

**Title** : YDOC TCP-Server manual

**Date** : Feb. 2019

**Version** : 1.09

**Author** : Your Data Our Care

## **PREFACE**

This manual is an add on to the logger (user) manuals and describes how to install and use the Java application: ydocTCPserver. This application can be used to collect log files from your YDOC loggers and provides a possibility to remotely configure the loggers from anywhere. ydocTCPserver is written in Java and should practically run on any Java SE virtual machine. Its allowed to run ydocTCPserver free of charge on any Java SE virtual machine when used to collect data from YDOC loggers only. ydocTCPserver is provided "As Is" and on the users own risk, no claims can be put when not meeting expectations, in case of mal functioning or causing data loss.

## **TABLE OF CONTETS**

PREFACE .....	1
TABLE OF CONTETS .....	1
1 INTRODUCTION .....	2
2 INSTALLATION & CONFIGURATION .....	3
2.1 Secure password authentication (CHAP) .....	4
3 LOG FILES .....	4
4 REMOTE CONFIGURATION .....	4
5 AUTOMATIC CONFIGURATION UPDATE .....	5

## 1 INTRODUCTION

We have implemented three different possibilities to transfer log data to a central system. By FTP, by e-mail and by raw TCP. You can skip this manual if you want to collect data by FTP or e-mail as the logger user manual should provide enough info about it.

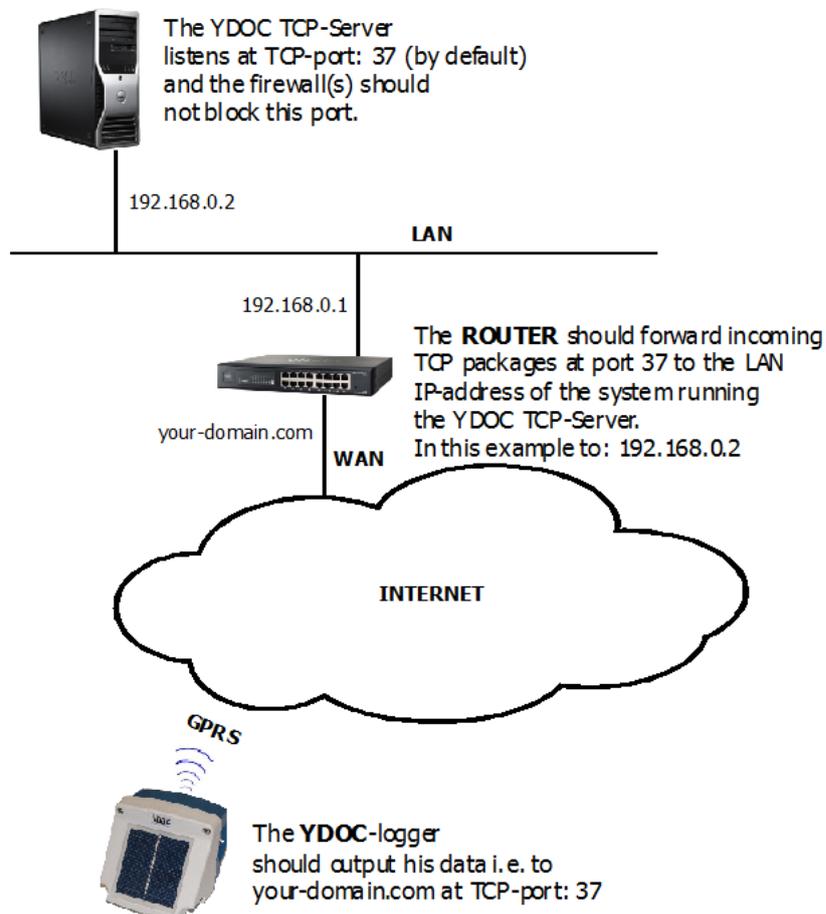
Why using a raw TCP protocol?

We are talking about low power devices and being low power involves two basic elements, at first using low power components and at second consuming power as short as possible. The throughput of mobile communications is great and sending the log data itself is done in a blink. However mobile communication knows latencies so every handshake from client to server will take almost 1 second. Protocols like FTP and E-mail use several handshakes, causing the communication to take seconds longer and therefore an increase in power consumption.

