SIMPLESOFT Inc.

SimpleMIBBrowser

User's Guide

For Technical Support:

Email: support@smplsft.com

Phone: (650) 965-4515

SIMPLESOFT Inc.
257 Castro Street, Suite 220,
Mountain View, CA 94041

Printed in the United States of America.

Trademarks

February 2007

Contents

Figures

Preface

Welcome to SimpleMIBBrowser, an easy to use, graphical tool that simplifies the process of interacting with SNMP enabled devices to retreive and configure management information.

Management Information Base (MIB) ASN.1 documents can be loaded into the SimpleMIBBrowser and their contents displayed in a graphical tree. Queries can then be generated by selecting nodes on the managed object tree and sent to the device and the responses shown in a tabular fashion.

Using the SimpleMIBBrowser, you can quickly and easily make SNMP-Get, SNMP-GetNext, SNMP-Set and also SNMP-GetBulk queries on any device and view the results in a tabular fashion. Multiple device profiles can also be stored for quickly establishing communication in the future.

SimpleMIBBrowser helps in:

- Making SNMP-GetNext (or walk) queries on the agent.
- Making SNMP-Set queries on the agent.
- Making SNMP-Get queries on the agent.
- Making SNMP-GetBulk queries on the agent.
- Displaying the contents of MIB tables in a corresponding tabular results form.
- Displaying Traps/Notifications/Informs received from agents.
- Conducting polling of performance related variables and displaying the results in a graph.
- Specifying various communication parameters.
- Comparing the results of the previous query and the latest query on same set of objects.

About Managers, Agents, MIBs and SNMP

Managers and Agents are able to communicate with each other because they share a common understanding of the data being exchanged and use a common mechanism to exchange that data.

In the world of SNMP, the common understanding of what data is being exchanged is specified through SNMP MIB definition files, while the common mechanism for data exchange is defined in the SNMP Protocol specification.

The SimpleMIBBrowser can be used to gather information from any SNMP enabled device as it supports the standard SNMP protocol and understands the underlying management data by importing the associated MIB definitions.

Hardware and Software Requirements

SimpleMIBBrowser requires one of the following operating systems:

- Microsoft Windows 98, WindowsNT, Windows2000 or WindowsXP
- Red Hat Linux 9.0/PC

Documentation Conventions

To help you locate and interpret information easily, this guide uses consistent visual cues and a few standard text formats listed in Table 1:

Table 1: Documentation Conventions

Convention	Meaning
Bold	Denotes characters that you must type exactly as they appear. For example, if you are directed to type **dir:**, you should type all the bold characters exactly as they are printed.
Arial font	Denotes menu names, command names, and file names.
<*italic text*>	Denotes a placeholder or variable. You must provide the actual value. For example, the statement set snmp getcommunity <*community string*> requires you to substitute a value for the <*community string*> parameter.
|	Separates an either/or choice.
SMALL CAPITALS	Indicates names of keys, key sequences, and key combinations. For example, ALT+SPACEBAR.
FULL CAPITALS	Indicates directory names and paths.
`Monospace`	Sets off code examples and shows syntax spacing.

Chapter 1
Installing SimpleMIBBrowser

This chapter describes how to install the SimpleMIBBrowser software on your computer.

A version of the SimpleMIBBrowser may get automatically installed along with some of SimpleSoft's other applications such as the SimpleAgentPro.

Installing SimpleMIBBrowser on Windows

To install the standalone version of SimpleMIBBrowser:

1. Start MS Windows 98/NT/2000/XP.
2. Insert the installation disk in your CD drive.
3. Invoke setup.exe found in the Win directory.
4. Follow the instructions on the screen.

   The install program will begin copying files from the CDROM to the directory you specify. By default, it copies the install files to the C:\Program Files\SimpleMIBBrowser directory. After copying the files, SimpleMIBBrowser creates the Windows Program Group 'SimpleMIBBrowser' and adds items to it.

   The SimpleMIBBrowser uses software license keys for software protection. Please contact SimpleSoft to request a key by sending an email to Email to sales@simplesoftinc.com and specify when you purchased the product.

   The information on how to obtain a software license key is also displayed when you first start the SimpleMIBBrowser application.

Installing SimpleMIBBrowser on Linux

To install the standalone version of SimpleMIBBrowser:

1. Start Unix.

2. Insert the installation disk in your CD drive.

3. Invoke install_smbrowser_linux_nn.sh.

4. Follow the instructions on the screen.

   The install program will begin copying files from the CDROM to the directory you specify. By default, it copies the install files to the /opt/smbrowser directory. After installing the files, add /opt/smbrowser/bin to the path environment variable and create an environment variable called, "SMB_DIR" and set it to the location of the SimpleMIBBrowser's installation.

5. The SimpleMIBBrowser uses software key for its software protection. Please contact SimpleSoft to request a license key by sending an email to sales@simplesoftinc.com.

   The information on how to obtain a software license key is also displayed when you first start the SimpleMIBBrowser application.

SimpleMIBBrowser Directory Structure

The directory structure created by the SimpleMIBBrowser installation will look like the following:

smbrowser (or install directory)

- **bin** (executables, main working directory)
- **doc** (manual, web help related pages)
- **images** (icons)
- **mib** (input mib files)
- **cmf** (ouput compiled mib files)
- **log** (ouput log files)

The install program adds the following files and subdirectories to the SMBrowser directory:

- smbrowser/bin

SimpleMIBBrowser.exe	main SimpleMIBBrowser executable (Windows only)
SimpleMIBBrowser	main SimpleMIBBrowser executable (Linux only)
qt-mt311.dll	QT dynamic library (Windows only)
SMBrowser.ini	settings/initialization file

- SMBrowser/images

**.png*	Images and icons used in the program

- SMBrowser/doc

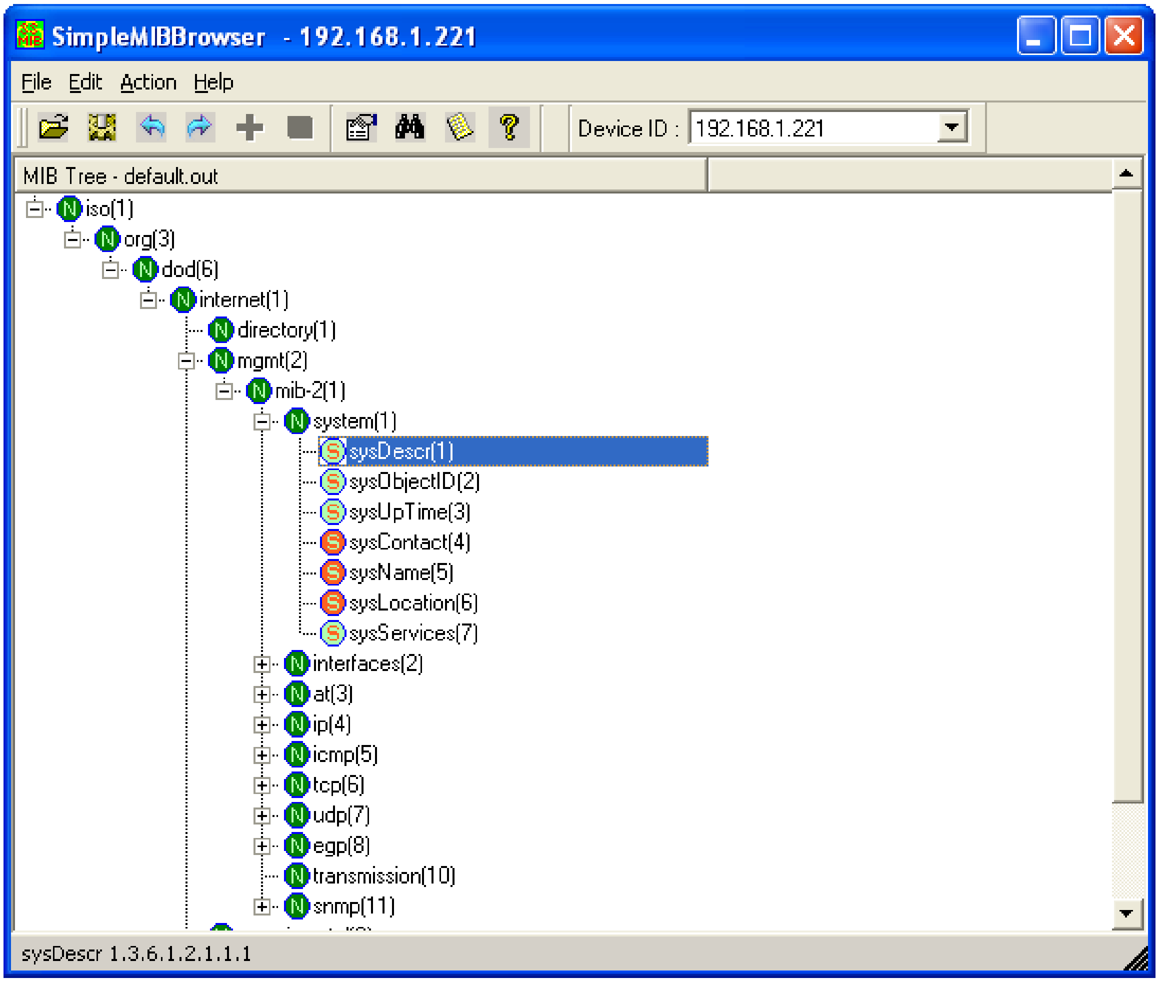
SMBrowser.pdf	Acrobat format manual
Webhelp/.html*	Web help related files

- SMBrowser/mibs

**.mib*	some standard MIB definition file

Chapter 2
Using SimpleMIBBrowser: An Overview

SimpleMIBBrowser is an easy to use application. To begin using this application to communicate with an SNMP enabled device, a few simple steps need to be followed. These steps are described in this chapter.



Device ID : 192.168.1.221

sysDescr 1.3.6.1.2.1.1.1

Figure 1: Main Screen

Basic Steps

The SimpleMIBBrowser requires two main pieces of information to communicate with an SNMP enabled device:

1. Information in the MIB definition files (default RFC1213 or MIB2 is already loaded).

2. SNMP communication settings that match those on the device.

With these two pieces of information, you can start talking SNMP with the device and retrieve data as well as configure it. The overall process is described below in general and in detail in the chapters to follow.

The basic steps for using the SimpleMIBBrowser are as follows:

1. Configure the device communication parameters.

   Use the ***Edit → Preferences*** menu option to configure communication parameters for interacting with the SNMP enabled device. See Chapter 3, Specifiying Communication Settings, for more details. Multiple device profiles can be stored and the currently remembered ones are displayed in an optional Device list panel to the left.

2. Retrieve and Configure MIB data from the device.

   You can now select any of the manageable objects displayed in the MIB tree and initiate SNMP queries for it. See Chapters 6, Retrieving Data, on how to get information about the selected node.

   See Chapter 7, Setting/Configuring Data, for more details on changing the values associated with managed objects.

   See Chapter 6, Retrieving Data, on generating SNMP Get/GetBulk requests to the SNMP enabled device.

   See Chapter 10, View Log, for more details on seeing the traces of the packets sent and received.

Menu Options in SimpleMIBBrowser

The Menu Options are described below:

Figure 2 : Main Menu Bar

- **File**: This menu option contains the following:

  1. **Open:** You can use this menu to open an existing Compiled MIB file that was created earlier by the SimpleMIBBrowser or some other SimpleSoft application. By default the current MIB repository is automatically opened.

  2. **Exit:** You can use this menu to close the application

- **Edit**: This menu option contains the following:

  1. **View Log:** You can use this menu option to open a log file. The Log file contains error and packet tracing information if enabled in the settings.

  2. **Show/Hide Device View:** You can use this menu to show/hide the device list panel to the left. The previously entered device profiles are displayed in this panel.

  3. **Find:** You can use this menu option to locate a variable by its name from the tree view.

  4. **Preferences:** You can use this menu to set the Communication Parameters to interact with the SNMP enabled device**.**

- **Action**: This menu option contains the following:

  1. **Get(walk):** You can use this menu option to to start a 'Get-Next' query on the agent. The results are displayed in a popup dialog in a tabular form.

  2. **Set**: You can use this menu option to set one or more writable variables in one go. *Set Variables* dialog box is displayed upon selecting this option, containing a list of all writable variables under the sub-tree of the selected node.

  3. **Add Row**: You can use this menu option to add a row to a table. This is enabled only when one table node or entry node is selected. *Table Query*

Results dialog box is displayed when this option is selected. The dialog box shows all the columns of the table, irrespective of whether they are Read-Only or writable. Instance values along with the column values have to be filled in.

4. **Get Table**: You can use this menu option to view the results of the query in a tabular format. This is enabled only when one table node or entry node is selected. The results are displayed in the *Table Query Results* dialog box, as and when they are available.

5. **SNMP Get**: You can use this menu option to start a SNMP Get query on the agent.

6. **SNMP GetBulk**: You can use this menu option to start a SNMP GetBulk query on the agent.

7. **Set From File**: You can use this menu option to load the .val file in a 'Set' dialog. This file can be created using the "Save" button on various dialogs. The variables from the .val file are loaded into the dialog, along with their respective values from the file, which can be used in the 'Set' operation.

8. **Ping/Traceroute:** You can use this menu to send ICMP ping and traceroute requests to the currently selected device to check for IP connectivity.

9. **Properties**: You can use this menu option to show the screen for a corresponding node with all the current information.

- **Help**: Help Topics.

Tool Bar within SimpleMIBBrowser

Toolbar and node popup menus are provided as shortcuts to the main menu options.

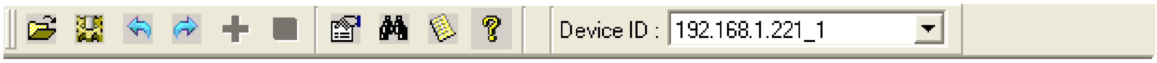


Figure 3 : Main Tool Bar

The functionality of each of these icons is exactly the same as the corresponding menu option. The icons simply provide a quicker way for you to invoke that function. These icons may get disabled / enabled in the same manner as the menu options, based on the selected MIB variable and state.

- This is a short cut to the "Open" menu option under the "File" menu.
- This is a shortcut to the *Hide Device View* menu option under *Edit* menu.
- This is a shortcut to the *Show Device View* menu option under *Edit* menu.
- This is a short cut to the "Preferences" menu option under the "Edit" menu.
- This is a short cut to the "Get(Walk)" menu option under the "Actions" menu.
- This is a short cut to the "Set" menu option under the "Actions" menu.
- This is a short cut to the "Add Row" menu option under the "Actions" menu.
- This is a short cut to the "Get Table" menu option under the "Actions" menu.
- This is a short cut to the "Properties" menu option under the "Action" menu to see the MIB information for the selected node.
- This is a short cut to the "Find" menu option under the "Edit" menu.
- This is a short cut to the "View Log" menu option under the "Edit" menu.
- This is a short cut to the "Help" menu option.
- **Device ID**: This will show the currently selected device identifier for the agent to query. It can be the IP address of the device or a more easily understandable name like "router1". The drop-down list associated with this field shows all the previously entered device profiles for easy access.

