



Info Mediators

This chapter introduces Cisco Info Mediators, their key features, and how they are used. It also describes the types of Cisco Info Mediators, their architecture and components, and how to run them. It contains the following sections:

- [Introduction to Cisco Info Mediators](#)
- [Types of Cisco Info Mediators, page 5-2](#)
- [Cisco Info Mediator Components, page 5-5](#)
- [Cisco Info Mediator Architecture, page 5-8](#)
- [Cisco Info Mediator Features, page 5-9](#)
- [Using a Specific Cisco Info Mediator, page 5-11.](#)

For reference information about Cisco Info Mediators, including descriptions of common properties and command line options, rules file syntax, troubleshooting hints, and descriptions of error and information messages, see [Appendix E, “Cisco Info Mediator Reference.”](#)

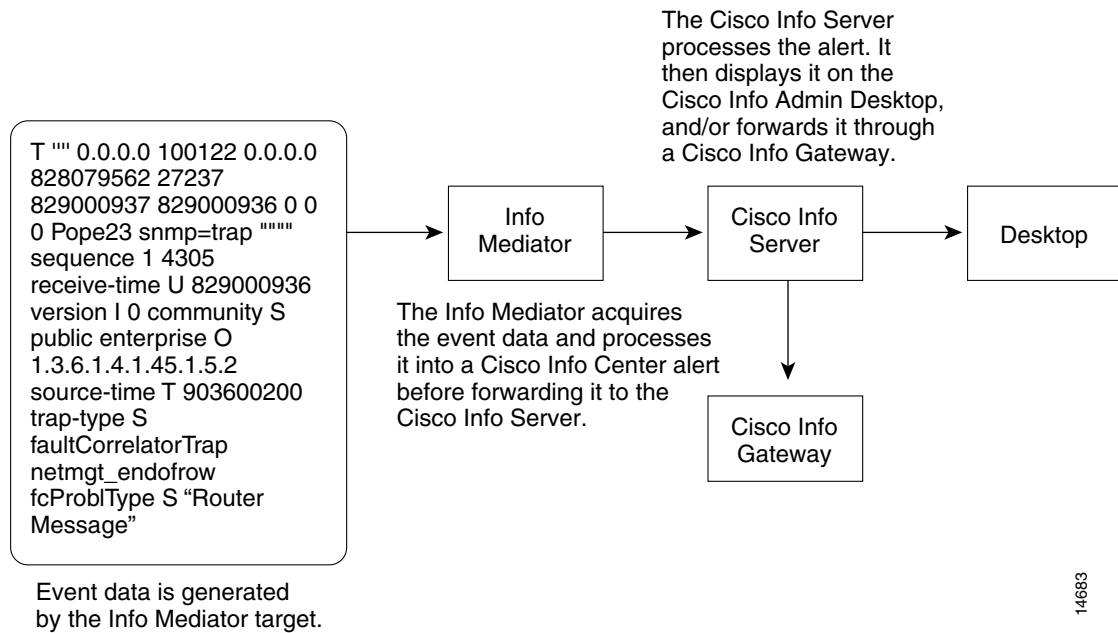
For more information about specific Cisco Info Mediators, see the individual guides available for each Cisco Info Mediator on the Cisco Systems Support Site.

Introduction to Cisco Info Mediators

A Cisco Info Mediator connects to an event source, detects and acquires event data, and forwards it to the Cisco Info Server to form alerts. The Cisco Info Mediator uses logic within a rules file to manipulate the event elements before converting them into fields of an alert in the Cisco Info Server **alerts.status** table.

[Figure 5-1](#) shows how Cisco Info Mediators fit into the Cisco Info Center architecture.

Figure 5-1 Event Processing in Cisco Info Center



Cisco Info Mediators can acquire data from any stable data source. These sources are described in [Types of Cisco Info Mediators, page 5-2](#).

Types of Cisco Info Mediators

Each Cisco Info Mediator is uniquely designed to acquire event data from a specific source. However, Cisco Info Mediators can be categorized based on how they acquire events. For example, the Cisco Info Mediator for Informix obtains event data from a database table, and is therefore classed as a database Cisco Info Mediator. The types of Cisco Info Mediators are:

- Device
- Database
- Log File
- API
- Miscellaneous.

