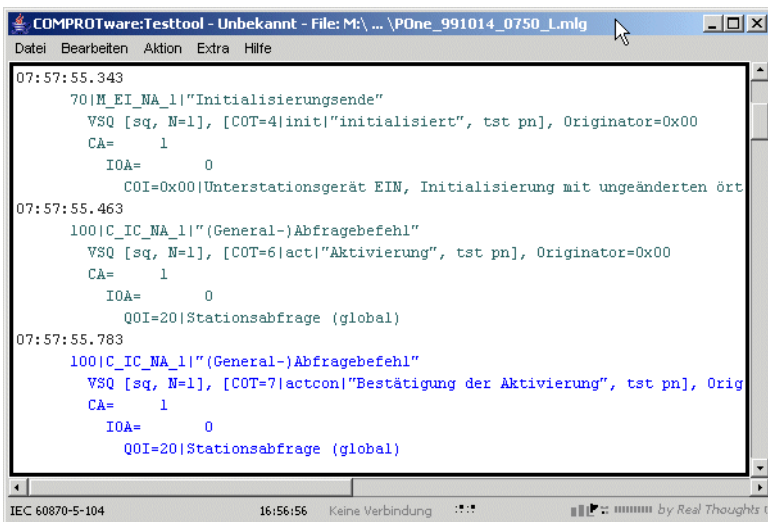


COMPROTware:Testtool

Integrated Testtool for Telecontrol Protocols

Since the mid 1990's, a radical change towards open and standardized protocols (like the IEC 608070-5 protocol family or DNP3) has been taking place. The main goal of these changes is to achieve a greater independence from suppliers; it should be possible to integrate devices and systems from arbitrary manufactures to form an optimal overall solution.

With **COMPROTware:Testtool**, *Real Thoughts* offers a supplier-independent test and analysis tool for protocols in this field.



```
COMPROTware:Testtool - Unbekannt - File: M:\... \POne_991014_0750_L.mlg
Datei Bearbeiten Aktion Extra Hilfe
07:57:55.343
70|M_EI_NA_1|"Initialisierungsende"
VSQ [sq, N=1], [COT=4|init|"initialisiert", tst pn], Originator=0x00
CA= 1
IOA= 0
COI=0x00|Unterstationsgerat EIN, Initialisierung mit ungeanderten ort
07:57:55.463
100|C_IC_NA_1|(General-)Abfragebefehl"
VSQ [sq, N=1], [COT=6|act|"Aktivierung", tst pn], Originator=0x00
CA= 1
IOA= 0
QOI=20|Stationsabfrage (global)
07:57:55.783
100|C_IC_NA_1|(General-)Abfragebefehl"
VSQ [sq, N=1], [COT=7|actcon|"Bestatigung der Aktivierung", tst pn], Orig
CA= 1
IOA= 0
QOI=20|Stationsabfrage (global)
IEC 60870-5-104 16:56:56 Keine Verbindung by Real Thoughts t
```

COMPROTware:Testtool can simulate the central system and the substation of a telecontrol system, as well as monitor the communication between both of them. You receive a continuous real-time representation with a flexible output of the different communication layers. During simulation/monitoring you can navigate in the communication traffic or search for text. Furthermore the filtering, formatting and output options can be changed on the fly. The protocol traffic can be logged to a file at any time. Single, periodic, or sequential message lists, containing correct or arbitrary erroneous information, can be sent. For the Offline Analysis, the log files can be used to draw right conclusions.

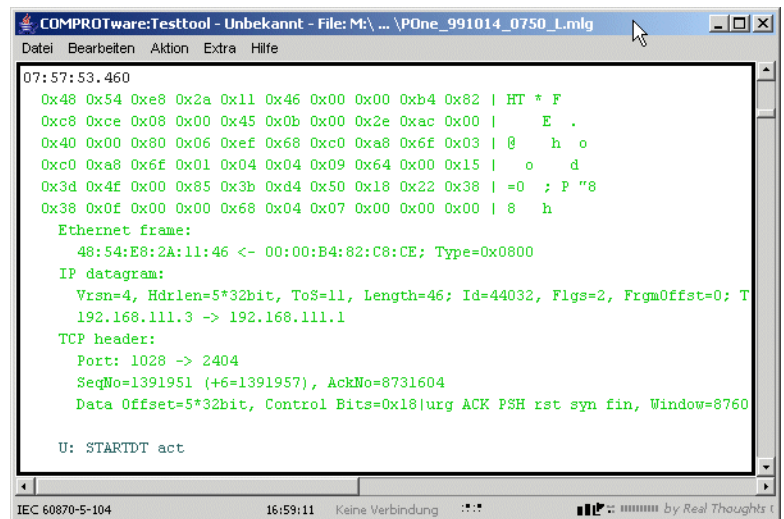
All protocol implementations are integrated into one Graphical User Interface. So you have the same state-of-the-art, windows-based look & feel to support the engineer's task of implementing and testing protocols, the delivery

of central systems/substations or the assurance of network availability.

