

Lightning PDF
User Guide
v3.0



Copyright

This publication contains information that is protected by copyright. No part of it may be reproduced in any form or by any means or adapted in any way without the prior written permission from the copyright holders.

This publication is provided for information purposes only. The manufacturer makes no representation or warranties with respect to the contents or use of this manual and specifically disclaims any expressed or implied warranties of merchantability or fitness for any particular purpose. The user will assume the entire risk of the use or the results of the use of this document. Further the manufacturer reserves the right to revise this document and make changes to its contents at any time, without obligation to notify any person or entity of such revisions or changes.

Copyright © Hamillroad Software 2007-2017.

All Rights Reserved.

Trademarks and Credits

Microsoft® Windows® NT, Windows® 2000, Windows® XP, Windows Vista®, Windows® 7, Windows® 8, Windows® 10 and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Intel® Core™ Duo, Intel® Core™2 Duo, Intel® Core™ i3 / i5 / i7 are registered trademarks of Intel Corporation. Apple® Mac OS® are registered trademarks of Apple Computers. Harlequin™ is a registered trademark of Global Graphics Software Ltd. Other trademarks and registered trademarks of products appearing in this manual are the properties of their respective holders.

Contents

| | |
|--|-----------|
| Introduction | 7 |
| About Lightning PDF..... | 7 |
| Conventions used in this reference manual | 8 |
| Differences between Mac and PC versions..... | 8 |
| Type of Files output by Lightning PDF..... | 8 |
| System Requirements | 9 |
| Operating system: | 9 |
| Harlequin RIP:..... | 9 |
| Installing and Activating Lightning PDF | 10 |
| Installing Lightning PDF | 10 |
| Activating Lightning PDF..... | 11 |
| Automatically..... | 11 |
| Manually..... | 12 |
| Configuring Lightning PDF..... | 13 |
| General | 13 |
| Dialog Layout..... | 14 |
| Options..... | 15 |
| Output Folder..... | 15 |
| Path includes..... | 15 |
| File name generation..... | 16 |
| Template Name Generation | 17 |
| Conventional Name Generation | 18 |
| Stem (edit box) | 18 |
| Use jobname as stem..... | 18 |
| Delete page number | 19 |

| | |
|------------------------------------|-----------|
| [Use] Jobname unchanged | 19 |
| Unique filenames..... | 19 |
| [Use] Filename as jobname..... | 20 |
| Delete filename suffix..... | 20 |
| Prefix color with “_” | 21 |
| Use 8.3 filenames | 21 |
| Suffix (edit box) | 21 |
| Performance..... | 22 |
| Threads | 22 |
| Parallel RIP..... | 24 |
| Output Format..... | 25 |
| Data [Format] | 25 |
| Compression | 25 |
| Pad to 32/64-bits (mono only)..... | 28 |
| Use Predictor | 28 |
| PDF Details..... | 29 |
| [Page] Style..... | 29 |
| [PDF] Format..... | 29 |
| PDF/X1a [Output Intent] | 29 |
| File Permissions | 30 |
| Unlimited (read and write)..... | 30 |
| PDF Info (Meta) Data | 31 |
| Update PDF creation date | 31 |
| Update PDF modification date..... | 31 |
| Mark raster as Trapped | 31 |
| Post-processing | 32 |
| Create Window | 32 |

| | |
|-----------------------------------|-----------|
| Command (File) | 32 |
| Command (Job) | 33 |
| Substitution Code Variables | 33 |
| Template Variables | 35 |
| Standard Variables | 35 |
| <colorant> variable | 35 |
| <colorname> variable | 35 |
| <date> variable | 35 |
| <exposure> variable | 35 |
| <job#> variable | 35 |
| <jobname> variable | 36 |
| <jobname1> variable | 36 |
| <page#> variable | 36 |
| <prefix> variable | 36 |
| <prefixonly> variable | 36 |
| <realpage#> variable | 37 |
| <time> variable | 37 |
| <unique> variable | 37 |
| <xres> variable | 37 |
| <yres> variable | 37 |
| Special Variables | 38 |
| <dot> variable | 38 |
| <dos> variable | 38 |
| <mac> variable | 39 |
| <macosx> variable | 39 |
| <unix> variable | 39 |
| <win32> variable | 39 |
| Extended Variables | 40 |

| | |
|--|-----------|
| <colorcode> variable..... | 40 |
| <filename{XYZ}> variable..... | 40 |
| <filename1{XYZ}> variable..... | 40 |
| <page#fromjobcomments> variable..... | 41 |
| <page#fromfilenameXYZ{+n}> variable..... | 41 |
| <page#fromjobnameXYZ{+n}> variable | 41 |
| Variable Modifiers..... | 42 |
| String modifier..... | 42 |
| Number modifier..... | 42 |
| Extended Variable Modifiers..... | 43 |
| [123-456] modifier..... | 43 |
| [123&456] modifier | 43 |
| [abc/def] modifier | 43 |
| Troubleshooting | 44 |
| Glossary..... | 47 |
| Contact..... | 48 |

Introduction

About Lightning PDF


Lightning PDF is a high performance PDF output plugin for the Harlequin RIP that is an alternative to the GlobalGraphics PDF Raster plugin. As well as providing a significant number of additional features, its main benefit is its lightning fast performance; users should see anything up to a 5x (or more) overall improvement in performance and throughput when it is used, depending on output mode, compression mode, resolution, screening, job content and platform.

The high performance that Lightning PDF achieves is obtained through a unique method for implementing CCITT and Flate compression, along with a significant number of code optimisation techniques that have been applied to all the various compression modes. In addition, Hamillroad engineers have built in the use of advanced multi-threading software techniques which when used with the latest multi-core CPUs, such as the Intel Core 2 Duo or Intel Core i3 / i5 / i7, provide a near linear additional increase in performance.

