PRISM MICROSYSTEMS

TrapTracker

# User's Guide

# Contents

# About this Guide

## Purpose of this guide

This guide educates the end-user to understand the interface better and to work with the application efficiently without any hassles.

## Who should read this guide

The targeted audience:

- Network Administrators, who are designated to monitor and manage the health of mission critical networks, RDBMS and applications.
- Technical support personnel who can identify the problems and take appropriate action before adverse situations happen.

## Typographical Conventions

Before you start, it is important to understand the typographical conventions followed in this guide:

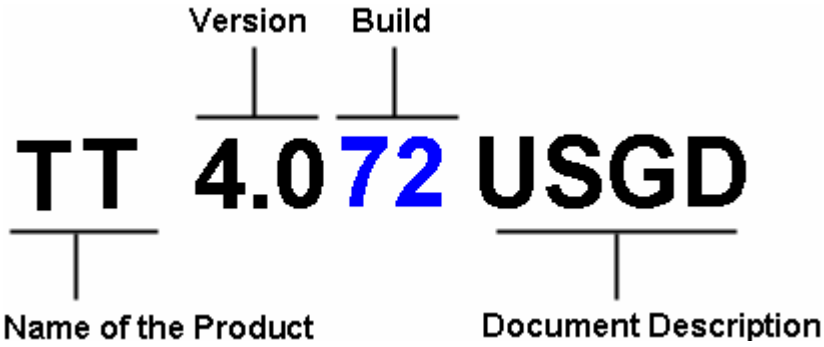
Table 1

| This | Represents |
|---|---|
| **Italics** | References to other guides and documents. |
| **Bold** | Input fields, radio button names, check boxes, drop-down lists, links on screens, menus and menu options. |
| **CAPS** | Keys on the keyboard and buttons on screens. |
| `{Text_to_customize}` | A placeholder for something that you must customize. For example, `{Server_Name}` would be replaced with the name of your server/ machine name or an IP address. |
| `Constant width` | Text that you enter, program code, files and directory names, function names. |
| 🗎 | A note, providing additional information about a certain topic. |

# Document Revision Control

Document Revision Control is an alphanumeric acronym. The components of the acronym identify the following:

- First two letters – name of the product
- Second two numbers – version of the product
- Third two numbers – build of the product
- Last four letters – document description




The document revision control for this guide is as given below:

Table 2

| | |
|---|---|
| **File Name** | TrapTracker v4.0 b72 User Guide |
| **Description** | Updated in accordance with release version 4.0 build 72. |
| **Status** | Final |
| **Release Date** | |

# How to Get In Touch

The following section provides information on how to obtain support for documentation and software.

## Documentation Support

We welcome your honest comments and thoughtful suggestions about the quality and usefulness of this document. For further questions, comments and suggestions on this documentation, contact us at support@prismMicroSys.com.

## Customer Support

For technical assistance regarding TrapTracker for Windows, contact us at support@prismmicrosys.com. While contacting technical support, please have the following information ready:

- Your name, e-mail address, phone number and fax number
- Topology of the network, type of hardware and the configuration you administered
- Version of TrapTracker for Windows
- Operating system
- The error message you encountered or any other error messages that appeared on your screen
- Description of how you tried to fix the problem

# Chapter 1
# Getting Started

*In this chapter, you will learn how to:*

- Start TrapTracker for Windows
- Work with Trap Windows
- Work with Traps
- Upgrade License
- Exit TrapTracker

# What is TrapTracker for Windows?

The Simple Network Management Protocol (SNMP) is today a de-facto industry standard for monitoring and managing devices on data communication networks, telecommunication systems and other globally reachable devices. Practically every organization dealing with computers and related devices aims to centrally monitor, diagnose and configure each such device across local and wide area networks. SNMP is the protocol that enables this interaction.

TrapTracker for Windows [TTW] is a scalable, standard-compliant framework that receives traps send by the SNMP compliant devices. TTW provides options to categorize traps, generate custom reports and configure notifications on occurrence of a specific trap.

TrapTracker for Windows helps the user to:

- Monitor, consolidate, and analyze traps sent by SNMP compliant devices
- Parse MIB (based on ASN-1 format) files.
- Retrieve object and trap definitions from MIB file. This implies that MIB modules describing the traps are compiled to facilitate the translation of SNMP PDUs into user understandable format. Traps that cannot be translated should not be discarded, but should be displayed and stored in raw format.
- View the contents of MIB files in a format easily understood by the user.
- Compile and store multiple MIBs in a single file.
- Collect and consolidate Trap details, Category details and Alert details into the database.
- Configure real-time notification by E-mail, beep, and custom action.
- Conform to audit requirements suggested by GLBA, HIPAA, Sarbanes-Oxley Changing Client Service Account, California Senate Bill 1386, the USA Patriot Act and NISPOM.

# TrapTracker Components

TTW version 4.x has the following components.

- A background process that receives and processes generic SNMP v1 and v2 traps; send by SNMP compliant devices.
- Feature-rich GUI application to categorize traps, filter traps for customized views, configure Alerts, upgrade license etc
- A MibCompiler

## TrapTracker Manager

TrapTracker Manager is the nerve center of the framework. It collects SNMPv1 and v2 traps sent by various SNMP compliant network devices, validates and logs them into the database and checks whether any Alert needs to be performed. The TTW Manager employs the MibCompiler to translate the traps received by TrapTracker Receiver service at port 162 into user-friendly names.

The GUI enables the user to:

- View live System window that provides information about trap activity on all monitored devices. Whenever a device generates traps, the criticality of the traps is indicated by visual indicators in the All Traps window as well as in the Systems window. The System window also provides a view, where only the latest traps that occurred on the system can be viewed and acknowledged.
- Filter Traps for view by setting criteria
- Acknowledge the traps that are viewed. The acknowledge traps are cleared from view, but are committed to the database. The TrapTracker Manager automatically acknowledges traps that are older than a specified time frame.
- Spawn multiple new windows on the console with each window showing only traps that match its own selection criteria.
- Script User Notes for any specific Trap. The User Notes is useful to keep track of what action was taken for a Trap, before the trap was acknowledged (cleared from view). The Notes entered here are visible in the reports/history.
- Import and Export Categories
- Generate customized reports

> **Note**
>
> By default, the TTW Manager uses port **162** to listen for SNMP traps sent by SNMP compliant devices.

Database is the repository of all received traps, configured Alerts and other configuration data.

## MibCompiler

The collection of related objects implemented by a system is called an MIB: Management Information Base. All network resources that are to be monitored are described in the form of objects using ASN-1 language and stored in a MIB file.

MibCompiler is responsible for parsing an input MIB file and checking its syntax and semantics for any error (if present). After successful compilation, it keeps object

information in binary format, which is used by TTW Manager from SNMP PDUs to traps translation.

The MibCompiler/Browser helps in compilation of custom MIBs into the TTW system.

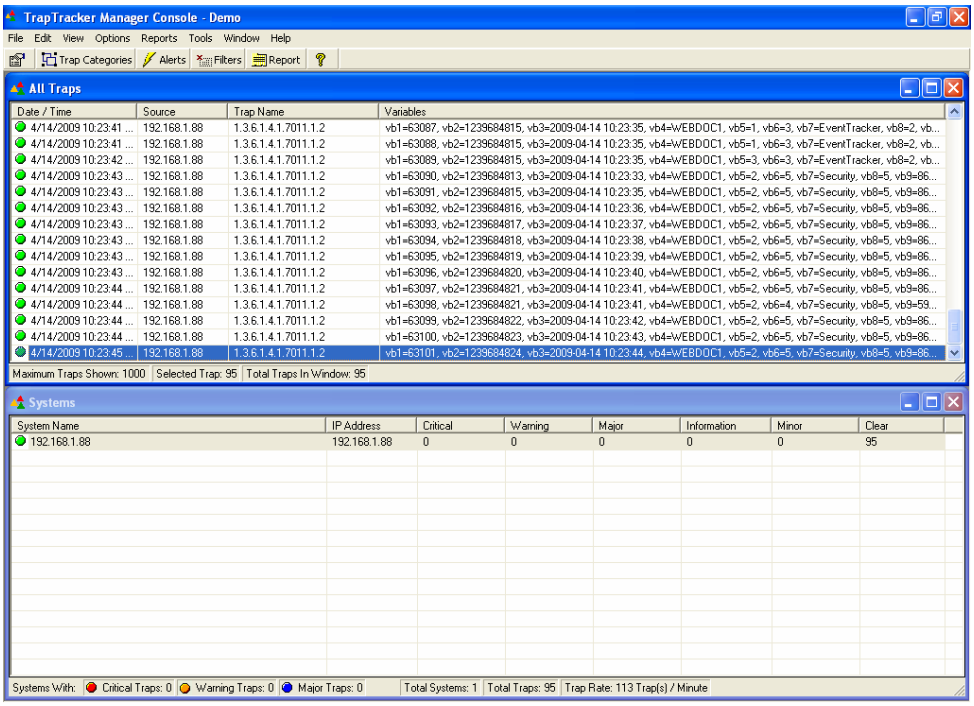
# Starting TrapTracker for Windows

## To start TrapTracker Manager

1. Double-click **TrapTracker** on the Control Panel.

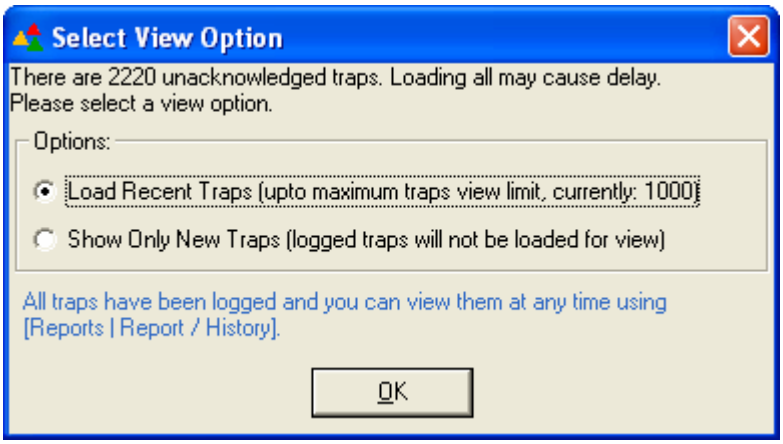
   TrapTracker displays the TrapTracker Manager Console.

Figure 1 TrapTracker Manager Console




If the number of unacknowledged traps exceeds the window view limit, TrapTracker displays the “Select View Option” dialog box.

Figure 2 Select View Option dialog box




2 To view only the recent traps, select the **Load Recent Traps (up to maximum traps view limit, currently 1000)** option.

3 To view new traps that are not logged into the database, select the **Show Only New Traps (logged traps will not be loaded for view)** option.

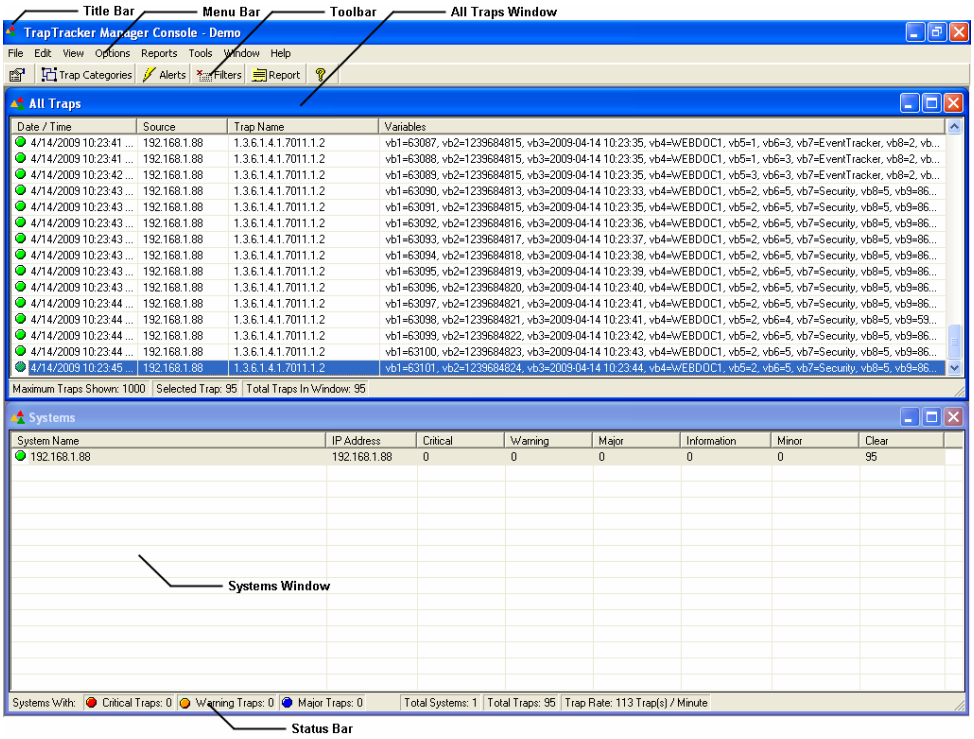
> **Note**
>
> TrapTracker does not display the “Select View Option” dialog box at the first instance you start the TrapTracker Manager Console.

4 Click **OK**.

# TrapTracker Manager Console

This section covers a conceptual overview of the TrapTracker Manager console. It helps you understand the menus and general interface of the TrapTracker Manager.

Figure 3 TrapTracker Console User Interface




### Title Bar

The top strip of the TrapTracker window is the Title Bar. The Title Bar shows the name of the application.

### Menu Bar

The menu bar contains menus with relevant commands. From the menus, choose appropriate commands or use shortcut keys to execute commands.

### Toolbar

The toolbar contains buttons with tool tips to perform basic tasks.

Table 3

| Click | To |
|---|---|
|  | View details of the trap selected from All Traps window. |
|  Trap Categories | Configure and manage Trap Categories |
|  Alerts | Configure Alerts and Alert actions. |
|  Filters | Configure and manage filters. |
|  Report | Generate consolidated reports by setting a wide range of parameters like Time Range, Categories, and VarBinds match. |

### All Traps Window

Displays all traps received by the TrapTracker Receiver. It can be resized, dragged and tiled vertically or horizontally. The maximum window view limit is 1000 and can be configured to display traps within this limit.

Table 4

| Field | Description |
|---|---|
| **Date/ Time** | Date and Time of the trap received by the TTW Receiver. |
| **Source** | Source from where the traps originated. |
| **Trap Name** | Name of the trap. |
| **Variables** | Variable definitions defined in the MIBs. |

### Status Bar

Displays the window view limit, serial number of the trap selected when a single trap is selected, or serial number of the last trap selected when multiple traps are selected and the total number of traps displayed currently in the window.

### Systems Window

Displays the name of all monitored SNMP compliant devices.

Table 5

| Field | Description |
|---|---|
| **System Name** | Name and domain of the SNMP compliant device. |
| **IP Address** | IP address of the device |
| **Critical** | Count of Critical severity traps |
| **Warning** | Count of Warning severity traps |
| **Major** | Count of Major severity traps |
| **Information** | Count of Information traps |
| **Minor** | Count of Minor severity traps |
| **Clear** | Count of Clear severity traps |

### Status Bar

The first section displays the trap criticality legend. The second section displays the total number of systems being monitored, total number of traps received from the monitored systems and the rate at which the traps are received.

# Working with Trap Windows

TrapTracker provides an option to open up multiple trap windows, with each window displaying only traps that satisfy its selection criteria. This feature is useful in viewing the trap activity of certain devices in isolation from the rest of the enterprise.

## Creating a New Trap Window

This option enables you to create a new trap window.

### To create a new trap window

1. Open the TrapTracker Manager console.
2. From the **File** menu, choose **New Window**.

   (OR)

   Press **Ctrl + N** on your keyboard.

   (OR)

   From the **Windows** menu, choose **New Window**.

   TrapTracker displays the “Select Window Parameters” window.

Figure 4 Select Window Parameters dialog box

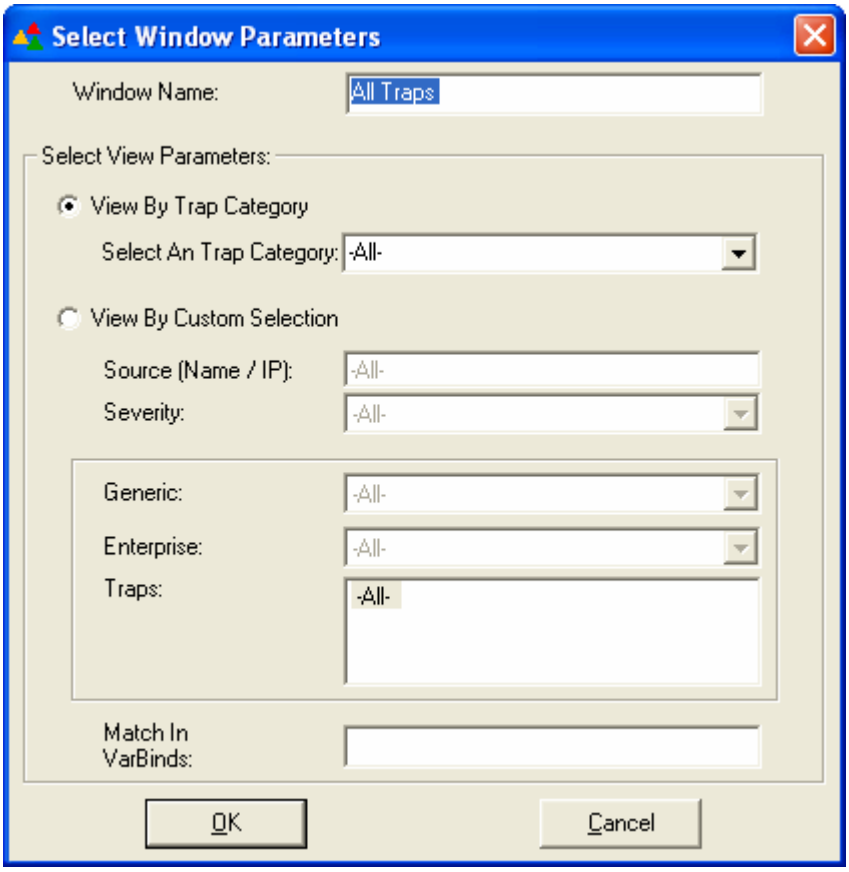



Table 6

| Field | Description |
|---|---|
| **Window Name** | Type a descriptive name of the window. |
| **View By Trap Category** | This option is selected by default.<br>Select a pre-defined Category or user-defined category from the “Select A Trap category” drop-down list.<br>By default, the drop-down list has the following values -All-, sysStartup Events, linkUp, linkDown.<br>The list gets populated along with the pre-defined Categories when you create new Categories. |
| **View By Custom Selection** | |
| **Source (Name/IP)** | Type the name or IP address of the source of traps. You can explicitly define the Name/IP address of SNMP compliant devices and monitor traps sent only by those devices. |

| Field | Description |
|---|---|
| **Severity** | Select a severity level of traps.<br>Available options are -All-, Clear, Minor, Information, Major, Warning, and Critical. |
| **Generic** | This drop-down list is populated with pre-defined generic traps, which are common to all the SNMP compliant devices.<br>The generic traps are as follows: -All-, coldStart, warmStart, linkDown, linkUp, authenticationFailure, egpNeighborLoss, and enterpriseSpecific.<br>EnterpriseSpecific are vendor-specific traps, which are defined by the vendors so that their devices can meet their special management needs.<br>coldStart - the sender is reinitializing and its configuration may change<br>warmStart - the sender is reinitializing but its configuration will not change<br>linkDown - failure in one of the agent's links<br>linkUp - one of the agent's links is up<br>authenticationFailure - the agent received an improperly authenticated protocol message authenticated<br>egpNeighborLoss - an Exterior Gateway Protocol neighbor is down<br>enterpriseSpecific - The trap is identified as not being one of the a basic one |
| **Enterprise** | This option is enabled only when you choose the "enterpriseSpecific" option in the "Generic" drop-down list.<br>This list box is populated with the available compiled MIBs. |
| **Traps** | This list box is populated with the traps that are available in the enterprise MIB you have chosen. |
| **Match in VarBinds** | To further narrow down your selection criteria, you can type a variable in this field.<br>The new window you create will display the traps that match the variable you have typed. |

*3* Select/enter appropriately in the relevant fields.

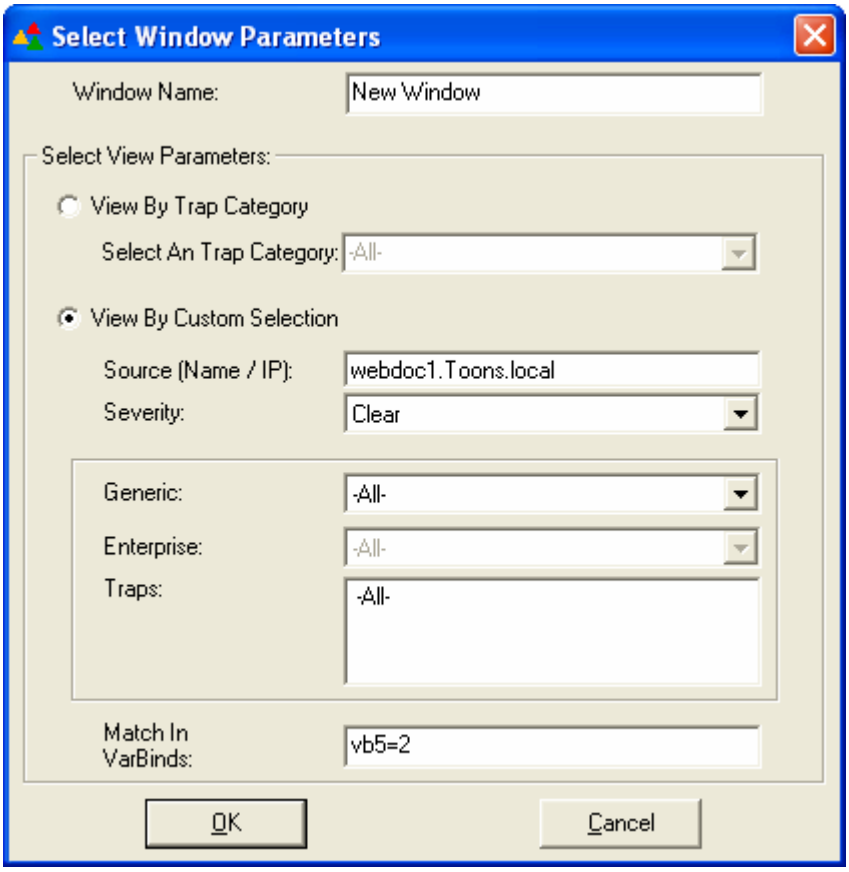
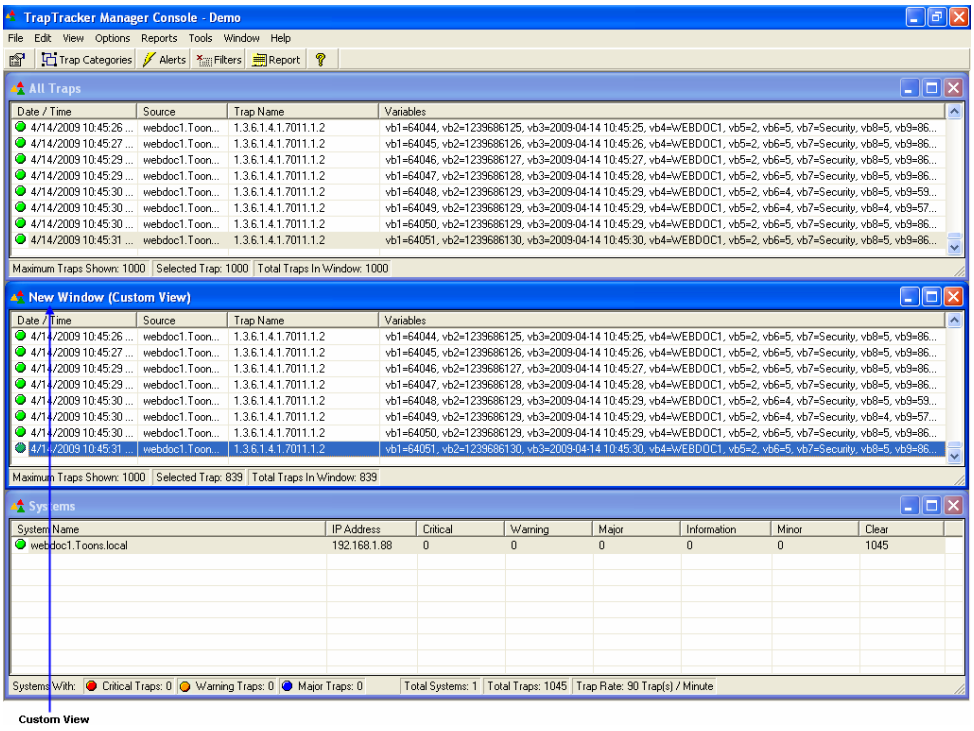



Figure 5 Select Window Parameters dialog box with user-defined parameters

TrapTracker displays the trap details for the aforementioned selection criteria in a new window.

Figure 6 New Trap Window




## Renaming a Trap Window

This option enables you to rename a trap window.

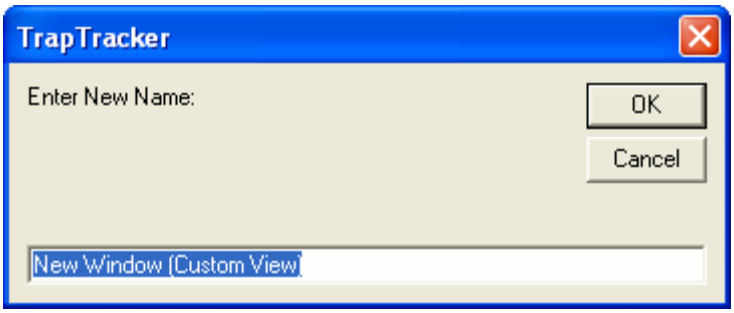
### To rename a trap window

1. Click the window that you want to rename.
2. From the **Edit** menu, choose **Rename Window**.

> **Note**
>
> This option is available for **All Traps** window, **Systems** window, and user-defined windows.

TrapTracker displays the “TrapTracker” dialog box.

Figure 7 TrapTracker rename dialog box




3 Type an appropriate name in the **Enter New Name** field

4 Click **OK**.

(OR)

Right-click any row on the window that you want to rename.

> **Note**
>
> This option is available only for **All Traps** window & user-defined windows and not for **Systems** window.

TrapTracker displays the shortcut menu.

From the shortcut menu, choose **Window Properties**.

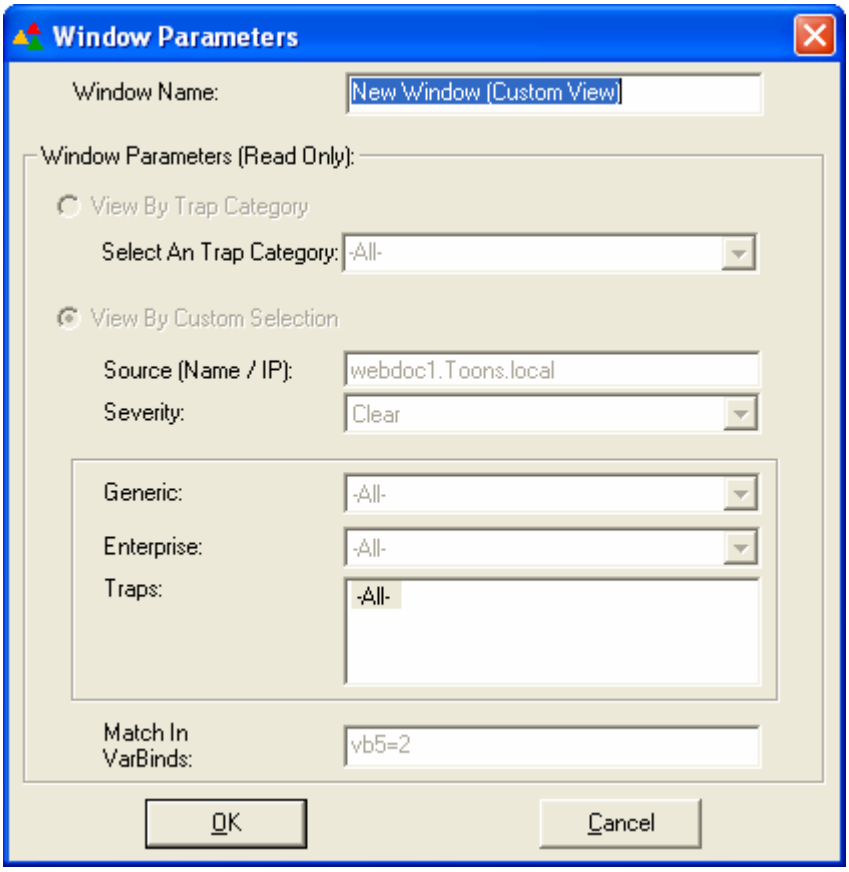
(OR)

From the **View** menu, choose **Window Properties**.

> **Note**
>
> This option is available only for **All Traps** window & user-defined windows and not for **Systems** window.

TrapTracker displays the “Window Parameters” window.

Figure 8 Window Parameters dialog box




5 Type an appropriate name in the **Window Name** field and then click **OK**.

> **Note**
>
> In Window Parameters dialog box, only the **Window Name** field is editable. The remaining fields are disabled and non-editable.

## Cascading Trap Windows

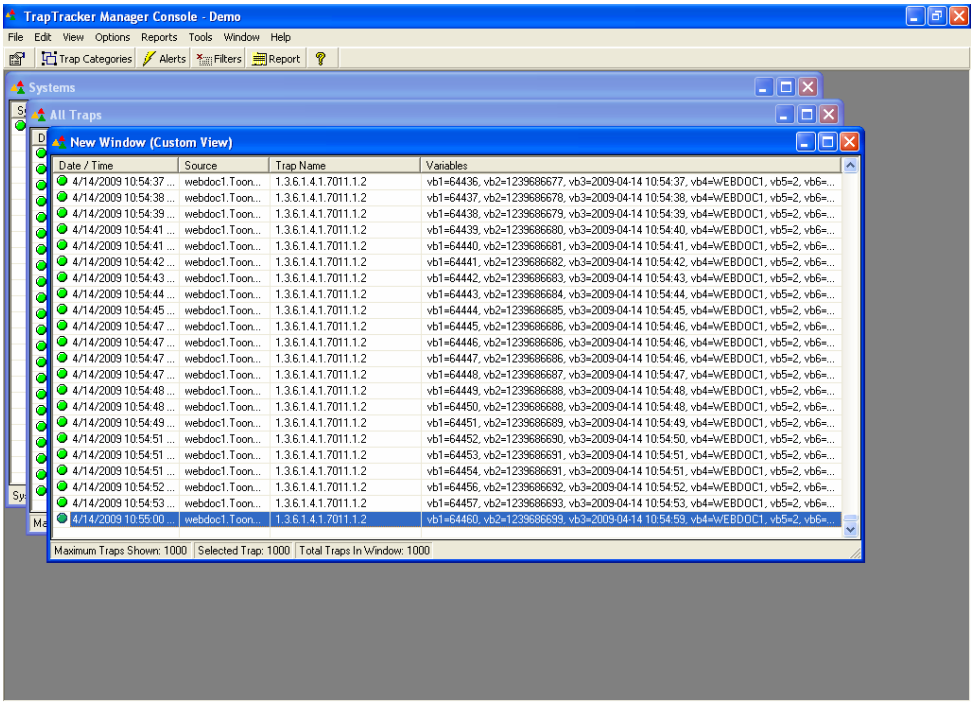
This option enables you to cascade all trap windows.

### To cascade trap windows

- From the **Window** menu, choose **Cascade**.

  TrapTracker displays the cascaded trap windows.

Figure 9 Cascaded Trap Windows.




## Tile Trap Windows Horizontally

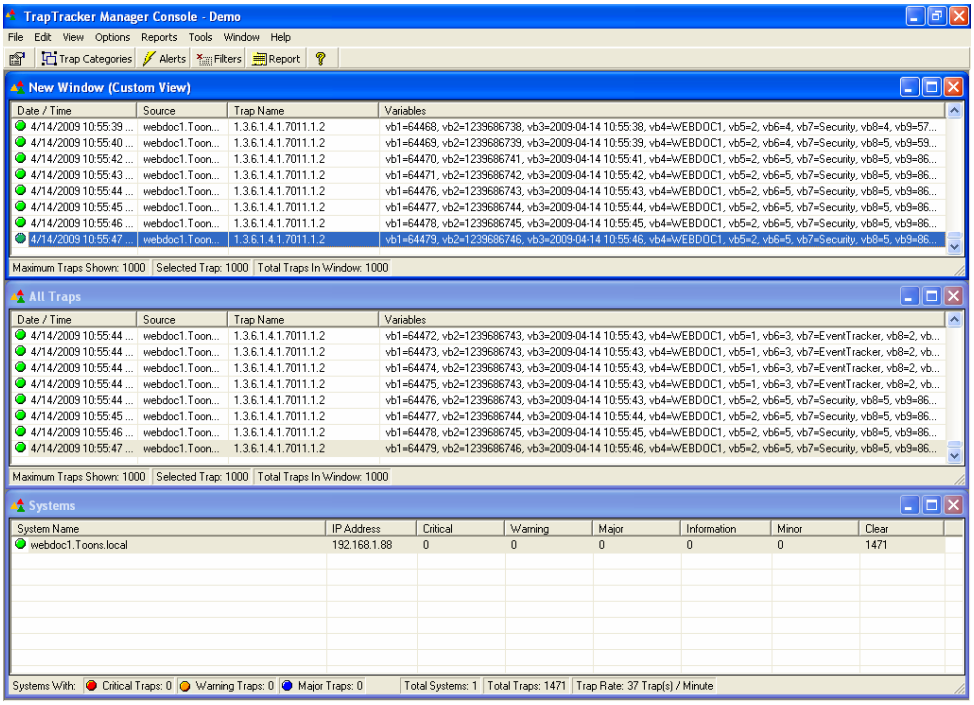
This option enables you to tile trap windows horizontally.

### Tile trap windows horizontally

- From the **Window** menu, choose **Tile Horizontal**.

  TrapTracker displays the trap windows tiled horizontally.

Figure 10 Trap Windows tiled horizontally.




