

User Manual

Prilog Alarm & Event logger



Table of contents

1. INTRODUCTION TO PRILOG	6
1.1. PRILOG WRITER / LOGGER.....	6
1.2. PRILOG READER / VIEWER.....	6
2. HISTORY, CONTACT, COPYRIGHTS AND CREDITS.....	7
3. PRILOG INSTALLATION.....	8
3.1. OVERVIEW	8
3.2. INSTALLATION PROCEDURE	9
3.3. INSTALLATION OF ADOBE ACROBAT READER (OPTIONAL)	9
3.4. AFTER INSTALLATION	9
3.5. FILES INSTALLED AND USED BY PRILOG	10
3.6. UPGRADE FROM EARLIER VERSIONS	10
3.7. LICENSING	11
4. PRILOG WRITER CONFIGURATION, STEP 1: DATA FILE SETUP	12
5. PRILOG WRITER CONFIGURATION, STEP 2: COMMUNICATIONS SETUP.....	13
5.1. COMMON PROPERTIES.....	13
5.2. COM PORT	14
5.3. TCP / IP CLIENT AND SERVER PORT TYPE	15
5.4. LPD SERVER SETUP	15
5.4.1. <i>Background on Unix printing</i>	16
5.4.2. <i>How to set up Windows to send print jobs to Prilog over network</i>	18
5.5. TEXT FILE PORT TYPE	19
5.6. PRILOG PORT TYPE.....	20
5.7. EMAIL PORT TYPE	20
5.8. FTP CLIENT PORT TYPE.....	21
5.9. PING PORT TYPE.....	22
5.10. STATUS PORT TYPE	23
5.11. UDP PORT TYPE.....	23
6. PRILOG WRITER CONFIGURATION, STEP 3: FILTER SETUP	25
6.1. GENERAL	25
6.2. EXCLUDE CHARACTERS FILTER.....	25
6.3. REPLACE CHARACTERS FILTER.....	26
6.4. FORMAT STRING FILTER	26
6.5. MARK PORT FILTER.....	26
6.6. INSERT TIME FILTER	26
6.7. COLOR MARK FILTER.....	26
6.8. LINE EXTRACTOR FILTER	26
6.9. REPLACE WORDS FILTER	27
6.10. EXTRACT TIME FILTER	27
6.11. ACTION FILTER	27
6.12. EXCLUDE PORT FILTER.....	28
7. DETAILS ON ACTION FILTER.....	29
7.1. GENERAL	29
7.2. DEFINING A SEARCH CRITERIA	29
7.3. COLOR GROUP ENABLE OPTIONS.....	31
7.4. THE "No" (NUMBER) COLUMN.....	31
7.5. THE "Use" COLUMN	31
7.6. REFRESH (READ FILES) BUTTON	31
7.7. ACTIONS	31
7.7.1. <i>Send email</i>	32
7.7.2. <i>Send SMS</i>	33

7.7.3. Play sound.....	34
7.7.4. Insert line.....	34
7.7.5. Say	34
7.7.6. Run.....	35
7.7.7. Enable printer	36
7.7.8. Disable printer	36
7.7.9. Enable SMS.....	36
7.7.10. Disable SMS.....	36
7.7.11. Delete Line.....	37
7.7.12. HTTP Post.....	37
7.7.13. Print Line.....	38
7.7.14. Write Line	39
7.7.15. Replace LX.....	40
7.8. DIALOG SETTINGS ACTIVATION	40
8. LINE EXTRACTOR FILTER (LX).....	41
8.1. USING VALUES FROM THE LX DATABASE.....	42
8.1.1. LX with the Insert Line filter.....	42
8.1.2. LX with the Replace words filter	42
8.1.3. LX with the HTTP Post filter	42
8.1.4. LX with the Write Line filter.....	42
8.2. CONFIGURING THE LX FILTER.....	43
8.3. CONFIGURING A FIELD	44
8.4. CONFIGURING A RULE	44
8.4.1. Position rules	45
8.4.2. TRUE / FALSE rules.....	47
8.5. LX MODEL1 AND MODEL2.....	48
8.5.1. Model1 example	50
8.5.2. Model2 example	50
8.5.3. Model1 configuration	50
8.5.4. Model2 configuration	50
8.6. LX STEP BY STEP CONFIGURATION WALKTHROUGH	51
8.6.1. Defining the comports.....	51
8.6.2. Defining the filters.....	52
9. SMS	56
10. GENERAL USAGE OF PRILOG WRITER / LOGGER.....	58
10.1. GENERAL INFORMATION.....	58
10.2. MENU ITEMS.....	58
10.2.1. File / Communication setup.....	58
10.2.2. File / Filter setup.....	58
10.2.3. File / Data file setup.....	58
10.2.4. File / Options	58
10.2.5. File / Debug mode settings.....	60
10.2.6. File / Block input from I/O ports	60
10.2.7. File / Exit.....	60
10.2.8. View / Serial port status.....	60
10.2.9. View / Line extraction database.....	60
10.2.10. Test / Generate lines	60
10.2.11. Help / Contents.....	60
10.2.12. Help / License information	60
10.2.13. Help / About.....	60
11. GENERAL USAGE OF PRILOG READER / VIEWER.....	61
11.1. MENU ITEMS.....	61
11.1.1. File / Open another channel	61
11.1.2. File / Save settings on exit.....	61
11.1.3. File / Stay on top.....	61

11.1.4. File / Analyze	61
11.1.5. File / View photo	61
11.1.6. File / Font setup	61
11.1.7. File / Set default font	61
11.1.8. File / Only fixed pitch fonts	61
11.1.9. File / Export	61
11.1.10. File / Print page	61
11.1.11. File / Print setup	62
11.1.12. File / Exit	62
11.1.13. Edit / Copy selection	62
11.1.14. Edit / Copy whole page	62
11.1.15. Edit / Find	62
11.1.16. Mode / True mode	62
11.1.17. Mode / Manual mode	62
11.1.18. Channels	63
11.1.19. Help / Contents	63
11.1.20. Help / License information	63
11.2. OBTAINING ASCII INFORMATION ON A CHARACTER	63
11.3. COLOR SELECTION FEATURE	63
11.4. SEARCH	64
11.4.1. Limit to color-selection	65
11.4.2. Search from which position?	65
11.4.3. Executing searches	65
11.4.4. Saving search results	65
11.4.5. Loading a saved search	65
11.5. THE LIST FUNCTION	66
11.6. HOW TO GO BACK TO THE LAST FOUND LINE	66
11.7. EXPORT	67
11.7.1. General	67
11.7.2. Export options	67
11.7.3. Exporting number of lines:	67
11.7.4. Export time backwards	68
11.7.5. Export time period	68
11.7.6. Options	68
11.7.7. Export file formats	68
11.7.8. Export filename	69
11.7.9. Executing the export	69
11.8. STATUS BAR HELP	69
12. CONFIGURING A READER (LOCALLY OR ON A NETWORK PC)	70
12.1. CONFIGURING A NETWORK READER	70
12.2. HOW TO CREATE A SHORTCUT TO A PRILOG CHANNEL (LOCALLY OR VIA NETWORK)	71
13. ESCAPE SEQUENCES AND SPECIAL CHARACTERS	72
14. ANALYZE FUNCTION	74
15. WORD DATABASE	75
16. PHOTO FUNCTION	76
17. HARDCOPY PRINTER	77
18. SCHEDULER	78
19. UNINSTALLING PRILOG	79
19.1. NORMAL UNINSTALL (PREFERRED)	79
19.2. MANUAL UNINSTALL	79

20. TROUBLESHOOTING COMMUNICATIONS	80
20.1. IF YOU CAN'T GET PRILOG TO OPEN THE COMPORT.....	80
20.2. PRILOG WILL OPEN THE COMPORT, BUT WON'T PRINT ANY DATA ON THE SCREEN	80
20.3. TROUBLESHOOTING NETWORKING AND TCP/IP PRINTING.....	81
21. SERIAL COMMUNICATION SPECIFICATIONS	82
22. FREQUENTLY ASKED QUESTIONS (FAQ).....	83

1. Introduction to Prilog

Prilog is an abbreviation of PRINTER LOGGER and is a replacement of log-printers. It is designed for logging text lines to hard disk from various sources. Prilog is maintenance free and offers extensive user features such as filtering, searching, listing, printing and exporting. Its main logging feature makes it the ideal 'black box' for storing alarms and events. Prilog also has advanced features like line modification based on content, automatic database lookup by word under mouse and line occurrence frequency analysis. In addition, Prilog may generate emails, extract and replace content based on variables, send SMS messages to mobile phone and has a text to speech function.

Prilog may receive text input from whatever source, hardware platform, operating system or software that sends text. Currently supported hardware and protocols are: Comport (RS-232/422/485 etc.), TCP/IP (as client or server), Unix lpd printer, another Prilog writer channel, email, ftp (file transfer protocol) Prilog may be shared among the users in a network so that any number of viewers may see the same data simultaneously.

Prilog is especially useful for logging alarms and events from process automation equipment or any other equipment that continually or periodically outputs a text log. Prilog consists of two programs:

1.1. Prilog writer / logger

This program collects data input port(s) and stores the data on a single file – a round buffer file. This file is created when you configure Prilog for the first time and only the hard disk size and file system limits its size. The file size is fixed. Prilog supports 64 bits file addressing, allowing 'unlimited' storage size (theoretically 16 exabytes, depending on underlying file system)

When the entire file is filled with data lines, Prilog writes further at the start of the file again - thus overwriting the oldest data. This makes Prilog maintenance-free because it will never run out of disk-space. It also makes Prilog very fast because it directly accesses the disk and does not have to access a database-layer to read/write its data. Prilog data may also be 'mirrored' to other physical locations to ensure availability of the data in case of a disaster.

The data file is in a proprietary format and difficult for unauthorized persons to modify. When Prilog is running, it is impossible to modify it by other programs or delete it. This increases the security.

The Prilog writer is configured at installation, then left running without any user interface. Normally the program should be run minimized because there are few user-functions in this program. Several Prilog programs can run simultaneously allowing 'unlimited' number of channels at each computer.

1.2. Prilog reader / viewer

This is the user program. It lets the user inspect, search, list, analyze, export and print Prilog data. It can connect to a local Prilog installation (on the same physical computer) or a remote Prilog in the network.

Several programs can run simultaneously for viewing different channels. Each program connects to one Prilog writer at a time, but a "hot" switching capability exist so that one can quickly open other channels.

2. History, contact, copyrights and credits

Prilog was first created to replace alarm printers. This product has over the years solved many problems all over the world. First developed for the marine industry, Prilog has also gained widespread use in other industries.

From solving the practical problems with a matrix printer, Prilog has also helped many times to quickly find the real cause that lead up to a problem or an emergency situation. While maintaining the basic simple philosophy for an easy user interface, Prilog has been extended to include many advanced features that have a real practical use in an ever more complex world. Prilog is continually enhanced to meet the current needs. The development cycle is very short from idea to implementation, and any suggestion for improvements is always welcomed.

Prilog is a commercial product and only sold by approved vendors. Any illegal use or sale is strictly forbidden. Modification, reverse engineering, reverse compiling or disassembly of this software and/or hardware is prohibited.

Vendor:

Høglund Marine Automation AS
Stålveien 11
N-3124 Tønsberg
Norway

Internet: <http://www.hma.no>
General inquiries: prilog@hma.no
Support: support@hma.no
Phone: +47 33414150
Fax: +47 33414946

Microsoft and the Microsoft logo are registered trademarks of the Microsoft Corporation.
Windows and the Windows logo are trademarks of Microsoft Corporation.
All products referenced throughout this guide are registered trademarks of their respective manufacturers.

Thanks to Nils Haeck, Netherlands, for his XML component.

Changes in this documentation can be made without warning and is not implying any obligations for the vendor or software producer. No part of this document may be reproduced, transmitted, transcribed or translated into any language (natural or computer) in any form by any means, without the written permission of the vendor or software producer.

IN NO EVENT WILL THE VENDOR OR SOFTWARE PRODUCER BE LIABLE TO YOU FOR ANY DAMAGES, INCLUDING BUT NOT LIMITED TO ANY LOST PROFITS, LOST SAVINGS, COMMERCIAL DAMAGE OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THIS PROGRAM, OR FOR ANY CLAIM BY ANY OTHER PARTY.

3. Prilog installation

3.1. Overview

Prilog can be installed on any Windows operating system ranging from Windows 98x, ME, NT, XP, 200X and later. Recommended platform is Windows XP. Prilog does not require any known service packs or patches.

Note: The TextToSpeech function lets Prilog 'say' certain text at specific search criteria. The runtime binaries for this is available free of cost, but must be installed separately. It also requires a special compile of Prilog (also free of extra charge). Contact the vendor for more information.

Note: For SMS sending, see own chapter for how to buy this service and other technical details.

The Prilog installation program installs one instance of Prilog on a specific directory on the hard disk. This directory is where the PRILOG.DAT file resides where the data is logged (the large data file). The installation program will install both the logger (PRILOG.EXE) and the viewer (PRILREAD.EXE). A complete installation is therefore necessary for:

- A) Each Prilog writer / logger. Meaning more precisely each data file because each data file can log text from various input ports if desirable.
- B) Each Prilog reader / viewer (each PC). Only *one* installation is necessary on a network viewer-PC, because this can "switch" between different channels, *or* be run more than one time against different writers / loggers (by means of specifying the base directory as parameter to PRILREAD.EXE).

Note that although one Prilog writer may receive text from more than one input port, sometimes this is not desired because it mixes the lines together and can be confusing. Therefore separate Prilog installations (as in point 'A' above) would be sometimes advisable.

Prilog may be uninstalled using the standard uninstall feature of Windows. This will not delete any data file created.

Prilog may be located anywhere you want to, but it would be advisable to store it under the root directory on a local hard disk, for example using the following scheme:

Example install directory (base directory)	Description
C:\ALARMS1	Base directory for Alarms system one (e.g. source=COM1)
C:\EVENTS1	Base directory for events channel 1 (e.g. source=COM2)
C:\EVENTS2	Base directory for events channel 2 (e.g. source=TCP/IP)

The directory where the channel resides is called the *base directory*. All files regarding to that Prilog installation is installed under this directory.

3.2. Installation procedure

Prilog should be installed as administrator or a user with administrator rights. Prilog is normally distributed as a single self-extracting executable on the format "Pril698.exe" (example for version 6.98). In this case, start this program by double-clicking it as a normal program to start installation. If Prilog is distributed on any other format, please follow the guidelines in each specific delivery.

Step-by-step Prilog installation:

1. Start the installation program (PrilNNN.exe) by double clicking it in the Windows explorer.
2. Press ok to install.
3. Change the default installation directory if desired (the Prilog base directory).
The default location is "C:\Prilog". This is the program directory where all Prilog files are kept. You should give the directory a name that corresponds to the Prilog channel name, for example "C:\Alarms". You can press the "Browse" button to change the location.
4. Enter the start menu name for this Prilog installation.
You would normally not change the default value, but it is recommended to change it to a logical name for this Prilog installation, especially if installing more than one instance of Prilog.

The files are then decompressed and copied to their respective destinations. When setup is complete, you will be asked to click finish to end setup.

3.3. Installation of Adobe Acrobat reader (optional)

All documentation for Prilog comes in the Adobe Acrobat format (*.PDF files). To read these files, the Acrobat reader program is needed. This program is available free-of-charge from the Adobe web site www.adobe.com. It is freely re-distributable and available in many different languages and for many different operating systems. On the distribution disk(s) of Prilog, the installation program for the English Acrobat Reader is included.

3.4. After installation

In addition to copy the install files to the destination directory, some shortcuts are created in the program manager folder on the desktop. The most important ones is a shortcut to the Prilog writer / logger and a shortcut to the Prilog reader / viewer. You use these shortcuts by going to the start-menu with the mouse, then "Programs", then the folder name you selected during installation. Note that you may create a shortcut to Prilog in the "startup" folder under the Start Menu / Programs to auto-start Prilog.

The next thing to do is to activate the license for Prilog (if not testing / demonstrating). This is explained in the next chapter. Afterwards, Prilog should be configured as explained later in this document.

Now, start the Prilog writer / logger for activating license and/or configuration.

3.5. Files installed and used by Prilog

The installation program copies a few programs and data files into the installation directory. No DLL's, ActiveX components or any other file is copied to other location on the hard disk apart from the installation directory. However, after configuring the data file by running the Prilog writer /logger for the first time, the files below are created:

Data file	Destination directory	Explanation
PRILOG.XML file	Base directory	A small text file containing the initialization settings for this Prilog channel. It is readable and editable (with caution – always make a backup copy).
PRILOG.DAT file	Base directory	Large binary data file with all the logged data-lines.
PRILOG.PEK file	Base directory	Small binary file with a decimal number, starting at 0 indicating which position in the data file that stored the last line received.

NOTE: These files will *not* be deleted when you remove the software from the control panel. All the other files will be deleted.

3.6. Upgrade from earlier versions

Always take a backup of all files from the running Prilog when upgrading to new versions. Prilog may be upgraded from version 5.xx by first deinstalling the earlier version from the control panel (or deleting all files except *.DAT, *.PEK, *.INI). Then install the new Prilog version. The uninstall program does not delete any data file, only program files. Prilog may also be installed directly over the old version without problems.

Another approach is to install the new Prilog in its own directory. This will not interfere with the old version in any way – not even the license dongle (hasp key). Version 6.xx does not use any hardware dongle (hasp key). A software-based license must be installed. See own chapter for details.

Upgrading from earlier versions than 5.xx is possible with a conversion program available upon request.

3.7. Licensing

You must enter a valid license code to be able to use Prilog – otherwise Prilog will only run a short time for demonstration or test purposes. A good time to do this is right after installing Prilog. You only need to do it once for the whole computer. If it is the first time you use Prilog, start the Prilog writer / logger first and configure a data file before opening the license dialog from the menu.

A Prilog license is a pure software license obtained from the vendor. This licensing method is an increasingly popular method for license control and many companies use its principle. It is easy to handle and robust. Both the reader and the writer have the menu item Help / License information. Either of these could be used to enter a license code or upgrade an existing license.

The license system works like this:

1. The customer sends a license request including the desired details of the license and the 'Code1' generated by Prilog on the computer to be used.
2. The vendor returns the 'Code2', which the customer enters into Prilog to activate the license.

The Code1 and Code2 are both 8 letter strings and easy to handle. Code1 is a scrambled version of the PC's (first) MAC address, or – if missing – the hard disk serial number – thus not containing any confidential information. The Code2 will then 'open' this PC for the license information requested. The Code2 is a checksum of all license information, so that it will only work with the license details given to the vendor.

The license is 'global' to the computer: There is only one license, and it applies to the whole computer, including all running Prilog's (readers and writers). Installing a new license later overwrites the old one. Licenses are therefore not accumulated, but replaced.

The actual license details are these:

Number of writers	The maximum number allowed of simultaneously running Prilog Writers on this computer.
Number of readers	The maximum number allowed of simultaneously running Prilog Readers on this computer.
Number of serial ports	The maximum number allowed of serial ports of <i>each</i> instance of a running Prilog Writer on this computer.
Prilog version	The highest version number allowed of any running Prilog Writer or Reader on this computer.
Code1	Computed by Prilog (read-only)
Code2	Computed by vendor, not by customer (write-only)

If you are going to install Prilog on a remote site and do not have access to the computer, it is important to note that Code2 can't be calculated until actually running Prilog on that computer. This is because Code1 must be generated on a Prilog running on the actual computer. Therefore, to obtain a license you must have access to email, phone, fax, internet etc. to be able to obtain a license from the vendor when you are on-site.

When a license is activated ok, it is shown in the upper part of the window in the 'Currently active license' box. In the upper right corner, Prilog shows the current number of running Prilog's. To upgrade a license, please follow the same procedure as with a new license.

The license details are stored as plain text string in the registry at this key:

HKLM\Software\Prilog Hoglund Marine Automation\License

If you are having problems with a license, this registry key may be deleted by the 'delete' button in the license form or manually with the REGEDIT.EXE or REGEDT32.EXE programs.

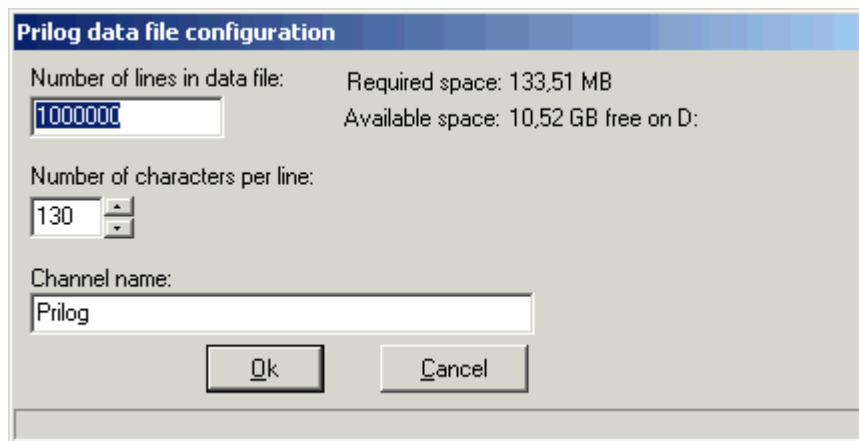
A 'Valid until' license gives the computer unlimited access until a given date.

4. Prilog Writer configuration, step 1: Data file setup

To completely configure the Prilog writer, you must cover these topics:

- Step 1: Data file setup [this chapter]
- Step 2: Communication setup [later chapter]
- Step 3: Filter setup [later chapter]

The first task is to create the main data file PRILOG.DAT. You do that with the Prilog writer /logger. The first time you run this program, you are presented with this dialog box:

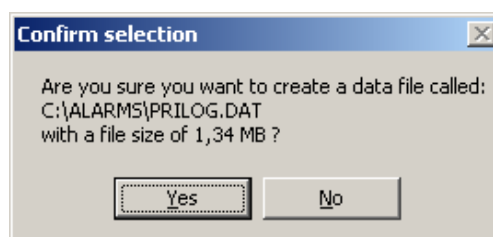


The dialog box is titled "Prilog data file configuration". It contains the following fields and labels:

- Number of lines in data file:** A text box containing "1000000".
- Required space:** 133,51 MB
- Available space:** 10,52 GB free on D:
- Number of characters per line:** A spin box set to "130".
- Channel name:** A text box containing "Prilog".
- Buttons: "Ok" and "Cancel".

You must change these settings according to local configuration!

Element in dialog box	Explanation
Number of lines in data file	This number is the total number of lines you are going to store at one time in the round buffer file. It will be the maximum number of lines possibly stored by Prilog. When Prilog reaches this limit, the oldest lines get overwritten. This number may later be changed if you would like to expand or shrink the PRILOG.DAT file size.
Number of characters per line	The maximum number of characters a stored line may occupy in the round buffer file. Use your experience with the actual data to determine this value so that Prilog will log the whole line and not too much (and waste space). This value may later be changed – either increased to make room for more data per line or decreased - which means loss of data.
Channel name	A text field that let's you give a logical, short name for the channel - most precisely defining its purpose. The channel name should be unique throughout the system of interconnected Prilog computers.
Required space	Shows the size of the data file PRILOG.DAT if you choose to create it by pressing the ok button.
Available space	Shows the available (free) space on the hard disk drive (partition) where Prilog is installed <i>before</i> creating the data file. It is advisable to leave some space free on the disk for other purposes like export files and print spooling (for example 1 GB).



The dialog box is titled "Confirm selection". It contains the following text and buttons:

Are you sure you want to create a data file called:
C:\ALARMS\PRILOG.DAT
with a file size of 1,34 MB ?

Buttons: "Yes" and "No".

Press 'Yes' to accept and create the data file. 'No' returns to the dialog.

5. Prilog Writer configuration, step 2: Communications setup

This dialog controls the input ports. One or more input ports might be created and configured, also during runtime. Changes are instantly activated. 'Ok' button closes the dialog and cancel restores the configuration to the earlier setup.

Use the Add/Delete buttons to add or delete an input port. The icons in the left pane reflect the currently installed ports. Each port type has its own icon. The default is comport. See below for an explanation of each port type. Any number of ports may be defined, but the license can limit the number of ports that actually works.

If, during runtime, one would like to inspect the current status of the ports, choose View / serial port status from the main menu. This will bring up a window showing the open/closed status of the individual ports and how much data transferred over them.

5.1. Common properties

Below each input port, there are some settings that are common to all ports:

Parameter	Description
CR Char	Carriage Return character in decimal. Normally 13 (0dh), indicating the end of the line. Prilog searches for this character to find end of line. Change it if the incoming text is not normal ASCII lines.
LF Char	Line Feed character in decimal. Windows/DOS text files usually also ends with a Line Feed character (value 10). Prilog will detect line-ends if the lines ends with CR, CRLF, or LF. Modify the value of the LF character if the incoming lines does not contain normal LF-terminated lines. If the incoming lines only uses <i>one</i> line-terminating character, it is recommended to enter that character value in the CR char above, and leave LF-char to value 0.
Receive timeout/reconnect	Timeout in seconds. The timer begins to count from the last received character on that port. If the timer reaches the timeout value, the link is closed and reopened to make sure that there is no communication problem that is blocking the reception. This parameter is mostly useful for the TCPClient port. The reason for this was that some specific equipment stopped sending after a while without any apparent reason, and while the tcp/ip connection was ok. What was needed was a tcp/ip disconnect and reconnect to reenale the data stream. Use this parameter to apply this behaviour. For the rest of the port types, this has no or little meaning and practical use.
Log error message	Check this box to enable trace (debug) messages of the communication link. Use

(trace)	this for troubleshooting the link. Not all ports give much information. The amount of information is related to the complexity of the protocol.
---------	---

5.2. COM PORT

This is the communication port under Windows that Prilog is using to collect data using RS232/422/485 communications. It normally goes under the name 'COMn' – where 'n' is the port number from 1 and upward. You can only choose from comports available to Windows. To set up extra ports, you must go to the Control Panel and select Add Hardware. If you install a COM-card, which uses a standard 16550 compatible UART, Windows has built-in drivers for it. You must know and specify the I/O address and IRQ number for each UART (Comport chip). If you install a multiport I/O board, for which Windows does not have drivers, the drivers come with the board and must be installed from a disk, supplied by the vendor.

BITRATE (BPS)

This indicates the rate at which the data is collected. You must obtain this value from the setup at the communicating device at the other end. The BPS rate must be identical in both ends. The word "baud rate" is also used for this parameter, although it isn't quite precisely the same.

