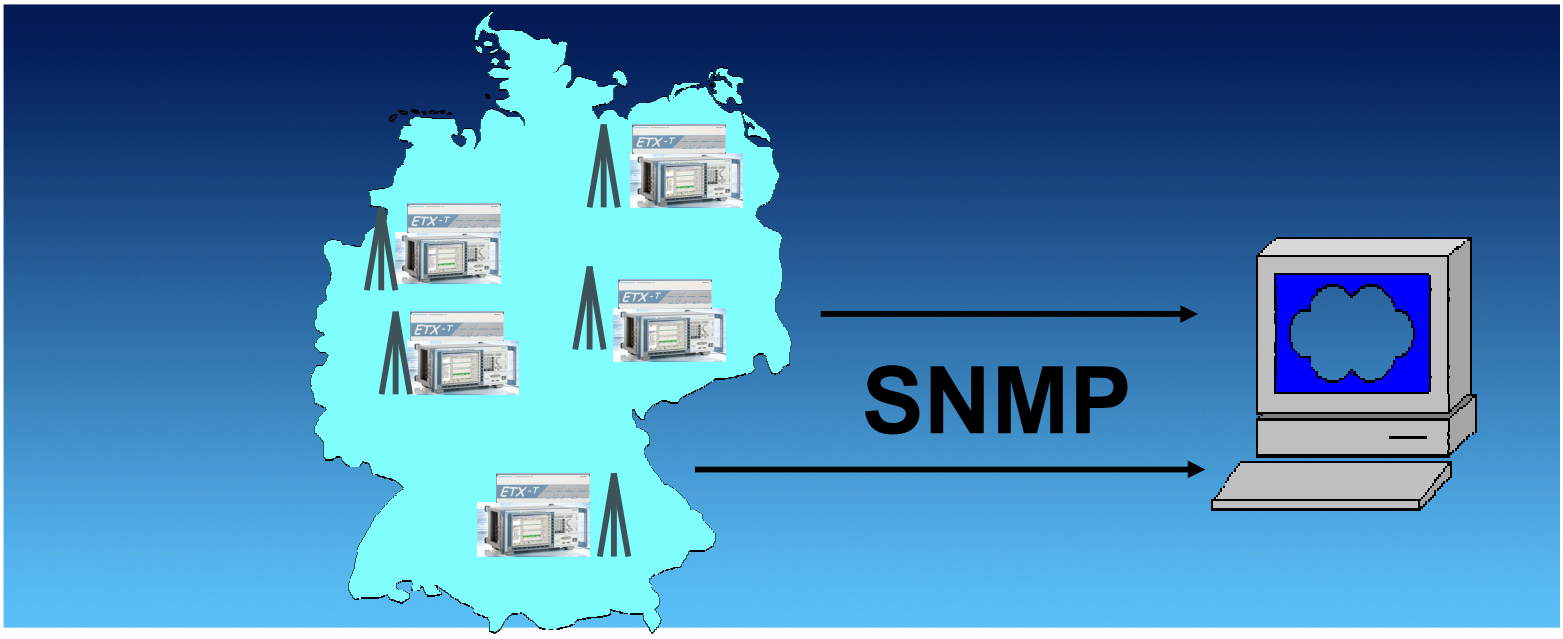


Products: R&S®DVM50, R&S®DVM100, R&S®DVM100L, R&S®DVM120, R&S®DVM400, R&S®ETX

Simple Network Management Protocol
Remote Controlling for Monitoring Devices
Basics, Tools, Examples, and Development Tips

Application Note

An operator of a network of terrestrial transmitter stations wants to have the most cost-effective means of monitoring the network's numerous transmitter sites from a main monitoring center. Modern error sensors such as the R&S®DVM50/100 and the R&S®ETX-T are used for diverse monitoring functions at the individual transmitter sites as well as to transmit monitoring status to the main monitoring center. These error sensors have a function-rich SNMP agent designed specifically for this purpose. Simple network management protocol (SNMP). Despite the "simple" in the protocol name, some users are still afraid to use it for remote control and monitoring.

This Application Note therefore provides a user-friendly and practical look at, for example, how the monitoring functions via SNMP can be used for the R&S®DVM. The following sections provide a brief look at the SNMP protocol, the R&S®DVM/ETX implementation, and useful tools for working with SNMP. A description of how SNMP can be linked to various development environments (C++, C#, Java) is also provided.

ROHDE&SCHWARZ

Subject to change – H. Gsoedl 03.2007 – 7BM65_1E

Content

1 Overview

The R&S®DVM family and the R&S®ETX-T from Rohde & Schwarz offer the functions required in order to monitor the MPEG-2 transport stream used in digital TV to transmit pictures, video, and other data, to monitor the digitally modulated RF signal (DVB-T), and to evaluate their quality at any time.

An operator of a network of terrestrial transmitter stations wants to have the most cost-effective means of monitoring the network's numerous transmitter sites from a main monitoring center.

Modern error sensors such as the R&S®DVM50/100 and the R&S®ETX-T are used for diverse monitoring functions at the individual transmitter sites as well as to transmit monitoring status to the main monitoring center. These error sensors have a function-rich SNMP agent designed specifically for this purpose. Simple network management protocol (SNMP). Despite the "simple" in the protocol name, some users are still afraid to use it for remote control and monitoring.

This Application Note therefore provides a user-friendly and practical look at, for example, how the monitoring functions via SNMP can be used for the R&S®DVM/ETX. The following sections provide a brief look at the SNMP protocol, the R&S®DVM/ETX implementation, and useful tools for working with SNMP. A description of how SNMP can be linked to various development environments (C++, C#, Java) is also provided. For your own development purposes, the programs and libraries that are used are supplied as a ZIP file accompanying this Application Note (7BM65_1E_Development.zip; available from the download area for Application Notes on the Rohde & Schwarz Internet site).

2 Introduction to SNMP

To understand why the R&S®DVM family and the R&S®ETX-T use SNMP for remote control, a brief overview of the origins and the actual function of this protocol follows.

History of SNMP

When the Internet age started in the late 1980s, many voices called for an administration tool for the global Internet network. A tool was needed that provided a simple and reliable means of performing functions such as configuring and monitoring the network components involved.

Given these requirements, the Internet Architecture Board (IAB) set out to create a protocol that permits exactly this. The IAB is an advisory group within the Internet Society (ISOC). Background information: As an umbrella organization, the ISOC is dedicated to the systematic development of the technical aspects of the Internet.

Based on earlier protocols (simple gateway monitoring protocol) and lengthy expert discussions with the participation of the International Organization for Standardization (ISO) and the International Telecommunication Union (ITU-T), version 1 of SNMP (SNMPv1) was then adopted. SNMPv1 covers the following "request for comment" (RFC) documents since the early 1990s:

- RFC 1155:

  Structure and Identification of Management Information for TCP/IP-based internets

- RFC 1156:

  Management Information Base for Network Management of TCP/IP-based internets

- RFC 1157:

  A Simple Network Management Protocol

In the ensuing years, further SNMP versions (SNMPv2, SNMPv3) were published. Their intent is to eliminate the security and performance problems found in SNMPv1.

SNMP – a typical client/server application

As mentioned earlier, the purpose of SNMP is to manage network components in a system. These administration tasks are performed on a central management console. The individual network components can communicate with the management console by means of a locally installed application, the agent.

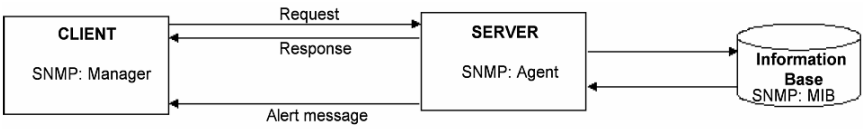


Fig. 1: Client/Server concept

Communication is normally initiated by the client. The client (manager) sends a request for read or write access to the server (agent). The agent reads or writes the wanted value from/to the local information base, and responses with a status information and the current data value.

In addition to the communication initiated by the client, unidirectional communication from the server to the client can take place. This type of communication is particularly useful for sending alert messages.

The management information base (MIB)

Each network component has a management information base (MIB). The MIB consists of "managed objects". It is a hierarchically arranged collection of information that lists all objects that can be accessed via SNMP (reading, writing). The objects can be accessed by means of an "object identifier" (OID). Each object has one instance (scalar: instance is normally addressed with "0") or

multiple instances (columnar: instance is selected by an index). Objects may be both tabular and multidimensional in structure.

The following figure illustrates the tree-like structure of the MIB. The tree consists of "public" and "private" nodes. The public nodes, located at the top of the hierarchy, are directly managed by the standardization bodies responsible for each. The "private Enterprises" node allows companies to define their own MIBs.

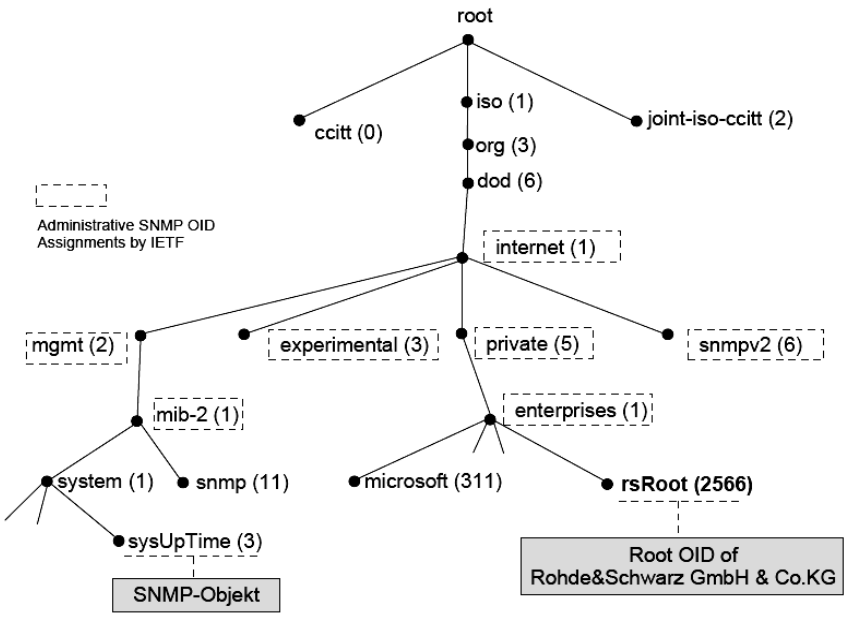


Fig. 2: Managed information base (MIB) tree

The SNMP makes it possible to access the instances of the individual objects by means of specific functions.

