

Prinergy 1.1: Changes to the Normalizer

FileName
Norm_1_1.doc

September 14, 1999

CreoScitex
3700 Gilmore Way
Burnaby, BC
Canada V5G 4M1
(604) 451-2700

Copyright 2001 Creo Products Inc. All rights reserved.

Revision History

Revision	Date	Author	Comments
1	9 Aug 1999	Bob Cook	Creation
2	10 Aug 1999	Bob Cook	Revisions
3	14 Sept 1999	Bob Cook	Rewrite to cover all of 1.1 features
4	20 Sept 1999	Bob Cook	Add suggested config (from Larry M.)
1	9 Aug 1999	Bob Cook	Creation



Prinergy 1.1: Changes to the Normalizer

CONTENTS

Revision History ..... **i**

1. New Features For Normalizer in Prinergy 1.1 ..... **1**

2. Direct Input of Encapsulated PostScript Files (EPS) **1**

3. Direct Input of Desktop Color Separation Files (DCS) **2**

4. Direct Input of PostScript from Imation Prepress Xtension **2**

5. Normalizer Input Configuration **2**

6. PostScript Bypass Feature **5**

7. PostScript Bypass – How It Works **6**

8. Recommendations for Using PostScript Bypass **7**

9. Further Information on the Configuration File **8**

9.1 Default NormInputConfig File Included with Prinergy 1.1 8

9.2 Adding PostScript Bypass for Shira CEPStoPS 9

9.3 Troubleshooting 9

1. NEW FEATURES FOR NORMALIZER IN PRINERGY 1.1

This document is intended as a reference for people supporting the Prinergy product. This is not a public document. This document contains confidential information. Do not distribute this document to customers.

The Normalizer JTP has a number of new features for Prinergy 1.1. This document outlines those changes, and documents the advanced features that are controlled via the new configuration file.

New features include:

- The ability to directly refine EPS files. The resulting PDF will contain a single composite page.
- The ability to directly refine DCS files. The resulting PDF will contain a single separated page, containing all the colors referenced from the DCS main file.
- The ability to directly refine PostScript generated from the Imation Prepress Xtension (also known as the Luminous Prepress Xtension) and preserve bleeds.
- The ability to customize the PDF page size based on input PostScript types.
- The ability to customize the PDF trim size based on input PostScript types.
- The ability to employ PostScript Bypass for certain types of EPS or DCS input PostScript types.

In addition to feature descriptions, each item contains information regarding known limitations, workarounds, etc.

2. DIRECT INPUT OF ENCAPSULATED POSTSCRIPT FILES (EPS)

The Normalizer can now accept EPS files from applications such as Quark, PhotoShop, Shira CEPStoPS, etc. These EPS files can contain previews (the Normalizer will safely ignore embedded previews even if saved using a PC-preview format).

The Normalizer converts the input EPS file to a composite single-page PDF. All spot colors are retained as specified in the original input file.

By default, the Normalizer will treat the bounding box (as indicated by either a `%%BoundingBox:` comment or `%%HiResBoundingBox:` comment in the PostScript) as the page size to use for the resulting PDF. This behavior can be modified using the Normalizer Input Configuration file (see that section for details).

There are no known limitations or issues noted for this new input format.

3. DIRECT INPUT OF DESKTOP COLOR SEPARATION FILES (DCS)

The Normalizer can now accept DCS files from applications such as Quark, PhotoShop, Shira CEPStoPS, etc. These can be DCS 1 or DCS 2 format, and can contain previews (the Normalizer will safely ignore embedded previews even if saved using a PC-preview format). The Normalizer supports DCS 2 single-file and DCS 2 multi-file format (DCS 1 is always a multi-file format). The details of the DCS specification can be viewed at the following web link: *http://www.quark.com/dcs*

The Normalizer converts the input DCS file to a separated single-page PDF. Any colors (either process or spot) defined in the DCS file are retained as separations.

