LabVIEW™ SNMP Programming
for the R&S® DVM, R&S® DVQ and R&S® ETX-T
Made Easy With the Viodia™ SNMP Toolkit

Application Note

In the T&M world, instrument remote control over SNMP (simple network management protocol) lives in the shadow of IEEE/GPIB (general purpose interface bus) based remote control, and only a handful of the T&M equipment available worldwide supports the protocol. It is therefore no surprise that for most T&M engineers familiar with text-based commands over IEEE/GPIB (and lately LAN), the SNMP protocol seems a bit odd and to some even complex at first. The SNMP Toolkit for LabVIEW™ from Viodia™ Inc. can save a considerable amount of development time and furthermore helps programmers to overcome the initial complexities of SNMP programming.
LabVIEW™ SNMP programming for the R&S® DVM, DVQ and ETX-T made easy with the Viodia™ SNMP Toolkit

Contents

1 Introduction ....................................................................................................................... 3
2 SNMP Terminology and Concepts .................................................................................. 3
3 Programming Examples .................................................................................................. 4

3.1 BasicVIs ....................................................................................................................... 4
3.1.1 General ..................................................................................................................... 4
3.1.2 SNMPset-request.vi ................................................................................................. 4
3.1.2.1 Parameter Description ....................................................................................... 4
3.1.2.2 Sample Program ............................................................................................... 8
3.1.3 SNMPget-request .................................................................................................... 9
3.1.3.1 Parameter Description ....................................................................................... 9
3.1.3.2 Sample Program ............................................................................................... 9
3.1.4 SNMPget-next-request ......................................................................................... 11
3.1.4.1 Parameter Description ....................................................................................... 11
3.1.5 Look-Up OIDs, Names ........................................................................................... 12
3.1.5.1 Parameter Description ....................................................................................... 12
3.1.5.2 Sample Code ..................................................................................................... 15

4 Appendix: OID Formation of the Most Common R&S® DVM100/400 Variable Types.......................................................................................................................... 16
4.1 Simple Variables ........................................................................................................ 16
4.2 Monitoring Configuration Variables ......................................................................... 17
4.3 Log Entries ................................................................................................................. 22

5 Summary ........................................................................................................................ 24
6 Literature ....................................................................................................................... 24
7 Additional Information .................................................................................................. 24
8 Ordering Information ..................................................................................................... 25

The R&S logo, Rohde & Schwarz and R&S are registered trademarks of Rohde & Schwarz GmbH & Co. KG and their subsidiaries.
LabVIEW™ SNMP programming for the R&S® DVM, DVQ and ETX-T
made easy with the Viodia™ SNMP Toolkit

1 Introduction

The simple network management protocol found a loyal group of users in the broadcasting industry. In this world, the border between network operation and T&M equipment is often vague and it is therefore no surprise that broadcasting T&M instruments are often fitted with an SNMP-based remote control interface. Products that fall into this category are the R&S®DVQ/DVQM digital video quality analyzers, R&S®DVM100/400 video measurement systems and R&S®ETX-T digital TV monitoring receivers.

The GPIB/IEEE interface commonly used for remote control of T&M instruments has been around for a few decades now, and most T&M engineers are familiar with it. Unfortunately this cannot be said of SNMP. The unfamiliar, object-based nature of this protocol, together with the numeric variable addressing format, induce a certain degree of hesitance to use SNMP when confronted with it for the first time.

Contrary to the GPIB/IEEE protocol, which works with straightforward, text-based commands, commonly known as standard commands for programmable instrumentation or SCPI, SNMP uses an object-based approach with numeric identifiers to address variables. It also requires a certain familiarity with LAN terminology like UDP and IP.

To shorten the learning curve and simplify SNMP programming, Viodia™ Inc. developed a LabVIEW™ Virtual Instrument library named SNMP Toolkit that largely permits the T&M engineer to employ a black box approach, eliminating the need for a thorough understanding of LAN technology. In this application note we explain how to use the SNMP Toolkit for remote control of the R&S®DVM100/400 video measurement systems.