The message types with SNMP

The manager as well as the agents are able to exchange messages with each other by means of various functions, or to issue queries as needed.

In SNMP, the essential basic functions (also referred to as *protocol data unit* (PDU) type) are as follows:

- GET (SNMPv1,2):

  Reads the value (content) of an SNMP object whose complete identification number (OID) is known (individual variables such as date or time).

- GETNEXT (SNMPv1,2):

  Reads the value (content) of the SNMP object that logically follows the object referenced by the given identification number (OID) within the MIB implemented in the agent.

- GETBULK (SNMPv2):

  Expansion of the GETNEXT function that makes it possible to issue only one request in order to obtain an entire sequence of objects as a response.

- SET (SNMPv1,2):

  Writes the value (content) of an SNMP object whose complete identification number (OID) is known (e.g. individual variables such as date or time).

- INFORM (SNMPv2):

  When a specific event occurs, an SNMP *informer* sends a message to one or more management systems. INFORM messages must be acknowledged by the manager.

- TRAP (SNMPv1,2):

  This message type enables agents to asynchronously report events to a manager. The manager does not issue an acknowledgment.

Access rights with SNMP

Special rights are required in order to read or write variables via SNMP. These rights are assigned by means of *community strings*. The following rights are provided for SNMP (v1 and v2):

- Read community
- Write community

These are basically string values, i.e. normal passwords.

3 Brief Look at Network Technology

UDP/IP networks serve as the communications path for the SNMP. This type of network represents a group of computer systems. The underlying purpose is to transmit data or to make central resources simultaneously available to multiple computers.

A typical network can be illustrated as follows:

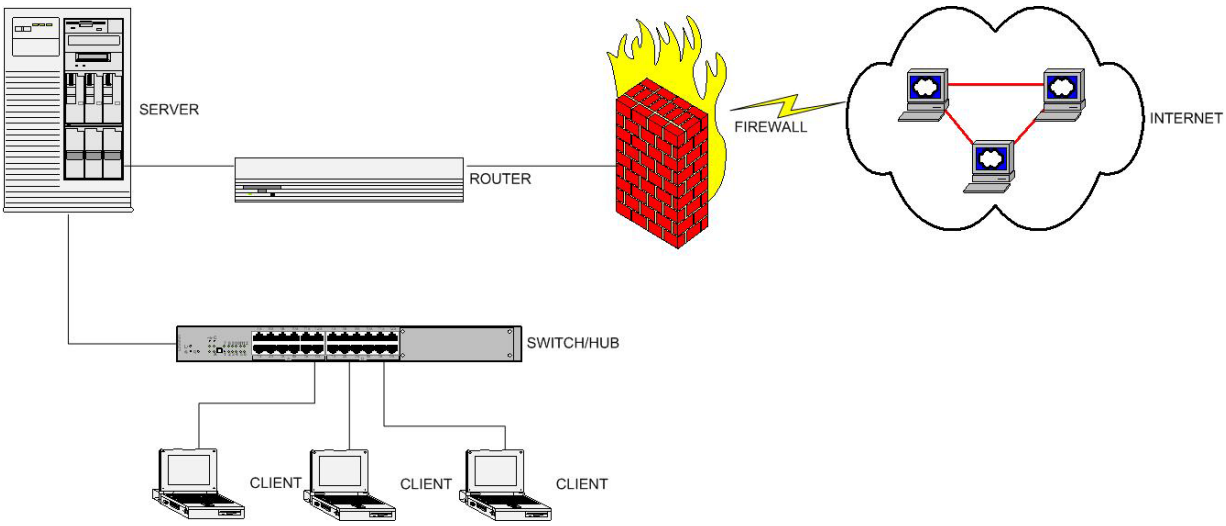


Fig. 3: Typical network configuration

As the illustration shows, a network consists of various components:

- Client:

  This term means a computer system that accesses non-local services of a server via a network.

- Server:

  In the classical sense, a server is a computer system whose sole purpose is to provide services for clients. Typical server applications include mail service and file server.

- Switch/Hub:

  Instruments that allow several computer systems to be interconnected to form a physical network.

- Router:

  The router handles the connection between individual network segments.

- Firewall:

  The purpose of a firewall is to block unwanted data traffic (security barrier). The firewall can be a software or hardware solution, or both.

How is data transmission carried out?

In the case of TCP/IP networks, packet-switched networks are involved. The information to be transmitted is divided into small data fragments (~ 1500 bytes). These data fragments are referred to as *packets*.

Before data can be transmitted, it must first be divided into packets at the transmitter site and then reassembled at the receiver end.

The TCP/IP protocol defines the format of these packets:

- Packet header with
  - Origin and destination address
  - Port number
  - Length of the packet
  - Type of the packet
  - Specifications about how a packet is received and, if necessary, forwarded
- Data portion (payload)

How are addresses defined within such a network?

By using IP addresses and subnet masks to define addresses, you can subdivide a physical network, i.e. a network created by means of hardware (cables and switches), into logical units.

Each device in the type of network described above has a unique identification number. This identification number is called an IP address. Both the sender address and receiver address of an IP packet can be found in the packet's header.

The IP address is subdivided into the following fields:

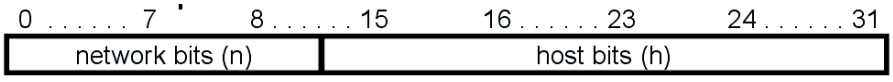


Fig. 4: Structure of the IP address

Normally, IP addresses are not represented bitwise, but, instead, by bytes in decimal format that are separated by decimal points:

Example: 10.124.10.187

The *network information center* (NIC), which is responsible for the assignment of IP addresses worldwide, distinguishes between several network classes.

	Address bits 0..31	First byte	Networks	Hosts
Class A	0nnnnnnn.hhhhhhhh.hhhhhhhh.hhhhhhhh	1-126	126	16777214
Class B	10nnnnnn.nnnnnnnn.hhhhhhhh.hhhhhhhh	128-191	16384	65534
Class C	110nnnnn.nnnnnnnn.nnnnnnnn.hhhhhhhh	192-223	2097375	254
Extended	111xxxxx. xxxxxxxx. xxxxxxxx. xxxxxxxx	224-254	undefined	undefined

Fig. 5: Network classes

As Fig. 4 shows, the IP address can be subdivided into two fields – the network bits and the host bits. The network bits are used to identify a specific network segment in the Internet, whereas the host bits are used to indicate the single devices in this network segment.

To improve administration within the network, for large networks instead of the above standard classification a specific network segmentation into subnets is possible. The subnet mask was created for this purpose:

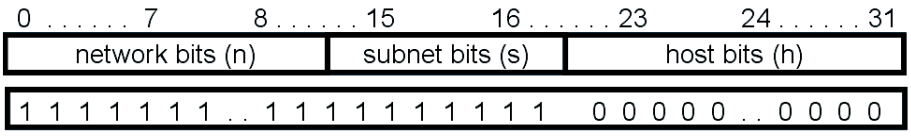


Fig. 6: Structure of the subnet mask

The IP address and subnet mask form a unit. The subnet mask determines which bits of the IP address belong to the network part and which to the host part. Starting with the first bit (at the left) of the subnet mask, "1" is set all the way across for the bits which belong to the network part (switching between 0 and 1 in the network part is not permitted). Network members whose IP addresses differ exclusively at "0" positions of the subnet mask are logically located in the same *network segment*. Bits belonging to the host part are marked in the subnet mask as “0” (switching between 0 and 1 in the host part is also not permitted). Data traffic to network members which do not belong to the same network segment is redirected to the gateway address (cf. indirect communication via routers).

Applications on the individual systems in the network use a further addressing method – the ports. Ports serve as multiplexers within a network. A port receives all packets addressed for specific services. An example is the HTTP protocol, where port 80 is the default port for receiving.

Conditions for successful communication between two members of a network

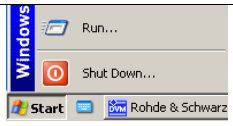
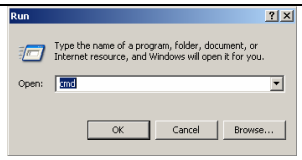
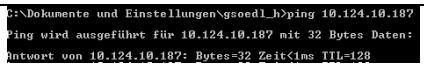
To permit mutual reachability, the following requirements must be met:

- The members communicating with one another must be located in the same subnet for direct communication, or interconnectivity is provided by a routing mechanism.
- Data transmission between the two network resources involved must not be restricted by any firewall mechanisms in the network or locally on the systems.

Basic reachability of the members of the network

Measures must be taken to ensure that specific IP addresses can be reached. A well-known utility has proven useful for this purpose: Ping.

To run the utility, you first must open the MS-DOS prompt under the Windows operating system. Under Windows XP, the procedure is as follows:

1.	From the Start menu, click Run.	
2.	In the window that opens, enter **cmd**.	
3.	Then enter **ping <IP address>** on the command line. If communication is successful, the response time to the query will be returned.	

Note: *The ping utility uses the Internet Control Message Protocol ICMP. Only in case, the ICMP packets can be sent and received between the devices without any blocking, the use of ping makes sense.*

Availability of specific ports for individual services such as SNMP

For individual services, specific communications ports are made available on the systems involved. However, if a service, e.g. the SNMP service, cannot be initiated even though the instruments can reach each other directly, the problem may be that a firewall is blocking the data traffic as shown in the following figure. Another possible reason is, that the SNMP service on the instrument is not active.

