

Papyrus ImageConverter

Reference Guide

ISIS Papyrus Europe AG

Papyrus Platz 1

A-2345 Brunn/Gebirge

Phone: +43-2236-27551

E-Mail: info@isis-papyrus.com

Website: www.isis-papyrus.com

Product Support:

Phone: +43 2236 27551-111

E-Mail: support@isis-papyrus.com

© ISIS Papyrus Europe AG 2012 - 2024

All rights reserved.

Reproduction in any form, in whole or in part, without express written permission is prohibited.

picrge V7.9 SP2

Last modified: 2024/11/13

This document complies with the PDF/UA-1 standard (ISO 14289-1) and was created by the Papyrus Academy using the Papyrus Document System to ensure that people with disabilities can access PDF content.

Contents

About this document	4
1 Introduction	5
2 Prerequisites	6
2.1 Common prerequisites	6
2.2 Specific prerequisites	7
3 Installation	8
3.1 Installation with the installation routine	8
3.2 Upgrade	9
4 PSEPIC parameters	10
4.1 INPUT parameters	10
4.1.1 PPMFInputName	10
4.1.2 PICMaxLen	10
4.2 OUTPUT parameters	10
4.2.1 TargetName	10
4.2.2 PICOutImgType	12
4.2.3 PICTiffMultipage	12
4.2.4 PICGenTiffMultipage	13
4.2.5 PICBmpMode	13
4.2.6 PICPhotometric	13
4.2.7 PICPsgPrefix	13
4.2.8 PICRotation	14
4.2.9 PICImgMode	14
4.2.10 PICColorMode	15
4.2.11 PICCompression	16
4.2.12 PICResolution	17
4.2.13 PICDitherMet	18
4.2.14 PICDitherAlg	18
4.2.15 PICDitherMax	19
4.2.16 PICCuttingArea	19
4.2.17 PICIBMOgl	19
4.2.18 PICOglEncoded	20
4.2.19 PICOglBitmap	20
4.2.20 PICTransparency	20
4.2.21 PICRemoveNOPs	20
4.3 DESIGN parameters	21
4.3.1 PICWidth, picheight	22
4.3.2 PICMaintDim	23
4.3.3 PICBestZoom	23
4.3.4 PICCellOpt	24
5 Sample ICD file for PSEPIC	25
6 Papyrus ImageConverter tool in Papyrus Objects	26
6.1 Ready to use setup	26
6.2 Technical details - available types of image conversions	30
Glossary	33
Support and contact information	40

About this document

This document describes the parameters of the **Papyrus ImageConverter** and is a supplement to the *Papyrus Server General Installation and Administrator Guide (pseiae).pdf*.

This document is addressed to business users who want to use the conversion facilities as well as to administrators who install the program and perform settings and configurations.

This document contains the following chapters:

Introduction

Chapter [1 Introduction](#) informs about the supported image types both as input type and output type and contains specific hardware and software requirements.

Prerequisites

Chapter [2 Prerequisites](#) contains both common prerequisites that are required for the operation of all ISIS products as well as specific requirements only applicable for successful operation of the OverView Image Editor.

Installation

Chapter [3 Installation](#) refers to the installation with the Papyrus installation routine and to the upgrade to a newer product version.

PSEPIC Parameters

Chapter [4 PSEPIC parameters](#) gives an overview of all ImageConverter parameters and contains detailed information on application and possible values of each parameter.

Sample ICD File for PSEPIC

Chapter [5 Sample ICD file for PSEPIC](#) contains a sample ICD file with possible parameter value settings.

Papyrus ImageConverter Tool in Papyrus Objects

Chapter [6 Papyrus ImageConverter tool in Papyrus Objects](#) describes how the ImageConverter is implemented as tool/material method in a Papyrus Objects environment and how image objects can be converted.

1 Introduction

Papyrus ImageConverter allows users to convert image types. In addition, it also provides options for resizing images and for resampling images based on a defined output resolution.

Following image types are supported both as input and output type:

- BMP - Windows Bitmap
- TIFF
- JPEG
- GIF
- ICO - Icons
- PTR - Pointer
- AFP - AFP Page Segment
- PNG

Papyrus ImageConverter embeds meta data information like creator tool and creation date in the converted image. This information is available for JPEG, PNG and TIFF. Metadata are only available when the output image will be produced on W32 platforms. To manage metadata, Papyrus ImageConverter uses the open source library Adobe that is available for Windows only.

Papyrus ImageConverter is also able to convert multi-page TIFF images into single-page images of any supported image type and vice versa.

For further information on general ICD parameters, see *Papyrus Server General Installation and Administrator Guide (pseiae).pdf*.



2 Prerequisites

This section describes common and specific prerequisites for successfully operating the Papyrus product or Papyrus solution you have purchased. The prerequisites section should cover all measurements and actions to be taken except for the actual installation and configuration of the product.

Common prerequisites are prerequisites that are common to each and every Papyrus product without exception. Such prerequisites are for example, the availability of an ISIS Papyrus Product Authorization Service (IPAS) and a KeySet file that was delivered by Papyrus Software shipment.

Specific prerequisites are prerequisites that are individual to the specific Papyrus product or Papyrus solution.

2.1 Common prerequisites

Operating system

Some Papyrus products are available for several operating systems. For details on which operating systems and versions are supported by Papyrus Software, see *Operating systems* in *Software Installation and Product Authorization Installation and Administrator Guide (iinste)*. For details if this Papyrus product places certain requirements concerning the operating system, see the following section "Specific Requirements".

Hardware and system sizing

The appropriate hardware and system sizes are dependent on the application field of the product. For example, a product operating on a client machine requires a different hardware base than a server application or a plugin. It is recommended that the product should run in an up-to-date production environment. For an approximate overview of hardware dimensioning, see *Hardware prerequisites* in *Project Management Guidelines (pmguide)*. For details on system requirements and hardware dimensioning, see *System requirements and hardware estimates* in *WebRepository and WebControl Installation and Administrator Guide (poinste)*.

Papyrus root directories access rights

In the course of product installation, the Papyrus root directories `\isis` and `\isiscomm` are created. After installation, the system administrator should restrict access rights to these folders and their subfolders for security reasons. For more information, see *Read and write access to Papyrus root directories* in *Software Installation and Product Authorization Installation and Administrator Guide (iinste)*.

Product authorization

Papyrus products require authorization to operate.

ISIS Papyrus Product Authorization Service (IPAS) provides authorization for Papyrus products. IPAS is the standard program to grant product authorizations.

In order to put any Papyrus product into operation, IPAS has to be installed and the service has to be accessible. Communication with IPAS takes place via a TCP/IP-based network. Additional environment variables (ISIS_OMS_DOMAIN, ISIS_OMS_PORT, ISIS_KEY_MODE) need to be set on the local machine where the Papyrus product is operated. For details on how to install and configure IPAS and how product authorizations are granted to Papyrus products, see *ISIS Papyrus Product Authorization Service (IPAS)* in *Software Installation and Product Authorization Installation and Administrator Guide (iinste)*.

The product authorization itself is available within a KeySet. The KeySet must be installed in IPAS which then grants authorization to the requesting Papyrus products. A KeySet is a package of all product authorizations required to be able to run the Papyrus product. A KeySet is delivered by ISIS Papyrus Software Shipment in form of a KeySet file. Which specific KeySet you need in order to be able to use your product is explained in the following section "Specific Prerequisites"

Messages and tracing

Message analysis and tracing helps optimize processes, avoid bottleneck situations, and detect errors. Every Papyrus process issues messages that report the status of the process. Messages and the trace log are written into a file called the "PCS log file".

Tracing is a specialized use of logging to record information about a program's execution. Trace messages are not created by default. To output trace messages, you must first configure and enable a trace. For more details, see *Papyrus product messages* in *Product Messages Reference Guide (ipmrge)*.

In Papyrus Objects of a WebRepository or WebControl installation, the Application Performance Analyzer (APA) provides a graphical user interface for analyzing the messages and doing a tracing by System Administrators as well as Application Developers. Even though APA is designed for analyzing Papyrus processes running with or without Papyrus Objects, the running of the APA itself needs a standard installation of the Papyrus WebRepository. The communication between APA and the Papyrus processes is enabled by setting the following environment variables:

```
ISIS_PCS_APANET=1 //means APA is activated
ISIS_PCS_APANET_BROKER
```

If these variables are set, tracing can be done at any time without interrupting the processes to be analyzed. This is important when production situations should be observed because of issues.

For more details on APA, see *Papyrus application and performance analyzer* in *Monitoring Analyzing Troubleshooting and Tuning Administrator Guide (pmatage)*.