2 SNMP Terminology and Concepts

An elementary understanding of SNMP, management information bases (MIBs), and object identifiers (OIDs) is required throughout this document. This application note does not have the intention to be yet another tutorial on SNMP programming, therefore for the newcomers to SNMP we would like to recommend:

• "The Cuddletech Guide to SNMP Programming” by Ben Rockwood.

This document explains everything a broadcasting or application engineer needs to know about OIDs, MIBs, and SNMP in general, without any unnecessary baggage. Some basics can also be found in the appendix as well as in:


• Application note 7BM66_1E “SNMP Example: DVM Management Center Monitoring in a Broadcast Network” by Harald Gsoedl.

This document also requires reader familiarity with LabVIEW™. Tutorials for this programming environment and sample code are available from the manufacturer National Instruments™.
3 Programming Examples

3.1 Basic VIs

3.1.1 General

The Viodia™ Inc. SNMP Toolkit contains five elementary Labview™ VIs which can be integrated into a larger VI in order to read and write data from and to the R&S®DVM100/400 or any other instrument which supports SNMP. These five elementary VIs are:

- **SNMPset-request**: Writes values to the R&S®DVM100/400 variables addressed by the passed OID.
- **SNMPget-request**: Returns the content of R&S®DVM100/400 variables addressed by the passed OID.
- **SNMPget-next-request**: Returns the contents of an R&S®DVM100/400 variable that's hierarchically next in line with the one previously addressed. Every time SNMPget-next-request is called the subsequent one is addressed.
- **SNMPtrap**: Reads the contents of a trap.
- **Look-up OID, Names**: Converts variable names defined inside MIBs located in the passed folder to OIDs and visa versa.

The SNMP Toolkit has additional VIs, like SNMPget-response, which are to be used in agent programming. Since the R&S®DVM100/400 generally acts as an agent and your LabVIEW™ program as a manager the use of these VIs is generally not required.

3.1.2 SNMPset-request.vi

3.1.2.1 Parameter Description

In the example at the end of this paragraph we make use of **SNMPset-request** to set the active Site Name in the R&S®DVM100/400. SNPset-request has 8 parameters as depicted below.
LabVIEW™ SNMP programming for the R&S® DVM, DVQ and ETX-T made easy with the Viodia™ SNMP Toolkit

Fig. 1 SNMPset-request parameters

- **Community**: Communities can be thought of as user access levels. The R&S® DVM100/400 recognizes two communities, “public” and “management”. Variables inside the R&S® DVM100/400 which have write access belong to the “management” community. Therefore, pass a text string containing “management” to Community. The community names “public” and “management” are stored in the text file `snmpd.conf` located in D:\usr\snmp\persist and can be changed if required.

Fig. 2 Assigned community names in R&S® DVM’s snmpd.conf file
LabVIEW™ SNMP programming for the R&S® DVM, DVQ and ETX-T made easy with the Viodia™ SNMP Toolkit

- **IP Address**: Note down the IP address setting in the Network Connections of the Windows OS on your R&S®DVM100/400 and pass this as a text string to the IP Address parameter.

![Image of IP address settings](image)

*Fig. 3 IP address settings*

The instrument has more than one LAN adapter. Make sure this is the IP address setting of the main network adapter instead of the local adapter used to communicate with the analyzer boards and R&S®DVM120 extension units.
LabVIEW™ SNMP programming for the R&S® DVM, DVQ and ETX-T made easy with the Viodia™ SNMP Toolkit

- **MIB variable values**: SNMP set-request can write multiple variables in one single SNMP set-request call. To support this, the MIB variable values parameter represents an array of clusters. One single cluster contains the:
  
  - **MibVariable**: OID of the variable you want to write to; e.g., “1.3.6.1.4.1.2566.127.1.1.157.3.1.1.0.” for the *Sitename* in the instrument. *LabVIEW™* type: string. This parameter also accepts the variable name associated with the OID when Look-up OIDs, Names has successfully initialized a resident MIB. See Look-up OIDs, Names.
  
  - **Syntax**: Variable type. Pass the enum from the table below that corresponds to the respective variable type. *LabVIEW™* type: U16 enum.