New devices can also be added by simply typing a new IP address in this box and it will inherit the previous devices SNMP profile.

In general, whenever the cursor moves over any one of these icons, a "tip" will also get displayed.

You can also "dock" the toolbar.

Popup Menu for MIB Nodes

The popup menus represent the various Actions that can be performed on the selected MIB Nodes.

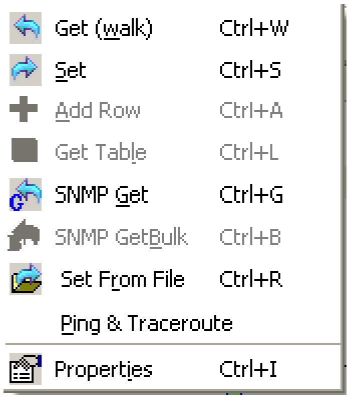


Figure 4 : Popup Menu for a node

Whenever the right button is clicked on a node, a context sensitive popup menu is displayed. If multiple nodes are selected, then options that make sense for multiple nodes appear enabled. The rest of the options are disabled in the menu. These menu options will have the same functionality as the main menu options. These are provided only to allow you to have quicker access to the equivalent function.

Chapter 3
Specifying Devices and their Settings

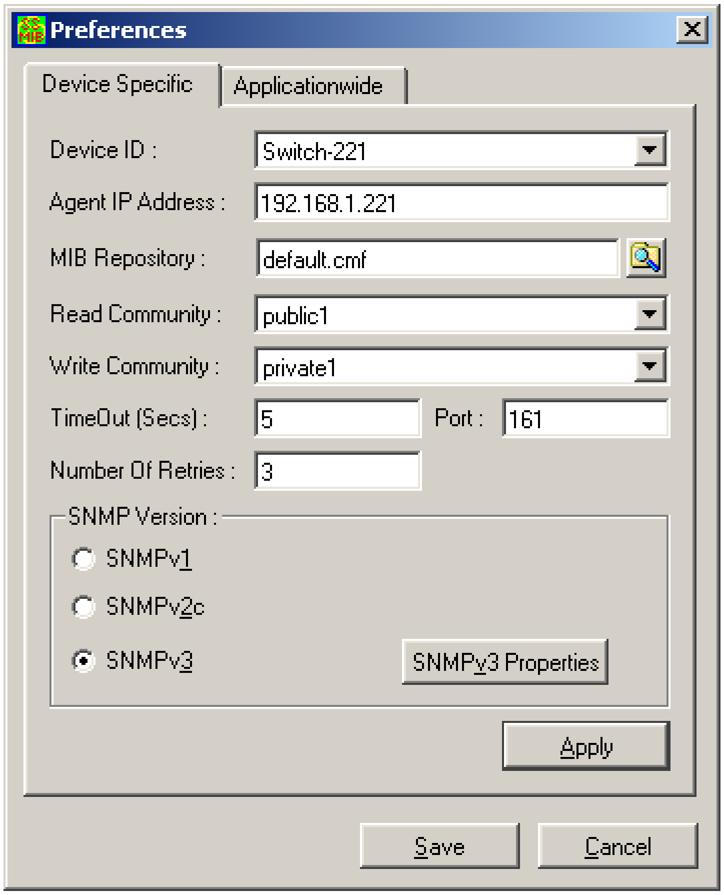
Before communicating with the SNMP enabled device, you must ensure that the communication settings in the SimpleMIBBrowser match those on the device. This chapter explains various preferences.

Individual Device Settings

When the Preferences is selected from the Edit menu (**Edit →Preferences)** or from the Toolbar, the following dialog is shown. The Preferences dialog contains two tabs: Device Specific and Application Specific.

Device Specific

SNMP related parameters can be set in this tab.



Device ID : Switch-221

Agent IP Address : 192.168.1.221

MIB Repository : default.cmf

Read Community : public1

Write Community : private1

TimeOut (Secs) : 5 Port : 161

Number Of Retries : 3

SNMP Version :

○ SNMPv1

○ SNMPv2c

● SNMPv3

Figure 5: Device Specific tab in the Preferences Screen

Device ID

This specifies a label to uniquely identify the SNMP enabled device that you are trying to communicate with. By default, it is the same as the IP address of the device, but it could be changed to another more understandable name like "router1".

Agent IP

This specifies the SNMP enabled device's IP address. All SNMP requests are sent to this IP address. This address will also be used in the case of community string based proxies and SNMPv3 context based proxies. Both IPv4 and IPv6 addresses are supported.

MIB Repository

This specifies the MIB repository associated with the device. By default it will be a repository called "default.cmf". However, you can specify different MIB respositories for different devices if you like.

Read Community

This is the community string you want the SimpleMIBBrowser to use when it issues SNMP Get/Getnext/GetBulk requests. The Read Community by default is set to "public." Set it to the community string that gives read privileges to access the part of the MIB that is of interest.

If you are communicating with SNMPv1-Proxies that use community strings for multiplexing requests, specify the appropriate community string that has access and identification information to allow the retrieval of the MIB variables.

The SimpleMIBBrowser remembers the set of communication parameters associated with a device. Selecting a device IP address, in the Agent IP field, causes its Read Community to also get set to the last specified value. You can always overwrite this.

Write Community

This is the community string you want the SimpleMIBBrowser to use when it issues SNMP Set requests. The Write Community by default is set to "private." Set it to the community string that gives write privileges to access the part of the MIB that is of interest.

If you are communicating with SNMPv1-Proxies that use community strings for multiplexed requests, specify the appropriate community string that has access and identification information to allow the setting of the MIB variables.

The SimpleMIBBrowser remembers the set of communication parameters associated with a device. Selecting a device IP address, in the Agent IP field, causes its Write Community to also get set to the last specified value. You can always overwrite this.

TimeOut (secs)

This is the amount of time you want the SimpleMIBBrowser to wait for a response when it issues SNMP requests. By default, this value is set to five (5) seconds. The amount of time the SimpleMIBBrowser should wait is a function of the type of SNMP Agent and its location in relation to the SimpleMIBBrowser. If the SNMP Agent is across five routers, it might take longer to receive a response. Also if the SNMP Agent has to translate the SNMP requests to an internal proprietary protocol and then issue these proprietary requests over a slow media, it might take longer to get a response back from the Agent.

Typically, if the SNMP Agent is on the local Ethernet, it is expected to respond back within one second. You can change the value to a more suitable number for your testbed. Note that if you set the TimeOut to a large number, it will slow down the execution of the tests, since the SimpleMIBBrowser will have to wait that much longer to determine that the Agent under test is not responding.

The SimpleMIBBrowser remembers the set of communication parameters associated with a device. Selecting a device IP address, in the Agent IP field, causes its Timeout to also get set to the last specified value. You can always overwrite this.

Num of Retries

This is the number of times you want the SimpleMIBBrowser to retry sending an SNMP request when it does not receive a response. By default, this value is set to three (3). You can set it to a value more suitable to your testbed. Typical things to consider are the load on the network, chances of the packet getting lost on the way, the number of receive buffers in the Agent, and how busy the Agent is doing other things. Note that if you set the number of retries to a large number, it will slow down the execution of the tests, since the SimpleMIBBrowser will have to wait that much longer to determine that the Agent under test is not responding.

The SimpleMIBBrowser remembers the set of communication parameters associated with a device. Selecting a device IP address, in the Agent IP field, causes its Retry Count to also get set to the last specified value. You can always overwrite this.

SNMP Port

This specifies the UDP port to which SNMP requests are sent to. By default it is set to port 161.

The SimpleMIBBrowser remembers the set of communication parameters associated with a device. Selecting a device IP address, in the Agent IP field, causes its SNMP Port to also get set to the last specified value. You can always overwrite this.

SNMP Version

This specifies the SNMP version to be used to communicate with the SNMP enabled device. The SimpleMIBBrowser supports SNMPv1, SNMPv2c, and SNMPv3. The SimpleMIBBrowser uses the value of this variable to determine what protocol to use for creating SNMP requests.

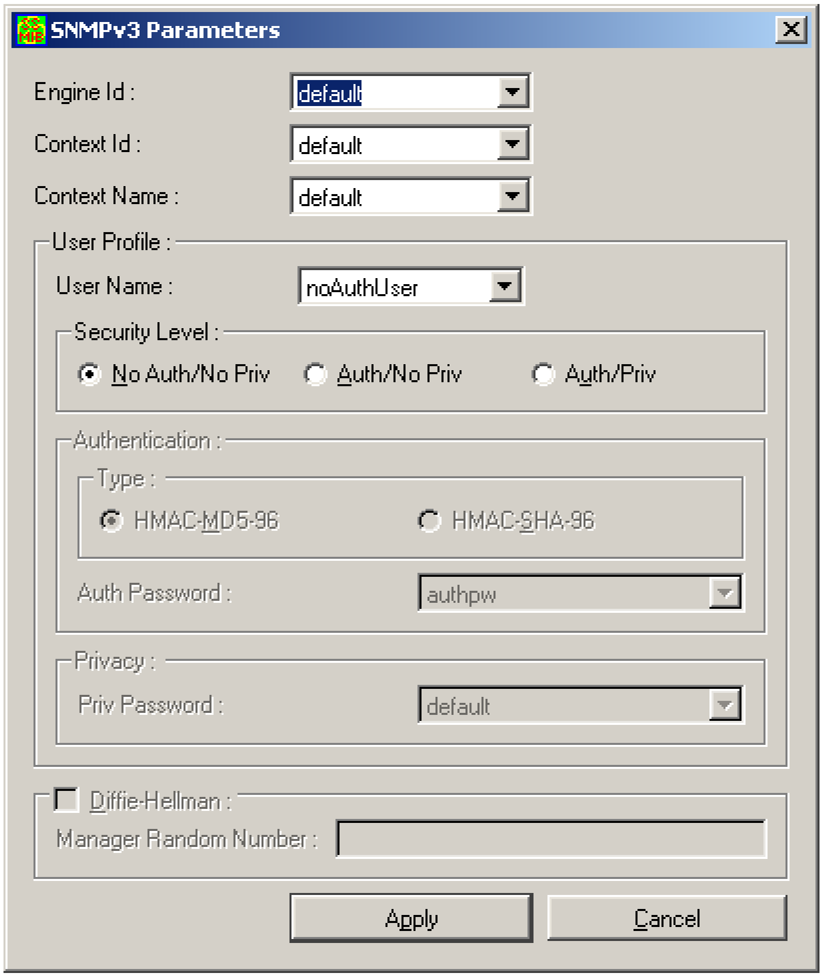
The SimpleMIBBrowser remembers the set of communication parameters associated with a device. Selecting a device IP address, in the Agent IP field, causes its SNMP version selection to also get set to the last specified value. You can always overwrite this.

Apply Button

When you click on this button, all the settings for the device will be saved. If the device modified is the currently displayed device and you have changed the MIB respository associated with it, clicking on this button will also automatically load the new MIB repository in the MIB tree view.

SNMPv3 Settings Screen

If the version chosen is SNMPv3, then the SNMPv3 Properties Button is enabled, which allows to set the SNMPv3 specific parameters to send the SNMPv3 request.



Engine Id : default

Context Id : default

Context Name : default

User Profile :

User Name : noAuthUser

Security Level : ● No Auth/No Priv ○ Auth/No Priv ○ Auth/Priv

Authentication :

Type : ● HMAC-MD5-96 ○ HMAC-SHA-96

Auth Password : authpw

Privacy :

Priv Password : default

□ Diffie-Hellman :

Manager Random Number :

Figure 6: SNMPv3 Parameters screen

Engine ID

This is used to identify the target SNMP agent. The Engine Id is an optional parameter. SNMPv3 defines a means to discover the value from the agent. Leaving this field blank causes the SimpleMIBBrowser to use the Discovery protocol to determine the Engine Id.

Context Engine ID

This is used to identify the agent that supports the MIB variables to be retrieved or set. Normally this value is the same as the SNMP Engine Id. If this field is left blank, then the SimpleMIBBrowser uses the same value as the SNMP Engine Id. This holds true whether the SNMP Engine Id was specified or discovered. This parameter will be different from the SNMP Engine Id when the target agent is acting as a proxy. In this case the value would identify the agent supporting the required MIB object, which will be different than the target agent.

Context Name

SNMPv3 allows for the grouping or collection of MIB variables; the Context Name identifies that grouping. The value depends on how these groups of variables are defined in the agent. This may or may not be a required parameter, depending on how the agent is configured.

User Name

User Name identifies the requestor's access to MIB variables. This setting may or may not be required – it is dependent on how the agent is configured. The name will specify which MIB variables are retrievable and which ones are settable. The default for this setting is "NoAuthUser."

SimpleMIBBrowser remembers the set of SNMPv3 settings associated with a specific user profile. Selecting a user name from the previously specified user list will cause all the associated v3 parameters like security level, authentication type, passwords, etc. to automatically get set to their last specified values.

Security Level

This controls the enabling or disabling of SNMPv3 authentication and privacy. There are only three combinations defined for SNMPv3: No Authentication and No Privacy, Authentication with No Privacy, and Authentication with Privacy. Authentication provides assurance that the request sent comes from an authorized source and that the message was not intercepted and altered in any way. Privacy provides the means to encrypt the message so that only authorized parties can understand the contents of the message.

Note

Privacy might be disabled in the demo version or expired demo version of SimpleMIBBrowser.

Authentication

This specifies the message digest algorithm used for authentication. The SimpleMIBBrowser supports two: HMAC-MD5-96 and HMAC-SHA-96. The algorithm must match the one used by the agent in test.

Authentication Password

This password is used as a seed to the authentication algorithm specified above. The value must match the password used by the agent in test or the authentication will fail.

Privacy Password

This password is used as a seed to the encryption algorithm specified for SNMPv3 User-based Security Model (USM). The value must match the password used by the agent in test or else the agent will fail to decrypt the messages sent by the SimpleMIBBrowser.