By default, the Normalizer will treat the bounding box (as indicated by either a `%%BoundingBox:` comment or `%%HiResBoundingBox:` comment in the PostScript) as the page size to use for the resulting PDF. This behavior can be modified using the Normalizer Input Configuration file (see that section for details).

There are some limitations to be noted regarding this new input format:

- DCS 1 and DCS 2 multi-file formats cannot be used with hot folder input. The Normalizer will fail to locate the external files associated with the DCS master file.
- DCS files are always treated as separated, thus color matching and trapping will not be possible for this input format.

4. DIRECT INPUT OF POSTSCRIPT FROM IMATION PREPRESS XTENSION

The Normalizer now correctly supports PostScript generated by the Imation Prepress Xtension for Quark Xpress. In Prinergy 1.0, these input files would not retain the correct page size, and any bleed would be removed from the left and bottom edges of each page.

The Normalizer will now adjust the page size of the resulting PDF such that there is a ¼" bleed area added around the original page size as defined in Quark Xpress. In addition, the trim box is automatically set to the original Quark page size. This behavior, including the size of bleed, can be modified using the Normalizer Input Configuration file (see that section for details).

It's recommended to use the Imation Prepress Xtension version 1.8. That version is currently available at: *http://www.luminous.com/support/02.04updatesdwndls.htm* Note: this link is valid as of September 14, 1999 but may not remain valid due to reorganization of that site.

5. NORMALIZER INPUT CONFIGURATION

The Prinergy Normalizer can be configured to detect certain types of PostScript input files, and to apply specific attributes while processing that input type. Attributes such as page size and trim size can be applied to some types of inputs. The Normalizer identifies input files using the PostScript comments that appear at the beginning of each input file.

Some experience with looking at PostScript files is useful when modifying the content of this configuration file. The name and location of this file is set in the registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\CREO\ARAXI\Build\DEFAULT\
      ENOR\NormalizeJTPName\Ps Input Config File
```

where *Build* and *NormalizeJTPName* will vary according to your installation (the JTP name is visible in the Prinergy Administrator).

By default, this entry contains the value **NormInputConfig,** which is the name of the default configuration file inside the AdobeExtreme\bin folder. In general it should not be necessary to edit this registry key. Instead, modify the indicated text file directly. Be sure to make a copy of the input file prior to modifications.

The Normalizer Input Configuration file is a text file, editable with Notepad. Text lines beginning with the # character are ignored; these are for human readers.

Each unique input file type is called a "target" and must be declared prior to describing the input file's PostScript comments. It must follow this format:

```
target name
```

where *name* is a unique string that describes the purpose of this target. This string will be visible in the diagnostics log, to indicate successful matching.

Following the target declaration, one or more PostScript comments are required, as a way to uniquely identify the input file. Any valid comment found in the header of the PostScript file can be used. In PostScript, comments begin a line with a % character, and are followed by one or more non-space characters. Here is an example using the `%%Creator:` comment:

```
%%Creator: = a unique creator string
```

where the first portion (in this case its `%%Creator:`) is the comment to look for, the comparison operator (in this case its =) specifies how to match the text, and the text to match against. Here is another example, testing for the type of input file:

```
%!PS-Adobe-3.0 = EPSF-3.0
```

This example is testing the PostScript comment `%!PS-Adobe-3.0` (usually found as the first line of a PostScript file) for the text `EPSF-3.0` (which is the standard required for EPS files). When specifying comparisons to the PostScript comments, there are four different text-matching operators:

=	Exact text match
~=	Text must contain
!=	Text must not equal
!~=	Text must not contain

When the text to match (the text after the comparison operator) is blank, the meaning is slightly changed:

`=`	Comment must be present and also have no trailing text
`~=`	Comment is required to be present, but no matching is performed
`!=`	Comment must be present and must have trailing text
`!~=`	Comment cannot be present in the input file

Following one or more lines of comparisons, one or more "attributes" can be assigned. Here is the list of supported attributes, with the specific parameters required:

`@bypass`

Indicates that this target can use the PostScript Bypass feature. If this attribute is missing then the target will not use PostScript Bypass. By default, no targets include the PostScript Bypass attribute.

`@bypassopi`

Indicates that this target can use the PostScript Bypass feature during OPI Image substitution. This feature is currently not implemented, hence this attribute is ignored. Using this attribute may cause the Normalizer JTP to terminate unexpectedly.

`@cleanmultipage`

Indicates that this target is a "well-behaved" input file that conforms to the format specified by the Encapsulated PostScript reference, except that it can be a multi-page input. Not compatible with the `@bypass` or `@bypassopi` options. Use this option with care. Two examples of such "well-behaved" input files are PostScript from the Imation Prepress Xtension, and from Creo PreScript.

`@pagebox bbox` *`llx lly urx ury`*
`@pagebox fixed` ***`llx lly urx ury`***

Specifies the page size for the input file type. When the first parameter is `bbox`, the remaining values are adjustments to the values found in the `%%BoundingBox:` comment. Typically negative numbers are used for *`llx`* and *`lly`*, while positive numbers are used for *`urx`* and *`ury`*. This will increase the size of the bounding box.

If the value of the first parameter is `fixed`, the remaining numbers replace the values found in the %%BoundingBox: comment.

The resulting values, as modified, are used to set the page size of the PDF. This attribute has no effect if the input file does not contain a `%%BoundingBox:` comment. Note that `%%HiResBoundingBox:` is used instead of `%%BoundingBox:` if present.

`@trimbox bbox  llx lly urx ury`
`@trimbox fixed llx lly urx ury`

> Much like the `@pagebox` attribute, except this modifies the TrimBox value in the PDF page. This affects the default offset and trim values when the PDF is registered, similar to the AutoGeometry feature. Ignored if the page does not have a `%%BoundingBox:` comment.

`@pscode postscript code to insert`

> Adds a raw PostScript fragment to the input for this target. Use with caution, as this can cause the Normalizer to abort the job if the PostScript string contains errors.

Following the PostScript comments and attributes, the next target should begin again with the `target` keyword.

When attempting locate a target match, comparisons are performed in the order presented in the text file (from top to bottom). The search stops upon finding a successful matching target. Note that for a target match to succeed, all PostScript comments must match as specified in the target list.

It is not necessary to restart the Normalizer JTP after modifying the configuration file. The Normalizer always re-reads the configuration file before beginning each job.

6. POSTSCRIPT BYPASS FEATURE

The most significant feature accessible via this configuration is PostScript Bypass.

The PostScript Bypass feature avoids processing some types of PostScript input files through the Normalizer, but instead keeps the file in PostScript format until output.

The Normalizer uses its special configuration file to enable and control the PostScript Bypass functionality. There is no control in the Process Plan Editor or Workshop.

Each input file generates a “proxy” PDF page that embeds reference information used during Output. A checker-board pattern is used to indicate PostScript Bypass during viewing in Acrobat or output to non-Prinergy RIP systems.

Specifically, PostScript Bypass is intended to be an efficient means to process CT/LW pages encoded as PostScript data, as generated by a tool such as Shira CEPStoPS. This feature is not intended to bypass general types of PostScript input.

When deciding to use PostScript Bypass, please understand:

1. PostScript Bypass does not prevent errors in PostScript input, nor can it fix errors. Files containing PostScript errors may not fail during Refining; however such files will certainly fail during Output. It is highly recommended to generate proofs prior to making plates, to insure the validity of the input files.
2. Soft-proofing with Acrobat is useless, as no page content will be visible. Output to VPS will still function normally, and is the recommended way to soft-proof the page content.