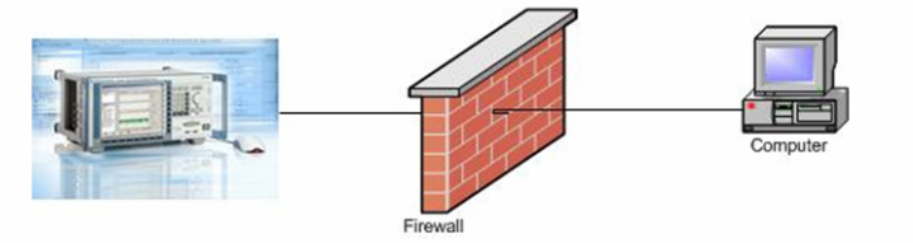


Fig. 7: A firewall can block data transfer

Note: *Rohde & Schwarz cannot provide support for the administration of customer networks. For matters regarding the configuration of your corporate network, contact your IT department.*

4 R&S®DVM/ETX and SNMP – An Overview of Functions

R&S®DVM MPEG-2 TS analyzer

The R&S®DVM supports the remote control of the following functions via SNMP:

rsDvmObjs

- Site configuration, analyzer configuration, configuration of the monitoring parameters (output, setting)
- Output of the monitoring results
- Output of the TS tree
- Output of the RF measurement values
- Output of the log data
- Configuration of the TS capture function
- Selection of specific R&S®DVM views, transport stream elements

rsDvmEvents

- Configuration of trap generation

Basic SNMP configuration on the R&S®DVM

To configure the community strings and the definition of the target IP addresses of R&S®DVM traps, you must edit the following configuration file on the R&S®DVM:

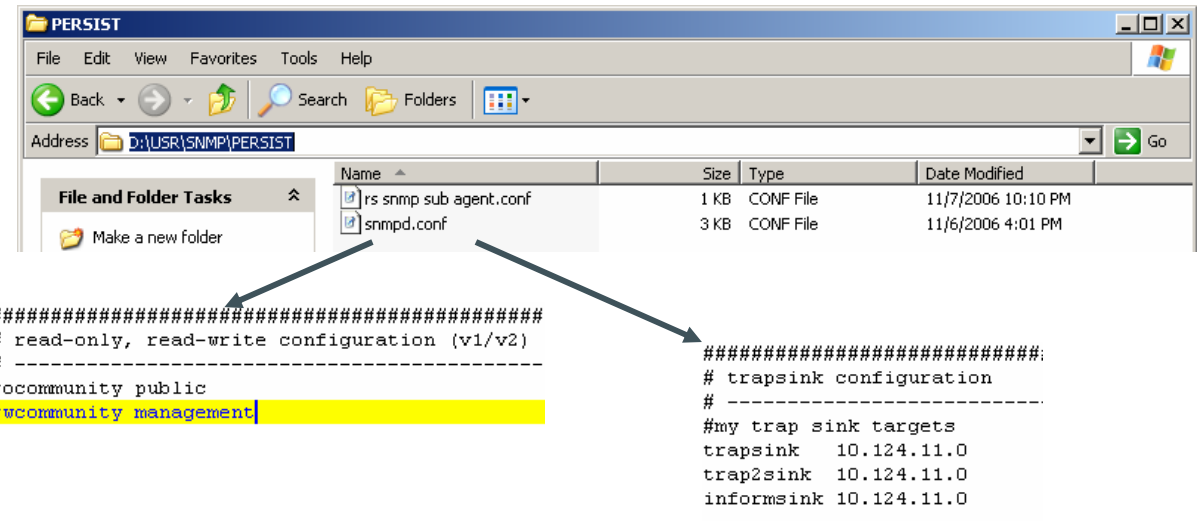


Fig. 8: Configuration file for the SNMP functionality

As shown in the figure above, making changes to the ReadOnly and ReadWrite community requires overwriting the *public* and *management* entries. For information on configuring the trap sinks, refer to section 7 of this document.

Rohde & Schwarz MIBs for the R&S®DVM

The instrument-specific MIB files of the R&S®DVM are located in directory D:\Programs\Rohde_Schwarz\DVM\Help of the R&S®DVM hard drive:

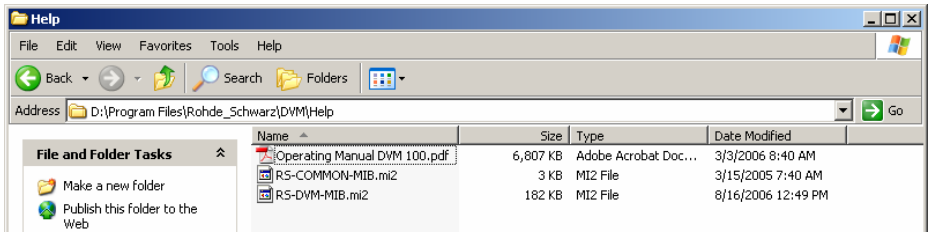


Fig. 9: Directory containing the instrument's MIB files

- RS-COMMON-MIB.mi2:

  Common MIB file for all Rohde & Schwarz instruments; linked in RS-DVM-MIB.mi2.
- RS-DVM-MIB.mi2: MIB file for the R&S®DVM

Self-monitoring of the SNMP agent on the R&S®DVM

To help ensure that the SNMP service functions properly, the R&S®DVM has a self-monitoring function – the SNMP watchdog. If an error prevents the SNMP service from functioning properly, the R&S®DVM automatically reboots.

To enable the watchdog, navigate to SNMP Configuration in the R&S®DVM GUI:

Fig. 10: Watchdog configuration

R&S®ETX RF monitoring system

The R&S®ETX-T supports the remote control of the following functions via SNMP:

- Output and setting of common system variables
- Configuration and reading of the MPEG-2 and RF measurement parameters/limit values
- Output of the MPEG-2 and RF measurement values
- Configuration of the trap function

The SNMP functions listed here can also be found in the MIB tree of the R&S®ETX:

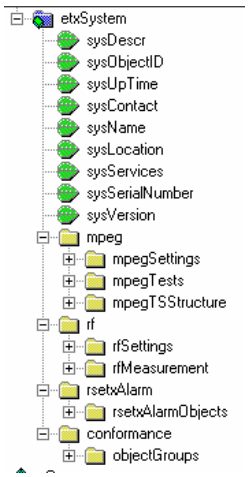
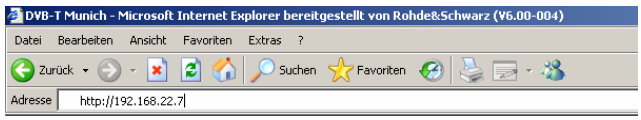


Fig. 11: MIB tree of the R&S®ETX-T

Basic SNMP configuration on the R&S®ETX-T

As also mentioned in section 3.2, *Administration*, of the R&S®ETX manual, the configuration of the SNMP functionality of the R&S®ETX can be carried out within the Administration → Network Configuration menu. To access this menu, do the following:

1.	Open a web browser and enter the IP address of the R&S®ETX: 

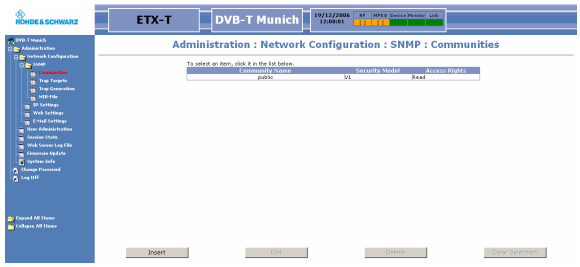
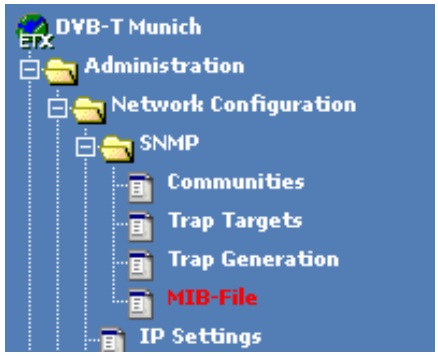
2.	Log on to the R&S®ETX as the super user (default login/password: super/super): 
3.	Open Administration → Network Configuration → SNMP: 

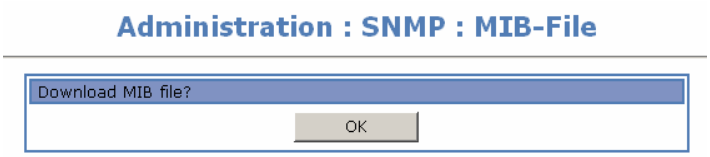
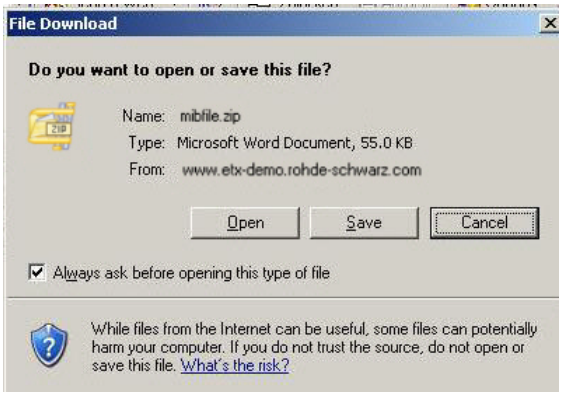
Fig. 12: SNMP configuration

In this subfolder, you can define the read and write community, configure the traps (see section 8), and download the instrument MIB (see next section).

Rohde & Schwarz MIB for the R&S®ETX-T

To obtain the MIB file of the R&S®ETX-T for local use, you must initiate a download from the web GUI. You can do this as follows (logon procedure identical to preceding section):

1.	From Administration → Network Configuration → SNMP, select MIB-File. 

2.	To start the download of the MIB file, select OK: 
3.	When the storage dialog opens, define the local folder in which you want to store the MIB file: 

5 Setups for SNMP Communication With the R&S®DVM/ETX

To communicate with the SNMP agent of the R&S®DVM/ETX, an Ethernet connection between the manager and the Rohde & Schwarz instrument is required. To demonstrate the configuration of agent (R&S®DVM or R&S®ETX-T) and manager system (user PC) for example, two common network configurations are presented in the following:

- Direct connection of the R&S®DVM/ETX with the management system
- Linking of the R&S®DVM/ETX to existing corporate networks

Note: *An inappropriate configuration of the network parameter on a single device in a network can result in severe troubles for the whole network system.*

Direct connection

One setup particularly of interest in development is the connection of the R&S®DVM/ETX directly with the management system:

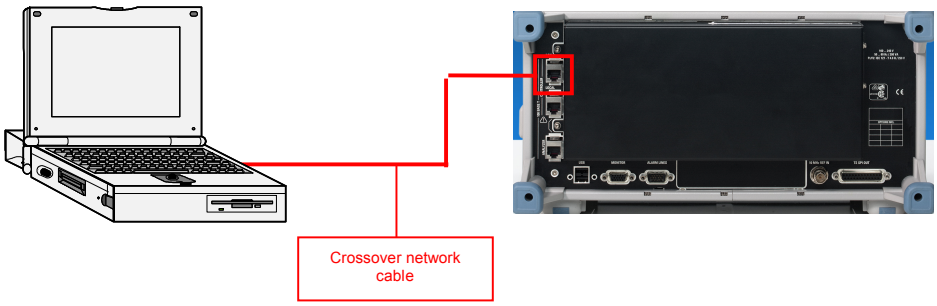


Fig. 13: Direct connection between measurement instrument and manager, using the R&S®DVM as an example

Note that this setup requires a crossover network cable. A crossover network cable can be identified by the differently arranged (crossover) wires on the RJ-45 connections:

Fig. 14: Crossover network cable

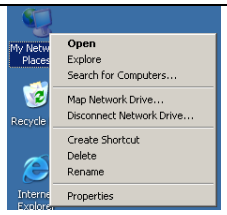
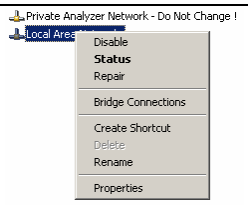
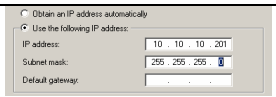
Manual configuration of the network address

Both the management system and the R&S®DVM/ETX must be configured properly with respect to the network address. As discussed in section 3, both instruments must be logically located in one subnet.