Diffie-Hellman

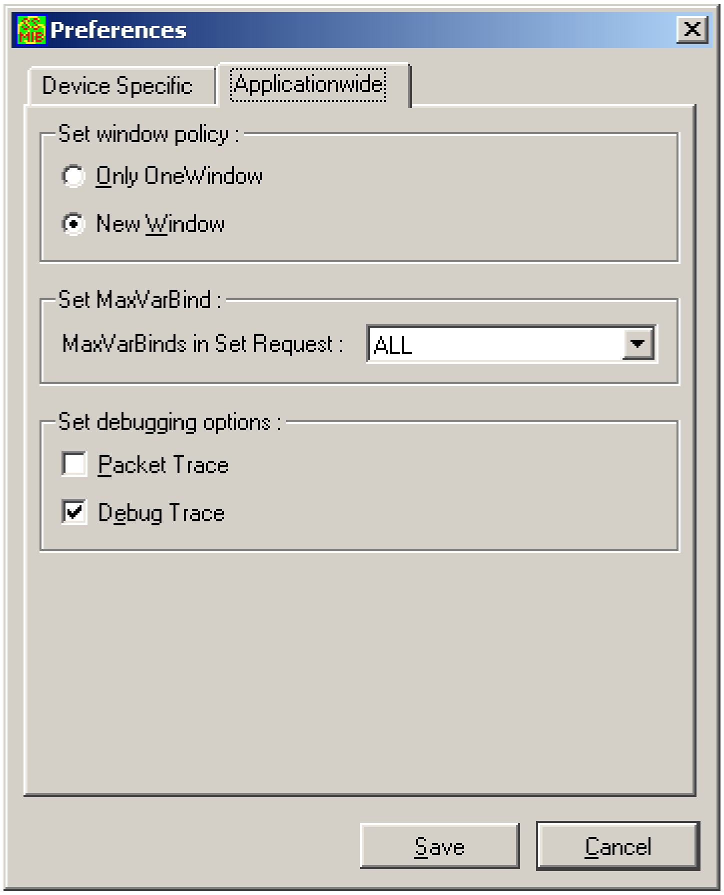
Diffie-Hellman defines an alternate means of deriving keys used for authentication and privacy. It uses a random number generated locally and the public key of its peer to derive a private, or secret, key. This secret key then becomes an input into the calculation of the authentication and privacy keys used by the SNMPv3 protocol. For the SimpleMIBBrowser to use Diffie-Hellman Key Agreement, the security level must be Auth/NoPriv or Auth/Priv. Merely select the *Yes* radio button for Key Agreement to enable this option. The authentication and privacy related fields will be grayed out and the Manager's Random Number field will be enabled. Enter the random number used to derive the Manager's Public Key. There should be a User Name that corresponds with the random number. Enter that name in the User Name field.

Application Settings

You can also specify settings that are pertinent to the MIB Browser application in the Applicationwide tab. This information is not specific to each device, but is common to all devices.

Applicationwide Tab

Advanced parameters like window policy and default number of varbinds in set request, etc. can be configured in this screen.



Set window policy :

○ Only OneWindow

● New Window

Set MaxVarBind :

MaxVarBinds in Set Request : ALL

Set debugging options :

□ Packet Trace

☑ Debug Trace

Figure 7: Advanced Tab in the Preferences Screen

Window Policy

This setting specifies the behavior of the SimpleMIBBrowser when displaying the results of a query. When the *One-Window* policy is selected, subsequent queries overwrite previously returned results. When the *New-Window* policy is selected, new queries get displayed in a new results window and the older result windows do not get overwritten.

MaxVarbinds in Set Request

This setting specifies the number of varbinds the SimpleMIBBrowser uses when grouping a large list of variables into smaller packets for doing SNMP Set requests.

Packet Trace

When you select this option, the information regarding the Request sent to the agent and Response sent by the agents to the SimpleMIBBrowser gets written in the log file called mibbrowser.log. The actual raw hex packet dumps are saved in this option.

Debug Trace

When you select this option, the information regarding the Request sent to the agent and Response sent by the agents to the SimpleMIBBrowser gets written in the log file called mibbrowser.log. Instead of the actual raw packet dump, a more intelligent dump of the various fields in the SNMP packet gets saved.

The outer buttons in the Preferences Screen apply to all the tabs and do the following:

Save:

When you click the Save button, then all the configurable options which are present in the three tabs, *viz,* Snmp, Advanced, and Debugging, get saved in the.ini file.

Cancel:

When you click the Cancel button, the settings dialog box gets cancelled.

Chapter 4
Viewing MIB Variable Information

SimpleMIBBrowser displays the information in the MIB files in a graphical tree format as shown below by displaying the node registration tree. The information related to each of the nodes as specified in the MIB can be viewed by selecting the node and chosing Properties in the popup menu.

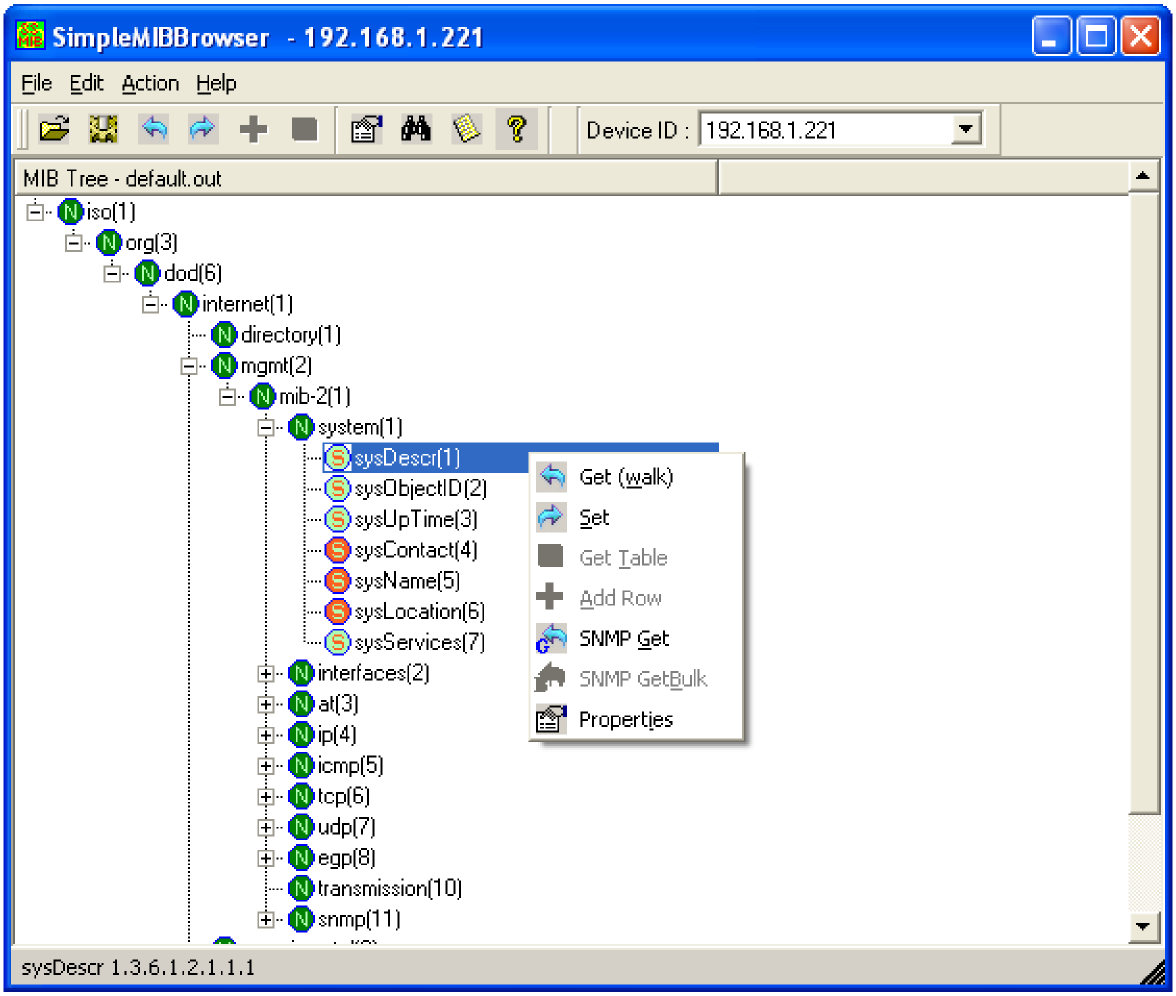
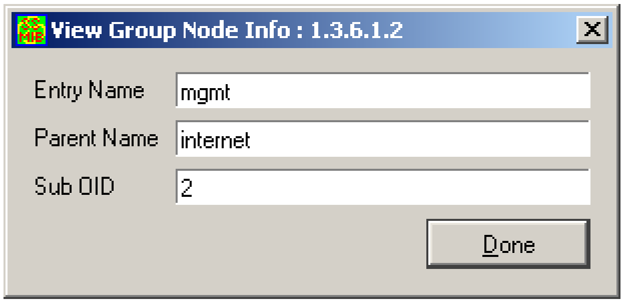


Figure 8: Viewing Properties of MIB nodes

When the Properties option is selected from the Node Popup menu, the following properties dialogs are displayed. Depending on the type of node selected, the associated information screen is displayed.

Group Node Info


View Group Node Info : 1.3.6.1.2

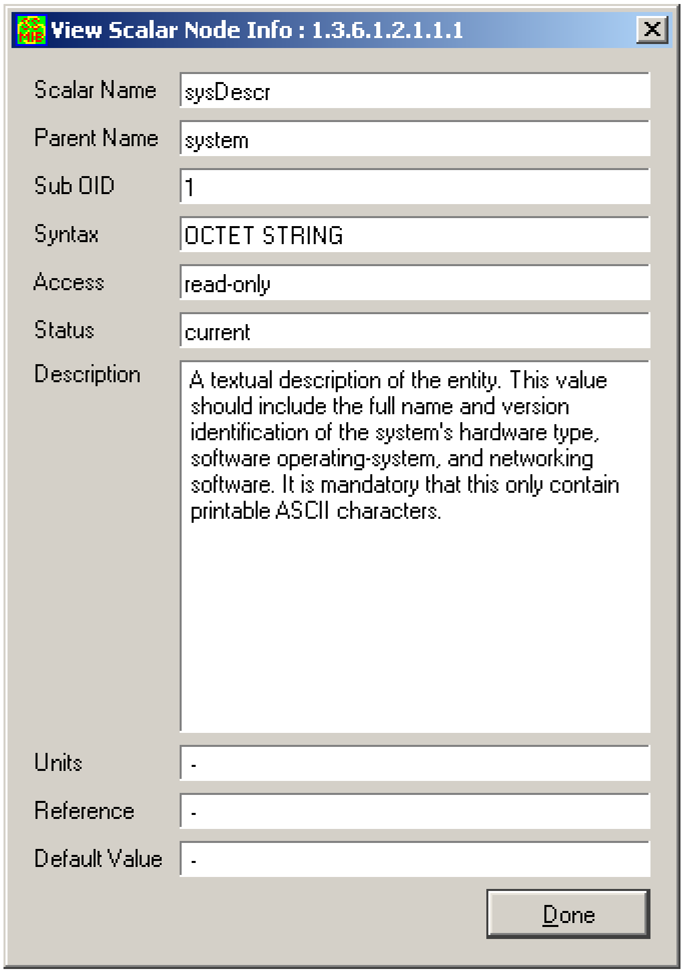
Entry Name: mgmt

Parent Name: internet

Sub OID: 2

Figure 9: Example Group Node Property Screen

Scalar Node Info

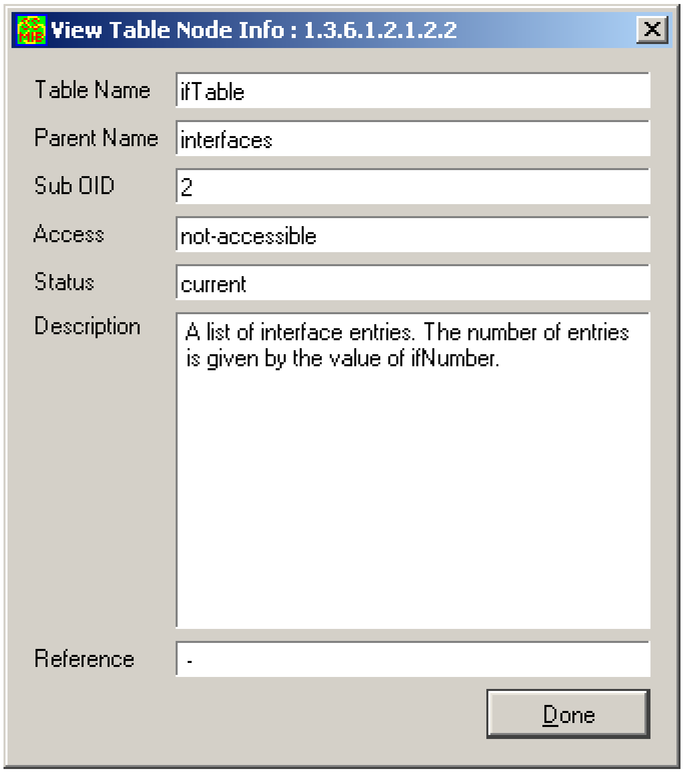


View Scalar Node Info : 1.3.6.1.2.1.1.1

Scalar Name	sysDescr
Parent Name	system
Sub OID	1
Syntax	OCTET STRING
Access	read-only
Status	current
Description	A textual description of the entity. This value should include the full name and version identification of the system's hardware type, software operating-system, and networking software. It is mandatory that this only contain printable ASCII characters.
Units	-
Reference	-
Default Value	-

Figure 10: Example Scalar Node Property Screen

Table Node Info

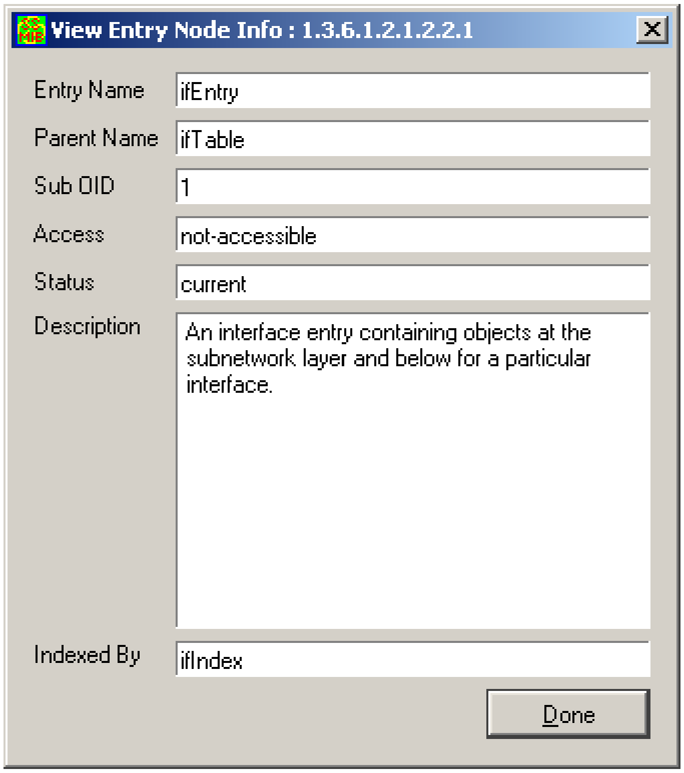


View Table Node Info : 1.3.6.1.2.1.2.2

Table Name	ifTable
Parent Name	interfaces
Sub OID	2
Access	not-accessible
Status	current
Description	A list of interface entries. The number of entries is given by the value of ifNumber.
Reference	-