We have chosen raw TCP communication to transfer the log data in one “dump” to the server, the server on his turn has to acknowledge the reception by one single response. So the total transfer is just one “handshake”, saving a lot of precious power seconds.

The log data transferred thru TCP has the same format as the log data contained in a log data FTP file (see chapter Data Format in the logger user manual).

Another advantage of using raw TCP is the possibility to have on-line bi-directional communication, to be able to implement remote configuration and transparent access to third party serial sensors.



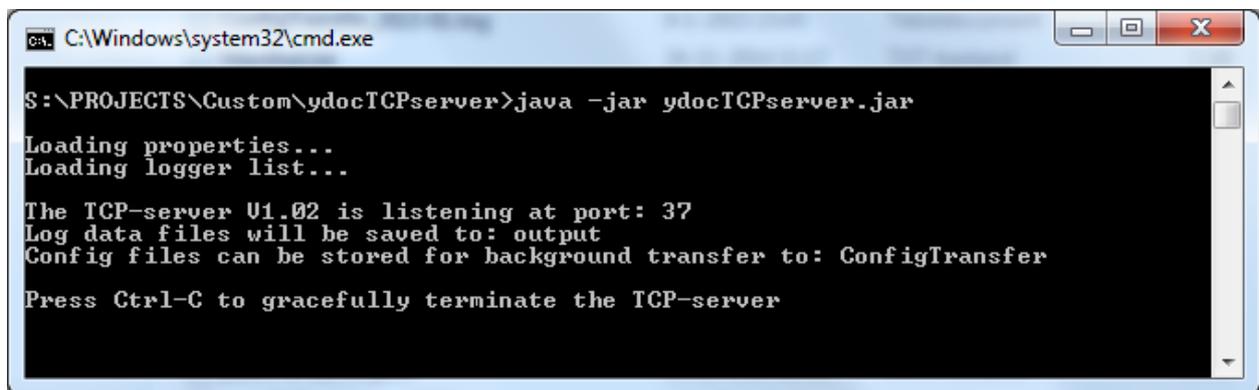
---

## 2 INSTALLATION & CONFIGURATION

Installation is quite simple, just download 'ydocTCPserver.jar' from 'www.your-data-our-care.com' and copy it to a folder on a machine with a running Java SE virtual machine.

Starting the jar on a Windows PC can be done by executing the java command.

```
java -jar ydocTCPserver.jar
```



```
C:\Windows\system32\cmd.exe
S:\PROJECTS\Custom\ydocTCPserver>java -jar ydocTCPserver.jar
Loading properties...
Loading logger list...
The TCP-server V1.02 is listening at port: 37
Log data files will be saved to: output
Config files can be stored for background transfer to: ConfigTransfer
Press Ctrl-C to gracefully terminate the TCP-server
```

When starting the jar for the first time it will run with default settings.

The default settings are:

1. Listening at TCP-port 37 (If your system does not allow this port, modify it)
2. No credentials are required for loggers to dump log data to this server.
3. Log files will be at default stored in a sub-folder (of the current working folder) with name 'output'.
4. It's possible to duplicate log files to a secondary output folder, which is left empty (disabled) at default.
5. Configuration files for automatic logger configuration updates, should be stored at default in the 'ConfigTransfer' folder.

After running for the first time the jar will create an ydocTCPserver.cfg file in the current working folder, which can be modified by a simple text editor to change the above listed defaults. The configuration file contains the following properties:

```
Output-path=output
SecondaryOutput-path=
TCP-port=37
User=
Password=
ConfigTransfer-path=ConfigTransfer
```

When specifying a user name and password, the same user name and password should be programmed into all loggers as well.

When changing the TCP-port don't forget to make sure that firewalls are not blocking this port and that the router will forward this port from outside INTERNET to the local IP-address of the system running the jar. The YDOC loggers should dump their data to the outside known port.

## 2.1 Secure password authentication (CHAP)

As of FW version V2.1B5 data loggers can be configured to use a more secure TCP authentication procedure based on CHAP (Challenged Handshake Authentication Protocol). With this protocol the password itself is not transmitted by the TCP-client (logger or terminal) but as an one way encrypted password using a challenge string received from the TCP-server.

CHAP support is implemented in ydocTCPserver V1.02 and ydocTerminal V1.5

---

## 3 LOG FILES & PICTURES

The contents of the log files (\*.csv, \*.txt or json) are having the same format as log files transferred by FTP (see chapter Data Format in the logger user manual).

