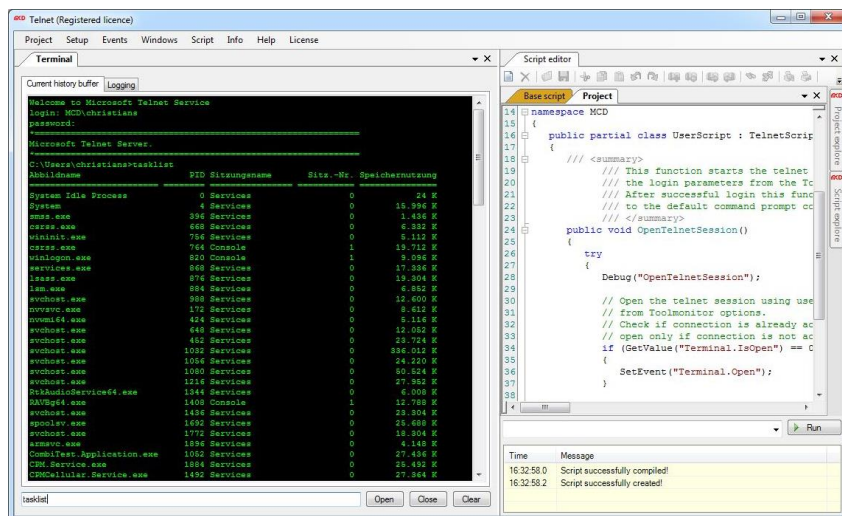


Manual

Toolmonitor Telnet



GET IN **touch**
WITH SENSITIVE TESTING

Softline

Modline

Conline

Boardline

Avidline

Pixline

Application

MCD Elektronik GmbH
Hoheneichstr. 52
75217 Birkenfeld
Tel. +49 (0) 72 31/78 405-0
Fax +49 (0) 72 31/78 405-10
info@mcd-elektronik.de
www.mcd-elektronik.com
HQ: Birkenfeld
Managing CEO: Bruno Hörter
Register Court Mannheim
HRB 505692

Template version: 5.4 / 2018-04-09
V1.2 2018-08-24CS (TS)

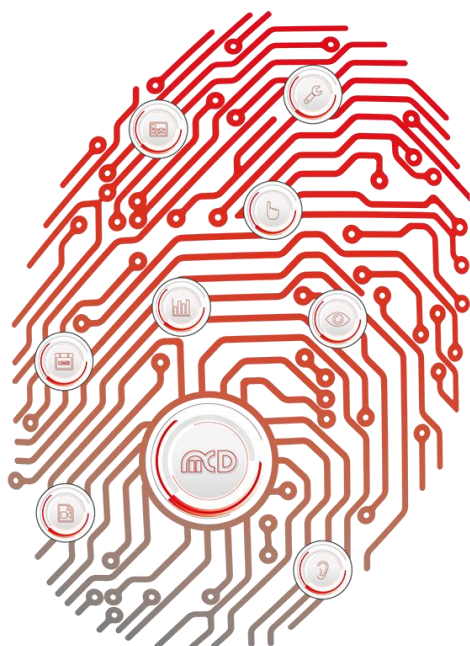


Table of Contents

1. GENERAL.....	3
2. INSTALLATION OF SOFTWARE.....	4
2.1. REQUIREMENTS.....	4
2.2. LICENSE	4
2.3. REGISTER COM SERVER.....	6
2.4. STARTING OF THE INSTALLED SOFTWARE	6
3. BRIEF INTRODUCTION TO OPERATION.....	7
4. SOFTWARE MANUAL	8
4.1. GENERAL.....	8
4.2. TERMINAL	8
4.3. OPTIONS.....	9
4.4. EVENTS.....	10
5. PROGRAMMING	11
5.1. CONTROLEVENTS.....	11
5.2. READVALUES	11
5.3. WRITEVALUES	11

1. General

This Toolmonitor can be used for the automatic control of Telnet sessions and the evaluation of results.

The connection can be established either manually or automatically depending on configuration. Arbitrary commands can be executed. Their results are returned and displayed. This permits the simple integration of auxiliary systems (such as Unix®, Linux®, Windows® etc.) for asynchronous measurement and control tasks without additional programming.

