

Running LogPlot with Command Line Parameters

Introduction

LogPlot supports command line processing and command line batch file processing. The implications of this range from simple loading of a data file upon program launch, to automatic display of the data file and establishment of program settings and/or compiling of data files and display of a log.

These command line tools are used for a single data file only.

! See also Compiling a Log Batch if you wish to compile multiple data files, within the program itself, using menu settings.

Option 1: Load a Data File at Program Startup

If you want a particular data file to be loaded when LogPlot is started, simply list the name of the LogPlot data file (.ldat8), including its full path, after the program name on the command line.

For example, to start up LogPlot and load the data file named "project_a.ldat8", you would use the following command line:

```
LogPlot8.exe C:\Users\username\Documents\project_a\project_a.ldat8
```

If there are any space characters in the path, such as for the directory "Project Data," be sure to enclose the information in quotation marks:

```
LogPlot8.exe "C:\Users\username\Documents\Project Data\DH-01.ldat8"
```

Option 2: Load a Data File and Establish Settings at Program Startup

If you want to automate the steps of starting the LogPlot program, automatically loading a particular data file AND establishing a number of program settings, you can do so by creating a "batch" file. This file contains a list of batch commands that declare file names and program settings to be honored when the program starts.

To process a batch file when the LogPlot program is started, you list the "BATCH=" command and then the name of the LogPlot Batch file after the program name on the command line.

```
LogPlot8.exe "BATCH=C:\Users\username\Documents\Project Data\batch_a.txt"
```

See the following topics for details about the batch command structure.

Option 3: Load a Data File, Establish Settings at Program Startup, AND Compile and Display the Log

If you want to automate the steps of starting the LogPlot program, automatically loading a particular data file and establishing a number of program settings, AND then compiling the data and displaying the log, this is also done by creating a "batch" file. This file contains a list of batch commands that declare file names and program settings to be honored upon program startup.

The difference between this batch and that in Option 2, above, is that this file contains the "DISPLAY" command.

To process a batch file when the LogPlot program is started, you list the "BATCH=" command and then the name of the LogPlot Batch file after the program name on the command line.

```
LogPlot8.exe "BATCH=C:\Users\username\Documents\Project Data\batch_a.txt"
```

See the following topics for details about the batch file structure.

Batch File Requirements

If you wish to run LogPlot in batch mode, you must list the available batch commands in a text file. This file could be created using any word processor or text editor that is capable of saving files in a text-only format.

Some commands are required, and some are optional. Any of the optional commands can be omitted if you want the program to use the current or default program settings (as stored in LogPlot's configuration file, resulting from previous uses of the program). This would include the name of the log design, the compile scale, log scale units, header status, and pattern density.

Commands are not case-sensitive. Commands can be listed in any order.

Batch File Commands

Following is a summary of the available commands and their syntax.

DATAFILE

REQUIRED

Format

```
DATAFILE=<name of data file to be loaded into LogPlot>
```

You may list the entire path of the data file, and the program accepts long file names with or without quotes.

Example:

```
DATAFILE=C:\Users\username\Documents\project_a\project_a.ldat8
```

Result: LogPlot will load the listed data file at startup.

FORMATFILE

Optional.

Format

```
FORMATFILE=<name of log design file to use>
```

You may list the entire path of the log design file, and the program accepts long file names with or without quotes. If none is listed in the batch file, the program will use the most recently-named design file, stored as default in the program's configuration file.

Example:

```
FORMATFILE=C:\Users\username\Documents\project_a\project_a.ldfx8
```

Result: LogPlot will set the listed log design file as default until a different ".ldfx8" file is declared in the program or batch.

SCALE for Log

Optional.

Format

```
SCALE=<real number>
```

This setting declares the vertical plotting scale for the log, expressed as a real number. If no scale is listed in the batch file, the program will use the default setting, stored in the program's configuration file.

Example:

```
SCALE=15
```

Result: LogPlot will set the declared vertical scale as default until a different scale is declared in the program or batch.

INCHES

Optional.

Format

```
INCHES=[YyTt or NnFf]
```

This setting sets the scale units to inches (if the parameter is "Y", "y", "T", or "t") or to centimeters (if the parameter is "N", "n", "F", or "f"). If no units are declared in the batch file, the program will use the default setting, stored in the program's configuration file.

Example:

```
INCHES=T
```

Result: LogPlot will set the declared scaling units as default until different units are declared in the program or batch.

