



## PageR Device Manager™

PageR Device Manager (PDM) is a freeware tool that enables you to easily find SNMP devices on the network such as temperature monitoring devices. It browses the Management Information Base (MIB) exposed by systems with SNMP agents.

This makes it easier to set up **PageR Enterprise** Monitored Objects.

### How it works

PageR Device Manager extracts or "walks" the MIB of a selected SNMP agent system and creates a structured view of the agent's MIB. A tree view of the MIB allows quick exploration of the MIB to locate objects of interest. The current value of MIB objects are displayed along with object descriptions. You can also view the MIB in a one dimensional list for a visual scan of the entire MIB.



Once loaded, the object values can be updated with a click of the mouse.

### Quick Start Hint

Click the AUTO DISCOVERY  button (3rd from the left).

### PDM also acts as a Trap Manager

PageR Device Manager can also act as an SNMP Trap manager. In this mode, PDM listens for and receives and logs any traps sent to the system on which PDM is executing. Traps can be displayed along with descriptive information.

## PDM Main Screen

When PDM first comes up, it will show nothing in either the MIB tree pane (left) or the MIB detail pane (right). You must set the name or IP address of the target SNMP agent system by

clicking the  button. Once an agent is selected, you set the community name used by the agent with the  button. The default community is "public". When the address and community have been set, click the  button to walk the MIB.

During the MIB walk, PDM reads the entire MIB from the agent into local memory. Then the retrieved MIB is formatted for viewing by loading it into the MIB tree pane and optionally the MIB detail pane. When formatting is completed, you can use either pane to explore the MIB.

You can switch between the tree driven MIB detail view and the full MIB dump view with the  button. When MIB objects appear in the right pane (as opposed to containers) the last retrieved value for the object is shown. You can left click on the object to update the value and display the object's description below the detail pane.

## Selecting Agent Systems

To select the Agent system whose MIB will be walked, click the  button. This will display a screen where you can enter the IP address or host name of the desired system. You can also select from previously entered values.

You can click the  button to see a list of known Agent systems. If the list has not been populated, you will be prompted to perform an auto discovery scan of your network to locate available Agent systems. You can directly perform an auto discovery by clicking the  button.

During an auto discovery, PDM sends queries to the IP addresses in the range to be scanned to determine if a system exists at that IP address and if a system exists, does it have an SNMP Agent available. The auto discovery screen shows the Agent systems found and you may click on a system to select it for MIB walking.

Note that during auto discovery, only Agents whose community match the currently selected community name will respond to the discovery query.

## Community Name

The Community name is set by clicking the  button. The Community name defaults to "public". The Community name acts like a password and is more intended to control access to SNMP Agents. Each Agent is configured with a Community name that it will respond to. SNMP management applications, like PDM, must supply the matching Community name to get a response from Agents.

## Viewing the Retrieved MIB

Once the Agent MIB has been walked, it is formatted for display. A tree view of the MIB is displayed in the left pane and the right pane reflects navigation through the tree. You can also navigate by clicking items in the right pane.

An SNMP MIB is a list of objects that represent actual data items that are tracked and reported by the Agent. These objects are organized into a hierarchy of containers or folders. Using the tree view and the detail view you navigate the MIB containers to locate the data object you are interested in.

When data objects are displayed in the right pane, their name, data type and current value are displayed. You can click on the object to update its value (the Agent is queried) and display the detailed description of the object (if known) in the description pane (lower right).



You can also click the  button to toggle Full List mode. When Full List is turned on, the entire MIB is listed at the object level in the right pane. If you click on an object in the right pane, the tree view will be adjusted to show the location of the object in the container hierarchy.

## Options Screen

This screen allows adjustment of PageR Device Manager basic settings.

### **SNMP Get/Set Port**

Sets the UDP port number that GET requests are sent on. 161 is the standard port.

### **SNMP Trap Send Port**

Sets the UDP port number that Traps are sent on. 162 is the standard port.

### **SNMP Trap Listen Port**

Sets the UDP port number that is used to listen for Traps. 162 is the standard port.

### **Timeout**

Sets the timeout for GET requests in seconds.

