1 IP-Proxy

1.1 Introduction

IP-Proxy is a command line tool that provides one or more receivers on one leg and one or more senders on the other leg. The tool is available for Windows and Linux. It can be used to perform message traffic between different protocols and/or different applications. IP-Proxy supports following protocols and applications both on receiver- and sender-side, except for IPC, see below:

TCP

- Client / Server

UDP

- Client / Server
- Unicast / Multicast

Serial (RS232 / RS485)

- Simplex / Duplex operation

IPC – Inter-Process Communication

- IPC-Server on receiver side
- IPC-Client on sender side

HTTP/HTTPS

- HTTP/HTTPS-Server on receiver side
- HTTP/HTTPS-Client on sender side
  Note: only HTTP/HTTPS-POST requests are supported

Both on receiver- and sender-side a Data-Processor can be connected to pre- and or post-process the incoming and/or outgoing data. This powerful feature allows adapting IpProxy according to the individual needs by adding of different simple but also of complex functions.
Note: The Data-Processor API will be described in a separate manual.
For easier and fast definition of receiver and sender configurations a separate tool is provided.

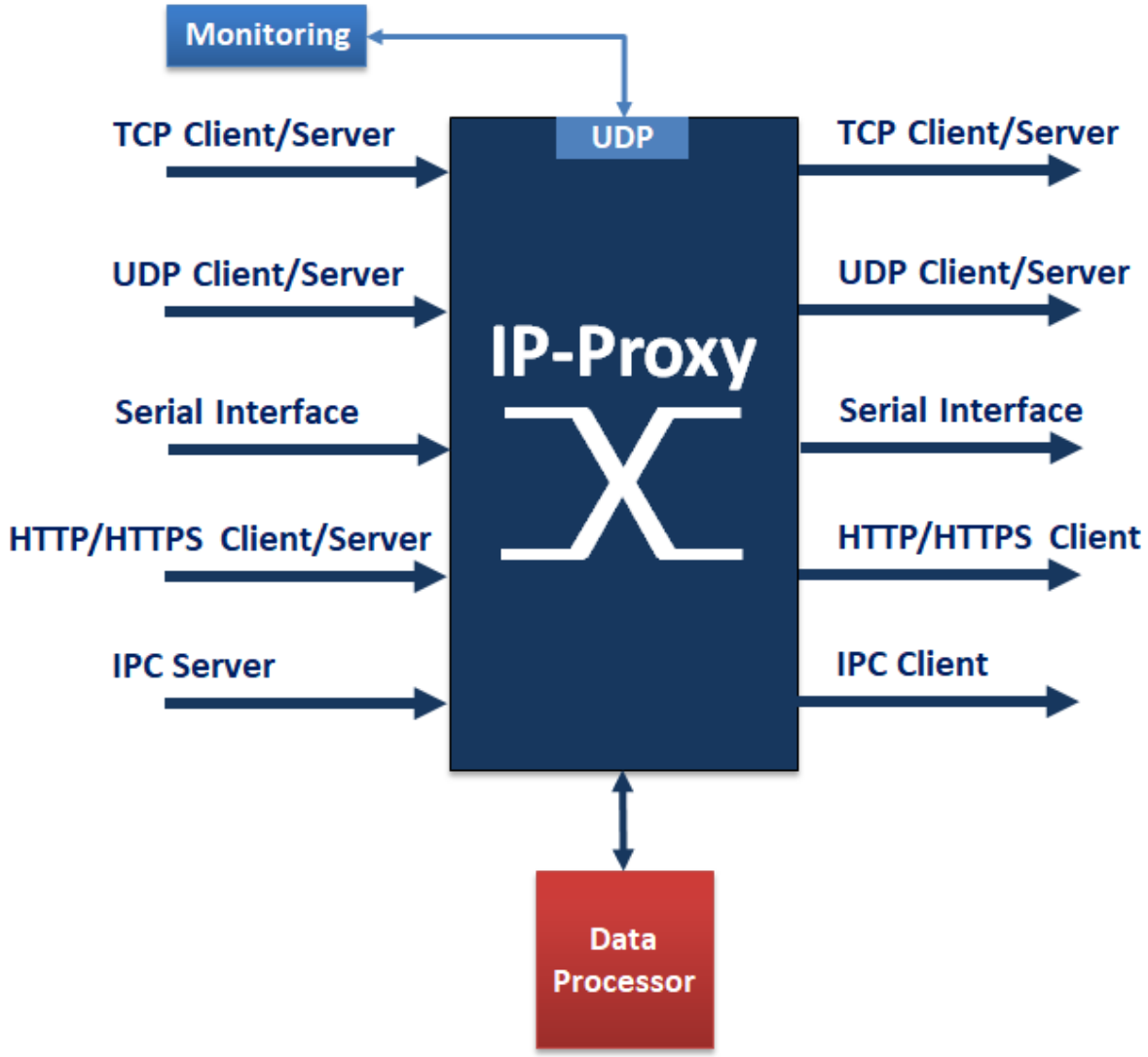


Figure 1: IP-Proxy Overview

1.2 Usage

1.2.1 Windows or Linux Desktop

IP-Proxy can be called from a Command-Line Window (Windows) or from a Terminal Window (Linux) as following

```
//Windows
IpProxy –f [configuration-file]

//Linux
ipproxy –f [configuration-file]
```

