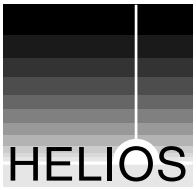


ImageServer G8

(Version 6.0.0)

User manual

September 10, 2020

Contents

1 About the chapters of this manual

In the following, we give a brief summary of each chapter of this manual. This summary is meant to help you find the information you are looking for.

Chapter 2 “An introduction to HELIOS ImageServer” gives background information. For readers who have little practical experience with the functionalities provided by ImageServer, it discusses basic concepts.

Good to know

Chapter 3 “Before getting started” provides background information about image file formats, color modes, image modifications, and printer characteristics. The explanations may be very useful for you, because they reveal how the settings you choose may influence your workflow. The chapter also describes how ImageServer behaves in specific situations.

How to administer the ImageServer settings

Chapter 4 “Set up ImageServer with HELIOS Admin” gives directions for the basic ImageServer configuration as well as the specific settings for each volume and printer queue.

How to use the ImageServer software

Chapter 5 “Using ImageServer” describes different utilities you may use to tune your layout files for specific purposes. The chapter then gives instructions on how to place layouts in a layout document and on how to set up the printer options in different OPI-compatible applications. This chapter is meant for those who simply want to use ImageServer and do not care about the programming or the administration of the program.

ImageServer utility programs and preferences

Chapter 6 “ImageServer utility programs” describes the programs of which ImageServer is composed, and describes in detail how to change options and attributes of these programs manually.

Chapter 7 “HELIOS Script Server” describes how to define hot folders in order to automate processing tasks. Users can then e.g. simply drop files into a specific folder which are then processed by a custom conversion script.

Chapter 8 “HELIOS Tool Server” allows integrating remote applications/tools to be used by a main server.

Chapter 9 “Preferences and notification features” lists the preferences of ImageServer and OPI-related printer queues. Additionally, OPI event notification features are covered in this section.

How to find help

Chapter 10 “Troubleshooting” describes specific settings or parameter conflicts that may lead to unexpected results, and explains how to avoid errors in particular situations.

Chapter 11 “Technical support” gives advice on how to set up a complete error description. Such a description is indispensable if you want to report a particular problem – which you cannot solve by using the documentation – to your dealer or to the HELIOS support department.

Additional information

Appendix A “About OPI settings, typical workflows, and conflicts” gives examples that illustrate how different OPI settings cooperate or how they may influence one another.

Appendix B “Colors, color matching, proof printing – basic concepts” gives background information about color management.

Appendix C “ImageServer and color space” describes how to preserve the original color space in a composite printing environment, and how the RGB-to-Hexachrome color space conversion workflow can be set up. In addition, based on the cooperation between HELIOS ImageServer and HELIOS PrintPreview, the spot color editor is described.

Appendix D “Raster image transparency” lists the categories of image documents with transparency and describes the several techniques of transparencies in raster images and their support in ImageServer.

Appendix E “Raster PDF format” introduces the HELIOS raster PDF and Adobe Photoshop PDF file formats and discusses their advantages and disadvantages. In the following the creation and use of the HELIOS raster PDF format is described.

Appendix F “Notes for PCShare users” explains the settings that are necessary if you want to use the ImageServer features from a PCShare client on a Windows PC.

Appendix G “Glossary” explains some product-related terminology.

2 An introduction to HELIOS ImageServer

ImageServer facilitates the handling of large image files in a typical prepress environment, and can be used for server-based repurposing of image files, including color management, color space transformation, compression, resolution, and image file format conversion between all supported image formats.

Note: HELIOS ImageServer runs on top of the foundation provided by HELIOS Base. Please read the HELIOS Base manual for installation instructions and other important details.

2.1 New ImageServer features

For new features in the ImageServer software see the HELIOS website:
www.helios.de Go to *HELIOS Product Versions – New Features*

For HELIOS Base, the foundation used by all HELIOS products, see the HELIOS Base product web page:
www.helios.de Go to *Products > Base*

2.2 ImageServer functionality

2.2.1 Overview

HELIOS ImageServer incorporates much more than just OPI image replacement. ImageServer is an integrated solution that makes the entire image management and print workflow more versatile and efficient. Here are some general advantages of OPI, along with additional features of ImageServer:

- Speed improvement and output optimization
- Remote workflow support
- Color management
- Composite PDF generation overcomes QuarkXPress limitations (QuarkXPress 3-9)
- Composite DeviceN printing support overcomes QuarkXPress limitations (QuarkXPress 3-9)
- Script Server workflow automation
- OPI image replacement
- Image conversion
- Tagging of ICC profiles
- Re-purposing the same print job
- Application compatibility
- Error checking
- QuarkXPress and InDesign document previews
- Tool Server remote application automation

These capabilities are detailed in the following sections, and in other areas of this manual.

2.2.2 OPI functionality (image replacement)

At first, you scan your original image with an appropriate (comparatively high) resolution. Then, you retouch this high-resolution image using an image processing application (such as Photoshop) and save the result. As soon as the new high-resolution original image file is saved in a HELIOS volume, the OPI software automatically generates a low-resolution duplicate (called a *layout file*) that needs much less disk space. OPI-compatible layout applications such as QuarkXPress or InDesign, which are running on network clients, can work with these small proxy files which leads to a reduction of processing time. All modifications like positioning, scaling or rotating are marked by OPI comments which are inserted in the PostScript when you launch the print command. During printing, the modifications are applied to the high-resolution original images that now replace the low-resolution substitutes.

With this concept, large image files are completely kept away from the layout computers in a network. Thus, printing from a client is much faster and the network traffic is reduced.

Fig. 2.1 shows a prepress environment and illustrates how you can benefit from HELIOS ImageServer.

This same concept is also used to enable remote workflow support via the internet.

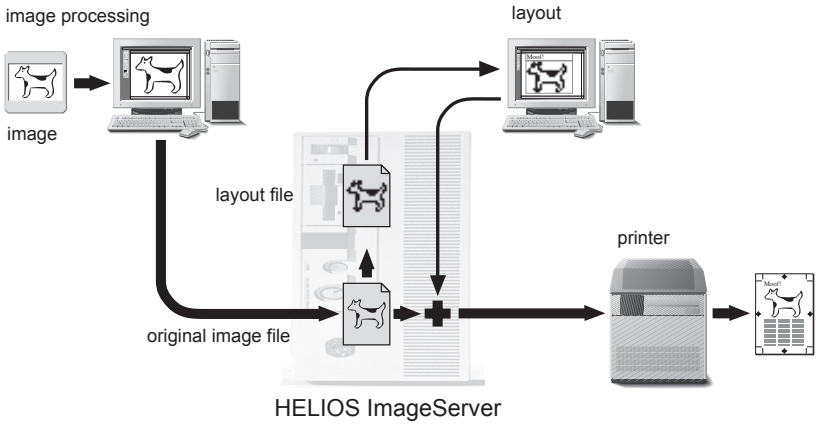


Fig. 2.1: The handling of image files on ImageServer

The OPI standard and the syntax and structure of OPI comments were created by the Aldus Corporation that now belongs to Adobe Systems, Inc.

2.2.3 Server-based image conversion

ImageServer supports a wide range of image formats and color spaces including clipping paths and spot colors. It can convert images to any supported file format, resolution, compression and color space to be used for printing, web, archive or any other purpose. HELIOS PDF HandShake adds PDF file format support. See also:

6.1 “layout”

6.4 “oiimginfo”

5.3 “Tag images with ICC profiles”

This image conversion capability is used extensively by ImageServer "behind the scenes". It can also be configured to provide automatic or hot folder based image conversion per user specifications.

2.2.4 Server-based color management

ImageServer supports ICC profiles and includes full ColorSync support, which has been co-developed with Heidelberger. ICC profiles will be applied for layout generation, image conversion, printing, and proofing. ICC profiles can be tagged to individual images, for a specific folder as well as for volume and server defaults. See 5 "Using ImageServer" for more information.

2.2.5 Script Server

ImageServer includes a Script Server service which implements a hot folder mechanism. You simply drop files into the hot folder. The idea of a hot folder is that, as soon as files are dropped into this folder, a process is automatically started which processes the files according to a given script. See 7 "HELIOS Script Server" for more information.

2.2.6 HELIOS OPI TuneUp XT

ImageServer supports foreground and background colorization of black & white and grayscale TIFF images. ImageServer includes the QuarkXPress extension "HELIOS OPI TuneUp XT" (Windows and Mac versions 3-9). It adds support for foreground and background colorized images in QuarkXPress. The extension includes additional ADSC comments in the print job, which allow colorizing images during OPI replacement. See 5.7 "Using QuarkXPress with ImageServer" for instructions on how to use the "HELIOS OPI TuneUp XT".

2.2.7 InDesign and QuarkXPress document previews

The HELIOS Preview plug-in allows creating preview files of Adobe InDesign 2.x (or newer) documents (see 5.8.1 “The HELIOS Preview plug-in”). HELIOS Preview XT creates previews of QuarkXPress 5 (or newer) documents utilizing these previews (see 5.7.3 “HELIOS Preview XT”). Local and remote web based users can preview InDesign or QuarkXPress documents from creation to final output, without needing access to the application, documents, fonts or images. Previews can be viewed, managed, and transferred in HELIOS WebShare, via any web browser. For details see 5.9.2 “Viewing preview images in WebShare”.

These multi-page preview files facilitate finding documents, content verification, and local and remote collaboration. Preference settings specify the preview resolution. ImageServer “hot folder scripts” allow fully automated processing for archiving, printing, converting to any major image file format or to a customized workflow. See 7 “HELIOS Script Server”.

2.3 Known limitations

There are still some situations that might cause problems when working with ImageServer. In most cases, this is due to application specific incompatibilities. The following list of known limitations is not “everlasting”. You should always compare it to the latest README files issued by HELIOS and “Tech Info” documents on the HELIOS website, in order to find out whether a problem is already known.

Various limitations

ImageServer does not support layer masks saved in Photoshop native format. When using layer masks in Photoshop and saving the images as Photoshop native, blank layout images will be generated. Please apply any layer mask to its layer and remove it before saving the image.

ImageServer does not fully support PICT images based on object descriptions other than PixMaps. Such images may be created by applications like Illustrator or FreeHand and may cause incomplete previews.

ImageServer will not work as expected if mounting NFS volumes using the UNIX auto-mount facility. Please hard mount NFS volumes by using the mount program.

QuarkXPress

We found that QuarkXPress matches spot color names found in placed EPSF files to its own ones by comparing the names of the imported colors with the built-in color tables. The program sometimes matches names which are only partially identical and then continues using its own names when editing colors and printing. This may result in missing spot color plates because the plate printed by QuarkXPress does not match the spot color name in the image. To avoid such problems, you should rename the spot colors in your images.

Example:

An image generated by Adobe Photoshop includes a channel named “PANTONE Orange 021 CVC”. QuarkXPress matches it to “PANTONE Orange 021 CV” in its own table and uses this name further on. While printing, a plate named “PANTONE Orange 021 CV” will be printed but no OPI image replacement can be done because of the different name stored in the image.

QuarkXPress 3.3 (Mac):
For PC-EPSF format DCS images in QuarkXPress, printing `Separations` will cause problems when you include images in the print job. `Include Images` for `TIFF` and `EPS` must NOT be selected. Otherwise, QuarkXPress will generate bad PostScript.

QuarkXPress 7 (PC/Mac):
Transparency features do not work with OPI images. This is a limitation in the XPress PDF Export, which always flattens PDF files instead of preserving transparencies.

3 Before getting started

3.1 Backward compatibility

ImageServer supports backward compatibility to EPSF and TIFF layout files that have been generated by previous ImageServer versions.

EtherShare OPI 1.x will not be able to handle and understand layout files generated by ImageServer UB or newer. However, ImageServer 2.5 will handle them, provided it already supports the used image format.

Please note that if you are using EPSF layout files which were generated by EtherShare OPI version 2.1 or earlier, downsampling, color matching and other new features will not be supported when you print separations. In that case, you have to generate new layouts from the high-resolution original images and thus create layout files that can be subjected to all features available in the new program version. You may use the HELIOS “touch” program for regenerating layouts (detailed instructions are given in 5.5 “Refresh layouts using the “touch” program”), or the HELIOS “opitouch” command (see 6.2 “opitouch”).

3.2 About images, color modes, file formats, and compression

This section provides an overview of ImageServer supported file formats, color spaces, and compression options for layout file generation. It is equally applicable to image conversion using ImageServer Script Server.

3.2.1 General remarks

An image file is defined by its file format. In addition to that, it contains information about its color mode and the mode of compression. These three characteristics depend on one another and may not be combined arbitrarily. An example is given in the following illustration.

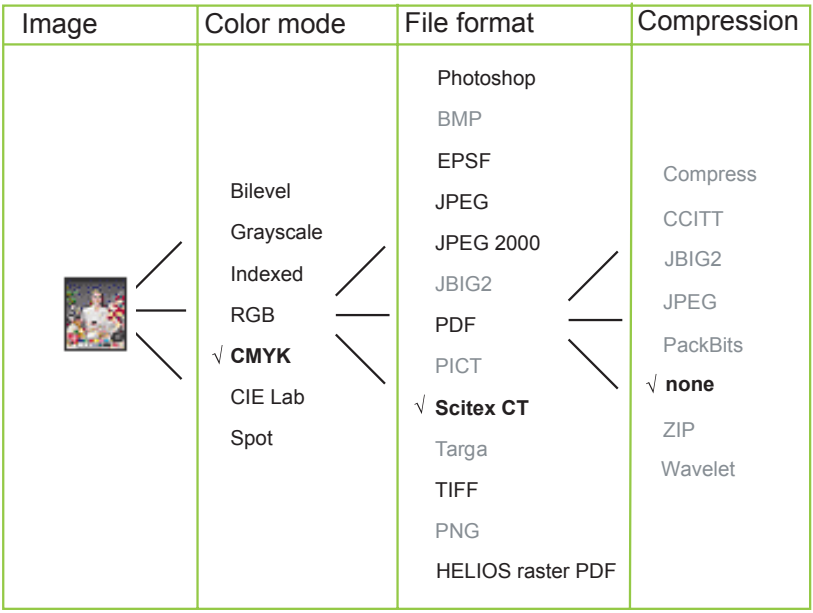


Fig. 3.1: The characteristics that define a given image file

Fig. 3.1, for example, shows a Photoshop CMYK image in the Scitex CT format, which cannot be compressed (unavailable options are grayed out in the illustration).

Note: Please note that some file formats are already compressed (e.g. JPEG).

In the following, we will ...

- ... give you a complete list of file formats supported by ImageServer
- ... describe the advantages/disadvantages of these formats
- ... discuss the different compression modes

3.2.2 64-bit file offset support

ImageServer handles file sizes up to 64-bit, which equals 8.3 million TB (*Terabyte*). The old 32-bit support can only handle files up to 2 GB. This is important because image files today can easily exceed 2 GB.

3.2.3 Supported file formats

ImageServer provides OPI facilities and, in addition, contains built-in color management features. The software accepts different file formats for the generation of (low-resolution) layout files and it supports various file formats for color matching. Please note that some file formats in connection with specific color modes allow the generation of layout files, but do not allow color matching.

Table 3.1 lists all supported file formats for the generation of layout files (and their possible color modes), whereas Table 3.2 only lists those file format/color mode combinations that allow color matching. If you want to achieve predictable color results on your printouts, you should select for your image files one of the file formats listed in Table 3.2. Table 3.3 gives an overview of the image file formats that are supported by ImageServer.

File format	Color mode
TIFF	Bilevel, Grayscale, Indexed, RGB, CMYK, CIELab, Multichannel (e.g. Hexachrome), YCbCr*
EPSF raster- and object-based images	Bilevel, Grayscale, Indexed, RGB, CMYK, CIELab, Spot, Multitones*
PICT	Bilevel, Grayscale, Indexed, RGB
JPEG	Grayscale, RGB, CMYK, CIELab, YCbCr*
JPEG 2000	Grayscale, RGB, CMYK, CIELab, YCbCr*
JBIG2*	Bilevel
Scitex CT	Grayscale, CMYK
Photoshop	Bilevel, Grayscale, Indexed, RGB, CMYK, CIELab, Spot, Multichannel*
ICS	CMYK
BMP	Bilevel, Indexed, RGB
PNG	Grayscale, Indexed, RGB
PDF**	Bilevel, Grayscale, Indexed, RGB, CMYK, CIELab, Spot, Multichannel*
HELIOS raster PDF	Bilevel, Grayscale, Indexed, RGB, CMYK, CIELab, Spot
DCS 1, DCS 2	CMYK, Spot, Multichannel*
Raw	RGB*

Table 3.1: High-resolution file formats that are supported by ImageServer

* Currently read-only support

** PDF HandShake option

File format	Color mode
TIFF	RGB, CMYK, Grayscale, CIELab, Multichannel (e.g. Hexachrome), YCbCr*
EPSF raster-based images	RGB, CMYK, Grayscale, CIELab
EPSF object-based images	Not supported!
PICT	RGB
JPEG	RGB, CMYK, Grayscale, CIELab, YCbCr*
JPEG 2000	RGB, CMYK, Grayscale, CIELab, YCbCr*
JBIG2*	Bilevel
Scitex CT	CMYK, Grayscale
Photoshop	RGB, CMYK, Grayscale, CIELab, Multichannel*
ICS	CMYK
BMP	Bilevel, Indexed, RGB
PNG	RGB, Grayscale
PDF**	RGB, CMYK, Grayscale, CIELab, Multichannel*
HELIOS raster PDF	RGB, CMYK, Grayscale, CIELab
DCS 1, DCS 2	CMYK, Multichannel*
Raw	RGB*

Table 3.2: File format/color mode combinations that are supported for color matching

* Currently read-only support

** PDF HandShake option

File format	Type	Suffix
Tagged Image File Format	TIFF	.tif, .tiff
JPEG or JFIF	JPEG	.jpg, .jpeg
JPEG 2000	JP2	.jp2, .jpx
JBIG2	JB2	.jb2, .jbig2
Encapsulated PostScript File	EPSF	.eps, .epsf
DCS 1	EPSF/DCS 1	.eps, .epsf
DCS 2	EPSF/DCS 2	.eps, .epsf
Binary Encapsulated PostScript File	EPSF/BIN	.eps, .epsf
Binary DCS 1	EPSF/DCS 1 BIN	.eps, .epsf
Binary DCS 2	EPSF/DCS 2 BIN	.eps, .epsf
Scitex CT Handshake Format	..CT	.sct
Adobe Photoshop Format	8BPS	.psd, .psb
Macintosh Picture Format	PICT	.pic
Macintosh Picture format Version 1	PICT/1	.pic
Macintosh Picture format Version 2	PICT/2	.pic
Macintosh Custom Icon Family Format	MacIcon	
Adobe Path Resource Format	AdobePath	
BMP	BMP	.bmp
Portable Network Graphics File Format	PNGf	.png
HELIOS Document Preview (XPPV)	XPPV	.xpv
Portable Document Format (raster-based)	PDF	.pdf
Portable Document Format (object-based)	PDF/Vector	.pdf
Unknown	UNKN	

Table 3.3: File formats, image types and suffixes

3.2.4 File format characteristics

Before you save an image in an image processing application (or scan or digital camera system), you should make a decision on the file format you want to choose. The advantages and disadvantages of the different file formats are described below.

Important: Again, the file formats have to be judged from different points of view. Some characteristics of a given file format may be advantageous in general, and for OPI processes, but – at the same time – disadvantageous with regard to color matching!

The format you choose for an original high-resolution image file determines the format of the generated layout file. Please refer to 3.2.6 “File formats of OPI-generated layout files” for explanations on the different layout file formats.

TIFF

❍ Advantages

- Many scan systems and image processing applications can generate TIFF files. Thus, they become – in principle – OPI-compatible
- TIFF files can be edited by all popular image processing applications
- TIFF files are independent of the output device
- For TIFF files, portions of an image can be sent to the output device (in case you have selected only a specific part of the image in the layout application)
- The resolution of TIFF files can be reduced during output and adapted to low-resolution output devices
- The TIFF format supports many color modes
- A clipping path that has been defined in image processing applications is preserved for printing (for details and *exceptions* see **Image paths and clipping paths**)
- May contain additional metadata (XMP, EXIF)

EPSF (raster-based)

❍ Advantages

- A clipping path that has been defined in image processing applications is preserved for printing (for details and *exceptions* see **Image paths and clipping paths**)

❍ Disadvantages

- Many image processing applications cannot re-edit EPSF files
- Complete images are sent to the output device, even if you have selected only parts of the images in the layout application
- When printing separations, the entire EPSF image must be sent to the RIP four times (ImageServer solves this problem)
- The resolution of an EPSF image remains unchanged (e.g. 1200 dpi) even if you are printing to an output device that only allows a lower resolution (e.g. 300 dpi). Thus, printing may become very slow (ImageServer solves this problem via optional downsampling)

EPSF (object-based)

❍ Advantages

- Smaller file size
- Files can be zoomed without any loss in quality

❍ Disadvantages

- There are few image processing applications that can read object-based EPSF files

PICT

❍ Advantages

- PICT files are small and preserve masks or paths

❍ Disadvantages

- The format is meant for Mac platforms only
- Not supported by OS X

JPEG

❍ Advantages

- Files are highly-compressed and thus are very small
- May contain additional metadata (XMP, EXIF)

❍ Disadvantages

- The mode of compression affects the quality of the images. Therefore, the JPEG format is mainly used for archiving purposes.

JPEG 2000

❍ Advantages

- Provides a higher compression rate than JPEG due to "Wavelet" compression
- Unlike JPEG, JPEG 2000 allows lossless compression methods
- Supports additional metadata and XML specifications within the JPEG 2000 file

❍ Disadvantages

- Limited support (few applications support JPEG 2000)

JBIG2

❍ Advantages

- Very high compression rate, which exceeds CCITT
- Both lossy and lossless compression possible
- Halftoning of Grayscale images

❍ Disadvantages

- Limited support (few encoders can write the JBIG2 image format)

Scitex CT

❍ Advantages

- Scitex CT files are pre-separated. Thus, they are already prepared for printing separations

❍ Disadvantages

- Pre-separation is disadvantageous for color matching, because the separation plates are created for an image that has not yet been subjected to the matching process
- This format does not allow any mode of compression
- This format is only available for Grayscale and CMYK images

Photoshop native

❍ Advantages

- The Photoshop Native format supports all color modes
- For multi-layer images, the layer information is preserved
- May contain additional XMP metadata

❍ Disadvantages

- None

BMP

❍ Advantages

- Supported by Windows

❍ Disadvantages

- No CMYK, Lab and clipping path support

PNG

❍ Advantages

- Supported by web browsers
- Supports transparent clipping paths (implemented via alpha channels)

❍ Disadvantages

- No CMYK and Lab support

DCS 1

❍ Advantages (compared to standard EPSF files)

- DCS 1 files are pre-separated and are composed of several files containing the EPSF separation plates for cyan, magenta, yellow, and black. Thus, for a given color, only the file that describes this specific separation is sent to the RIP (if you are working with the EPSF format, you only have one file that contains the complete image description. This large file has to be sent to the RIP four times)

❍ Disadvantages

- Pre-separation is disadvantageous for color matching, because the separation plates are created for an image that has not yet been subjected to the matching process
- Composite printers, usually, are unable to handle the four EPSF separation plates at a time and thus cannot produce a composite printout

DCS 2

❍ Advantages (compared to the DCS 1 format)

- They contain the EPSF separation plates for cyan, magenta, yellow, and black, and – optionally – additional spot colors, in only one file

❍ Disadvantages

- See **DCS 1**

PDF

❍ Advantages

- Allows storing illustrations, text, and images in one file
- Supports several color spaces like RGB, CMYK or LAB
- Supports several compression modes such as JPEG, ZIP, CCITT or JBIG2
- Supports several types of fonts (PostScript and TrueType)
- Editing is possible using Acrobat

❍ Disadvantages

- The various content types can present printing problems (PDF HandShake solves these problems)

HELIOS raster PDF

❍ Advantages (compared to the PDF format)

- Can be repurposed, i.e. converted back into any supported image format

❍ Disadvantages

- No support for Photoshop resources, which means no clipping path support

3.2.5 Supported compression modes

ImageServer supports several file formats for the generation of layout files, even if the data contained in these files has been subjected to compression. The original high-resolution files remain compressed, but usually do not pass on the compression to the layout files. If you want the layout files to be compressed as well, you may use HELIOS Admin or the respective parameters (see 4.1 “ImageServer settings” and 6.1 “layout” for details). Table 3.4 lists various modes of compression supported by ImageServer and indicates whether they can be applied to the different file formats.

File format	Compression mode
TIFF	Compress°, CCITT RLE, CCITT G3 Fax, CCITT G4 Fax, JPEG, PackBits RLE, Flate (Zip)°
EPSF	Compress***, CCITT RLE***, CCITT G3 Fax***, CCITT G4 Fax***, JPEG, Flate (Zip)
PICT*	
JPEG	JPEG
JPEG 2000	JPEG 2000°
JBIG2*	JBIG2
Scitex CT	Compress, CCITT RLE, CCITT G3 Fax, CCITT G4 Fax
Photoshop	PackBits RLE
Photoshop PSB	PackBits RLE°
ICS	Compress***, CCITT RLE***, CCITT G3 Fax***, CCITT G4 Fax***
BMP	PackBits RLE
PNG	Flate (Zip)°
PDF**	Compress, CCITT RLE, CCITT G3 Fax, CCITT G4 Fax, JPEG, JPEG 2000°, JBIG2, PackBits RLE, Flate (Zip)
HELIOS raster PDF	Compress, CCITT RLE, CCITT G3 Fax, CCITT G4 Fax, JPEG, JPEG 2000, JBIG2, PackBits RLE, Flate (Zip)
DCS 1, DCS 2	Compress***, CCITT RLE***, CCITT G3 Fax***, CCITT G4 Fax***, JPEG

Table 3.4: Compression modes and their relation to different file formats

* Currently read-only support

** PDF HandShake option

*** This compression mode is allowed but will cause restrictions: raster-based data is no longer recognized by ImageServer

° 16-bit compatible

3.2.6 File formats of OPI-generated layout files

The layout files that are generated by the OPI software are either EPSF or TIFF, by default, depending on the color mode and file format of the high-resolution original image.

EPSF layout files

The OPI server generates EPSF layout files if

- ❍ the high-resolution original image is an EPSF, ICS, PDF or DCS image or
- ❍ the high-resolution original image is not an EPSF, ICS, PDF or DCS image, but:
  - it contains color channels that do not belong to the selected color mode (e.g. spot colors)
  - it contains a clipping path

TIFF layout files

TIFF layout files are generated of every high-resolution original image that

- ❍ is not an EPSF, PDF, ICS or DCS image
- ❍ does not contain clipping paths

Note: For additional information about PDF-native OPI workflow see “PDF-native OPI” in the HELIOS PDF HandShake manual.

EPSF and TIFF layout files

In the following, we briefly discuss the differences between EPSF and TIFF layout files. Knowing these differences can help you decide which layout file format you want to use. (You may then select the color mode and file format for your high-resolution original images and thereby indirectly determine the desired layout file format.)

There are mainly two aspects that make up the difference between EPSF and TIFF layout files:

- EPSF files do not allow source cropping. If you crop an image in a layout application, this does not reduce the amount of data that is sent to the print server, because you can include only complete EPSF layout files into a print job. This is different if you are using TIFF layout files and then crop the images. In that case, the print job will only contain the data that describes the visible part of the image.

- Many applications – even on different platforms – are able to place EPSF layout images. Thus, these applications automatically become (quasi-) OPI-compatible (see **Layout applications** in 3.3.1 “OPI-compatible applications”).

Other layout file formats?

In most cases, it is convenient and sensible to accept the default EPSF or TIFF layout file format. Nevertheless, you may as well choose any other format for your layout files. 5.2 “Define folder specific OPI settings” explains how to set custom settings on a “per volume” or “per folder” basis.

Color modes for layout images

The color modes of the high-resolution originals are not passed on to the layouts. By default, layout images are Euroscale- or SWOP-CMYK, to allow the layout applications to print separations. In special situations, you may also select the RGB or Lab color mode for layout images (see **Layout Color** in 4.1 “ImageServer settings”).

3.3 About applications, OPI comments, and modifications

3.3.1 OPI-compatible applications

The ImageServer software requires two types of OPI-compatible applications, namely:

- Scan systems, image processing applications or similar programs that are able to generate one of the image file formats which allow the generation of layout files
- Layout applications which allow importing layout files *and* are able to generate OPI comments

Image processing applications

Most scan systems and image processing applications are OPI-compatible as far as the "input" side is concerned. They all offer the possibility of creating at least one or two file formats that are supported by ImageServer for the generation of layout files and for color matching (e.g. TIFF-RGB or raster-based EPSF-CIELab). One of the most frequently used image processing applications is Adobe Photoshop – ImageServer supports the Photoshop Native format.

Layout applications

Strictly speaking, OPI-compatible layout applications have to meet two different demands: they have to be able to import/place layout images and they have to be able to generate OPI comments according to the Adobe OPI standard and place these comments into the PostScript that is generated for each print job.

There is, however, one exception from this rule: layout applications that are able to import EPSF layout files are quasi-OPI-compatible, even if they do not generate OPI comments. This is due to the characteristics of EPSF layout files which simply do not allow applications to modify their contents. If an EPSF layout file already contains OPI comments – e.g. a reference to the location of

the original high-resolution image file – these comments are preserved and passed on to the PostScript file.

Note: Some layout applications that are fully OPI-compatible also offer OPI options in their print dialogs, namely “Omit TIFF” and “Omit TIFF & EPS”, and thus provide additional means of speeding up print jobs when working with OPI. Details about the “Omit . . . ” options are given in 5.6 “OPI print options – Printing from QuarkXPress and InDesign”.

The most frequently used layout applications that are – in principle – fully OPI-compatible are listed in Table 3.5.

Application	Mac & Windows PC
QuarkXPress	3.1-9
InDesign	1.0 and higher

Table 3.5: OPI-compatible layout applications

Other applications like CorelDraw, FrameMaker, Word, and MS Publisher may also be involved in the process of creating documents that include text and images. Strictly speaking, they were not designed for layout purposes in prepress environments. These applications are – if at all – only quasiOPI-compatible, when they use EPSF layout files.

The mentioned restrictions are also applicable to layout applications for UNIX systems. Most of these applications are able to import EPSF layout files, and thus can be used with ImageServer.

3.3.2 Supported image modifications and OPI comments

Image modifications

Different layout applications provide different features for image modifications. Depending on your application, you can e.g. *move*, *scale*, *rotate*, *skew*, *reflect*, *tint* or *crop* your images, and apply a *clipping path*, or *overprinting*. All these modifications are supported by ImageServer and are automatically applied to the high-resolution original images during printing.

Important: *Transparency* definitions for grayscale and color images are the only modifications that are not supported by ImageServer, due to limitations in the OPI specifications. You may overcome this limitation with the PDF-native OPI workflow, which is described in the HELIOS PDF HandShake manual.

OPI comments

If you print from a layout application, the OPI server requires different information about the images that are included in the document. There are several OPI PostScript comments that contribute to a complete image description. Some comments only serve to clearly identify the high-resolution image; others are used to describe the modifications that have been applied to the layout.

For example, the comments "%ALDImage Dimensions" and "%ALDImage-Position" define the dimensions and the position of an image, and thus may also include information about whether you have moved or scaled the layout image.

Image paths and clipping paths

In Photoshop, the outline of a selected area can be saved as a vector outline or *path*. Multiple paths can be created and saved with the file. One of the paths can also be selected as a *clipping path*. A clipping path can be activated to mask part of an image, so that only the desired portion is seen in the printout. Some page layout programs recognize and utilize these paths. However, the behavior varies depending upon the program and file format. For example, some applications have poor support for TIFF files containing paths, while

others have excellent support. ImageServer can be set to produce layout files with the desired path properties for your specific page layout programs and objectives.

Multiple Photoshop paths

ImageServer supports multiple paths, any one of which may also be designated as the active clipping path. Multiple paths are migrated into converted files. Layout images inherit these paths if no path is defined as clipping path. If a clipping path is defined, only this is applied. Photoshop paths can be exported/imported into existing files, so users can re-use the images and use different paths for different jobs (see `-o ImportMask <string:None>` in 6.1.8 "Image conversion options" and 6.1.12 "Tagging options"). In layout applications such as QuarkXPress or InDesign, it is possible to select a defined path and make it the active clipping path.

TIFF files with clipping paths

Clipping paths can be saved within TIFF images in an image processing application. Even though the image will print correctly and the clipping path will be applied, some applications cannot display a TIFF clipping path, and hence are unable to properly use features such as text "Runaround". Therefore, by default, ImageServer generates EPSF layout images of TIFF hi-res files that contain a clipping path. Layout applications can then properly display and use the clipping path.

If your layout application, e.g. QuarkXPress or InDesign, does support the TIFF clipping path feature then you can use either EPSF or TIFF layout images. Some applications also have the capability to add or modify the image clipping path in TIFF files. If you wish to take advantage of this feature, layout images should be generated in TIFF format (see 5.2 "Define folder specific OPI settings"). TIFF layout images will always inherit all paths from the high-resolution image.

PC-EPSF files with clipping paths

Unlike Mac-EPSF files, PC-EPSF files contain a TIFF screen preview of the respective image (instead of a PICT screen preview).

Note: For the generation of PC-EPSF layouts see 4.2 “Volume settings”.

EPS files and clipping paths

Some layout applications lack full support for TIFF images containing a clipping path. For these applications, EPS layout images should be used (the server default). In this case, you need to specify whether the server image replacement should use the clipping path applied by the image processing application or the path applied by the page layout application. See 3.3.3 “Application clipping paths” for details.

3.3.3 Application clipping paths

A bitmap image file that already contains clipping path information and then has path changes applied in a layout program, may cause interferences and lead to unwanted results. HELIOS Admin offers an option (see checkbox `Ignore Mask for Bitmap Images During Output` in Fig. 3.2) that makes the server ignore existing masks for bitmap images so a new clipping path can be defined from within the layout application (QuarkXPress, InDesign, etc.). This option can also be set for individual folders via folder naming syntax (see 5.2 “Define folder specific OPI settings”).

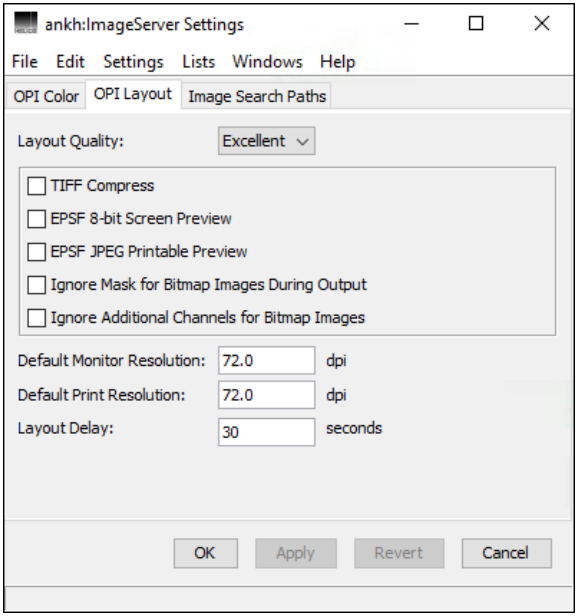


Fig. 3.2: HELIOS Admin `OPI Layout` preferences

Note: Multiple Photoshop paths are migrated into layout and converted files. Photoshop paths can be exported/ imported into existing files, so customers can re-use the images and use different clipping paths for different jobs.

Table 3.6 gives an overview on the compatibility of layout software programs and the used layout file formats which have application paths included.

Layout program	Application path compatibility
InDesign CS1 or newer	TIFF layout, JPEG layout, EPS layout, EPS layout with default application path
QuarkXPress 4-9	TIFF layout, JPEG layout, EPS layout

Table 3.6: Application path compatibility

Use always the folder specific option "%x" (see 5.2 "Define folder specific OPI settings") or check the `Ignore Mask for Bitmap Images During Output` option.

3.3.4 Additional channels in bitmap images

ImageServer recognizes and supports additional channels (e.g. Hexachrome, spot colors or alpha channels). All additional channels will be used as spot color channels.

It is often required to ignore additional channels and use the default printing channels, e.g. CMYK, RGB or CIELab. Many layout applications ignore additional channels in order to be compatible with the printout. Printing from these applications, ImageServer can ignore additional channels (see `Ignore Additional Channels for Bitmap Images` in Fig. 3.2) server-wide or enable/disable this setting on a folder tree (see 5.2 "Define folder specific OPI settings").

3.4 About output devices

PostScript Level 1&2, PostScript 3

ImageServer requires PostScript output devices for printing. There are three types of PostScript RIPs: Level 1, Level 2, and PostScript 3 compatible devices.

The Level 2 and PostScript 3 specifications are follow-ups to the Level 1 specification and provide many more features, as e.g. support for different color spaces such as CMYK, RGB and CIE-based color spaces, and provisions for color matching and transformations into CMYK.

PPD files

Different printers may behave differently and provide different features. PPD files contain comprehensive printer descriptions – with the supported PostScript level being only one aspect. ImageServer (just like any application) refers to the PPD file to get information about the features of the output device. So, for high-quality printouts, always make sure that you have activated a PPD file that properly corresponds to your current output device. For details, see the HELIOS Base manual.

Are PostScript levels relevant to ImageServer?

Usually, if you are printing with ImageServer, our software takes care about the separation into CMYK. The separation modules of the RIP are not required. Thus, print jobs do not depend on the final output device. It does not make any difference whether you are using a Level 1, Level 2 or PostScript 3 printer.

It is, however, possible to leave the separation to the PostScript RIP. In that case, you have to select the Lab D65 color space as output color space (see **Default Printer Profile** in 4.4 “ICC printer settings”) and you must use a Level 2 (or PostScript 3) output device. The RIP will then be initialized with a specific CRD (*Color Rendering Dictionary*) and, using this dictionary, will perform the separation into CMYK. This workflow may be sensible if you are sending your documents (e.g. a magazine) to different production sites. Your print job is not yet separated for a given output device and the Lab D65 color space is device independent. The different RIPs transform the color data into the gamut of the respective printer or press and the output results should all be identical. Please note that this workflow does not allow proof printing (see B “Colors, color matching, proof printing – basic concepts” for details about proofs).

3.5 How ImageServer modifies color modes

If you retouch an image using an image processing application, you also define a color mode for this specific image. This color mode will be preserved as far as the high-resolution original image file is concerned – but it will not be passed on to the layout file and may be changed for printing purposes, i.e. transformed into the color space of the output device.

In the following, we briefly describe how ImageServer handles or modifies color modes and thus relieves you of taking care about it.

Color modes for printing

The color mode OPI selects for printing depends on whether you print composite or separations and whether or not you have activated color matching. The color mode of the high-resolution original image does not matter, unless it is Bilevel or Grayscale – these two color modes are preserved for printing.

Note that only images which are replaced by the OPI server, and which are raster images, may be subject to color space conversion. All other images remain as they are. Whenever a color space conversion is performed, the source ICC profile associated with each image is used. In the absence of a source profile, the server defaults are used (see 4.1 “ImageServer settings” and 9.1 “ImageServer preferences”).

- Printing composite (without active color matching):

  Images that are printed as rastered images are transformed into the color space which is specified (see **CompositeColorspace** in 9.1.5 “OPI-related printer queue preferences” and “psresolve” in 6.5 “psresolve” for details). For vector images, color spaces remain as they are.

- Printing composite (if color matching is active):

  For all images that are printed as rastered images, OPI uses the color space that is specified in the printer profile. For vector images, color spaces remain as they are.

❍ Printing separations (without active color matching):

For all images that are printed as rastered images, the color space is CMYK. For other images, color spaces remain as they are.

❍ Printing separations (if color matching is active):

For all images that are printed as rastered images, OPI uses the color space that is specified in the printer profile. For other images, color spaces remain as they are.

Color modes for layout images

You may select a color mode for the layout files. You can choose between CMYK, RGB, and CIELab (see **Layout Color** in 4.1 “ImageServer settings”). By default, OPI sets the Euroscale CMYK color space for layouts, to allow layout applications to print separations.

3.6 How ImageServer applies color matching

The settings for color matching may be different for every printer queue you have defined on your server. The ImageServer CMM (*Color Matching Module*) is activated by selecting an ICC printer profile for the respective printer queue (see also 4.3 “OPI printer queue settings”).

However, in some specific situations ImageServer performs color matching “behind the scenes”, even though it has not been activated explicitly. These situations are described below (please remember that you must not delete the “ICC-Profiles” volume, because this would disable indirect color matching, which would entirely disable OPI).

The color matching module is active …

❍ … whenever OPI transforms an image into another color space (see also 3.5 “How ImageServer modifies color modes”).

- … for all Lab images, because they always have to be transformed into RGB or CMYK (unless you are working with Lab layouts and explicitly choose “Lab D65” as output profile – compare chapters 4.1 “ImageServer settings” and 4.3 “OPI printer queue settings”).
- … for all Multichannel images (Hexachrome, HiFi Color, etc.). For these images, there are no standard profiles OPI can refer to. Therefore, you are forced to specify an ICC printer profile and thus activate color matching.