PARITY

This sets the sender's system parity setting. No parity is mostly used, but with parity checking (odd/even/mark/space), each character received has an add-on bit which is calculated on the 5-8 preceding bits.

DATA BITS

Sets how many bits represent one character. A standard US-ASCII character set uses 7 bits, but does not include some foreign-country characters. The normal is 8 bits per character. Anyhow, this must be identical to the setup at the other end.

STOP BITS

How many bits that indicates the stop of a character. This must be identical on both sides and is somewhat dependent on the parity and data bits.

Note:

Much equipment may require the Data Terminal Ready signal (DTR) in order to print. This is one of the RS232C signals and is therefore available electrically. Prilog activates this signal when coming online. When the cable from Prilog is connected, the sending equipment therefore knows that it is safe to send its data because there is a connected receiver. This way, lines will not be lost. Pulling the plug or terminating Prilog will lower the DTR line and by that terminate the sending from the originating equipment.

OUT CTS FLOW CONTROL

Disables or enables output Clear To Send flow control. If enabled, transmitting is suspended while the CTS input line is low. In this case, the sending system is not ready to accept data. When CTS input is high, transmitting may be started. This is to avoid buffer overflow in the other end.

OUT DSR FLOW CONTROL

Disables or enables output Data Set Read flow control

DTR MODE

Data Terminal Ready mode:

- Enabled: Enables the DTR line when the device is opened and leaves it on.
- Disabled: Disables the DTR line when the device is opened and leaves it disabled.
- Handshake: DTR Handshaking enabled. DTR line signals request to send.

DSR SENSITIVITY

Disables or enables Data Set Ready sensitivity. If enabled, the driver ignores any bytes received unless the DSR input line is high.

RTS MODE

"Request To Send" mode:

- Enabled: Enables the RTS line when port is open and leaves it on.
- Disabled: Disables the RTS line when port is open and leaves it disabled
- Handshake: Enables RTS handshaking. The driver raises the RTS line when the "type-ahead" (input) buffer is less than one-half full and lowers the RTS line when the buffer is more than three-quarters full. If handshaking is enabled
- Toggle: Specifies that the RTS line will be high if bytes are available for transmission. After all buffered bytes have been sent, the RTS line will be low. Note: This feature is not available in the standard Windows 9x comm. driver.

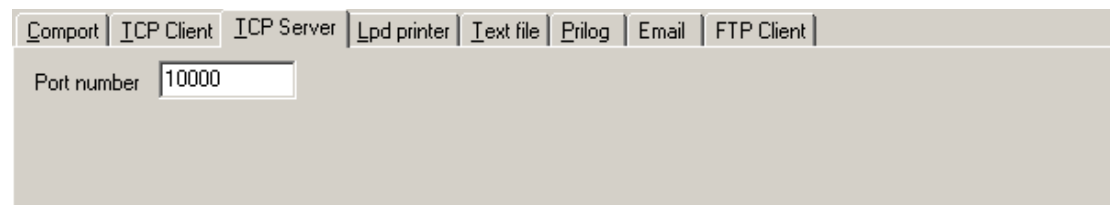
5.3. TCP / IP Client and Server port type

Prilog supports general-purpose TCP/IP client and server ports. A TCP/IP server is a TCP socket that listens for incoming connections and spawns a connection for each client that connects to it. A TCP/IP client is a TCP port that initiates and connects to a listening TCP socket. The server and client socket may or may not be physically located on the same computer. This is basically how the internet communicates. These port types can also be compared to telnet connections, although not adhering to the telnet protocol. Actually, one can telnet to a Prilog server port and enter lines.

Prilog supports these port types because much equipment may send their data over such connections. A TCP connection is a full duplex 8-bit per byte transparent data connection. This means that any data may be sent over such a connection. Prilog uses this to receive normal 8-bit ASCII values that are terminated with line-ends (CR, CRLF or LF, ASCII 13, 13+10 or 10).

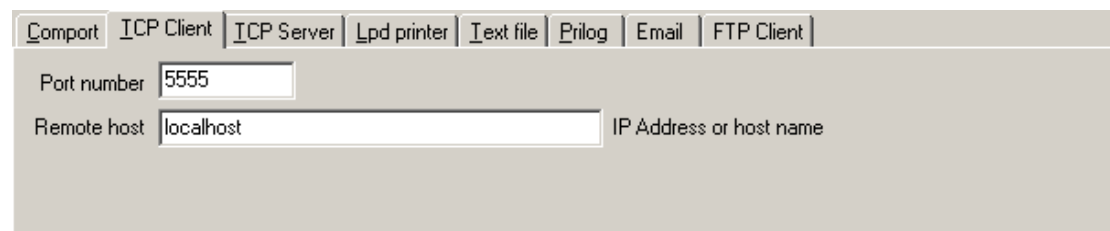
Prilog treats a TCP server or client connection in just the same manner as a normal COM port connection. If a connection is lost on a TCP client port, Prilog tries continually to re establish the connection. If a connection is lost on a TCP server port, Prilog still listens to the server port, hoping the remote client will re-connect. A TCP server can connect to any number of connecting clients.

There is no built-in firewall support or secured connection in Prilog.



The screenshot shows the 'Prilog' tab selected in a menu bar. Below the menu bar, there is a 'Port number' label followed by a text input field containing the value '10000'.

The TCP server only needs the port number on which to listen for incoming connections. In a computer with more than one IP interface (NIC), Prilog listens to all interfaces.



The screenshot shows the 'Prilog' tab selected in a menu bar. Below the menu bar, there are two labels: 'Port number' with a text input field containing '5555', and 'Remote host' with a text input field containing 'localhost'. To the right of the 'Remote host' field is the text 'IP Address or host name'.

The Port number specifies the TCP port number of the host to connect to.

The TCP client also needs a remote IP address or host name. If hostname is used, Windows should be configured with a DNS server or a host file to resolve the hostname to an IP address. Since this is not always available, specifying the IP address is the alternative.

5.4. Lpd server setup

Prilog implements the Lpd protocol that comes from the Unix environment. This is the standard protocol in Unix to print documents to a network printer (Line Printer). It uses the underlying TCP/IP protocol to achieve this.

The above settings are available to the Lpd printer setup. By default, **no configuration is required!** The parameters here are changed only in the special case where the LPD server LPD.EXE is run on another computer, which is highly unlikely.

The default behavior is to listen to all print-jobs directed to this computer. It does this by talking to the Lpd server at port 5555 (default) on the local computer (default hostname: 'localhost'). Since no printer name is specified, Prilog emulates all printers.

The actual Lpd server is the LPD.EXE file that must be running to work. It is included in the Prilog installation files and is a part of the Prilog product. Prilog automatically starts LPD.EXE when necessary. Only one LPD.EXE running is required, no matter how many Prilog writers that are running. When LPD.EXE starts, it goes down as a tray bar icon in Windows with a blue icon that can be activated by double clicking on it.

In very special cases the LPD.EXE can be running on another computer. This would be necessary, if for example the alarm generator is located on a network inaccessible to Prilog, but the LPD server computer has access to both networks. In this case enter the other computer's IP address or host name in the 'Remote host' edit field. Change the port number also if necessary. LPD.EXE uses 5555 as default, but may be changed by giving the port number as the first parameter to LPD.EXE. The LPD protocol uses normally port 515 to receive print jobs. Prilog looks up the etc/services file for the 'printer' entry. Alias for this service is 'spooler'. To change this port number, you must edit the etc/services file located normally within the windows directory structure and restart LPD.EXE.

If Prilog should listen to only one or a few printer, enter their names in the memo field below 'List printer names....' in the dialog. Enter each printer name on a separate name. The printer name is the actual remote printer name the line-generating system uses to identify the printer queue. Note that on some unix system, both a local and remote printer name can be specified.

5.4.1. Background on Unix printing

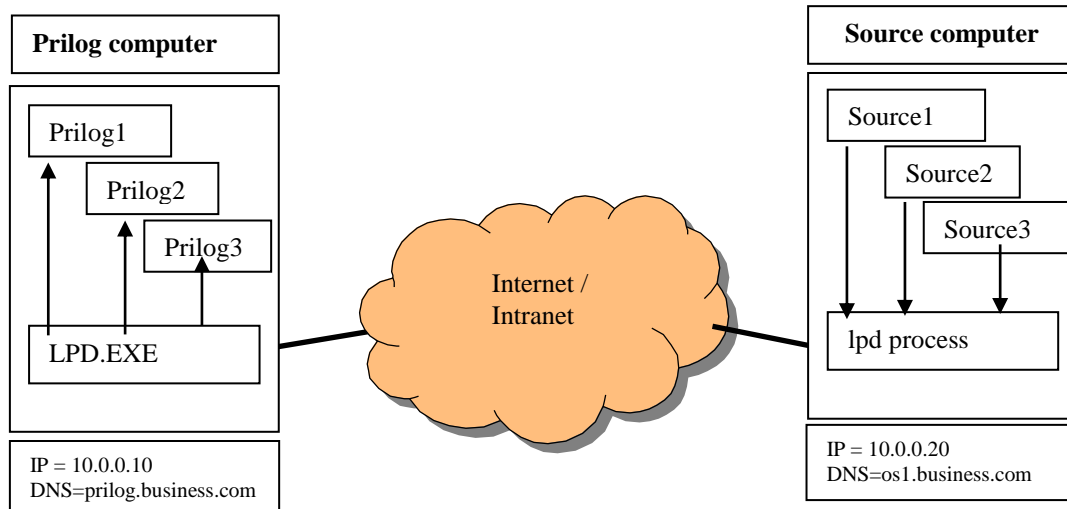
Without a network, a printer has to be connected to a physical port on the source computer to be able to use it. To limit the number of printers (and the economic cost), network printing was therefore developed. A printer is then connected physically to a print-server. The print-server in turn is connected to the network. Other workstations may then send print-jobs to its printer via the network.

The workstation has a list of printers installed. Usually this list is the /etc/printcap file. This list includes all printers, both local and remote printers (network printers). Each printer has a name and some attributes which describes its behavior. If it is a local printer, the local name of the port is specified where the printer is physically connected. If it is a remote printer, the IP-address and remote printer-name is specified.

The base software to accomplish this is called **lpd**, short for line printer daemon [a background process]. It will receive print-jobs, queue them or relay them to the print-server if the print job is sent to a remote printer.

Details on Prilog TCP/IP printing

In general, Prilog uses the Unix LP-protocol to receive its data. The implementation is based on the internet standard RFC1179, august 1990. To implement this system, we need to have an extra program running for TCP/IP-printing to work – the lpd background process, which is called LPD.EXE.



Prilog will emulate a local attached printer. In the drawing above, there are 3 different Prilog writer / loggers running emulating 3 printers. The LPD.EXE program receives print-job request from net network (source computer). In the drawing only one source computer is shown, but more than one source computer may be used.

Setup on the source computer

Source 1-3

The source programs must have one parameter defined, the printer-name that the local LP-server knows about (see below).

LP-Setup on the source computer [/etc/printcap]

The lpd process on the source computer must have defined remote printer(s). In the example above, the information will include this:

Printer 1:

Local Name of printer: Prilog1 (may be any printer name)
Local port / remote machine: 10.0.0.10 (IP-addr. or the Internet name, e.g. prilog.business.com)
Remote printer name: Prilog1 (must match the "tcp/ip printer name" on the Prilog computer)

Printer 2:

Local Name of printer: Prilog2
Local port / remote machine: 10.0.0.10
Remote printer name: Prilog2

Printer 3:

Local Name of printer: Prilog3
Local port / remote machine: 10.0.0.10
Remote printer name: Prilog3

It is recommended to have the local and remote printer name equal for simplicity. It is also recommended to keep the remote printer name short, e.g. 8 characters.

On a Unix system these printers are usually defined in a text file under /etc/printcap. Many parameters can be defined to describe a printer in this file, but you probably don't want any special formatting, but a simple raw throughput of the data.

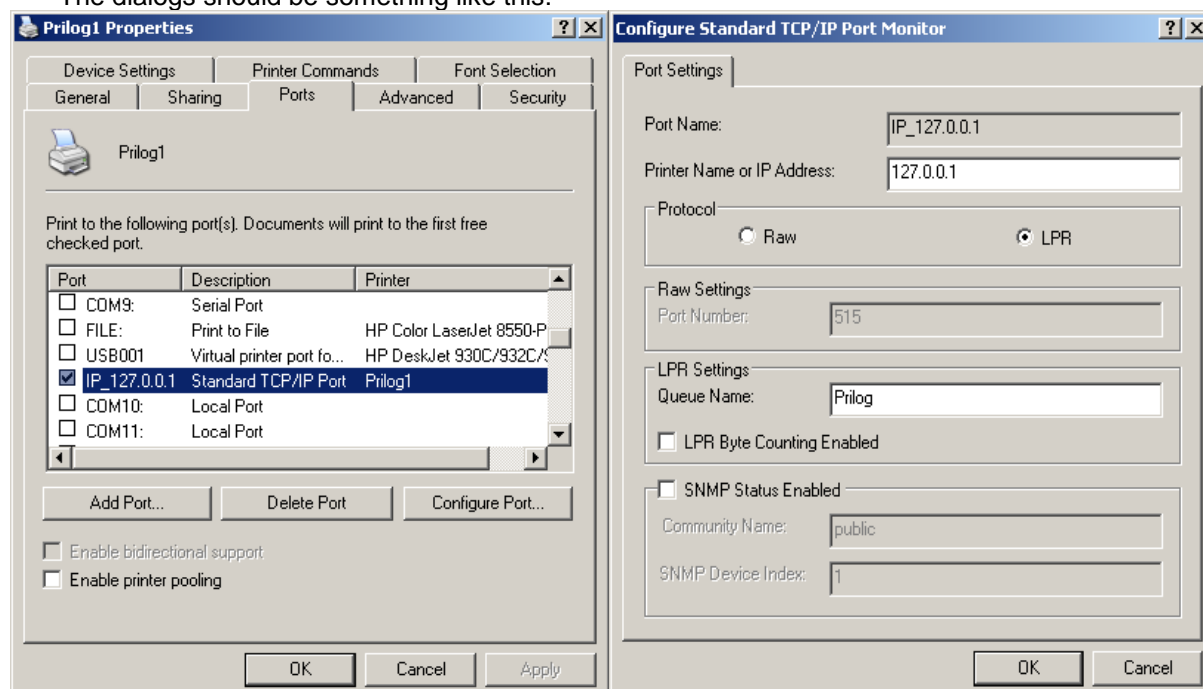
The actual setup of this varies from operating system to operating system. On the HP-UX Unix, there is a tool called *Sam* (systems administration manager), which manages this. First, log in as root in a terminal window. Then start sam with the command: **sam &** The "&" will run the program in the background.

5.4.2. How to set up Windows to send print jobs to Prilog over network

To make Windows send print jobs to Prilog using LPD, please follow the following steps:

1. On the *sender pc*: Add a new **local** printer. **Don't** detect automatically.
2. Choose the default recommended printer port, for example "LPT1"
3. Choose manufacturer "Generic" and the printer "Generic / Text Only"
4. If it asks to use the current driver, choose Yes.
5. Give the new printer a meaningful name, for example Prilog1.
6. You can make it shareable if you want to. Don't send a test page.
7. When created, choose properties on the new printer. You get a dialog window for the printer.
8. Go to the "Ports" page. You will see a checkbox on the "LPT1" port you selected above.
9. Choose the "Add port..." button.
10. Choose "Standard TCP/IP port", and you get a new dialog wizard.
11. In "Printer Name or IP Address" write the IP address of the Prilog computer (where LPD.EXE runs)
12. In "Port Name" write something meaningful, for example "IP_" plus the IP address above.
13. When you press "Next", Windows tries to communicate. It could take a while...
14. You get asked about "Device Type". Choose "Custom"
15. In the new dialog choose "LPR", (not "RAW")
16. In the same dialog choose a "Queue name", for example "Prilog" (Printer Queue Name)
17. "LPR Byte Counting Enabled" can be left unchecked. "SNMP Status Enabled" also.
18. Now press Finished/Ok/Close etc and you are back at the printer properties
19. You will see that out new port "IP_xx.xx.xx.xx" is checked (uncheck all others if necessary)
20. You are now ready to print to your new Prilog printer. Open Notepad and print to the "Prilog1" printer. You should immediately (within a second) see the line coming into Prilog.

The dialogs should be something like this:



The left dialog above is the generic dialog for the printer. The "Ports" page show which physical I/O port the printer sends its data to. In this case, the port "IP_127.0.0.1". The details for this port is shown to the right. It is a standard TCP/IP port using the LPR (LPD) protocol with the queue name "Prilog".

Note that under Windows 2000 there are two ways of setting up a TCP/IP port with a printer. Using the above method should produce a working result. If not, try re-creating the printer and the port.

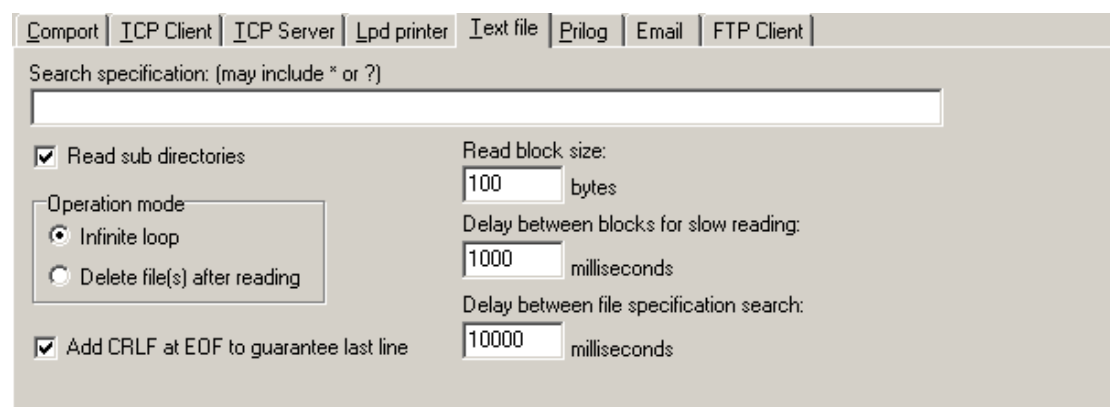
If no lines appear in Prilog, make sure the that Prilog is set up to use the "LPD" protocol in the File / Communication setup. Make also sure that the LPD.EXE program is running and isn't blocked on TCP port 515 on any firewall setting. It is listening for print jobs on port 515, and the TCP port 5555 for Prilog connections. Whenever a Prilog wants to read LPD print jobs, it does a TCP/IP connection to port 5555. You can actually do a telnet <ipaddress> 515 and check the debug information in LPD.EXE to see if a connection is made. However, you can't generate any print job this way. You can also "tap" the print jobs by logging onto the port 5555 on LPD.EXE to see the actual printer jobs that goes into Prilog (you must write a printer queue name and press enter to "log on". Use "*" for any printer queue). If you still don't see any lines, please also check in Prilog under File / Communication setup that the LPD port hasn't configured any printer names. If not, it will listen by default on **all** printer queues and log them into Prilog.

A last note: The sending system must be sending *raw text* with as little special characters and formatting as possible. Postscript, PCL etc won't work as it contains loads of commands to the printer. Therefore a "Generic / Text only" printer driver must be used. If you use a laser printer and not the "Text Only", Prilog will only display "garbage". Prilog has many functions to edit the incoming data, so formatting data *can* actually be used if set up cleverly. It is for example possible to recreate the color of the lines using the escape codes sent (simple Epson color printers for example).

5.5. Text file port type

Suppose you have a file, an ASCII dump, of a transmitting system. Maybe even a system periodically outputs a text log to a specific directory. Prilog may then read this file at specified intervals and insert its contents into Prilog as if it were a com port. Prilog may even 'listen' to a directory structure to read all files within it – using a search specification. For example to read all *.TXT files in the C:\DUMP directory each 20 seconds and delete them afterwards. Set up a text file port to achieve this.

Or, if you want to debug an output of a system, set up a text file port to read the file in an infinite loop so that when the file is read, it starts over again.



- Search specification:** Specify the full file name of the text file you want to read. If you include wildcards (* or ?), the directory is search for files matching this specification. If no wildcards are specified and the search specification is a directory, a search specification of *.* is assumed – reading all files in that directory.
- Read sub directories:** Check to also include scanning of sub-directories below the search specification directory. Only files matching the wildcards will be accepted.
- Operation Mode:** Infinite loop means that the files are not deleted and that the search specification will over and over again read the same files unless files are added or removed from 'outside' of Prilog.
Delete files(s) after reading is the opposite. After a file is read, it is deleted. In this mode Prilog then 'eats' the files in the search specification.
- Read block size:** Specify the size in bytes to be read each time.
- Delay between blocks**

for slow reading: Specify the delay in milliseconds between each read from the text file. These two last parameters controls how fast Prilog is fed with the text file.

Delay between file specification search: Specify the delay in milliseconds between each scanning of the search specification.

5.6. Prilog port type

The Prilog port type connects to another running Prilog writer and reads the lines generated into it. This port type also treats the data as if they were read from a COM port. This feature is useful for mirroring purposes.

Specify the Prilog directory where the PRILOG.XML, PRILOG.DAT and PRILOG.PEK files reside (the Prilog's base directory). When activated, all text written to the other Prilog writer is also inserted into this Prilog. Note that the color information is lost, if not regenerated by any action filter on this Prilog.

5.7. Email port type

The email port type reads emails from a specific email account on an email server using the popular POP3 protocol. It will poll the account according to the setup and feed Prilog with the text that is received. This includes headers and body text (if selected). Attachments are optionally decoded and saved to a given directory. Body text may also be saved as file. Emails are deleted after they have been read.

Email server: Enter the name or IP address of the POP3 server

User name: Enter the account name on the email server. Note that some POP3 servers require full email address including domain name.

Password: Password for the email account

Read interval: How many seconds between each login to the email server to read emails.

POP3 port: Specify the TCP port number to which the POP3 server listens.

Leave messages on server: If checked, all emails are read from the server, but they are never deleted. So if the emails are not deleted within the next 'Read interval', the same emails are read twice.

Allow into application: Check all options that you want to send into Prilog. 'Everything' overrides all other settings and simply copies the whole mail into Prilog with headers, attachments etc. The 'force' box will insert the current date if the date header is missing from the mail. 'Include prop. name' will include the property name, e.g. 'Subject:' when inserting the headers into Prilog. 'Props on same line' will join all headers on the same line instead

of each getting its own. 'Body' text will forward all plain text body text into Prilog. 'Same line' will include the body text on the same line as the headers. 'Save notification' inserts the filename into Prilog when an attachment is saved.

File actions:

Attachments directory:

Specify the directory in which you want to store attachments. Attachments are files that are MIME encoded and not have content type 'text/plain'.

Save as filename: If left blank, the filename indicated in the received mail is used. If desired, enter a filename in this field to override the filename in the email. If the mail contains more than one attachment, they will get the same filename. However, the attachment options are still in effect, so that if any of the append/prepend options are used, the files may still get a unique filename.

Note that in emails, the directory is never stored, only the filename part.

Options:

Overwrite if existing: Overwrites existing attachments if they exist.

Append YYDDMM HHMMSS: Appends the date/time to the filename

Prepend YYDDMM HHMMSS: Prepends the date/time to the filename

Sequence number: Assigns a sequence number to each attachment.

Max sequence number:

If this is checked, specify a maximum sequence number the attachments are given. When reaching this number, the sequence number will wrap back to '1'. If no maximum sequence number is used, the counter increases to a very large value, then wraps to 1.

Save body text as file to directory:

Specify the name of the directory to store body text. In case of a MIME encoded mail, these are marked with content type 'text/plain'.

'Allow give directory in subject':

Check this to be able to give the directory for attachments in the subject field of the email. If it is an absolute directory, this is used. If it is a relative directory, the 'attachment directory' in the dialog is prepended.

5.8. FTP Client port type

The FTP Client can read cyclically a file from a FTP server.

The screenshot shows the 'FTP Client' tab selected in a window with multiple tabs. The fields are as follows:

- FTP server: ftp.business.com
- User name: joe
- Password: secret
- Remote ftp file: /home/system/snapshot.txt
- Local ftp file: (leave blank to feed into application)
- Read interval: 60 Seconds
- FTP Port: 21 Default 21
- ☒ Delete remote file after reading

FTP server: Specify the remote ftp server to log into (IP address or hostname)

User name: Specify the ftp user account to log in as

Password: Specify the password for the above user name

Read interval: Specify the time, in seconds, to wait between to logins at the ftp server

FTP port: Specify the TCP port the ftp server listens to.

Remote ftp file: Specify the file name with full path of the remote file to read

Local ftp file: If simply downloading a file to store on the local disk, specify the local filename here. If this field is left blank, the contents of the file is fed into Prilog as if it were read from a com port.

Delete remote

file after reading: Checking this checkbox will make Prilog send a delete file command to the remote ftp server when the download is finished. This requires delete-access on this file for the current ftp user.

5.9. Ping port type

The screenshot shows the 'Ping' port type configuration window in Prilog. The window has a tabbed interface with tabs for Comport, ICP Client, ICP Server, Lpd printer, Text file, Prilog, Email, FTP Client, and Ping. The 'Ping' tab is selected. Inside the window, there is a list box on the left labeled 'List of hosts or IP addresses to ping:' containing the text 'www.hma.no'. To the right of the list box are several configuration fields: 'Timeout:' with a value of '5000' and unit 'ms'; 'Delay between ping loops:' with a value of '1000' and unit 'ms'; 'Status report interval:' with a value of '60' and unit 'min. (0 disables)'; a checked checkbox labeled 'Status report at startup'; 'Message when up:' with a text field containing 'Server %s is up'; and 'Message when down:' with a text field containing 'Server %s is down'.

With this port type Prilog acts like an IP monitor program that displays the status if host(s) respond to ping requests. Ping is a low level network tool that sends a message over an IP (internet) network to a given host (IP address or hostname). The host can send a ping reply indicating that it is alive. This way ping is an excellent tool for checking for the availability of servers on the net. You can set up a list of hosts or IP addresses to monitor. Prilog will then ping all the hosts in the list and send a text line to the screen when a server goes down or becomes available again.

List of hosts or IP addresses to ping:

In this field, enter - line-by-line - hosts to ping. Both IP addresses (1.2.3.4) and hostnames are allowed.

Timeout:

Enter timeout in milliseconds. If a host doesn't respond to a ping within this timeout it is 'down'.