## Tile Trap Windows Vertically

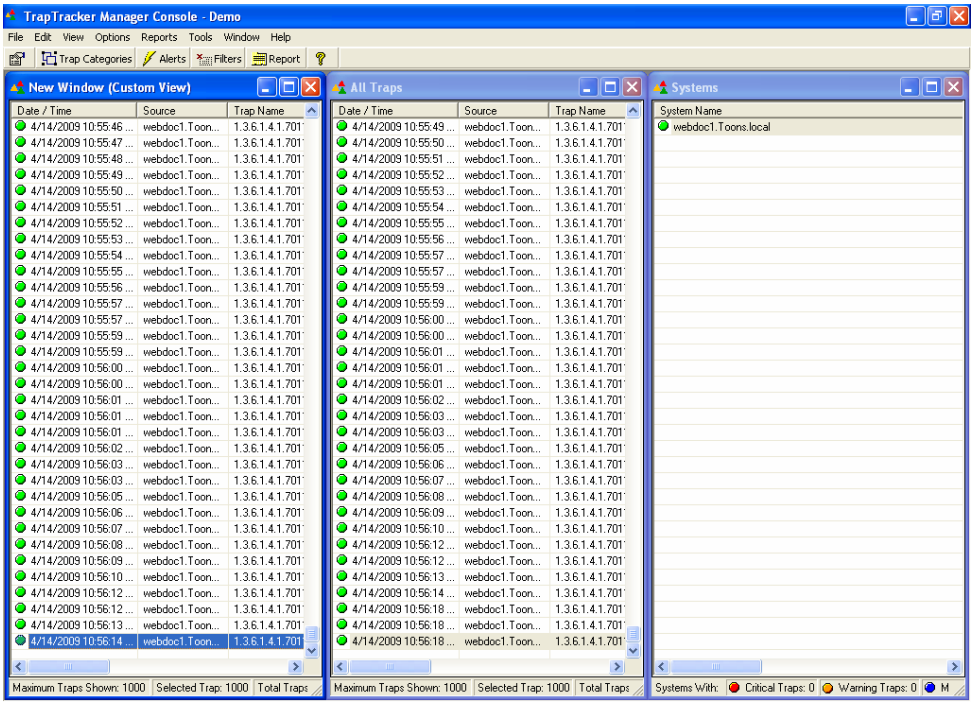
This option enables you to tile trap windows vertically.

### Tile trap windows vertically

- From the **Window** menu, choose **Tile Vertical**.

  TrapTracker displays traps windows tiled vertically.

Figure 11 Trap Windows tiled vertically.




## Closing a Single Trap Window

This option enables you to close a single trap window.

### To close a single trap window

1. Click the window that you want to close.
2. From the **Window** menu, choose **Close**.

   TrapTracker closes the selected window gracefully.

**Note**

TrapTracker minimizes, when you try to close the **Systems** window.

## Closing All Trap Windows

This option enables you to close all trap windows

### To close all windows

- From the **Window** menu, choose **Close All**.

  TrapTracker closes all windows except the **Systems** window.

## Viewing Window Properties

This option enables you to view properties of a selected window

### To view properties of a selected window

1. Click the window that you want to view properties.
2. From the **View** menu, choose **Window Properties**.

   (OR)

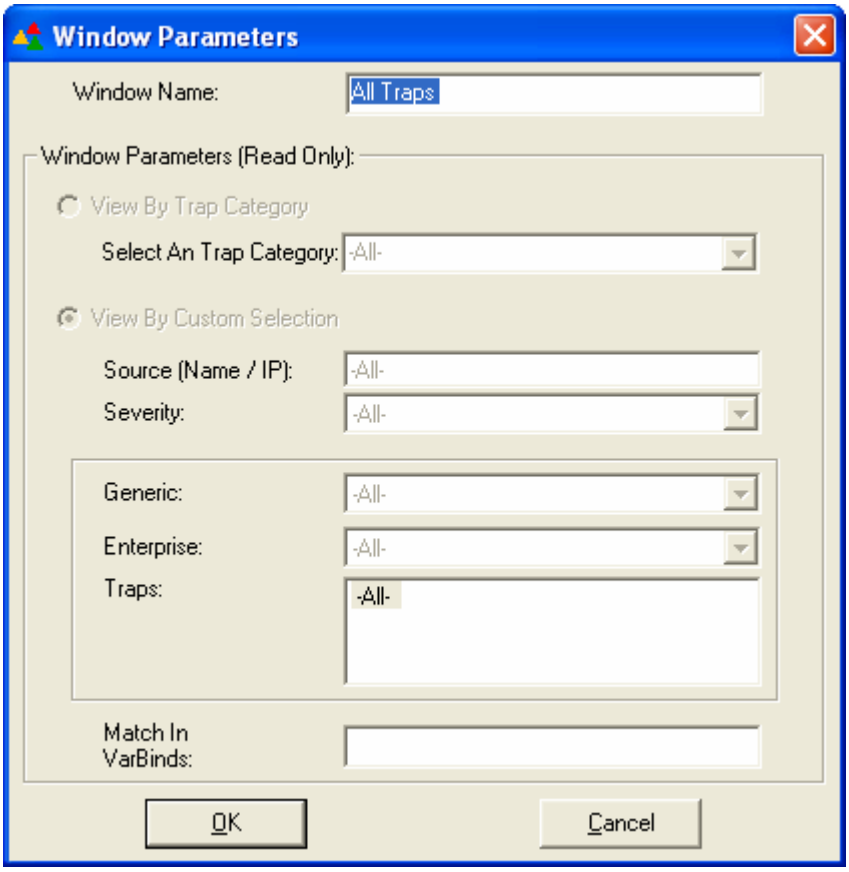
   Right-click a trap detail record on the All Traps window or the new trap window created by you.

   TrapTracker displays the shortcut menu.

   From the shortcut menu, choose **Window Properties**.

   TrapTracker displays the “Window Parameters” window.

Figure 12 Trap window parameters dialog box.




Only the Window Name field is editable, while the remaining fields are disabled.

**Note**

You cannot view properties of the Systems window. When you choose Systems window to view properties, TrapTracker disables the **Trap Details** command button on the toolbar, and **Window Properties** command on the **View** menu.

## View All Trap Details in the Notepad

This option enables you to view all trap details in the Notepad.

### To view all trap details (Print Preview)

1 Click the **All Traps** window or any other trap window you have created.

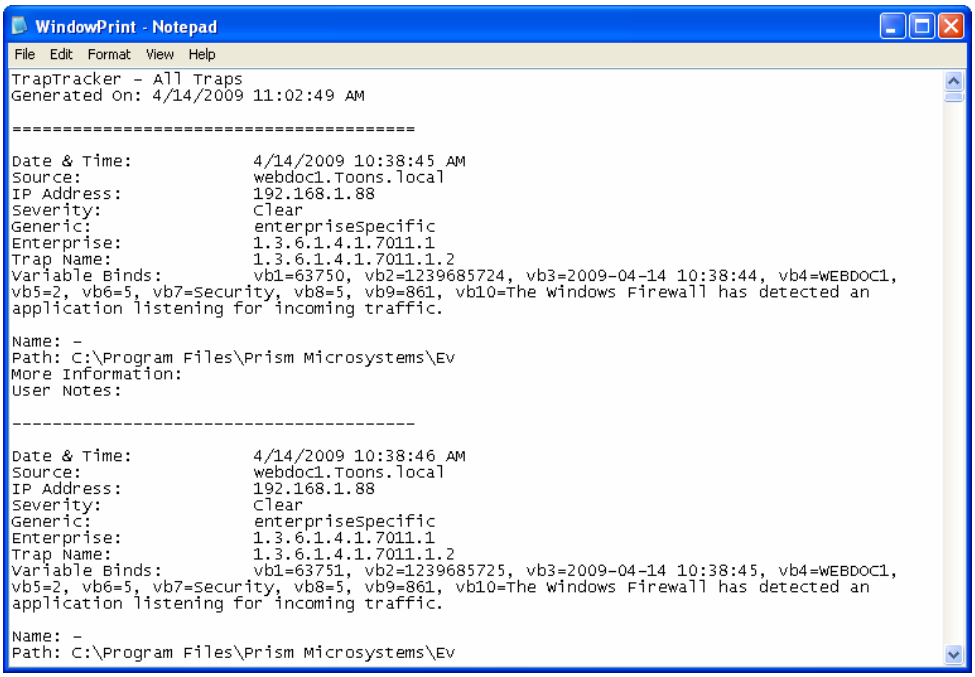
2 From the **File** menu, choose **Print Preview.**

(OR)

Press the shortcut keys **CTRL+P** on you keyboard.

TrapTracker displays the trap details in the Notepad.

Figure 13 Print Preview.




You can print the trap details by selecting appropriate commands in the Notepad.

**Note**

You cannot view the print preview when you choose the **Systems** window. TrapTracker disables **Print Preview** command on the **File** menu.

## View All Trap Details in a Window

This option enables you to view trap details in a window.

### To view trap details

1 Click the **All Traps** window or any other trap window that you have created.

2 Right-click any row that you want to view details.

TrapTracker displays the shortcut menu.

From the shortcut menu, choose **Trap Details**.

3 You can also view the trap details by choosing **Trap Details** from the **View** menu.

(OR)

Click on the toolbar.

(OR)

Double-click the row that you want to view details.

TrapTracker displays the "Trap Detail" window.

Figure 14 Trap Detail dialog box

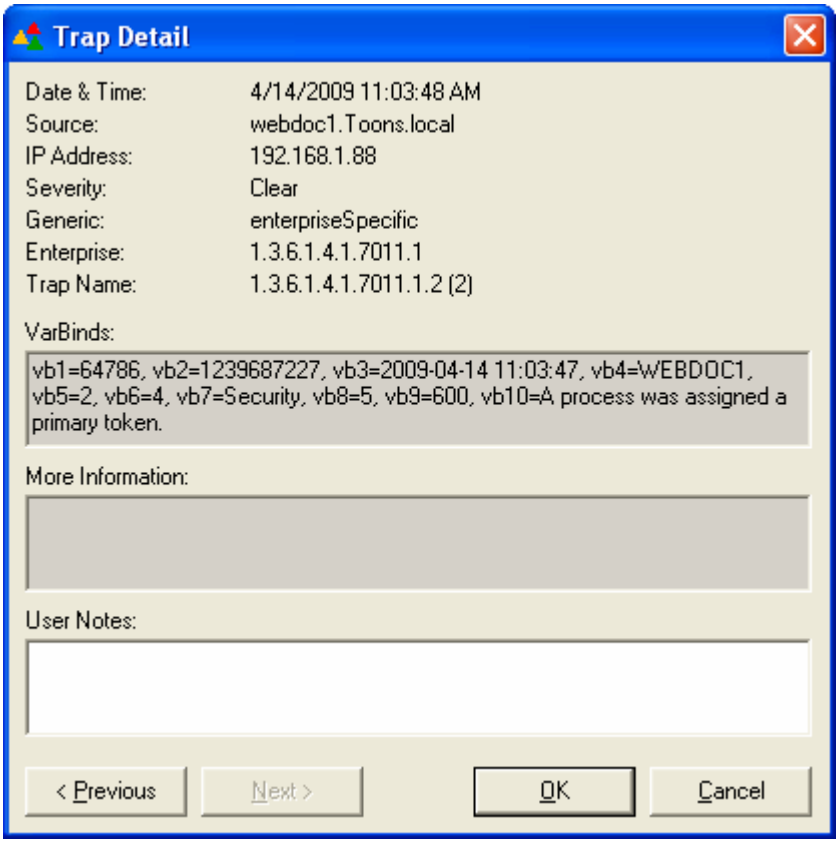



Table 7

| Field | Description |
|---|---|
| **Date & Time** | Date and time when the TTW Manager received the trap. |
| **Source** | Name and domain of the SNMP complaint device. |
| **IP Address** | IP address of the SNMP complaint device. |

| Field | Description |
|---|---|
| **Severity** | Severity level of the trap. |
| **Generic** | Category type that the trap belongs to. |
| **Enterprise** | Object Id |
| **Trap Name** | Name of the trap. |
| **VarBinds** | Variables associated with the trap. |
| **More Information** | Nature of the trap. Details like when it was triggered and to which server it was sent are displayed. |
| **User Notes** | These notes are useful to keep track of what action was taken for a trap, before it is acknowledged (cleared from view and committed to the database). The notes entered are also visible in the reports / history. |

4 Click **< Previous** to view trap details of the previous trap from the current position on the trap window.

5 Click **Next >** to view trap details of the next trap from the current position on the trap window.

## View Trap Details of a Selected System

This option enables you to view trap details of the selected system.

### To view trap details of the selected system

1 Right-click a row on the **Systems** window that represents the system details.

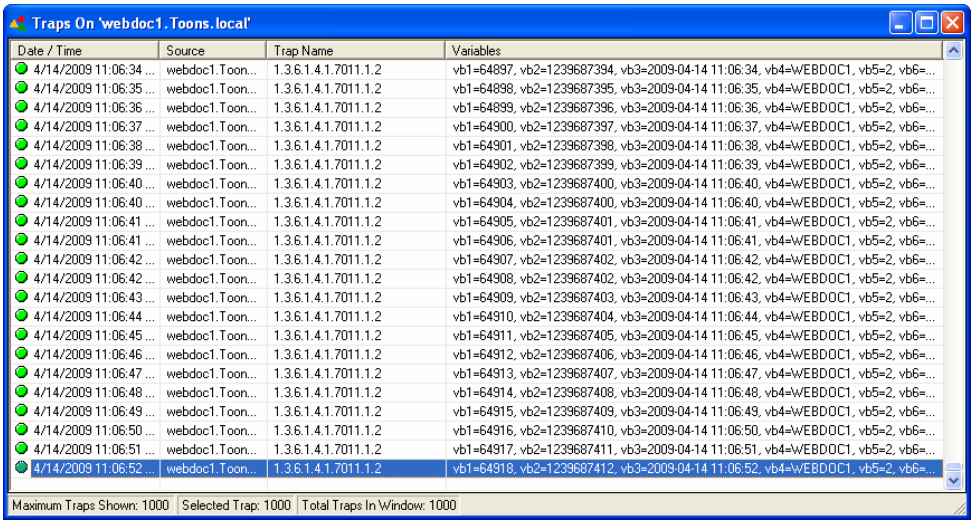
TrapTracker displays the shortcut menu.

From the shortcut menu, choose **View Traps For 'webdoc1.Toons.local'**.

> **Note**
>
> 'webdoc1.Toons.local' is the system name and the domain where it resides. The name and the domain vary according to your selection.

TrapTracker displays the traps sent by the selected system in a pop-up window (**Traps On 'webdoc1.toons.local'**).

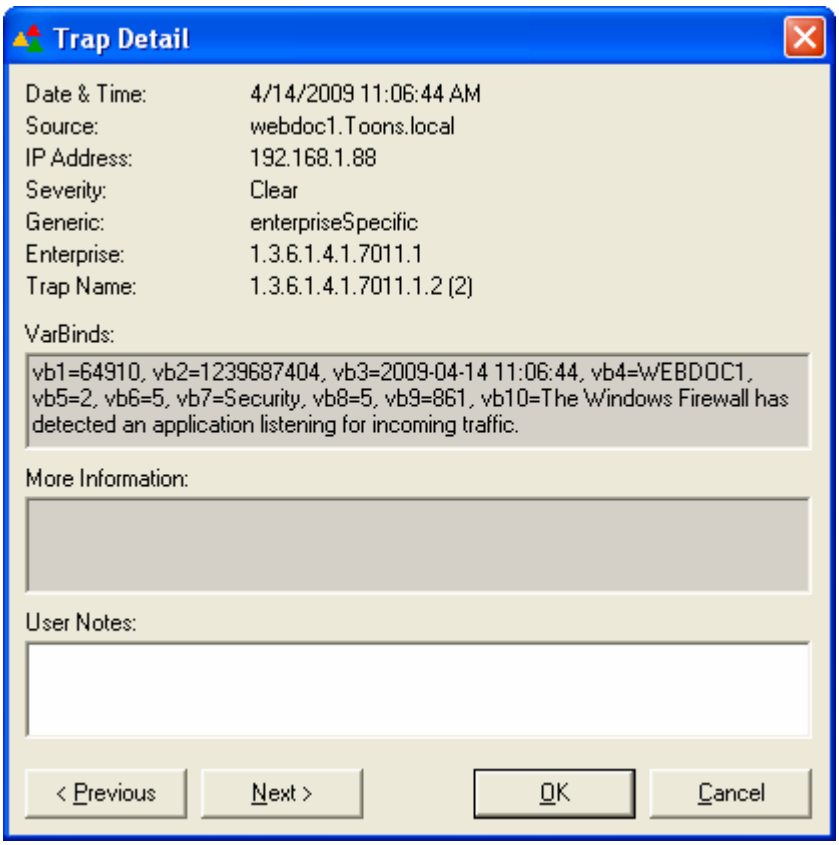
Figure 15 Trap detail window for the selected system




*2* Double-click the trap that you want to view details.

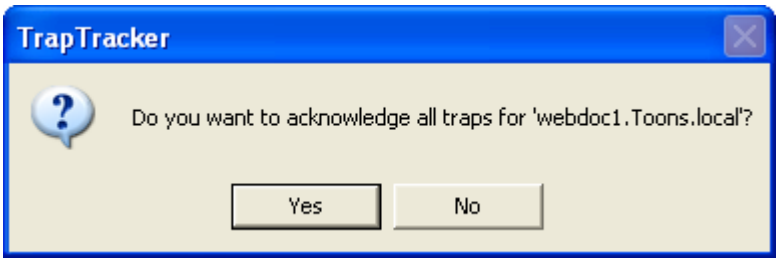
TrapTracker displays the trap details of that particular trap.

Figure 16 Trap Detail dialog box



TrapTracker displays the message box, when you close **Traps On 'webdoc1.toons.local'** pop-up window.

Figure 17 TrapTracker message box.




3 Click **Yes** to acknowledge traps and close the window.

4 Click **No** to close the window without acknowledging the traps.

# Clearing/Acknowledging Trap Details

While monitoring traps from the centralized console, traps that have already been viewed and acted upon can be acknowledged. On acknowledging, traps are just removed from the view and are not deleted or purged from the database. This feature ensures that only new traps received by the TTW Manager are displayed on the console.

## Clear/Acknowledge a single trap

This option enables you to clear/acknowledge a single trap.

### To clear/acknowledge a single trap

- Right-click the trap that you want to acknowledge on the All Traps window or on the custom trap window.

  TrapTracker displays the shortcut menu.

  From the shortcut menu, choose **Clear / Acknowledge Trap(s)**.

  (OR)

  From the **Edit** menu, choose **Clear/Acknowledge Trap(s)**.

  TrapTracker clears the selected trap from view.

## Clear/Acknowledge Multiple Traps

This option enables you to clear/acknowledge a group of traps.

### To clear/acknowledge a group of traps

1. Select a trap on the All Traps window or on the custom trap window you have created.
2. Hold the **SHIFT** key and point the mouse pointer to the trap until it reaches the group you want to clear/acknowledge.
3. Right-click any selected row.

   TrapTracker displays the shortcut menu.

   From the shortcut menu, choose **Clear / Acknowledge Trap(s)**.

   (OR)

   From the **Edit** menu, choose **Clear/Acknowledge Trap(s)**.

   TrapTracker clears the selected trap from view.

## Clear/Acknowledge All Traps of a Selected System

This option enables you to clear/acknowledge traps of a selected system.

### To clear/acknowledge all traps of a selected system

- Right-click the system that you want to clear from the **Systems** window.

  TrapTracker displays the shortcut menu.

  From the shortcut menu, choose **Clear / Acknowledge Trap(s) For 'gijoe.Toons.local'**.

  TrapTracker clears all traps of the selected system.

# *Auto Scroll*

Auto Scroll is an option that intimates you the arrival of a new trap. Suppose you are browsing through the All Traps window and have selected the first trap record to view its details, you will never come to know that a new trap has arrived unless you scroll down to the last record. But, by selecting Auto Scroll option, TrapTracker turns its focus on the new trap, scrolls down and highlights it.

## Viewing New Trap

This option enables you to view new trap.

### To view new trap

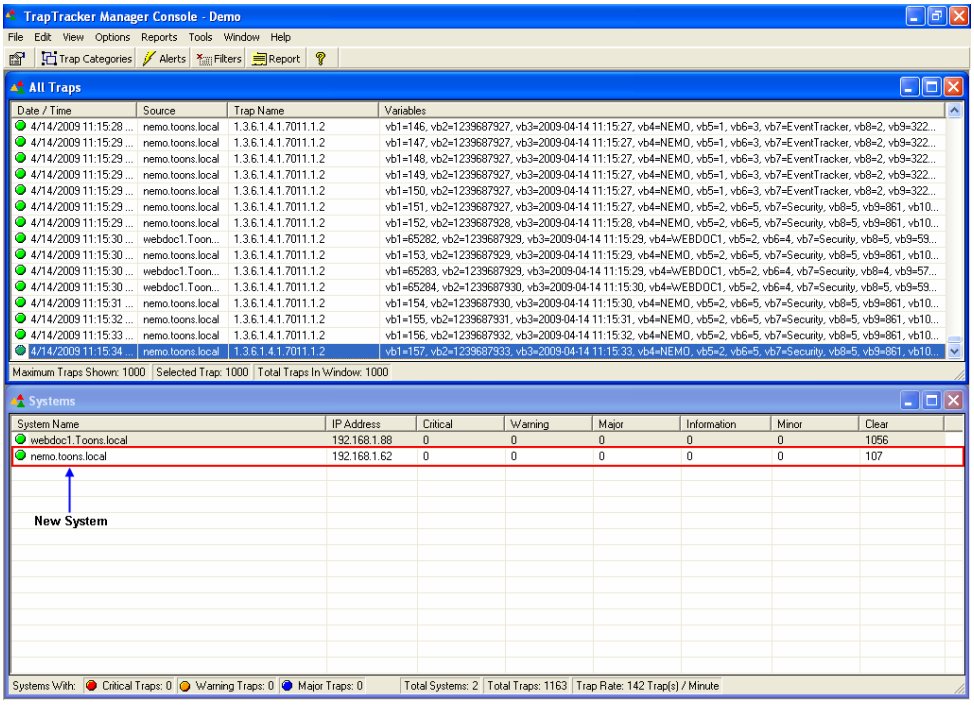
- From the **View** menu, choose **New Traps (Auto Scroll)**.

> **Note**
>
> This option is selected by default to intimate you the arrival of a new trap.

# *Adding a New System*

When you install a new SNMP compliant device in your enterprise, you will be prompted to enter the name or IP address of the Manager that receives traps sent by SNMP compliant device you have installed. Enter appropriate details. No further configuration is needed. TTW Manager listens at port 162 for incoming traps.

Figure 18 Systems window after adding a new system



# Upgrading License

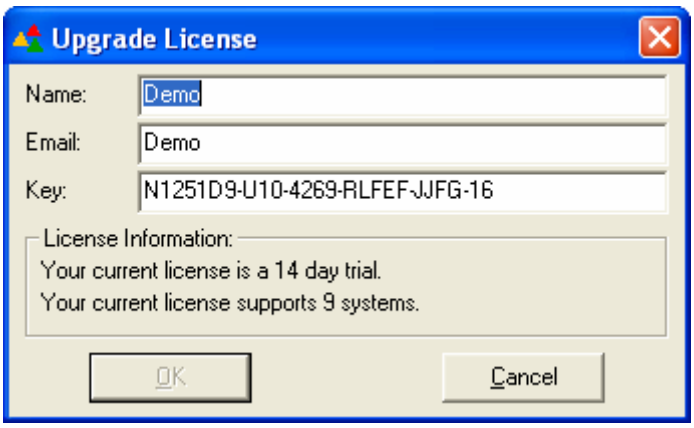
This option enables you to upgrade your license from trial version to registered version.

## To upgrade license

1. Open the TrapTracker Manager console.
2. From the **Help** menu, choose **Upgrade License.**

   TrapTracker displays the "Upgrade License**"** dialog box.

Figure 19 Upgrade License dialog box




**Note**

You can get the upgrade license information from sales@prismMicroSys.com.

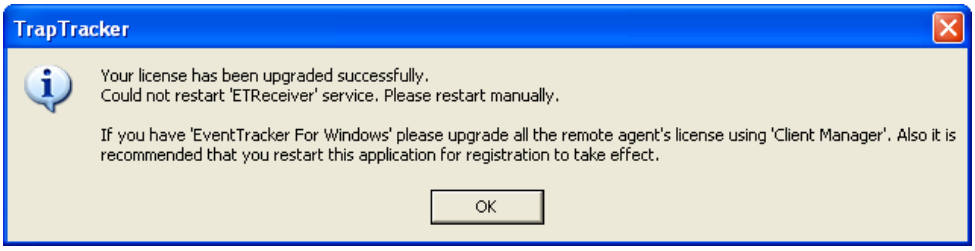
Table 8

| Field | Description |
|---|---|
| **Name** | Type your name. |
| **Email** | Type your e-mail address. |
| **Key** | Type the license key. Since the license key is case-sensitive, care should be taken while entering the key. |

3. Click **OK**.

   On successful acceptance of the license details, TrapTracker displays the TrapTracker confirmation message box.

Figure 20 Upgrade License dialog box




4 Click **OK**.

5 Restart the TrapTracker Receiver service manually as advised on the "TrapTracker" message box.

# Exiting TrapTracker

This option enables you to exit TrapTracker gracefully.

## To exit TrapTracker

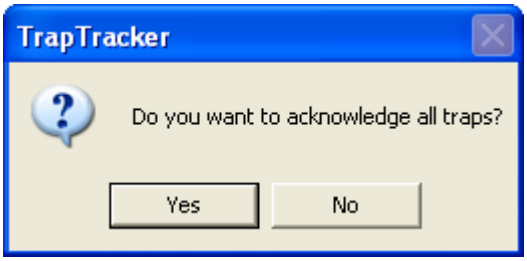
1 From the **File** menu, choose **Exit**.

(OR)

Click  at the upper-right corner of the TrapTracker Manager console.

TrapTracker displays the confirmation message box.

Figure 21 Exit confirmation dialog box




2 Click **Yes** to acknowledge all traps and exit.

3 Click **No** to exit without acknowledging the traps.

# Chapter 2
# Managing Traps

*In this chapter, you will learn how to:*

- Auto-Acknowledge traps
- Filter Traps
- Configure Alerts

# Auto-Acknowledge Traps

This option enables you to acknowledge traps older than a specific period of time, set window view limit, and purge traps from the database.

## To auto-acknowledge traps

- From the **Options** menu, choose **Configuration**.

  TrapTracker displays the “Configuration” window.

Figure 22
Configuration dialog box

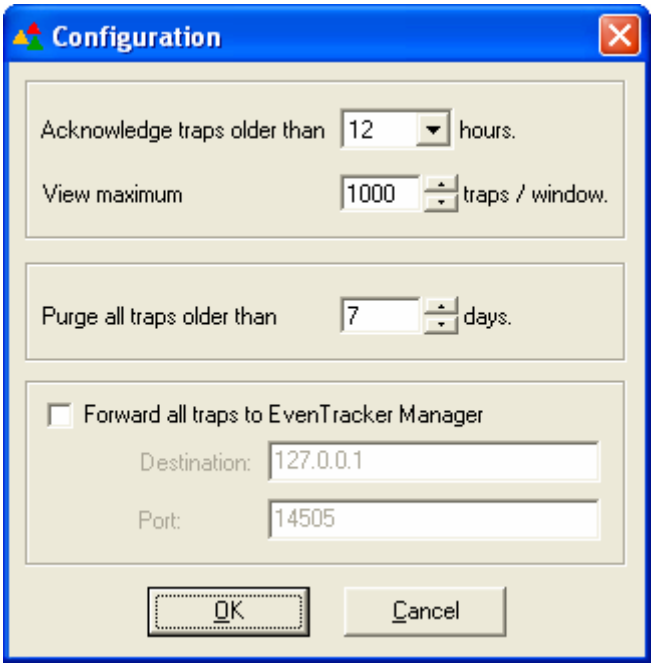



Table 9

| Field | Description |
|---|---|
| **Acknowledge traps older than** | Select an option from the drop-down list. Traps are cleared /acknowledged that are older that specified period of time. |
| **View maximum** | Set the number of traps that you want to view in a window. |
| **Purge all traps older than** | Select the number of days.<br>TrapTracker deletes traps permanently from the database after the specified number of days. |

| Field | Description |
|---|---|
| **Forward all traps to EventTracker Manager** | Select this check box.<br>Type the name or IP address of the EventTracker Manager in the Destination field.<br>Type port number through which the EventTracker Receiver receives traps in the Port field. |

# *Filtering Traps from View*

You can filter traps of minor significance from the view. They are not purged from the database but are logged into the database and available for reports and history views.

## Adding Trap Filter

This option enables you to add a trap filter.

### To add a trap filter

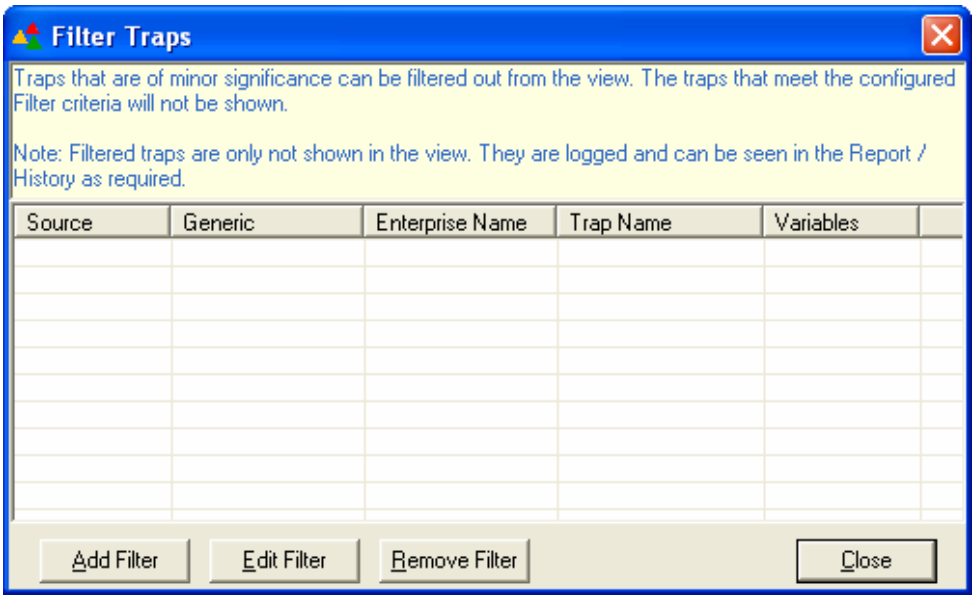
1. From the **Options** menu, choose **Filters**.

   (OR)

   Click **Filters** on the toolbar.

   TrapTracker displays the “Filter Traps” console.

Figure 23 Filter Traps dialog box.




*2* Click **Add Filter**.

TrapTracker displays the “Trap Filter” window.

Figure 24 Trap Filter dialog box

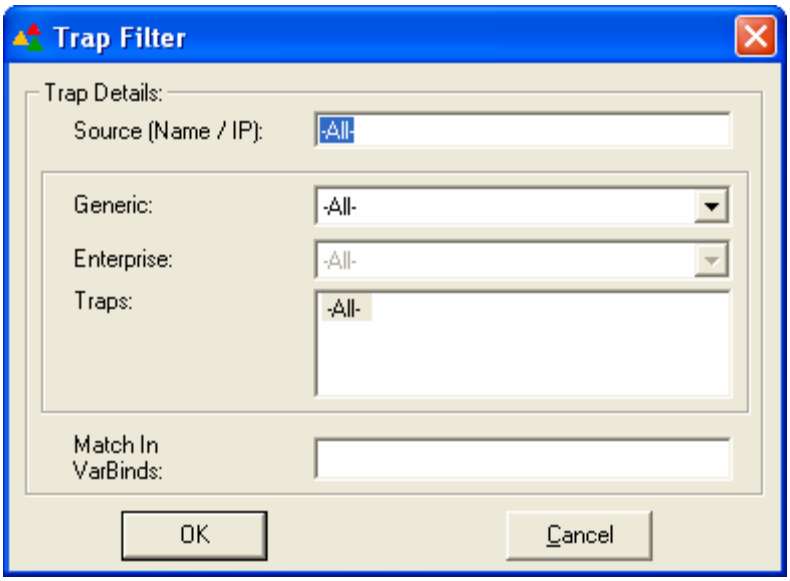


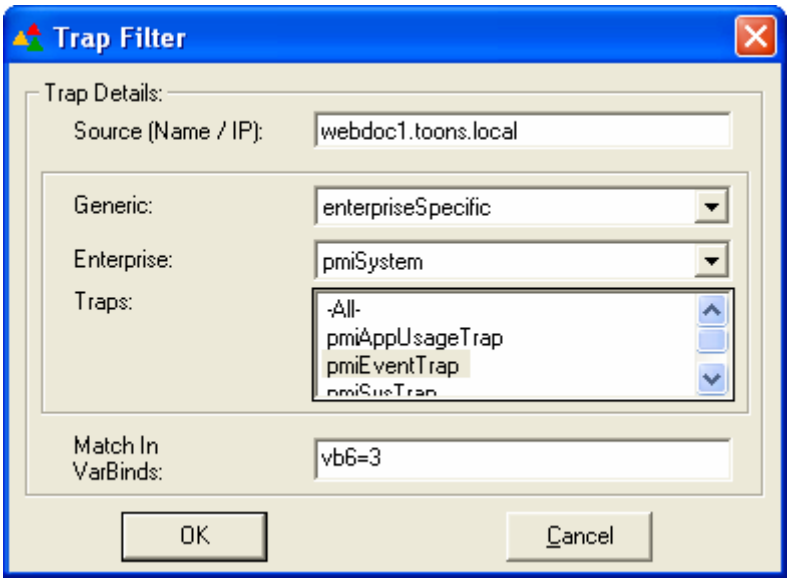

Table 10

| Field | Description |
|---|---|
| **Source (Name/IP)** | Type the name or IP address of the source of traps that you want to filter out. |

| Field | Description |
|---|---|
| **Generic** | This drop-down list is populated with pre-defined generic traps, which are common to all SNMP-compliant devices. |
| **Enterprise** | This option is enabled only when you choose the enterpriseSpecific option in the Generic drop-down list. This list box is populated with the available compiled MIBs. |
| **Traps** | This is a list box, which is populated with the traps that are available in the enterprise MIB you have chosen. |
| **Match in Varbinds** | To further narrow down your selection criteria, you can enter a variable in this field. The new window you create will display the traps that match the variable you have entered. |

*3* Select/enter appropriately in the relevant fields.