ImageServer needs an ICC profile on the output side (printer profile) and an ICC profile on the input side (image source profile) for proper color matching. This means that you need to tag an image profile to every high-resolution image you wish to print with predictable colors, and you have to make sure that your layouts are tagged in case you want to print layouts (see also 5.3 “Tag images with ICC profiles”).

For a general overview of how ICC based color matching works, as well as examples for specific objectives, see B “Colors, color matching, proof printing – basic concepts” and C “ImageServer and color space”.

3.7 How ImageServer finds and replaces image files

Whenever you print a document from a layout application, the OPI software has to locate and find the images that are to replace the low-resolution images you have used for layout purposes. These images, which are required for replacement, may be either high-resolution originals or the layout files again (in case you print your documents with layouts).

If you do not use the layouts at all, but import high-resolution images into your document, there will be no image replacement for printing unless you set the `Replace Images` option to `All` for your current printer queue (see also 4.3 “OPI printer queue settings”).

Important: If you want to use OPI from a Windows system, you should always maintain the 3-character file name extension. The extension may be any extension that is supported by Photoshop for Windows!

The finding strategies

ImageServer offers complex finding strategies for image replacement. The software is able to locate images even after you have moved or renamed them – as long as they remain in the same server volume.

- First of all, OPI will check the folder that contained the image file during layout generation.
- Secondly, the software will check the folder that contains the placed image, ignoring the "layouts" subfolder.
- After that, OPI will check all image search paths you have defined using HELIOS Admin (see 4.1 “ImageServer settings”). These search paths help the OPI software trace images you have moved after layout generation.
- Finally, if a high-resolution file cannot be found with the methods described above, OPI will try to resolve, with the server desktop database, the file ID, which has automatically been saved in the low-resolution file during layout generation. With this method, it is possible to find image files which were renamed and/or moved *in the same volume*. In case the HELIOS volume desktop database is outdated, searching via file IDs may find the wrong image.

Please note that the finding procedures are started automatically during printing. If you want OPI to report missing image files, make sure that the `Check Images` option is active (default) when setting up the print preferences (see Fig. 4.6 in 4.3 “OPI printer queue settings”). Missing images will then stop the print job. With `Check Images` switched off, the documents will be printed, but the printouts will show blanks or layout images where the high-resolution images could not be found.

Note: You may use the HELIOS “Error Queue” feature to save and restart print jobs that have produced an error message (see HELIOS Base manual).

4 Set up ImageServer with HELIOS Admin

The following chapters explain OPI related dialogs. For an overall description of the HELIOS Admin program, please refer to the HELIOS Base manual.

The HELIOS Admin program lets you specify various parameters that are required for using OPI. However, some specific parameters can be set or changed manually using command line commands. For advice on how to handle these commands, please refer to the reference guide in this manual (see 6 “ImageServer utility programs”).

Important: You may not skip the administration before using OPI because, by default, OPI is not active on printer queues and color matching has to be activated explicitly as well.

Note: The following chapters describe the effects single settings will have. In Appendix A “About OPI settings, typical workflows, and conflicts” you can see how different settings influence one another and how the software behaves if application specific OPI settings intervene.

4.1 ImageServer settings

The “ImageServer Settings” dialog contains three tabs in which you can define general OPI settings. These are valid for all layout images that are generated by the server, unless overridden by folder specific or command line settings.

➢ Select `Settings > ImageServer Settings` from the HELIOS Admin menu to open the `OPI Color` tab (Fig. 4.1).

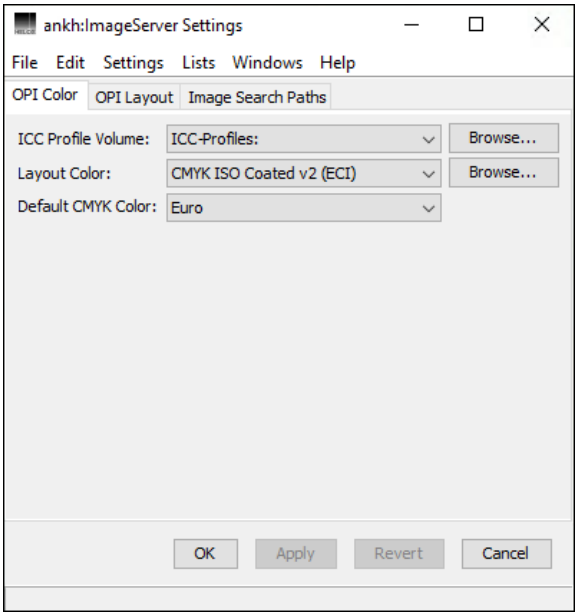


Fig. 4.1: Defining `OPI Color` preferences

Note: The `Revert` and `Apply` buttons remain disabled until you apply changes in the dialog.

ICC Profile Volume

Color matching with ImageServer is based on ICC profiles. All profiles that you want to use must be stored in a central repository. Setting up `ICC Profile Volume` tells the server where to find the profiles.

A pull-down menu shows by default the entry `ICC-Profiles`. It lets you select the volume "ICC-Profiles" that has automatically been created during the software installation. This volume contains all the profiles we have included in our software package.

The `Browse...` button is used to select the current profile repository (in case it is not the "ICC-Profiles" volume).

Layout Color

This item serves to select a color space for all layout files that are generated on the server, and at the same time defines an ICC profile which is used for color matching the layouts.

You can choose a profile via the `Browse...` button, or select one of the four predefined profiles from the pop-up menu:

- `CMYK ISO Coated v2 (ECI)`
  is the default setting and is best-suited if you want to separate your images according to the Euroscale standard.
- `CMYK GRACoL2006_Coated1v2`
  may be selected if you are working with the SWOP standard.
- `sRGB IEC61966-2-1`
  may only be selected if your layout application is able to print composite and separations using RGB images.
- `Lab D65`
  may only be selected if your layout application is able to import Lab layout images.

Default CMYK Color

The CMYK default profile is used whenever OPI has to transform an image from another color space into CMYK or vice versa *and if* – in a situation like that – a CMYK output (printer) or source (image) profile has not explicitly been defined. If you e.g. select `sRGB IEC61966-2-1` from the `Layout Color` menu

and your high-resolution image is a CMYK image, OPI has to perform a transformation from CMYK into RGB during layout generation. If your high-resolution original has no source profile, OPI will use the `CMYK Default Profile`. The pull-down menu lets you select either `SWOP` or `Euro`.

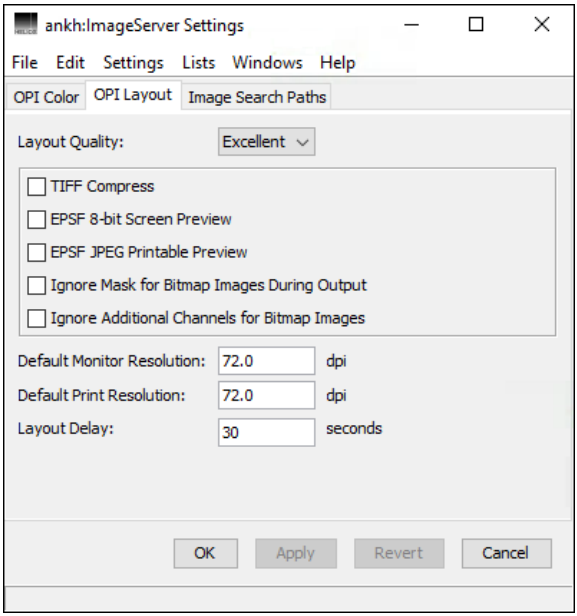


Fig. 4.2: Defining `OPI Layout` preferences

➢ Open the `OPI Layout` tab (Fig. 4.2).

Layout Quality

The quality of the generated image may be either `Excellent` or `Standard`.

❍ `Excellent`
Images are automatically color matched and subjected to a high-quality downsampling method. `Excellent` is the default setting.

- `Standard`
  Images are automatically color matched and generated with a slightly less precise but faster algorithm.

Layout Compression

Layout images do not inherit any compression from the high-resolution originals. You may choose a compression mode, though. There are three possible options which can be – even more than one concurrently – selected from the checkboxes:

- `TIFF Compress`
  Compressed TIFF layout images
- `EPSF 8-bit Screen Preview`
  EPSF screen preview containing indexed colors = max. 256 colors
- `EPSF JPEG Printable Preview`
  JPEG compressed EPSF printable preview

Clipping paths

- `Ignore Mask for Bitmap Images During Output` causes an existing clipping path to be ignored when printing images, and therefore during final high-resolution output (see also 3.3.3 “Application clipping paths”).

Additional channels

- `Ignore Additional Channels for Bitmap Images` causes additional channels (e.g. spot colors) to be omitted when creating a layout image, and therefore during final high-resolution output (see also 3.3.4 “Additional channels in bitmap images”).

Default Monitor Resolution

This item serves to define the default monitor resolution for your layout files. Monitor resolutions usually have a range between 72 and 200 dpi.

Default Print Resolution

EPSF files contain a monitor preview and a printable preview of the respective image. It is therefore sensible to allow you to define for your layout images two different dpi values. You can e.g. enter 72 dpi monitor resolution and 96 dpi print resolution if you print your documents with EPSF layout images and want to achieve a printout quality better than 72 dpi.

Note: For files that have only one preview (for printing and monitor display) the OPI software compares the `Default Monitor Resolution` to the `Default Print Resolution` and uses the higher value as layout resolution. This applies e.g. to TIFF images.

Layout Delay

In this field you may define the time in seconds before the layout creation process is started by the OPI server. The purpose is to prevent layout file generation from temporary files or files that are in use.

Important: Please note that all above mentioned server settings are only applied to future layout files. If you want to update any existing layouts, you must re-generate them. For that purpose you can either open and save the high-resolution original image file once again, use the HELIOS “touch” program (see 5.5 “Refresh layouts using the “touch” program”), or use the “opitouch” program (see 6.2 “opitouch”).

➢ Open the `Image Search Paths` tab (Fig. 4.3).

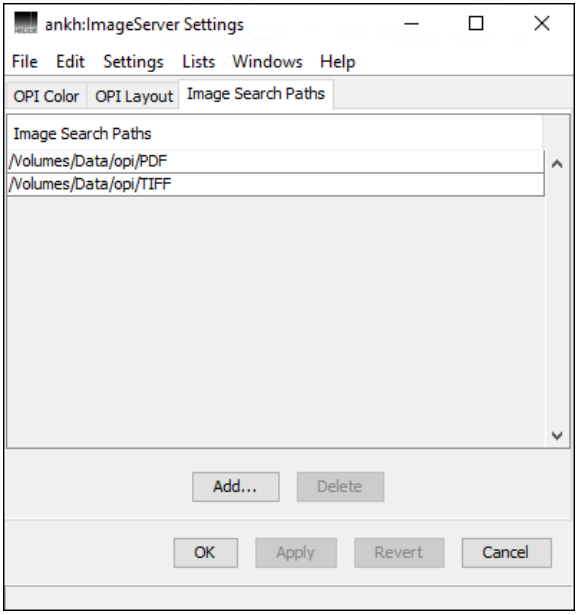


Fig. 4.3: Defining `Image Search Paths` preferences

Image Search Paths

In some situations you may want to move your image files to another volume or folder. You can help OPI find these images for printing if you define search paths that lead to the locations you usually use.

There are, in principle, no limits as far as the number of search paths is concerned. However, you can save time if you make sure that the locations you use frequently are checked first.
Please note that no subdirectories are searched for images.

The “Server Settings” window shows the image search paths that are already defined. You can delete any of these paths or add new ones. The `Add...` button

opens the dialog that serves to select new paths (Fig. 4.4).

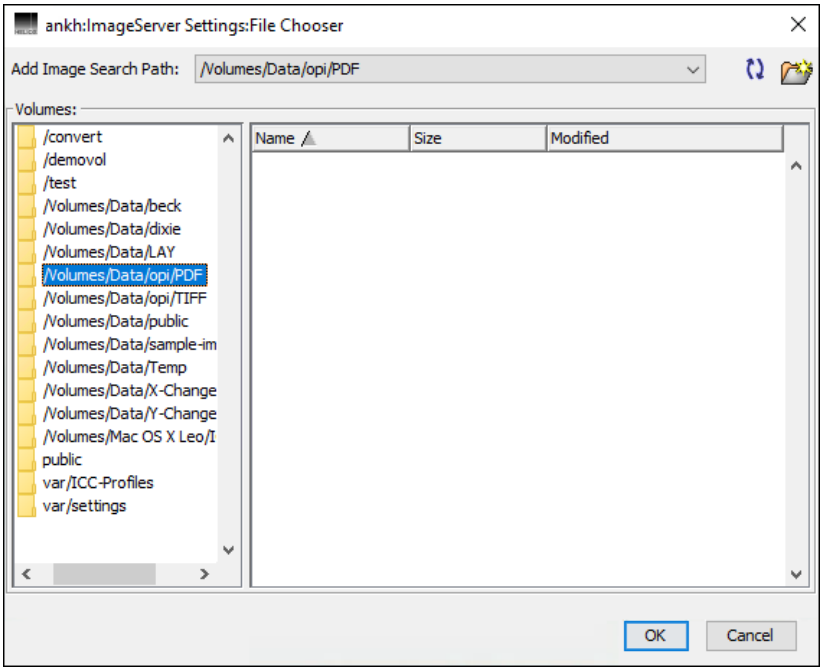


Fig. 4.4: Defining image search paths

Note: Detailed explanations on how ImageServer deals with image search paths are given in 3.7 “How ImageServer finds and replaces image files”.

4.2 Volume settings

Some OPI settings are volume-dependent. To edit the settings for a specific volume do the following:

➢ In HELIOS Admin open the `Volumes` tab and highlight the volume you wish to configure. Then open the volume settings and select the `OPI` tab (Fig. 4.5).

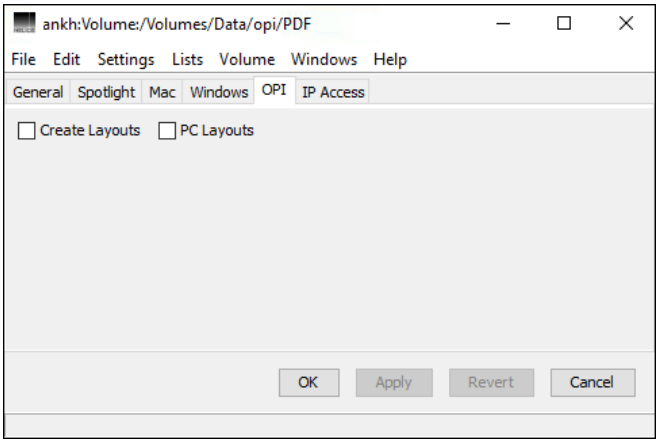


Fig. 4.5: Defining volume-dependent OPI settings

Create Layouts

This option controls the automatic generation of layout files. `Create Layouts` will make the OPI server generate a low-resolution layout file of each image file you save to this volume. The generation of layouts will take a few moments, it starts after a user-defined time delay. By default this is a delay of 30 seconds (see **Layout Delay** above in this chapter).

Important: When copying OS X packages, e.g. application packages or RTFD documents, to a HELIOS volume with automatic layout generation active, layouts of all images within the package will be generated. Though this will not corrupt the package it can considerably enlarge the package size.

PC Layouts

File characteristics are slightly different on Mac and Windows computers. Switch `PC Layouts` on if you want OPI to always generate PC-compatible layouts – even if the original high-resolution images are coming from a Mac computer; OPI generates layout files always according to the `PC Layouts` checkbox in the HELIOS Admin volume settings (Fig. 4.5).

Note: Since the Mac is also able to recognize PC-compatible layouts, make sure to always have checked the `PC Layouts` checkbox when working in a volume shared with PCShare (see also appendix F “Notes for PCShare users”).

Please note that working with `PC Layouts` would mean that if OPI generates EPSF layouts, these layouts will be PC-EPSF files with TIFF previews (instead of PICT previews).

4.3 OPI printer queue settings

Some OPI settings are only meant for printing, and have to be set up for every single printer queue.

Note: It may be helpful to define an “Error Queue” for your OPI printer queue. This allows you to save print jobs that have produced an error message (e.g. because of missing image files). After solving the problem, the print jobs can be started again directly from the error queue. For details refer to the HELIOS Base manual.

➢ Highlight a printer from the `Printers` list and then select `Printer > Settings` from the HELIOS Admin menu and go to the `OPI` tab to open the dialog shown in Fig. 4.6.

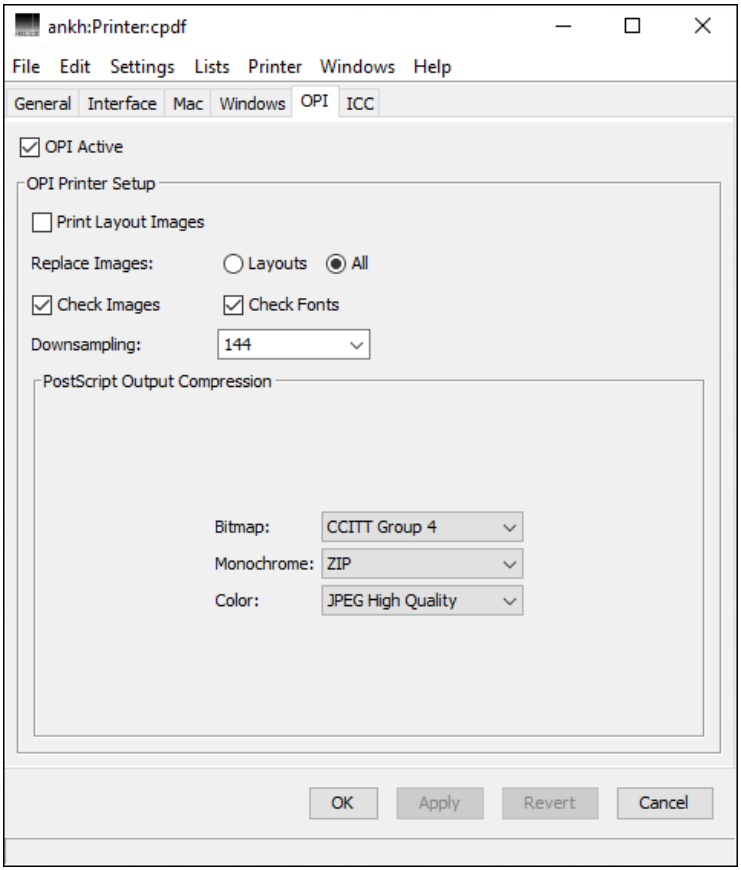


Fig. 4.6: Defining OPI settings for a printer queue

OPI Active

The `OPI Active` option enables OPI for this specific printer queue, and at the same time activates the "OPI Printer Setup" section.

Print Layout Images

With `Print Layout Images` enabled, printouts from this specific printer queue will only contain layout images. The print job will be sent to the OPI server where the OPI comments of the placed layouts are resolved, but no image replacement is performed.

Replace Images

`Replace Images = Layouts` is the default resolve option. If this setting is active, the OPI server will only replace layout images during printing. High-resolution originals you may have placed in your document, will not be replaced. In case you want the OPI server to replace all images during printing (e.g. if you want to use the "Omit ..." options print dialog) you should switch to `Replace Images = All` (see also 5.6 "OPI print options – Printing from QuarkXPress and InDesign").

Check Images

The `Check Images` option will induce ImageServer to stop the print job and issue an error message if the current print job contains any images that are not available. The strategies the OPI server uses for locating images are described in 3.7 "How ImageServer finds and replaces image files". If `Check Images` is not active, single images may be missing in the printouts.

Check Fonts

The `Check Fonts` option will induce ImageServer to stop the print job and issue an error message if the current print job contains any fonts that are not available.

Downsampling

The `Downsampling` item serves to define a uniform output resolution for all raster-based images – no matter whether you print layout images or high-resolution originals. If the resolution of a given image is higher than the

value in this dialog (144 dpi in Fig. 4.6), the OPI server will apply the value you have chosen for downsampling and then re-calculate and scale the image accordingly. Images with a resolution smaller than the value entered in this dialog, will remain unchanged because increasing the resolution automatically decreases the dimensions of an image. This would entail upward scaling and lead to a loss in quality.

The Downsampling pull-down menu contains a number of dpi values and the PPD Default option. If you select PPD Default, ImageServer will use the resolution that is stated in the PPD file. You may also specify a custom value for downsampling from the pop-up menu. Selecting None from the list will switch downsampling off. Every image will then be printed according to its particular resolution.

PostScript Output Compression

There are efficient compression methods you can apply to the high-resolution images. Depending on the image format there are different types of compression that can be selected from the following pull-down menus. This considerably reduces the size of the PostScript printing data:

Bitmap
For "bitmap color mode" (bilevel) images, the CCITT Group 4 compression is a very good lossless linework compression. It can easily reduce linework file sizes by the factor of 50. It can be used with PostScript Level 2 or higher printers.

Monochrome/Color
The ZIP compression, which is compatible with PostScript 3 compatible printers, is the best lossless continuous tone compression. Compress works with PostScript 2 and 3 and has good compression results. JPEG (Low Quality, Medium Quality, High Quality, Maximum Quality) compression will always lose some image details but offers the best data reduction, it is supported with PostScript Level 2 or higher.

4.4 ICC printer settings

➢ Highlight a printer from the `Printers` list and then select `Printer > Settings` from the HELIOS Admin menu and go to the `ICC` tab to open the dialog shown in Fig. 4.7.

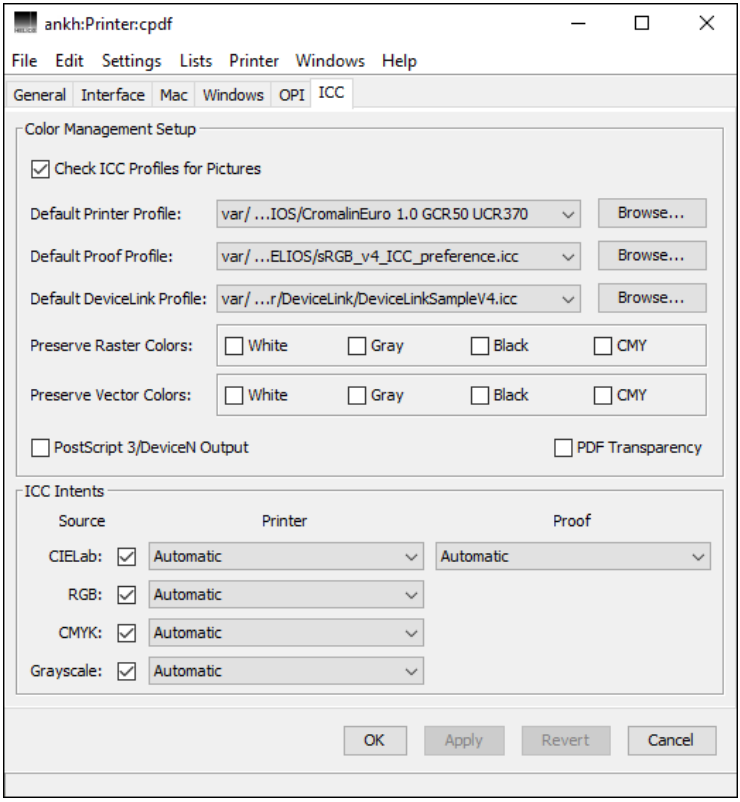


Fig. 4.7: Defining ICC settings for a printer queue

Check ICC Profiles for Pictures

Activating `Check ICC Profiles for Pictures` (the `Check Images` OPI option must be activated, otherwise this option has no effect!) induces the OPI server to check whether all image profiles that have been tagged to any of the image files are available. The server will automatically stop the print job if a single profile is missing and issue a corresponding error message. If this option is not active, OPI will execute all print jobs and use standard profiles for color matching, whenever the correct profile is missing, unless the input color space is the same as the output color space, e.g. Image CMYK = Printer CMYK.

Note: Grayscale images without ICC profile information remain unaltered. Grayscale images with profile information are matched to the printer profile (e.g. CMYK). Grayscale images are ignored by the `Check ICC Profile for Images` option.

Default Printer Profile

The `Default Printer Profile` item lets you choose the correct profile for your output device and, at the same time, serves to switch on color matching for this printer queue. The pull-down menu contains four different options.

- `None`
  Switches color matching off.

- `sRGB IEC61966-2-1`
  May be selected if you want to output an RGB document.

- `Lab D65`
  Best-suited if you want to select a device independent output color space, e.g. if you send your documents to different production sites (see also 3.4 “About output devices”).

- `Composite - Preserve Colors`
  No color matching for composite print jobs; all colors are preserved.

The `Browse...` button opens a dialog that lets you select a specific profile.

Default Proof Profile

This item becomes only available if you have selected a default printer profile. Here, you can select the correct profile for your proof printer.

Default DeviceLink Profile

This pop-up menu allows you to specify a default DeviceLink profile for color matching. DeviceLink profiles have a higher priority than standard ICC profiles, i.e. standard ICC profiles are overridden by DeviceLink profiles. If a CMYK DeviceLink profile is specified, all CMYK images are converted using this profile. If CMYK images have a tagged profile it will be ignored and the DeviceLink profile is used instead.

Preserve Raster Colors

Exclude `White`, `Black`, `Gray` or `CMY` colored *raster* objects from color transformation, so that they preserve the selected color(s), even if color matching is done. For example, gray raster image pixels in arbitrary color space are detected and converted to Gray only for CMYK output, if the `Gray` option is checked.

Preserve Vector Colors

Exclude `White`, `Black`, `Gray` or `CMY` colored *vector* objects from color transformation, so that they preserve the selected color(s), even if color matching is done. For example, gray text and other gray vector objects in arbitrary color space are detected and converted to Gray only for CMYK output, if the `Gray` option is checked.

PDF Transparency

This option is PDF related and hence described in the HELIOS PDF HandShake manual.

PostScript 3/DeviceN

The `PostScript 3/DeviceN Output` checkbox should only be selected if you print to a PostScript 3 device with in-RIP separation. The PostScript output of the original files will be optimized to use the PostScript 3/DeviceN features. This applies to Hexachrome printing, colorized images (Duotone), etc. DCS-2,

TIFF, PDF, and Photoshop native files containing spot colors are converted into DeviceN. When printing host-based separations with applications like QuarkXPress, this feature must be turned off, otherwise the output will lead to unexpected results.

ICC Intents

This feature offers different "strategies" for rendering an image file from one color space to another. The following settings are available:

```
1) Automatic
2) Perceptual
3) Relative Colorimetric
4) Saturation
5) Absolute Colorimetric
6) Perceptual with BPC[1]
7) Relative Colorimetric with BPC
8) Saturation with BPC
9) Perceptual with Black Plane Compensation
10) Relative Colorimetric with Black Plane Compensation
11) Saturation with Black Plane Compensation
```

Please make sure that you have assigned a printer profile to the printer queue according to the instructions given in **Default Printer Profile** above. This ensures that the values you specify in the "ICC Intents" section are used for printing.

Note: Please keep the intent setting `Automatic`, unless you have a special reason to change it. Your profiling software will explain the intents meanings.

---

[1] Black Point Compensation

4.5 Using the HELIOS Admin printer log file to check OPI activities

You may use the HELIOS Admin program for both setting up OPI (before printing) and proofing OPI (after printing). In fact, the printer log files contain comprehensive information about how OPI has handled the images included in a given print job.

➢ Select `Printer Log Files` from the `Lists` menu. Then click on the print job on which you wish to obtain OPI-related information.

Another possibility is to make a double-click on the print job in the "Hold queue" (Fig. 4.8).

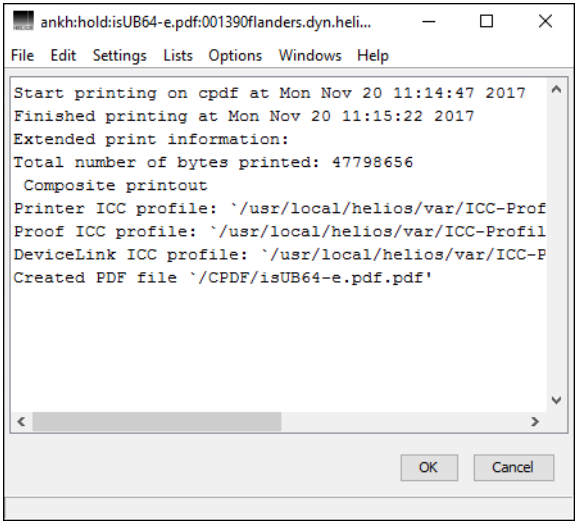


Fig. 4.8: Contents of a printer log file

The log file will contain various messages that describe the respective print job. Some of the messages are OPI-specific. The OPI-related messages in this file inform you about:

- the name of the layout images that have been included in the document
- the location and name of the high-resolution original images that have replaced the layouts
- the number of printed separations
- the name of each separation and related page number

For further information about HELIOS log files, please refer to the HELIOS Base manual.

5 Using ImageServer

The preceding chapter described how to set up ImageServer volumes and printer queues via HELIOS Admin. This chapter gives notes on image layout generation, and describes folder-specific OPI settings, tagging images with ICC profiles, the export/import of metadata such as XMP information, the "touch" program, which allows refreshing layout images, print options for printing from QuarkXPress and InDesign, and how to view HELIOS XPPV documents.

5.1 Notes about layout generation

The automatic generation of layout files is volume dependent. If you have retouched your high-resolution original image files using an image processing application, make sure that you save them in a volume on which `Create Layouts` is enabled (see also 4.2 "Volume settings"). OPI will not generate layouts for files that you save on your local hard disk.

Important: You should always select unique file names for your high-resolution original images, at least if you want to save them in the same folder! Different extensions will not do. For example, if you save "flowers.psd" (Photoshop) and "flowers.tif" in the same folder, they both will produce a "flowers.tif" layout image. The contents of this layout may show either the content of "flowers.psd" or "flowers.tif".

The generation of layouts will take a few moments, it starts after a user-defined time delay. This is – by default – after a delay of 30 seconds (see **Layout Delay** in 4.1 "ImageServer settings") before the OPI server starts generating them.

Layout files are generated in a “layouts” subfolder. ImageServer always generates layouts, even if the resolution of the original image is lower than the default layout resolution. In this case the layout would inherit the resolution of the original.

5.1.1 Notes for Photoshop users

Since Photoshop 6, layers can be vector-based. So you should consider before saving your high-resolution originals, that not every image format can handle such layers. The file format of the high-resolution image determines whether the vector data (not the vector layer!) in the document is maintained in the layout file, although the preview in the Finder may show it anyway.

It is recommended to stick to the Photoshop native (PSD) format when vector-based data is employed within the document.

When saving the file to EPS-format Photoshop, the following warning is issued:

```
Files with vector data will be rasterized by Photoshop when opened.
```

Note: You can switch off the `Include Vector Data` option in the `EPS-Options` dialog. Then the layout file, generated by ImageServer, will contain the (rasterized) text.

We recommend to activate the option `Maximize PSD File Compatibility` in the Photoshop preferences in order to provide a better compatibility with ImageServer.

5.2 Define folder specific OPI settings

Usually, all OPI settings that define the attributes of layout files are server or volume specific (see Fig. 4.1 in 4.1 “ImageServer settings” and Fig. 4.5 in 4.2 “Volume settings”). It is, however, possible to specify alternative settings for individual folders, for automatic layout generation on the OPI server, by simply changing the folder name, e.g. from a Mac or Windows computer.

The “%” character always introduces a new specification. You can add different definitions following that character, including: resolution, file format, and color space.

Resolution

A 1, 2 or 3 digit dpi value for the resolution of the layout files that are to be generated within this folder. `%0` means that no layout is generated at all!

File format

The `t` flag will produce TIFF layouts even if the high-resolution image contains a clipping path or spot colors. If you want to use TIFF layout files for multi-path handling, do not use high-res EPSF files with a clipping path (see also **MaskToEps** and **SpotToEps** in 9.1.2 “Global OPI preferences”). An EPS image is special since the “opisrv” does only check the raster-based information of the image and does not look in additional object/vector based image parts. ImageServer supports both vector-based and raster-based EPSF files.

Note: As a workaround, you may use the “layout” command (see 6.1 “layout”) to create TIFF layouts from an EPSF image.

For other layout file formats use the following flags:

`%t` TIFF (and enforce raster layout images)
`%e` EPS
`%j` JPEG (and enforce raster layout images)
`%g` PNG (and enforce raster layout images)
`%n` Neutral

The `%n` flag causes the folder to use the OPI server settings, ignoring all folder specific OPI settings of any parent folder.

Platform

`%m` Mac only
`%p` Cross-platform/PC

Example:

`Folder_name%80em` will produce 80 dpi Mac EPSF layouts.

Color space

`%b` Grayscale
`%r` RGB
`%c` CMYK
`%l` Lab

Specify the color mode of the printable and screen previews. If you use `l`, TIFF layouts and the printable previews of EPSF layouts will be in Lab, the screen previews of EPSF layouts will be in the RGB color space.

General

`%u` Ignore spot colors
`%v` Enable spot colors
`%x` Ignore masks
`%y` Enable masks

Important: The chosen parameters in a folder name affect all subfolders, i.e. they are automatically valid for the complete directory tree!

Folder name syntax

The folder name must end in %, followed by none or more digits for the resolution, and by the other options in the order as listed below. Options are not case-sensitive. Other characters or spaces in the options string are not allowed.

1. *Resolution (in dpi)*
2. *e, t, g, j, n*
3. *m, p*
4. *b, c, r, l*
5. *u, v*
6. *x, y*

Example:

The appended "%0" of the "test-images%0" folder will prevent the OPI server from creating layout images from the high-resolution originals saved to that folder.

The folder named "test-images%36pb" is for high-resolution images you want to substitute with 36 dpi layouts containing PC previews in Grayscale color mode for printable and screen EPSF and TIFF previews.

5.3 Tag images with ICC profiles

Note: See the client requirements for HELIOS ICC Tagger in chapter 4.1.3 "Client requirements" of the HELIOS Base manual.

For high-quality color matching, the OPI server needs an ICC source profile (image profile) for each image and an ICC output profile (printer profile) for the printer queue.

HELIOS ICC Tagger serves to tag high-resolution original images with an ICC source profile and thus allows correct color data transformation (for details

about color data transformation with ICC profiles see Fig. B.1 in B “Colors, color matching, proof printing – basic concepts”).

Please note that for tagging both the images and the profiles have to be available on the server. The volume in which the tagging will be done must have enough disk space available – at least as much space as the largest image would need, because “opisrv” might temporarily duplicate data.

➢ Mount the “HELIOS Applications” volume, open the “Windows” folder (Mac: “MacOS”) and then “ImageServer Tools”. Double-click the “HELIOS ICC Tagger” program icon.

The HELIOS ICC Tagger program allows creating so-called ICC info files that contain tagging instructions for all images in a given folder or tagging single images individually.

➢ Select `File` and `New` from the program menu. If HELIOS ICC Tagger issues an error message that the profile repository (by default this is the “ICC-Profiles” volume) cannot be found, mount either the volume or, if your profiles are stored in another volume, change the profile search path by clicking the `Change` button and specify the correct path.
Confirm with `OK`.

Creating an ICC info file

The dialog (Fig. 5.1) lets you set general tagging options as well as specific options for all the different file formats that are supported by ImageServer.

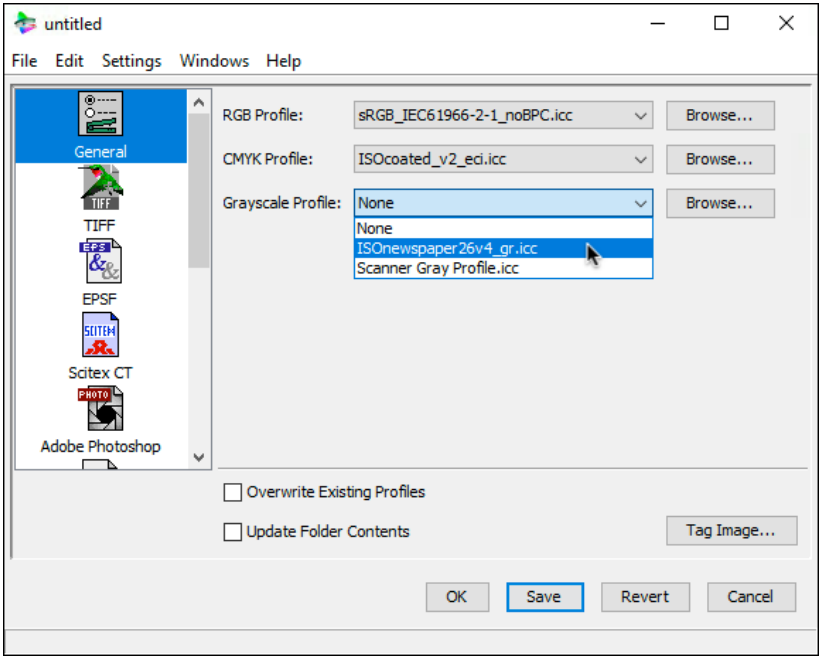


Fig. 5.1: Setting general options for a new ICC info file

You can define separately for each single file format (TIFF, EPSF, etc.) whether you wish to tag the profiles by reference or embed them into the image file.

➢ In the corresponding pop-up menu select an RGB, CMYK, and Grayscale profile for the specific file format.

The `Overwrite existing profiles` checkbox should be used carefully. Whenever you save an image that already contains an image profile to the folder that contains this particular ICC info file (or if you move the image to that folder) the old profile will be overwritten.

By selecting `Update Folder Contents`, current ICC profile settings are applied immediately to the images in the respective folder (allow “opisrv” some time to apply the changes).

After setting the “General” options (Fig. 5.1), which apply the chosen CMYK, RGB, and Grayscale profile for different file formats, individual profiles can be assigned in the same way for specific file formats, which can be selected from the scroll list at the left. Selecting `General` from the pop-up menu means that the profiles which you have specified in the “General” section are applied. Fig. 5.2 shows an example for the TIFF file format.

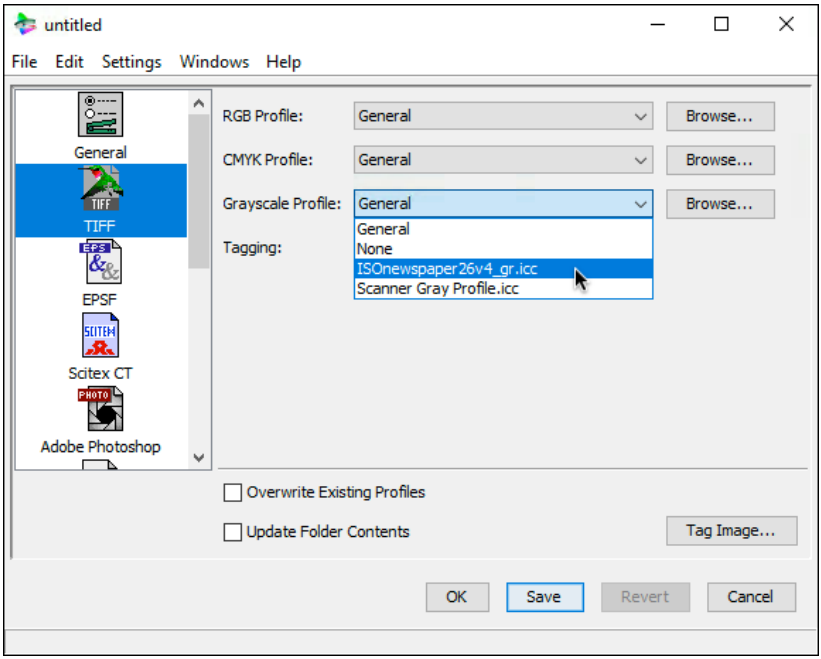


Fig. 5.2: Setting “TIFF” options for a new ICC info file

Note: `General` is the default setting in the pop-ups for the different file formats unless the file format denies tagging of a particular profile type, e.g., PICT images cannot be tagged with a CMYK profile.

Saving an ICC info file

When you have set up tagging for all required file formats, you can save your new ICC info file.

➢ Click the `Save` button to open the dialog that lets you select the destination folder (Fig. 5.3).

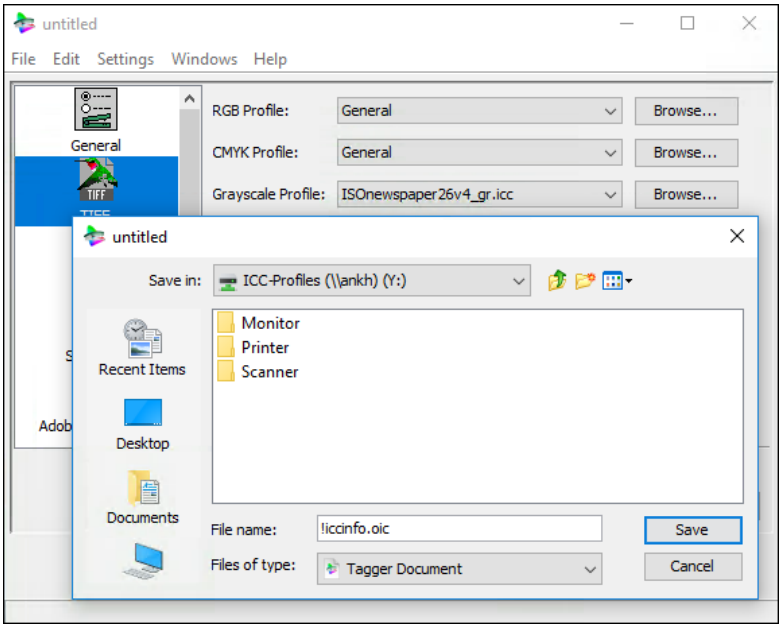


Fig. 5.3: Saving an ICC info file

Please note that there are a few important things you should know about saving:

- The file name for an ICC info file is "!iccinfo.oic". This name must not be changed.
- The tagging instructions that are contained in the file are applied to images in the destination folder only – they are not valid for subfolders. But: if you save an ICC info file in the volume root directory, it will be valid for all folders in the volume that do not contain their own ICC info file.
- Images that were already stored in the destination folder are not tagged automatically. Tagging is only applied to images coming in later. If you want to tag existing image files, the `Update Folder Contents` checkbox must be active (as already described in **Creating an ICC info file** above).