<table>
<thead>
<tr>
<th>Items</th>
<th>Digital Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTEGER</td>
<td>0</td>
</tr>
<tr>
<td>OCTET STRING</td>
<td>1</td>
</tr>
<tr>
<td>OBJECT IDENTIFIER</td>
<td>2</td>
</tr>
<tr>
<td>IpAddress</td>
<td>3</td>
</tr>
<tr>
<td>Counter32</td>
<td>4</td>
</tr>
<tr>
<td>Counter64</td>
<td>5</td>
</tr>
<tr>
<td>Gauge32</td>
<td>6</td>
</tr>
<tr>
<td>TimeTicks</td>
<td>7</td>
</tr>
<tr>
<td>opaque</td>
<td>8</td>
</tr>
<tr>
<td>NULL</td>
<td>9</td>
</tr>
<tr>
<td>noSuchObject exception</td>
<td>10</td>
</tr>
<tr>
<td>noSuchInstance exception</td>
<td>11</td>
</tr>
<tr>
<td>endOfMibView exception</td>
<td>12</td>
</tr>
</tbody>
</table>

*Fig. 4 Enumerations of different SNMP variable types*

- **Variable value**: The variable value is passed as a string irrespective of the actual type addressed by the respective OID.

The full variable type is depicted in figure 4. Remember that the variable must be an array type. Single clusters are not accepted. To write single variables, fill out array element 0 only.

*Fig. 5 Array of MIB variable values cluster*

- **Time-out**: Time-out value in milliseconds. SNMP set-request returns with error code 56 in Error-out when the operation could
LabVIEW™ SNMP programming for the R&S® DVM, DVQ and ETX-T made easy with the Viodia™ SNMP Toolkit

not be completed within the specified time. The default value is 3 seconds. Type: I32.

- **Version**: The R&S®DVM100/400 supports SNMP version 1 and 2. Set this parameter to 1 (Version 2). Type: U16.
- **Error-in**: Connect this terminal to the error output of the preceding component. If SNMPset-request is the first one, connect it to a standard LabVIEW™ Error-in cluster which is set to zero (No error).

![Error-in and Error-out clusters](image)

- **Error-out**: Contains the error code when SNMPset-request reports an error. Pass this output to any subsequent functions, or if SNMPset-request is the last call, connect it to an Error-out cluster.

### 3.1.2.2 Sample Program

The sample code below sets the current site name at "My Site" (OID = 1.3.6.1.4.1.2566.127.1.1.157.3.1.1.1.0.) on an R&S®DVM100/400 with IP address 168.198.30.27. The time-out value is 3000 milliseconds.

![Sample program diagram](image)

*Fig. 6 SNMPset-request sample code*
3.1.3 SNMPget-request

3.1.3.1 Parameter Description

The parameters of SNMPget-request are almost identical to the ones of SNMPset-request. The difference lies in the passing of the MIB variables parameter.

- **MIB variables input**: SNMPget-request only needs to know the OID(s) for a read operation. This parameter is therefore passed as an array of OIDs or variable name strings instead of cluster types containing OID, type, and value.

- **MIB variables output**: Identical to SNMPset-request. The returned values and types are passed back to the second and third fields of the cluster (Syntax and Value) of the MIB variables output parameter after successful completion of the operation.

- When reading the variable content with SNMPget-request, like in the example on the next page, use “Public”. Read-only variables are members of the “Public” community.

![SNMPget-request parameters](image)

3.1.3.2 Sample Program

The sample code below reads:

- The **Upper period** limit setting of the PAT repetition in a Monitoring Configuration named “Config (DVB)”.

- The **Loss after packets** limit of the TS Synchronization setting in the “Config (ATSC)”. 
If Monitoring Configurations with such names are available in the instrument, the MIB Variables output parameter contains the requested values. Refer to the appendix to understand how the object identifiers 1.3.6.1.4.1.2566.127.1.1.157.3.1.3.1.3.67.111.110.102.105.103.32.40.68.86.66.41.1301 and 1.3.6.1.4.1.2566.127.1.1.157.3.1.3.1.3.67.111.110.102.105.103.32.40.65.84.83.67.41.1101 for these variables are formed.