- **COMPROTware:Testtool** runs on Microsoft Windows 2000/XP (32 bit).
- It is available for PC Linux (upon request).
- Available in the following languages: German, English, French, Italian and Spanish.

One of **COMPROTware:Testtool**'s basic concepts is to be open to the user:

- All configuration files are text-based and easy to understand, making it simple for different kinds of external software to generate configurations.
- With User Engine Classes (Plug-Ins) you are able to integrate all kinds of user-specific procedures into the testtool.
- The Offline Analysis of protocol communication requires no license. So you can install **COMPROTware:Testtool** on every desktop to do your daily work and also distribute it freely.
- **CPTT** integrates into your working environment by supporting imports of text files/Pcap files and by exporting data to HTML, MS Word, Wireshark or ASCII formats.
- With our Remote I/O Server it is possible to show communication captured by a remote system.



```
COMPROTware:Testtool - Unbekannt - File: M:\... \POne_991014_0750_L.mlg
Datei Bearbeiten Aktion Extra Hilfe
07:57:53.460
0x48 0x54 0xe8 0x2a 0x11 0x46 0x00 0x00 0xb4 0x82 | HT * F
0xc8 0xce 0x08 0x00 0x45 0x0b 0x00 0x2e 0xac 0x00 | E .
0x40 0x00 0x80 0x06 0xef 0x68 0xc0 0xa8 0x6f 0x03 | @ h o
0xc0 0xa8 0x6f 0x01 0x04 0x04 0x09 0x64 0x00 0x15 | o d
0x3d 0x4f 0x00 0x85 0x3b 0xd4 0x50 0x18 0x22 0x38 | =0 ; P "8
0x38 0x0f 0x00 0x00 0x68 0x04 0x07 0x00 0x00 0x00 | 8 h
Ethernet frame:
48:54:E8:2A:11:46 <- 00:00:B4:82:C8:CE; Type=0x0800
IP datagram:
Vrsn=4, Hdrlen=5*32bit, ToS=11, Length=46; Id=44032, Flgs=2, FrgmOffst=0; T
192.168.111.3 -> 192.168.111.1
TCP header:
Port: 1028 -> 2404
SeqNo=1391951 (+6=1391957), AckNo=8731604
Data Offset=5*32bit, Control Bits=0x18|urg ACK PSH rst syn fin, Window=8760
U: STARTDT act
IEC 60870-5-104 16:59:11 Keine Verbindung by Real Thoughts t
```

Specification

- Simulation of Master and Slave Station or Controlling and Controlled Station respectively.
 - ▲ Continuous real-time display of messages (bidirectional, time-stamped) and of interface/Protocol Engine events.
 - ▲ Interactive build up of messages to send (even during operation) from a logical down to a byte-by-byte level (to generate incorrect data).
 - ▲ Single or periodical sending of message lists, that can be linked to complex successions.
 - ▲ Log, storage and retrieval of protocol communication data for later analysis.
 - ▲ Special operation modes and parameter sets to support protocol stack development (e.g. to provoke stress situations, for load analysis or endurance tests, ...).
- Monitoring of communication traffic (for serial line- and network-based communication).
 - ▲ Continuous real-time display of messages (bidirectional, time-stamped) and of interface events.
 - ▲ Log, storage and retrieval of protocol communication data for later analysis.
- Offline Analysis of protocol communication.
 - ▲ Access to stored protocol communication data.
 - ▲ Export of protocol communication messages to text, HTML, MS Word and Pcap files.
 - ▲ Import from text and Pcap files.
- Graphical User Interface.
 - ▲ Configuration of the protocol profile and the protocol engine.
 - ▲ It is always possible to navigate within protocol traffic, to search for text, to change formatting options, and to set a filter (even during simulation

and monitoring).

