

*COMPROTware:Testtool*

# **COMPROTware:Testtool**

## **Introduction and Training**

**Real Thoughts GmbH**

Haid-und-Neu-Straße 7

76131 Karlsruhe

Germany

Fon +49-721-6276730, Fax +49-721-6276731

Website [www.realthoughts.de](http://www.realthoughts.de)

E-mail [info@realthoughts.de](mailto:info@realthoughts.de)

All rights reserved. No part of this publication may be reproduced, by any means whatsoever,  
without the prior written permission of **Real Thoughts**.

## COMPROTware:Testtool General

### General

- **CPTT** is an integrated testtool for transmission protocols of telecontrol systems
- ... combines different protocols in one user interface
- ... simulates master or slave station of a telecontrol system

```

COMPROTware:Testtool - Hongkong - File: L:\... \adnp3_001.mlg
File Edit Operate Extra Help
11:11:33.963
  Used protocol profile: "DNP3-1999: Source Addr.: 1, Destination Addr.: 1"
  Used timeout intervals: "Message T0=1s, Link Down T0=7s"
  Serial device "COM5" opened: 9600 baud, 8 data bits, no parity and 1 stop bits, 3ms gap supervision time
11:11:33.963
  Test for Slave ...
11:11:34.976
  Station B->A Link established
11:11:35.017
  Station A->B Link established
11:11:37.070  1 -> 1
  Read #0
    Class 0 Data (Static Data): All objects
11:11:37.124  1 -> 1    Confirm #0
11:11:37.290  1 -> 1
  Response #0 [IIN: C1s1Av DvceRestr]
  Binary Input: 14..+16..29
    Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
  ( 14+ 0) 0x5001|off ON  off ON  off off off off off off off off off off ON
  Binary Input: 33..+16..48
    Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
  ( 33+ 0) 0x0000|off off off off off off off off off off off off off off off
  Binary Output Status: 0..+4..3
  ( 0+ 0) 0x01|off local remote coalst restrt ONLINE
  ( 0+ 1) 0x01|off local remote coalst restrt ONLINE
  
```

(For details enlarge picture!)

### Delivery

- The delivery includes
  - ▲ printed installation manual
  - ▲ CD-ROM with installation, documentation and examples
  - ▲ USB-, parallel port- or PCMCIA-dongle to license the protocols

## COMPROTware:Testtool CD-ROM & Document

### CD-ROM

- CD-ROM contains the installation, documentation and examples.

### Documentation

- The directory `\doc` contains the description of the software its self: Installation Manual, User's Manual, and Programmer's Guide; all material comes in German and English, and the User's Manual additionally in Italian
- `\doc\Support` contains descriptions of **CPTT**-related problems and advanced topics, like remote license updating
- In `\doc\Marketing` the data sheet, the end user price list and the presentation can be found.

## COMPROTware:Testtool Installation

### Installation of CPTT

- The **CPTT** installation is based on Microsoft Windows Installer
- It is described in detail in the Installation Manual
- New releases are installed parallel to old ones; in case of problems with the new release you still have access to the old one. Old releases can be very quickly uninstalled using Microsoft Windows Installer
- The installation occurs in five steps (sequence at will, dongle **not** necessary):
  - ▲ Installation Java Runtime Environment for Windows x86 (32 Bit):  
<http://java.com/download>
  - ▲ Installation WibuKey Runtime for Windows (Windows 32/64 Bit, multilanguage):  
<http://www.wibu.com/us/downloads-user-software.html>
  - ▲ Installation of **COMPROTware:Testtool**  
Start with \iX86\_WIN32\CPTT\setup.exe
  - ▲ Installation WinPcap  
<http://www.winpcap.org/>
  - ▲ Installation of **RIO Server**  
The installation of this option is described later on
- Normaly, the standard values can always be used

## COMPROTware:Testtool Licensing

### Licensing

- For simulation and monitoring a license is required; Offline Analysis can be done without a dongle any time
- **CPTT** may be installed on any computer; a license is only required for simulation and monitoring
- The protocol specific licenses are stored in the dongle; only if a license for a protocol is stored in the dongle simulation/monitoring is possible
- Dongles are programmable:
  - ▲ A bit pattern in the dongle is used to control which protocols are licensed
  - ▲ Only one license per protocol can be stored on a dongle, but for all protocols a license can be stored on a dongle.
  - ▲ With remote programming, by exchanging a context file and an update file, the content of a dongle can be altered over electronic media
- The dongles are deliverable as
  - ▲ Dongle for USB
  - ▲ Dongle for parallel port
  - ▲ Dongle for PCMCIA slot
  - ▲ For more information about dongles, see <http://www.wibu.com/>

## COMPROTware:Testtool

### Start and General Handling

#### Start

- As usual under MS Windows, **CPTT** is started from the Start menu or by double-clicking on the icon on the desktop
- On program startup the configuration of the last program execution is restored.
- **CPTT** can be started several times, even several simulations may run at the same time
- **User Engine Classes** (see further below) can be loaded and started during program startup

#### General Handling

- The handling of **CPTT** meets the typical rules under MS Windows; with shortcut keys frequently used functions can be accessed very quickly
- The following file extensions are associated to **CPTT**:
  - ▲ .mlg for Message Log files (protocol traffic logs)
  - ▲ .mls for Message List files (message lists)
  - ▲ .cptt for configuration files (parameter sets)
- **CPTT** distinguishes between four operating modes:
  - ▲ Pure display of protocol traffic (*Operate -> Stop*)
  - ▲ Simulation of a Master (*Operate -> as Master*)
  - ▲ Simulation of a Slave (*Operate -> as Slave*)
  - ▲ Monitoring (*Operate -> Monitor*)

## COMPROTware:Testtool Examples

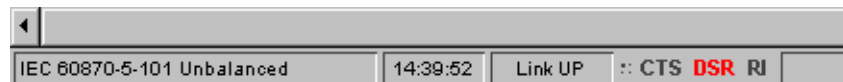
### Examples

- The delivered Messages List examples under `c:\Program files\realthoughts\comprotware\testtool\doc\<RELEASE>\MLS_Examples` with name suffix `*_Std_Example.mls` always carry out a complete demo of both Master and Slave; the Message Lists must be imported through *Edit -> Message List Storage ...* and then the menu item *File->Open from ...*
- Depending on which protocol you have licensed:
  - ▲ Is it a serial line-based protocol? Then connect two serial ports and run one tool on one port as Master and another tool on the other port as Slave.
  - ▲ It's even easier with network-based protocols: one instance of the testtool runs as Slave, the other runs as Master Station (hence the IP address has to be set to 127.0.0.1 (Localhost))
- The delivered examples should help you understand **CPTT** and give you some insight on the protocol

## COMPROTware:Testtool Simulation

### Simulation

- **CPTT** can simulate Master as well as Slave (Controlling and Controlled Station or Master and Outstation respectively) for all implemented transmission protocols
- First through *Edit -> Protocol Profile ...* choose the desired protocol from a protocol family; the station name makes the grouping of the different windows easier
- In the next window, the protocol specific parameter must be specified; for each protocol, there's a separate set of parameters and a separate input mask; the default values of **CPTT** meet common usage conditions
- Now with *Operate -> as Controlling/Operate -> as Master* or with *Operate -> as Controlled/Operate -> as Slave* you can simulate a control system or a sub device; *Operate -> Stop* stops the simulation
- In the window's title, the operation mode will be displayed: Controlling, Master, Controlled, Slave, Monitor, ...
- In the footer, the chosen protocol, the current time, state of the link (UP, down), the receiving of characters through animation and the modem signal CTS, DSR and RI are displayed:





## COMPROTware:Testtool Simulation

### Simulation - Continuation

- The protocol traffic is classified by colors:
  - ▲ The border color of the display window indicates the operation mode
  - ▲ Messages sent by **CPTT** are in bold
  - ▲ In IEC 60870-5-104: **green** is always Controlling, **blue** is always Controlled
  - ▲ In serial protocols: **green** is always Master, **blue** is always Slave; except with IEC 60870-5-101 balanced: **green** is always dir, **blue** is always DIR

```

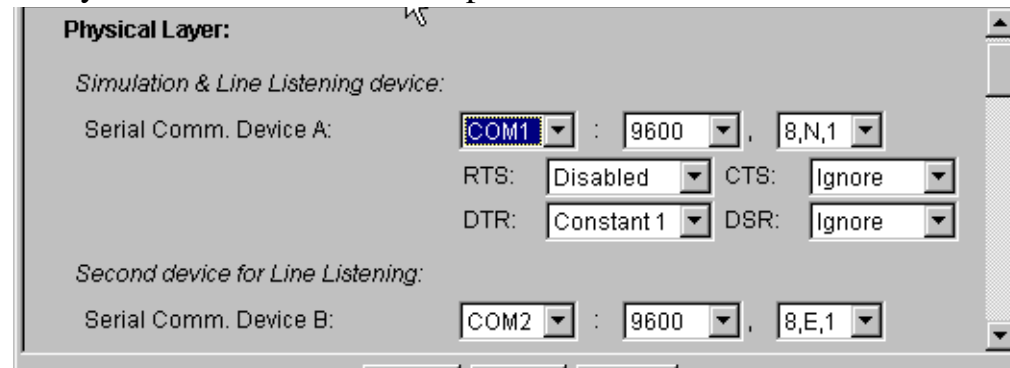
COMPROTware:Testtool - Karlsruhe - Controlling
File Edit Operate Extra Help
Used protocol profile: "IEC 60870-5-101: Transm.proc.: unbalanced, Link Addr.: 1, C
Used timeout intervals: "Message T0=1s, Link Down T0=7s"
Serial device "COM2" opened: 9600 baud, 8 data bits, no parity and 1 stop bits, 100
18:25:10.293
Test for Controlled Station ...
18:25:10.450
Link established
18:25:10.538 M_EI_MA_1 [init +] 1 0 COI=local power switch on, initializat
18:25:10.574 C_IC_MA_1 [act +] 1 0 QOI=Station interrogation (global)
18:25:10.730 C_IC_MA_1 [actcon +] 1 0 QOI=Station interrogation (global)
18:25:10.806 M_SP_MA_1 [inrogen +] 1 1 SPI=1|0n [IV nt sb bl]
18:25:10.870 M_DP_MA_1 [inrogen +] 1 2 DPI=0|Int [IV nt sb BL]
18:25:10.926 M_DP_MA_1 [inrogen +] 1 10203 DPI=1|0ff [IV nt sb BL]
18:25:10.994 M_DP_MA_1 [inrogen +] 1 10204 DPI=2|0n [IV nt sb BL]
18:25:11.072 M_DP_MA_1 [inrogen +] 1 10205 DPI=3|Ind [IV nt sb BL]
18:25:11.123 M_SP_TB_1 [inrogen +] 1 10101 SPI=1|0n [IV MT sb bl] 99
18:25:11.203 M_SP_TB_1 [inrogen +] 1 10102 SPI=0|0ff [IV MT sb bl] 99
IEC 60870-5-101 Unbalanced 18:25:21 Link UP :: CTS DSR RI
    
```