The following functions are available in the MCD Toolmonitor Telnet:

- Automatic or manual opening of Telnet sessions
- Reading of responses with a previously configured trigger character
- Evaluation of responses by user - defined script functions
- Simultaneous use of the standard console

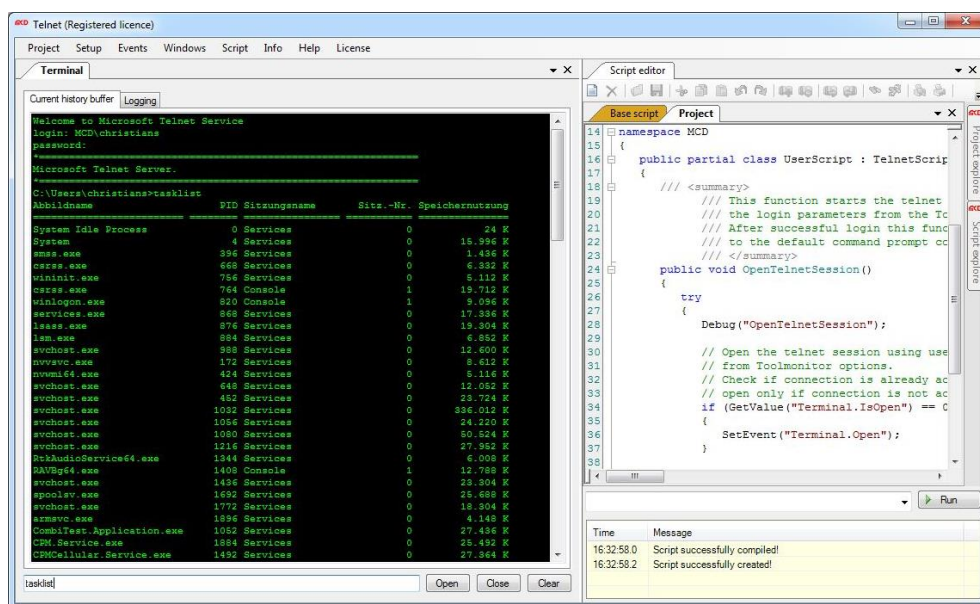


Figure 1: Toolmonitor Telnet with Script Editor

The program interface can largely be freely laid out and adapted to user requirements. Once created, configurations can be stored in **Project Files** and loaded again when needed. The installation files include example projects.

An integrated **Script Engine** can be used to carry out all measurements and settings automatically. In combination with the **Visualization module**, custom user interfaces can also be created.

For third - party software, the Toolmonitor Telnet can be fully remote controlled. The interface can be either COM / DCOM or a .Net assembly. This allows the Toolmonitor to be integrated into many different applications (Microsoft Visual Studio® (C#, C++, Visual Basic), Microsoft Office® (for example Excel®), Open Office®, LabVIEW®, MCD TestManager CE, standard Windows DLL).

Order number: # 118954

2. Installation of Software

2.1. Requirements

- Operating system: Windows XP® - Windows 8.1®
- Architecture: 32 bit or 64 bit
- .Net framework: Version 3.0

To install the MCD Toolmonitor Telnet, it is sufficient simply to copy *TelnetMonitor.exe* file into any directory on the target system.

2.2. License

To protect the software from unauthorized use, it is required to license the Toolmonitor after installation.

For purposes of demonstrations and testing the Toolmonitor can also be operated for 30 minutes at a time without a license. Some program functions are disabled. A 24 - hour temporary license can also be activated while waiting for permanent activation (for example on a weekend).

To activate the Toolmonitor, please open the **License administration** dialog from the menu item **License → Register**.



Figure 2: Calling up the Registration Dialog

1.) The **Current licensing** dialog shows the status of your current license:

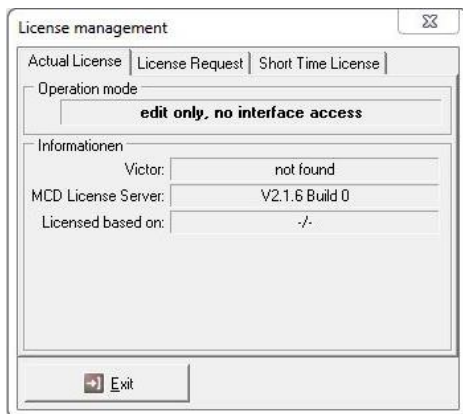


Figure 3: Retrieving License Status

2.) To request a permanent license for your software, please proceed as follows:

- Select the *Request license* dialog
- Specify the number of licenses needed (for your PC) in the *Number of licenses* field
- Click the *Generate request file* button
- Another window then opens to ask you to save the *MCD Licenser Request file (*.mlr)*
- Please save this file and send it by email to info@mcd-elektronik.de. Please specify an order number or project number to simplify allocation
- You will then receive an email from MCD Elektronik with your license file (*MCD License Key *.mlk*) attached
- Finally, save this file either under *C:\Windows* or into the folder where the .exe file for your software was installed
- After the next start of your software, it will then be available with its full functional scope



Figure 4: Requesting a Permanent License

3.) To activate a temporary license (24 hours), please select the **Temporary license** tab. Then enter the sequences of digits from the left window into the right window. If you cannot interpret the numbers, please click the *New number* button to receive a new number. Once you have correctly entered the number, you can activate the temporary license using the *Activate license* button. Please note that the temporary license will expire as soon as you stop the software. However, you can activate the temporary license as often as you like.

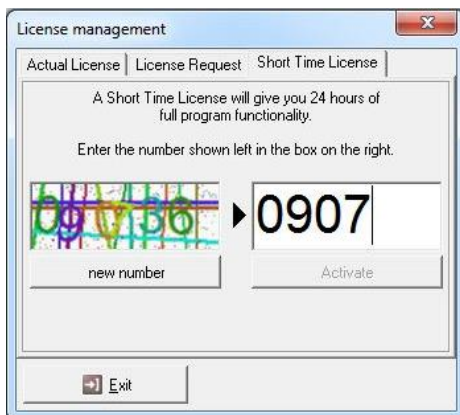


Figure 5: Requesting a Temporary License

2.3. Register COM Server

This command registers the Toolmonitor as a COM server. This is required if the Toolmonitor will be remote controlled by other programs, such as the MCD TestManager.

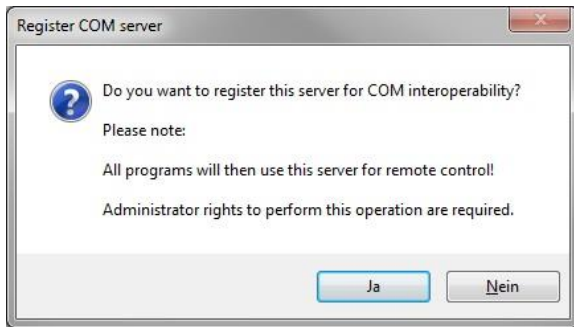


Figure 6: Register COM Server

2.4. Starting of the Installed Software

The Toolmonitor Telnet is started by executing *TelnetMonitor.exe*.

3. Brief Introduction to Operation

- 1.) Starting the Toolmonitor Telnet.
- 2.) Edit the setup with the desired hostnames / login data (activate "Auto open" to perform logins automatically).

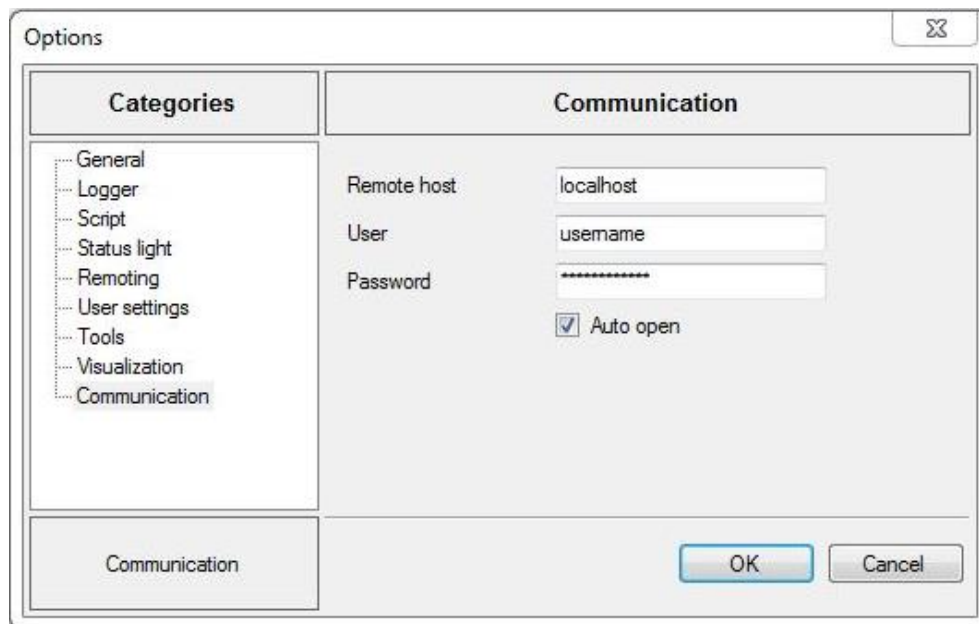


Figure 7: View of Options for Telnet Communication

- 3.) Once the connection is established and the user is logged in, commands can be entered into the text box on the **Windows** → **Terminal** window, then execute by pressing the enter key.