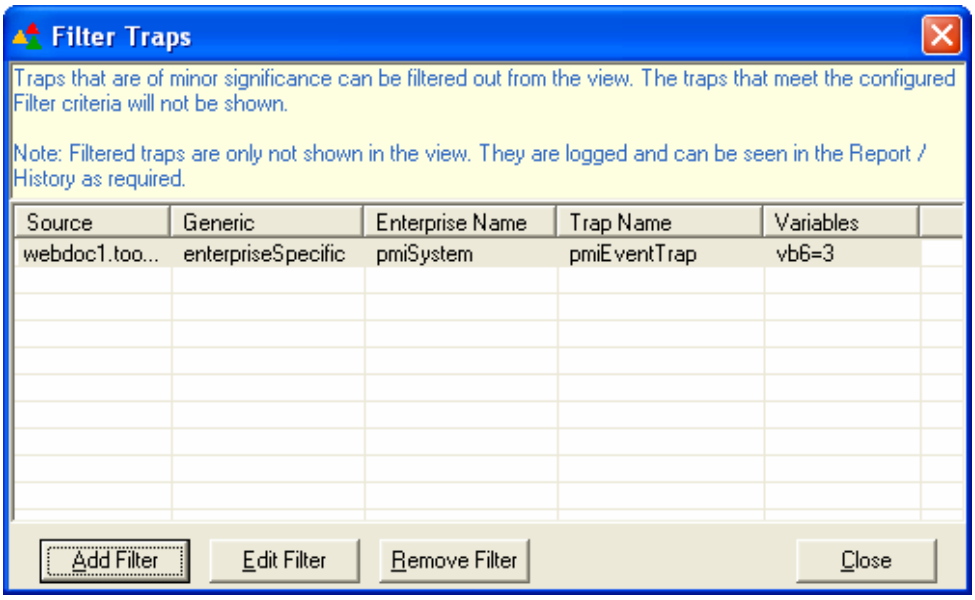
Figure 25 Trap Filter dialog box with data




*4* Click **OK**.

TrapTracker adds the newly created trap filter to the Filter Traps pool.

Figure 26 Filter Traps dialog box with newly added filter trap.




*5* Click **Close**.

## Modifying Trap Filter

This option enables you to modify the trap filter.

### To modify the trap filter

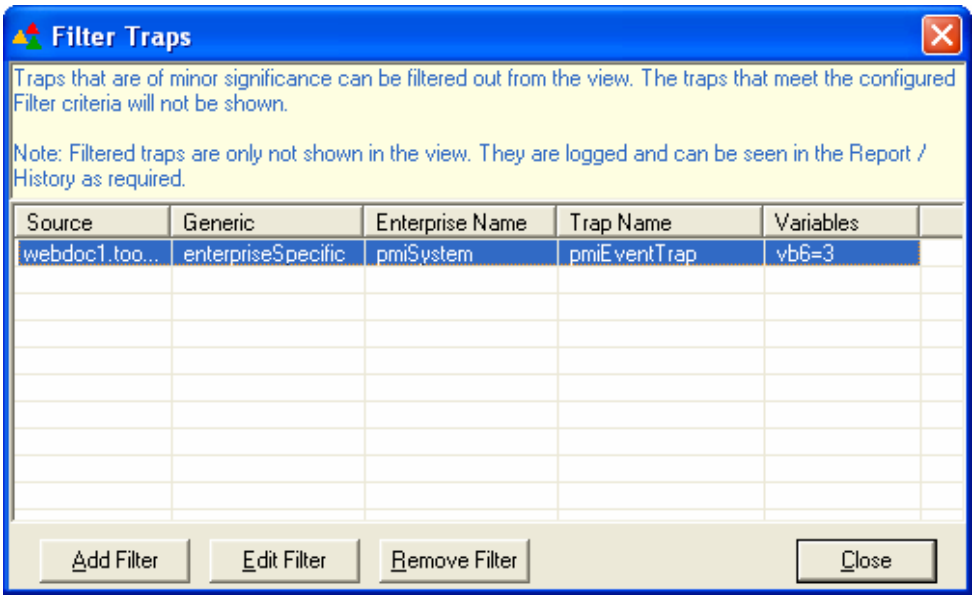
*1* From the **Options** menu, choose **Filters**.

(OR)

Click **Filters** on the toolbar.

TrapTracker displays the “Filter Traps” console.

Figure 27 Filter Traps dialog box.

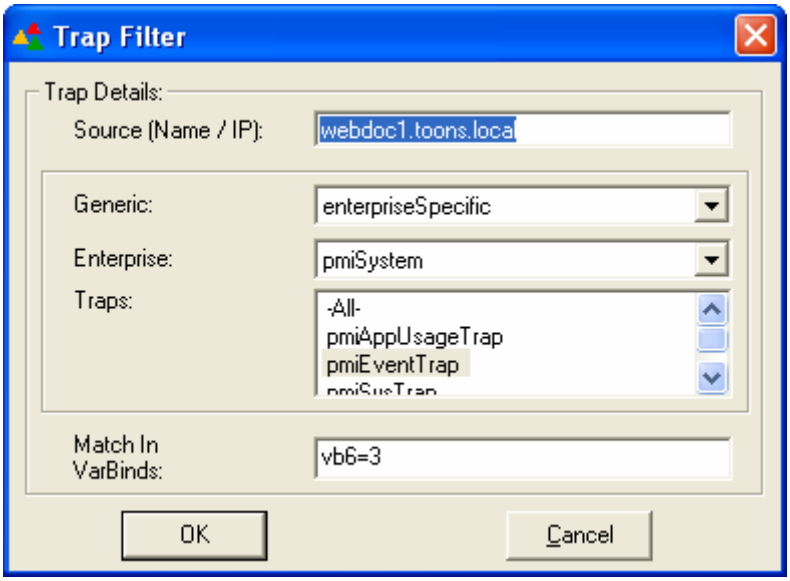



2 Select the filter that you want to modify.

3 Click **Edit Filter**.

TrapTracker displays the "Trap Filter" window with the configuration details.

Figure 28 Trap Filter dialog box




4 Select/enter appropriately and then click **OK**.

5 Click **Cancel** to retain the previous settings.

6 Click **Close** on the "Filter Traps" console.

## Deleting Trap Filter

This option enables you to delete the trap filter.

### To delete trap filter

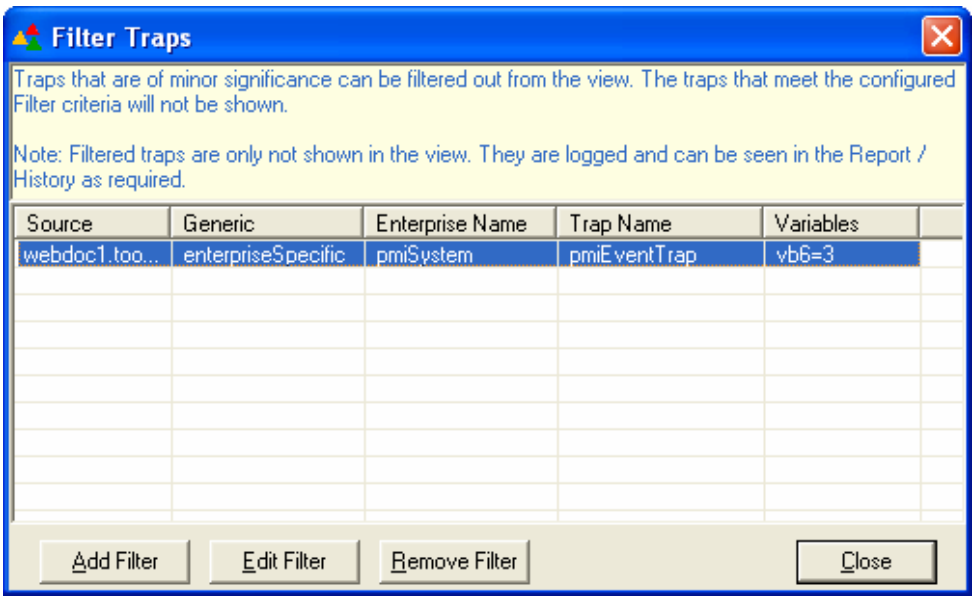
1. From the **Options** menu, choose **Filters**.

   (OR)

   Click **Filters** on the toolbar.

   TrapTracker displays the "Filter Traps" console.

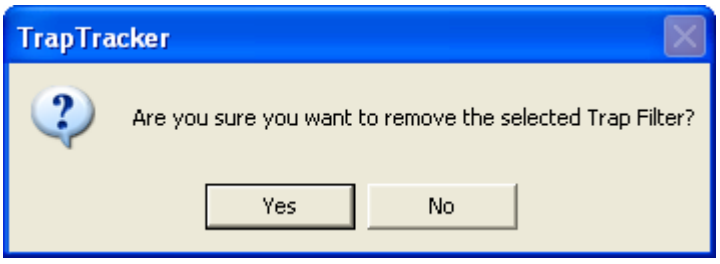
Figure 29 Filter Traps dialog box.




2. Select the filter that you want to delete.
3. Click **Remove Filter**.

   TrapTracker displays the confirmation message box.

Figure 30 Filter Traps remove confirmation message box.



4 Click **Yes** to remove or **No** to retain.

5 Click **Close**.

# Alerts

TrapTracker Manager provides an option to notify the user when traps matching the criteria set are received by the TTW Receiver.

You can associate Alert actions to the Alert configuration. Alert notification mechanisms available are, a beep on the system hosting the TrapTracker Manager, an e-mail to a specified e-mail id, a console message to a specified system, and a custom action like running an EXE or .a BAT job.

A typical example would be the Routers, which are critical to every enterprise. Alerts can be configured to notify the user when the monitored routers' normal functionality goes down.

## Adding Alerts

This option enables you to configure Alerts

### To configure Alerts

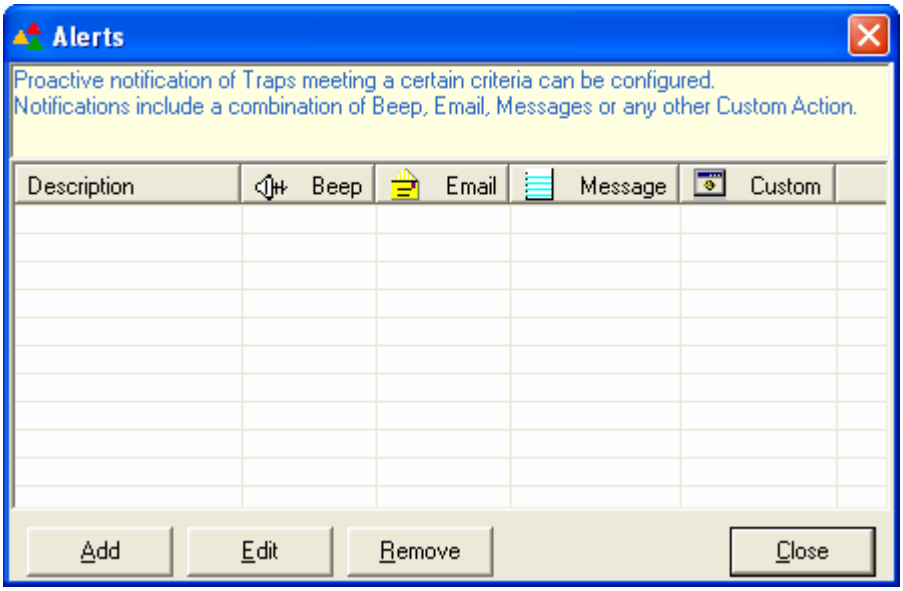
1 From the **Options** menu, choose **Alerts**.

(OR)

Click **Alerts** on the toolbar.

TrapTracker displays the "Alerts" console.

Figure 31 Alerts dialog box




2 Click **Add**.

TrapTracker displays “Alert Configuration” window.

Figure 32 Alert Configuration dialog box

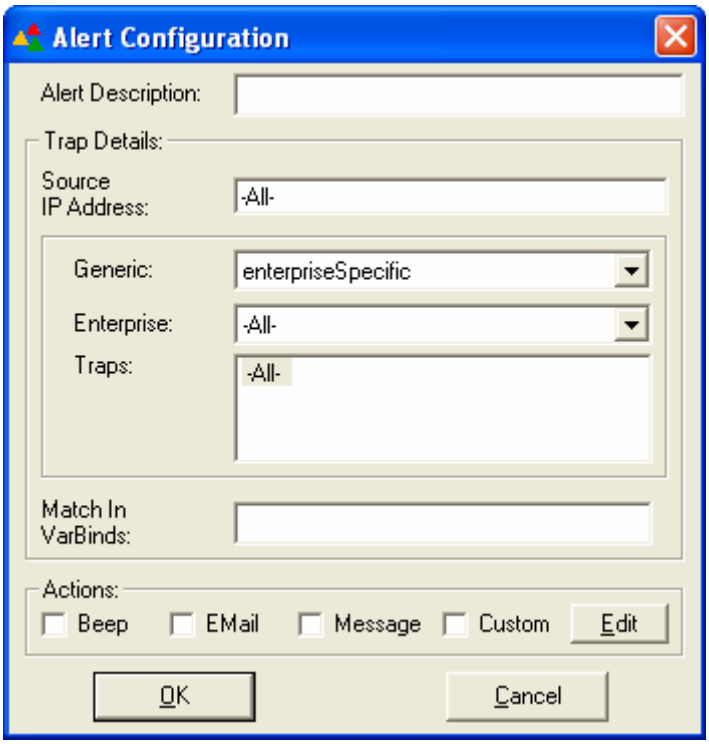


Table 11

| Field | Description |
|---|---|
| **Alert Description** | Type a brief description about the Alert. |
| **Trap Details** | |
| **Source (Name/IP)** | Name or IP address of the source of traps. You can explicitly define the Name/IP address of SNMP compliant devices and monitor traps sent only by those devices. |
| **Generic** | This drop-down list is populated with pre-defined generic traps, which are common to all the SNMP compliant devices. |
| **Enterprise** | This option is enabled only when you choose the enterpriseSpecific option in the Generic drop-down list. This list box is populated with the available compiled MIBs. |
| **Traps** | This is a list box, which is populated with the traps that are available in the enterprise MIB you have chosen. |
| **Match in VarBinds** | To further narrow down your selection criteria, you can enter a variable in this field. The new window you create will display the traps that match the variable you have entered. |
| **Actions** | |
| **Beep** | A beep is heard when the TrapTracker Manager receives the specific configured trap. |
| **Email** | An e-mail is sent to the configured recipient address when the Manager receives the specific configured trap. |
| **Message** | A network message is sent to the console of the configured system when the Manager receives the specific configured trap. |

| Field | Description |
|---|---|
| **Custom** | Certain situations may arise when the administrator needs to perform some customized action on receiving a trap. In this case, the recommended practice is to create a batch file or program and select that batch file or program in the Custom Action screen.<br>On receiving a matching trap, the Manager will execute the batch file or program file selected, and will pass the following as parameters to the batch file or program. The order of parameters is also the same as below.<br>IP Address<br>Enterprise OID<br>Community<br>Generic Trap ID<br>Specific Trap ID<br>A recommended method to write a custom program is to print the received parameters and then build the program. This enables the user to understand the way the parameters are being passed to the program. |
| **Edit** | Edit the previously configured alert notification mechanism. |

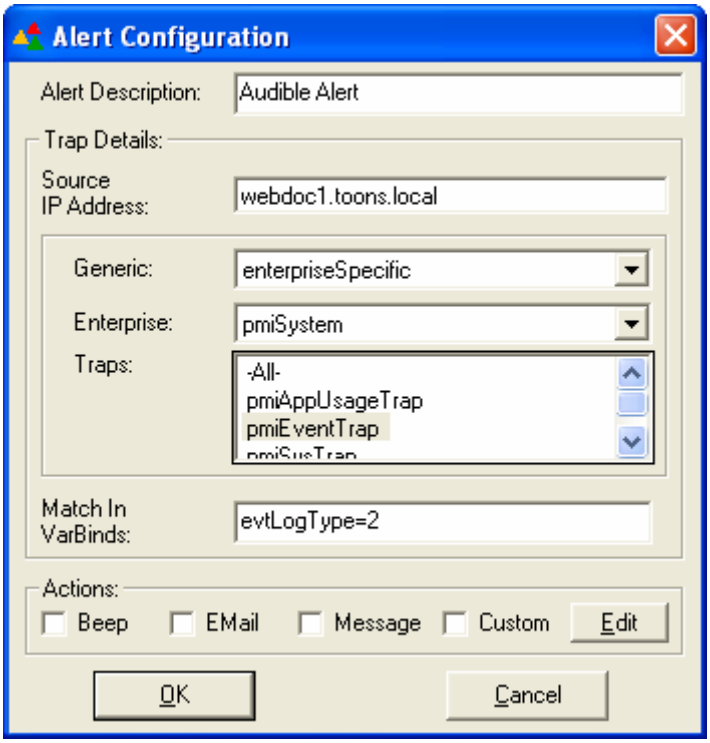
## Configuring Audible Alert action

This option enables you to configure audible Alert action.

### To configure audible Alert action

1. Select/enter appropriate trap details in the “Alert Configuration” window.

Figure 33 Alert Configuration dialog box – set up audible alert




*2* Select the **Beep** check box

TrapTracker displays the "Configure Action – Beep" window.

Figure 34 Configure Action dialog box

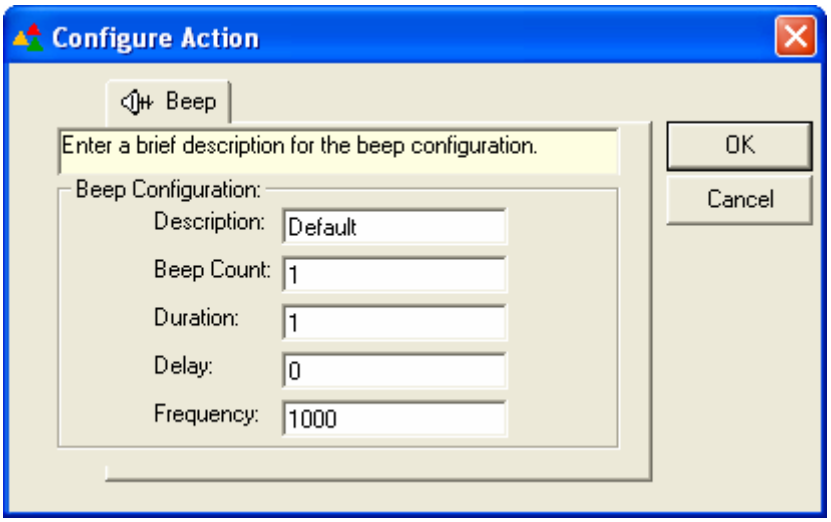



Table 12

| Field | Description |
|---|---|
| **Beep Configuration** | |

| Field | Description |
|---|---|
| **Description** | Type a brief Alert description. |
| **Beep Count** | Type the number of beeps that should be generated on the PC speaker.<br>This field supports numeric datatype only. |
| **Duration** | Type how long the beep should be sustained.<br>This field supports numeric datatype only. |
| **Delay** | Type the time interval to pause between consecutive beeps.<br>This field supports numeric datatype only. |
| **Frequency** | Type the frequency of the beep in Hertz.<br>This field supports numeric datatype only. |

3 Type appropriately in the relevant fields and then click **OK**.

4 Click **OK** on the "Alert Configuration" window.

TrapTracker displays the "Alerts" console with the newly created audible Alert.

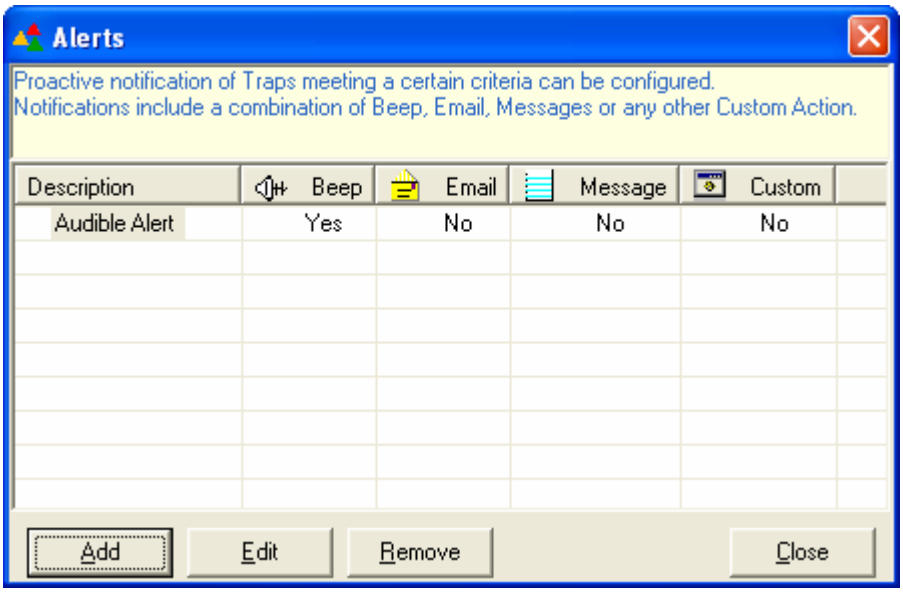



Figure 35 Alerts dialog box with newly added audible alert.

5 Click **Close**.

## Configuring E-mail Alert action

This option enables you to configure e-mail Alert action.

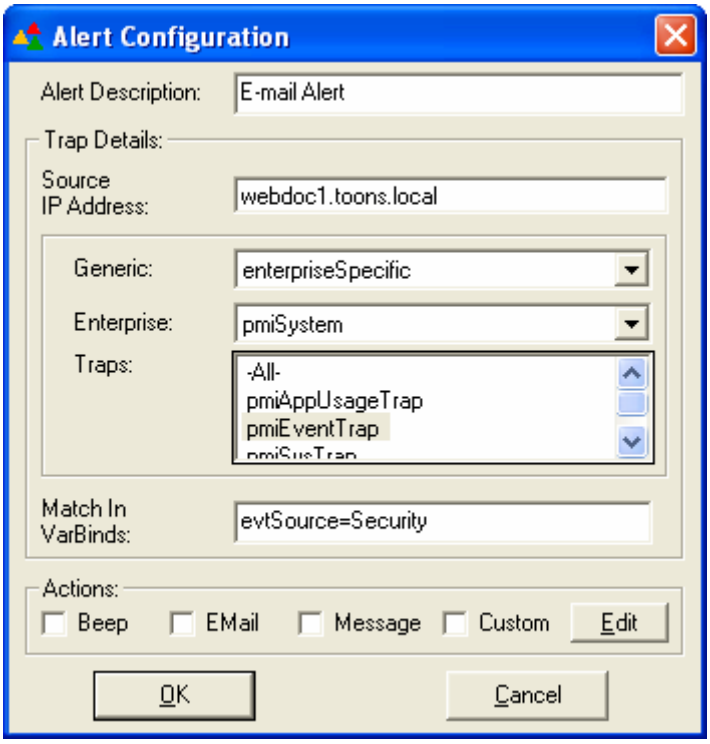
### To configure e-mail Alert action

**Note**

The SMTP server must be accessible from the Console system. That is either the system must be able to access internet or the SMTP server must be reachable over the LAN. Ensure valid email id's are provided in both "To Address" and "From Address".

1 Select/enter appropriate Trap Details in the “Alert Configuration” window.

Figure 36 Alert Configuration dialog box – set up audible alert




2 Select the **Email** check box.

TrapTracker displays the “Configure Action – Email” window.

Figure 37 Configure Action dialog box

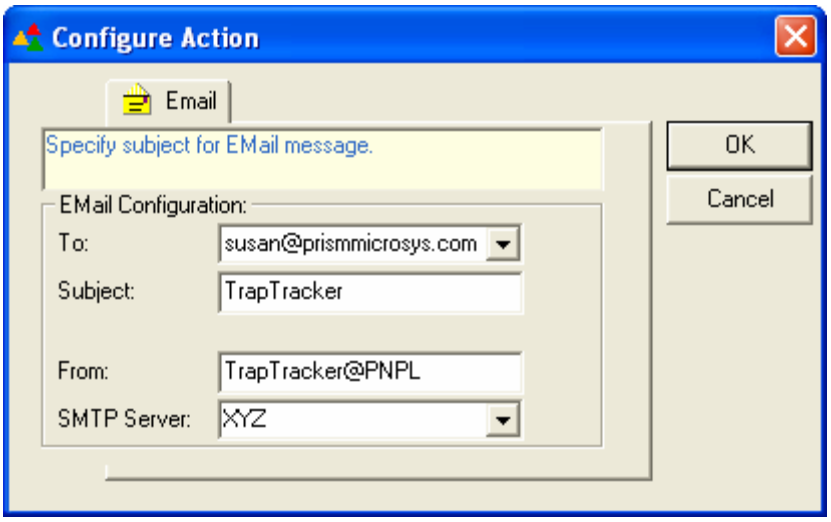



Table 13

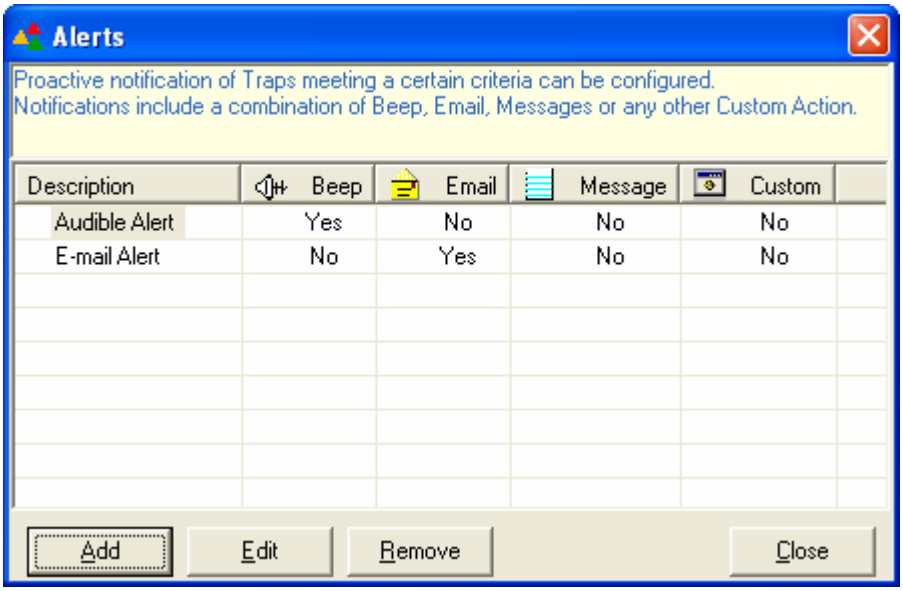
| Field | Description |
|---|---|
| **Email Configuration** | |
| **To** | Type a valid recipient e-mail address. |
| **Subject** | Subject of the e-mail. |
| **From** | Type a valid sender e-mail address. |
| **SMTP Server** | Type the SMTP Server name or IP address or select it from the drop-down list. |

3 Select/enter appropriately in the relevant fields and then click **OK**.

4 Click **OK** on the “Alert Configuration” window.

TrapTracker displays the “Alerts” console with the newly created e-mail Alert.

Figure 38 Alerts dialog box with newly added audible alert.




5 Click **Close**.

## Configuring Console Message Alert action

This option enables you to configure console message Alert action.

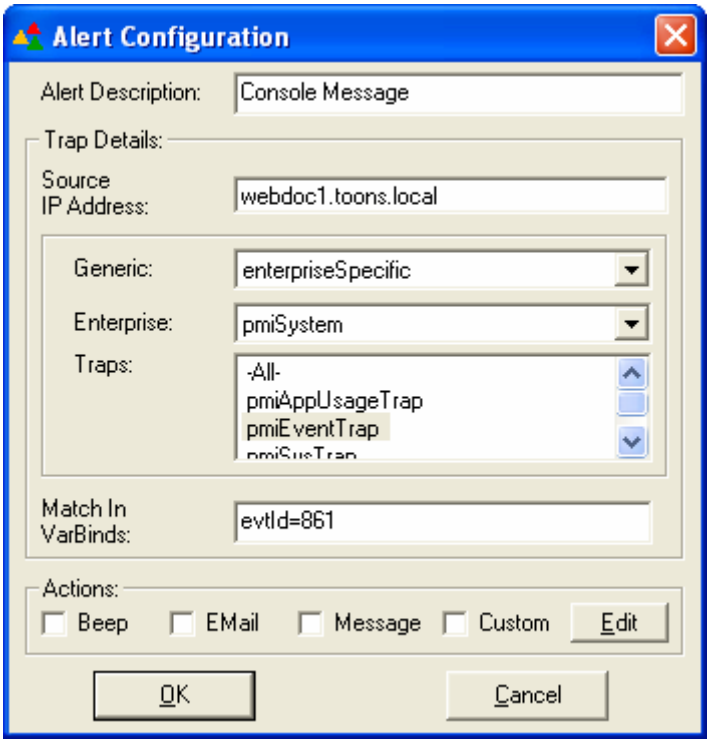
### To configure console message Alert action

**Note**

Messenger service must be available to configure this action.

1 Select/enter appropriate Trap Details in the “Alert Configuration” window.

Figure 39 Alert Configuration dialog box – set up console message alert.




*2* Select the **Message** check box.

TrapTracker displays the "Configure Action – Message" window.

Figure 40 Configure Action dialog box.

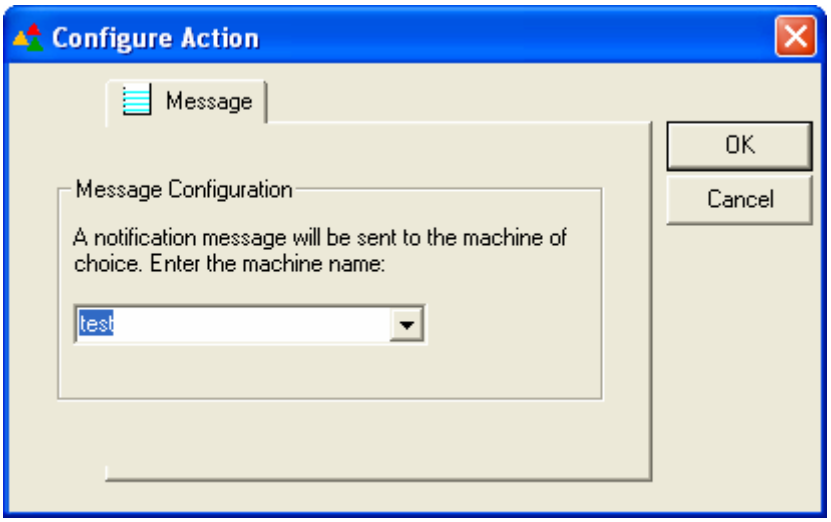



Table 14

| Field | Description |
|---|---|

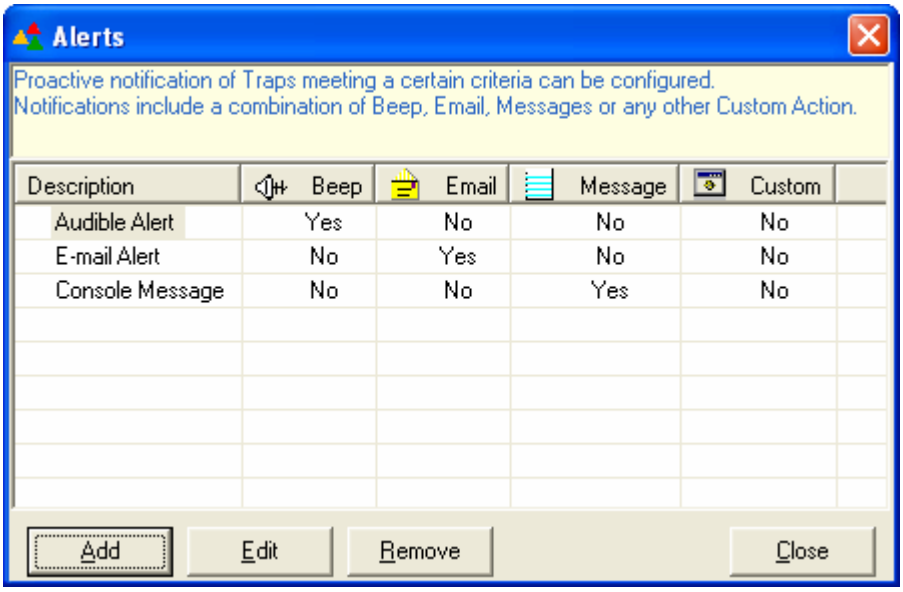
| Field | Description |
|---|---|
| **Message Configuration: A notification message will be sent to the machine of your choice.** | |
| | Type the name or IP address of the machine or you can select from the drop-down list. |

3 Type appropriately in the relevant fields and then click **OK**.

4 Click **OK** on the "Alert Configuration" window.

TrapTracker displays the Alerts console with the newly created console message Alert.

Figure 41 Alerts dialog box with newly added audible alert.




5 Click **Close**.

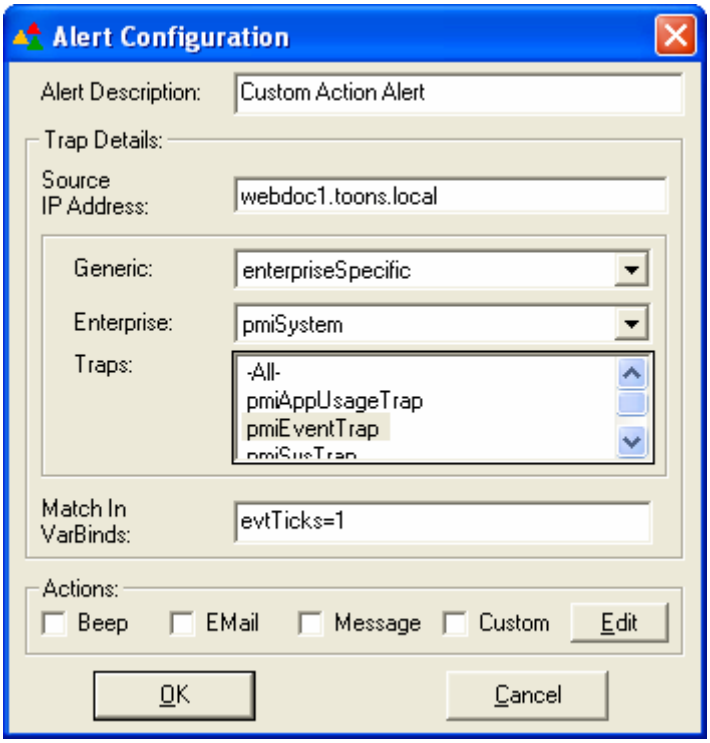
## Executing Custom Alert action

This option enables you to set up a custom Alert action.