- About the state of the connection (displayed in the footer):
  - ▲ Link down: no connection established; with Master simulation **CPTT** tries to initiate a connection by sending restart frames
  - ▲ Link UP: connection established; messages on Application Layer are exchanged

## COMPROTware:Testtool Protocol Profiles

### Protocol Profiles

- Physical Layer with serial line-based protocols:



- ▲ The simulation runs with Serial Comm. Device A; the setting for the modem signal displayed here represents a direct connection
- ▲ Both Serial Comm. Device B and Serial Comm. Device A are used for the monitoring of serial line-based protocols; the modem settings will be ignored

COMPROTware:Testtool  
Protocol Profiles

## Protocol Profiles - Continuation

- Physical Layer in network-based protocols:

**Physical Layer:**

*According to standard:*

Controlled Station IP Address: 127 . 0 . 0 . 1

Port No.: 2404

*Implementation specific:*

Controlling Station IP Address: 127 . 0 . 0 . 1

Network Adapter (for Listening): [2] SiS 900 PCI Fast Ethernet Adapter

- During simulation of a Controlling Station, the Controlled Station IP Address indicates the IP Address of the Controlled Station; 2404 is the port no. for the connection establishment according to the IEC 60870-5-104-standard
- During monitoring, the Controlled Station IP Address and the Controlling Station IP Address are filter for the display of network packets; the value 255 is the wild-card value

COMPROTware:Testtool  
Protocol Profiles

## Protocol Profiles - Continuation

- Link Layer, parameters according to the standard:

**LPCI:**

*According to standard:*

Transmission Procedure: unbalanced

Direction Bit: 1

Address length [octets]: Link: 1

Link Address: 1

Struct. Link Address Format: %d Decimal

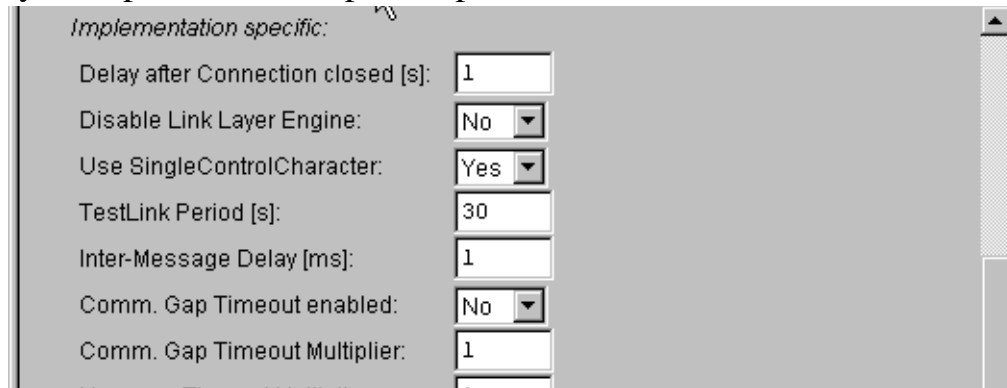
Frame length (net) [octets]: 255

- ▲ Transmission procedure: unbalanced (polling) or balanced (spontaneous)

## COMPROTware:Testtool Protocol Profiles

### Protocol Profiles - Continuation

- Link Layer, implementation-specific parameters:



Implementation specific:	
Delay after Connection closed [s]:	1
Disable Link Layer Engine:	No
Use SingleControlCharacter:	Yes
TestLink Period [s]:	30
Inter-Message Delay [ms]:	1
Comm. Gap Timeout enabled:	No
Comm. Gap Timeout Multiplier:	1

- ▲ Delay after connection closed: delay after a connection loss, before a new connection is tried to be established
- ▲ Disable Link Layer Engine: Link Layer Engine is deactivated (e.g. for connection reestablishment)
- ▲ Use SingleControlCharacter: should the Single Control Character 0xe5 be used?
- ▲ TestLink Period: duration period for TestLink-cycles
- ▲ Inter-message Delay: min. delay between the last received and next sent message
- ▲ Communication gaps are delays between two characters on a serial line; within a frame, gaps are not allowed; here the supervision of comm. gaps can be enabled and a timeout multiplier for tolerated gaps can be specified

COMPROTware:Testtool  
Protocol Profiles

## Protocol Profiles - Continuation

- Link Layer, implementation-specific parameters:

Message Timeout Multiplier:	1
Link Down Timeout Multiplier:	1
Test&Act - 2x-, 1xACD:	No
Test&Act - 2x-, 1xDFC:	No

- ▲ Message Timeout Multiplier: message reception timeout multiplier; reference time is calculated based on frame length and baudrate
- ▲ Link Down Timeout Multiplier: multiplier for link down timeout; reference time is calculated based on frame length and baudrate
- ▲ Test&Act - 2x-, 1xACD: if set, during Controlled Station simulation the ACD bit is set in every third response frame
- ▲ Test&Act - 2x-, 1xDFC: if set, during Controlled Station simulation the DFC bit is set in every third response frame

COMPROTware:Testtool  
Protocol Profiles

## Protocol Profiles - Continuation

- Application Layer, according to the standard and implementation-specific:

**ASDU:**

*According to standard:*

Address length [octets]: Common: 1 Inform.Obj.: 2

Field length [octets]: Cause of Transm.: 1

Struct. Common Address Format: %d Decimal

Struct. Info. Obj. Address Format: %d Decimal

*Implementation specific:*

Auto ACTCON&ACTTERM: Yes

- Auto ACTCON&ACTTERM: if set, then on each command an ACTCON and if necessary an ACTTERM will be sent

## COMPROTware:Testtool Simulation Examples

### Simulation Examples

- Now, the examples for IEC 60870-5-101 or -104 resp. should be read: open the files IEC60870\_5\_101n104\_Std\_Example.mls and IEC60870\_5\_101n104\_All\_TypeIdent.s.mls (when reading the last file, don't delete the Message List Storage)
- **CPTT** can be started several times on a computer: use your computer to simulate both Master and Slave; with a network-based protocol use the IP address 127.0.0.1 (local-host) for simulation, with a serial line-based protocol connect two serial ports with a cable



## COMPROTware:Testtool Navigation

### Navigation

- Using the keys *cursor up* und *down*, *Page up* and *Page down* and *Pos1*, *STRG+Pos1* and *End*, you can navigate through the window.
- Normally the view is updated „in place“, so that the newest protocol traffic is shown continuously.
- By scrolling up the snap-on is freed. The scroll back buffer allows the examination of previously exchanged messages.
- You return to the updated display by pressing the *End* key
- Navigation within the storage can be done anytime, even during monitoring and simulation

# COMPROTware:Testtool Formatting Options

## Formatting Options

- The window with the formatting options appears over the background menu (right mouse button) and *Formatting Options ...*; alternatively, the formatting options can be changed with the shortcuts <b>, <v>, <h> and <l>
- Formatting options (all combinations allowed):
  - ▲ Time only or with date
  - ▲ Verbose, multiline

```
09:45:28.465
 9|M_ME_NA_1|measured value, normalized value
VSQ [SQ, N=7], [COT=20|inrogen, tst pn], Originator=0x00
CA=4351
IOA=131124
NVA=0x110b|0.133148
QDS [iv nt sb bl ov]
```

or brief, single-lined

```
09:45:28.465 M_ME_NA_1 [inrogen +] 4351 131124 NVA=0.133 [iv nt sb bl ov]
( 131124+1 ) NVA=0.298 [iv nt sb bl ov]
( 131124+2 ) NVA=-0.421 [iv nt sb bl ov]
( 131124+3 ) NVA=0.260 [iv nt sb bl ov]
```

- ▲ Hexdump included

```
09:45:28.465
0x68 0x22 0x18 0x00 0x02 0x00 0x09 0x87 0x14 0x00
0xff 0x10 0x34 0x00 0x02 0x0b 0x11 0x00 0x17 0x26
0x00 0x2b 0xca 0x00 0x57 0x21 0x00 0xc0 0xe4 0x00
0xeb 0xd5 0x00 0xd7 0x21 0x00
M_ME_NA_1 [inrogen +] 4351 131124 NVA=0.133 [iv nt sb bl ov]
( 131124+1 ) NVA=0.298 [iv nt sb bl ov]
```