Support for the latest Harlequin Multi RIP (HMR) 10 and 11 is also included, both 32-bit and 64-bit, with the 64-bit version having been fully optimised to take advantage of 64-bit CPUs.

The results of this are the creation of PDF files at extremely high speed, allowing for greatly increased throughput.

Conventions used in this reference manual

- Instructions may be indicated by a numbered paragraph or a (•) bullet point.
- The  icon is used to identify important information.
- Screen Shots are always laid out with the Windows version preceding the Macintosh version.

Differences between Mac and PC versions

While every effort has been made to keep the PC and Mac versions the same, there are some minor differences that cannot be avoided.

This manual will always explain the procedure for both platforms where they differ from the above.

Type of Files output by Lightning PDF

Lightning PDF produces PDF (raster) files which are compatible with the Adobe PDF Reference Third Edition Version 1.4 specification. In addition to PDF 1.4, files can be produced that conform to the PDF/X1a:2003 (ISO 15930-1:2001) standard.

Each PDF file contains either a single page or multiple pages, depending on what options have been chosen.

Each page consists of a single (raster) image, containing either 1-bit screened image data or 8-bit contone image data, that uses a color space of DeviceGray, DeviceCMYK or DeviceN, depending on if Gray, CMYK or spot colors are used in the page.

To minimise the size of the PDF file, the image data is compressed with one of a number of compression algorithms; CCITT Fax G4/G3, Flate (ZIP), LZW or Packbits.

System Requirements

Operating system:

| | |
|-----|--|
| PC | Windows XP / Vista / 7 / 8 / 10 or later |
| Mac | OS X - 10.2 to 10.10 or later |

Hardware:

| | |
|-------------|--|
| PC | Pentium-4 2GHz Processor, or Core 2 Duo 1.66GHz Processor, or Core i3 / i5 / i7 2GHz Processor 512Mb RAM 10Gb free disk space CD-ROM Drive |
| Mac (Intel) | Core 2 Duo 1.66GHz Processor, or Core i3 / i5 / i7 2GHz Processor 512Mb RAM 10Gb free disk space CD-ROM Drive |

Harlequin RIP:



A Harlequin RIP version 5.0 or later is required.



PowerPC Mac RIPs are not supported.

Installing and Activating Lightning PDF

Installing Lightning PDF

Installing Lightning PDF on your Harlequin RIP is as simple as ripping a Postscript job.

- From the first menu on the RIP's screen, select the **Print File** option.



The menu's name will vary depending on the RIP vendor.

- If your Harlequin RIP is on a PC, then depending on your RIP version print one of:

Install Lightning PDF (Win-Hqn9).ps

Install Lightning PDF (Win-HMR10-32).ps

Install Lightning PDF (Win-HMR10-64).ps

Install Lightning PDF (Win-HMR11-64).ps

- If your Harlequin RIP is on a Mac (OSX), then depending on your RIP version print one of:

Install Lightning PDF (Mac-OSX).ps

- Once the job is complete you must reboot the RIP.

Lightning PDF can now be accessed from the Edit Page Setup dialog when creating new Page Setups, although jobs cannot be output with it until it is activated.

Activating Lightning PDF

Automatically

Activating Lightning PDF on your Harlequin RIP is as simple as ripping a Postscript job.

- From the first menu on the RIP's screen, select the **Print File** option.



The menu's name will vary depending on the RIP vendor.

- Print the Lightning PDF Activation File you have been supplied with (to a Page Setup that uses Lightning PDF):

Lightning PDF Activation - 12345-67.ps


Where '12345-67' is the security number of the Harlequin RIP you want to use Lightning PDF on.

- Once the job is complete you must output the page it produces (in the Output Controller / Monitor).

Once the job is output, you will see in the RIP's Monitor window the following message:

! Lightning Plugin (PDF)

MESSAGE: Activated Lightning PDF.

and a flashing  icon in the Output Controller / Monitor. This indicates that Lightning PDF has successfully been activated.

If there is a problem with the Activation Key, then you will see in the RIP's Monitor window the following message:

! Lightning Plugin (PDF)

ERROR: Invalid Product Key.

and a flashing  icon in the Output Controller / Monitor.

Manually

To activate Lightning PDF on your Harlequin RIP you need to enter the 25 digit Activation Key (including '-'s):

- From the first menu on the RIP's screen, select the **Device Manager** option.



The menu's name will vary depending on the RIP vendor.

-
- From the "Plugin" popup, select **LightningHIFF.i32 (Hqn89 & HMR10/32)**, **LightningHIFF.i64 (HMR10/64 & HMR11/64)** or **LightningHIFF (Mac (OSX))**.
 - From the list below, either double-click the **Lightning PDF** entry or select the **Lightning PDF** entry and click **Edit**.
 - The **Device Manager Edit** screen will launch.
 - Enter your 25 digit Activation Key (including '-'s) into the Address field, preferably copy-n-pasting it from the file name of the supplied Activation Key Certificate.

Lightning PDF can now be accessed from the Edit Page Setup dialog when creating new Page Setups and jobs can also be output with it.

Configuring Lightning PDF

General

To use Lightning PDF you need to configure a Page Setup in your Harlequin RIP to use it:

- From the first menu on the RIP's screen, select the **Page Setup Manager** option.



The menu's name will vary depending on the RIP vendor.

