

Table of Contents

1 About this document.....	1
1.1 Legend.....	1
1.2 Links.....	2
1.3 Compatibility.....	2
1.4 Files and Directories.....	2
1.4.1 Important DEWESoft® 7 Directories.....	2
1.4.1.1 DEWESoft® Measurement Unit [recommended].....	2
1.4.1.2 Windows Standard.....	2
1.5 Licensing.....	3
1.5.1 Requesting an Evaluation license.....	3
1.5.2 Activating the Evaluation license.....	3
1.6 Plug-in Installation.....	3
1.7 Input Fields.....	4
1.7.1 Input Confirmation.....	4
1.7.2 Input Warnings/Errors.....	4
2 Protocol.....	5
2.1 Basics.....	5
2.1.1 Compatibility Specification.....	5
2.1.2 Connection establishment.....	5
2.2 Command Protocol Specification.....	5
2.2.1 Welcome Message.....	6
2.2.2 Protocol Version.....	6
2.2.3 Get Sample Rate.....	7
2.2.4 Get Sample Rates.....	7
2.2.5 Set Sample Rate.....	7
2.2.6 Get Available Channels.....	7
2.2.6.1 Channel List.....	8
2.2.6.2 Unique Id.....	8
2.2.7 Start Transfer.....	9
2.2.8 Stop Transfer.....	9
2.2.9 Disconnect.....	9
2.2.10 Exit.....	10
2.3 Data Protocol Specification.....	10
2.3.1 Header Data.....	10
2.3.2 Channel Data.....	10
2.3.3 Trailing Data.....	11
2.3.4 Example.....	12
2.3.5 Example-Byte Level.....	12
3 Configuration.....	14
3.1 Hardware Setup.....	14
3.1.1 OS Commands.....	14
3.1.1.1 Server Command.....	14
3.1.1.2 Exit Command.....	15
3.1.2 Command Connection.....	15
Connection Timeout.....	15
Read Timeout.....	15
3.1.3 Data Connection.....	15
3.1.4 Ignore Errors on Start of Acquisition.....	16
3.1.5 Log files.....	16
3.1.5.1 Log levels.....	16
3.2 Channel Setup.....	17
3.2.1 Connection.....	17
3.2.2 First start.....	17
3.2.3 Channel Setup Grid.....	18
3.2.3.1 Header pop-up.....	19

3.2.3.2Multi select.....	19
4Version History.....	20

1 About this document

This is the user documentation for DEWESoft® TCP/IP Binary Client Plugin Version 1.1.5.

The plugin can be used to get data from another program (called *Server*) which implements the TCP/IP protocol specified in chapter 2 Protocol (on page 5): e.g. LabVIEW™.

The data that the Server sends, will be added to asynchronous DEWESoft® channels.

1.1 Legend

The following symbols and formats will be used throughout the document.

IMPORTANT



Gives you an important information about a subject.
Please read carefully!

HINT



Gives you a hint or provides additional information about a subject.

EXAMPLE



Gives you an example of a specific subject.

Example	Meaning	Description
Cancel	Button	a button that you can click
<i>File</i>	Menu Item	a menu item, will open a sub menu or a dialog
<i>Times New Roman</i>	List Item	an item in a list (or tree) that you can select
Events	Tab Sheet	a tab sheet that you can select
C:\Program Files\OpenOffice.org 3\readme.txt	File Path and Name	a file name or path
<i>Windows Key</i>	a term	any kind of term (maybe also compound)

Table 1: Layout formats used in the documentation

1.2 Links



DEWESoft® homepage

<http://www.dewesoft.com>

you can download DEWESoft® plugins when you go to: [Support](#) - [Downloads](#) - [Plugins](#)

1.3 Compatibility

The plugin is compatible with DEWESoft® X.

It has been tested with the version DEWESoft® XI on Windows 7 (64-bit).

1.4 Files and Directories

The actual location of the directories on your computer may vary dependant on your computer's locale settings and the settings you chose when installing DEWESoft®.

1.4.1 Important DEWESoft® 7 Directories

1.4.1.1 DEWESoft® Measurement Unit [recommended]

Directory name	Explanation	Default path
Bin	contains DEWESoft.exe	D:\DEWESoft7\Bin\V7_1
Addons	.dll files for plugins must be copied into this directory	D:\DEWESoft7\Bin\V7_1\Addons
Data	this is where DEWESoft® will store your measurement data	D:\DEWESoft7\Data
Setups	this is where your DEWESoft® setup files will be stored	D:\DEWESoft7\Setups
System	this is where DEWESoft® project files are stored	D:\DEWESoft7\System\V7_1
Log	this is where DEWESoft® will store log files	D:\DEWESoft7\System\V7_1\Logs

The paths may be different depending on your DEWESoft® version.

1.4.1.2 Windows Standard

Directory name	Default path
Bin	language dependant directory: C:\Programme\DEWESoft7\Bin\V7_1
Addons	language dependant directory: C:\Programme\DEWESoft7\Bin\V7_1\Addons
Data	user/language dependant directory: C:\Dokumente und Einstellungen\All Users\Dokumente\DEWESoft7\Data
Setups	user/language dependant directory: C:\Dokumente und Einstellungen\All Users\Dokumente\DEWESoft7\Setups
System	user/language dependant directory: C:\Dokumente und Einstellungen\All Users\Dokumente\DEWESoft7\System\V7_1
Log	user/language dependant directory: C:\Dokumente und Einstellungen\All Users\Dokumente\DEWESoft7\System\V7_1\Logs

The paths may be different depending on your DEWESoft® version.

1.5 Licensing

The plugin requires a valid DEWESoft® license.
To test the plugin you can use an *Evaluation license*.