Default PDF reader

In order to view PDF files from within a Papyrus application, Adobe Acrobat Reader with the same register width as the Papyrus application has to be installed on the machine where the PDF is viewed. This Adobe Acrobat Reader has to be the system default for viewing PDFs. If a Papyrus application with PDF display such as Papyrus Desktop/EYE Widgets or Papyrus Designer is installed as 64-bit version, then also Adobe Acrobat Reader has to be a 64-bit version. The same applies to 32-bit variants. If the installed Adobe Acrobat Reader does not correspond to the Papyrus application's register width, error messages like "Failed to load plugin nppdf32.dll" will be displayed when attempting to view a PDF document.

TCP/IP-based network

For communication with IPAS a TCP/IP-based network has to be set up. If IPAS and Papyrus products run on the same machine, a TCP/IP-based network is not required for authorization purposes. But, as most Papyrus products are run in the context of a Papyrus WebRepository installation, which means a distributed multi-machine and multi-node environment, a TCP/IP-based network is required anyway.

2.2 Specific prerequisites

Product authorization

Papyrus ImageConverter requires a Papyrus ImageConverter product authorization.

3 Installation

The following chapter describes how to install Papyrus Image Converter with the Papyrus Installation Routine and how to upgrade to a newer product version.

3.1 Installation with the installation routine

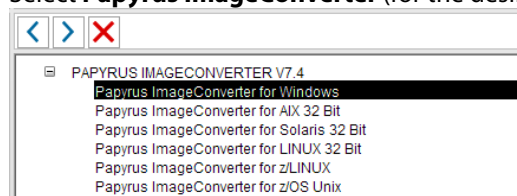
To install Papyrus ImageConverter, proceed as follows:

- (1) Insert the Papyrus Software DVD. The Papyrus Installation Routine starts automatically and guides you through the installation process.

Note: You can start the installation routine also manually by opening the directory `\install` and starting the file `setup.exe`.

For more details on the installation routine, see document *Software Installation and Product Authorization Installation and Administrator Guide (iinste)*.

- (2) Select **Papyrus ImageConverter** (for the desired platform) in the product selection panel:



- (3) Ensure that the required product authorization is installed on your system. For more details, see [2.2 Specific prerequisites](#).

After successful installation, the application is installed - for example for Windows - under the directory `... \psew3<vvv> \pic (<vvv> stands for the version number, for example 740)`.

z/OS and codepages

On z/OS all Papyrus modules are compiled with EBCDIC codepage 273. Thus, all scripts and configuration files on z/OS have to be in codepage 273 as well. Existing configuration files like `pcscfgv<vv>.cfg` already contain a hint to use the correct codepage.

If you are using a shell that runs with a different codepage (e.g. 1047) and have a script or configuration file that uses any of the non-alphanumeric characters in the following figure, use the `iconv` utility from IBM to convert its codepage to 273. In order to pass a parameter with one of these characters via command line, we recommend that you create a script.

ß ä Ä ö Ö ü Ü { } [] ! ~ \$ % & ' () * + , - . / : ; < > [\] ^ _ ` { | } ~

Characters with different code in EBCDIC 273 and 1047

Technical background: Some non-alphanumeric characters have different codes in different EBCDIC codepages. For instance `{` has the number 192 (00xC0) in EBCDIC 1047 and 67 (00x43) in EBCDIC 273. A regular expression or a GUID passed in curly brackets is not recognizable to the system when a "wrong" character is passed.

3.2 Upgrade

The procedure for upgrading Papyrus products follows the same principle on all platforms and is described in *Upgrading Papyrus installations* in *Software Installation and Product Authorization Installation and Administrator Guide (iinste)*.

4 PSEPIC parameters

On Windows, each parameter can be specified in uppercase or lowercase characters. On Unix and Linux platforms, parameters, path specifications, and file names are case-sensitive.

Some parameters are initialized with default values. The application checks all parameter values to verify, if they are compatible with each other.

For more details, see the subsequent parameter descriptions.

4.1 INPUT parameters

4.1.1 PPMFInputName

PPMFInputName = "[<drive>:]<path><filename>[.<ext>]"

This parameter describes the input file(s) to be converted. If the path is not specified, the image will be searched in the current directory. It is possible to specify a single file name or a list of file names separated by blanks. It is also possible to use wildcards "*" (asterisk) to specify all images contained in a directory, or "?" (question mark) to specify all images in a directory, whose file names are different in one character.

Windows:

PPMFInputName = C:\ISIS\in\A*.*

Unix:

PPMFInputName = /ISIS/in/A*

It is **not** possible to specify "*" and single file names at the same time, for example:

C:\ISIS\pseg* s1image.240

4.1.2 PICMaxLen

PICMaxLen = "<number>"

It is possible to specify as an input parameter the maximum length of the output file name. The specified length is considered as file name without extension. If not specified or 0, the maximum file name length is set to 8+3 (name + extension).

Note: If the output file name should not be defined, use the wildcard "*".

4.2 OUTPUT parameters

4.2.1 TargetName

TargetName = "[<drive>:]<path><filename>[.<ext>]"

This parameter describes the output file(s). The parameter PICMaxLen is only intended, when the output file name is "*". In this case the length of the output file name is set to the value specified for the parameter PICMaxLen (if not specified, the default value is 8+3). If the path is omitted, the PSEG(s) is/are generated in the current directory. If the extension is omitted, the default is PSG. If the output file already exists, it will not be replaced but renamed with the extension .bak.

Note: It is possible to use wildcard "*" (asterisk). In this case the output file name is equal to the input file name.

Example: See the following sample set of PPMFInputName and TargetName parameter combinations for Windows and Unix-based operating systems:

1. Windows:

```
PPMFInputName = C:\ISIS\in\A*.*
TargetName = C:\ISIS\out\A*.300
```

Unix:

```
PPMFInputName = /ISIS/in/A*.*
TargetName = /ISIS/out/A*.300
```

Every image file with "A" (or "a" but only under Windows because of its case insensitivity!) as its first letter will be taken as input file and converted to an output file of the same name plus extension .300. For example, for input file A001.bmp an output file A001.300 would be created.

2. Windows:

```
PPMFInputName = C:\ISIS\in\<file name>.<ext>
TargetName = C:\ISIS\out\*
```

Unix:

```
PPMFInputName = /ISIS/in/<file name>.<ext>
TargetName = /ISIS/out/*
```

Every image file will be taken as input file and converted to an output file using the same file name plus extension .PSG. For example, for input file ABC.psg an output file ABC.psg would be created.

3. Windows:

```
PPMFInputName = C:\ISIS\in\*.*
TargetName = C:\ISIS\out\*.300
```

Unix:

```
PPMFInputName = /ISIS/in/*.*
TargetName = /ISIS/out/*.300
```

Every image file will be taken as input file and converted to an output file using the same file name plus extension .300. For example, for input file ABC.bmp an output file ABC.300 would be created.

4. Windows:

```
PPMFInputName = C:\ISIS\in\*.tif
TargetName = C:\ISIS\out\*.300
```

Unix:

```
PPMFInputName = /ISIS/in/*.tif
TargetName = /ISIS/out/*.300
```

Every image file with extension .tif will be taken as input file and converted to an output file using the same file name plus extension .300. For example, for input file ABC.tif an output file ABC.300 would be created.

4.2.2 PICOutImgType

PICOutImgType = "{ 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 }" or "{ AFP | BMP | GIF | JPG | TIF | ICO | PTR | PNG }"

This parameter specifies, which output image types are allowed.

Value	Image format
1	AFP (=PSEG)
2	BMP
3	GIF
4	JPG
5	TIF
6	ICO
7	PTR
8	PNG

Samples:

PICOutImgType="1"

PICOutImgType="PTR"

Note: This is a mandatory parameter. It is required to specify the output image type.

4.2.3 PICTiffMultipage

PICTiffMultipage = "{ 0 | 1 }"

This parameter defines whether to convert a multi-page TIFF file into a series of single-page image files.

The parameter only takes effect if the input file is a multipage TIFF format and each of the images contained is to be converted.

Any supported output format, i.e. AFP, BMP, GIF, JPG, ICO, PTR, or PNG, can be created, depending on the setting of parameter **PICOutImgType**.

Values	Description
1	Converts a multi-page TIFF input file into a series of single-page files.
0	The Tool is not able to create single-page files from multi-page TIFF.

Default: 0

4.2.4 PICGenTiffMultipage

PICGenTiffMultipage = "{ 0 | 1 }"

This parameter defines whether to create a multi-page TIFF file from a series of single-page image files of type AFP, BMP, GIF, JPG, ICO, PTR or PNG. The input files can be mixed format types - for example, the input images include BMP, GIF and PNG.