### To execute custom Alert action

1 Select/enter appropriate Trap Details in the "Alert Configuration" window.

Figure 42 Alert Configuration dialog box – set up custom action alert.




*2* Select the **Custom** check box.

TrapTracker displays the "Configure Action – Custom" window.

Figure 43 Configure Action dialog box.

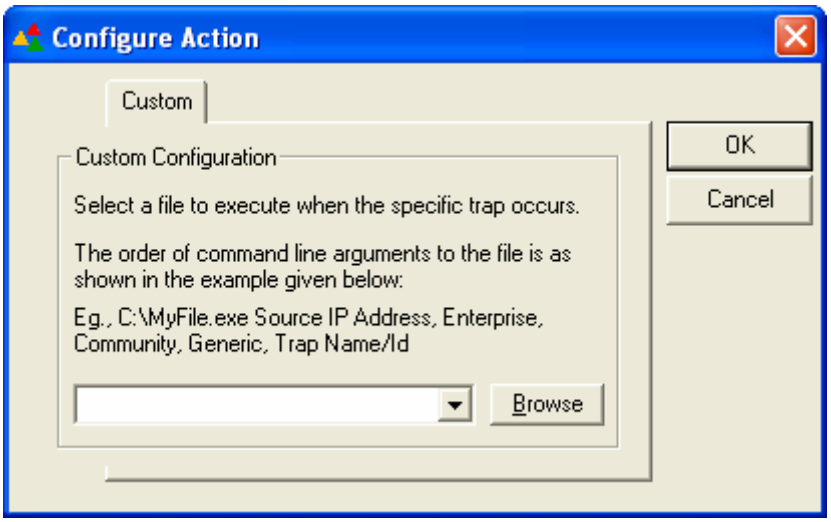



Table 15

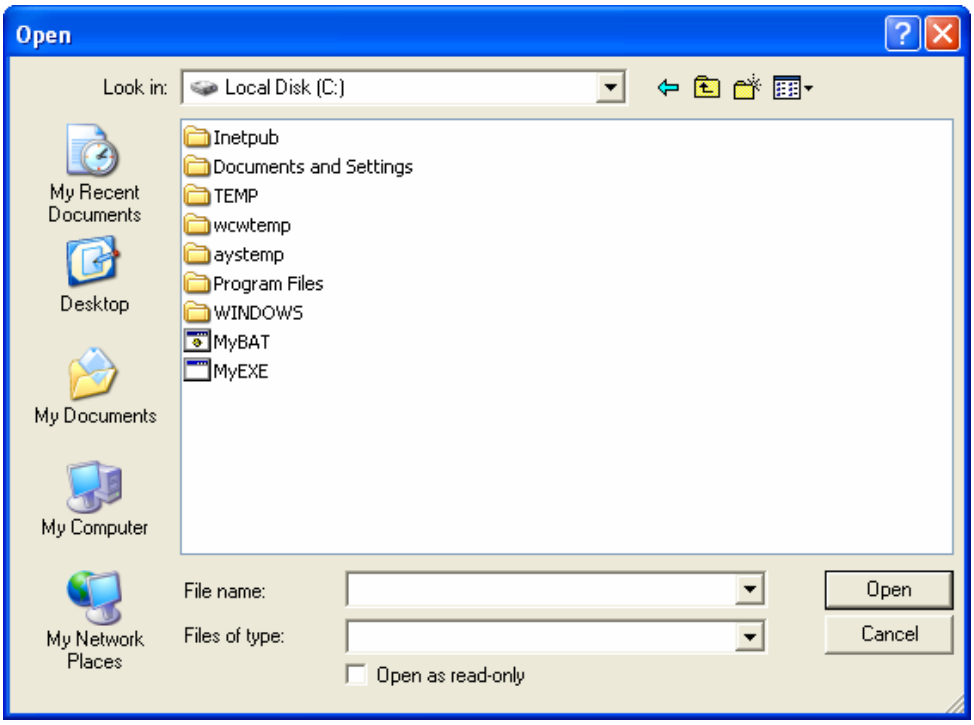
| Field | Description |
|---|---|

| Field | Description |
|---|---|
| **Custom Configuration: Select a file to be executed when a specific trap occurs.** | |
|  | Select a file from the drop-down list or click **Browse** to browse for the file. |

*3* Click **Browse**.

TrapTracker displays the "Open" window.

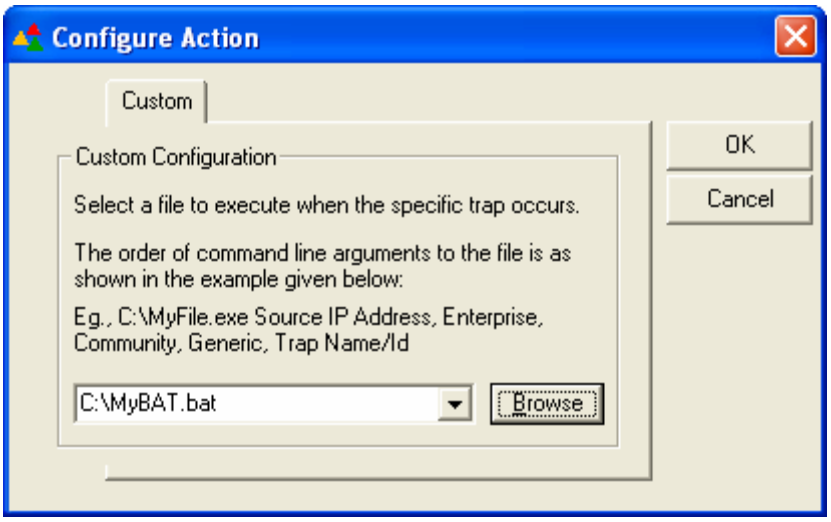
Figure 44 Open dialog box.




*4* Select a custom file and then click **Open**.

TrapTracker updates the Configure Action window with the path of the custom file.

Figure 45 Configure Action dialog box.

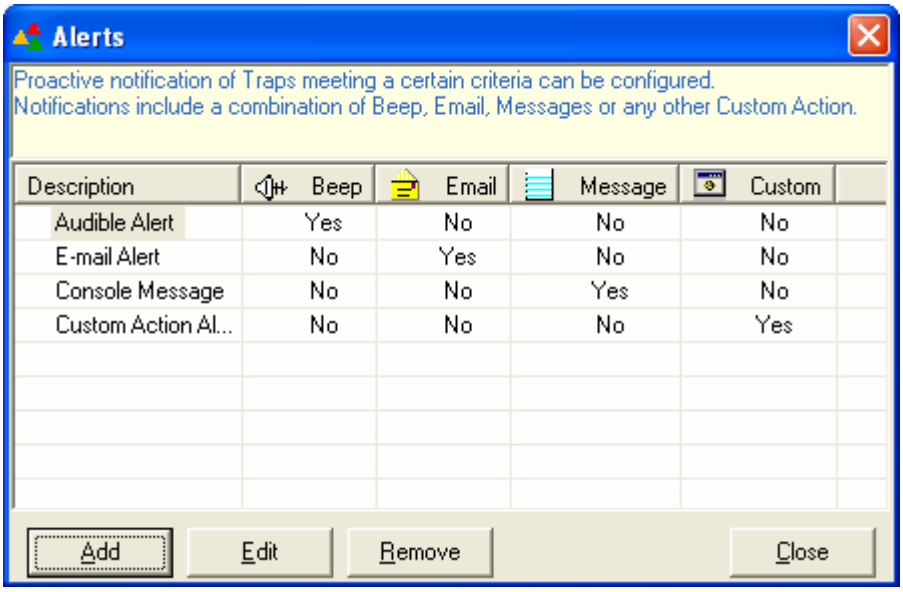



5 Click **OK**.

6 Click **OK** on the "Alert Configuration" window.

TrapTracker displays the Alerts console with the newly created custom Alert.

Figure 46 Alerts dialog box with newly added custom action alert.




7 Click **Close**.

## Modifying Alert Configuration Details

This option enables you to modify Alert and Alert action configuration settings.

## To modify Alert and Alert action configuration

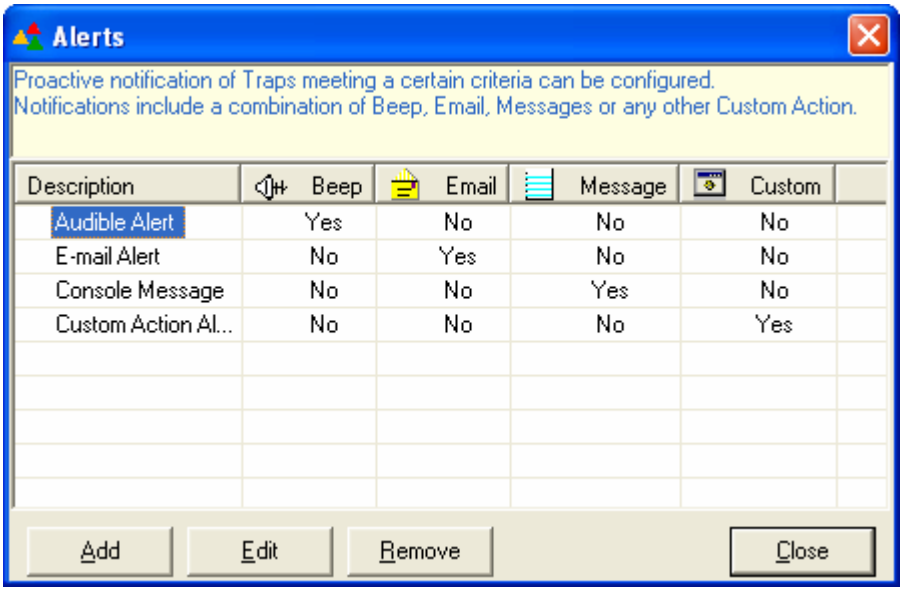
1 From the **Options** menu, choose **Alerts**.

(OR)

Click **Alerts** on the toolbar.

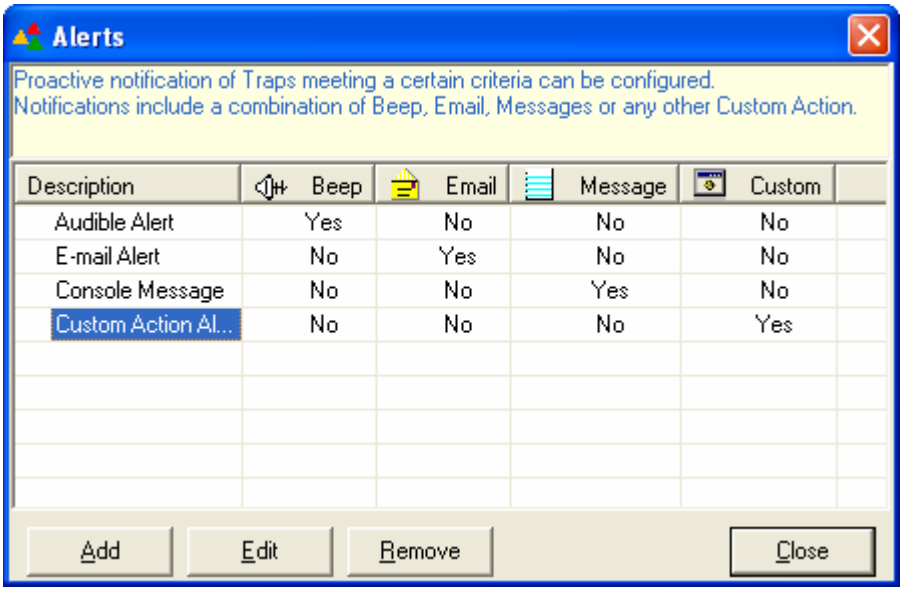
TrapTracker displays the “Alerts” console.

Figure 47 Alerts dialog box.




2 Select the Alert that you want to modify.

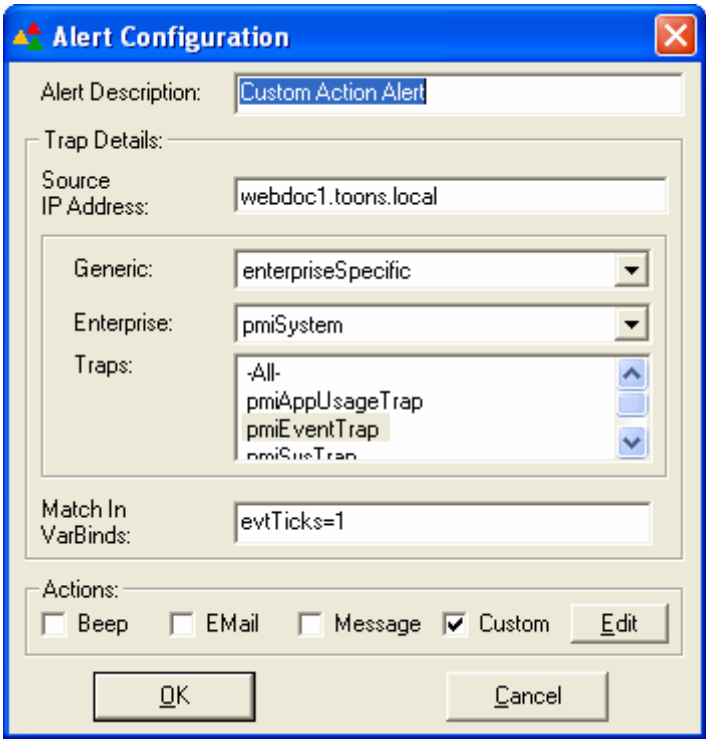
Figure 48 Alerts dialog box.



3 Click **Edit**.

TrapTracker displays “Alert Configuration” window with the configuration settings set earlier.

Figure 49 Alert Configuration dialog box.

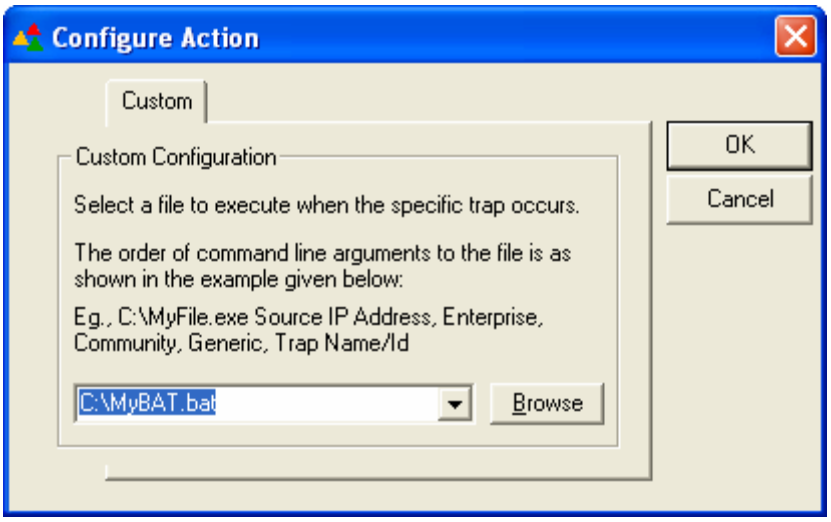



4 Select/enter appropriately in the relevant fields under Trap Details.

5 Click **Edit** to modify the Alert actions.

TrapTracker displays “Configure Action – Custom” window.

Figure 50 Configure Action dialog box.

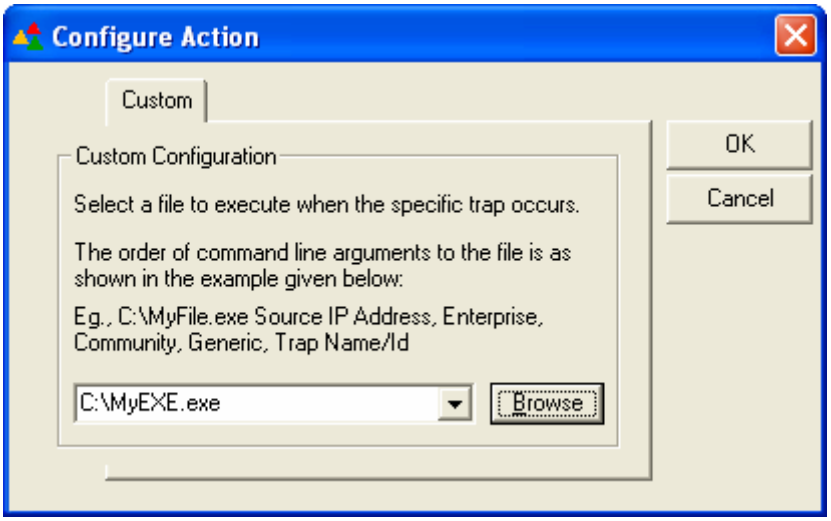



6 Click **Browse** to select a custom file from "Open" window.

7 Select a file that you want to execute and click **Open**.

TrapTracker updates the "Configure Action – Custom" window with the path of the new file chosen.

Figure 51 Configure Action dialog box.




8 Click **OK**.

9 Click **OK** on the "Alerts Configuration" window.

10 Click **Close**.

## Deleting Alert Configuration Details

This option enables you to delete Alert configuration settings.

### To delete Alert configuration setting

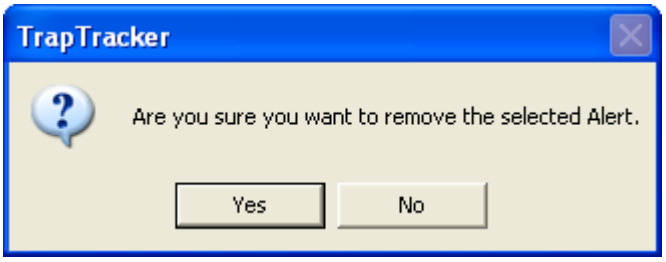
1. From the **Options** menu, choose **Alerts**.

   (OR)

   Click **Alerts** on the toolbar.
2. Select the Alert that you want to delete.
3. Click **Remove**.

   TrapTracker displays the confirmation message box.

Figure 52 Remove alert – confirmation dialog box.




4. Click **Yes** to delete or **No** to retain.
5. Click **Close**.

# Chapter 3
# Reports & Categories

*In this chapter, you will learn how to:*

- Create Categories
- Modify Categories
- Delete Categories
- Add Trap Details to a Category
- Modify Trap Details in a Category
- Delete Trap Details from a Category
- Export Category
- Import Category
- Generate Historical Reports

# Managing Trap Categories

TrapTracker has categories feature, where a set of related traps are grouped together as your needs dictate into a category. Whenever a trap in a category occurs, the Reports will highlight that category with an appropriate trap severity indicator. This feature enables easy identification of critical traps that are generated by a device that belongs to a specific category.

The steps involved in using categories are:

1. Creating a Category
2. Adding traps to that Category
3. Defining the Trap Severity for each trap added
4. Monitoring the status of the Category by running Reports

**Example:**

Take the case of a category that monitors all critical traps generated from a Router. This category will contain all the important traps that are generated by the router. Each trap will have its own severity level. When the user monitors the status of this category using the **Reports** console, the category will be highlighted with the color code of the trap with the maximum severity that has been generated by the router.

If the category contains 2 traps,

1. First trap with Trap ID = 1.3.6.1.4.1.618, Source = 192.244.88.11, Severity=Major
2. Second trap with Trap ID = 1.3.6.8.4.1.724, Source = 192.244.88.11, Severity=Critical

If the device (192.244.88.11) had generated a trap with trap ID = 1.3.6.1.4.1.618, then the Reports would indicate the presence of a Major Severity trap in the device (192.244.88.11).

If the device (192.244.88.11) had generated a trap with trap ID = 1.3.6.1.4.1.618 and another trap with trap ID = 1.3.6.8.4.1.724, then the Reports would indicate the presence of a Critical Severity trap from device 192.244.88.11.

Therefore, the most significant severity level takes precedence over its peers.

If either of these two traps were not generated, then the Reports would make no indication for this category.

## Creating Trap Category

This option enables you to create a trap category.

## To create a trap category

1 From the **Reports** menu, choose **Categories**.

(OR)

Click **Trap Categories** on the toolbar.

TrapTracker displays the “Manage Categories” console.

Figure 53 Manage Categories dialog box.

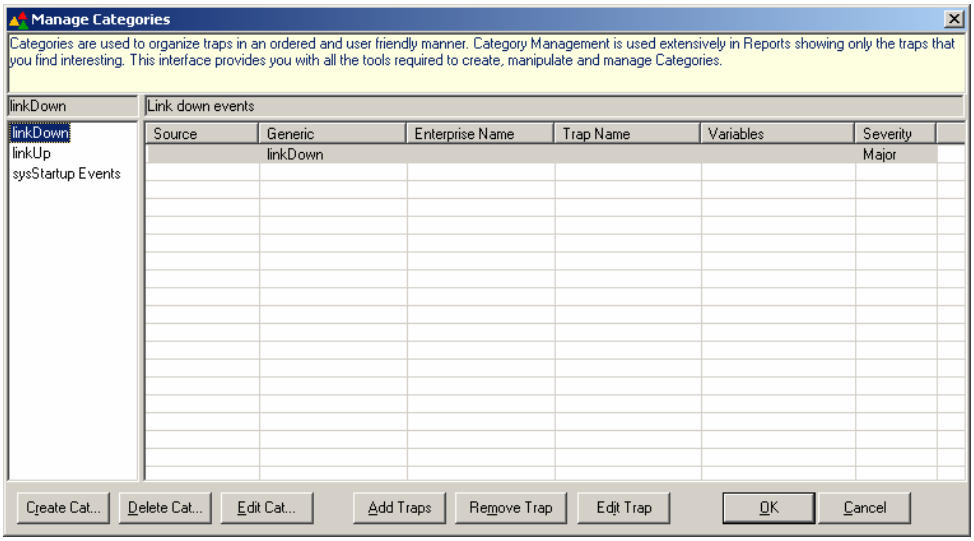



Table 16

| Click | To |
|---|---|
| **Create Category** | Create a category. |
| **Delete Category** | Delete a category. |
| **Edit Category** | Modify category details. |
| **Add Traps** | Add trap details to a category. |
| **Remove Traps** | Delete traps details from a category. |
| **Edit Trap** | Modify trap details in a category. |

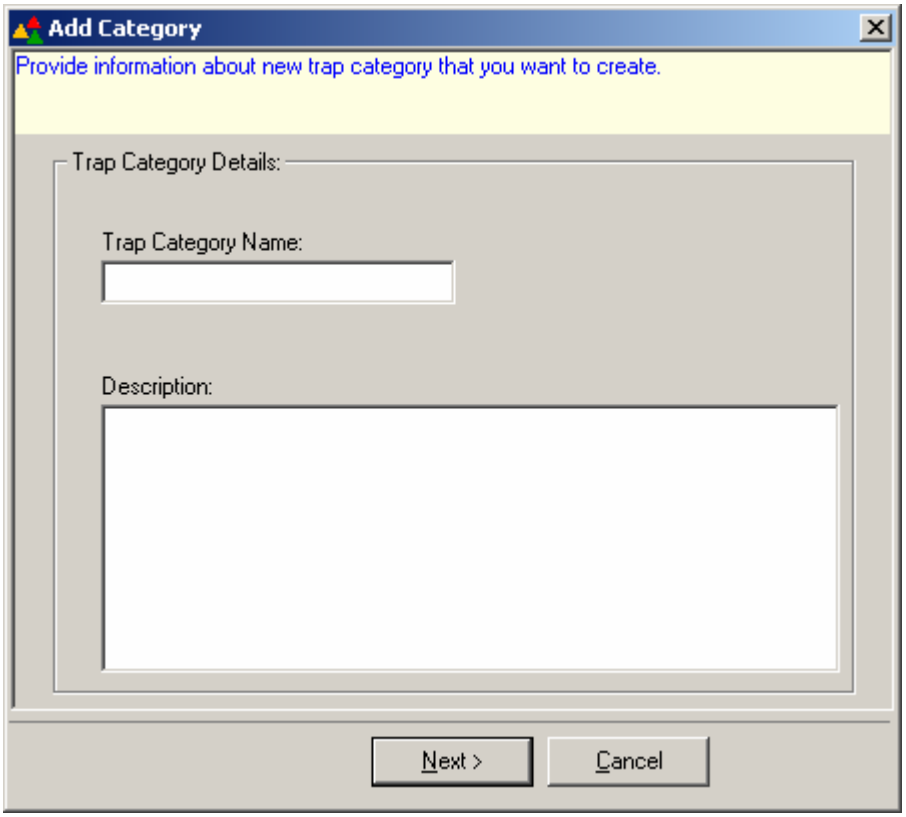
**Note**

Pre-defined categories are **linkDown**, **linkUp** and **sysStartup Events**.

2 Click **Create Category**.

TrapTracker displays the “Add Category” window.

Figure 54 Add Category dialog box.

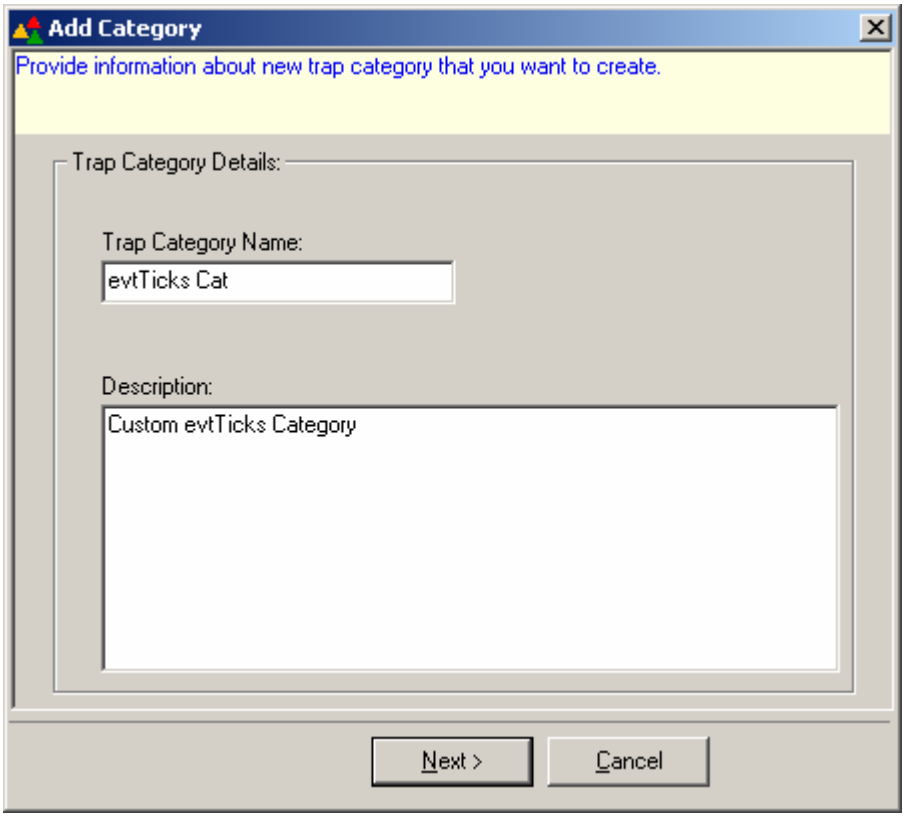



*3* Type an appropriate name in the **Trap Category Name** field. This field is mandatory. If you skip this, TrapTracker displays the TrapTracker message box with appropriate message.

Figure 55 TrapTracker message box.


*4* Type a brief description of the Category in the **Description** field. This field is not mandatory.

Figure 56 Add Category dialog box




***5*** Click **Next** > to add trap details.

TrapTracker displays the “Add Trap Detail” window.

Figure 57 Add Trap Detail dialog box.

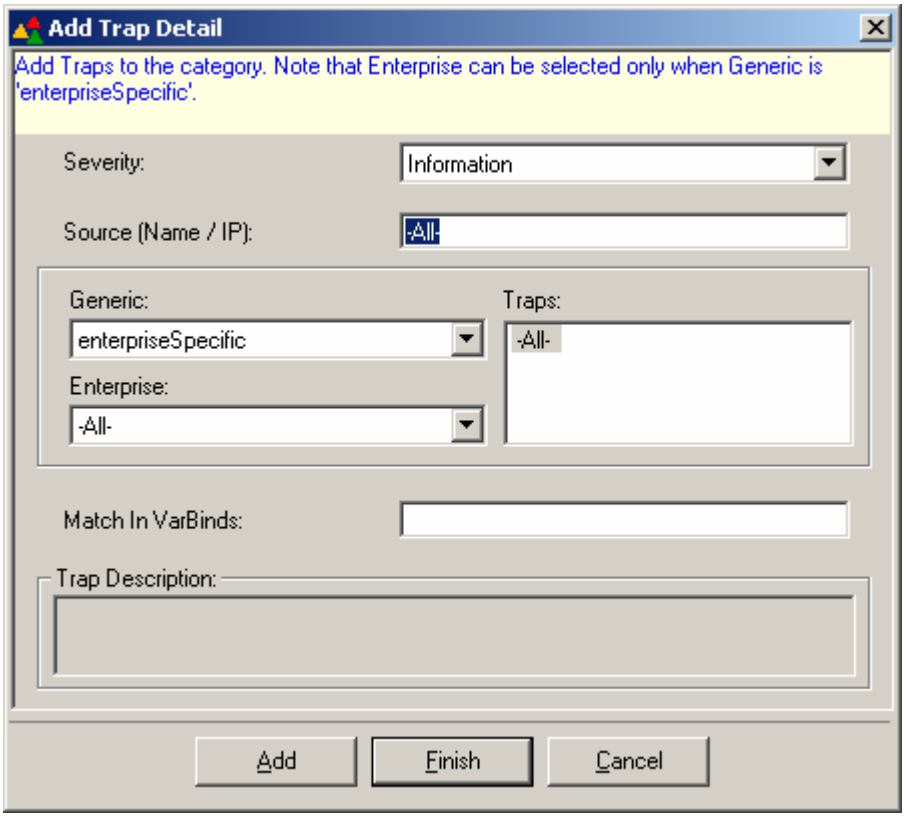
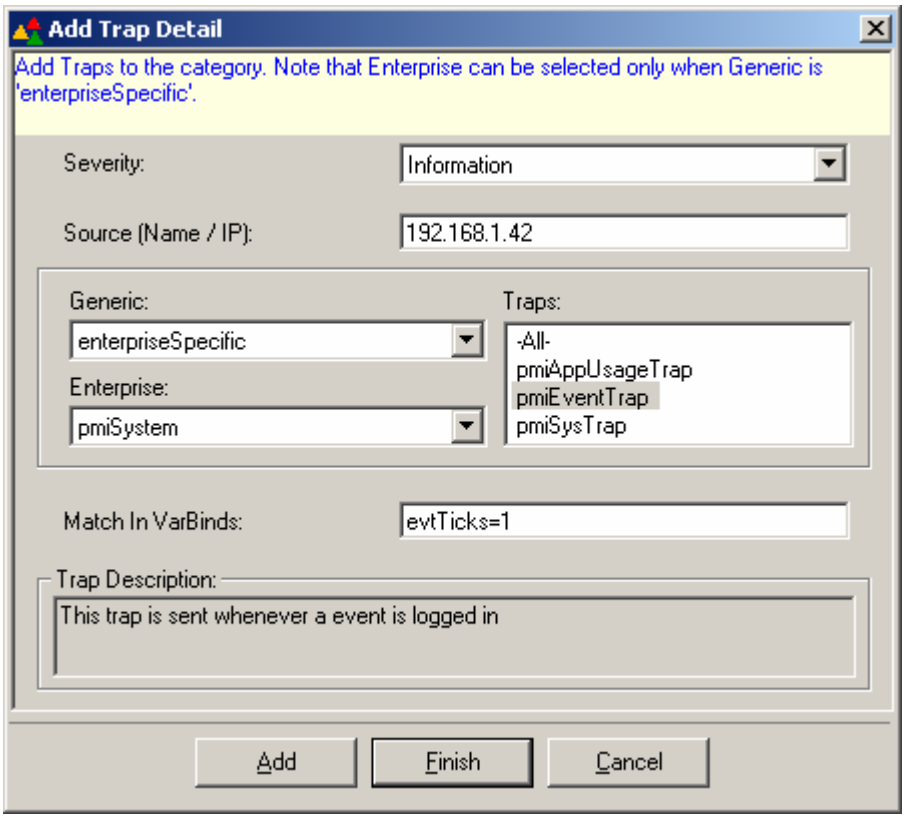



Table 17

| Field | Description |
|---|---|
| **Severity** | Select a severity level from this drop-down list. Available options are -All-, Clear, Minor, Information, Major, Warning, and Critical. |
| **Source (Name/IP)** | Type the name or IP address of the source of traps. |
| **Generic** | This drop-down list is populated with pre-defined generic traps, which are common to all SNMP compliant devices. |
| **Enterprise** | This option is enabled only when you choose the enterpriseSpecific option in the Generic drop-down list. This list box is populated with the available compiled MIBs. |
| **Traps** | This list box is populated with the traps that are available in the enterprise MIB you have chosen. |
| **Match in Varbinds** | To further narrow down your selection criteria, you can enter the variables associated with the chosen MIB, in this field. |
| **Trap Description** | The trap description defined in the MIB is displayed in this display box. |

6 Select/enter appropriately in the relevant fields.

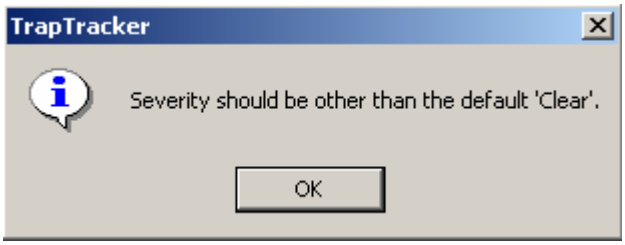
Figure 58 Add Trap Detail dialog box




7 Click **Add** to add trap details.

Had you selected the default severity level, TrapTracker displays the TrapTracker message box with appropriate message.

Figure 59 TrapTracker message box.



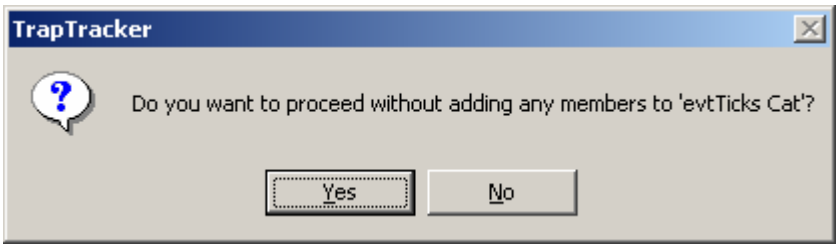

8 Click **OK**.

9 Choose a severity level other than the default value and click **Add**. This way you can add n number of trap details to your category.