1.5.1 Requesting an Evaluation license

You can request an *Evaluation license* from our homepage:

<http://www.dewesoft.com/registration>

- (1) Click on **Evaluation license**
- (2) Fill out all the required fields
- (3) Click the **Request license** button

The screenshot shows a web browser window with the URL www.dewesoft.com/registration#evaluation-license. On the left, there are three buttons: 'Online registration', 'Offline registration', and 'Evaluation license' (which is highlighted). The main content area is titled 'Evaluation license'. Below the title, there is a text block explaining that a 30-day evaluation license can be requested by filling out a form. The form contains the following fields: Name (text input), Company (text input), Country (dropdown menu showing 'Austria'), Email (text input), and Application (dropdown menu). A 'Request license' button is located at the bottom right of the form.

Illustration 1: Request Evaluation License

1.5.2 Activating the Evaluation license

When you have received your trial licence key, open DEWESoft®, go to **Settings - Hardware Setup...**, select the **Registration** tab sheet and enter the license code (if you already have other licenses, you may need to click the **Create** button).

Now enter the license code and click the **Register online** button.

Then your new license key will show up in the list and should have the *Status Valid*.

The screenshot shows the 'Hardware setup' dialog box with the 'Registration' tab selected. The 'Enter new license' section is active, showing a text input field with the license key 'DW7-7QAA-9SW4-26A2'. Below the input field, there are buttons for 'Register online', 'Register offline', and 'Cancel'. A warning message states: 'Warning: License will be valid ONLY for the measurement hardware (or computer MAC) and options which are chosen at this moment. Please make sure that you choose all the devices, options and plugins which you will use.' The 'Registration status' at the bottom is 'DEMO'.

Illustration 2: Enter license key

The screenshot shows the 'Hardware setup' dialog box with the 'Registration' tab selected. The 'Existing license(s)' section is active, displaying a table with the following data:

Used	License key	Software version and options	Status
✓	DW7-7QAA-9SW4-26A2	TRIAL	Valid

Buttons for 'Create', 'Delete', and 'Update' are located to the left of the table. The 'Registration status' at the bottom is 'TRIAL (03.09.2011)'.

Illustration 3: Valid trial license

1.6 Plug-in Installation

Simply copy the file `TcpIpBinaryClient.dll` into the Addons folder of your DEWESoft® installation.
(e.g. `D:\DEWESoft7\Bin\V7_1\Addons\`) and then start DEWESoft®).

HINT

When you are using Windows® 7, then you must click the **Register plugins** button (at the bottom of the Hardware setup dialogue) once and restart DEWESoft® before the plugin shows up in the list of available plugins.

1.7 Input Fields

1.7.1 Input Confirmation

When you change the value of an input field, the background colour of the input field will turn yellow to indicate that you have changed something and that this change has not been confirmed yet.

Your input will automatically be confirmed when set the focus to another input field (i.e. by clicking with the mouse or by pressing the key).

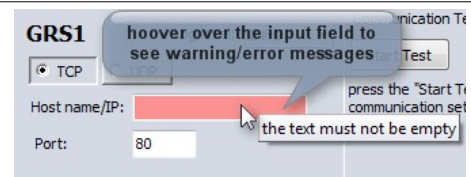
You can also press to manually confirm your change.

After the input has been confirmed the background colour of the input field will be white again (or **red/orange**, when there are **errors/warnings**).

1.7.2 Input Warnings/Errors

An invalid input may cause a **warning** and **error**.

Warnings will be highlighted in **orange**, **errors** in **red**. When you hover over the input field you will see a hint with a detailed description of what is wrong:



2 Protocol



This section provides detailed specifications of the protocol used for communication between the plugin and the *Server*.

2.1 Basics

The DEWESoft® plugin will act as a TCP/IP client that connects to the *Server* (e.g. LabView™ program). The plugin can send commands to the server and the server must respond within a given timeout to those commands.

The plugin can also request from the server to start sending data. In this case the plugin will listen to a specified port and the *Server* (e.g. LabView™ program) must send the data to this port.

So the plugin will use 2 TCP/IP connections:

-  the *Command Connection* (plugin connects to the Server)
-  the *Data Connection* (the Server connects to the plugin)

2.1.1 Compatibility Specification




The compatibility specification is strict: i.e. the client and the server must have the same protocol version (see also: 2.2.2 Protocol Version on page 6).

2.1.2 Connection establishment

First the Server must be running and listening to the *Command Port*.

Then the DEWESoft® plugin will open a connection to the *Command Port* and expects a valid Welcome Message (see 2.2.1 Welcome Message on page 6). Next, the plugin will request the Server's protocol version (see 2.2.2 Protocol Version on page 6).

The plugin must also know the following things:

-  which *Sample Rates* are supported (see 2.2.4 Get Sample Rates on page 7)
-  what is the current *Sample Rate* (see 2.2.3 Get Sample Rate on page 7)
-  which *Channels* does the server provide. (2.2.6 Get Available Channels on page 7)

When the plugin needs to get data from the Server (i.e. when you go to *Ch. Setup* or *Measure mode* in DEWESoft®), the plugin will start to listen on the specified *Data Port* and send the *Start Transfer* command (see 2.2.7 Start Transfer on page 9) to the server.

Then the server sends the data for all it's channels in the specified format (see 2.3 Data Protocol Specification on page 10).

When the measurement is stopped in DEWESoft®, the plugin will send the Stop Transfer command (2.2.8 Stop Transfer on page 9).

Finally when the plugin stops it's work (i.e. DEWESoft® is closed) it will send the *Disconnect* command (see 2.2.9 Disconnect on page 9) to the server to indicate that the Server can stop to listen for commands or send data.