- ▲ With or without Link Layer

```
09:45:28.465
I: SSN=12, RSN=1
M_ME_NA_1 [inrogen +] 4351 131124 NVA=0.133 [iv nt sb bl ov]
( 131124+1 ) NVA=0.298 [iv nt sb bl ov]
( 131124+2 ) NVA=-0.421 [iv nt sb bl ov]
( 131124+3 ) NVA=0.260 [iv nt sb bl ov]
( 131124+4 ) NVA=-0.213 [iv nt sb bl ov]
```

## COMPROTware:Testtool Formatting Options

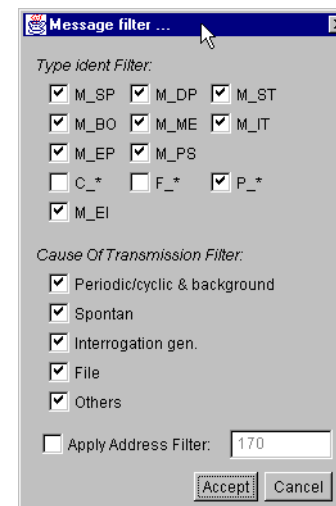
### Formatting Options

- Specially for information captured on the network (not for simulation): display of Ethernet Frames and/or TCP Headers is possible

```
00-09-22 09:45:28.465
Ethernet frame:
 00:10:7B:7F:FC:07 <- 00:00:23:09:01:27; Type=0x0800
IP datagram:
  VerSn=4, HdrLen=5*32bit, ToS=0, Length=184; Id=1120, Flgs=0, FrgmOffst=0; TtL=64, Prtcl=6, Chksm=2723
 172.28.144.3 -> 172.28.131.1
TCP header:
 Port: 2404 -> 3494
 SeqNo=2404421902, AckNo=1037098064
 Data Offset=5*32bit, Control Bits=0x18|urg ACK PSH rst syn fin, Window=8192; Cksm=53765, UrgntData=0

      M_ME_NA_l [inrogen +] 4351 131124 NVA=0.133 [iv nt sb bl ov]
                          ( 131124+l ) NVA=0.298 [iv nt sb bl ov]
```

- Filter for type indication, cause of transmission und station address through *Formatting Options ... -> Message Filter ...*



*COMPROTware:Testtool*  
*Formatting Options*

## Formatting Options

- The formatting options can be modified any time, even during simulation or monitoring
- A grey wavy line indicates that on that spot a message is being filtered (not displayed)

```

09:45:28.465 M_ME_NA_1 [inrogen +] 4351 131153 NVA=0.105 [iv nt sb bl ov]
( 131153+1 ) NVA=0.294 [iv nt sb bl ov]
( 131153+2 ) NVA=0.209 [iv nt sb bl ov]
( 131153+3 ) NVA=0.189 [iv nt sb bl ov]
( 131153+4 ) NVA=0.000 [iv nt sb bl ov]
09:45:32.211 M_ME_NA_1 [inrogen +] 4351 131159 NVA=0.248 [iv nt sb bl ov]
( 131159+1 ) NVA=0.000 [iv nt sb bl ov]
( 131159+2 ) NVA=0.000 [iv nt sb bl ov]

```

## COMPROTware:Testtool Display Info-elementes

### Display Information Elementes

- General rules for presentation:

```
09:45:28.465
  9|M_ME_NA_1|measured value, normalized value
    VSQ [SQ, N=7], [COT=20|inrogen, tst pn], Originator=0x00
    CA=4351
      IOA=131124
        NVA=0x110b|0.133148
          QDS [iv nt sb bl ov]
```

- ▲ Alternative representations (e.g. decimal and clear text) are separated by |; helpful, because very often both coding and clear text or decimal and hexadecimal value are of interest
  - ▲ Groups are embraced in [ and ], and are mostly named; this allows to easily link to hex representation
  - ▲ All bits are displayed, set bits in capital, clear bits in small letters; you always know which qualifiers an element has
- In the display mode BRIEF only a few representations are shown, in VERBOSE all necessary ones are there

## COMPROTware:Testtool IEC - Structured Addresses

### IEC 60870-5-101/-104 - Structured Addresses

- Structured addresses divide the address octets into smaller units
- **CPTT** offers a very flexible display:
  - ▲ Bit groups can be named
  - ▲ Display in decimal and hexadecimal
  - ▲ Predefined templates make the selection easier
- Format: V=%23\_16d Fld=%15\_8d Dev=%7\_0d is 8-8-8
  - ▲ Strings can contain any desired character
  - ▲ % (escape symbol) initiates a value, %% results in %
  - ▲ Optional: bit positions are specified from higher to lower bit; a higher bit position is separated from a lower one by \_
  - ▲ For decimal representation use d, for hexadecimal x
  - ▲ If no bit positions are given, then the entire address field is used
  - ▲ %23\_16d means that bit 23 to 16 (starting with bit 0) are shown decimal
  - ▲ %d|%x means that the address field is first shown in decimal and then in hexadecimal, separated by a |, e.g. like 192|0xc0

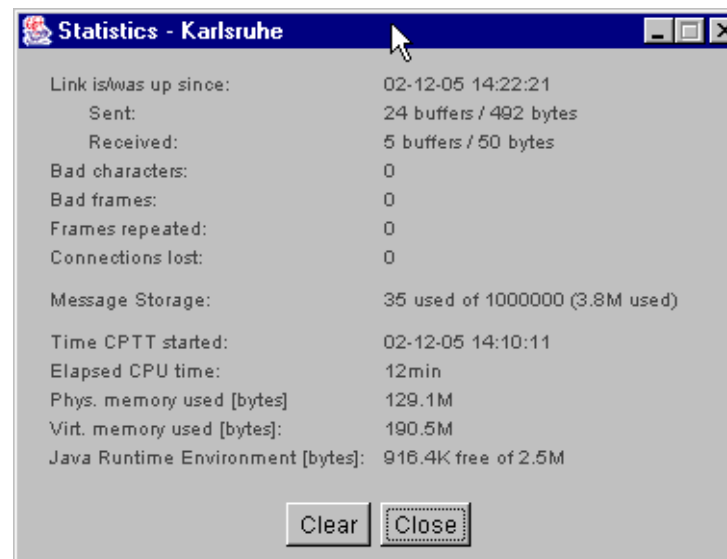
## COMPROTware:Testtool Settings & Statistic

### Settings

- To control the memory use, the size of the Message Storage (memory for the protocol traffic) can be restricted to a value between 3.000 and 1.000.000 (*Edit -> General Preferences ...*)
- Pure Link Layer information may be dropped directly after reception (*Edit -> General Preferences ...*). Thus, the event polling doesn't waste any entries in Message Storage

### Statistic

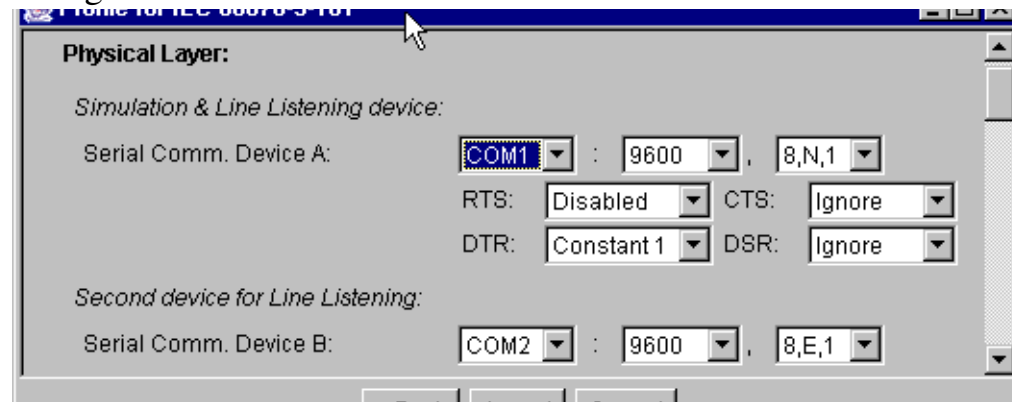
- Miscellaneous statistical values about running/last connection and about **CPTT** are shown by the window *Extra -> Statistics*; the values are cyclically refreshed



## COMPROTware:Testtool Monitoring

### Monitoring

- Monitoring is possible with serial line-based protocols through two serial interfaces, and with network-based protocols through the network adapter
- Concerning monitoring of network-based protocols: with MS Windows no additional components are required but you get better results if WinPcap is installed; monitoring of IEC 61850 GOOSE is only available if WinPcap is installed
- Monitoring is started in *Operate* -> *Monitor* and through *Operate* -> *Stop* stopped again
- With serial line-based protocols: both serial devices used for monitoring will be chosen from the list of available devices within the protocol profile; the modem signal settings are ignored:

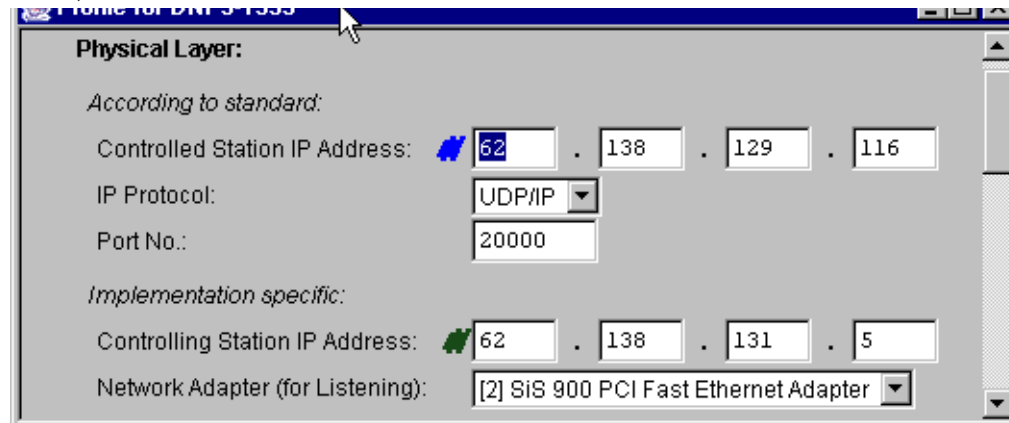




COMPROTware:Testtool  
Monitoring

## Monitoring - Continuation

- With network-based protocols: to do Network Monitoring, the IP address of both participants and the network adapter must be set in the protocol profile; if an IP address is unknown, 255 can be used as a wildcard value:



- Only with DNP3 over LAN/WAN: select transmission protocol: UDP/IP or TCP/IP
- Network Adapter specifies the network adapter which is used to monitor on the network
- Only the network traffic that passes by the network adapter can be monitored.
- Switches are used for balancing the network load, hubs are used to distribute network packets; a switch only forwards network packets to its destination, preventing network monitoring on arbitrary packets

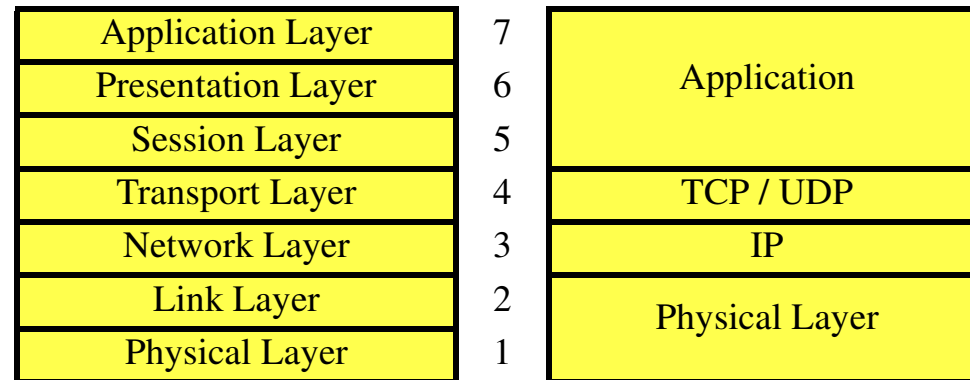
COMPROTware:Testtool  
Network Monitoring

## Network Monitoring

- Standardized model for information exchange:

### ISO/OSI Referenzmodell

### Internet Protocol

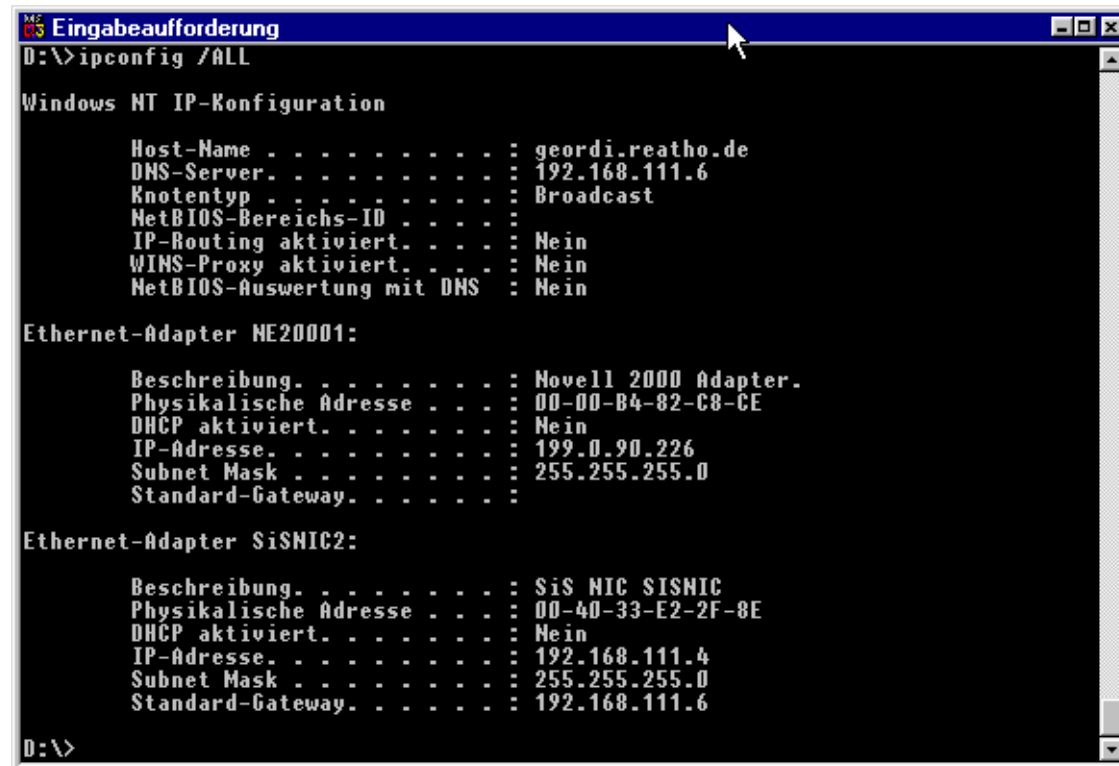


- MAC-Address: 00-00-b4-82-c8-ce  
IP-Address: 192.168.111.1 with network mask ff:ff:ff:00 or /24  
TCP/UDP Ports: port no. at both client and server
- Hub: physical link between network adapters  
Switch: like a hub, but with filtering on the physical layer  
Router: connects networks on the IP layer

## COMPROTware:Testtool Network Monitoring

### Network Monitoring

- Additional tools delivered by the operating system:
  - ▲ With all these tools, you get help through the option “-?”
  - ▲ ipconfig.exe - IP configuration of MS Windows



```
D:\>ipconfig /ALL

Windows NT IP-Konfiguration

    Host-Name . . . . . : geordi.reatho.de
    DNS-Server . . . . . : 192.168.111.6
    Knotentyp . . . . . : Broadcast
    NetBIOS-Bereichs-ID . . . . . :
    IP-Routing aktiviert. . . . . : Nein
    WINS-Proxy aktiviert. . . . . : Nein
    NetBIOS-Auswertung mit DNS : Nein

Ethernet-Adapter NE20001:

    Beschreibung. . . . . : Novell 2000 Adapter.
    Physikalische Adresse . . . . . : 00-00-B4-82-C8-CE
    DHCP aktiviert. . . . . : Nein
    IP-Adresse . . . . . : 199.0.90.226
    Subnet Mask . . . . . : 255.255.255.0
    Standard-Gateway. . . . . :

Ethernet-Adapter SiS NIC2:

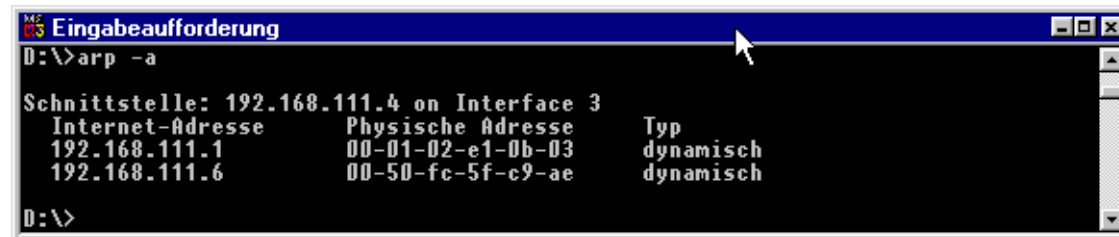
    Beschreibung. . . . . : SiS NIC SiS NIC
    Physikalische Adresse . . . . . : 00-40-33-E2-2F-8E
    DHCP aktiviert. . . . . : Nein
    IP-Adresse . . . . . : 192.168.111.4
    Subnet Mask . . . . . : 255.255.255.0
    Standard-Gateway. . . . . : 192.168.111.6

D:\>
```

## COMPROTware:Testtool Network Monitoring

### Network Monitoring

- arp.exe - IP/physical address mapping table (Address Resolution Protocol)  
Delivers list of all known pairs



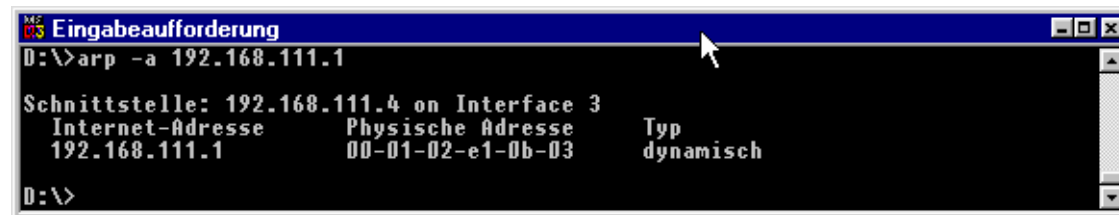
```
Microsoft Windows [Version 6.0.6002.18005]
(c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.

D:\>arp -a

Schnittstelle: 192.168.111.4 on Interface 3
Internet-Adresse    Physische Adresse    Typ
192.168.111.1      00-01-02-e1-0b-03    dynamisch
192.168.111.6      00-50-fc-5f-c9-ae    dynamisch

D:\>
```