These types of Cisco Info Mediators are described in the following sections.

The Cisco Info Mediator type is determined by the method in which the Cisco Info Mediator detects events. For example, the Cisco Info Mediator for Agile ATM Switch Management detects events produced by a device (an ATM switch), but it acquires events from a log file, not directly from the switch. Therefore, this Cisco Info Mediator is classed as a log file Cisco Info Mediator and not a device Cisco Info Mediator.

Introduction to Device Cisco Info Mediators

A device Cisco Info Mediator acquires events by connecting to a remote device, such as an ATM switch. Device Cisco Info Mediators run on a separate machine to the one they are probing and connect to the target machine through a network link, modem, or physical cable. Often, device Cisco Info Mediators can use more than one method to connect to the target machine—for example, the Cisco Info Mediator for Ericsson ACP1000 can use Telnet, tip, or rsh protocols to connect to the target switch.

Once connected to the target machine, the Cisco Info Mediator detects events and forwards them to the Cisco Info Server. Some device Cisco Info Mediators are passive, waiting to detect an event before forwarding it to the Cisco Info Server—for example, the Cisco Info Mediator for Telematics Switch. Other device Cisco Info Mediators are more active, issuing commands to the target device in order to acquire events—for example, the Cisco Info Mediator for Ericsson ACP1000.

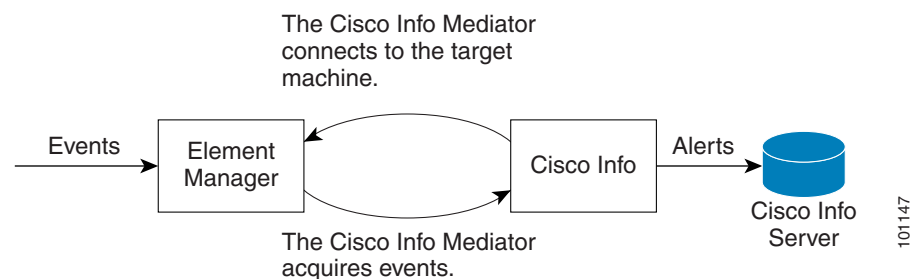


Note

Some Cisco Info Mediators, including the Cisco Info Mediator for KBU Fivemere and Cisco Info Mediator for Polycenter Watchdog, can acquire events from either log files or devices.

Figure 5-2 shows a typical device Cisco Info Mediator environment.

Figure 5-2 Typical Device Cisco Info Mediator Environment

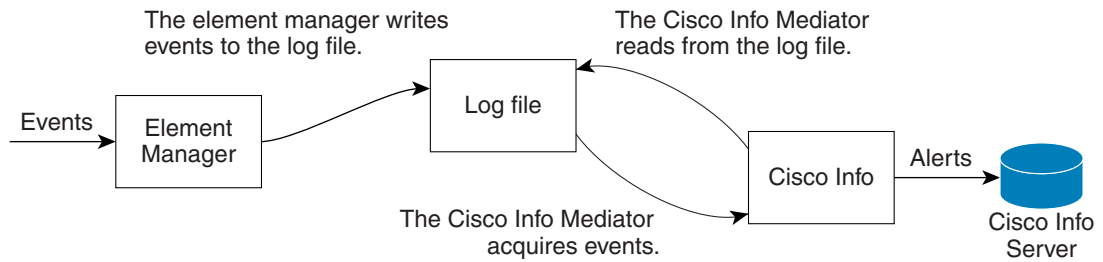


Introduction to Log File Cisco Info Mediators

A log file Cisco Info Mediator acquires events by reading a log file created by the target system.

Most log files Cisco Info Mediators run on the machine where the log file resides; this is not necessarily the same machine as the target system. The target system appends events to the log file. Periodically, the Cisco Info Mediator opens the log file, acquires and processes the events stored in it, and forwards the relevant events to the Cisco Info Server as alerts. You can configure how often the Cisco Info Mediator checks the log file for new events and how events are processed.