2.2 Command Protocol Specification

The command protocol is in plain text, to make it easy to debug and test (i.e. plain text can be sent/received via simple tools, such as `telnet`). Characters should be encoded in UTF-8 format (i.e. this only affects characters with an ordinal number > 127).


The commands are case-insensitive: e.g. the following 3 commands will all work:


-  `getProtocolVersion`


 GETPROTOCOLVERSION


 gEtPrOtOcOlVeRsIoN

The server-response must always start with the plus (+) character:

 successful responses:

 single line responses usually start with +OK (except for the the Welcome Message) optionally followed by a space character and textual data

 multi-line responses start with a +STX (followed by a data id) line , followed by the data lines and end with an +ETX line: e.g. see 2.2.4 Get Sample Rates
fields in the multiline response will be separated with the TAB character

 erroneous responses start with +ERR followed by an optional error message describing the problem in detail

Each command must end with the following newline suffix: carriage-return line-feed (hexadecimal notation: 0x0D 0x0A, \r\n)– see also Table 2: Representation of non-printable characters.

Numbers never contain a thousand-separator. The decimal-separator is a dot (.).

The following characters will be used to represent non-printable characters in the protocol specification:

Representation	Meaning
→	Represents the TAB character (dec: 09, hex: 0x09)
↓	Represents a line-break sequence of carriage-return and line-feed (dec: 13 10, hex 0x0D 0x0A, \r\n)

Table 2: Representation of non-printable characters

2.2.1 Welcome Message

When a client opens a connection to the command-port of the TCP/IP server, the server must respond with a welcome message in the following format:

+CONNECTED to: **ServerId**

where the variable **ServerId** can be an arbitrary string, describing the server: e.g. LabVIEW

Full example:

+CONNECTED to: LabVIEW

2.2.2 Protocol Version

The client can request the protocol version of the server:

Entity	Format	Examples
Client Request	getProtocolVersion↓	
Server Response	+OK ProtocolVersion ↓	+OK 1.0↓ +OK 2.12↓ see: 2.1.1 Compatibility Specification on page 5

2.2.3 Get Sample Rate

Returns the current sample rate in Hz from the server.

Entity	Format	Examples
Client Request	<code>getSampleRate↓</code>	
Server Response	<code>+OK SampleRate↓</code>	<code>+OK 10000↓</code> <code>+OK 0.1↓</code>

2.2.4 Get Sample Rates

Returns a list of available sample rates in Hz from the server.

Entity	Format	Examples
Client Request	<code>getSampleRates↓</code>	
Server Response	<code>+STX SampleRates↓</code> <code>SampleRates↓</code> <code>+ETX↓</code>	<code>+STX SampleRates↓</code> <code>SR →0.1↓</code> <code>SR →1↓</code> <code>SR →10↓</code> <code>SR →20↓</code> <code>+ETX↓</code>

2.2.5 Set Sample Rate

Allows the client to set the sample rate (in Hz) of the server.

Entity	Format	Examples
Client Request	<code>setSampleRate SampleRate↓</code>	<code>SetSampeRate 10↓</code>
Server Response	<code>+OK setSampleRate to SampleRate Hz↓</code> Note: the text after +OK is optional, but it is strongly recommended that the server uses this text, as it makes debugging much easier)	<code>+OK setSampleRate to 10 Hz↓</code>

2.2.6 Get Available Channels

The client can request information about the available channels on the server with this command.

Entity	Format	Example
Client Request	<code>getChannels↓</code>	
Server Response	<code>+STX Channels↓</code> <code>ChannelList↓</code> <code>+ETX↓</code>	<code>+STX Channels↓</code> <code>CH →0 →12 →Pressure →Bar →Sync →16iu↓</code> <code>CH →1 →8 →In Temp. →°C →Sync →32fs↓</code> <code>+ETX↓</code> for specification of ChannelList see below

2.2.6.1 Channel List

Data	Description	Data Type	Examples
CH	Fixed string		
Number	Consecutive channel number in this list, starting with 0	Integer	0, 1, 2, ...
UniqueID	A unique ID representing the channel: see 2.2.6.2 Unique Id below	Text	
Name	Channel name	Text	Pressure, In Temp., AI0, ...
Unit	Physical measurement unit of the channel data	Text	bar, kg, km/h, °C, ..
Sync Type	Type of synchronization. Currently only <i>Sync</i> is supported in the future maybe <i>Async</i> , <i>SingleValue</i> , ..	Text	Sync
Data Type	The data type of the channel in the format ¹ : Bits NumberType Signedness <ul style="list-style-type: none"> Bits: the number of bits that are used to store this data type (16, 32, 64) NumberType: the number type: i (integer), f (Float) Signedness: s (signed), u (unsigned) valid supported types: 32is, 32fs 	Text	32is, 32fs

2.2.6.2 Unique Id

The server has to make sure that this text is unique (2 channels MUST NOT have the same Id) and that it does not change.

EXAMPLE 1



Even if a user renames a channel on the server, DEWESoft® could still keep all the visual controls associated with the same channel – only the name has changed.