10 Click **Finish** to save the category details.

If you click **Finish** without adding trap details, TrapTracker displays the TrapTracker message box with appropriate message.

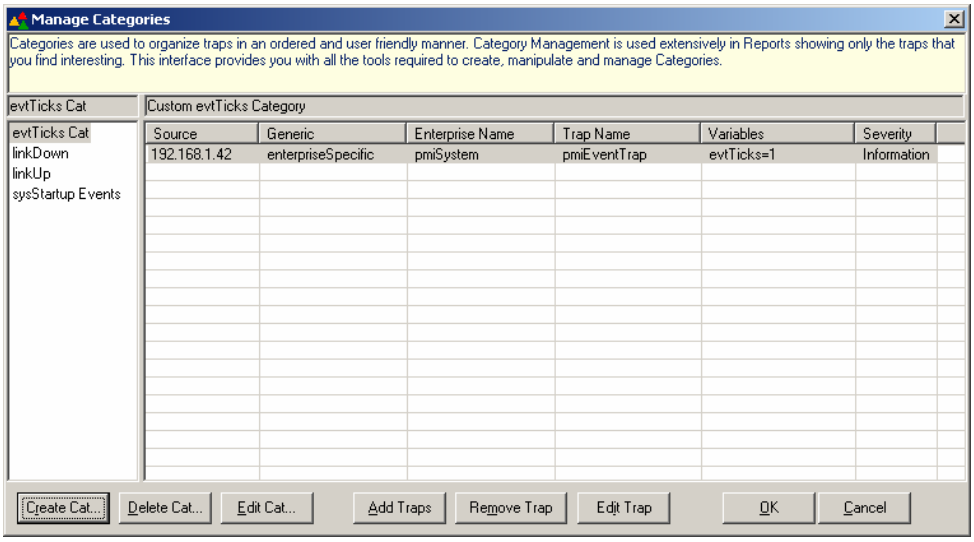
Figure 60 TrapTracker message box.




*11* If you click **Yes**, the category is created without any trap details. Later, you can add trap details to this category.

TrapTracker displays the "Manage Categories" console with the newly created category.

Figure 61 Manage Categories dialog box




*12* Click **OK**.

## Monitoring Custom Categories

### To monitor custom categories

*1* From **Reports** menu, choose **Report/History**.

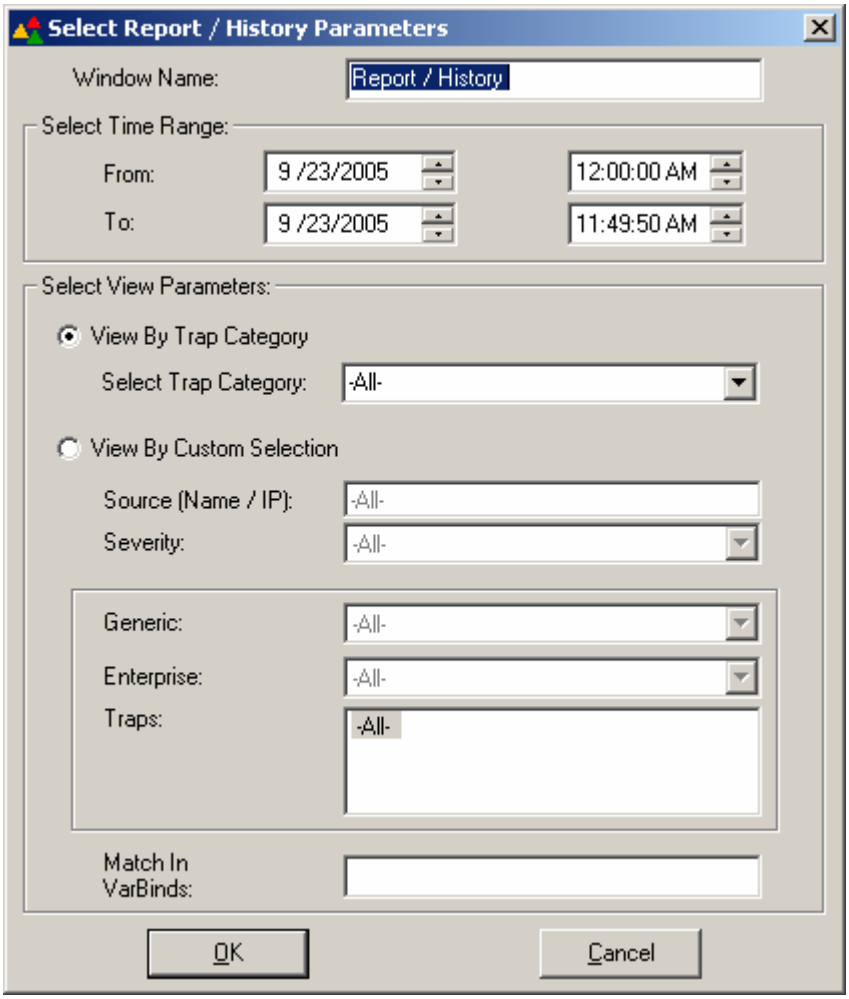
(OR)

Click 

on the toolbar.

TrapTracker displays the "Select Report / History Parameters" window.

Figure 62 Select Report/History Parameters dialog box.



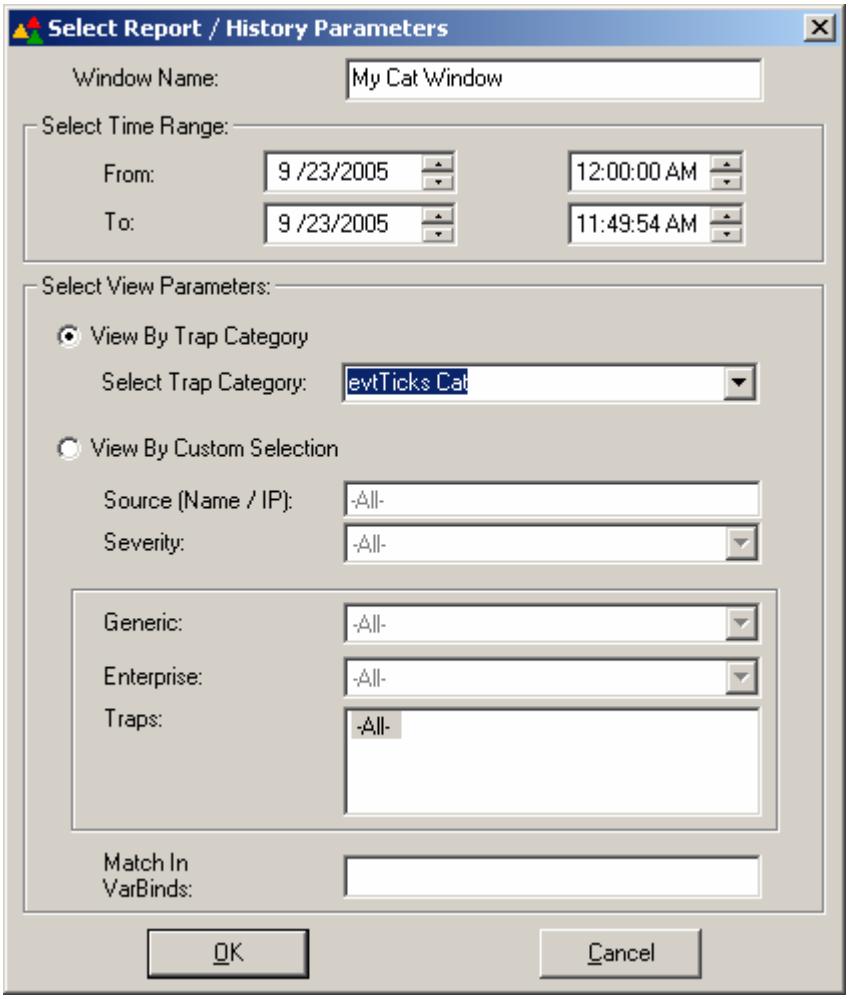

2 Type the name of the window in the **Window Name** field.

3 Select **From**, **To** Date and Time.

4 Select the **View By Trap Category** option.

5 Select the category from the **Select A Trap Category** drop-down list.

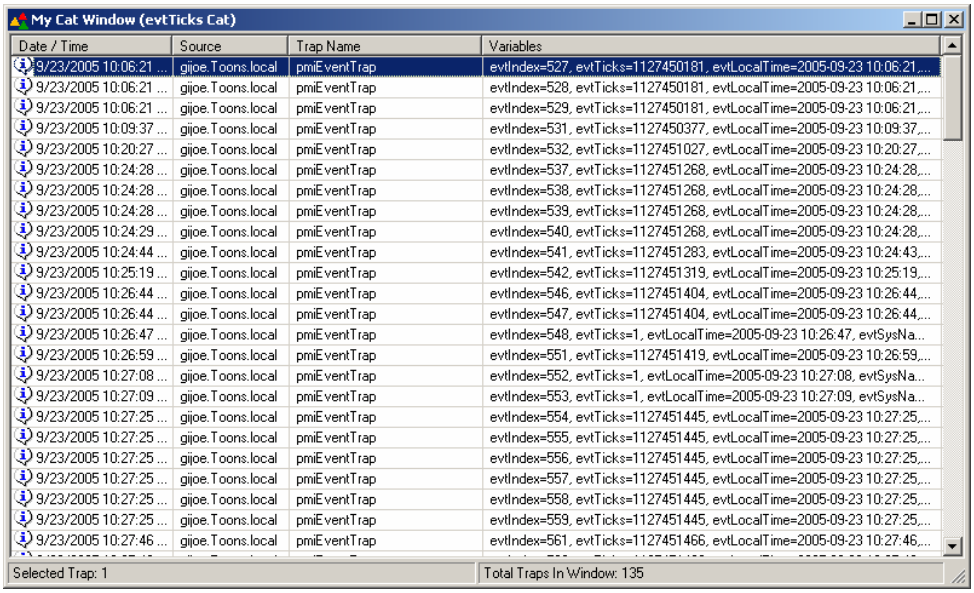
Figure 63 Select Report/History Parameters dialog box.




*6* Click **OK**.

TrapTracker displays the trap details of the selected category in a new window.

Figure 64 New window displaying selected category details




***7*** Close the window by clicking  at the upper-right corner of the window.

## Modifying Category Details

This option enables you to modify category details.

### To modify category details

***1*** From the **Reports** menu, choose **Categories**.

(OR)

Click **Trap Categories** on the toolbar.

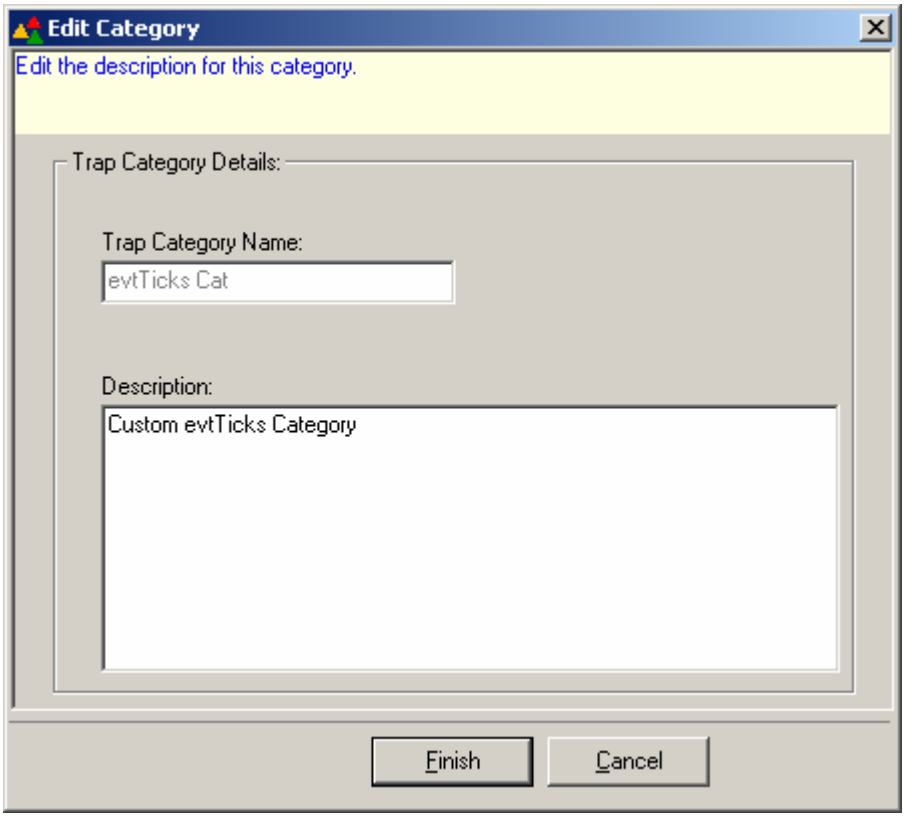
***2*** On the left pane, select the category that you want to modify and then click **Edit Category**.

(OR)

Double-click the category that you want to modify.

TrapTracker displays the “Edit Category” window.

Figure 65 Edit Category dialog box




3 You can edit **Description** alone and not the **Trap Category Name**.

4 Click **Finish**.

5 Click **Cancel** to retain the previous configuration settings.

6 Click **OK**.

## Deleting Category

This option enables you to delete a category.

### To remove a category

1 From the **Reports** menu, choose **Categories**.

(OR)

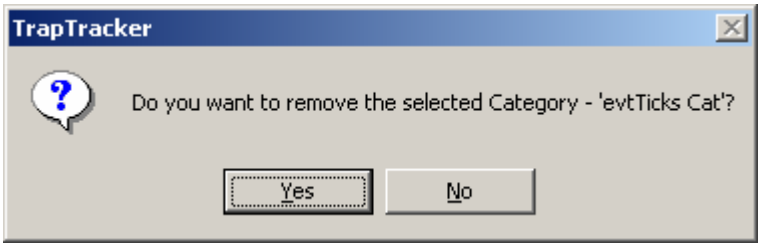
Click **Trap Categories** on the toolbar.

2 On the left pane, select the category that you want to delete.

3 Click **Delete Category**.

TrapTracker displays the confirmation message box.

Figure 66 TrapTracker message box




4 Click **Yes** to delete the selected category or **No** to retain.

5 Click **OK**.

# Adding Trap Details to a Trap Category

This option enables you to add trap details to a trap Category.

## To add trap details to a trap category

1 From the **Reports** menu, choose **Categories**.

(OR)

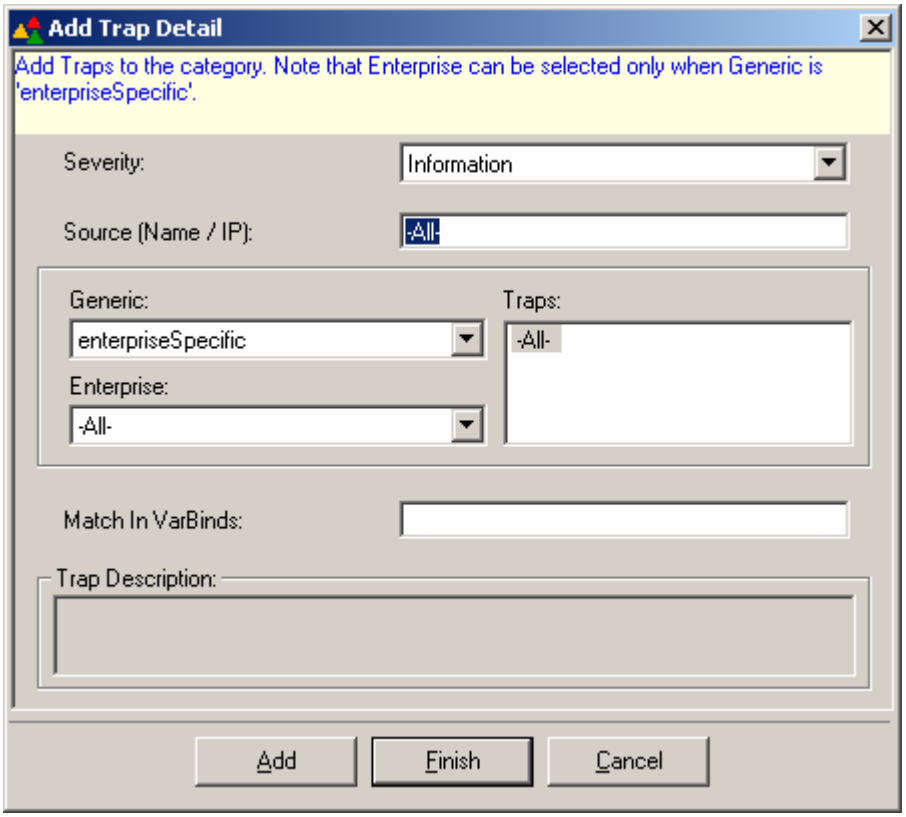
Click **Trap Categories** on the toolbar.

2 On the left pane, select a category.

3 Click **Add Traps**.

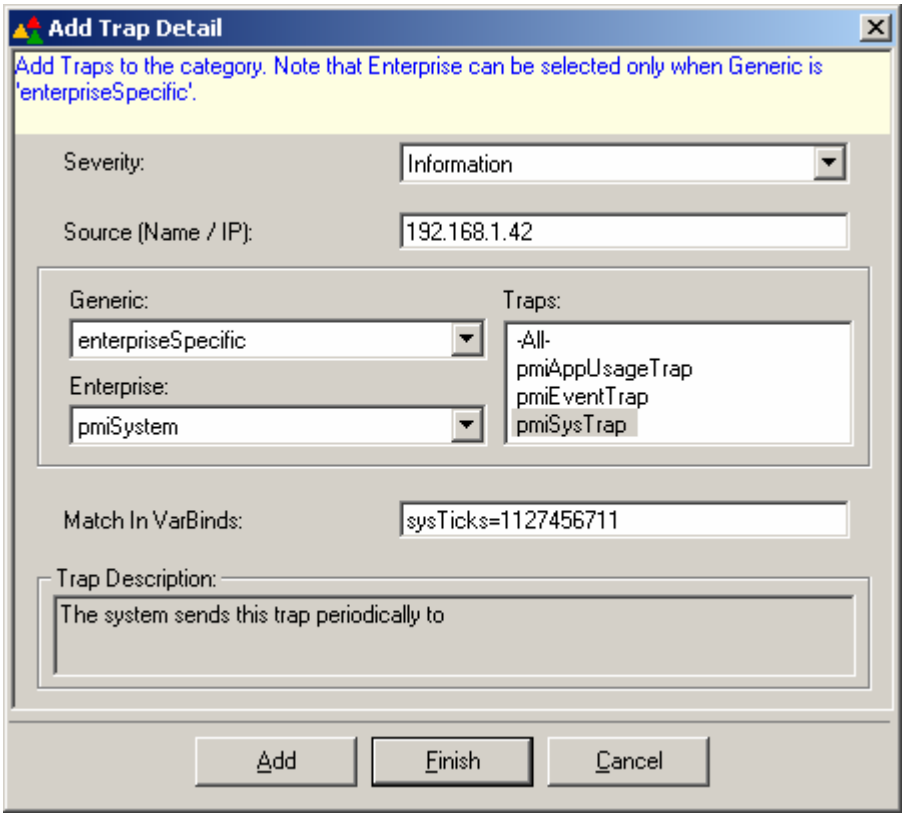
TrapTracker displays the “Add Trap Detail” window.

Figure 67 Add Trap Detail dialog box




*4* Select/enter appropriately in the relevant fields.

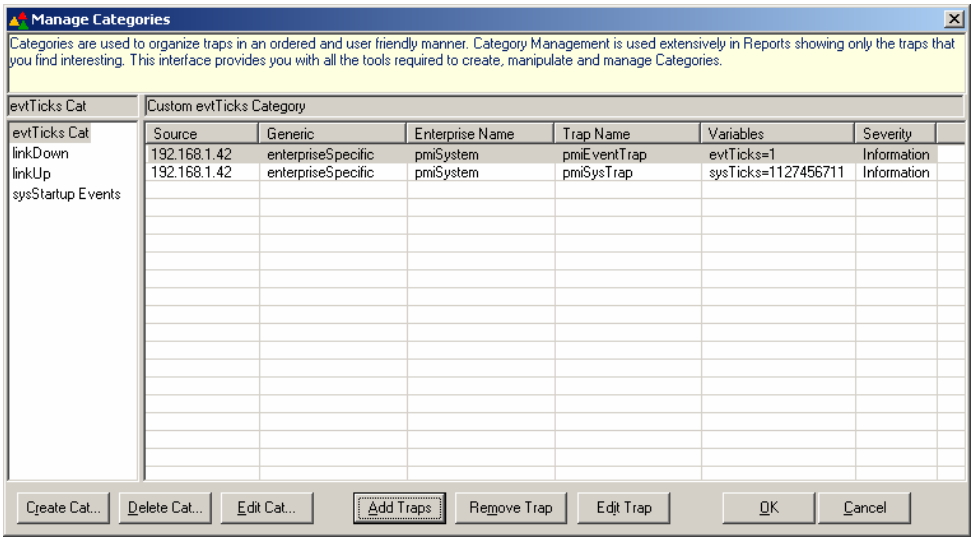
Figure 68 Add Trap Detail dialog box




***5*** Click **Add** and then click **Finish**.

TrapTracker displays the “Manage Categories” console with the newly added trap details.

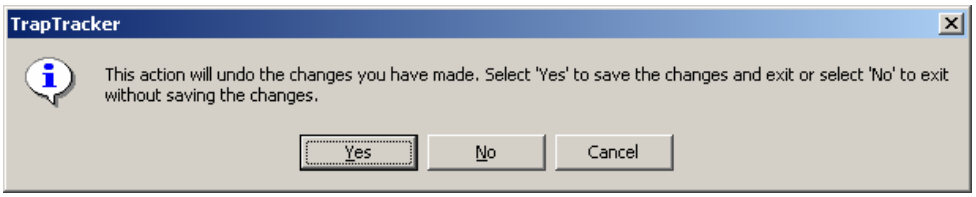
Figure 69 Manage Categories dialog box with newly added trap details.



6 Click **OK**.

If you click **Cancel**, TrapTracker displays the TrapTracker message box with appropriate message.

Figure 70 TrapTracker message box.




7 Click **Yes** to save the changes and exit or click **No** to exit without saving.

8 Click **Cancel** and then click **OK** on the Manage Categories console.

# Modifying Trap Details in a Trap Category

This option enables you to modify trap details in a trap category.

## To modify trap details in a trap category

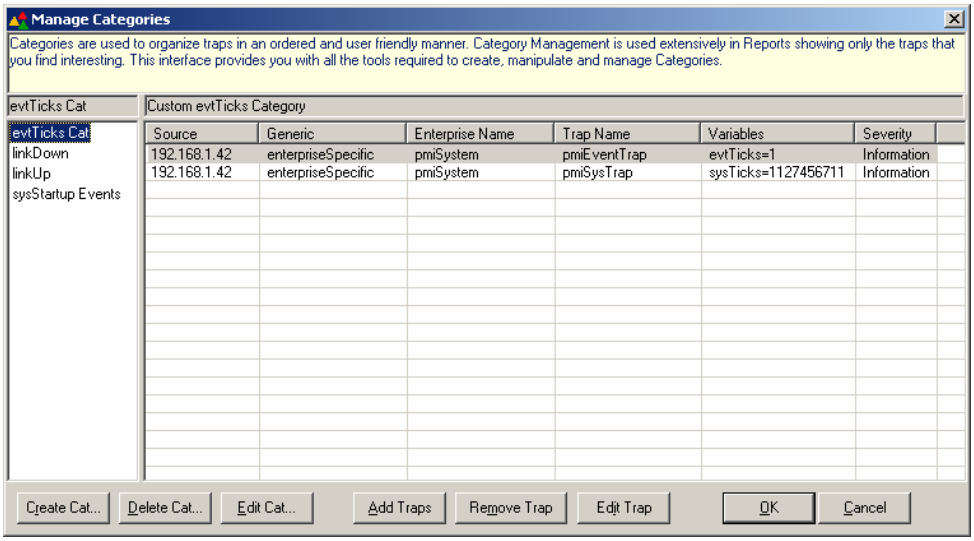
1 From the **Reports** menu, choose **Categories**.

(OR)

Click **Trap Categories** on the toolbar.

2 On the left pane, select the category.

Figure 71 Select Category.



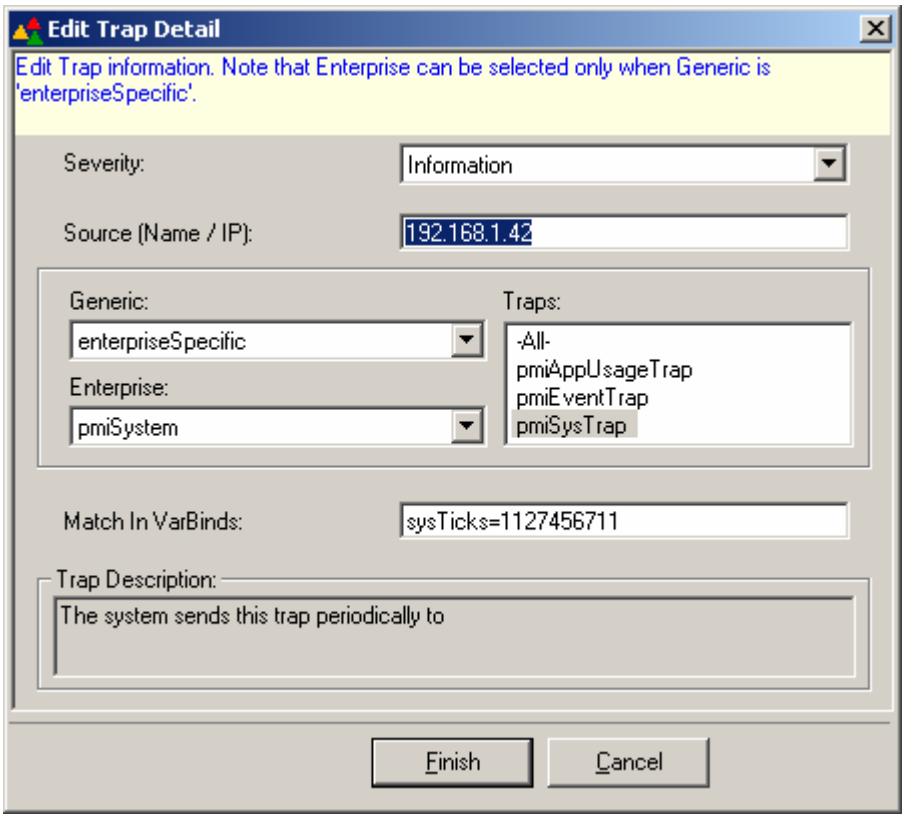
3 On the right pane, double-click the trap detail you want to modify.

(OR)

Select the trap detail on the right pane and then click **Edit Trap**.

TrapTracker displays “Edit Trap Detail” window.

Figure 72 Edit Trap Detail dialog box




4 Select/enter appropriately in the relevant fields and click **Finish**.

5 Click **OK**.

# Deleting Trap Details from a Trap Category

This option enables you to delete trap details from a trap Category.

## To delete trap details from a trap category

1 From the **Reports** menu, choose **Categories**.

(OR)

Click **Trap Categories** on the toolbar.

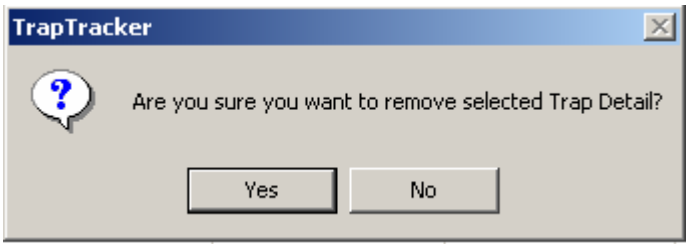
2 On the left pane, select the Category.

3 On the right pane, select the trap detail you want to delete.

4 Click **Remove Trap**.

TrapTracker displays the confirmation message box.

Figure 73 TrapTracker message box.




5 Click **Yes** to delete the selected category or **No** to retain.

6 Click **OK**.

# Import and Export Trap Categories

The category import and export feature is provided to ease the category creation process and to help in category redistribution. . Categories can be exported to a file and these files can be imported by any other systems in an enterprise. This concept is concerned with creating categories in a system and utilizing them on any other systems in an enterprise that may need it.

## Exporting Trap Categories

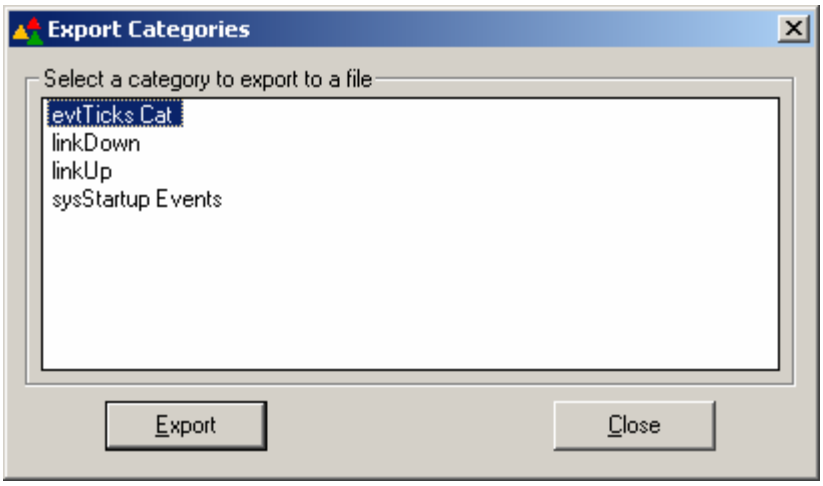
This option enables you to export a trap Category.

### To export a trap category

1 From the **Reports** menu, choose **Export Categories**.

TrapTracker displays the “Export Categories” window.

Figure 74 Export Categories dialog box.

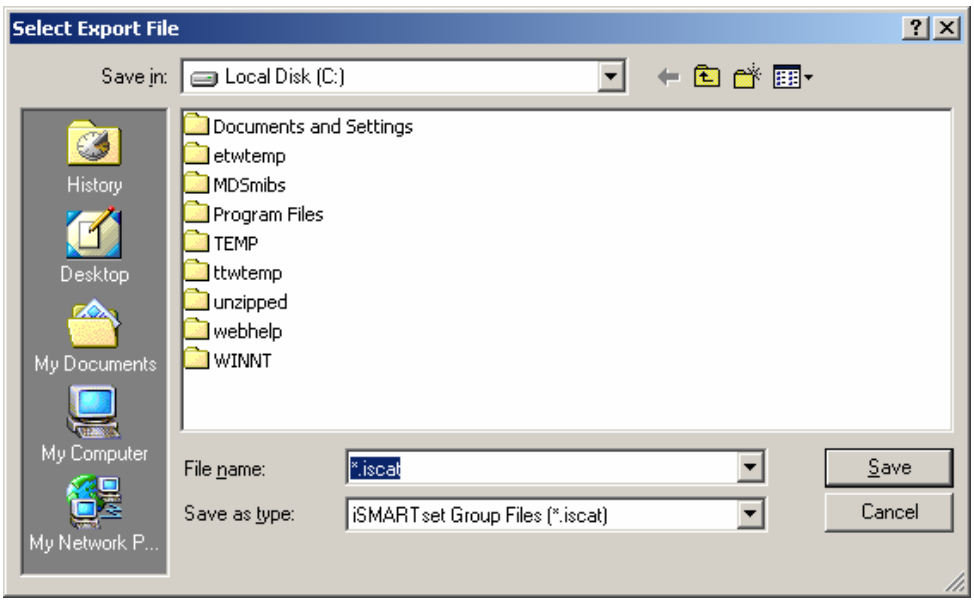



TrapTracker displays all available pre-defined and user-defined categories.

2 Select a single category or hold **Shift** key and select multiple categories and then click **Export**.

TrapTracker displays the “Select Export File” window.

Figure 75 Select Export File dialog box




**Note**

The file extension of the export file is **.iscat**.

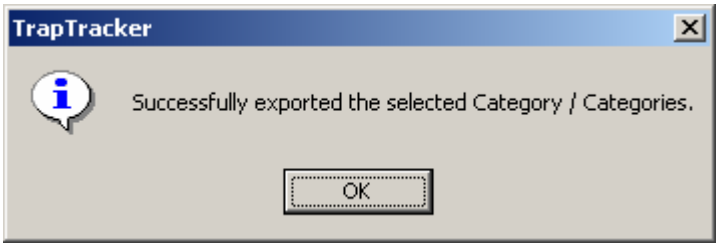
3 Go to the directory where you want to save the export file.

4 Type a name in the **File name** field.

5 Click **Save**.

TrapTracker displays the TrapTracker message box with appropriate status of the export.

Figure 76 TrapTracker message box.




6 Click **OK**.

7 Click **Close** on the “Export Categories” window.

## Importing Trap Categories

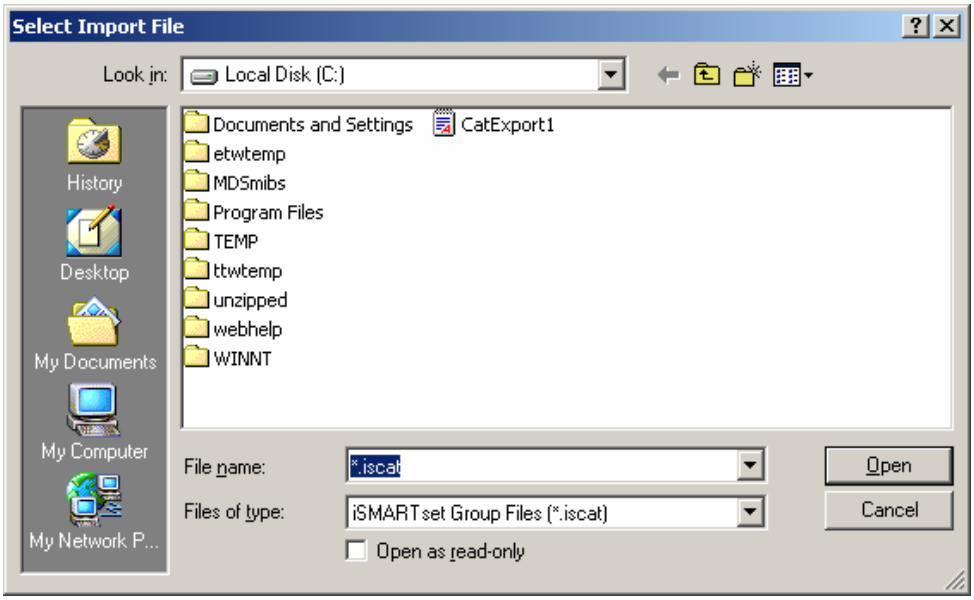
This option enables you to import a trap category.