Parameters	
-f [config-file]	load the configuration provided by the configuration-file
-c	hide console window *Note: does only work for Windows OS*
-h	get help

Note: if no configuration file is provided then following default configuration is used
RECEIVER : UDP-Server, localhost, port 12345
SENDER_0 : UDP-Client, localhost, port 54321

1.2.2 Linux without GUI

On Unix Server IP-Proxy can be called as following:

```
ipproxy –f [configuration-file] > [log-fil] &
```

Parameters	
> [log-file]	the whole output will be redirected into a log-file
&	the application will started as background process Note: the application has to be terminated by killing the appropriate job, e.g. ***kill %1*** for killing job-number '1'

1.2.3 Monitoring

The operation of IP-Proxy can be monitored via external applications. Such a monitoring can especially useful if IP-Proxy is running in a 24/7 environment. By sending of special pre-defined commands via UDP, see below, IP-Proxy will provide information about the current receivers and senders status. An external application, e.g. decontev's IP-Toolbox, can use this status information as trigger for special actions. Such an action could be the re-start of IP-Proxy in case of interface errors.

The configuration of the monitoring UDP-server can be done in the 'General' tab of the settings dialog.

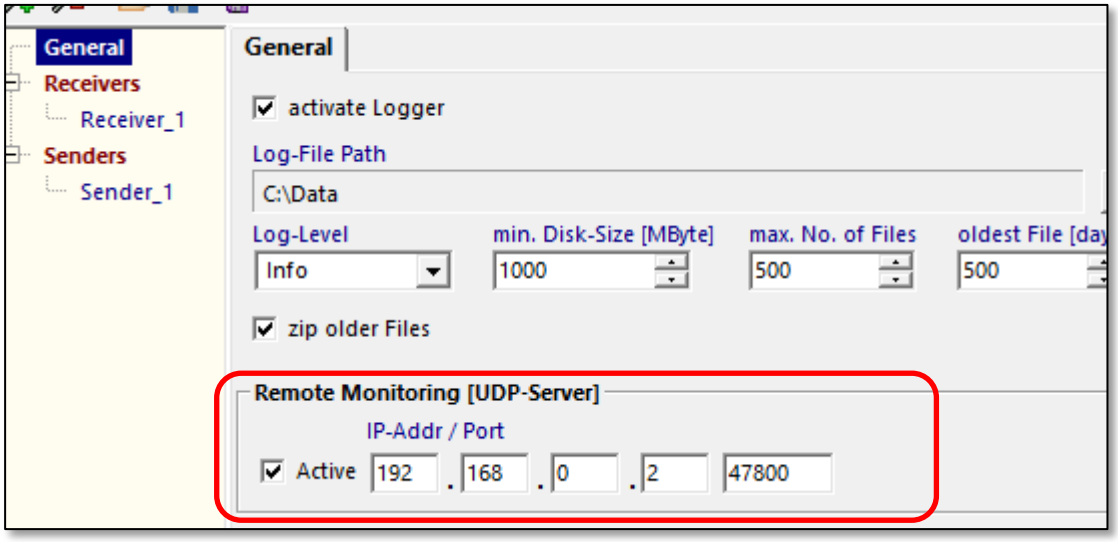


Figure 2: configuration of monitoring server

For each receiver and sender the monitoring can be activated and a message receiving/sending timeout can be set.

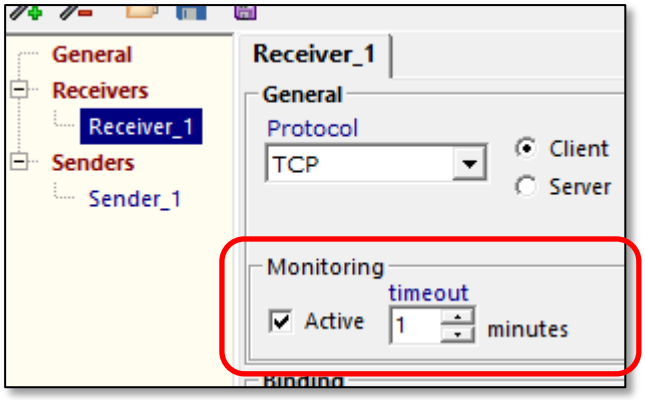


Figure 3: receiver/sender monitoring configuration

Following monitoring commands are defined:

STATUS_IN

- if IP-Proxy receives this command then all receivers are checked for a timeout
- return value
  -1 = not applicable (e.g. if there is no receiver with active monitoring)
  0 = no timeout
  1 = at least one receiver signals timeout

```
21:20:48,256 >> out_0 [5991] : 636D643D7375626D697...
CtrlServer : client connected 192.168.0.2:54085
CtrlServer : cmd received STATUS-IN
CtrlServer : sends 'ok' -> 0
21:20:58,242 << inc_0 [50655] : 7B0A202022666F726D...
```

Figure 4: monitoring command STATUS_IN

STATUS_OUT

- if IP-Proxy receives this command then all senders are checked for a timeout
- return value
  -1 = not applicable (e.g. if there is no sender with active monitoring)
  0 = no timeout
  1 = at least one sender signals timeout

```
21:19:28,257 :: dpi_0 end: ok
21:19:28,257 >> out_0 [5991] : 636D643D7375626D697...
CtrlServer : client connected 192.168.0.2:55268
CtrlServer : cmd received STATUS-OUT
CtrlServer : sends 'ok' -> 0
21:19:38,245 << inc_0 [50655] : 7B0A202022666F726D...
```