Figure 5-3 shows a typical log file Cisco Info Mediator environment.

Figure 5-3 Typical Log File Cisco Info Mediator Environment

101149

Introduction to Database Cisco Info Mediators

A database Cisco Info Mediator acquires events from a single database table—the source table. Depending on the configuration, any change (insert, update, or delete) to any row of the source table can produce an event.

When a database Cisco Info Mediator is started, it creates a temporary logging table and adds a trigger to the source table. When a change is made to the source table, the trigger forwards the event to the logging table. Periodically, the events stored in the logging table are forwarded to the Cisco Info Server as alerts and the contents of the logging table are discarded. You can configure how often the Cisco Info Mediator checks the logging table for new events.

Existing triggers on the source table may be overwritten when the Cisco Info Mediator is installed.

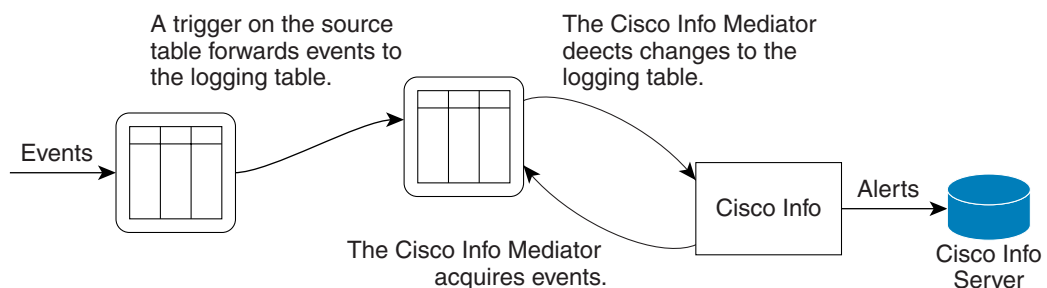
Database Cisco Info Mediators treat each row of the source table as a single entity. Even if only one field of a row in the source table changes, all of the fields of that row are forwarded to the logging table and from there to the Cisco Info Server. If a row in the source table is deleted, the Cisco Info Mediator forwards the contents of the row before it was deleted. If a row in the source table is inserted or updated, the Cisco Info Mediator forwards the contents of the row after the insert or update.



Note

The Cisco Info Mediator for StrataView only detects inserts into the source table.

Figure 5-4 shows a typical database Cisco Info Mediator environment.

Figure 5-4 Typical Database Cisco Info Mediator Environment

101148

Introduction to API Cisco Info Mediators

An API Cisco Info Mediator acquires events through the API of another application.

API Cisco Info Mediators use specially designed libraries to acquire events from another application or management system. These libraries contain functions that connect to the target system and manage the retrieval of events. The API Cisco Info Mediators call these functions which connect to the target system and return any events to the Cisco Info Mediator. The Cisco Info Mediator processes these events and forwards them to the Cisco Info Server as alerts.

**Note**

The Cisco Info Mediator for Aprisma Spectrum Alarm Notifier reads events from a log file, but first it issues a command which generates the events. It is therefore classified as an API Cisco Info Mediator. For more information about how log file Cisco Info Mediators work, see [Introduction to Log File Cisco Info Mediators, page 5-3](#).

Introduction to Miscellaneous Cisco Info Mediators

All of the miscellaneous Cisco Info Mediators have characteristics that differentiate them from the other types of Cisco Info Mediators and from each other. Each of them has been designed to carry out a specialized task that requires them to work in a unique way.

For example, the Email Cisco Info Mediator connects to the server, retrieves emails, processes them, deletes them, and then disconnects. This is useful on a workstation not having sufficient resources to permit an SMTP server and associated local mail delivery system to be kept resident and continuously running. Similarly, it may be expensive, or even impossible, to keep a personal computer connected to an IP-style network for a long period of time.

Another example of a Cisco Info Mediator in the miscellaneous category is the Ping Info Mediator. It is used for general purpose applications on UNIX platforms and does not require any special hardware. You can use the Ping Info Mediator to monitor any device that supports the ICMP protocol, such as switches, routers, PCs, and UNIX hosts.