- ▲ Protocol specific formatting that is easy to understand.
- Highly efficient implementation of simulation and message display/formatting, and a robust and intuitive Graphic User Interface.
- User Engine Class (plug-ins) for user specific processing functions.
- No additional interface hardware necessary.
- Implemented telecontrol protocols (march '09):
 - ▲ IEC 60870-5-101 Ed. 2.0
 - ▲ IEC 60870-5-102
 - ▲ IEC 60870-5-103
 - ▲ IEC 60870-5-104 Ed. 2.0
 - ▲ IEC 61850 MMS and GOOSE (monitoring only)
 - ▲ DNP3
 - ▲ DNP3 over LAN/WAN (TCP or UDP-based)
 - ▲ MODBUS (RTU and ASCII)
 - ▲ MODBUS TCP/IP
 - ▲ ABB RP570/571
 - ▲ ABB SPA-Bus
 - ▲ Landis&Gyr TG 809(The up-to-date list of supported protocols can be found on our website)

COMPROTware:Testtool is based on the idea that the implementation of an integrated testtool for telecontrol protocols brings advantages with the operation of every single protocol: if the user knows the tool and the handling of a single protocol, then he knows the handling for all other protocols, too. This concept allows the user to develop, implement and maintain RTUs, IEDs and control systems at a constant high level.

Contact us ...

Real Thoughts GmbH

Haid-und-Neu-Straße 7, 76131 Karlsruhe, Germany

Vanity-No. +49-700-REALTHOU

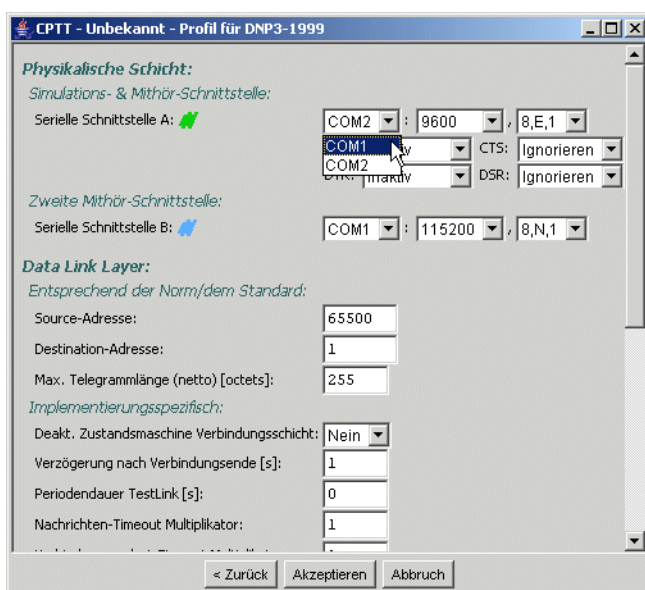
Fon +49-721-6276730

Fax +49-721-6276731

Website www.realthoughts.com

E-mail info@realthoughts.com

Price, specifications and availability may change without notice. Last update: March 2009



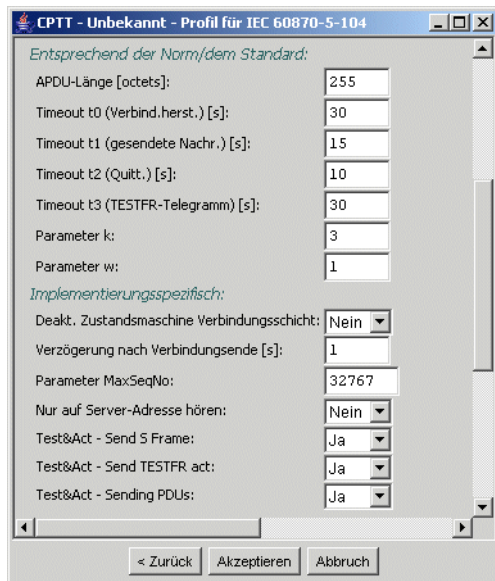
Specific Characteristics

■ General:

- ▲ **COMPROTware:Testtool (CPTT)** is an **integrated test and analysis tool** for telecontrol protocols. It supports the user in his tasks of guaranteeing the reliable functioning of the telecontrol system. It is a useful tool for the implementation of protocols, and assures an efficient start up of control systems and substations. It helps in the search for errors and problem analysis, and it makes the testing of new devices and protocols possible.
- ▲ The **CPTT Graphical User Interface** can be **intuitively and efficiently operated**, which makes handling the displayed communication traffic very easy.
- ▲ **CPTT** integrates **all protocols in a single tool**. Thus, making it fast to change from one protocol to another. This also means that the handling of one protocol is the same to that of every other protocol.
- ▲ **CPTT** is a **modern software**, whose core is in C++ and its Graphical User Interface is implemented in Java. Consequently, the advantages of both languages are conveniently combined, giving the user a **reliable, robust and trustworthy software**.
- ▲ **CPTT** is **future-proof** for it can be installed on Microsoft Windows 2000 and XP (as a 32 bit application). A PC Linux version can be provided upon request.
- ▲ **CPTT** is **multi-lingual**: German, English, French, Italian, and Spanish.
- ▲ **CPTT** can **run on a computer multiple times**. Each instance can communicate with an outstation through its own interface and with a different protocol/profile.
- ▲ The basic settings of the protocol profile, together with the Message List examples from the installation, and the training document, lead to a **promptly applicable simulation environment**. No time is lost on elaborate preparations, so work can begin right away.
- ▲ The **CPTT** configuration, consisting of the **general preferences**, the **protocol profile** and the **Message Lists**, can be **saved and retrieved**. Subsequently, one-time acquired equipment specific configurations can be rapidly restored.
- ▲ The installation (on MS Windows) is carried out by **Microsoft Windows Installer**, the installation and configuration service from Microsoft.

- ▲ Through the **integration of CPTT into Microsoft Windows**, you can work seamlessly with logfiles.
 - ▲ All configuration and description files (not the Message Log files) are text based and easy to understand. Due to the import/export functions, **CPTT** is ready for information exchange with others, and through the User Engine Classes upgradable at will. Therefore, **CPTT** is **open, flexible, and adaptable to collaborating with other applications**.
 - ▲ **CPTT** is **constantly being maintained and enhanced**. The **feedback from our users** combined with our own experience (from our projects) flows continuously into the tool, assuring its steady improvement. Technological updates in protocols and in operating systems are promptly implemented in **CPTT**.
 - ▲ We successively upgrade **CPTT** with new protocols and functions.
 - ▲ **CPTT** can be updated through our website <http://www.realthoughts.com/>, where the latest approved releases or the latest development snapshot can be downloaded.
- Implemented protocols:
- ▲ The complete IEC 60870-5-family: **IEC 60870-5-101 Ed. 2.0, -102, -103 and -104 Ed. 2.0**.
 - ▲ **IEC 61850 MMS and GOOSE (monitoring only)**.
 - ▲ **DNP3 and DNP3 over LAN/WAN**.
 - ▲ **MODBUS and MODBUS TCP/IP**.
 - ▲ **ABB RP570/571 and ABB SPA-Bus**.
 - ▲ **Landis&Gyr TG 809**.
(The up-to-date list of implemented protocols can be found on our website)
 - ▲ As soon as a protocol is licensed, the protocol **Raw data** is also available. Through this (pseudo) protocol, it is possible to monitor any character stream and to perform simulations based on hexadecimal character strings.
 - ▲ In the Graphical User Interface, a protocol from a protocol family is chosen, and the profile (consisting of parameters defined in the standard and extended by **CPTT** specific parameters) edited.
 - ▲ With all telecontrol protocols, it is possible to **simulate** the central station (**Master or Controlling Station**) or the substation (**Slave or Controlled Station**) of a protocol and to **monitor** the protocol traffic.

- About IEC 60870-5-101/-104:
 - ▲ The implementation is based on the standards **IEC 60870-5-101-Ed. 2.0, March 2003** and **IEC 60870-5-101-Ed. 1.0, December 2000**.
 - ▲ The IEC 60870-5-101 balanced and unbalanced transmission procedures, as well as all address and



field lengths, are implemented on the Link and Application Layer.