The ICC info file can be re-edited at any time. Just double-click the file icon to open the HELIOS ICC Tagger.

Note: If you have added your own image profiles to the "ICC-Profiles" volume issue the following command to update the ICC index file, in order to notify the OPI server about the changes:

```
# srvutil reconf opisrv
```

Tagging single images

You may also tag single images one by one with the HELIOS ICC Tagger program. The procedure is similar to that described above.

➢ Select `File` and `New` from the program menu to open the ICC info file dialog.

➢ Select the file format of the image you want to tag and choose the correct profile. Then click the `Tag Image` button to open the dialog that lets you choose the desired image file.

The HELIOS ICC Tagger program internally creates an ICC info file which is used for tagging the images and then deletes it afterwards.

About tagging layout images

The HELIOS ICC Tagger program may only be used for tagging high-resolution original images. If you happen to tag a layout image this will be silently ignored because layouts are tagged automatically by the OPI server. The rules for layout tagging are as follows:

- If the layout inherits the color space of the high-resolution image (e.g. CMYK) it also inherits the profile. For a high-resolution image without a profile the layout will not have a profile either.
- If the high-resolution image and the layout have a different color space the layout will be tagged with the profile that has been used by the OPI server for color data transformation. See also **OmitProfile** in 9.1 “ImageServer preferences”.

Note: For details about color spaces of layout images refer to 3.5 “How ImageServer modifies color modes”.

Information about tagged profiles

There are two methods to find out about profile information that has been tagged to an image file with the HELIOS ICC Tagger program:

➢ Select `File > Open Tag Info`. Then drag and drop the file(s) you want to get profile information about into the window.

Another possibility is to select `Open` from the `File` menu, navigate to the image and highlight it and then click Open. The “Tagging Information Window” (Fig. 5.4) lists the file(s) and gives information about possible profile tagging and whether the profiles are referenced or embedded.

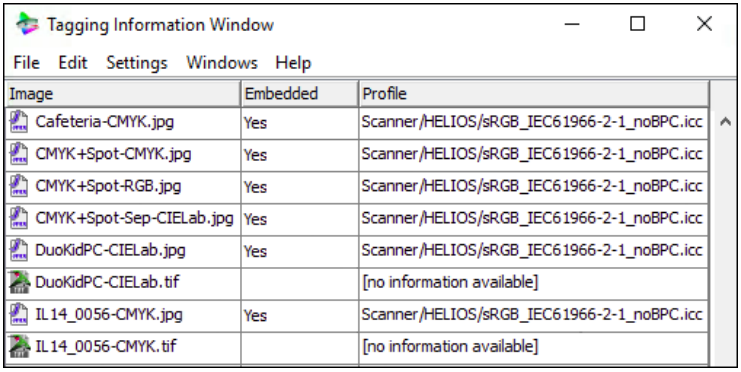


Fig. 5.4: “Tagging Information Window”

You may also use our “oiimginfo” program to find out whether a specific image is already tagged with a profile. See 6.4 “oiimginfo”.

5.4 Export/Import XMP etc. information

An image file can contain XMP (*Extensible Metadata Platform*), metadata or clipping path information. ImageServer enables users to export this information from a file, and likewise, to import such information into an image file.

5.4.1 Export

➢ Specify “oiimginfo” for an image file with the `-f` option and an export file name (See also `-E <ExportType>` in 6.4 “oiimginfo”).

Example:

```
$ cd /usr/local/helios
$ bin/oiimginfo -E XMP -f /demovol/Info.xmp image.jpg
```

ImageServer will create a file in the specified directory (in the example above in “/demovol”), which contains the extracted XMP information.

5.4.2 Import

➢ Make sure the XMP file is UTF-8 coded.

➢ Specify “layout” with the `-f` option (see 6.1.11 “Tagging ICC profiles, path information, and metadata to images”).

Example:

```
$ cd /usr/local/helios
$ bin/layout -f "XMP:ADD:/demovol/Info.xmp" image.jpg
```

5.5 Refresh layouts using the “touch” program

The “touch” program is meant to help you generate new layouts. This might be necessary either …

❍ … if you want to replace layouts that have been generated with an earlier ImageServer version and thus make them fully compatible with the latest program version, or

❍ … if you have changed any OPI settings (using the HELIOS Admin program) and now want to refresh your existing layout files accordingly.

Usually, a layout file is generated shortly after you have saved a high-resolution original image on the server. The layout file will then bear the attributes that

are defined at that moment in HELIOS Admin. If you change settings in the Admin, the existing layouts will not be updated automatically.

In case you decide to update an old layout, you could open your high-resolution original image file and save it once again. Or, you can use the “touch” program for that purpose. This is much more convenient because the program simulates the saving procedure without opening any image processing application.

➢ Mount the “HELIOS Applications” volume, open the “Windows” folder (Mac: “MacOS”) and then “ImageServer Tools”.

➢ Drag the “touch” program icon onto your local desktop.

➢ Drag the image file(s) you want to re-layout onto the “touch” icon (Fig. 5.5).

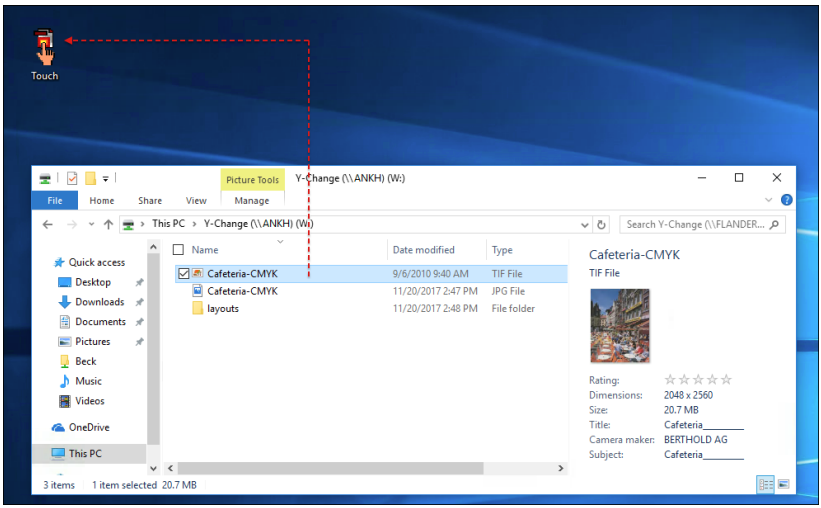


Fig. 5.5: “touching” an image

The “touch” program will – after a delay of 30 seconds (default) – automatically lead to the generation of new layout files.

Note: You may select several images at a time and drag them onto the “touch” icon. Likewise, you can drag a complete folder onto the “touch” icon. However, this can lead to a high server load on the host, depending on the number of files in that folder because its content will then be processed recursively.

Please note that the time the server needs for the generation of new layouts, depends on the number of layout jobs and processes.

The “touch” program may also be used for tagging ICC profiles to high-resolution images. This is done by “touching” a complete directory which contains high-resolution images and an “!iccinfo.oic” file. Due to the fact that the “Touch.exe” program enforces a new layout process in the directory, existing source profiles are read out anew according to the information stored in the “!iccinfo.oic” file.

5.5.1 Using “Touch.exe” under Windows

Under Windows, only a certain amount of files can be dragged onto the “Touch.exe” icon at a time, otherwise Windows will issue an error message.

In this case do the following:

➢ Start “Touch.exe” and drag all selected files into the program window. You may also copy the selected files in the Explorer and paste them into the program window.

5.5.2 opitouch

“opitouch” is the command line version of this program. It is described in 6.2 “opitouch”.

5.6 OPI print options – Printing from QuarkXPress and InDesign

OPI options in the print dialog are application specific. In the following, we explain how to specify OPI settings in the QuarkXPress and InDesign applications, and the effects on the printing results.

5.6.1 QuarkXPress

For example, QuarkXPress offers the `Include Images` option (Fig. 5.6), as long as `OPI Active` is checked.

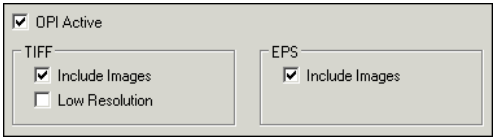


Fig. 5.6: OPI print options (QuarkXPress 6)

Include Images

With this option selected, the PostScript job which is sent from the workstation to the OPI server will include the image data and all OPI comments that describe these images. Once the job has arrived on the server, ImageServer will replace the image data with the high-resolution originals – or print the

layouts that have been placed (see **Print Layout Images** in 4.3 “OPI printer queue settings”).

If `Include Images` is not selected, the amount of data that is sent from the client to the server will be reduced. The PostScript file will no longer contain the image data of the layout files. However, the OPI comments are still included. In this case the OPI server does not *replace images*, but *fills blanks*. The printouts will be complete and correct.

It may be reasonable to deactivate this `Include...` option whenever documents contain a lot of TIFF or EPS images and thus become rather large (even with layout files).

Important: Only deactivate the `Include...` options if all your images are stored on the OPI server. This works properly if you are using layout files only. In case you place high-resolution originals in your layout document, this might cause problems, because the OPI server, by default, does not replace high-resolution images nor fills blanks either. If you deactivate one of the `Include...` options for high-resolution images, these images will be missing on the printout.
You can solve this problem by switching the `Replace Images` option to `All` on ImageServer – see 4.3 “OPI printer queue settings”.

Low Resolution

You may additionally select `Low Resolution` to downscale high-res TIFF images in the document when printing out.

5.6.2 InDesign

When printing a document from InDesign, the OPI options in the “Advanced” section let you selectively omit different imported graphics types when sending image data to a printer or file, leaving only the OPI links (comments) for later handling by an OPI server. This behavior is almost the same as that described in 5.6.1 “QuarkXPress”.

OPI Image Replacement

If the checkbox `OPI Image Replacement` (Fig. 5.16) is active, InDesign replaces low-resolution EPS images with high-resolution graphics at output time. You must deselect this option to have ImageServer perform server-based OPI image replacement.

Omit for OPI

These options allow you to selectively omit different imported graphics types (`EPS`, `PDF`, and `Bitmap Images`) when sending image data to a printer or file, leaving only the OPI comments for later handling by ImageServer OPI. The comments contain the information needed to locate a high-resolution image in a HELIOS OPI volume. InDesign only includes the comments in the print job. Please note that the `Omit for OPI` option does not apply to embedded graphics.

5.7 Using QuarkXPress with ImageServer

ImageServer supports background tinting of Bilevel and Grayscale raster-based images that are used by QuarkXPress. To benefit from this functionality you must first copy the "HELIOS OPI TuneUp XT" module to the QuarkXPress "XTension" folder:

➢ Mount the "HELIOS Applications" volume, open the "Windows" folder (Mac: "MacOS"), then "ImageServer Tools" and "HELIOS OPI TuneUp".

➢ Then copy the "HELIOS OPI TuneUp XT" file from the appropriate folder (e.g. "XPress 8") into the "XTension" folder of your QuarkXPress application.

Now the installation of the module is complete and you can immediately utilize it after a restart of QuarkXPress. Once you have designed your QuarkXPress document, you can view information about the images you have placed in it:

➢ For that purpose, select `Utilities` and `Usage...` and then the `Pictures` tab from the QuarkXPress menu.

The "Usage" window will provide details you might need for setting up the print dialog for OPI. It tells you e.g. whether you have imported layout files only (this is indicated by the path and file name) and what kind of file formats you are dealing with. Fig. 5.7 shows an example document with the "Usage" window.

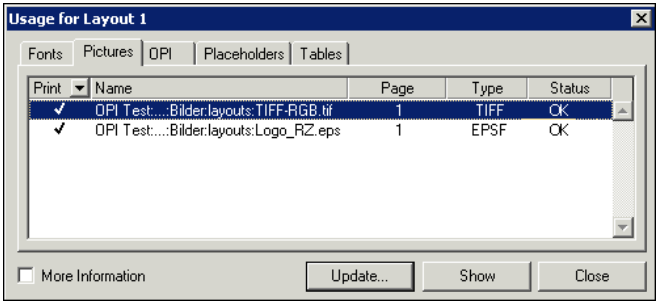


Fig. 5.7: Prompting information about included images

5.7.1 Print settings in QuarkXPress

"Layout" tab

In the "Layout" tab of the print dialog you can specify whether to print composite or separations (Fig. 5.8). If you switch on `Separations`, please note that – in case you have selected a particular ICC printer profile for your printer queue – only CMYK and Multichannel (e.g. Hexachrome) output profiles cooperate with this option. Other profiles do not allow printing separations and will cause error messages. In case you get such messages, please check the printer queue settings in the respective HELIOS Admin dialog (4.3 "OPI printer queue settings").

Printing separations may fail if you have placed RGB high-resolution images in your document or if you have selected `sRGB IEC61966-2-1` as default color space for layouts (4.1 “ImageServer settings”). This restriction is only caused by older QuarkXPress versions, because the software does not separate RGB images (even if the high-resolution originals are CMYK).

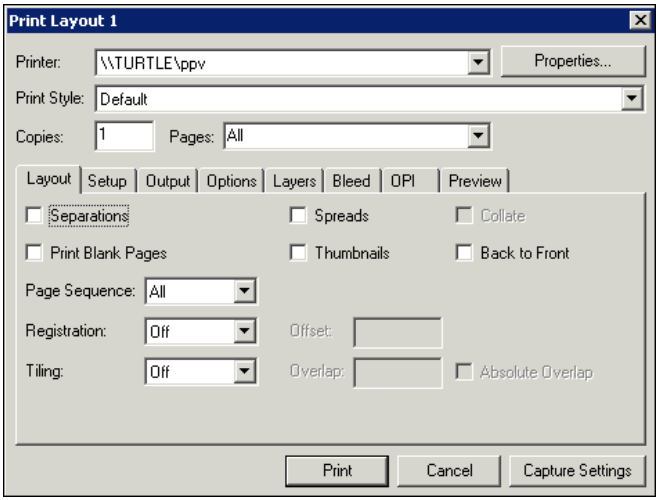


Fig. 5.8: “Layout” tab in QuarkXPress “Print” dialog

“Setup” tab

The “Setup” tab lets you check paper settings for the document and select a printer type.

➢ Select the printer in the `Printer Description` pull-down menu (Fig. 5.9). `Generic Color` may be selected if your printer type is not stated in the list and you plan to print colored images, but *do not* select `Generic B&W`.

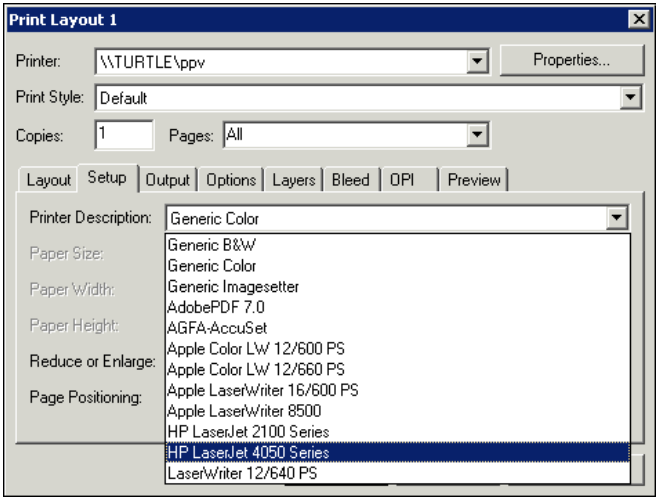


Fig. 5.9: “Setup” tab in QuarkXPress “Print” dialog

“Output” tab

In the “Output” tab (Fig. 5.10) you may select printer settings for your print job.

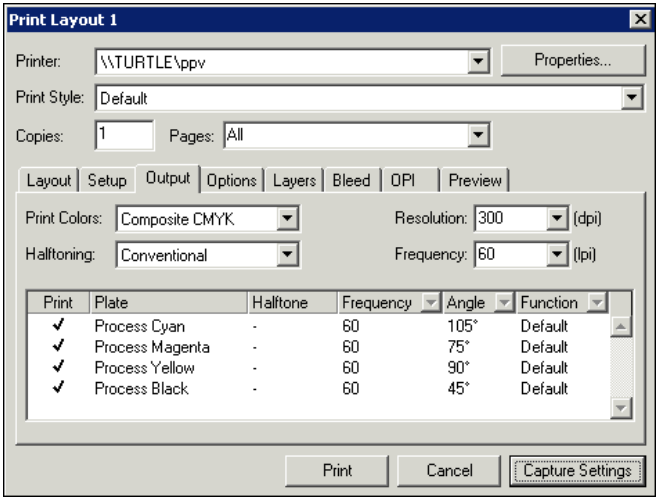


Fig. 5.10: “Ouput” tab in QuarkXPress “Print” dialog

“OPI” tab

The “OPI” tab lets you specify whether to include the images of the document in the print job or not. This can be set for TIFF and EPS images separately (Fig. 5.11).

Details about these settings are described in 5.6 “OPI print options – Printing from QuarkXPress and InDesign”.

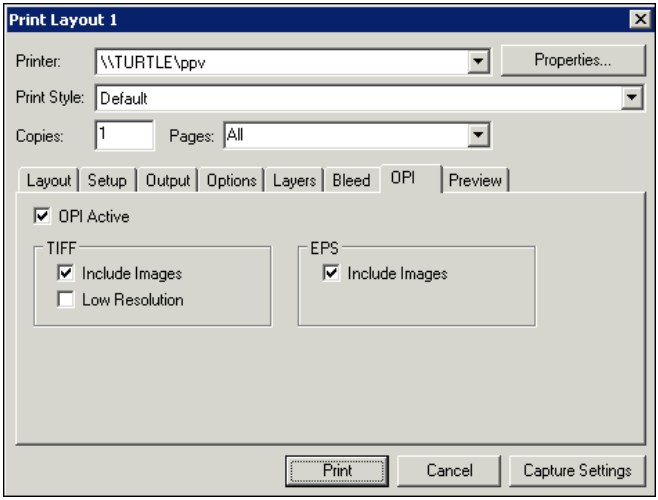


Fig. 5.11: “OPI” tab in QuarkXPress “Print” dialog

5.7.2 Overcoming QuarkXPress limitations with ImageServer and PDF HandShake

You can use ImageServer to overcome some QuarkXPress limitations. QuarkXPress 3 through 5 cannot produce DeviceN output, and QuarkXPress 3 through 6 cannot produce in-RIP separations. But if you use QuarkXPress in connection with ImageServer, you can produce in-RIP separations with DeviceN output. Here is how you can do it:

➢ Print your jobs composite from QuarkXPress to a “Create PDF” printer queue (requires PDF HandShake) to create PDF documents, using the ImageServer OPI image replacement functionality. Make sure that DeviceN printing is enabled on this queue (see Fig. 4.7 and **PostScript 3/DeviceN** in 4.3 “OPI printer queue settings”).

These PDF documents will include full composite DeviceN spot color information generated from the high-resolution DCS-2, TIFF and Photoshop images with spot colors, as well as from colorized grayscale images. You can print these PDF documents, e.g. via PDF HandShake with its variety of options including host-based and in-RIP separation to your PostScript 3 compatible output device.

5.7.3 HELIOS Preview XT

With this extension installed, a preview file with the same name as the document and the suffix ".xpv" will be generated each time you save a QuarkXPress document. The following steps describe how to install the extension and how it is used:

➢ Mount the "HELIOS Applications" volume, open the "Windows" folder (Mac: "MacOS"), then "ImageServer Tools" and "HELIOS Preview".

➢ Copy "HELIOS Preview XT.xnt" from the appropriate QuarkXPress version folder into the "XTension" folder of the QuarkXPress application.

Note: For QuarkXPress 6 or newer, you need to mount the proper "HELIOS Preview XT.dmg" disk image before getting access to the "HELIOS Preview XT.xnt" extension.

Now the installation of the module is complete and you can immediately utilize it after a restart of QuarkXPress:

Note: Open `Edit > HELIOS Preview XT...` from the QuarkXPress menu bar.

The checkbox `Create preview on saving` in the `General` tab allows you to specify whether previews should be generated each time a QuarkXPress document is saved (Fig. 5.12).

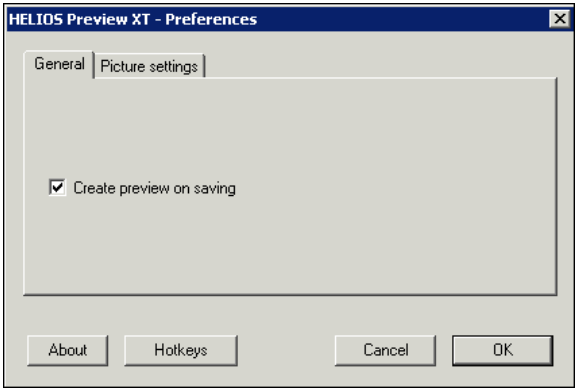


Fig. 5.12: HELIOS Preview XT: “General” tab

The `Picture settings` tab allows the specification of the preview image file format, PNG or JPEG (Fig. 5.13).

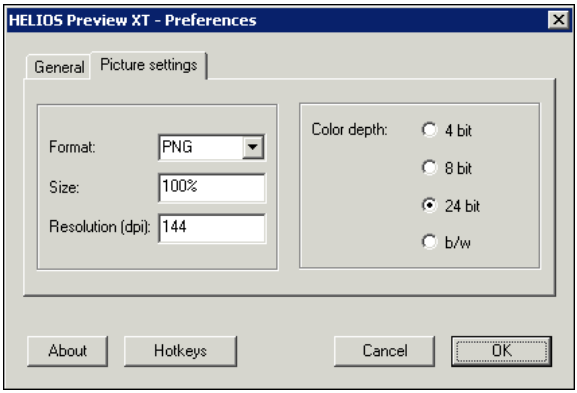


Fig. 5.13: HELIOS Preview XT: “Picture settings” tab

In the “Hotkeys” window tab you may specify a shortcut which enforces the preview generation without saving the document (Fig. 5.14).

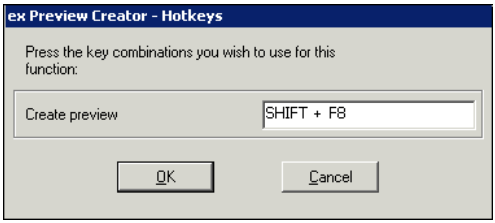


Fig. 5.14: HELIOS Preview XT: “Hotkeys”

5.8 Using InDesign with ImageServer

In this section we focus on the InDesign print output settings (Fig. 5.15), which allow you to specify if printing composite or separations, and the advanced settings (Fig. 5.16), in which OPI related settings are provided.

“Output” settings

In the `Color` pull-down menu you can select whether to print composite, separations or in-RIP separations. In order to have ImageServer perform composite color matching, thus keep image data unchanged, you should select `Composite Leave Unchanged`. If you wish to print documents from InDesign to an already defined color space, select `Composite Gray`, `Composite RGB`, `Composite CMYK`.

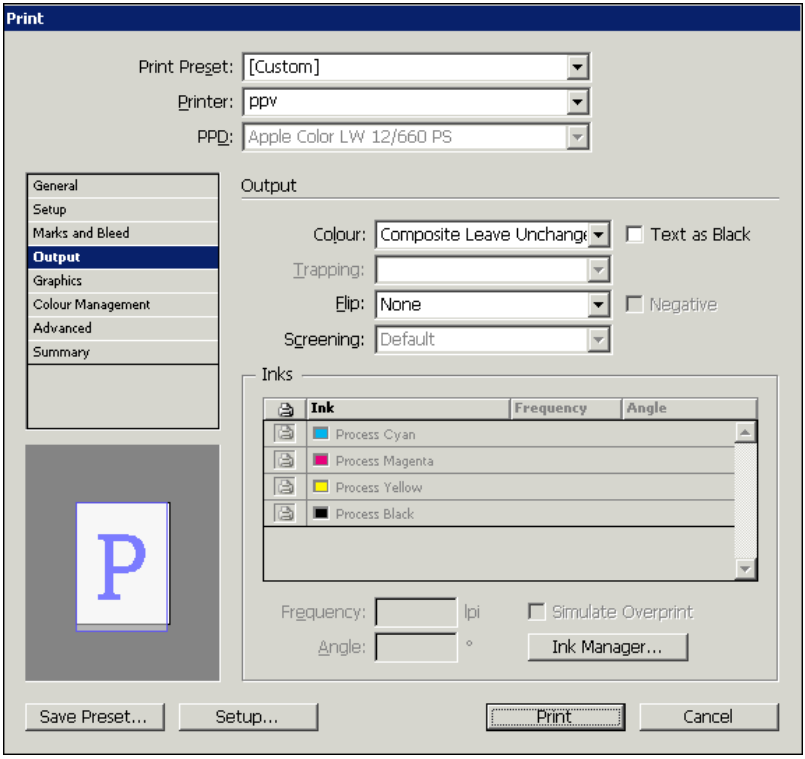


Fig. 5.15: “Output” tab in InDesign print dialog

“Advanced” settings

Here you can omit EPS, PDF or bitmap layout images for OPI. If one or more of these options are checked, InDesign does not include the placed layout images in the document, but will include all OPI references. This reduces the document size significantly while preserving all the essential information which is needed to have ImageServer perform proper OPI image replacement. Furthermore the option `OPI Image Replacement` must be unchecked. This en-

sures that ImageServer performs server-based image replacement.

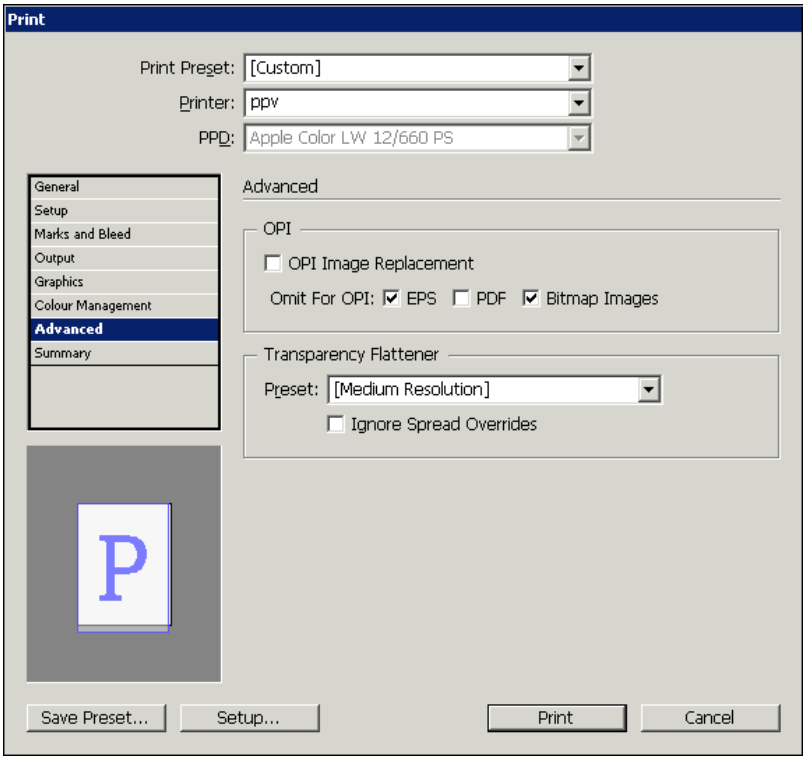


Fig. 5.16: “Advanced” tab in InDesign print dialog

When selecting an EPS layout image from the `Place...` dialog box, select the option `Show Import Options`. The option `Read Embedded OPI` Image Links tells InDesign to read links from OPI comments for images which are included (or nested) in the graphic (Fig. 5.17).

This option must be deselected if you are using layout images and plan to have the ImageServer OPI functionality perform the image replacement. When this option is deselected, InDesign preserves the OPI links, but does not read them. When you print or export, the layout images and the links are passed on to the output file.

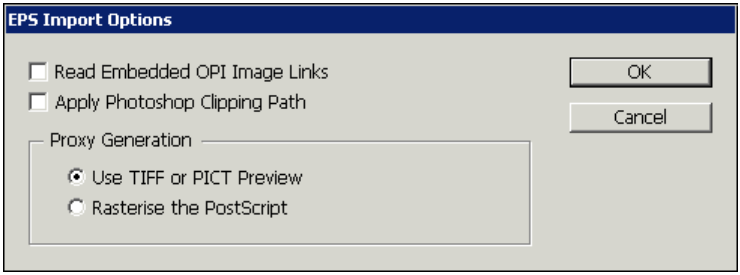


Fig. 5.17: InDesign “Import Options” dialog

Select the option `Apply Photoshop Clipping Path` to import a clipping path from a Photoshop EPS file.

5.8.1 The HELIOS Preview plug-in

With this plug-in installed, a preview file with the same name and the suffix “.xpv” will be generated each time you save an InDesign document. The following steps describe how to install the plug-in and how it is used:

➢ Mount the “HELIOS Applications” volume, open the “Windows” folder (Mac: “MacOS”), then “ImageServer Tools” and “HELIOS Preview”.

➢ From the appropriate InDesign version folder mount the “HELIOS Preview XT.dmg” disk image and copy the “HELIOS Preview XT.pln” plug-in into the “Plug-Ins” folder of the InDesign application.

Note: InDesign 2, CS and Windows clients:
Copy “HELIOS Preview XT.pln” into the “Plug-Ins” folder of the InDesign application.

Now the installation of the plug-in is complete and you can immediately utilize it after restarting InDesign:

➢ Open `Plug-Ins > HELIOS Preview > Preferences...` from the InDesign menu bar.

The “HELIOS Preview” window appears (Fig. 5.18). It allows you to specify the preview image file format, which can be PNG or JPEG, and their respective settings. In addition, you may specify with the `Preview on Saving` option whether previews should be generated each time an InDesign document is saved.

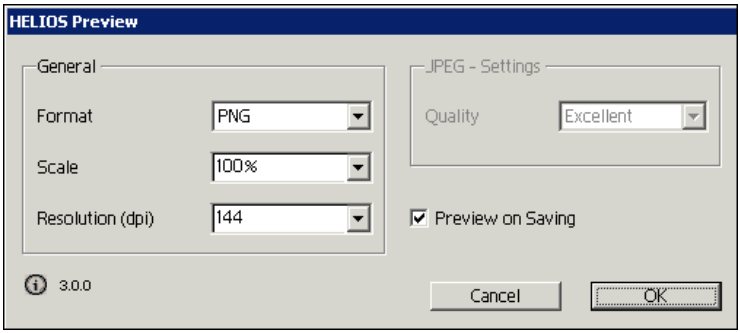


Fig. 5.18: HELIOS Preview: Settings

5.8.2 PDF-native OPI workflows

ImageServer provides a native PDF-based OPI image replacement solution that replaces, within in a PDF file, the low-resolution bitmap images with the high-resolution originals. This preserves transparency and metadata, and allows server-based PDF workflows utilizing OPI low-resolution images.

Modern pre-press applications, e.g. Adobe InDesign, are moving to a direct PDF output workflow allowing advanced features like PDF transparency, drop shadows, and embedded XMP metadata. These features are incompatible with today's PostScript output, but will be fully supported by this solution.

The main advantages of the HELIOS server-based OPI solutions are:

- Server based color management to color transform images for different printing processes. ICC profiles can be applied or embedded into PDF print jobs
- Submit PDF print jobs before all images are ready – the server will generate the final high-resolution PDFs when all images are completed
- Much faster PDF export from InDesign, using `Omit for OPI`
- Server defines color management for all OPI images

Remote users using WebShare can download low-resolution images and return PDF print jobs with images omitted, which are then replaced on the server for final output. For complete details, see "11 PDF-native OPI workflow" in the HELIOS PDF HandShake manual.

5.9 Viewing HELIOS XPPV documents

XPPV is a file format developed by HELIOS used for storing previews of multi-page documents, e.g. Quark XPress documents. The Mac file type for these document preview files is “XPPV” and the standard suffix is “.xpv”.

HELIOS WebShare allows conveniently previewing and proofing XPPV documents via any web browser (see 5.9.2 “Viewing preview images in WebShare” below).

5.9.1 Converting the preview image format

The preview of a page in an XPPV file can be converted to other image formats using the “layout” command (see 6.1.8 “Image conversion options”).

Example:

```
# layout -oSpotToProcess=true -oScreenResolution=0 -oPrintColor=RGB
  -aPageNumber=1 -T JPEG -l doc.xpv page1.jpeg
```

5.9.2 Viewing preview images in WebShare

HELIOS WebShare can display previews of QuarkXPress and InDesign document pages. The HELIOS “Preview” extensions automatically save an “.xpv” file containing the document preview together upon each QuarkXPress or InDesign document save. WebShare users can click on this “.xpv” file to view the document previews of each page. For more information see the section “Image and document previews” in the HELIOS WebShare manual.

5.9.3 XPV action scripts in WebShare

WebShare includes two sample action scripts related to XPV preview files. Within WebShare, first select an ".xpv" file, and then "XPV info" on the `Action >` menu (if enabled). It will list extensive information about the images, colors, text blocks, fonts, and other details about each page in the corresponding InDesign or QuarkXPress document. If the `Collect files from XPV` action script is selected, it will collect the referred data (QuarkXPress or InDesign document and images) from the XPV document, and save it into a "collection" folder. It also creates a report about the used fonts. See "8.4.3 Sample action scripts" in the HELIOS WebShare manual.

6 ImageServer utility programs

This chapter describes some modules of our software that are only accessible when working on a command line. It is meant for those who want to change or check specific OPI settings manually or to set up automated workflows, e.g. with Script Server (see 7 “HELIOS Script Server”).

ImageServer programs

ImageServer is composed of many individual programs e.g. “opisrv”, “opitouch”, “layout”, “oiimginfo”, and “hirespath”. In addition, it also makes use of the modules “papsrv” (see EtherShare manual), “psresolve” and different printer interface programs (“tcpif”, “smbif” and others). Some of the above mentioned programs are described in the following chapters.

The HELIOS utility programs described in this chapter are located in the “HELIOSDIR/bin” directory.

You may use them for setting specific parameters manually. This may be useful because some of these parameters were especially designed to max out application specific features.

For programs that are not described in this manual and for instructions on how to integrate plug-ins, please refer to the “HELIOS OpenImage Software Developer Kit”. This documentation can be found on the HELIOS website, section *Support > Developer Specifications*, and is meant for third-party developers only.

About defaults

Please note that parameters you set, using e.g. the “layout” program, override the settings in the preference data base. If you do not specify a specific

parameter, ImageServer will check the entry in the preference database. In case the preference data base does not contain an entry for this parameter, the default will be used. For changing entries in the preference data base, see 4.1 “ImageServer settings” and 9 “Preferences and notification features”.

6.1 layout

The “layout” program has three purposes:

- Layout file generation (6.1.3 “Layout file generation”)
- Image conversion (6.1.7 “Image conversion”)
- Tagging of ICC profiles, path information and XMP metadata (6.1.11 “Tagging ICC profiles, path information, and metadata to images”)

The files processed by “layout” differ in some respects; layout files that have been generated from high-resolution files include embedded OPI references to the original images, whereas converted image files do not. XMP and IPTC metadata are preserved by both generated layouts and converted images.

6.1.1 Syntax conventions

Irrespective of whether you call the “layout” program for a set of files or use it to poll permanently in the background, use the options described below. Type and default values are shown in angle brackets just before the description. The following types are used:

`uint32`	32-bit integer value (e.g. `72`)
`double`	Floating point value (e.g. `72.6`)
`string`	Any string enclosed in quotation marks (e.g. `"RGB"`)
`string list`	Comma-separated strings (e.g. `"RGB","CMYK"` – *no spaces!*)
`boolean`	`TRUE` or `FALSE`

6.1.2 General “layout” options

These options can be used for either layout file generation, or image conversion.

Note: Options that can also be specified in one of the HELIOS Admin dialogs are described with the note *“Compare HELIOS Admin!”*.

-h Display help file.

-v Display progress on “stdout”.

-H Ignore all user home directories as HELIOS volumes.

-o PreserveResources <string list:IPTC,PATH,XMP>

Determines additional resources, which will be preserved by “OpenImage” plug-ins. If the string contains `PATH`, `IPTC` or `XMP`, the named additional resource will be preserved. If no resources should be preserved, specify `none`.

-o OmitProfile <boolean:FALSE>

Specifies whether to tag a profile to the resulting output file. This option is useful if the generated layout/original is small while the specified profile was large.

-o DcsSuffixes <string list:"C","M","Y","K">

Suffixes of DCS plates that need to be processed.

-o DcsCodings <string list:"MacRoman",”PC850”>

The generation of EPS files requires an internal character encoding. This option allows specifying the desired encoding.

-o ProfileRepository <string:"ICC-Profiles">

This preference specifies the volume name containing the ICC profiles. You may also specify the absolute path to the repository or the path relative to the "HELIOSDIR" directory.

Compare HELIOS Admin!

-o ProfileSearchPaths <string list:None>

Additional directories to search for ICC profiles.

-o ProfilePaths <string list:
(RGB="ICC-Profiles:Scanner:HELIOS:sRGB_IEC61966-2-1_noBPC.icc",
CMYK="ICC-Profiles:Printer:ECI_Offset_2007:ISOcoated_v2_eci.icc")>

Defines RGB/CMYK profiles used if the color space for the output file and the color space of the high-resolution file are different. These profiles are used to match between color spaces properly. Each string has the format `<color space>=<path>`.

-o RenderingQuality <uint32:2>

Defines ICC rendering quality. You may choose one of the values listed below:

- `0` (normal quality)
- `1` (draft quality)
- `2` (best quality)

-o MissingIccProfileIsError <boolean:TRUE>

Defines missing profile as error when creating output file.

-o LogMissingIccProfile <boolean:TRUE>

Logs missing profile in "syslog" file.

-o UseProfile <boolean:TRUE>

Process color matching for output files, if the high-resolution files have tagged ICC profiles.

-o IncludeProfile <boolean:FALSE>

This option embedds the entire profile in the output file as a value (copy) instead of a reference (`-o OmitProfile`). Only applicable if `UseProfile` option is set.

-o ImageProfilePaths <string list:None>

If PDF input files are not yet tagged with ICC profiles, this parameter can be set for attaching a profile to RGB, CMYK or Grayscale objects within the document. Each string has the format `<color space>=<Pathname>` and defines the ICC profile path name for the given color space. This setting overrides the default profiles in the HELIOS Admin `PDF HandShake Settings`.

-o DcsCompatible <boolean:FALSE>

Lets the program generate DCS 1 or DCS 2 EPSF layout images if the raster-based high-resolution images include colors other than Black&White or Grayscale.

-o KeepDcs <boolean:TRUE>

Lets EPSF layout images of DCS 1 or DCS 2 high-resolution images keep the DCS structure. Changing the option to `FALSE` will lead to simple EPSF layout files.

-o Fast <boolean:FALSE>

If set to `TRUE`, the used image generation algorithm will be slightly less precise but faster, while consuming less resources. However, the rendering quality should be sufficient for smaller images.

-o PreserveDeviceN <boolean:FALSE>

This option optimizes the PostScript output of PDF original files to use PostScript 3/DeviceN features. It should only be set to `TRUE` if you print to a PostScript 3 device with in-RIP separation. This applies for Hexachrome printing, colorized images (Duotone), etc. When printing host-based separations with applications like

QuarkXPress, this feature must be turned off, otherwise the output will lead to unexpected results.

-o PDFTransparency <boolean:TRUE>

If a PDF document that contains transparencies is processed with this option, PostScript with "pdfmark" constructs is generated.

-o PDFPageBox <str:"CropBox">

Specifies the page box of PDF documents that should be used for output. PDF page objects always contain a MediaBox entry and may specify boxes within the MediaBox as CropBox, BleedBox, TrimBox and ArtBox. Valid values of this option are: `"MediaBox"`, `"CropBox"`, `"BleedBox"`, `"TrimBox"`, `"ArtBox"`.

-o SpotToEps <boolean:TRUE>

Causes the creation of EPSF layout images if a high-resolution raster image contains extra colors like spot colors.

-o MaskToEps <boolean:TRUE>

Causes the creation of EPSF layout images if a high-resolution raster image contains a clipping path.

-o FilterInks <string:<None>

This option can be used to omit specific channels from the target image. The argument is a comma-separated list of each separation color (Inks) as displayed in "oiimginfo". Thus, the first color has the index 0, the second 1, etc. All colors that are not listed are removed from the target image. For a CMYK image, to filter *Cyan* and *Magenta* only, this option must be set to *0,1*.