Each time a logger connects to dump its log data a log file will be created. The format of the log file name is: YDOC\_<logger name>\_<logger SN#>\_<log file date yymmdd>\_<log file time hhmmss.zzz>.txt

e.g. YDOC\_TEST\_4371293\_141124\_12:00:04.300.txt for a log file in native txt format

or YDOC\_TEST\_4371293\_141124\_12:00:04.300.jpg for a camera picture

---

## 4 REMOTE CONFIGURATION

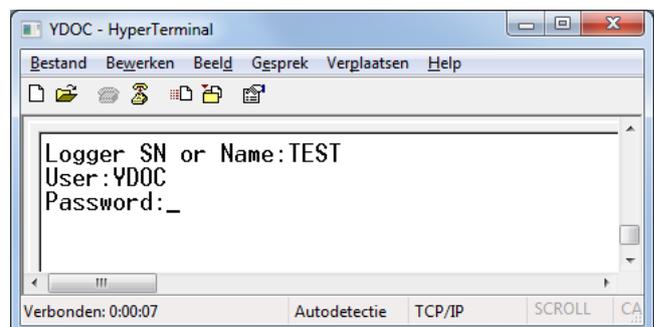
To save power the logger is switched off most of the time and can't be reached, therefore using an integrated web server in the logger to configure the loggers is useless. To not rely on external software (versions) we have implemented a terminal interface to configure the logger locally by USB and remotely by TCP. The terminal interface is described in the logger user manual.

To be able to do remote configuration by TCP, the ydocTCPserver can intercept the TCP connection when a logger has just dump its data.

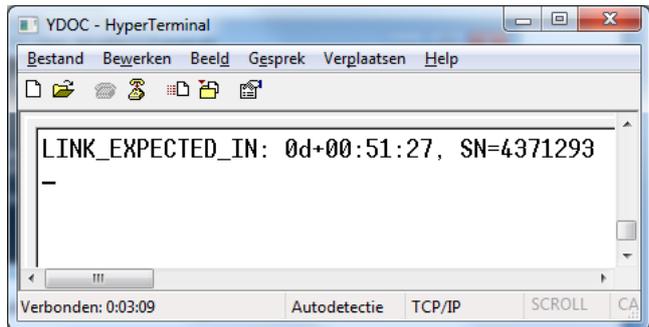
To remotely configure a logger you can use any terminal emulation program supporting TCP socket communication (e.g. hyperterminal of our free ydocTerminal). Its more secure to use ydocTerminal as it supports secure login (CHAP).

Within the terminal emulator, connect to the IP-address/domain name of ydocTCPserver and the TCP-port it is listening at.

After connecting you will be prompted to specify the SN (serial number) or the name of the logger you want to configure, followed by prompting for credentials.



If the logger is known to the ydocTCPserver and credentials are supplied correctly, the terminal window will display a text telling when the logger is expected to be awake. This line will be updated every second.



When the logger awakes, the terminal window will display the terminal interface of the logger.

---

## 5 AUTOMATIC CONFIGURATION UPDATE

If you have a lot of loggers and don't want to configure them all manually, you can just configure one and use that configuration for automatic background transfer to all other loggers.

Automatic update requires that the logger firmware has version 1.8 or higher.

To automatic upgrade logger configurations, copy the configuration files to the 'ConfigTransfer' folder, you can specify the used folder in the ydocTCPserver.cfg file (See chapter 2). **Note:** When you copy a file, be aware that it should literally be a copy as the file will be deleted after processing.

The file name syntax of an individual configuration file is: <your file name prefix>\_<logger SN#>.cfg

To update multiple loggers with the same configuration you can:

- a) obviously copy and rename them to individual logger configuration files.
- b) Specify a configuration file with a <logger name prefix> instead of an individual <logger SN#> (e.g. when you rename a configuration file to LOGGER\_TEST.cfg then all loggers with their name starting with TEST will be updated).

When a logger connects to the TCP-server to dump its data, the server will check if a configuration file for the logger is available and will try to transfer the file to the logger directly after he dumped its log data.

Successful or failed configuration updates will be listed in a monthly log file in the current working folder. The log file has the following format: ConfigTransfer\_YYYY-MM.log