### To import a trap category

1 From the **Reports** menu, choose **Import Categories**.

TrapTracker displays “Select Import File” window.

Figure 77 Select Import File dialog box




2 Go to the directory where you have stored the category file.

3 Select the file and then click **Open**.

TrapTracker displays TrapTracker message box with appropriate status of the import.

Figure 78 TrapTracker message box.


4 Click **OK**.

5 From the **Reports** menu, choose **Categories** to view the imported categories.

# Reports

The Reports feature presents a very simplified picture of all the trap activities in your enterprise or any specific device. This feature is designed to help you obtain a high-level perspective about the health of a critical device. Using a combination of pre-defined categories and user-defined categories, the Reports feature provides you a quick insight into the trap activities that have occurred in a specific time frame.

User-defined categories are populated in the **Select a Trap Category** list box in the **Select / History Parameters** dialog box. Using a combination of Alerts, Categories and Reports, you can easily isolate the devices and resolve issues.

Reports present the data in two formats:

1. Based on categories.
2. Based on custom trap selection criteria.

## Generating Reports

This option enables you to generate reports.

### To generate reports

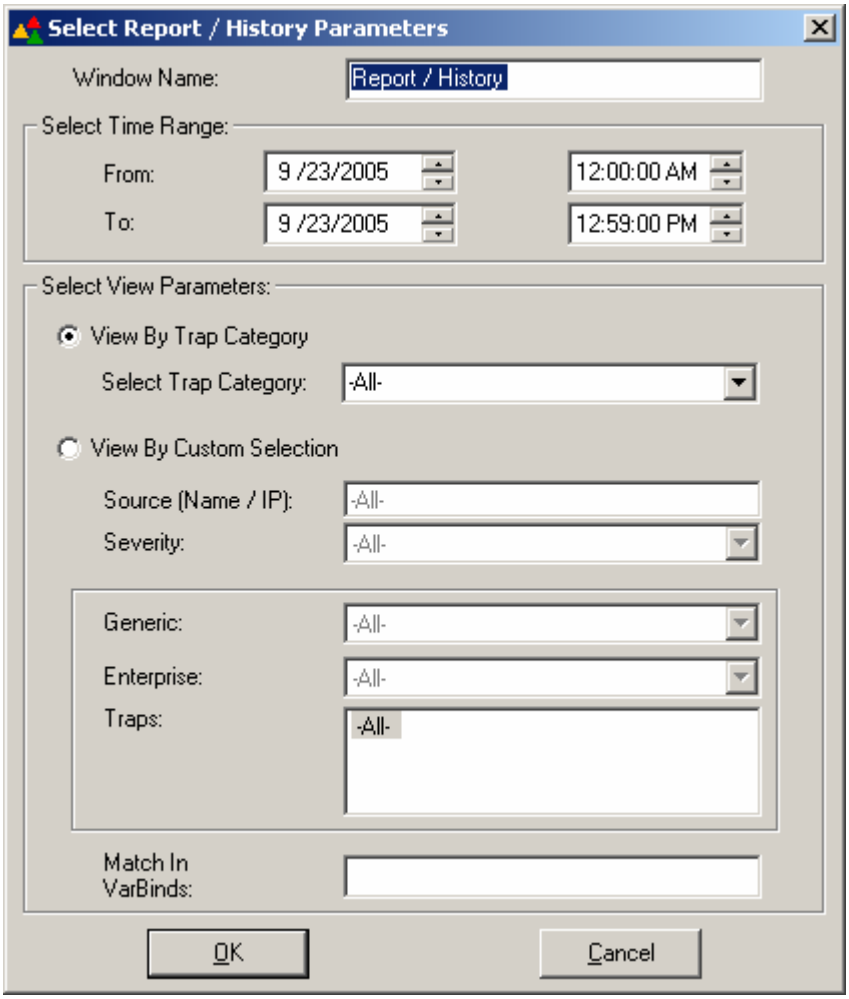
1. From the **Reports** menu, choose **Report/History**.

   (OR)

   Click 
   
   on the toolbar.

   TrapTracker displays “Select Report / History Parameters” window.

Figure 79 Select Report/History Parameters dialog box.




2 Type the window name in the **Window Name** field.

3 Select the **From** and **To** date and time.

## VIEW TRAP BY TRAP CATEGORY

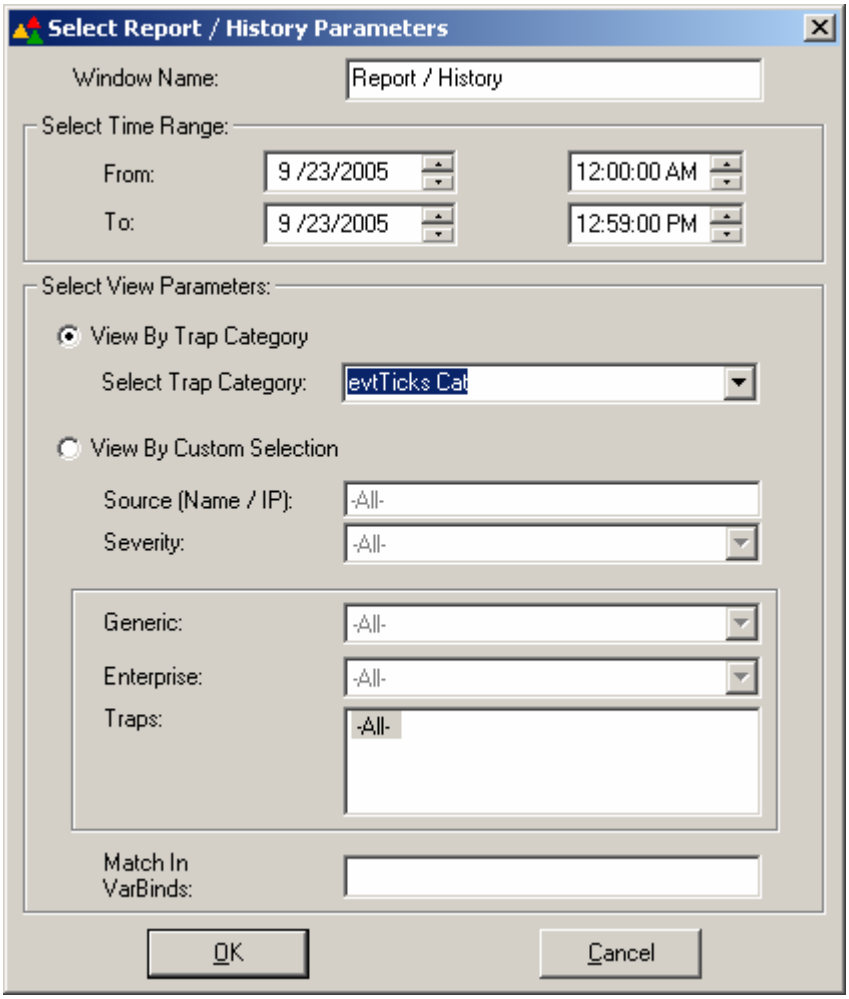
This option enables you to view traps by trap category.

### To view trap by Trap Category

1 Select the **View By Trap Category** option.

2 Select a category from the **Select A Trap Category** drop-down list.

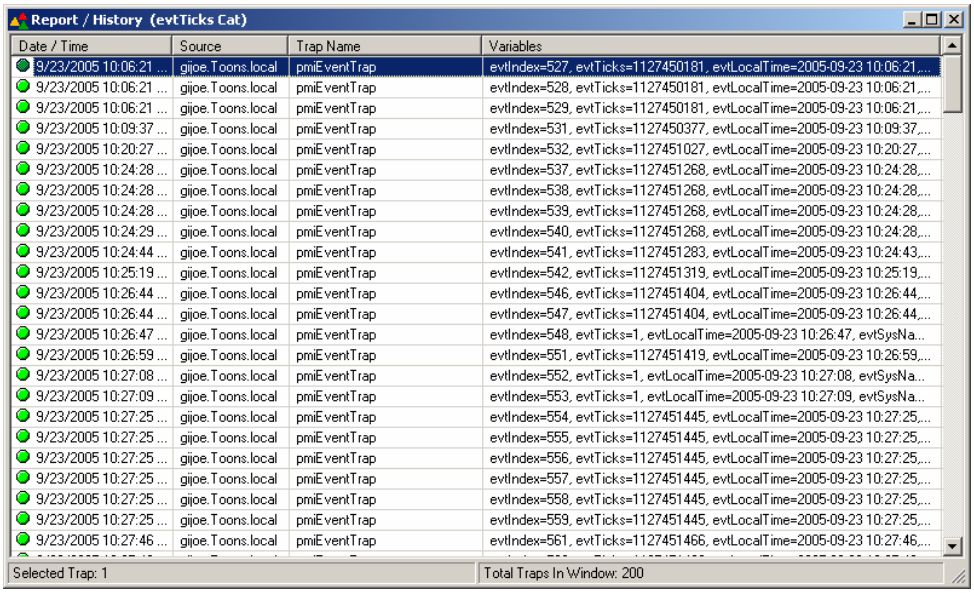
Figure 80 Select Report/History Parameters dialog box.




*3* Click **OK**.

TrapTracker displays the trap details of the selected category in a new window.

Figure 81 New window displaying selected category details.




4 Double-click a trap to view details.

5 Click  at the upper-right corner of the “Reports / History (evtTicks Cat)” window to close.

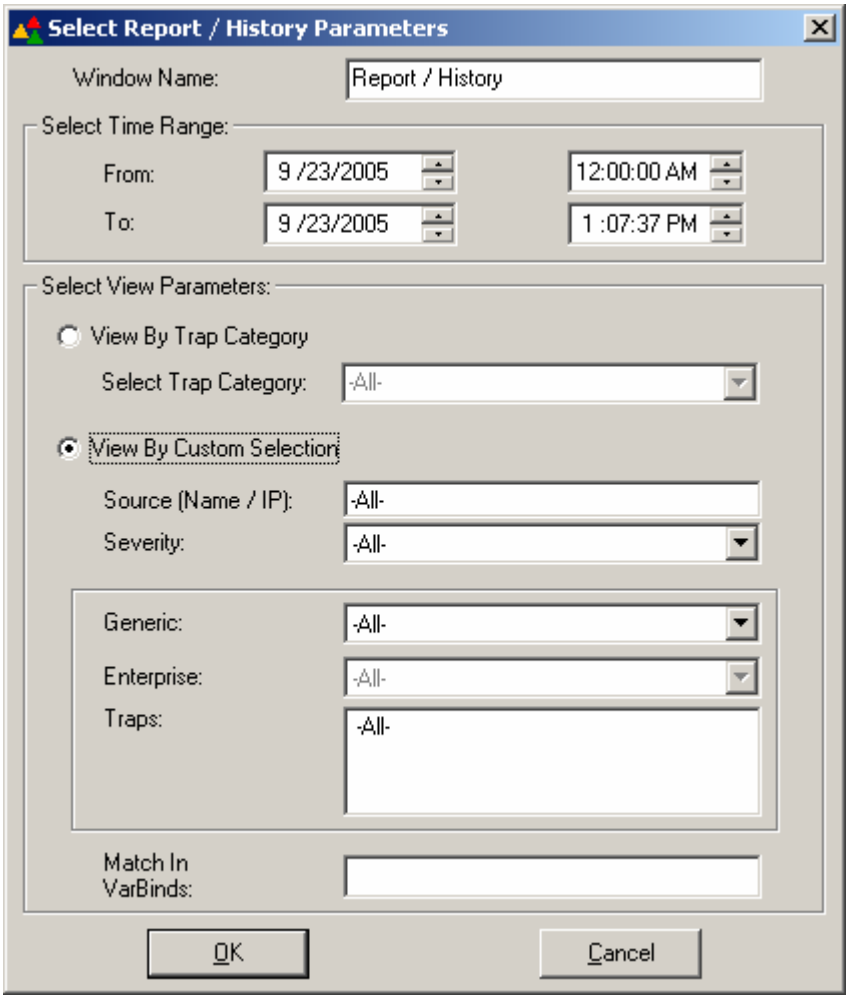
## VIEW TRAP BY CUSTOM SELECTION

This option enables you to view traps by custom selection.

### To view trap by custom selection

1 Select the **View By Custom Selection** option.

Figure 82 Select Report/History Parameters dialog box.




2 Type the name of IP address of the source in the **Source [Name / IP]** field.

3 Select the severity level from the **Severity** drop-down list.

4 Select the generic trap type from the **Generic** drop-down list.

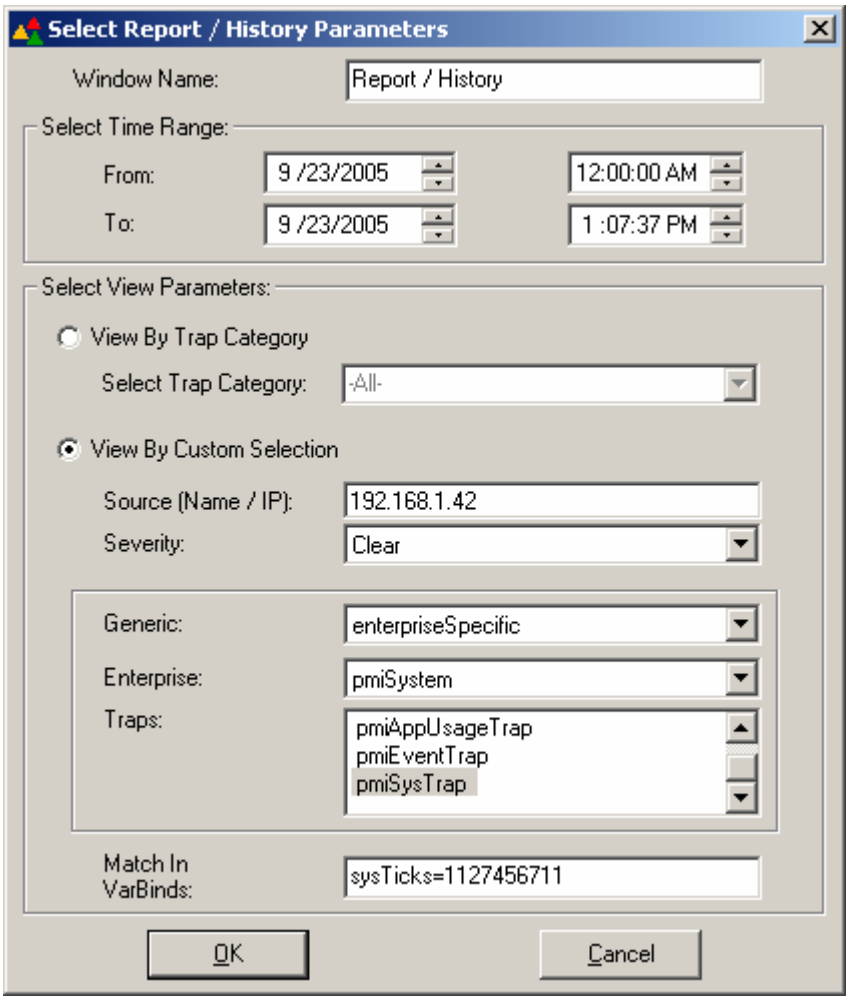
> **Note**
>
> The **Enterprise** list box is enabled, only when you choose **enterpriseSpecific** trap type from the **Generic** drop-down list.

5 Select the enterprise from the **Enterprise** drop-down list.

6 Select the traps associated with the selected enterprise from the **Traps** drop-down list.

7 Type the varBinds associated with the selected trap in the **Match In VarBinds** field.

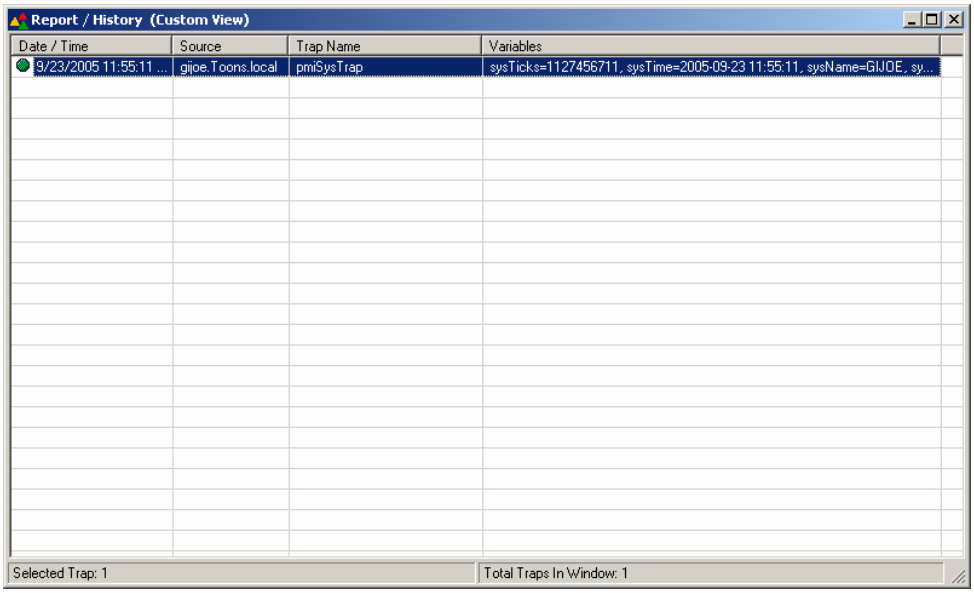
Figure 83 Select Report/History Parameters dialog box with user entered values.




8 Click **OK**.

TrapTracker displays the trap details of the selected category in a new window.

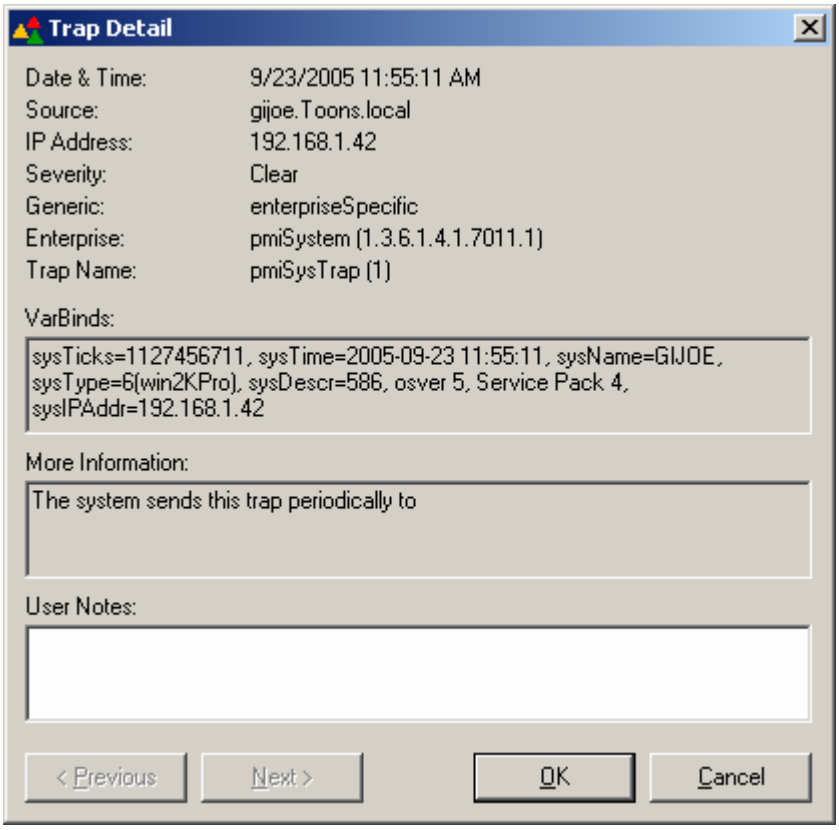
Figure 84 New window displaying selected category details.




***9*** Double-click the trap record to view details.

TrapTracker displays the trap details in the “Trap Detail” window.

Figure 85 New window displaying selected category details.




10 Type notes in the **User Notes** field and then click **OK**.

11 Click **< Previous** to view details of the previous trap.

12 Click **Next >** to view the details of the next trap.

13 Click  at the upper-right corner of the Report / History (Custom View) window to close.

# Chapter 4
# Tools

*In this chapter, you will learn about:*

- SMI
- SNMP
- MIB
- MIB II Tree
- SNMP Datatypes
- UDP
- MibCompiler / Browser
- Starting MibCompiler
- Understanding MibCompiler Console
- Need for MIB Compilation
- Compiling a MIB File
- Viewing MIB Details
- Viewing Trap Details
- Search and Find
- Deleting MIB
- Exiting MibCompiler
- DB Compaction

# What is SMI?

**SMI** stands for **Structure of Managed Information** and represents the notation by which an SNMP MIB must be written. Another way to look at SMI is that it is the grammar to write SNMP MIBs. There are two types of SMI: SMIv1 and SMIv2 with SMIv1 being the earlier version, of course, back in 1990.

**SMIv1** is now an obsolete notation. However, there are still many SNMP MIBs written before SMIv2 arrived in 1993. SMIv1 is represented by the following IETF RFCs (Request for Comments):

- RFC 1155 for Structure and Identification of Management Information for TCP/IP-based Internets
- RFC 1212 for Concise MIB Definitions
- RFC 1215 A Convention for Defining Traps

**SMIv2** is the new notation, which should be used whenever you create a new MIB. SMIv2 is represented by the following MIBs:

- RFC 2576 for Coexistence between Version 1, Version 2, and Version 3
- RFC 2578 for Structure of Management Information Version 2
- RFC 2579 for Textual Conventions for SMIv2
- RFC 2580 for Conformance Statements for SMIv2

"In order for the MIB to serve the needs of a network-management system, it must meet two objectives:

1. The object or objects used to represent a particular resource must be the same at each node. [...]
2. A common scheme for representation must be used to support interoperability." - William Stallings.

In both Internet and OSI (Open System Interconnection) network management, these two objectives are met by a common structure of management information (SMI), which is defined in RFC 1155. The SMI is the specification for the MIB object tree, which provides a means of associating a common numerical identification code for a given object.

For more information, refer

TCP/IP MIB Objects, Object Characteristics and Object Types.

# What is SNMP?

The SNMP Management Framework presently consists of five major components:

- An overall architecture, described in RFC 2571 [RFC2571].

- Mechanism for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIv1 and is described in STD 16, RFC 1155 [RFC1155], STD 16, RFC 1212 [RFC1212] and RFC 1215 [RFC1215]. The second version called SMIv2 is described in STD 58, RFC 2578 [RFC2578], RFC 2579 [RFC2579] and RFC 2580 [RFC2580].
- Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and is described in STD 15, RFC 1157 [RFC1157]. A second version of the SNMP message protocol, which is not an Internet standards track protocol is called SNMPv2c and is described in RFC 1901 [RFC1901] and RFC 1906 [RFC1906]. The third version of the message protocol is called SNMPv3 and is described in RFC 1906 [RFC1906], RFC 2572 [RFC2572] and RFC 2574 [RFC2574].
- Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [RFC1157]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [RFC1905].
- A set of fundamental applications is described in RFC 2573 [RFC2573]. The view-based access control mechanism is described in RFC 2575 [RFC2575]. A more detailed introduction to the current SNMP Management Framework can be found in RFC 2570 [RFC2570].

The Simple Network Management Protocol is a protocol for Internet network management services. It is formally specified in a series of related RFC documents.

(Some of these RFCs are in "historic" or "informational" status)

- RFC 1067 - A Simple Network Management Protocol
- RFC 1089 - SNMP over Ethernet
- RFC 1140 - IAB Official Protocol Standards
- RFC 1147 - Tools for Monitoring and Debugging TCP/IP Internets and Interconnected Devices [superceded by RFC 1470]
- RFC 1155 - Structure and Identification of Management Information for TCP/IP based Internets.
- RFC 1156 (H)- Management Information Base Network Management of TCP/IP based internets
- RFC 1157 - A Simple Network Management Protocol
- RFC 1158 - Management Information Base Network Management of TCP/IP based internets: MIB-II
- RFC 1161 (H)- SNMP over OSI
- RFC 1187 - Bulk Table Retrieval with the SNMP
- RFC 1212 - Concise MIB Definitions

- RFC 1213 - Management Information Base for Network Management of TCP/IP-based internets: MIB-II
- RFC 1215 (I)- A Convention for Defining Traps to be used with the SNMP
- RFC 1224 - Techniques for Managing Asynchronously-Generated Alerts
- RFC 1270 (I)- SNMP Communication Services
- RFC 1303 (I)- A Convention for Describing SNMP-based Agents
- RFC 1470 (I)- A Network Management Tool Catalog
- RFC 1298 - SNMP over IPX (obsolete, see RFC 1420)
- RFC 1418 - SNMP over OSI
- RFC 1419 - SNMP over AppleTalk
- RFC 1420 - SNMP over IPX (replaces RFC 1298)

SNMPv1 is historic and SNMPv3 is now standard and is described by

RFCs 3410-3418 (note: 3410 is informational)

**What do SNMPv1 and SNMPv2 have to do with SMIv1 and SMIv2?**

SNMPv1 and SNMPv2 are transport protocols to carry MIB information, while SMIv1 and SMIv2 only specify the grammar by which SNMP MIBs are written. In fact, there is an SNMPv3 protocol definition, which relies on SMIv2.

# *What is MIB?*

An MIB is not a database but a file written in a specific language that lists variables. It assigns each variable a name, a number, and a set of permissions. It may also provide a description of what the variable is supposed to represent. Since everything in SNMP is a "simple" action on a variable, this is important.

The MIB files define a hierarchy. Each MIB variable is a leaf in the MIB tree. So how are names translated into numbers that the device will understand? In the MIB tree, each level is responsible for numbering itself in relation to the level above. .1.3.6.1.2.1.1.1 But that is not quite all. There can actually be many instances of the same variable on any single device, and so you must specify an instance number with any request.

To obtain values of objects from the agent, you need to specify the instance of the object. Appending an instance index to the object identifier specifies the instance of an object. For example, the last 0 in: **.iso.3.dod.1.mgmt.mib.1.sysUpTime.0** is the instance index. An instance index of "0" (zero) specifies the first instance, "1" specifies the second instance, and so on. Since sysUpTime is a scalar object, it has only one instance. Therefore, an instance index of zero is always specified when retrieving the

value of a scalar object. An instance index higher than 0 can only be used in the case of columnar objects (in table), which can have multiple instances.

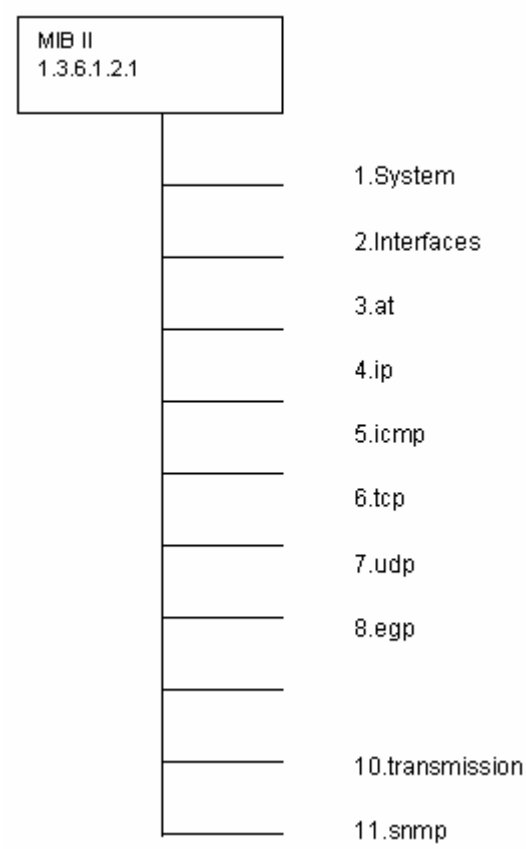
# *MIB-II Tree*

MIB variables have to be simple elements because of the purpose of the SNMP. Therefore, these variables are elementary stand-alone quantities, integers, octet strings, or object identifiers. Sometimes, these variables are organized into tables.

The tree of MIB variables does not have limits; it can grow and grow. Therefore, it is not surprising that definitions need to be improved or updated occasionally. If the improvement involves a major change, an updated version of the MIB will simply define a completely new variable and mark the old one as deprecated or outright obsolete.

Eleven groups are referenced in the original MIB-II document (RFC 1213). One of these, CMOT (ISO Common Management Information on top of TCP/IP) is no longer used because this project was abandoned. The 10 remaining groups describe the basic information required to manage a TCP/IP Internet.

Figure 86 MIB II Tree view




# Groups of MIB-II

Refer the following links for detailed information on MIB-II groups.

- System group (1.3.6.1.2.1.1) Defines a list of objects that pertain to system operation, such as the system uptime, system contact, and system name.
- Interfaces group (1.3.6.1.2.1.2) Keeps track of the status of each interface on a managed entity. The **interfaces** group monitors, which interfaces are up or down and track such things as octets sent and received errors and discards, etc.
- Address Translation group (1.3.6.1.2.1.3) The address translation (**at**) group is deprecated and is provided only for backward compatibility. It will probably be dropped from MIB-III.
- Internet Protocol group (1.3.6.1.2.1.4) Keeps track of many aspects of IP, including IP routing.

- Internet Control Message Protocol group (1.3.6.1.2.1.5) Tracks things such as ICMP errors, discards, etc.
- Transmission Control Protocol group (1.3.6.1.2.1.6) Tracks, among other things, the state of the TCP connection (e.g., **closed**, **listen**, **synSent**, etc.).
- User Datagram Protocol group (1.3.6.1.2.1.7) Tracks UDP statistics, datagrams in and out, etc.
- Exterior Gateway Protocol group (1.3.6.1.2.1.8) Tracks various statistics about EGP and keeps an EGP neighbor table.
- SNMP group (1.3.6.1.2.1.11) Measures the performance of the underlying SNMP implementation on the managed entity and tracks things such as the number of SNMP packets sent and received.
- Transmission group (1.3.6.1.2.1.10) There are currently no objects defined for this group, but other media-specific MIBs are defined using this subtree.

# *SNMPv1 Datatypes*

SNMP uses the following basic ASN.1 datatypes as the most important ones:

Table 18

| Datatype | Description |
|---|---|
| **INTEGER** | A 32-bit number often used to specify enumerated types within the context of a single managed object. For example, the operational status of a router interface can be up, down, or testing. With enumerated types, 1 would represent up, 2 down, and 3 testing. The value zero (0) must not be used as an enumerated type, according to RFC 1155. |
| **OCTET STRING** | A string of zero or more octets (more commonly known as bytes) generally used to represent text strings, but also sometimes used to represent physical addresses. |

| Datatype | Description |
|---|---|
| **Counter** | A 32-bit number with minimum value 0 and maximum value 232 - 1 (4,294,967,295). When the maximum value is reached, it wraps back to zero and starts over. It's primarily used to track information such as the number of octets sent and received on an interface or the number of errors and discards seen on an interface. A Counter is monotonically increasing, in that its values should never decrease during normal operation. When an agent is rebooted, all Counter values should be set to zero. Deltas are used to determine if anything useful can be said for successive queries of Counter values. A delta is computed by querying a Counter at least twice in a row, and taking the difference between the query results over some time interval. |
| **OBJECT IDENTIFIER** | A dotted-decimal string that represents a managed object within the object tree. For example, 1.3.6.1.4.1.9 represents Cisco Systems' private enterprise OID. |
| **NULL** | Not currently used in SNMP. |
| **SEQUENCE** | Defines lists that contain zero or more other ASN.1 datatypes. |
| **SEQUENCE OF** | Defines a managed object that is made up of a SEQUENCE of ASN.1 types. |
| **IpAddress** | Represents a 32-bit IPv4 address. Neither SMIv1 nor SMIv2 discusses 128-bit IPv6 addresses; this problem will be addressed by the IETF's SMI Next Generation (SMING) working group (see http://www.ietf.org/html.charters/sming-charter.html). |
| **NetworkAddress** | Same as the IpAddress type, but can represent different network address types. |
| **Gauge** | A 32-bit number with minimum value 0 and maximum value $2^{32}$ - 1 (4,294,967,295). Unlike a Counter, a Gauge can increase and decrease at will, but it can never exceed its maximum value. The interface speed on a router is measured with a Gauge. |
| **TimeTicks** | A 32-bit number with minimum value 0 and maximum value $2^{32}$ - 1 (4,294,967,295). TimeTicks measures time in hundredths of a second. Uptime on a device is measured using this datatype. |
| **Opaque** | Allows any other ASN.1 encoding to be stuffed into an OCTET STRING. |

# SNMPv2 Datatypes

SNMP uses the following basic ASN.1 datatypes as the most important ones:

Table 19

| Datatype | Description |
|---|---|
| **Integer32** | Same as an INTEGER. |
| **Counter32** | Same as a Counter. |
| **Gauge32** | Same as a Gauge. |
| **Unsigned32** | Represents decimal values in the range of 0 to $2^{32}$ - 1 inclusive. |
| **Counter64** | Similar to Counter32, but its maximum value is 18,446,744,073,709,551,615. Counter64 is ideal for situations in which a Counter32 may wrap back to 0 in a short amount of time. |
| **BITS** | An enumeration of nonnegative named bits. |

# SNMPv2 Object Definition Enhancements

Table 20

| Object Definition Enhancements | Description |
|---|---|
| **UnitsParts** | A textual description of the units (i.e., seconds, milliseconds, etc.) used to represent the object. |
| **MAX-ACCESS** | An OBJECT-TYPE's ACCESS can be MAX-ACCESS in SNMPv2. The valid options for MAX-ACCESS are read-only, read-write, read-create, not accessible, and accessible-for-notify. |
| **STATUS** | This clause has been extended to allow the current, obsolete, and deprecated keywords. current in SNMPv2 is the same as mandatory in an SNMPv1 MIB. |
| **AUGMENTS** | In some cases it is useful to add a column to an existing table. The AUGMENTS clause allows you to extend a table by adding one or more columns, represented by some other object. This clause requires the name of the table the object will augment. |

# Textual conventions for SMIv2

Table 21

| Textual Convention | Description |
|---|---|
| **DisplayString** | A string of NVT ASCII characters. A DisplayString can be no more than 255 characters in length. |
| **PhyAddress** | A media- or physical-level address, represented as an OCTET STRING. |
| **MacAddress** | Defines the media-access address for IEEE 802 (the standard for local area networks) in canonical order. (In everyday language, this means the Ethernet address.) This address is represented as six octets. |
| **TruthValue** | Defines both true and false Boolean values. |
| **TestAndIncr** | Used to keep two management stations from modifying the same managed object at the same time. |
| **AutonomousType** | An OID used to define a subtree with additional MIB-related definitions. |
| **VariablePointer** | A pointer to a particular object instance, such as the ifDescr for interface 3. In this case, the VariablePointer would be the OID ifDescr.3. |
| **RowPointer** | A pointer to a row in a table. For example, ifIndex.3 points to the third row in the ifTable. |
| **RowStatus** | Used to manage the creation and deletion of rows in a table, since SNMP has no way of doing this via the protocol itself. RowStatus can keep track of the state of a row in a table, as well as receive commands for creation and deletion of rows. This textual convention is designed to promote table integrity when more than one manager is updating rows. The following enumerated types define the commands and state variables: active(1), notInService(2), notReady(3), createAndGo(4), createAndWait(5), and destroy(6). |
| **TimeStamp** | Measures the amount of time elapsed between the device's system uptime and some event or occurrence. |
| **TimeInterval** | Measures a period of time in hundredths of a second. TimeInterval can take any integer value from 0-2147483647. |
| **DateAndTime** | An OCTET STRING used to represent date-and-time information. |
| **StorageType** | Defines the type of memory an agent uses. The possible values are other(1), volatile(2), nonVolatile(3), permanent(4), and readOnly(5). |
| **Tdomain** | Denotes a kind of transport service. |

| Textual Convention | Description |
|---|---|
| **TAddress** | Denotes the transport service address. TAddress is defined to be from 1-255 octets in length. |

# UDP

UDP has been chosen and recommended for SNMP transport protocol. Initially, SNMP was targeted at managing Internet nodes and the predominant Internet protocol suite TCP/IP. The choice of TCP/IP suite is viable because IP became the protocol for commercial backbone networks and users can count on a TCP/IP implementation available on any type of host and router.