Figure 5: monitoring command STATUS_OUT

STATUS

- if IP-Proxy receives this command then all receivers and senders are checked for a timeout
- return value
  -1 = not applicable (e.g. if there are no receivers and no senders with active monitoring)
  0 = no timeout
  1 = at least one receiver or one sender signals timeout

STATISTICS

- if IP-Proxy receives this command then a short statistical report will be returned

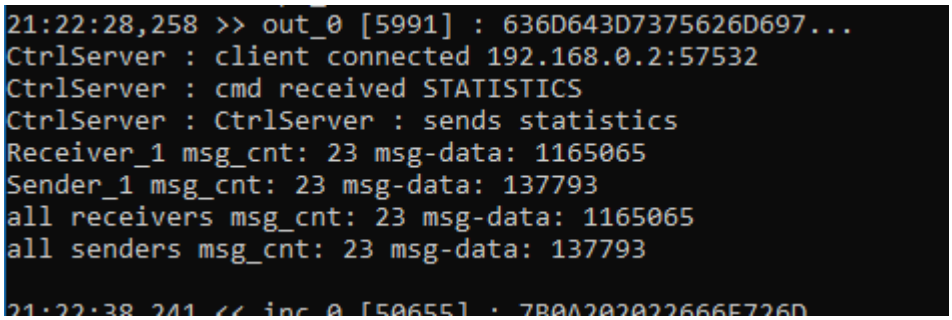


Figure 6: monitoring command STATISTICS

1.2.4 Terminate IP-Proxy

The application can be terminated by pressing the key combination **Ctrl+C**

1.3 Configuration

The configuration of IP-Proxy is done by means of a configuration file. This file has to be formatted as the well-known Ini-File format, i.e.

```
[Section_1]
parameter_1 = value_1
parameter_2 = value_2
:
[Section_2]
:
```

***Note**: there is a sample configuration file 'my_proxy.ini' in the installation directory of IP-Proxy*

In following an example is shown:

```
[Log]
LogAct=1
:

[General]
NoOfReceivers=1
:

[Receiver_1]
IpAddr    =127.0.0.1
:

[Serial_R_1]
BaudRate=11
:

[Receiver_Http_1]
HttpIpPort=443
:

[Sender_1]
IpAddr    =192.168.0.11
:

[Serial_S_1]
BaudRate=11
:

[Sender_Http_1]
HttpClientUrl=https://192.168.0.58/nrdp/
:
```

All parameters are described in following:

[General]	
IniNoOfReceivers	number of receivers, i.e. number of sections [Receiver_x] which have to be provided
IniNoOfSenders	number of senders, i.e. number of sections [Senders_x] which have to be provided
CtrlIp	IP address of monitoring UDP server
CtrlPort	Port of monitoring UDP server
CtrlAct	monitoring server on/off 0 = off, 1 = on

[Receiver_X], [Sender_X]	
IpAddr	IP-Address of Client/Server
IpPort	IP-Port of Client/Server
bBind	Binding of Client/Server 0 = off, 1 = on
BindIp	Binding IP-Address
BindPort	Binding IP-Port
bMcast	Multicast (only UDP) 0 = off, 1 = on
McastIP	Multicast IP-Address
Protocol	0 = UDP, 1 = TCP, 2 = Serial
bIpServer	Application (only TCP/UDP) 0 = acts as Client 1 = acts as Server
RcvTimeOut	receiver timeout in msec
Duplex	Duplex Operation Mode (only Serial) 0 = off, 1 = on
ProcFn	file name of an external data-processor library (including path)
ProcParam	list of parameters for the external data-processor note: the parameters and their format are defined by the data-processor
bLogHex	the data will be logged as hexa-decimal data string
bLogDpi	the data-output of the data-processor is logged note: only available for receiver data-processors
IpcId	unique ID for an inter-process communication (IPC)
IpcTime	time in seconds for waiting of an IPC connection
bMonitoring	monitoring on/off 0 = off, 1 = on
iMonTimeout	monitoring message timeout (minutes)

[Serial_R_X], [Serial_S_X]	
BaudeRate	***Windows*** 0= 110, 1 = 300, 2 = 600, 3 = 1200, 4 = 2400, 5 = 4800, 6 = 9600, 7 = 14400, 8 = 19200, 9 = 38400, 10 = 56000, 11 = 57600, 12 = 115200, 13 = 128000, 14 = 230400, 15 = 256000, 16 = 460800, 17 = 921600 ***Unix*** 0 = 0, 1 = 50, 2 = 75, 3 = 110, 4 = 134, 5 = 150, 6 = 200, 7 = 300, 8 = 600, 9 = 1200, 10 = 1800, 11 = 2400, 12 = 4800, 13 = 9600, 14 = 19200 15 = 38400, 16 = 57600, 17 = 115200, 18 = 230400, 19 = 460800, 20 = 500000, 21 = 576000, 22 = 921600, 23 = 1000000, 24 = 1152000, 25 = 1500000, 26 = 2000000, 27 = 2500000, 28 = 3000000, 29 = 3500000, 30 = 4000000
DataBits	0 = 8bits, 1 = 7bits, 2 = 6bits, 3 = 5bits
StopBits	0 = 1, 1 = 1,5, 2 = 2
Parity	0 = none, 1 = Xon/Xoff, 2 = Hardware
Device	Name of the Serial Device (only Serial) ***Windows*** e.g. COM1 ***Unix*** e.g. ttyS0
LineCrLf	Line-Feed Control (only Serial) 0 = off, 1 = on