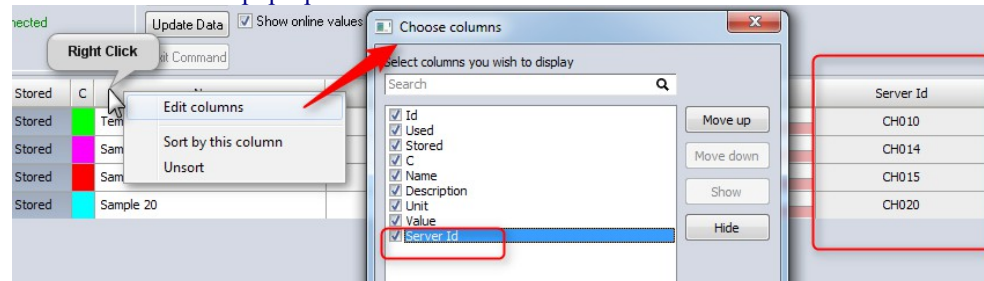
As another example: if the server chooses to use the channel name also as unique ID, it means that the names of all channels must be unique. And if a user ever renames a channel, the visual controls that have been assigned to this channel will not be assigned any longer (i.e. the user must now assign the new channel to the visual controls).

¹ The spaces have been inserted only for readability – the are not part of the data string

The unique ID will be used to sort the items in the Channel Setup Grid (see also 3.2.3 Channel Setup Grid on page 18). Note. That the sorting is smart enough to recognize positive integer numbers: e.g. Ch1 will be sorted before Ch10, etc.

HINT

You can show the *Server-Id* column in the Channel Setup Grid (see also 3.2.3 Channel Setup Grid on page 18). Right click on any of the header cells in the grid and then select *Edit columns* from the pop-up menu. Then activate the *Server Id* column.



2.2.7 Start Transfer

Requests to start the data transfer from the server. The server must connect to the given **PortNumber** and start to stream it's data in the binary format specified in 2.3 Data Protocol Specification (on page 10).

Note: the timeout used for this command is Connection Timeout (see 3.1.2 Command Connection on page 15), because it may take longer for the server application to open the connection (as the normal Read Timeout).

The data must be sent in the order specified by the channel list (see 2.2.6.1 Channel List on page 8).

Entity	Format	Example
Client Request	<code>StartTransfer PortNumber↓</code>	<code>StartTransfer 8090</code>
Server Response	<code>+OK streaming data to PortNumber</code> Note: the text after +OK is optional, but it is strongly recommended that the server uses this text, as it makes debugging much easier)	<code>+OK streaming data to port 8090</code>

2.2.8 Stop Transfer

Tells the server to stop the data transfer (which has been started via the Start Transfer command). The server must close the data-port (which has been specified in the `StartTransfer` command).

Entity	Format	Example
Client Request	<code>StopTransfer</code>	<code>StopTransfer</code>
Server Response	<code>+OK</code>	<code>+OK</code>

2.2.9 Disconnect

Will be sent by the DEWESoft® plugin right before DEWESoft® will close the TCP/IP connections to the server.

The server should respond with a final positive response and can then close it's TCP/IP connections.

Note that this command is optional and it might not be sent at all (e.g. in the very unlikely event that DEWESoft® crashes).

Entity	Format	Example
Client Request	<code>Disconnect</code>	<code>Disconnect</code>
Server Response	<code>+OK</code>	<code>+OK</code>

2.2.10 Exit

Will be sent by the DEWESoft® plugin right before the plugin will be closed: this is the case when DEWESoft® is closed but also when the user deactivates the plugin in Hardware setup.

In this case the plugin will not wait for any response from the server – it's just a fire and forget command. The TCP/IP connection will be closed immediately after sending the command.

Note that this command is optional and it might not be sent at all (e.g. in the very unlikely event that DEWESoft® crashes).

Entity	Format	Example
Client Request	<code>Exit</code>	<code>Exit</code>
Server Response	<code>–</code>	<code>–</code>

2.3 Data Protocol Specification

The binary data is specified as a subset of the DEWE-NET option *Binary data format* (currently only synchronous channels are supported). See also 2.3.4 Example on page 12 for a simple example.

The byte-endianess is Windows standard (Little-Endian).

2.3.1 Header Data

Every data packet starts with the given header:

Offset [bytes]	Length [bytes]	Data type	Description	Comment
0	8		Start packet string	0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07
8	4	32iu	Packet size	Size in bytes without Start and Stop packet string
12	4	32iu	Packet type	Always 0 for data packets
16	4	32iu	Samples in packet	Number of synchronous samples per channel
20	8	64iu	Samples acquired so far	The total number of samples that have been acquired by the Server since the <i>Start Transfer</i> command has been sent (see 2.2.7 Start Transfer on page 9) i.e. the first packet will have the value 0, the second packet 0+Samples in the 1 st packet, etc.
28	8	64fu	relative time	Number of seconds since start of the acquisition

Off = 36 bytes

2.3.2 Channel Data

The channel data block of the response is repeated for each channel. The data packets must be in the order of the channel list (see 2.2.6.1 Channel List on page 8).

Offset [bytes]	Length [bytes]	Data type	Description	Comment
Off	4	32iu	Number of samples	Since we currently support only synchronous channels without sample rate divider, this number will always be the same as the Samples in packet of the header
Off + 4	X*SampleSize	Channel data type	The data samples	The size of the samples for each channel is dependant on the channel data type. Since the plugin currently only supports the data types 32is and 32fs, the <i>Sample Size</i> is always 4.

Off = Off + 4 + X * SampleSize

2.3.3 Trailing Data

After the data of the last channel the server must send the trailing data to indicate that the data transfer is complete.