Delay between ping loops:

Enter delay in milliseconds. This is the time Prilog waits after having pinged all hosts in the list.

This shouldn't be 0 to not stress the network unnecessary.

Status report interval:

Interval in minutes of status report of all hosts. Setting this value to greater than 0 enables periodic status reports. A status report lists all hosts and indicating if they are 'up' or 'down'

Status report interval:

Check this checkbox to enable a status report when Prilog starts up.

Message when up:

Enter a text message to be sent to Prilog when a server goes up, or is up in a status report.

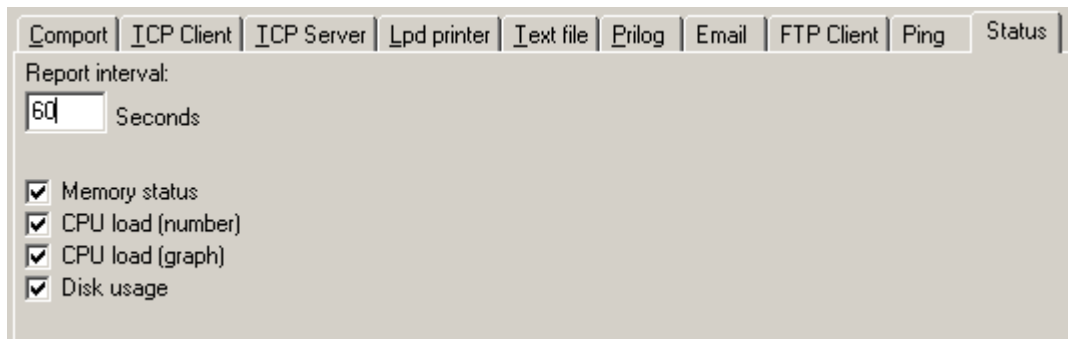
Insert a '%s' in the text (if desired) to insert the actual host name.

Message when down:

Enter a text message to be sent to Prilog when a server goes down, or is down in a status report.

Insert a '%s' in the text (if desired) to insert the actual host name.

5.10. Status port type



The status port reports various status messages from the computer. It could be used to monitor resources over time and to check that everything is ok. The following options are available:

Report interval: Time, in seconds, between each status report is printed. In a status report, every category that is checked below is included.

Memory status: Gives detailed information about the memory usage of the computer:

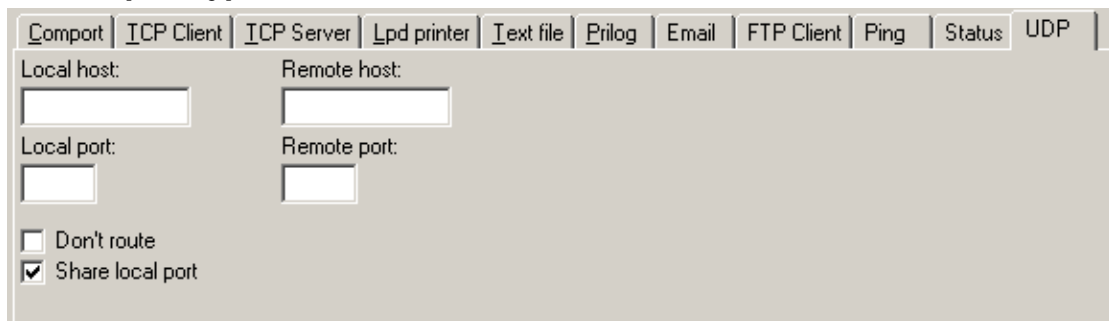
 %: Memory usage in percent
 Avail: Available physical memory in bytes
 TotPage: Total bytes of paging file
 AvailPage: Free bytes of paging file
 TotVirt: User bytes of address space
 AvailVirt: Free user bytes

CPU load (number): Shows the cpu load as percent

CPU load (graph): This will show a graph with #-signs for every percent of CPU load.

Disk usage: This will show free space on each fixed or remote disk or ramdisk.
Removable disks or CDRom's are not included.

5.11. UDP port type



The UDP port type uses the UDP / IP protocol. It is a protocol in the TCP/IP family and very similar to TCP / IP but far more simpler in the following aspects:

- It is connectionless (data is sent and forgotten, no sequence control etc.)
- Does not provide an error-free transmission
- Frequently used to broadcast data on a local area network

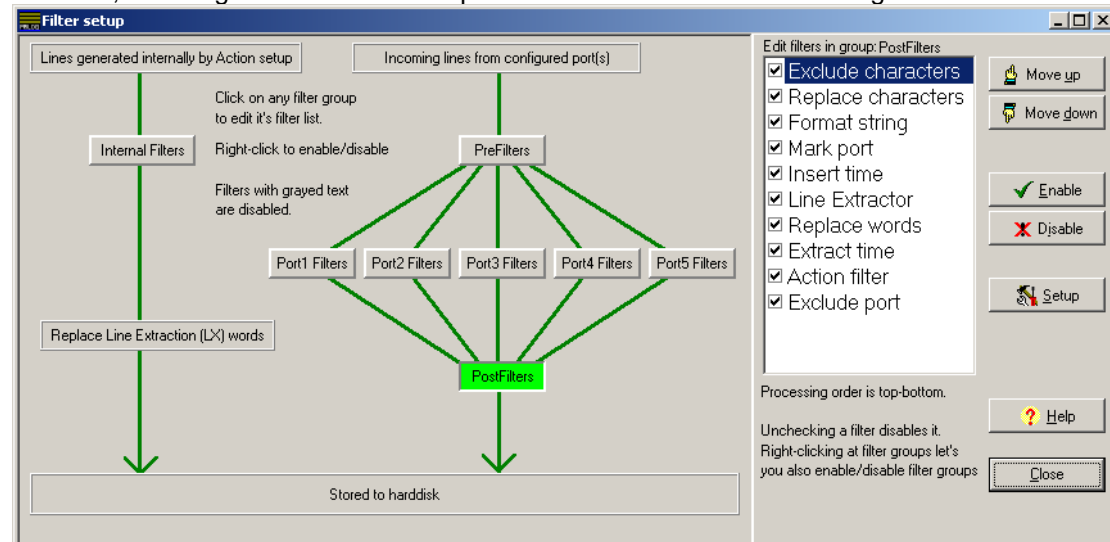
This is an ideal transmission method when a box wants to transmit data continuously to one or more recipients in a network, but without the need to ensure that each and every packet arrive. For example, a GPS receiver may transmit the GPS position continuously using UDP. The receivers in the network will not fail or generate errors if one transmission fails. It will just wait for the next and be happy with it. Each UDP datagram has a receiver address (IP address) and a port number. For Prilog to receive UDP data, these two parameters are the most important. This means that the local host and local port must always be defined. The local host should be the IP address of the network card (interface) that you want to receive the data from (or 0.0.0.0 to denote all available interfaces). The local port is the port on which to listen for data. Consult with the sending equipment to determine the local port.

- Local host: IP address (or host name) of local interface to accept data. Leave blank or enter 0.0.0.0 to listen to data on all interfaces. This setting is useful if there is more than one network card (interface) in the computer (a so called multi homed host).
- Local port: The port on which to accept incoming data. Must be the same port number the sending equipment is configured to send to. This is the most important setting to receive data into Prilog with the UDP protocol.
- Don't route: If checked, this forces the socket to send data directly to the interface without routing.
- Share local port: Check to allow more than one program to listen to the same local port. Unfortunately the default IP stack under the Windows operating system won't allow the distribution to other than the latest program that has attached itself to an UDP port.
- Remote host: Define the IP address (or host name) if this port is to send data. In the current Prilog version, no data is sent, so this option should be left blank.
- Remote port: Define the port number of the receiving system to where to send data. As above, Prilog can't send any data in this version, so leave this option blank.

6. Prilog Writer configuration, step 3: Filter Setup

6.1. General

By default, the filters don't modify the contents of the incoming lines. Several filters may be configured, however, to change the lines or do a specific action. This is the filter dialog:



The left hand side contains a graphical overview of the filter groups. The green arrows indicate the flow of data lines. Click on the filter group button that you want to configure. This button is then marked green. The right hand side lists the filters in that group. Double-clicking on a filter in this list opens the setup dialog for this filter.

All lines from the I/O ports are first run through the PreFilters. Then run through the filter group corresponding to the originating port. At last, all lines are run through the PostFilters group. If only a simple filter setup is required, using the PostFilters is recommended. If it turns out later that individual filter setting per port is needed, the port filters may then be edited.

The port filters are numbered from 1 and upwards. The numbering is the same as the numbering in the communication setup. Port1 is the first configured port. Port2 is the second configured port etc. This has nothing to do with for instance the COM port number or TCP port number.

The internal filter group is made to treat especially any internally generated lines in the action filter / insert lines function. Normally there is no need to configure this.

The 'Replace Line Extraction word' is a special filter that replaces any occurrences of the LX database words in the line with it's corresponding value taken from the current LX database. For instance, if a line contains the word 'Heading' and the LX database contains the name 'Heading' that equals value '234 DEG', then the word 'Heading' will be replaced with '234 DEG'.

Within a filter group, a line is run through the filters in the order that they appear. To configure a filter, first select a filter group, and then double click on the filter in the right pane. Note that it is possible to disable or enable an entire filter group by right clicking on its button. The button text is shown gray if it is disabled. An individual filter may also be disabled by un-checking the checkbox at its left side.

A filter in the filter-box is mark with a bold font if it contains a configuration. 'Non-bold' filters don't modify the lines or create any action.

6.2. Exclude characters filter

Here, you may check each character you want to remove from the incoming lines before they are processed with the format string. These are characters that appear as "garbage". By default, the ASCII

value 12, Form Feed, is the only character in this list. You may check or uncheck any ASCII value from 1 to 255 during runtime and the change is effective immediately.

6.3. Replace characters filter

You can here define a replacement table for single characters. In the leftmost three columns you can choose either to enter the character itself, or its ASCII value in decimal or hexadecimal (if it is difficult to enter it from the keyboard). Likewise, you enter the character you want to replace the first with in the three rightmost columns.

You may this way add and delete replace characters. You may also disable this feature by un-checking the checkbox at the bottom of the dialog.

If you don't know the ASCII value and can't enter it from the keyboard, Prilog has a useful feature to obtain this information. First make sure to log at least one line that contains the character you want to replace. Then, start the Prilog reader / viewer and click with the right mouse-button on the character. You will then be presented with the character's ASCII value in both decimal and hexadecimal.

6.4. Format string filter

This function works like earlier versions of Prilog. The format string is a character string with a length of a Prilog line containing either + signs or - signs. The purpose is to be able to edit the line received to remove unwanted space or information. A minus sign will remove a character at that position, and a plus sign will let it stay.

Example:

```
0123456789 ABCDEFGHIJKLMNOPQRSTUVWXYZ (original line as it was received)
+++++-----+ (Format string)
0123456 ABCDEFGHIJOPQRSTUVWXYZ (resulting line)
```

6.5. Mark port filter

This filter inserts text string(s) at a certain position in each line passing the filter. The text string is inserted and the text to the right is shifted rightwards.

More than one text can be defined. For example, the filter can insert the text 'ABC' at position 10, then insert the text 'DEF' at position 20.

If the idea is to tag the different I/O ports with different text, then use the 'Mark port' filters in the individual port filters.

6.6. Insert time filter

This filter inserts the current date/time at the given position. It uses the Windows date/time format. Enter 0 to disable function. Position 1 is the first position in a line.

6.7. Color mark filter

This filter sets a specific color number on all the lines that passes it, regardless of the color it had before. The filter dialog has a checkbox that must be checked before this filter starts to work. When checked, the line get the color number in the edit in the dialog. Color number 0 is the default color that all lines get. Numbers above 0 indicate different color groups whose colors can be set in the action filter. The colors are common to all filters.

6.8. Line extractor filter

The Line Extractor filter (LX) is a general-purpose rule-based method of extracting data out of text files. Editing this filter brings up a big dialog allowing to visually edit a set of rules in a hierarchy. The details are explained in its own chapter.

6.9. Replace words filter

This is a list of words, each with an existing or not-existing corresponding word. If the first word is found in a line, the corresponding word is replaced with this one. If the corresponding word does not exist, the word is simply removed (i.e. replaced with an empty word).

The words may contain escape-sequences or any other special code (except file-references). This allows for removal of unwanted "noise". Note, however, that this should be processed *after* the action-setup allowing the action-setup to match these sequences or codes before they are removed.

The "replace words" dialog can handle a huge amount of word replacements. Sometimes, however, it could be easier to define this database in Excel format. Therefore, below the text "Alternative database as external file", enter the full path to an MS Excel file that holds more word replacements. These words are processed after the words in the dialog itself, and the workbook sheets are processed from left to right. The Excel file can contain any number of sheets and each sheet have the *from* words in the "A" column and the corresponding *to* word in the "B" column, starting from row 1. The sheet name can also limit the number of color groups the word substitution works for. If the sheet begins with "Color", the rest of the sheet name are color numbers separated with a "-" character that the sheet will work on. For example:

Sheet name = Color0-1-5-6-7

- will make the sheet word substitute only on lines having color numbers 0, 1, 5, 6 and 7.

If the sheet name does not begin with "color", then the sheet is enabled for all colors.

Prilog will reload the XLS file if you press the Apply button or Ok button.

The dialog accepts drag-drop operations from the Windows file explorer, so you don't have to type in the whole filename.

Check the checkbox "Trim all words from spaces before and after" if you want to make sure that there is no extra space before or after any of the words in the database by accident.

6.10. Extract time filter

This filter tries to decode a date and/or time in the actual line data. If successfully decoded, Prilog adjusts the PC clock (date or time). If the date differs, the new date is set immediately. If the time differs, there is a hysteresis value – default 5 seconds – that avoids Prilog constantly changing the PC clock. This allows the PC clock to drift a maximum of 5 seconds until the clock is adjusted the next time.

The extract time filter dialog lets you set the specific time and date format. Prilog depends on this and can't 'guess' what is the date and time in the incoming lines because there are so many different standards. Study the explanation carefully in the dialog and define the date/time standard that your incoming lines have. Note that it is possible to only define one date format or a time format.

Usually the time extraction is most useful. This can for example be used to synchronize the PC clock with the clock of the automation system.

If defined in the dialog, Prilog inserts a line into Prilog when it adjusts the date or time.

6.11. Action filter

The action-setup lets you define a search criteria and a variety of actions that are done when a line matching the search criteria arrives. Some of the possible actions are:

- Color marking the line: You define a background and foreground color of the line
- Should line be blinking? Double click this column to mark it for blinking.
- Should something happen when a line matching the search criteria arrives? Here are some example actions:
 - E-mail sending to a specific receiver via the intranet or internet
 - SMS-sending to a mobile phone with a cutout of the actual line.
 - SMS-sending may also be enabled or disabled as an action which means that for example it can be turned off during the working hours of the day.
 - Playback of a pre-recorded sound-file.
 - TextToSpeech. Enter a text line the computer 'says' on the loudspeakers.

- Insert Line. Enter text line(s) to be inserted after the line.
- Delete line – deletes all lines passing this filter
- HTTP Post

The search criteria may be large and complex with And/Or/Nand/Nor operators. See the chapter "Details on action setup" for more information.

6.12. Exclude port filter

This filter can remove all lines coming from a specific port. Mark the port desired, and the line will not be passed on the next filter or the DAT file.

This filter could be used to block any unwanted port to appear in the DAT file, but still having it processed in the earlier filters – for line extraction for instance.

7. Details on Action filter

Action setup

Here you may define colors and actions. If a text-string appears in a incoming line, the whole line's color is assigned these colors. Click or double-click cells to change values. You may also set an E-mail address, sound file to be played, or a phone number to send a SMS message.

Color group enable options:
☐ Enable all
☐ Disable all
☒ Individual settings (in 'Use' column)

SMS Enable options:
☒ SMS Enable

Ok Cancel Help Refresh (read files)

No	Use	Text 1	Oper	Text 2	Font	Back	Blink	Example	Action
1	✓	Alarm 33	AND	Pressure	Green			Example	
2	✓	Pipe 3	OR	Pipe 4			✓	Example	
3	✓	TIAL_4FF1	AndNot	Maximum	Blue			Example	
4	✓	ORfile:PRIW24.TXT	AND	FIO_771L	Pink			Example	
5	✓	ANDfile:PRIW25.TXT	OR	ANDfile:PRIW45.TXT	Brown			Example	
6	✓		OR		Red			Example	
7	✓	SMS Disactivate	OR		Yellow			Example	
8	✓	SMS Activate	OR		Yellow			Example	
9	✓	Alarm	OR		Red		✓	Example	
10	✓		OR		Yellow			Example	
11	✓		OR		Yellow			Example	
12	✓		OR		Yellow			Example	
13	✓		OR		Yellow			Example	
14	✓		OR		Yellow			Example	
15	✓		OR		Yellow			Example	
16	✓		OR		Yellow			Example	

7.1. General

Up to 255 different search criterias may be defined – with an associated font color, background color and blink status. For each of these, zero or more actions may be defined, meaning the action(s) are executed if a line that is received matches the search-criteria. If an incoming line matches more than one search criteria, all actions are executed, in the order they are defined (beginning with line no. 1 in the dialog). However, as search-criteria can give a specific foreground and background-color on the incoming line, it is the *last* search criteria that define its color.

The colors are common to all action filter groups (that is – for prefilter, postfilter, internal filter and port filters). This means that if you define the first line to give red font color, the first line will also show red in all other action filter dialogs – even if the action filter belongs to another filter group. If you want to set a unique color, please use a line that is not already in use. The 'No' column corresponds to 'Color number'.

7.2. Defining a search criteria

A Search criteria is entered in the "Text 1" and/or "Text 2" columns in the dialog. The search-criteria is one of the following five possibilities:

Text 1			Searches for the text 'Text 1'
Text 1	OR	Text 2	Searches for the text 'Text 1' OR 'Text 2'
Text 1	AND	Text 2	Searches for the text 'Text 1' AND 'Text 2'
Text 1	AND-NOT	Text 2	Searches for the text 'Text 1' AND NOT 'Text 2'
	AND-NOT	Text 2	Searches for ANYTHING BUT 'Text 2'

Here, Text 1 and Text 2 may be any combination of printable and non-printable characters or a file reference. The search is case-insensitive, meaning that "alarm" will match "ALARM", "Alarm" etc. The text can include more than one word separated with spaces, but spaces are not treated special, but as a letter subject to string matching. This means that the search expression "alarm 2" (which includes one space between "alarm" and "2") will only match lines which includes only *one* space (and not more) between the two words.

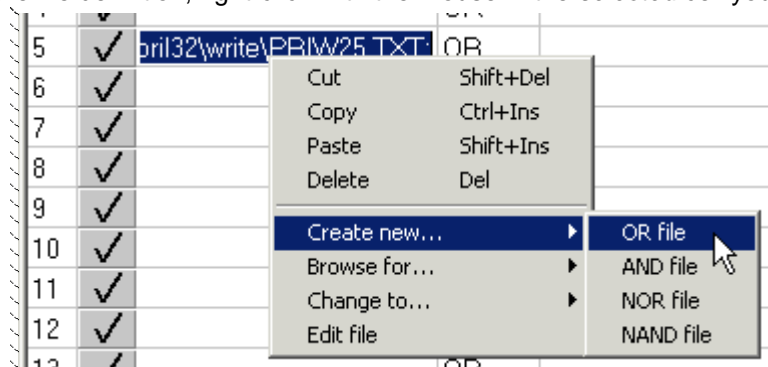
It is possible to define escape-sequences. This means that if the received text includes special printer-formatting codes, Prilog may recognize them (e.g. color codes) and do the appropriate action, for example setting a specific color. See the chapter "Escape sequences and special characters" for details.

Text 1 and/or Text 2 may also instead of being a sequence of characters be a **file name**. In this case, Prilog reads the file as a text-file. The file is an ordinary text file, editable by for example notepad. It should contain one or more lines, each with search criteria as described above, or another definition of a filename, allowing recursive processing of files (which again allows for large, complex search criteria's with many nesting-levels).

The file may be of either AND, OR, NAND or NOR-type:

- AND file There is an AND function between each line in the file
- OR file There is an OR function between each line in the file
- NAND file There is an AND function between each line in the file and afterwards, the result is inverted (true to false and false to true)
- NOR file There is an OR function between each line in the file and afterwards the result is inverted (true to false and false to true)

The file type is given by the syntax of entering the file name in the Text1 / Text2 columns. To easy create or change the file definition, right-click with the mouse in the *selected* cell you want to define:



With this popup menu, you can create a new file, browse for an existing file, change the file type or edit the existing file. The actual syntax for the file is as follows:

```
\ORFILE C:\Filename.txt;
\ANDFILE C:\Filename.txt;
\NORFILE C:\Filename.txt;
\NANDFILE C:\Filename.txt;
```

Note that the syntax begins with a backslash and ends with a semicolon.
 You can also double-click on the filename in the grid to edit its file directly.
 The 'Create new' menu creates default filenames that you might want to change.

Example search criteria:

<i>Text 1 Column</i>		<i>Text 2 Column</i>
\ANDFILE A.TXT;	AND-NOT	\ORFILE O.TXT;
<i>A.TXT contains</i>		<i>O.TXT contains:</i>
ABC		123
DEF		456
GHI		789

Resulting search criteria:

("ABC" **AND** "DEF" **AND** "GHI") **AND-NOT** ("123" **OR** "456" **OR** "789")

Note: Within a OR/AND/NOR/NAND file, the same filename syntax may be used in any of the lines to create nested levels of search expressions (gets very complex).

7.3. Color group enable options

The "Enable all"-switch will enable all 16 search criteria's, thus overriding the individual "use" column. The "Disable all"-switch will disable all 16 search criteria's, thus overriding the individual "use" column. The "Individual settings" opens for activating or deactivating individually the 16 search criteria's by means of the "use"-column. Double-click on a hatch in this line, and the corresponding search-criteria will either be enabled or disabled.

7.4. The "No" (number) column.

This is just for a simple numbering of the colors. The default color number is 0. This is given all lines that aren't recognized in any action filter. If a line matches a search criteria in an action filter, the line is given the color number as indicated in this column. A font color and background color is assigned to each color number.

7.5. The "Use" column

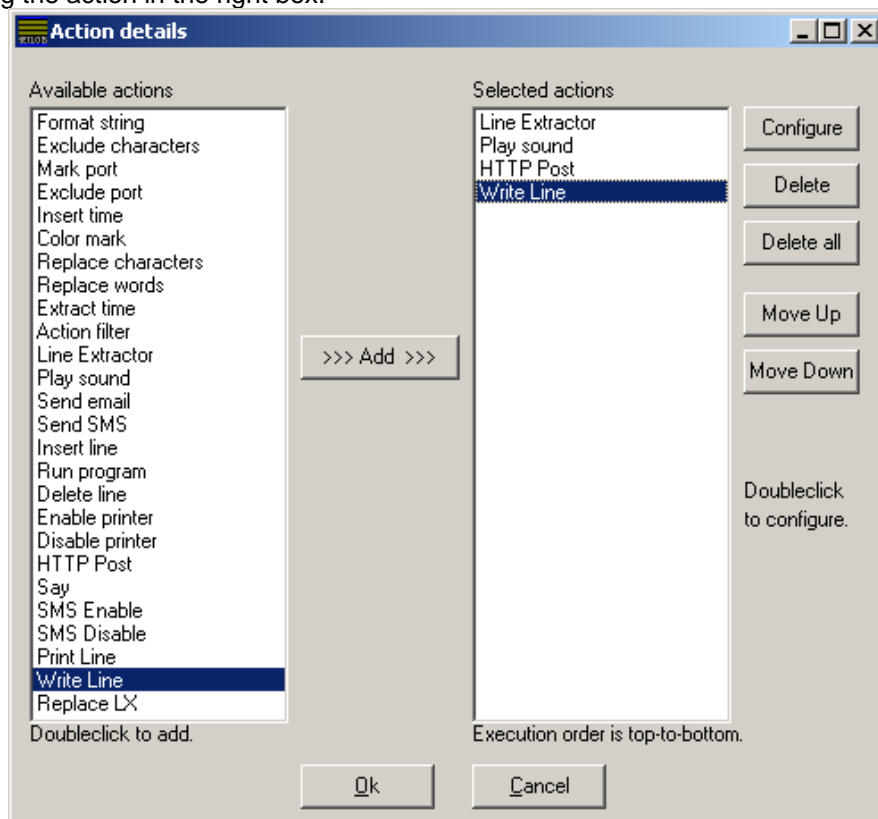
The "Use" column tells Prilog if this line (search criteria) is activated if you are using "individual settings" (see the actual dialog window). If you use "Enable all", all lines are activated regardless of the "Use" column. If you use "Disable all", all groups are de-activated regardless of the "Use" column.

7.6. Refresh (read files) button

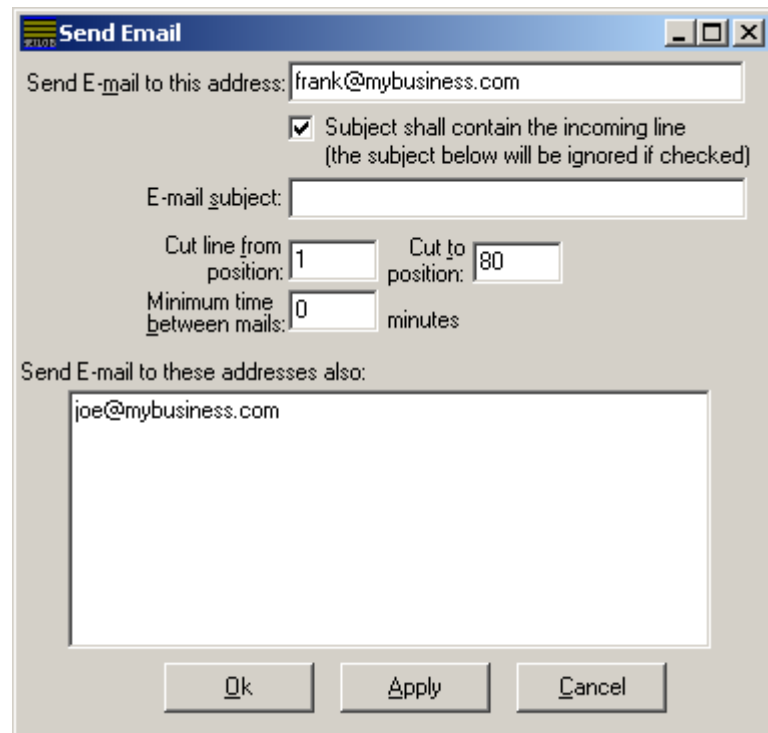
If there is a reference to an AND/OR/NAND/NOR file as shown above, the refresh button will reload these files into memory. This is useful if you are editing these files while Prilog is running and just want to activate a change after saving to disk without having to restart Prilog.