If the input consists of one or several multi-page TIFF file(s), only the first page of each TIFF file will be converted.

To generate a TIFF requires that parameter **PICOutImType** is set to "5".

Values	Description
1	Creates a multi-page TIFF file from a series of (single-page) files. If one or more multi-page TIFF files used as input, only the first page of each file will be converted.
0	The tool is not able to create multi-page TIFF files.

Default: 0

4.2.5 PICBmpMode

PICBmpMode = "{ 0 | 1 }"

This parameter specifies the bitmap image type: 0=Windows, 1=OS2

4.2.6 PICPhotometric

PICPhotometric = "{ 0 | 1 }"

When you save TIFF images, you can select the photometric interpretation. This parameter is only relevant for bilevel and grayscale TIFF images.

Value	Description
0	Creates a TIFF with photometric interpretation "White is Zero".
1	Creates a TIFF with photometric interpretation "Black is Zero".

Default: 1

Important note: To create a TIFFG4 DLL compatible TIFF it is mandatory to use

- PICPhotometric= "0" ("White is Zero") and
- PICCompression = "2" (CCITT fax4)

4.2.7 PICPsgPrefix

PICPsgPrefix = "{ 0 | 1 }"

This parameter puts the prefix "S1" to the output file name if set to "1". If it is specified, the file name length is limited to 8 digits including the prefix. The prefix is not inserted by default.

Default: 1

4.2.8 PICRotation

PICRotation = "{ 0 | 90 | 180 | 270 | <xx> }"

This parameter specifies the rotation of the output image. Following values are possible:

Values	Degree
0	0 degrees
90	90 degrees
180	180 degrees
270	270 degrees
<xx>	It is possible to specify a non-standard rotation angle

Default: 0

4.2.9 PICImgMode

PICImgMode = "{ 0 | 1 | 2 | 3 | 4 | 5 }"

This parameter specifies the image mode of the output image.

The parameter only takes effect if **PICOutImgType** is set to "AFP".

Following values are possible:

Values	Image mode
0	IM1
1	GOCA RASTER
2	IOCA FS10
3	IOCA FS11
4	IOCA FS42
5	IOCA FS45

Default: 0

Note: IOCA FS11 only allows the JPEG compression and RGB color. Other compression types "no compression", "MMR" and "G4MMR" are only possible with OCA colors (base and extended).

4.2.10 PICColorMode

PICColorMode = "{ 0 | 1 | 2 | 3 }"

This parameter specifies the color mode of the output image. Following values are possible:

Values	Color mode
0	OCA
1	OCA EXTENDED
2	RGB
3	CMYK

Default: 0

Please note that this parameter follows these rules:

Colors	IM	GOCA				IOCA			
		OCA	RGB	CMYK	Highlight	FS10	FS11	FS42	FS45
B/W	X	X	X	X		X	X	X	X
Default OCA	X	X	X	X		X	X	X	X
Extended OCA	X	X	X	X					X
Highlight					X				
<= 1000 colors			X	X			X		X
True Color							X		X

Note: FS10 is the standard format for IOCA Black and White and/or OCA colors

4.2.11 PICCompression

PICCompression = "{ 0 | 1 | 2 | 3 | 4 }" (for PSEG, depending on image mode)

PICCompression = "{ 0 | 1 | 2 | 3 | 4 | 5 }" (for TIFF, depending on color mode (B/W or color TIFF))

This parameter specifies the compression of the output image.

The parameter only takes effect if parameter **PICOutImgType** is set either to "AFP" or "TIF".

Default: 0

If the output image type is a **PSEG** (PICOutImgType="1" or "AFP"), following values are possible:

Values	Image compression
0	NONE
1	MMR
2	G4MMR
3	TIFF
4	JPEG

Please note that this parameter follows these rules:

- IM1 images: no compression
- IOCA FS10 images: no compression + MMR + G4MMR
- IOCA FS11 images: no compression + MMR + G4MMR (requires OCA colors, RGB colors require JPEG compression)
- IOCA FS42 images: no compression + MMR + G4MMR
- IOCA FS45 images: no compression + TIFF + JPEG (It is not possible to use IOCA FS45 JPEG compression with RGB colors!)
- GOCA images: no compression

If the output image type is **TIFF** (PICOutImgType="5" or "TIF"), the following values are possible:

Values	Image compression
0	NONE
1	CCITT fax3
2	CCITT fax4
3	LZW
4	JPEG
5	PACKBITS

Please note that this parameter follows these rules:

For black and white TIFF

- NONE
- CCITT fax3
- CCITT fax4
- LZW
- JPEG
- PACKBITS

For multi color TIFF

- NONE
- LZW

4.2.12 PICResolution

PICResolution = "{ 240 | 300 | 600 }"

This parameter specifies the resolution of the output image. It is possible to specify any number.

Default: 240

Related parameter: PICMaintDim

Note: If the defined output resolution differs from the input image's resolution, the input image will either be resampled or resized depending on the value set for **PICMaintDim**.

4.2.13 PICDitherMet

PICDitherMet = "{ 0 | 1 | 2 | 3 | 4 | 5 }"

This parameter specifies the method to be used to dither the output image. The following values are possible:

Values	Description
0	No dither
1	Black and White
2	Gray scale
3	IBM OCA Colors
4	IBM OCA Extended Colors
5	The image will be dithered to the number of colors defined with the parameter PICDitherMax (for example "1000").

Default: 0

Related parameter: PICDitherMax

Note: If an input image contains transparency dithering should not be used. Otherwise transparency gets lost.

4.2.14 PICDitherAlg

PICDitherAlg = "{ 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 }"

This parameter specifies the algorithm to be used to dither the output image. The following values are possible:

Values	Description
0	Nearest color
1	Horizontal
2	Floyd
3	Jarvis
4	Stucki
5	Stevenson
6	Burkes
7	Sierra
8	Random Error Diffusion

Default: 0

4.2.15 PICDitherMax

PICDitherMax = "<number>"

This parameter specifies the maximum number of colors that can be used to dither the output image. The range of possible values is from 0 to n ('n' means any number)

Default: 1024

Related parameter: PICDitherMet

Note: This parameter only takes effect if parameter **PICDitherMet** is set to 5.

In case of dithering, compression types increasing the number of colors - for example, JPEG Compression (PICCompression=4) - should not be used.

4.2.16 PICCuttingArea

PICCuttingArea = "<res>, <x>, <y>, <wid>, <hei>, <R>,<G>,"

This parameter specifies all the characteristics of the area to be cut from the original image; the area will be filled with the specified RGB color.

Values	Description
res	specifies the resolution used as base (the area coordinates are calculated with this resolution).
x	the x coordinate of area upper left corner (in PEL according to base resolution)
y	the y coordinate of area upper left corner (in PEL according to base resolution)
wid	width of the area (in PEL according to base resolution)
hei	height of the area (in PEL according to base resolution)
R,G,B	RGB value of the color to be used (range 0 ... 255). The default value is white.

If the parameter is omitted, no cutting will be performed. Every value has to be specified, otherwise the message PPIC0204E; "Bad cutting parameter format." is displayed.

Sample: PICCuttingArea = "300,50,50,419,207,255,138,0"

4.2.17 PICIBMOgl

PICIBMOgl = "{ 0 | 1 }"

This parameter enables the creation of IBM OGL/370 pattern from any image type.

Default: 0

Note: The image will be automatically dithered into B/W before the pattern generation takes place. It is recommended to specify the dither algorithm in the ICD file, otherwise the default value will be used (0=nearest color).

4.2.18 PICOglEncoded

PICOglEncoded = "{ 0 | 1 }"

When this parameter is set to "1", an image with encoded pattern can be created, otherwise it is created with pels ("0").

Default: 0

4.2.19 PICOglBitmap

PICOglBitmap = "{ 0 | 1 }"

If set to "1", this parameter creates a bitmap without OGL syntax.

Note: If **PICOglEncoded** and **PICOglBitmap** are activated, only **PICOglEncoded** will be considered and the output will be an encoded pattern.

4.2.20 PICTransparency

PicTransparency = "1, <R>, <G>, , 0" (for RGB colors)

PicTransparency = "2, <C>, <M>, <Y>, <K>" (for CMYK colors)

With this new parameter you can create transparent images from any image type. It is only possible to save these images with IOCA FS45.