Offset [bytes]	Length [bytes]	Data type	Description	Comment
0	8		Stop packet string	0x07 0x06 0x05 0x04 0x03 0x02 0x01 0x00

2.3.4 Example

Let's say, we have acquired the first 3 samples of 2 channels:

Sample #	Channel 1	Channel 2
1	1.1	10
2	2.2	11
3	3.3	12

Then the data-packet must look like this

Offset [bytes]	Length [bytes]	Data type	Description	Data
Header				
0	8		Start packet string	0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07
8	4	32iu	Packet size (28 for header, plus 32 for the data)	60
12	4	32iu	Packet type	0
16	4	32iu	Samples in packet	3
20	8	64iu	Samples acquired so far	0
28	8	64fu	relative time	0.0
Channel 1				
36	4	32iu	Number of samples for Channel 1	3
40	4	32fu	1 st sample of Channel 1	1.1
44	4	32fu	2 nd sample of Channel 1	2.2
48	4	32fu	3 rd sample of Channel 1	3.3
Channel 2				
52	4	32iu	Number of samples for Channel 2	3
65	4	32iu	1 st sample of Channel 2	10
60	4	32iu	2 nd sample of Channel 2	20
64	4	32iu	3 rd sample of Channel 2	30
Trailing Data				
68	8		Stop packet string	0x07 0x06 0x05 0x04 0x03 0x02 0x01 0x00

Total Data Packet Size: 76

2.3.5 Example-Byte Level

This example shows the data packet sent from the server in byte level (recorded with Wireshark) to explain the expected byte order (Windows standard – Little Endian).

The servers sends one sample for 2 channels – both samples have the value 0.

Part	Hex/Dec Value	Header Bytes
(1) Start packet	00 01 02 03 04 05 06 07	<pre> Ethernet II, Src: Dell-C0-11-19-14-1E-07, Dst: Cisco-C4-9E-24-C4-07-0A, Len: 144 Internet Protocol Version 4, Src: 192.168.1.146 (192.168.1.146), Dst: 192.168.1.146 (192.168.1.146) Transmission Control Protocol, Src Port: 53209 (53209), Dst Port: 8114 (8114), Seq: 1000000000, Win: 65535, Len: 36 Data (36 bytes) Data: 00010203040506072c000000000000000100000001000000... [Length: 36] </pre>
(2) Packet Size	2c 00 00 00 44	
(3) Packet Type	00 00 00 00 0	
(4) Samples in Packet	01 00 00 00 1	
(5) Sample so far	01 00 00 00 00 00 00 00 1	
(6) Relative Time	80 ae 3c 3e f2 1a f0 3f 1.0065786772856	
		Data Bytes
(7) Samples for Ch. 1	01 00 00 00 1	<pre> Internet Protocol Version 4, Src: 192.168.1.146 (192.168.1.146), Dst: 192.168.1.146 (192.168.1.146) Transmission Control Protocol, Src Port: 53209 (53209), Dst Port: 8114 (8114), Seq: 1000000000, Win: 65535, Len: 24 Data (24 bytes) Data: 01000000000000000100000000000000706050403020100 [Length: 24] </pre>
(8) Value of Ch. 1	00 00 00 00 0	
(9) Samples for Ch. 2	01 00 00 00 1	
(10) Value for Ch. 2	00 00 00 00 0	
(11) Stop packet	07 06 05 04 03 02 01 00	

Table 3: Binary Data: Byte Level

3 Configuration

3.1 Hardware Setup

Go to the **Plugins** tabsheet ❶.
In the grid at the top, look for the row with the name *Tcp Client* ❷ (if you don't find this row, see: 1.6 Plug-in Installation on page 3) - make sure that the button in the left *ON/OFF* column shows **Used**.

When you click on the *Tcp Client* row in the grid, you will see the properties of the plugin in the bottom of the screen:

❸ see 3.1.1 OS Commands on page 14

❹ see 3.1.2 Command Connection on page 15

❺ see 3.1.3 Data Connection on page 15

❻ see 3.1.5 Log files on page 16

❼ see 3.1.1.1 Server Command on page 14

❽ see 3.1.4 Ignore Errors on Start of Acquisition on page 16

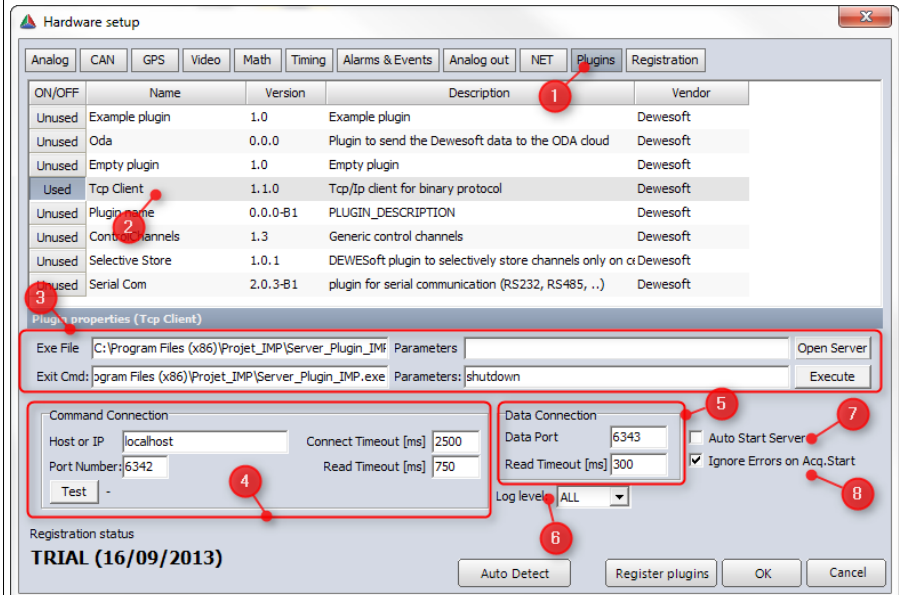


Illustration 4: Hardware setup

3.1.1 OS Commands

The OS Commands are optional. There are 2 commands available:

- ⚠ the first is the Server Command
- ⚠ and just below it, there is the Exit Command

Both consist of a command filed and a parameters field. The command plus the optional parameters will be passed to the operating system for execution.