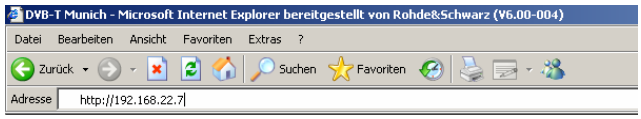
For example, the following address configuration may be used:

	Manager	R&S® DVM
IP address	192.100.10.201	192.100.10.202
Subnet mask	255.255.255.0	255.255.255.0

The network configuration of a fixed IP address may be done as follows under Windows:

1.	Using the right mouse button, click My Network Places and select Properties.	
2.	Right-click the required LAN adapter of the system; the properties of this network connection will be accessed. ***Note:*** *In the case of the R&S®DVM, do not change the configuration of the Private Analyzer Network adapter.*	
3.	Access the properties of the Internet Protocol (TCP/IP) entry.	
4.	Enter the IP address and subnet mask you want by enabling "Use the following IP address".	
5.	To accept your settings, close all windows with OK.	-

The configuration of a new network address for the R&S®ETX differs from the R&S®DVM. Settings are entered via the web GUI:

1.	Open a web browser and enter the IP address of the R&S®ETX:  ***Note:*** If you do not know the old IP address of the R&S®ETX, you can display it during bootup via a serial connection; see the R&S®ETX operating manual for details.
2.	Log on to the R&S®ETX as the super user (default login/password: super/super): 
3.	From Administration → Network Configuration, select IP Settings and configure the new IP: 

Connecting the R&S®DVM/ETX to an existing network

The following setup is relevant particularly when operating the R&S®DVM/ETX in the actual monitoring environment:

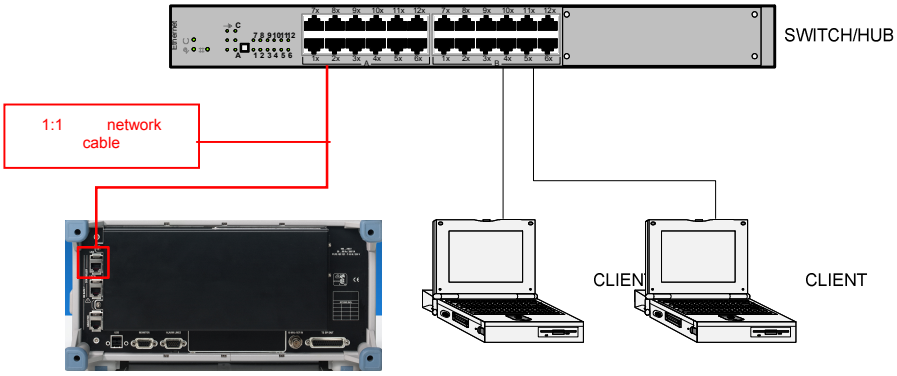


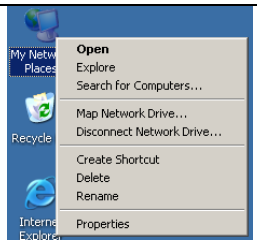
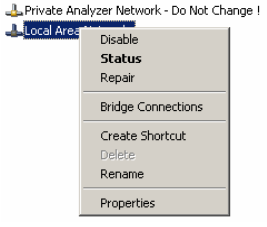
Fig. 15: Typical setup within a corporate network

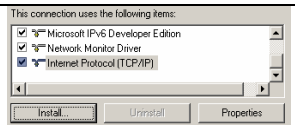
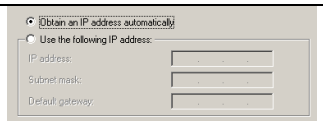
In contrast to a crossover network cable, the two leads of straight-through network cables have identical wire arrangements.

Dynamic configuration of the network address

A network configuration commonly used in companies is the dynamic allocation of IP addresses by dynamic host configuration protocol (DHCP) servers. Here, the configuration data (IP address, subnet mask) for the clients is automatically assigned.

To enable an R&S®DVM to support this dynamic configuration, the network adapter must be configured as follows:

1.	Using the right mouse button, click My Network Places and select Properties.	
2.	Right-click the required LAN adapter of the system. The properties of this network connection will be accessed. ***Note:*** *In case of the R&S®DVM, do not change the configuration of the Private Analyzer Network adapter.*	

3.	Access the properties of the Internet Protocol (TCP/IP) entry.	
4.	Enable the DHCP configuration by selecting "Obtain an IP address automatically".	
5.	To accept your settings, close all windows with OK.	-

In the case of the R&S®ETX, automatic configuration of the IP address via DHCP is not supported. Manual configuration as described in the preceding section is required.

6 Tools for Development and Working with SNMP

The MIB browser for everyday SNMP use

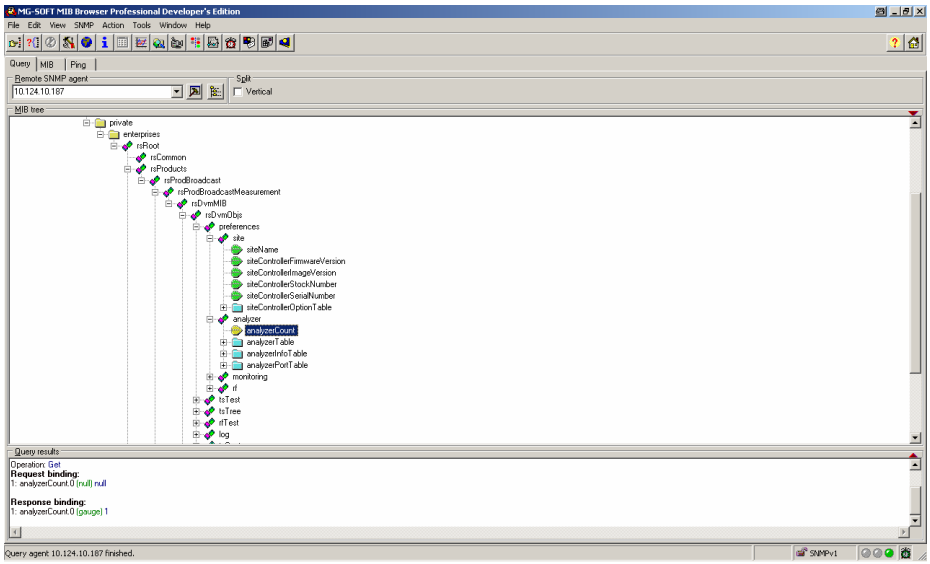


Fig. 16: MG-SOFT MIB Browser

To permit access to the various *managed objects* of an MIB managed in the SNMP agent and their instances, MIB browsers are available. These browsers communicate with the agent via the SNMP protocol, and they can read and write variables or they can receive traps by means of the SNMP functions presented above.

The market offers numerous MIB browsers as freeware or for a fee. The following examples are based on the experience Rohde & Schwarz accumulated using the MIB browser from the MG-SOFT company. The settings in interaction with the R&S®DVM, plus simple write and read accesses and the trap sink, are presented.

Note: *The procedures for SNMP interaction presented here are just simple examples. To read more about R&S®DVM examples such as how to output specific logs or TS states, refer to the manual.*

Procurement

The MG-SOFT MIB Browser Professional software can be obtained from the MG-SOFT website: http://www.mg-soft.com . Versions for both Windows and Linux are available. If you first want to try out the software, a trial version is also available from the website.

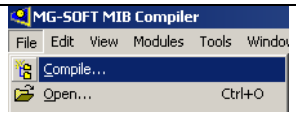
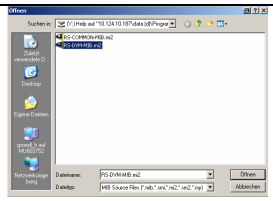
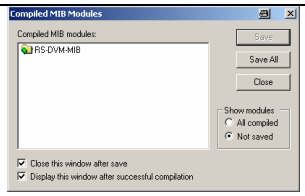
Basic operation using the R&S®DVM MIBs as an example

The operation of basic remote-control functions is shown below using the R&S®DVM MIBs. The procedure is similar for the R&S®ETX MIB.

Importing the R&S®DVM MIBs into the MIB browser

To make it possible to use the specific MIBs of a wide variety of manufacturers in the MG-SOFT MIB Browser, the MIBs are prepared using the MG-SOFT MIB Compiler.

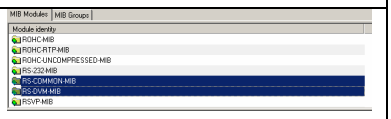
The procedure is as follows:

1.	Call the MIB compiler application from the Start menu or program directory.	
2.	Select Compile from the File menu.	
3.	Select the file named RS-DVM-MIB.mi2.	
4.	Click Save All.	
5.	Save in the default subdirectory ..\SMIDB.	
6.	Repeat steps 3 through 5 for the file named RS-COMMON-MIB.mi2 and then exit the application.	-

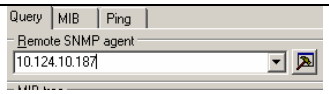
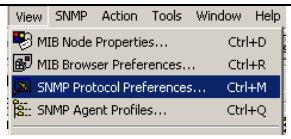
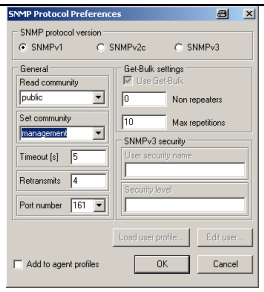
Initial steps with the MG-SOFT MIB Browser

This section shows you how the basic settings of the MIB browser for accessing the R&S®DVM must appear.