Figure 11: Example Table Node Property Screen

Entry Node Info


View Entry Node Info : 1.3.6.1.2.1.2.2.1

Entry Name: ifEntry

Parent Name: ifTable

Sub OID: 1

Access: not-accessible

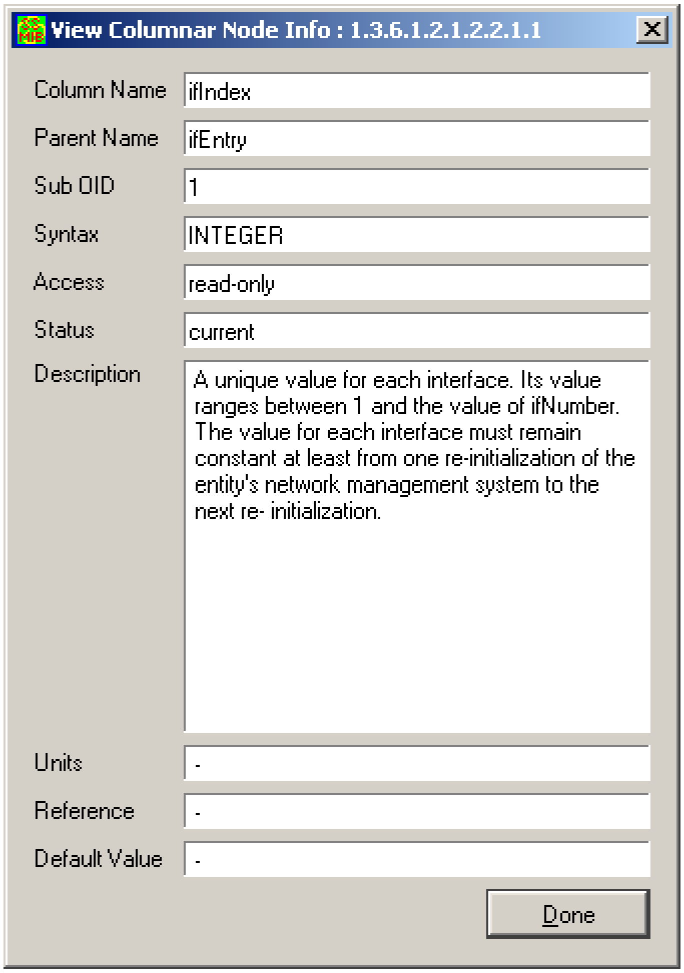
Status: current

Description: An interface entry containing objects at the subnetwork layer and below for a particular interface.

Indexed By: ifIndex

Figure 12: Example Entry Node Property Screen

Columnar Node Info



Column Name: ifIndex

Parent Name: ifEntry

Sub OID: 1

Syntax: INTEGER

Access: read-only

Status: current

Description: A unique value for each interface. Its value ranges between 1 and the value of ifNumber. The value for each interface must remain constant at least from one re-initialization of the entity's network management system to the next re- initialization.

Units: -

Reference: -

Default Value: -

Figure 13: Example Columnar Node Property Screen

Chapter 5
Retrieving Data

Once the MIB information is loaded and the communication settings are specified, you are ready to talk with the SNMP enabled device and examine the management information within it. You can use the SimpleMIBBrowser to retrieve as well as change the management information.

This chapter describes the various ways in which the data can be retrieved from the SNMP enabled device.

Getting MIB data using the SNMP GetNext Operator (Walk)

By default, the SimpleMIBBrowser uses the SNMP GetNext Operator to retrieve data from the device An SNMP walk is done on the selected node, and the data is displayed in the results window as it is retrieved. The following paragraphs describe the "SNMP Walk" process.

All the manageable objects supported by an SNMP Agent can be viewed as a long, flat, sequential list of attribute instances, ordered in a lexicographic manner. The SNMP GetNext operator can be used to traverse this list.

Let us assume that the list is made up of [a.0, b.0, b.x.1, b.y.1, c.0, d.0]. where a.0 is lexicographically less than b.0, and b.0 is lexicographically less than c.0, and c.0 is lexicographically less than d.0. a.0, b.0, c.0, and d.0 are attribute object identifiers for the manageable objects that they represent.

GetNextRequest of ["1.0" or a] will give back

GetResponse[a.0, datatype of a.0, value of a.0]

You can then send a *GetNextRequest[a.0]* to get back

GetResponse[b.0, datatype of b.0, value of b.0]

You can then send a *GetNextRequest[b.0]* to get back

GetResponse[b.x,1, datatype of b.x.1, value of b.x.1]

You can then send a *GetNextRequest[b.x.1]* to get back

GetResponse[b.y.1, datatype of b.y.1, value of b.y.1]

You can then send a *GetNextRequest[b.y.1]* to get back

GetResponse[c.0, datatype of c.0, value of c.0]

You can then send a *GetNextRequest[c.0]* to get back

GetResponse[d.0, datatype of d.0, value of d.0]

Now sending a *GetNextRequest[d.0]* will result in

GetResponse[NoSuchName Error],

which indicates the End *Of All MIBs*.

This process of making a series of sequential *GetNext* requests to retrieve the list of attribute instances supported by an Agent is often referred to as "walking" the MIB.

To get sub-trees, this walking can be initiated at the root of the subtree and terminated when the returned object identifier no longer matches the starting object identifier or you get a NoSuchName Error. For instance, suppose if we wanted to get the "b" sub-tree, we could have done the following:

Send a *GetNextRequest[b]* to get back

GetResponse[b.0, datatype of b.0, value of b.0]

Compare [b.0] with the starting object id [b]. If it still matches, continue

Send a *GetNextRequest[b.0]* to get back

GetResponse[b.x.1, datatype of b.x.1, value of b.x.1]

Compare [b.x.1] with the starting object id [b]. If it still matches, continue

Send a *GetNextRequest[b.x.1]* to get back

GetResponse[b.y.1, datatype of b.y.1, value of b.y.1]

Compare [b.y.1] with the starting object id [b]. If it still matches, continue

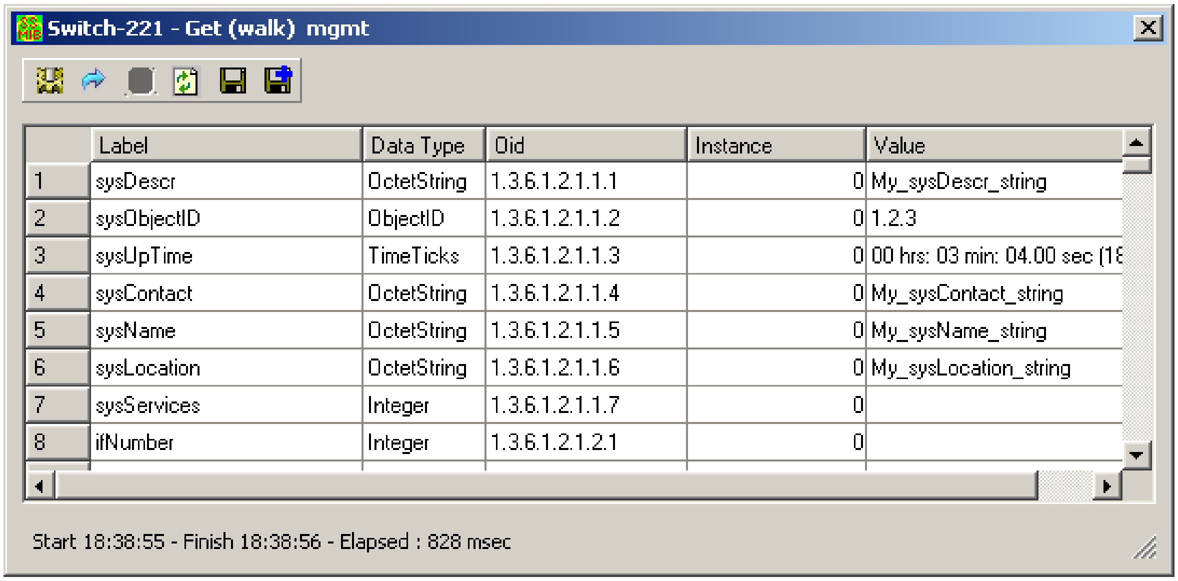
Send a *GetNextRequest[b.y.1]* to get back

GetResponse[c.0, datatype of c.0, value of c.0]

Compare [c.0] with the starting object id [b]. Now it no longer matches the starting object id. So we stop the sequential list traversal.

Our attribute instance list will now include attributes [b.0, b.x.1, b.y.1]. c.0 is *not* included, although it is retrieved.

The retrieved list is shown in a results screen as seen below. The retrived variables (scalar as well as columnar nodes) are displayed in a lexicographical order.



Switch-221 - Get (walk) mgmt

	Label	Data Type	Oid	Instance	Value
1	sysDescr	OctetString	1.3.6.1.2.1.1.1	0	My_sysDescr_string
2	sysObjectID	ObjectID	1.3.6.1.2.1.1.2	0	1.2.3
3	sysUpTime	TimeTicks	1.3.6.1.2.1.1.3	0	00 hrs: 03 min: 04.00 sec (18
4	sysContact	OctetString	1.3.6.1.2.1.1.4	0	My_sysContact_string
5	sysName	OctetString	1.3.6.1.2.1.1.5	0	My_sysName_string
6	sysLocation	OctetString	1.3.6.1.2.1.1.6	0	My_sysLocation_string
7	sysServices	Integer	1.3.6.1.2.1.1.7	0	
8	ifNumber	Integer	1.3.6.1.2.1.2.1	0	

Start 18:38:55 - Finish 18:38:56 - Elapsed : 828 msec

Figure 14: Results of data retrieval on a MIB Node

You can start multiple queries. Based on the window policy selected (in 'Set Window Policy', in the Perferences dialog box under 'Advanced Tab'), one or more result windows will be opened.

If you had selected multiple MIB nodes in the MIB tree prior to starting the Get(Walk) query, then based on the Window policy, you will either be notified that only the first selected node will be used and its results displayed, or multiple result windows will get opened which display the results of carrying out walks on each of the selected nodes.

You can select one or more rows from this table. For multiple row selection, control or shift key can be used along with the mouse click.

The tool bar on the top of the results dialog is described below:

- **Preferences**: You can use this toolbar icon to change the communication settings. This option displays the Preferences dialog.

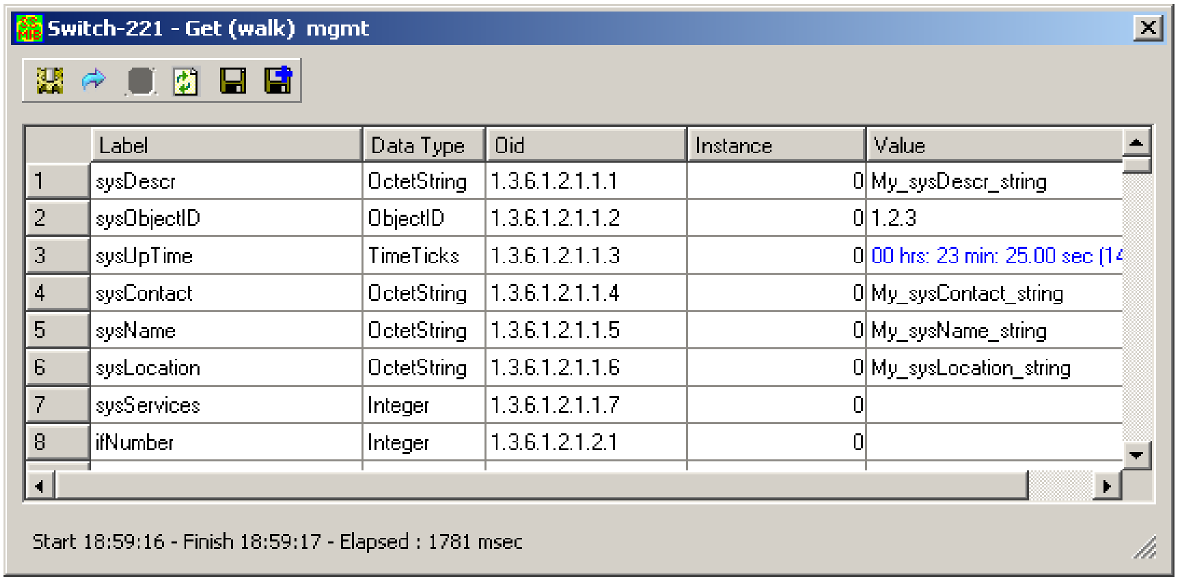
- **Set**: When you click on this icon, all the writeable variables which are present in the results window are shown in the set dialog box with PDU split. The PDU split takes place according to the configurable option named ‘MaxVarbinds in Set Request’, which is available in the Perferences dialog box under ‘Advanced Tab’.

- **Stop Query**: This icon is enabled only until the results of the query are still being retrieved from the agent. You can click on this icon to abort the query while it is in progress, when, say, the agent is not responding or some unexpected error has occurred.

- **Refresh**: You can click on this icon to redo the original query. If the values are different than when they were retrieved previously, they are displayed in the blue color, as shown in the figure below.

- **Write To File:** When you click on this icon, a File Save Dialog box with save as type ‘.val’ is shown. If the “.val” file type is used, then only writeable variables which are present in the results window are written in the file for future “setting from file” operation. If the file already exists, you will be asked whether to over-write it.

- **Append To File**: When you click on this icon, an Open File Dialog box with file type as ‘.val’ is shown. If the ‘.val’ file type is used, the writeable varaiables which are present in the results window are appended to this file.