Or only the physical address to an IP address



```
Microsoft Windows [Version 6.0.6002.18005]
(c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.

D:\>arp -a 192.168.111.1

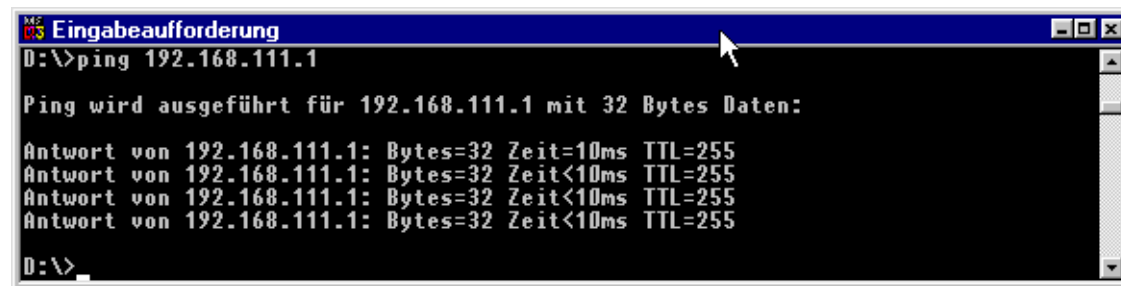
Schnittstelle: 192.168.111.4 on Interface 3
Internet-Adresse    Physische Adresse    Typ
192.168.111.1      00-01-02-e1-0b-03    dynamisch

D:\>
```

## COMPROTware: Testtool Network Monitoring

### Network Monitoring

- ▲ ping.exe - checks accessibility on IP layer



```
D:\>ping 192.168.111.1

Ping wird ausgeführt für 192.168.111.1 mit 32 Bytes Daten:

Antwort von 192.168.111.1: Bytes=32 Zeit=10ms TTL=255
Antwort von 192.168.111.1: Bytes=32 Zeit<10ms TTL=255
Antwort von 192.168.111.1: Bytes=32 Zeit<10ms TTL=255
Antwort von 192.168.111.1: Bytes=32 Zeit<10ms TTL=255

D:\>
```

- ▲ pathping.exe

COMPROTware:Testtool  
Network Monitoring

## Network Monitoring

- ▲ route.exe - network-routing table  
Which target (host or network) is reachable using which gateway through which interface?

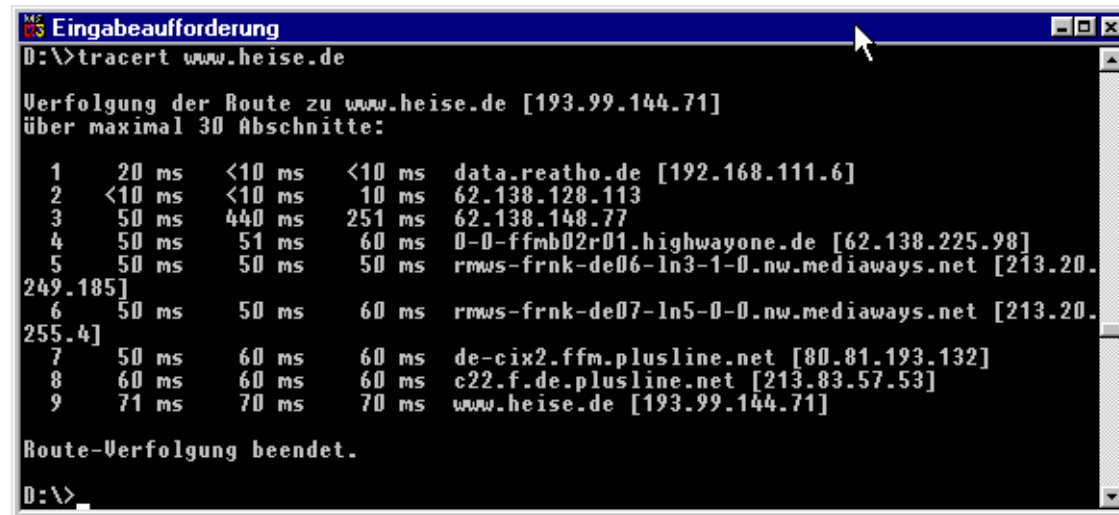
```

D:\>route PRINT
=====
Schnittstellenliste
0x1 ..... MS TCP Loopback interface
0x2 ...00 00 b4 82 c8 ce ..... Novell 2000 Adapter.
0x3 ...00 40 33 e2 2f 8e ..... SiS NIC SISNIC
=====
Aktive Routen:
Netzwerk  Ziel           Netzmaske      Gateway        Schnittst.  Metrik
-----
0.0.0.0   0.0.0.0         0.0.0.0        192.168.111.6  192.168.111.4  1
127.0.0.0 127.0.0.0       255.0.0.0      127.0.0.1     127.0.0.1     1
192.168.111.0 255.255.255.0 192.168.111.4  192.168.111.4  192.168.111.4  1
192.168.111.4 255.255.255.255 127.0.0.1     127.0.0.1     127.0.0.1     1
192.168.111.255 255.255.255.255 192.168.111.4  192.168.111.4  192.168.111.4  1
199.0.90.0 255.255.255.0  199.0.90.226  199.0.90.226  199.0.90.226  1
199.0.90.226 255.255.255.255 127.0.0.1     127.0.0.1     127.0.0.1     1
199.0.90.255 255.255.255.255 199.0.90.226  199.0.90.226  199.0.90.226  1
224.0.0.0  224.0.0.0       224.0.0.0      192.168.111.4  192.168.111.4  1
224.0.0.0  224.0.0.0       224.0.0.0      199.0.90.226  199.0.90.226  1
255.255.255.255 255.255.255.255 199.0.90.226  199.0.90.226  199.0.90.226  1
=====
D:\>
  
```

## COMPROTware:Testtool Network Monitoring

### Network Monitoring

- ▲ tracert.exe - way to a host  
Show me all hosts on the way to my target



```
MS-DOS Batch Eingabeaufforderung
D:\>tracert www.heise.de

Verfolgung der Route zu www.heise.de [193.99.144.71]
über maximal 30 Abschnitte:

  1  20 ms  <10 ms  <10 ms  data.reatho.de [192.168.111.6]
  2  <10 ms  <10 ms  10 ms  62.138.128.113
  3  50 ms  440 ms  251 ms  62.138.148.77
  4  50 ms  51 ms  60 ms  0-0-ffmb02r01.highwayone.de [62.138.225.98]
  5  50 ms  50 ms  50 ms  rmws-frnk-de06-ln3-1-0.nw.mediaways.net [213.20.
249.185]
  6  50 ms  50 ms  60 ms  rmws-frnk-de07-ln5-0-0.nw.mediaways.net [213.20.
255.4]
  7  50 ms  60 ms  60 ms  de-cix2.ffm.plusline.net [80.81.193.132]
  8  60 ms  60 ms  60 ms  c22.f.de.plusline.net [213.83.57.53]
  9  71 ms  70 ms  70 ms  www.heise.de [193.99.144.71]

Route-Verfolgung beendet.

D:\>
```

COMPROTware:Testtool  
Network Monitoring

## Network Monitoring

- ▲ netstat.exe -e  
Statistic Ethernet-Frames

```

D:\>netstat -e
Schnittstellenstatistik

                Empfangen                Gesendet
Bytes                20385660                9425644
Unicast-Pakete        25865                   23363
Nicht-Unicast-Pakete  3208                    1824
Verworfen              0                       0
Fehler                 0                       257
Unbekannte Protokolle 0
D:\>
    
```

- ▲ netstat.exe -s  
Statistic IP-Layer

```

Empfangene Pakete = 28542
Empfangene Vorspannfehler = 0
Empfangene Adreßfehler = 0
Weitergeleitete Datagramme = 0
Empfangene unbekannte Protokolle = 0
Empfangene verworfene Pakete = 0
Empfangene übermittelte Pakete = 28542
Ausgabeaufforderungen = 24930
Verworfen Routing-Pakete = 0
Verworfen Ausgabepakete = 0
Ausgabepakete ohne Routing = 0
Neuaufbau erforderlich = 0
Neuaufbau erfolgreich = 0
Neuaufbau erfolglos = 0
Erfolgreiche Datagrammfragmentierung = 0
Erfolglose Datagrammfragmentierung = 0
Erzeugte Fragmente = 0
ICMP-Statistik
    
```



COMPROTware:Testtool  
Network Monitoring

## Network Monitoring

```

ICMP-Statistik
      Empfangen  Gesendet
Nachrichten      46      46
Fehler           0       0
Ziel nicht erreichbar  0       1
Zeitüberschreitung 30       0
Parameterprobleme  0       0
Quelldrosselung   0       0
Redirects         0       0
Echos            0      45
Echo-Antworten   16       0
Zeiteinträge      0       0
Zeiteintrag-Antworten 0       0
Adreßmasken      0       0
Adreßmaske-Antworten 0       0

TCP-Statistik
Aktiv geöffnet           = 465
Passiv geöffnet         = 2
Erfolgreiche Verbindungsversuche = 2
Zurückgesetzte Verbindungen = 8
Aktuelle Verbindungen   = 4
Empfangene Segmente     = 25413
Gesendete Segmente      = 22856
Erneut übertragene Segmente = 9

UDP-Statistik
Empfangene Datagramme = 3083
Keine Anschlüsse      = 46
Empfangsfehler        = 0
Gesendete Datagramme  = 2019

D:\>
    
```