-o KeepPhysicalSize <boolean:FALSE>

If set to `TRUE`, this option preserves the physical size of an image when the image size is changed via the resolution.

-o BitsPerComponent <uint32:0>

Enforces the color depth of the target image. Valid values are `8` and `16`.

-o SpotToProcess <boolean:FALSE>

Forces output images to always contain only the default inks of the output image color space. Spot colors defined in a high-resolution image will be converted into the appropriate values of the output image color space.

-o ForceMatch <boolean:FALSE>

Forces ICC color matching while creating layout images or converting images, even if no color matching would be applied by the "layout" default behavior.

Note: Do not apply this option when converting images to grayscale or when having grayscale layouts created.

-o Compression <boolean:FALSE>

Forces the output image to be compressed the same way the high-resolution file is compressed. This option will only be used if the high-resolution file and the output image have the same file format.

-o ColorAliases <string list:None>

Defines mappings from application or localization dependent color names to the names used in "OpenImage". String format is `<Color-Name>=<ColornameAlias>`.

-o Signature <string:None>

Defines a signature filter configuration file. See "HELIOS OpenImage Software Developer Kit" for details.

-o RssLimit <uint32:0>

Controls resource usage of main memory that is used from the "layout" application. The specified value limits the maximum use of main memory in kilobytes. Zero means no limits.

-o ConvertRenderingIntents

Allows setting an ICC rendering intent. This may be one of:

- Perceptual (0)
- Relative colorimetric (1)
- Saturation (2)
- Absolute colorimetric (3)
- Perceptual with BPC (*Black Point Compensation*) (4)
- Relative colorimetric with BPC (5)
- Saturation with BPC (6)
- Perceptual with Black Plane Compensation (7)
- Relative Colorimetric with Black Plane Compensation (8)
- Saturation with Black Plane Compensation (9)

The following matrix displays the default ICC rendering intents that are used if not otherwise specified:

	To:	S	R	G	H	H	C	C	M	Y	C	C	C	C	Y
From:	Spot	1	1	1	1	1	0	0	1	1	1	1	1	1	1
	RGB	1	1	1	1	1	0	0	1	1	1	1	1	1	1
	Grayscale	1	1	1	1	1	0	0	1	1	1	1	1	1	1
	HSV	1	1	1	1	1	0	0	1	1	1	1	1	1	1
	HLS	1	1	1	1	1	0	0	1	1	1	1	1	1	1
	CMY	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	CMYK	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	Multitone	1	1	1	1	1	0	0	1	1	1	1	1	1	1
	YCbCr	1	1	1	1	1	0	0	1	1	1	1	1	1	1
	CIELab	1	1	1	1	1	0	0	1	1	1	1	1	1	1
	CIEXYZ	1	1	1	1	1	0	0	1	1	1	1	1	1	1
	CIELuv	1	1	1	1	1	0	0	1	1	1	1	1	1	1
	CIEYxy	1	1	1	1	1	0	0	1	1	1	1	1	1	1
	YCC	1	1	1	1	1	0	0	1	1	1	1	1	1	1

Example:

```
-o ConvertRenderingIntents="RGB:CMYK:2,CMYK:CMYK:0"
```

Converts RGB to CMYK with the intent “Saturation”, and also CMYK to CMYK with the intent “Perceptual”.

-o RenderColor <string:CMYK>

Sets the color space for rendered image objects. It is important that for PDF files this option is set to CMYK, to show proper overprints.

6.1.3 Layout file generation

Usually, layout files will be generated automatically by ImageServer, and new layouting parameters can be set using the HELIOS Admin program on any client. If you want to create a layout manually or if you want to use a special set of parameters for the creation, you may call the “layout” program directly. This procedure may optionally be automated by means of Script Server.

There are two main possibilities of calling the “layout” program. If you want to create layout files for a single file or a set of known files, you can use the following call:

```
layout [options] files... or
layout [options] -L dir file(s)
```

If you want the “layout” program to scan permanently for high-resolution files in a special directory, use the call:

```
layout [options] -p polltime directory or
layout [options] -p polltime -L dir <polled directory>
```

Now, the layout process will go into the background and scan the given directory constantly with the given interval polltime in seconds for a new or changed high-resolution file to create a layout file from. This feature should only be

used for small directories because it is not as fast and efficient as automatic layout generation.

Instead, we recommend to use Script Server, which is described in 7 “HELIOS Script Server”.

6.1.4 Options for the layout file generation

Note: Options that can also be specified in one of the HELIOS Admin dialogs are described with the note *“Compare HELIOS Admin!”*.

-E

Return the path to the generated layout.

-G

Send notify event for generated file.

-p <int: (see text)>

Check the directory regularly for new image files (“polling”). The default is `0 = never`.

-L dir <string:None>

Set destination directory for layout file generation. It is sufficient to state the relative path for manual (*NOT* for automatic “polling”!) mode. This option creates all layouts in a “layouts” subdirectory of the indicated directory. This option is mostly needed when the high-resolution files reside on a read-only medium e.g. a CD-ROM.

Note: If the source directory is located in a HELIOS volume, but the destination directory is not, first copy the high-res original into the destination directory using the “dt” tools (see the Base manual). Then perform a regular “layout” call without specifying `-L`.

-o MinLayoutSize <uint32:0>

Defines minimum size in bytes needed to start layout process (*0* = any size).

-o KeepImageResolution <boolean:FALSE>

Do not consider `PrintResolution` (see below); the layout file will inherit the resolution of the high-resolution image.

-o PrintResolution <double:72.0>

Defines the resolution used for the printable part of the layout (`0` = omit printable part).

Compare HELIOS Admin!

-o ScreenResolution <double:72.0>

Defines the resolution used for the screen preview part of the layout (*0* = omit screen preview).

Compare HELIOS Admin!

-o PrintColor <string:"CMYK">

Defines the color space used for printable part of layout file. If the string is set to *None*, the color space of the high-resolution file is used. For valid color spaces, see Table 6.1 below. Please note that some color spaces cannot be applied to certain file formats.

Compare HELIOS Admin!

Note: This option is ignored for Bilevel and Grayscale images if the target format is specified as EPS.

-o ScreenColor <string:"RGB">

Defines the color space used for the screen preview part of the layout file. If the string is set to `None`, the color space of the high-resolution file is used. For valid color spaces, see Table 6.1 below.

Note that some color spaces cannot be applied to certain file formats.

Name of color space		
None	HSV	YCbCr
Spot	HLS	CIELab
Bilevel	CMY	CIEXYZ
Grayscale	CMYK	CIELuv
Indexed	Multi	CIEYxy
RGB	Duotone	YCC

Table 6.1: List of layout color spaces

-o CompressPrint <string:None>

Defines the compression mode used to generate the printable part of the layout file using a string from Table 6.2. Please note that some compression modes cannot be applied to certain file formats.

Compare HELIOS Admin! (HELIOS Admin provides less options.)

-o CompressScreen <string:None>

Defines the compression mode used to generate the screen preview of the layout file using a string from Table 6.2. Note that some compression modes cannot be applied to certain file formats.

Name of compression mode		
None	JPEG	Berthold Lineart
Compress	JPEG 2000	Pixar
CCITTRLE	PackBits	Flate
CCITTG3	NextRLE	RLE
CCITTG4	Thunder	

Table 6.2: List of layout compression modes

-o CompressQuality

Defines the compression quality of layout files or image components:

For JPEG:

-o CompressQuality <uint32:75>

Creates poor...high quality JPEG image (1...100).

For JPEG 2000:

-o CompressQuality <uint32:0>

Specifies the image quality of a JPEG 2000 image in relation to the uncompressed high-resolution original. The values range from 1...100. Specifying "0" means *lossless*.

Note: `CompressQuality` specifies the percentage size of the JPEG 2000 image data compared to the size of the uncompressed image, which the JPEG 2000 should not exceed. For example, if an image with lossless compression has only 50% the size of an uncompressed image, a `CompressQuality` value between 50% and 100% produces no visible differences.

-o LayoutHighQuality <boolean:TRUE>

Create the best layout quality possible. Otherwise (`FALSE`), the smallest version of the image that matches the defined layout resolution is used. The high quality setting uses a bilinear resolution converter, low quality setting uses the "nearest neighbor" algorithm.

-o LayoutCreator <string:"8BIM">

Defines Mac file creator used for layout creation (The default value `"8BIM"` defines Adobe Photoshop).

-o LayoutEpsCreator <string:None>

Some application programs, e.g. Adobe InDesign, parse Photoshop resources only if they find a suitable creator in the file header. This option allows specifying a creator for creating OPI layouts.

-o LayoutAlpha <boolean:FALSE>

Most layout programs do not support OPI for transparent images, so transparency channels are stripped from layout images by default. If Alpha support is active (`UseAlpha = TRUE`) and `LayoutAlpha=TRUE`, layout images will not be stripped of transparency channels. Using this option can make sense in a PDF-native OPI workflow.

-o ForceRasterLayout <boolean:FALSE>

By specifying this option, all layout images will be created in raster format (provided that `ForceEpsLayout` is not specified!). However, the original image should contain a raster image.

-o ForceEpsLayout <boolean:FALSE>

By specifying this option all layout images will be created in EPSF format (provided that `ForceRasterLayout` is not specified!).

-o PDFNativeWorkflow <boolean:FALSE>

This preference enforces the generation of PDF-native workflow compatible layout images from vector-based EPS images, even if TIFF layout images are supposed to be generated (e.g. via the "%t" folder syntax).

Note: This option must be set to `FALSE` if `RasterImageType` and `RasterImageExt` are used. Otherwise no raster layout can be enforced.

-o SingleEpsPath <boolean:TRUE>

For most images, ImageServer supports handing over different paths. However, most application programs cannot cope with different paths when it comes to EPS images. So only the clipping path is currently evaluated by default, which is controlled by this option.

-o UseAlpha <boolean:TRUE>

If set to FALSE, ImageServer ignores alpha channels in image files.

-o RasterImageType <string:"TIFF">

If the high-resolution file uses raster format and does not include any mask, the resulting layout file will inherit the raster format. The used layout file format can be overridden using this option. You may use any string defined for the file type (e.g. "JPEG"). See valid file types in Table 3.3 in 3 "Before getting started".

-o RasterImageExt <string:None>

Specifies the file type by means of the file's subformat when generating layouts from raster-based images.

-o EpsImageType <string:"EPSF">

Defines the file type which is used to select the manager which creates layout files. You may use any string defined for the file type. See valid file types in Table 3.3 in 3 "Before getting started".

-o EpsImageExt <string:None>

Specifies the file type by means of the file's subformat when generating layouts from EPSF images.

Please note that the following four options can only be used if the created layout file is an EPSF file.

-o EpsKeepSize <double:10.0>

Defines the allowed difference in size between the layout file and the high-resolution file (in percent). 20.0, e.g., specifies that the layout file must be smaller than 120% of the high-resolution file or else only a copy of the high-resolution file will be used as layout file. You may also specify negative values here. Valid values range from -99 to any positive value. *Applies only to EPSF layout files*.

-o EpsPrintablePreview <boolean:TRUE>

If set to `FALSE`, "layout" creates a screen preview of the image but no printable layout. *Applies only to EPSF layout files*.

-o EPSPreviewPSLevel<int:1>

Determines the preview PostScript level for EPSF files. By default, this is PostScript level 1, optionally PostScript level 2 can be specified. *Applies only to EPSF layout files*.

-o EpsBinaryEncoding <boolean:TRUE>

Defines that image data is stored using 8-bit values (otherwise 7-bit hex is used). *Applies only to EPSF layout files*.

-o PreviewFromPict <boolean:TRUE>

Mac PICT may be used as source for layout generation.

-o OmitScreenPreviewTypes <string list:"TIFF","JPEG","PNGf","BMP ","PDF ">

By default, all provided managers create a screen preview in addition to the print preview. Thus, TIFF layouts may contain an additional PICT element. With this option you can define a list of layout file types for which the screen preview should not be included in the layout file, e.g.: `"TIFF","JPEG"`

-o MaxScreenSize <int:800>

Defines the maximum allowed image size in pixels for screen previews.

-o MaxThumbnailSize <int:200>

Defines the maximum allowed image size in pixels for thumbnails.

-o FailureInfoLayouts <boolean:TRUE>

Defines whether a standard default layout (indicating the error) should be created in case of layout generation failure.

-o CrossPlatformLayout <boolean:FALSE> (see also `-P -M` options below)

Defines whether created layouts are usable on all supported platforms (e.g. all relevant data is stored in the data fork). If the high-resolution file has a resource, the layout file contains a resource when stored in a HELIOS volume. This option can be used to specify the preview image format of the generated layout file; in a cross-platform layout the preview is TIFF, otherwise a Mac PICT preview is used.

-o CrossplatformFiles <boolean:FALSE>

Defines whether a resource fork is created (see also the `CrossPlatformLayout` option above).

-o Thumbnail <boolean:TRUE>

Include Mac icon in layout file (not applicable to cross-platform files).

-o LayoutComment <string:"HELIOS ImageServer 5.0.0 layout of %T file \"%f\", size %k kBytes, created by %C">

Overrides the default comment field for layout images. The option is a string which is placed in the comment section of a Mac file and can be viewed in the Finder using the `Get Info` dialog box. This string may include the following symbols (the comment may contain up to 199 bytes):

- `%f` File name of the original image
- `%b` Size (bytes) of the original image
- `%k` Size (kilobytes) of the original image
- `%m` Size (megabytes) of the original image
- `%t` File type of the original image printed as 4 byte signature
- `%c` File creator of the original image printed as 4 byte signature
- `%T` File type of the original image printed as clear text if possible (otherwise printed as with %t)
- `%C` File creator of the original image printed as application name if possible (otherwise printed as with %c)

-o LayoutLabel <uint32:0>

Sets the label color of a layout file. An integer number between `1` and `8` reflects the Mac Finder label colors, `0` turns this option off.

-o FailureComment <string:"Failed HELIOS ImageServer 5.0.0 layout file of \"%f\"">

Defines the comment string for a failed layout image (see also `LayoutComment`).

-o FailureLabel <uint32:0>

Specifies the label color of a failed layout image (see also `LayoutLabel`).

-o IgnoreMasks <boolean:FALSE>

If `TRUE`, this option causes an existing clipping path to be ignored when creating a layout image. See 3.3.2 “Image paths and clipping paths” and 5.2 “Define folder specific OPI settings” for a description.

-o IgnoreSpots <boolean:FALSE>

If `TRUE`, this option specifies that additional channels (e.g. spot colors) are ignored when creating a layout image, printing the resolved high-resolution image or converting an image. See 3.3.4 “Additional channels in bitmap images” and 5.2 “Define folder specific OPI settings” for a description.

-o upscale <boolean:FALSE>

Allows upward scaling so that the layout image may have a resolution higher than the resolution of the original image.

-o ProtectLayouts <boolean:FALSE>

With this option set to `TRUE` generated layout files will have the Mac Finder flag Locked turned on. Before setting this flag to `TRUE` make sure that the DTP application you use can handle locked layout files!

The following options are only supported for backward compatibility with previous ImageServer versions; please do not use these options for new installations or configurations!

-r dpi <double:72.0>

Set resolution (*in dpi*) used for creating layout files. Use `PrintResolution` and `ScreenResolution` instead.

-M <boolean:TRUE> (opposite to -P)

Produce layout files for the Mac platform only. Use `CrossPlatformLayout` instead.

-P <boolean:FALSE> (opposite to -M)

Generate cross-platform layouts. Use `CrossPlatformLayout` instead.

-n <boolean:TRUE> (opposite to -N)

Limit the size of EPSF layout files on generation. Use `EpsKeepSize` instead.

-N <boolean:FALSE> (opposite to -n)

Do not limit the size of EPSF layout files on generation. Use `EpsKeepSize` instead.

-c <boolean:FALSE>

Compress screen previews. Use `CompressScreen` instead.

-d dcsSuffixes <string:"CMYK">

Defines valid DCS suffixes. You may specify any number of characters. These characters are handled case-insensitive. Use `DcsSuffixes` instead.

-C [rgb|cmyk] <string:"CMYK">

Defines the color space used for raster-based layouts (e.g. TIFF layouts). Use `PrintColor` instead.

-e [rgb|cmyk|bw] <string:"CMYK">

Defines the color space used for EPSF printable previews. Use `PrintColor` instead.

-t [rgb|cmyk|bw] <string:"CMYK">

Defines the color space used for EPSF screen previews. Use `ScreenColor` instead.

-a attributes

Allows you to specify a comma-separated list of attributes. Use `-o Attributes` instead.

6.1.5 Attributes for "layout" options

Attributes are only applicable to specific file types. They are entered as command line options using the syntax:

```
layout -o Attributes=<attributes>=<value>
```

Additional attributes are delimited by commas.

Attributes <string list:"key=value">

Consists of `key=value` pairs, delimited by comma characters. These attributes are passed to the image managers of OPI. The keys and possible values recognized depend on the image managers.

Baseline (JPEG only) <boolean:FALSE>

Saves JPEG layout images per default in progressive method. If this option is set to `TRUE`, JPEG layout images are saved as baseline JPEGs instead.

DctMethod (JPEG only) <uint32:0>

Use slow, fast or precise DCT (0, 1, 2).
DCT = *Discrete Cosinus Transformation*

PDFLayer <string:"">

Select or deselect layers in PDF documents. See also the appendix "PDF layers" in the PDF HandShake manual.

The following two attributes are only supported for backward compatibility with previous ImageServer versions; please do not use these options for new installations or configurations. Use the option `-o CompressQuality` *instead!*

Quality (JPEG only!) <uint32:75>

Create poor…high quality JPEG image (1…100).

Quality (JPEG 2000 only!) <uint32:0>

Specifies the image quality of a JPEG 2000 image in relation to the uncompressed high-resolution original. The values range from 1…100. Specifying "0" means *lossless*.

Note: `Quality` specifies the percentage size of the JPEG 2000 image data compared to the size of the uncompressed image, which the JPEG 2000 should not exceed. For example, if an image with lossless compression has only 50% the size of an uncompressed image, a `Quality` value between 50% and 100% produces no visible differences.

TileWidth (JPEG 2000 only!) <uint32:256>

Specifies the horizontal size of JPEG 2000 tiles. A JPEG 2000 image file consists of juxtaposed tiles. If the value specified is `0`, the whole image merely consists of one tile.

TileHeight (JPEG 2000 only!) <uint32:256>

Specifies the vertical size of JPEG 2000 tiles. A JPEG 2000 image file consists of juxtaposed tiles. If the value specified is `0`, the whole image merely consists of one tile.

Note: The subdivision of a JPEG 2000 image in tiles accelerates the coding and decoding process significantly, and minimizes the memory requirement during the operational time. The best value, which meets operational time as well as memory requirement, is 256.

AntiAlias (PDF only) <boolean:TRUE>

This attribute determines whether the image is smoothed.

PageNumber <uint32:1>

This attribute lets you specify a certain page of a multi-page PDF, XPV or TIFF document for image conversion or creating layouts.

Password <string:"None">

This attribute allows image conversion from a password protected PDF file. Note that this attribute requires that "layout" is used in the image conversion mode (see 6.1.8 "Image conversion options").

SaveUnseenColor (PNG only) <boolean:FALSE>

Usually, transparent points become white. This attribute leaves them in their original color if this option is set to `TRUE`. Does not work with indexed images.

ColorMasking (PNG only) <string:"None">

This attribute allows the definition of a transparency color. This color is set in the form `<r>:<rd>,<g>:<gd>,<b>:<bd>`. Each of these variables is a positive integer with values between 0 and 255. "r", "g" and "b" are the *red*, *green* and *blue* values of the color that shall be transparent, while "rd", "gd" and "bd" stand for the distance that a color can have to the transparent color and still be transparent. So `ColorMasking=255:1,255:1,255:1` would mean: Every white pixel with RGB values between `255,255,255` to `254,254,254` shall be transparent.

Note: This attribute only works with True Color or Indexed Color images.

WebOptimize (PNG only) <boolean:TRUE>

By default, all PNG-layouts are interlaced, so a web browser can show an impression of this picture while the image is still loading. If you want to turn this feature off you will have to use this option, e.g.:

```
layout -o Attributes "WebOptimize=FALSE"
```

6.1.6 Layout generation example

If you wish to create a JPEG-RGB layout from a TIFF-CMYK high-resolution image without a clipping path you have to induce the “layout ” program to convert CMYK color data into RGB and the TIFF file format into JPEG. In that case you use the `PrintColor` option to specify RGB. Furthermore, use the `RasterImageType` option to create a JPEG layout file. The complete command line is given below:

```
layout -v -o PrintColor=RGB -o RasterImageType=JPEG /images/cmyktif.tif
```

6.1.7 Image conversion

ImageServer can only convert raster images. If you wish to convert other image formats, you may create a PDF file from the vector image and convert it, or use the utility program **psrip**.

Note: The HELIOS CD contains the “imageconv” script (in “sample-images/template-images%0”) as well as various sample images. “imageconv” can be used as an excellent source of “layout” command image conversion examples.

6.1.8 Image conversion options

-l file(s) destination

The `-l` option specifies that instead of generating a layout file, image conversion should occur. Highest quality converted images are then produced according to options which specify the output file format, color space, resolution, and compression. Image conversion can be executed from the command line, but more commonly is scripted in Script Server to enable interactive image conversion via hot folders and fully automated background image conversion.

When using the `-l` option, specify one or more files to convert as well as a destination file (if converting a single file) or directory (if converting several files).

Usage:

```
layout [options] -l file(s) destination
```

Example:

```
$ layout -o PrintColor=RGB -T TIFF -l file1.jpg file2.jpg
  /tmp/images
```

will convert “file1.jpg” and “file2.jpg” to RGB TIFF images and store them in “/tmp/images”.

The `-l` option may be combined with the `-p` option:

```
layout [options] -p polltime -l directory destination
```

When using the `-l` option together with the `-p` (*polling*) option, a destination directory must be specified with its *absolute path*.

Example:

```
$ layout -o PrintColor=RGB -p 45 -l /user/dir
  "{unix,native,,TIFF}/tmp/images"
```

Now, the layout process will go into the background, scan the given directory every 45 seconds for a new or changed image file, and then converts this file into a TIFF-RGB image which will be stored in the directory "/tmp/images".

Note: Script Server provides a faster, more efficient automation method, and is therefore recommended instead of polling. See 7 "HELIOS Script Server".

-T (Format)

Set the target file format (JPEG, PNGf, TIFF, etc.) for an image conversion.

-o MultiInkSeparationColors <string list:None>

Defines the names and order of the color components when using Multichannel ICC profiles (5-8 channels). This option can be used if converting to the "Multi" color space. Usage of the "Multi" color space requires the use of ICC profiles. Future ICC profiles that include the names of the color components, will dispense with this option.

-o xPix || yPix <uint32:None>

Allows the user to define the size of the output file in pixels either in x or y direction. The layouts always maintain their aspect ratio. "layout" guarantees that the resulting output files will always be smaller than or equal to these sizes. If the use of `xPix` would mean that a given `yPix` value cannot be maintained, `yPix` will control the output file dimensions and vice-versa.

Example:

```
$ layout -o xPix=150 -o yPix=150 -l image.tif new.tif
```

This will create an output file named "new.tif" which is equal to or smaller than 150 by 150 pixels.

Note: Due to the fixed aspect ratio, the image resolution may not be identical for x- and y-direction. When such an image is opened, e.g. using Adobe Photoshop, a pixel aspect ratio warning is shown. The automatic resolution can be overridden additionally using the `PrintResolution` (see 6.1.4 “Options for the layout file generation”) option which defines a fixed resolution in each direction. However, this may break the exact aspect ratio of the converted image.

-o ExifRotate <boolean:TRUE>

If the original image contains an Exif parameter specifying an orientation, the image is rotated so that the target image has the correct orientation.

-o ShowIncomplete <boolean:FALSE>

By default “layout” generates an error if an image was not written completely. If this option is set to `TRUE` there is no error if at least 5 lines were written. Instead, missing image lines will be filled with the content of the last read image line.

-o ApplyClippath <boolean:FALSE>

If the source image contains a clipping path, areas outside of this path will become white. If the target format supports transparencies the target image will get a new transparency channel, which makes areas outside of the clipping path transparent.

-o ChangeClippath <string:None>

If an image is converted into a different format that does not support multiple paths, this option can be used to generate a picture with the specified path as a mask object. It contains the name of the path object that shall be the new clipping path.

-o PureBlack <boolean:FALSE>

Exclude black colored *raster* objects in PDF files from color transformation, so that they remain *black* even if color matching is done.

Black raster objects in Gray/RGB/CIELab/Indexed color spaces are detected and converted to *Black* only for CMYK output. This option can be used to override the default **PureBlack** preference described in 9 “Preferences and notification features”.

-o PureWhite <boolean:FALSE>

Exclude white colored *raster* objects in PDF files from color transformation, so that they remain *white* even if color matching is done. CMYK values will be zero, i.e. no ink is applied to the plates. This option can be used to override the default **PureWhite** preference described in 9 “Preferences and notification features”.

-o PureGrays <boolean:FALSE>

Exclude gray colored *raster* objects in PDF files from color transformation, so that they remain *gray* even if color matching is done. Gray raster objects in Gray/RGB/CIELab/Indexed color spaces are detected and converted to *Gray* only for CMYK output. This option can be used to override the default **PureGrays** preference described in 9 “Preferences and notification features”.

-o PureCMY <boolean:FALSE>

Applies to CMYK to CMYK conversions of *pixels of raster images*, and preserves the input color if, and only if, black is 0% and exactly one CMY primary color is 100% and all other primary colors are 0%. This option can be used to override the default **PureCMY** preference described in 9 “Preferences and notification features”.

-o PureVectorBlack <boolean:FALSE>

Exclude black colored *vector* objects in PDF files from color transformation, so that they remain *black* even if color matching is done. Black text and vectors in Gray/RGB/CIELab/Indexed color spaces are detected and converted to *Black* only for CMYK output.

-o PureVectorWhite <boolean:FALSE>

Exclude white colored *vector* objects in PDF files from color transformation, so that they remain *white* even if color matching is done. CMYK values will be zero, i.e. no ink is applied to the plates.

-o PureVectorGrays <boolean:FALSE>

Exclude gray colored *vector* objects in PDF files from color transformation, so that they remain *gray* even if color matching is done. Gray text and vectors in Gray/RGB/CIELab/Indexed color spaces are detected and converted to *Gray* only for CMYK output.

-o PureVectorCMY <boolean:FALSE>

Applies to CMYK to CMYK conversions of *text and vector objects*, and preserves the input color if, and only if, black is 0% and exactly one CMY primary color is 100% and all other primary colors are 0%.

-o ImportMask <string:None>

This option allows importing a clipping path from a different image into the target picture. The main reason for this feature is the ability to add a different or more precise clipping path into an image. Its parameter is the name of the picture that contains the clipping path that shall be imported (both pictures must be of the same size!).

Note: For image formats with a lossy compression we recommend to tag the original image with the new path information instead of saving it anew. See 6.1.11 “Tagging ICC profiles, path information, and metadata to images”.

-o ConvertEpsCreator <string:None>

Some application programs, e.g. some Adobe InDesign versions, parse Photoshop resources only if they find a suitable creator in the file header. This option allows specifying a creator for converting images.

-o OutputProfile <string:None>

This option receives the path or file name of a suitable output profile, which is mapped to the target. In doing so, ICC color matching is enforced and the appropriate color space is defined for the profile. This option overrides the `PrintColor` option.

-o DevLinkProfile <string:none>

This option receives the path or file name of a Device Link profile as a parameter. In doing so, the standard ICC handling is overridden, and the target color space is internally set to the profile target color space. Then the source image is converted according to the DeviceLink profile directions and saved in the target format.

Note: For easier identification, the target image metadata obtains the attribute Used DevLinkProfile with the path or file name as a parameter. The attribute can be displayed with “oiimginfo” (see 6.4 “oiimginfo”).

For the following 4 options, a number between 0.0 and 1.0 specifies a per cent value. A number > 1.0 specifies a pixel value.

-o cropLeft <double:0.000000>

Applies left margin cropping to the output image.

-o cropRight <double:0.000000>

Applies right margin cropping to the output image.

-o cropTop <double:0.000000>

Applies top margin cropping to the output image.

-o cropBottom <double:0.000000>

Applies bottom margin cropping to the output image.

The next three options only work for the generation of non-EPS images. They do not work if the specified target image format is EPS.

-o rotate <double:0.000000>

Rotates the generated output image clockwise a certain number of degrees (`0`, `90`, `180`, `270` are permitted).

-o flipVertical <boolean:FALSE>

Causes the generated output image to be flipped around the vertical centerline (swap `right` and `left`).

-o flipHorizontal <boolean:FALSE>

Causes the generated output image to be flipped around the horizontal centerline (swap `top` and `bottom`).

-o RemoveAfterConvert <boolean:FALSE>

If set to `TRUE`, original files which have been converted will be removed after a successful conversion.

-o SkipLikeLayout <boolean:FALSE>

If set to `TRUE`, `layout -l` will skip files that are not image files, just as "layout" in standard mode would.

-o IncludeRef <boolean:FALSE>

If set to `TRUE`, a converted file will contain a reference to the original file. This is similar to a generated layout file, which contains a reference to the original high-resolution file.

6.1.9 Image conversion examples

Example 1:

If you want to create a JPEG-RGB image from a TIFF-CMYK high-resolution image without a clipping path, you have to induce the "layout"

program to convert CMYK color data into RGB and the TIFF file format into JPEG. In that case you use the `PrintColor` option to specify RGB. Furthermore, you must use the file format option to create a JPEG file. The complete command line is given below:

```
$ layout -v -o PrintColor=RGB -T JPEG -l cmyktif.tif rgb.jpg
```

Example 2:

The "imageconv" script which uses many of the image conversion features can be found on the HELIOS CD in the `sample-images/template-images%0` folder.

The "template-images%0" directory contains about 90 MB of sample images and the "imageconv" Perl script. This script uses the "layout" command with several options to convert the original images to any supported image type with different color spaces. After the conversion is finished all images will take about 1.8 GB of hard disk space.

Note: The "imageconv" script will run – depending on the used server – in as little as 4 minutes on a fast server, or over 30 minutes on a slower server. "imageconv" converts images sequentially. Therefore only one CPU is used.

6.1.10 "OpenImage" file specification

Though the classic "OpenImage" file specification is still supported by HELIOS, we recommend to use the `-T` option instead (see 6.1.8 "Image conversion options").

If used, the "OpenImage" file specification is included as part of the `-l` option.

Usage:

```
layout [options] -l file(s) destination
```

Example:

```
$ layout -o PrintColor=RGB -l file1.jpg file2.jpg
"{unix,native,,TIFF}/tmp/images"
```

will convert “file1.jpg” and “file2.jpg” to RGB TIFF images and store them as plain UNIX files in “/tmp/images”.

The image type of the destination file(s) is given by specifying the destination as a complete “OpenImage” file specification.

The “OpenImage” file specification precedes the file or directory name and is put in braces. It is structured as follows:

{File System Type, FileRepresentation, Creator, ImageType, ImageTypeExtension}

FileSystem Type must be `unix`.

FileRepresentation may be either `native`, `ES` or `PCS`.

Creator is a Mac file creator signature (if left blank, `8BIM` for Adobe Photoshop is used).

ImageType is a file type signature (as e.g. `TIFF` or `EPSF`; see Table 3.3 in 3.2.3 “Supported file formats”).

If the target image file format (*ImageType*) is omitted, the file format of the source image is used, e.g.:

```
$ layout -l image.eps "{unix,native,,}TARGETIMAGE"
```

creates an EPS image and

```
$ layout -l image.tif "{unix,native,,}TARGETIMAGE"
```

creates a TIFF image.

ImageTypeExtension is an optional extension to the file type and may be empty.

6.1.11 Tagging ICC profiles, path information, and metadata to images

The "layout" program can also be used to tag ICC profiles to image files. In addition, path information and image metadata can be added, deleted or modified. To do so, the `-f` option must be specified with appropriate substrings, which are described below. This method has the advantage (over the `-o ImportMask` method; see 6.1.8 "Image conversion options"), that it does not save the image data completely anew, but only the tagging information, and thus does not deteriorate the image quality when processing lossy files.

6.1.12 Tagging options

-f <string list:None>

This option sets "layout" into a *tag only* mode, so only the specified file is modified (irrespective of an existing "!iccinfo.oic" file for ICC tagging). The `-f` option gets a string with up to three substrings, that are concatenated by a colon (":"). You can issue multiple tag orders in one layout command by specifying multiple `-f` options.

Note: *Exporting* metadata (paths, ICC profiles, etc.) is done via the "oiimginfo" utility (see 6.4.3 "Export").

Object defining substring

The first substring contains an object type; valid object types are:

ICC Add or delete a named ICC profile.

PATH An image mask or path.

CLIP Modify the clipping path parameter. It can either be deleted or will define a named path as the clipping path.

IPTC Delete the IPTC information of this image or import IPTC information of another image.

XMP Delete XMP information or import XMP information of another image or special XMP file.

RESO Change image resolution (implemented for TIFF, JPEG, PNG and Photoshop images).

Action defining substring

The second substring defines an action; valid actions are:

ADD Add ICC profiles, a clipping path or metadata to a file. For ICC profiles this will only work if the substring also contains an "EMB" or "REF" to define how a profile will be added. To "ADD" paths, clipping path information or IPTC information, that were exported by "oiimginfo", use the substring "ADDRES" (short for *add resource*). Note that "ADD" will not override an existing resource, for this, use "OVER".

DEL Delete the specified object.

OVER Almost the same functionality as "ADD", with the difference that "OVER" *overrides* existing ICC profiles.

Parameter defining substring

The third parameter will either contain the name of a profile, a color space name, the name of an image, a path name, the name of a special file, or resolution or size values. The following table summarizes possible combinations:

Object	Action	Parameter
ICC	ADDEMB/OVEREMB ADDREF/OVERREF DEL	<profile_name> <profile_name> (<color_space>)
PATH	ADD ADDRESS DEL	<image_name> <path_name> (<path_name>)
CLIP	ADD DEL	<path_name> (<path_name>)
IPTC	ADD DEL	<path_name> <path_name>
XMP	ADD DEL	<path_name> <path_name>
RESO	INCH CM SIZE SIZECM	<xRES>,<yRES> <xRES>,<yRES> <xSIZE>,<ySIZE> <xSIZE>,<ySIZE>

Examples:

```
$ layout -f "ICC:ADDEMB:Testprofile" test.jpg
```

tags the image “test.jpg” with the embedded ICC profile “Testprofile”.

```
$ layout -f "ICC:DEL" test.jpg
```

deletes all embedded or referenced ICC profiles from the image “test.jpg”.

```
$ layout -f "ICC:DEL:RGB" test.pdf
```

deletes only the RGB profile from the PDF file “test.pdf”.

```
$ layout -f "PATH:ADD:Import.jpg" test.jpg
```

imports the clipping path from the image “Import.jpg” and adds this to “test.jpg” as a path, but not as the clipping path. If the imported path should also be designated as the clipping path, then the “CLIP” tag order must be specified as well.

```
$ layout -f 'PATH:ADDRES:Sky.path" test.tif
```

adds the path “Sky”, that was exported by “oiimginfo” to the file “Sky.path”, to the image “test.jpg”. If “test.jpg” already contains a path named “Sky”, the new path will be renamed to “Sky-1”. If there is already a “Sky-1”, it will be renamed to “Sky-2” and so on.

```
$ layout -f "PATH:DEL:Pfad 1" test.jpg
```

deletes the path “Pfad 1” from “test.jpg”. The clipping path resource will not be changed by this order. So you should delete or change the clipping path resource also if “Pfad 1” is the clipping path.

```
$ layout -f "CLIP:ADD:exactMask" test.jpg
```

sets the path “exactMask” as the clipping path for the image “test.jpg”.

```
$ layout -f "CLIP:DEL" test.jpg
```

deletes the clipping path definition from “test.jpg”.

```
$ layout -f "XMP:ADD:test.xmp" image.jpg
```

adds the content of the special XMP file “test.xmp” to the file “image.jpg”, replacing all existing XMP metadata. Such an XMP file can be generated either by Photoshop or you can get the XMP information of an image as an XMP file by use of the command `oiimginfo -E xmp -f <exportfile.xmp> file`. The tagging option recognizes special XMP files by their “.xmp” suffix. If the named file would have a suffix other than “.xmp” it will try to open this file as an image, and retrieve the XMP resources of the given image.

```
$ layout -f "RESO:INCH:237,237" test.jpg
```

sets the resolution of the file “test.jpg” to 237 dpi.

```
$ layout -f "RESO:SIZE:10,8" test.jpg
```

defines the image dimensions to 10x8 inches.

```
$ layout -f "RESO:SIZECM:1,2" test.jpg
```

defines the image dimensions to 1x2 cm.

Note: The “oiimginfo” program can be used to list path and ICC profile information. For further details, see 6.4 “oiimginfo”. The PDF HandShake tool “pdfinfo” can also be used to list ICC profiles in a PDF document.

Examples:

```
$ oiimginfo -E path Cafeteria.tif
Path PSID=2000 , Name: Sky , size: 6396
Path PSID=2001 , Name: House , size: 3068
Path PSID=2002 , Name: Table , size: 338

$ oiimginfo -E clip Cafeteria.tif
Table

$ oiimginfo -E icc Cafeteria.tif
1: embedded profile , colorspace: RGB , size: 201892 bytes ,
   Id: (C) Linotype-Hell TOPAZ Fuji Transparent
```

The PDF HandShake “pdfinfo” program can also be used to list tagged ICC profiles:

```
$ pdfinfo -o Profile Duo2.pdf
# pdfinfo 5.0.0
Profile: ColorSpace=DeviceCMYK,
   Path="ICC-Profiles:Printer:HELIOS:Euro 2.6 UCR-370", Reference
Profile: ColorSpace=DeviceRGB,
   Path=ICC-Profiles:Scanner:HELIOS:CCIR-EBU-RGB, Reference
```

Limitations

Layouting or converting in a non-HELIOS volume does work. However, file events are not reported to the HELIOS notification server because the HELIOS file events only notify about events in HELIOS volumes.

In addition, Mac resources like icons, file comments, etc. are not supported in non-HELIOS volumes.

6.2 opitouch

The “opitouch” command allows triggering the generation of low-resolution OPI layout files. Although this could also be done with the “layout” command, the benefit of “opitouch” is that jobs are processed by “opisrv”, which uses the global OPI parameters and OPI folder options for layout generation.

Usage:

```
opitouch [-h host][-s service][-dr] files ...
opitouch [-h host][-s service] -e event ...
```

Important: Make sure that you specify files with their absolute path name!

“opitouch” is the command line version of the program “Touch”, allowing virtual touching of image files *without* modifying the file date or time of the high-resolution files, while causing automatic OPI layout generation. Of course, if “opitouch” is used to trigger a “Tagger” ICC info file to tag ICC profiles to images, then their file date would be changed as a result. “opitouch” allows touching single files, all files of a directory or even a complete directory tree on the local or remote system.

Important: “opitouch” does not recognize the types of the files specified. So touching whole directories containing a non-image file does not produce any error, even if no layout image is generated! However, when touching a single non-image file, “opisrv” issues a syslog error message.

Unless otherwise specified, “opitouch” touches files on the local host.

-h Allows you to select a remote host by name or its IP address. You may only gain access to ImageServer systems with appropriate IP access control setup. This can be done with HELIOS Admin, using the `IP Access` dialog. The remote host must have the OPI server preference **RemoteAccess** set to `TRUE`.

-s Allows you to select the OPI event service port of the “opisrv” by name or port number. Specifying a service port that is served by anything other than “opisrv”, may cause “opitouch” to hang. The default service used is `OPIEvent`, or its default port number `2002`.

Unless otherwise specified (see `-d` and `-r` below), “opitouch” assumes that the path name arguments refer to plain files. In this case, directories will be ignored by the contacted “opisrv”.

-d Requires all path names to be directories. In this case, all plain files within the specified directories will be touched. Subdirectories will be skipped.

-r Requires all path names to be directories. In this case, all plain files and subdirectories contained in the specified directory path names will be touched *recursively*.

Note: Be aware that the usage of this option may cause heavy system load on the OPI server system.

-e This option sends events to the “opisrv”:

```
sendlayout "file name"
sendclose "file name" [file type]
sendrename "file name old" "file name new" [file type]
senddelete "file name" [file type]
sendcreatedir "directory name"
sendrenamedir "directory name old" "directory name new"
senddeletedir "directory name"
```

Example:

```
$ opitouch -e sendclose "/data/images/cover.tif"
```

The optional file type is the Mac four character type code. If the file type is not specified it is determined from the file’s resource fork. For the delete event (`senddelete`) a file type should be specified because it is not possible to determine the type of a file that is already deleted. The file and directory names are distributed to the registered listeners “as is”, the existence of the objects is not verified.

Note: All files and directories must be specified with their absolute path names beginning with a "/" character. The path names must NOT end with a "/" character!

