Title : YDOC-Datacollector manual **Date** : April 2020

Version : 1.0

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: ydocDataCollector, successor of the ydocTCPserver. This application can be used to collect log files from your YDOC loggers to a disk share and provides a possibility to remotely configure the loggers from anywhere. ydocDataCollector is written in Java and should practically run on any Java SE virtual machine. Its allowed to run ydocDatacollector free of charge on any Java SE virtual machine when used to collect data from YDOC loggers only. ydocDatacollector 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.

The differences with its predecessor ydocTCPserver are:

1. Support for multiple users, loggers can be configured with different credentials to access the data-collector and files can be transferred to individual user directories.
2. Support for scheduled remote configuration sessions.
3. Act as a secure tunnel server to make scheduled remote configurations sessions for loggers using FTP, MQTT or HTTP for transferring log files to a central system (See logger user manual).

TABLE OF CONTETS

1 INTRODUCTION

We have implemented several different possibilities to transfer log data to a central system. By FTP, by e-mail, MQTT, HTTP and by raw TCP.

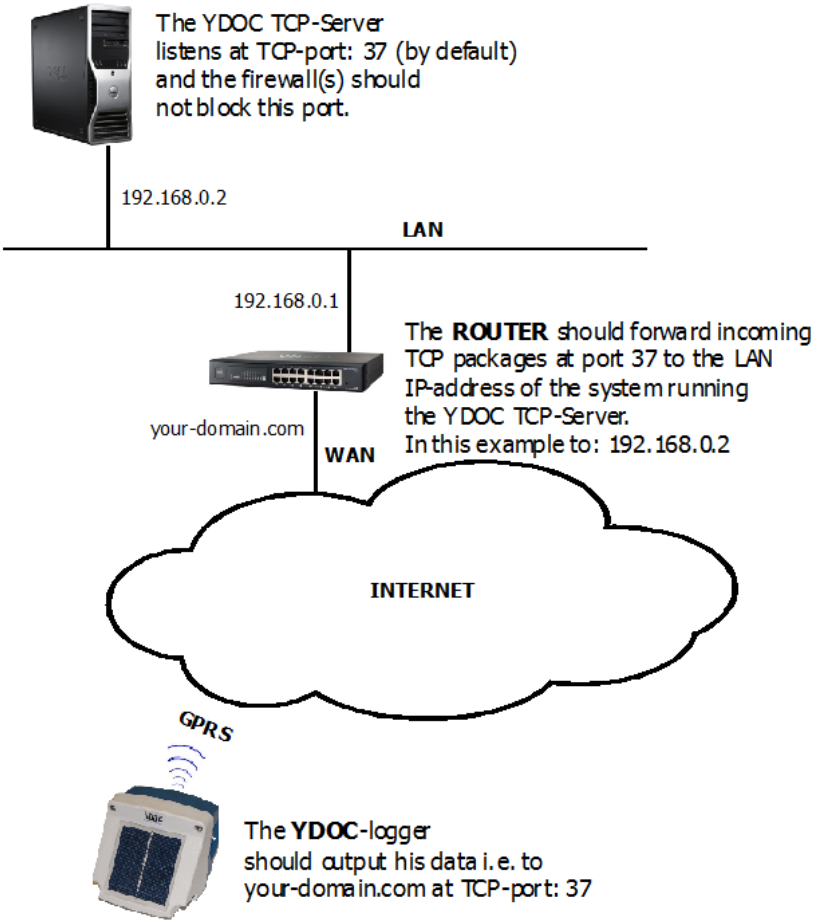
Why using a raw TCP protocol?

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 takes time, especially when using SSL/TLS. Protocols like FTP and E-mail use several handshakes, causing the communication to take longer and therefore an increase in power consumption. The data payload could be another reason as protocols like HTTP and FTP are coming with overhead, especially when using SSL/TLS.

We have chosen raw TCP communication to transfer the log data in one (encrypted) “dump” to the server, the server on his turn has to acknowledge the reception by one single (encrypted) response. So the total transfer is just one “handshake” with little overhead, saving a lot of precious power time and payload..

The format of the transferred log files; txt, csv or json is describe in the logger user manual. The data collector can receive camera pictures in jpg format as well.

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.