3. Color matching, Trapping, and Delta output functions are not compatible with the "proxy" PDF pages. These functions cannot be used with pages that employ PostScript Bypass. These functions will continue to work with pages that do not use PostScript Bypass.

4. PostScript Bypass does not perform any checks for fonts, spot colors, etc. PostScript Bypass does not perform OPI during Output.

5. PostScript Bypass does not optimize images, nor does it take advantage of the performance improvements for Copydot data available in Extreme. PostScript Bypass of Copydot pages will consume significantly higher amounts of memory during Output than non-Bypass cases.

6. PostScript Bypass cannot specify specialized screening. Pages that employ PostScript Bypass will only use the screening specified in the Output Process Plan. Attempting to defeat this will probably result in unsatisfactory screening, and is not recommended.

By default, the Prinergy system does not enable the PostScript Bypass feature. Users must enable this feature explicitly, and must be aware of the implications of doing so.

There are some limitations to be noted regarding PostScript Bypass:

- Re-refine of the proxy will fail. Do not re-refine a PDF page containing a PostScript Bypass proxy.
- Do not re-use PDF pages containing a PostScript Bypass proxy. Registration will fail. Refine the original input file again to generate a new PDF page.
- It is not recommended to mix composite and separated files in the same job when using PostScript Bypass.
- Be sure to apply Prinergy 1.1 Build 159 Patch 2.

7. POSTSCRIPT BYPASS – HOW IT WORKS

The PostScript Bypass feature is controlled by the Normalizer. It is applicable to PostScript input files of EPS or DCS only. Multi-page inputs cannot use this feature; errors will result if attempts to use this feature with such multi-page inputs.

When the Normalizer detects an input file qualifies for PostScript Bypass, it does the following:

1. Copies the input file(s) to the System\BypassParts folder of the Job Folder (the location on disk where the PDF pages and other job-related data is stored).

2. Creates a "proxy" PDF that contains references to the data files now stored in the System\BypassParts folder. This proxy should not be moved, renamed or otherwise altered. It should not be moved to another job (it will fail to work in other jobs).

During Output, the "proxy" PDF locates the System\BypassParts folder, and retrieves the PostScript portions for generating raster. These "proxy" PDF pages can be mixed with normal PDF pages, each type gets rasterized correctly based on its input type.

8. RECOMMENDATIONS FOR USING POSTSCRIPT BYPASS

Unless a specific configuration is desired, please consider this example as a good starting point. It involves setting up an extra Normalizer with bypass enabled, while leaving the others with it disabled. Next, set up 2 sets of process plans: one set that references the bypass-enabled Normalizer, the other that references the bypass-disabled Normalizers. Although there is a slight penalty associated with having an extra Normalizer in the system, the advantage is that you can switch between bypass-enable and disable without reconfiguring anything or starting or stopping a Normalizer or altering the registry or configuration file.

Details on how to set it up:

1. Create a new Normalizer called Bypass.
2. Create a set of process plans that reference the Bypass Normalizer (put the word *Bypass* in the Process Plan name, or put them in a group called *Bypass*). Simply edit any existing plan, change the JTP to point to the Bypass JTP and save the Process Plan with the same name, but add Bypass to it. For example, edit *1stRef-Normz* and save it out as *Bypass1stRef-Normz*.
3. Refine any file with one of the new process plans -- this is required to flesh out the registry that's created for a new JTP, in particular to create a key that's used to point to the **Normalizer Input Configuration** file.
4. Create a **Normalizer Input Configuration** file that contains the bypass targets of relevance to your site and call it *NormInputConfigBypass*. Copy this file to the folder %ARAXIHOME%\AdobeExtreme\bin. If you are unsure about the content of this file, you can copy the existing *NormInputConfig* file and edit that copy.
5. Change the value of the following registry key for the newly created JTP to point to your **Normalizer Input Configuration** file:

   Hklm\software\creo\araxi\build\default\enor\Bypass\Ps Input Config File

   FROM: NormInputConfig

   TO: NormInputConfigBypass:

Now you can refine a file with bypass enabled by selecting the process plan called *Bypass1stRef-Normz* or bypass disabled by selecting *1stRef-Normz*.

This is just the mechanics of how to set up one configuration. You should tune the naming and grouping of Process Plans and JTPs (and Pools) to the customer's workflow.

If their predominant workflow will use PostScript Bypass, and they will only occasionally not use the non-PostScript Bypass Normalizer, then you could set up all Process Plans and JTPs to use PostScript Bypass as the default. Leave the Process Plan names untouched as they come out of the box. Create one JTP called BypassDisabled and a couple Process Plans that would go in the *Troubleshooting* Process Plan group.

9. FURTHER INFORMATION ON THE CONFIGURATION FILE

9.1 Default NormInputConfig File Included with Prinergy 1.1