Cisco Info Mediator Components

A Cisco Info Mediator has the following primary components:

- an executable file
- a properties file
- a rules file.

These components are described in the following sections.

Executable File

The executable file is the core of a Cisco Info Mediator. It connects to the event source, acquires and processes events, and forwards the events to the Cisco Info Server as alerts.

Cisco Info Mediator executable files are stored in the directory `$OMNIHOME/probes/<arch>`, where `<arch>` is the name of the architecture. For example, the executable file for the Ping Info Mediator running on HP-UX 11.00 is:

`$OMNIHOME/probes/hpux11/ncp_p_ping`

When the Cisco Info Mediator is started, it obtains information on how to configure its environment from the properties and rules files, described in the next sections. The Cisco Info Mediator uses this configuration information to customize the data it forwards to the Cisco Info Server.



Note

When you run the Cisco Info Center configuration utility, `ncp_config`, Cisco Info Mediators are configured to run under Process Control. Process Control is described in [Chapter 7, “Process Control.”](#)

Properties File

Cisco Info Mediator properties define the environment in which the Cisco Info Mediator runs. For example, the **Server** property specifies the Cisco Info Server to which the Cisco Info Mediator forwards alerts. Cisco Info Mediator properties are stored in a properties file.

By default, the properties file is stored in the same directory as the executable file, with the extension `.props`. For example, the properties file for the Ping Info Mediator running on HP-UX 11.00 is:

`$OMNIHOME/probes/hpux11/ping.props`

Properties files are formed of name-value pairs separated by a colon. For example:

Server : “INFOSERVER”

In this name-value pair, **Server** is the name of the property and **INFOSERVER** is the value to which the property is set. String values must be enclosed in quotes; other values do not require quotes.

Cisco Info Mediator Property Types

Properties can be divided into two categories: common properties and Cisco Info Mediator-specific properties.

For example, the **Server** property is a common property, because every Cisco Info Mediator needs to know to which Cisco Info Server to send alerts. Common properties are described in [Common Cisco Info Mediator Properties and Command Line Options, page E-1](#).

Cisco Info Mediator-specific properties vary by Cisco Info Mediator. Some Cisco Info Mediators do not have any specific properties, but most have additional properties relating to the environment in which they run. For example, the Ping Info Mediator has a **Pingfile** property which specifies the name of a file containing a list of the machines to be pinged. Cisco Info Mediator-specific properties are described in the individual Info Mediator guides, available on the Cisco Systems Support Site.

You can change the values of Cisco Info Mediator properties by editing the properties file with a text editor or the Properties Editor, described in [Properties Editor, page 3-35](#).

Properties and Command Line Options

There is a command line option corresponding to each Cisco Info Mediator property. For example, the Server property can be set in a properties file as:

Server : “INFOSERVER”

It can also be set on the command line using the option:

probename -server STWO

The command line option overrides the property when both are set. In the preceding example, where the property sets **Server to INFOSERVER** and the command line option sets **Server to STWO**, the value **STWO** is used for the Cisco Info Server name. For more information on using command line options to override properties, see [How Properties and Command Line Options Are Processed at Startup](#), page E-2.

The Rules File

The rules file defines how the Cisco Info Mediator should process event data to create a meaningful Cisco Info Center alert. In addition, the rules file creates an identifier for each alert—the **Identifier** field, described in [Creating a Unique Identifier](#), page 5-9. The **Identifier** field is used to uniquely identify the problem source, so repeated events can be deduplicated.