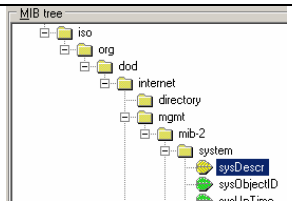
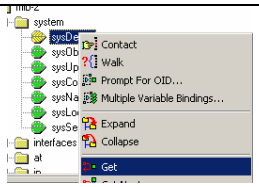
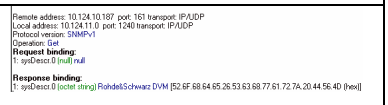
I. Load the compiled MIBs into the MIB browser

1.	Call the MIB browser application from the Start menu or program directory.	
2.	Select the MIB tab.	
3.	Select the two previously compiled MIBs.	
4.	Move the two R&S®DVM MIBs to the "Loaded MIB modules" section.	

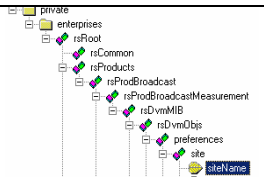
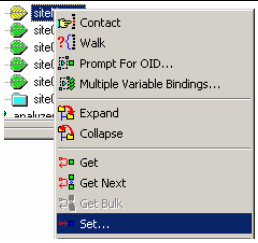
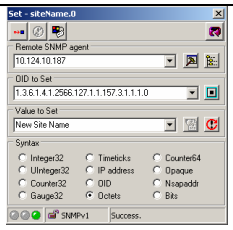
II. Set the IP and community strings / first communications test

1.	Enter the IP address of the R&S®DVM to be addressed.	
2.	Select SNMP Protocol Preferences from the View menu.	
3.	Enter the community strings configured on the R&S®DVM in the marked area at the right. After you confirm with OK, the browser will attempt to read the object named sysUptime from the agent of the R&S®DVM as a communications test.	
4.	If contact with the instrument is successful, the "Query results" window will display the following message.	

Application example: Outputting the system description

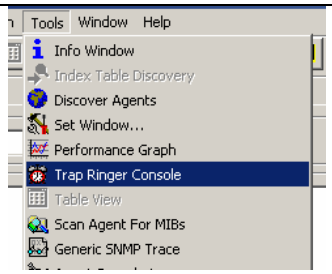
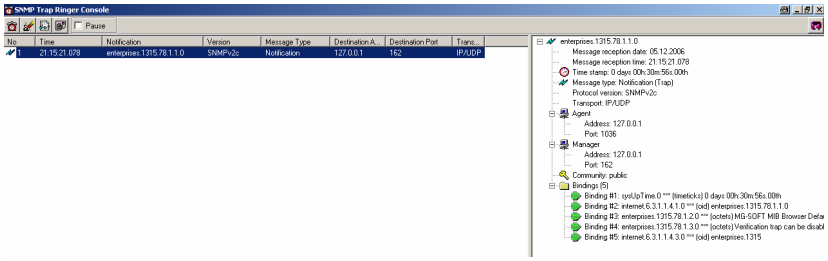
1.	Select sysDescr from the MIB tree. ***Note:*** *In order to read the sysDescr object, ensure that the RFC1213-MIB is loaded.*	
2.	Click sysDescr using the right mouse button, and then select the SNMP Get function from the menu.	
3.	The system description will now be output in the "Query results" window.	

Application example: Changing the site name of the R&S®DVM

1.	Select siteName from the MIB tree.	
2.	Click siteName using the right mouse button, and then select the SNMP Set function from the menu.	
3.	Enter the new site name in the "Value to Set" entry field. Now send ( ).	
4.	After the information is written successfully, the "Query results" window will display the current site name.	

Application example: Receiving traps

The Trap Ringer Console function in the MG-SOFT MIB Browser enables you to receive SNMP traps and display them.

1.	Select Trap Ringer Console from the Tools menu of the MIB browser.	
2.	Every received trap will be displayed in this window together with its specific parameters in the list. You can analyze the trap message further by clicking such an entry. 	

The network sniffer for detailed protocol analysis: Ethereal

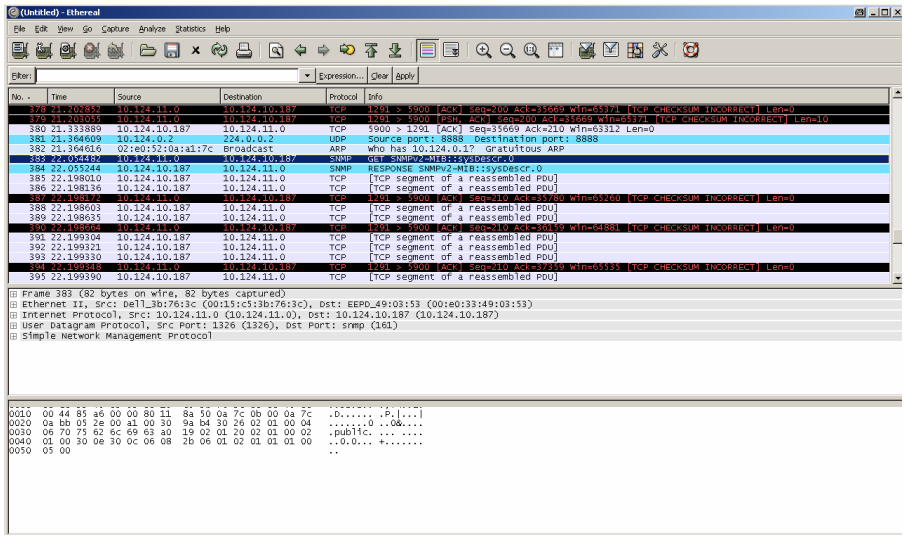


Fig. 17: Ethereal

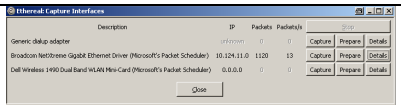
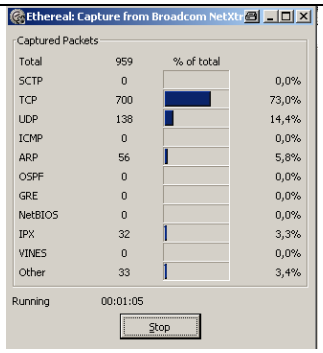
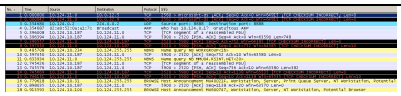
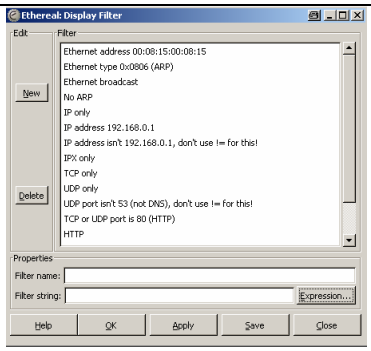
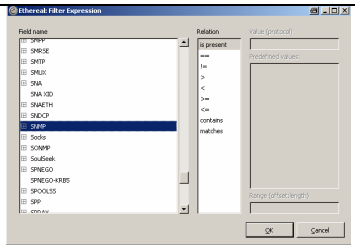
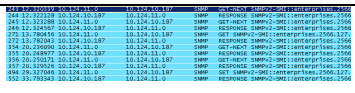
Ethereal enables you to record data as it is being transmitted across a network interface. The exact and manual analysis of the transmitted and received data is often indispensable, particularly in the development of protocol-handling software components.

Ethereal enables you to display precisely the data packets you want by using a wide selection of filter criteria. For example, filter criteria are available for the SMTP, SNMP, and FTP protocols, plus many others.

Procurement

The Ethereal software is available as freeware from http://www.ethereal.com/. Implementations/installations are offered for the numerous computer platforms in use.

Application example: Monitoring of SNMP traffic on the Ethernet interface of the internal network card

1.	Select the button labeled "List the available capture interfaces".	
2.	Select Capture for the capture interface you want to use.	
3.	After the capture has run as long as you want, stop it with Stop.	
4.	The log window will now display the recorded protocol packets.	
5.	To enable the selection of corresponding filters, click the Filter button.	
6.	Assign a filter name and click Expression in order to select the corresponding filter string.	
7.	Select SNMP from the menu that appears. Now close the window.	
8.	The log window will now display only the SNMP packets.	

7 Trap Configuration in the R&S®DVM Family

Configuration of the trap receiver (target)

The target for trap messages must be configured manually using a text editor. To do this, open the following file:

D:\USR\SNMP\PERSIST\SNMP.CONF

```
###############################################################################################################
# trapsink configuration
# ----------------------------------------------------------------------------------------------------------
#my trap sink targets
#trapsink   89.10.69.28
#trap2sink  89.10.69.28
#informsink 89.10.69.28
```

Fig. 18: Configuration of the trap targets

To make it possible to transmit SNMP traps in the various SNMP protocol versions, three variables are available to which target addresses for SNMP messages can be assigned:

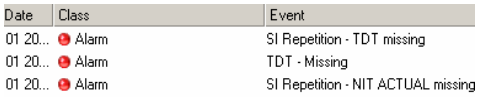
- trapsink: traps in line with SNMPv1
- trap2sink: traps in line with SNMPv2
- informsink: inform messages in line with SNMPv2

To enable the SNMP message you want, delete the pound sign (#) preceding the specific entry. The corresponding target IP will be appended to the individual variable (89.10.69.28 in the above example). Then restart the R&S®DVM.

Note: *If SNMP messages must be sent to more than one target, you must append the specific IP addresses to additional trapsink, trap2sink, or informsink entries line-by-line.*

Overview of the various trap types of the R&S®DVM

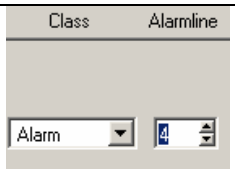
The R&S®DVM may only generate two different types of traps:

	rsDvmAlarmLineEvent	**rsDvmLogEvent**
Trap information	- Time stamp - Current state of all alarm lines at generation time	- Time stamp - Analyzer MAC address - Port number - Content of all columns of log line causing the trap
Event	Setting one of the alarm pins 	Entry in the monitoring log 

rsDvmAlarmLineEvent configuration

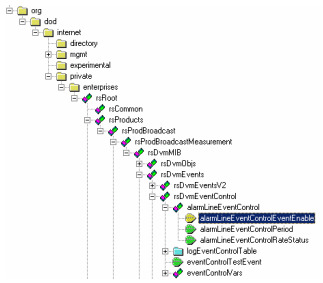
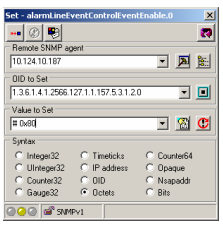
The source for rsDvmAlarmLineEvent is an active alarm line which could be assigned to each possible TS test. This can be done in two ways: by means of the R&S®DVM GUI or, alternatively, by means of SNMP (not shown in this document).