TCP and UDP provide transport services. However, UDP was preferred. This is due to TCP characteristics, it is a complicate protocol and it consume to many memory and CPU resources, whereas UDP is easy to build and run. Vendors have built simple versions of IP and UDP in devices (repeaters and modems). Thus the total software needed is small and can be stored in a ROM. UDP is well suited to the brief request / response message used in network management communication.

The SNMP protocol is UDP-based. Each message is sent in an atomic UDP packet. From RFC1157: "A message consists of a version identifier, an SNMP community name, and a protocol data unit (PDU)." Version is the SNMP version, community name is the password, and a PDU is just data.

# SNMP PDU

There are five types of PDUs: Authorization

- **Get-request** - is used to request the values of one or more MIB variables.
- **Get-next-request** - is used to read variable values in the MIB sequentially. It is often used to read though a table of values. After a first read with the get-request, get-next-request is used to read the remaining rows.
- **Set-request** - is used to update an MIB value.
- **Get-response** - is returned as an answer to a get-request, a get-next-request or set-request message.
- **Trap** - is used to support significant events (e.g. a cold or a warm restart or a link that has gone down).

# *MibCompiler / Browser*

The MibCompiler is a standard framework to compile and store MIB files that are represented in ASN1.0 format to a binary format. This manner of storing the MIB documents makes it easier to parse the document and represent it in a user understandable format.

The MIB Browser is an indispensable tool that can walk the user through the MIB trees, view MIB tables, search MIBs, remotely modify SNMP values, and perform many other SNMP functions.

The most critical part of a MIB Browser is the number of standard and proprietary MIBs it supports. Without the correct MIBs, the data collected from a remote device is difficult to interpret and use. TTW MIB Browser is shipped with over 10,000 precompiled unique OIDs from hundreds of standard and vendor MIBs.

## Scope

The MibCompiler supports the following features.

- Parse the ASN format (.mib) file and convert it to a binary format
- Retrieve the different objects (Mib Objects/Traps) from the MIB file
- A Viewer is available to facilitate viewing the contents of the MIB file in a user understandable format
- Multiple MIBs can be compiled and stored in a single file
- Support nested compilation, collection of standard mibs along with installation.
- Caching compiled and imported mibs for interconnected MIB compilation.
- Providing help on all the above features

## References and Terminology

The term MIB stands for Management Information Base, which is an integral part of the SNMP architecture. The MIB is a storage area for the resources being monitored. The description of this information is represented in an ASN1.0 format file. The MibCompiler parses these files in order to generate an output that can be used by the Manager System to procure information from the remote system's MIB.

## Architectural Overview

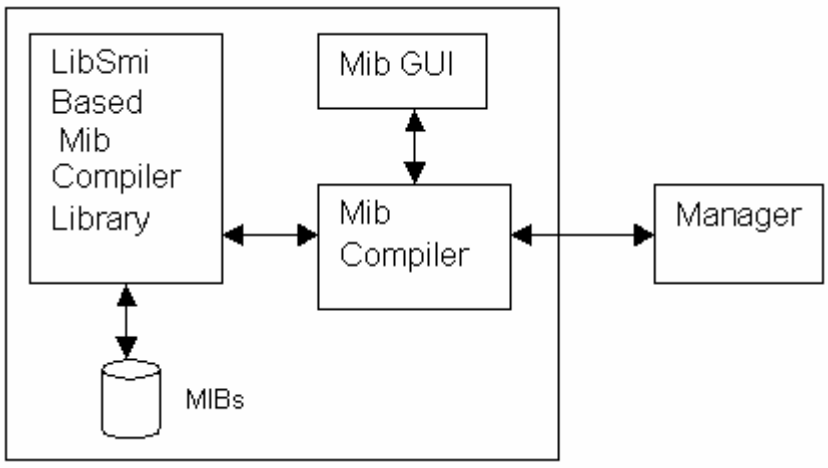
The MibCompiler has the following components.

- A customized parser that parses the MibComp generated output file
- A Program to configure Alerts, License etc
- A Viewer to browse through the contents of the Mib File

**Note**

The Parser API library from MibComp is used for parsing the (.mib) file.

Figure 87 Architectural Overview




The **MibCompiler** is used to compile MIB modules (and therefore, the trap definitions) and store into a binary file. The Manager when it receives a trap can lookup the trap definitions and use them to translate the trap into a user-friendly format before inserting into the Database. These translations are used by the Manager Console to show the trap information in a friendly format.

The **Mib GUI** is used to display the compiled contents of the MIB components in a user readable format. Multiple MIB components can be seen simultaneously.

## Functional Definition

- Compile the Mib modules using the MibComp Compiler libraries to generate an intermediate format.
- Parse this intermediate format and convert it to a user comprehendible format that can be used by the TrapTracker and the Mib Console to display the Mib.

**Note**

The Mibs are stored in a binary format in the “**mymibs.bin**” file.

# Starting MibCompiler

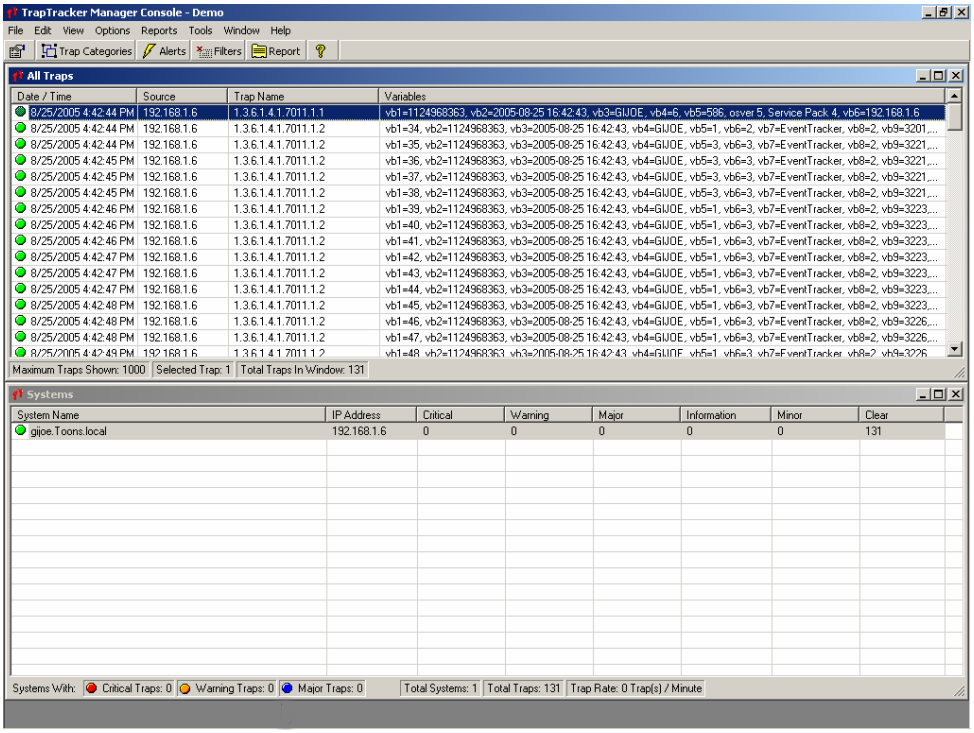
This option enables you to start MibCompiler.

## To start MibCompiler

1 Double-click TrapTracker on the Control Panel.

TrapTracker displays the TrapTracker Manager console.

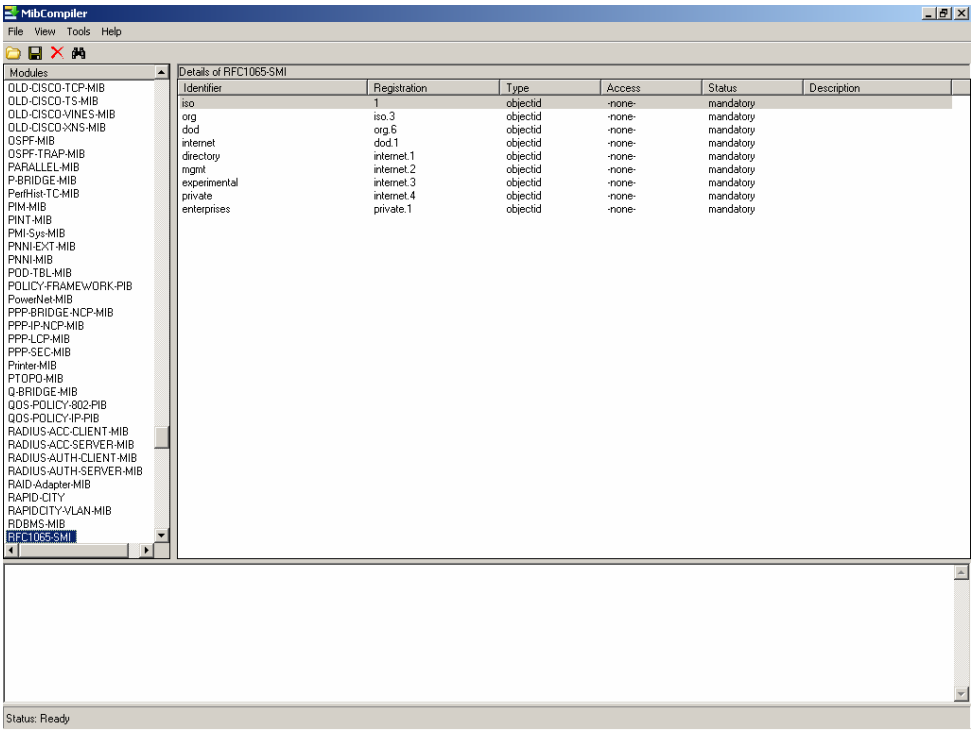
Figure 88 TrapTracker Manager Console




2 From the **Tools** menu, choose **MibCompiler**.

TrapTracker displays the MibCompiler console.

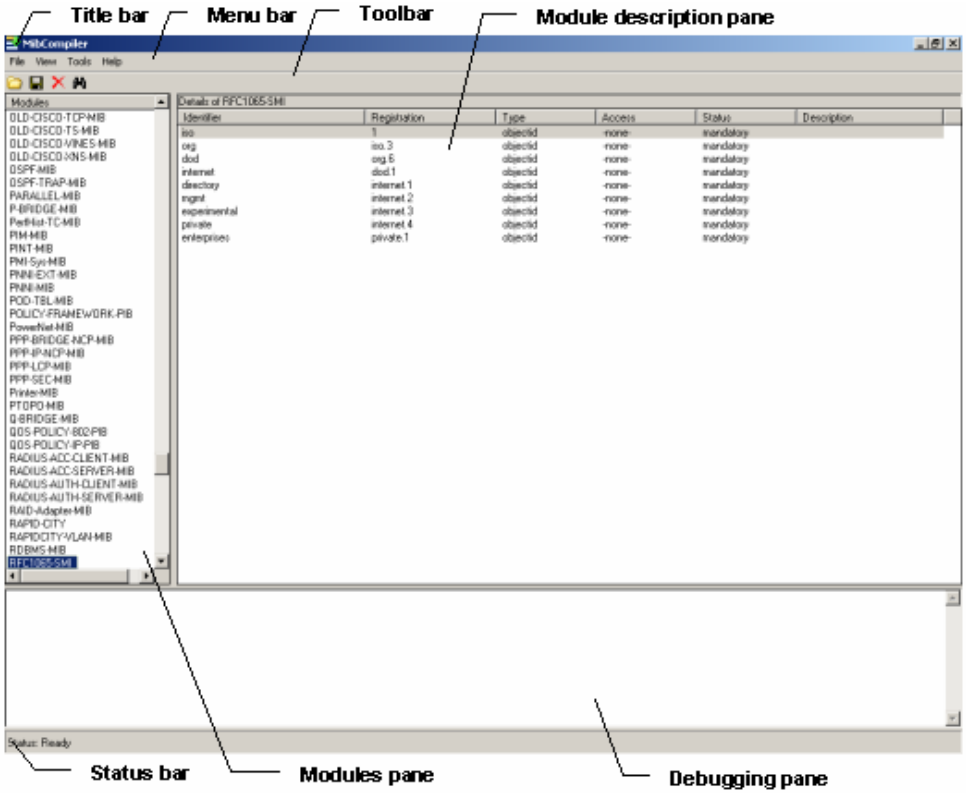
Figure 89
MibCompiler Console




# *Understanding MibCompiler Console*

The GUI contains five parts: Menu bar, Toolbar, Modules window, Module description window, Debugging window and Status bar.

Figure 90
MibCompiler Console




### Title Bar

The top strip of MibCompiler window is the Title Bar. The Title Bar shows the name of the application.

### Menu Bar

The menu bar contains menus with relevant commands. From the menus, choose appropriate commands or use shortcut keys to execute the commands.

### Tool Bar

The toolbar contains buttons with tool tips to perform basic tasks.

Table 22

| Click | To |
|---|---|
|  | Compile a MIB module. |
|  | Save the compiled module(s). |
|  | Delete the compiled module(s). |
|  | Search and find a word or phrase in Module Names, OidIdentifiers or Trap Names. |

### Modules Pane

It's a resizable window; displays sorted list of all the compiled Mib modules.

### Module Details Pane

It's a resizable window; displays description of the module selected in Modules window.

Table 23

| Field | Description |
|---|---|
| **Identifier** | Objects associated with the selected MIB. |
| **Registration** | Hierarchy details of the objects. |
| **Type** | Datatype of the objects:<br>table<br>row<br>counter64<br>objectid<br>bitstring<br>integer<br>octectstring<br>ipaddress<br>Timeticks |
| **Access** | Permission on manipulation of objects:<br>read-write<br>not accessible<br>read-only<br>counter<br>gauge<br>-none- |
| **Status** | Status of the objects:<br>current<br>mandatory<br>deprecated |
| **Description** | Description of the objects. |

### Debugging Pane

It's a resizable window; displays success / failure status of the compiled MIBs and other debugging information.

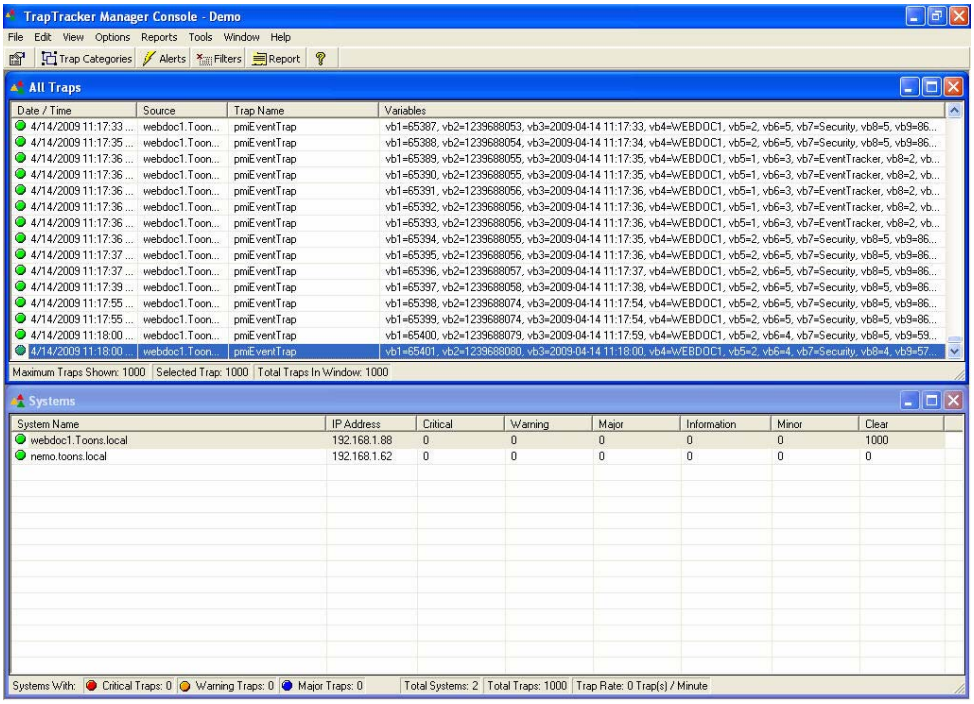
**Status Bar**

Activity status of MibCompiler like Ready, Compiling is displayed here.

# Need for MIB Compilation

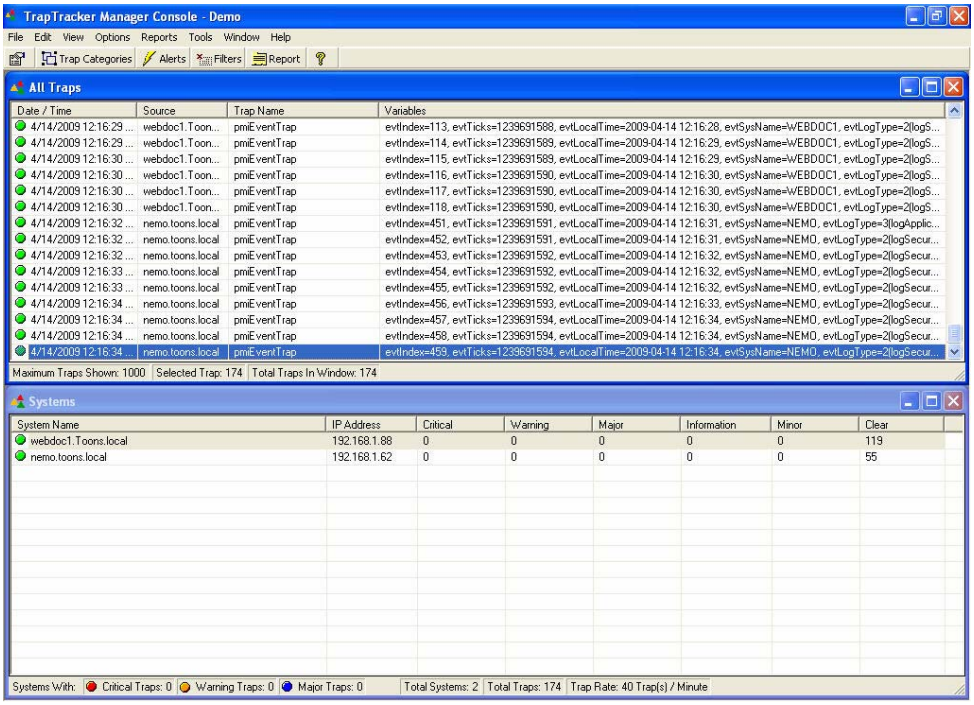
TTW is incapable of interpreting the traps sent by the SNMP compliant devices in human understandable form when there is no reference to those traps in the bin file. For example, the following figure displays the varBinds associated with their respective traps in the All Traps window, which is incomprehensible to the end-user.

Figure 91 varBinds before Mib compilation.




But when the Mib module **PMI-Sys-MIB**, which is the enterprise specific Mib is compiled and saved, TrapTracker Manager displays the varBinds associated with that Mib in user comprehensible way.

Figure 92 varBinds after Mib compilation.




So it is mandatory to compile the enterprise specific MIBs and save them in bin file.

# Compiling a Single MIB Module

This option enables you to compile a single MIB module.

## To compile a single MIB module

1. Open the MibCompiler console.
2. From the **File** menu, choose **Compile one MIB**.

   (OR)

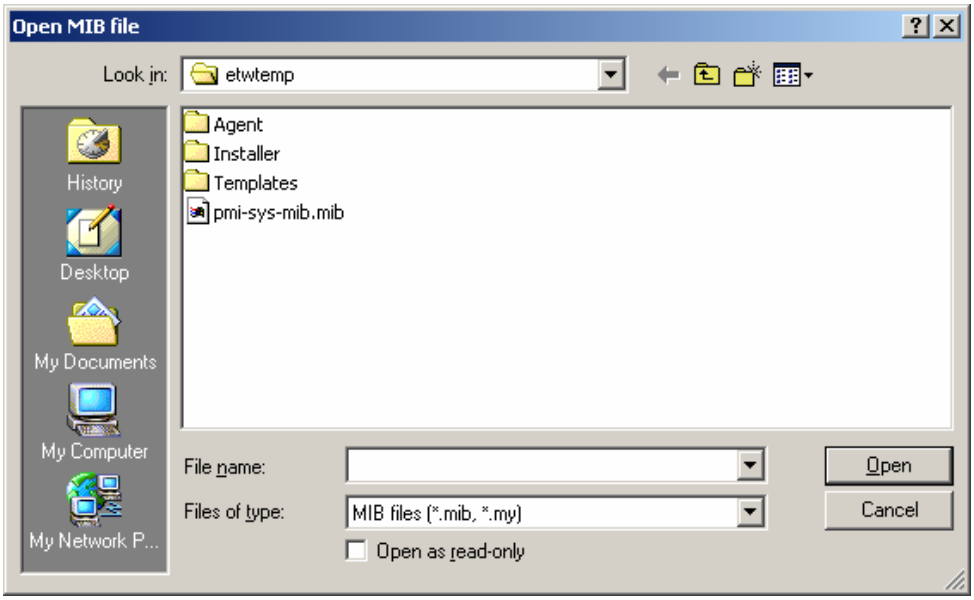
   Press **Ctrl + N** on your keyboard.

   (OR)

   Click  on the tool bar.

   MibCompiler displays “Open MIB file” window.

Figure 93 Open Mib file dialog box.

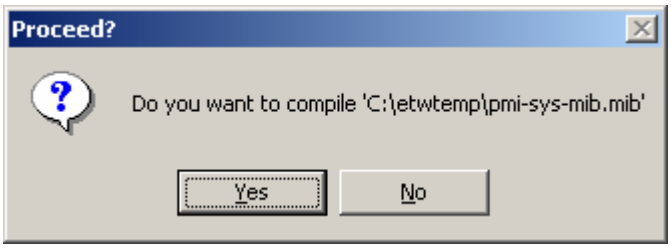



3 Go to the appropriate folder and select the MIB file that you want to compile.

4 Click **Open**.

MibCompiler displays the confirmation message box.

Figure 94 Proceed? message box.




5 Click **Yes** to start compilation.

TrapTracker displays the compilation success / failure status in the debugging window.

Figure 95 MIB compilation status

```
C:\etwtemp\pmi-sys-mib.mib:98: type `EvtEntry' of node `evtEntry' does not resolve to a known base type
C:\etwtemp\pmi-sys-mib.mib:135: scalar's parent node must be simple node
C:\etwtemp\pmi-sys-mib.mib:143: scalar's parent node must be simple node
C:\etwtemp\pmi-sys-mib.mib:151: scalar's parent node must be simple node
C:\etwtemp\pmi-sys-mib.mib:159: scalar's parent node must be simple node
C:\etwtemp\pmi-sys-mib.mib:167: scalar's parent node must be simple node
C:\etwtemp\pmi-sys-mib.mib:179: scalar's parent node must be simple node
C:\etwtemp\pmi-sys-mib.mib:193: scalar's parent node must be simple node
C:\etwtemp\pmi-sys-mib.mib:201: scalar's parent node must be simple node
C:\etwtemp\pmi-sys-mib.mib:209: scalar's parent node must be simple node
C:\etwtemp\pmi-sys-mib.mib:217: scalar's parent node must be simple node
C:\etwtemp\pmi-sys-mib.mib:225: scalar's parent node must be simple node
C:\etwtemp\pmi-sys-mib.mib:233: scalar's parent node must be simple node
C:\etwtemp\pmi-sys-mib.mib:253: type `AppUsageEntry' of node `appUsageEntry' does not resolve to a known base type
C:\etwtemp\pmi-sys-mib.mib:262: SEQUENCE element #9 `appUsageDescr' is not a child node under `appUsageEntry'
C:\etwtemp\pmi-sys-mib.mib:288: scalar's parent node must be simple node
C:\etwtemp\pmi-sys-mib.mib:296: scalar's parent node must be simple node
C:\etwtemp\pmi-sys-mib.mib:304: scalar's parent node must be simple node
C:\etwtemp\pmi-sys-mib.mib:313: scalar's parent node must be simple node
C:\etwtemp\pmi-sys-mib.mib:321: scalar's parent node must be simple node
C:\etwtemp\pmi-sys-mib.mib:329: scalar's parent node must be simple node
C:\etwtemp\pmi-sys-mib.mib:337: scalar's parent node must be simple node
C:\etwtemp\pmi-sys-mib.mib:345: scalar's parent node must be simple node
C:\etwtemp\pmi-sys-mib.mib:281: unknown object identifier label `appUsageDescr'
C:\etwtemp\pmi-sys-mib.mib:353: scalar's parent node must be simple node
C:\etwtemp\pmi-sys-mib.mib:361: scalar's parent node must be simple node
MibComp: compiled module `C:\etwtemp\pmi-sys-mib.mib' with errors/warnings
Compiled Successfully: PMI-Sys-MIB mib inserted.
Compilation completed
Select save from menu or toolbar to save the added MIB to disk
```

*6* From the **File** menu, choose **Save**.

(OR)

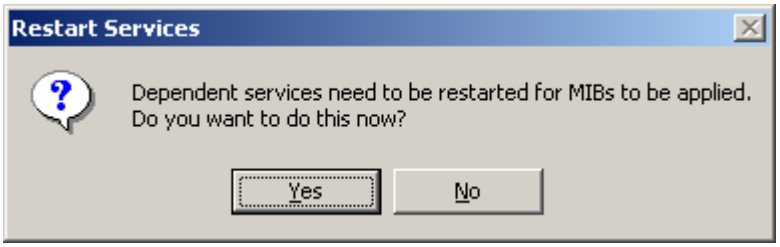
Press **Ctrl + S** on your keyboard.

(OR)

Click  on the tool bar to save.

MibCompiler displays the “Restart Services” confirmation message box.

Figure 96 Restart Services message box



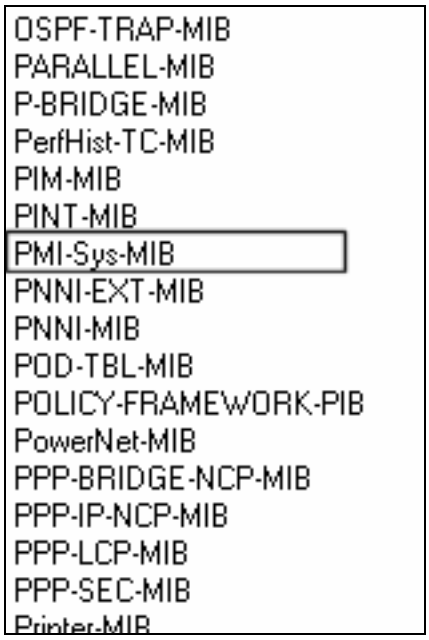

*7* Click **Yes** to restart the dependent services.

**Note**

TrapTracker inserts the newly compiled MIB to the bin file and to the database.

MibCompiler displays the newly compiled **PMI-sys-MIB** in the Modules window.

Figure 97 Partial Modules window




# *Compiling Multiple MIB Modules*

In group compilation mode you can compile a group of related MIB modules. The MibCompiler randomly chooses the MIB modules irrespective of their dependency and compiles iteratively. Consider there are three MIB modules “A”, “B” and “C”, where “C” has dependency on “B” and in turn “B” has dependency on “A”. Suppose TrapTracker compiles “C” first, the compilation will complete with errors since “C” has dependency on “B”. But the MIB module that has no reference will get compiled successfully without errors; in this case “A”. The MIB modules, which are compiled successfully, will also be compiled for every iteration. You can refer the **gcreport.txt**, which is generated by the MibCompiler for group compilation or the debugging window to get to know what has happened to your MIB modules and compile them iteratively until all the modules are compiled successfully. If there are references in a MIB module to other modules, which could not be resolved, then the compilation process terminates unsuccessfully, irrespective of the number of iterations you attempt. The gcreport.txt is

generated only for group compilation and not for single Mib compilation. Its contents are over written for every compilation.

This option enables you to compile multiple MIB modules.

## To compile multiple MIB Modules

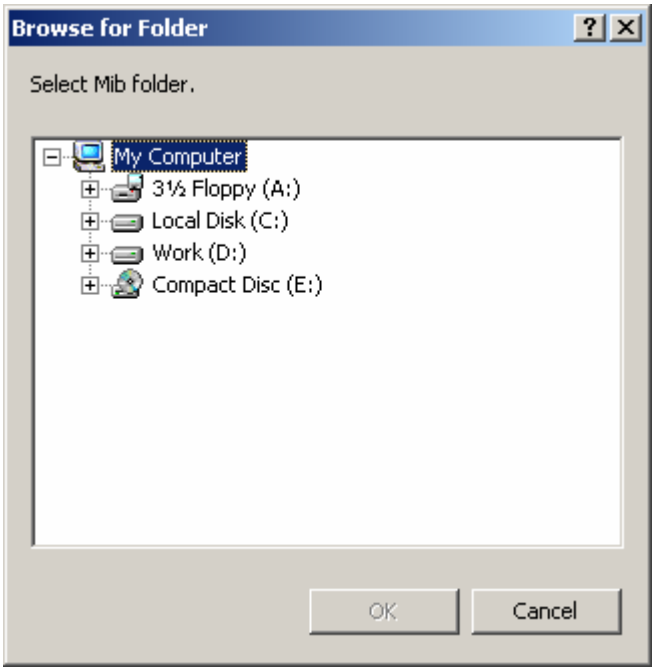
1. Open the MibCompiler console.
2. From the **File** menu, choose **Compile multiple MIBs**.

   (OR)

   Press **Ctrl + G** on your keyboard.

   MibCompiler displays “Browse for Folder” window.

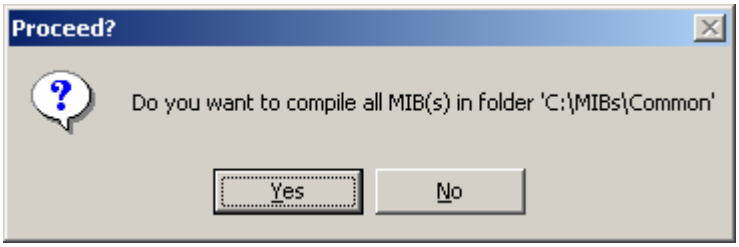
Figure 98 Browse for Folder dialog box.




3. Go to the appropriate drive and select the folder that you want to compile.
4. Click **OK**.

   MibCompiler displays the confirmation message box to proceed further.

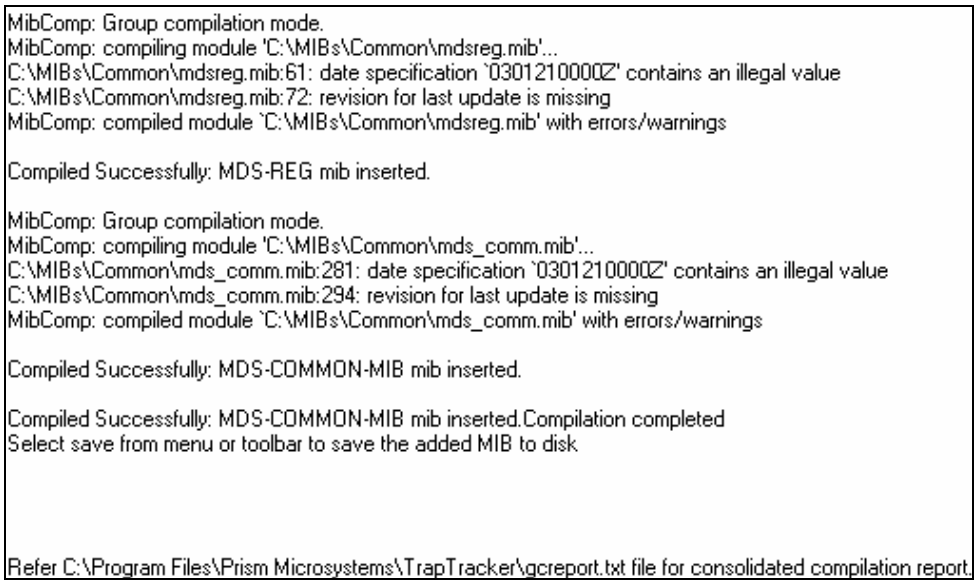
Figure 99 Proceed? message box.



5 Click **Yes** to start compilation.

TrapTracker displays the compilation success / failure status in the debugging window.