Switch-221 - Get (walk) mgmt

	Label	Data Type	Oid	Instance	Value
1	sysDescr	OctetString	1.3.6.1.2.1.1.1	0	My_sysDescr_string
2	sysObjectID	ObjectID	1.3.6.1.2.1.1.2	0	1.2.3
3	sysUpTime	TimeTicks	1.3.6.1.2.1.1.3	0	00 hrs: 23 min: 25.00 sec (1
4	sysContact	OctetString	1.3.6.1.2.1.1.4	0	My_sysContact_string
5	sysName	OctetString	1.3.6.1.2.1.1.5	0	My_sysName_string
6	sysLocation	OctetString	1.3.6.1.2.1.1.6	0	My_sysLocation_string
7	sysServices	Integer	1.3.6.1.2.1.1.7	0	
8	ifNumber	Integer	1.3.6.1.2.1.2.1	0	

Start 18:59:16 - Finish 18:59:17 - Elapsed : 1781 msec

Figure 15: Refresh shows the changed values in a different color

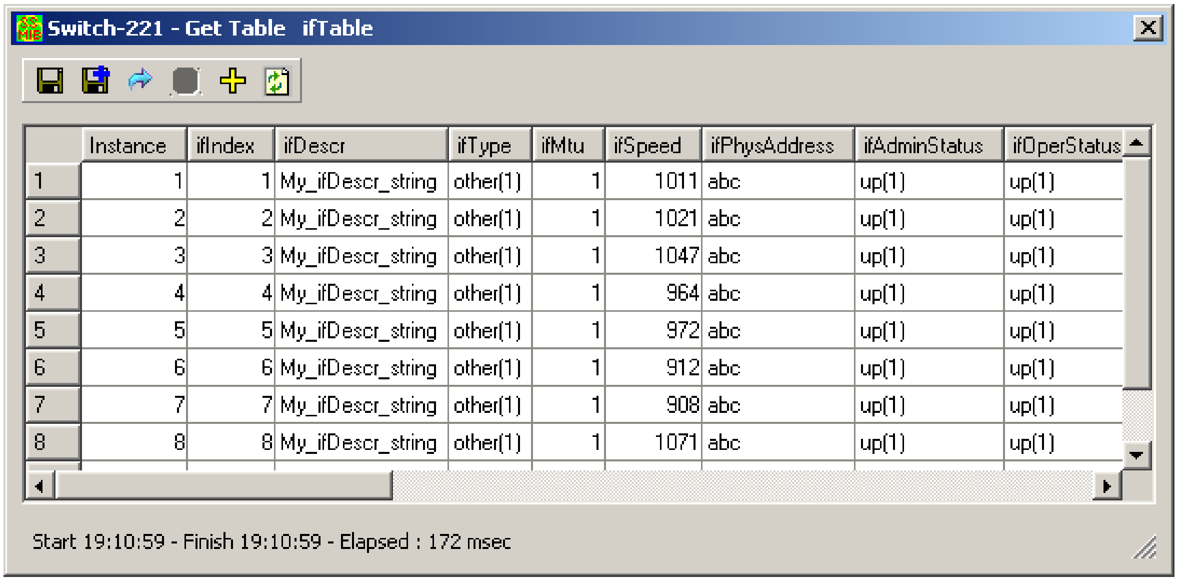
Getting Tabular Data

SNMP data is generally of two types: scalar and tabular. Data that is only present once in a device's data model is typically represented as a scalar (eg: sysLocation-physical location of the device), while data that is present multiple times in a device's data model is represented within a table (ifOperStatus – status of one of the device's many interfaces).

Getting the data by selecting any node as described earlier, displays the data in a lexicograhical manner. For tabular data, an additional menu option is available that allows you to see the data in a form that is similar to its definition in the MIB. This option is available for table and entry nodes and can be chosen from edit menu, node popup menu or from the toolbar.

When this option is chosen, all the columnar nodes are collected in a single PDU and a GetNext query (walk) is performed in a loop. The results of the query are displayed as they are retrieved by adding rows to the display. If there are individual cells that are empty, then they are kept blank. The results for the interfaces table is shown below.

An additional column called Instance is shown at the start of the table, while the rest of the columns match the MIB definition. If index columns are not-accessible, their value is determined based on the value of the instance component, and displayed in Light Grey color.



	Instance	ifIndex	ifDescr	ifType	ifMtu	ifSpeed	ifPhysAddress	ifAdminStatus	ifOperStatus
1	1	1	My_ifDescr_string	other(1)	1	1011	abc	up(1)	up(1)
2	2	2	My_ifDescr_string	other(1)	1	1021	abc	up(1)	up(1)
3	3	3	My_ifDescr_string	other(1)	1	1047	abc	up(1)	up(1)
4	4	4	My_ifDescr_string	other(1)	1	964	abc	up(1)	up(1)
5	5	5	My_ifDescr_string	other(1)	1	972	abc	up(1)	up(1)
6	6	6	My_ifDescr_string	other(1)	1	912	abc	up(1)	up(1)
7	7	7	My_ifDescr_string	other(1)	1	908	abc	up(1)	up(1)
8	8	8	My_ifDescr_string	other(1)	1	1071	abc	up(1)	up(1)

Start 19:10:59 - Finish 19:10:59 - Elapsed : 172 msec

Figure 16: Tabular display corresponding to the associated MIB table definition

The tool bar icons are similar to the general results windows described earlier, except that the tabular display includes one more icon described below.

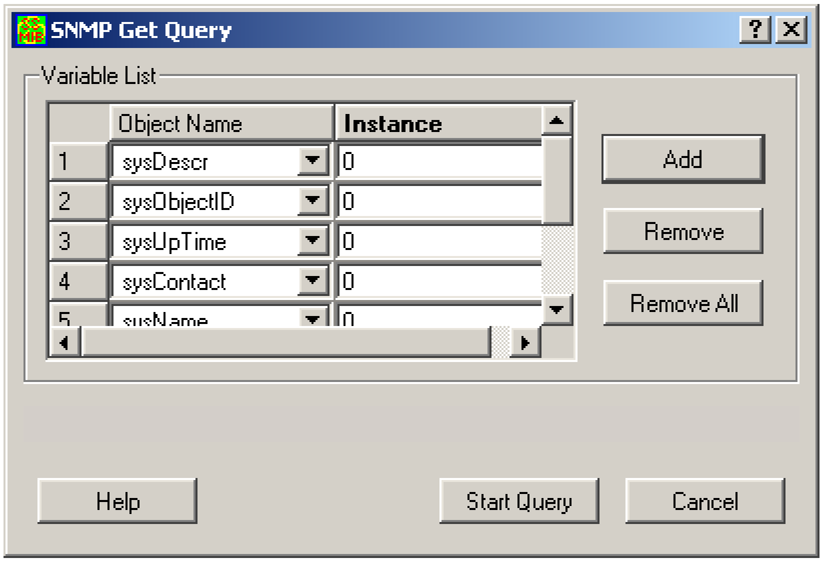
✚ **Add Row:** When you click on this icon, a dialog similar to the set dialog is shown, allowing you to add rows to this table. This functionality is further described in the subsequent chapter on Setting MIB data.

Getting MIB data using the SNMP Get Operator

The SimpleMIBBrowser uses the SNMP GetNext operator by default when retrieving MIB data. However, if you want to explicitly send a SNMP Get Request, you can do that also, by selecting SNMP Get Option in the Action Menu.

If you had selected multiple MIB nodes in the MIB tree prior to starting the Get query, then they or the leafnodes underneath them will show up pre-entered in the variable list. If the list is too large, a prompt will be displayed to limit the list to the first 20.

The window shown below allows you to compose the SNMP Get Request with one or more variables. Please note that when sending a SNMP Get Request, the instance component of the object identifier also has to be specified.

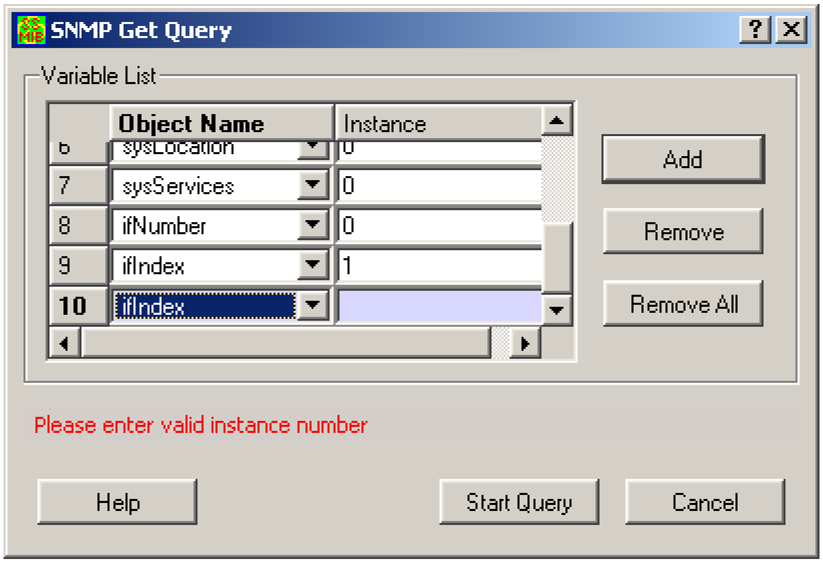


	Object Name	Instance
1	sysDescr	0
2	sysObjectID	0
3	sysUpTime	0
4	sysContact	0
5	sysName	0

Figure 17: SNMP Get Operation

The functionality of the buttons in the window is described below:

- **Add**: When you click on the 'Add' button, a new row gets added. The cell has a drop down list box which contains all the variables that are present in the MIB. If a scalar variable is selected, the Instance cell will be set to 0 and this cell becomes non-editable, other wise it will be left blank and the Instance cell becomes editable. If no instance is specified for a tabular variable, the new variable does not get added in the dialog, but an error message in red is displayed as shown below.

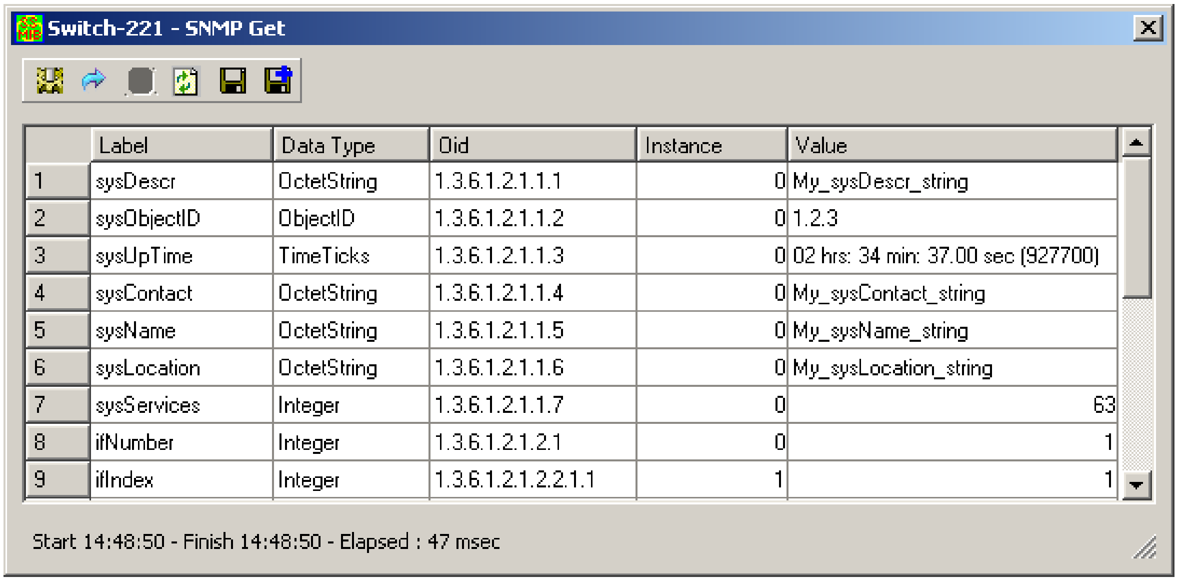


	Object Name	Instance
6	sysLocation	0
7	sysServices	0
8	ifNumber	0
9	ifIndex	1
10	ifIndex	

Please enter valid instance number

Figure 18: SNMP Get Operation with error display

- **Remove**: When this is clicked, the selected variable gets deleted from the list.
- **Remove All**: When this is clicked, all variables, which were previously added in the dialog, get deleted.
- **Help**: When this is clicked, a popup help window is displayed.
- **Cancel**: When this button is clicked, the SNMP Get Query dialog is cancelled.
- **Start Query**: When this button is clicked, then the actual SNMP Get query is sent to the agent. After the Start Query button is clicked, the results will be shown in the SNMP Get dialog as shown in the following figure.



	Label	Data Type	Oid	Instance	Value
1	sysDescr	OctetString	1.3.6.1.2.1.1.1	0	My_sysDescr_string
2	sysObjectID	ObjectID	1.3.6.1.2.1.1.2	0	1.2.3
3	sysUpTime	TimeTicks	1.3.6.1.2.1.1.3	0	02 hrs: 34 min: 37.00 sec (927700)
4	sysContact	OctetString	1.3.6.1.2.1.1.4	0	My_sysContact_string
5	sysName	OctetString	1.3.6.1.2.1.1.5	0	My_sysName_string
6	sysLocation	OctetString	1.3.6.1.2.1.1.6	0	My_sysLocation_string
7	sysServices	Integer	1.3.6.1.2.1.1.7	0	63
8	ifNumber	Integer	1.3.6.1.2.1.2.1	0	1
9	ifIndex	Integer	1.3.6.1.2.1.2.2.1.1	1	1

Start 14:48:50 - Finish 14:48:50 - Elapsed : 47 msec

Figure 19: SNMP Get Operation Results display

You can start multiple queries. Based on the window policy selected (in 'Set Window Policy', in the Perferences dialog box under 'Advanced Tab'), one or more result windows will be opened.

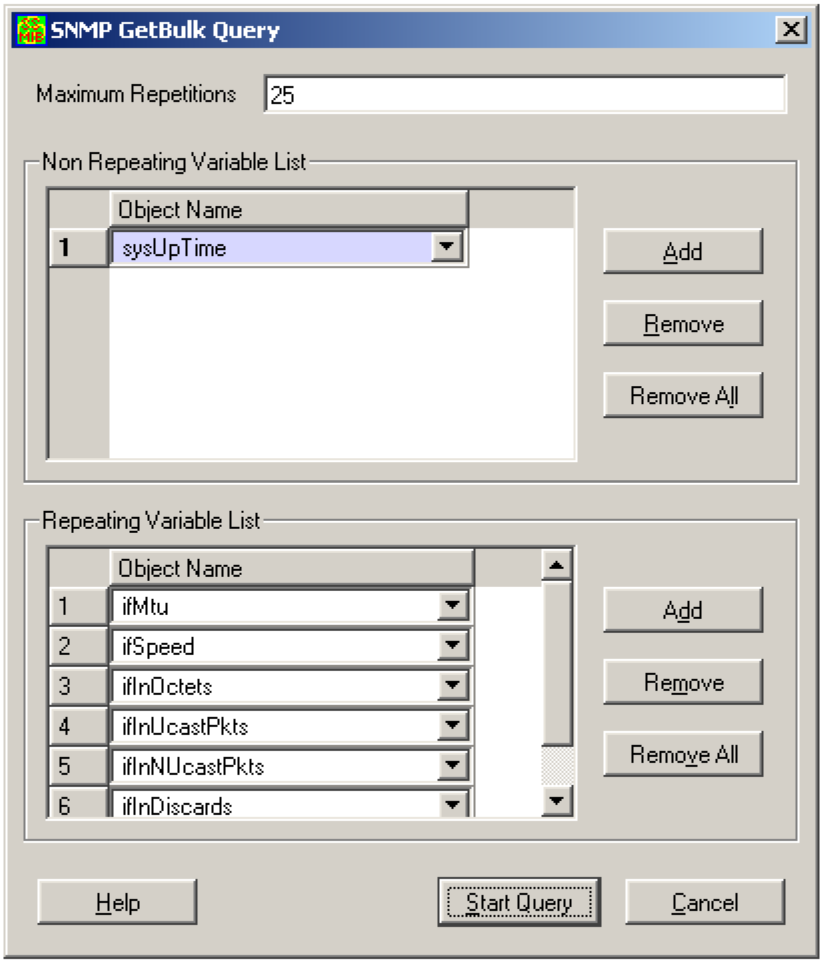
The tool bar icons are similar to those described earlier.