7.7. Actions

Double-click on any cell in the 'Action' column to add actions. Actions are filters or a specific action taken when a line that matches the search criteria passes it. All the normal filters are available in addition to some special ones explained below. Add and remove actions in the dialog. Available actions are on the left, selected ones on the right. Most actions have configuration settings. They are edited by double clicking the action in the right box.



7.7.1. Send email



The 'Send Email' dialog box is shown with the following fields and options:

- Send E-mail to this address:
- ☒ Subject shall contain the incoming line (the subject below will be ignored if checked)
- E-mail subject:
- Cut line from position: Cut to position:
- Minimum time between mails: minutes
- Send E-mail to these addresses also:
- Buttons: , ,

If search-criteria is a match, Prilog may send an email to a specific address given for this search criteria. Included in the message is an optional custom-subject and, as the body-text, the received line which generated the action. To limit the amount of information sent in the body-text, one may define to cut out only the most interesting part of it. This is because the e-mail function may be used to send SMS-messages for e-mail accounts that support forwarding to mobile phones.

This option requires either a direct connection to the internet/intranet or via dial up networking. Prilog uses an external program called PRILMAIL.EXE to actually send the email. It comes with the Prilog distribution files and is automatically launched when an email is to be sent. It then stays active until the computer is shut down. It starts as a tray icon in the lower right of the screen, and can be double-clicked to inspect details in the email sending.

The first time you define this action, Prilog will bring up the options dialog to allow entering an outgoing mail server, SMTP-id, sender address and reply-to address. With the exception of SMTP-id, the rest is vital to make the email sending work. Otherwise, choose File / Options in the Prilog writer to bring the email options up:



The 'Prilog options' dialog box is shown with the following fields and options:

- Prilog options
- Email options
- Email server (SMTP):
- SMTP User ID (not required by most servers):
- From email address:
- Reply to address:

Prilmail details

Prilmail.exe will be started from the Prilog directory when there are emails to be sent. Emails are generated in the OUT-MAIL directory under the Prilog directory. Prilmail cyclically scans this directory and sends it to the email server (SMTP protocol). Email server is specified in the Prilog writer under File / Options. Optionally, a file called PRILMAIL.INI may be created in the Prilog directory. The contents are as follows:

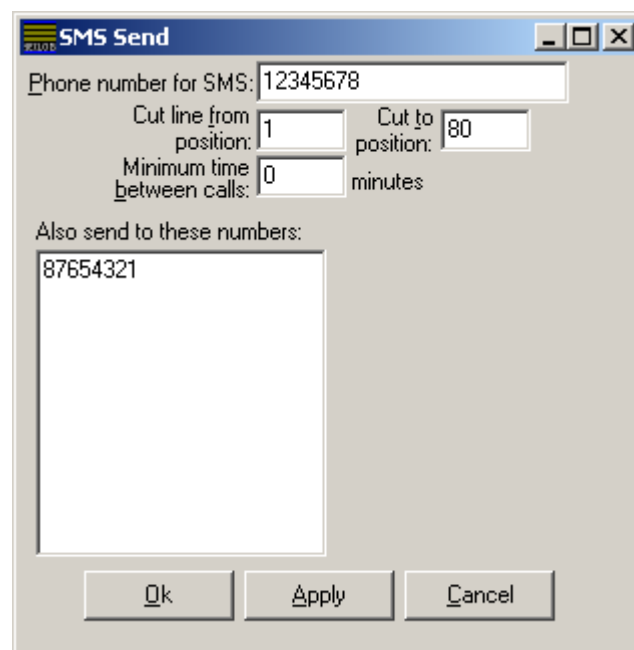
```
[General]
PhoneBookEntry=MyInternetConnection
RASTIMEOUT=15
```

If this is defined, Prilmail will try to call the dial-up connection in Windows called 'MyInternetConnection' in this example (substitute with your local setting). When no mail has been sent for 'RASTIMEOUT' seconds, Prilmail will disconnect.

If PRILMAIL.INI does not exist, Prilmail relies on a direct connection to the email server – for example through the local network or a broadband connection.

If for any reason an email couldn't be sent, Prilmail will reschedule it to be sent later. The waiting interval increases as it continues to fail. If the OUT-MAIL directory grows, it would mean that email can't be sent and the problem should be investigated. Prilog won't let the directory grow so big that it fills the hard disk completely. As the emails are stored in temporary files in the OUT-MAIL directory, the computer may be restarted without losing them. When an email is sent, its corresponding file is deleted.

7.7.2. Send SMS

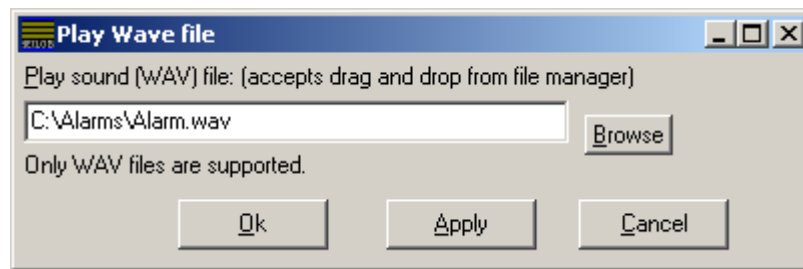


This action sends one or more SMS'es with the contents of the line. Define the phone number in the first edit box. Additional recipients can be defined below. Enter new numbers on separate lines. Cut from- and to positions can be used to extract only a portion of the line to be sent.

To limit the amount of messages sent, one can limit the sending by setting a minimum time between SMS-messages sent. Prilog keeps count of this timeout individually on each color number. A value of 0 means that every time an incoming line matches the search criteria (and SMS is globally enabled), Prilog will send the SMS-message.

For more details about SMS, see own chapter.

7.7.3. Play sound

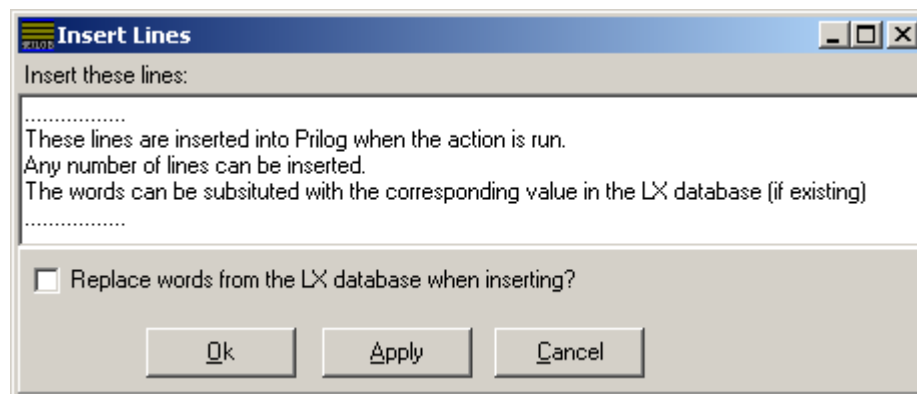


If a sound file is defined, Prilog will do a playback on this file at once a line matching the search-criteria is met. This option requires pre-recording sound-files - which could for example be a message in the local language with specific information about this very alarm.

The only sound files supported are files with the WAV extension. This is a standard Windows file format (meaning waveform). Prilog itself plays the sound file in the background. If another sound is played before the first one finishes, the first one is stopped. Prilog caches the last sound file for quick playing.

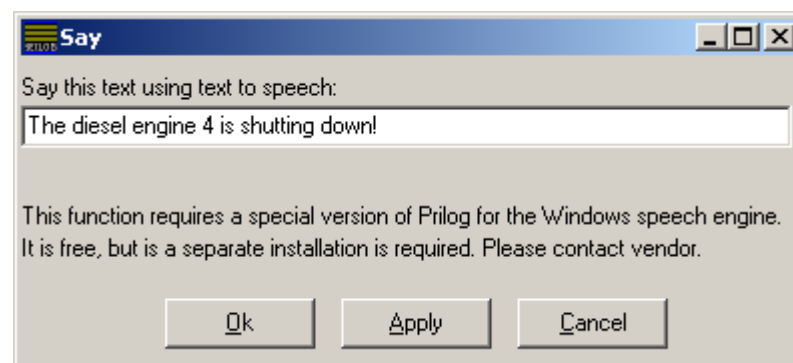
Note: This pane accepts drag-and-drop files from the Windows explorer for instance.

7.7.4. Insert line



Enter text line(s) in the edit field to be inserted into the text stream just after the current line. If the checkbox is checked, these lines are subject to replacement of Line Extractor value in the LX database. Any occurrence of a word in the line that also exists in the LX database will be replaced with the corresponding value in the LX database. The lines are inserted into the incoming stream of lines, but will be run through the "internal filters" and not the normal prefilters/portfilteres/postfilters.

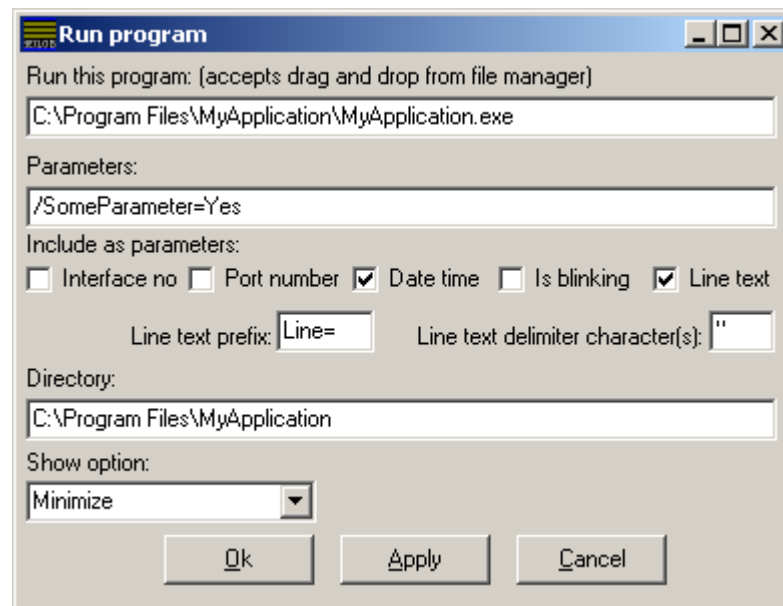
7.7.5. Say



Enter a text line to be output to loudspeakers using a built in text to speech engine.

Note that this only is available on Windows 2000 and newer. Not all computers have the Microsoft programs required for this function. If this is the case, these binaries (programs), can freely be downloaded from the Microsoft website.

7.7.6. Run



This configures an application to be launched when the action occurs. It is the same as launching a program from the desktop by double-clicking it. This application should be specially crafted so that it does a specific action and terminates afterwards so that it will not consume resources if run many times.

A BAT file could be invoked by using the CMD.EXE command interpreter.

Note: This pane accepts drag-and-drop files (from the Windows explorer for instance).

Parameters

To use some fixed parameters, enter them in the free-text field below 'Parameters'. If you want to add some of the information from the incoming line itself, you have the possibility to insert any combination of these parameters:

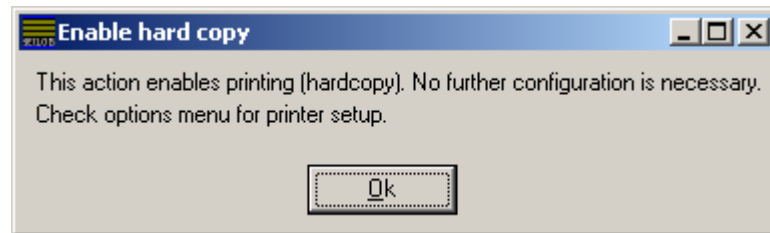
Parameter	Description	Parameter prefix
Interface no	Indicates which interface in File / Communications that originated the line. It is an ordinal number and can be used by the application to see where the line came from.	IFNo=
Port number	The port number the physical interface above is set up to use. For serial ports, this is the COM port number. For TCP connections, it is the TCP number.	PortNo=
Date time	The timestamp the line has – meaning the time it was received on the physical interface. The datetime is formatted according to the Windows control panel setting. However, all spaces are substituted with a underscore character ('_').	DateTime=
Is blinking	Indicates if the line is blinking (1) or not (0) at the time it passes the filter. Note that there can be another filter afterwards that enables blinking before it is written to disk.	Blink=
Line text	The actual line contents. Because there probably are spaces in the line text, the whole text line can be "encapsulated" in delimiter characters so that the application easier can see what is line text in the parameter list. The delimiter character(s) are defined by the edit box just below the line text checkbox. This parameter can also be preceded by a special prefix also defined below. Default is 'Line='.	User configurable. The contents of the 'Line text prefix' is used. Default: Line=

The parameter order is the same order as the above table (top – to – bottom) with a space between each. If a fixed parameter is defined (in the upper edit), it is appended at the *end*.

Example of actual parameters given to a program when all parameter options are elected:

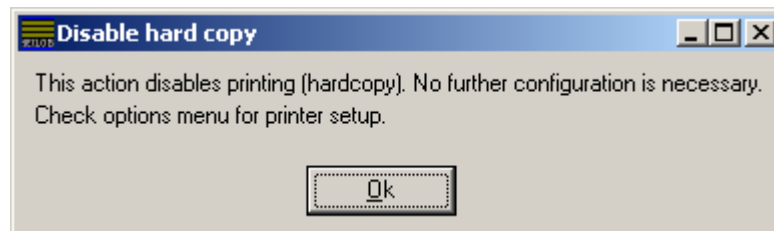
```
IFNo=0 PortNo=0 DateTime=08.01.2006_20:24:24 Blink=0 Line="XY_432_34 DG1 Alarm" /SomeParameter=Yes
```

7.7.7. Enable printer



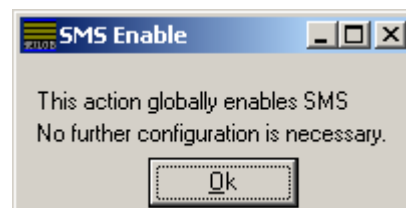
Adding this action will enable printing. This is observed in the options dialog and at the status bar in the main Prilog window. From the moment a line passes this filter, printing is enabled. The line will then be printed when it is written to hard disk, unless its color group is disabled in the options dialog.

7.7.8. Disable printer



Adding this action will disable printing. This is observed in the options dialog and at the status bar in the main Prilog window. The line will not be printed when it is written to hard disk.

7.7.9. Enable SMS



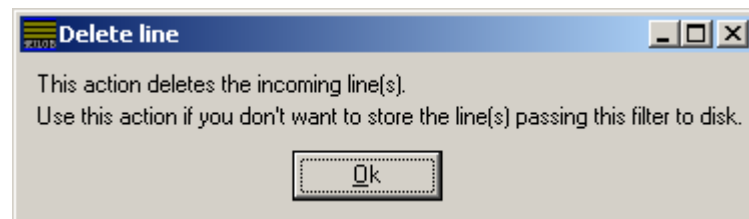
Adding this action filter will enable SMS. This means that any line from this point in time that reaches a SMS Send filter (in any action filter) will actually be sent as an SMS. The status of the SMS Enable flag is observed in the options dialog and at the status bar in the main Prilog window.

7.7.10. Disable SMS



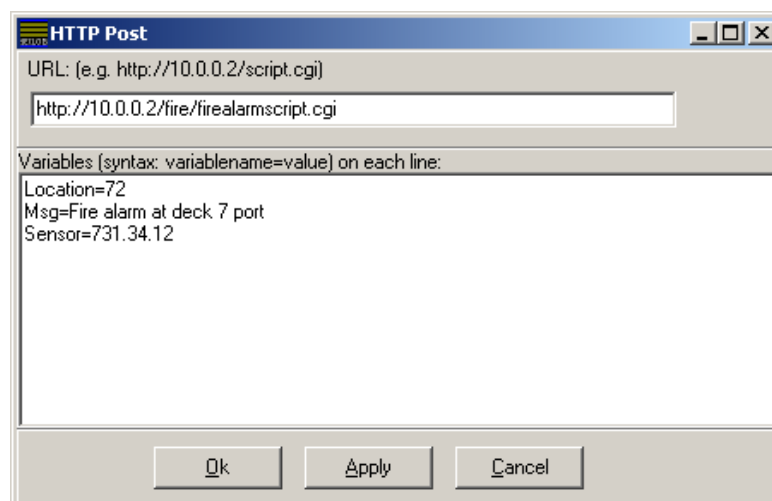
Adding this action filter will disable SMS. This means that any line from this point in time that reaches a SMS Send filter (in any action filter) will *not* be sent as a SMS. The status of the SMS Enable flag is observed in the options dialog and at the status bar in the main Prilog window.

7.7.11. Delete Line



Any line passing this action filter is deleted immediately and not written to hard disk, printed, sent anywhere or processed more by any other filter. This could be useful for sentences that only should generate an action, but never displayed on the screen or stored to hard disk. Therefore: Add this filter at the bottom of the “selected actions” so that all the filters above are executed and this filter finally deletes it.

7.7.12. HTTP Post



This action is used to send information to for example a web server or a pager system. Lines passing this filter causes a HTTP Post action to be generated and sent by TCP/IP to a host. Another word for HTTP Post is Web Form. HTML pages that are programmed as a form (dialog) accepts a HTTP posts with information from the user. The information consists of an address (URL) and a set of Name=Value variable pairs. Information from the LX database (the incoming lines itself) can be included in the post.

In it's simplest form, the HTTP Post action can be set up as above with all information statically configured. The variables are sent directly to the host without modification. But a LX filter can be inserted just *before* the HTTP action to extract certain values from the incoming line before used in the HTTP Post action. This is how:

LX Lookup Possibility 1 of 2:

The variable names (text before the equal sign) is looked up in the LX database. If found, all text *after* the equal sign is replaced with the contents of that LX variable. Example:

LX Database:

Location = Staircase 3

HTTP Post variable box:

Location=

When the HTTP Post action is run, the text “Location=Staircase 3” is actually sent to the host.

As the LX database is constantly changing, the HTTP Post is also constantly sending the latest value for the ‘Location’ variable.

Note: If the HTTP Post variable box says “Location=some text here”, and “Location” is a LX variable, the text “some text here” is totally overwritten with the value of the LX variable.

LX Lookup Possibility 2 of 2:

If the variable name does *not* correspond to a LX variable name, each word of the text *after* the equal sign is looked up in the LX database and replaced with the contents if existing. Example:

LX Database:

DG1_RPM = 690

DG1_Alarm = Yes

HTTP Post variable box:

RPM=DG1_RPM AlarmStatus: DG1_Alarm

When the HTTP Post action is run, the text "RPM=690 AlarmStatus: Yes" is actually sent to the host.

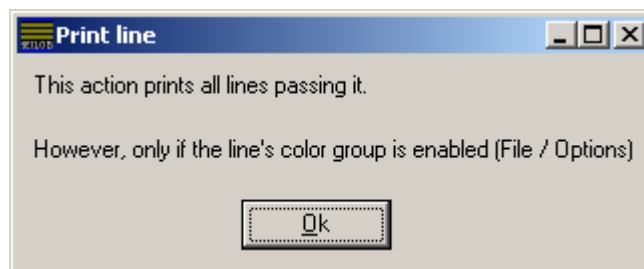
Note that for this to work, the LX variable name must be a complete word, surrounded by blanks or line begin/end. LX is not case sensitive. Text comparison is used using the current Windows locale.

HTTP Poster engine

Prilog has one instance of the HTTP Poster engine installed. This 'engine' collects all HTTP request from all filters and sends them out in an orderly fashion. There are certain important aspects of this:

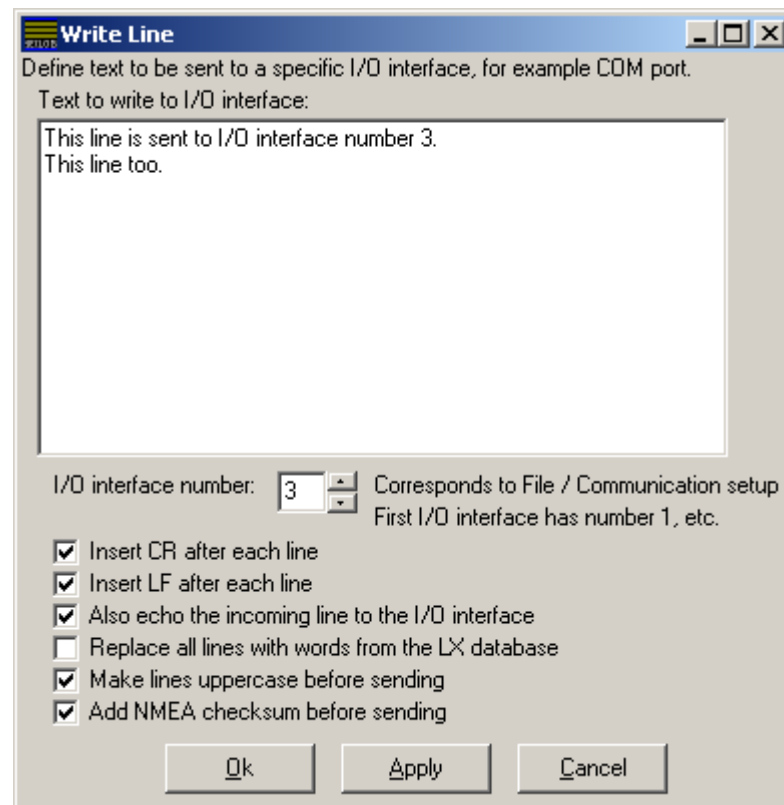
- The posts are queued and sent one at a time
- Max queue size is 100 posts, but will normally never be reached
- Minimum delay between each post is 100 milliseconds, allowing a theoretical max output of 10 HTTP Posts per second.
- If the receiving host does not respond or does not close the TCP connection, Prilog has a timeout of 3 seconds, thus blocking the others in the queue for 3 seconds.
- Each post is tried sent maximum 20 times, afterwards discarded from the queue.
- The newest unsent posts are given priority. Then the "oldest" that failed earlier.
- When Prilog terminates, the queue contents are lost as they are only stored in RAM.

7.7.13. Print Line



This action prints the line passing it if the line's color is enabled in the options menu. Note that the line is printed even if printing is not enabled. The line is also not printed if there is a large amount of print-jobs pending in the corresponding printer queue. This is to avoid crashing Windows with too many print jobs.

7.7.14. Write Line



The Write Line action filter is used to send data out to a specific I/O interface. You can define any number of text lines in the upper part of the dialog. Every time a line is passing through the filter, these lines are then written to the I/O interface.

The I/O interfaces are numbered from 1 and upwards and corresponds to the number in the File / Communication setup or the View / Serial port status dialog. You can write to any defined I/O interface, but not all I/O types supports writing. You can even write to interfaces that you are receiving data from. Typical interface types that supports writing include COM port, TCP/IP client and TCP/IP server.

These additional checkboxes can be used:

Insert CR after each line: Inserts ASCII character 13 to terminate the line sent

Insert LF after each line: Inserts ASCII character 10 to terminate the line sent

Also echo the incoming line to the I/O interface: The incoming line itself is sent to the I/O interface

Replace all lines with words from the LX database: Check this to replace any word in the line(s) sent to the I/O interface with the LX database.

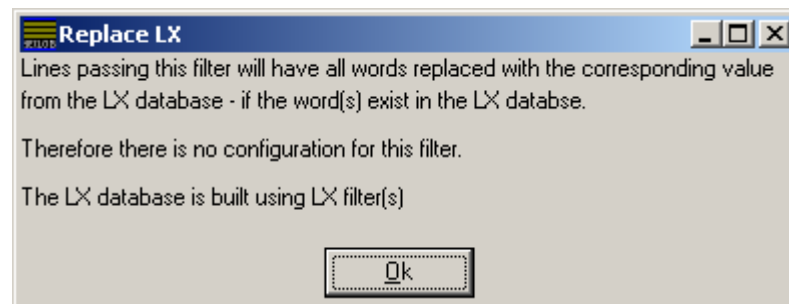
Make lines uppercase before sending: If checked, the sent line is made all uppercase before sending

Add NMEA checksum before sending: If checked, Prilog calculates the NMEA checksum based on all characters in the line to be sent, minus the first one, which is assumed to be '\$' as the NMEA standard says. Then, the checksum is appended along with its asterix, for example '*34'

The lines passing this filter are not modified in any way. They pass the filter and are eventually stored to the hard disk.

Note: Prilog maintains a sending-queue for each I/O interface that can hold a maximum of 1000 lines. If the I/O interface is temporary down, the lines are accumulated until it is opened again. If it isn't opened, lines are lost if the queue is full.

7.7.15. Replace LX



By adding this filter, all lines passing it will be subject to word-replace from the LX database. The lines are therefore modified. The filter has no configuration.

7.8. Dialog settings activation

When you alter anything in the action setup dialog, the new setting is instantly activated. This means that as you type, the search criteria's changes according to what you type. If lines are coming in at the same time, you may instantly see the result.

The exception is when you are using files to store the search-criteria's (by means of And/Or/Nand/Nor files). Prilog reads these files then pressing OK at the main action setup dialog. But as Prilog doesn't know when these files are updated, you should press the Refresh-button to re-read them if you update them when Prilog is running.

However, if you press the Cancel-button on the main action setup dialog, all the old setup is restored and the changes are lost.

If you press the OK-button on the main action setup dialog, all the information is stored.

8. Line Extractor filter (LX)

The Line Extractor filter (LX) is a general purpose rule based mechanism to extract values from text lines. It does not modify the lines passing it, just extracts values and stores them in a common internal database stored in Prilog memory (RAM). The purpose is to insert these values again in later lines using either of these filters: Word Replace, Insert Lines, HTTP Post or Write Line.

When Prilog starts, the database is always empty, but built dynamically during runtime based on the configured rules. The database is therefore lost when Prilog terminates. In runtime, it can be viewed in the Prilog Writer under the menu item View / Line extraction database. Here, the database can also be cleared by pressing the 'clear database' button.

The database consists of 'VariableName = Value' pairs. The variable names are always unique. If a 'VariableName = Value' pair is inserted in the database, but 'VariableName' already exists, the new value overwrites the old. Variable names are *not* case sensitive.

The variable names are either extracted from the incoming lines themselves (e.g. tagname), or you predefine it when you configure the LX filter.

