR you would change:

log4j.logger.com.cisco.mnfc.concentrator=INFO, nfcLog

to:

log4j.logger.com.cisco.mnfc.concentrator=ERROR, nfcLog

See <http://jakarta.apache.org/log4j> for more details on how these configuration files work.



APPENDIX **A**

Troubleshooting the Multi NetFlow Collector

This appendix provides helpful information and procedures in case you encounter problems while using Cisco Multi NetFlow Collector (MNFC).

Solving Multi NetFlow Collector Problems

This section discusses some basic problems that you might encounter while attempting to run Multi NetFlow Collector.

Symptom When creating an aggregator, MNFC displays a collision error.

Recommended Action Do the following:

1. Verify that the aggregation table has been dropped.
 2. Echo **select sum(size) size, tablename from sysmaster:sysexents where chunk=10 group by tablename order by size desc** in the `/opt/informix/bin/dbaccess nfc_db` file. If the table has not been dropped, manually drop the table.
 3. Create the aggregator using the same ID. Click **Discard**, then recreate the aggregator using the same ID.
-

Symptom NFC Data Files are not being imported from the NFC server.

Possible Cause Changes were made to the NetFlow Collector (NFC) or NFC ran out of disk space.

Recommended Action Restart **cscmnfc**. If the problem persists, remove files from the `/opt/CSCOnfc/Data` file.

Symptom A backlog of NFC data files has occurred on MNFC. Files are not being imported into the MNFC database.

Possible Cause Check the `/opt/mnfc/nfcd.db.log` file for a **No free disk space** error.

Recommended Action If after several hours MNFC does not recover, do the following:

- a. Drop tables
 - b. Reduce traffic
 - c. Move to a larger disk
-

Symptom MNFC displays the message **No source data available** when generating report .

Possible Cause Data no longer exists in the table, data is not in database yet due to latency, or the wrong time span is specified in the report spec.

Recommended Action Query the table to see what data is currently available. See the [“Working with MNFC Report” section on page 2-18](#) for instructions on how reports are generated.

Symptom Unable to create TopN summary aggregator and no error message is displayed.

Recommended Action Disable the pop-up blocker in your web browser.

Symptom Memory errors related to reporting in the `mnfcrd.log` or `mnfcweb.log` file.

Possible Cause Report daemon (web process) maximum memory allowance is set too low for the time coverage and/or number of keys requested for the report.

Recommended Action Increase the report daemon maximum memory using the **RD_MEM_MAX** [**WEB_MEM_MAX**] setting in the `/opt/CSCOmnc/config/mnfcmem` file and decrease the time coverage and/or number of keys requested in the report. See the *Cisco Multi NetFlow Collector User Guide*.

Symptom Egress PE column shows empty values in PE-PE or CE-CE reports.

Possible Cause NFCs from which MNFC collects data failed to resolve egress PEs.

Recommended Action Log into each NFC server and verify if the `/opt/CSCOnfc/config/peList.conf` file contains the whole list of all PEs in the network. If not, make sure the list is complete by adding any missing PE addresses. Then restart NFC on the server.

Symptom After starting the `vpnmibcltr` process, the data source `ifindex_to_vrfname` is not created automatically.

Possible Cause The `/opt/CSCOmnc/config/vpnmibcltr.xml` file does not specify the list of PEs from which it collects data.

Recommended Action Make sure the `<pe-list>` element contains a row like `<pe id=x.x.x.x>` for every PE in the network. Then restart the `vpnmibcltr` process.

Symptom Ingress CE or customer name columns shows empty values in CE-CE reports.

Possible Cause NFCs from which MNFC collects data fails to resolve the ingress CE or customer name fields.

Recommended Action Log into each NFC managed by this MNFC and check if the `/opt/CSCOnfc/config/vpn.conf` file contains the correct configuration. Refer to the [“Defining a CE-CE Report” section on page 2-29](#) for the correct format of this configuration file. Make sure that there is a row corresponding to each VPN interface of every PE that exports NetFlow data to that NFC server.

Symptom The Storage Manager Process is down.

Possible Cause The Informix IDS is not running. The Informix Utility tool `onstat` reports **no shared memory is initialized for server nfc**.

Recommended Action Review the information and messages in the `/opt/informix/online.log` file that corresponds with the time of the malfunction. Change the SHMBASE, and if the problem persists recheck the information and messages in the `/opt/informix/online.log` file for different messages. Contact support.

Symptom No aggregated data is uploaded and the HPL consistently exits with code **255** without connecting to **database onpload** or **database nfc_db**.

Possible Cause HPL fails to connect to database server.

Recommended Action Re-start the server.

Symptom Significant degradation in performance of database activities and possible loss of summarized data and/or report results.

Possible Cause Informix IDS server enters **Long Transaction** as the transaction gets aborted due to an insufficient number of available transaction buffers. In addition, `onstat` shows **LONGTX** in status position in title line.

Recommended Action Do the following:

1. Shut down MNFC.
 2. Shut down IDS. Note that before the Informix database is shutdown or restarted, the Multi NetFlow Collector application must be shutdown. For information on operating the Informix server, refer to IDS 9.40 documentation by IBM.
 3. Increase the number of **LOGFILES** in the `/opt/informix/etc/onconfig.nfc` file. The default value is 18. Set the number of **LOGFILES** to a value such as 24, 30, or 26.
 4. Restart MNFC.
-

Symptom The Summarization table with **Top N Values** storage method contains only a subset of the data loaded or no data is uploaded at all.

Possible Cause Since the last Metadata Transfer was completed on the 1st Tier NFC Collector, the configuration on the 1st tier has been changed so that the Aggregator Sort Output is not selected.