COMPROTware:Testtool  
Network Monitoring

## Network Monitoring

- Typical connection establishment

```

11:58:36.777
  IP datagram: 171.26.177.141 -> 171.26.177.150
  TCP header:
    Port: 1034 -> 2404
    SeqNo=41421, AckNo=0
    Data Offset=6*32bit, Control Bits=0x02|urg ack psh rst SYN fin, Window=8192; Cksm:
    TCP Option: Kind=2|Maximum Segment Size, Length=4: 1460
11:58:36.777
  IP datagram: 171.26.177.150 -> 171.26.177.141
  TCP header:
    Port: 2404 -> 1034
    SeqNo=3813791165, AckNo=41422
    Data Offset=6*32bit, Control Bits=0x12|urg ACK psh rst SYN fin, Window=8192; Cksm:
    TCP Option: Kind=2|Maximum Segment Size, Length=4: 1460
11:58:36.777
  IP datagram: 171.26.177.141 -> 171.26.177.150
  TCP header:
    Port: 1034 -> 2404
    SeqNo=41422, AckNo=3813791166
    Data Offset=5*32bit, Control Bits=0x10|urg ACK psh rst syn fin, Window=8760; Cksm:
11:58:37.108
  IP datagram: 171.26.177.141 -> 171.26.177.150
  TCP header:
    Port: 1034 -> 2404
    SeqNo=41422, AckNo=3813791166
    Data Offset=5*32bit, Control Bits=0x18|urg ACK PSH rst syn fin, Window=8760; Cksm:

U: STARTDT act
11:58:37.108
  IP datagram: 171.26.177.150 -> 171.26.177.141
  TCP header:
    Port: 2404 -> 1034
  
```

COMPROTware:Testtool  
Network Monitoring

## Network Monitoring

- While connection is established

```

11:58:37.168
  IP datagram: 171.26.177.150 -> 171.26.177.141
  TCP header:
    Port: 2404 -> 1034
    SeqNo=3813791172, AckNo=41444
    Data Offset=5*32bit, Control Bits=0x18|urg ACK PSH rst syn fin, Window=8192; Cksm:

I: SSN=0, RSN=1
  C_IC_MA_1 [actcon +]      12      0  QOI=Station interrogation (global)
11:58:37.268
  IP datagram: 171.26.177.141 -> 171.26.177.150
  TCP header:
    Port: 1034 -> 2404
    SeqNo=41444, AckNo=3813791188
    Data Offset=5*32bit, Control Bits=0x10|urg ACK psh rst syn fin, Window=8738; Cksm:
11:58:37.278
  IP datagram: 171.26.177.150 -> 171.26.177.141
  TCP header:
    Port: 2404 -> 1034
    SeqNo=3813791188, AckNo=41444
    Data Offset=5*32bit, Control Bits=0x18|urg ACK PSH rst syn fin, Window=8192; Cksm:

I: SSN=1, RSN=1
  M_SP_MA_1 [inrogen +]    12      263  SPI=1|0n      [iv nt sb bl]
                                267  SPI=1|0n      [iv nt sb bl]

I: SSN=2, RSN=1
  M_SP_MA_1 [inrogen +]    12    10016  SPI=0|0ff      [iv nt sb bl]
                                ( 10016+1 )  SPI=0|0ff      [iv nt sb bl]
                                ( 10016+2 )  SPI=0|0ff      [iv nt sb bl]
                                ( 10016+3 )  SPI=0|0ff      [iv nt sb bl]

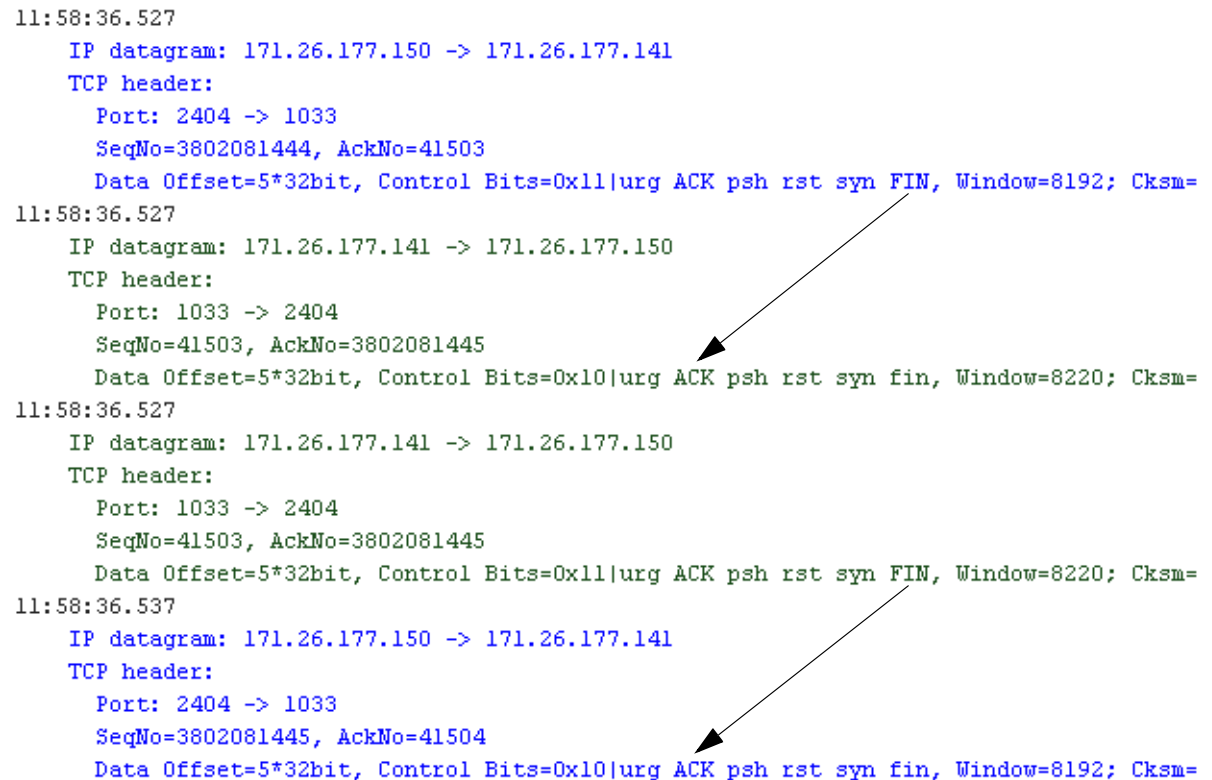
```

## COMPROTware:Testtool Network Monitoring

### Network Monitoring

#### ■ Typical connection shutdown

```
11:58:36.527
  IP datagram: 171.26.177.150 -> 171.26.177.141
  TCP header:
    Port: 2404 -> 1033
    SeqNo=3802081444, AckNo=41503
    Data Offset=5*32bit, Control Bits=0x11|urg ACK psh rst syn FIN, Window=8192; Cksm=
11:58:36.527
  IP datagram: 171.26.177.141 -> 171.26.177.150
  TCP header:
    Port: 1033 -> 2404
    SeqNo=41503, AckNo=3802081445
    Data Offset=5*32bit, Control Bits=0x10|urg ACK psh rst syn fin, Window=8220; Cksm=
11:58:36.527
  IP datagram: 171.26.177.141 -> 171.26.177.150
  TCP header:
    Port: 1033 -> 2404
    SeqNo=41503, AckNo=3802081445
    Data Offset=5*32bit, Control Bits=0x11|urg ACK psh rst syn FIN, Window=8220; Cksm=
11:58:36.537
  IP datagram: 171.26.177.150 -> 171.26.177.141
  TCP header:
    Port: 2404 -> 1033
    SeqNo=3802081445, AckNo=41504
    Data Offset=5*32bit, Control Bits=0x10|urg ACK psh rst syn fin, Window=8192; Cksm=
```



## COMPROTware:Testtool Message Log files

### Message Storage

- During simulation or monitoring, the entire protocol traffic is saved in the Message Storage
- The content of the Message Storage can be written into a file
- Always raw data is saved in the Message Log files, not processed data; for Offline Analysis all formatting options keeps available
- The protocol profile is also saved, so that when the file is loaded again, the right parameters are immediately set
- The current content of the Message Storage can be saved through *File -> Save as ...* ; don't forget to use the filename extension *.mlg* in the directory dialog

### Logging

- Additionally, during simulation or monitoring the protocol traffic can be logged in a file (*File -> Log to file ...* and *File -> Close Log file*); this file can be of any length, allowing extensive recordings (that last days) to be created, that would otherwise not fit in the Message Storage

### Access to Message Log files

- Through *File -> Open from ...* a Message Log file can be read
- A very long log file can be divided into smaller pieces through *Extra -> Split Log file*  
...