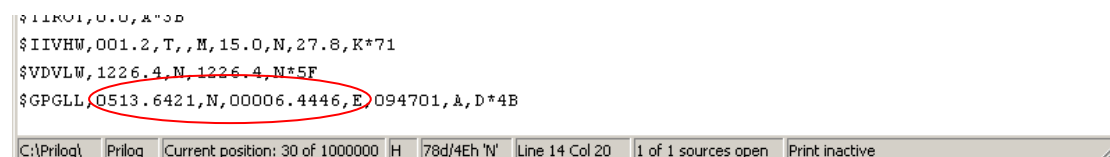
The value part is extracted by rules giving a character position in the incoming lines. The first character from the left has position 1. If a field rule has *one* position sub-rule, the word at that position is extracted (or character if SharpCut=TRUE). If a field rule has *two* position sub-rules, the word(s) between these positions are extracted (or, if SharpCut=TRUE, the exact characters between these positions). An exception is that some position rules actually return two positions, like the string block rule. In this case, SharpCut=TRUE must be used in the field.

The LX filter also have TRUE / FALSE rules that allow you to establish complex logical expressions on whether to accept or reject value extraction of each passing line.

There is by default one LX filter for each filter group. In the action filter, you can add as many LX filters as you want. But since the LX filter can contain as many sentences as you want, normally one instance of the LX filter is enough.

The LX database in Prilog is common for all LX filters. Each LX filter updates this one.

Consider the following lines read from NMEA instruments:



```
$IHK01,U,U,A"0D
$IIVHW,001.2,T,,M,15.0,N,27.8,K*71
$VDVLW,1226.4,N,1226.4,N*5F
$GPGLL,0513.6421,N,00006.4446,E,094701,A,D*4B
```

LX can be configured to read the GPS position this way:

Position = 0513.6421,N,00006.4446,E

This is done with a field called 'Position' with a String Block rule (Separator=',', BlockNo=2, BlockCount=4)

Then, we define an action filter that has 'Alarm' as search-word. Below the action-filter, we define an action: 'Insert Lines'. We enter 'The ship is at Position' in the Insert Lines filter. The result is that each time a line enters Prilog that includes the text 'Alarm', a new line is immediately inserted that is like this : 'The ship is at 0513.6421,N,00006.4446,E' (only that the latest GPS position is used The keyword 'Position' in the Insert Lines filter is substituted with the corresponding LX variable.

8.1. Using values from the LX database

There are 4 ways of using the LX database:

8.1.1. LX with the Insert Line filter

The Insert Line filter can insert one or more lines in the line stream when the action criteria are met. Optionally, before the line is actually inserted it is run through a 'Search and Replace' function for LX database names. For example, if a line to be inserted is:

'The ship is now at Position'
the line actually inserted would be:
'The ship is now at 5915.628,N,01025.200,E'
because 'Position' is in the LX database and it's value is the ship's position.

8.1.2. LX with the Replace words filter

1. Select a from-word, for example 'SHIP' and type this in the 'From word' column.
2. In the to-word column for example enter \LX Position;
Remember to include the backslash and the trailing semicolon. 'Position' is chosen as an example of LX variable name that is going to be fetched from the database. Using this example, each time the word 'SHIP' is read from the incoming lines, this word is replaced with the value of the 'Position' variable in the LX database.

8.1.3. LX with the HTTP Post filter

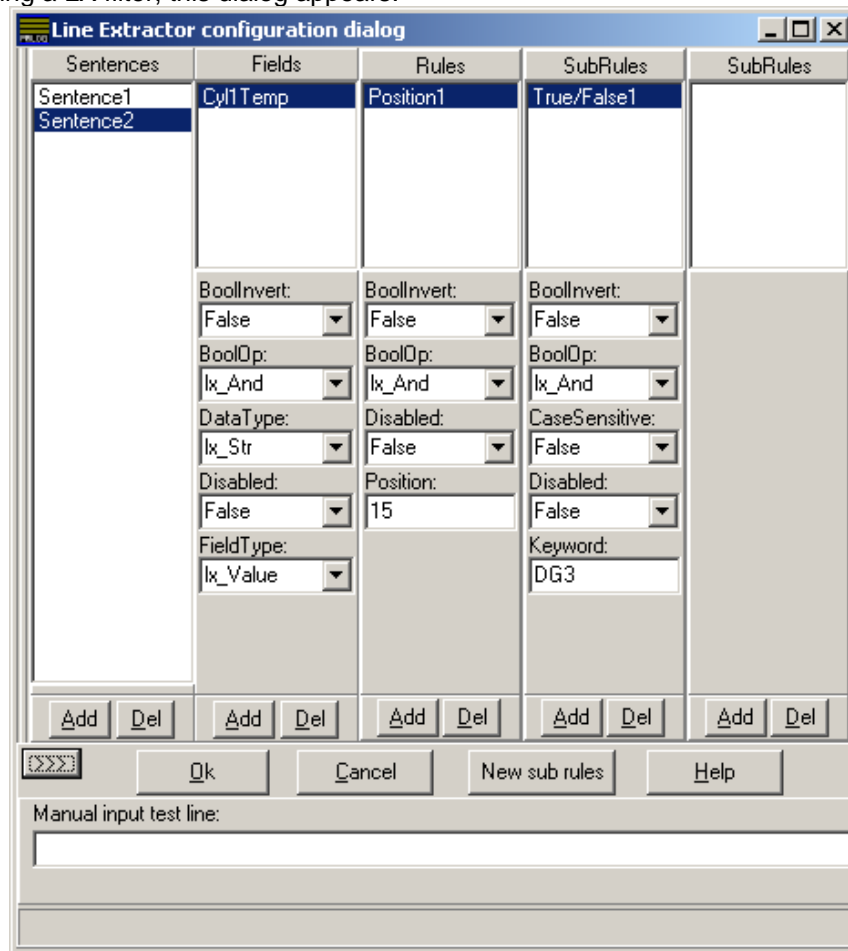
This action filter sends a TCP/IP packet across the network to another host. The variables configured within the HTTP Post filter are substituted with LX variables before sent. See more details in this manual where the HTTP Post filter is described.

8.1.4. LX with the Write Line filter

The write line filter can send lines to a physical I/O interface. If checking a special checkbox in this filter dialog, all lines sent are subject to word replace with the LX database. This could be used to construct custom-made NMEA sentences.

8.2. Configuring the LX filter

When configuring a LX filter, this dialog appears:



The dialog is titled "Line Extractor configuration dialog". It features a table with five columns: Sentences, Fields, Rules, SubRules, and SubRules. The first row contains "Sentence1" under Sentences, "Cyl1Temp" under Fields, "Position1" under Rules, and "True/False1" under SubRules. The second row contains "Sentence2" under Sentences, and the other columns are empty. Below the table, there are three columns of settings. The first column (under Sentences) has: BoolInvert: False, BoolOp: lx_And, DataType: lx_Str, Disabled: False, and FieldType: lx_Value. The second column (under Rules) has: BoolInvert: False, BoolOp: lx_And, Disabled: False, Position: 15, and Keyword: DG3. The third column (under SubRules) has: BoolInvert: False, BoolOp: lx_And, CaseSensitive: False, Disabled: False, and Keyword: DG3. At the bottom, there are buttons for Add and Del for each column, and a "Manual input test line:" text area.

Sentences	Fields	Rules	SubRules	SubRules
Sentence1	Cyl1Temp	Position1	True/False1	
Sentence2				

BoolInvert: False
BoolOp: lx_And
DataType: lx_Str
Disabled: False
FieldType: lx_Value

BoolInvert: False
BoolOp: lx_And
Disabled: False
Position: 15
Keyword: DG3

BoolInvert: False
BoolOp: lx_And
CaseSensitive: False
Disabled: False
Keyword: DG3

Add Del Add Del Add Del Add Del Add Del

Manual input test line:

The LX filter is organized in sentences. Each sentence owns one or more fields. A field is a text string (data value). A field owns one or more rules of how to extract the field data (at least one position rule). In turn, a rule may also have sub-rules to allow for more complex boolean evaluations of whether to accept or reject certain lines.

When a line is run through the LX filter, it is run sentence by sentence. The resulting extracted values are shown in the left pane of the dialog below 'Resulting extracted fields' (using the '<<<' and '>>>' buttons in the dialog). This left pane is cleared before new line received, so it will always reflect the extracted values of the last received line. When configuring, it is useful also to have the LX database visible to see the actual database (choose Status / Line extraction).

Remember that there is a LX filter in the PreFilter group, in the current Port filter group and in the PostFilter group (actually also in the Internal filter group). Usually, when the setup isn't complex, the PostFilter group's LX filter would be a good place to use because normally there isn't a need to distinguish the LX filters by different physical ports.

In other words, to set up LX rules, use the PostFilter group if there is no special need.

Sentences can contain any number of fields and any number of sentences are allowed. It is not important which field goes into which sentence or if you choose to have only one sentence to include all fields. The important thing is that each field results in one value in the LX database. There is one exception: LX Model 2. Here, each sentence should have only two fields, one tagname field and one value field (see further on for deeper explanation).

8.3. Configuring a field

A field has these properties:

Name:	Double click on the field, and a dialog pops up asking for a field name. Note that Model2 does not care about the field name, as the field name is taken from the line itself (the 'tagname' field')
BoolInvert:	Set to TRUE if the result of the TRUE/FALSE evaluation is to be inverted. Any field may have one or more TRUE/FALSE rules attached. Together they form a boolean expression. By using this option, the result of this expression may be inverted, thus making possible NAND/NOR expressions etc. possible.
BoolOp:	Which boolean operator there should be between each boolean rule. Any field may have one or more TRUE/FALSE rules attached. This parameter indicates whether to use AND or OR function between them. Default is AND, so that each TRUE/FALSE rule must evaluate to TRUE if the line is to be accepted.
DataType:	Is not used by Prilog. Prilog treats every field as a string value.
Disabled:	Disables this field if set to TRUE
FieldType:	lx_Value or lx_Tag. Prilog uses no other options. lx_Value indicates that this field is the data value. lx_Tag indicates that this field is a tag name (or 'variable name').
CutSharp:	If to extract whole word surrounding the position(s), or cut word exactly at position(s) If a field only has 1 position rule: CutSharp=TRUE means extract the character at that exact position CutSharp=FALSE means extract the word appearing at that position, regardless if it began before the given position, or ends after. If the position points to blank space, nothing is extracted. If a field has 2 position rules (or a position rule that returns two positions) CutSharp=TRUE means cut string from start position to end position regardless if it means cutting a word. CutSharp=FALSE means include whole word at start position even if it begins before start-position, and also include whole word at stop position, even if it ends after the stop-position.

8.4. Configuring a Rule

A rule is either a position rule or a TRUE/FALSE rule.

A position rule returns a position. A TRUE/FALSE rule returns either TRUE or FALSE.

Each field must have at least one position rule. This is because we must know at which position the field is located in the incoming lines. The first character has position 1. If a field has one position rule, the extracted value is word at that position. This means that Prilog searches to the left and right of the position to the first blank character (space). If a field has two position-rules they actually then transform their meaning to TO-and-FROM position. If the first position rule returns 10 and the second 20, the resulting value is then all characters from 10 to 20, totally 11 characters.

TRUE/FALSE rules are optional. They are used to either accept or reject a line for a given field. For example we would want to configure the LX filter only to extract the GPS position from GPS1, but not GPS2. You can add as many TRUE / FALSE rules as you like.

Each rule has different properties, but the 'BoolInvert', 'BoolOp' and 'Disabled' properties are common and have the same meaning as within the field configuration. Rules need not have a special name, and are given a unique name when adding them. Each rule can also have sub-rules. Sub-rules are only useful for specifying complex (nested) boolean expressions.

8.4.1. Position rules

Declare the exact position of the field:

This simple rule specifies the exact position of the field in the 'Position' property.

Example:

TT_440_005 FLV1 RM AHU5 WTR IN Sig.Err Normal 01-07 09:29:11.713

Rule Parameters:

Position = 13

Explanation:

This position rule always returns position 13.

Resulting line extraction is:

'FLV1' if SharpCut = FALSE

'L' if SharpCut = TRUE

Keyword gives field position (+options)

This rule is more complex. You may specify a keyword that Prilog searches for either from the left or the right (by specifying the 'FromRight' property). When found, the position is moved 'Offset' characters to the left or right. If the line contains more than one occurrence of the keyword, the 'WordNo' property specifies which one, beginning with 1 as the first occurrence.

Example:

TT_440_005 FLV1 RM AHU5 WTR IN Sig.Err Normal 01-07 09:29:11.713

Rule Parameters:

Keyword = 'Normal'

CaseSensitive = FALSE

WordNo = 1

From Right = FALSE

Offset = -8

Free text keyword you want to search for

Indicates if the keyword search is case sensitive

Occurrence number of keyword in line

WordNo counting from right or left?

Character offset from position of keyword in line

Explanation:

Searches for the first occurrence from the left of the non case-sensitive text string 'Normal'. Uses the position of the word (its first character, in this case the 'N'), counts 8 characters to the left (which gives the position of the 'S').

Resulting line extraction is:

'Sig.Err' if SharpCut = FALSE

'S' if SharpCut = TRUE

String block number

Use this rule in sentences that separate the values with a fixed character, for example a comma. A block is a value separated by the separator character. The first block is number 1.

There are two key parameters:

Separator: Character that separates each block

BlockNo: Indicate which block number to extract (default: 1)

BlockCount: How many blocks to extract from 'BlockNo' (default 1) to the right

This is especially useful for NMEA sentences that are separated by commas. Indicate in the property 'BlockNo', starting with 1, which comma separated block is to be returned. Note that this function returns two positions, start position and stop position of the block so that Prilog cut the text out exactly at the block delimiters. Therefore the owning field must have SharpCut = TRUE in order to work.

Example:

\$GPGLL,0513.6421,N,00006.4446,E,094701,A,D*4B

Rule Parameters:

BlockNo = 2

BlockCount = 4

Separator = ,

Explanation:

Searches for the second block that is separated by commas. Returns this block and the next 3, a total of 4 text blocks.

Resulting line extraction is:

'0513.6421,N,00006.4446,E' if SharpCut = TRUE

Returns whole sentence if SharpCut = FALSE because there is no blank space in the sentence and the whole sentence is therefore considered a word.

Word number + offset

Returns the position of the n.th word from either left or right, then adds an option offset. Words are separated by space(s) or tab(s) (ASCII 32 or 9).

Example:

TT_440_005 FLV1 RM AHU5 WTR IN Sig.Err Normal 01-07 09:29:11.713

Rule Parameters:

WordNo = 4

From Right = FALSE

Offset = 5

Word number

WordNo counting from right or left?

Character offset from word position in line (+/-)

Explanation:

Searches for the fourth word from the left (finds 'AHU5'), then goes 5 characters to the right (Offset = +5) from the first letter in the word. In this case: stops at the 'W'.

Resulting line extraction is:

'WTR' if SharpCut = FALSE

'W' if SharpCut = TRUE

8.4.2. TRUE / FALSE rules

The only TRUE/FALSE rule is 'A certain word must exist'. This rule searches the incoming line for this word and returns TRUE if it does. If not, the line is rejected.

8.5. LX Model1 and Model2

There are two models of line extraction: One *without* tagname in the text, the other *with* tagname.

Model1: Sentences include only data values without any tag names (of interest at least).
Values are just spread around in the text and you want to extract them.
You therefore want to give each value its own unique LX variable name.
Each field in the LX dialog will then be a data value.
The field name you then assign it in the LX dialog is the variable name in the LX database (you double click the field directly to change it).

Model2: Sentences include a Tag Name in addition to its value.
You want to associate the tagname appearing in the sentence with the value.
Therefore you must specify both where the tagname and the value is in each sentence.
In one sentence: One field must point to the tagname, and another to the data value.
The LX variable name is the tagname and is assigned its associated value.
The field names in the LX dialog are therefore not used and not important.
Although the field names could be changed to easier understand what it is.

See below for an example of these two models.

Model1 dialog example

Line Extractor configuration dialog

Sentences	Fields	Rules	SubRules	SubRule
Sentence1	Cyl1Temp	Position1	True/False1	

BoolInvert: False
 BoolOp: lx_And
 DataType: lx_Str
 Disabled: False
 Position: 10
 FieldType: lx_Value

Add Del Add Del Add Del

Manual input test line:

This model1 example shows one sentence and its corresponding field. The field name is Cyl1Temp and its position is 10. It contains one sub-rule which requires this sentence to contain the word 'DG3'.

The resulting LX database will then have a variable name called Cyl1Temp with the value extracted from position 10 of lines that contains the keyword 'DG3'.

Note that the Cyl1Temp field is of field type 'lx_Value'.

Model2 dialog example

Line Extractor configuration dialog

Sentences	Fields	Rules	SubRules	SubRule
Sentence1	Tagname Cyl1Temp	Position1	True/False1	

BoolInvert: False
 BoolOp: lx_And
 DataType: lx_Str
 Disabled: False
 Position: 1
 FieldType: lx_Tag

Add Del Add Del Add Del

Manual input test line:

This model2 example shows one sentence with a tagname/value field combination. The tagname field is on position 1 (the fieldname 'tagname' is unimportant). Note that the tagname field has FieldType=lx_Tag, thus letting Prilog know that here is the tagname.

The 'Cyl1Temp' field (field name also unimportant) is at position 30 and requires the text 'DG3' to also appear in the text. Note that FieldType=lx_Value, thus letting Prilog know that here is the data value. Prilog then inserts the tagname / value pair into the LX database.

8.5.1. Model1 example

We have for example incoming lines in this format:

Azimuth event 150 degrees RPM 110 Event 01/12/2003 18:58:03

Example Model1 configuration:

Field1: 'AZ1Pos' at position 15

Field2: 'AZ1RPM' at position 31

8.5.2. Model2 example

We have for example incoming lines in this format:

Az1Deg 150 Az1RPM 110 Event 01/12/2003 18:58:03

Example Model2 configuration:

Sentence1

Field1: Tag Name at position 1

Field2: Value at position 8

Sentence 2

Field1: Tag Name at position 12

Field2: Value at position 19

The difference with these to sentence types is that Model1 has no Tag Names or variable names in the actual text that is useful. We therefore use the field name in the LX dialog as a value name (variable name). Model2, however, has excellent variable names: 'Az1Deg' and 'Az1RPM' that we can use as LX variable names.

8.5.3. Model1 configuration

Shortly, all fields have FieldType = lx_Value.

You start by adding or selecting a sentence. Then adding fields (one ore more) that all have FieldType = lx_Value (as they have by default). The FieldType is selected using the dynamically created controls that appears in the pane just below when selected fields in the dialog. Each field must have at least one position rule to indicate where in the line it is located.

When this is done, select a name for the field by double clicking the field. This name constitutes the LX name in the 'Name = Value' pair in the LX database.

8.5.4. Model2 configuration

Shortly, there must be two fields defined: One TagName field and one Value field.

This means: One field must have FieldType = lx_Tag and the other must have FieldType = lx_Value.

Here, as always, each field must have at least one position rule to indicate where in the line it is located. The first field specifies where the tagname is, the other field where the value is. These form a 'Name = Value' pair in the LX database.

8.6. LX Step by Step configuration walkthrough

Scenario: We receive events on COM1/9600 and GPS on COM2/4800. We want to log the ship's position every time 'OVERBOARD VALVE' is found in the event text. Like this:

```
...
...
XV23_3040ZSH OVERBOARD VALVE OPENED 12-03-2003 13:23:21
Ship's Pos 6023.2090,N,00518.7260,E
PS34_029340101 ANY OTHER EVENT TEXTTEXT 12-03-2003 13:23:21
...
...
```

The GPS lines we are interested in are the following:

\$GPGLL,6023.2090,N,00518.7260,E,083400.00,A,A*65

```

|
|
|
|
|
|
|
|
|----- E=East
|----- Position east
|----- N=North
|----- Position north
```

8.6.1. Defining the comports

Let's start with a Prilog you just installed.

Go to File / Communication and add 2 COM ports:

Port 1: COM1 using 9600. From event generator (automation equipment)

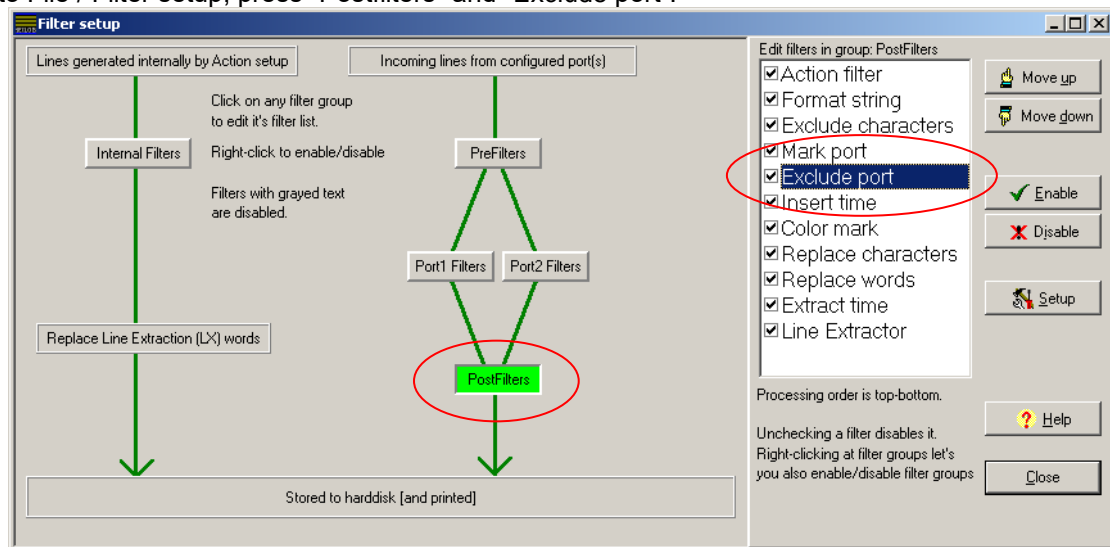
Port 2: COM2 using 4800. From GPS.

The ports probably use the default properties.

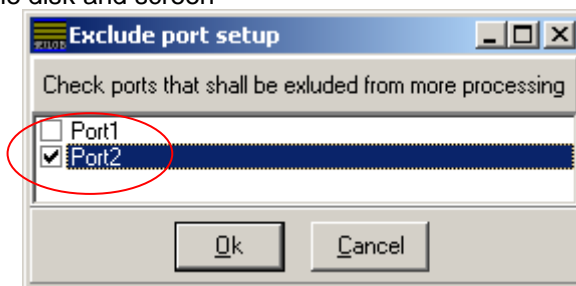
Pressing OK, you should now see both event text and GPS lines fed into Prilog.

8.6.2. Defining the filters

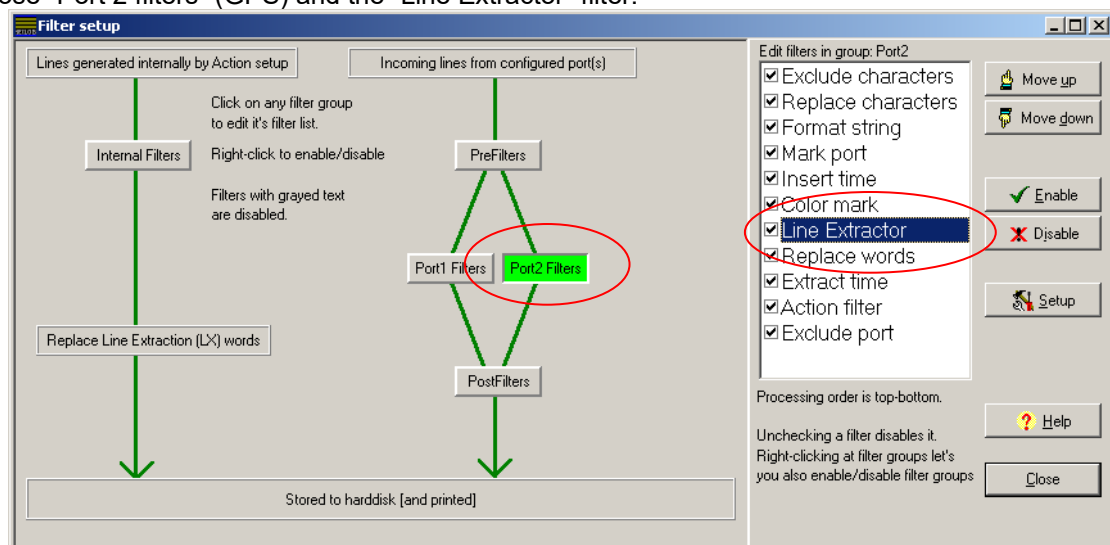
First we want to avoid the GPS lines from coming into the screen (and data file). Go to File / Filter setup, press “Postfilters” and “Exclude port”:



Double click the “Exclude port” filter and check “Port 2” inside the dialog to exclude from GPS lines passing further down to the disk and screen



Next, we want to extract some data from the GPS into the LX database. Go to the File / Filter setup, choose “Port 2 filters” (GPS) and the “Line Extractor” filter.



When double clicking the Line Extractor filter, you get a dialog. What we want to do is define a way to read one LX variable from the GPS data called “Position”. The rules are hierarchical, and this is a easy way of showing the setup:

Sentence

Field: "Position" (DataType = lx_Str, FieldType = lx_Value)
 Block number rule (BlockNo 2, BlockCount=4, Separator =',')
 TRUE / FALSE Rule: Word must exist: Keyword=GPGLL

Here is the same represented in the dialog (the first showing the BlockNo rule, the second showing the TRUE/FALSE rule):

The image shows two screenshots of the 'Line Extractor configuration dialog' window. Both windows have a 'Resulting extracted fields' pane on the left showing 'Sentence1.Position.Value = "6023.2090,N,00518.7260,E"' and a 'Manual input test line' at the bottom with the text '\$GPGLL,6023.2090,N,00518.7260,E,083400.00,A,A*65' and its decoded version '\$GPGLL,6023.2090,N,00518.7260,E,083400.00,A,A*65'.