[Receiver_Http_X]	
Https	enable HTTPS 0 = HTTPS disabled 1 = HTTPS enabled
HttpCertFile HttpCAFile HttpPrivKey HttpKeyPw	HTTPS certificate Note: a decontev certificate is available in the subdirectory 'Certificate' of the IP-Tools installation directory example using decontev certificate: HttpCertFile=<install_dir>\Certificate\d-ca-cert.pem HttpCAFile==<install_dir>\Certificate\d-pub.pem HttpPrivKey=<install_dir>\Certificate\d-ca-key.pem HttpKeyPw=decontev_123
HttpClientGetFreq	frequency of HTTP Get requests
HttpClientUrl	URL of the HTTP Get request

[Sender_Http_X]	
HttpClientUrl	destination URL
HttpClientUserAgent	replace UserAgent in the HTTP header
HttpClientMimeType	replace MimeType in the HTTP header
HttpClientAddHeaderCnt	number of additional HTTP header entries
HttpClientAddHeader_x	additional HTTP header entry x

[Log]	
LogAct	activate Logger 0 = inactive , 1= active
LogPath	log-file path
LogLevel	1 = error, 2 = warning, 3 = info, 4 = debug
DiskSize	minimum disk size in MB for logging
NoOfFiles	number of log-files in the log-file path
ZipArchive	0 = none, 1 = older files are zipped

1.4 Setup-Tool

For easier configuration, especially several listeners and/or sender has to be defined, the IpProxy-Setup tool can be used. The tool is installed in the same directory of IpProxy. The setup-tool is also available for Windows and Linux.

Start the tool by calling the appropriate application file *'IpProxy_Setup'* from the installation directory. The use of the tool is largely self-explanatory.

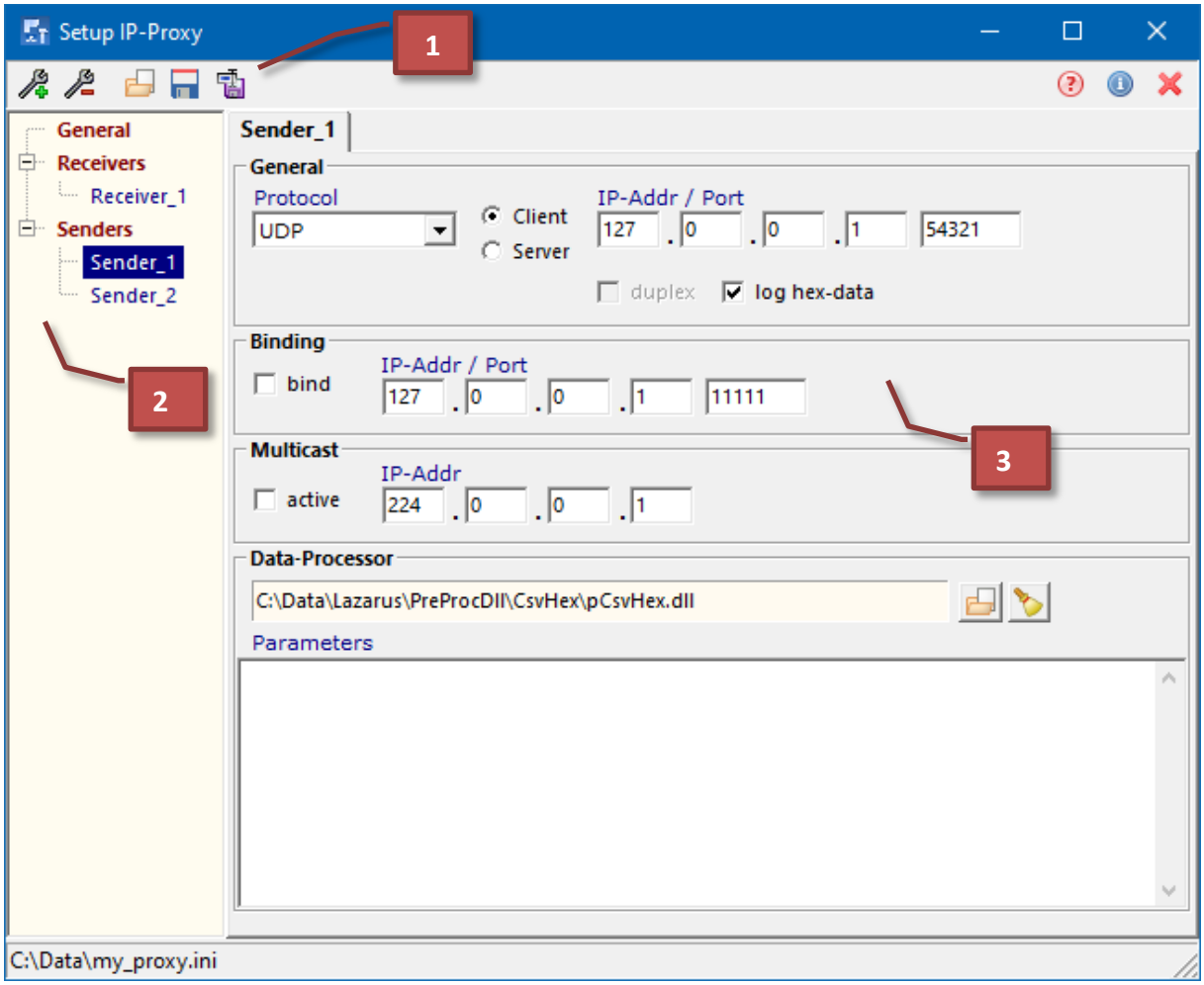


Figure 7: IP-Proxy Setup-Tool

[1] tool-bar
[2] tree-view of defined receivers and senders
[3] set-up panel for the selected item