## COMPROTware:Testtool Offline Analysis

### Offline Analysis

- For postprocessing of the recorded protocol traffic
- The Offline Analysis **requires no license**
- Enables you to calmly go through the protocol traffic, to document and to draw the right conclusions
- Through *File -> Open from ...* a Message Log file can be retrieved again; with the cursor keys, the protocol traffic can be navigated, and using the shortcut keys the desired formatting options can be chosen

### Exporting to file

- *Extra -> Export to file ...* allows the content of the Message Storage to be exported to a file based on the previously chosen formatting options

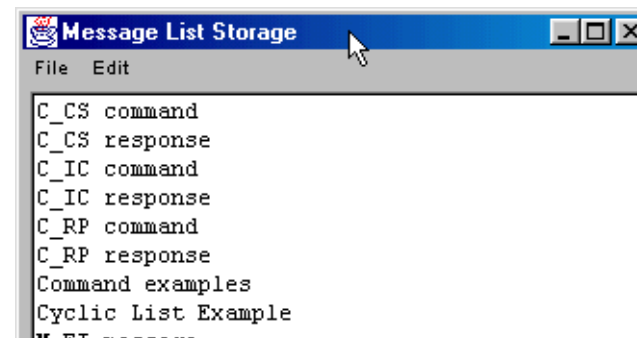
### Erase Message Storage

- *Clear* in the background menu deletes the Message Storage content

## COMPROTware:Testtool Message List Storage

### Message List Storage

- The Message List Storage contains all the message lists
- Through *Edit -> Message List Storage ...* a view of the Message List Storage can be opened

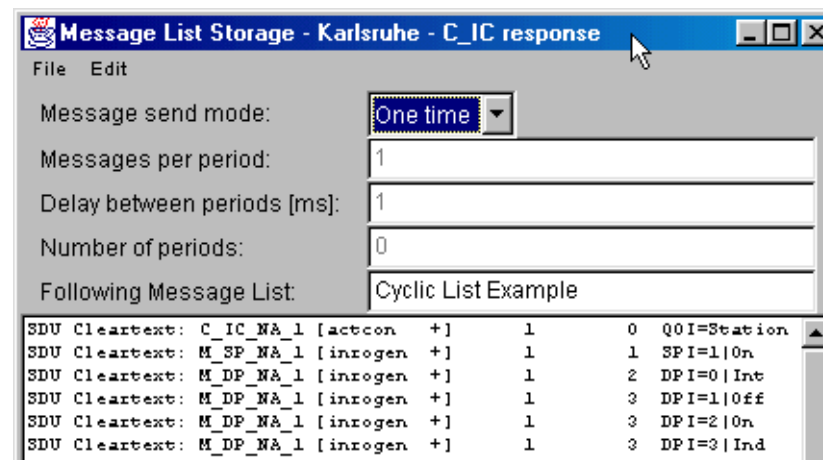


- The message lists names can be chosen freely, but must be unique; in distinct protocols, distinct message lists have a special meaning, e.g. C\_IC response contains the answer to a IEC 60870-5-101/-104 general interrogation
- In menu *File* through *New*, a new (empty) message list can be created, *Open from ...* imports message lists and *Save as ...* saves all message lists in Message List Storage in a file; the background menu and the menu *Edit* allow the editing, renaming, cutting, copying, pasting, sending and stopping of a message list; double-clicking on a list name opens a editing window for the list

## COMPROTware:Testtool Message Lists

### Message Lists

- A single message list can be stored in a file (*File -> Save as ...*), by *Edit* meta-messages can be edited, copied, pasted, and deleted; double-clicking on a meta-message opens the editing window
- A number of attributes and a list of meta-messages together setup a message list (the interpretation depends on the chosen protocol!)



- The attributes determine:
  - ▲ If the list is processed once (*One time*) or cyclically (*Cyclic*)
  - ▲ In cyclic processing: how many messages should be sent in a period, the delay between periods and the number of periods
  - ▲ The next message list



*COMPROTware:Testtool*  
*Specific Message Lists*

## Specific Message Lists

- Message list names reserved for distinct protocols:
  - ▲ IEC 60870-5-101/-104:
    - M\_EI message - initialization message (after connection establishment)
    - C\_IC command - general interrogation (after connection establishment)
    - C\_IC response - response to general interrogation
    - C\_RP response - response to reset process command
    - C\_CS response - response to synchronization command
  - ▲ IEC 60870-5-102/-103:
    - Init message - initialization message (after connection establishment)
    - GI request - general interrogation (after connection establishment)
    - GI response - response to general interrogation
    - ResetProcess response - response to reset process command
    - ClockSync response - response to synchronization command
  - ▲ DNP3 and DNP3 over LAN/WAN:
    - GI request - general interrogation (after connection establishment)
    - Reset Device Restarted - command to reset the Device Restarted bit
    - Timesync - response to time synchronization request
    - Read Class1 - response to indication class 1 data available
    - Read Class2 - response to indication class 2 data available
    - Read Class3 - response to indication class 3 data available

*COMPROTware:Testtool*  
*Specific Message Lists*

## Specific Message Lists - Continuation

▲ ABB RP570/571:

Init message - initialization message (after connection establishment)

Status Check request - general interrogation (after connection establishment)

Status Check response - response to general interrogation

Coldstart response - response to reset process command (FCOM 1)

VersionId response - response to inquiry firmware version (FCOM 4)

▲ MODBUS:

Cyclic Queries - cyclic list with queries

Response Read coils 1...8 - response; adjust value range

Response Read input discretes 1...8 - response; adjust value range

Response Read multiple registers 1...8 - response; adjust value range

Response Read multiple registers 1...8 - response; adjust value range

Response Read input registers 1...8 - response; adjust value range

Response Read exception status

Response Read general reference 1/2/3/4 - response; adjust value range

Response Read/write registers 1...8 - response; adjust value range

Response Read FIFO queue 8 - response; adjust value range

## COMPROTware:Testtool Meta-Messages

### Meta-Messages

- The message lists are composed of single meta-messages; the meta-messages can consist of:
  - ▲ A complete frame (Link Layer frame) as a hexstring (PDU transparent)
  - ▲ Payload data (Application Layer message) in clear text or as a hexstring (SDU clear text or SDU transparent)
  - ▲ User String
  - ▲ Delay
  - ▲ Abrupt, uncontrolled connection break-off (Abort)
  - ▲ Controlled shutdown of the connection (Shutdown)
- The clear text entry of messages offers a very fast and efficient way of entering message lists

Construct IEC 60870-5-101 message:

SDU Cleartext Class: 2 Options: Send/Confirm

Type Ident: 1 M\_SP\_NA\_1

COT: 20 inrogen

Common Address: 1 Information Object Address: 1

S/Q:  
Value: 0x81 Descriptor:

QDS:  
Value:

Date: 98 - 12 - 24 9 : 36 12000 ms

Duration: ms

Set Send Cancel

## COMPROTware:Testtool

### Sending of single messages

## Sending of single messages

- Besides the possibility of editing and sending message lists, single messages can easily be sent;  
therefore, *Send message...* is available in the background menu of the display windows

Send message - Karlsruhe

Construct IEC 60870-5-101 message:

SDU transparent Class: 2 Options: Send/Confirm

Type Ident: 0

COT: 0

Common Address: 0 Information Object Address: 0

Delay:  
Value: 0x00 Descriptor:

Hexstring:  
Value: 0x01 0x02 0x03

Date: 98 - 12 - 24 9 : 36 12000 ms

Duration: ms

Send Close

- The advantage here is that the entry text boxes can be changed rapidly and the message can be sent immediately

## COMPROTware:Testtool User Comments

### User Comments

- Through the window “Send Message“ it is possible to add user comments:

Send message - Unknown

Construct IEC 60870-5-104 message:

User String: [dropdown] Class: 1 Options: Send/Confirm

Type Ident: 9 M\_ME\_NA\_1

COT: 1 per/cyc Pos.Con. No Test

Common Address: 1 Information Object Address: 15

String:  
Value: Test 64.3 Descriptor: [empty]

QDS:  
Value: 0x01

Date: Edit Date&Time  
3 - 8 - 11 14 : 31 39569 ms

Duration: [empty] ms

Send Close

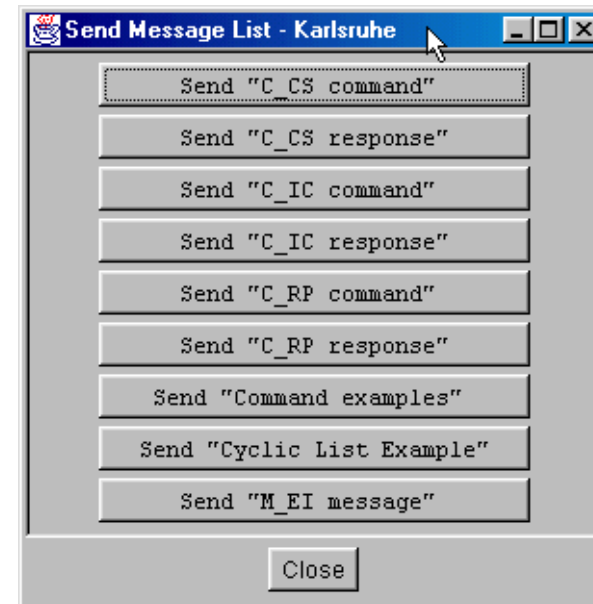
They are immediately added to the message storage

## COMPROTware:Testtool

### Sending Message Lists

## Sending Message Lists

- A single window allows it to easily activate message lists and to stop their execution; therefore, in the background select menu item *Send Message List ...*:



- Multiple message lists can be simultaneously executed; but every message list can only be processed once.
- Through *Send* the execution will begin and with *Stop* it will end