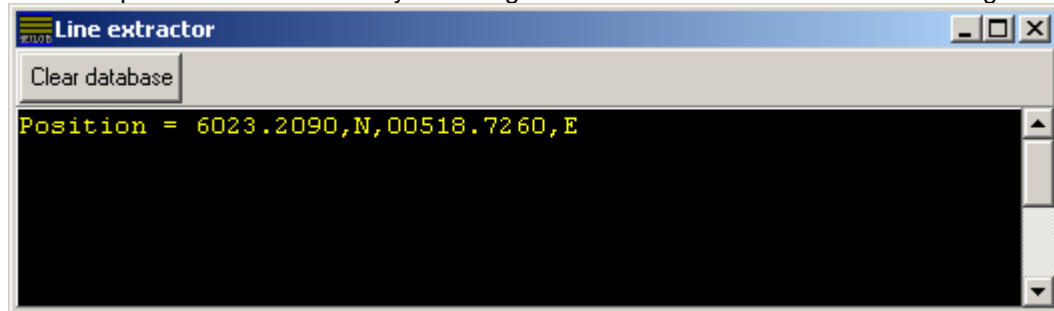
Top Screenshot: The 'Rules' tab is selected. It shows a rule named 'BlockNo1' with the following configuration:

BlockNo1
True/False1
BlockCount: 4
BlockNo: 2
BoolInvert: False
BoolOp: lx_And
DataType: lx_Str
Disabled: False
FieldType: lx_Value
SharpCut: True

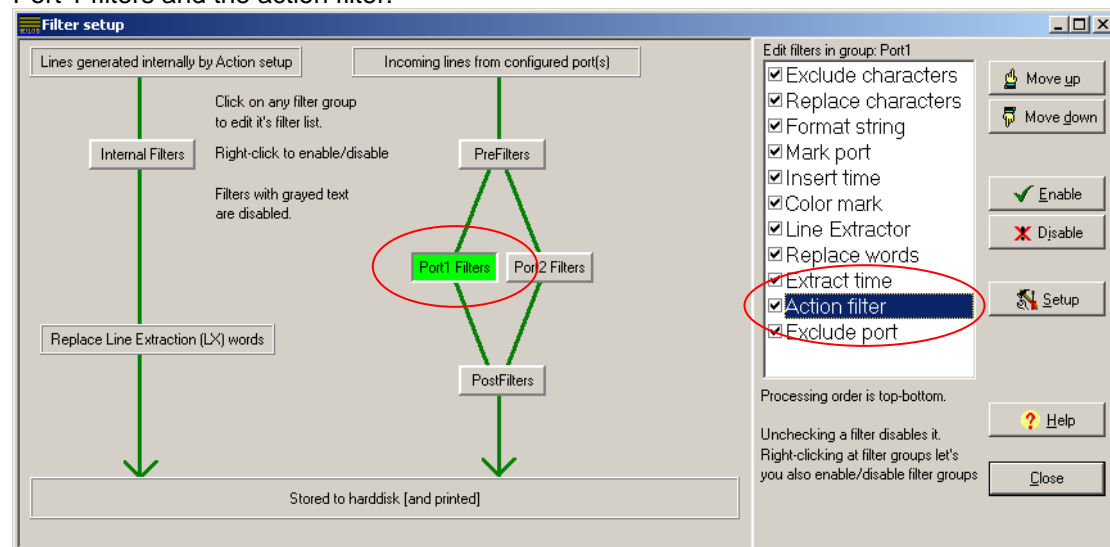
Bottom Screenshot: The 'Rules' tab is selected. It shows a rule named 'True/False1' with the following configuration:

True/False1
BlockCount: 4
BlockNo: 2
BoolInvert: False
BoolOp: lx_And
CaseSensitive: False
Disabled: False
Keyword: GPGLL
SharpCut: True

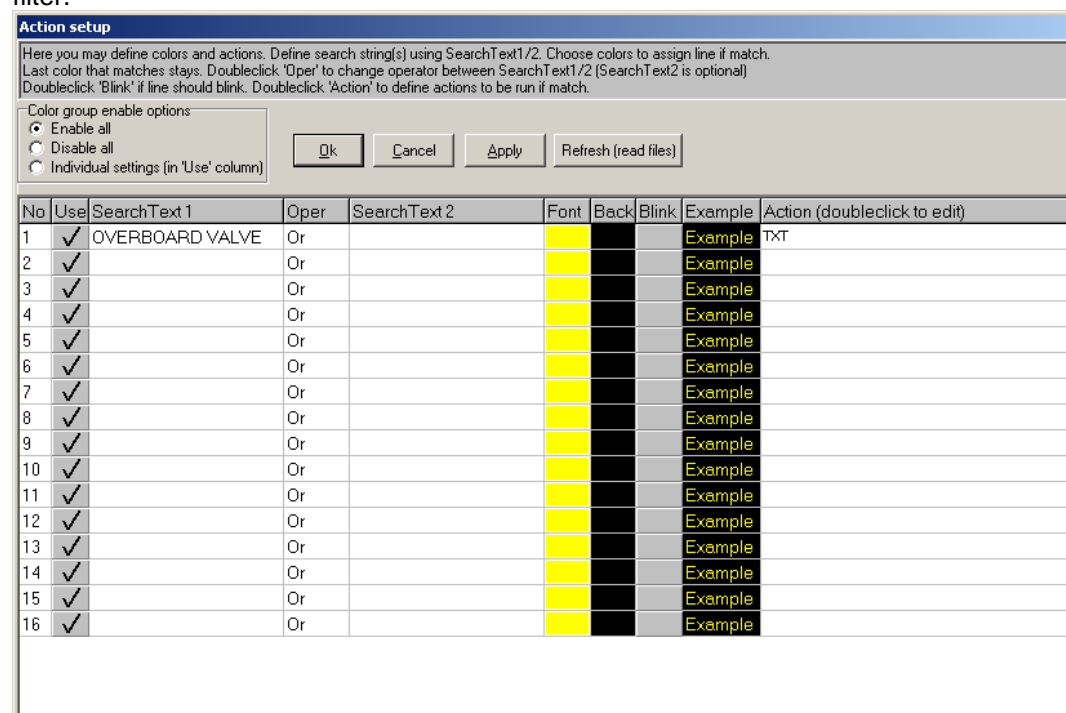
Now, we can inspect the LX database by choosing View / Line extraction database in Prilog Writer:

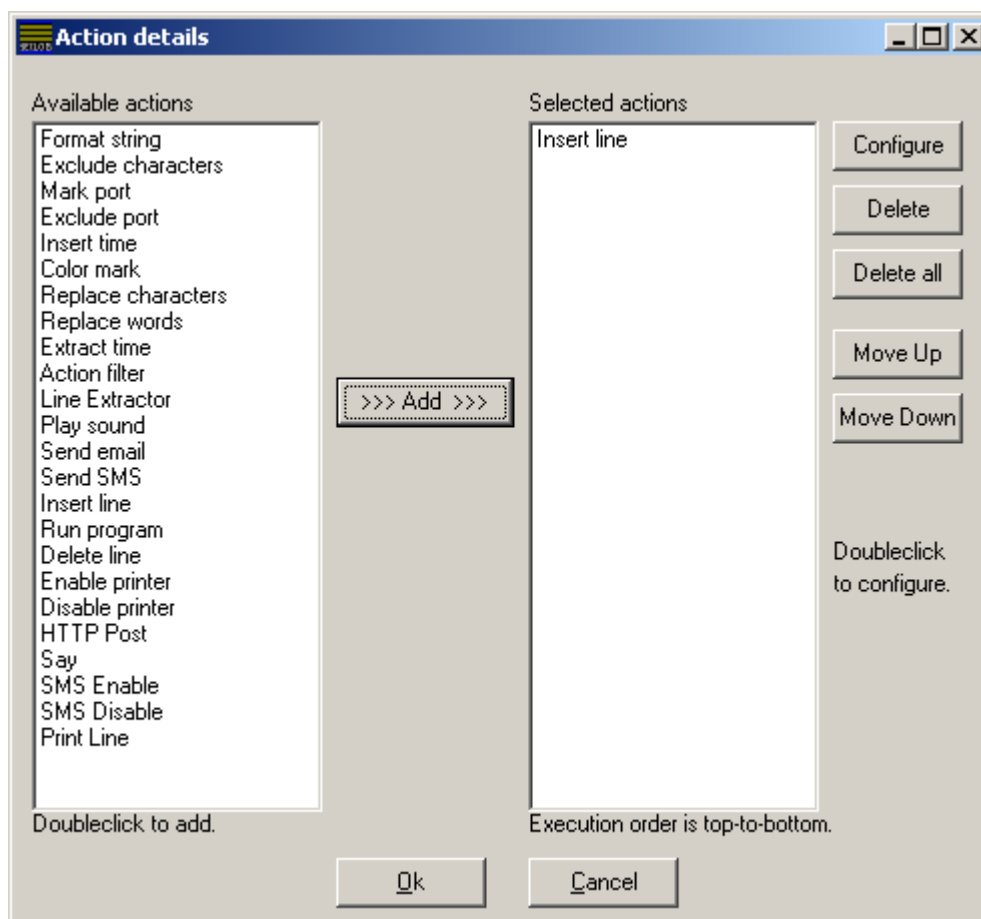


The last step is to insert the position into the event text. Let's go to File / Filter setup again and choose Port 1 filters and the action filter:

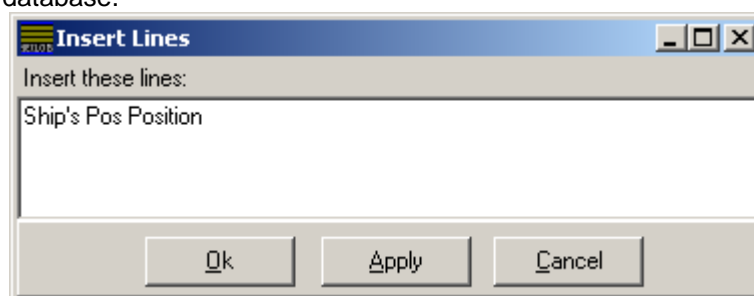


Double clicking the action filter will bring up a new dialog. Write 'OVERBOARD VALVE' in the Search text 1 for color group 1. Then double click the rightmost column in the same line and add a "Insert Line" filter.





Double click the 'Insert line' action filter in the right pane to edit the settings. Enter the text 'Ship's Pos Position' into the dialog. Doing this inserts that text every time the text 'OVERBOARD VALVE' occurs in the event text. It also substitutes every word it can find in the LX database. Because we only have one LX variable called 'Position', only the word 'Position' in the Insert Line filter is exchanged with the actual contents of the LX database.



The filter will then insert the text (for example)
Ship's Pos 6023.2090,N,00518.7260,E

This results in Prilog inserting the ship's position every time the overboard valve is mentioned. There are of course many more things you could do. For example the position could be marked with red color by entering the 'Internal filters' and choosing the 'Color mark' filter. This is because all lines inserted by the 'Insert Lines' filter are passed through the 'Internal filters'.

Hopefully, this example will help configuring the more advanced features of Prilog.

9. SMS

SMS (Short Message Service) is a very useful method of transferring alarms or events to a cellular phone. Prilog can be configured to send selected alarms or events based on a search criteria. Prilog will send the actual alarm/event text so that the recipient can instantly see what is happening. Any number of recipients can be configured. Delivery time is 'as fast as possible', probably in the neighbourhood of a few seconds from when the alarm is generated until it is received.

Prilog supports worldwide SMS sending. However, because SMS sending costs money, one must normally buy services from a SMS provider to accomplish it. SMS receiving is not supported unless Prilog can receive it using any of the standard ports - for example email or from a serial port, in which case Prilog would not treat these lines differently than any other text lines. From here on, we therefore only talk about SMS *sending* from Prilog to a SMS receiver (usually mobile phones) using a public telecom network.

There are several technical solutions for sending SMS, as presented by the following table.

Note: If you want to send alarms to a pager system, Prilog can also do this. See HTTP Post action.

SMS Send Method	Minimum Requirements	Reliability	Protocols
Dialup directly to SMS provider	<ul style="list-style-type: none">• Phone line (analog or ISDN)• Modem connected to PC• Agreement with SMS provider• Software from SMS provider	Good. Depends on the phone network which almost always works.	UCP Proprietary
Internet to SMS provider	<ul style="list-style-type: none">• Modem / ISDN / Broadband Internet connection• Agreement with SMS provider• Software from SMS provider	Good. Depends on the internet service provider and the internet.	TCP, HTTP, HTTPS, SOAP, XML
Email to SMS provider	<ul style="list-style-type: none">• Internet access (or at least an Email gateway)• Agreement with SMS provider	Good. Depends on the internet service provider and the internet.	SMTP
Email to Email recipients	<ul style="list-style-type: none">• Internet access (or at least an Email gateway)• Email Recipients that route their emails to SMS, like the norwegian mobilpost.no• No SMS provider is needed, but recipients need it on an individual basis.	Good. Depends on the internet service provider and the internet.	SMTP

In the above table "SMS provider" is a commercial company that specializes in serving as an SMS gateway. Earlier the telecom companies themselves had SMS gateways. Because of bad security and abuse, these services has been shut down (at least in Norway). Therefore there has come a new market for SMS gateways that are much better, fairly cheap and demands user authentication. Contact the Prilog vendor for a good suggestion for world-wide SMS coverage (that is – send SMS from anywhere in the world to anywhere in the world).

If you already have a broadband connection or at least an Email gateway (server), consider using this. Dialup is always more unsafe because something usually goes wrong when dialing sooner or later. Ships with occasional satellite coverage would probably be best off using the onboard email gateway and synchronize email when online. Ships may also use dialup. In this case Prilog will dial and send SMS as described in the above table.

Note that the "Email to" alternatives above actually involves using the "Send Email" action filter, and not the "Send SMS" action filter. It is included in this consideration, however, to clarify which options exist to send SMS.

Basically the "Send SMS" action filter in Prilog creates a queue of text files in the OUT-SMS directory below the Prilog directory – one text file for each SMS. These files are picked up by the external software from the SMS provider which in turn sends the SMS to the SMS provider using one of its methods (direct dialup, transfer via internet or as Email).

Prilog has a global flag that enables or disables SMS sending. This is useful for avoiding sending SMS'es after working hours. The status of the flag is observed in the status bar at the bottom of the Prilog Writer main window. This flag is only checked by the "Send SMS" action filter (not the "Send Email" action filter). It can be manually set in the File / Options dialog. It can also be set or cleared by a command in an incoming text line in Prilog. Define a search criteria in an action filter and add either a "SMS Enable" filter or a "SMS Disable" filter as actions. This lets Prilog turn SMS on/off by the automation system itself.

In the "Email to Email recipients" method, you usually have email addresses like 12345678@mobilpost.no. The subject and body of the email determines how the SMS looks like on the mobile.

MMS Sending from a web camera is also available upon request.

10. General usage of Prilog writer / logger

10.1. General information

Prilog writer / logger is run minimized when started. You maximize Prilog only when you are going to change setup or view the current setup / status of the running program. Do this by double-clicking the program at the taskbar.



Note that when Prilog is closed by using ALT-F4, the window menu (system menu) or by the X in the upper right window, Prilog is not terminated, but minimized to a taskbar-icon. In earlier chapters, the main configuration settings for communication, filters and data file is covered in detail. The file menu invokes these. Here is an explanation of other user features of the Prilog writer.

10.2. Menu items

10.2.1. File / Communication setup

This enters the communication setup dialog for configuring serial input ports. See own chapter for explanation.

10.2.2. File / Filter setup

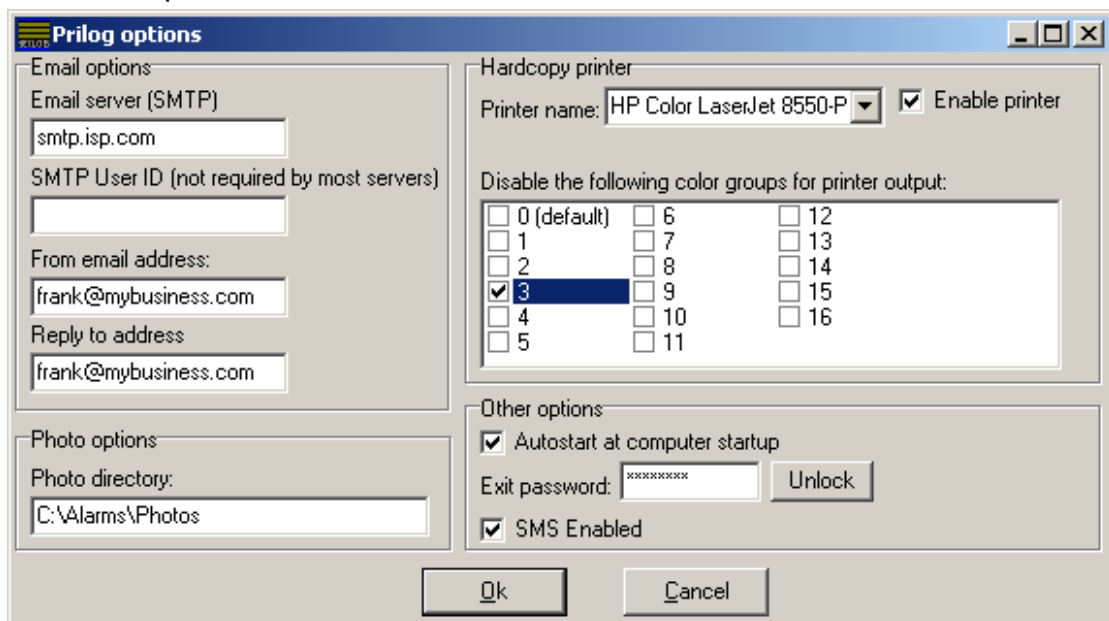
This enters the filter setup dialog. See own chapter for explanation.

10.2.3. File / Data file setup

This enters the setup for the data file PRILOG.DAT.

The only changeable option in this version is the channel name.

10.2.4. File / Options



The options dialog contains various global Prilog settings:

Email options:

Email server (SMTP). This specifies the IP address or host name of the outgoing email server. SMTP stands for Simple Mail Transfer Protocol and is the standard protocol on the internet for sending mail.

SMTP User ID: Most servers doesn't need to specify a SMTP user ID, so in most cases this can be left blank, but if needed, the network administrator will supply the correct information.

From email address: Specify the email address that gets to be the 'sender address' of the outgoing emails.

Reply to address: Specify the email address that gets to be the 'reply to address' of the outgoing emails.

Photo options:

The photo directory specifies where the Prilog reader looks for photo images (explained in other chapter).

Hardcopy printer:

Specify the printer name to be used to print on paper. Check the 'enable printer' to enable printing. Check also any of the color groups below if you want a particular group to *not* be printed. See own chapter of 'hardcopy printer' for more details.

Other options:

Autostart at computer startup. If checked, Prilog generates a unique entry in the Windows registry under HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run that launches Prilog during PC boot. If unchecked, Prilog deletes the registry entry. Note that you must have registry write access to enable this.

Exit password: Define a password here to be used if Prilog should be terminated. If the user doesn't supply Prilog with the correct password, Prilog won't be terminated. If a password is defined, you must first click the 'Unlock' button and write the password to be able to change or delete the password.

The checkbox "SMS Enable" has two functions, (1) Indication and (2) Manual override.

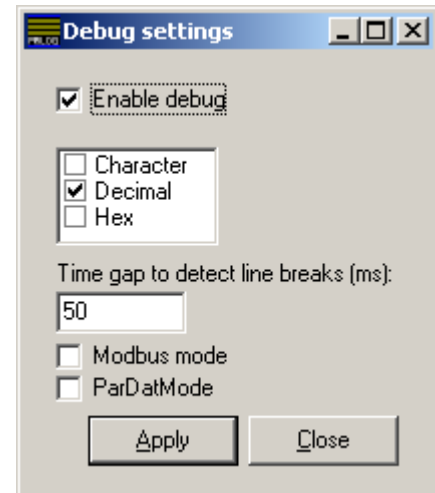
1. Indication. At any time, the checkbox is marked, this means that SMS-sending is globally enabled in Prilog (only for search-criteria's which are enabled). If there is no hatch, no SMS-sending will occur. If a "SMS enable" color-action is defined, it will activate SMS and be indicated by this checkbox with a hatch. If a "SMS disable" color-action is defined, it will deactivate SMS and be indicated with no hatch in the checkbox.

2. Manual override. By clicking in the checkbox, the state of the SMS-sending will toggle - either activate or deactivate.

10.2.5. File / Debug mode settings

This enters a dialog for specifying a special debugging mode. In this mode, Prilog bypasses all filters and character/word handling and rather passes the data stream through a 'debug' filter. The output of this filter is then sent through the filters as normal, although normally no filters would be configured. The debug mode outputs data in decimal, hexadecimal or ASCII format. The purpose is to debug the characters more deeply. It could be used as a protocol analyzer to inspect binary data. If there is two-way communication, a 'listener' cable could be attached so that both send and receive would be received into Prilog. By specifying a time-gap between characters, you can separate the two channels data into its own line. This is because the debug mode treats the data as a binary stream rather than lines terminated with a CR or LF. Therefore time should be used as a separator.

If ModBus mode is checked, Prilog attempts to decode the data directly to ModBus format. ParDat mode denotes using 32 bits values rather than 16 bits values.



Note that this function is not necessary working 100% as suspected and may not be suitable for each specific situation.

10.2.6. File / Block input from I/O ports

Enabling this, Prilog blocks any input from the input ports until disabled again by choosing this menu.

10.2.7. File / Exit

Terminates Prilog writer after a confirmation.

10.2.8. View / Serial port status

Displays a list of all configured port and indicates their status.

10.2.9. View / Line extraction database

Displays the current live LX database. Use the scrollbar to scroll through the list. The 'Clear database' button clears all names and values, but Prilog constantly updates it with new values as received from the incoming lines.

10.2.10. Test / Generate lines

Generates demonstration lines for debugging / testing. These lines emulate port number 1, and are therefore run through the normal Pre/Port/Post filters. This can be used for debugging or demonstration purposes.

10.2.11. Help / Contents

This opens the Prilog user manual.

10.2.12. Help / License information

To view current license or upgrade Prilog, use this menu. See own chapter for how to upgrade.

10.2.13. Help / About

Shows program information and version number.

11. General usage of Prilog reader / viewer

11.1. Menu items

11.1.1. File / Open another channel

This menu item will connect to another Prilog writer. It will bring up a file selection box where you can browse through your own computer or the network and specify the directory. You may select any file in the directory. Chose "open" and Prilog will connect to this writer and let you inspect the data in it.

11.1.2. File / Save settings on exit

Selecting this menu item will toggle on / off the saving of the prilog reader settings. The settings include colors and font information used. The settings are stored within Windows at the computer you are running and the next time you start Prilog reader, it will use this information. See the chapter called "Prilog reader: saving setup in windows registry" for details.

11.1.3. File / Stay on top

Check this menu item to make the reader window a 'stay on top' window – thus staying on top of other windows – even if they have focus. Other 'stay on top' windows may, however, stay on top of Prilog even if this option is enabled.

11.1.4. File / Analyze

Lets you analyze – or sort – the Prilog data on number of occurrences. Prilog analyzes the lines and recognizes similar alarm/events and shows a sorted list with the most-occurring lines at the top. Please find more details in own 'Analyze' chapter.

11.1.5. File / View photo

The photo function is a way of associating alarms/events with a photo taken at that time. Prilog relies on a photo directory for photos and displays the photo nearest in time of any selected line on the screen. Double-click on a line to 'select' it loading its associated photo in the photo window. The photo directory is defined in the writer under File / Options. Note that only jpegs and bitmaps are supported.

11.1.6. File / Font setup

Let's you choose the font type and color used.
SHORTCUT: Press the "Font" button on the tool bar.

11.1.7. File / Set default font

Sets the default main font. The change takes place immediately in the main window.

11.1.8. File / Only fixed pitch fonts

Enabling this option, only fixed pitch fonts are shown in the font setup. The pitch of the font is the width of each character. Normally each letter has a different width, but fixed pitch fonts are all equal width – thus making it very suitable for Prilog which often displays text with columns given by the text itself using space characters as separators. This makes sure the text is shown pretty on screen.

11.1.9. File / Export

Let's you export data from Prilog to a file. See the chapter explaining export for details.
SHORTCUT: Press the "export" button on the tool bar.

11.1.10. File / Print page

This menu item will print the complete Prilog window as you see it, but with white background. To print larger amounts of lines, use the export function or search/list function. For example, export to file and open it in a word processor and print the text from there.

11.1.11. File / Print setup

Lets you chose printer and setup for the printer.

11.1.12. File / Exit

Terminates the Prilog reader / viewer.

11.1.13. Edit / Copy selection


You can mark a selection of lines with the mouse by clicking and dragging inside the text window. By selecting this menu, this portion of the text is copied to the clipboard.


11.1.14. Edit / Copy whole page

This function will copy all Prilog data visible in the window to the clipboard. All data existing on the clipboard is possible to paste into other program - for instance word processors. For instance, use this function to quickly paste lines from Prilog into a report among other text.

11.1.15. Edit / Find

This function lets you search through all Prilog data or a limited time period. It includes AND / OR / NOT operators. See the chapter about the find function for details.

SHORTCUT: Press the button on the toolbar with a magnifying  glass:

NOTE: If you press mouse button 2 (usually the right button) on  the search-button above, you will (if you are in manual mode) be positioned at the last search-position. The found line is highlighted.

11.1.16. Mode / True mode

In true mode, you will only see new lines coming in to Prilog. You can't move up or down in the lines as you can in manual mode. If the screen is blank, it therefore means you have not received any lines since Prilog was started.

As lines are received, they will be printed out on the screen. You will only see the amount of new lines that will fit into the window. To increase the amount of lines, either enlarge the Prilog window or use a smaller font. If you go to manual mode and back to true mode, the screen is updated to print the newest lines.

SHORTCUT: Press the button on the toolbar marked "TRUE".

11.1.17. Mode / Manual mode

In manual mode, you will "freeze" at the current line. You may scroll upwards or downwards with keyboard, mouse click or using the scroll-bar to the right of the screen.

These buttons on the toolbar work in manual mode:

HOME or H: Takes you to the line that is newest. The newest line is the line at the bottom of the screen.

UP: Scrolls upwards - thus backwards in the data towards older data.

Down: Scrolls downwards - thus forward in the data towards newer data.

Page up: Will scroll one screen upward towards older data.

Page down: Will scroll one screen downward towards newer data.

Note that these keyboard shortcuts also work:

Up/Down/Page up/Page down work the same way as above.

CTRL-Page Up scrolls approximately 25 pages upwards.

CTRL-Page Down scrolls approximately 25 pages downwards.

In addition, there is a vertical scroll-bar at the right of the screen. Clicking it with the mouse and holding the button down lets you scroll backwards in the data file. If the scroll bar is at the lowest position, you are at the newest line. If the scroll bar is at the highest position, you are at the end of the lines (oldest data).