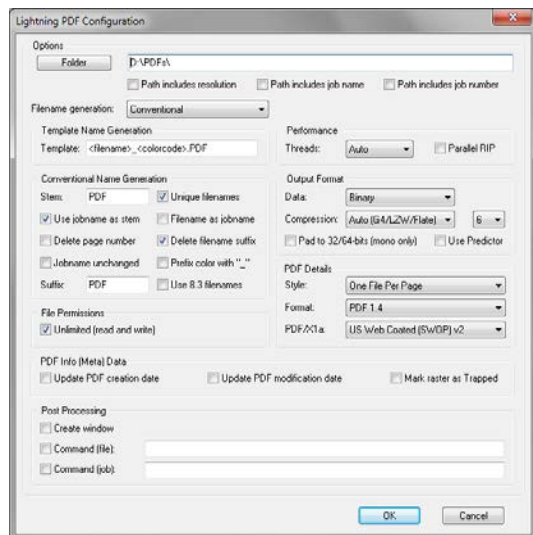
- Click on **New**.
- The **Edit Page Setup** screen will launch.
- From the Device section, select **Lightning PDF** from the **Device** popup menu.
- You may now select the usual set of options from the rest of the Page Setup dialog.

To configure Lightning PDF:

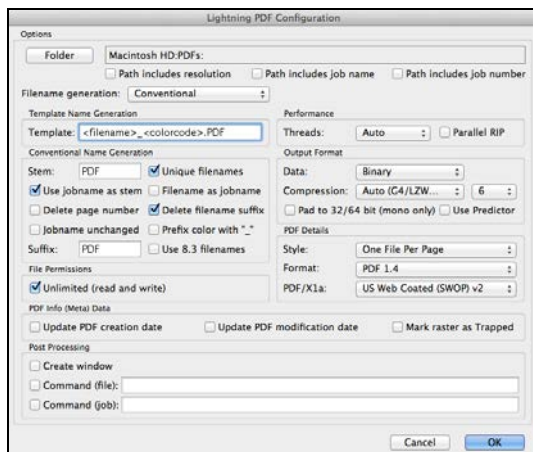
- From the **Edit Page Setup** screen, select the **Configure device** button.
- The **Lightning PDF Configuration** screen will launch.
- You may now modify the various options.

Dialog Layout

PC



Mac



Options

Output Folder

The output folder is the location where the PDF files will be output. This can be either on a local disk, or on a remote computer via a mounted share.

To select the output folder:

- Select the **Folder** button.
- The **Select Folder** screen will launch.
- Browse to the required folder and select **OK**.

Alternatively, enter the full path name of the folder in the box.

Path includes...

In addition to selecting the output folder where the PDF files will be written, the user can also select from any of three further options which augment the selected output folder:

1. Path includes resolution:

This option appends the (horizontal) device resolution of the job to the chosen Folder.

For example if the folder was "D:\PDFs\" and the device resolution was "1200" dpi then the folder used would be "D:\PDFs\1200\".

2. Path includes job name:

This option appends the job name of the job to the chosen Folder.

For example if the folder was "D:\PDFs\" and the job name was "Job1" then the folder used would be "D:\PDFs\ Job1\".

3. Path includes job number:

This option appends the (unique) job number of the job (as allocated by the Harlequin RIP) to the chosen Folder.

For example if the folder was "D:\PDFs\" and the job number was "123" then the folder used would be "D:\PDFs\123\".

The most useful of these are option 2 and 3, where all the PDF files from a given job will be output into their own individual folder.

File name generation

The file name that each PDF file is given depends on whether template base name generation or conventional name generation is chosen. The former allows for very sophisticated and precise control over exactly how the file name is composed; quite often a requisite of many workflows. The latter allows for much simpler creation of file names, following a small number of easily understandable rules.

Template Name Generation

Template name generation uses a mixture of literal text (used as is) and variables (which are replaced) to create the resulting file name.

A simple example of a template is “PDF<2unique>.pdf”; this simply creates files named as “PDF00.pdf”, “PDF01.pdf”, “PDF02.pdf”, etc... In this example, the initial “PDF” and final “.pdf” are both literal text (used as is) and the middle “<2unique>” is a variable that is replaced with a unique number that is (at least) two digits long.

All sorts of complex names can be created, and furthermore, names can include sub-folders, allowing for accurate control over where files are written. To specify a sub-folder on the PC use a “\” and on the Mac a “.”.

For example “<jobname>\<page#>_<colorname>.pdf” will put all files for a given job into it’s own sub-folder (named after the job name), naming each file with the page number (e.g. 123) and color name (e.g. “Cyan”).

Although there are two types of template naming possible, Standard and Extended (see Lightning TIFF), Lightning PDF only supports the more advanced Extended Templates. Extended Template naming uses an extended set of template variables that provide significantly more options, as well as more powerful options.

For further details on all the possible variables that templates can use and what they each do, please see the chapter on Template Variables.

Conventional Name Generation

Conventional name generation, as mentioned before, uses a small number of options to specify exactly how each PDF file name is constructed.

All conventionally named PDF files start off with a stem prefix, followed by a period "." and end with a suffix; a simple example is "1Job.pdf". In this example, the initial "1Job" is the stem and the final "pdf" is the suffix.

Stem (edit box)

A fixed stem can be chosen by entering whatever is required in the Stem edit box.

For example if "PDF" is entered (and "Unique filenames is enabled"), then the PDF files produced will be created as "PDF00.PDF", "PDF01.PDF", etc...

Use jobname as stem

Instead of using a fixed string for the prefix, when it is difficult to see which files belong to which job, the "Use jobname as stem" option when enabled replaces the fixed string with the job name.

For example if the job name was "Job", then instead of producing PDF files named "PDF00.pdf", "PDF01.pdf", etc..., the PDF files produced would be named "Job00.pdf", "Job01.pdf", etc...

However, because of the way the Harlequin RIP works, the job name as seen by Lightning PDF always contains the page number at the beginning and the color name (of the separation) at the end.