Fig. 8 SNMPget-request sample
3.1.4 SNMPget-next-request

3.1.4.1 Parameter Description

The number, types, and function of the parameters of SNMPget-next-request are identical to the ones of SNMPget-request. SNMPget-next-request returns the contents of an R&S®DVM100/400 variable that's hierarchically next in line with the one previously addressed. Every time SNMPget-next-request is called the subsequent one is addressed.
3.1.5 Look-Up OIDs, Names

3.1.5.1 Parameter Description

Look-up OIDs, Names converts the variable names passed via the MIB variables values input to OIDs. This allows the programmer to make use of variable names, which are easier to handle and far more readable than OIDs. SNMPget-request, SNMPget-next-request, and SNMPset-request call Look-up OIDs, Names internally. This means that there’s no need for an additional external call anymore. It is however useful to understand its parameters and operation.

Look-up OIDs, Names makes use of the UC Davis Open Source MIB compiler to convert a variable name to an OID. This compiler comes in the form of DLLs included in the SNMP Toolkit. It therefore must have access to the RS-DVM-MIB.mi2 and RS-COMMON-MIB.mi2 files on the R&S® DVM100/400. These files contain all the instrument’s variables and
corresponding OIDs. Copy these files onto the computer that runs your LabVIEW™ program (manager) and rename them to RS-DVM-MIB.mib and RS-COMMON-MIB.mib. Renaming is necessary since the MIB compiler only recognizes MIBs with a *.MIB extension.

Variable names are never carried over the SNMP protocol. If the compiler for whatever reason is unable to perform a successful conversion, no OID is available at the output. If an OID is passed to Look-up OIDs, Names, no conversion takes place and the OID is passed directly to the ObjectIdentifiers output.

- **Mode**: Determines operational mode of Look-up OIDs, Names. There are 3 modes:
  - **1) Init**: Initializes the UC Davis Open Source MIB compiler. Look-up OIDs, Names must be called at least once in this mode to compile the MIBs present in the MIBDirectoryPath prior to use in any of the two other modes.
  - **2) Name to OID**: Converts the names in the MIB variables input to OIDs.
  - **3) OID to Name**: Converts OIDs to the corresponding names in the MIB variables.

- **MIBDirectoryPath**: Pass the path variable of the folder in which you have copied DVM-MIB.mib and RS-COMMON-MIB.mib. This folder must also contain the SNMPv2-SMI.mib, SNMPv2-TC.mib, and SNMPv2-TM.mib SNMP foundation included in the SNMP Toolkit.

- **MIB variables**: This parameter is a string array to support conversion of multiple variables in one single call.

- **ObjectIdentifiers**: Only applies to OID to Name mode. Array of strings containing the OIDs after successful conversion of the variable names. ObjectIdentifiers is an array of simple strings and can’t be wired directly to the MIB variable input of SNMPget-request and other VIs. Use a conversion method like the following one to convert the simple array type to an array of clusters.

\[
\text{Fig. 12 MIB variable conversion}
\]

- **Names**: Array containing the names of the OIDs after successful Name to OID conversion.
- **Error in**: Connect this terminal to the error output of the preceding circuit. If **Look-up OIDs, Names** is the first one, connect it to the **Error-in cluster** which is set to zero (No error).

```
+---+   +---+
|   |   |   |
|   |   |   |
|   |   |   |
+---+   +---+
```

- **Error out**: Contains the error code and index when **Look-up OIDs, Names** reports an error. Pass this output to any subsequent functions.
3.1.5.2 Sample Code

The LabVIEW™ sample code below converts the *Sitename* and *Analyzer Serial Number* (declared as “siteName” and “analyzerSerialNumber” inside the *R&S®DVM100/400 MIB DVM-MIB.mib*) to their respective OIDs.

Fig. 13 LabVIEW™ code to convert variable name to OID
Appendix: OID Formation of the Most Common
R&S® DVM100/400 Variable Types

4.1 Simple Variables

Simple variables like the `Sitename` in the R&S® DVM100/400 follow the general SNMP OID principles; i.e., the full OID is formed by appending the OIDs of the individual objects in the instrument’s variable structure separated by periods.

Figure 14 The instrument’s current site name “Rohde & Schwarz”

Example: Fig. 15 shows the hierarchical variable structure of the `Sitename`. `Sitename` is a member of `site`, `preferences`, `rsDvmObjs`, `rsDvmMIB`, ....