6.3 hirespath

The "hirespath" program can be used for three purposes. In the first mode, it can resolve an OPI layout file name to the appropriate high-resolution path name. It receives the low-resolution file name as a command line argument or on "stdin" and prints the file name of the high-resolution image on "stdout". In doing so, it recognizes certain OPI printer interface parameters (see -o below):

hirespath [-P <printer>] [-d <pathprefix>] [<loResRef>]

<printer>

The name of a printer queue. "hirespath" will then access the OPI parameters for that printer queue.

<pathprefix>

A HELIOS server directory path name which precedes <loResRef> or <clientName>. If this parameter is supplied, <loResRef> or <clientName> has to be a file name without directory specification. <pathprefix> must NOT end with a "/" character!

<loResRef>

File name or reference of the low-resolution image. It can be in the format of the client's file system (Mac, PC). The file name is quoted to allow for spaces. If <loResRef> is omitted, "hirespath" will read <loResRef> from "stdin".

***Result:* <hiResRef>**

File name and path of the high-resolution image. Nothing is printed on "stdout" if the program cannot resolve <loResRef>. The high-resolution file location has the format of the native file system on which "hirespath" is running.

The return code can be any of the following:

0

Success; <loResRef> was successfully resolved into <hiResRef>, the high-resolution file is accessible. <hiResRef> is printed on "stdout".

1

Error; An error occurred. An explanatory error message is printed on "stderr". Nothing is printed on "stdout".

2

Error; <loResRef> cannot be resolved into <hiResFile>. If the theoretical <hiResFile> is reasonable, it is printed on "stdout".

3

Error; <loResRef> can be accessed but is a <hiResRef> and the parameter resolveall is not set. If the theoretical <hiResFile> is reasonable, it is printed on "stdout".

4

Error; <loResRef> can be accessed but is not a valid layout file. Nothing is printed on "stdout".

"hirespath" may also check a complete PostScript job for OPI references and give a detailed report on "stdout". If checking a PostScript job, "hirespath" receives a file name of the job to check as a command line argument or uses "stdin" as source:

hirespath -j [-P <printer>] [<PostScriptFileName>]

-j

When the `-j` option is specified, the argument to "hirespath" is a `<PostScriptFileName>`. "hirespath" will then check the PostScript job for all OPI references and check each reference as described above. On exit a report will be printed on "stdout" as a list of `<opiRef>` references, each on a single line.

The return code can be any of the following:

`0`
Success. All OPI references could be resolved.

`1`
Error; an (UNIX) error occurred. An explanatory error message is printed on "stderr". Nothing is printed on "stdout".

`2`
Error; one of the OPI references could not be resolved.

<PostScriptFileName>

Name of a file containing a PostScript print job.

"hirespath " returns the name or reference on "stdout", and returns an integer to indicate the result of the operation:

Result: **<opiRef>**

`<code>\t(<reference>)\t(<resolvedName>)`

`<code>` is one of the return codes listed above, `<reference>` is the OPI reference found in the PostScript file and `<resolvedName>` is the name of the replaced image. `<resolvedName>` may be empty. The names are quoted in `()`, using the ADSC quoting conventions.

In a third mode, "hirespath" can map client (Mac or PC) file names to host path names:

hirespath -m [-d <pathprefix>] [<clientName>]

-m

When this option is specified, “hirespath” is a <clientName> and the result will be a <hostPath>. In this case, “hirespath” will map a client file name <clientName> to a host system path name <hostPath> without OPI resolving. This works for Mac and PC client file names. There are a few restrictions:

- Mac client file names must begin with an AFP volume name as specified in the volume preferences
- PC client file names must refer to existing files or directories on a PCShare volume

The return code can be any of the following:

0
Success; <clientName> was successfully mapped into <hostPath>, and the file is accessible. <hostPath> is printed on “stdout”.

1
Error; An error occurred. An explanatory error message is printed on “stderr”. Nothing is printed on “stdout”.

2
Error; <clientName> cannot be mapped into <hostPath> or <hostPath> is not accessible. If the theoretical <hostPath> is reasonable, it is printed on “stdout”.

A file that resides on a HELIOS host system may have its name presented in a different syntax when viewed from a client system. The “hirespath” -m option will accept any <clientName> file name, in a client system syntax. The output will be the corresponding path and file name of <clientName>, in the native file naming syntax of the HELIOS host. This is a name mapping function, and does not require that the file be a layout file or PostScript job. Even if the <clientName> file does include OPI references, they will be ignored, and only the <clientName> file name will be mapped to a host system file name.

<clientName>

File name of a Mac or PC file. If `<clientName>` is omitted, "hirespath" will read `<clientName>` from "stdin".

Result: **<hostPath>**

Mapped UNIX path name of `<clientName>`.

In this case, "hirespath" will map a client file name `<clientName>` to a host system path name `<hostPath>` without OPI resolving.

6.3.1 Additional "hirespath" arguments

"hirespath" always understands the arguments:

-e Enable explanatory error messages of unresolved references.

-o Set an option using the `-o Parameter=Value` syntax. **ImageSearchPaths**, **ImageSearchVolumes**, and **ImageIDSearch** (see 9.1.5 "OPI-related printer queue preferences") can be specified.

6.3.2 Example

The example shows the command you have to enter if you want the program to return the path name of a high-resolution image for a given layout image in a print job:

```
$ hirespath -P lw /Volumes/OPI-Test/layouts/Band.tif
```

The server returns the following statement:

```
/Volumes/OPI-Test/Band.tif
```

6.4 oiimginfo

"oiimginfo" prints information about a given image file. The `-v` option can be used to restrict the amount of output. The default is to print all OpenImage-related information.

If the `-E` option (export mode) is set, only special information is printed to "stdout" or to a file.

Usage:

```
oiimginfo [-v] [-a] [-E [-I] [-f]] [-p] [-h] file
```

6.4.1 Options

-v <VerboseLevel>

Restrict the "oiimginfo" output to the given levels. See 6.4.2 "Verbose/Filter levels" below.

-a <Attributes>

Comma-separated list of attributes for the image manager. See 6.4.4 "Attributes for "oiimginfo"" below.

Example:

```
oiimginfo -aPageNumber=28
```

-E <ExportType>

Print the selected type of information to standard output or to a file. See 6.4.3 "Export" below.

-I <ExportId>

Select a particular object of the given `<ExportType>`, used e.g. with text boxes.

-f <ExportFile>

Print the export information to `<ExportFile>` instead of standard output.

-p

Provides basic information (e.g. path, class and colorspace) on the *ICC profile*, identified by the string.

-h

Display help file.

6.4.2 Verbose/Filter levels

`-v <VerboseLevel>` is a comma-separated list of keywords.

Keywords are case-insensitive and can be abbreviated as long as the abbreviation is unique.

Filter keywords

Filter keywords can be used to specify which objects in the image are described in detail, depending on the image type, quality or access method.

Type filter keywords:	`Image,Plate,OtherTypes`
Quality filter keywords:	`Print,Proof,Screen`
Access filter keywords:	`Raster,PostScript,Unspec`

If a filter keyword is given for one of these categories, only the corresponding objects in the image are described in detail.

Selector keywords

Selector keywords can be used to specify the amount of information printed.

Generic selector keywords:	`ObjBase,LFile,RFile,Manager`
Default:	`ObjBase`
GFX selector keywords:	`GFXBase,GFXInk,GFXICC,GFXAttributes, GFXResources,GFXVerboseResources`
Default:	`GFXBase,GFXInk,GFXICC,GFXAttributes, GFXResources`

`NoGFX` can be used to suppress all GFX output.

Shortcuts:

```
GFXContent = GFXBase,GFXVerboseResources,GFXAttributes
IInfo      = Image,GFXContent
RImage     = Image,Print,Raster
PImage     = Image,Print,Postscript
XPVInfo    = Image,Print,GFXVerboseResources,GFXAttributes
```

6.4.3 Export

With the `-E <ExportType>` option, special types of information can be extracted from the image file and printed to "stdout" or to the file given by `-f <Export-File>`.

`<ExportType>` can be one of: `XMP`, `Text`, `Attributes`, `Base`, `Clip`, `IPTC`, `Path`, `ICC`.

`<ExportType>` is case-insensitive and can be abbreviated, as long as the abbreviation is unique.

The `-v` option can be used with filter keywords to select an object. If there is more than one object containing the wanted export type, the content of the best object is selected.

Export types in detail:

XMP Dumps the XMP information of a file. You can set an empty file name to generate an `<ImageFileName>.xmp` in the directory of the image file.

Text Dumps the contents of all text boxes or of the text box specified by `-I <ExportId>`. You can set an empty file name to generate the file `<ImageFileName>.txt[number]` in the directory of the image file.

Attributes Dumps the attributes of an image.

Base Dumps the basic image information of an image.

Clip Dumps the name of the clipping path.

IPTC Dumps the binary content of the IPTC objects.

Path Without `-I` you get an overview of all included paths. `-I <PathName>` dumps the binary path.

ICC Without `-I` you get an overview of all included profiles. `-I <ProfileNumber>` dumps the binary profile.

Example:

```
$ oiimginfo -E PATH -I Sky -f Sky.path export.tif
```

Export the path "Sky" from "export.tif" into the file "Sky.path", that can later be added by the "layout" program to another image, see 6.1.12 "Tagging options".

6.4.4 Attributes for "oiimginfo"

PageNumber <uint32:1>

This attribute allows you to specify a certain page of a document for which image information is provided.

Example:

```
$ oiimginfo -a PageNumber=3 test.xpv
```

6.5 psresolve

Note: Technically speaking, "psresolve" is a HELIOS Base utility program. However, most of the options are set in an ImageServer environment. Therefore, "psresolve" is described in this manual.

The “psresolve” program acts like a printer driver, e.g. “smbif”. Instead of sending the PostScript output to the network, “psresolve” passes all PostScript data to “stdout” which can be redirected into a file or piped to an application.

The “psresolve” program lets you set printer interface OPI options manually on the server for one single print job. These options override the settings in the preference database for the specific spooler:

```
psresolve [options] [SpoolerName]
```

In the following, we list the parameters that may be set with “psresolve” and then – at the end of this chapter – give you an example that may be used for a specific workflow.

6.5.1 Options

Printer interface related parameters are entered as command line options using the `-o Preference=Value` syntax.

OpiEnabled <boolean:FALSE>

This option turns OPI functionality on or off. If it is set to `TRUE`, OPI image replacement is enabled.

ImageSearchPaths <string list:None>

Specifies a list of path names where to search high-resolution images during image replacement, in case they cannot be found in the standard locations.

ImageIDsearch <boolean:TRUE>

This parameter enables the use of the HELIOS desktop database to locate moved image files.

PrintDraft <boolean:FALSE>

If set to TRUE, the printouts from the specific printer queue will only contain layout images. The copies of the images – which you have used for layouting your document – will be sent to the OPI server and then be replaced by the layout files instead of the high-resolution originals.

CheckFonts <boolean:TRUE>

This parameter ensures that all fonts used in the document will be checked before printing. Every missing font will be reported and this will stop the print job.

CheckImages <boolean:TRUE>

This parameter ensures that all images placed in the document will be checked before printing. Every missing image will be reported and this will stop the print job.

CheckICCProfiles <boolean:TRUE>

Specifies whether existence of ICC image profiles has to be verified before printing. The server will automatically stop the print job if a single profile is missing and issue a corresponding error message.

CompatCheck <boolean:TRUE>

Specifies whether special PostScript setup has to be included to avoid application specific problems during printout. This is necessary for FreeHand 3.1, PageMaker 5 and QuarkXPress 3.0.

ResolveAll <boolean:FALSE>

If this parameter is set to TRUE, all images will be replaced during printout. Otherwise only layout images are resolved.

PureBlack <bool:FALSE>

Determines whether black pixels are excluded from color matching.

PureWhite <bool:FALSE>

Determines whether white pixels are excluded from color matching.

PureGrays <bool:FALSE>

Determines whether gray pixels are excluded from color matching.

PureCMY <bool:FALSE>

This option determines whether pure *Cyan*, *Magenta* or *Yellow* pixels are excluded from color matching in CMYK to CMYK conversions. Pure Cyan pixels have 100 percent Cyan and 0 percent for other colors. Pure Magenta pixels and pure Yellow pixels are defined similarly.

KeepInfo <boolean:TRUE>

Specifies whether in case of resolving an already resolved PostScript job, the use of fonts and other resources should still be accounted.

DefaultPrinterProfile <string:None>

This parameter specifies the path name of the default ICC profile describing the printing device.

DefaultProofProfile <string:None>

This parameter specifies the path name of the default ICC profile describing the proof device. The string will only be recognized if a default printer profile is set.

DownSampling <boolean:FALSE>

This option – if it is set to `TRUE` – allows downsampling of images to a given output resolution. The value that has to be used for printing is specified by the `Resolution` parameter below.

FixedSampling <boolean:FALSE>

This option allows fixing the downsampling resolution (see the `Resolution` parameter below) to the given value. Usually, if this

parameter is set to FALSE, the output resolution is adjusted to the image resolution if the image resolution is smaller. When setting this parameter to TRUE the printer interfaces will use the given Resolution as is for downsampling. This may cause upsampling of images with resolutions smaller than the given one. This parameter makes only sense when setting the DownSampling parameter to TRUE.

FastDownSampling <boolean:FALSE>

This option controls the downsampling algorithm. By default, a bilinear algorithm is used. If you set this option to TRUE you switch to a fast picking algorithm. This parameter is only meaningful when setting the DownSampling parameter to TRUE.

Resolution <double:0.0>

This option controls the resolution of downsampled images. Zero induces the software to use the default resolution of the printing device as specified in the queue's PPD file. This parameter is only meaningful when setting the DownSampling parameter to TRUE.

ProcessColorspace <string:"CMYK">

This option sets the color space used while printing separations. The default is CMYK for a four color print process. Spot colors are not affected by this parameter. Valid color spaces are "CMYK" and "Multi". If you set the string to "Multi" you have to define the names and order of the color components. Furthermore, the usage of the "Multi" color space requires the use of ICC profiles. For (future) ICC profiles that include the names of the color components, this option will become obsolete. Please note that the value you choose for this option can be overridden by the printer profile you have selected using the DefaultPrinterProfile option.

ProcessInks <string list:"Cyan","Magenta","Yellow","Black">

This option controls, together with the ProcessColorspace option, the names of the process inks used for separations. Spot colors are not affected by this parameter.

CompositeColorspace <string:"CMYK">

This option sets the color space used while printing composite. The default is `CMYK`. Setting this option to `None` causes all color images in a print job to be kept and printed in their original color space. This applies to CMYK, CIELab and RGB images only. Valid strings are `CMYK`, `RGB`, and `CIELab`. Please note that the value you choose for this option can be overridden by the printer profile you have selected using the `DefaultPrinterProfile` option.

CompressPostScript <string:None>

This option selects a compression mode for images while being printed. CCITT Group 4, JPEG (Low to Max.), ZIP, and Compress are currently supported.

MaxLowResolution <double:72.0>

When printing layout images, ImageServer is able to scale down a high-resolution image to layout quality on the fly if the layout image cannot be found. This parameter controls the maximum resolution used for printing these transformed images.

IgnoreMissingLowRes <boolean:TRUE>

Use this parameter to specify whether non-existing layout images should be ignored when printing layout quality. If set to `FALSE`, and any layout images are missing, then the print job will be passed to the error queue.

TemporaryTagging <boolean:TRUE>

This parameter enables the use of the ICC info files while printing images. If it is set to `TRUE`, every non-tagged image will be temporarily tagged during printing provided that the ICC info file specifies tagging for that kind of image.

IgnoreUntagged <boolean:TRUE>

This parameter controls whether non-tagged images should be ignored when printing to an ICC color matched queue.

ScitexApr <boolean:FALSE>

This parameter – if it is set to TRUE – tells ImageServer not to resolve OPI comments if printing a PSImage file to an APR-aware RIP.

Note: The parameters which are described in the following two paragraphs will usually be specified when setting up EtherShare. They may, however, be relevant to ImageServer as well and can therefore be set as preferences, too.

IgnoreResolveOpts, IgnoreProcsetResolveOpt, IgnoreFontResolveOpt, IgnoreIncludeResolveOpt and IgnoreOpiResolveOpt <boolean:FALSE>

When printing from a HELIOS spooler to a second HELIOS spooler, print jobs will usually not be resolved a second time. Specifying one or more of these options for the second spooler re-enables resolving of fonts, procsets, included files or *OPI comments*.

ExtendedInfo <boolean:TRUE>

Enables explanatory accounting. The accounting file will e.g. include the OPI image replacement list.

6.5.2 Example

Imagine you want to print a document using the `CompressPostScript` option. For that purpose you have to change the default setting for this particular print job using the "psresolve" program. The complete procedure is described below:

➢ First of all, set up two different printer queues using HELIOS Admin. One queue (e.g. "real_printing") is to be used for printing to an output device and the other one (e.g. "print_to_disk") is to be used for printing to a PostScript file (spool only).

➢ Make sure that `Spool only` has been activated for the "print_to_disk" queue (`Printer` menu) and then print to this queue from your layout application.

➢ Then, on your server, change to the spool directory that now contains your print job (e.g. "/usr2/spool/print_to_disk"):

```
$ cd /usr2/spool/print_to_disk
```

➢ Start the "psresolve" program, enter the parameters you want to use, and specify the queue name, the input file name (e.g. "your_print_job") and the output file name:

```
/usr/local/helios/bin/psresolve -o
CompressPostScript=Compress print_to_disk
<your_print_job >your_print_job_output
```

Finally, you may send the file "your_print_job_output" to your `Spool & Print` printer queue for output:

```
/usr/local/helios/bin/
lpr -Preal_printing your_print_job_output
```

Please note that the "real_printing" printer queue will not resolve your print job again. The job will be printed according to the settings that had been specified for the "print_to_disk" queue using HELIOS Admin and/or according to the parameters set by the `psresolve` command line.

6.6 psrip

The "psrip" utility is a command line PostScript RIP application which is suitable for creating previews of vector EPS and PostScript files.

Usage:

```
psrip [-v] [-o Option] [-m] [-T] [-A] [-l <level>] [-V version]
      [-e <xpixel,ypixel>] [-p <width,height>] [-s <xshift,yshift>]
      [-N <serialno>] [-i <InputFileName>]
      <OutputFilenameBase>
```

The following options are supported:

-v Display progress.

-o PrintResolution=<resolution> or <XResolution,YResolution>

Device resolution in dpi (default is `72.0`). Ignored, if page dimension in inch (`-p`) or page dimension in device pixels (`-e`) is set.

-o PrintColor=<color space>

Select device color space `CMYK`, `RGB` or `Grayscale` (default is `CMYK`). For best color space conversion of EPS images, specify the image's current color space for PrintColor, and then use "layout" to perform an ICC color space transform on the output file.

-o inripseparation

Enable in-RIP separation. One image file per each colorant is written, unless you want to produce a multichannel image file (`-m`).

-o TransparencyMask=<flag>

`<flag>` can be "on" or "force". "on" induces "psrip" to generate a transparency mask, as long as the PostScript file contains transparencies. "Force" enforces the generation of a transparency mask even though there are no transparencies in the document and hence the mask is empty. In doing so, a consistent paging within the output file(s) is guaranteed.

The transparency mask is saved into a separate grayscale file by "psrip", unless a multichannel image file is generated (see `-m` option below). In such a case, the transparency mask appears as a separate channel in the output file.

-o KeepAspectRatio

This flag is an addition to the `-e` option. The page dimension in device pixels of the raster image has the same aspect ratio as specified by width and height (`-p`). The `-e` option then specifies maxima: The raster image is smaller than or equal to these values.

This option has no result if the device resolution is set (`-o PrintResolution`).

-o SeparationOrder=<separation order list>

List of colorants, written as output. This option does only work with the `inripseparation` option. The value is a list of colorant names, separated by commas.

If a colorant name contains spaces, the whole list must be surrounded by quotes. The colorant names must be defined either implicitly by color space (Cyan, Magenta, Yellow, Black, Red, Green, Blue or Gray) or by `SeparationColorNames`. If `SeparationColorNames` is not used, each unknown colorant (not implicitly defined by color space) is supposed to be a spot color.

-o TextQuality=<str:"high">

Determines the text rendering quality. There are two values available, “high”, which is the default, and “low”. “high” means that a font-specific algorithm is utilized for rendering the glyphs more accurately. “low” has the advantage of faster rendering.

-o ImageQuality=<str:"high">

Determines the image rendering quality. There are two values available, “high”, which is the default, and “low”. If an image has a higher resolution than the output device, and is therefore downscaled, a better image rendering algorithm is used. This needs some time, so this option is primarily meant to switch this algorithm off.

-o SeparationColorNames=<spot color list>

List of available spot colors. This option does only work with `SeparationOrder` and the `inripseparation` option. The value is a list of colorant names, separated by commas.

If a colorant name contains spaces, the whole list must be surrounded by quotes. If this option is specified, only colorant names

defined here can be used in `SeparationOrder` as spot colors. If a PostScript job uses a separation or DeviceN color space, the color space tint transform is not used, if the according (spot) color(s) are known (and thus listed for this option).

-o RasterImageType=<image format>

The default raster image format is `TIFF`. If ImageServer is not installed, this option is not available and images are written as raw files. Supported output file formats are listed in 6.6.1 “Output file formats”.

Image format specific settings, e.g. quality, can be specified via attributes (see `-o Attributes=<attributes>` below).

-o RasterImageSuffix=<output file suffix>

Suffix of the raster image output file. The default is derived from the image format, e.g. `.tif` for TIFF.

-o CompressQuality

Defines the compression quality of output image files or image components:

For JPEG:

-o CompressQuality <uint32:75>

Creates poor…high quality JPEG image (1…100).

For JPEG 2000:

-o CompressQuality <uint32:0>

Specifies the image quality of a JPEG 2000 image in relation to the uncompressed high-resolution original. The values range from 1…100. Specifying “0" means *lossless*

Note: `CompressQuality` specifies the percentage size of the JPEG-2000 image data compared to the size of the uncompressed image, which the JPEG 2000 should not exceed. For example, if an image with lossless compression has only 50% the size of an uncompressed image, a `CompressQuality` value between 50% and 100% produces no visible differences.

-o CompressPrint=<compression>

Compression used by the raster image file. Some raster image file formats support several compressions (e.g. Zip compression in TIFF files).

By default, there is no compression method set. To use a compression method, specify this option with one of the following supported methods:

`Flate` Zip compression format
`JPEG` JPEG compression format

-o vmsize=<VM size>

Size of the local VM in bytes (default is 60 MB). If `vmsize` ends with the suffix "K" or "M", kB or MB are specified, e.g.: `-o vmsize=75M`.

-o gvmsize=<global VM size>

Size of the global VM in bytes (default is 10 MB). If `gvmsize` ends with the suffix "M" and "K", kB or MB are specified, e.g.: `-o gvmsize=15M`.

-m Use this option and enable in-RIP separation (`-o inripseparation`) to create multichannel image files instead of one file per separation. The following image formats support multichannel image files:

```
TIFF
Adobe Photoshop
JPEG 2000
```

Note: Photoshop displays spot colors in multichannel images as process colors during the presentation. "psrip" does the calculation by use of spot color tables (see "Proof printing files with spot colors" in the HELIOS PrintPreview manual).

-T Create an image with a transparent background. This option is only available for TIFF and PNG images (see `-o RasterImageType=<image format>`).

-A Do antialiasing (four-times oversampling).

-l (level)

PostScript language level (1, 2 or 3). Default is `2`.

-V <version>

Set the PostScript interpreter version (e.g. `3015.102`).

-e <xpixel,ypixel)>

Page dimension specified in device pixels. `xpixel` and `ypixel` must be separated by a comma. The default values are `595` (xpixel) by `842` (ypixel).

-p <width,height> or -p <paper format>

Page dimension specified in inches. `width` and `height` must be separated by a comma. The default is `8.26` (width) by `11.69` (height). Alternatively, the paper format can be specified. The default is `<A4>`.

Supported page sizes:

`Letter, Legal, Executive, Ledger, A0, A1, A2, A3, A4, A5, B0, B1, B2, B3, B4, B5, B6`

-s <xshift,yshift>

Shift coordinate system in inches. Positive is direction upper/right. `xshift` and `yshift` must be separated by a comma. The default values are `0` (xshift) by `0` (yshift).

-N <serial number>

PostScript RIP serial number.

-i <input file name>

File name of the PostScript or vector EPS file. If no file name is specified, PostScript is read from "stdin" (reading from "stdin", "psrip" **cannot handle** binary EPS headers).

If PostScript is read from a file (instead of "stdin"), "psrip" **can handle** files containing a binary EPS header.

OutputFilenameBase

Output files will be named:

`<OutputFilenameBase>.<PageNumber>.<OutputFileSuffix>`

or if in-RIP Separation is enabled (`-o inripseparation`) and one image file per colorant is written:

`<OutputFilenameBase>.<PageNumber>.<ColorantName>.<OutputFileSuffix>`

Multichannel image files are named in the standard manner of output files.

-o Attributes=<attributes>

Attributes specifying details of the used image format. This option is only available, if ImageServer is installed. `<Attributes>` is a comma-separated list of `key=value` pairs. For a list of available attributes see 6.6.2 "Attributes for "psrip" options".

-h Display help file.

6.6.1 Output file formats

The default raster image format is TIFF. To use a different format specify the `-o RasterImageType` option with one of the following supported formats:

`TIFF`	for TIFF image format
`JPEG`	for JPEG image format
`JP2`	(*4 chars!*) for JPEG 2000 image format
`8BPS`	for Adobe Photoshop format
`PNGf`	for PNG image format (RGB and Grayscale only)
`PICT`	for Apple PICT image format (RGB and Grayscale only)
`..CT`	for Scitex CT format

6.6.2 Attributes for “psrip” options

Attributes specify details of the used image format. Attributes are entered as command line options using the syntax (additional attributes are delimited by commas):

```
psrip -o Attributes=<attributes>=<value>
```

(deprecated but possible for compatibility reasons): `psrip -a <key>=<value>`

Baseline <boolean:FALSE>

(*JPEG only*) Saves JPEG images per default in progressive method. If this option is set to `TRUE`, JPEG images are saved as baseline JPEGs instead.

DctMethod <uint32:0>

(*JPEG only*) Use slow, fast or precise DCT (*Discrete Cosine Transformation*). Values are `0`, `1` or `2`. The default is `0`.

TileWidth <uint32:256>

(*JPEG 2000 only*) Specifies the horizontal size of JPEG 2000 tiles. A JPEG 2000 image file consists of juxtaposed tiles. If the value specified is `0`, the whole image merely consists of one tile.

TileHeight <uint32:256>

(*JPEG 2000 only*) Specifies the vertical size of JPEG 2000 tiles. A JPEG 2000 image file consists of juxtaposed tiles. If the value specified is `0`, the whole image merely consists of one tile.

Note: Partitioning a JPEG 2000 in tiles accelerates encoding and decoding significantly and minimizes the amount of required memory during this time. The default value 256 makes a good compromise between speed and memory requirement.

The following two attributes are only supported for backward compatibility with previous ImageServer versions; please do not use these options for new installations or configurations. Use the option `-o CompressQuality` *instead:*

Quality (JPEG only!) <uint32:75>

Create poor … high quality JPEG image (1 … 100).

Quality (JPEG 2000 only!) <uint32:0>

Specifies the image quality of a JPEG 2000 image in relation to the uncompressed high-resolution original. The values range from 1…100. Specifying “0" means *lossless*.

6.6.3 Examples

Example: (Convert a PostScript page)

```
psrip -o PrintColor=Grayscale -p 10,15 -o RasterImageType="JP2 "
      -i example.ps example
```

Prints a PostScript job onto a 10x15 inch Grayscale page and writes the result into a JPEG 2000 file (“example.1.jp2”).

Example: (Convert a vector EPS image to the bounding box size)

```
psrip -o RasterImagetype=JPEG -i example.eps example
```

Converts a vector EPS file to a JPEG file. Since no page formats or sizes are specified, the image is converted to the bounding box size.

6.7 pdfresolve

The “pdfresolve” program allows replacing OPI images in PDF documents. In addition, “pdfresolve” preserves transparencies, that have been applied to layout images, in the OPI image replacement process. A complete description can be found in the HELIOS PDF HandShake manual.

7 HELIOS Script Server

HELIOS ImageServer includes the Script Server service, which implements a so-called hot folder mechanism. The idea of a hot folder is that, when files are dropped into this folder, a process is automatically started, acting upon the files according to a given script.

Major features of the Script Server are:

- It is a server-based solution that utilizes the HELIOS event notification service, which is much faster and efficient than polling
- Only events containing the folder path are processed by the Script Server
- Easy local or remote setup and monitoring via HELIOS Admin
- Monitor current jobs and configurations on telnet port 2024
- General purpose scripts can be re-used for different purposes by specifying environment variables in HELIOS Admin. Pre-defined script environment variables are e.g. event type, event user, hot folder path
- Support for multiple file types or suffixes as well as directories
- Less traffic for wildcard registrations
- Process auditing and tracing compatibility

Any folder within a HELIOS volume can act as a hot folder for the Script Server. HELIOS provides a variety of sample scripts for different file formats and purposes.

A detailed description on starting/stopping and refreshing the Script Server process, the included sample scripts, and on debugging scripts can be found in 7.3 “Start, stop and refresh the Script Server process” and 7.4 “Included sample scripts”.

7.1 Script Server settings

The HELIOS Script Server is configured in HELIOS Admin.

➢ From the `Settings` menu in HELIOS Admin open the “Script Server Settings” window (Fig. 7.1).

The `Script Delay` field allows specifying the time interval before the script is started after the file event is received. `Script Timeout` specifies the maximum script runtime. The value `0` may let the script run forever. `Simultaneous Scripts` determines the maximum number of concurrently executing scripts in order to limit server load. The value for `Simultaneous Scripts` should be less or equal to the number of physical CPUs. If you leave this field empty Script Server determines an appropriate value by checking the number of CPUs.

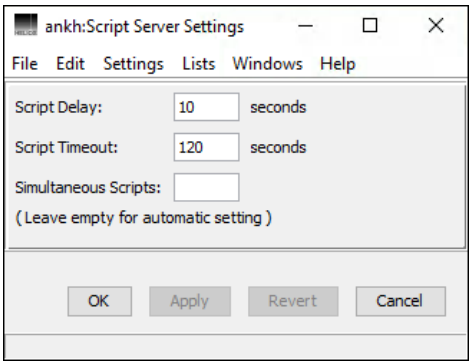


Fig. 7.1: HELIOS Admin “Script Server Settings” window

7.1.1 Configure a script queue

In the Scripts (Fig. 7.2) tab you can set up script queues with different configurations for different purposes.

Note: If the Scripts tab is not available after the first login, activate Scripts in the Lists menu.

All script queues configured on the host are listed in this window. The HELIOS Admin server automatically creates it by inspecting script related entries in the "Preferences" file.

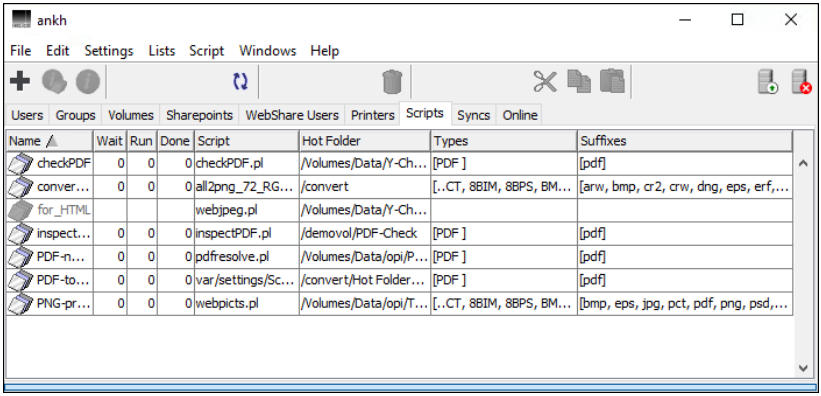


Fig. 7.2: HELIOS Admin Scripts tab

The Name column shows the script queue name, while the used script itself is displayed in the Script column.

Wait, Run and Done list the status of the script queues. Wait means that they are spooled but not yet processed. Run signifies that a script queue is being processed, and Done that the script has finished.

The path to the specified hot folder, the place where the script “looks” for files or directories to be processed, is displayed in the `Hot Folder` column. `Types` and `Suffixes` list all defined file types and file suffixes per queue.

Change script data

Script queue settings for a particular script queue are changed in the `Script` window (Fig. 7.3):

➢ Select the desired script queue from the list and choose `Settings` from the `Script` menu.

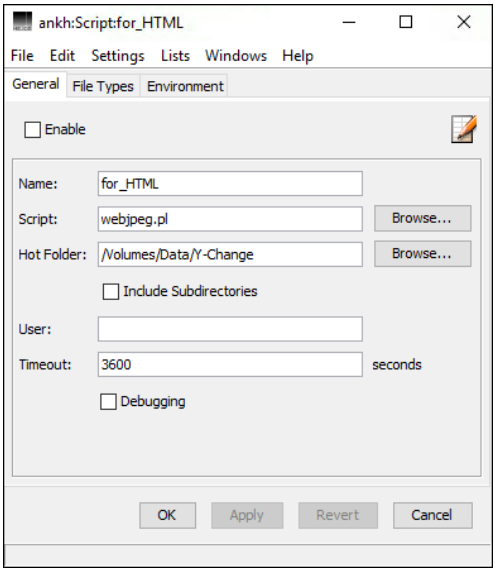


Fig. 7.3: HELIOS Admin “Script” window – `General` tab

The `General` tab (Fig. 7.3) shows the general settings for the particular queue. The `Enable` checkbox activates the queue for the Script Server. If a script

queue is not enabled the queue name in the Scripts list appears grayed out (see "PDF_print" entry in Fig. 7.2).

➢ Specify a name for the script queue in the Name field and the desired script in the Script field. You may write the script path and name directly into the field or select it by use of the Browse... button.

Note: If two or more script queues use the same hot folder, and do not differ in file type or suffix specification, the scripts are executed in alphabetical order. You may have to rename the script queue names according to your needs.

A click on the icon (see arrow in Fig. 7.3) opens the selected script file for editing. You may also choose Open from the File menu.

➢ Specify the path to the hot folder in the Hot Folder field, either manually or via the Browse... button.

Important: Make sure that the specified path already exists. Otherwise Script Server will ignore the configuration!

The Include Subdirectories checkbox is inactive by default. This avoids subdirectories in the specified path being considered as hot folders as well. However, if it is desired that all subdirectories underneath the specified path are considered, this checkbox must be switched on.

The user name the script should run as can be entered in the User field. If this field is left empty, the script is executed as *superuser*.

Note: On Windows systems the User field requires setting the existing user password again using the HELIOS "authutil" tool, e.g.: `authutil passwd -n user -p passwd`. This information will be used to run the script under the specified user account.

The Timeout field determines the maximum script runtime. The value 0 may let the script run forever. If a value is specified in this field, it overrides the global setting (compare to Fig. 7.1).

The Debugging checkbox specifies whether verbose output is written. The value is exported as SCRIPTDEBUG environment to the script.

To make the hot folder react only on certain file types and suffixes, you must define these for the script queue. File types and suffixes are added in the File Types tab (Fig. 7.4).

➢ Enter the 4-character file type code into the respective field and click the Add button. Do likewise for the file suffix.

Note: The file type specification always demands 4 characters, even if blanks are included, e.g. “PDF ”.

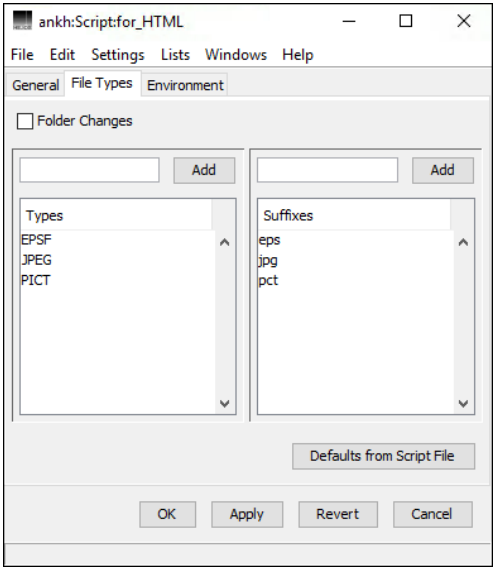


Fig. 7.4: HELIOS Admin "Script" window – `File Types` tab

Enable the option `Folder Changes` if you wish that directory events rather than file events are notified to the Script Server. In this case, file events are ignored and the `Types` and the `Suffixes` column in the `Scripts` list appear grayed out and the settings are ignored. In addition, the leading icon changes from a "files" symbol to a "directory" symbol (see third entry in Fig. 7.2).

Script Server scripts can be formatted in such a way (see 7.1.2 "Automatic Script Server configuration"), that specified variables and their default values are shown in the `Environment` tab (Fig. 7.5). Such variables can be viewed and edited by double-clicking the variable name, entering the new value, and then clicking `Add`.

In addition, other variables in the script can be entered here, along with their value. Note that the script must check for (and use) these declared environment values in order for them to have any effect.

Important: Environment variables are handed over to Perl via shell. Therefore, used characters are limited to ASCII. This is because some shells (especially on Windows) as well as Perl are not able to handle all UTF-8 characters.

To remove an entry from the `Environment` list, highlight it and select `Clear` from the `Edit` menu. Another way is to double-click the entry, which makes it disappear from the list, whereas variable and value are displayed in the definition fields.

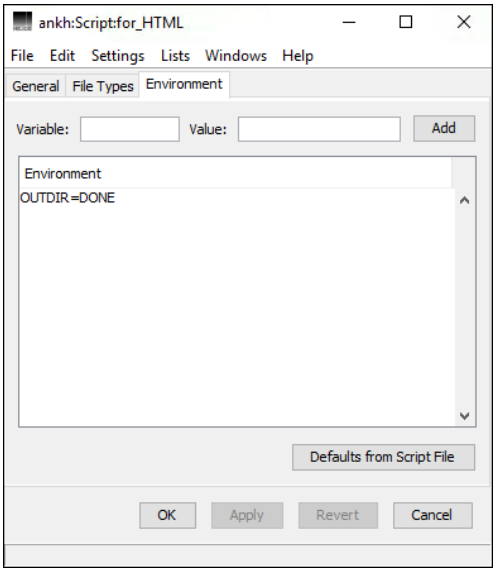


Fig. 7.5: HELIOS Admin “Script” window – `Environment` tab

Create a script queue

A new script queue is created as follows:

➢ In HELIOS Admin activate the `Scripts` tab and choose `New` from the `File` menu.

If you wish to create a new script queue that should adopt most of the settings of an already existing queue, a very convenient way is to duplicate the script queue data by copying a queue entry from the `Scripts` list and then pasting it into the list. A script queue data window opens to adjust the settings for the new queue. Instead of copying/pasting the script queue entry you may also highlight it in the `Scripts` list and drag it with the mouse within the list.

After filling in the script queue data window and confirming the changes, the new script queue is added to the list. If the script queue data window is closed without saving, the new script queue entry is discarded.

Delete a script queue

A script queue is deleted as follows:

➢ In the `Scripts` list highlight the script queue that you wish to delete and choose `Clear` from the `Edit` menu, or just click the toolbar trash can icon.

7.1.2 Automatic Script Server configuration

When setting up a new script queue, HELIOS Admin can read out the configuration from the referenced script and complete the respective fields with default values. For this to work, the script must contain the configuration data in the *SETTINGS* section. Furthermore, the script must be selected via the `Browse...` button (compare Fig. 7.3). The following example shows, in a script excerpt for the queue “For_HTML”, the defined default configuration values:

```
# $Id: webjpeg.pl
```

```
# Sample configuration
#
my $default_settings = <<'</SETTINGS>'; # Let Perl ignore the settings block
<SETTINGS>
<General
        Enable="true"
        Hot_Folder="/demovol/webjpeg"
        Include_Subdirectories="false"
        User=""
        Timeout=""
/>
<File_Types
        Types="TIFF,JPEG,EPSF,8BIM,8BPS,PICT,BMP ,PNGf,..CT,PDF "
        Suffixes="tif,jpg,eps,psd,pct,bmp,png,sct,pdf"
        Folder_Changes="false"
/>
<Environment
        OUTDIR="DONE"
/>
</SETTINGS>
```

Each time a script has been selected via the `Browse...` button in the `General` tab, an additional button – `Defaults from Script File` – becomes available in both the `File Types` and `Environment` tabs (Fig. 7.6 and Fig. 7.7). `Defaults from Script File` allows restoring the default settings which are defined in the selected script.