## About MIBs

A MIB or Management Information Base, is a data structure that defines the data objects exposed by an SNMP Agent. The basic entity in a MIB is an Object, or a unique identifiable data item. Objects are placed into a multi level hierarchy of containers to organize the many objects in a typical MIB in a logical manner.

The MIB in an Agent is defined by a schema language and stored in a text file. The schema itself is also called a MIB. This can be a bit confusing as the "MIB" for an agent is both a physical implementation in the Agent and a definition or schema for that implementation stored in a text file. This is important because the schema or MIB file contains detailed descriptive information about the MIB objects and containers that are not stored in or reported by Agents.

A MIB object is identified by a unique Object Identifier. The object further has a name, a data type, a textual description and a number of other attributes. Communication between an Agent and Management application is by the Object Identifier alone. When queried for an object or list of objects, an Agent will return the object identifier(s) and data values requested and nothing else. In order for a management application such as PDM to present the Agent MIB in a useful manner, ie with container names, object names, data types (and appropriate formatting) and object descriptions, the manager must link the object ids returned by the Agent to object schema information stored in the MIB file.

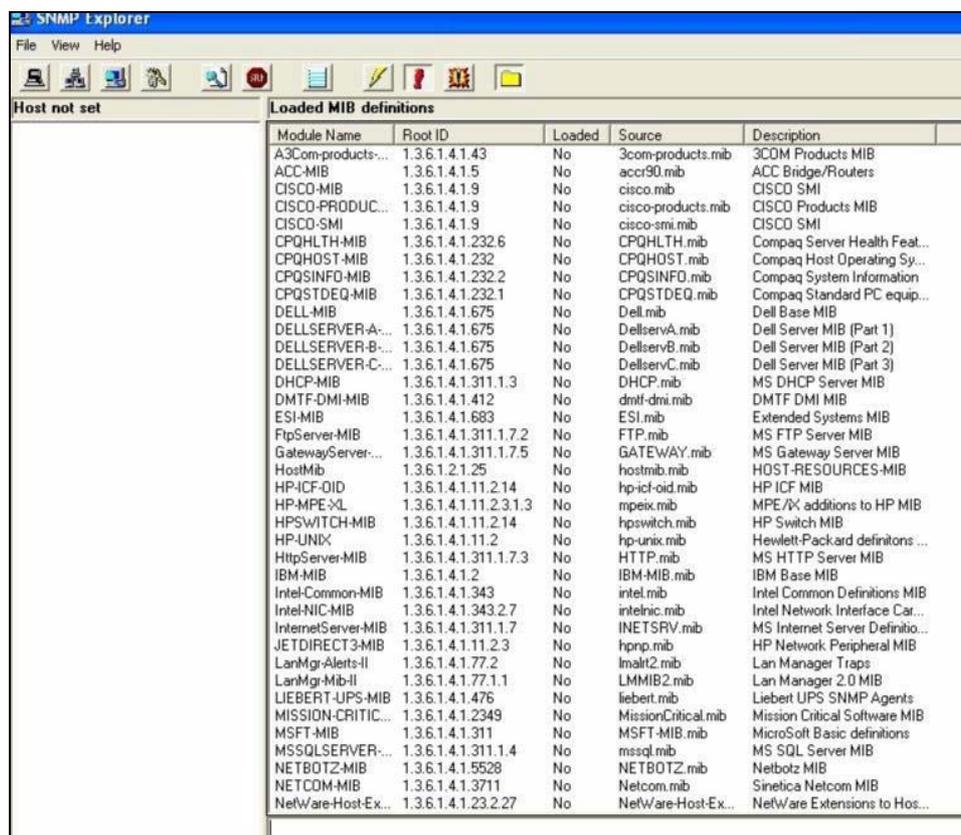
It is for this reason that SNMP applications ship with a set of MIB schema files. These files are compiled by the manager application so that it can present a useful view of the object data values returned by Agents.

## Loaded MIB List

PageR Device Manager ships with a set of common MIB files stored in the Mibs directory. Each MIB file is described in the MibRoot.txt file. When PDM starts up, the MibRoot.txt file is loaded into memory so the PDM knows what MIB files are available. MIB files themselves are not loaded at start up except for the set of files that describe the standard MIB object set required of all SNMP Agents.