The individual OIDs of the objects are given in table 1 and can be found in the MIBs `RS-DVM-MIB.mi2` and `RS-COMMON-MIB.mi2`, both of which are stored in the instrument. The contents can be observed with a MIB browser or plaintext editor.
Table 1 OID of “Sitename”

<table>
<thead>
<tr>
<th>Variable</th>
<th>OID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rohde &amp; Schwarz GmbH &amp; Co.KG</td>
<td>2566</td>
</tr>
<tr>
<td>rsRoot (Rohde &amp; Schwarz GmbH &amp; Co.KG)</td>
<td>2566</td>
</tr>
<tr>
<td>rsProducts</td>
<td>127</td>
</tr>
<tr>
<td>rsProdBroadcast</td>
<td>1</td>
</tr>
<tr>
<td>rsProdBroadcastMeasurement</td>
<td>1</td>
</tr>
<tr>
<td>rsDvmMIB</td>
<td>157</td>
</tr>
<tr>
<td>rsDvmObjs</td>
<td>3</td>
</tr>
<tr>
<td>Preferences</td>
<td>1</td>
</tr>
<tr>
<td>Site</td>
<td>1</td>
</tr>
<tr>
<td>siteName</td>
<td>1</td>
</tr>
</tbody>
</table>

To assemble the full OID to access the Sitename, append all OIDs, separated by periods, in hierarchical order and terminate the string with a 0 trailer “1.3.6.1.4.1.2566.127.1.1.157.3.1.1.1.0”. The final string can be passed to the MIB variables parameter of the respective VI as explained in the previous chapter. The full formation process is shown in figure 16.

4.2 Monitoring Configuration Variables

The OID formation of Monitoring Configuration variables is more complex than that of simple variables.

A Monitoring Configuration allows logging and reporting of certain MPEG transport stream parameters that go beyond a predefined limit. A Monitoring Configuration includes configuration of these predefined limits, classification of the seriousness, disabling/enabling of reporting, and more. Multiple Monitoring Configurations can be saved under different names and recalled at a later stage.

- Manual control: From a user interface point of view, the parameters that are configurable in a Monitoring Configuration can be seen as a parameter or property of the actual MPEG transport stream parameter to be monitored. For example, the maximum limit or Upper Period, Class, etc., of
the **PAT** (Program Association table), are all grouped under the same header **PAT** as shown in the figure below.

![Fig. 17 R&S® DVM100/400 monitoring parameters settings from a user interface point of view](image)

- **Remote control**: The situation is different from a remote control point of view. Here the parameter to be monitored (e.g., **PAT**) is a property of the **Monitoring Configuration** filename under which it is saved on disk. The filename of the **Monitoring Configuration** itself is a member of the actual variable (**Upper Period**, **Class**) that is to be accessed.

![Fig. 18 Variable structure of monitoring configuration](image)
LabVIEW™ SNMP programming for the R&S® DVM, DVQ and ETX-T made easy with the Viodia™ SNMP Toolkit

From a MIB point of view, a parameter **Parameter to be monitored** is an element of a two-dimensional array indexed as:

\[
[\text{Name monitoring configuration}][\text{Parameter to be monitored}]
\]

This array is defined in the MIB as `controlMonitoringConfigAlarmTable`. The matching row object defines `controlMonitoringConfigName`, which serves as a row index; i.e., for every Monitoring Configuration, there exists an instance or row element in the table. The **Parameter to be monitored** is addressed with the column index `controlMonitoringConfigAlarmTestIndex` via fixed enumerations defined in `IndexTransportStreamTestConfig`. A graphical representation of this concept is shown below.

This also means that the variable structure is dynamic and the number, length, and precise OID depend on the available Monitoring Configurations and names stored on disk.