Figure 100 MIB compilation status




6 From the **File** menu, choose **Save**.

(OR)

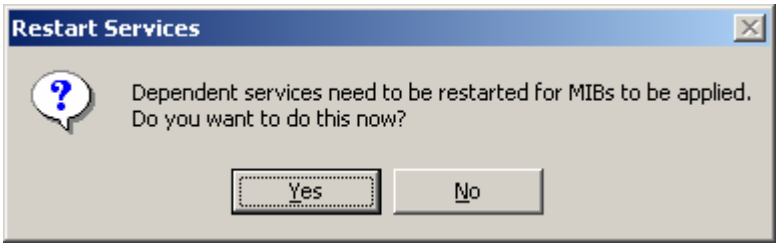
Press **Ctrl + S** on your keyboard.

(OR)

Click  on the tool bar to save.

MibCompiler displays the “Restart Services” confirmation message box.

Figure 101 Restart Services message box




1 Click **Yes** to restart the dependent services.

# Saving MIB Compilation Report

By default, MibCompiler generates gcreport.txt for group compilation. Apart from this, you can manually copy, paste, delete, save and organize those details in separate text files for future reference for group compilation and single Mib compilation.

This option enables you to save the Mib compilation report in a text file.

### To save MIB compilation report

1. Right-click the debugging window.

   MibCompiler displays the shortcut menu.

   From the shortcut menu, choose **Select All** and then **Copy**.

2. Create a text file and paste the copied details in that text file.

   You can also select and copy a portion of the details.

# Viewing MIB Details

This option enables you to view the MIB details.

### To view MIB details

1. Open the MibCompiler console.
2. From the **View** menu, choose **Mibs**.

   (OR)

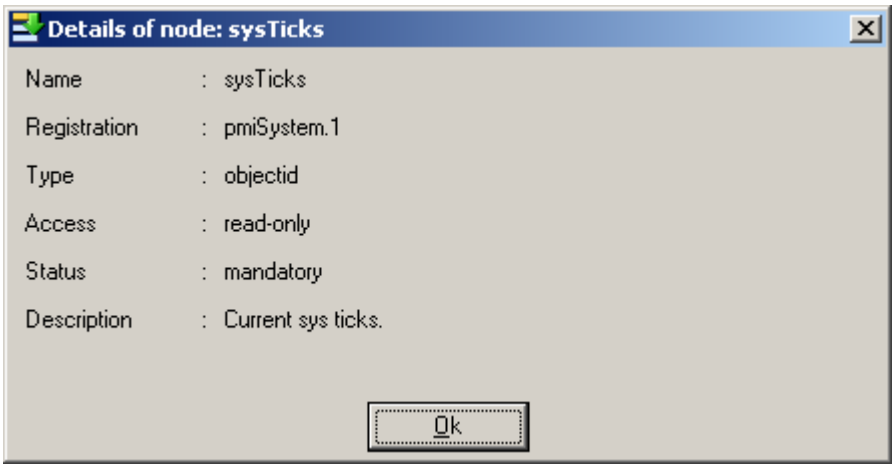
   Press **Ctrl + M** on your keyboard.

   By default, MibCompiler selects this view.
3. Select a MIB module on the **Modules** window.

   MibCompiler displays the details of the selected MIB in the **Details** window.
4. Double-click an object on the “Details” window.

   MibCompiler displays the details of the selected object.

Figure 102 Details of node: sysTicks.




5 Click **OK**.

# Viewing Trap Details

This option enables you to view the trap details.

## To view trap details

1 Open the MibCompiler console.

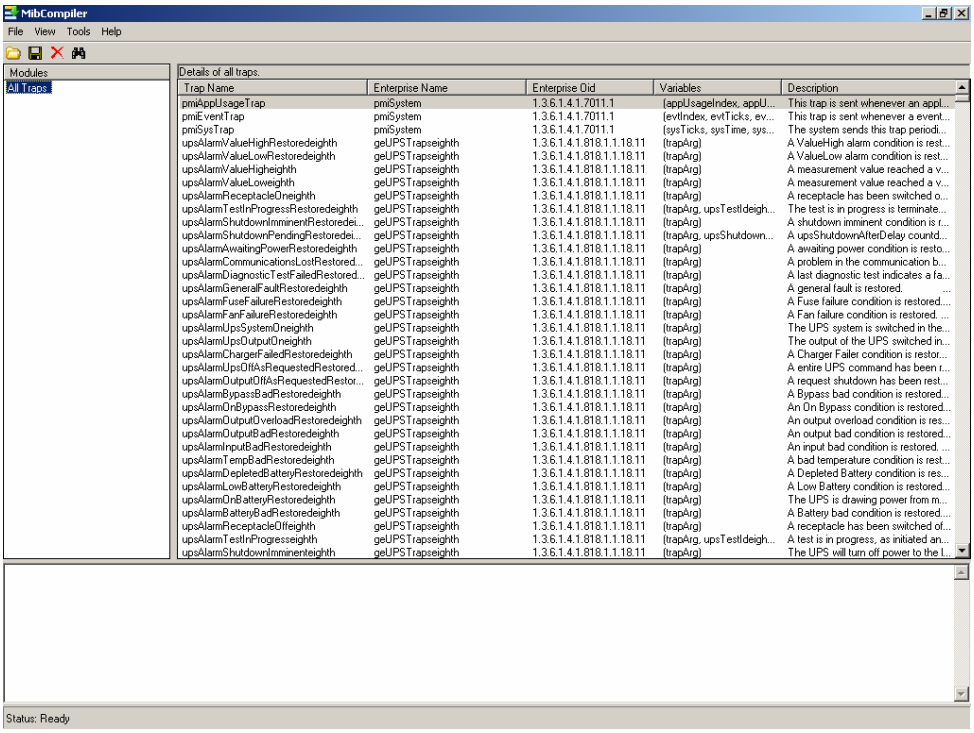
2 From the **View** menu, choose **Traps**.

(OR)

Press **Ctrl + T** on your keyboard.

MibCompiler displays all traps in the “Details” window.

Figure 103 Details of all traps.




***3*** Double-click the trap that you want to view details.

MibCompiler displays the details of the selected trap.

Figure 104 Details of trap: pmiSysTrap.

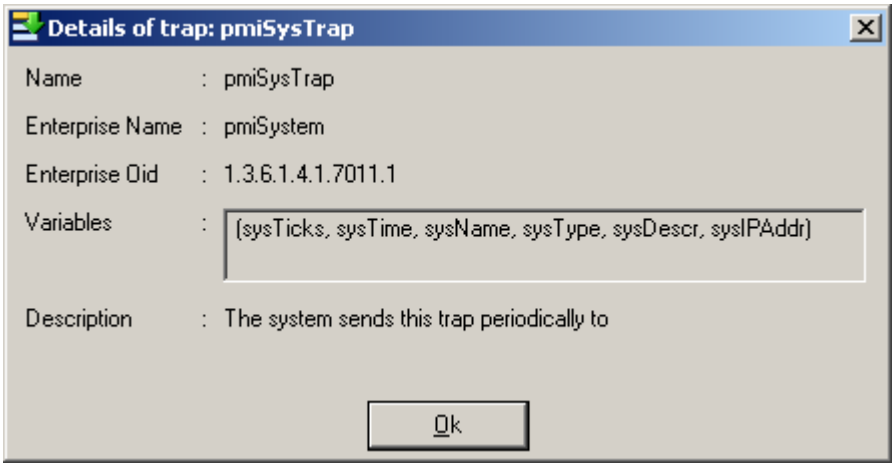



Table 24

| Field | Description |
|---|---|
| **Name** | Name of the trap. |
| **Enterprise Name** | Name of the enterprise that defined the trap. |

| Field | Description |
|---|---|
| **Enterprise Oid** | Object id of the trap.<br>1 – ISO<br>3 – ORG<br>6 – DOD<br>1 – Internet<br>4 – Private<br>1- Enterprise<br>7011 – pmiSystem<br>1 – pmiSystrap |
| **Variables** | varBinds associated with the trap. |
| **Description** | Description of the trap. |

# *Browsing MIB Tree*

This option enables you to search through the MIB tree for an MIB, OID, Identifier and Trap names. Searching with regular expression is not permitted.

## To browse the MIB tree

1. Open the MibCompiler console.
2. From the **Tools** menu, choose **Find**.

   (OR)

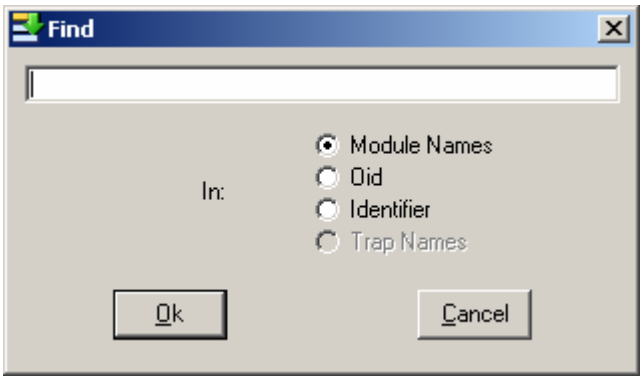
   Press **Ctrl + F** on your keyboard.

   (OR)

   Click  on the tool bar.

   MibCompiler displays “Find” dialog box.

Figure 105 Find dialog box



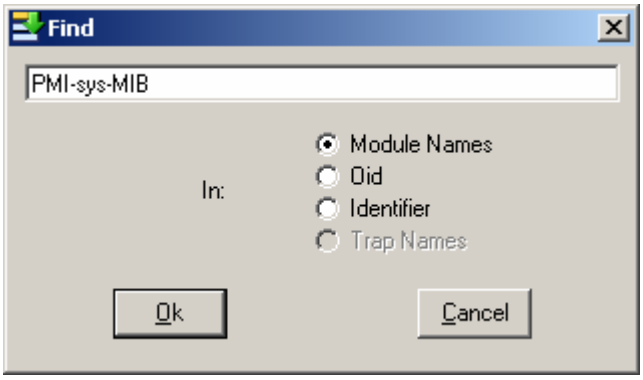

**Note**

By default, MibCompiler displays the MIB modules and the details of those modules. To view the trap details, follow the procedures in the **View Trap Details** section.

3 Type the name of a module in the search field.

4 Select **Module Names** option, if not selected.

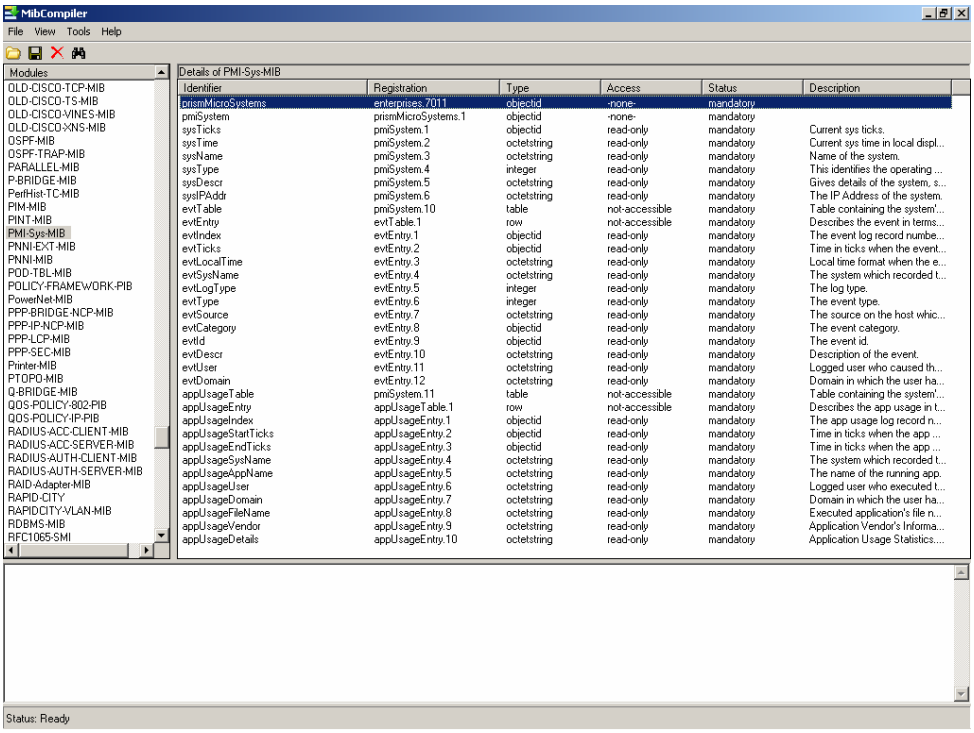
Figure 106 Find Module Name




5 Click **Ok**.

MibCompiler displays the MIB module and its details.

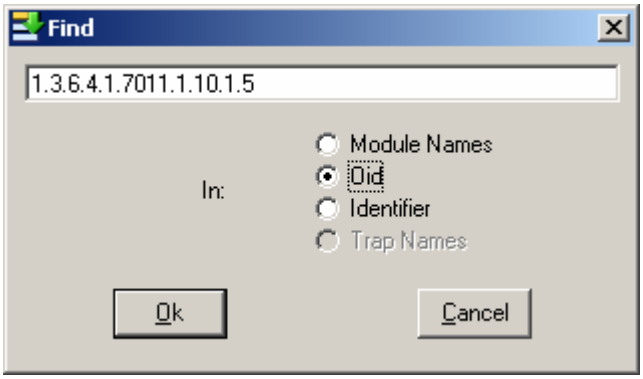
Figure 107 Search result.




6 Type an **OID** in the search field.

7 Select the **Oid** option.

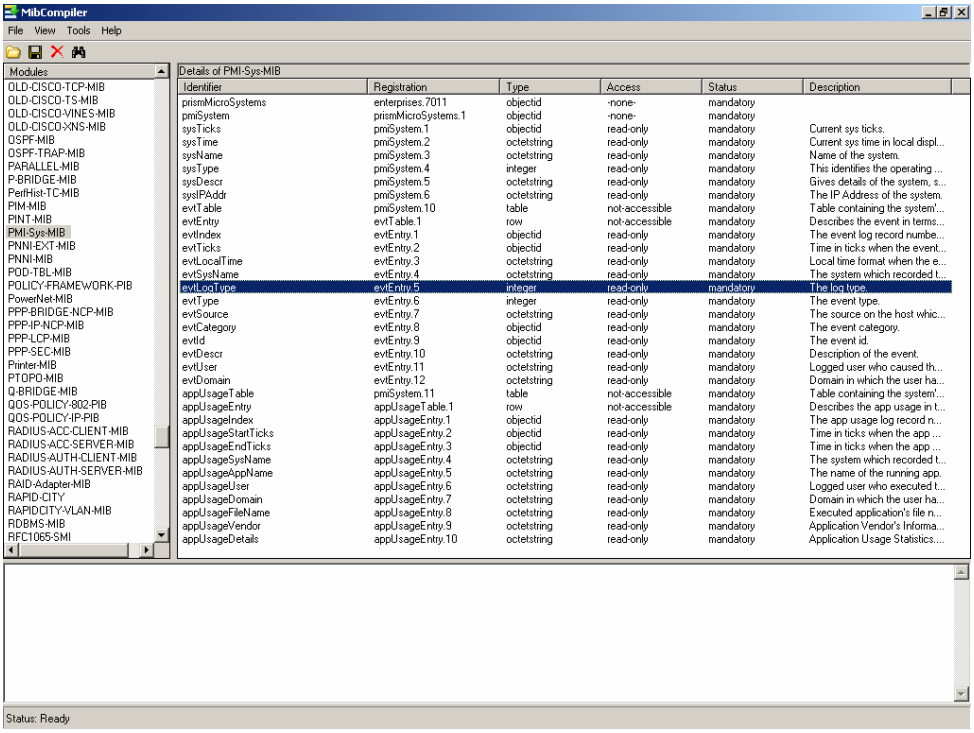
Figure 108 Find OID.




8 Click **Ok**.

MibCompiler displays the result.

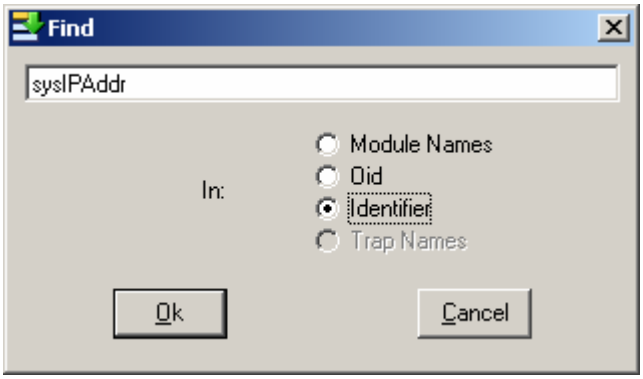
Figure 109 Search result




***9*** Type an **Identifier** in the search field.

***10*** Select the **Identifier** option.

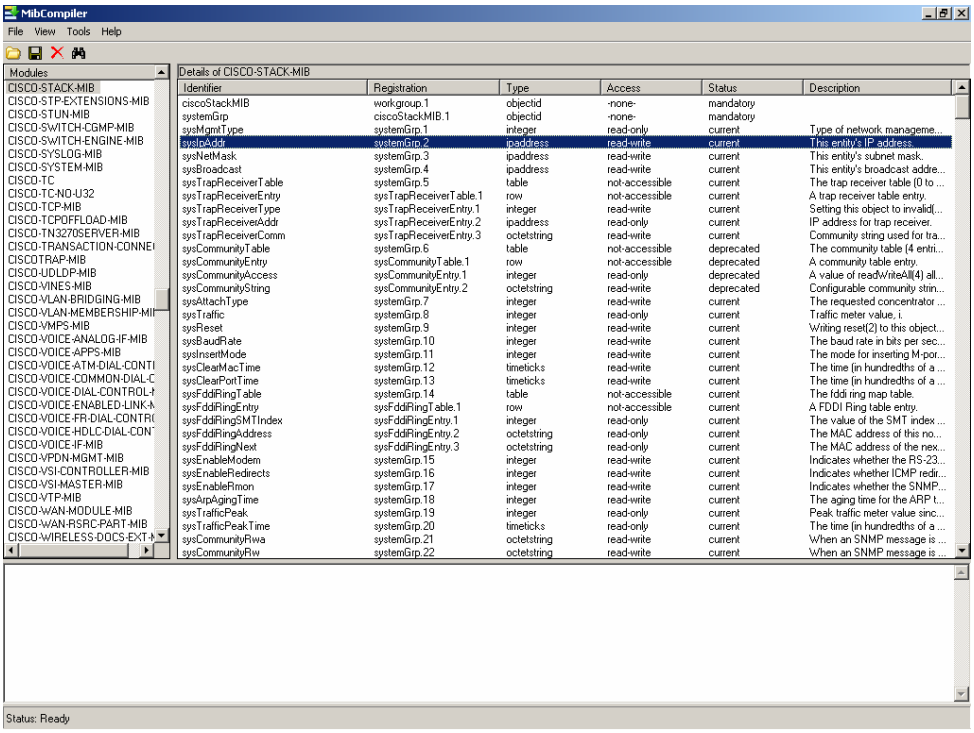
Figure 110 Find Identifier




***11*** Click **Ok**.

MibCompiler displays the result.

Figure 111 Search result.




## Searching Trap Details

This option enables you to search the Trap details.

### To search trap details

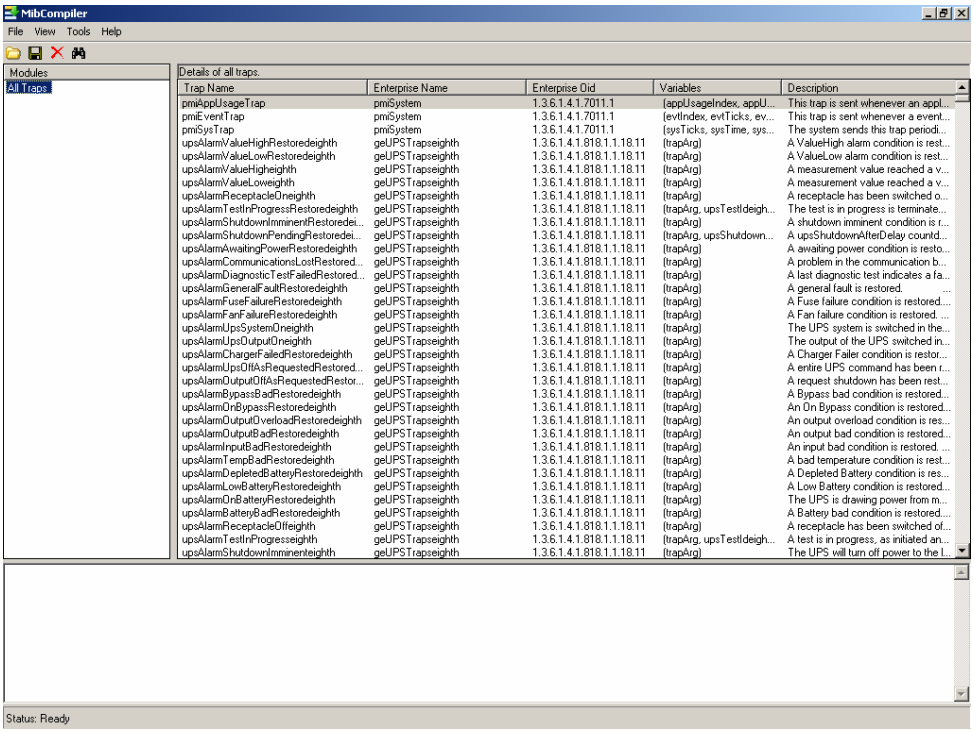
1. Open the MibCompiler console.
2. From the **View** menu, choose **Traps**.

   (OR)

   Press **Ctrl + T** on your keyboard.

   MibCompiler displays all trap details.

Figure 112 Details of all traps.




***3*** From the **Tools** menu, choose **Find**.

(OR)

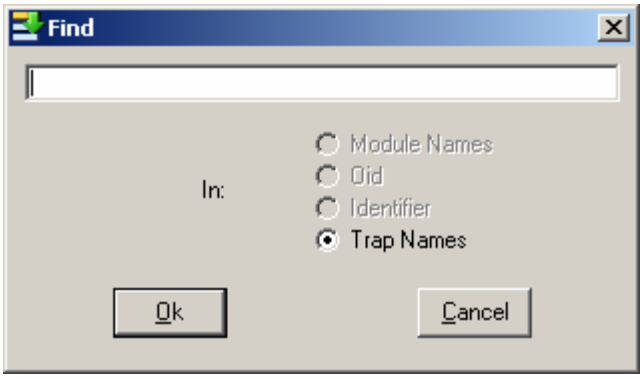
Press **Ctrl + F** on your keyboard.

(OR)

Click  on the tool bar.

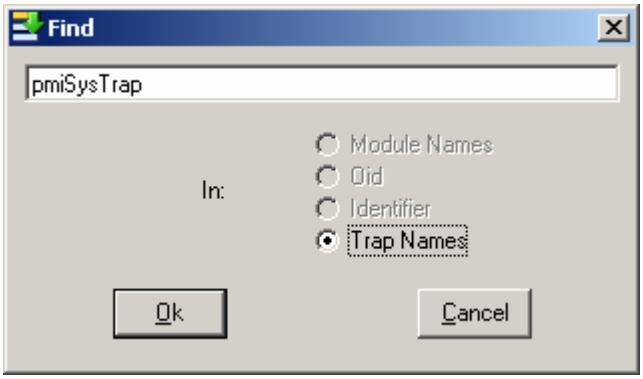
MibCompiler displays the “Find” dialog box.

Figure 113 Find dialog box




***4*** Type the name of the trap in the search field.

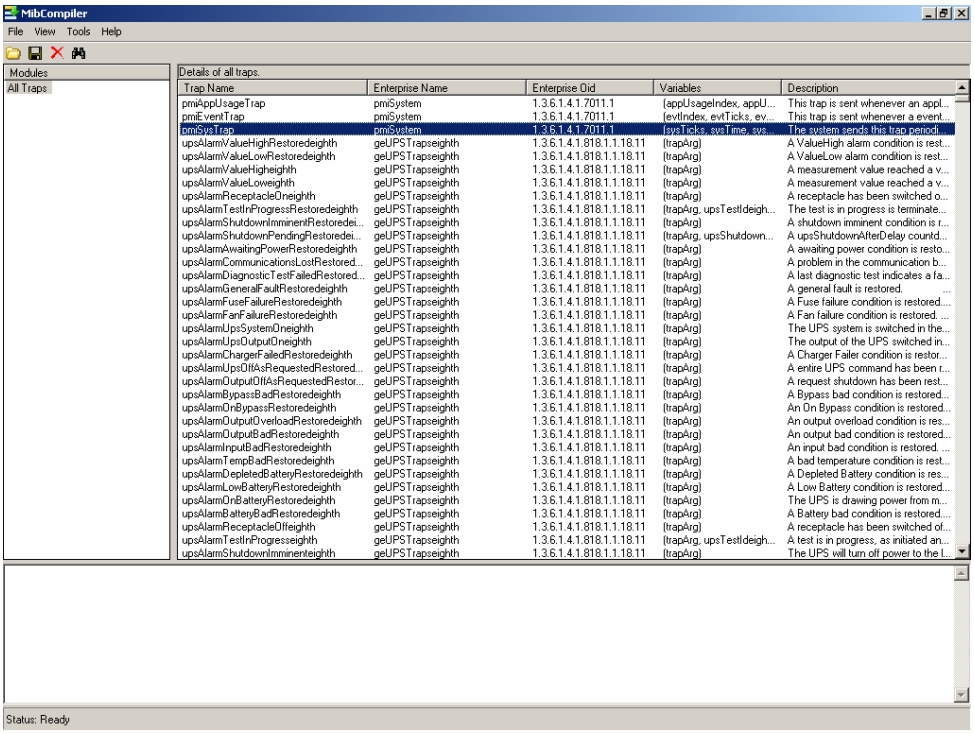
Figure 114 Find Trap name




***5*** Click **Ok**.

MibCompiler displays the result.

Figure 115 Search result




# *Deleting MIB*

This option enables you to delete a MIB from the bin file.

### To delete a MIB

1. Open the MibCompiler console.
2. Select the module that you want to delete from the “Modules” window.
3. From the **File** menu, choose **Delete**.

   (OR)

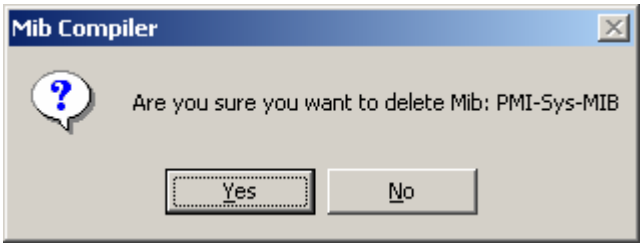
   Press **Ctrl + D** on your keyboard.

   (OR)

   Click  on the tool bar.

   MibCompiler displays the Mib Compiler confirmation message box.

Figure 116
MibCompiler message box.




4. Click **Yes**.

   MibCompiler displays the Mib Compiler message box with appropriate message.

Figure 117
MibCompiler message box.


5. Click **OK**.

# Exiting MibCompiler

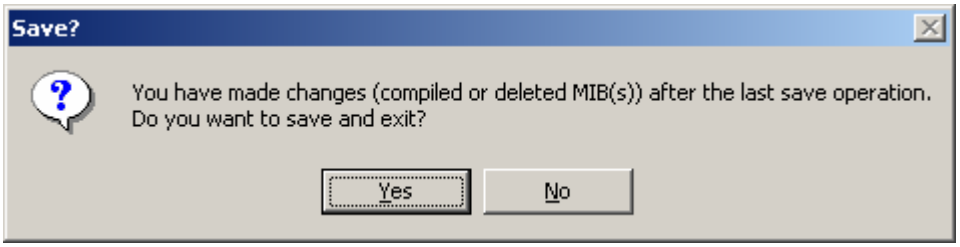
This option enables you to exit MibCompiler.

### To exit MibCompiler

1. From the **File** menu, choose **Exit**.

   MibCompiler displays the “Save?” confirmation message box.

Figure 118 Save? message box

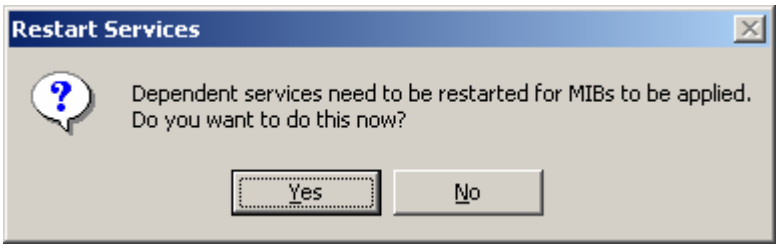



2 Click **Yes** to save the changes you have made and exit the MibCompiler.

3 Click **No** to exit MibCompiler without saving the changes.

MibCompiler displays the “Restart Services” message box.

Figure 119 Restart Service? Message box




4 Click **Yes** to restart the dependent services.

# Glossary

| Term | Description |
|---|---|
| **Alert Configuration** | Process of configuring alert notifications in the form of Sound, E-mail, Console message or any Custom action. |
| **Alerts** | A feature that instructs programs to notify timely information about the events. |
| **ASN-1** | Abstract Syntax Notation One (ASN.1) is an internationally accepted formal language or notation system used for describing data to be exchanged between distributed computer systems or by telecommunications protocols, regardless of language implementation, physical representation of these data, or type of application. It is a formal system for the specification of abstract data types. ASN.1 uses sets of encoding rules to transform data specified in the ASN.1 language into a standard format that can be decoded on any system that 'knows' the same rules. |
| **Audible Alert** | A feature that instructs programs that usually notifies information by sound. |
| **Console Message Alert** | A feature that instructs programs to notify information to the selected machine. |
| **Custom Alert** | A feature that instructs programs to execute custom action on receipt of an event. |
| **Email Alert** | A feature that instructs programs to notify information by E-mail. |
| **Entity** | Refers to both a server and a client. |
| **Event** | A condition or state of change that may cause a trap message to be generated. |
| **Filters** | The process to filter out events that you do not want to monitor. |
| **Historical Report** | The report generated based on the selection criteria. |

| Term | Description |
|---|---|
| **Hubs** | A common connection point for devices in a network. Hubs are commonly used to connect segments of a LAN. A hub contains multiple ports. When a packet arrives at one port, it is copied to the other ports so that all segments of the LAN can see all packets. A passive hub serves simply as a conduit for the data, enabling it to go from one device (or segment) to another. So-called intelligent hubs include additional features that enable an administrator to monitor the traffic passing through the hub and to configure each port in the hub. |
| **IP Subnet** | A 32-bit address used to identify a node on an IP internet. The address is typically represented with a decimal value of each octet separated by a period. For example: 192.168.7.27. |
| **Logfiles** | The process to monitor textual log files such as SQL or ISA logs, created by any vendor. You can also configure the strings to search. If any record matching the search string is found, an event will be generated. |
| **MIB** | Management Information Base; a collection of managed objects residing in a virtual information store. |
| **Network element** | Also known as a managed device- a hardware device, such as a PC or a router. |
| **NOC** | Network Operations Center. A location from which the operation of a network is monitored. Additionally, this center usually serves as a clearinghouse for connectivity problems and efforts to resolve those problems. |
| **Notification** | A message that indicates a status change (equivalent to a trap). |
| **Objects** | A passive entity that contains or receives information. Access to an object potentially implies access to the information it contains. Examples of objects are: records, blocks, pages, segments, files, directories, directory trees, and programs, as well as bits, bytes, words, fields, processors, video displays, keyboards, clocks, printers, network nodes, etc. |
| **OID** | An Object Identifier (OID) is the identification value of an object that is defined in a MIB. OIDs are arranged in a hierarchical tree structure compliant with Internet standard, that consists of roots and branches. An OID is written as a sequence of sub identifiers, starting with the tree root in dotted decimal notation. For example, the Microsoft branch of the MIB naming tree is expressed as 1.3.6.1.4.1.311. |

| Term | Description |
|---|---|
| **Parse** | Using algorithms to analyze data into components. Semantic parsing involves trying to figure out what the components mean. Lexical parsing refers to the process of deconstructing the data into components. |
| **Protocol** | A set of rules that computers use to communicate across networks on the internet. |
| **Router** | A device that determines the next network point to which a data packet should be forwarded enroute toward its destination. The router is connected to at least two networks and determines which way to send each data packet; based on its current understanding of the state of the networks it is connected to. Routers create or maintain a table of the available routes and use this information to determine the best route for a given data packet. |
| **SMI** | **SMI** stands for **Structure of Managed Information** and represents the notation by which an SNMP MIB must be written. Another way to look at SMI is that it is the grammar to write SNMP MIBs. There are two types of SMI: SMIv1 and SMIv2 with SMIv1 being the earlier version, of course, back in 1990.<br><br>**SMIv1** is the old notation that nobody should use any more. However, there are still a lot of SNMP MIBs written before SMIv2 came about in 1993.<br><br>**SMIv2** is the new notation that you should use whenever you create a new MIB. |
| **SNMP** | Simple Network Management Protocol -- A set of standards for communication with devices connected to a TCP/IP network. Examples of these devices include routers, hubs, and switches. A device is said to be “SNMP compatible” if it can be monitored and/or controlled using SNMP messages. SNMP messages are known as “PDU's” - Protocol Data Units. Devices that are SNMP compatible contain SNMP “agent” software to receive, send, and act upon SNMP messages. Software for managing devices via SNMP are available for every kind of commonly used computer and are often bundled along with the device they are designed to manage. |
| **SNMP Community Strings** | An SNMP community string is a text string that acts as a password. It is used to authenticate messages that are sent between the management station (the SNMP manager) and the device (the SNMP agent). The community string is included in every packet that is transmitted between the SNMP manager and the SNMP agent. |

| Term | Description |
|---|---|
| **SNMP Traps** | The process to receive trap messages generated by local or remote SNMP agents And forwards the messages to third party vendor software such as an NOC. |
| **Switch** | A device that improves network performance by segmenting the network and reducing competition for bandwidth. When a switch port receives data packets, it forwards those packets only to the appropriate port for the intended recipient. This further reduces competition for bandwidth between the clients, servers or workgroups connected to each switch port. |
| **TCP** | Transmission Control Protocol. TCP is responsible for verifying the correct delivery of data from Agent to server. TCP adds support to detect errors or lost data and to trigger transmission until the data is correctly and complete received. |
| **Trap** | Message sent by an SNMP server to a client to indicate the occurrence of a significant event, such as a specifically defined condition or a threshold that was reached. Managed devices use traps to asynchronously report certain events to clients. |
| **UDP** | User Datagram Protocol. A connectionless protocol that, like TCP, runs on top IP networks. Unlike TCP/IP, UDP/IP provides very few error recovery services, offering instead a direct way to send and receive datagrams over an IP network. |

# Index

## R



## S



## T



## U



## W