```
#  NormInputConfig
#
#  This configuration file controls the behavior of the PNormalizerJTP
#  application for certain specific input PostScript files.
#
#  Each specific configuration starts with the keyword "target", and
#  is followed by one or more PostScript comment comparators. Four
#  different comparisions can be made, using this syntax:
#
#       =          exact text match
#       ~=         text contains
#       !=         text does not match
#       !~=        text does not contain
#
#  All the PostScript comment comparators must succeed for target match.
#  After the PostScript comments, specific attributes can be assigned,
#  from this list: @pagebox, @trimbox, @bypass, @bypassopi, @pscode,
#  @cleanmultipage.
#
#  Target matching is attempted from the top of this file. Comparisions
#  stop when finding the first match.
#
#  Caution: It is not advised to make changes to this file, erroneous
#  results are possible.
#
#  Copyright (c) Creo Products Inc, 1999 All Rights Reserved.
#

#  Imation Prepress XTension, for use with Quark XPress 3 or 4

target Imation Prepress XTension
%%Creator: = Imation Prepress XTension
@pagebox bbox -18 -18 +18 +18
@trimbox bbox 0 0 0 0
@cleanmultipage

#  Creo Renaissance Scanner, EPS or DCS output

target Creo Renaissance Scanner
%%Creator: ~= Creo
%%Creator: ~= Renaissance
@pagebox bbox -18 -18 +18 +18
@trimbox bbox 0 0 0 0

#  Creo Copydot Toolkit, EPS or DCS output

target Creo Copydot Toolkit
%%Creator: ~= Creo
%%Creator: ~= Copydot
%%Creator: ~= Toolkit
@pagebox bbox -18 -18 +18 +18
@trimbox bbox 0 0 0 0

#  End of NormInputConfig
```

9.2 Adding PostScript Bypass for Shira CEPStoPS

Add the following lines to the NormInputConfig file:

```
#  Shira CEPStoPS, Normalize with Bypass

target Shira CEPS2PS CT/LW Bypass
%!PS-Adobe-3.0 = EPSF-3.0
%%Creator: = Shira CEPStoPS
%%BoundingBox: ~=
@bypass
```

When creating output files from the Shira CEPStoPS product, use either the EPS or DCS output function. Do not use the other options. See the Shira CEPStoPS manual for further information if necessary.

9.3 Troubleshooting

Enabling the extra debugging for the Normalizer can often be a useful way to see why your input files are not matching the given target description. This can be enabled in the registry by changing the key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\CREO\ARAXI\Build\DEFAULT\
      ENOR\ENOR_diagDebug
```

where *Build* will vary according to your installation. Change the value to 2 and restart the Normalizer JTP. The diagnostics log will trace the loading and processing of the configuration file during each job.

The diagnostic log will always contain information about the activation of the PostScript Bypass feature. It will provide the information pertaining to the target match found from the configuration file.

Be aware that PostScript Bypass will mask the errors in the input files until output. Always verify that the files successfully Normalize without PostScript Bypass enabled when troubleshooting specific problematic input files.

Be aware that PostScript Bypass does not function with every flavor of input file. It is not recommended to use this feature in production with untested input data types. Always test carefully and compare the results obtained with PostScript Bypass disabled.