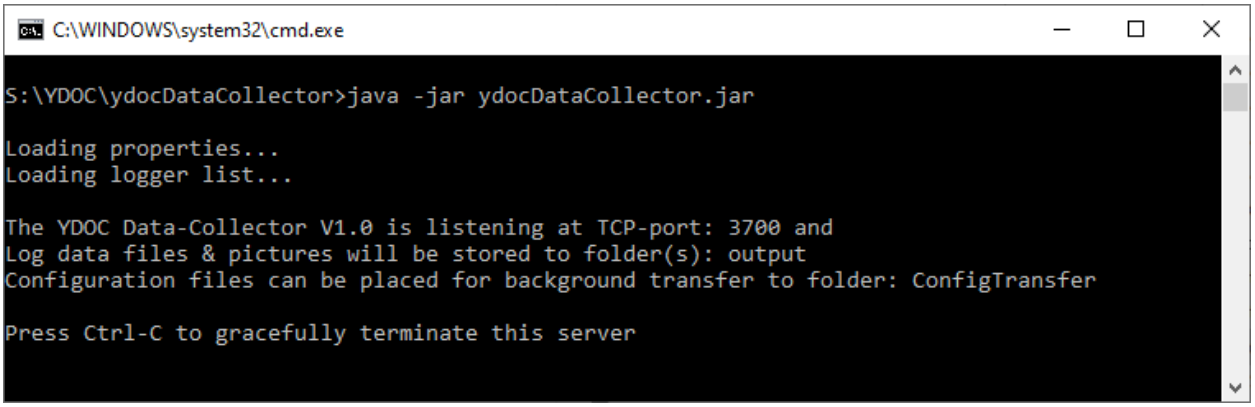


2 INSTALLATION & CONFIGURATION

Installation is quite simple, just download 'ydocDatacollector.jar' from 'www.ydoc.biz' 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 ydocDataCollector.jar



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

The default settings are:

1. Listening at TCP-port 3700 (If your system does not allow this port, modify it)
2. Credentials are required for loggers or terminals to connect to this server (see config file).
3. Log files will be at default stored in a sub-folder (of the current working folder) with name 'output'.
4. Configuration files for automatic logger configuration updates, should be stored at default in the 'ConfigTransfer' folder.
5. The data-collector can also act as a secure tunnel server for YDOC-loggers (see: https://www.ydoc.biz/datalogger-remote-configuration.html). To enable the tunnel server you have to specify a 'TunnelToken' (e.g. yTunnel) and specify this access token in the 'Tunnel Sever'-properties of ydocTerminal as well.

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

```
#The TCPcollector should listen at port (0=disable):
TCP-port=3700

#Log files & pictures should be stored in output folder:
Output-folder=output

#Folder for configurations files to be synchronized with the
corresponding data logger(s):
ConfigTransfer-path=ConfigTransfer

#User name and password of the main user:
User=YDOC
Password=<some random#>

#ydocTerminal secure tunnel access token (make empty to disable):
TunnelToken=T3f21b0a5
```

```
#Additional users can be added with a property starting with the text
User-
#User-XYZ=<user name>;<user password>;<optional output path>
User-ABC=YourName;yourpsw;your-output
User-123=HisName;hispsw;his-output
```

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)

The 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.

2.2 AES-128 data encryption

The data loggers can be configured to use AES-128 data encryption, the encryption tokens change every session.

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

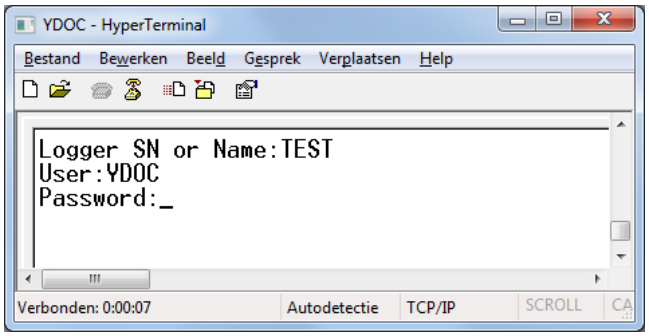
To save power the logger is switched off most of the time and can't be reached, therefore using an integrated web server in the logger to configure the loggers is useless. To not rely on external software (versions) we have implemented a terminal interface to configure the logger locally by USB and remotely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

To be able to do remote configuration by TCP, the data-collector can intercept the TCP connection when a logger has connected and has finished transferring its data.

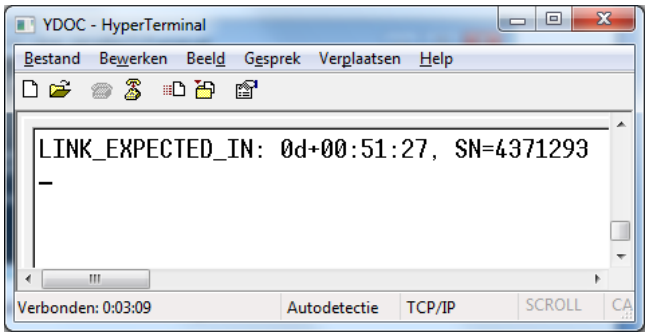
To remotely configure a logger you can use any terminal emulation program supporting TCP socket communication (e.g. hyperterminal of our free ydocTerminal). Its more secure to use ydocTerminal as it supports secure login (CHAP) and AES encryption.

Within the terminal emulator, connect to the IP-address/domain name of the data-collector 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 data-collector 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.

To automatic upgrade logger configurations, copy the configuration files to the 'ConfigTransfer' folder, you can specify the folder in the ydocDatacollector.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 data-collector 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