CONTINUOUS

Optional.

Format

```
CONTINUOUS=[YyTt or NnFf]
```

This setting sets the "continuous" log setting to "Yes" (if the parameter is "Y", "y", "T", or "t") or to "No" (if the parameter is "N", "n", "F", or "f"). If no setting is declared in the batch file, the program will use the default setting. If set to Continuous, a header is plotted at the top of the log, and a footer at the end of the log only, with middle pages having no header or footer.

Example:

```
CONTINUOUS=T
```

Result: LogPlot will establish the declared header-on-every-page setting as default until a different setting is declared in the program or batch.

Note: This command is equivalent to the "HEADER=" command in previous versions of LogPlot.

HEADERS

Optional

Format

```
P1_HDR=[0 or 1 or 2]
P_HDR=[0 or 1 or 2]
PN_HDR=[0 or 1 or 2]
```

Use these commands to establish which header is to be used on the first page (P1_HDR), the middle pages (P_HDR), and the last page (PN_HDR) of the log. A setting of "0" means no header, "1" is header 1, "2" is header 2. Note that if you are compiling a continuous log, above, you can define only the first page header.

Example

```
P1_HDR=1
P_HDR=2
PN_HDR=2
```

Result: LogPlot will compile the log such that header 1 is on the first page, and header 2 is on the middle and last page.

FOOTERS

Optional

Format

```
P1_FTR=[0 or 1 or 2]
P_FTR=[0 or 1 or 2]
PN_FTR=[0 or 1 or 2]
```

Use these commands to establish which footer is to be used on the first page (P1_FTR), the middle pages (P_FTR), and the last page (PN_FTR) of the log. A setting of "0" means no footer, "1" is footer 1, "2" is footer 2. Note that if you are compiling a continuous log, above, you can define only the last page footer.

Example

```
P1_FTR=1
P_FTR=2
PN_FTR=2
```

Result: LogPlot will compile the log such that footer 1 is on the first page, and footer 2 is on the middle and last page.

MARGIN

Optional.

Format

```
MARGIN=<real number>
```

This setting sets the margin below the header/above the footer setting to the real number value you declare. The value represents either inches or centimeters, depending on how you have configured the program (see INCHES, above). If no setting is declared in the batch file, the program will use the default setting.

Example:

```
MARGIN=0.25
```

Result: LogPlot will establish the declared header/footer margin as default until a different setting is declared in the program or batch.

TOP

Optional.

Format

```
TOP=<real number>
```

This setting establishes the top of the log, in real number depth or elevation units. If no setting is declared in the batch file, the program will use the top-of-log declaration in the data file's BH Info worksheet.

Example:

```
TOP=0
```

Result: LogPlot will establish the declared top-of-log depth or elevation as default until a different setting is declared in the program, batch, or data file's Setup tab.

BOTTOM

Optional.

Format

```
BOTTOM=<real number>
```

This setting establishes the base of the log, in real number depth or elevation units. The base of the log must be greater in value than the top of the log. If no setting is declared in the batch file, the program will use the base-of-log declaration in the data file's BH Info worksheet.

Example:

```
BOTTOM=100
```

Result: LogPlot will establish the declared bottom-of-log depth or elevation as default until a different setting is declared in the program, batch, or data file's Setup tab.

DATE FORMAT

Optional

Format

```
DATE_TYPE=[0 or 1 or 2]  
USE_TIME=[YyTt or NnFf]  
DATE=<date>  
DATE_START=<starting date>  
DATE_END=<ending date>
```

This command is used to activate a date filter for any time-based data you have entered. Setting the DATE_TYPE to "0" means no date filter, "1" means a single date (use DATE= to define that date), and "2" means a date range (use DATE_START= and DATE_END= to define the beginning and ending dates). Setting USE_TIME to true (if the parameter is "Y", "y", "T", or "t") indicates that the dates you specify will include time values.

Example

```
DATE_TYPE=2  
USE_TIME=N  
DATE_START=01/01/2020  
DATE_END=03/31/2020
```

Result: LogPlot will plot only the date measurements from Jan 1 2020 through Mar 31, 2020.

DISPLAY

Optional.

If this command is present it compiles the declared data file and displays the completed log. The program will use the parameters declared in the batch file for scaling, log design file name, and other settings. If these are not present it will use the most recently used settings (stored in the program's configuration file).