The parameter will be defined by the parameter name, followed by the color to be considered as transparency color. The first parameter value defines the color mode (1 ... RGB, 2 ... CMYK) and the color (R,G,B or C,M,Y,K) that should be converted to transparency:

Samples:

PICTransparency="1,255,255,255,0" ... RGB color white

PICTransparency="2,0,0,0,0" ... CMYK color white

Note: Transparency can only be used for output image type PSG (requires to select IOCA and Function Set 45), GIF, and PNG.

4.2.21 PICRemoveNOPs

PICRemoveNOPs = "{ 0 | 1 }"

This parameter defines whether to remove user-defined NOPs (No Operation Records) in the generated PSEG.

Values	Description
1	Does remove NOP's
0	Does not remove NOP's

Default: 0

Note: This function can only be used when the input and the output images are PSEGs.

Example:

The input image (listed) has for instance the following NOPs:

```
NOP(00000013/0000001) : 'OverView Image Editor V.7.01/w3 ('2010-03-15
(build:7.01.1.10110)') Gen.:Fri Mar 19 15:20:57 2010'

'.@.@.@.K.K..a..@M}....'..'..@M.....z.K..K.K.....]]@...
.Kz...@...@...@...z...z...@....'

NOP(0000007f/0000002) : 'Creation Date: Fri Mar 19 15:20:57 2010'
'.@.....z@...@...@...@...z...z...@....'
NOP(000000b1/0000003) : 'Resolution: 3000-3000, Size: 72x36 pel'
'.@.....z@...@...@...@...z...z...@....'
NOP(000000e2/0000004) : 'Image type: IOCA FS10, Color type: OCA,
Color Number: 2'
'.@.....z@...@...@...@...z@...@...@...@...z@...@...@...z@...'
NOP(00000124/0000005) : 'This is the user information!.'
```

With PICRemoveNOPs="1" all NOPs are removed.

Setting PICRemoveNOPs="0" or leaving this parameter undefined, the generated PSEG has the NOP of its creator program Papyrus ImageConverter:

```
NOP(00000013/0000001) : 'ISIS IMAGE Converter V.7.01/w3 ('2010-03-18
(build:7.01.1.10112)') Gen.:Wed Mar 24 10:59:50 2010'

'.@.....@.....@.....@.K.K..a..@M}....'..'..@M.....z.K..K.K.....]]@...
.Kz...@...@...@...z...z...@....'
```

4.3 DESIGN parameters

4.3.1 PICWidth, picheight

PICWidth, PICHeight = "<image dimensions>"

With these parameters, you can specify new image dimensions. You have the following possibilities to define new dimensions:

- **pels:** Enter image values in pels for both parameters
- **auto:** Enter the keyword 'auto' for one of the two parameters, for the other enter a value in pels. In this case, the 'auto' dimension is automatically calculated in order to maintain the image aspect ratio.
- **percentage:** Express dimensions as percentage 'p' or '%' of the input image.
- **thumbnail:** For one parameter, enter the keyword 'thumbnail'. For the second parameter, enter a value in pels to specify the dimensions of the thumbnail image.

Example:

PICWidth (as well as **PICHeight**) can assume one of the following sample values:

Values	Description
220	The output image width is set to 220 pels
p200 or 200%	The output image width is calculated as 200% of the input image
auto	The output width is calculated in order to maintain the image aspect ratio; the other value must be set to a valid non-zero value.
thumbnail	See Thumbnail Examples below

Thumbnail Examples:

PICHeight = "thumbnail", PICWidth = "100" --> generates a thumbnail image of size 100x100 pels

PICHeight = "75", PICWidth = "thumbnail" --> generates a thumbnail image of size 75x75 pels

Note: ImageConverter supports a maximum width of 7000 pels and a maximum height of 9900 pels for output images. Otherwise the image conversion fails and the error message PPIC0416E: "The output image is too large to be supported" is returned.

4.3.2 PICMaintDim

PICMaintDim = "{ 0 | 1 }"

This parameter allows to maintain the input image's physical dimensions (i.e. its print size in mm or inches) during resampling operations based on a change of resolution.

Value	Description
0	Do not maintain physical dimensions.
1	Maintain pyhsical dimensions.

Default: 1

The following table gives an overview of conversion results with various parameter combinations for **PICResolution**, **PICWidth/PICHeight** and **PICMaintDim**. The results shown are based on the use of an input image with 100 pels width, 100 pels height and a resolution set to 300 ppi:

PICResolution	PICWidth/PICHeight	PICMaintDim	result in PEL(width/height)
-	p200/auto	0	200/200
-	200/200	0	200/200
600	-	0	100/100
600	p200/auto	0	200/200
600	200/200	0	200/200
-	p200/auto	1	error message*
-	200/200	1	error message*
600	-	1	200/200

Note: It is not possible to maintain the physical image dimensions (**PicMaintDim="1"**) and define **PICWidth/PICHeight** at the same time. This leads to the following error message:

PPIC0114E:"Conflict between specified Image Dimensions and Maintain Image Dimensions parameter".

4.3.3 PICBestZoom

PICBestZoom = "{ 0 | 1 }"

With this parameter, an improved functionality in rescaling and resizing operations can be applied, resulting in a higher image quality.

In rescaling and resizing image operations, an enhanced zoom routine can be used, which applies a more efficient algorithm that improves the resulting image quality. Unlike the default function, it does not simply indiscriminately delete or add pixels in resizing and rescaling operations, but it adds some new pixels.

Note: In combination with proportional values of the parameters **PICWidth** and **PICHeight**, the number of colors increases.

Value	Description
0	Do not use enhanced zoom functionality.
1	Use enhanced zoom functionality.

Default: 1

4.3.4 PICCellOpt

PICCellOpt = "{ 0 | 1 }"

This parameter enables users to optimize the size of the image. This is a method to compress IM1 images. It means, that only valorized pixels are saved to reduce the size of the IM1 image as much as possible.

Value	Description
0	Do not use IM1 cell optimization.
1	Use IM1 cell optimization.

Default: 1

5 Sample ICD file for PSEPIC

The following examples show the parameters for the PSEPIC section only. For a description of the general parameters, see *Papyrus Server General Installation and Administrator Guide (pseiae).pdf*.

[PPMF]

```
PPMFInputName ="C:\isis\IN_PIC\sample.gif"
TargetName   ="C:\isis\OUT_PIC\pseg.300"
```

[PIC]

```
PICOutimgType ="1"
PICPsgPrefix  ="1"
PICImgMode    ="0"
PICColorMode  ="0"
PICCompression="0"
PICRotation   =" "
PICResolution ="240"
PICWidth      ="p50"
PICHeight     ="auto"
PICMaintDim   ="1"
PICBestZoom   ="1"
PICCellOpt    ="1"
PICCuttingArea="300,50,50,419,207,255,138,0"
PICIBMOgl     ="0"
PICOglEncoded ="1"
PICOglBitmap  ="1"
PICTransparency="1,255,255,255,0"
PICDitherMet   ="0"
PICDitherAlg   ="0"
PICDitherMax   ="1024"
PICMaxLen      =" "
PICTiffMultipage="1"
PICGenTiffMultipage=" "
PICBmpMode     =" "
PICRemoveNOPs  =" "
PICPhotometric ="0"
```

6 Papyrus ImageConverter tool in Papyrus Objects

The ImageConverter tool can be used to convert images in combination with the ImageConverter material. The internal names of their attributes are aligned to the ICD parameters used by the file-based Papyrus ImageConverter.

The ImageConverter tool is an instance of class **Repository|ISIS TOOL|PRINTER|IMAGE CONVERTER** and the ImageConverter material of **Repository|MATERIAL|CONVERT IMAGE**.

Important note for working with different nodes

If the input/output image is on a different node than the worker node where the ImageConverter tool is located, a Proxy Management definition must push them there.

To perform an image conversion, ensure that the material for the conversion step is in state "Ready". Then drag the material onto the ImageConverter tool and select **Start ...** in the context menu.

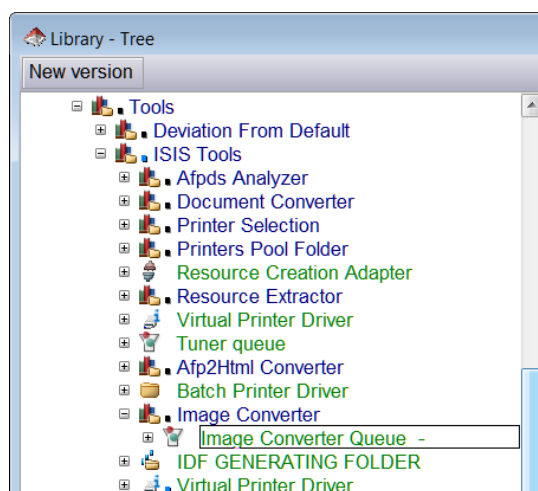
To enable the ImageConverter tool to update an existing image or generate a new version, ensure that the **Allow Update/New Version** option of the material used for the conversion is checked (related parameter "PicAllowUpdateNewVersion" of **Start** method in material class **CONVERT IMAGE**).