Tool-Bar

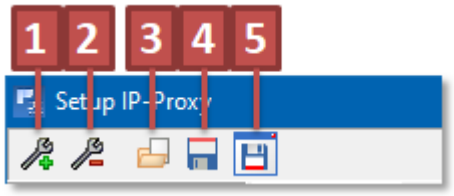


Figure 3: Tool-Bar

[1] add receiver or sender
[2] delete selected item
[3] load configuration file
[4] save current configuration file
[5] save configuration file under new filename

1.4.1 General Configuration

Select 'General' in the tree-view to setup general configurations.

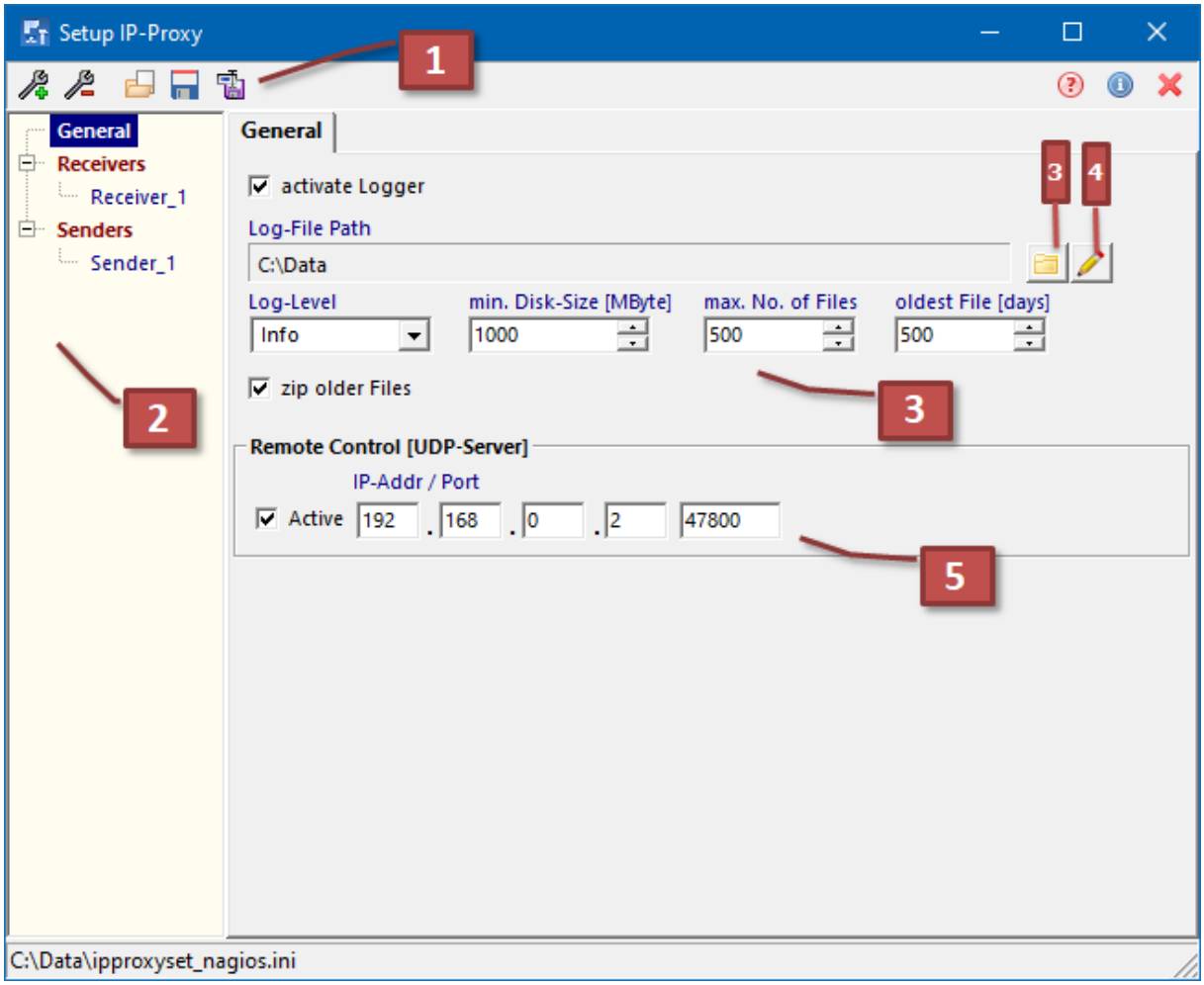


Figure 8: IP-Proxy Setup-Tool - general configuration

[1] activate logging
[2] setup the logging parameters
Note: the parameters are described above in the table [Log]
[3] click to open a dialog for selecting the log-directory
[4] click to edit the log-directory manually
[5] setup UDP-Server parameters for remote monitoring

1.4.2 Receiver / Sender Configuration

The configuration panels for receiver and sender are identical and largely self-explanatory.