Taking our previous example, where the job name was "Job", then the job names used by the Harlequin RIP for the four separations of a single page job would be "1. Job (C)", "1. Job (M)", "1. Job (Y)" and "1. Job (K)", and the PDF files

produced would be named “1JobC00.pdf”, “1JobM00.pdf”, “1JobY00.pdf” and “1JobK00.pdf”¹.

Delete page number

If “Use jobname as stem” is enabled and the initial page number is not required, then this can be removed by enabling “Delete page number”.

Taking our previous example, the PDF files produced in this case would then be “JobC00.pdf”, “JobM00.pdf”, “JobY00.pdf” and “JobK00.pdf”.

[Use] Jobname unchanged

As mentioned previously (and see the foot note), illegal characters are always removed from file names before trying to create a file. That’s fine for job names that use single byte characters, but for job names that use double byte characters, this can remove part of a valid 2-byte character that can in turn result in an unreadable file name.

The “[Use] Jobname unchanged” option can be therefore be useful if job names are written using double byte characters. However, care should be taken when using this option, as it is possible to create file names that are invalid on the operating system and so PDF files fail to get created.

Unique filenames

In the case where we set the stem explicitly (and the PDF files were named “PDF00.pdf”, “PDF01.pdf”, etc...), we also had “Unique filenames” enabled too. Without this the PDF output would fail on the second and subsequent files, as

¹ The initial “1. ” is reduced to “1” and the final “ (C)” is reduced to “C” because illegal characters for a file system are always removed (and “:”, “ ”, “(” & “)”) are all illegal characters in file names on the PC.

they would all be named “PDF.pdf”. Therefore if using an explicit stem to name files, “Unique filenames” should always be enabled.

However, if “Use jobname as stem” is used (instead of an explicit stem), then (assuming the job names are always unique) the “00” that is always at the end of the prefix is quite unnecessary, unless you want to generate multiple revisions of the same file (for example after making changes). In this case “Unique filenames” can be disabled.

[Use] Filename as jobname

Instead of using the job name, which is not always known by the user or set sensibly in a job (for example if a customer supplies a job), it is sometimes better to be able to use the file name instead.

This provides more flexibility, since the operator can rename the file from whatever it is to a known name and the output will be guaranteed to use that name.

Enabling the “[Use] Filename as jobname” therefore replaces the job name with the file name.

Taking our previous example, if the filename was “File.pdf”, the PDF files produced in this case would be “FilepdfC00.pdf”, “FilepdfM00.pdf”, “FilepdfY00.pdf” and “FilepdfK00.pdf”.

Delete filename suffix

If “[Use] Filename as jobname” is enabled and any trailing file name extension is not required (such as “.pdf” or “.ps”), then this can be removed by enabling “Delete filename suffix”.

Taking our previous example, the PDF files produced in this case would then be “FileC00.pdf”, “FileM00.pdf”, “FileY00.pdf” and “FileK00.pdf”.

Prefix color with “_”

Sometimes for a given file name it is not clear which part of it is the job name and which part of it is the color. Although obvious for CMYK, it is not so obvious for other names such as Pantone 123 CVC (which could be C for Cyan).

If “Prefix color with ‘_’” is enabled, then the color part of the name is prefixed (and separated from the job name) by “_”.

Taking our previous example the PDF files produced in this case would then be “File_C00.pdf”, “File_M00.pdf”, “File_Y00.pdf” and “File_K00.pdf”.

Use 8.3 filenames

With the advent of long file names in Windows, this option is rarely used. However, it remains an option for backwards compatibility.

If this option is set, then the resulting file name stem is truncated to 8 characters and the resulting file name suffix is truncated to 3 characters. This therefore results in a file name that is 8.3 characters and valid on a DOS file system.

Suffix (edit box)

Finally, for a conventional name, we need to specify the suffix. This is normally “pdf” if the PDF specification is followed.

Performance

As mentioned previously, Hamillroad engineers have built in the use of advanced multi-threading software techniques which when used with the latest multi-core CPUs provide a near linear additional increase in (output) performance.

As you make use of more and more threads though, the memory requirements do go up (each thread needs its own local memory to work in) and since you are only speeding up the output times, the overall speed increase is less. There is therefore an issue of diminishing returns, where other cores might be better used to service other software (for example other modules in a workflow system). Using two threads is certainly a very good idea, as is anything up to four. Above that, it is recommended that the user considers carefully what else might be running on the computer.

If the computer on which Lightning PDF is running does not have multiple CPUs (or cores), then this option is disabled (only the “None” option is available).

If running the RIP on a workstation which is also running other workflow modules, then it might be sensible to disable this (or use a smaller number of threads than CPUs/cores), so that the other workflow modules get more CPU time to do their work.

Threads

The options available in this popup menu depend on the number of CPUs (or cores) available. If there is only 1 CPU available, then the only choice here will be “None”. If there is more than 1 CPU (or core) available, then there will be a number of options:

1. None:

This option disables the multi-threading capabilities.

2. Auto:

This option enables the multi-threading capabilities (on computers with more than one core) and uses a carefully selected number of threads based on the number of available cores. This leads to a high level of output performance, but without consuming too many resources on the latest Intel Core i7 CPUs that can have up to 12 cores (6 cores with 2 hardware threads per core to be precise). The exact formula is $\min(\max(2, N / 2), 4)$.

3. Max(N):

This option enables the multi-threading capabilities (on computers with more than one core) and uses as many threads as there are cores. This leads to the best possible output performance, but for large band sizes and/or a large number of cores can cause memory issues (requiring you to also increase your Printer Buffer and the memory available to each RIP thread).

4. N - 1:

This option enables the multi-threading capabilities (on computers with more than one core) and makes use of a single thread for output (in addition to that of the RIP). It therefore overlaps the rendering of the job (in Single mode) or the reading of the page buffer (in Multiple mode) with the output.