As Agents are walked, PDM will load MIB files as appropriate. A MIB file is not required to access an Agent but the descriptive information available about retrieved objects will be limited without a MIB file.

You can click the  to see a list of the MIB files that PDM knows about and see which ones are loaded. You can force a MIB file to be loaded by clicking on it.



The screenshot shows the 'SNMP Explorer' application window. The 'Host not set' pane is empty, and the 'Loaded MIB definitions' pane displays a table of MIB files. The table has columns for Module Name, Root ID, Loaded status, Source, and Description.

Module Name	Root ID	Loaded	Source	Description
A3Com-products...	1.3.6.1.4.1.43	No	3com-products.mib	3COM Products MIB
ACC-MIB	1.3.6.1.4.1.5	No	accr90.mib	ACC Bridge/Routers
CISCO-MIB	1.3.6.1.4.1.9	No	cisco.mib	CISCO SMI
CISCO-PRODUC...	1.3.6.1.4.1.9	No	cisco-products.mib	CISCO Products MIB
CISCO-SMI	1.3.6.1.4.1.9	No	cisco-smi.mib	CISCO SMI
CPQHLTH-MIB	1.3.6.1.4.1.232.6	No	CPQHLTH.mib	Compaq Server Health Feat...
CPQHST-MIB	1.3.6.1.4.1.232	No	CPQHST.mib	Compaq Host Operating Sy...
CPQSINFO-MIB	1.3.6.1.4.1.232.2	No	CPQSINFO.mib	Compaq System Information
CPGSTDEQ-MIB	1.3.6.1.4.1.232.1	No	CPGSTDEQ.mib	Compaq Standard PC equip...
DELL-MIB	1.3.6.1.4.1.675	No	Dell.mib	Dell Base MIB
DELLSERVER-A...	1.3.6.1.4.1.675	No	DellservA.mib	Dell Server MIB (Part 1)
DELLSERVER-B...	1.3.6.1.4.1.675	No	DellservB.mib	Dell Server MIB (Part 2)
DELLSERVER-C...	1.3.6.1.4.1.675	No	DellservC.mib	Dell Server MIB (Part 3)
DHCP-MIB	1.3.6.1.4.1.311.1.3	No	DHCP.mib	MS DHCP Server MIB
DMTF-DMI-MIB	1.3.6.1.4.1.412	No	dmtf-dmi.mib	DMTF DMI MIB
ESI-MIB	1.3.6.1.4.1.683	No	ESI.mib	Extended Systems MIB
FtpServer-MIB	1.3.6.1.4.1.311.1.7.2	No	FTP.mib	MS FTP Server MIB
GatewayServer...	1.3.6.1.4.1.311.1.7.5	No	GATEWAY.mib	MS Gateway Server MIB
HostMib	1.3.6.1.2.1.25	No	hostmib.mib	HOST-RESOURCES-MIB
HP-ICF-OID	1.3.6.1.4.1.11.2.14	No	hp-icf-oid.mib	HP ICF MIB
HP-MPE>XL	1.3.6.1.4.1.11.2.3.1.3	No	mpeix.mib	MPE/IX additions to HP MIB
HPSWITCH-MIB	1.3.6.1.4.1.11.2.14	No	hpswitch.mib	HP Switch MIB
HP-UNIX	1.3.6.1.4.1.11.2	No	hp-unix.mib	Hewlett-Packard definitions ...
HttpServer-MIB	1.3.6.1.4.1.311.1.7.3	No	HTTP.mib	MS HTTP Server MIB
IBM-MIB	1.3.6.1.4.1.2	No	IBM-MIB.mib	IBM Base MIB
Intel-Common-MIB	1.3.6.1.4.1.343	No	intel.mib	Intel Common Definitions MIB
Intel-NIC-MIB	1.3.6.1.4.1.343.2.7	No	intelnic.mib	Intel Network Interface Car...
InternetServer-MIB	1.3.6.1.4.1.311.1.7	No	INETSrv.mib	MS Internet Server Definitio...
JETDIRECT3-MIB	1.3.6.1.4.1.11.2.3	No	hnpn.mib	HP Network Peripheral MIB
LanMgr-Alerts-II	1.3.6.1.4.1.77.2	No	lmal2.mib	Lan Manager Traps
LanMgr-Mib-II	1.3.6.1.4.1.77.1.1	No	LMMIB2.mib	Lan Manager 2.0 MIB
LIEBERT-UPS-MIB	1.3.6.1.4.1.476	No	liebert.mib	Liebert UPS SNMP Agents
MISSION-CRITIC...	1.3.6.1.4.1.2349	No	MissionCritical.mib	Mission Critical Software MIB
MSFT-MIB	1.3.6.1.4.1.311	No	MSFT-MIB.mib	Microsoft Basic definitions
MSSQLSERVER...	1.3.6.1.4.1.311.1.4	No	mssql.mib	MS SQL Server MIB
NETBOTZ-MIB	1.3.6.1.4.1.5528	No	NETBOTZ.mib	Netbotz MIB
NETCOM-MIB	1.3.6.1.4.1.3711	No	Netcom.mib	Sinetica Netcom MIB
NetWare-Host-Ex...	1.3.6.1.4.1.23.2.27	No	NetWare-Host-Ex...	NetWare Extensions to Hos...