<table>
<thead>
<tr>
<th>monitoring Configuration</th>
<th>TS Synchronisation</th>
<th>Sync Byte</th>
<th>PAT</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of monitoring configuration 1</td>
<td>TS Synchronisation</td>
<td>Sync Byte</td>
<td>PAT</td>
<td>...</td>
</tr>
<tr>
<td>Name of monitoring configuration 2</td>
<td>TS Synchronisation</td>
<td>Sync Byte</td>
<td>PAT</td>
<td>...</td>
</tr>
<tr>
<td>Name of monitoring configuration 3</td>
<td>TS Synchronisation</td>
<td>Sync Byte</td>
<td>PAT</td>
<td>...</td>
</tr>
</tbody>
</table>

**Fig. 19 R&S® DVM/100/400 monitoring parameters settings from a remote control point of view**

When we bear the previous in mind, we can determine how to form an OID to access a particular Monitoring Configuration variable via SNMP remote control.
LabVIEW™ SNMP programming for the R&S® DVM, DVQ and ETX-T made easy with the Viodia™ SNMP Toolkit

Example: The **TS Synchronization** alarm determines after how many TS packets with a missing sync byte (0x47) a loss of synchronization is reported. The OID for the **Loss After Packets** setting of the **TS Synchronization** parameter is formed by appending the OID of **tsSyncLoss** (1101) (refer to the R&S®DVM/100/400 manual) to the ASCII values (decimal) of the current **Monitoring Configuration**; e.g., **Config (DVB)**.

![Fig. 20 Loss after packets setting for TS Synchronization parameter](image)

Since we are referring to the limit setting, this string is then appended to the OID of the **controlMonitoringConfigAlarmLimit** object. Figures 21 and 22 show the process in detail.

![Fig. 21 OID formation process of a monitoring configuration variable](image)

The process applies to all the parameters in the **Monitoring Configuration**.

---

7BM70_0E  20  Rohde & Schwarz
LabVIEW™ SNMP programming for the R&S® DVM, DVQ and ETX-T made easy with the Viodia™ SNMP Toolkit

controlMonitoringConfigAlarmLimit
object ID

<table>
<thead>
<tr>
<th>ASCII value</th>
<th>67</th>
<th>111</th>
<th>110</th>
<th>102</th>
<th>105</th>
<th>103</th>
<th>32</th>
<th>40</th>
<th>68</th>
<th>86</th>
<th>66</th>
<th>41</th>
</tr>
</thead>
<tbody>
<tr>
<td>Character</td>
<td>C</td>
<td>o</td>
<td>n</td>
<td>f</td>
<td>i</td>
<td>g</td>
<td>(</td>
<td>D</td>
<td>V</td>
<td>B</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

OID 1101
Object tsSyncLoss

+ ASCII values of the name of the Monitoring Configuration separated by periods

+ object ID of Monitoring parameter e.g. tsSyncLoss

Fig. 22 OID formation process of a monitoring configuration variable

SNMPset-request and SNMPset-request return with an error when the OID is formed with ASCII codes of a non-existent Monitoring Configuration.
4.3 Log Entries

If necessary, the Statistics & Log contents can be retrieved via SNMP. Just like the Monitoring Configuration, the Statistics & Log variable structure in the R&S DVM/100/400 is dynamic in nature.

From a remote control point of view, the different columns (Time, Class, Event, Detail, ...) of one single line in a Statistics & Log entry are split up. In addition, entries are to be addressed line by line. For example, the full content of entry 991 “TDT missing” in Fig. 23 requires 7 SNMPset-request operations.

The OID is formed by appending the Log Sequence number, say the sequential order of entry (First column), to the analyzer input port (1, 2, 3, 4) to which the TS stream that generated the log is connected. This OID is then appended to the hexadecimal representation of the MAC address of the analyzer board on which the port resides, and subsequently to the OID of the actual parameter to be read (Time, Class, Event, Detail, ...).
LabVIEW™ SNMP programming for the R&S® DVM, DVQ and ETX-T made easy with the Viodia™ SNMP Toolkit

The full formation process is depicted below.

Fig. 25 OID formation process to access a log entry

Fig. 26 OID formation process to access a log entry
Summary

LabVIEW™ programming enthusiasts who want to control the R&S® DVM100/400, DVQ, DVQM, ETX-T, and other instruments which support SNMP from a host computer will find a great tool in the Viodia™ SNMP Toolkit. It not only shortens the learning curve, and therefore development time, but also reduces the required network programming know-how to the essentials. This is beneficial, especially for the RF and broadcasting engineer who rarely deals with computer network technology.