IMPORTANT



Not that you can enter any command. So be carefully which programs you start. They could harm your PC!

3.1.1.1 Server Command

This is optional. If you leave the fields empty, you must manually start the server so that DEWESoft® can read the data (i.e. double click the icon of your server application on the Windows Desktop).

However, it can be very useful and convenient to use this function. If you enter data (at least the *Exe File*), then the *Tcp Client* plugin will always try to start the *Exe File* when the *Hardware setup* is loaded (if the *Auto Start Server* check-box is activated). You can click the **Open Server** button to test the configuration.

The **Open Server** button is also available in the Channel Setup.

If the executable is already running, it will not be started again: it simply becomes the foreground Window.

3.1.1.2 Exit Command

This command will be executed once, just before you exit DEWESoft® (i.e. when you close DEWESoft® by clicking the x-icon of the main window).

Use-case: you can use this command to shut-down your server when DEWESoft® is closed.

3.1.2 Command Connection

These settings are relevant for the *Command Connection*. The plugin will try to open the given *Command Port* on the given *Host or IP* (note: it's recommended that the server program also runs on the same PC as DEWESoft®) which will then be used to send commands to the *Server*.

Connection Timeout

The *Connection Timeout* is the maximum number of milliseconds that the plugin will try to open the connection.

Read Timeout

The *Read Timeout* is the maximum number of milliseconds that the plugin will wait for a response from the server. If the server does not respond within the specified time-out it considers the command to have failed.

When you press the **Test** button, the plugin will try to open a connection to the server and read the server's welcome message and protocol version. The server will then show a corresponding message about the test.

Everything is okay (welcome message and protocol version)

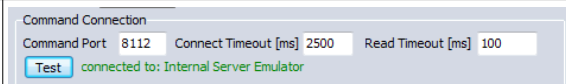


Illustration 5: Test okay

In this case the TCP/IP connection to the specified *Command Port* could not be opened within the specified timeout. Check if the *Command Port* is correct and that your Server is listening on that Port.

Also try to disable any security software that might block the communication (i.e. firewalls, Anti-virus software, etc.)

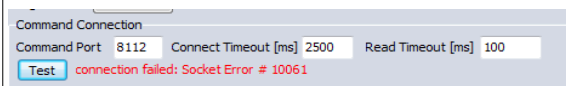


Illustration 6: Socket Error

In this case the TCP/IP connection has been established successfully, but the server did not respond with the specified welcome message (see 2.2.1 Welcome Message on page 6).

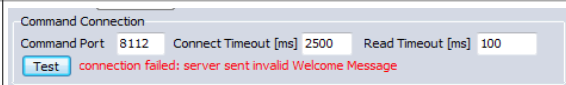


Illustration 7: Invalid Welcome Message

In this case the TCP/IP connection has been established successfully, the *Welcome Message* was okay, but the server uses another protocol version as the plugin: (see 2.2.2 Protocol Version on page 6).

Check if there is a newer version of the plugin available (see 1.2 Links on page 2).

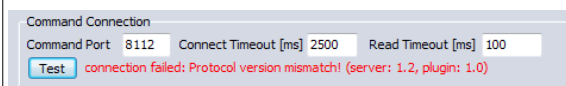


Illustration 8: Protocol Version Mismatch

3.1.3 Data Connection

When DEWESoft® needs to get the measurement data from the server it will start to listen on the specified *Data Port* (see also 2.1.2 Connection establishment on page 5). The Read Timeout specified the time in milliseconds that the

plugin will wait for the data (i.e. if the *Server* sends the *Data Header*, and no data follows within the given timeout, then the received data header will be skipped).

3.1.4 Ignore Errors on Start of Acquisition

When you activate this check-box any errors on the start of the acquisition will silently be ignored and the plugin will acquire any data from the Server.

IMPORTANT



It is NOT recommended to activate this feature because any errors on the start of the acquisition will not be shown. You will NOT get any data from the Server in this case. INSTEAD you should deactivate the plugin in Hardware setup when it's not needed. Because even when you ignore the error, the plugin will still be active: i.e. it will constantly try to connect to the server and takes up CPU and memory from DEWESoft®.

3.1.5 Log files

The plugin will write log files during operation. The amount of log messages is configurable via the *Log level* drop down box in the *Hardware setup*. The name of the logfile is `TcpIpBinaryClient.log`.

When the plugin is started, it will immediately start to log to the windows temporary directory.

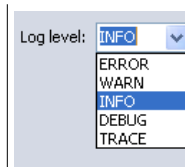
As soon as the DEWESoft® application is available to the plugin, all subsequent logs will be written to the standard DEWESoft® log directory (e.g. `D:\DEWESoft7\System\V7_1\Logs`).

Note: There is also a log file called `TcpIpBinaryClient.dll.log` in the *Addons* directory (see 1.4 Files and Directories). This will normally be empty. It will only contain messages when there is a bug very early in the plugin initialization.

3.1.5.1 Log levels

With the *log level* drop down box you can set the detail level of the logging function.

If you set a high log level (e.g. *TRACE*, *ALL*) a lot of log messages will be written and the logfiles will roll over quite often. This is also dependent on the sample rate – the higher the sample rate is, the more often data will be fetched and thus more log messages will be written.



For production-use the log level *INFO* is recommended.

Log level	Description
<i>Error</i>	Will only log error messages
<i>Warn</i>	Will also log warning messages
<i>Info</i>	Will also log info messages – this is recommended for production use
<i>Debug</i>	Will also log debug messages
<i>Trace</i>	will also log trace messages
<i>All</i>	will log all messages

3.2 Channel Setup

The *Tcp Client* channel setup will show you some status information of the TCP/IP connections, information about the channels and also online data from the server.