- ▲ **All type indications**, and **causes of transmissions** from IEC 60870-5-101 Ed. 2.0 and -104 are implemented.
 - ▲ **Input of structured addresses** and **display** of arbitrarily formattable structured addresses (with descriptive text for every level).
 - ▲ A file transfer implementation comes as an example of a User Engine Class.
- About IEC 60870-5-102:
 - ▲ The implementation is based on the standard **IEC 60870-5-102-Ed. 1.0, June 1996**.
 - About IEC 60870-5-103:
 - ▲ The implementation is based on the standard **IEC 60870-5-103-Ed. 1.0, December 1997**.
 - About IEC 61850:
 - ▲ The implementation is based on the standard **IEC 61850-8-1-Ed. 1.0, May 2004**.
 - ▲ The mappings MMS and GOOSE are implemented.
 - ▲ Monitoring is implemented, Master or Slave simulation are currently not available.
 - About DNP3 and DNP3 over LAN/WAN:
 - ▲ The implementation is based on **DNP3 Basic 4 Documents** plus **technical bulletins**.

- ▲ The **Link, Transport, and Application Layer** are **completely implemented**. **Unsolicited responses** are supported, all object groups/variations up to **level 3** are provided, DNP3 over LAN/WAN is implemented for **TCP/IP** as well as for **UDP/IP**.
- About MOBUS:
 - ▲ The implementation is based on the **MODICON-Standard-Documents, Rev. J. June 1996**.
 - ▲ Both transmission modes, RTU and ASCII, are implemented.
 - ▲ **Automatic generation of responses** to requests for substation simulation.
 - About MOBUS TCP/IP:
 - ▲ The implementation is based on the **MODBUS Messaging on TCP/IP Implementation Guide V1.0b, 2006**.
 - ▲ **Automatic generation of responses** to requests for substation simulation.
 - About ABB PR57x:
 - ▲ The implementation is based on the documents **RTU Protocol 570 and 571, Rev. 09, August 1998** and **Implementation of the RTU Protocol 570 and 571, Rev. 02, November 1998**.
 - About ABB SPA-Bus:
 - ▲ The implementation is based on the documents **SPA-Bus Communication Protocol V2.5, January 1996**.
 - ▲ **Automatic generation of responses** to requests for substation simulation.
 - About Landis&Gyr TG 809:
 - ▲ The implementation is based on the **TG US-Konventionsbeschreibung. 1995**.

Message View:

- ▲ Is the window **displaying the messages**, that are **stored in the Message Storage**.
- ▲ **The clear text** for messages is determined by the Message Decoder on the fly based on raw data. There are numerous formatting options and filters available:
 - The digital character stream is formatted in a **comprehensive manner**. Bits are transformed into clear text, and measured data is displayed in the correct representation. Thereby, **CPTT** always stays close to the guidelines (denominations, conventions, ...) of the implemented transmission protocols.
 - The color of the message identifies its origins. Erroneous data and error messages are displayed in red.
 - Messages can be shown as a **brief single-line** up to a **verbose multiline** display.
 - With or without **Link Layer**.
 - With or without **hexdump** of the message.
 - **Time stamp** with or without date.
 - With IEC 60870-5-104, IEC 61850, MODBUS TCP/IP or DNP3 over LAN/WAN (over TCP/IP or UDP/IP) the **Ethernet frame** and the **TCP/UDP header** can also be shown for monitored messages.
 - **Filter for substation addresses, message types, ...** reduce the displayed information to a user desired and manageable amount. **Wavy lines** mark the position in message view, where the **messages were filtered**.
- ▲ The cursor keys (arrow keys), page up and page down, Pos 1 and End allow **rapid navigation** through messages. When the end of the Message Storage is reached the view snaps on and shows henceforth the latest information. As soon as the cursor moves the view remains in the new position, while simulation or monitoring continues running.
- ▲ The **search** function can find any text in the Message Storage, forwards and backwards.

CPTT - Unbekannt - Statistiken	
Verbindung besteht/bestand seit:	05-08-03 15:27:56 (54sec)
Gesendet:	5521 Zeichen / 333 Puffer / 334 Telegramm(e) Durchsatz: akt: 1375 / durchschn: 1100 / max: 1705 28%
Empfangen:	2028 Zeichen / 338 Puffer / 334 Telegramm(e) Durchsatz: akt: 385 / durchschn: 407 / max: 649 8%
BREAK Signale:	0
Falsche Zeichen:	0 (0.0%)
= Falsche Parität:	0 (0.0%)
+ Falscher Zeichenrahmen:	0 (0.0%)
Falsche Telegramme:	0 (0.0%)
Wiederholte Telegramme:	0 (0.0%)
Verlorene Verbindungen:	0
MessageStorage:	2621 benutzt von 100000 (676.8K benutzt)
Startzeitpunkt CPTT:	05-08-03 15:23:40 (5min 10sec)
Verstrichene CPU-Zeit [s]:	10sec
Benutzer Phys.Speicher [bytes]:	268.9M
Benutzer Virt.Speicher [bytes]:	262.4M
Java Runtime Environment [bytes]:	848.9K frei von 2.5M
<input type="button" value="Rücksetzen"/> <input type="button" value="Schließen"/>	