5. N - 2:

This option is the same as the previous one, but two threads are used for output.

6. N - #:

This option is the same as the previous one, but # threads are used for output (where $\# > 2$).

Parallel RIP

This option when enabled causes the RIP to render (and so screen) separations in parallel, whilst still producing separate PDF files. It is only used if the RIP is set to produce separations in the Page Setup; for example if the “CMYK Separations (Halftone)” (or similar) “Separations, Screening & Colour” is used.

This option is intended to be used to improve the performance of screening in version 7 of Auraia which was released at the end of 2015.

It can also be used to optimise the RIP performance when jobs contain transparency (or masked images), enabling the RIP to handle these types of job much faster (in some cases significantly faster).

Output Format

These options determine how data is stored in the PDF file.

Data [Format]

The available options for data storage are:

1. Binary:

This is the default/standard option and should rarely be changed. It produces the smallest possible file size as each byte in the image is stored 'as-is', using an 8-bit binary byte.

2. ASCII (85):

This option allows a PDF file to be stored using only 7-bit ASCII character codes, by encoding the image data in the PDF file with the ASCII 85 method. It uses less space than the ASCII (Hex) method, but decoding is slower and some applications may have issues with it.

3. ASCII (Hex):

This option allows a PDF file to be stored using only 7-bit ASCII character codes, by encoding the image data in the PDF file with the ASCII Hex method. It uses more space than the ASCII (85) method, but decoding is simpler and faster.

Compression

A large number of compression methods are available in Lightning PDF. Of these, the group of CCITT and Flate (ZIP) methods are in particular highly optimised, using unique algorithms along with a significant number of code optimisation techniques to achieve extremely fast performance.

The available options for compression are:

1. None:

This option is in general not recommended, as it performs no compression whatsoever and simply writes the image data out.

For high resolution output this can lead to extremely large PDF files on disk and in some cases might actually be slower than compressing the data and writing (a much smaller amount) to disk.

2. Auto (G4 / LZW / Flate):

This option is the default. For 1-bit / 1-channel data it uses CCITT G4 and for 8-bit or N-channel data it uses LZW. You can therefore use this setting for a Page Setup that can produce either 1-bit or 8-bit, monochrome or CMYK PDF files and it saves you from having to change this option if you change the Page Setup from generating 1-bit to 8-bit data.

Note that if "PDF/X 1a:2003" files are being produced, then this option automatically switches from using LZW to Flate (ZIP), since LZW is not allowed in the ISO 15930-1:2001 standard.

3. CCITT G4 / CCITT Fax Group 4:

For 1-bit data this is the recommended option to use. Especially in Lightning PDF, where some of the highest compression ratios are achieved with extremely fast performance.

4. CCITT G3 (2D) / CCITT Fax Group 3-2D:

This option is similar to CCITT G4, but produces slightly larger files. It only makes sense to use if the target application needs this type of compression.

5. CCITT G3 (1D) / CCITT Fax Group 3-1D:

This option is similar to CCITT G3 (2D), but produces slightly larger PDF files. It only makes sense to use if the target application needs this type of compression.

6. Flate (ZIP):

For 8-bit or N-channel data this is a good option to use. It produces files that are very highly compressed (more so than LZW), but does take time to do so (being slower than LZW).

7. LZW:

For 8-bit data this is the recommended option to use. It produces files that are nicely compressed without taking too much time to do so. For 1-bit data, it is a good option to use instead of CCITT Group 4.

8. Packbits:

This option is in general not recommended, as it produces very large files, often 100s of megabytes.

Although it is very fast at compressing files, it typically does not compress files that much (other than areas of white space or areas of a flat tint) and so can lead to extremely large PDF files on disk. In some cases it might actually be slower than compressing the data and writing a smaller amount to disk using LZW or another option.

The Flate option mentioned above can further be controlled by changing the compression level from 1 (fastest, largest) to 12 (slowest, smallest). Levels 1 to 9 correspond to that of the public domain gzip software (but produce slightly smaller files, up to 30% faster). Levels 10 to 12 achieve slightly higher compression levels, but can take considerably longer; levels 11 and 12 are not recommended as they are slow.

Pad to 32/64-bits (mono only)

This option is another one of those that is rarely used, but included for compatibility. The standard option here is to generate PDF files “of the correct expected width”; that is this option should usually be disabled and it should be rarely enabled, unless advised to do so by an ‘expert’.

If it is enabled, it should not normally be a problem, as all it will do is result in an image that is slightly wider than anticipated. If that image is just being output, then it will (should) not cause any problems. However, if the image is (for example) rotated and stitched together with another, then the extra white space (up to 31/63 pixels) on the right hand side could lead to problems with ‘gaps’ appearing.

Use Predictor

This option is one that can sometimes reduce the size of the compressed file (by anything up to 20%). It does so by differencing each pixel of a given color component with the pixel (of the same color component) to the left; that is it subtracts the pixel on the left from the current pixel, before then compressing the result. In some cases depending on the type of image produced it can actually make the resulting file larger, so a few tests should be done to see if it is beneficial to the type of image being produced.

PDF Details

[Page] Style

By default, Lightning PDF produces PDF files with a single page in each file. This is fine if just dealing with single page PDFs. However, if multi-page PDFs are being RIPped, for example which will then be taken back into a workflow and imposed, then all the pages need to be in a single file.

The available options for Style are:

1. **One File Per Page:**

This is the default/standard option and produces one file per page; a 64 page PDF job will therefore produce 64 raster PDF files.

2. **One File Per Job:**

This option allows all the pages in a job to be stored in a single file; a 64 page (single file) PDF job will therefore produce 1 raster PDF file.

[PDF] Format

By default, Lightning PDF produces PDF files that are compatible with the PDF version 1.4 specification. If jobs are to be RIPped for a system that only accepts PDF/X 1a:2003 files, then this option should be used instead.