Assignment of alarm lines via the R&S®DVM GUI

1.	Select the TS monitoring GUI of the R&S®DVM by clicking the monitoring icon.	
2.	You can access the monitoring configuration via Config (in the lower right-hand corner).	
3.	Once there, enter the number of the relay contact you want in the Alarm Line column.	
4.	To accept the changes, select OK or Apply.	

Activating the Alarm Line event

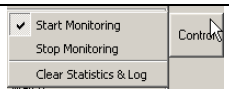
To activate the trap generation of an Alarm Line event on the R&S®DVM, you must configure a control variable (alarmLineEventControlEventEnable) on the instrument end via SNMP. You can do this as follows:

1.	Select the control variable labeled alarmLineEventControlEventEnable by means of an MIB browser: 
2.	Write-access the variable and transfer the string labeled **# 0x80**. 

rsDvmLogEvent configuration

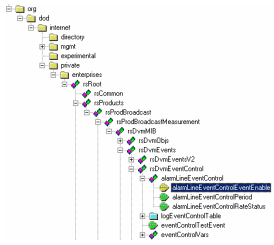
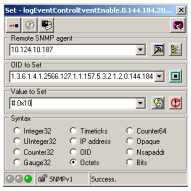
The source for rsDvmLogEvent is a new line appended to the monitoring log of the R&S®DVM. A new line is generated if the current value of an enabled TS test violates the limits or is caused by a system message. A log entry caused by a TS test can be classified as info, warning, or alarm with the object controlMonitoringConfigAlarmClass. A system message in the log is automatically classified as "system". Before generating traps, make sure that monitoring of the desired input has been activated. Again, you can activate it manually or via SNMP (not shown in this document).

Manual activation of TS monitoring

1.	Select the TS monitoring GUI of the R&S®DVM by clicking the monitoring icon.	
2.	You can enable TS monitoring by selecting the Control button (upper right-hand corner).	

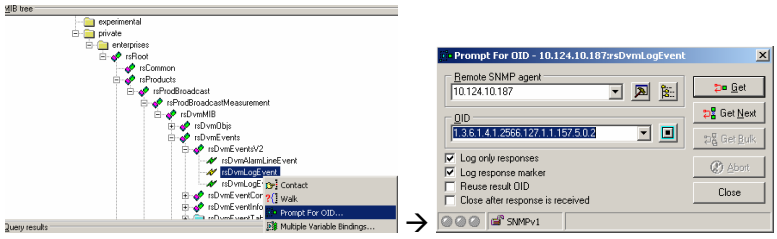
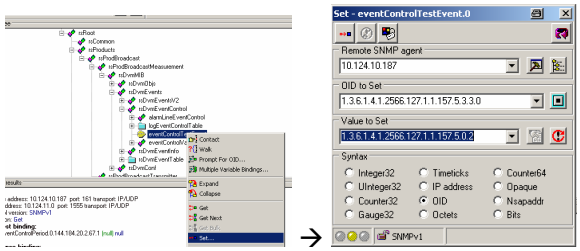
Activating the Log event

To activate the trap generation of a Log event on the R&S®DVM, you must configure a control variable (logEventControlEventEnable) on the instrument end via SNMP.

1.	Select the control variable labeled logEventControlEventEnable by means of an MIB browser: 
2.	The control variable represents a bit field. Write-access the variable and transfer the string you want.  logAlarmEventEnable (# 0x80), logWarningEventEnable (# 0x40), logInfoEventEnable (# 0x20), logSystemEventEnable (# 0x10), logEnableCompactReportEvent(# 0x01) You can enable several events simultaneously by using bit addition. Example: # 0x81 for logAlarmEventEnable and logEnableCompactReportEvent

The generation of test traps

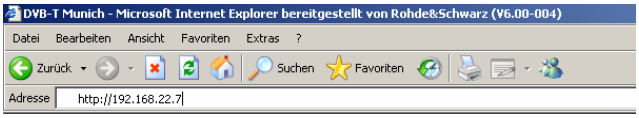
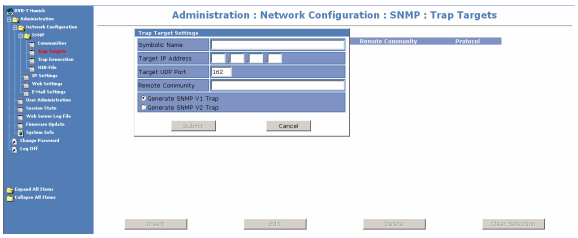
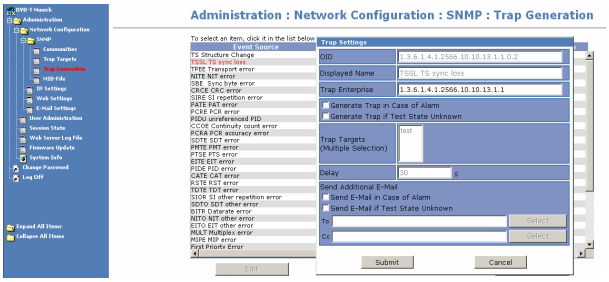
To provide support on the management end for the development of systems that evaluate received traps, the R&S®DVM MPEG-2 monitoring system has the capability to generate test traps. You can generate such a trap as follows:

1.	Select the control variable labeled rsDvmLogEvent or rsDvmAlarmLineEvent by means of an MIB browser, and output the OID: 
2.	Write-access the eventControlTestEvent variable, and transfer the previously determined OID: 
3.	A test trap has been generated.

For more information about R&S®DVM test traps, refer to the operating manual.

8 Generating Traps on the R&S®ETX

You can easily configure traps on the R&S®ETX by using the instrument's web GUI. Proceed as follows:

1.	Open a web browser and enter the IP address of the R&S®ETX: 
2.	Log on to the R&S®ETX as the super user (default login/password: super/super): 
3.	To configure the target of a trap message, go to Trap Targets under Administration → Network Configuration → SNMP. Here, you can define the IP data and other configuration data for sending the traps to specific targets. 
4.	Go to Trap Generation under Administration → Network Configuration → SNMP. You can now assign trap events to the specific MPEG-2 and RF alarm conditions.  To do this, click the events you want and select Edit. In the Trap Settings window, you can now select the two items labeled "Generate Trap in case of alert" and "Generate Trap if Test State unknown".
5.	After you complete the above steps, a trap will be generated when the corresponding event occurs.

9 The Development of SNMP Applications Made Easy

An integral part of this document is to provide a hands-on introduction to SNMP development. The objective is not to focus on SNMP programming at the lowest protocol layer, but, rather, to provide a solid basis for actual application development by means of *public domain* libraries provided as freeware on the Internet for the C#, C++, and Java programming languages. Thus, you can simply implement existing basic read and write accesses and reliable error handling routines.

Sections 10, 11, and 12 of this document provide a brief look at the development environment used, a recommendation for an SNMP stack, and the procedure, e.g. how to get an SNMP application up and running.

Note: The scope of this document does not allow for a discussion of the fundamentals of object-oriented development in the individual development languages and environments. The following explanations and sample programs are intended for experienced users.

The network protocol stacks mentioned in these sections are supplied separately in a ZIP file accompanying this Application Note. In addition, source code samples are supplied in subfolders. You may use them for your own development projects.

The ZIP file mentioned above is structured as follows:

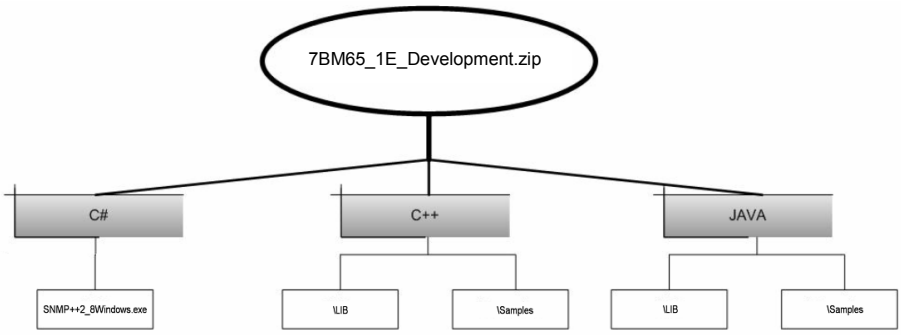


Fig. 19: Structure of the source code archive

After unpacking the file to any directory, you will see three main directories: C#, C++, and JAVA. Each of these directories comprises either two subdirectories named \LIB and \Samples or alternatively a self-extracting file.

10 Example of SNMP Implementation for C#

If you need to develop a C# application, various development tools such as the following are available:

- Microsoft Visual Studio .Net from Microsoft
- Visual C# Express from Microsoft
- C# Builder from Borland
- SharpDevelop (open source)
- MonoDevelop (open source)

The example of SNMP implementation for C# presented here is based on the Microsoft MS Visual Studio .NET 2005 development environment.

The solution for the example project described below is available in subdirectory C# of the separate source code archive. SNMP++.NET v. 1.21 is used as the SNMP stack for the .NET development:

The library consists of following five DLLs:

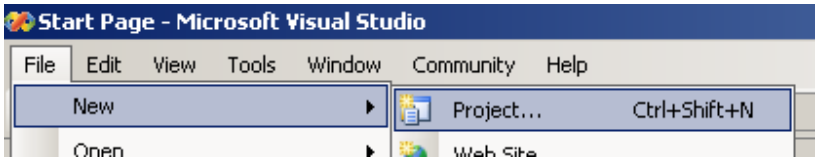
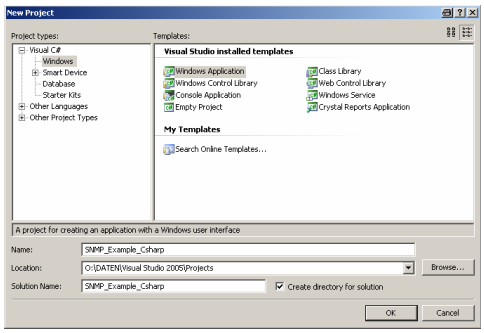
- Mib.Dll
- SnmpComp.dll
- SNMPDll.dll
- TableReader.dll
- Tools.dll

For further information about SNMP++.NET, see [7].