Getting MIB data using the SNMP GetBulk Operator

As mentioned earlier, the SimpleMIBBrowser uses the SNMP GetNext operator by default when retrieving MIB data. However, if you want to explicitly send a SNMP GetBulk Request, you can do that also, by selecting SNMP GetBulk Option in the Action Menu. This option is not available in SNMPv1, and only present in SNMPv2c and SNMPv3.

The window shown below allows you to compose the SNMP GetBulk Request with one or more variables in the Non-Repeating Variable List and in the Repeating Variable List.

If you had selected multiple MIB nodes in the MIB tree prior to starting the GetBulk query, then they or the leafnodes underneath them will show up pre-entered in the repeating variable list. If the list is too large, a prompt will be displayed to limit the list to the first 20.



Maximum Repetitions: 25

Non Repeating Variable List

	Object Name
1	sysUpTime

Repeating Variable List

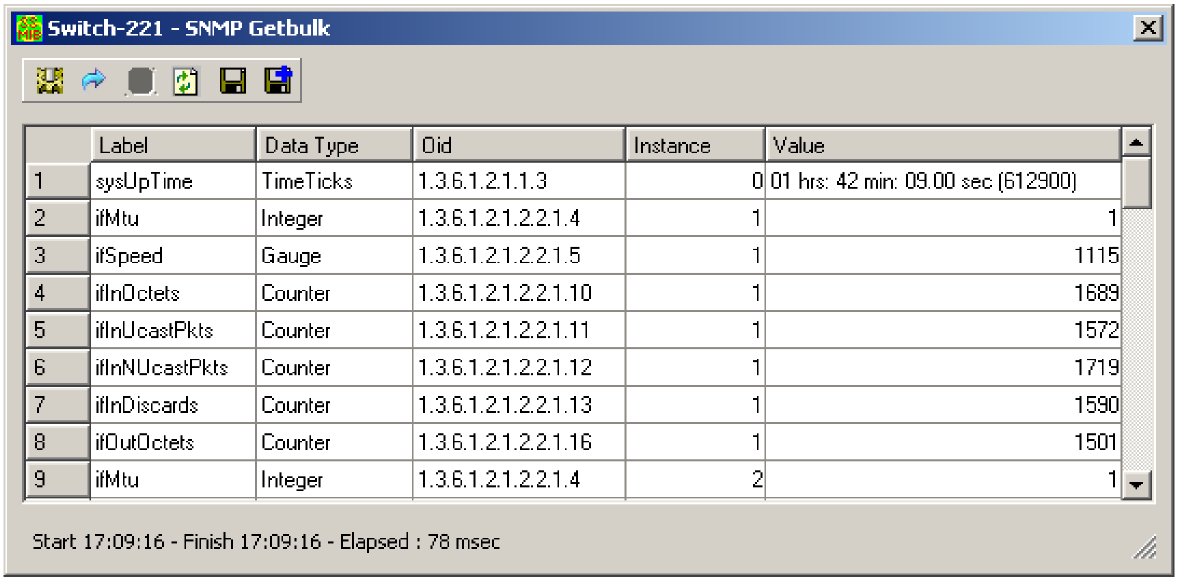
	Object Name
1	ifMtu
2	ifSpeed
3	ifInOctets
4	ifInUcastPkts
5	ifInNUcastPkts
6	ifInDiscards

Figure 20: SNMP GetNext Operation Screen

The following are the buttons, which are present in the 'SNMP GetBulk Query' dialog.

- **Add**: When you click on the 'Add' button, a new row gets added. The cell has a drop down list box which contains all the variables that are present in the MIB.
- **Remove**: When this is clicked, the selected variable gets deleted from the list.
- **Remove All**: When this is clicked, all variables, which were previously added in the dialog, get deleted.
- **Help**: When this is clicked, a popup help window is displayed.
- **Cancel**: When this button is clicked, the SNMP GetBulk Query dialog is cancelled.

- **Start Query**: When this button is clicked, then the actual SNMP GetBulk query is sent to the agent. After the Start Query button is clicked, the results will be shown in the SNMP Get dialog as shown in the following figure.



	Label	Data Type	Oid	Instance	Value
1	sysUpTime	TimeTicks	1.3.6.1.2.1.1.3	0	01 hrs: 42 min: 09.00 sec (612900)
2	ifMtu	Integer	1.3.6.1.2.1.2.2.1.4	1	1
3	ifSpeed	Gauge	1.3.6.1.2.1.2.2.1.5	1	1115
4	ifInOctets	Counter	1.3.6.1.2.1.2.2.1.10	1	1689
5	ifInUcastPkts	Counter	1.3.6.1.2.1.2.2.1.11	1	1572
6	ifInNUcastPkts	Counter	1.3.6.1.2.1.2.2.1.12	1	1719
7	ifInDiscards	Counter	1.3.6.1.2.1.2.2.1.13	1	1590
8	ifOutOctets	Counter	1.3.6.1.2.1.2.2.1.16	1	1501
9	ifMtu	Integer	1.3.6.1.2.1.2.2.1.4	2	1

Start 17:09:16 - Finish 17:09:16 - Elapsed : 78 msec

Figure 21: SNMP GetBulk Operation Results Display

You can start multiple queries. Based on the window policy selected (in ‘Set Window Policy’, in the Perferences dialog box under ‘Advanced Tab’), one or more result windows will be opened.

The tool bar icons are similar to those described earlier.

Chapter 6
Setting/Configuring Data

You can use the SimpleMIBBrowser to also set or change the value of variables. When variables are defined in the MIB, they can be defined as read-only, which implies that their values can only be read, or read-write, which implies that their value can be read as well as changed. Other types such as read-create also allow changing of the values of variables. An older type of write-only (which is now not in use) also allows management applications to send SNMP Set requests to it.

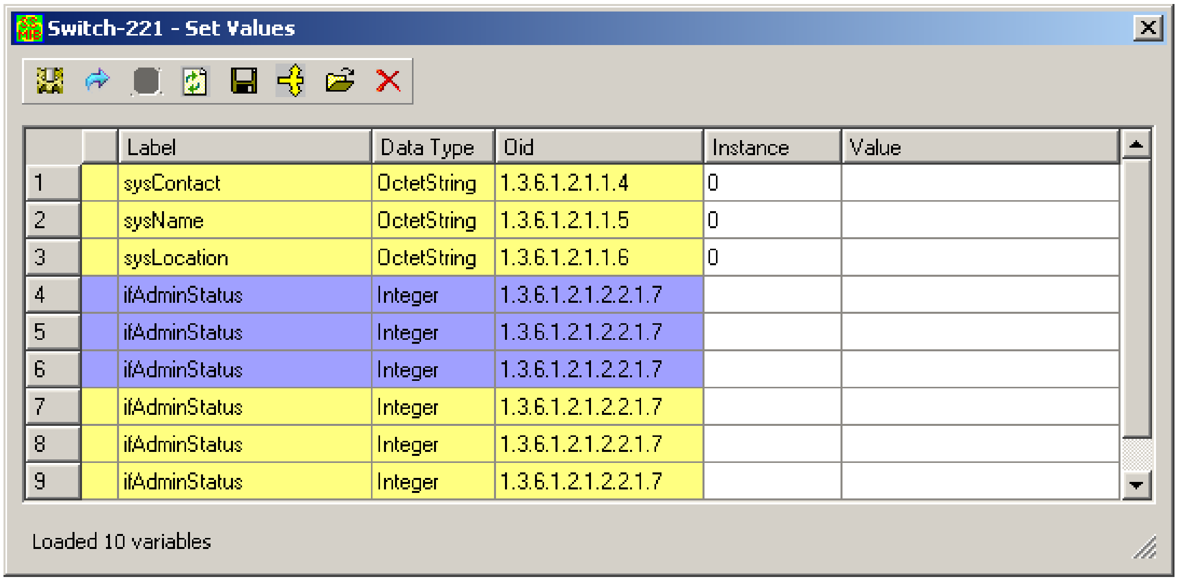
SNMP SET Operation

The Set operation can be initiated in three ways:

- Select a node from the tree view and then select the 'Set' operation from the popup menu. In the case of leaf-nodes, this operation is only available on nodes that are of type Read-Write, Read-Create or Write-Only (changeable leaf nodes). In the case of intermediate nodes, a list of changeable leaf nodes that are underneath the selected node is created.
- Click on the "Set" icon in the results window that is displayed after doing a SNMP GetNext (walk), SNMP Get or SNMP GetBulk operation.
- Selecting 'Set From File' menu under 'Action' menu.

In all the cases above, the PDU containing a list of changeable variables is created and displayed in two different colors based on the value of the configurable option 'MaxVarbinds in Set Request' which is present in Preference dialog box under 'Advanced Tab' or the special '%PduSplitMarker' lines if loading from a file.

If the instance information is not known, (as when initiating SET from the tree view) it will be filled up as ".0" for scalar variables and left blank for tabular variables. Please note that for the SET operation, the instance information needs to be filled prior to sending the SET request.



Switch-221 - Set Values

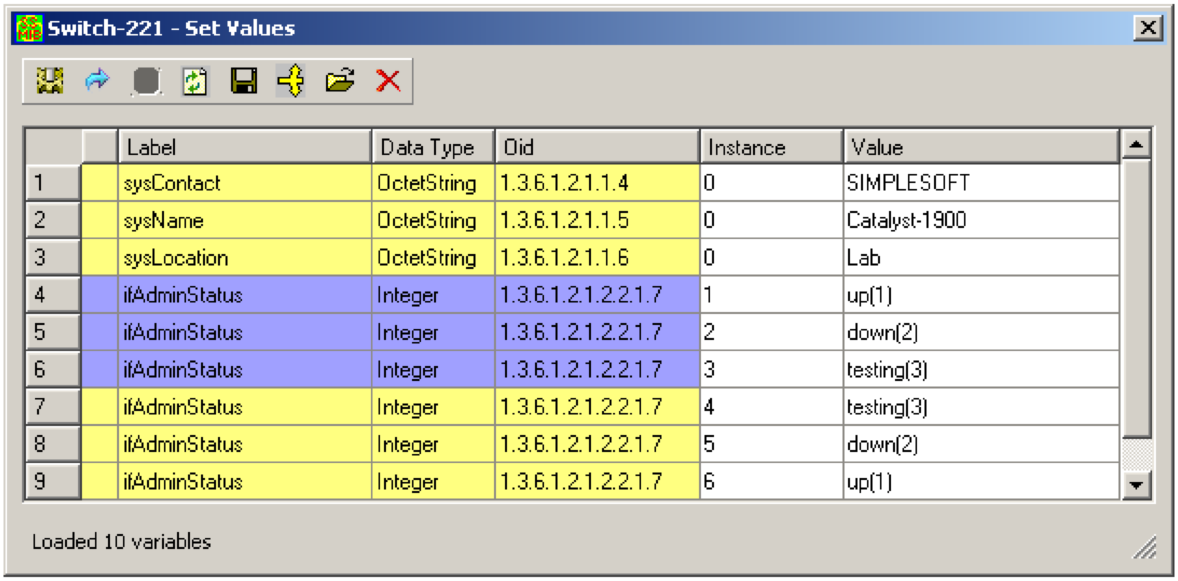
		Label	Data Type	Oid	Instance	Value
1		sysContact	OctetString	1.3.6.1.2.1.1.4	0	
2		sysName	OctetString	1.3.6.1.2.1.1.5	0	
3		sysLocation	OctetString	1.3.6.1.2.1.1.6	0	
4		ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7		
5		ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7		
6		ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7		
7		ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7		
8		ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7		
9		ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7		

Loaded 10 variables

Figure 22: Setting many variables initial screen

The variables displayed in the above dialog are grouped together into separate PDUs and shown as bands, based on the value of 'MaxVarbinds in Set Request' (in the above example, it was set to 3).

In the case where the SET is initated from the results window of a previous GET operation, the instance information will be already present. Only variables that are of type Read-Write, Read-Create, and Write-Only are selected and displayed in the window shown below.



Switch-221 - Set Values

		Label	Data Type	Oid	Instance	Value
1		sysContact	OctetString	1.3.6.1.2.1.1.4	0	SIMPLESOFT
2		sysName	OctetString	1.3.6.1.2.1.1.5	0	Catalyst-1900
3		sysLocation	OctetString	1.3.6.1.2.1.1.6	0	Lab
4		ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	1	up(1)
5		ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	2	down(2)
6		ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	3	testing(3)
7		ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	4	testing(3)
8		ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	5	down(2)
9		ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	6	up(1)

Loaded 10 variables

Figure 23: Setting many variables with values

The tool bar on the top of the set dialog is described below:

- **Preferences**: You can use this toolbar icon to change the communication settings. This option displays the Preferences dialog.

- **Set**: You can use this toolbar icon to initiate the set action. All the variables, which are present in one PDU, are sent out together using the currently configured write community string/user profile. Internally, multiple Set operations are performed, one for each PDU is displayed.

- **Stop Query**: You can use this toolbar icon to abort a Set operation that is currently in progress when, say, the agent is not responding or some unexpected error has occurred. This icon is enabled only while "Set" operation is in progress.

- **Refresh**: You can use this toolbar icon to get the values of the variables once again. It will initiate a Snmp Get Operation.

- **Save Val File**: You can use this toolbar icon to save the current list of variables and their instances and values for future use against the same or another device. The File Save screen will prompt you to specify the name of the .val file. The variable and the values displayed are saved into this file. If the file already exists, you will be asked whether to over-write it.

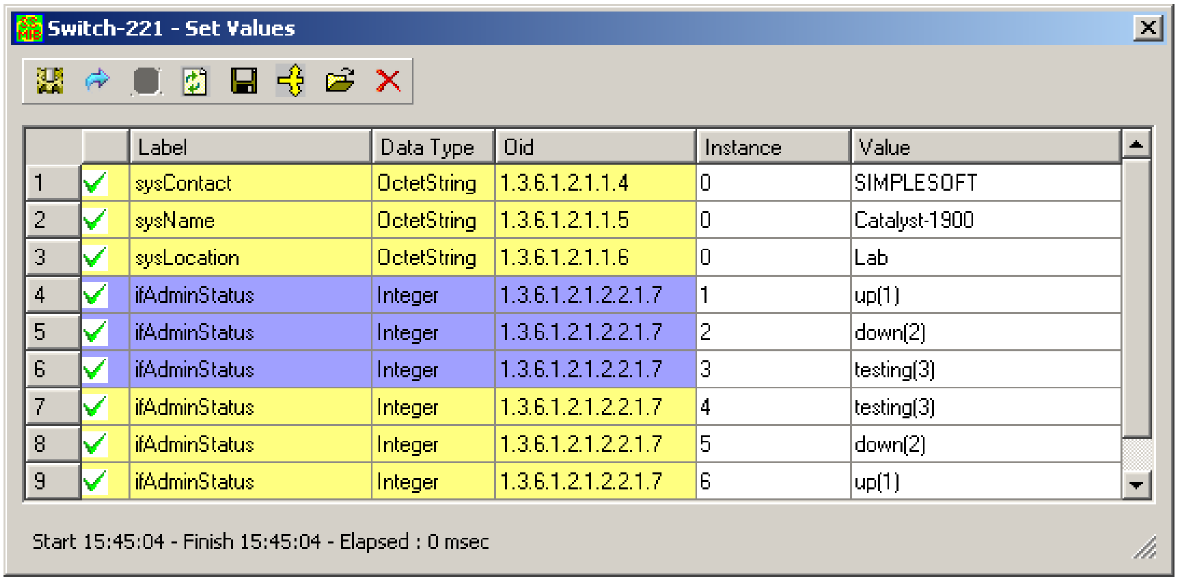
- **Pdu Split**: You can use this toolbar icon to manually split the PDU in different ways. If you select a line/row and then click on the split PDU icon, then it will split the existing PDU in two parts. From second part onwards the color of the pdu split gets toggled. The Pdu Split does not take place if you select the last line from that existing PDU.