By default, the rules file is stored in the same directory as the executable file, with the extension.rules. For example, the rules file for the HP-UX 11.00 version of the Ping Info Mediator is:

\$OMNIHOME/probes/hpux11/ping.rules

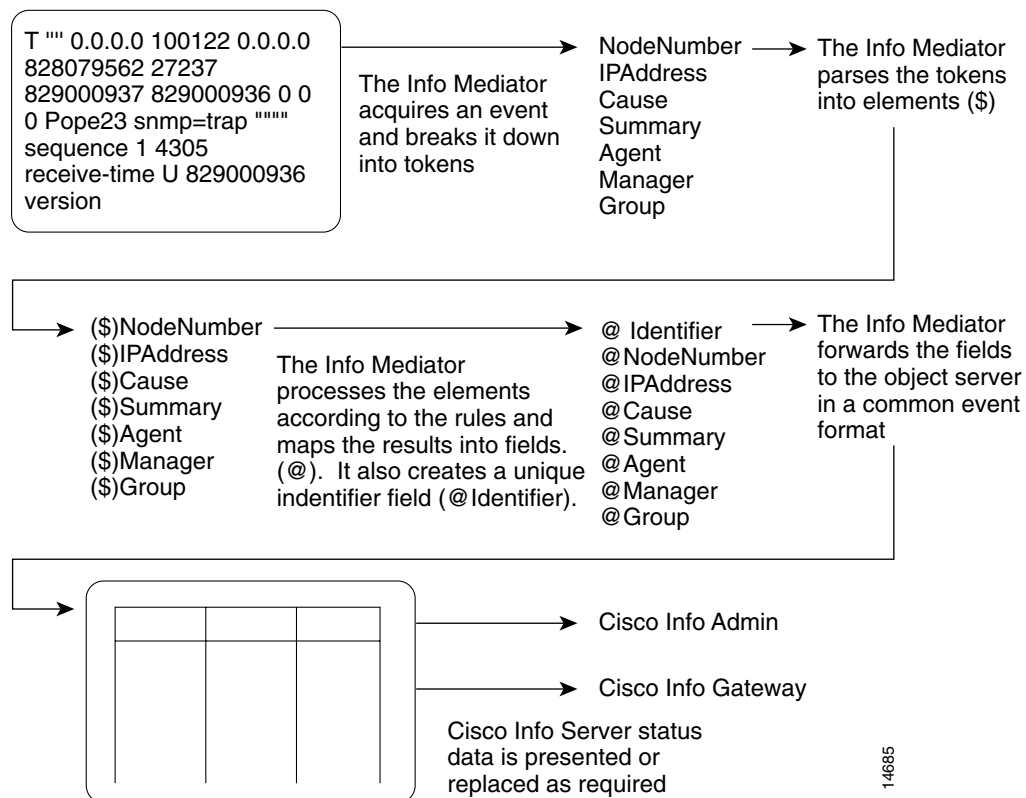
See [Cisco Info Mediator Rules File Processing](#), page E-6 for more detailed information about rules files and how to modify them.

For rules file changes to take effect, the Cisco Info Mediator must re-read the rules file. You can force the Cisco Info Mediator to re-read the rules file by issuing a **kill -HUP pid** command on the Cisco Info Mediator process ID (PID). This is preferable to restarting the Cisco Info Mediator, because the Cisco Info Mediator will not lose events.

Cisco Info Mediator Architecture

The function of a Cisco Info Mediator is to acquire information from an event source and forward it to the Cisco Info Server. [Figure 5-5](#) shows how Cisco Info Mediators process the event data acquired from the event source using rules

Figure 5-5 Event Mapping Using Rules



The raw event data a Cisco Info Mediator acquires cannot be sent directly to the Cisco Info Server. The Cisco Info Mediator breaks the event data down into tokens. Each token represents a piece of event data.

The Cisco Info Mediator then parses these tokens into elements and processes the elements according to the rules in the rules file.

Elements are identified in the rules file by the \$ symbol. For example, **\$Node** is an element containing the node name of the event source. Elements are used to assign values to Cisco Info Server fields, indicated by the @ symbol. The field values contain the event details in a form understood by the Cisco Info Server. Fields make up the alerts which are forwarded to the Cisco Info Server, where they are stored and managed in the **alerts.status** table.

The **@Identifier** field is also generated by the rules file, as described in the next section.

For more information about manipulating fields and elements, see [Cisco Info Mediator Rules File Processing](#), page E-6.

Creating a Unique Identifier

The identifier field (**@Identifier**) is used to uniquely identify a problem source. Like other Cisco Info Server fields, the identifier field is constructed from the tokens the Cisco Info Mediator acquires from the event stream according to the rules in the rules file.