PDF/X1a [Output Intent]

This option is only used when PDF/X 1a:2003 files are produced and specifies the extra OutputIntents information required by that format. See the PDF Reference sixth edition v1.7, Section 10.10.4, or the ISO 15930-1:2001 standard for further details.

File Permissions

By default, Lightning PDF produces PDF files with permissions set to that of the current user (and their group). This typically allows the current user (and members of their group) to read or write (delete) the file, but not anyone else.

Unlimited (read and write)

If unlimited file permissions are required, then this option can be enabled. This can be useful if the RIP is running on a mixed network; for example if the RIP is on a PC but all the users are on Macs. This simplifies the setup of the system so that you don't have to create a common user/group between the PC and Macs.

PDF Info (Meta) Data

Update PDF creation date

This option when enabled updates (or sets) the “CreationDate” entry in the Info dictionary.

If this option is not enabled, a PDF file is the source of the job being processed and the input PDF file itself contains a “CreationDate” entry in its Info dictionary, then this entry is copied from the input PDF file to the output PDF file.

See the PDF Reference, sixth edition v1.7, Section 10.2.1.

Update PDF modification date

This option when enabled updates (or sets) the “ModDate” entry in the Info dictionary.

If this option is not enabled, a PDF file is the source of the job being processed and the input PDF file itself contains a “ModDate” entry in its Info dictionary, then this entry is copied from the input PDF file to the output PDF file.

See the PDF Reference, sixth edition v1.7, Section 10.2.1.

Mark raster as Trapped

This option when enabled sets the “Trapped” entry in the Info dictionary to “True”. It should be enabled if using TrapPro in the Page Setup to automatically trap the page, so that subsequent software knows that the pages have already been trapped and so this does not need doing again.

See the PDF Reference, sixth edition v1.7, Section 10.2.1.

Post-processing

When integrating a Harlequin RIP into a workflow, it is often necessary to perform an action on the PDF files after they have been output - for example one might want to notify another part of the system that all the PDF files from a job have arrived, or one might want to backup and archive all the PDF files to DVD, or one might want to print all the pages of a job to a proofer, etc...

This is where the post-processing in Lightning PDF comes in to its own. You have the option to perform a post-process operation both after each file has been output, and at the end of a job. The former is especially useful if you use the options to include the job name or the job number in the output path.

The post-process operation consists of running a DOS command (or script), to which you can supply various arguments. These arguments can either be literal text, or special substitution code variables which get replaced by their values.

Create Window

This option when enabled causes a Windows DOS box to be created. This allows you to see any text output that might appear, such as warning or error messages. It is mainly useful when writing and debugging scripts.

The DOS box goes away when the command completes, so to see any error message when creating/debugging your script, you might want to add a pause at the end of the script.

Command (File)

This option gets called at the end of outputting a PDF file.

All of the substitution variables can be used to pass arguments to the command or script.

Command (Job)

This option gets called at the end of outputting all the PDF files in a job.

The substitution variables which refer to individual PDF files ('c', 'f', 'j', 'p' & 'z') can NOT be used to pass arguments to the command or script; they will either be blank (empty strings - "") or set to the value from the last PDF file produced.

Substitution Code Variables

All substitution code variables can have a number between the '%' and the variable code. This number is used to restrict the maximum number of characters in the result.

The substitution code variables are:

| Code | Value |
|------|---|
| %c | The separation color, default length 1 (e.g. "C" for "Cyan"). |
| %d | The current date, default length 6, in the form YYYYMMDD. |
| %e | The job exposure as entered in the Page Setup dialog. |
| %f | The output PDF file name (e.g. "1BrochureC00.PDF"). |
| %g | A fixed job name; without the leading number prefix and with all non-alphanumeric characters removed. |
| %i | The input file name (e.g. "MyJob." - any suffix is removed). |
| %j | The page buffer name as shown in the RIP's Output Controller / Monitor (e.g. "1. Brochure (C)"). |
| %n | The current job number (as assigned by the RIP). |
| %o | The output folder; without any sub-folder added as a result of one of the "Path includes..." options. |
| %p | The current page number (as assigned by the RIP). |
| %r | The job resolution (as set in the RIP). |
| %s | A fixed job name; without the leading number prefix, with all parenthetical expressions removed, with all characters before any delimiter character removed (these being ":", ";", "@", "-" and all control characters) and with non-alphanumeric characters removed ¹ . |
| %t | The current time, default length 6, in the form of HHMMSS. |
| %x | The current file name suffix (e.g. "PDF"). |
| %z | The output file stem (e.g. ""1BrochureC00" or "PDF00" depending on the options set). |

¹ If this results in an empty name, then we effectively start again and only non-alphanumeric characters are removed.

Template Variables

Standard Variables

Standard variables are those which are compatible with Harlequin TIFF.

<colorant> variable

This generates a string of either “DeviceCMYK”, “DeviceRGB” or “DeviceGray”, depending on the output color mode of the Harlequin RIP.

<colorname> variable

This generates a string containing the full color name of the separation, or the string “Composite” for a composite page.

For example: “Cyan”.

<date> variable

This generates a string containing the current date in the form YYYYMMDD.

For example: “20070530”.

<exposure> variable

This generates a string containing the job exposure as entered in the Page Setup dialog, with a default length of 2.

For example: “100”.

<job#> variable

This generates a string containing the current job number (as assigned by the RIP), with a default length of 3.

For example: “012”.

<jobname> variable

This generates a string containing the job name, without the leading number prefix and with all illegal characters removed.

For example: "Brochure (C)".

<jobname1> variable

This generates a string containing the job name, without the leading number prefix and with all non-alphanumeric characters removed.

For example: "BrochureC".