- **Load Val File**: You can use this toolbar icon to load variables from an existing val file. When you click on this icon, a File Open Dialog box will allow you to select the existing val file. Once a file is selected, its contents will be displayed. If the file contains the '%PduSplitMarker', then PDU is created according to '%PduSplitMarker' and not using the 'MaxVarBinds in Set Request' which is present in 'Advanced Tab' under Preferences Dialog. If the file does not contain '%PduSplitMarker tokens, the PDU is created according the MaxVarBinds in 'Set Request' which is present in 'Advanced Tab' under Preferences Dialog.

- **Delete a Row**: You can use this toolbar icon to remove a variable from the list of variables to be set. If that variable was the last variable in the PDU, the PDU itself will be deleted, and the subsequent PDU colors will be toggled, if any.

Once the Set action is initiated, it can result in the following screen being displayed.

a) If the Set was successful, then a dialog similar to the following is displayed.

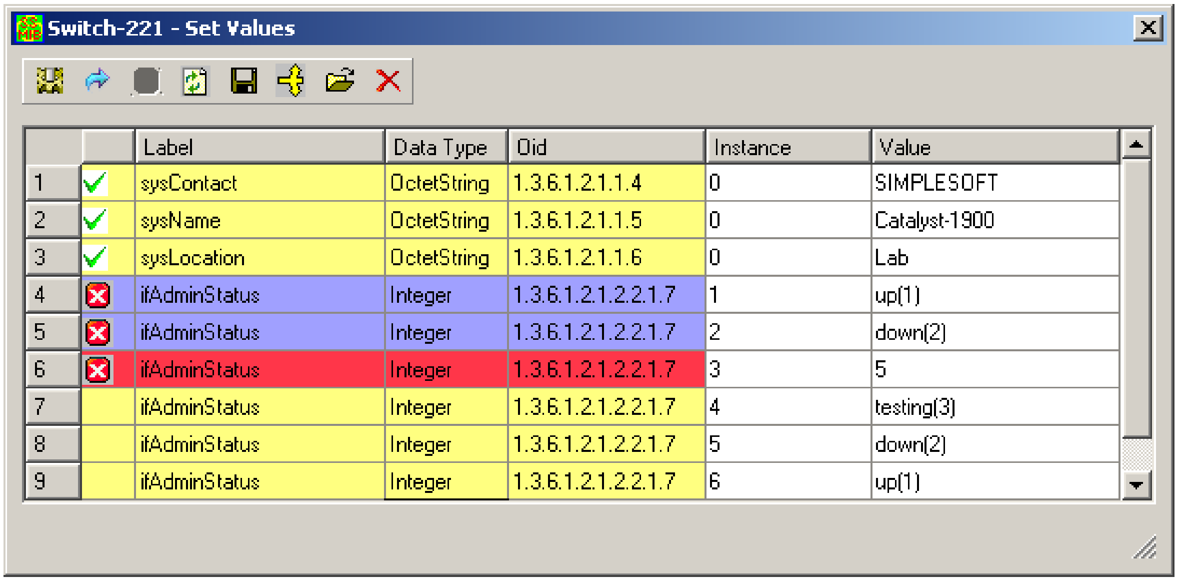


		Label	Data Type	Oid	Instance	Value
1	✓	sysContact	OctetString	1.3.6.1.2.1.1.4	0	SIMPLESOFT
2	✓	sysName	OctetString	1.3.6.1.2.1.1.5	0	Catalyst-1900
3	✓	sysLocation	OctetString	1.3.6.1.2.1.1.6	0	Lab
4	✓	ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	1	up(1)
5	✓	ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	2	down(2)
6	✓	ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	3	testing(3)
7	✓	ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	4	testing(3)
8	✓	ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	5	down(2)
9	✓	ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	6	up(1)

Start 15:45:04 - Finish 15:45:04 - Elapsed : 0 msec

Figure 24: Set results

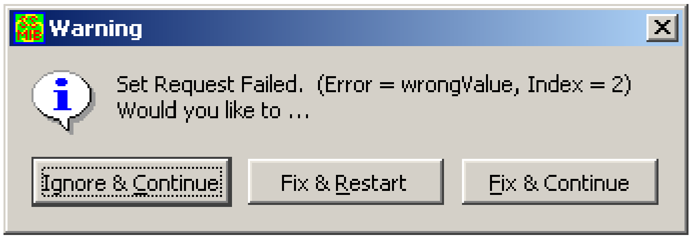
b) If the Set was unsuccessful, due to any SNMP Error, then a dialog similar to the following is displayed.



		Label	Data Type	Oid	Instance	Value
1	✓	sysContact	OctetString	1.3.6.1.2.1.1.4	0	SIMPLESOFT
2	✓	sysName	OctetString	1.3.6.1.2.1.1.5	0	Catalyst-1900
3	✓	sysLocation	OctetString	1.3.6.1.2.1.1.6	0	Lab
4	☒	ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	1	up(1)
5	☒	ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	2	down(2)
6	☒	ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	3	5
7		ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	4	testing(3)
8		ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	5	down(2)
9		ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	6	up(1)

Figure 25: Set Results containing errors

The exact SNMP Error is also displayed in the error message box as shown below.

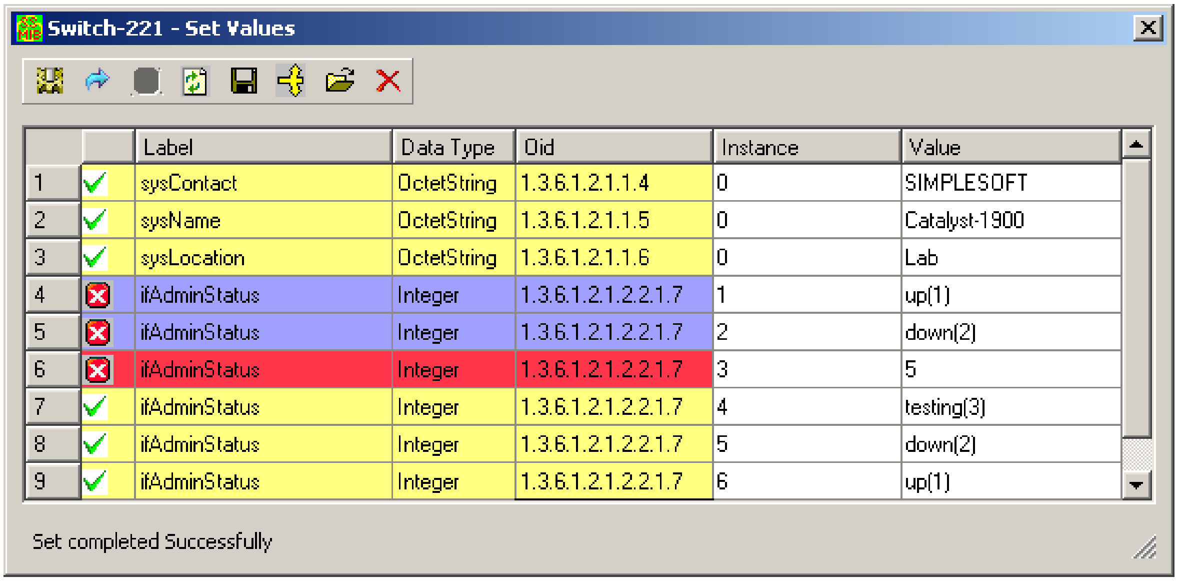


Set Request Failed. (Error = wrongValue, Index = 2)
Would you like to ...

Figure 26: Error display

In the above case, the first PDU was successfully set and so it was indicated by the green tick-mark signs. The second PDU was, however, unsuccessful and therefore shown with red cross signs with the variable causing the actual error being highlighted in red. The message box explains the reason for the error.

If you click on the 'Ignore & Continue' button, the existing error PDU is ignored and the next PDU is sent for the 'Set' operation. Then a dialog similar to the following is displayed.

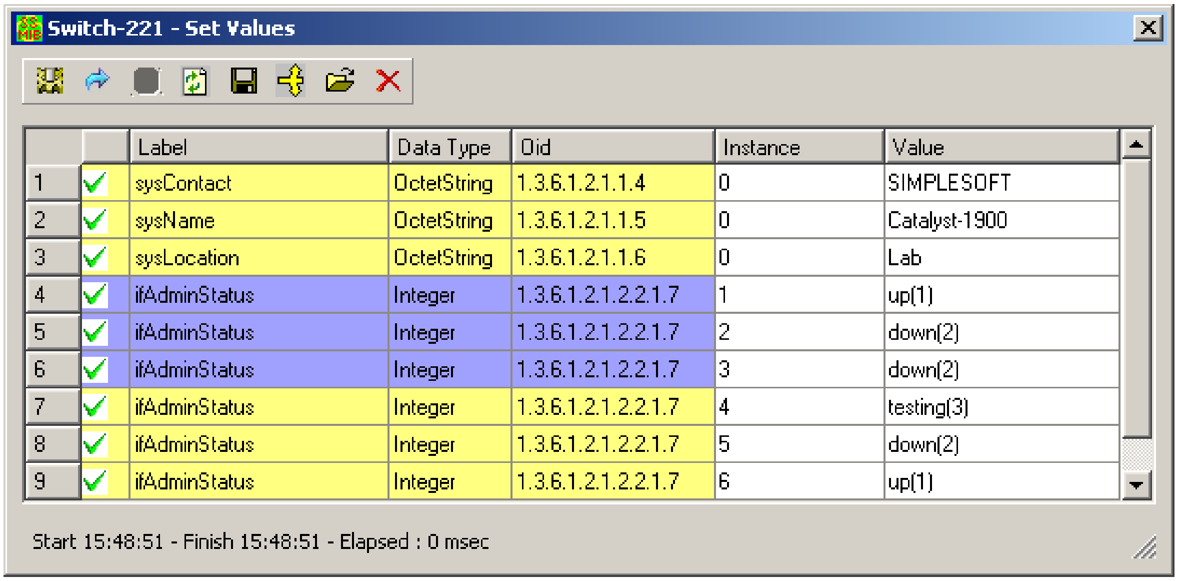


		Label	Data Type	Oid	Instance	Value
1	✓	sysContact	OctetString	1.3.6.1.2.1.1.4	0	SIMPLESOFT
2	✓	sysName	OctetString	1.3.6.1.2.1.1.5	0	Catalyst-1900
3	✓	sysLocation	OctetString	1.3.6.1.2.1.1.6	0	Lab
4	☒	ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	1	up(1)
5	☒	ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	2	down(2)
6	☒	ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	3	5
7	✓	ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	4	testing(3)
8	✓	ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	5	down(2)
9	✓	ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	6	up(1)

Set completed Successfully

Figure 27: Set results with errors but continuing to subsequent varbinds

If you select the 'Fix & Restart' button, the existing 'Set' operation is stopped. You can now change the values or the settings and then choose to click on the "Set" toobar icon again. The actual 'Set' operation is now once again initiated from the very first PDU. Then a dialog similar to the one below might get displayed if the errors were corrected.



		Label	Data Type	Oid	Instance	Value
1	✓	sysContact	OctetString	1.3.6.1.2.1.1.4	0	SIMPLESOFT
2	✓	sysName	OctetString	1.3.6.1.2.1.1.5	0	Catalyst-1900
3	✓	sysLocation	OctetString	1.3.6.1.2.1.1.6	0	Lab
4	✓	ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	1	up(1)
5	✓	ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	2	down(2)
6	✓	ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	3	down(2)
7	✓	ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	4	testing(3)
8	✓	ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	5	down(2)
9	✓	ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	6	up(1)

Start 15:48:51 - Finish 15:48:51 - Elapsed : 0 msec

Figure 28: Set results with corrections

If you select the 'Fix & Continue' button, the existing error PDU index is located and given for correction. You can either remove the variable, change its value, or change the settings to fix the problem. Once you decide that the problem could be fixed, you can click on the "Set" toolbar icon again. This time, the Set operation is started from the errror PDU and not from the very beginning as was done with the "Stop" button. A dialog similar to the one above might get displayed if the errors were corrected.

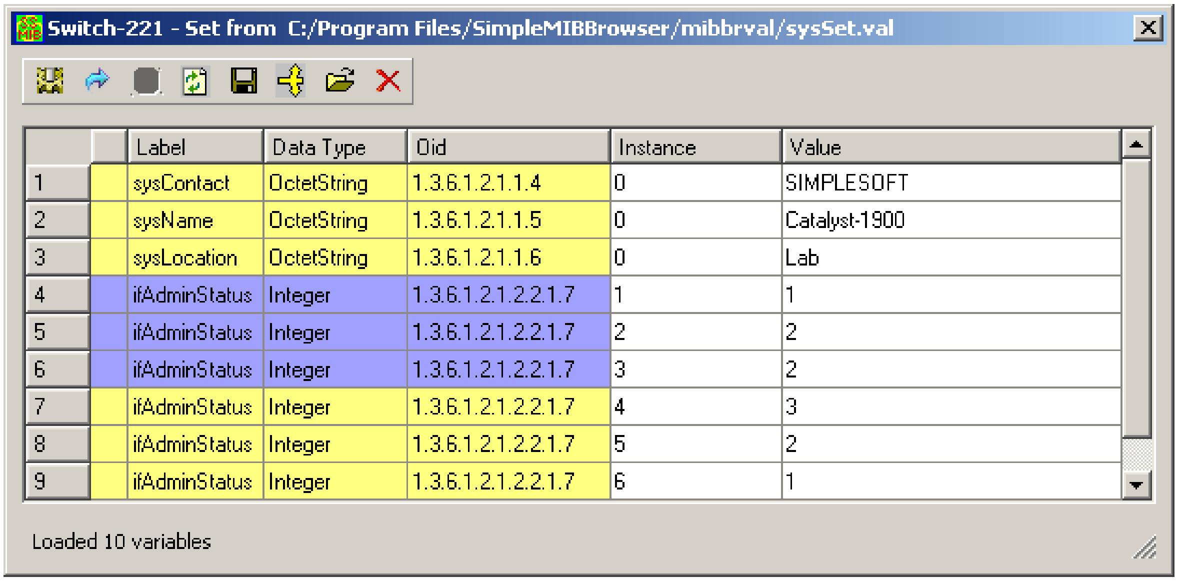
Note in all the three cases above, you can see the status of the each PDU in the status bar, by selecting a row from the dialog box.