SAVEFILE

Optional.

Format

```
SAVEFILE=<filename>
```

This command tells the program to compile the declared data file and to store the completed log plot on disk, under the ".lpt8" file name that you specify. You may list the entire path for the lpt8 file, and the program accepts long file names with or without quotes. When used with the DISPLAY command the named file will be loaded into the LogPlot viewer. Without the DISPLAY command LogPlot will exit once the lpt8 file has been saved.

Example:

```
SAVEFILE=C:\Users\username\Documents\project_a\bh1.lpt8
```

PDF Export

Optional.

Format

```
PDF=<filename>
```

This command tells LogPlot to compile the declared data file and to output the completed plot to PDF format, under the file name that you specify.

Example:

```
PDF=C:\Users\username\Documents\project_a\bh1.pdf
```


HTMLFILE Export

Optional.

Format

```
HTMLFILE=<filename>
```

This command tells the program to compile the declared data file and to export the completed log plot to HTML format, under the HTML file name that you specify. (As discussed in the Log View section of the LogPlot help, this export involves saving JPEG image(s) of each log page and then listing these in a HTML table.) You may list the entire path for the output file, and the program accepts long file names with or without quotes. The JPEG images will be stored with the same file name as the one you declare for the HTML file, with a "1", "2", etc. appended.

When used with the DISPLAY command (below) the HTML file will be loaded into your default browser, and the LogPlot program will be closed. Without the DISPLAY command LogPlot will exit once the HTML file has been saved, but the log will not be displayed in the browser.

Example:

```
HTMLFILE=C:\Users\username\Documents\project_a\bh1.html
```

Accessory Commands:

HTMLDISPLAY

Format

```
HTMLDISPLAY=[YyTt or NnFf]
```

This defines whether the HTML log is to be loaded into a browser on completion.

Example:

```
HTMLDISPLAY=T
```

JPEG Export

Optional.

Format

```
JPEG=<filename>
```

This command tells the program to compile the declared data file and to export the completed log plot to JPEG format, under the JPG file name that you specify.

Example:

```
JPEG=C:\Users\username\Documents\project_a\bh1.jpg
```

Accessory Commands:

JPEGFORMAT

Format

```
JPEGFORMAT=[0 or 1 or 2]
```

This sets the format of the JPEG image output. "0" represents black and white, "1" represents grayscale, "2" represents true color.

BITMAP Export

Optional.

Format

```
BITMAP=<filename>
```

This command tells the program to compile the declared data file and to export the completed log plot to a Bitmap format, under the BMP file name that you specify.

Example:

```
BITMAP=C:\Users\username\Documents\project_a\bh1.bmp
```

Accessory Commands:

BMPFORMAT

Format

```
BMPFORMAT=<integer>
```

This sets the color depth of the BMP image output. Enter 5 for 16-bit color, 6 for 24-bit color, or 7 for 32-bit color. If not listed, the program will use the default setting from the last time the program was run.

PNG Export

Optional.

Format

```
PNG=<filename>
```

This command tells the program to compile the declared data file and to export the completed log plot to a PNG raster format, under the PNG file name that you specify.

Example:

```
PNG=C:\Users\username\Documents\project_a\bh1.png
```

TIFF Export

Optional.

Format

```
TIFF=<filename>
```

This command tells the program to compile the declared data file and to export the completed log plot to a TIFF raster format, under the TIF file name that you specify.

Example

```
TIFF=C:\Users\username\Documents\project_a\bh1.tif
```

Accessory Commands

TIFFFORMAT

Format

```
TIFFFORMAT=<integer>
```

This sets the color depth of the TIFF image output. Enter "0" for black & white, "1" for 16-color grayscale, "2" for 216-color grayscale, "3" for 16-color, "4" for 256 color, and "5" for true color. If not listed, the program will use the default setting from the last time the program was run.

PRINT

Optional.

This command sends the compiled log to the current printer. Make sure that your printer page size matches the LogDesign file, otherwise your batch job will be stopped by a page size warning screen.

DEBUG Log

Optional.

Format

```
DEBUG=<output file name>
```

This command creates a text file that shows how the batch file was interpreted and what the defaults settings were.

Example:

```
DEBUG=C:\Users\username\Documents\project_a\debug.txt
```

Result: LogPlot will perform the listed operations in the batch file, and will record a summary of actions and settings as text under the file name listed above.