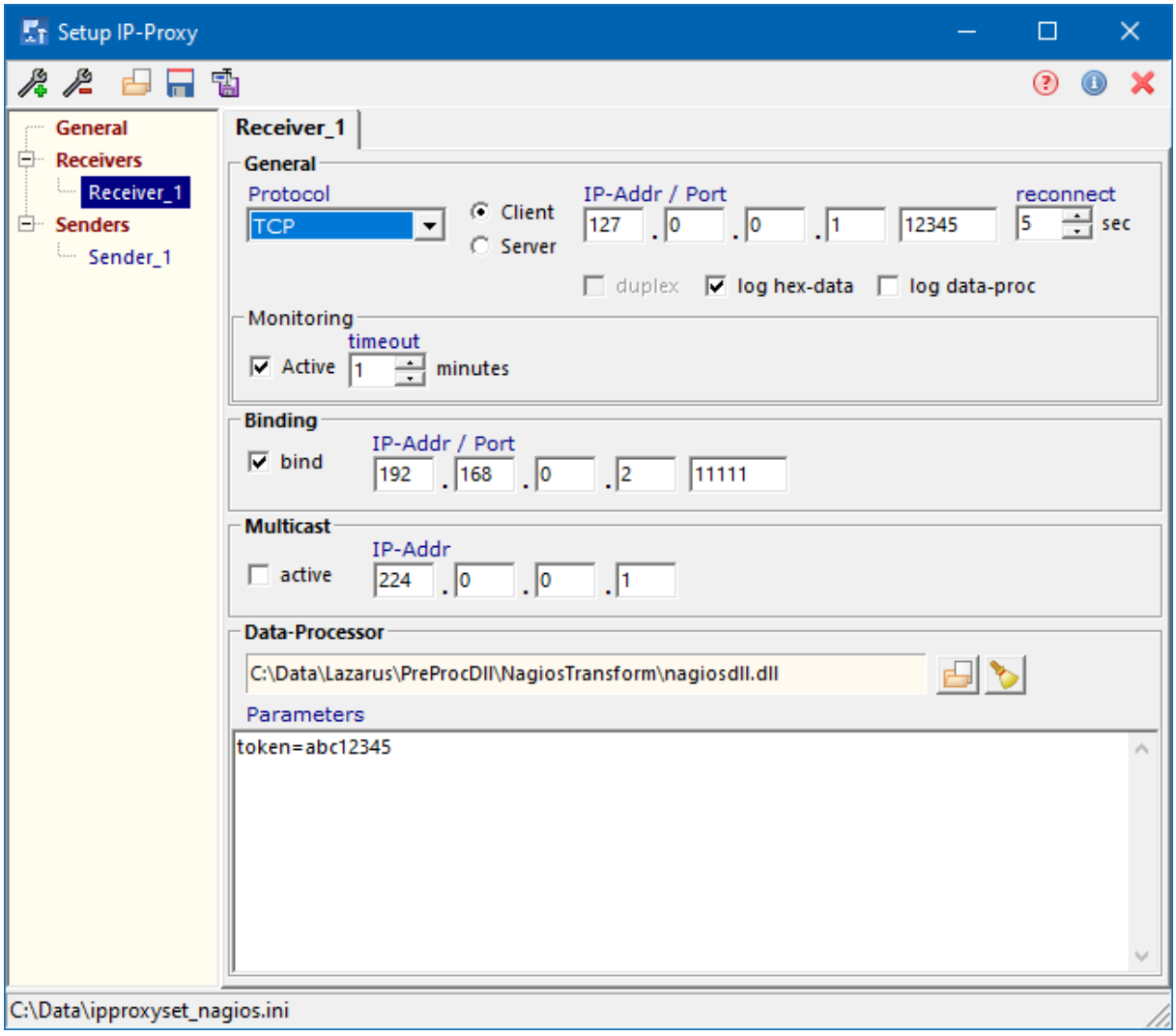
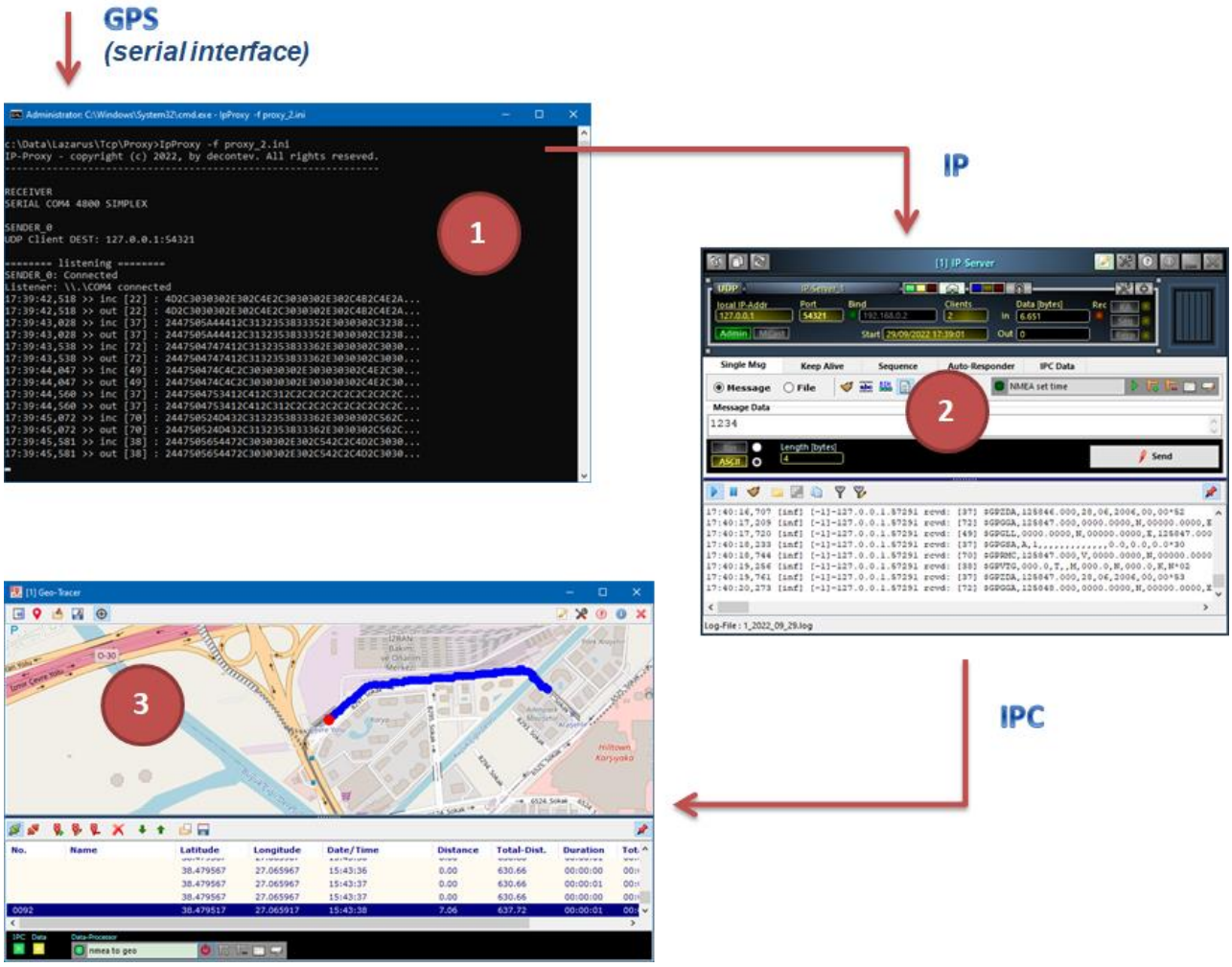