4. Software Manual

4.1. General

The main sections of this documentation are the **Functions and Programming** sections. The **Functions** section provides descriptions of the individual modules of the software. The *Programming* section provides a tabular list of the interfaces provided to users of this software. The interface documents the functions for access to the Toolmonitor and the different modules specify the valid parameter values for those functions.

4.2. Terminal

In the **Terminal window** of the Toolmonitor Telnet, the *Open* and *Close* buttons can be used to start and stop the Telnet session using the connection data from the *Options*. The **History display** can be cleared by clicking the *Clear* button.

The *History display* only has a limited number of lines available; this can be adjusted for long commands or responses using the programming interface (*Command HistoryBufferSize*).

Commands can be entered into the text box and sent with the enter key.

All commands in the programming interface work in the *History view*. If this is cleared with the *Clear* button or the *Clear* command, this data will no longer be available.

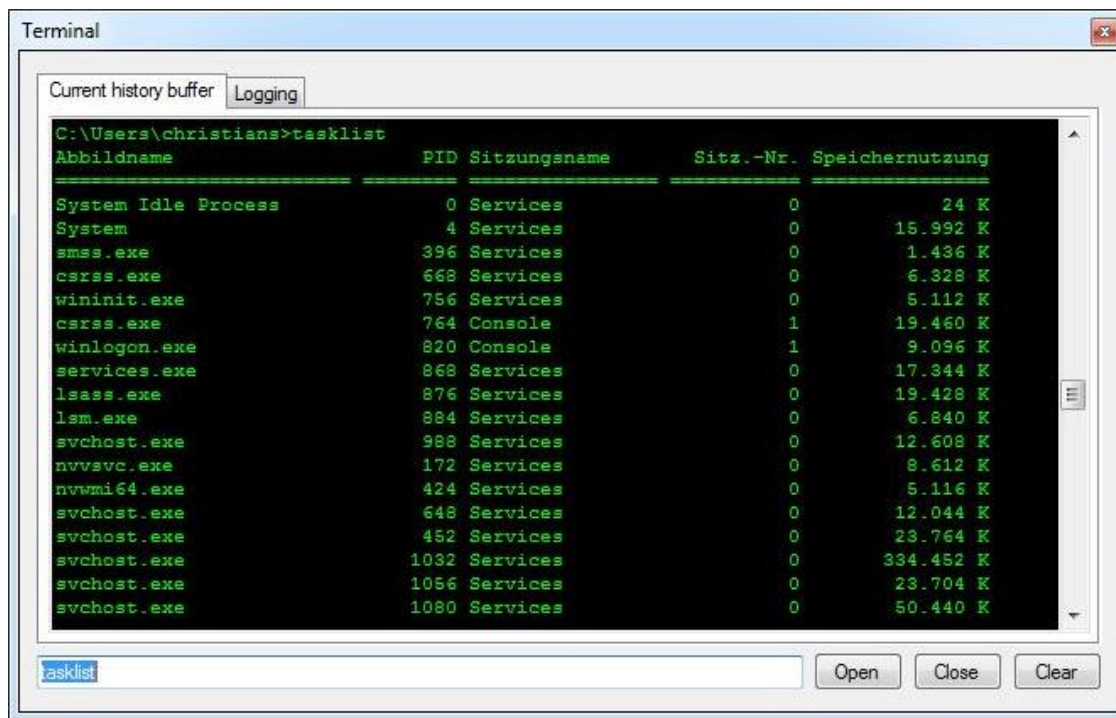


Figure 8: Terminal Window with History View

In contrast to the *History view*, the **Log view** records the entire communication session.