Attributes of the ImageConverter

The attributes of the ImageConverter tool and material have the same internal name as their corresponding ICD parameters and provide the same functionality as described in chapter [4 PSEPIC parameters](#).

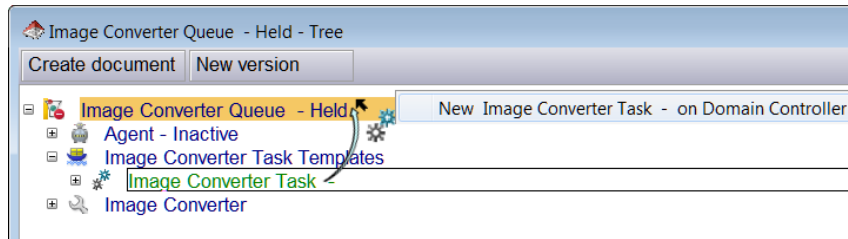
6.1 Ready to use setup

- (1) Instantiate the queue template **Image Converter Queue** in library folder **Library|Generic Frameworks|Tools|ISIS Tools|Image Converter** on a node.



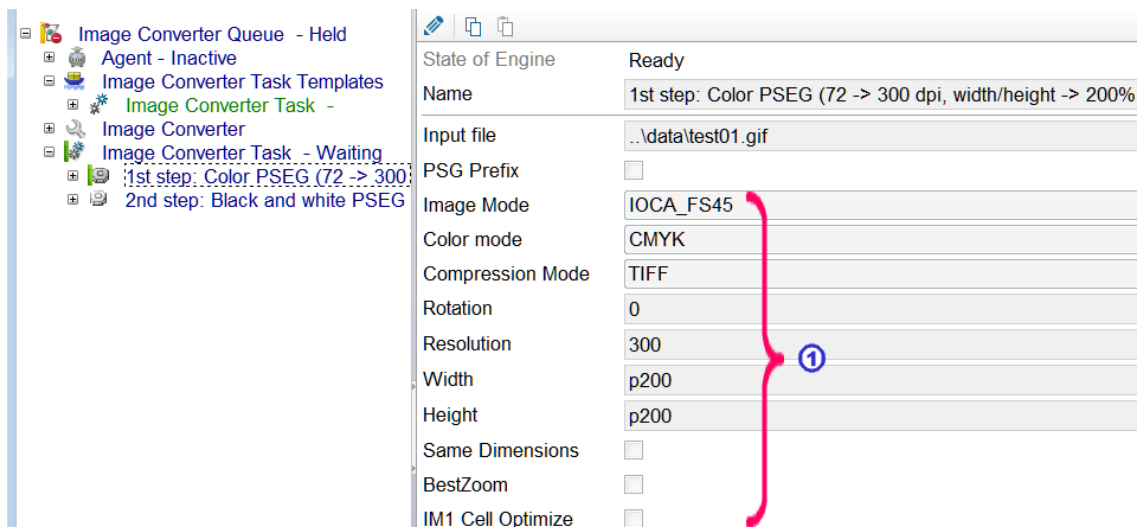
Note: Do not modify the settings of the task template provided by ISIS Papyrus, because with the next update of the Object Space these settings would be lost. Therefore please duplicate the provided task template including its children instead and modify the duplicate accordingly.

- (2) In the new queue instance, instantiate the **Image Converter Task** template by dragging it onto the **Image Converter Queue** and selecting **New Image Converter Task**:

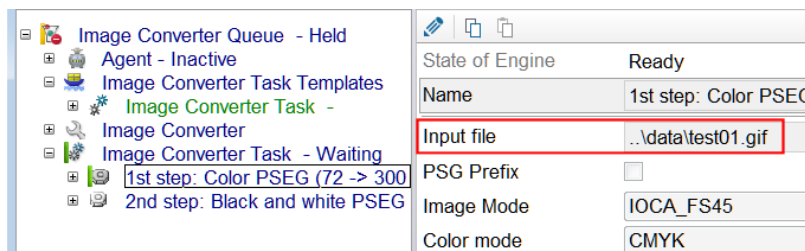


- (3) The first step of the converter queue converts a GIF image to a 256 color PSEG. The second step converts it into a black/white PSEG.

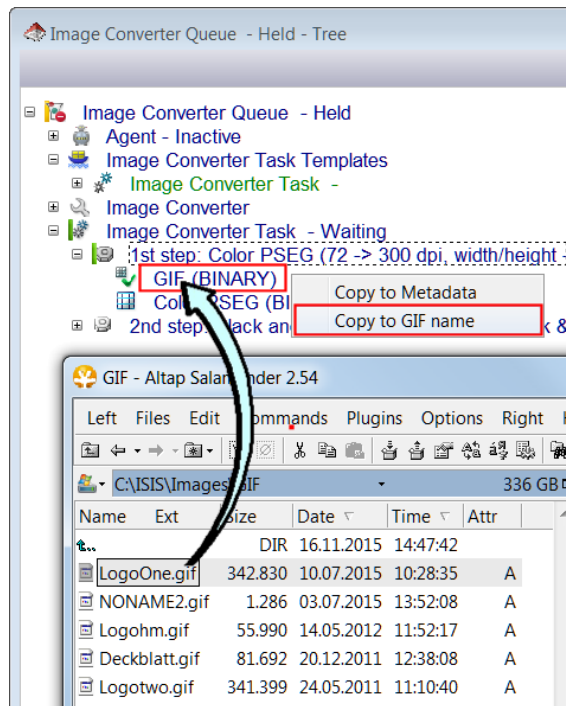
If you want, you can change the settings in the material objects **1st step** and **2nd step (1)**:



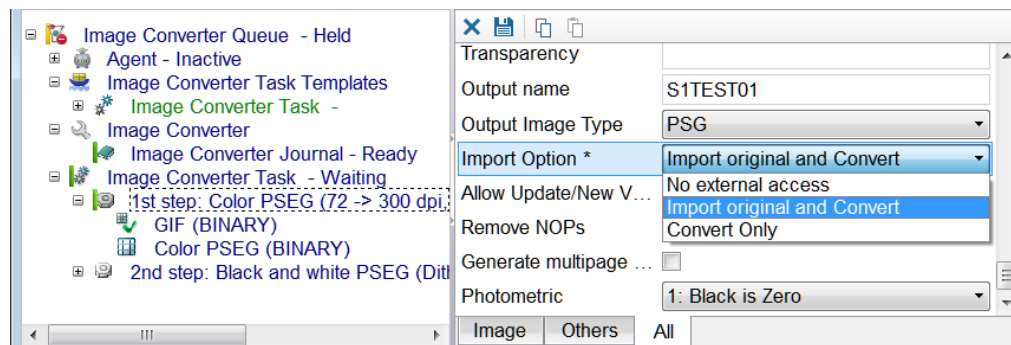
- (4) In the Image Converter Queue, the input image file is defined by the **Input file** attribute:



If you want to convert an image from the file system, empty the **Input file** attribute and drag an image file from the file system onto the **GIF (BINARY)** data object and select **Copy to GIF name** in the context menu:

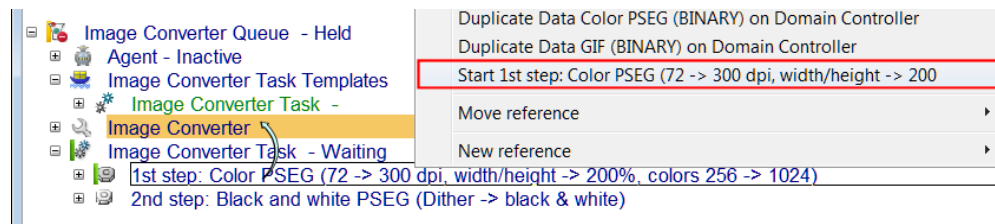


- (5) Select the **Import option** attribute of the material according to the following criteria:



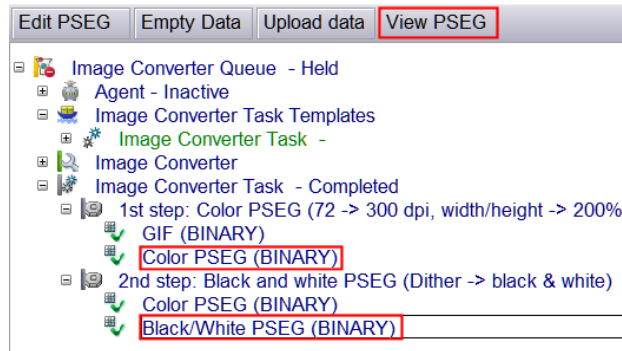
- **No external access:** converts the binary attribute of the input image object (**GIF(BINARY)**). In this case, the **Input file** attribute must be empty.
- **Import original and Convert:** converts the file specified in the **Input file** attribute and stores it in the binary attribute of the input image object (**GIF(BINARY)**).
- **Convert only:** converts the file specified in the **Input file** attribute

- (6) Now drag the **1st step** material onto the **Image Converter** tool and select **Start 1st step**:



Proceed likewise with the **2nd step** material.