Figure 9: IP-Proxy Setup-Tool – receiver/sender configuration

1.5 Applications

There are a lot of applications imaginable for using an IP-Proxy. In following 3 possibly applications are described.

GPS Data Evaluation

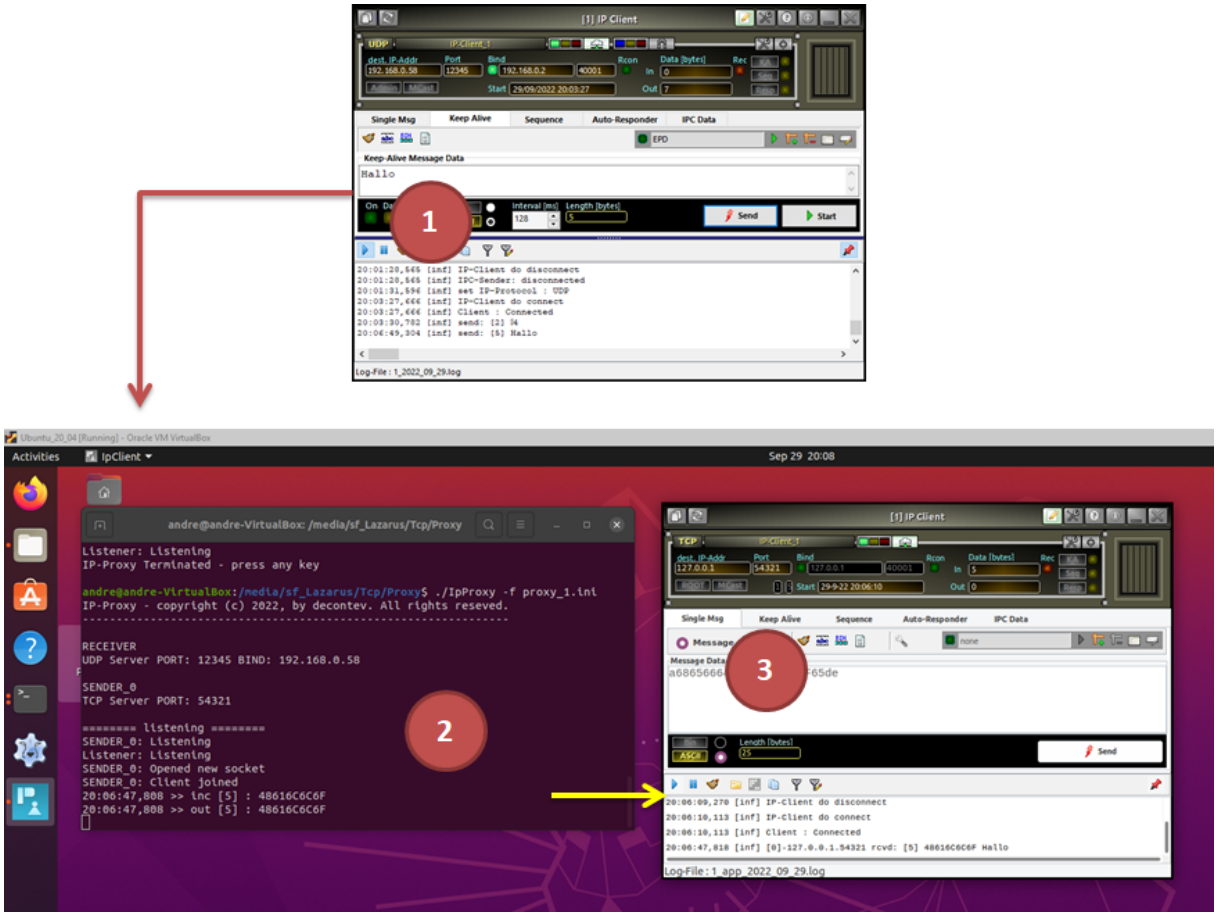


[1] IP-Proxy receives GPS data from a serial interface (e.g. GPS mouse) and redirects these data via IP/UDP to a remote location

[2] on remote side an IP-Server application receives the GPS data and forwards these data via Inter-Process Communication (IPC) to a Geo-Tracer application

[3] the Geo-Tracer application visualize the GPS data on a map

Protocol Conversion

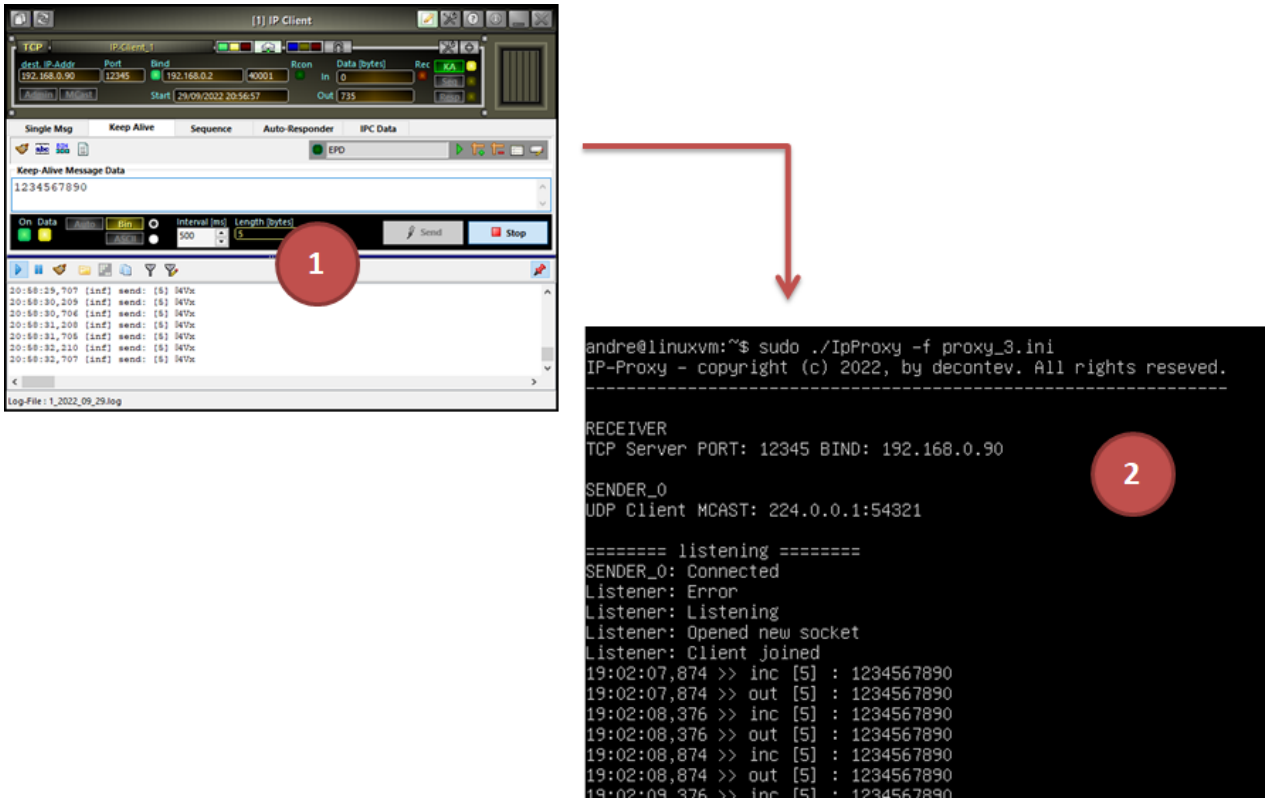


[1] an UDP-Client application, running on a Windows system, sends out data via IP/UDP thereby these data shall be processed by a TCP-Client application on a Linux system

[2] on the Linux system an IP-Proxy receives the data from the UDP-Client and redirects these via IP to a TCP-Client application on the same system

[3] the TCP-Client application on Linux system processes the data

Data for a Multicast-Group on Linux Server



[1] a TCP-Client application, running on a Windows system, sends out data via IP/TCP thereby these data shall be provided to a multicast-group on a Linux-Server system without graphical user interface (GUI)

[2] on the Linux-Server system, an IP-Proxy receives the data from the TCP-Client and forwards this data via IP/UDP to the multicast-group defined on this system