Set From File

You can use this menu to set a group of variables to the values specified in the file. This can be used when trying to restore a pre-existing configuration or making similar changes to many devices.

A File Open dialog with file type as '.val' file allows you to select a pre-existing file. Its contents will be displayed in the set dialog and broken up into PDUs as per the configurable option named 'MaxVarBinds in Set Request' which is present in 'Advanced Tab' under Preferences Dialog. If the file contains '%PduSplitMarker' token, then it is used instead. PDU is represented using two different colors.

All columns will be filled from reading the .val file and presented to you. The Oid column will not contain the instance part, but only what is available from the file. The Instance column will contain a "0" for the scalar variables and Non-zero for columnar variables. The value column contains value which is present in the file.



Switch-221 - Set from C:/Program Files/SimpleMIBBrowser/mibbrval/sysSet.val

		Label	Data Type	Oid	Instance	Value
1		sysContact	OctetString	1.3.6.1.2.1.1.4	0	SIMPLESOFT
2		sysName	OctetString	1.3.6.1.2.1.1.5	0	Catalyst-1900
3		sysLocation	OctetString	1.3.6.1.2.1.1.6	0	Lab
4		ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	1	1
5		ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	2	2
6		ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	3	2
7		ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	4	3
8		ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	5	2
9		ifAdminStatus	Integer	1.3.6.1.2.1.2.2.1.7	6	1

Loaded 10 variables

Figure 29: Example where Set is saved to a file

The Instance for columnar variables and Value columns are editable and will remain so even if you edit them and add a value into the cell.

Adding a new row to a SNMP table

You can use this toolbar icon (from the Get Table Dialog) when trying to add a row to an existing table.

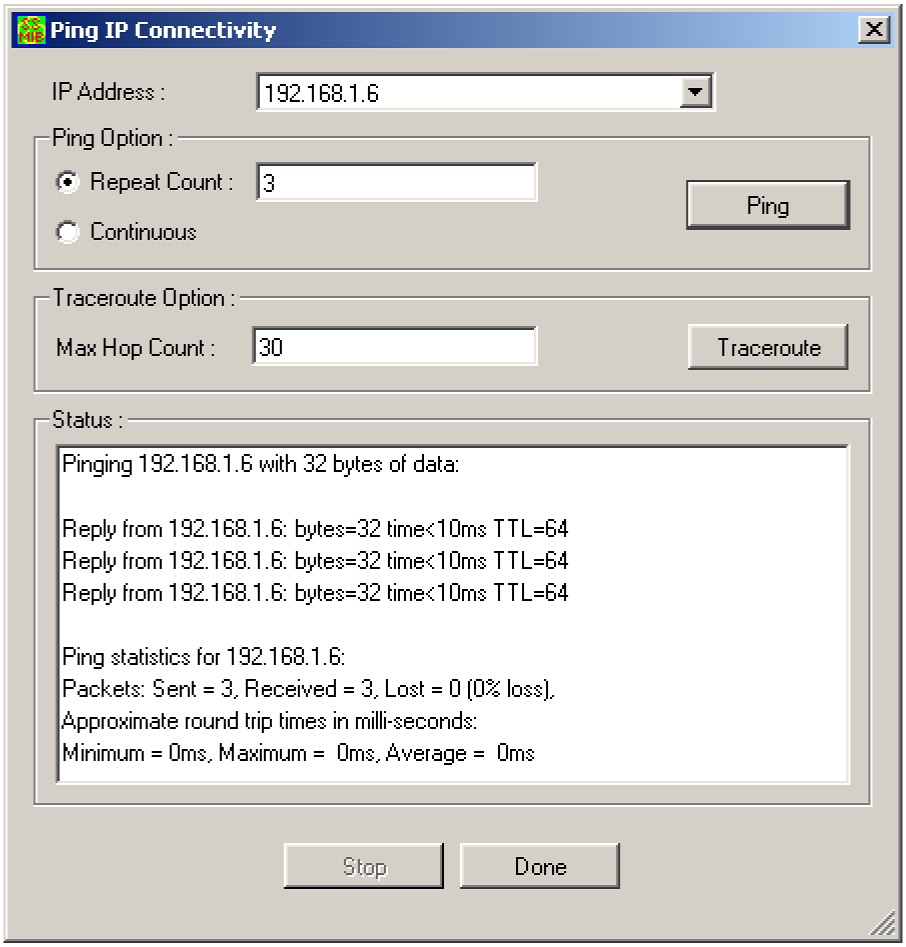
Currently, it does not understand the semantics of RowStatus, EntryStatus, etc., but future versions will include this functionality.

This menu option is provided for adding a row to the selected table. This menu option will be enabled if only one table or entry node is selected. An empty row in the non-modal dialog box will be displayed in the form of "Table Query Results". All the column cells will be editable, and the instance value will be generated automatically from the values specified in the index columns. This is the same as the Set Operation.

Chapter 7
Checking for IP connectivity

SimpleMIBBrowser also includes the capability to check for IP connectivity by sending ICMP Ping and Traceroute commands to the device that you are trying to communicate with. This allows you to diagnose the network connectivity problems.

The Ping/Traceroute connectivity checking is available on the toolbar and also under the Action menu.



Ping IP Connectivity

IP Address : 192.168.1.6

Ping Option :

● Repeat Count : 3

○ Continuous

Traceroute Option :

Max Hop Count : 30

Status :

```
Pinging 192.168.1.6 with 32 bytes of data:

Reply from 192.168.1.6: bytes=32 time<10ms TTL=64
Reply from 192.168.1.6: bytes=32 time<10ms TTL=64
Reply from 192.168.1.6: bytes=32 time<10ms TTL=64

Ping statistics for 192.168.1.6:
Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum =  0ms, Average =  0ms
```

Figure 30: Ping/Traceroute screen

IP Address:

This field shows the currently selected device and the drop down list contains all the previously entered devices.   You can also enter new IP addresses in this field.

Ping Button:

You can click on the Ping button to send ICMP ping requests to the selected device. Ping requests will be sent base on the selected option of Repeat Count or Continuous. If the Continuous option is selected, the application will periodically keep sending requests until you click on the "Stop" button.

The results of the ping operation are shown in the Status Window.

Traceroute Button:

You can click on this button to send a traceroute request to the selected device.   The maximum specified hop count is used to generate a series of traceroute requests with increasing hop counts.

The results of the trace route operation are also shown in the Status Window.

Stop Button:

You can click on this button to stop a currently running ping or traceroute. A confirmation screen will be displayed prior to stopping the current action.

Done Button:

You can click on this button to close the window if no operation is currently running. If a ping or traceroute is currently running, you will be prompted with a confirmation screen to verify that you want to stop the current operation and close the window.

Chapter 8
View Log

SimpleMIBBrowser also includes the capability to log all the packets sent and received by it. Based on the debugging options selected in the Prefrences settings, the log file can contain raw packet dumps or formatted PDU information or both.

Sample Log File

Example contents of the log file are shown below.

1) Packet Trace while doing Get (walk) on variable named as sysDescr

```
****Sending Pkt to 192.168.1.221/161****
PACKETTRACE:(Mon Mar 14 19:02:41 2005):SENT:
30 82 00 28 02 01 00 04 07 70 75 62 6c 69 63 31 a1
1a 02 01 00 02 01 00 02 01 00 30 82 00 0d 30 0b 06
07 2b 06 01 02 01 01 01 05 00

****Received Pkt from 192.168.1.221/161****
PACKETTRACE:(Mon Mar 14 19:02:41 2005):RECEIVED:
30 82 00 3b 02 01 00 04 07 70 75 62 6c 69 63 31 a2
2d 02 01 00 02 01 00 02 01 00 30 82 00 20 30 1e 06
08 2b 06 01 02 01 01 01 00 04 12 4d 79 5f 73 79 73
44 65 73 63 72 5f 73 74 72 69 6e 67

****Sending Pkt to 192.168.1.221/161****
PACKETTRACE:(Mon Mar 14 19:02:41 2005):SENT:
30 82 00 29 02 01 00 04 07 70 75 62 6c 69 63 31 a1
1b 02 01 01 02 01 00 02 01 00 30 82 00 0e 30 0c 06
08 2b 06 01 02 01 01 01 00 05 00


****Received Pkt from 192.168.1.221/161****
PACKETTRACE:(Mon Mar 14 19:02:41 2005):RECEIVED:
30 82 00 2b 02 01 00 04 07 70 75 62 6c 69 63 31 a2
1d 02 01 01 02 01 00 02 01 00 30 82 00 10 30 0e 06
```

```
08 2b 06 01 02 01 01 02 00 06 02 2a 03
```

2) PDU dump when doing Get (walk) on sysDescr variable

```
Mon Mar 14 19:14:46 2005
GetNext Request :  Version = SNMPv1, Community = public1
Request Id = 0, Error Status = No Error , Error Index = 0
Oid1 = 1.3.6.1.2.1.1.1 , Type = NULL, Value = NULL

Mon Mar 14 19:14:46 2005
Response :  Version = SNMPv1, Community = public1
Request Id = 0, Error Status = No Error , Error Index = 0
Oid1 = 1.3.6.1.2.1.1.1.0 , Type = OctetString, Value = My_sysDescr_string

Mon Mar 14 19:14:46 2005
GetNext Request :  Version = SNMPv1, Community = public1
Request Id = 1, Error Status = No Error , Error Index = 0
Oid1 = 1.3.6.1.2.1.1.1.0 , Type = NULL, Value = NULL

Mon Mar 14 19:14:46 2005
Response :  Version = SNMPv1, Community = public1
Request Id = 1, Error Status = No Error , Error Index = 0
Oid1 = 1.3.6.1.2.1.1.2.0 , Type = ObjectID, Value = 1.2.3
```

Appendix A:
Reference Books and Material

This appendix lists some of the reference material available that provides guidelines for MIB design and provides general information on SNMP.

SNMP MIB related

1. "Understanding SNMP MIBs" by David Perkins and Evan McGinnis, Prentice Hall Inc. (ISBN 0-13-437708-7)

2. Internet Draft , "Guidelines for MIB Authors and Reviewers", C.M.Heard , Editor. <draft-ietf-ops-mib-review-guidelines-02.txt>.

SNMP related

1. "SNMPv1, SNMPv2, SNMPv3, RMON and RMON2" by William Stallings, Third Edition, Addison-Wesley. (ISBN 0-201-48534-6)

2. SimpleTimes (http://www.simple-times.org/)

Appendix B: INI File Options

This Appendix lists some of the options available in the SMBrowser.ini file that allows you to customize the runtime execution environment of the SimpleMIBBrowser. The mibbrowser.ini file is found in the <*install_dir*>/SimpleMIBBrowser/bin directory.

The Device and User Profiles are stored in this file, so that the SimpleMIBBrowser may be more easily configured when sending subsequent requests.

Appendix C: Frequently Asked Questions

Questions

1) **Does the SimpleMIBBrowser support hardware or dongle based licensing like SimpleSoft's other products ?**

   No, the SimpleMIBBrowser only supports software key based licensing. If a hardware key based scheme is required, please contact SimpleSoft's Sales department and let them know your preferences.

2) **What are the differences in demo mode and full version mode ?**

   The demo version is provided to give you an idea about the product's capabilities prior to purchase. All the capabilities of the full version (version received after purchase) are available in the demo version, except that the demo version limits the number of nodes displayed in the MIB tree, and restricts the number of SNMP requests sent out and traps received. An expired demo version further curtails these limits.

3) **What kind of help is available to use the SimpleMIBBrowser ?**

   The manual for the SimpleMIBBrowser is available in the <install_dir>/SimpleMIBBrowser/docs directory as a Adobe PDF file to allow platform independent viewing. Information is also available in html format by using the Help menu. .

4) **Where can I get the IETF RFC documents referred by the manual and in the on-line help ?**

   The authoritative way to get IETF related documents is by going to their website at http://www.ietf.org. For convenience, some of the MIBs are also included in the installation and available in the <install_dir>/SimpleMIBBrowser/mibs/ietf folder.

.

Appendix D: Copyright Notices

This Appendix lists all the related copyright notices.

SIMPLESOFT Inc.:

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph(c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

SIMPLESOFT Inc. 257 Castro Street, Suite 220, Mountain View, CA 94041

/***

Copyright 1988, 1989, 1991, 1992 by Carnegie Mellon University

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of CMU not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

CMU DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL CMU BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
***/

This software is copyrighted by the Regents of the University of California, Sun Microsystems, Inc., and other parties. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files.

The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions. No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

RESTRICTED RIGHTS: Use, duplication or disclosure by the government is subject to the restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software Clause as DFARS 252.227-7013 and FAR 52.227-19.

Compilation Copyright (c) 1995-2000 by Wei Dai. All rights reserved.
This copyright applies only to this particular software distribution
package, and does not imply a copyright on any particular module in
the library.

All modules in this library except the following are placed in
the public domain by Wei Dai and other contributers.

The following modules are copyrighted by their respective original authors.

cast.cpp - Copyright 1996 Peter Gutmann.
haval.cpp - Copyright 1992 Yuliang Zheng.
idea.cpp - Copyright 1992 Colin Plumb.
mars.cpp - Copyright 1998 Brian Gladman.
md2.cpp - Copyright 1994, 1995 Sun Microsystems, Inc.
md5.cpp md5mac.cpp - Copyright 1995 Eric Young.
rijndael.cpp - Copyright 1998, 1999 Brian Gladman.
serpent.cpp - Copyright 1998, 1999 Brian Gladman and Sam Simpson.
sha.cpp - Copyright 1992 Peter Gutmann.
twofish.cpp - Copyright 1998, 1999 Brian Gladman.
zbits.cpp ztrees.cpp zdeflate.cpp - Copyright 1990-1992 Mark Adler,
Richard B. Wales, Jean-loup Gailly.

Permission to use, copy, modify, and distribute this compilation for
any purpose is hereby granted without fee, subject to
the following license:

1. Any copy or modification of this compilation must include the above
copyright notice and this license.

2. Users of the software included in this compilation agree to use
their best efforts to provide Wei Dai with any modifications containing
improvements or extensions and hereby grant Wei Dai a perpetual,
royalty-free license to use and distribute such modifications under
the terms of this license.

3. Wei Dai makes no warranty or representation that the operation of the
software in this compilation will be error-free, and Wei Dai is under no
obligation to provide any services, by way of maintenance, update, or
otherwise. THE SOFTWARE AND ANY DOCUMENTATION ARE PROVIDED "AS IS"
WITHOUT EXPRESS OR IMPLIED WARRANTY INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. IN NO EVENT WILL WEI DAI OR ANY OTHER CONTRIBUTOR BE LIABLE FOR
DIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF
ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.