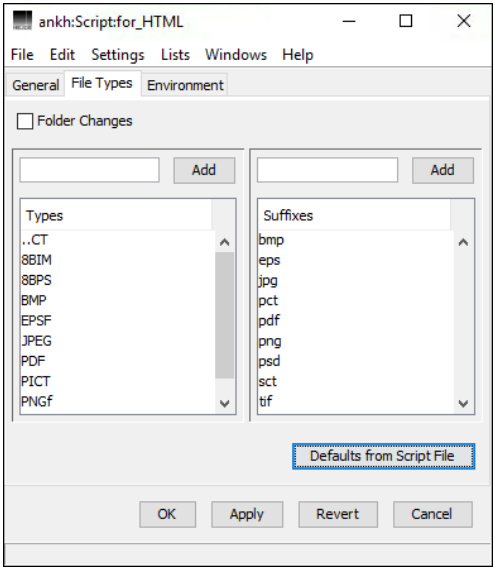


Fig. 7.6: HELIOS Admin “Script” window – `File Types` tab

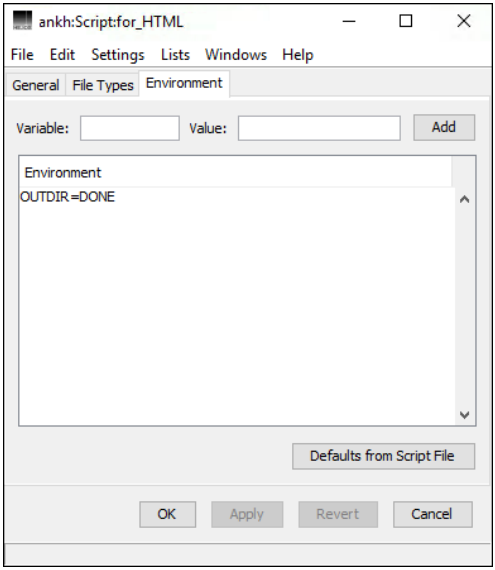


Fig. 7.7: HELIOS Admin “Script” window – `Environment` tab

7.1.3 “Script Jobs” window

The status of a script queue is displayed in the “Script Jobs” window (opened by double-clicking the script queue). As soon as a script starts being spooled, “Waiting” is displayed in the `Script Status` column. When it is being executed, the status changes to “Running”. The moment the script has been accomplished, “Running” disappears from the `Script Status` column. The accounting (see arrow in Fig. 7.8) informs about the status of all processed jobs, and additionally counts the accomplished jobs (“Done”).

Note: The accounting is reset if the configuration is changed or the “scriptsrv” process is restarted.

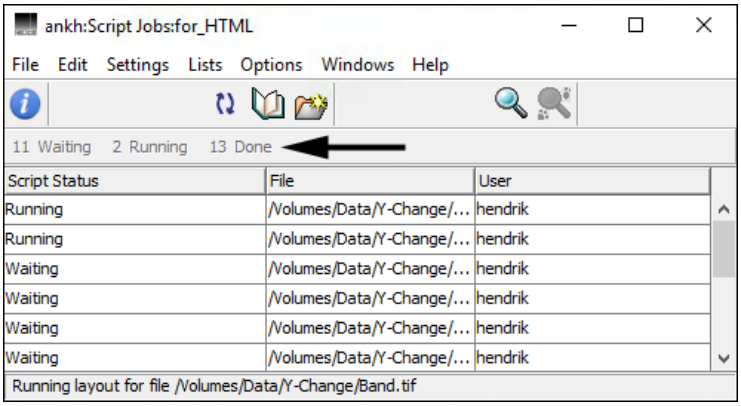


Fig. 7.8: HELIOS Admin “Script Jobs” window

7.2 Script Server log files

The Script Server log file lists script actions or Script Server processes, with date and time (Fig. 7.9).

Script Server log files are arranged by date. Every night at midnight the log files are reset, i.e. the logs of the last seven days are renamed; the file “Today” becomes the file “Yesterday”, “Yesterday” is renamed into “Two Days Ago”, and so on. You can then select, for instance, the server log file of “Three Days Ago”. Log files which are older than seven days are deleted automatically.

➢ Select `Script Server Log Files` from the `Lists` menu and specify the required day.

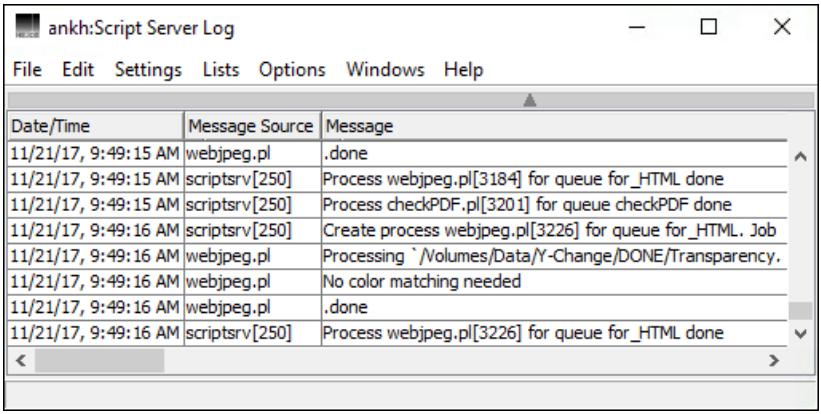


Fig. 7.9: Script Server log file on host “ankh”

➢ Choose `Save as...` from the `File` menu to save the complete Script Server log file as a text file.

You can then read this information into a word processor for further use. HELIOS Admin gets its information from the files “HELIOSDIR/var/adm/scriptsrv.log” (“Today”) to “HELIOSDIR/var/adm/scriptsrv.log.6” (“Seven Days Ago”).

7.2.1 Script Server log file structure

You might be interested in the Script Server log file, e.g. for troubleshooting purposes. It can easily be accessed from within HELIOS Admin.

Each entry in “HELIOSDIR/var/adm/scriptsrv.log” (with the appendices “.0”, *yesterday* to “.6”, *seven days ago*) has the following format:

```
date time scriptsrv[pid] or script: message
```

Note: HELIOS Admin does a live update of the displayed Script Server log files (`Lists > Script Server Log Files`).

7.3 Start, stop and refresh the Script Server process

The “scriptsrv” process will automatically be started using the “start-helios” and “stop-helios” commands. The status of the “scriptsrv” service can be changed via “srvutil”:

Note: The “srvutil” utility can be found in “HELIOSDIR/bin”.

Stop “scriptsrv”

```
# srvutil stop scriptsrv
```

Stops the “scriptsrv” services, new events will not be processed. All queued events will be processed up to 30 seconds after receiving the stop event. All executed scripts will receive a TERM signal to notify them of the “scriptsrv” service shutdown.

Start “scriptsrv”

```
# srvutil start scriptsrv
```

Starts the “scriptsrv” services.

7.4 Included sample scripts

Note: HELIOS ScriptAssistant (see 7.6 “ScriptAssistant”) simplifies the creation of standard hot folder setups. It generates scripts for the HELIOS Script Server hot folder automation system, to automate HELIOS server capabilities.

A variety of sample scripts can be accessed after mounting the “Settings” volume:

action-psd-cs2.pl

Call a Photoshop action (CS2)

printps.pl

Print PostScript files to a specified printer.

printtext.pl

Print text files to a specified printer.

sharpen-psd.pl

Applies the Photoshop sharpening filter to an image (CS3).

The following scripts require that PDF HandShake be installed on the same host:

HTML2PDF.pl

Convert a HTML page to a PDF document.

inspectPDF.pl

Analyze and preflight PDF files. Also performs PDF-to-PDF color conversion.

pdfflatten.pl

Flatten transparencies in a PDF document.

pdfresolve.pl

Replace OPI images and forms in PDF documents. The environment variable RESOLVEOPTION allows specifying additional command line parameters for "pdfresolve.pl".

PDF2BMP.pl

Convert a PDF document to a BMP image file.

printpdf.pl

Print a PDF file to a specified PostScript printer.

printpdfOptions.pl

A more advanced PDF printing script, which allows specifying common parameters.

splitPDF.pl

Split a PDF file into single pages and move final PDF files into a directory specified in the OUTDIR parameter.

convpdf2eps.pl

Convert a PDF page to an EPSF file and move final EPSF files into a directory specified in the OUTDIR parameter.

In the following scripts, image conversion of PDF files does require PDF HandShake:

convert2Lab.pl

Convert all supported ImageServer file types into TIFF CIELab and move converted files to OUTDIR.

webpicts.pl

Generate a 128x128 pixel PNG file and move final PNG file into a directory specified in OUTDIR. All colors (including spot colors) are converted to RGB.

webjpeg.pl

Generate a 128x128 pixel JPEG file and move final JPEG files into a directory specified in OUTDIR. All colors (including spot colors) are converted to RGB.

7.4.1 HELIOS default scripts location

We recommend to copy all scripts into the "Script Server" directory in the "Settings" volume. The benefits of the HELIOS default script location are:

- ❍ All scripts will automatically be migrated in case of a server change by moving the "var" folder to the new server
- ❍ The "Settings" volume simplifies the installation of new customized scripts by users, just copy the new script into the directory "Script Server"
- ❍ There is no need to specify the absolute path to the script. The script name specified in the configuration will automatically be searched in the "Settings" volume
- ❍ All scripts will automatically be changed to be executable (the executable flag will be set for this script) when the script is run

7.5 Script Server service port

The Script Server service port is 2024.

➢ Issue the command `socket localhost 2024`, type `help` for the command overview and `quit` to leave.

Note: By default, the Script Server service port can only be reached from *localhost*. See **RemoteAccess** in 9.1.7 "Script queue preferences".

---

showconf

Show validated and active configuration lines.

Example:

```
$ cd /usr/local/helios/
$ bin/socket localhost 2024
Welcome to the HELIOS Script Server service port
showconf
# Config of scriptsrv
Hold time: 10
Default script timeout: 120
# Queue name:Ftype:Suffix:Hotfolder:Script:User:Environment
PNG_convert:::/scriptserver/for_HTML:convert2Lab.pl::
OUTDIR=Done,SCRIPT_TIMEOUT=120,SCRIPTDEBUG=0
OK
```

Additional global information:

- `Hold time` (see **HoldTime** in 9.1.6 “Script Server preferences”)
- `Default script timeout` (see **RunTime** in 9.1.6 “Script Server preferences”)

showq

Show status of current jobs queued for processing in the `hold`, `termination` and `run` syntax:

```
h   File Scriptname User
t   File Scriptname User
r   File Scriptname User Status
```

Example:

```
showq
r "/demovol/Hotfolder%0/pdfresolve/PDF-native OPI.pdf" "pdfresolve"
  "hendrik" "Status: Running pdfresolve for file /demovol/Hotfolder%0/
  pdfresolve/PDF-native OPI.pdf"
h "/demovol/Hotfolder%0/pdfresolve/manual.pdf" "pdfresolve" "hendrik"
h "/demovol/Hotfolder%0/pdfresolve/data_sheet.pdf" "pdfresolve"
  "hendrik"
OK
```

Status field is filled by script writing `Status:...` to “stdout” or “stderr”, see also 7.7.2 “Status (shown on service port 2024)”.

showstat

Shows waiting, running, and finished jobs since the service has started. This can be used to debug/watch if the script is working correctly.

Example:

```
showstat
Total jobs done: 109
# Queue: Hold, Run, Done
HTML2PDF: 3, 0, 50
macpdfflatten: 5, 1, 28
pdfresolve: 0, 1, 14
OK
```

7.6 ScriptAssistant

HELIOS ScriptAssistant simplifies the creation of standard hot folder setups on Mac and Windows. It generates scripts for the HELIOS Script Server hot folder automation system, to automate server capabilities. Simply select the desired options, and ScriptAssistant will create a Perl script, ready for use by Script Server. No scripting or programming knowledge is needed.

ScriptAssistant helps create custom scripts for image conversion, PDF and PostScript printing, and PDF-native OPI image replacement, as well as automation of client applications via HELIOS Tool Server.

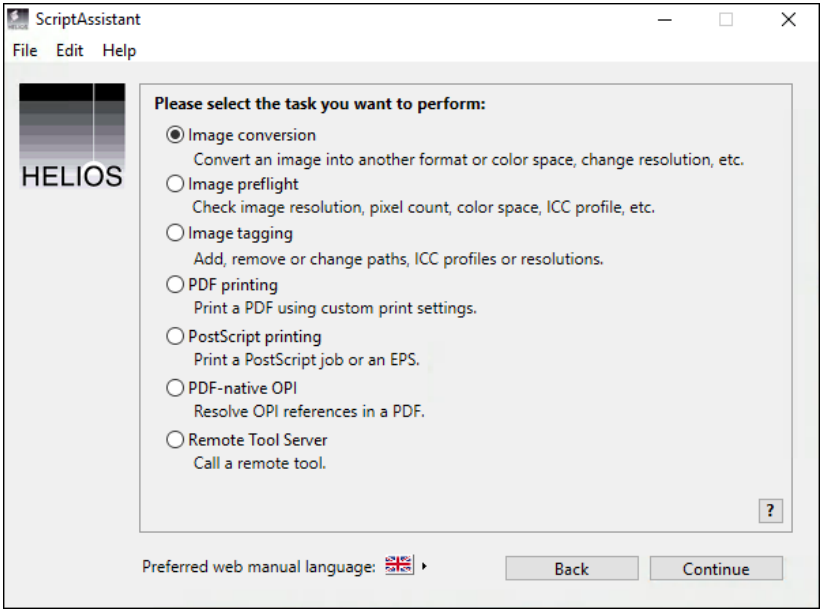


Fig. 7.10: The ScriptAssistant task selection dialog

7.6.1 Usage

General

When HELIOS ScriptAssistant is launched, the “Introduction” dialog window automatically opens. Here the preferred language for the manual pages can be specified. Menu items allow opening new ScriptAssistant windows, or opening an existing ScriptAssistant script. The “Help” menu opens a feedback form. On each dialog page, there is a [?] link to the relevant manual page on the HELIOS website. This help describes each option more fully. All HELIOS manuals are also included in PDF format in the “HELIOS Applications” volume.

Note: Each control provides a tooltip which references e.g. the command-line option behind it.

Saving scripts

Once a script configuration is completed, the “Save Script” dialog contains instructions on how to save the script, and copy it to the “Script Server” folder in the “Settings” volume on your HELIOS server. It is suggested that scripts be saved with descriptive file names. It may also be desirable to differentiate ScriptAssistant created scripts from other Script Server scripts, either by file naming convention, or by saving them into a subfolder.

Revising existing scripts using ScriptAssistant

Existing ScriptAssistant scripts can be reopened in ScriptAssistant and revised as needed. For example, on occasion it may be desirable to modify the settings for an existing hot folder. Or, if many hot folders are to be enabled, with only slight differences (e.g. different ICC profiles), a script could be created, then reopened and revised for each different hot folder, with each revised version saved under a new name.

Manually editing ScriptAssistant scripts

For those with Perl scripting know-how, it may on occasion be desirable to manually edit ScriptAssistant created scripts. For example, if many hot folders are to be enabled, with only slight differences (e.g. different ICC profiles), then an environment variable could be added so that users could use HELIOS Admin to easily specify the value for each hot folder (see 7.1.2 “Automatic Script Server configuration”). Thus only one master script would be needed instead of multiple scripts. Or, it may be necessary to manually edit a script in order to add options not available in ScriptAssistant. Note that any manual changes will be lost if a manually revised script is reopened in ScriptAssistant and then saved. A warning dialog alerts the user upon opening such files, and the “Save Script” window provides an additional warning (see Fig. 7.11).

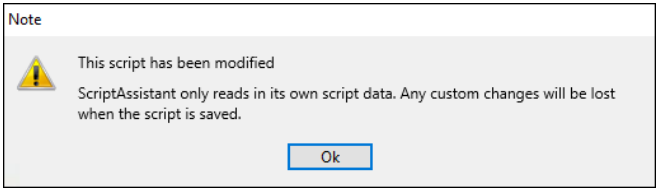


Fig. 7.11: Modified script alert

ScriptAssistant as a learning tool

ScriptAssistant generated scripts, along with the included sample scripts, can be used to learn the proper syntax of the various HELIOS commands, and to study Perl scripting itself. The actual command line output for each script can be viewed in the Script Server log file if the `Debugging` checkbox is selected in the Script Server script setup dialog.

7.6.2 The "Tasks"

Image conversion

Generates scripts utilizing the ImageServer "layout" command. The "layout" program is a general purpose image conversion engine, with the highest quality output. With over 100 image conversion options, it is very powerful, but can also be complex. ScriptAssistant simplifies the procedure by allowing the easy selection of conversion options such as image file format, color space, compression, resolution, dimensions, color and channel options, bit-depth, ICC settings, metadata options, and more. As you proceed through the dialogs, ScriptAssistant shows only options that are compatible with the options already selected.

Image preflight

Generates scripts utilizing the ImageServer "oiimginfo" command. It allows checking image resolution, pixel count, color space, ICC profile, etc.

Image tagging

Generates scripts utilizing the HELIOS "layout" command. It allows adding or removing ICC profiles, removing paths and IPTC or XMP metadata, setting or removing a clipping path, and changing image resolution or dimensions.

PDF printing

Generates scripts utilizing the PDF HandShake "pdfprint" command. The "pdfprint" program prints a PDF file to PostScript, with options to specify the printer queue, page setup, color and font options, and ICC settings.

PostScript printing

Generates scripts utilizing the HELIOS "lpr" command. The "lpr" program prints a PostScript or EPS file to a print queue.

PDF-native OPI

Generates scripts utilizing the PDF HandShake "pdfresolve" command. The "pdfresolve" program performs PDF-native OPI image replacement and content repurposing (e.g. ICC color matching) in PDF files, with extensive options, for color and ICC settings, image bit-depth, compression, PDF page box, and more.

Remote Tool Server

Generates scripts utilizing the HELIOS "toolclient" command. The "toolclient" program finds, connects to, and sends/receives files to/from a tool server. Tool Server makes automating remote applications/tools very easy. A click on the ScriptAssistant ? button allows you to read the Tool Server manual for instructions on how to install the HELIOS Tool Server on each Mac or Windows or UNIX client to work as a tool server, and how to enable the desired tool server script(s) by copying them into the "Tool Server" folder.

7.6.3 Requirements

- ScriptAssistant: macOS 10.11 or newer, Windows 7 (SP1) or newer
- HELIOS G8 server installation (based on HELIOS CD G8) or newer
- ImageServer for image conversion
- PDF HandShake for PDF conversion and printing
- Universal File Server for PostScript printing
- ImageServer and PDF HandShake for PDF-native OPI
- ImageServer for Tool Server (PDF HandShake also required for some tools)

7.6.4 Installation

➢ Mount the “HELIOS Applications” volume, open the “Windows” folder (Mac: “MacOS”), then “ImageServer Tools” and “HELIOS ScriptAssistant”.

Mac only:

➢ Mount “HELIOS ScriptAssistant.dmg” and copy the application to your local harddisk.

Windows only:

➢ Extract the “HELIOS ScriptAssistant.zip” archive to your local harddisk. “HELIOS ScriptAssistant.exe” needs the “HELIOS ScriptAssistant Libs” directory. “HELIOS ScriptAssistant.exe” must not be moved alone.

7.6.5 Uninstallation

To uninstall HELIOS ScriptAssistant delete the “HELIOS ScriptAssistant.app” (Mac) or the “HELIOS ScriptAssistant” directory (Windows), respectively.

7.7 Developing and debugging scripts

Script Server is started automatically by the Service Controller. Please make sure via “srvutil status” that the “scriptsrv” service is running and there are no error messages in the system messages file.

All valid registrations for different scripts are displayed via port 2024.

Every time a custom script is called, the “scriptsrv.log” file contains a information similar to the following:

```
29.07.2005 18:00:07 scriptsrv [1747]: Create process
myscript.pl[1854] job added by tim
...
29.07.2005 18:00:17 scriptsrv [1747]: Process
myscript.pl[1854] done
```

All standard output, as well as error output of the script is automatically redirected into the “scriptsrv.log” file. Each line is prefixed by time stamp and script name.

7.7.1 Script parameters

Scripts are called with the file name as the first argument, e.g. a host shell script will contain the file name in `$1`, a Perl script will contain it in `$ARGV[0]`.

These parameters are provided via environment variables:

SCRIPTDEBUG

Verbose output is written.

SCRIPT_EVENT

Is set by the server. Available events are `close`, `rename`, `dircreate`, `dirclose`, `delete`, `dirdelete`, `dirrename`, and `exchange`.

SCRIPT_EVENTUSER

Name of the user that triggered the event.

IMAGE_SERVER_LICENSE

Map script to serial number of the ImageServer installation. This environment variable can be set, e.g. to prevent that a script is distributed.

SCRIPT_HOTFOLDER

Absolute path of the hot folder.

SCRIPT_NAME

Name of the script.

SCRIPT_TIMEOUT

If set, it overrides the global preference **RunTime** (see 9.1.6 “Script Server preferences”).

SCRIPT_OLDNAME

Is set by the server and can be evaluated in the script containing the old name, if the event is `move` or `dirmove`.

Additional custom parameters can be specified in the `Environment` tab (Fig. 7.5). The custom environment will be available as environment variables in the script. This allows developing one script and using it for different hot folders by using different parameters (e.g. printer, ICC profiles, color spaces).

7.7.2 Status (shown on service port 2024)

Script output starting with `Status:` is not written to "scriptsrv.log" but displayed in the **showq** command output on service port 2024. Intention: if a job takes long time to finish, the script can tell the user (who is monitoring the process on the service port) what is actually done by the script.

7.7.3 Debugging scripts stand-alone (without Script Server)

It is a good idea to test your scripts first manually in a host terminal session to verify that they work. The Script Server calls all scripts with the current directory set to the HELIOS product directory. Therefore it is required that you first change into the HELIOSDIR before executing a script manually.

Manual call of a stand-alone script, ("printps.pl"), with a bash or Bourne shell:

```
# cd /usr/local/helios
# chmod +x var/settings/Script Server/printps.pl
# export HELIOSDIR="/usr/local/helios"
# export PRINTER="lw"
# export SCRIPTDEBUG=1
# export PRINT_TIMEOUT=60
# export SCRIPT_EVENT=close
# var/settings/Script Server/printps.pl "/data/demovol/TestFolder/test.ps"
```

If this produces error messages or the script does not do what it is expected to do, it needs to be solved before using it in the automatically event-driven Script Server.

Calling standalone scripts allows debugging, using the Perl debugger or a host debugger for C/C++ applications. Simple debugging by printing messages will work as well within the automated Script Server environment. All script output will later automatically be redirected into the "scriptsrv.log" file.

Note: Every HELIOS Perl sample contains the following file magic:

```
#!var/run/runperl -w
```

This means the script is executable just by calling it. It always needs to be called from the HELIOS directory (e.g. "/usr/local/helios"), otherwise "var/run/runperl" cannot be found. The file "runperl" will be a symbolic link to your Perl runtime, e.g. "/usr/bin/perl" on OS X or Linux systems.

Whenever "start-helios" is issued, the "runperl" link will automatically be created. This might be helpful if Perl is installed at a later time (no changes in scripts required).

An alternative option to test your scripts in a different directory is to create the runperl link in your script directory.

Example (e.g. on OS X or Linux):

```
# cd /home/myhome
# mkdir -p var/run
# ln -s /usr/bin/perl var/run/runperl
```

7.7.4 Debugging scripts in the Script Server environment

To inspect "holdQueue" and "runQueue", and the configuration of the running "scriptsrv" daemon, connect to service port 2024, and use "showq" for inspecting the queue and "showconf" for inspecting the configuration.

If a job matches more than one configuration, only the first matching configuration in alphabetical order is considered.

To switch on the script debug variable `SCRIPTDEBUG` it is required to set the value to `1`. You may also set the global debug preference `scriptdebug`:

```
# prefvalue -k Programs/scriptsrv/scriptdebug -t int 1
```

Delete the `scriptdebug` preference via:

```
# prefvalue -k Programs/scriptsrv/scriptdebug -d
```

Find a description of the “prefvalue” program in the HELIOS Base manual.

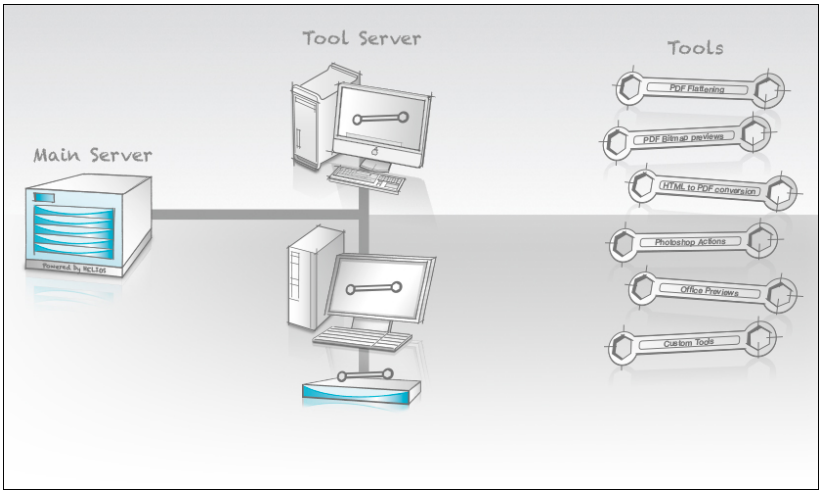
Important: The global preference `scriptdebug` activates debugging for all scripts. Thus, we do not recommend to turn on the debug preference for production servers because it may produce heavy load on the server and can cause significant slow-down of the server!

7.8 Additional considerations

You should restrict the registered types/extensions and directories to those that are really processed by your scripts, so that the communication and processing overhead does not become too large.

When using scripts for layout generation (with “layout” in normal or convert mode) you should deactivate automatic layout generation for the hot folder or the complete volume in which the hot folder resides.

8 HELIOS Tool Server



HELIOS Tool Server allows integrating remote applications to be used by a main server. Certain applications and technologies are only available on a special platform. Instead of reinventing these required applications and technologies, Tool Server makes them remotely available on any major server platform. Tool Server provides the platform and already includes several useful tools – additional tools can be added by the user. HELIOS provided tools and samples are installed in minutes.

The main goal is to make remote applications easy to use, therefore HELIOS Tool Server comes with a dedicated license-free installation which installs on any major platform, i.e. Mac/Windows. The “toolclient” application detects

all servers automatically by using mDNS (“Bonjour”) and choosing the server that offers the required service and provides best computing performance. Automatic job load balancing, file transfers, processing, auditing, and error handling makes it quite easy.

See the HELIOS Tool Server online user manual at:
www.helios.de/web/EN/support/manuals.html

9 Preferences and notification features

9.1 ImageServer preferences

This chapter lists all the preferences that are pertinent to ImageServer. Find a description of how to set, view, change or delete preferences, with the HELIOS utility programs “prefdump”, “prefvalue”, and “prefrestore”, in the HELIOS Base manual.

Important: Make sure that preference keys *DO NOT* start or end with a slash (“/”) character, and note that they are case-sensitive! Also, if any preference key or preference value includes spaces, that key or value must be enclosed in quotes.

9.1.1 OPI server preferences

```
Key:  Programs/opisrv/<preference>
```

HoldTime `int` `30`

(HELIOS Admin option `Layout Delay`)

This preference determines the time delay in seconds after which the layout file is generated from the original image file.

MaxProc `uint` `<see description>`

This preference controls the maximum number of “opisrv” processes that may run simultaneously. Default is the number of CPU cores.

NiceIncr `int` `10`

The priority of the “opisrv” processes – compared to other executable programs on the HELIOS server – can be changed. This preference lets you increase/reduce the priority, and therefore the speed of a program. The higher the priority of a program, the less “nice” its behavior towards other applications. The values for NiceIncr are as follows:

`NiceIncr` > 0 Slower / “nice” towards other applications
`NiceIncr` < 0 Faster / less “nice” towards other applications

Facility `str` `"daemon" or "user"`

Specifies the facility which is used for system log messages. The string names the log facility. The default value depends on the program and is `daemon` for the “opisrv” program and `user` for the “layout” program. Valid names are listed below (please note that some names are not supported on every architecture):

“audit”, “sat”, “auth”, “security”, “cron”, “daemon”, “kern”, “lpr”, “mail”, “news”, “syslog”, “user”, “uucp”, “local0”, “local1”, “local2”, “local3”, “local4”, “local5”, “local6”, and “local7”.

RegisterTypes `strlist` `""`

Allows defining additional file types (Mac), which are registered for file events.

RegisterSuffixes `strlist` `""`

Allows definition of additional file suffixes (Windows PC), which are registered for file events.

RemoteAccess `bool` `FALSE`

If set to `TRUE`, access to the ImageServer Event port (2002) is enabled for remote users (subject to the restrictions in the "ipaccess" list). The default is that access to the ImageServer Event port is only enabled for users on the same host.

PreserveResources `strlist` `"IPTC,PATH,XMP"`

Determines additional resources, which will be read and preserved by "OpenImage" plug-ins. If the string contains `PATH`, `IPTC` or `XMP`, the named additional resource will be read. If no resources should be preserved, specify `none`.

RenderColor `str` `"CMYK"`

Sets the color space for rendered image objects. It is important that for PDF files this option is set to CMYK, to show proper overprints.

KeepPhysicalSize `bool` `FALSE`

If set to `TRUE`, this preference preserves the physical size of an image when the image size is changed via the resolution.

LayoutAlpha `bool` `FALSE`

The "layout" option `-o LayoutAlpha` can also be set as a server preference either for the "opisrv" service or globally, with the key: "Global/Opi/LayoutAlpha".

9.1.2 Global OPI preferences

Key: Global/Opi/<preference>

Fast bool FALSE

(HELIOS Admin pull-down menu Layout Quality)

This preference determines the quality of the generated image. If set to TRUE, the used image generation algorithm will be slightly less precise but faster, while consuming less resources. However, the rendering quality should be sufficient for smaller images.

PrintColor str "CMYK"

(HELIOS Admin pull-down menu Layout Color)

The print color space for all layout files that are generated on the server can be specified by use of this preference. At the same time, it defines an ICC profile which is used for color matching the layouts. This profile is taken from the **ProfilePaths** setting.

ScreenResolution double 72.000000

(HELIOS Admin option Default Monitor Resolution)

Defines the default monitor resolution for layout files. Usually monitor resolutions have a range between 72 and 200 dpi.

PrintResolution double 72.000000

(HELIOS Admin option Default Print Resolution)

Defines the default resolution for printouts of layout files.

ProfileRepository str "ICC-Profiles"

This preference specifies the volume name containing the ICC profiles. You may also specify the absolute path to the repository or the path relative to the “HELIOSDIR” directory.

ImageSearchPaths `strlist` `""`

Specifies a list of path names to search for high-resolution images during image replacement, in case they cannot be found in the standard locations. The setting reflects the entries in HELIOS Admin (`ImageServer Settings > Image Search Paths...`)

ProfilePaths `strlist` `(see description)`

Defines RGB and CMYK profiles used if the color space for the destination layout file and the color space of the high-resolution file are different. These profiles are used to match between color spaces properly. Each string has the format `<color space>=<path name>`.

The default profiles are "RGB=sRGB_IEC61966-2-1_noBPC.icc" for RGB, and "CMYK=ISOcoated_v2_eci.icc" for CMYK.

IgnoreSpots `bool` `FALSE`

(HELIOS Admin option `Ignore Additional Channels for Bitmap Images`)

This preference default causes that additional channels (e.g. spot colors) are used when creating a layout image, printing the resolved high-resolution image or converting an image. See 3.3.4 "Additional channels in bitmap images" and 5.2 "Define folder specific OPI settings" for a description.

EpsPreviewPSLevel `int` `1`

Determines the preview PostScript level for EPSF files. By default, this is PostScript level 1, optionally PostScript level 2 can be specified.

EventPort `int` `2002`

Controls the TCP/IP port number where to listen for OPI event registration requests. If this preference is not set, "opisrv" uses the port number given by the TCP service "OPIEvent". If this service is not defined it uses the default port number 2002.

AutomaticLayout bool TRUE

If set to TRUE, this preference causes that a layout image is created from a PDF file. The setting of this attribute reflects that of the PDF OPI Layout Generation checkbox in the HELIOS Admin Settings > PDF HandShake Settings menu.

For this preference, the key must be extended to:
Global/Opi/Formats/PDF ///AutomaticLayout

SpotToEps bool TRUE

Causes the creation of EPSF layout images if a high-resolution raster image contains extra colors like spot colors.

MaskToEps bool TRUE

Causes the creation of EPSF layout images if a high-resolution raster image contains a clipping path.

AntiAlias bool TRUE

If set to TRUE, the screen preview in layout files, generated from PDF documents, offers an enhanced readability. This is recommended for monitor screen resolutions up to 96 dpi. The setting of this attribute reflects that of the Antialias Screen Preview checkbox in the HELIOS Admin Settings > PDF HandShake Settings menu.

For this preference, the key is:
Library/OpenImage/ManagerPrivate/HeliosPDF/AntiAlias

UsePredictors bool TRUE

The HELIOS TIFF plug-in uses so called “predictors” for the “Flate” and “Compress” compression methods, which results in smaller file sizes after compression. If this preference is set to FALSE, no “predictors” are used.

For this preference, the key is:
Library/OpenImage/ManagerPrivate/HeliosTIFF/UsePredictors

UseEuroCMYK `bool` `TRUE`

(HELIOS Admin pull-down menu `CMYK Default Profile`)

This preference defines the type of CMYK color space, if an image which is transformed into CMYK or vice versa *and if* – in a situation like that – a CMYK output (printer) or input (image) profile has not been explicitly defined.

For this preference, the key is:

`Library/OpenImage/UseEuroCMYK`

UseProfile `bool` `TRUE`

Process color matching for layout files, if the high-resolution files have tagged ICC profiles.

IncludeProfile `bool` `FALSE`

If set to `TRUE`, a generated layout will contain an *embedded* profile, even if the original image only has a *referenced* profile. This preference is only applicable if the `UseProfile` option is also set. It is useful, if the layout will be used either on a different computer or in an environment where the specified profile is not available.

OmitProfile `bool` `FALSE`

If set to `TRUE`, a generated layout/original will omit any specified profile. This preference is useful if the generated layout/original has a smaller size than the specified profile.

CompressPrint (TIFF) `str` `""`

(HELIOS Admin option `TIFF Compress`)

If the value “Compress” is specified for this preference, TIFF layout images are compressed.

For this preference, the key must be extended to:

`Global/Opi/ImageCompress/TIFF///CompressPrint`

CompressPrint (EPSF) `str` `""`

(HELIOS Admin option `EPSF JPEG Printable Preview`)

If the value “JPEG” is specified for this preference, EPSF layout images are compressed.

For this preference, the key must be extended to:

`Global/Opi/ImageCompress/EPSF///CompressPrint`

ScreenColor `str` `""`

(HELIOS Admin option `EPSF 8-bit Screen Preview`)

If this preference is set, EPSF layout images for monitor use contain indexed colors (max. 256 colors).

For this preference, the key must be extended to:

`Global/Opi/ImageColor/EPSF//ScreenColor`

PDFNativeWorkflow `bool` `FALSE`

This preference enforces the generation of PDF-native workflow compatible layout images from vector-based EPS images, even if TIFF layout images are supposed to be generated (e.g. via a “%t” folder syntax).

MaxLayoutSize `uint32` `2540`

Defines the maximum allowed rows and columns in a layout image.

9.1.3 OPI-related volume preferences

`Key: Volumes/<volume_path>/<preference>`

Important: If the key contains a path, the “/” characters *within the path* must be “marked” by preceding each with a “\” character.

Example:

```
# prefvalue -k '/Volumes/\/images/SpotToEps' -t bool FALSE
```

AutomaticLayout `bool` `TRUE`

(HELIOS Admin option `Create Layouts`)

Activates the automatic generation of layout files.

CrossPlatformLayout `bool` `FALSE`

(HELIOS Admin option `PC Layouts`)

Causes that "opisrv" generates PC-compatible layouts.

SpotToEps `bool` `TRUE`

Causes the creation of EPSF layout images if a high-resolution raster image contains extra colors like spot colors.

MaskToEps `bool` `TRUE`

Causes the creation of EPSF layout images if a high-resolution raster image contains a clipping path.

9.1.4 Global printer preferences

`Key: Global/Printers/<preference>`

IgnoreMasks `bool` `FALSE`

(HELIOS Admin option `Ignore Mask for Bitmap Images During Output`)

If set to `TRUE`, this preference causes that an existing clipping path is ignored when printing an image. See **Image paths and clipping paths** and 5.2 "Define folder specific OPI settings" for a description.

UseAlpha `bool` `TRUE`

If set to `FALSE`, ImageServer ignores alpha channels in image files.

9.1.5 OPI-related printer queue preferences

Key: Printers/<printer queue>/<preference>

OpiEnabled bool FALSE

(HELIOS Admin option OPI Active)

If this preference is set to TRUE it activates OPI for the specified printer queue.

CustomColorTinting bool TRUE

This preference applies to composite printing only: Spot colors are output in Level 2 Separation color space. This becomes necessary when printing to a Create PDF printer queue or to in-RIP separation devices, in order to preserve the spot colors. If set to FALSE, all spot colors that are incorporated in the image are converted to the process colors of the output color space, e.g. CMYK.

PrintDraft bool FALSE

(HELIOS Admin option Print Layout Images)

If set to TRUE, the printouts from the specific printer queue will only contain layout images. The copies of the images – which you have used for layouting your document – will be sent to the OPI server and then be replaced by the layout files instead of the high-resolution originals.

CheckImages bool TRUE

(HELIOS Admin option Check Images)

With this preference set to TRUE, the OPI server will try to find all images that are included in a print job. If any images are missing, the job will be stopped, and issue an error message for every missing file. If CheckImages is set to FALSE, single images may be missing on the printouts.

CheckICCProfiles `bool` `TRUE`

(HELIOS Admin option `Check ICC Profiles for Pictures`)

This preference will induce the OPI server to check whether all image profiles that have been tagged to any of the image files are available (the `Check Images` option must be activated, otherwise this preference has no effect!). The server will automatically stop the print job if a single profile is missing and issue a corresponding error message. If you set `CheckICCProfiles` to `FALSE`, OPI will execute all print jobs and use standard output profiles for color matching whenever the correct profile is missing.

PreserveDeviceN `bool` `FALSE`

(HELIOS Admin option `PostScript 3/DeviceN Output`)

This preference optimizes the PostScript output of PDF original files to use PostScript 3/DeviceN features. It should only be set to `TRUE` if you print to a PostScript 3 device with in-RIP separation. This applies for Hexachrome printing, colorized images (Duotone), etc. When printing host-based separations with applications like QuarkXPress, this feature must be turned off, otherwise the output will lead to unexpected results.

PureWhite `bool` `FALSE`

(HELIOS Admin option `Preserve Raster Colors - White`)

Exclude white colored *raster* objects in PDF files from color transformation, so that they remain *white* even if color matching is done. CMYK values will be zero, i.e. no ink is applied to the plates.

PureGrays `bool` `FALSE`

(HELIOS Admin option `Preserve Raster Colors - Gray`)

Exclude gray colored *raster* objects in PDF files from color transformation, so that they remain *gray* even if color matching is done. Gray raster objects in Gray/RGB/CIELab/Indexed color spaces are detected and converted to *Gray* only for CMYK output.

PureBlack `bool` `FALSE`

(HELIOS Admin option `Preserve Raster Colors - Black`)

Exclude black colored *raster* objects in PDF files from color transformation, so that they remain *black* even if color matching is done. Black raster objects in Gray/RGB/CIELab/Indexed color spaces are detected and converted to *Black* only for CMYK output.

PureCMY `bool` `FALSE`

(HELIOS Admin option `Preserve Raster Colors - CMY`)

Applies to CMYK to CMYK conversions of pixels of raster images, and preserves the input color if, and only if, black is 0% and exactly one CMY primary color is 100% and all other primary colors are 0%.

CheckFonts `bool` `TRUE`

(HELIOS Admin option `Check Fonts`)

This preference will induce the OPI server to stop the print job and issue an error message if the current print job contains any fonts that are not available.

ResolveAll `bool` `FALSE`

(HELIOS Admin option `Replace Images`)

If set to `FALSE`, the OPI server will only replace *layout* images during printing. High-resolution originals you may have placed in your document, will not be replaced. In case you want the OPI server to replace all images during printing (e.g. if you want to use the `Omit...` options in the print dialog) you should set this preference to `TRUE`.

DownSampling `bool` `FALSE`

(HELIOS Admin pull-down menu `Downsampling`)

Defines a uniform output resolution for all raster-based images – no matter whether you print layout images or high-resolution

originals. The resolution for downsampling may be specified with the **Resolution** preference (see below).

Resolution `double` `0.0`

(HELIOS Admin pull-down menu `Downsampling`)

With **DownSampling** (see above) set to `TRUE`, the resolution of an image can be specified with this preference. If the resolution of a given image (e.g. 300 dpi) is higher than the value specified for this preference (e.g. `222,30` dpi), the OPI server will apply the value you have chosen for downsampling and then re-calculate and scale the image accordingly. Images with a resolution smaller than the value entered in this dialog, will remain unchanged because increasing the resolution automatically decreases the dimensions of an image. This would entail upward scaling and lead to a loss in quality. Zero induces the software to use the default resolution of the printing device as specified in the queue's PPD file.

FixedSampling `bool` `FALSE`

This preference allows fixing the downsampling resolution (see `Resolution` preference above) to the given value. Usually, if this preference is set to `FALSE`, the output resolution is adjusted to the image resolution if the image resolution is smaller. When setting this preference to `TRUE` the printer interfaces will use the given **Resolution** as is for downsampling. This may cause upsampling of images with resolutions smaller than the given one. This preference is only meaningful when setting the `DownSampling` preference to `TRUE`.