IEC 60870-5-104-Nachricht erstellen:

SDU Cleartext | Klasse: 2 | Optionen: Send/Confirm

Type Ident: 100 | C_IC_NA_1

COT: 6 | act | Pos.Con. | No Test

Gemeinsame Adresse: 1 | act | Subjektadresse: 0

QOI: | 0x14 | deact | deactoon | actterm | reterm | retloc | file

QDS: | |

Datum: Edit Date&Time | 5 | 4 | 26 | 17 | 12 | 13724 | ms

Time valid | Calculate Day Of Week | Signal DST

Dauer: | ms

Buttons: Setzen, Sende, Abbruch

- ▲ Through **shortkeys**, all the important functions can be rapidly reached: change of the formatting options, change of message filter, sending of a single message, starting of simulation or monitoring, ...
- ▲ During simulation and monitoring, you get a **continuous real-time representation** with a **flexible display** of the **different communication layers**. The display can be changed at any time, even during simulation and monitoring.
- ▲ During simulation and monitoring, continuously determined **statistical values** are presented in a separate window: counter of sent/received characters, buffers and frames, current, average, and maximal throughput, erroneous characters and frames (also in percentages relative to the total), repeated telegrams and lost connections, ...

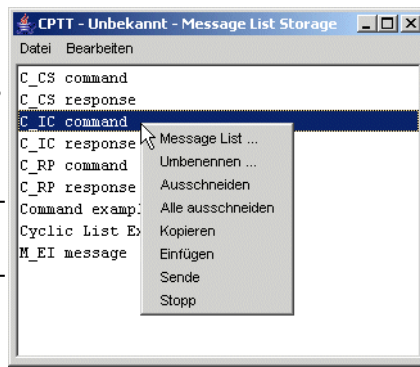
Message Storage:

- ▲ Is the central storage for all the sent/received messages. Each message has a time stamp with **milli-second accuracy**.
- ▲ The Message Storage stores up to **1.000.000 entries**. The size can be configured, to assure the safe functioning of the computer.
- ▲ Additionally, it **can be configured** so that **pure Link Layer frames will not be saved**, and thereby economize resources in terms of working memory and disk space.
- ▲ The Message Storage can be **saved in a Message Log file** and then be **retrieved** from it.

Message List:

- ▲ The Message List is a list of meta-messages. A meta-message can be a **Link Layer frame** or an **Application Layer message** (payload data). Furthermore, a meta message can cause a **delay**, the **controlled shutdown** of a connection or its **sudden abortion**, or trigger the **output of a user's text**.
- ▲ The number of Message Lists and the number of

messages per list is not limited. Each Message List is identified by its name.