NOTE: Scrolling downwards may cause you to “turn over” when you reach the newest incoming line to the oldest data. You can still scroll downwards, but the effect is that you just scroll around the round buffer. The same happens if you scroll upwards. When you reach the oldest line and scroll more, you will see the newest line.

KEYBOARD: M, ALT-M + M

SHORTCUT: Press the button on the toolbar marked “MAN”.

11.1.18. Channels

This menu contains a list of all channels that are registered (except the currently selected) - which Prilog knows about. When a channel is opened, it is remembered, so that this menu can select it later - quickly. It is the reader that maintains this information and stores the information in the Windows registry.

11.1.19. Help / Contents

Opens the Prilog user manual.

Prilog also has a help file



Pressing will also open it.

(PRILOG.HLP) giving short help on some issues. These are mostly issues regarding problems when an error box appears and you press the help-button.

11.1.20. Help / License information

To view current license or upgrade Prilog, use this menu. See own chapter about licensing for details.

11.2. Obtaining ASCII information on a character

To obtain the decimal and hexadecimal value of any character Prilog receives, move the mouse pointer over the character and read its value in the status bar at the bottom of the Prilog window.

11.3. Color selection feature



The color selection bar let's the user make visible or invisible the different colors that are defined in the Prilog writer / logger. The numbers 1 to 16 correspond to the color group 1 to 16 in the action-setup in the writer. The "Std" button means the "standard" color the lines get if they don't belong to a color group. These lines can also be invisible. This feature works both in true and in manual mode.

Clicking on either of these buttons disables the rest of the buttons (colors), so you only see the color group you have clicked. The disabled groups are marked with an 'X'.

Clicking with the other mouse-button has the opposite effect.

In the search-function, you can choose to search within only the visible lines, or also in the invisible ones. The export-function does not care whether a color is visible or not. It will export lines, even if they are not visible. Otherwise, this has no effect. It has only a visible effect on the reader.

KEYBOARD: Std = | (upper left key, below ESC)

(Will toggle Color group 1..9: Press 1..9

visible/ Color group 10: Press 0

invisible) Make all colors visible: Press BACKSPACE

11.4. Search

Search - enter your case-insensitive search expression

Expression: Alarm OR Warning OR

AND

Generator 2 OR OR

AND

NOT ☒ Fuel OR OR

☐ Limit to this time period? From: 15.10.2004 09:19:44 To: 15.10.2004 09:19:44

☐ Limit to colour-selection?

Search from which position?

☐ From last search ☒ From newest line

Search! Stop search List Earlier saved searches: Save this search

Help Close Last list

Prilog lets you search all stored data in several manners. Every search is executed backwards in time, so that you always will first get the newest line(s). The search criterion is applied for each line in the data file at a time.

The main part of the search form is the search criteria. Here you enter text in the open field(s). At least ONE field must be defined, but up to 9 fields can be used.

You move between the fields by clicking the mouse.

KEYBOARD: Press the TAB button on the keyboard to switch between the fields.

The AND / OR / NOT mechanism works as follows:

The first horizontal line consists of three open fields. There is a logical OR operator between these. Example: ALARM OR PUMP OR 4539 will search for lines containing either "ALARM", "PUMP" or "4539", two of them or all three. The same goes for the last two horizontal lines, BUT:

Between line 1 and 2 there is a logical AND operation.

Between line 2 and 3 there is also a logical AND operation, with an optional inversion (NOT operator).

This means in practice that word(s) you enter in line 1 (first line), MUST exist together with word(s) defined in line 2. The same goes for line 3, but if you check the "NOT" box, word(s) defined in line 3 must NOT exist in the lines.

This means that the complete boolean expression is as follows:

A	B	C
D	E	F
G	H	I

Result = (A OR B OR C) AND (D OR E OR F) AND (G OR H OR I)

Alternatively with the NOT box checked:

Result = (A OR B OR C) AND (D OR E OR F) AND (NOT (G OR H OR I))

You can limit your entire search to a specified time period. You do that in two steps:

First you check the square box to the left of the "Limit to time period" text, second you modify the From/To date and time controls.

NOTE: You can modify the dates/times in more than one way. You can position yourself with the mouse or the keyboard using the TAB key and left/right keys. Then you can use the up/down keys to change the value one step up or down. This goes for dates and times (the individual numbers). Alternatively, you can use the mouse and press the arrow(s) to the right of each box. For the date fields, a calendar will pop up letting you easily see all the days in a month and a red ring around the current day if the current month/year is displayed. For the time fields, the number by which the "cursor" is next to, is increased/decreased pressing the arrows with the mouse.
KEYBOARD: ALT-I

11.4.1. Limit to color-selection

The current color-selection determines which color-groups of lines that are visible. If you would like to limit the search to only be applied to visible lines, check this. If you don't have checked this and Prilog finds a line that is invisible, you are told so and you need yourself to make it visible to observe it.

11.4.2. Search from which position?

You have two possibilities:

Search from last search: This will start the search at the line that is one older than the last found line (remember that search always goes backward in time) and go backwards. This enables the "Find next" by pressing one more time at the "Search" button.

The very first time you use this function the "last" position operates so that the latest arrived line (true position) is found first.

Search from newest line always makes the search start at the newest line (true position) and goes backwards in time.

11.4.3. Executing searches

Press the "Search" button to start the search.

You will then see a progress-indicator telling you how far the search has come in the whole search. You also see the date and time of the current position in the data file where Prilog is searching. This way, you may stop the search if Prilog searches older data than you think is necessary.

KEYBOARD: Press ALT-S.

During search, you are not allowed to close the window. To abort the search, press the "Stop search" button.

KEYBOARD: Press ALT-T.

11.4.4. Saving search results

First, define the search you want as described above. You may then test it.

Then press the "Save this search" button to save the search criteria. You will then be prompted to give a name to this search. You may enter anything you like, for instance "Weekly search for thruster alarms".

Press Cancel or leave field empty to abort save.

The search is saved in the Prilog writer's INI file in its own section.

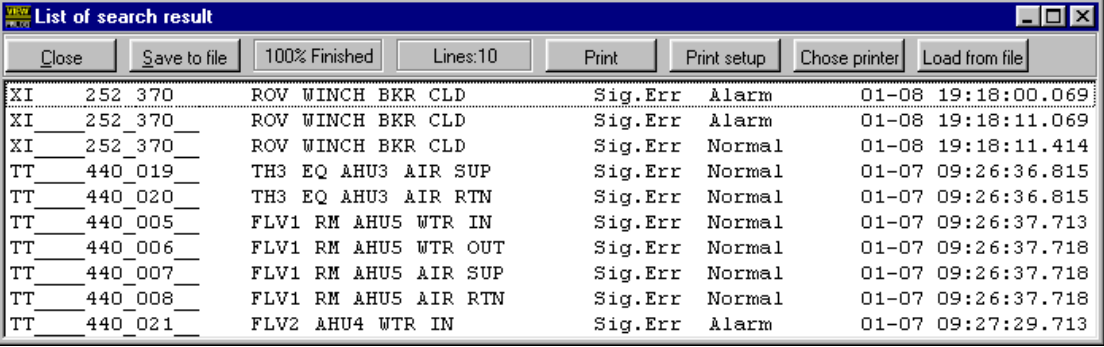
KEYBOARD: Press ALT-A.

11.4.5. Loading a saved search

To load an already saved search, simply press the box below the "save this search"-button and select one from the list.

KEYBOARD: Press ALT-O and then press Up / Down to browse the saved searches.

11.5. The LIST function



Close	Save to file	100% Finished	Lines:10	Print	Print setup	Chose printer	Load from file
XI	252_370	ROV WINCH BKR CLD	Sig.Err	Alarm	01-08 19:18:00.069		
XI	252_370	ROV WINCH BKR CLD	Sig.Err	Alarm	01-08 19:18:11.069		
XI	252_370	ROV WINCH BKR CLD	Sig.Err	Normal	01-08 19:18:11.414		
TT	440_019	TH3 EQ AHU3 AIR SUP	Sig.Err	Normal	01-07 09:26:36.815		
TT	440_020	TH3 EQ AHU3 AIR RTN	Sig.Err	Normal	01-07 09:26:36.815		
TT	440_005	FLV1 RM AHU5 WTR IN	Sig.Err	Normal	01-07 09:26:37.713		
TT	440_006	FLV1 RM AHU5 WTR OUT	Sig.Err	Normal	01-07 09:26:37.718		
TT	440_007	FLV1 RM AHU5 AIR SUP	Sig.Err	Normal	01-07 09:26:37.718		
TT	440_008	FLV1 RM AHU5 AIR RTN	Sig.Err	Normal	01-07 09:26:37.718		
TT	440_021	FLV2 AHU4 WTR IN	Sig.Err	Alarm	01-07 09:27:29.713		

The list function is intended for display a list of found lines in a separate window. A list will search through the data and display the lines satisfying the search criteria. A maximum of 32.700 lines can be generated in this way.

You generate a list by pressing the "List"-button within the search function.
KEYBOARD: ALT-L

To create a list, you define search criteria, and optionally set a time-limitation. When pressing the List-button, you are prompted to change the maximum number of resulting list lines. The search will stop when it reaches this amount of lines. By checking the "newest first" box, the order of the list result is reversed. This controls how Prilog displays and prints the lines on the printer.

All search options are applied to the list function, also the time-period limitation (except the "From position", see below under the note).

A list-window will appear and lines that satisfy the search-criteria will be inserted. After searching, you may use the buttons inside the window, and close the window when you are finished. If you want to delete a single line or a range of lines, you may select the line(s) and then press DELETE. You may also press the "Save to file" button inside the list-window to save all list-text in a text file (*.TXT). The lines are separated with CR and LF. This file may then be used for whatever program that can import or manipulate / use text files. To load an already saved list, the "Load from file" button can be used.

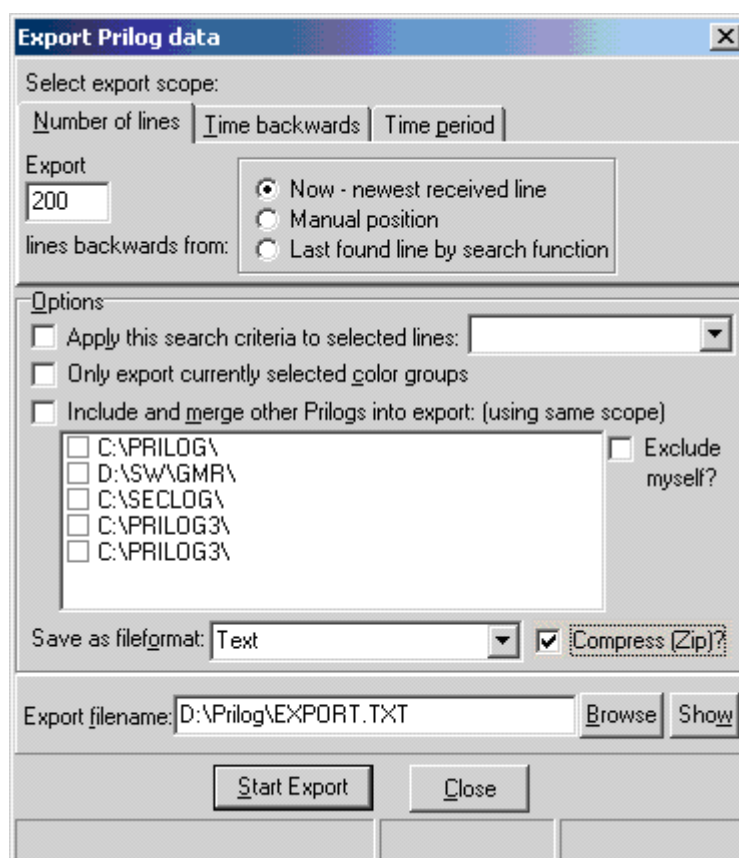
Note: The "From position" is not applied when a list-search is executed with a time-period limitation. The searching is always started at the "To-time" position and searched backwards in time.

The 'Last list' button shows the list window if it is closed.

11.6. How to go back to the last found line

If you "loose" the line you found by a search (for example if you moved with the scroll-bar), you may go back to the line by pressing mouse button 2 (usually the right button if you have a right-handed mouse) on the search-button on the tool-bar.

11.7. Export



11.7.1. General

Export means grabbing a set of lines out of the main data file (DAT file) and saving it to another file, called the export file. The export file is then completely independent of the main data file and is therefore “dead”. The export file can be used for these purposes among others:

- Moving the selected data from one geographical location to another
- Inspecting, manipulating and calculating the data with another program
- Importing the data in another program, database, spreadsheet etc.
- Printing the data in other formats supported by Prilog.

11.7.2. Export options

You can either:

- Export a number of lines from the last received line and backwards
- OR
- Export a number of days/hours/minutes backwards from now
- OR
- Export between to dates

You select one of these three modes by selecting the correct "tab" in the upper part of the dialog.

11.7.3. Exporting number of lines:

You can export a specific number of lines starting from the newest line (latest incoming). This is useful for extracting only the newest lines.

From now - newest received line

This will make the start-point the newest incoming line and count backwards in time until one has reached the desired number of lines to export.

From Manual position

If Prilog is in Manual mode, it is actually scrolled backwards in time to a specific time location. Then, export will start with the line at the bottom at the screen and export backwards in time from this line.

From Last found line by search function

Independent of whether you are in True or Manual mode, or where you are in manual mode, selecting this option will start exporting from the last found line of the search function. This requires that you earlier have used the search function and have found a line, which then is highlighted in Prilog. Then, the export function will start exporting from this line and go backwards in time.

11.7.4. Export time backwards

This will export a number of days, hours or minutes backwards in time. This is useful for periodical exporting of data for permanent storage. You have then the three options explained above for indicating the start-position of the export.

11.7.5. Export time period

This will export data between the two specific times you indicate in the dialog.

11.7.6. Options.

Apply this search criteria to selected line

Select this checkbox and a previously saved search in the combo-box at the right. Then, Prilog will subject all exported lines to the search filter and only export those matching the filter.

Only export currently selected color groups

Select this checkbox if you want to limit the export to only the currently selected color groups. If for example only color groups 1,2 and 3 are visible in Prilog, then only these groups will be subject to export.

Include and merge other Prilogs into export (using same scope)

Select this checkbox and one or more Prilog channels below to merge two or more channels into one export file and sorting the exported lines by time. If no channels are listed in the box below, they must first be opened in the Prilog reader to be registered. Then, later, open the export dialog and they will appear in the list. The 'Exclude myself?' checkbox may be checked to merge together only the other channels, not the currently opened one. This way, one has total freedom of selecting which channels should be merged.

11.7.7. Export file formats

The current supported export formats are:

- Normal DOS text file with CR + LF as line-ends. This is the main format used by PC's and DOS/Windows. There is no maximum line length.
- Normal UNIX textile with LF as line-ends. This format is exactly identical to the above with the exception of the CR character that is not used. This format is widely used in the Unix world, but most programs on a PC (Dos/Windows) can also read this format. There is no maximum line length.
- Prilog native data format. This format is readable for Prilog. This makes the exported data available in its own Prilog reader window as if it were an ordinary Prilog channel. The line length is equal to the original Prilog data.

KEYBOARD: ALT-O

11.7.8. Export filename

Specify the name of the file that will contain the exported lines. You can enter this manually in the open field, or browse the file system using the BROWSE button.

Chose a file location either locally or on a remote network computer.

Pressing the 'Show' button will launch the export file. Because it normally ends with 'TXT' in text file exports, notepad will be launched because normally notepad is associated in Windows with TXT files.

KEYBOARD: ALT-F, ALT-B, ALT-W

You can compress the resulting export file by checking the 'Compress (ZIP)' box. This generates a new file called the same as the export file, only with a .ZIP extension. The original export file is not deleted.

11.7.9. Executing the export

Press the "Start export" button to start generating the export file. A progress-bar will be displayed and feedback given.

KEYBOARD: ALT-S

You must close the export window to return to the main Prilog window.

KEYBOARD: ALT-C

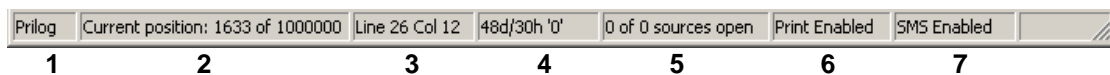
NOTE: Prilog will be running normally in the background during export.

You may press the HELP button to open this page.

KEYBOARD: ALT-H

11.8. Status bar help

The status bar in the bottom of the Prilog window has the same layout on the reader and the writer except for positions 5-7. It is divided into these sections:

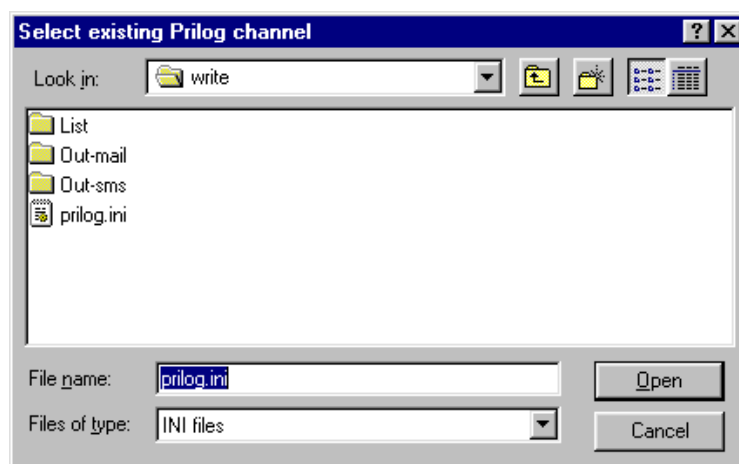


No.	Description
1.	Channel name. You can change it in the options dialog. (File / Options)
2.	Current line position in the DAT file. It is showing where are currently writing in the DAT file. For reader: The manual position is also showed.
3.	The line number and column of the mouse cursor in the text data.
4.	ASCII value of character below mouse
5.	Writer only: Input port status. Tells how many of the configured ports are open.
6.	Writer only: Printing active or inactive. If active, lines within print-enabled color groups are printed to a hardcopy printer if one is defined. See File / Options for printer setup. Here you can also select which color groups that are enabled. By default all color groups are enabled.
7.	Writer only: SMS Enabled or Disabled. Global parameter that shows if SMS is allowed to be sent or not. It is possible to manually override it in the options menu. The action filter also has actions that enable or disable sending of SMS.

12. Configuring a reader (locally or on a network PC)

There is only one installation program for Prilog and this installs both a reader and a writer. So, if there are X writers installed on a computer, there are also X readers installed.

A Prilog reader / viewer may connect to one Prilog writer / logger at a time for inspection of the data. More readers may run at the same time, connected to the same or different writers. However, a reader has the File / Open command that will let the user browse to another directory and open another channel. This will "switch" a reader to another channel. This dialog-box will also appear if a Prilog reader / viewer is *started* in a directory where there is no existing channel. After a reader has opened a channel, it remembers where it is and puts it on the "channel" menu for easy access later.



12.1. Configuring a network reader

To set up a network reader of Prilog, the installation must be run according to the description in another chapter in this manual. The correct HASP key should be inserted which has a license for a reader (but not necessarily a writer). Since there is no need for a writer on this computer, it should just be disregarded and not started. There is no need to install more than once, because the same reader can be started more than one time (for example via shortcuts for easy access, see below).

The following description assumes that the network is installed properly. Now, after starting the reader, the above dialog-box appears. From this box, navigate to the network and find the computer and base-directory where an already-configured Prilog writer exists. If you know the computer name of the other computer, the form \\ComputerName may be written as filename. Pressing enter, opens the available resources (shared directories) on that computer. Open the "PRILOG.INI" file of the Prilog channel you want to connect to. If the channel opens and you can see the lines, you have successfully opened the channel.

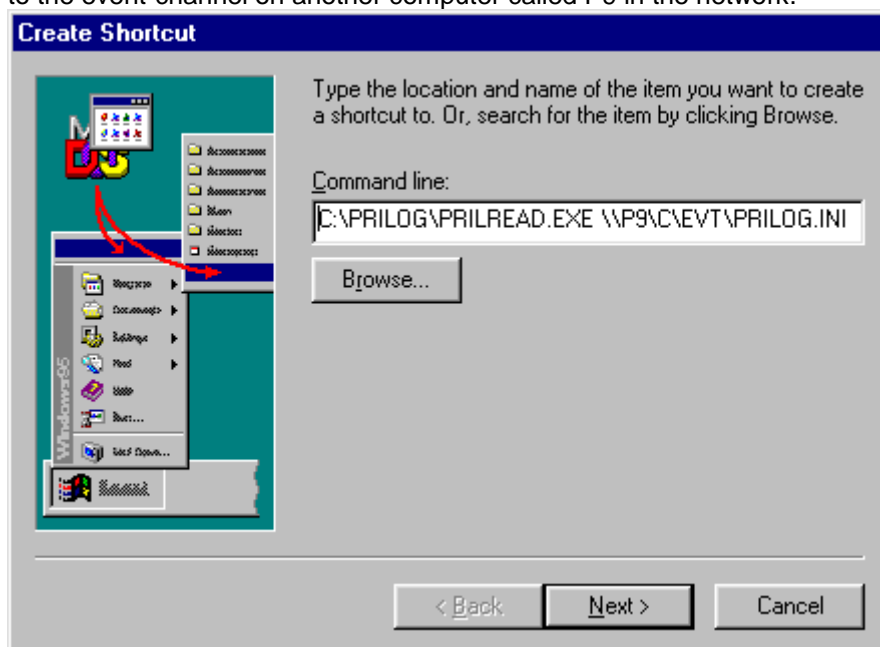
To save the time it takes to open the channel, you could create a shortcut to it:

12.2. How to create a shortcut to a Prilog channel (locally or via network)

1. Start the windows file explorer on the computer where the Prilog reader is.
2. Browse to the directory where you want to store the shortcut (e.g. the desktop¹ or in the Startup directory²).
3. Press mouse button 2 (property button) and chose New / Shortcut
4. Specify under "Command line" the full path to the local base directory of the Prilog reader.³
5. Add "PRILREAD.EXE" to the command line.⁴
6. Add a space and append as a parameter the complete path and name for the PRILOG.INI file of the channel you will connect to⁵. Note that you have the possibility to use drive mapping (the form U:\ALARMS\PRILOG.XML) or UNC-mapping (Universal Naming Convention, the form \\ComputerName\ShareName\Directory\File). The only difference is that the first requires the drive-letter to be mapped before use. Mapping is done under the windows file explorer under tools / map network drive.
7. Press next and give the shortcut a meaningful name.

You can now double-click the new shortcut to open the new channel.

The example below is a PC that is a Prilog network reader. We wish to make a shortcut on the desktop to the event-channel on another computer called P9 in the network.



This example shows creating a shortcut to the "EVT" channel on the computer "P9" under the share name "C" and directory "\EVT".

¹ E.g. C:\WINDOWS\DESKTOP

² E.g. C:\WINDOWS\START MENU\PROGRAMS\STARTUP

Windows automatically starts files or shortcuts existing in this directory when Windows starts.

³ E.g. C:\PRILOG\

⁴ E.g. C:\PRILOG\PRILREAD.EXE

⁵ E.g. C:\PRILOG\PRILREAD.EXE U:\ALARMS\PRILOG.XML

Or: C:\PRILOG\PRILREAD.EXE \\LOGGERPC\CDISK\ALARMS\PRILOG.XML

13. Escape sequences and special characters

Prilog recognizes escape sequences and special characters by means of the system described below. A "special sequence" starts with the backslash character "\" and must terminate with the semicolon ";". If a special sequence does not start or end with these characters, it is regarded as a plain search-text – which is treated case-insensitive.

A special sequence may be mixed with other text and there may be more than one in the search criteria. The exception is the FILE-sequences, which allows for *only* this sequence, and is only allowed in the action-setup.

It is important to know that characters generated in a special sequence *are* case-sensitive, meaning that "\A;" is not the same as "\a;"

These sequences are used for these purposes (and possibly more):

- Defining characters which are impossible to enter via the keyboard
- Defining characters that do not appear correctly on the screen.
- Defining search-criteria's which are case-sensitive
- Defining a wildcard using the "?" character (matches any character)
- Defining search-characters which matches special codes and escape sequences

A special sequence may not be nested (one appear inside another).

This is a complete list of the special sequences:

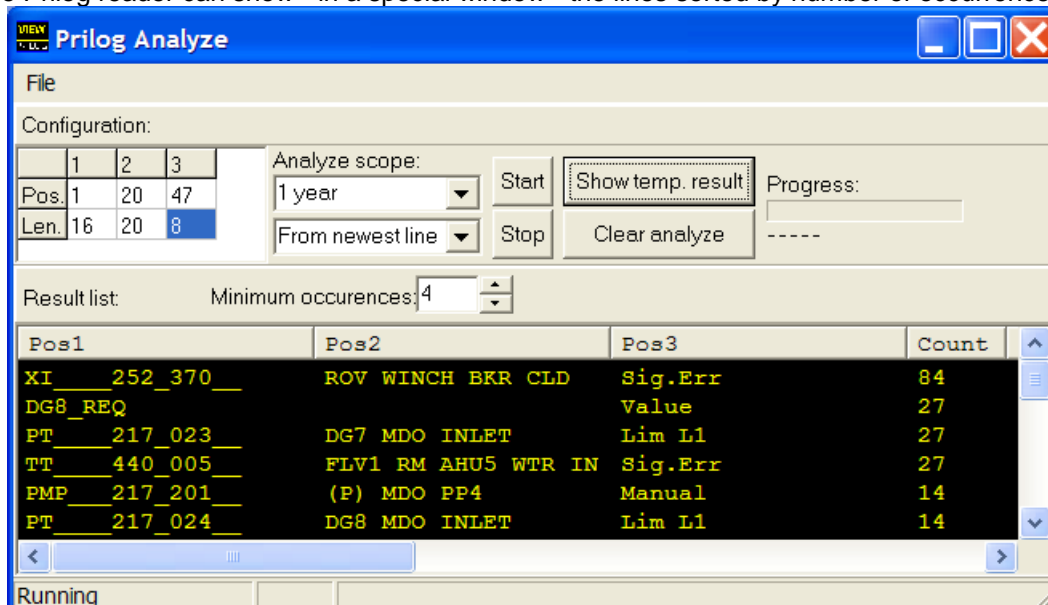
\ESC;	Denotes the ASCII ESCAPE (ESC) character that is used in start escape-characters. The ASCII value is 27.
\n;	n is a ASCII value from 1 to 255. For example \94; denotes a "^". Use this special sequence to define other ASCII character or binary values other than ESC. Hint: For a list of the ASCII characters, go to the "Exclude characters" setup.
\?;	Will match any ASCII character. Use this as a wildcard when for example an escape-sequence includes a number that varies.
\\;	Because the "\" character starts a macro, enter this to denote the actual character "\" in the search criteria. However, if the semicolon does not exist in the rest of the string, Prilog understands that this is not a start of a special-sequence and thus treats it as a plain "\".
\ORFILE x;	"x" is a file name that contains search-criteria used as a substitute for this special sequence. Between the lines in the file x there is the OR operator.
\ANDfile x;	"x" is a file name that contains search-criteria used as a substitute for this special sequence. Between the lines in the file x there is the AND operator.
\NORFILE x;	"x" is a file name that contains search-criteria used as a substitute for this special sequence. Between the lines in the file x there is the OR operator and the final result is inverted logically.
\NANDFILE x;	"x" is a file name that contains search-criteria used as a substitute for this special sequence. Between the lines in the file x there is the AND operator and the final result is inverted logically.
\any-other-text;	Any other text between "\" and ";" is used as-is and is case-sensitive, meaning that text must match <i>exactly</i> as it appears. Note: This is the method of inserting a bit of text that should not be case insensitive.