The identifier field allows the Cisco Info Server to correlate alerts so duplicate alerts only appear in the Event List once. Rather than inserting a new alert, the existing alert is updated. The updates are configurable, as described in [Introduction to Deduplication, page 2-2](#). For example, typically the tally field (**@Tally**) is incremented to keep track of the number of times the event occurred.

It is essential the identifier appropriately identifies repeated events. The following identifier is not specific enough, resulting in the deduplication of events that are not actually repeated:

@Identifier=@Manager+@Node

In this example, any events with the same manager and node are treated as duplicates.

However, if the identifier field is too specific (for example, if it contains a time value) the Cisco Info Server is not able to correlate and deduplicate repeated events.

For more information on deduplication, see [Introduction to Deduplication, page 2-2](#).

Deduplication with Cisco Info Mediators

Deduplication is managed by the Cisco Info Server, but can be configured in the Cisco Info Mediator rules file. This enables you to set deduplication rules on a per-event basis. You can specify which fields of an alert are to be updated if the alert is deduplicated using the update function, described in [Update on Deduplication Function, page E-22](#).

Cisco Info Mediator Features

This section describes some of the key features of Cisco Info Mediator operation.

Store and Forward Mode

Cisco Info Mediators can continue to run if the Cisco Info Server is down. When the Cisco Info Mediator detects the Cisco Info Server is not present (usually because it is unable to write an alert to the Cisco Info Server), it switches to store mode. In this mode, the Cisco Info Mediator writes all of the messages it would normally send to the Cisco Info Server to a file named:

\$OMNIHOME/var/<probename>.<destserver>.store

In this file name, *<probename>* is the name of the Cisco Info Mediator and *<destserver>* is the name of the Cisco Info Server to which the Cisco Info Mediator is attempting to send alerts.

When the Cisco Info Mediator detects the Cisco Info Server is back online, it switches to forward mode and sends the alert information held in the **.store** file to the Cisco Info Server. Once all of the alerts in the **.store** file have been forwarded, the Cisco Info Mediator returns to normal operation.

Store and forward functionality is enabled by default, but can be disabled by setting the **StoreAndForward** property to **0** (False) in the properties file.

**Note**

Do not run Cisco Info Mediators in store and forward mode if the Cisco Info Server is taken offline for changes in its database and table definitions; instead, shut down the Cisco Info Mediators. Any alerts generated during this time will be discarded by the Cisco Info Server, because the new database and table definitions will not match those of the incoming alerts.

The **RetryConnectionCount**, **RetryConnectionTimeout**, and **MaxSAFFileSize** properties also control the operation of store and forward mode. See [Common Cisco Info Mediator Properties and Command Line Options, page E-1](#) for more information about these properties.

Automatic Store and Forward

By default, store and forward mode is active only after a connection to the Cisco Info Server has been established, used, and then lost. If the Cisco Info Server is not running when the Cisco Info Mediator starts, store and forward mode is not triggered and the Cisco Info Mediator terminates.

However, if you set the Cisco Info Mediator to run in automatic store and forward mode, it will go straight into store mode if the Cisco Info Server is not running, as long as the Cisco Info Mediator has been connected to the Cisco Info Server at least once before. Enable automatic store and forward mode by setting the **-autosaf** command line option or the **AutoSAF** property.

Raw Capture Mode

Raw capture mode enables you to save the complete stream of event data acquired by a Cisco Info Mediator into a file without any processing by the rules file. This can be useful for auditing, recording, and debugging the operation of a Cisco Info Mediator.

A Cisco Info Mediator running in raw capture mode does not process the events it acquires. The captured data is in a format that can be replayed by the Generic Info Mediator, as described in the Generic Info Mediator guide.

You can enable raw capture mode by setting the **-raw** command line option or the **RawCapture** property.

You can also set the **RawCapture** property in the rules file, so you can send the raw event data to a file only when certain conditions are met. See [Assigning Properties, page E-8](#) for more information on using Cisco Info Mediator properties in rules files.