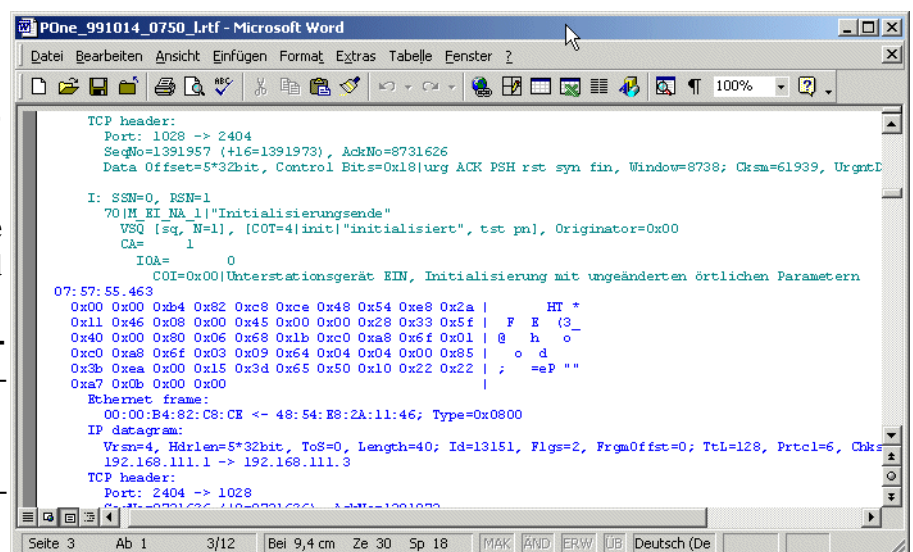


- ▲ Messages are entered as a hexadecimal character string or as **clear text, with interactive support at a logical level**. Any desired number of correct or erroneous messages may be created.
- ▲ A Message List can be **sent once or periodically**.
- ▲ Sequences of Message Lists for extensive test scheduling can be defined.
- ▲ **From the protocol traffic**, it is possible to **create Message Lists** for each direction separately. These lists may, for example, contain all data points from a station (function not available for all protocols).
- ▲ Message Lists can be stored individually or as a compilation in a file.
- ▲ With the help of common Window's functions like **copy&paste**, Message Lists or single messages can be easily constructed and duplicated.
- ▲ Message Lists and messages can be processed even while simulation is running.
- ▲ There are protocol specific reserved Message Lists names for particular tasks: after an IEC 60870-5-101 connection establishment, the Message List "C_IC command" will be searched for and executed. This Message List can contain, for example, the general interrogation message.
- ▲ Single message lists can be started or stopped by a button.
- Communication:
 - ▲ The communication occurs through serial interfaces (made available by the operating system) or the network stack depending on the protocol.
 - ▲ About serial interfaces: transfer speeds from **50-115200 Baud** are possible. All standard character framings (**bits/characters, stopbits, parity bits**) are allowed. **One interface** is required for **simulation** and for **monitoring** generally **two** (for specific protocols and under certain conditions one interface is sufficient). **No additional hardware** is necessary. USB-serial converters and PCMCIA cards are supported. Elaborate and efficient access to the serial interfaces guarantees that even with slow baud rates no discontinuous behavior appears and that also with high baud rates monitoring leads

to good results.

- ▲ About the network: The simulation is based on the **standard operating system network stack**. Monitoring is carried out through Windows drivers or through WinPcap. The available network adapters are used for network access.
- Simulation:
 - ▲ **CPTT** always **simulates both central system and substation** of a telecontrol protocol (not for IEC 61850).
 - ▲ The simulation is based on the implemented **Protocol Engine with procedures for Link and Application Layer**. The Protocol Engine establishes a correct connection, exchanges frames reliably, etc.
 - ▲ The Protocol Engine also comprises general interrogation, command responses, and time synchronization.
 - ▲ You get a **continuous real-time representation** of messages (bidirectional, time-stamped) and of interface and Protocol Engine events (like parity error, bad characters, bad checksum, unexpected frames, bad frame sequence bit, ...).
 - ▲ During simulation, **message traffic** can be continuously stored in a **Message Log file, as long as desired**.
- Monitoring:
 - ▲ Possible with **serial communication** and with **network-based communication** (TCP/IP oder UDP/IP, e.g. with IEC 60870-5-104, IEC 61850, MODBUS TCP/IP or DNP3 over LAN/WAN). **GRPP/PPP-Communication** to a **GSM modem** can also be monitored and displayed.
 - ▲ You get a **continuous real-time representation** of messages (bidirectional, time-stamped) and of interface events (like parity error, bad characters, bad checksum, ...).
 - ▲ During monitoring, **message traffic** can be continuously stored in a **Message Log file, as long as desired**.
 - ▲ For **Network Monitoring** it is necessary to be connected to a **hub** or a **suitably configured switch** on the network path between the participants.