6 Literature

User’s manuals:


Articles, books and application notes:


7 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@rohde-schwarz.com.
8 Ordering Information

**R&S® DVM family**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVM10</td>
<td>Depiction</td>
<td>0865.1900.03</td>
</tr>
<tr>
<td>DVM10L</td>
<td>MPEG-4 Monitoring System</td>
<td>0865.1950.02</td>
</tr>
<tr>
<td>DVM20</td>
<td>MPEG-2 Monitoring System</td>
<td>0865.1900.03</td>
</tr>
<tr>
<td>DVM-B1</td>
<td>MPEG-4 Analysis Board</td>
<td>0865.2283.02</td>
</tr>
<tr>
<td>DVM-K1</td>
<td>TS-Monitoring</td>
<td>0865.5211.02</td>
</tr>
<tr>
<td>DVM-K2</td>
<td>TS-Capture</td>
<td>0865.5234.02</td>
</tr>
<tr>
<td>DVM-K10</td>
<td>In Depth Analysis</td>
<td>0865.5238.02</td>
</tr>
<tr>
<td>DVM-K11</td>
<td>Data Broadcast Analysis</td>
<td>0865.5311.02</td>
</tr>
<tr>
<td>DVM-K12</td>
<td>Template Monitoring</td>
<td>0865.5320.02</td>
</tr>
<tr>
<td>ZZA-111</td>
<td>Basic support kit</td>
<td>1096.3254.00</td>
</tr>
<tr>
<td>DVM95</td>
<td>MPEG-2 Monitoring System</td>
<td>0865.1900.03</td>
</tr>
<tr>
<td>DVM50-K9</td>
<td>In Depth Analysis</td>
<td>0865.5434.02</td>
</tr>
<tr>
<td>DVM400</td>
<td>Digital Video Measurement System</td>
<td>0865.1800.03</td>
</tr>
<tr>
<td>DVM400-B1</td>
<td>MPEG-4 Analysis Board</td>
<td>0865.5530.02</td>
</tr>
<tr>
<td>DVM400-B3</td>
<td>TS Generator</td>
<td>0865.5511.02</td>
</tr>
<tr>
<td>DVM400-B4</td>
<td>Upgrade TS Generator, TRP Recorder</td>
<td>0865.5523.03</td>
</tr>
</tbody>
</table>

**R&S® ETX-T**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETX-T1</td>
<td>DTV Monitoring Receiver</td>
<td>0868.0189.40</td>
</tr>
<tr>
<td>ETX-B1</td>
<td>MPEG-2 Real Time Analysis w Decoder</td>
<td>0868.0415.02</td>
</tr>
<tr>
<td>ETX-B2</td>
<td>MPEG-2 Real Time Analysis w Decoder</td>
<td>0868.0401.02</td>
</tr>
<tr>
<td>ETX-B3</td>
<td>MPEG-2 Real Time Analysis w Decoder</td>
<td>0868.0401.33</td>
</tr>
<tr>
<td>ETX-R1</td>
<td>MPEG Saw Filter</td>
<td>0868.0412.02</td>
</tr>
<tr>
<td>ETX-B12</td>
<td>MPEG Saw Filter</td>
<td>0868.0444.02</td>
</tr>
<tr>
<td>ETX-B13</td>
<td>MPEG Saw Filter</td>
<td>0868.0444.33</td>
</tr>
<tr>
<td>ETX-K10</td>
<td>TP Option</td>
<td>0868.0480.02</td>
</tr>
<tr>
<td>ETX-DCV</td>
<td>Documentation of ETXs</td>
<td>0868.0480.28</td>
</tr>
</tbody>
</table>

ROHDE & SCHWARZ GmbH & Co. KG  Mühldorferstrasse 15  D-81671 München  P.O.B 80 14 69  D-81614 München  Telephone +49 89 4129 - 0  Fax +49 89 4129 - 13777  Internet: http://www.rohde-schwarz.com

This application note and the supplied programs may only be used subject to the conditions of use set forth in the download area of the Rohde & Schwarz website.