Update to PPD Specification
Version 4.3
Adobe® Developer Support

Technical Note #5645

29 April 1997
Contents

1 Introduction 5
2 *FastImage 6
3 Resource Entries 7
  *Resource 7
  *categoryNameInstance 8
  *Resource and *categoryNameInstance combinations 9
4 *GuaranteedMaxSeparations 9
5 *NonUIConstraints 10
6 *Font 10
Update to PPD Specification
Version 4.3

1 Introduction


This amendment does not introduce any new required keywords. All PPD files compliant with PPD Spec 4.3 will continue to work with drivers, applications, and plug-ins that have been modified according to this document. PPD files compliant with this updated specification will continue to use

*PPD-Adobe: "4.3"
*FormatVersion: "4.3"

Three entries are introduced—*FastImage, resources, and
*GuaranteedMaxSeparations. *NonUIConstraints is extended to work with
*GuaranteedMaxSeparations. *Font is extended to work with the charset named ExtendedRoman and the encoding named Win1250.

The new optional *FastImage UI entry is associated with the HalftoneMode user parameter, introduced in PostScript language version 3010 as an optional feature. This entry is recognized by the Windows AdobePS 4.2 and Macintosh AdobePS 8.5 printer drivers.

Resource-related keywords allow specification of both actual and virtual resource categories and instances. This will initially indicate whether in-RIP trapping is supported. In-RIP trapping is an optional feature of PostScript language versions 2017 and later. These entries will be recognized by certain Adobe plug-ins.
The new optional *GuaranteedMaxSeparations entry is associated with the MaxSeparations page device parameter, introduced in PostScript language version 3010. This will be recognized by a future release of Adobe PageMaker.

2 *FastImage

*FastImage modeOption: "invocation"

*DefaultFastImage: modeOption

*?FastImage: "query"

*FastImage is similar to *OutputMode except that it is specifically for the fast image processing feature introduced in PostScript language version 3010, and controlled by the new HalftoneMode user parameter. This feature must be thoroughly understood for a particular product before that product’s *FastImage entry is written.

This feature is only applicable to monochrome products; the fast image keywords should be omitted on products incapable of a monochrome-only mode. OEMs with products capable of both monochrome and color modes may want to constrain the *FastImage choices according to *ColorModel choices, via *UIConstraints.

The currently valid modeOptions are:

- None or False—No fast image processing will be done by either the host or the product.

- 1 or True—The host may optionally produce images optimized for speed with insignificant loss of quality. The product may override any requests to change the halftone, and may process images using the Adobe fast image feature (HalftoneMode) with no significant loss of quality.

- 2—The host behaves the same as for modeOption 1. The product behaves the same except that it should process images faster with a possible minor loss of quality, on average.

The following is sample code for the PickOne variant:

*OpenUI *FastImage/Showing Images: PickOne
*DefaultFastImage: None
*OrderDependency: 90 AnySetup *FastImage
*FastImage None/Document Settings: "</HalftoneMode 0>> setuserparams currenthalftone sethalftone"
*End
*FastImage 1/Optimized for Quality: "
<</HalftoneMode 1>> setuserparams
currenthalftone sethalftone"
*End
*FastImage 2/Optimized for Performance: "
<</HalftoneMode 2>> setuserparams
currenthalftone sethalftone"
*End
*?FastImage: "
<<0 (None) 1 1 2 2>>
currentuserparams /HalftoneMode get get = flush"
*End
*CloseUI: *FastImage

The value for *DefaultFastImage should reflect the value of the
HalftoneMode user parameter when the product is first initialized.

If output from HalftoneMode 2 is not supported or if the OEM wishes to hide it from the user, but HalftoneModes 0 and 1 are supported then a Boolean
*OpenUI style may be best, with an *OpenUI translation string such as
“Faster, uniform images?”.

A product is entitled to have HalftoneMode 1 mean that a product-specific halftone is imposed on the job. If such a product also allows HalftoneMode 0 and 2 then the above sample code would typically be used.

If a product has HalftoneMode 1 mean that a product-specific halftone is imposed on the job, and HalftoneMode 0 is supported but HalftoneMode 2 is not, then another UI entry such as *OutputMode should be used.

Adobe’s Windows® 95 printer driver AdobePS 4.2 and Adobe’s Macintosh®
printer driver AdobePS 8.5 recognize the *FastImage keyword and optimize for speed with insignificant loss of quality for modeOptions 1 and 2.

3 Resource Entries

The fact that certain resources exist within a product may be specified with the *Resource and *categoryNameInstance keywords.

These keywords are optional. Omitting these keywords says nothing about whether unmentioned PostScript language resource categories or instances exist.

3.1 *Resource

*Resource categoryName: Implicit | Regular | Virtual
This entry declares that the *categoryName* resource category applies to this product and that the PPD file may also contain *categoryNameInstance* keyword(s). For example, if *categoryName* were TrappingType then *TrappingTypeInstance* keywords may follow.

The currently defined values for *Resource* are:

- **Regular** — The product has the regular resource category named *categoryName*. The PostScript operator findresource is meaningful for this category.

- **Implicit** — The product has an implicit resource category named *categoryName*. The PostScript operator findresource is not meaningful for this category; resourceforall and resourcestatus are. Implicit resource instance names or numbers represent some built-in capability of the product.