## Adding Your Own MIBs

You can add your own MIB files to PDM's list of MIBs. Place your files in the Mibs directory. Add a definition for your MIBs to the **MibRoot.txt** file. Then test against an Agent system to see if object names and descriptions appear.

The MibRoot.txt file contains definitions of MIB files in XML format. The format for a MIB file entry is described below:

```
<IBM-MIB load="auto" display="yes">  
  <Module>IBM-MIB</Module>  
  <RootId>1.3.6.1.4.1.2</RootId>  
  <Desc>IBM Base MIB</Desc>  
  <File>IBM-MIB.mib</File>  
</IBM-MIB>
```

The **<block-name** and **</block-name** elements define the start and end of a MIB file definition. The block-name is free but by convention is the same as the Module Name. The **load** attribute can be "yes" or "auto". Yes means the MIB will be loaded at start up time. Auto means the MIB will be loaded if needed. Always set the **display** attribute to "yes".

The **<Module>** element is the name of the MIB. This is found on the DEFINITIONS ::= BEGIN line at the top of the MIB file.

The **<RootID>** element is the Object ID of the highest Level item in the MIB.

The **<Desc>** element is a description of the MIB file and displayed on the MIB List screen.

The **<File>** element defines the name of the MIB file. This file is expected to be in the same directory as the MibRoot.txt file.

MIB definition blocks should be placed in the MibRoot.txt file in order sorted by the root object identifier. Out of order definitions may not be automatically loaded when the Agent is accessed.

When PDM loads a MIB file, that MIB file is compiled based on the standards that describe the MIB schema language. If detectable errors occur, a window will appear describing the error. Due to limitations in the PDM MIB compiler, some errors are not reported but cause the MIB to load incorrectly. If a MIB file loads without errors but does not display object names and descriptions correctly or seems to interfere with the correct operation of other MIBs, contact technical support for assistance.

MIB files typically import some number of objects or other definitions from other MIBs. If your MIB imports from other MIBs, those MIBs must be loaded before your MIB. In most cases, the MIBs typically imported are part of the standard set of MIBs delivered with PDM and will already be loaded.

## Sending a Trap

SNMP Traps are alert messages sent by Agents to Management applications. SNMP Explorer supports both sending traps and receiving traps.

Sending Traps is mainly used to test other Management applications. You can send a Trap by clicking the  button. You are prompted for the IP address or name of the system where the trap will be sent, the Enterprise ID for the desired Trap family and the specific Trap number.

## Receiving Traps

PageR Device Manager can act as a Trap manager. Click the  button to enable receiving of Traps. Traps are received and logged for review. Click the  button to view the Trap Log.

Note that only one application may act as Trap Manager on a system. If any other Trap Manager, such as the Windows SNMP Trap Service is running, enabling Traps in PDM will fail.