Recommended Action Make sure that the **Sort Output** checkbox is selected in the **Modify Aggregator** window for aggregators on all NFCs supplying records to the MNFC Aggregator to be used by the Summarization in question.



APPENDIX **B**

Database Fragmentation Profiles

This appendix provides information on database fragmentation profiles.

- Database fragmentation (sizes and quantity of allocated dbspaces) is determined by amount of initially available raw disk space and is labeled **fragmentation profile**.
- In MNFC Release 6, unfragmented storage is not supported.
- For all profiles, the size of root database space is allocated as one fifth of the total raw disk space.
- The remaining four fifths of raw disk space is fragmented into equal dbspaces named **nfc_data**. See [Table 1](#).

Table 1 *Database Fragmentation Profiles*

Profile Name	Available Raw Disk Space	Number of nfc_data dbspaces
SMALL	50 GB to 100 GB	24
MEDIUM	100GB to 200 GB	48
LARGE	200 GB to 500 GB	96
HUGE	500 GB or more	192



Note

The database partitions' fragmentation is done in post-installation phase by the script **configuredb.sh** but can be re-done again at any convenient time provided that the MNFC is shut down and the configuration is reset.



APPENDIX **C**

MNFC Aggregated File Transfer and Internal Tables

This appendix provides information on the following:

- [Aggregated File Transfer and Uploading Sequence](#)
- [MNFC Internal Tables](#)

Aggregated File Transfer and Uploading Sequence

The following is the aggregated file transfer and uploading sequence:

1. Aggregated file is exported by the Collector (1st tier NFC).
2. Aggregated file is transported over FTP/SFTP to MNFC by the File Transport Manager (in Concentrator process).
3. Aggregated file is converted to HPL file and appended to HPL bundle (in the **/tmp/nfc** directory) by the Upload Manager (in Concentrator process).
4. HPL bundle is submitted to the Storage Manager, the status record is created in the **upload_info** file with status value as **Submitted**.
5. Storage Manager spawns HPL, which uploads the bundle into table named **<rec-type>_a**, after HPL returns the status is updated to **Uploaded**.
6. Storage Manager attaches table named **<rec-type>_a** to table named **<rec-type>** (primary storage); if this fails the status is updated to error message with failure code.

MNFC Internal Tables

MNFC maintains number of internal tables for its own use in the Informix database named **nfc_db**. See [Table C-1](#)

Table C-1 *MNFC Database Tables*

Table Name(s)	Purpose
Aggregator_metadata	Metadata on application (NetFlow data) table
Field_metadata	
Aggregator_hostaddr	Allocation of Aggregators to 1st tier Collectors
Device_store	Encoding of device names to internal integer values
Retention_info	Details on retention of application tables (for Aggregators, Summarizations, and Correlators)
Storage_directory	Timestamps for latest data available in application tables
Upload_info	Stats on upload of application data into database
Qtriggers and other named q*	Repository for Quartz (the scheduler)



INDEX

A

aggregated file transfer sequence [C-1](#)
audience [vii](#)

B

basic problems
 recommended actions [A-1](#)

C

Cisco Multi NetFlow Collector
 functions [1-5](#)
 overview illustration [1-5](#)
Cisco NetFlow Collector
 Device and IOS Release Support [1-2](#)
 functions [1-4, 1-5](#)
 overview illustration [1-4](#)
command conventions [viii](#)
compatibility
 IOS software [1-2](#)
configuration [2-6](#)
conventions, command [viii](#)

D

database fragmentation profiles [B-1](#)
data export
 compatibility matrix [1-2](#)
 format [1-3](#)
 mechanism [1-2](#)
documentation

obtaining [ix](#)

F

flow cache [1-2](#)
flows
 defined [1-1](#)

I

internal tables [C-2](#)
IP address
 for configuration [1-4](#)
IP packets [1-1](#)

J

JRE 1.5 [2-3](#)

L

Log4J [4-1](#)
logging [4-1](#)
login screen [2-4](#)

M

MNFC [2-3](#)
 configuration [2-6](#)
 intrnal tables [C-2](#)
 login screen [2-4](#)
 overview [1-1, 3-1](#)
 reports [2-18](#)

Multi NetFlow Collector

- configuration [2-6](#)
- functions [1-5](#)
- overview [1-1, 3-1](#)
- overview illustration [1-5](#)
- starting [2-3](#)

N

NetFlow Collector

- Device and IOS Release Support [1-2](#)
- functions [1-4, 1-5](#)
- overview illustration [1-4](#)

NetFlow data export

- hardware supported [1-2](#)

NetFlow services

- device and IOS release support [1-2](#)
- overview [1-1](#)

P

packets

- IP [1-1](#)

Process Watcher [3-2](#)

Rreports [2-18](#)

S

SCTP

- port number configuration [1-4](#)
- security guidelines [ix](#)
- starting [2-3](#)
- starting MNFC [2-3](#)
- Storage Manager [3-1](#)
- support

- obtaining [ix](#)

T

traffic flows

- description [1-1](#)

traffic statistics

- information types [1-3](#)

Transport [3-3](#)troubleshooting [A-1](#)

U

UDP

- exporting NetFlow data to port [1-4](#)
- port number configuration [1-4](#)

uploading sequence [C-1](#)

user interface

- configuration [2-6](#)
- navigation [2-5](#)
- Reports [2-18](#)

V

Version 1 NetFlow export datagram

- description [1-3](#)

Version 5 NetFlow export datagram

- description [1-3](#)

Version 7 NetFlow export datagram

- description [1-3](#)

Version 8 NetFlow export datagram

- description [1-3](#)

Version 9 NetFlow export datagram

- description [1-3](#)