- (7) As results of the image converter process, notice the binary image objects below **1st step** and **2nd step**. You can view these image objects by invoking the **View PSEG** method:



Further processing of the converted image

To view, modify or export the image of a data object into the file system, drag it onto an **IMAGE EDITOR** tool and select **Edit ...** in the context menu.

For more details, see *OverView Image Editor in Papyrus Objects* in .

Converting a data object into a resource object

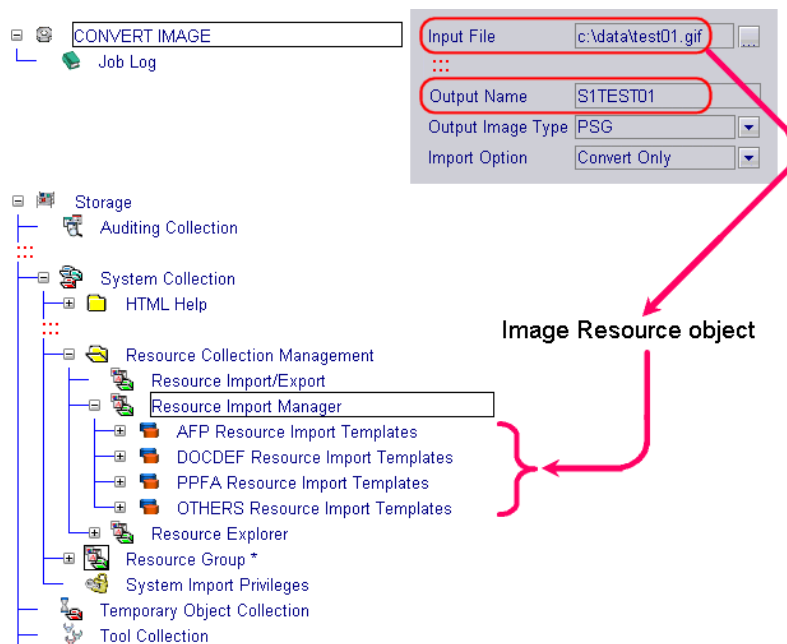
It is also possible to convert the binary attribute of a **data object** into a **resource object**. For details on how to modify the task of this ready-to-use setup, see [6.2 Technical details - available types of image conversions](#).

Note that the LOG object below the material object stays empty if option **2: use only PCSLOG and overrule all individual ISISLOG instance settings (recommended)** has been chosen in the **LOG Settings**.

6.2 Technical details - available types of image conversions

Depending on the input, the Image Converter tool automatically uses one of the following conversion mechanisms:

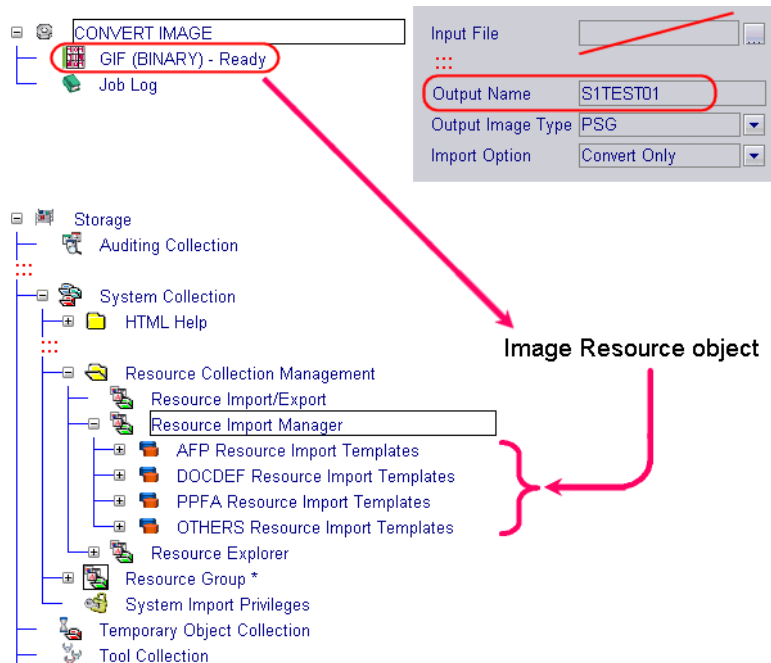
1. Convert an **image file** into the **resource object** with the specified visible name:
 - Attribute **Input File** (= method parameter "PicInputFile") references an image file
 - Attribute **Output Name** (= method parameter "PicOutputFile") is assigned with the visible name of a Resource object
 - No data object below material



Note: If the resource object does not exist yet, a new one with the specified visible name is generated, otherwise the existing resource object is overwritten. An existing resource object must be in deployment state "Development", otherwise the Image Converter tool is not able to store the image in it.

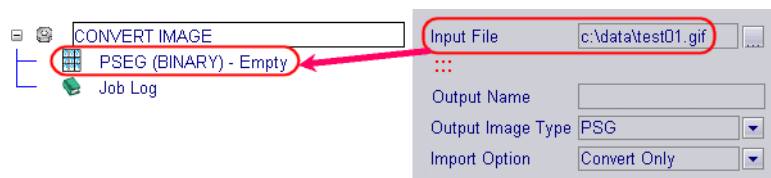
2. Convert a binary attribute of a **data object** into the **resource object** with the specified visible name:

- Data object in state "Ready" with valid image data in binary attribute
- Attribute **Output Name** is assigned with the visible name of a resource object
- Attribute **Input File** MUST be empty!



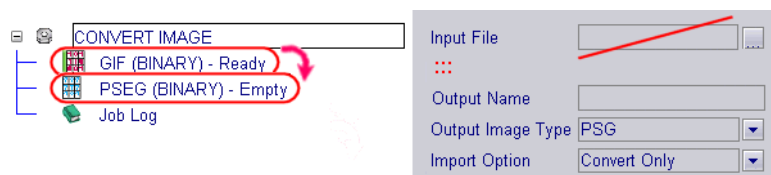
3. Convert an **image file** into the binary attribute of a **data object**:

- Attribute **Input File** references an image file
- Data object in state "Empty" or a resource object with empty attribute **CONTENT**



4. Convert a binary attribute of a **data object** into the binary attribute of another **data object**:

- Data object in state "Ready" with valid image data in binary attribute
- another data object in state "Empty" or a Resource object with empty attribute **CONTENT**
- Attribute **Input File** MUST be empty!



Online Help with F1

For details on individual elements in the object tree and data panel of these tools, see the Papyrus Objects Online Help.

Select an object in the object tree or an attribute in the data panel and press [F1].

The help panel is displayed on the right side of the Papyrus Desktop. It always shows the help information corresponding to the object that is currently selected.

Glossary

ACIF

AFP Conversion and Indexing Facility - an IBM AFP program included with PSF (Print Services Facility) to convert, index and embed resources for a MO:DCA archive document. The same function is available in Papyrus PageEXEC and Papyrus DocEXEC.

AFIG

Advanced Function Image and Graphic Feature - used in IBM printers to enable the usage of IOCA and GOCA formats.

AFP

Advanced Function Presentation. Format for the print data stream to print data on a wide variety of printers or to display data on various display devices. AFP includes creating, formatting, archiving, retrieving, viewing, distributing, and printing information. The abbreviation AFP was also used by IBM as an abbreviation for "Advanced Function Printing". Since 1995 it has been renamed "Advanced Function Presentation".

AFP Splitter

Papyrus WebRepository implementation of Papyrus Archive Module. This tool splits an AFP into its component documents according to their TLE indices and triggers the check-in of the split documents into a (printpool) Depot. Additionally, AFP Splitter can convert the split AFP into PDF, which are checked-in as well.