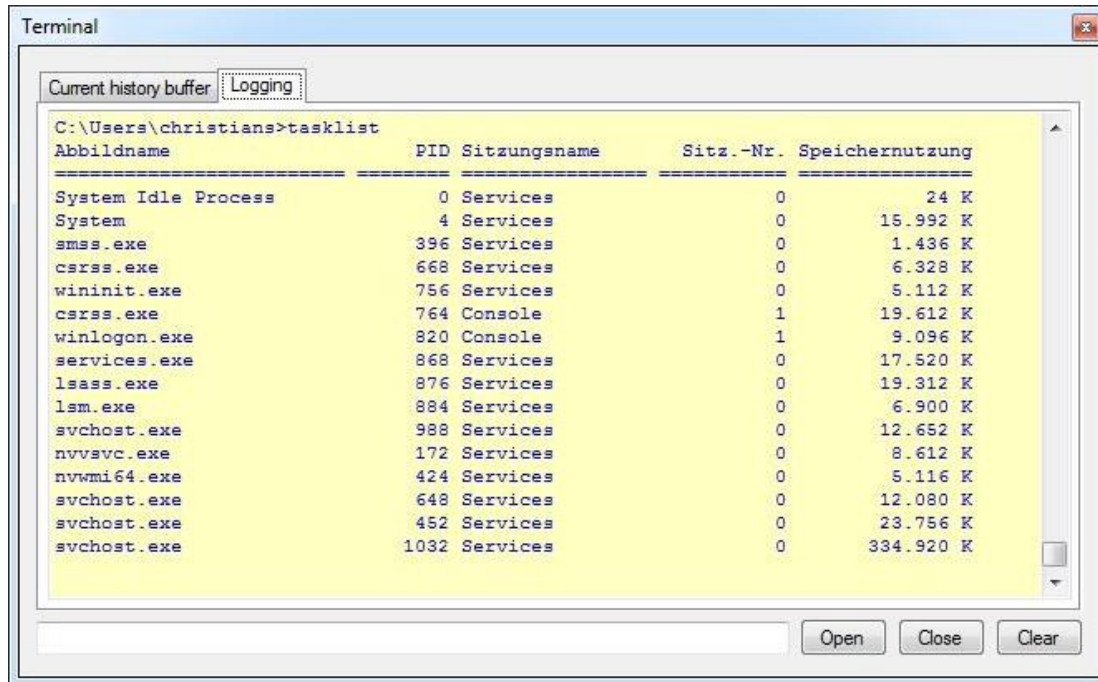


Figure 9: Terminal Window with Log View

4.3. Options

On the **Communication** options dialog, parameters can be specified for the current application.

- Remote host: The IP address or hostname to which the Telnet connection should be established can be entered here. If a name is used, it must be ensured that a name server is available
- User: The username for logging into the Telnet session
- Password: The password for logging into the Telnet session
- Auto open: If this checkbox is checked, then when the Toolmonitor starts it immediately attempts to establish a telnet connection to the specified host (port 23). If the host is unreachable, the message "Connection timeout ..." will be displayed

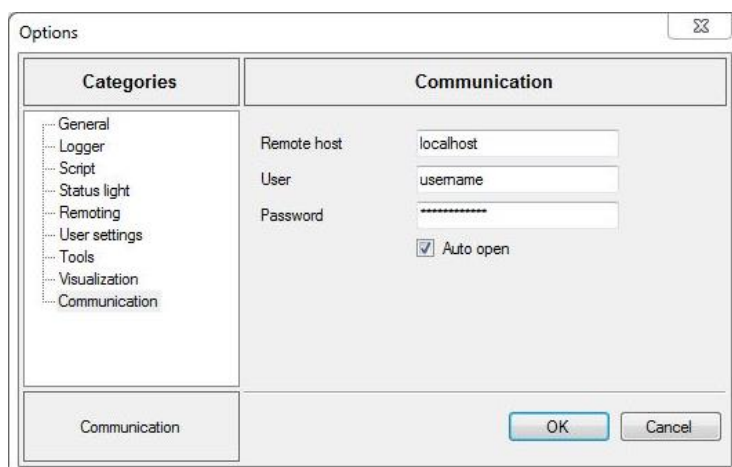


Figure 10: Telnet Options

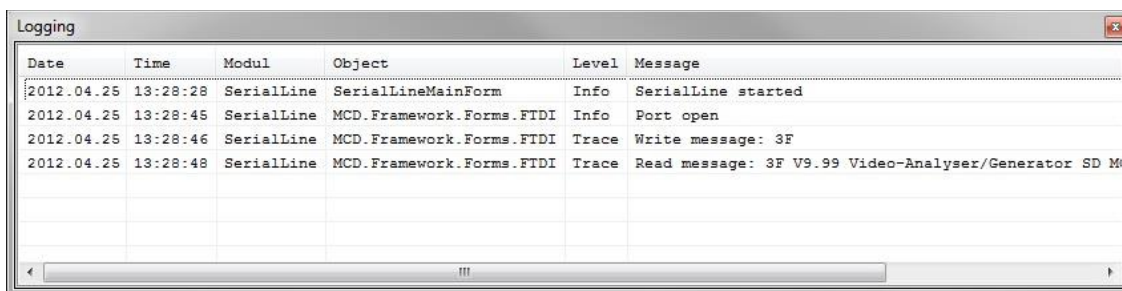
4.4. Events

This can be used to call up the displays for **log and trace messages**.