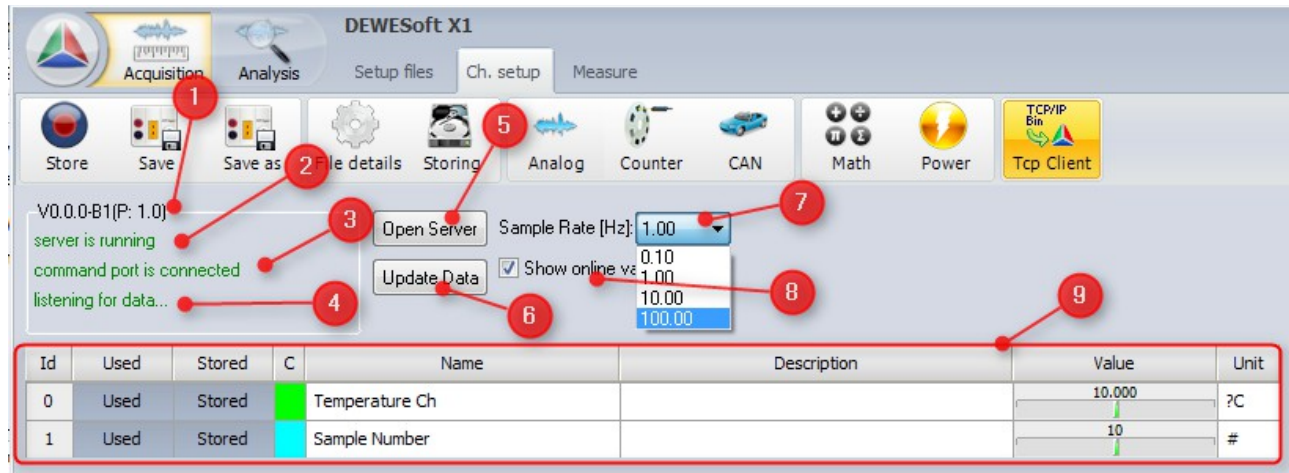


Illustration 9: Channel Setup

- ❶ Will show the version of the plugin (0.0.0-B1 in this case) and in brackets the version of the *Protocol* (1.0 in this case – see also: 2.2.2 Protocol Version on page 6)
- ❷ Shows the status of the server executable (see also 3.1.1 OS Commands on page 14).
- ❸ Shows the status of the *Command Connection*: see 3.1.2 Command Connection on page 15
- ❹ Shows the status of the *Data Connection*: see 3.1.3 Data Connection on page 15
- ❺ Will open the Server (start the executable if the Server is not running or make the Server the foreground Window)
- ❻ Will read the configuration from the Server and update the DEWESoft® configuration (Sample rates, Channel List)
- ❼ Shows a list of all sample rates from the server and when the user changes the sample rate, it will send the *Set Sample Rate* command (see 2.2.5 Set Sample Rate on page 7) command to the server
- ❽ If this check-box is checked, the plugin will listen for the data from the server and show the values in the Value column of the channel setup grid (see 3.2.3 Channel Setup Grid on page 18)
- ❾ List of all channels that the server supports: see 3.2.3 Channel Setup Grid on page 18

3.2.1 Connection

The plugin will automatically try to connect to the *Command Port* of the *Server* every-time you enter the *Ch. Setup*. If the connection is not okay, the plugin will constantly retry to open the connection.

The plugin will only request channel data from the *Server* if the *Show online values* check-box is checked.

3.2.2 First start

When you create a *New Setup* or enter the *Ch. Setup* for the first time after installing the plugin, then the plugin will try to read the information about the sample rates and channels from the server.

If the server is not running (or not listening to the *Command Port*), you will see an empty list:

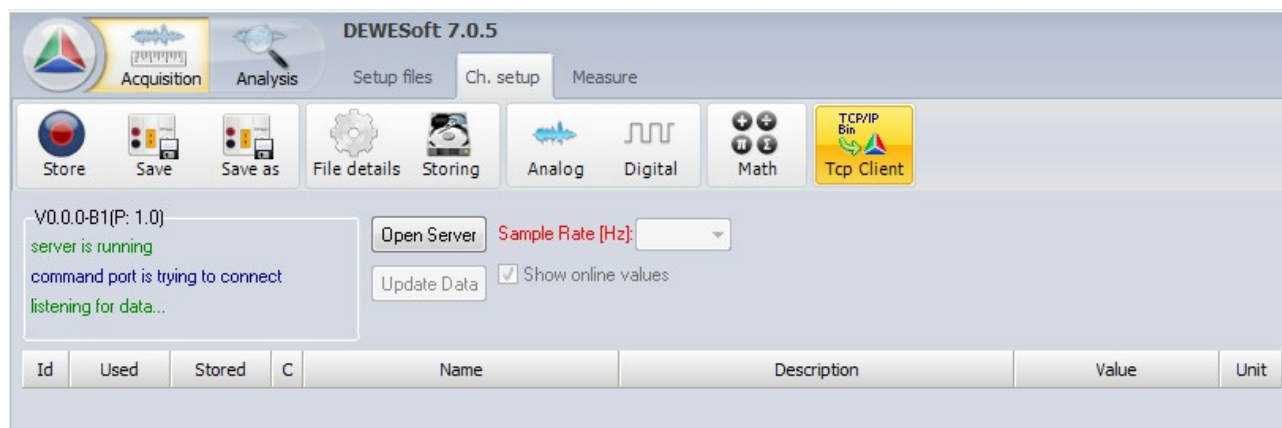


Illustration 10: Channel Setup: No server data

3.2.3 Channel Setup Grid

The Channel Setup Grid shows information about all channels of the plugin and also live values