<page#> variable

This generates a string containing the current page number (as assigned by the RIP), with a default length of 3.

For example: "001".

<prefix> variable

This generates a string containing the prefix from the job name, including the period and space.

For example: "1. ".

<prefixonly> variable

This generates a string containing the prefix from the job name, excluding the period and space.

For example: "1".

<realpage#> variable

This generates a string containing the real page number (as detected by the HqnPageCounter procset¹), with a default length of 3.

For example: "001".

<time> variable

This generates a string containing the current time in the form HHMMSS (using the 24 hour clock).

For example: "172405".

<unique> variable

This generates a unique number, with a default length of 3.

In any one template, the unique number used will be the same, unless the Extension Variables are enabled, in which case the number increases each time.

<xres> variable

This generates a string containing the (integer) horizontal job resolution (as set in the RIP), with no default length.

For example: "1270".

<yres> variable

This generates a string containing the (integer) vertical job resolution (as set in the RIP), with no default length.

For example: "1270".

¹ Note that this procset does not work very well in v5.5 of the Harlequin RIP, so this variable is not recommended for use there.

Special Variables

Special variables are also compatible with the Harlequin PDF raster plugin, but are used for specifying the particular platform file name requirements that the PDF file is being created for. If the resulting file name would be invalid for that particular platform, then an error is produced. Additionally, the platform type is used when removing illegal characters from the file name; the platform type should therefore appear at the beginning of any template.

None of these variables (apart from <dot>) produce any output.

<dot> variable

This variable produces a string containing “.”.

Furthermore, this variable is used to indicate which part of the file name consists of the prefix (everything before this) and which part of the file name consists of the suffix (everything after this). This then allows the lengths of the prefix and suffix to be checked against any platform requirements.

<dos> variable

This variable indicates that the file name will be created on a DOS file system. The prefix is checked to be no longer than 8 characters in length and the suffix to be no longer than 3 characters in length.

Illegal characters that are removed include “””, “*”, “:”, “\”, “|”, “/”, “?”, “<”, “>”, “ ”, “.”, (tab), (line feed), (carriage return) and any control characters (less than 32).

<mac> variable

This variable indicates that the file name will be created on a Mac (HFS) file system. The prefix and suffix total length are checked to be no longer than 31 characters in length.

Illegal characters that are removed include “'”, “*”, “:”, “\”, “|”, “/”, “?”, “<” and “>”.

<macosx> variable

This variable indicates that the file name will be created on a Mac (OSX / HFS+) file system. The prefix and suffix total length are checked to be no longer than 256 characters in length.

Illegal characters that are removed include “'” and “*”.

<unix> variable

This variable indicates that the file name will be created on a Unix file system. The prefix and suffix total length are checked to be no longer than 256 characters in length.

Illegal characters that are removed include “'”, “*”, “:”, “\”, “|”, “/”, “?”, “<”, “>”, “ ”, (tab), (line feed) and (carriage return).

<win32> variable

This variable indicates that the file name will be created on a Windows NTFS file system. The prefix and suffix total length are checked to be no longer than 256 characters in length.

Illegal characters that are removed include “'”, “*”, “:”, “\”, “|”, “/”, “?”, “<” and “>”.

Extended Variables

The extended variables are ones that have been added in addition to that of the Harlequin PDF raster plugin, providing extra functionality.

<colorcode> variable

This generates a string containing the standard letter designating the color name of the separation (for the standard colors of Red, Green, Blue, Cyan, Magenta, Yellow and Black), or the full name of the color in all other cases (which will be the string “Composite” for a composite page).

For example: “C” - Cyan, “K” – Black, etc...

<filename{XYZ}> variable

This generates a string containing the file name, without any trailing suffix and with all illegal characters removed.

If any additional text is present at the end of the variable name {XYZ}, then this is searched for in the file name. If the text is found, then it and all subsequent characters are removed.

<filename1{XYZ}> variable

This generates a string containing the file name, without any trailing suffix and with all non-alphanumeric characters removed.

If any additional text is present at the end of the variable name {XYZ}, then this is searched for in the file name. If the text is found, then it and all subsequent characters are removed.

<page#fromjobcomments> variable

This generates a string containing the real page number (as detected by the HqnPageCounter procset), with a default length of 3.

It is equivalent to using <realpage#>

<page#fromfilenameXYZ{+n}> variable

This generates a page number from the file name by searching for it in the file name. The additional text at the end of the variable name (XYZ) is used as the search string which must precede the page number.

If the (optional) “+” is present, this indicates the difference in page numbers between pages. Negative differences may be used. Lightning PDF will then use the page number for the first page, and add the difference to this for each subsequent page.

For example if you print pages 5, 7 & 9 of a job, you could call the file name “job_Page5.pdf” and use the variable “<page#fromfilenamePage+2>”.

The file name may also explicitly contain the page numbers of all pages in it, as a comma separated list (including ranges). In this case the option “+0” should be used.

For example the file name might be “job_Page1,2,9-11,15,22-30.pdf”, in which case the variable should be “<page#fromfilenamePage+0>”. This will therefore produce page numbers of 1, 2, 9, 10, 11, 15, 22, 23, 24, 25, 26, 27, 28, 29 & 30.

<page#fromjobnameXYZ{+n}> variable

This works in the same way as <page#fromfilename> but uses the job name instead.

Variable Modifiers

These are used to modify any of the string or integer strings that are produced. They must appear after the initial “<” and before the variable name.

String modifier

All string variables can have a number “max” after the initial ‘<’ and before the variable name. This number is used to restrict the maximum number of characters in the result; if a string is longer, it is truncated from right-to-left.

For example: “<2colorname>” with “Cyan” produces “Cy”.