CompressPostScript `str` `""`

(HELIOS Admin section "PostScript Output Compression")

This option specifies the compression mode for images while being printed. Two preferences are needed to set this preference: color space and compression mode. JPEG 2000 makes sense only in a "pdfresolve" workflow.

Modes are:
`None`, `CCITTG4`, `JPEG`, `JPEG 2000`, `Flate`, and `Compress`

Color spaces are:
`RGB`, `CMYK`, `CIELab`, `Grayscale`, and `Bilevel`

For this preference, the key must be extended to:
`Printers/<printer queue>/PostScriptImage/<colorspace>/CompressPostScript`

PrintCompressQuality `double` `(see description)`

HELIOS Admin section "PostScript Output Compression")

With the **CompressPostScript** preference selected, this preference controls the output quality of JPEG compressed PostScript images (valid values `1...100`; default value `75`) or JPEG 2000-compressed images (valid values `0...100` default value `0` for lossless).

Color spaces are:
`RGB`, `CMYK`, and `Grayscale`

For this preference, the key must be extended to:
`Printers/<printer queue>/PostScriptImage/<colorspace>/PrintCompressQuality`

KeepInfo `bool` `TRUE`

Specifies whether in case of resolving an already resolved PostScript job, the use of fonts and other resources should still be accounted.

ImageIDSearch `bool` `TRUE`

This preference – if it is set to `TRUE` – enables the use of the HELIOS desktop database to locate moved image files.

ImageSearchVolumes `strlist` `""`

Specifies a list of volume names to search for high-resolution images during image replacement, in case they cannot be found in the standard locations.

Note: When remote OPI is used, the remote client must not use “layouts” as directory name for low-res images, in case the “ImageSearchVolumes” feature is used.

DefaultPrinterProfile `str` `""`

(HELIOS Admin pull-down menu `Default Printer Profile`)

This preference specifies the path name of the default ICC profile describing the printing device.

DefaultProofProfile `str` `""`

(HELIOS Admin pull-down menu `Default Proof Profile`)

This preference specifies the path name of the default ICC profile describing the proof device. The string will only be recognized if a default printer profile is set.

DefaultDevLinkProfile `str` `""`

(HELIOS Admin pull-down menu `Default DeviceLink Profile`)

This preference specifies the path name of the default DeviceLink profile. The string will only be recognized if a default printer profile is set.

ProcessColorspace `str` `"CMYK"`

This preference sets the color space used while printing separations. The default is `"CMYK"` for a four color print process. Spot colors are not affected by this preference. Valid color spaces are `"CMYK"` and `"Multi"`. If you set the string to `"Multi"` you have to define the names and order of the color components. Furthermore, the usage of the `"Multi"` color space requires the use of ICC profiles. For (future) ICC profiles that include the names of the color components this preference will become obsolete. Note that the value you choose for this preference can be overridden by the printer profile you have selected using the `DefaultPrinterProfile` preference.

ProcessInks `strlist` `"Cyan","Magenta","Yellow","Black"`

This preference controls, in conjunction with the `ProcessColorspace` preference, the names of the process inks used for separations. Spot colors are not affected.

CompositeColorspace `str` `"CMYK"`

This preference sets the color space used while printing composite. The default is `CMYK`. Setting this preference to `None` causes all color images in a print job to be kept and printed in their original color space. This applies to CMYK, RGB and CIELab images only. Valid strings are `"CMYK"`, `"RGB"`, and `"CIELab"`. Please note that the value you choose for this preference can be overridden by the printer profile you have selected using the `DefaultPrinterProfile` preference.

Note: In a “Create PDF” queue, the default behavior is “None”.

TemporaryTagging `bool` `TRUE`

This preference enables the use of the ICC info files while printing images. If it is set to `TRUE`, every non-tagged image will be temporarily tagged during printing provided that the ICC info file specifies tagging for that kind of image.

IgnoreMissingLowRes `bool` `TRUE`

Use this preference to specify whether non-existing layout images should be ignored when printing layout quality.

IgnoreUntagged `bool` `TRUE`

This preference controls whether untagged images should be ignored when printing to an ICC color matched queue. If set to `TRUE`, the images are printed with the default profile. If set to `FALSE`, the print job is passed to the error queue.

PrintSpotToProcess `bool` `FALSE`

If this preference is set to `TRUE`, all spot colors in the print job will be printed with the appropriate process color values.

9.1.6 Script Server preferences

The following keys require a restart of the service to take effect:

`Key: Programs/scriptsrv/<preference>`

TelnetPort `int` `2024`

Specifies the Script Server service port.

HoldTime `int` `10`

Specifies the time interval (in seconds) after the file event is received before the script is started. The setting reflects that of the `Script Delay` field in the HELIOS Admin "Script Server Settings" window.

RunTime `int` `120`

This preference determines how long a script may run. Specifying `0` disables the running time (i.e. "run forever"). The setting reflects that of the `Script Timeout` field in the HELIOS Admin "Script Server Settings" window. Compare **Timeout** in 9.1.7 "Script queue preferences".

TermTime `int` `30`

This preference determines how long "scriptsrv" should wait before the process is killed. This happens if **RunTime** has timed out.

MaxProc `int` `<see description>`

This preference controls the maximum number of "scriptsrv" processes that may run simultaneously. Default is the number of CPU cores.

Note: It is recommended that the value for MaxProc be less than or equal to the number of CPU cores.

scriptdebug int32 0

Specifies whether verbose output is written. The value is exported as the SCRIPTDEBUG environment variable.

RemoteAccess bool FALSE

If set to TRUE, access to the Script Server service port is enabled to remote users (subject to the restrictions specified in the IP access list).

ipaccess str ipaccess

Specifies the file name of the IP access list that controls client access to the Script Server.

9.1.7 Script queue preferences

Key: Programs/scriptsrv/Config/<queue>/<preference>

Path str ""

Specifies the path to the Script Server hot folder. The setting reflects that of the Hot Folder field in the General tab of the HELIOS Admin "Script" configuration window.

Script str ""

Specifies the desired script for the script queue. The setting reflects that of the Script field in the General tab of the HELIOS Admin "Script" configuration window.

User `str` `(see description)`

Specifies the user whose permissions are used to execute the script. The setting reflects that of the `User` field in the `General` tab of the HELIOS Admin `Script` configuration window. If this preference is not specified, the script is executed as *superuser*.

Enabled `bool` `FALSE`

Determines whether the script queue is active at all. The setting reflects that of the `Enable` checkbox in the `General` tab of the HELIOS Admin "Script" configuration window.

Timeout `int` `(see description)`

This preference determines how long the script may run. Specifying `0` disables the running time (i.e. "run forever"). The setting reflects that of the `Timeout` field in the General tab of the HELIOS Admin `Script` configuration window. If this preference is not specified, Script Server uses the global value **RunTime** (see above).

Recursive `bool` `FALSE`

Determines that all subdirectories underneath the specified directory for the hot folder (see Path) are considered. The setting reflects that of the `Include Subdirectories` checkbox in the `General` tab of the HELIOS Admin "Script" configuration window.

Directory `bool` `FALSE`

Determines that directory events rather than file events are notified to the Script Server. The setting reflects that of the `Folder Changes` checkbox in the `File Types` tab of the HELIOS Admin "Script" configuration window.

RemoveEvents `bool` `FALSE`

If set to `TRUE`, this preference allows getting notified via a ScriptServer script when files are being deleted. The same works for deleted folders when the **Directory** preference is set.

scriptdebug `int32` `(see description)`

Specifies whether verbose output is written. The value is exported as the `SCRIPTDEBUG` environment variable. The setting reflects that of the `Debugging` checkbox in the `General` tab of the HELIOS Admin "Script" configuration window. If this preference is not specified, Script Server uses the global value **scriptdebug**.

Ftypes `strlist` `""`

Allows specifying file types for which script execution should be triggered. If no type is specified, the script is executed for any file event. The setting reflects that of the entries in the `Types` list in the `File Types` tab of the HELIOS Admin `Script` configuration window.

Suffixes `strlist` `""`

Allows specifying file suffixes for which script execution should be triggered. If no suffix is specified, the script is executed for any file event. The setting reflects that of the entries in the `Suffixes` list in the `File Types` tab of the HELIOS Admin "Script" configuration window.

Environment `strlist` `""`

Allows specifying additional environment variables, which are considered while the script runs. The setting reflects that of the entries in the `Environment` list in the `Environment` tab of the HELIOS Admin "Script" configuration window.

9.1.8 Spot color editor preferences

```
Key:  Library/OpenImage/CustomColors
```

CustomColors* `str` `<color value>`

(HELIOS Admin `Spot Colors`)

Defines a preview color for a given spot color name either in the *Lab* or in the *CMYK* color space. If the color is defined in *Lab* (1), the string is preceded by an "L". If the value is entered in *CMYK* (2), the string has a leading "C".

For *C*, *M*, *Y*, *K*, and *L* color values of *0%* are represented as `0.0`, while values of *100%* are represented as `1.0`.
For *a* and *b*, values of *-128* are represented as `0.0`, while color values of *127* are represented as `1.0`.

* *The actual key is* `Library/OpenImage/CustomColors/color name`.

Examples:

```
(1) prefvalue -k "Library/OpenImage/CustomColors/Bleen"
        -t str "L0.450000000 0.757812480 0.527343744"
(2) prefvalue -k "Library/OpenImage/CustomColors/Grellow"
        -t str "C0.0214 0.8991 0.1767 0.0091"
```

9.2 Notification features

File change events will be reported to custom applications by the ImageServer TCP/IP connection via port `2002`.

The output is preceded by a number. and followed by the event type (see 9.2.1 "Event types" below) and the file path:

0 Simple OK
1 Informational message
2 Command syntax error
3 UNIX error
4 OPI event

➢ Issue the command `socket localhost 2002`, type `help` for the command overview and `quit` to leave.

Note: By default, the ImageServer event port can only be reached from localhost. See **RemoteAccess** in 9.1.1 “OPI server preferences”.

9.2.1 Event types

ushowq/showq

Show the “opisrv” event queue as a list of files which are waiting to be processed into layout images. `ushowq` additionally includes the *user ID* of the job initiator.

showvols

Show list of exported volumes by EtherShare and PCShare. Can be used to map the path name to the corresponding volume. The following flags are possible:

`E` EtherShare
`P` PCShare
`U` UTF-8 encoded
`A` Automatic layout generation

touchfile

Issue a change event for a given file, e.g. to generate a layout file.

touchdir

Issue a change event for a directory, e.g. to generate layouts for all files in the directory.

touchtree

Same as `touchdir` above, but used *recursively*.

Note: For the next three event features a file type can optionally be specified for the current event.

---

sendclose

Send a file close event to the “opisrv”:

```
sendclose "file name" [file type]
```

sendrename

Send a file rename event to the “opisrv”:

```
sendrename "file name old" "file name new" [file type]
```

senddelete

Send a file delete event to the “opisrv”:

```
senddelete "file name" [file type]
```

sendlayout

Send a layout event to the `registerlayout` clients. This is usually done by the programs “layout” and “opisrv”:

```
sendlayout "file name"
```

sendrenamedir

Send a directory rename event to the “opisrv”:

```
sendrenamedir "directory name old" "directory name new"
```

sendcreatedir

Send a directory create event to the “opisrv”:

```
sendcreatedir "directory name"
```

senddeletedir

Send a directory delete event to the “opisrv”:

```
senddeletedir "directory name"
```

registerlayout

Register for layout events. Whenever a new layout file is generated, the path name of the new file will be reported.

registerfiles

Register for file change events. This allows receiving all events for newly created files, as well as for files renamed or deleted by Mac and Windows clients. Optionally, you may delimit the file events by specifying a path to a certain directory. The specified path must be absolute.

```
registerfiles [directory path]
```

This option should be used carefully because it can cause hundreds of events per second and hence may dramatically slow down the server performance.

registerdirs

Register for directory change events. This allows receiving all events for newly created directories, as well as for directories renamed or deleted by Mac and Windows clients. Optionally, you may delimit the file events by specifying a path to a certain directory. The specified path must be absolute.

```
registerdirs [directory path]
```

registertype

Register for file change events with a particular Mac file type. The “opisrv” process will report all changes for a given file type done by a HELIOS file server. For example, *registertype "EPSF"* will report a list of all file changes of the type EPSF. Optionally, you may delimit the file events by specifying a path to a certain directory. The specified path must be absolute.

When used with *protocol 0*, `registertype "\0\0\0\0"` is a wildcard register to report all file changes done by clients. This option should be used carefully because it can cause hundreds of events per second and hence may dramatically slow down the server performance. See also **Wildcard in *protocol 1* or *protocol 2***.

registersuffix

Register for file change events with a particular Windows file suffix. The "opisrv" process will report all changes for a given file suffix done by a HELIOS file server. For example, `registersuffix ".eps"` will report a list of all file changes of the file suffix ".eps". Optionally, you may delimit the file events by specifying a path to a certain directory. The specified path must be absolute.

When used with *protocol 0*, `registersuffix "\0"` is a wildcard register to report all file changes done by clients. This option should be used carefully because it can cause hundreds of events per second and hence may dramatically slow down the server performance. See also **Wildcard in *protocol 1* or *protocol 2***.

switchprotocol0

Switch to old output format (up to ImageServer 2.5 compatibility; *protocol 0*). This behavior (*protocol 0*) is the default.

switchprotocol1

Switch to extended output format.

switchprotocol2

Switch to advanced format including *dirclose* events and flags (*stream*, *resource*, *Finder info*, etc.)

9.2.2 Notification features (newer protocols)

In addition to the event type and the path to the file, the protocols *protocol 1* and *protocol 2* show also the ID of the user who caused the file change event.

For “close” events, the string may contain flags that show which parts of the file have been changed:

```
event path [path2] uid [flags] [stream [stream2]] [fileid]
```

`event`	close, rename, exchange, delete, dirclose, dircreate, dirrename, dirdelete
`path`	File path
`path2`	Second file path (rename, exchange, dirrename)
`uid`	User ID
`flags`	`r` Resource; `f` Finder info; `s` Stream; `i` File ID; `c` File created (Windows only)
`stream`	Name of stream
`stream2`	Name of second stream
`fileid`	File ID

Example: (`r` = Resource fork)

```
4 - close "/opitest/InDesign/Project1" 105 r
```

Example: (`s` = File stream)

```
4 - close "/opitest/InDesign/Project1" 105 s "Comments"
```

Example: (`f` = Finder info)

```
4 - close "/opitest/InDesign/Project1" 105 f
```

Example: (`i` = File ID)

```
4 - dirrename "/opitest/untitled folder" "/opitest/WORK" 105 i 10736
```

Wildcard in *protocol 1* or *protocol 2*

The file event commands `registertype` and `registersuffix` may also be specified with the newer output formats (see `switchprotocol1` and `switchprotocol2`). If you wish to set wildcard registrations for file types and suffixes under *protocol 1* or *protocol 2*, use the `registerfiles` feature instead. The old protocol wildcard statement `registersuffix "\0"` is not allowed with *protocol 1* and *protocol 2*, whereas `registertype "\0\0\0\0"` will be interpreted as an empty file type.

Example:

```
$ socket localhost 2002
0 - Welcome to the HELIOS ImageServer event listener
```

The protocol for the session is switched to *protocol 1*:

```
switchprotocol1
0 - OK
```

Then a wildcard command is set, which limits the notifications to files below the stated path:

```
registerfiles "/opitest/InDesign"
0 - OK
```

When duplicating the file “Project1”, the following file event is issued (Note that the user ID “105” is displayed when *protocol 1* is active):

```
4 - close "/opitest/InDesign/.DS\_Store" 105
4 - close "/opitest/InDesign/Project1 copy" 105
4 - delete "/opitest/InDesign/Project1 copy" 105
```

10 Troubleshooting

This chapter is meant to offer assistance in case your ImageServer software does not work as expected. It summarizes important facts that have already been mentioned, but are distributed among various chapters of this manual.

The OPI server does not generate a layout image when you save or copy a high-resolution original.

There may be different reasons for this problem. First of all, the generation of layouts must be active for the network volume that contains your original image file (see 4.2 “Volume settings”). ImageServer does not generate layouts on your local hard disk.

Secondly, make sure that you did not save the original image in a folder (or subfolder) for which the “%0” option is set (see 5.2 “Define folder specific OPI settings”).

Finally, it might be possible that the file format you use for your high-resolution original is not supported by ImageServer (see 3.2.3 “Supported file formats”).

Please note that – as long as you do not change certain parameters with the “layout” program – resolution conflicts are impossible. ImageServer always generates layouts even if the resolution of the original file is smaller than the value specified with the `-r dpi` parameter for layouts (or smaller than 72 dpi, if the default setting has not been changed). In that case, the layout image inherits the resolution of the original.

If the generation of a layout image has failed, an image file with the error message shown in Fig. 10.1 is placed instead.

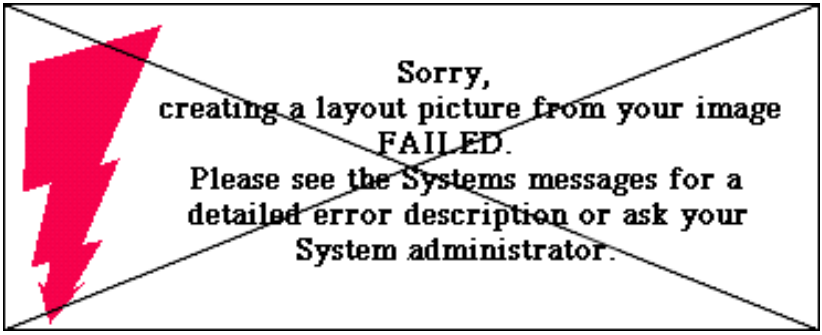


Fig. 10.1: Error message for failed layout generation

Images are missing on the printouts.

If images are missing on your printouts, check if you have placed any high-resolution images in your document and selected one of the `Omit ...` options in the print dialog. This combination of settings may lead to missing images because, by default, high-resolution images are not replaced during printing (see 5.6 “OPI print options – Printing from QuarkXPress and InDesign”). Images may also be missing if you have moved the image files to a location OPI cannot resolve, e.g. your local hard disk. In that case, however, you get an error message (see 3.7 “How ImageServer finds and replaces image files”). The printer log file includes more information on the exact cause of the error.

Images on your printouts do not show the resolution you expected.

This problem may occur if you have specified a certain value for downsampling on your printer queue (see 4.3 “OPI printer queue settings”). In that case, some images will be re-calculated according to the value you entered for output resolution and others will not be changed at all. In fact, images with a resolution smaller than that specified for downsampling will not be re-calculated by the OPI server, because increasing the resolution automatically decreases the dimensions of an image. This would entail upward scaling and lead to a loss in quality. Furthermore, downsampling cannot be applied to object-based image files, as e.g. FreeHand illustrations.

Please note that if you switch off downsampling, the output resolution depends on the individual image files. Even if you only print layouts, the resolution may be different for each image. This is due to the fact that you may re-define at any time the layout resolution on your server and that existing layouts are not updated automatically (see 4.1 “ImageServer settings”).

Printing separations does not work as expected.

This problem may occur due to application-specific settings that do not properly cooperate with ImageServer. Details are given in chapters 5.7 “Using QuarkXPress with ImageServer” to 5.8 “Using InDesign with ImageServer”.

Other problems:

Select proper file names. Do not use the same name twice in one folder, even if the extension is different. For example, if you save “flowers.psd” and “flowers.tif” to the same folder, they both will produce a “flowers.tif” layout image. The contents of this layout can be from either of the original files.

You may not remove the “ICC-Profiles” volume from the server – even if you do not activate color matching. This would disable OPI completely, because in specific situations OPI uses the standard profiles from the “ICC-Profiles” volume “behind the scenes” and must therefore be able to access this volume at any time (see 3.6 “How ImageServer applies color matching”). If you have already removed the “ICC-Profiles” volume you must re-install ImageServer. You can start the installation program from your server, but you will need the CD to re-install the profiles. Color matching mainly depends on the profiles you use. In case your printouts show unexpected results you should check the image profiles (see 5.3 “Tag images with ICC profiles”) and the profiles that have been selected for printer and/or proof printer (see 4.3 “OPI printer queue settings”).

Note: If you detect a problem that is not mentioned above, you may also refer to the explanations given in 2.3 “Known limitations”.

---

11 Technical support

If your ImageServer software does not work as expected, there are different ways of looking for help:

- Check the `Printer Log Files` and `System Messages` via HELIOS Admin for error messages.
- For errors that are reported by error messages, please read the "Error description" carefully; it may give you the very hint you need for proceeding. In case it does not, check the user manual again (pay special attention to 2.3 "Known limitations"). If the information you need is not in the manual, it might be found on the HELIOS website (**www.helios.de**).
- Make sure that your HELIOS software is up to date. Product updates are listed on the HELIOS website.
- Check the HELIOS support website for any "Tech Info" related to the problem.
- For problems you cannot solve on your own, you may request help from your dealer. If you do so, please note that for a fast problem localization the dealer needs some specific information from you (see **Set up an error report**).

Set up an error report

A complete error report should include:

- Contact information (such as company name, name of the person to contact, e-mail address, fax/phone number)
- Product information

- ❍ A problem description
- ❍ Printouts or screenshots that illustrate the problem
- ❍ Additional information (e.g. about workarounds you may have tried)
- ❍ Also always include the complete "Versions" file. This file can be accessed and saved from HELIOS Admin (`Lists > Versions`).

To allow your HELIOS dealer to solve the problem, be prepared to submit the following information in writing:

- ❍ Include the complete "Versions" file (see above)
- ❍ Specify the type of your Mac/PC and the Mac OS/Windows version number
- ❍ List the exact error messages – if there are any
- ❍ List the Mac/PC applications you are using (name and version number)
- ❍ Which network and printer drivers are currently installed (include version numbers)?
- ❍ Give us the settings you have entered in the HELIOS Admin dialogs (screenshots may be helpful)
- ❍ If printing does not work properly:
  - First of all – What is the problem?
  - Specify vendor and model name of your printer
  - Specify document and image file formats. Does the problem also occur with other document or image file formats?
  - Is the problem restricted to only one client computer?
  - Does the problem affect all printers or only some of them?
  - Are you able to print successfully from a different Mac/PC OPI application?
  - Does the problem persist if you place and print the high-resolution original images with no OPI settings active on the printer queue?

- Does the problem persist if you change the resolution of the high-resolution image, the low-resolution image or the output device?

❍ If generating/placing layouts does not work properly:

- First of all – What is the problem?
- Specify the exact file name of the high-resolution image including the complete Mac/PC volume and folder path
- How did you create the layout file?

  a): Automatically
  b): Using the "touch" or "opitouch" programs
  c): Using the "layout" program

- Specify the file format and color mode of the image file
- Which application has been used to create or edit the image file?
- Have you saved your image file from a Mac, PC or UNIX computer?
- Test: Does the problem persist if you open the image file with Photoshop and save it under a different name?

❍ If color matching does not work properly:

- First of all – What is the problem?
- List all active profiles (scanner/image, printer, proof)
- Specify image file formats and color modes. Does the problem also occur with other images?

A About OPI settings, typical workflows, and conflicts

In the following, we give a few examples of how to combine OPI settings and application specific print options when aiming at a particular workflow. Fig. A.1 illustrates a typical situation when working and making use of all advantages offered by ImageServer:

Workflow 1

- *Step A*: The document in your layout application only contains low-resolution layout images. Thus, you can work rather fast on your Mac or PC, because you do not have to handle huge high-resolution image files.
- *Step B*: `Print Layout` Images is switched off on your printer queue. This is an HELIOS Admin default setting and lets you print the high-resolution originals.
- *Step C*: `Include Images` for `TIFF` and `EPS` images is *NOT* selected in the print dialog of your layout program. Thus, the print job that is sent from your Mac computer to the server does not contain image data and is therefore rather small.
- *Step D*: `Replace Layouts` is active on your printer queue. This is a HELIOS Admin default setting and induces the OPI server to replace all layout images in the document (or to fill the blanks if *NONE* of the `Include...` options is selected).
- *Step E*: A source profile is tagged to every high-resolution image that will now be included in the print job.
- *Step F*: An ICC printer profile has been selected for your printer queue. Thus, the OPI server will perform color matching.

❍ *Result*: The printout contains your high-resolution images. Each image will be color matched correctly.

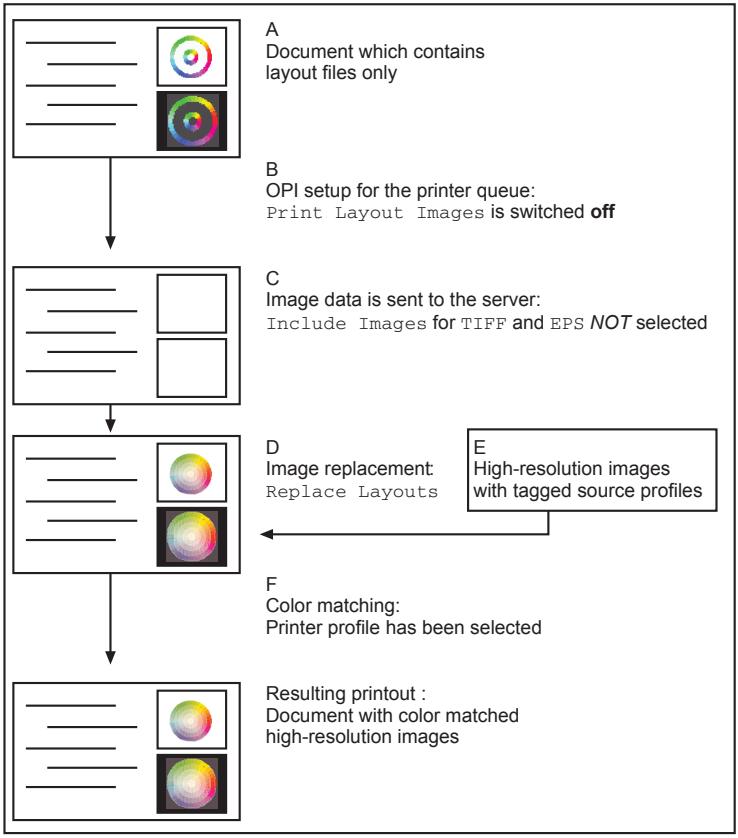


Fig. A.1: A typical workflow when using ImageServer

Workflow 2

If you have pre-separated images and want to print them "as is" without applying color matching, you can do that without changing a lot of parameters. The only thing you have to do is set the `Default Printer Profile` to `None` on your printer queue (*Step F* in Fig. A.1).

Workflow 3

If you want to carry your documents to a third-party company for test printouts, it might be difficult when using a slow Internet connection. In that case, you may provide the company with the document files and the layout images only and then let them use the `Print Layout` Images option (*Step B* in Fig. A.1). You can even ask them to color match the images on the printouts; for that purpose, however, you have to make sure that all the layouts have a source profile.

Example of possible conflicts

The moment you add high-resolution originals to your document (*Step A* in Fig. A.1), you have to change some settings for printing. You may either use the `Include Images` option (*Step C*) or switch the `Replace Images` option to `All` (*Step D*). Replace `All` must be selected if you want OPI to apply color matching to all your images. If you do not induce the replacement of your high-resolution originals, they will not be color matched.

The image that is finally included in the print job has to have a tagged source profile if you want to achieve high-quality color results (*Step E* in Fig. A.1). If you print layouts (*Step B*), you have to make sure that these layouts have a source profile, too. Layouts are not always tagged with a source profile during layout generation and they do not automatically inherit the profile of the original image. OPI will display a warning if image profiles are missing.

Note: For problems that arise even though you have chosen your OPI settings properly, you may refer to 2.3 "Known limitations".

B Colors, color matching, proof printing – basic concepts

About color modes/spaces

Color is created through the interaction of light, an object, and the eye. The "visible spectrum" contains millions of colors.

Screens and monitors produce colors by means of red, green and blue light (RGB). The light intensities make up a given color. Scanners and digital cameras also work with RGB colors. They read the amounts of red, green, and blue light that are reflected from an image (or transmitted if you scan transparent images). The RGB color space is smaller than the visible spectrum of light. RGB colors are device dependent, they vary with scanner or monitor characteristics.

Color printing is based on the CMYK color space. Cyan, Magenta, Yellow, and Black inks are mixed on paper to produce a given color. The CMYK color space is even smaller than the RGB color space. CMYK colors vary with printer, ink, and paper characteristics.

The CIE (*Commission Internationale de l'Eclairage*) created different color spaces that specify colors in terms of human perception. One example is the CIELab color space. Lab colors are device independent.

About color matching

If you want to capture, edit, place, and print an image, you have to transform the scanned RGB values into CMYK values for the output device. This process is called separation. If you want to have predictable color results, you must have calculation models that match RGB to CMYK during separation. As

RGB and CMYK values are device dependent, you can never exactly define a color. E.g. capturing one particular color with three different scanners or cameras will produce three different sets of RGB values. Thus, you will have different input values for the transformation into CMYK. You must have a specific transformation table for every possible scanner–printer combination. If you have a device independent color space like Lab, you can exactly define the color you scanned. Considering the color characteristics of the different scanners – which are described in the device profiles (see **ICC, ICC profiles** in G "Glossary") – you can transform the different sets of RGB values into one single Lab color definition. Now, you have only one single input value for the transformation into CMYK. Fig. B.1 and Fig. B.2 illustrate the effect of a device independent color space (Please note that you may as well have an RGB device on the output side, e.g. an imagesetter).

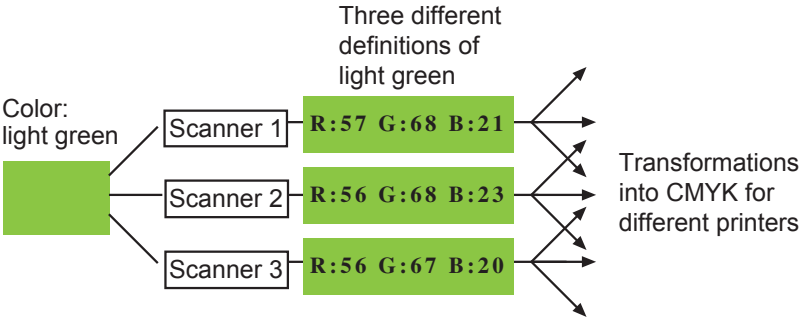


Fig. B.1: Color transformation without Lab values + profiles

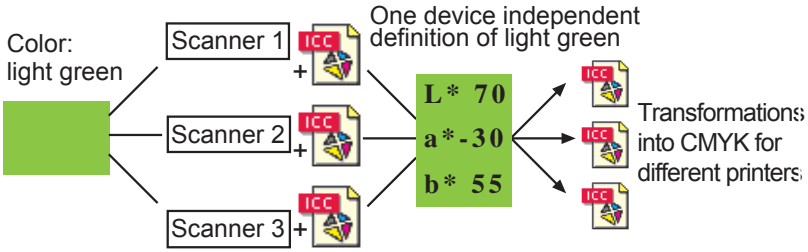


Fig. B.2: Color transformation with Lab values + profiles

Note: Color matching does not always lead to precise reproduction. Some Lab or RGB colors cannot be printed – some can, but not by every given printer. The color matching module has to make sure that deviations are as small as possible. This can be achieved by *gamut mapping* (see G “Glossary”).

ImageServer is able to transform colors from different color spaces like RGB and CMYK into Lab values and vice versa. Thus, a device independent color space is used for the interchange of color data. The color matching module that is available with ImageServer uses ICC based profiles for color matching.

With ImageServer, the color spaces of single images may be changed as necessary for layout generation and printing. The transformations, however, do not affect the original high-resolution images. These images remain unchanged in terms of color space and file format.

About proof printing

All the color data that can be matched for a printer can also be matched for a proof printer. A proof printer is used to simulate the output results of another printer or a press. The color data transformation and gamut mapping are accomplished for the printer you want to use as final output device (e.g. an offset press). The resulting CMYK values are then transformed into the specific

CMYK values of the proof printer. To guarantee a precise simulation, the gamut of the proof printer must not be smaller than that of the printer because there is no gamut mapping from printer to proof. Fig. B.3 shows how color data are transformed if you print to a proof.

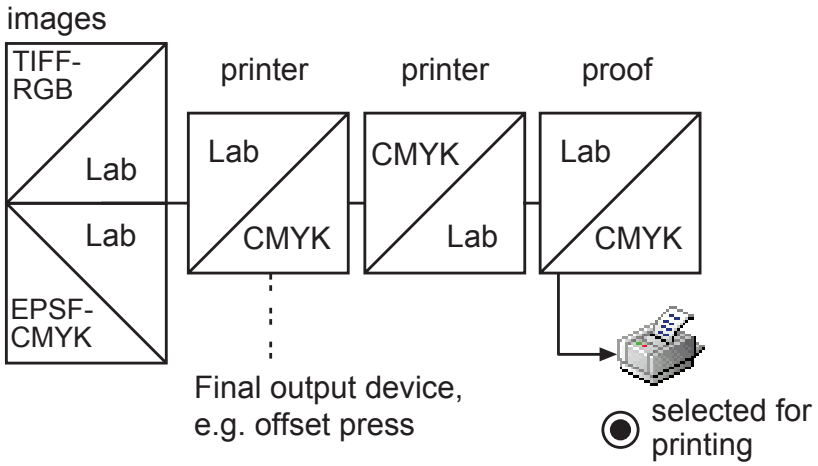


Fig. B.3: Transformation of color data during proof printing

With ImageServer, color matching must be activated separately for each printer queue. You have to define a printer profile for your output device and, optionally, a proof profile if you want to simulate your output results on a proof printer. Fig. 4.7 in 4.4 “ICC printer settings” shows the required settings for simulating a newspaper press on a color LaserWriter.

B.1 DeviceLink

ImageServer supports DeviceLink ICC profiles, which are primarily used to convert CMYK colors of one device to CMYK colors of another device, e.g.

Euroscale Offset CMYK to gravure printing CMYK. Color conversions using DeviceLink profiles can have properties that are impossible using conventional ICC profiles. Conventional ICC profiles represent color characteristics of color devices (such as scanners, digital cameras, monitors, displays, and printers) and abstract color spaces (such as CIELab). In contrast to conventional ICC profiles, DeviceLink profiles represent direct mappings from one specific color space to another specific color space. Thus, DeviceLink profiles have two color spaces: an input color space and an output color space. Conversions between CMYK color spaces using two conventional ICC profiles have the fundamental disadvantage that the 3 component CIELab space is used as an intermediate step between the input and output 4 component CMYK color spaces. The use of CIELab as an intermediate color space results in the loss of information, which objects should be printed using only the black channel, e.g. black text. Typically conventional conversions from CMYK to CMYK convert Black channel only objects to black objects using Cyan, Magenta and Yellow. Additional ICC profiling software can create DeviceLink profiles for CMYK to CMYK conversions that preserve black channel only objects or which replace Cyan, Magenta and Yellow with Black ink while preserving the color appearance to cut cost and to improve ink drying properties during printing.

Batch image conversion using DeviceLink

The "layout" image conversion tool supports DeviceLink conversions via the `-o DevLinkProfile=mydevlink.icc` option. If a DeviceLink profile is specified, the input image must have the same color space as the input color space of the DeviceLink profile. The color space of the output image is determined by the output color space of the DeviceLink profile.

DeviceLink support when printing PDF files, in PostScript OPI and PDF-native OPI

When printing PDFs directly via "pdfprint" or Acrobat plug-in or indirectly via OPI, a DeviceLink profile can be specified within the printer queue. All input color objects with the same color space as the input color space of the DeviceLink profile are color matched using the DeviceLink profile. All other color objects are converted using the input profile and printer profile. Printer

queues with DeviceLink profile require a printer profile with the same color space as the output color space of the DeviceLink profile. Assumed that a CMYK printer profile and a CMYK to CMYK DeviceLink profile have been specified, the color data will be:

1. RGB and Lab input data

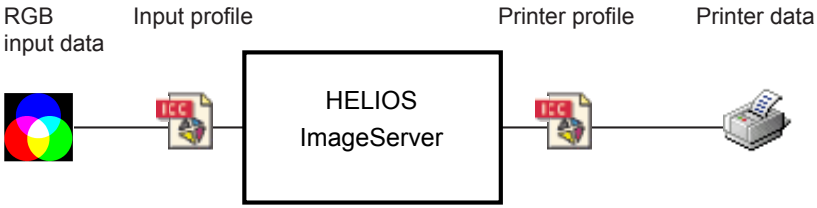


Fig. B.4: Conversion using input and printer profile

2. CMYK input data

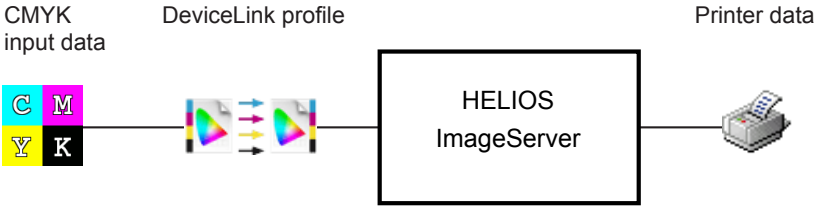


Fig. B.5: Conversion using DeviceLink profile

Proofing DeviceLink

PDF printing, PostScript OPI, and PDF-native OPI using DeviceLink profiles supports proofing, too. Assumed that a CMYK printer profile, a CMYK proof profile and a CMYK to CMYK DeviceLink profile have been specified, the color data will be:

1. RGB and Lab input data

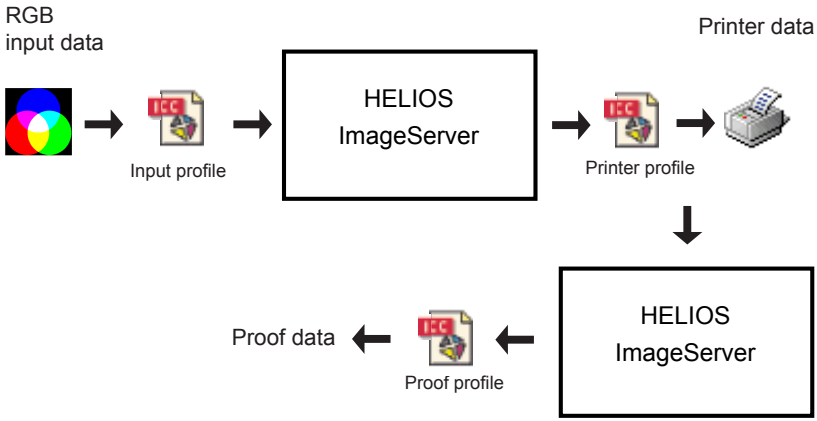


Fig. B.6: Conversion using printer and proof profile

2. CMYK input data

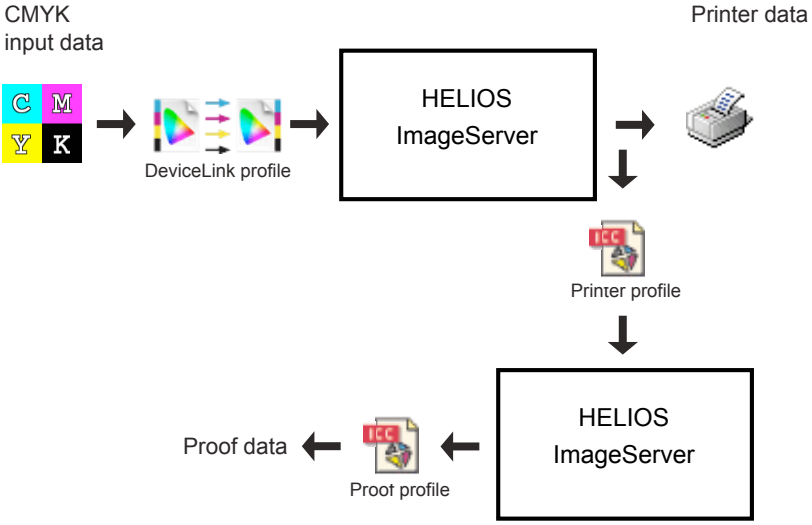


Fig. B.7: Conversion using DeviceLink, Printer, and proof profile

Note: As specified by the ICC standard, DeviceLink conversions do not have rendering intent support. Rendering intent settings do not alter DeviceLink conversions.

Preserve colors

ImageServer includes options to preserve black, white, gray, and primary CMY colors. E.g., with the "Preserve … Colors – Black" option RGB black is detected and converted to CMYK black without Cyan, Magenta, and Yellow. The "Preserve … Colors – CMY" option applies to CMYK to CMYK conversions only and preserves the input color if, and only if black is 0% and exactly one CMY primary color is 100% and all other primary colors are 0%.

The preserve color options can be specified separately for raster image data and PDF vector data.

Grayscale ICC support

ImageServer supports grayscale ICC profiles, i.e. grayscale images with an ICC profile are properly processed. Grayscale ICC profiles can be specified as printer profiles and as output profiles for image conversions.

Black Point Compensation (BPC)

BPC is not part of the ICC specifications that have been issued by now. With BPC, the CMM evaluates the lightness range of the used devices and scales between them. Until now, BPC has been available nearly exclusively in some Adobe applications. The BPC option within HELIOS products is available for the *Perceptional*, *Relative Colorimetric* and *Saturation* rendering intents. HELIOS provides the BPC option as 3 additional rendering intents:

4 – for *Perceptual with BPC*
5 – for *Relative Colorimetric with BPC*
6 – for *Saturation with BPC*

Colors converted with BPC are highly source and destination gamut dependent. Some images look better with BPC, others do not. Therefor, it is recommended to check carefully when BPC should be used. However, a refinement of this approach is now available, the *Black Plane Compensation*.