## COMPROTware:Testtool Preparing Configurations

### Preparing Configurations

- Configurations can be prepared comfortably right from your desk and then saved; in doing so an important step in the preparation for an on-site customer visit can be previously solved
- The following belong to a configuration: protocol profile, loaded **User Engine Classes**, general preferences and position/size of windows
- Save the configuration using *File -> Save Configuration as ...*, import the configuration using *File -> Open Configuration from ...*
- If **CPTT** must run several times on the desktop with the same configurations, there is a trick: start **CPTT**, set configuration and then save it using *File -> Save Configuration to user default*; the new user configuration has been saved; now you can run **CPTT** as often as wanted... all **CPTTs** have the same configuration

## COMPROTware:Testtool User Engine Classes

### User Engine Classes

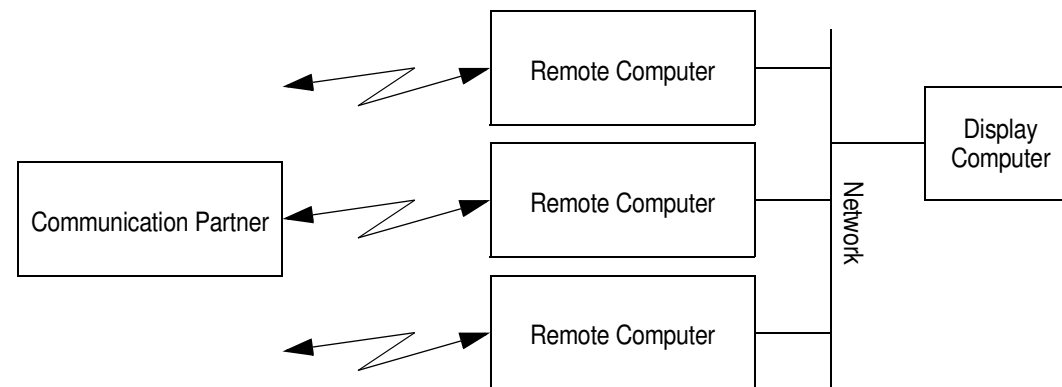
- Are plug-ins for *CPTT*, that are implemented in Java. In the Programmer's Guide the API of the *User Engine Classes* is described; a precondition for the development of *User Engine Classes* is an installed Java Development Kit
- Because Java is very similar to ANSI C and C++, it is easy to write *User Engine Classes*
- Java helps to protect programs against crashes from *User Engine Classes*
- An example for *User Engine Classes* will be delivered:  
IEC 60870-5-101/-104-Filetransfer
- Through *Edit -> User Engine Classes ...* you are able to choose and load *User Engine Classes*; if the path in the text field is cleared, then the class will be unloaded again
- Many other applications are imaginable:
  - ▲ File transfer and other Application Layer procedures
  - ▲ Recreating of device specific pseudo points
  - ▲ Controlling the basic functions of a device



## COMPROTware:Testtool Remote I/O Server

### Remote I/O Server

- The Remote I/O Server **separates the display from the reception of protocol traffic**. It enables data streams, which are not directly available on a **CPTT**-running display computer but can be monitored by remote computers, to be analyzed and displayed by **CPTT**. The **RIO Server** exchanges data with a communication partner through the remote computer's communication interface (serial interface, network, files). The data frames are handed off to **CPTT** through the network, to be analyzed and displayed.



- The **RIO Server** is an independent program, that operates separately from **CPTT** on another computer. While **CPTT** is limited to an MS Windows computer, the light weight **RIO Server** runs on many computer architectures (MS Windows, Linux, Solaris, ...).

*COMPROTware:Testtool*  
*Remote I/O Server*  
**RIO Server**

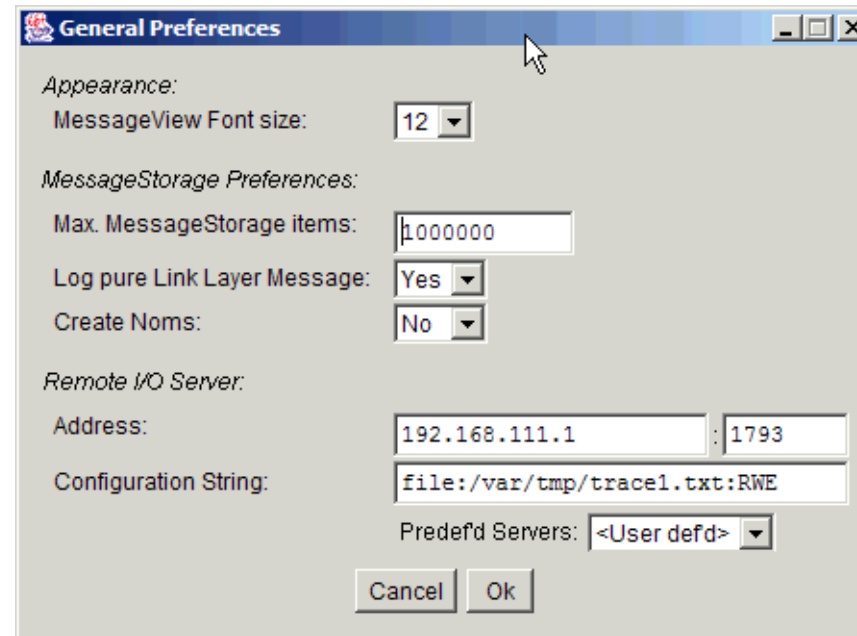
## **RIO Server**

- The executable version of **RIO Servers** for the required computer architecture can be found on the CD-ROM in the directory \CPRioServer:
  - ▲ CPRioServer\_iX86\_WIN32 for MS Windows
  - ▲ CPRioServer\_iX86\_Linux for PC Linux 2.x
  - ▲ CPRioServer\_sun4\_SunOS\_5 for Sun SPARC Solaris 7
- Copy the executable version on your target system.
- Start the **RIO Server** on the remote computer. So that the **RIO Server** is always available, it should be started when the computer is booted or the supervised application is started:
  - ▲ Under MS Windows: Put RIO Server in the registry
  - ▲ Under UNIX or Linux within the /etc/rc.d scripts
  - ▲ Or through a script related to the supervised application

COMPROTware:Testtool  
Remote I/O Server  
RIO Server

## RIO Client in CPTT

- The configuration of the *RIO Clients* in *CPTT* can be carried out through *Edit->General Preferences*:



- ▲ In this window, the address (computer name or IP address) of the *RIO Servers*, the TCP port no. and the configuration strings can be entered. The configuration string must be specified from the *RIO Servers* perspective (see below).

COMPROTware:Testtool  
Remote I/O Server  
RIO Client in CPTT

## RIO Client in CPTT

- The following applies for the inputs:
  - ▲ If there is a correct **RIO Server** address given, with immediate effect communication takes place through the **RIO Server**. If you like to continue without **RIO Server**, just clear the address.
  - ▲ If predefined settings (as CPTT configuration files) were read, then these can be selected from *Predef'd Servers* and accepted.
- “@Rio“ in the title of the window shows that the communication occurs indirectly through the **RIO Server**.
- Predefined **RIO Server** configuration can be read through *File->Open Configuration from*. The maintenance of these configurations must occur through an external editor.
- The configuration string have the following structure:  
<Medium>:<location>[;<location>]:<Format>
  - ▲ dblsr1:com1,9600,8n1;com2,9600,8n1:PPP  
From two serial interfaces (com1 and com2 each running on 9600 baud, 8 data-, 1 stop- and no parity bits) in format PPP (Point-to-Point Protocol) successive PPP-frames are caught and forwarded to **CPTT**.
  - ▲ file:/var/tmp/trace1.txt;/var/tmp/trace2.txt:RWE  
From a file, path and filename is “/var/tmp/trace1.txt“ or “/var/tmp/trace2.txt“, in format RWE all added data frames are consecutively read out and forwarded to **CPTT**.

## COMPROTware:Testtool

### We have observed...

### We have observed...

- Using *Edit -> Protocol Profiles ...* a protocol can be chosen from a protocol family, and its profile can be determined,
- the simulation as Controlling Station/Master or as Controlled Station/Slave is started through *Operate -> as Controlling/as Master* or *Operate -> as Controlled/as Slave*,
- the simulation can be finished again through *Operate -> Stop*,
- the display formatting of the protocol traffic can be changed using the *background menu* (right mouse button) -> *Formatting Options ...*
- the messages can be saved on a disk (*File -> Save as ...*) and be opened again (*File -> Open from ...*)
- The size of the Message Storage can be controlled using *Edit -> General Preference*, as well as whether all Link Layer information should be saved
- For long lasting tests, the protocol traffic can be saved directly into a file through *File -> Log to file ...*, logging will be stopped through *File -> Close Log file*
- This is specially useful during monitoring (*Operate -> Monitor* und *Operate -> Stop*)
- Large Message Log files can be split into smaller files using *Extra -> Split Log file ...*,
- the content of the Message Storage, with the selected formatting options, can be saved in a file by using *Extra -> Export to file*

*COMPROTware:Testtool*  
*We have observed...*

## **We have observed... - Continuation**

- Message lists are processed by *Edit -> Message List Storage ...*, **User Engine Classes** (plug-ins) can be added to **CPTT** through *Edit -> User Engine Class ...*
- For questions, *Help -> About* gives you the contact information to contact us!

**Real Thoughts GmbH**

Haid-und-Neu-Straße 7

76131 Karlsruhe

Germany

Fon +49-721-6276730, Fax +49-721-6276731

Website [www.realthoughts.de](http://www.realthoughts.de)

E-mail [info@realthoughts.de](mailto:info@realthoughts.de)