## Appendix 1 – A MIB Tutorial

A MIB or Management Information Base, is a data structure that defines the data objects exposed by an SNMP Agent. The basic entity in a MIB is an Object, or a unique identifiable data item. Objects are placed into a multi level hierarchy of containers to organize the many objects in a typical MIB in a logical manner.

The MIB in an Agent is defined by a schema language and stored in a text file. The schema itself is also called a MIB. This can be a bit confusing as the "MIB" for an agent is both a physical implementation in the Agent and a definition or schema for that implementation stored in a text file. This is important because the schema or MIB file contains detailed descriptive information about the MIB objects and containers that are not stored in or reported by Agents.

A MIB object is identified by a unique Object Identifier. The object further has a name, a data type, a textual description and a number of other attributes. Communication between an Agent and Management application is by the Object Identifier alone. When queried for an object or list of objects, an Agent will return the object identifier(s) and data values requested and nothing else. In order for a management application such as PageR Device Manager to present the Agent MIB in a useful manner, ie with container names, object names, data types (and appropriate formatting) and object descriptions, the manager must link the object ids returned by the Agent to object schema information stored in the MIB file.

It is for this reason that PageR Device Manager applications ship with a set of MIB schema files. These files are compiled by the manager application so that it can present a useful view of the object data values returned by Agents.

PageR Device Manager ships with a set of common MIB files stored in the Mibs directory. Each MIB file is described in the MibRoot.txt file. When PageR Device Manager starts up, the MibRoot.txt file is loaded into memory so the PageR Device Manager knows what MIB files are available. MIB files themselves are not loaded at start up except for the set of files that describe the standard MIB object set required of all SNMP Agents.

As Agents are walked, PageR Device Manager will load MIB files as appropriate. A MIB file is not required to access an Agent but the descriptive information available about retrieved objects will be limited without a MIB file.

You can click an icon to see a list of the MIB files that PageR Device Manager knows about and see which ones are loaded. You can force a MIB file to be loaded by clicking on it.

You can add your own MIB files to PageR Device Manager's list of MIBs. Place your files in the Mibs directory. Add a definition for your MIBs to the MibRoot.txt file. Then test against an Agent system to see if object names and descriptions appear.

The MibRoot.txt file contains definitions of MIB files in XML format. The format for a MIB file entry is described below:

```
<IBM-MIB load="auto" display="yes">
  <Module>IBM-MIB</Module>
  <RootId>1.3.6.1.4.1.2</RootId>

  <Desc>IBM Base MIB</Desc>
  <File>IBM-MIB.mib</File>
</IBM-MIB>
```

The <block-name and </block-name elements define the start and end of a MIB file definition.

The block-name is free but by convention is the same as the Module Name. The load attribute can be "yes" or "auto". Yes means the MIB will be loaded at start up time. Auto means the MIB will be loaded if needed. Always set the display attribute to "yes".

The <Module> element is the name of the MIB. This is found on the DEFINITIONS ::= BEGIN line at the top of the MIB file.

The <RootID> element is the Object ID of the highest Level item in the MIB.

The <Desc> element is a description of the MIB file and displayed on the MIB List screen.

The <File> element defines the name of the MIB file. This file is expected to be in the same directory as the MibRoot.txt file.

MIB definition blocks should be placed in the MibRoot.txt file in order sorted by the root object identifier. Out of order definitions may not be automatically loaded when the Agent is accessed.

When PageR Device Manager loads a MIB file, that MIB file is compiled based on the standards that describe the MIB schema language. If detectable errors occur, a window will appear describing the error. Due to limitations in the PageR Device Manager MIB compiler, some errors are not reported but cause the MIB to load incorrectly. If a MIB file loads without errors but does not display object names and descriptions correctly or seems to interfere with the correct operation of other MIBs, contact technical support for assistance.

MIB files typically import some number of objects or other definitions from other MIBs. If your MIB imports from other MIBs, those MIBs must be loaded before your MIB. In most cases, the MIBs typically imported are part of the standard set of MIBs delivered with PageR Device Manager and will already be loaded

(c)2012 CPL Systems All Rights Reserved.