Black Plane Compensation

While the BPC transformation only affects pure gray tones, the Black Plane Compensation in the CMYK color space limits the maximum ink coverage. Therefore, not only pure black is black, but in reality many dots in the vicinity. Thus, not only a single dot is transferred to the target black point, but a whole layer of dots. This means that shadows are still retained, but they appear calmer and less noisy:

7 – for *Perceptual with Black Plane Compensation*
8 – for *Relative Colorimetric with Black Plane Compensation*
9 – for *Saturation with Black Plane Compensation*

C ImageServer and color space

C.1 Hexachrome printing

This section describes, how the RGB-to-Hexachrome color space printing workflow can be set up.

The workflow which is described below only applies when using QuarkXPress (versions 3-9). Adobe InDesign does currently not support Hexachrome printing.

To enable Hexachrome output on a queue, certain preferences for this specific queue must be set:

➢ For this to do, login as “root” and type:

```
# cd /usr/local/helios
# bin/prefvalue -k 'Printers/<printer queue>/ProcessInks' -t strlist
                  "Cyan","Magenta","Yellow","Black","Orange","Green"
```

➢ To remove the preference again, type the following:

```
# cd /usr/local/helios
# bin/prefvalue -k 'Printers/<printer queue>/ProcessInks' -d
```

See also **ProcessInks** in 9.1 “ImageServer preferences”.

Additionally, the order of the specified colors must match the order in the used Hexachrome profile. If the order in the profile is different, modify the above mentioned command line according to the order in the profile.

Note: Please note that an inaccurately set preference may lead to faulty results! Read more about the usage of “prefvalue” in the HELIOS Base manual.

➢ Assign the Hexachrome profile as *printer profile* to the specific queue you want to use.

To enforce a Hexachrome color separation, the “ForceHexachrome.eps” must be placed in each page of the document (see below).

If you are using PrintPreview, the spot colors “Green” and “Orange” must be defined in the HELIOS Admin Spot Color editor (see C.3 “The Spot Color editor”). This enables PrintPreview to create a correct composite preview.

All placed images have to be “tagged” with an ICC profile, or must be assigned a default profile per folder/volume.

The HELIOS CD includes a file named “ForceHexachrome.eps” in the `sample-images` folder. This file defines six process colors: *Cyan*, *Magenta*, *Yellow*, *Black*, *Green* and *Orange* and needs to be placed in the layout application to enforce the six-channel separation output for printing. The file does not contain printable data.

Some Hexachrome profiles are not compatible with the ImageServer software. E.g., a Hexachrome profile using the data color space “MCH6” is not supported by ImageServer. However, in order to still make use of the profile the data color space “6CLR” must be specified.

➢ Modify the profile in a data editor, e.g. “HexEdit”, (**hexedit.sourceforge.net**). Open the profile and replace in bytes 17 to 20 “MCH6” with “6CLR” (without quotes!).

C.2 Multicolor support for image conversion

It is possible to create multicolor images (e.g. Hexachrome) if appropriate Multichannel ICC profiles (5-8 channels) are used. The option `MultiInkSeparationColors` (see 6.1.8 “Image conversion options”) is optional and can be used to define color names. Otherwise either color names from the ICC profile are used or own color names are generated (many Hexachrome profiles do not contain predefined color names).

Example:

```
$ layout -oOutputProfile="Hexachrome.icc" -l PCRI_chart.jpg
  "{8BPS}hexachromeimage.psd"
```

C.3 The Spot Color editor

C.3.1 What is the spot color editor good for?

ImageServer and PrintPreview cannot calculate a composite preview of a separated print job with spot colors if the application which has created the document does not send any process color information for the spot colors in the document.

For this purpose, PrintPreview has built-in color tables which recognize various spot color names, so process color information can be provided for the composite preview generation in PrintPreview. However, if a used spot color name does not match the exact entry in the built-in table, PrintPreview generates a composite preview which lacks the spot colors.

The spot color editor enables the user to define a preview color for a custom spot color name. This allows PrintPreview to include spot color objects in the composite preview.

C.3.2 Setting up the Spot Color editor

Define spot color names

A list of spot colors can be created as follows:

➢ In HELIOS Admin, open `Spot Colors` from the `Settings` menu.

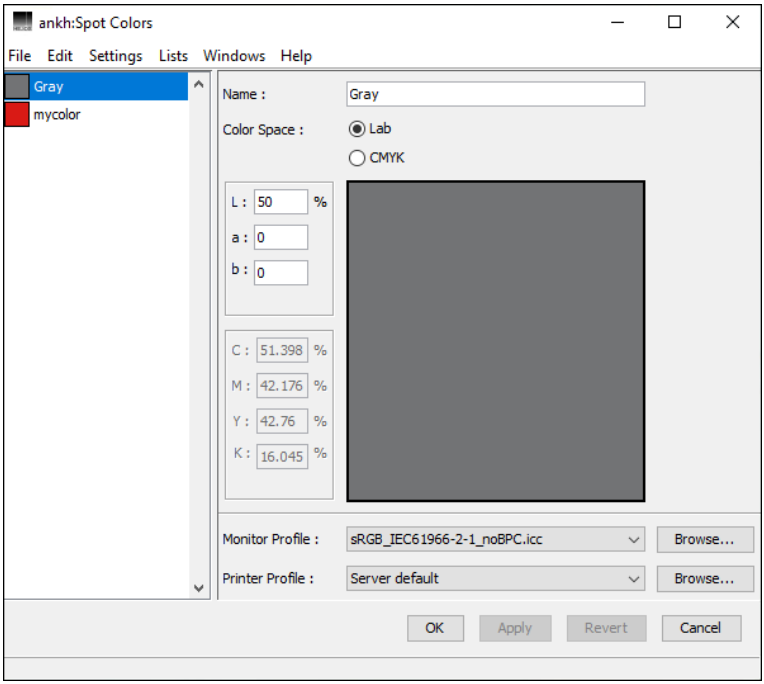


Fig. C.1: The `Spot Colors` definition window

The “Spot Colors” definition window (Fig. C.1), which may already contain the names of some previously defined spot colors, opens. For the definition of a new spot color in the spot color editor, CMYK or Lab values are accepted.

➢ Select `File > New` from the menu bar.

➢ In the `Color Space` section, select the appropriate color space and enter the values of your spot color. After that, specify a name for the defined spot color and confirm the settings by clicking the `Save` button.

The `Revert` button sets all entries back to default or, if you have only modified an entry, reverts to the latest saved version. After closing the window the new spot color appears in the list (Fig. C.1).

C.3.3 Accurate spot colors with ColorSync/ICC

RGB profiles

Fig. C.2 shows the ColorSync profile settings for an RGB monitor on Mac OS 9.

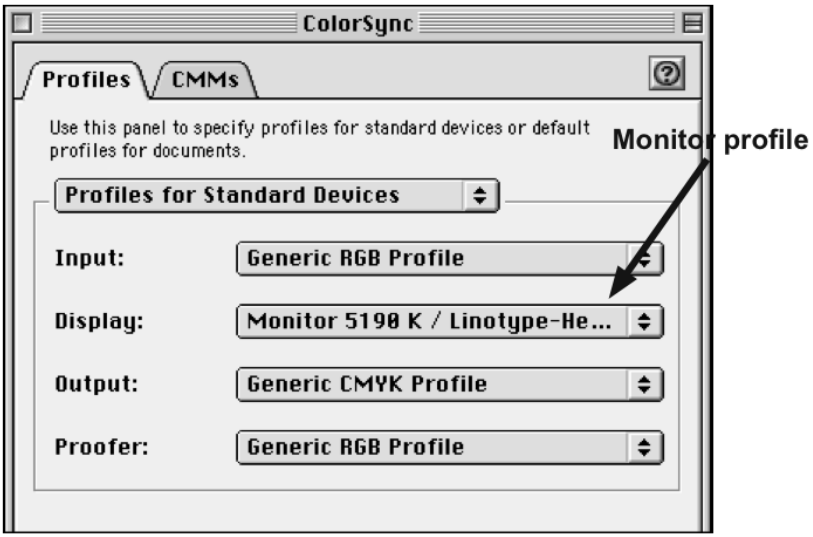


Fig. C.2: ColorSync profile for RGB

Note: On OS X, go to “System Preferences > Displays” and select in the `Color` tab the display profile that suits your monitor.

CMYK profiles

Fig. C.3 shows the profile settings for a CMYK printer on OS X. Both settings are used to represent accurate previews in the “Spot Colors” definition window.

On OS X, the ColorSync utility is used to set profiles.

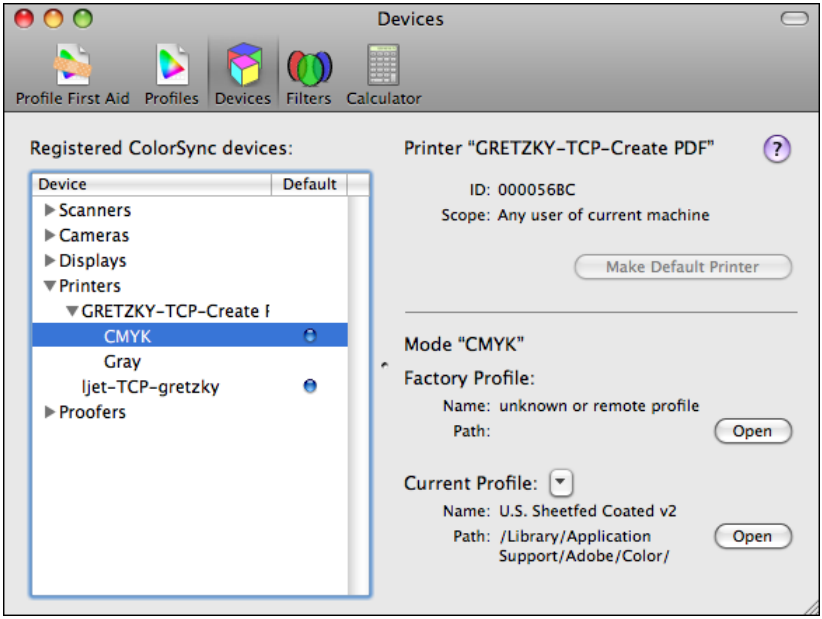


Fig. C.3: `ColorSync` profile for CMYK

On Mac OS 9, suitable printer profiles can be chosen in the `ColorSync` dialog:

➢ Open `Control Panels > ColorSync` from the Apple menu and assign appropriate profiles from the pull-down menu to the output devices that you want to use.

These chosen profiles are displayed in the Spot Colors definition window. Fig. C.4 shows an example where the spot color "Gray" has been defined in the Lab color space. The color patch displays the accurate color according to the monitor profile. The CMYK values displayed adjacent to the color patch indicate how the Lab color is converted for printing.

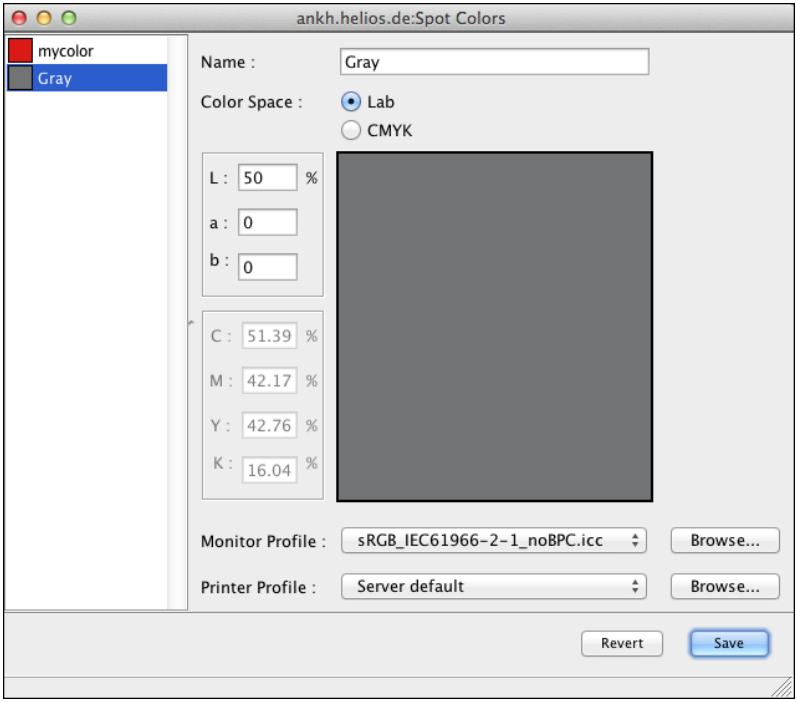


Fig. C.4: Lab to CMYK

Fig. C.5 shows an example where the spot color “Gray” has been defined in the CMYK color space, according to the selected printer profile. The color patch displays the accurate color according to the printer and monitor profiles. The Lab values displayed adjacent to the color patch represent the CMYK color in the Lab color space.

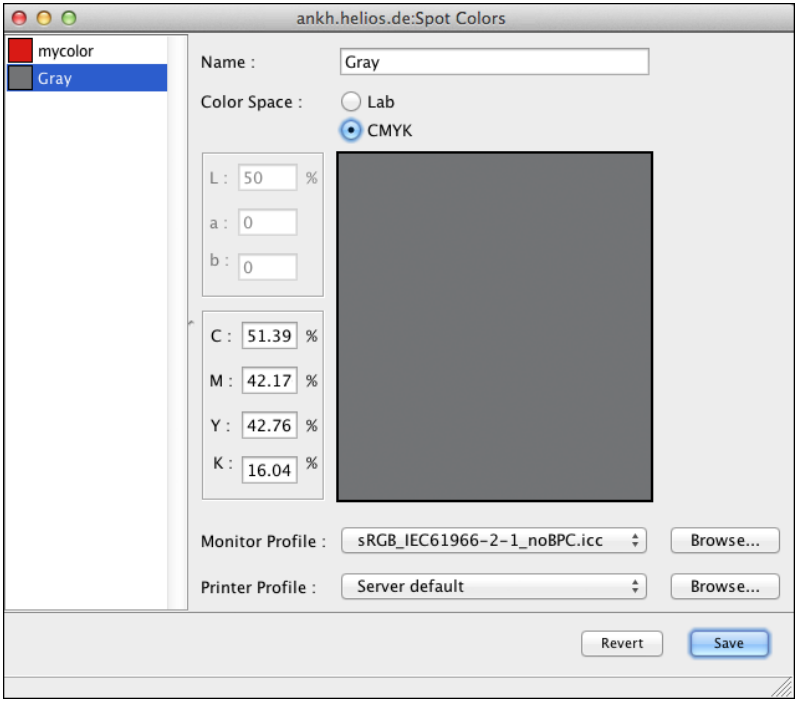


Fig. C.5: CMYK to Lab

Important: Make sure to use the appropriate monitor and printer profiles, which can be created with ICC profiling tools. Wrong profiles, e.g. the default profiles used by Mac OS 9 or bad profiles, will lead to inaccurate results!

Since colors are best and most reliably represented in the Lab color space, it is sensible to enter the color in Lab values which have been determined by a spectrophotometer, e.g. “GretagMacbeth”.

Note: To make profile changes in the `ColorSync` dialog valid, the `ColorSync` dialog and the `Spot Colors` definition window must be closed. The new `ColorSync` profiles will be used at reopening the Spot Colors definition window.

D Raster image transparency

D.1 Categories of image documents with transparency

In recent years the use of transparency has become more and more common in media production. There are three different categories of image documents with transparency:

1. Raster image files (e.g. TIFF)
2. Native page layout application documents (e.g. Adobe InDesign)
3. PDF 1.4 and later documents

PDF 1.4 and later documents are supported by PDF HandShake. The support of ImageServer for native page layout application documents consists of the InDesign plug-in to generate previews of native InDesign documents (see 5.8.1 “The HELIOS Preview plug-in”) and of the QuarkXPress XTension to generate previews of native QuarkXPress documents (see 5.7.3 “HELIOS Preview XT”). The following text explains transparency in raster images only.

D.2 Transparency in raster images

There are several different techniques to create transparency in raster images:

- Clipping paths

- Alpha and mask channels
- Transparency channels
- Multiple layers

D.2.1 Clipping paths

The most common way of creating transparency in prepress production uses clipping paths. Clipping paths isolate foreground objects and make their background transparent. Clipping paths are vector-based line structures which create hard edges between foreground objects and transparent background.

D.2.2 Alpha and mask channels

Some image formats support alpha or mask channels.

In all raster images each pixel is represented by a set of numerical values for all colorants of the image, depending on its color space. If an image contains an alpha channel, each image pixel is represented additionally by a numerical value which specifies the opacity of that pixel. In alpha channels, 100% specifies an opaque pixel and 0% specifies a transparent pixel. Mask channels are exactly like alpha channels except for their polarity. In mask channels, 100% specifies a transparent pixel and 0% specifies an opaque pixel. Alpha and mask channels offer multilevel transparency for drop shadows and soft edges of objects, also known as *feather effects*. If an image format supports alpha and mask channels, any image of that format may contain zero, one or multiple alpha or mask channels. The extra channels can be identified by their unique names. If an image contains one or more alpha or mask channels, the user has the option to apply none of the extra channels or exactly one of them.

D.2.3 Transparency channels

Transparency channels are much like alpha channels. Raster image formats supporting transparency channels can contain exactly one or no transparency channel. The user usually does not have the choice whether to apply the transparency channel or not. The application of a transparency channel is mandatory. Each image pixel is represented by a set of numerical values for the visible colorants and one additional numerical value containing the opacity information. 100% specifies an opaque pixel and 0% specifies a transparent pixel. Transparency channels offer multilevel transparency, too. A well-known example of an image format with transparency channels is PNG with RGBA color.

D.2.4 Multiple layers

The most complex way of creating transparency currently is to combine multiple layers in Adobe Photoshop. Only TIFF and Photoshop native documents support multiple layers. TIFF images always contain a main image which is a combination of all Photoshop layers. But Photoshop native images contain a main image which is not necessarily a combination of all Photoshop layers. If ImageServer reads an image with multiple Photoshop layers, it uses a combination of all layers.

The following table lists the transparency capabilities of the most important image file formats:

Raster image format	Clipping paths	Alpha/Mask channels	Transparency channels	Multiple layers
BMP	–	–	√*	–
DCS 1	√	–	–	–
DCS 2	√	–	–	–
HELIOS raster PDF	–	–	–	–
JBIG2	–	–	–	–
JPEG	√	–	–	–
JPEG 2000	√*	–	√*	–
Photoshop	√	√	√	√
PICT	–	–	√*	–
PNG	–	–	√	–
Scitex CT	–	–	–	–
TIFF	√	√	√	√
Raster EPSF	√	–	–	–

* Image format standard or Photoshop support this feature but ImageServer does not.

D.3 How transparency properties of images are displayed in Adobe Photoshop

- If an image contains a clipping path, the clipping path is displayed in the Photoshop “Paths” window with its name displayed in bold font.
- If an image contains an alpha or mask channel, the extra channel is displayed in the “Channels” window. Every channel in the “Channels” window has an icon of the current image. If you double-click this icon, nothing happens for process colors, but a “Channel Options” window is opened for masked, alpha, and spot channels. This “Channel Options” window displays properties of extra channels.

- If an image contains a transparency channel, it is NOT displayed in the "Channels" window. Instead, the "Layers" window contains no background layer entry.
- If an image contains multiple layers, the "Layers" window contains more than one layer entry.

D.4 Raster image transparency support in ImageServer

Adobe InDesign and QuarkXPress (versions 3-9) support OPI workflows using high-resolution images with clipping paths, but they do not support OPI workflows using high-resolution images with alpha, mask or transparency channels or with multiple layers. More precisely, if ImageServer generated layouts with alpha, mask or transparency channels or with multiple layers, and these layouts are placed in Adobe InDesign, then InDesign does no longer generate OPI information in PostScript print output or in exported PDF documents. The same applies to QuarkXPress. Therefore ImageServer does not generate layouts with alpha, mask or transparency channels or with multiple layers.

ImageServer supports transparency via alpha, mask and transparency channels, and via multiple layers for the file formats TIFF, Photoshop native, and PNG. However, support for multiple layers in ImageServer is limited; it cannot write images with multiple layers. If an image with multiple layers is converted to another image, the multiple layer information is converted to an alpha or transparency channel. If ImageServer reads an image with multiple layers, it uses the main image only, see D.2.4 "Multiple layers". The main image of TIFF images is always the combined image of all layers. When converting a TIFF image to another image using ImageServer, only the main image is transferred. Viewing the converted image in Photoshop, it is visually equal to the original image viewed with all layers, but the individual layers of the original image can no longer be viewed separately in Photoshop. This conversion loses information only in the sense that further editing of converted images is limited.

The combination of all layers is the default view of an image in Photoshop, and the Photoshop default view is the very result of an image conversion users usually expect. On the contrary, the main image of Photoshop native images may not include all visible layers. For Photoshop native images, there may also be visible losses.

Another limitation is that transparency channels can be read from Photoshop native images, but only alpha channels can be written to Photoshop native images.

ImageServer tries to preserve transparency information in image conversions if the `IgnoreMasks` preference with default `FALSE` and the `UseAlpha` preference with default `TRUE` remain unaltered. If the `IgnoreMasks` preference is changed to `TRUE`, clipping paths in input images are ignored. If the `UseAlpha` preference is changed to `FALSE`, alpha, mask, and transparency channels in input images are ignored. If an input image contains transparency information that the output image format does not support, ImageServer tries to convert the transparency information to a type which the output image format does support. ImageServer does currently not support conversions between transparency information types that are not induced by output image format limitations. Here are some examples of raster image transparency support in ImageServer:

- If an input image with a clipping path is converted to the PNG image format, the clipping path is converted to a transparency channel since PNG does not support clipping paths.
- If you want to convert a TIFF image with a clipping path to a TIFF image with a transparency channel, the `ApplyClippath` option (see 6.1.8 “Image conversion options”) can be used.
- If a PNG image with transparency channel is converted to a Photoshop Native image, the transparency channel is converted to an alpha channel.
- If a TIFF or Photoshop image with an alpha channel is converted to a PNG image, the resulting PNG image does not contain transparency information. ImageServer does not convert an optional alpha channel to a mandatory

transparency channel. If the input image has more than one alpha channel, the resulting PNG image does not contain transparency information because only one transparency channel can be created, and there is no way currently to select one alpha channel to convert.

E Raster PDF format

E.1 Photoshop PDF format

Adobe Photoshop offers the Photoshop PDF file format. Photoshop help pages state implicitly that generic PDF documents and Photoshop PDF documents are different. Adobe did not specify the differences between generic PDF and Photoshop PDF in the years before 2008, and there is no indication that Adobe intends to document these differences in the future. In contrast to the documented and open Photoshop EPSF format, the Photoshop PDF file format is currently undocumented and proprietary.

You can safely assume that Photoshop PDF documents are valid PDF documents. Most applications that can read PDF documents can also read Photoshop PDF documents. With well-known PDF analysis tools it can be observed that Photoshop PDF documents are standard PDF documents with additional private Photoshop data contained in so-called page-piece dictionaries. Photoshop PDF documents store two copies of the image they contain. The first copy is stored according to the PDF standard and uses the compression filter specified during saving, and this compression filter may be lossy such as JPEG. The second copy is stored in the additional private Photoshop data and is compressed with the lossless ZIP compression. This redundant storage of image information requires a lot of disk space and makes the Photoshop PDF format ill-suited for prepress production. While some properties of Photoshop PDF documents can be observed without official documentation, many details of the Photoshop PDF format stored within the page-piece dictionaries remain unknown outside of Adobe.

When images are saved as Photoshop PDF documents, the “Preserve Photoshop Editing Capabilities” option is important. If this option is turned off, the saved PDF documents do not contain additional private Photoshop data, hence they are not Photoshop PDF documents, but generic PDF documents. Photoshop can open generic PDF documents only by rasterizing them. The Photoshop import of generic PDF documents often results in substantial loss of quality. A PDF document created by Photoshop without preserved editing capabilities contains one raster image. If this PDF document is re-opened in Photoshop, the raster image is interpreted as vector data which is rasterized. The raster of the raster image within the PDF document and the raster of the rasterization may not fit together well. Another problem with Photoshop rasterization of generic PDF documents is that it does not feature in-RIP separation. In Photoshop rasterization spot colors are converted to process colors or are lost entirely.

E.2 HELIOS raster PDF

The PDF format can be a very efficient container for raster images. PDF viewing applications are available for a wide range of platforms, the PDF format includes support for many color spaces, for many image compressions, for ICC profiles and for spot colors. Since the Photoshop PDF format is undocumented, it cannot be supported by ImageServer. Therefore ImageServer supports a new raster image format derived from PDF called “HELIOS raster PDF”. All non-HELIOS applications view HELIOS raster PDF documents as generic PDF documents. This fact has both advantages and disadvantages. The advantages of the HELIOS raster PDF format are:

- HELIOS raster PDF documents are as portable as generic PDF documents; applications that can read generic PDF documents can read HELIOS raster PDF documents, too
- All major color spaces such as Bilevel, Grayscale, Indexed, RGB, CMYK, and CIE Lab are supported

- All major image compressions such as CCITT G4, ZIP, JPEG, and JPEG 2000 are supported
- ICC profiles are supported
- Spot colors are supported
- XMP is supported
- OPI is supported

The disadvantages of HELIOS raster PDF documents result from the fact that non-HELIOS applications view HELIOS raster PDF documents as generic PDF documents which do not contain Photoshop resource block data. Hence many features that require Photoshop resource blocks are not supported in HELIOS raster PDF documents:

- Spot colors can be identified by their name string only, identification of a spot color as a Photoshop spot color table entry is not available
- No clipping path support
- No alpha and mask channels
- No transparency channels
- No multiple layers
- IPTC is not supported

If HELIOS raster PDF supports a specific color space, an image with this color space can also have spot colors. The only way in the PDF versions up to 1.7 to combine arbitrary color space images with spot colors is the use of overprints. Overprints are a device-dependent feature by definition of all PDF versions up to 1.7. Note that the term “device-dependent” in prepress production should read “software-dependent” today. Applications often do not support spot colors or do not support overprints. Using “HELIOS raster PDF” documents with spot colors in these applications often results in partial or complete loss of spot color information.

E.3 Creating HELIOS raster PDF documents

Images can be converted to HELIOS raster PDF using the “layout” command line tool in convert mode:

```
$ layout -v -T PDF -l rabbit.tif rabbit.pdf
```

Note: Photoshop resource block information such as a clipping path is lost during a conversion to HELIOS raster PDF.

E.4 Using HELIOS raster PDF as OPI layouts

The layout command line tool can generate layouts in HELIOS raster PDF format with OPI references, too:

```
$ layout -v -o RasterImageType="PDF " rabbit.tif
```

The resulting PDF layouts can be used in InDesign for OPI workflows.

E.5 Embedded and referenced ICC profiles

The PDF standard supports embedded ICC profiles only. ImageServer supports both embedded and referenced ICC profiles in HELIOS raster PDF documents. HELIOS raster PDF documents with referenced ICC profiles are an extension of the PDF standard. Consequently, non-HELIOS applications view HELIOS raster PDF documents with referenced ICC profiles as untagged PDF documents.

E.5.1 Rendering intent in PDF documents

The ICC standard defines four rendering intents for viewing or printing color images:

- Perceptual (0)
- Relative Colorimetric (1)
- Saturation (2)
- Absolute Colorimetric (3)

PDF documents can contain rendering intent information, unlike images in any other common raster image format. If PDF documents with rendering intent information are viewed in applications such as Adobe Acrobat 8, then the appearance of these documents depends on their rendering intent information. ImageServer supports the specification of rendering intents for the HELIOS raster PDF format during document creation.

The PDF standard defines that rendering intent information applies to CIE-based color spaces only. This means for HELIOS raster PDF documents that rendering intent information can be specified for images with color space CIE Lab and for images tagged with an embedded ICC profile. ICC profile references are not supported, because they are not part of the PDF standard. The rendering intent of images with a color space suitable for rendering intents can be specified using the “PDFIntent” attribute of the layout command line tool:

```
$ layout -o Attributes PDFIntent=<intent> -T PDF -l input.tif output.pdf
```

The value of `<intent>` can be `0`, `1`, `2` or `3`. The ImageServer values 4, 5, and 6 for rendering intents with BPC (*Black Point Compensation*) are not allowed here, because they are neither part of the ICC standard nor part of the PDF standard.

F Notes for PCShare users

If `Create Layouts` is active in a specific volume, ImageServer will create layouts of all image files, regardless of whether they come from a Mac or PC client. It generates layout files always according to the `PC Layouts` checkbox in the HELIOS Admin `Volume` settings. Thus, if you want to have PC-compatible layouts of all images (e.g. for working in a volume shared with PCShare) – even if they are coming from a Mac – you have to activate the `PC Layouts` option for this specific volume (see **PC Layouts** in 4.2 “Volume settings”).

UTF-8 support

PCShare support includes UTF-8 volume character sets, long file names, and file locking. ImageServer does not make any precautions for sticking to the *8dot3* file name syntax when creating layout files, etc. Files with long file names will not be seen anymore by older PCShare clients.

Important: “opisrv” does not automatically create a layout image if a PCShare-based file is renamed and the new file name does not end with one of the known image file suffixes!

“Close option” flag

If you want to use image files cross-platform and share volumes between Mac and Windows clients, we recommend to set the `Close option` flag in the HELIOS Admin `Volumes` dialog. The `Close option` guarantees that PCShare always closes a file completely to allow a Mac client to operate on consistent data.

OPI printing to Windows PostScript printers (via SMB/CIFS)

The advantage of the HELIOS “Windows Printer” support is that the OPI server will do the OPI processing *on-the-fly* during output, e.g.:

- Image replacement
- ColorSync/ICC separation/proof
- Scaling/Cropping
- Output optimization

G Glossary

ADSC

Adobe Document Structuring Conventions (ADSC) provide directions for the structuring of PostScript files.

AFP server

Apple file server using the *Apple Filing Protocol* (AFP).

Alpha channels

The high-order 8 bits of a 32-bit graphics pixel used to manipulate the remaining 24 bits for purposes of coloring or masking. Also called *Additional channels*.

Bilevel

Bilevel images contain only two colors: black and white. “Lineart” is one example of bilevel images.

BMP

Acronym for *Bitmap*. A simple graphics file format developed by Microsoft for its Windows operating system. BMP files can store graphics from 1-bit (2-color) up to 24-bit (16.7 millions of colors). But since the BMP format does not support any method of compression, images may consume considerable space on your hard drive.

BPC

Acronym for *Black Point Compensation*. A software setting that maps the black point of the source profile to the black point of the destination profile.

CCITT

Comité Consultatif International Téléphonique et Télégraphique, renamed to *ITU-T* in 1993. This group specified communication protocols for a particular class of devices (e.g. facsimile devices) and represents specific modes of compression (CCITT Group 3, CCITT Group 4).

CIE

The *Commission Internationale de l'Eclairage* (CIE) is responsible for the definition of color models and the standardization of color descriptions. The Lab color space has been defined by the CIE.

CIELab

CIELab colors are defined by the *L-value* for lightness and the coordinates *a* and *b* defining the quantitative distance of a color from a reference white point. The Lab color space includes all visible colors and is device independent.

Clipping paths

A clipping path is a mask you can apply to an image if you want to cut out a specific part. The clipping path may be a rectangle or an ellipse; some applications also allow defining an irregular clipping path.

CMYK

Color printing is based on the CMYK color space. *Cyan*, *Magenta*, *Yellow*, and *Black* inks are mixed on paper to produce a given color. The definitions of the basic colors Cyan, Magenta, Yellow, and Black are slightly different in Europe (**Euroscale**), the US (**SWOP**), and Japan.

Color matching

The process of adjusting or matching colors appropriately to achieve maximum similarity from the gamut of one color space to the other. In practice, the color data delivered by a given device – e.g. a scanner – have to be transformed so that the colors can exactly be reproduced by a second device – e.g. a printer.

Color mode

See: **Bilevel**, **Grayscale**, **Indexed colors**, **RGB**, **CMYK**, **CIELab**, **Spot color**, **Multichannel**, **Multitones**, and **YCbCr, YCbCrK**.

ColorSync

ColorSync offers a programming interface to a fast computing engine which uses ICC profiles as parameters to perform color transformations between different devices. Apple ColorSync 2 has been co-developed by Apple Computer and Linotype-Hell.

Compress

Compress is a lossless compression mode that can be used for different color modes and file formats. It collects repetitive patterns in a table and saves references to this table whenever possible.

Compression

See: **Compress**, **CCITT**, **Flate**, **JPEG**, **JBIG2**, **PackBits RLE**, **Wavelet**, and **Zip**.

DCS 1, DCS 2

The *Desktop Color Separations* file format (DCS) has been developed by Quark, Inc. and is based on the standard EPSF format. DCS 1 images are composed of five files, namely the preview and the EPSF separation plates for Cyan, Magenta, Yellow, and Black. DCS 2 is a single-file format that includes the preview and the separation plate information.

DCT

Discrete Cosine Transformation is a technique for representing waveform data as a weighted sum of cosines. DCT is commonly used for data compression, as in JPEG. This usage of DCT results in lossy compression. DCT itself does not lose data; rather, data compression technologies that rely on DCT approximate some of the coefficients to reduce the amount of data.

DeviceLink profile

The *International Color Consortium* (ICC) has defined ICC profiles which describe the color characteristics of graphics devices (such as scanners, monitors,

and printers) and of abstract color spaces (such as CIELab D50, CIELab D65, and CIELuv). Additionally, the ICC has defined DeviceLink ICC profiles, which describe mappings from one particular color space to another particular color space, e.g. from an offset press CMYK color space to a gravure press CMYK color space. Many DeviceLink ICC profiles maintain special characteristics of the black channel and fulfill special conditions for color sums according to the drying capabilities of a particular printing process.

DeviceN

DeviceN is the PostScript 3 device-dependent instance of a multi-component color space. This type of color space allows for the specification of color components other than the standard set of three (RGB) or four (CMYK) color components usually used by most applications.

Downsampling

The process of transforming a high-resolution image into a low-resolution image.

dpi

See: **Resolution**

EPSF

The *Encapsulated PostScript File* (EPSF) format is meant for pictures that are to be used in different applications or on different platforms. EPSF files contain a text file that lists the PostScript instructions necessary to create the picture and, in addition to that, may contain a PICT preview of the image. If an EPSF file is created by – or exported from – an illustration or DTP application (e.g. InDesign, QuarkXPress) this file contains object-based PostScript instructions and can only be placed in other documents; it cannot be reloaded or edited again. This is only possible, if you are using an image processing application like Photoshop which is able to create raster-based EPSF files. Please note that EPSF files and PC-EPSF files (for Windows computers) are not identical. PC-EPSF files contain compressed TIFF previews (instead of PICT previews) and, therefore, behave differently in specific situations.

Euroscale

Euroscale defines the European ink set for the process colors Cyan, Magenta, Yellow, and Black.

File format

See: **BMP**, **DCS 1, DCS 2**, **EPSF**, **ICS**, **JPEG**, **JPEG 2000**, **JBIG2**, **PDF**, **Photoshop**, **PICT**, **PNG**, **Scitex CT**, **TIFF**.

Flate

Flate (also called Zip) is a compression method that works well on images with large areas of single colors or repeating patterns. The Flate method is lossless, which means it does not remove data to reduce file size and so does not affect an image's quality.

Gamut, gamut mapping

The gamut is the range of color that a given device can produce. Gamut mapping means re-defining the colors of a given device so that its gamut becomes (approximately) identical to that of a second device.

GFX

GFX describes the internal HELIOS data structure. It may contain color space, compression method, plate information, etc. of each object in a file.

Grayscale

Grayscale images are a generalization of Bilevel images. They contain black and white and different shades of gray.

Home directory

Private directory provided for each UNIX user. The home directory is the current directory as soon as you login to a server via Terminal.

ICC, ICC profiles

The *International Color Consortium* (ICC) is a group of vendors who defined the ICC-profile format. This format is a cross-platform specification which

allows third party vendors to develop profile tools and applications supporting the standard. The founding members of this consortium include: Adobe Systems Inc., Agfa-Gevaert N.V., Apple Computer, Inc., Eastman Kodak Company, FOGRA (Honorary), Microsoft Corporation, Silicon Graphics, Inc., Sun Microsystems, Inc., and Taligent, Inc. These companies have committed themselves to fully support this specification in their operating systems, platforms and applications. (See also **Profiles**.)

ICS

ICS is a proprietary format that has been created by Linotype-Hell. It is a modification of the EPSF/DCS 1 format.

Indexed colors

Images with indexed colors use colors from a given RGB color map. Every pixel of such an image contains a reference to a specific color in the map.

JPEG

JPEG is a file format and – at the same time – a mode of compression. Images are compressed by replacing several similar colors by one color only. Thus, some color data is lost and cannot be recreated when the files are opened again. JPEG has been developed by the *Joint Photographic Experts Group* (JPEG).

JPEG 2000

A potential successor to JPEG with better compression and multiresolution images. JPEG 2000 gives reasonable quality down to 0.1 bits/pixel (JPEG quality drops dramatically below about 0.4 bits/pixel).

JBIG2

A compression standard for lossless bilevel image coding. Its lossless algorithm features compatible progressive coding and sequential coding – the image is unaltered after compression and decompression. JBIG2 has been developed by the *Joint Bi-level Image Experts Group*.

Multichannel

A Multichannel image contains different color modes – e.g. CMYK colors and several spot colors – at the same time.

Multitones

Colorized Grayscale

PackBits RLE

Apple Mac OS compression, mainly used for PICT files.

PDF

The *Portable Document Format* (PDF) is an Adobe Acrobat file format that has been created for application independent file exchange. With the Adobe Reader software – which can be downloaded from the Adobe website – you may read and print any given PDF document.

Photoshop

Photoshop is the native Adobe Photoshop application specific file format.

PICT

The PICT file format is the native classic Mac OS image format.

PNG

Acronym for *Portable Network Graphics*. Bitmapped graphics format conforming to the "Portable Network Graphics Tenth Specification" (Version 1.0). The PNG format is similar to GIF (*Graphics Interchange Format*). However, it does not use patented data compression and is license-free. The World Wide Web consortium approved it as a standard to replace GIF. Most browsers support the PNG format.

PPD

PostScript Printer Description (PPD), is a file format developed by Adobe Systems, Inc. PPD files contain information enabling software to produce the best results possible for each type of designated printer.

Profiles

A profile is a device description. It contains information about how a given device (scanner, monitor or printer) mixes and reproduces colors.

Rendering intent

The approach taken when a *Color Management Module* (CMM) maps or translates the colors of an image to the color gamut of a destination device. Each profile supports different rendering intents. Changing the rendering intent may lead to a different output result, even though you did not change the profile.

Resolution

The *dots per inch* (dpi) value of an image indicates its resolution. The resolution of a given device (e.g. scanner, printer) defines its resolution capacity. Very clear and sharp images require input/output devices with a high resolution (300 dpi or more). Monitor resolutions usually range from 72 to 200 dpi.

RGB

Screens and monitors produce colors by means of *Red*, *Green* and *Blue* light (RGB). The light intensities make up a given color. Scanners also work with RGB colors. They read the amounts of red, green, and blue light that are reflected from an image (or transmitted if you scan transparent images). RGB images contain three components per pixel, namely a specific amount of red, green, and blue.

RIP

A *Raster Image Processor* (RIP) performs the final calculation of the data which are sent to the output device. The RIP may be either an external unit or part of the output device itself. A PostScript laser printer for example contains its RIP.

RLE

Run-Length Encoding (RLE) is a mode of compression that saves repetitive patterns only once and adds the number of repetitions. RLE is a lossless compression.

Root directory

The top-most directory on a UNIX computer is called “root” directory. If you are logged in as “root”, you can access all other directories and subdirectories on the system.

Scitex CT

Scitex *Continuous Tone* (CT) is a proprietary image file format that has been developed by Scitex Corporation, Ltd.

Spooler

A spooler is a set of programs which manage print jobs. A spooler acts as a buffer for the files that have been sent to an output device.

Spot color

A spot color is an exactly defined full tone color. Spot colors are printed on their own separation plates when separations are specified.

SWOP

The *Specifications for Web Offset Publications* (SWOP) include US standards for color separation and color printing.

TCP

The protocols and services included in Apple’s network architecture. The access to the EtherShare file server and print server use TCP/IP.

TIFF

The *Tagged Image File Format* (TIFF) is a standard graphics file format for files that are to be exchanged among several applications and environments, including Mac, DOS/Windows, and UNIX.

Wavelet

A lossy compression method of an image. Compression is achieved by converting the image into a set of mathematical expressions. The **JPEG 2000** image format uses wavelet compression.

YCbCr, YCbCrK

The YCbCr color mode is used by the television industry for digital videos. There are transformation tables that allow transforming YCbCr into RGB and vice versa. YCbCrK is a JPEG-based format that has been developed by Adobe.

Zip

A lossless file compression method, which is based on the DEFLATE algorithm for compression (see **Flate**).

Index

Q