Column	Information
Id	This is just a unique consecutive number to identify the row/channel of the grid
Used	You can click on the buttons in this row to toggle the Used status from Used to Unused . Only channels that are set to Used will show up in <i>Measure Mode</i> and can be stored in DEWESoft® data files.
Stored	This is only useful if the channel is set to Used (see description above). For Used channels you may want to deactivate the Store button. Then you can see and use the values of this channel in <i>Measure Mode</i> , but the channel data will not be stored to the DEWESoft® data file. This can be useful if you just want to check the data, but don't need it after the measurement. Another use-case is to use the data of the channel in other <i>Math</i> channels (e.g. to do some statistics) and then only store the <i>Math</i> channel to the DEWESoft® data file (but not the original data).
C	This colour will be used by the displays in <i>Measure Mode</i> . You can click on the colour to change it.
Name	This is the name of the channel as it will show up in the channel list of the Measure mode. Make sure to enter a useful name for the channel (also it makes sense to use unique names to avoid confusion). If you enter a blank name, then you will get a warning.
Description	Detailed description of the channel.
Value	This column will show the online data of the channels. Note: you will only see live data of channels that are set to Used (see description above) and only if the <i>Show online values</i> check-box is checked. If the plugin does not receive any data within 5 seconds it will close the connection and try to reopen it.
Unit	This is the unit that will be displayed for the channel. Note: you can change this value, but it is not recommended.
Server Id	This is the Unique ID of the channel as defined by the Server application (see 2.2.6.2 Unique Id on page 8). Note: the default sort-order of the grid will be based on this ID. This column is hidden per default.

3.2.3.1 Header pop-up

When you right-click on the header row of the channel grid, you can specify which columns you want to see (i.e. show/hide columns) and you can also sort/unsort the grid by certain columns.

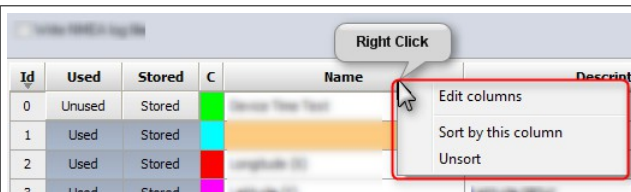


Illustration 11: Grid: Header Pop-up

3.2.3.2 Multi select

You can select and edit multiple rows at once.

1. left-click a cell and hold the mouse button
2. move the mouse (while still holding down the mouse button) to the target cell and then release the mouse button

The selected cells will now be surrounded by a black rectangle. When you now click into the selected region, you can change the Used status of all selected cells at once. Note: this does also work for text-columns: i.e. you can select several cells of the *Name* column and change all selected names at once.

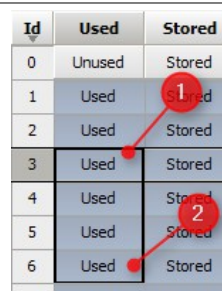


Illustration 12: Grid: Multi Select

4 Version History

Revision number: 62

Last modified: Thu 10 Nov 2016, 13:32

Plugin-Version	Date [dd.mm.yyyy]	Notes
0.0.0-B1	21.06.2013	initial beta revision for testing only!
1.0.0	12.08.2013	Initial release version
1.1.0	06.09.2013	<input checked="" type="checkbox"/> New Icon <input checked="" type="checkbox"/> Hardware setup: <input type="checkbox"/> Added Exit Command <input type="checkbox"/> Added Auto Start Server checkbox <input type="checkbox"/> Added Ignore Errors on Acq. Start checkbox
1.1.1	12.09.2013	<input checked="" type="checkbox"/> Added exit command
1.1.2	02.03.2016	<input checked="" type="checkbox"/> Fixed problem when the server only had a single Sample Rate <input checked="" type="checkbox"/> Fixed possible problem with Decimal-separator (must always be a dot in the protocol) <input checked="" type="checkbox"/> Grid uses hints to display warnings/errors <input checked="" type="checkbox"/> Internal: updated to Delphi XE8 <input checked="" type="checkbox"/> Internal: using DEWESoft® X2 XML <input checked="" type="checkbox"/> Internal: using grid controller structure <input checked="" type="checkbox"/> Internal: using new plugin structure <input checked="" type="checkbox"/> Corrected locking when updating the server data (i.e. the Grid must be freed, so that the data is not accessed from 2 threads at the same time, and then recreated after the update) <input checked="" type="checkbox"/> Corrected problem with live-values when show online values checkbox was deactivated and activated again <input checked="" type="checkbox"/> Documentation <input type="checkbox"/> Orange Layout & DEWESoft® <input type="checkbox"/> Fixed some typos: i.e. SamplRate → SampleRate <input type="checkbox"/> Improved the Data Protocol example (now using colours, and fixed some errors)
1.1.3	06.07.2016	<input checked="" type="checkbox"/> When the sample rate is changed, the ExpectedAsyncRate of the channel is now immediately updated <input checked="" type="checkbox"/> The default sort order of the Channel Setup Grid is based on the Server-Id
1.1.4	13.07.2016	<input checked="" type="checkbox"/> Fixed possible startup issue for non-English locale settings <input checked="" type="checkbox"/> Documentation: now we use only the plugin version (removed the doc-version)
1.1.5	10.11.2016	<input checked="" type="checkbox"/> Start Transfer now uses the Connection Timeout <input checked="" type="checkbox"/> Increased default Read Timeout for command connection from 100 to 200ms