- **Offline Analysis:**
 - ▲ For Offline Analysis, message traffic retrieved from a Message Log file is formatted by the protocol's Message Decoder and displayed in the Message View. The analysis allows you to draw the right conclusions and to gather the necessary documentation.
 - ▲ Results can simply be exported from **CPTT** into the documentation.
 - ▲ **Offline Analysis requires no license.**
- **Protocol Engine:**
 - ▲ The Protocol Engine **implements a telecontrol protocol**, and thereby defines action and reaction. It assures a proper connection build up, a reliable frame exchange, the repetition of a frame if necessary, etc.
 - ▲ Besides the **protocol profile parameters**, there are **implementation-specific parameters** available, that support the protocol developer and the project engineer during testing:
 - **Stress scenarios** can be setup.
 - **Error situation** are reproducible.
 - **Load analysis and endurance tests** are feasible.
 - The Protocol Engine can be **completely deactivated**, in order to perform all actions manually or to implement a protocol as a User Engine Class.
- **Message Log files:**
 - ▲ The contents of the Message Storage or the continuous message traffic during simulation/monitoring, can be stored in a Message Log file.
 - ▲ **Contains raw data** but no processed data. For its display, the raw data is **decoded according to the selected protocol and the view formatting options**.
 - ▲ Each message gets a **time stamp with millisecond accuracy**. Text messages describe **interface and Protocol Engine events** and elucidate processes in **CPTT**.
 - ▲ The **protocol profile is included in the Message Log file** and will be recovered when the file is retrieved. A saved file will **immediately after retrieval be displayed comprehensibly on screen** (nevertheless, the profile can be altered).
 - ▲ **Size is only limited by the available disk space**. Therefore, a recording duration of days or weeks is no problem.
- **Import/Export:**
 - ▲ Import and export make **CPTT** open to other programs, assist in performing **advanced analysis**, and support the user by the **documentation of its results**.
 - ▲ It is possible to **import** line oriented hexdumps from a **text file** or network based communication from **Wireshark (Pcap) files**.
 - ▲ The content of the Message Storage can be exported as an **HTML** or as a **text file**, to a **MS Word (RTF file)** or to an **Wireshark (Pcap) file**.
- **User Engine Classes:**
 - ▲ The objective of User Engine Classes is the creation and integration of advanced functions in the form of **plug-ins** to **CPTT**. It is possible to integrate all kinds of **user specific procedures** to **CPTT**:
 - You can program **monitoring functions**, to verify communication (e.g. the integrity of indication changes).
 - Or, you implement your **own Graphical User Interface**, that emulates **device specific details** (e.g. pseudo data points with system information).
 - **Application Layer procedures** (e.g. **file transfer**) are also feasible as User Engine Classes.
 - ▲ The Java environment makes sure that the User Engine Classes **won't cause CPTT** to crash.
 - ▲ In the document "Programmer's Guide", you can find a description of the interface between the User



Engine Class and **CPTT**.

- ▲ Different examples for User Engine Classes are included, such as **IEC 60870-5-101/104 File transfer**.
- Remote I/O Server:
 - ▲ The Remote I/O Server **separates the display from the reception of message 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**. Only a network connection is required between a display and receiving computer.
 - ▲ Source for the message traffic can be a **serial interface, a network adapter or a file**.
 - ▲ The Remote I/O Server is an independent program, available for various computer architectures, **including MS Windows, PC Linux, Solaris,** For the integration in a customer device we additionally deliver the sourcecode.
 - ▲ A typical application is remote surveillance of communication systems or the integration of the Remote I/O Server in a device (RTU, IED, gateway, ...) for the diagnosis of the message traffic on its interfaces.
 - ▲ **In CPTT the option "Remote I/O Client" is licensed separately.**
- Documentation:
 - ▲ **Multi-lingual printed installation instructions** in English and German.
 - ▲ **Training documents** exist in English and German in electronic form.
- Licensing:
 - ▲ **A license is necessary for the simulation of central or substations** of a telecontrol protocol, or for the **monitoring of message traffic** between two stations.
 - ▲ **Each protocol requires a separate license.**
 - ▲ **For the Offline Analysis no license is required.**
 - ▲ **CPTT can be freely installed on all computer systems and be freely distributed to others.** As explained previously, a license is required only for protocol simulation and monitoring.
 - ▲ The license(s) are **stored in a dongle**:
 - Dongles are available for various interfaces: **printer port or USB.**
 - The dongles are programmed **for each customer individually**. The programming can sub-

sequently be **changed on site**, using an **update file**. Therefore, it is very fast and easy to add an additional license to a dongle, making it unnecessary to return it to us.

1) Depends on the computer systems capacity.

2) Network Monitoring is not possible with all network adapters. The completeness of the monitored protocol traffic can not be guaranteed for all network adapters.

Real Thoughts GmbH

Haid-und-Neu-Straße 7

76131 Karlsruhe

Germany

Vanity-No. +49-700-REALTHOU

Fon +49-721-6276730

Fax +49-721-6276731

Website www.realthoughts.com

E-mail info@realthoughts.com

Price, specifications and availability may change without notice. Last updated: March 2009