Secure Mode

You can run the Cisco Info Server in secure mode. When you specify the **-secure** command line option, the Cisco Info Server authenticates Cisco Info Mediator, gateway, and proxy server connection requests by requiring a user name and an encrypted password. When a connection request is sent, the Cisco Info Server issues an authentication message. The Cisco Info Mediator, gateway, or proxy server must respond with the correct user name and password.

If you do not specify the **-secure** option, no security checks are performed on the connection request.

When connecting to a secure Cisco Info Server or proxy server, each Cisco Info Mediator must have the **AuthUserName** and **AuthPassword** properties in its property file. If the user name and password combination is incorrect, the Cisco Info Server issues an error message and rejects the connection.

The passwords used in secure mode must be encrypted using the **nco_crypt** utility, described in [Running the Cisco Info Server in Secure Mode, page 1-13](#). Then, add the **AuthUserName** and **AuthPassword** properties to the Cisco Info Mediator properties file with the corresponding user name and encrypted password before running the Cisco Info Mediator.

Proxy Server

You can use the **ConnectionRatio** property to configure the proxy server to maintain the ratio of incoming connections from Cisco Info Mediators to outgoing connections to a Cisco Info Server.

The default value of **10** creates a 10:1 ratio of incoming to outgoing connections. If ten Cisco Info Mediators are connected, only one Cisco Info Server connection is required. However, if an additional Cisco Info Mediator connects to the proxy server, a second connection to the Cisco Info Server is established.

A proxy server requires an entry in the Server Editor, as described in the *Cisco Info Center Installation and Configuration Guide, 3.6*. The Cisco Info Mediator must be configured to connect to the proxy server (named **NCO_PROXY** by default) by setting the **Server** property in the Cisco Info Mediator properties file. For more information about the proxy server, see [Chapter 1, “Configuring the Cisco Info Server and Proxy Server.”](#)

Using a Specific Cisco Info Mediator

Each Cisco Info Mediator has an abbreviated name that is used to identify the Cisco Info Mediator executable and its associated files. For example, the abbreviated name for SunNet Manager is **snmlog**, the abbreviated name for HP Network Node Manager is **nnm**, and the abbreviated name for IBM Netview/6000 is **nv**.

For the Cisco Info Mediator for SunNet Manager, the executable is named:

\$OMNIHOME/probes/<arch>/nco_p_snmlog

The properties file is named:

\$OMNIHOME/probes/<arch>/snmlog.prop

The rules file is named:

\$OMNIHOME/probes/<arch>/snmlog.rules

In these paths, **<arch>** is the name of the architecture on which the Cisco Info Mediator is installed. See the guide for each Cisco Info Mediator, available on the Cisco Systems Support Site, for details about specific Cisco Info Mediators, their defaults, and which of their properties can be changed.

Running a Cisco Info Mediator

This section describes how to run a Cisco Info Mediator from the command line.

**Note**

When you run the Cisco Info Center configuration utility, **nco_config**, Cisco Info Mediators are configured to run under Process Control. Process Control is described in [Chapter 7, “Process Control.”](#)

Once you have installed the Cisco Info Mediator, you must configure the properties and rules files to fit your environment. For example, if you are using the HTTP Common Log Format Cisco Info Mediator, you must set the **LogFile** property, so the Cisco Info Mediator can connect to the event source. See [Cisco Info Mediator Components, page 5-5](#) for more information about properties and rules files.

To run a Cisco Info Mediator, enter:

```
$OMNIHOME/probes/nco_p_<probenam>
```

where *<probenam>* is the abbreviated name of the Cisco Info Mediator you want to run. You can also specify command line options when you run a Cisco Info Mediator. For example, to run the Sybase Info Mediator in raw capture mode, enter:

```
$OMNIHOME/probes/nco_p_sybase -raw
```

**Note**

If you are running a proxy server, you must connect your Cisco Info Mediators to the proxy server rather than to the Cisco Info Server. To do this, use the **Server** property or the **-server** command line option and specify the name of the proxy server. For more information, see [Proxy Server, page 5-11](#).