- Logging: This menu is used to call up the display of *log messages* for general events, warnings, errors, etc.
- Trace: This menu is used to call up the display of *Trace messages* (messages sent or received)

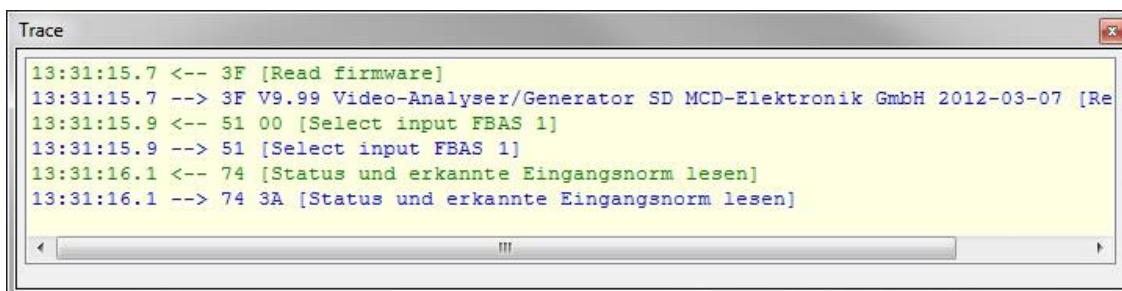


Figure 11: Events Menu



Date	Time	Modul	Object	Level	Message
2012.04.25	13:28:28	SerialLine	SerialLineMainForm	Info	SerialLine started
2012.04.25	13:28:45	SerialLine	MCD.Framework.Forms.FTDI	Info	Port open
2012.04.25	13:28:46	SerialLine	MCD.Framework.Forms.FTDI	Trace	Write message: 3F
2012.04.25	13:28:48	SerialLine	MCD.Framework.Forms.FTDI	Trace	Read message: 3F V9.99 Video-Analyser/Generator SD M

Figure 12: Log Monitor



13:31:15.7	<-- 3F [Read firmware]
13:31:15.7	--> 3F V9.99 Video-Analyser/Generator SD MCD-Elektronik GmbH 2012-03-07 [Re
13:31:15.9	<-- 51 00 [Select input FBAS 1]
13:31:15.9	--> 51 [Select input FBAS 1]
13:31:16.1	<-- 74 [Status und erkannte Eingangsnorm lesen]
13:31:16.1	--> 74 3A [Status und erkannte Eingangsnorm lesen]

Figure 13: Trace Monitor

5. Programming

5.1. ControlEvents

These names are used to start events.

Example: `SetEvent("Terminal.Open");`

Enumeration values:

- **Open:** Open a Telnet connection with the login data from the *Options*. The hostname can be set in the *Options*, or using the command "*HostName*"
- **Close:** Close a Telnet connection
- **Clear:** Delete the *History* in the Terminal window
- **Update History:** Update the *History* in the Terminal window

5.2. ReadValues

These names are used to read out data, settings or parameters.

Example: `string history = GetValue("Terminal.History");`

Enumeration values:

- **Read:** Read out the characters currently displayed in the *History* window. The *History* is then cleared
- **History:** Read out the characters currently displayed in the *History* window. The *History* is preserved, and can be deleted using the command *Clear*
- **HistoryEnd:** This command can be passed a number after a period (".") at the end. As a response, the specified number of characters will then be passed at the end of the *History*.
- **CheckHistoryEnd:** This command can be passed a character after a period (".") at the end. The response is then either "0" or "1" to denote whether the desired character was found at the end of the *History*
- **DataAvailable:** Command to check whether data has been received
- **BytesAvailable:** Command to retrieve the number of bytes received

5.3. WriteValues

These names are used to set values or start actions that require a parameter to be passed.

Example: `SetValue("Terminal.Write", "telnet localhost");`

Enumeration values:

- **Write:** Write commands over the Telnet interface
- **HostName:** Command to dynamically set the host to be used at runtime. This stops the current connection when set
- **HistoryBufferSize:** Set the number of characters that can be printed in the *History*. Important is, if this number is too small, parts of longer responses may be missing