AFPDS

Advanced Function Presentation Data Stream. The printer data stream used for printing AFP data. The AFPDS includes composed text, page segments, electronic overlays, form definitions, and fonts that are downloaded from the operating system to the printer. It is generated, for example, by DCF/Script or by Papyrus DocEXEC.

agent

An automated user moving tasks into the appropriate queues based on conditions defined in rule objects (to check attributes and states) and starting matching tool/material methods.

ASCII

American Standard Code for Information Interchange (Standard character array for letters and symbols).

BCOCA

Bar Code Object Content Architecture - a part of MO:DCA.

bit field

A bit field is a data structure used in computer programming. The individual bits within the field can represent on/off switches for a number of options.

BLOB

Binary Large Object - collection of binary data stored as a single entity in a database management system, e.g. AFP documents, images etc.

case sensitivity

Only passwords and XML tags are treated as case sensitive by Papyrus applications. For other type of data, Papyrus applications make no difference in interpreting uppercase and lowercase letters.

CCITT

Consultative Committee for International Telegraph and Telephone. This body defines among others compression/decompression standards for black and white images. CCITT has set the Group 3 and Group 4 compression methods used originally for facsimile communications. They are used today as bitmap formats for archiving and printing. IOCA supports G3 and G4 compression.

CDP

Codepage file.

channel code

Channel codes are the very first byte in a line. They are mostly used in legacy data and are utilized to group data, as they can work as a delimiter for variable sized groups.

character set

A character set is the amount of all characters displayable on a computer system. Depending on the application there are 8 bit characters sets with 256 characters (e.g. ANSI or ASCII) and 16 bit character sets with many thousand characters (Unicode). For 8 bit character sets there are many national variations, so called code pages for displaying language-specific characters.

CHS

Character set file.

CID fonts

These PostScript fonts are Unicode outline fonts developed by Adobe. The glyphs are accessed by CID (character identifier). The CID-keyed font file format was designed for large character set fonts for use with PostScript. It is the ideal format for Chinese, Japanese, or Korean fonts (referred to collectively as CJK fonts). There are several font types available, whereas Type 1 is used in Adobe Reader for the substitution fonts.

CMYK

Cyan, Magenta, Yellow and Key. Key is usually the black ink cartridge. CMYK is a subtractive color model for color printing. The CMYK model is called subtractive, because each ink "subtracts" brightness from white (masking partially or entirely certain colors on the white paper).

code point

A code point value represents the position of a character in the coded character set. In Unicode, the code range (or code space) is an integer between 0 and 10FFFF. The letter "A", for instance, is represented by the Unicode code point U+0041 (decimal: 65).

coded font

AFP resource that associates a specific codepage with a specific character set. A character must be included in the specified character set and listed on the specified codepage before it can be printed. Coded fonts are mainly used for presenting text, character strings, and barcode HRI.

codepage

Characters are internally represented as numbers in computers. The mapping of a (control) character to a number is done with a codepage. Sometime the terms character map or - especially in the HP world - symbol set are used.

comb field

A comb is a set of vertical lines within a field that separate each character typed. By forcing the user to enter each character in its own box optical character recognition becomes a lot easier and more reliable. It can also be used for fields with a fixed length such as an IBAN.

copy group

A copy group is a subset of a form definition that specifies a physical page. A form definition can contain one or several copy groups. Each copy group can specify characteristics such as page position, paper source, and duplex printing. Synonym for medium map.

Db2

IBM Database 2. Db2 is a Relational Database Management System.

DBCS

Double Byte Character Set. Character encodings that use exactly two bytes for encoding of graphic characters. Used for presenting documents in Korean, Chinese and Japanese languages, which have up to 17.000 characters per font. To represent a single character, two bytes are used instead of one (SBCS). DBCS also requires operating system support to be able to input the language from the keyboard. AFP and therefore all Papyrus products support DBCS.

DFA

"DFA" is the file extension of a DOCDEF, a document definition, which is the template for formatting input data with Papyrus DocEXEC. A DOCDEF can be designed with Papyrus Designer. Sometimes "DFA" is used synonymously with "DOCDEF". "DFA" also stands for "Document Formatting Aid" referring to Papyrus DocEXEC as a document formatting tool following the IBM term PPFA, Page Printing Formatting Aid, which is a page describing tool. "DFA" is also the short form of resource type DOCDEF in the Papyrus WebRepository's Resource Collection.

DOCDEF

Document definition. DOCDEF is the source code written in document definition language processed by Papyrus DocEXEC to produce documents. A DOCDEF consists of the Format Group Definitions and Document Format Definitions, which are handled in separate windows in Papyrus Designer.

DocEXEC

The document formatting and rendering engine used by Papyrus Designer and other tools.

dpi

Dots per inch. A measurement of resolution, e.g. the number of pixels per inch on a workstation display monitor or printing device.

EBCDIC

Extended Binary Coded Decimal Interchange Code

FGI

Font Global Identifier

FLSF

Font Library Service Facility

FOCA

Font Object Contents Architecture - a part of MO:DCA.

FON

Coded font file that associates a codepage (CDP) and character set (CHS). See also [coded font](#).

form definition (FORMDEF)

A form definition (FORMDEF) is a resource that is required when printing an AFPDS. It is an interface between the generated AFPDS and the printer. A form definition determines the physical properties of the medium on which printing is to take place and regulates how the document pages of the AFPDS are to be mapped to the physical pages of the paper stored in the printer's paper trays. It describes the properties of the physical page, e.g. simplex/duplex printing, paper tray selections for printing, the placement of logical pages on the physical page (N_UP), etc. Form definitions can be created and compiled with OverView AFP Designer in Data Placement mode. A compiled form definition has the file extension *.FDF.

FS

An IOCA Function Set is a set of self-defining fields that describe an image object. With their identification number, products determine the level of support they must provide to generate or receive IOCA image objects. [IOCA](#) defines the following function sets:

- **FS10** for bilevel images
- **FS11** and **FS20** for bilevel, grayscale, and color images
- **FS42** for tiled bilevel images and tiled CMYK images with one bit per spot
- **FS45** for tiled bilevel and tiled CMYK images with either one or eight bits per spot

GCSGID

Graphical Character Set Global Identifier

GOCA

Graphic Object Content Architecture - a part of MO:DCA. Papyrus products support GOCA for generating chart graphics.

GUID

Globally Unique Identifier. A worldwide unique 128-bit number used as identifier for various things. Generally, GUIDs are shown as a sequence of 32 hexadecimal characters, in an 8-4-4-4-12 representation.

The GUIDs used in Papyrus products (see [logical GUID](#), [version GUID](#), and [physical GUID](#)) to identify objects and versions are based on the standard for UUIDs (Universally Unique Identifiers), defined in RFC 4122 and Standard ISO/IEC 9834-8:2005. In contrast to Object IDs, GUIDs do not store the information on which node the object is located and on which one it has been created.

ICD

ISIS Control Data. The parameters in an ICD file define settings used by the File Receiver (Dispatcher, ICD Receiver).

IEL

Index Element - a part of MO:DCA.

IMM

Image Object Content Architecture terminology. IMM stands for the Invoke Medium Map structured field.

See [copy group](#).

IOCA

Image Object Content Architecture - a part of MO:DCA. IOCA data architecture specifies resolution-independent images and also defines data compression, color and gray-scale encoding. Valid only on printers supporting IOCA.

IOCA compression types

- **None** - for binary image data without any compression
- **MMR** - for a compression algorithm that allows both one-dimensional and two-dimensional coding
- **RL4** - for a binary, one-dimensional, run-length coding algorithm of compression
- **G4 MMR** - for two dimensional image compression
- **FS45 JPEG** or **FS45 TIFF** - to compile colored images into inline FS45 images. The profile parameter ICOMP will show the values "JPEG" or "TIFF". See also [FS](#).

IPAS

ISIS Papyrus Product Authorization Service. Standard mechanism for authorizing Papyrus products and for centrally managing all Papyrus product authorizations.

ISPF

Interactive System Productivity Facility. ISPF is an interface for Time Sharing Option under z/OS.

JCL

Job Control Language. JCL is a scripting language used on IBM mainframe operating systems to instruct the system how to run a batch job or start a subsystem.

Library

Folder object where all templates are stored.

logical GUID

An object's logical GUID (system attribute \$logicalGUID) remains constant over all versions. This attribute can be used to find the latest version of any object.

logical page

A logical page is a presentation space on the physical page and is specified by characteristics such as size, shape, orientation, and offset. Orientation and offset are specified relative to the [physical page](#). The logical page is described by the page definition, which itself contains one or more page formats.

LPI

Lines per inch

material

An object may act as material if it has at least one material method defined that can be called in combination with the matching tool method (same "Method internal name" or "Method implementation name") of an object acting as tool.