Examples of search criteria:

INLET	Will match lines containing "INLET", "Inlet", "INLET034", "#####InLet.2" etc.
\ESC;	Will match lines containing the escape character (ASCII 27).
\ESC;A	Will match lines containing an "A" or "a" after the escape character.
\ESC;\4;o	Will match lines containing the escape character, followed by ASCII character 4, then a "o" or a "O".
\ESC;\?;t	Will match lines containing the escape character followed by ANY character and then a "t" or a "T".
\ESC;\?;\?;o	Will match lines containing the escape character followed by ANY character, followed by ANY character, followed by a "o" or "O".
\ESC;\14;	Will match lines containing the escape character followed by the ASCII value 14.
\198;\233;	Will match lines containing the ASCII character 198 followed by the ASCII character 233.
PUMP2\?;A	Will match lines containing "PUMP2" (or for example "Pump2"), followed by ANY character, followed by an "a" or an "A".
\ALARM;	Will match lines containing "ALARM", but not "Alarm" or "ALarm" etc
\A;\L;\?;x	Will match the uppercase characters "AL", followed by any character, then a "x" or a "X".
A	Will match lines containing "A" or "a".
\A;	Will match lines containing "A", but will not be triggered by "a".
\65;	The same as \A; Will match lines containing the ASCII character 65 ("A"), but will not be triggered by "a".

14. Analyze function

The Prilog reader can show - in a special window - the lines sorted by number of occurrences.



You activate this feature by entering numbers in the fields in the upper left corner. 'Pos' is position and 'Len' is length. It gives the possibility to specify one, two or three fixed position columns in the incoming lines. The position is the character position of the first character in the column (for example 1, 20 and 47 in the above example). The length value is the length (or width) of the column in characters (for example 16, 20 and 8 in the above example). If the actual word at this position is shorter than this length, the blank space is trimmed away. If the word is longer, the word is truncated.

The analysis function will then, line by line, try to extract these column(s) from the incoming lines. If one position is defined, Prilog extracts only one word. If two positions are defined, Prilog extracts two words and so on. The result is for each line one, two or three words that join a unique entity.

Prilog then inserts this word (or these words) into the analysis database to see if they are there already. If so, it adds its counter by one. If not, it inserts it. When all lines are read, Prilog then knows which lines occur with the most frequency.

For example, by defining only one position, Prilog can examine how often different tagname occur. However, this is often not useful because a tagname may occur not only in alarms that go active, but also when it goes inactive or changes value. Therefore, you can define up to three columns so that for example description and event text also may be added as a criterion.

The result is displayed in the window when finished. The values are showed for the columns (one, two or three) and the number of occurrences for this combination.

This function may be manually started and stopped. If running (as indicated on the status bar), a temporary result may be showed. Press the 'clear analyze' button to empty the analysis database to start over – for example to change the analysis scope. The scope may vary in time backwards and either from the newest line or the current position (manual position) in the Prilog viewer. The list can also be reduced to only include frequencies over a specified minimum (default 4 occurrences). If desired the list (which normally covers much more than visible at one time in the window) can be saved to CSV or XLS file from the file menu.

If this function is defined, Prilog automatically starts to analyze when started. The analysis is done in the background so it doesn't interfere with the user interface. When the whole scope is covered, Prilog continues to add new incoming lines to the analysis.

15. Word database

The Prilog Reader can read a database of words along with a description of each word. If the user moves the mouse over a word on the screen that is found in the database, the description will appear in a box on the screen. If the user double-clicks on a word, a dialog-box will appear with the description also.

This system is designed for recognizing single words. A word is defined as a group of characters that is terminated on the left and right hand side of one or more spaces (ASCII value 32). A word may appear in any position on the screen, both in manual and true mode.

Database format

The database has the name WORDINFO.TXT and must be in the same directory as the executables. It is a normal DOS text file. This format makes it easy to auto-generate it from another database format.

The database has this format:

```
Word1;Description1;Description2;Description3;.....DescriptionN
Word2;Description1;Description2;Description3;.....DescriptionN
Word3;Description1;Description2;Description3;.....DescriptionN
(etc)
```

Word: A word consisting of all characters, but not space.

Description 1-n: A free-text description (may contain spaces)

Each description will be presented in its own line in the box on the screen, below each other

Each line in WORDINFO.TXT is one database entry. Add as many lines as necessary. The database size is limited by the amount of memory and CPU speed. 10.000 entries should work reasonably fast on a Pentium processor of 200Mhz.

Changing the field separator

The field-separation character is semicolon (;). It is possible to change this to another character by using this method in the WORDINFO.TXT file:

Separator=&

Word1&Description1&Description2&Description3

16. Photo function

The Prilog reader can relate a line (for example an alarm) to a photo and display this when double-clicking on the line. The purpose is to directly associate a photo to the event so that the time is as equal as possible between the alarm text and photo.

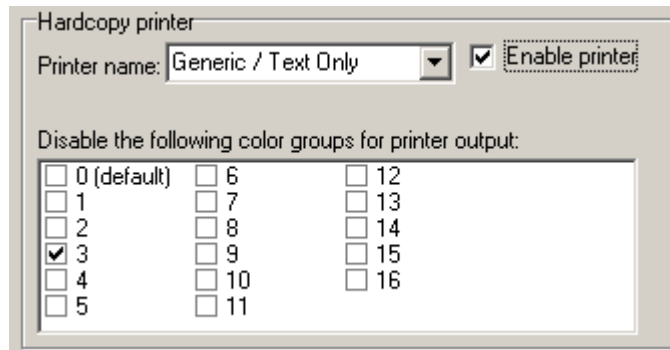
Prilog does not generate any photos, but only reads a given directory of image files and searches for the image with a timestamp closest to the timestamp of the line. Prilog relies on external programs to insert image files into this directory. The directory is specified in the Prilog writer / logger under the File / Options menu. Remember to restart the Prilog reader / viewer after changing this.

In the Prilog reader / viewer, select File / Photo on the menu to bring up a window that will contain the photos. When double-clicking on a line, Prilog scans the photo directory and shows the closet image file in time. Prilog doesn't care about the time difference. It only finds the closest in time, although this can be big. With the scrollbar, one can also skip to the next/last image to inspect a sequence of images.

17. Hardcopy printer

Prilog may print the lines it receives directly to a printer (usually a dot matrix printer). This is usually to secure a hardcopy of the lines in case of something happening to the computer. The best solution is of course to connect the printer directly to the sending equipment - not Prilog. This is because the printer then relies on the computer and Prilog being up and running ok. If this is not possible, printing from Prilog may be a good alternative.

Enter the File / Options menu, and see the 'Hardcopy printer' section of this dialog:



First, select a printer that is going to be used. Tip: some printers require entering properties on the printer, choosing the Advanced pane / Print Processor, then choosing WinPrint and TEXT as default datatype.

Prilog sends no formatting codes to the printer. Only raw text. Each line is only terminated by a Carriage Return (CR, ASCII 13), and a Line Fee (LF, ASCII 10). Some printers may accumulate enough lines in its buffer before it actually prints a whole page.

The preferred printer type is of course a dot matrix printer (or 'impact printer') with a continuous paper feeder ('tractor feeder'). Prilog can print to any printer that is supported by the Windows operating system. This means that Prilog supports any connection method to the printer available - parallel, serial, network, USB and so on.

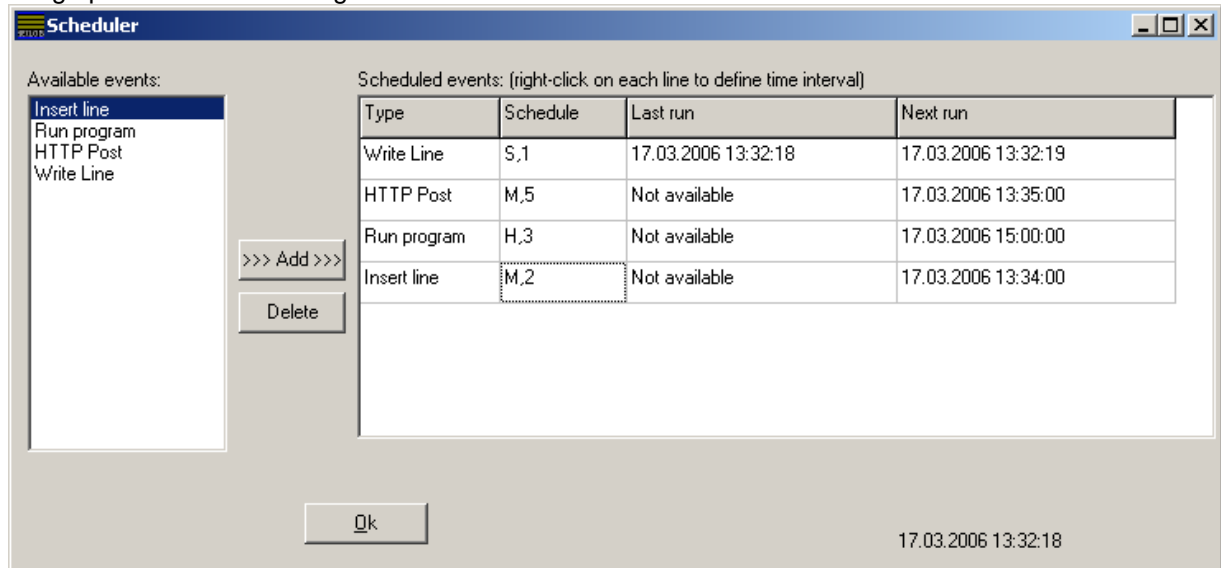
By default, Prilog sends everything that goes to the screen (and data file) to the printer. Optionally, the checkbox (as shown above) may be used to disable printing of certain color groups. Check the color group that you *don't* want to go to the printer. For example, this allows for printing only the alarms that go active, even if there exist other types of alarms and events. Accomplish this by the action filter(s) to group the lines into different color groups (see own chapter on action setup).

The 'Enable printer' checkbox serves as indication as well as command. Click this box to either enable or disable printing. However, some action filter may also enable or disable printing by itself. Therefore, this checkbox will also indicate the current status by automatically being checked or unchecked when the action filter changes it.

Note that there is a safety mechanism regarding printer queue size. Before printing each line, Prilog checks the number of printer jobs in the queue. If it exceeds 100, Prilog doesn't print the line. This to avoid accumulating excessive amounts of printer jobs in Windows. This may lead to missing lines in the printout.

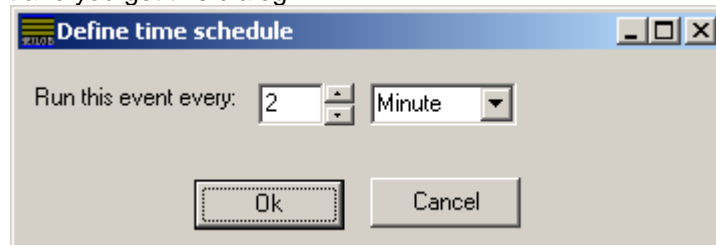
18. Scheduler

The Prilog scheduler can execute certain actions at a specific time interval. Choose File / Scheduler to bring up the scheduler dialog.



The available event types are shown in the left box. The scheduled events are shown in the right box. Double click an event in the left box to add it into the scheduler, or press the add button. To delete an event, select a line in the right box and click the delete button. To edit an event, double click it. To change the schedule, right-click with the mouse on the corresponding event and select the popup-menu.

When you add an event, there is no schedule defined, and the “Schedule” column is blank. To define a schedule, right-click it and you get this dialog:



Define an interval that suits you and press Ok. The schedule column then reflects your choice. “S” stands for seconds, “M” for minute and “H” for hour. “M,2” means the event is run every two minutes. The schedules are aligned to midnight. If an event is scheduled to be run every hour, it is run 00:00, 01:00, 02:00 etc.

The events are actually selected filters, and their description is found elsewhere in this document. The only difference, however, is that there is no “incoming line” that is inserted into the filter for each scheduled event. This hasn’t a big impact, but for example for the “Write Line” filter has an option to echo the incoming line to the port. This isn’t possible, as there isn’t any incoming line.

The scheduler dynamically shows the last and next scheduled time for each event.

19. Uninstalling Prilog

19.1. Normal uninstall (preferred)

To uninstall Prilog, first stop all writers and readers. Then open the control panel and the “Add/Remove software” icon. You should then be able to see a line containing “Prilog”. Click on this line and then the “Add/Remove” button. Prilog will then be uninstalled automatically. This is the preferred way to do it. Repeat this procedure if you have installed more than one Prilog.

If Prilog is not in the add/remove software list, it is probably because that Prilog was installed by copying the files manually onto the hard disk. In this case, please follow the instructions below for how to manually delete Prilog.

Uninstallation - Step - by - step procedure:

1. Open the Control Panel
2. Open Add / Remove software
3. Click on the line containing “Prilog” and then the Add/Remove button
4. Delete the DAT, PEK, XML-file and base directory manually if you want to.

NOTE: The fact that the DAT/PEK/INI files are not deleted could be used to your advantage. You may copy these files to another channels directory to use the data you already have collected. You may, however, just delete these files - including the base directory to completely delete Prilog.

19.2. Manual uninstall

This is the manual method to delete everything that has to do with Prilog if you won't (or can't) use Windows built-in de-installer. Note that you will continue to have Prilog in the control panels software-list if you do not follow the standard uninstall method above.

1. Delete the base directory (e.g. C:\EVENTS 1) and all files contained
2. Delete the start-menu folder by exploring to it, (e.g. C:\WIN95\START MENU\PROGRAMS\ALARM CHANNEL NO 1).
3. Delete the following registry keys (if they exist):
 - HKEY_LOCAL_MACHINE\SOFTWARE\HOGLUND MARINE AUTOMATION
 - HKEY_LOCAL_MACHINE\SOFTWARE\Prilog HOGLUND MARINE AUTOMATION

First, start the program REGEDIT.EXE or REGEDT32.EXE. Then click on the hierarchical “tree” on the left-hand side and move from HKEY_LOCAL_MACHINE and down to SOFTWARE. You delete the to Prilog keys just as if it was a directory (folder) in the Windows explorer (by clicking on it and pressing delete).

20. Troubleshooting communications

20.1. If you can't get Prilog to open the comport.

If Prilog is unable to open a COM port, it will appear at the rightmost field in the status line. Check these things:

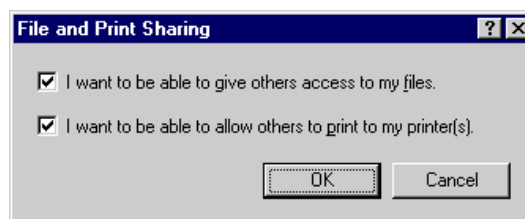
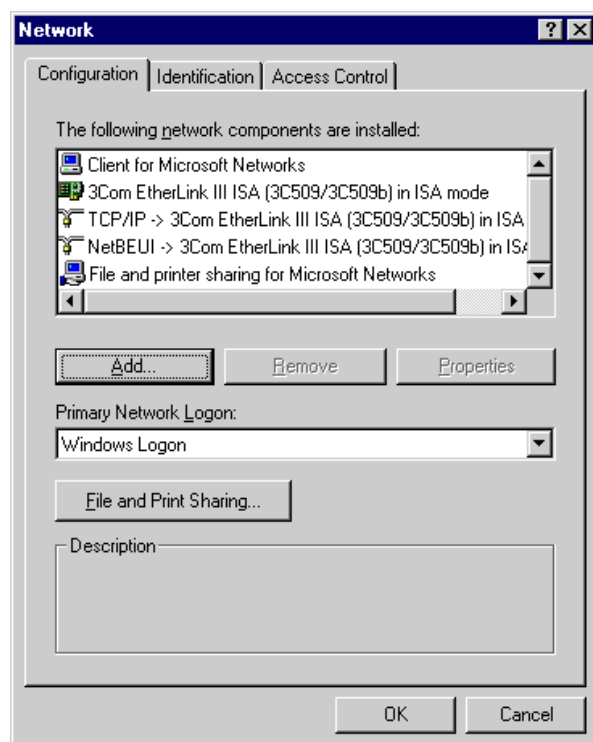
1. Is the COMn: port defined in the Control Panel / System / Device Manager/Ports (COM & LPT)? If it is not, you must install the driver for the port and specify the correct IRQ and I/O address.
2. If the driver is installed, check the "general" tab on the port and see if it says "this device is working properly". If not, check board, IRQ and I/O address. Try changing the IRQ on the board and within Windows, often there is a conflict.
3. If you are using a proprietary I/O board, be sure to install the drivers that came with the board. This will probably install the COM-driver in Windows. Otherwise, you must do it yourself afterwards.
4. Check the DIP switches or jumper settings so that the I/O board's IRQ and I/O address is correct. Some boards have more settings, like baud rate frequency selector that should be 1.843 MHz as a default.
5. If everything with the Windows driver seems OK, then double-check that you have the correct comport setting and baud rate in the Prilog comport setup screen.
6. Start another comport (terminal) program, like HyperTerminal that comes with Windows. Connect to your com-port with the correct baud rate and see if you can see any data. HyperTerminal will print everything it receives. If HyperTerminal cannot open the port, something still is wrong with the driver. If you receive data OK, then Prilog should also be capable of opening the port. Re-start your computer and start Prilog again. Important: A program will "LOCK" a comport if it run. If an error occurs, it will continue to be "locked" until you restart your computer. If you receive "garbage", the port works, but you probably have the wrong baud rate.
7. An MS-DOS program which uses the comport is detected by windows. Windows denies access to that comport when the program uses it. However, the comport may still be occupied after the MS-DOS program has terminated if the DOS-window is still open. Therefore: Close all DOS windows. This may release the comport you are trying to open.

20.2. Prilog will open the comport, but won't print any data on the screen

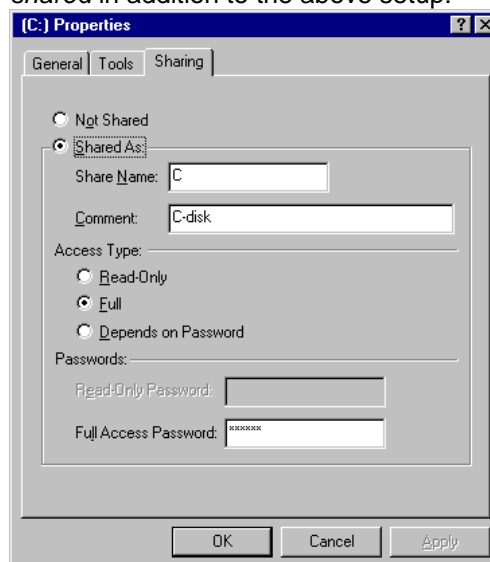
1. First, Prilog needs to detect a line-termination that is a CR, LF or CRLF characters to print it on the screen. This means that Prilog may actually be receiving data without displaying them on the screen. To check if this is the case, Chose Information / Communication status and see if the buffer-sizes changes. If they do, Prilog is receiving something.
2. Check the baud rate (bit rate) setting. If the baud rate is wrong, Prilog probably receives "garbage" characters and no line-termination and thus not prints anything on the screen. You may use the built-in Windows program "HyperTerminal" to check this. Stop Prilog (terminate program) and start HyperTerminal and connect it to the same comport. If you receive "garbage" on the screen, you have the wrong baud rate. Change the baud-rate until you find the correct one and use that in Prilog.
3. The other end does not send any data! Check the wiring. Check the other end that it is actually sending something. When Prilog starts, it sends an X-ON character that means "Transmit On" or "Start transmitting". This is ASCII character 17. (When Prilog ends, it sends X-OFF, ASCII 19.)
4. The interrupt setting in hardware or operating system could be incorrect. Check that the port uses an available interrupt and that Windows uses this. Check the hardware manual for details on the comport. Check the "system properties/device manager" under Windows. Note the following problem-situation: If you have installed add-on cards with comport(s) to use with Prilog, the card may be delivered with standard IRQ settings, for example 4 for COM3 and 3 for COM4. In theory Windows manages that the COM1 also uses IRQ4 and COM2 uses IRQ3 at the same time. However, the add-on card may not support this electrically on the machine bus (the IRQ lines don't go tri-state when inactive). Therefore, it is often necessary for add-on boards to have their own dedicated IRQ's.

20.3. Troubleshooting networking and TCP/IP printing

1. First of all the network drivers in the computer must be installed. It is necessary with a network card with a network driver and the TCP/IP protocol. It is not the scope of this document to describe the full setup of this. However, this is an example screenshot of the Windows 95 network setting:



The file and print sharing is not necessary if no Prilog reader / viewer is going to connect to the logger via the network. If, however, this is the case, the drive or directory where Prilog resides, need also be *shared* in addition to the above setup:



The sharing dialog-box may be invoked in the Windows explorer (file/directory-explorer) and pressing the right mouse-button and choosing sharing on the popup-menu.

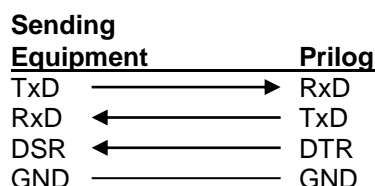
Note that a network administrator should consider the security and safety regarding network before implementing this. As no system is 100% secure, especially on a network, a password and the latest software-release of the operating system from Microsoft should at least protect the resources.

21. Serial communication specifications

This is the pin layout of the common RS232 port:

Pin name	25-pin	9-pin	Direction ⁶	Full name	Remarks
TxD	2	3	Out	Transmit Data	Data
RxD	3	2	In	Receive Data	Data
RTS	4	7	Out	Request To Send	Handshaking
CTS	5	8	In	Clear To Send	Handshaking
DTR	20	4	Out	Data Terminal Ready	Status
DSR	6	6	In	Data Set Ready	Status
RI	22	9	In	Ring Indicator	Status
DCD	8	1	In	Data Carrier Detect	Status
GND	7	5	-	Signal ground	Reference level
PG	1	-	-	Protective ground	Don't use this one as signal ground.

RS232C cabling varies somewhat because there are different needs for different equipment. Here is a basic cable that would work in most cases. The arrow indicates the signal's direction.



These are the recommended settings for I/O address and IRQ for each COM port. It is just an example. Your computer may be set up differently. You can observe the current IRQ usage under Control Panel/System/Device Manager and double-click on "Computer" and observe the "Interrupt request (IRQ)" list.

Comment	Com port	Address	IRQ
Windows mouse?	COM1	3F8	4
Prilog channel	COM2	2F8	3
Prilog channel	COM3	3E8	5 (may be used for printer)
Prilog channel	COM4	2E8	7 (may be used for printer)
Prilog channel	COM5	2F0	10
Prilog channel	COM6	3E0	11
Prilog channel	COM7	2E0	12

NOTE: You should ONLY choose IRQ's from the above table if you do not explicitly know that an interrupt is free. This is due to that some hardware is bound to interrupts for historical reasons and cannot be changed, nor inspected (sometimes). Different Prilog sessions CAN share IRQ together if the hardware supports it electrically. This is because the Windows serial communications driver supports IRQ sharing in software. Checking which port originated the interrupt when the interrupt occurs does this. Not all serial-boards support sharing IRQ's with other serial-boards, unfortunately. Prilog will also work with proprietary a multi-serial board that has a Windows driver written for it. This allows for one Prilog computer to log several hundred channels simultaneously theoretically. The vendor can give advice on this.

⁶ As seen from the PC (Data Terminal Equipment)

22. Frequently Asked Questions (FAQ)

Q. Which types of data can Prilog receive?

A. Any single-byte ASCII text data. Binary values can be discarded or replaced with ASCII. Unicode is not supported yet.

Q. I get the error message “Illegal record size: 0”. What’s wrong?

A. There is a bug in some versions that overwrites the PrilogRoot/DatFile/Recordsize XML node in the PRILOG.XML file with the value 0 instead of the default 140. This occurs if one starts a Prilog Writer which is already running. To fix this error, stop the Prilog Writer, open PRILOG.XML in notepad, search for ‘RecordSize’ and set its value to 140 (or 10 more than your linelength). Save the file and start Prilog Writer. This bug is avoided by not starting the Prilog Writer twice. This bug is fixed, and a free upgrade is recommended.

Q. How can Prilog send data?

A. Two ways. Adding a “Write Line” action filter or adding a “Write Line” scheduler action.

Q. Can Prilog send NMEA data?

A. Yes, the “Write Line” filter can be used in conjunction with the LX replacement feature. This way Prilog can send NMEA lines that contains values that are dynamically read from other source and is custom formatted. A NMEA checksum can even be added.

Q. I’m having license problems

A. Prilog versions from 5.00 to approx. 6.45 uses hardware key HASP. Versions above uses software license.

If you’re having HASP problems, you most probably need to reinstall the HASP driver with a current one for your new computer or new Windows version. The source for these drivers is <http://www.aladdin.com>

If you are having software license problems, then please check the Help / License information. For versions earlier than 6.97, you must have administrator rights to install and use Prilog. Later versions require you to be administrator only when installing the license code (because it is used by the whole computer). Another problem could be that a network card has been added or removed. Prilog binds to a network card with a MAC address (also called a hardware or ethernet address). If anything changes with this, Prilog won’t recognize the license code. A new code must be obtained from the vendor, by simply stating the reason.