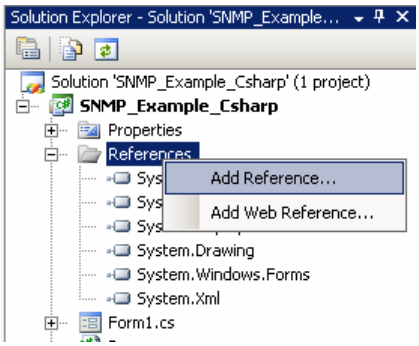
Procedure for implementing SNMP functions in a C# application

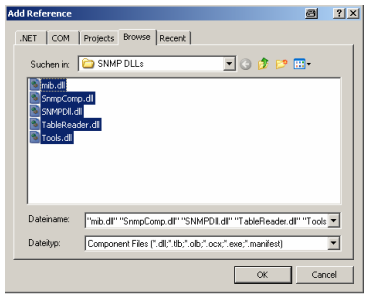
To output the system designation (sysDescr, OID: 1.3.6.1.2.1.1.1.0) of the R&S®DVM via a C# application, complete the following steps:

Create a new project

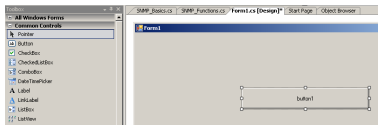
1.	Start the Visual Studio application and select New → Project from the File menu: 
2.	From the New → Project window, select Windows Application and assign a project name. In this case, use the following name: SNMP_Example_Csharp: 

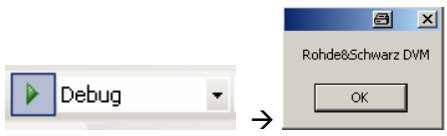
Add the SNMP library

1.	With your project's work area, you can now assign new references to external program libraries via the Solution Explorer in order to link the SNMP stack mentioned above: 

2.	Select the DLLs of the library: 

Your first executable program

1.	Add a button whose event outputs the message box with the system description:
	 
	→
	`using System.Windows.Forms;`
	`using SNMPDll;  //Including namespace from the SNMPDLL`
	
	`namespace SNMP_Example_Csharp`
	`{`
	`  public partial class Form1 : Form`
	`  {`
	`  public Form1()`
	`  {`
	`  InitializeComponent();`
	`  }`
	
	`  private void button1_Click(object sender, EventArgs e)`
	`  {`
	`  //Create an instance for the SNMP Agent on the DVM/ETX`
	`  SNMPAgent snmpAg = new SNMPAgent("10.124.10.187","public","management");`
	
	`  //Create an instance of the managed object sysDescr`
	`  SNMPObject snmpOb = new SNMPObject("1.3.6.1.2.1.1.1.0");`
	
	`  //Read the current value of the object sysDescr and output it via a MessageBox`
	`  string sysDescr = snmpOb.getSimpleValue(snmpAg);`
	`  MessageBox.Show(sysDescr);`
	`  }`
	`  }`
	`}`

2.	You can now run the application: 

11 Example of SNMP Implementation for Visual C++ 6

If you need to develop a C# application, various development tools such as the following are available:

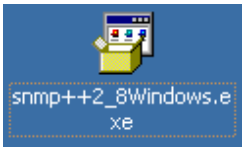
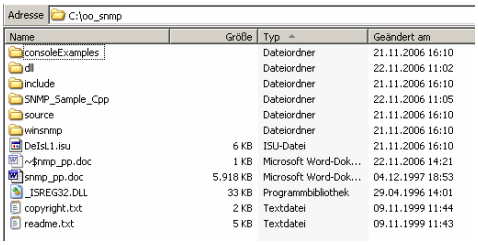
- MS Visual C++
- g++ (open source)
- Intel C++ Compiler
- Borland C++ Builder
- Comeau's C++ Compiler

An example of how to implement SNMP functionality in C++ is provided below in the Visual C++ 6 development environment.

If you want to implement SNMP functions in your C++ application, we recommend using the HP SNMP++ library. You can find this library in the C++ subdirectory of the separately downloadable source code archive.

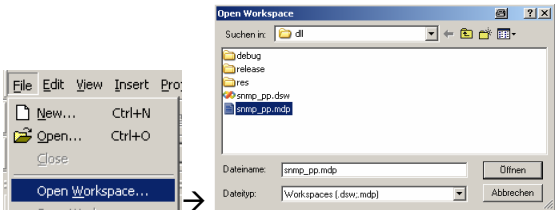
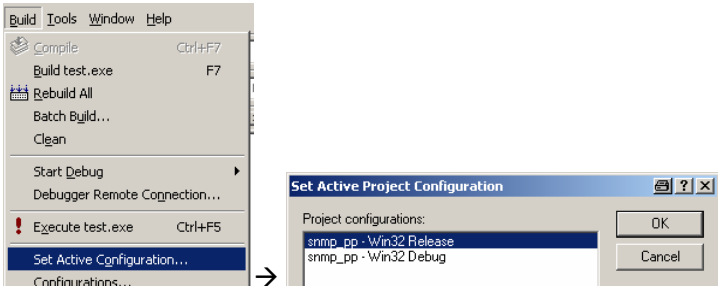
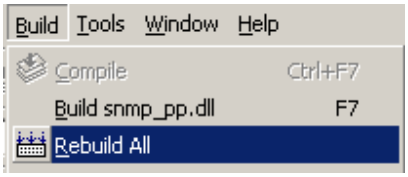
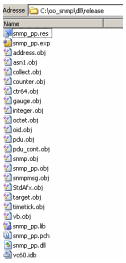
To link the HP SNMP++ library to your application, do the following:

Unpack the HP SNMP++ library

1.	Unpack the library archive: 
2.	After you run the setup routine, the source code and description of the SNMP library will be located in the c:\oo_snmp directory by default: 

Generate the DLL and LIB files

To use the SNMP++ library in your own applications, you must first generate the dynamic linked library (DLL) and its associated import libraries (LIB). The project files found in the c:\oo_snmp directory make this possible. Open them and run target in release mode:

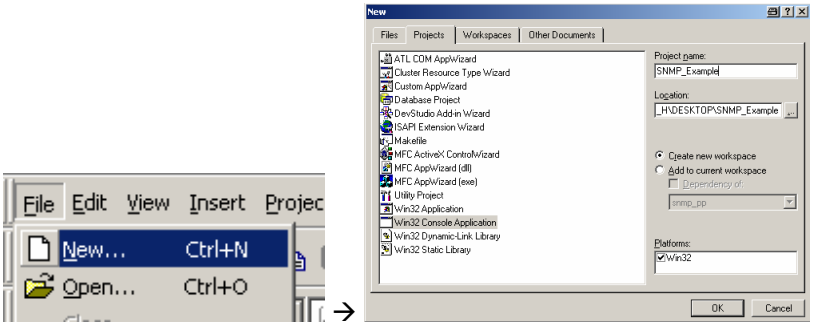
1.	Start the Microsoft Visual C++ 6.0 development environment: 
2.	Open the snmp_pp.mdp workspace: 
3.	Configure the Release mode for the project under Build → Set Active Configuration: 
4.	Generate the project: 
5.	The Release directory contains various files, including snmp_pp.dll and snmp_pp.lib: 

Link the library to your current Visual C++ 6.0 project

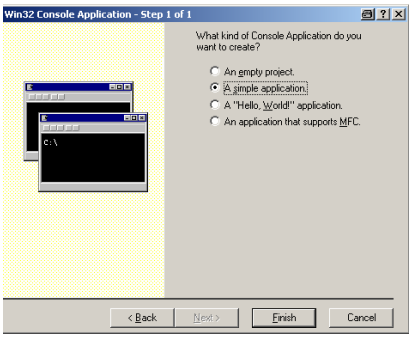
To see how this is done, you must first create a new C++ console application and then add the SNMP++ library to it.

Create a console application

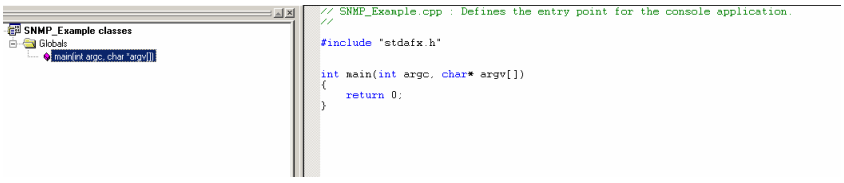
1. Start the Microsoft Visual C++ 6.0 development environment. Specify "Win32 Console Application" as the project. Define the storage location, and assign the specific project name. In this case: SNMP_Example



2. Select "A simple application" and close the wizard:



3. A simple C++ console application is now available:



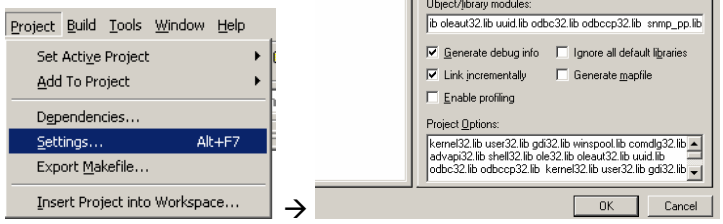
Copy the required library files to your current project folder

Copy the following files of the c:\oo_snmp directory to the specified locations of your current project folder:

c:\oo_snmp	current project directory
\dll\release\snmp_pp.lib	\snmp_pp.lib
\dll\release\snmp_pp.dll	\release\snmp_pp.dll
\include*.*	\include*.*

Now link the library to the existing project

1. Link the import library snmp_pp.lib to the linker process by selecting the Link tab under Project → Settings and appending the file name in "Object/library modules":



2. Now include the snmp_pp.h header file in your class:

```
// SNMP_Example.cpp : Defines the entry pc
//

#include "stdafx.h"
#include "Include\snmp_pp.h"


int main(int argc, char* argv[])
{
    return 0;
}
```

Example applications for SNMP and C++

For some example applications (such as read or write access to SNMP variables), take a look at the example applications of the SNMP++ library, located under \consoleExamples in the installation directory of the SNMP++ library.

12 Example of SNMP Implementation for Java

If you need to develop a Java application, various development tools such as the following are available:

- Eclipse (open source)
- Sun ONE Studio
- IntelliJ IDEA from JetBrains
- JBuilder from Borland

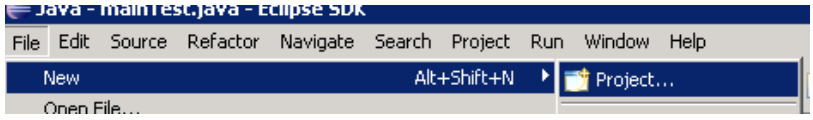
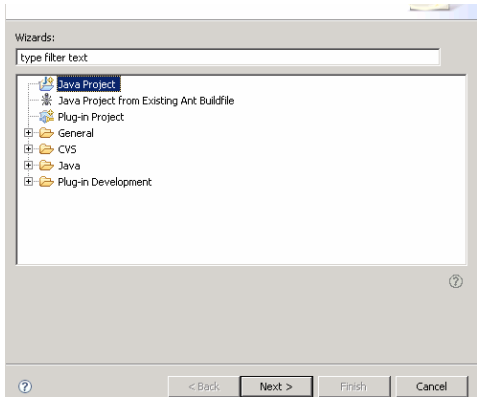
The example of SNMP implementation for Java presented below is shown using the Eclipse development environment, which is available as freeware.