MBCS

Multi Byte Character Set. Character encodings that use a varying number of bytes for encoding of graphic characters. Most prominent MBCS encodation schemes are UTF-8, UTF-16, UTF-7, Shift-JIS, Big5 und GB2312. UTF-8, for example, uses between one and four byte.

MCF

Map Coded Font. A structured field.

medium map

See [copy group](#).

MO:DCA

Mixed Objects : Document Contents Architecture. A compound document format for text and graphics elements. 'Mixed Object' refers to the fact that an MO:DCA file can contain various types of objects, including text, images, vector graphics, and barcodes.

node

A node consists of a Papyrus Objects Kernel and its related Object Space running on a machine which is connected via TCP/IP to the Papyrus Objects Domain (LAN or WAN). Several nodes can run simultaneously on one physical machine. The graphical user interface Papyrus Desktop is optional and is automatically installed if available for the operating system specified via the installation routine. Alternatively it is also possible to use a web browser, if a Papyrus WebPortal is installed on the node whose Object Space should be accessed (thin client solution).

NOP record

No operation record. NOP records can be part of an AFPDS and contain user-specified information and/or metadata, which are used by some 3rd-party postprocessing applications.

Object Space

The Object Space is an object oriented database where the objects (data units), logs, and settings of Papyrus WebControl and WebRepository are stored. Papyrus Objects employs an integrated, compiled-in object database that is fully licensed from Oracle and does not need to be purchased separately and requires no additional installation, administration or maintenance.

OGL

Overlay Generation Language. A script language for designing overlays (electronic forms) with OverView AFP Designer (in Forms Design mode). The overlay script can be compiled by OGL compiler in OverView AFP Designer resulting in an overlay. The script language is published by IBM.

overlay (OVL)

Overlay is the technical term for an electronic form. An overlay is a AFP resource object. It is a collection of predefined data, such as lines, shading, text, boxes, or logos, that can be merged with variable data on a page at print time. Overlays can be created with Overview AFP Designer by means of the Overlay Generation Language (OGL).

page definition

A page definition (PAGEDEF) specifies characteristics of the logical page such as dimension, coordinates of the first printed line, and print direction of the logical page. It can be overridden at a lower level by a page format specification. It has the file extension *.PDF. Page definitions can be created with OverView AFP Designer in Data Placement mode.

page format

Page formats are subsets of page definitions that specify a logical page. Different page formats are used if more specifications (as e.g. print direction, size of the logical page) are needed to format a page within a single print job. It can be overridden at a lower level by a subpage specification. Synonym for data map.

page segment

Page segment, segment or PSEG. An AFP resource that contains composed text and images, prepared before formatting and included in a document when it is printed. Page segments can be created with Overview Image Editor or Papyrus ImageConverter.

Papyrus Desktop

Papyrus Desktop is the graphical user interface to visualize the Object Space; it has to be connected to a Papyrus Objects Kernel and represents the respective objects on the local or on remote nodes.

PCS

Process Control System. PCS is represented by the ISIS Papyrus Common Libraries (also known as PCSDLLs). PCS is a layer between the operating system and Papyrus applications that automatically handles a Shared Memory Segment (SHM, PCSDATA) which is used for the inter-process and inter-thread communication. It also handles the creation of the PCS log files.

pel

picture element. The smallest printable or displayable unit on a physical medium.

physical GUID

An object's physical GUID (system attribute \$physicalGUID) changes each time it is *modified*. This attribute, together with modification count, is used internally for various purposes such as proxy management.

physical page

A physical page can be seen as a sheet of paper (front or back), on which data is printed. The coordinates of the physical page are always absolute, as opposed to the logical page (LOGICALPAGE), which is defined relative to the physical page. A physical page is specified by form definition (FORMDEF), copy group (COPYGROUP) and subgroup (SUBGROUP).

pixel

See [pel](#).

PMF

Print Management Facility. Was used to generate print resources (outdated). ISIS Papyrus offers a conversion tool to convert the generated objects back to PPFA source code.

POSIX

Portable Operating System Interface. POSIX is a family of standards specified by the IEEE Computer Society for maintaining compatibility between operating systems. POSIX defines an API, along with command line shells and utility interfaces, for software compatibility with variants of Unix and other operating systems.

PPFA

Page Printing Formatting Aid. An IBM licensed program to create and store form definitions and page definitions. These stored resource objects are used to format printed output. OverView AFP Designer in PPFA mode allows to create page definitions and form definitions as well.

PQL

Papyrus Query Language. PQL is a script language to interact with the Object Space.

product authorization

A key entitling a customer to use certain Papyrus product(s). Colloquially often incorrectly referred to as license.

PSEG

See [page segment](#).

PSF

Print Services Facility. An IBM software for driving IPDS printers on various platforms.

PTOCA

Presentation Text Object Contents Architecture. A part of MO:DCA.

PTX

Presentation Text. A structured field used to specify text data and text position, rotation, and fonts to be used when presenting text data.

queue

A queue is the place where tasks are processed. A queue usually contains an agent and a tool. The agent operates the tool(s) of the queue to process incoming tasks and their materials (tool/material matching).

Repository

Folder object where all classes are stored.

resolution

AFP resources as raster fonts (FON), overlays (OVL), and page segments (PSEG) may exist in different resolutions (240, 300, or 600 dpi). The resolution of the existing resource has to be considered especially for printing, otherwise this might lead to a loss in quality of the printed document. See also [dpi](#).

resource library

AFP resources as fonts, images, overlays, page definitions, and form definitions are stored as files in the file system. A resource library profile determines their location in the file system, and Papyrus products access resources by means of this profile. Resource libraries are commonly used by Papyrus products running standalone, i.e. outside a Papyrus WebRepository Domain.

return code

After a server module has completed its run it returns a code informing about its success. RC=0 means that no errors occurred. 4 reports a warning, 8 an error, 12 warning and error and 15 a fatal error.

RGB

Red, green and blue. RGB is an additive color model for the display of colors on screens. The RGB model is called additive, because when equal amounts of the light dots red, green, and blue are added up at full intensity, they create white.

SAA

Systems Application Architecture. A set of standards for computer software developed by IBM.

SBCS

Single Byte Character Set. Character encodings that use exactly one byte for each graphic character.

starter

Used by the Papyrus Objects Kernel to start the DLLs behind the Papyrus Server Modules, which would otherwise be a task of Papyrus Server. Starter Modules are DLLs of Papyrus Server (e.g. File Receiver (ICD) is the Dispatcher DLL of Papyrus Server).

Storage

A folder object where system relevant instances are stored.

structured field

Structured fields are used to define composed text pages and line format data. The structured field is a self-identifying string of bytes containing data or parameters and must have an introducer, which contains a length field, an identifier, a flag, and a sequence number. This is followed by parameter bytes that contain control information or data to be printed.

subgroup

A subgroup, which is incorporated in a copy group, provides further possibilities to specify a physical page. With a subgroup, characteristics as number of copies, overlays, text suppression, single sided or duplex printing, and forms flash can be defined. See also [copy group](#) and [physical page](#).

subpage

A subpage is a part of a logical page on which line data may be placed. It is incorporated in a page format, which may have multiple subpages.

task

A task is a job description including the sequence of operations to be performed through its material objects, all specific parameters applied to that job and input data and produced output data attached as data objects to its material objects.

TLE

Tag Logical Element. Structured field for group indexes.

tool method

A method type that can be used in combination with the related [material](#) method(s).

TRC

Table Reference Character

TSO

Time-Sharing Option. TSO is an interactive command line interpreter for z/OS.

USS

Unix System Services. USS is a component of z/OS.

version GUID

An object's version GUID (system attribute \$versionGUID) changes each time a new version of it is created. In unversioned instances, this attribute holds the version of the class or template from which the instance was derived.

Support and contact information

If you need support or general information, do not hesitate to contact ISIS Papyrus:

General contact information

ISIS Papyrus Europe AG
Phone: +43 2236 27551
E-mail: info@isis-papyrus.com
Website: www.isis-papyrus.com

Product & project support

Phone: +43 2236 27551-111
E-mail: support@isis-papyrus.com

We will attempt to answer all inquiries within 24 hours.

Click here to open the **ISIS Papyrus Product Support Request** form where you can enter and submit information:

www.isis-papyrus.com/support-request

It will help us answer your inquiries if you send us the version number and compilation date of the relevant Papyrus products. To find out this information, choose **Help | About...** or **Help | Product information** from the main menu. By left-clicking on the version number line you can copy the data.

For more details on support types and what kind of information is essential, see *Contacting ISIS Papyrus for Support General Information (supporte)*.