- **Virtual** — The product does not have a resource category named *categoryName*. Virtual resource instance names or numbers represent some built-in capability of the product that is not otherwise conveniently representable.

*Resource* may not appear within *OpenUI* and *CloseUI*.

The absence of a particular *Resource* entry says nothing about the existence of that category on the product. Features reflected in existing PPD file entries, such as *Font*, should continue to use those PPD file entries.

### 3.2 *categoryNameInstance*

*categoryNameInstance* instanceName: “...”

This means that *categoryName* has been declared in a *Resource* entry and that instanceName is the name of an instance for this category. Because the resource may be Virtual, the category and instance may not be visible from a PostScript job.

**Note** instanceName is restricted to the syntax for PPD option keywords. This means that instanceName may not contain special characters, even though they are legal in a PostScript program.

The meaning of “...” is undefined. Parsers should accept a normal InvocationValue here, but its use is undefined. PPD files should supply an empty InvocationValue, “”, to avoid undefined effects. A future version of the PPD File Format Specification may define a meaning.
The absence of a particular entry says nothing about the existence of the *categoryName* category or its instance named *instanceName*. Features reflected in existing PPD file entries, such as *Font*, should continue to use those PPD entries.

*Comment*  
This style was chosen over more compact representations because many PPD parsers already recognize this style.

### 3.3 *Resource and *categoryName*Instance combinations

The following combination is understood by certain plug-ins. This combination should be included in PPD files that may be used with such plug-ins.

#### 3.3.1 TrappingType

The entries below indicate that the product supports in-RIP trapping as defined in the Supplements for PostScript language versions 2017 and later. Subsequent products may have Implicit category TrappingType with several instance numbers. These particular PPD file entries are recognized by certain plug-ins that will be available in 1997.

*Resource TrappingType: Virtual
*TrappingTypeInstance 1001: *

### 4 *GuaranteedMaxSeparations

*GuaranteedMaxSeparations number: "..."

*DefaultGuaranteedMaxSeparations: number

*?GuaranteedMaxSeparations: "query"

This entry indicates the maximum number of selectable separations that are guaranteed to be supported on all configurations. In other words, this is the smallest value of the MaxSeparations page device parameter among all available configurations. *NonUIConstraints may be used to affect the applicable value.

Several *GuaranteedMaxSeparations keywords may be supplied. Some of these choices may be unavailable due to the effect of *NonUIConstraints entries. The largest number from the available choices indicates the maximum number of selectable separations that are guaranteed to be supported in available configurations.

For example, a product may support MaxSeparations of 3 for all configurations except duplex or 11x17 paper, when it supports only 1. The following example illustrates this situation.
*DefaultGuaranteedMaxSeparations: 3
*GuaranteedMaxSeparations 3: ""
*GuaranteedMaxSeparations 1: ""
?GuaranteedMaxSeparations: 
   currentpagedevice /MaxSeparations get -*  
*End

*NonUIConstraints: *Duplex DuplexTumble *GuaranteedMaxSeparations 3  
*NonUIConstraints: *GuaranteedMaxSeparations 3 *Duplex DuplexTumble  
*NonUIConstraints: *Duplex DuplexNoTumble *GuaranteedMaxSeparations 3  
*NonUIConstraints: *GuaranteedMaxSeparations 3 *Duplex DuplexNoTumble  
*NonUIConstraints: *PageSize Tabloid *GuaranteedMaxSeparations 3  
*NonUIConstraints: *GuaranteedMaxSeparations 3 *PageSize Tabloid  
*NonUIConstraints: *PageRegion Tabloid *GuaranteedMaxSeparations 3  
*NonUIConstraints: *GuaranteedMaxSeparations 3 *PageRegion Tabloid

If only one value for *GuaranteedMaxSeparations is applicable for a product, then only *DefaultGuaranteedMaxSeparations may be supplied. Here is an example.

*DefaultGuaranteedMaxSeparations: 2

If this entry is omitted for a product compliant with PostScript language version 3010 or later a *DefaultGuaranteedMaxSeparations value of 1 is implied. If this entry is omitted for a product compliant with PostScript language version 2017 or prior, a value of 0 is implied.

This entry uses the UI style solely to take advantage of *NonUIConstraints entries. Since this entry does not indicate a user selectable feature, it should not appear between *OpenUI and *CloseUI.

The meaning of "..." is undefined. Parsers should accept a normal InvocationValue here, but its use is undefined. PPD files should supply an empty InvocationValue, "", to avoid undefined effects. A future version of the PPD File Format Specification may define a meaning.

5  *NonUIConstraints

The *NonUIConstraints keyword has been extended so it may be used with *GuaranteedMaxSeparations.

6  *Font

The *Font keyword has been extended to allow a new encoding name and a new charset name. This allows use of fonts designed to work with Central European alphabets. The AdobePS 4.2 driver understands these.

ExtendedRoman may be specified as *charset*. This is the combination of:

- Standard charset
- the characters mentioned in the Win1250 encoding
- certain other characters

The precise definition of ExtendedRoman is in the *Supplement: PostScript Language Reference Manual, For Version 3010*. 