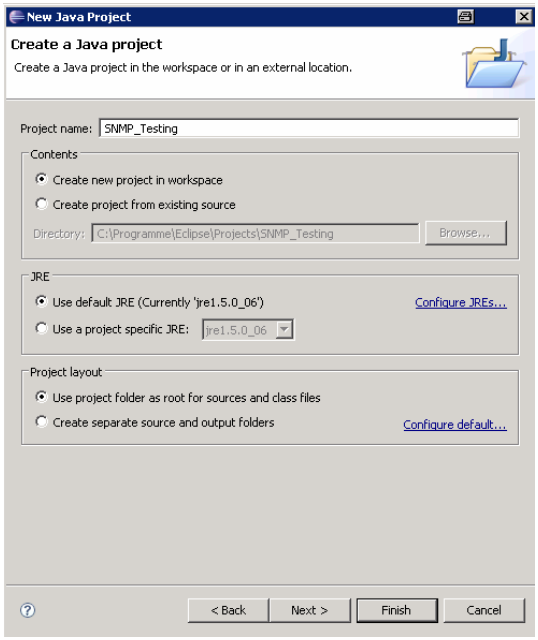
To use the SNMP functions in Java, we recommend the SNMP4J Java library. You can find the library in the \Java directory of the separately downloadable source archive.

Link the SNMP4J library to Eclipse

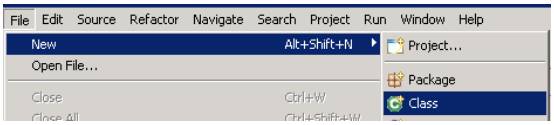
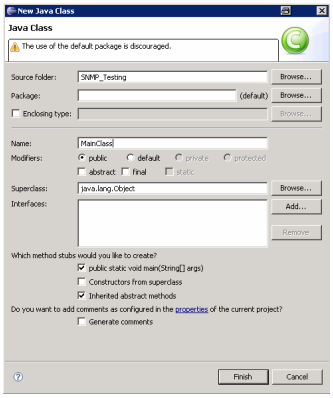
To see how this is done, you must first start a new Eclipse project. Then link a Java archive file (JAR) of the SNMP4J library to this project.

Create a new project

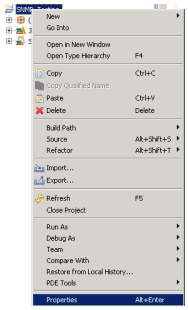
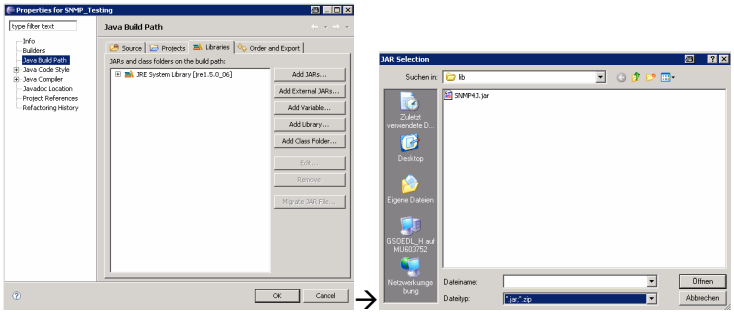
1.	After starting the Eclipse application, select New → Project from the File menu: 
2.	Select Java Project and go to the next window by selecting Next. 

3.	Assign a project name (in this case: SNMP_Test) and click Finish. 

Create a new class

1.	To create a new class, select New → Class from the File menu: 
2.	Specify a class name (here: MainClass) and enable the "public static void main()" option: 

Link the SNMP4J library

1.	In the Package Explorer, click the current project using the right mouse button, and select Properties: 
2.	In the Java Build Path selection, the SNMP4J.JAR file is linked by means of ADD External JARs: 

3.	By adding import instructions, you can link specific namespaces to the current class:
	`import org.snmp4j.*;`
	`import org.snmp4j.event.*;`
	`import org.snmp4j.transport.*;`
	`import org.snmp4j.smi.*;`
	
	`import java.io.*;`

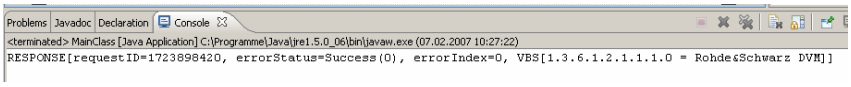
Create your SNMP Application

1. Now enter the actual source code for the SNMP application within the main() function.

```
public static void main(String[] args) throws IOException {
    //Define community target
    CommunityTarget target = new CommunityTarget();
    Address targetaddress = new UdpAddress("10.124.10.187/161");
    target.setAddress(targetaddress);
    target.setCommunity(new OctetString("public"));
    target.setTimeout(1000);
    target.setVersion(0);
    target.setRetries(1);

    //Define SNMP request
    PDU pdu = new PDU();
    pdu.setType(PDU.GET);
    pdu.add(new VariableBinding(new OID("1.3.6.1.2.1.1.1.0")));

    //Define transport mapping
    TransportMapping transport = new DefaultUdpTransportMapping();
    transport.listen();

    //The actual snmp request
    try
    {
        Snmp snmp = new Snmp(transport);
        ResponseEvent respEv = snmp.send(pdu, target);
        PDU response = respEv.getResponse();
        System.out.println(response);
    }
    catch (Exception e)
    {
        System.out.println("Error in SNMP communication");
    }
}
```

2. Execute the application:

After the communication has been executed successfully, the system designation shows up in the console:



13 References

Books:

[1] Fischer, Walter (2005). Digital Television. A Practical Guide for Engineers. Berlin: Springer.

[2] Liberty, MacDonald (2006). Learning C# 2005: Get Started with C# 2.0 and .NET Programming (2nd Edition). Sebastopol: O'Reilly Media.

[3] Stephen Prata (2004). C++ Primer Plus (5th Edition). Indianapolis: Sams.

[4] Liang, Y. D. (2006). Introduction to Java Programming-Comprehensive Version (6th Edition). Upper Saddle River: Prentice Hall.

Manuals:

[5] Rohde & Schwarz (Ed.) (2006). R&S®ETX DTV Monitoring Receiver operating manual, 2068.0909.12 – 02. Munich: Rohde & Schwarz.

[6] Rohde & Schwarz (Ed.) (2006). R&S®DVM100/120 MPEG-2 Monitoring System, 2085.1639.12-05. Munich: Rohde & Schwarz.

Internet Sources:

[7] SNMP library for .NET:

http://www.codeproject.com/useritems/SNMPDLL.asp

[8] SNMP++ library for C++:

http://www.agentpp.com/

[9] SNMP library for Java:

http://www.snmp4j.org/

14 Additional Information

Our Application Notes are regularly revised and updated. Check for any changes at http://www.rohde-schwarz.com.

Please send any comments or suggestions about this Application Note to Broadcasting-TM-Applications@rsd.rohde-schwarz.com

15 Ordering Information

R&S®DVM family

Option		Number
DVM100	Description	2085.1600.03
DVM100L	MPEG2 Monitoring System	2112.7050.02
DVM120	MPEG2 Monitoring System	2085.1700.03
DVM-B1	MPEG Analysis Board	2085.3283.02
DVM-K1	TS-Monitoring	2085.5211.02
DVM-K2	TS-Capture	2085.5234.02
DVM-K10	In Depth Analysis	2085.5228.02
DVM-K11	Data Broadcast Analysis	2085.5311.02
DVM-K12	Template Monitoring	2085.5328.02
ZZA-111	Rack mount kit	1096.3254.00
DVM50	MPEG2 Monitoring System	2085.1900.03
DVM50-K10	In Depth Analysis	2085.5434.02
DVM400	Digital Video Measurement System	2085.1800.03
DVM400-B1	MPEG Analysis Board	2085.5505.02
DVM400-B2	TS Generator	2085.5511.02
DVM400-B3	Upgrade TS Generator TRP Recorder/Player	2085.5528.03
DVM400-B4	Upgrade TS Generator TRP Recorder/Player (214MBIT/S)	2085.5534.03
IP		
DVM400-B40	Gigabit Ethernet interface module	2085.5557.02
Decoder		
DVM-B30	Video and audio hardware decoder	2085.5570.02
DVM400-B30	Video and audio hardware decoder	2085.5540.02
DVM-K30	HD/SD-SDI output	2085.5440.02
DVM-K31	Video and audio monitoring	2085.5457.02
DVM-K32	HDTV decoding upgrade	2085.5486.02
RF		
DVM-B50	DVB-C, J83.A/C Receiver Module	2085.5605.02
DVM-B51	DVB-S/DVB-S2 Receiver Module	2085.5611.02
DVM-B52	DVB-T/H Receiver Module	2085.5628.02
DVM-K52	Second DVB-T/H receiver path	2085.5470.02
DVM-B500	RF carrier board	2085.5634.02
DVM-B520	RF carrier board	2085.5640.02
DVM400-B500	RF carrier board and decoder extension	2085.5563.02
Streams		
DV-HDTV	HDTV Sequences	2085.7650.02
DV-DVBH	DVB-H Stream Library	2085.8704.02
DV-H264	H.264 Stream Library	2085.9052.02
DV-TCM	Test Card M Streams	2085.7708.02
DV-ASC	Advanced Stream Combiner	2085.8804.02/03

R&S®ETX-T

Option		Number
ETX-T	DTV Monitoring Receiver	2068.0109.40
ETX-B2	MPEG2 Real Time Analysis w/o Decoder Output	2068.0415.02
ETX-B3	MPEG2 Real Time Analysis w/ Decoder Output	2068.0450.02
ETX-B11	6MHZ-Saw-Filter	2068.0421.02
ETX-B12	7MHZ-Saw-Filter	2068.0438.02
ETX-B13	8MHZ-Saw-Filter	2068.0444.02
ETX-K10	SFN Option	2068.0480.02
ETX-DCV	Documentation of ETX	2082.0490.28

ROHDE & SCHWARZ GmbH & Co. KG · Mühldorfstraße 15 · D-81671 München · Postfach 80 14 69 · D-81614 München ·
Tel (089) 4129 -0 · Fax (089) 4129 - 13777 · Internet: http://www.rohde-schwarz.com