Number modifier

All number variables can have a “min:max” after the initial ‘<’ and before the variable name, where either of the two numbers (min/max) (and ‘:’) are optional.

The first number defines the default (minimum) number of characters used to represent the result, with ‘0’ characters being added (on the left) if insufficient exist.

The second number defines the maximum number of characters in the result; if a number is longer, it is truncated from left-to-right. The second number must be larger than the first number (or the default).

For example: “<2page#>” with page #1 produces “01”
and with page #100 produces “100”.

For example: “<3:page#>” with page #12 produces “012”
and with page #1234 produces “1234”.

For example: “<:4page#>” with page #12 produces “012”
(default length is 3) and with page #12345
produces “2345”.

For example: “<2:3page#>” with page #1 produces “01” and
with page #12345 produces “345”.

Extended Variable Modifiers

These are used to modify any of the job name or file name strings that are produced. They must appear after the initial "<" and before the variable name.

[123-456] modifier

This is used to indicate a range of characters to keep. The first number indicates the first character to keep and all characters up to and including that indicated by the second number are used.

Both numbers must be non-zero.

[123&456] modifier

This is used to indicate a range of characters to discard at the start and end. The first number indicates the number of characters to remove at the start and the second number indicates the number of characters to remove at the end.

Either number may be zero.

[abc/def] modifier

This is used to indicate two sequences of characters in the name; all characters up to and including the first sequence are discarded, as are all characters from and including the second sequence.

Either sequence may be 'empty'.

Troubleshooting

Lightning PDF has been designed to take care of most processes automatically, however, if any errors do occur you will be kept informed with appropriate error messages.

These will appear in the RIP's Monitor Window prefixed with the string "! Lightning Plugin (PDF)". If the RIP is running in Multiple Mode, then an error icon and message will also appear in the Output Monitor / Controller, however, due to the restricted space there, this message will only indicate that there has been an error and the message in the RIP's Monitor Window should be checked.

The following table shows all the error messages that Lightning PDF will produce and gives a short explanation of what they all are.

| Problem | Solution |
|-------------------------------------|--|
| File Exists: %s | This is most likely caused by not using any uniqueIDs in the file name. Add some. |
| File Post processing command failed | The error here depends on the command or script being used. In this case, check with your supplier or the person who wrote the script. |
| Job Post processing command failed | The error here depends on the command or script being used. In this case, check with your supplier or the person who wrote the script. |
| PDF Writer Aborted | The user aborted the job or the output. |
| Memory Allocation Failed | Not enough memory was available to the plugin. Increase the amount of virtual memory on the system, or decrease the amount of memory given to the RIP. |
| File Creation Failed | Could not create the PDF file. This could be due to illegal characters in the file name, a file already existing with that name, insufficient access permissions, etc... |

Lightning PDF User Guide

| | |
|--|--|
| File Open Failed | Could not open the PDF file to write to it after creating it. A very rare error. |
| File Seek Failed | Could not seek in the open PDF file to write to it. A very rare error. |
| File Write Failed | Could not write data to the PDF file. Most likely caused by either a disk getting full, or by a remote file connection being dropped. |
| File Close Failed | Could not close the PDF file and flush any remaining data to it. Most likely caused by either a disk getting full, or by a remote file connection being dropped. |
| File Delete Failed | Could not delete the (temporary) file. Most likely caused by a remote file connection being dropped. |
| File Rename Failed | Could not rename the (temporary) file. Most likely caused by either the target file suddenly existing, or a remote file connection being dropped. |
| Illegal CCITT Compression for 8 bits/pixel | An internal error when the compression method is not valid. |
| Compression ran out of temporary memory (Internal Error) | An internal error when the compression method runs out of memory. This should not happen with a 64-bit RIP (or even a 32-bit RIP). |
| Could not create a Unique File | The conventional or template method used could not generate a unique file that doesn't already exist. Make sure that the file name includes unique ids of sufficient size for your jobs. |
| Resulting File Name already exists | The conventional or template method used generated a file name that already exists. Make sure that the file name includes unique ids of sufficient size for your jobs. |
| DOS File Stem too long | The length of the file name stem exceeds 8 characters. Change your template to reduce its length. |
| DOS File Extension too long | The length of the file name stem exceeds 3 characters. Change your template to reduce its length. |

Lightning PDF User Guide

| | |
|--|--|
| Mac File Name too long | The length of the file name exceeds 31 characters. Change your template to reduce its length. |
| File Name too long | The length of the file name exceeds 256 characters. Change your template to reduce its length. |
| Malformed Template | There was an error in the template – for example a right bracket “]” was missing. Check and fix the template. |
| Unknown Template | The template variable name is unknown. Check and fix the template. |
| Template too long | The template results in a file name that is too long. Shorten the template. |
| Conventional File Name too long | The conventional name results in a file name that is too long. Either enable the “Use 8.3 filenames” or use a template name instead. |
| Unknown PostProcess Substitution character | The post process operation encountered a substitution character that was not recognized. Check and fix the post process string. |
| Post-Process Command too long | The resulting post process string is too long. Shorten the post process string. |
| Post-Process Command failed | Execution of the post-process command failed. The error here depends on the command or script being used. In this case, check with your supplier or the person who wrote the script. |

Glossary

| | |
|-----------------|---|
| PDF file | A PDF file as per the Adobe PDF Reference Edition, which can be anything from the Third Edition version 1.4, to the Sixth Edition version 1.7. See www.adobe.com for further details. |
|-----------------|---|

Contact

Use the following contact details if you wish to contact Hamillroad Software.



E-mail technical support will be provided to registered users free of charge for a period of 60 days from date of registration. Thereafter a charge may be levied.

| | |
|-------------------|------------------------|
| Technical Support | support@hamillroad.com |
| General | info@hamillroad.com |
| Website | www.hamillroad.com |