Appendices

English

Brisque

version **4.0**

www.creo.com | creo

Table of Contents

A PostScript and PDF-Related Issues 1

Application Note #1 2
The Error Message Format 2
Locating the Source of an Error 2
Typical Error Messages 3
Application Note #2 7
Application Note #3 8
Application Note #4 11

B The PS/PDF Utils Folder 13

A

PostScript and PDF-Related Issues

Application Note #1 2
Application Note #2 7
Application Note #3 8
Application Note #4 11

Application Note #1

Understanding PostScript Error Messages

PostScript is a page description language as well as a computer programming language. This means that PostScript is an open-ended system with a very large range of possible combinations and constructs. Its error messages describe very technical aspects of a PostScript program execution, but this usually means very little for the non-programmer user.

The same error message may have completely different causes, thus complicating the task of describing the possible solution.

Given all this background, it is still useful to list some of the more common error messages and their possible causes. Please remember that there are no fixed, guaranteed answers.

The Error Message Format

PostScript errors have a standard structure which is composed of two parts:

- The first part identifies the error.
- The second part describes which PostScript command caused this error.

See the following example:

%% [Error: <error name>; OffendingCommand: <bad command>] %%

When you are trying to identify a problem, or reporting errors to Creo, it is important to note both parts of the error message.

Locating the Source of an Error

With other PostScript devices it is quite difficult to locate the source of the PostScript error. Fortunately, the Brisque has a unique Preview operation of PostScript files that draws one element of the file after the other. By previewing a bad PostScript file you can quickly locate the element that caused the error.

The common sources of potential problems fall into two categories:

- Limit stretching
- Corrupted or erroneous PostScript.

Limit stretching:

- Very complex graphic elements such as color blends or gradations, complex clipping paths (masks), excessive nesting (Paste Inside) and very complex fill patterns.



Note: Fill patterns from Freehand and Illustrator are not supported by the Brisque.

- Very large pages with lots of images and graphics.
- Pages with many fonts.

Corrupted or erroneous PostScript:

- Fonts of suspicious origin.
- Fonts that were converted from other formats.
- Nested EPS images (created by importing an EPS element to a page then saving the page as an EPS).
- Files that were corrupted by going through multiple communication channels and/or magnetic media.

Typical Error Messages

This section lists some error messages and some of their common causes.

dictstackunderflow

The PostScript code has an unmatched end command. Usually this means that the file is corrupted.

invalidfont

Corrupted font. First check if this font is embedded in the PostScript file or located in one of the font folders. If it is in one of the folders, you can try and replace it with a 'fresh' copy. If it is embedded, you may need to use a different font. If the font is a composite font (such as Japanese or Chinese), which is protected, you may have a problem with the protection mechanism. Verify that you have a legal copy of the font.

ioerror

This may be an actual disk fault, or some other hardware problem. It also may indicate file corruption in PostScript Level 2 files that contain filtered data such as compressed images.

limitcheck

The PostScript interpreter reached an internal limit. This limit may be an architectural limit, such as a file name that is too long or that is located in a deeply nested folder. The limit may be memory, disk space or buffer size. In some cases this problem can be solved by increasing the memory allocated to the RIP.

rangecheck

Some parameter in the PostScript program was out of the acceptable range for the specified offending command. In some cases this is caused by file corruption. The offending command may indicate the problematic element. (For example, **image** or **colorimage** indicate a scanned image.)

syntaxerror

The PostScript code is bad. This is most likely due to a corrupted file.

typecheck

A PostScript command was expecting parameters of one type, but got a different type. Usually this is caused by a corrupted file.

- If the offending command is **aload**, it may be caused by a compatibility problem with LaserWriter 7.x. Try using a later version.
- If the offending command is **get**, it may be caused by bad DOS PostScript code. Try selecting the multiple page option in the DOS application, and recreate the PostScript file.

undefined

The PostScript interpreter encountered an unrecognized command. This usually indicates file corruption, but it may be tracked to a more specific reason:

- If the offending command is composed of garbage characters (such as `$#@†®ƒb8`), it may be caused by a corrupted image (TIFF or other scanned image). Use **Preview** to locate the bad image and try replacing it. Also try saving the file in ASCII rather than BINARY format. (This is an option in the **print** dialog box when you **print to file**).
- If the offending command is composed of hexadecimal numbers (hexadecimal numbers are composed of the usual digits 0-9 and characters A-F), this may also be a corrupted image that was saved in ASCII. Use **Preview** to identify the corrupted image. In rare cases it can indicate a corrupted font. Try locating the bad font and replacing it.
- If the offending command is `@PJL` or `%-12345X`, this file was intended for a non-PostScript printer such as a Hewlett-Packard printer. Select the correct driver in your Windows Printer Control panel and print again.

undefinedfilename

The PostScript interpreter could not find the specified file. This may be caused by a bad PSImage element. Check the name of the high resolution file. In some non-Latin environments (such as Arabic, Japanese etc.) certain file names may cause this error.

undefinedresource

The PostScript program is looking for some PostScript resource that is not found. PostScript resources can be Fonts, ProcSets, ColorRendering dictionaries and Forms. Normally, this should not happen.

undefinedresult

The PostScript interpreter encountered a calculation that cannot be made mathematically. For example, division by zero, or calculating the square root of a negative number.

In some cases the division by zero is caused by a bad scaling. This can happen when you use a scale factor or a resolution that is close to zero.

VMError

The PostScript interpreter ran out of VM memory. VM memory is the part of memory used by the interpreter to store strings, arrays, dictionaries and fonts. This problem can be solved by increasing the memory allocated to the RIP, using lower resolution, simpler files, or by separating a multi-document file into several sections.

Application Note #2

Reporting PostScript Problems

Before reporting PostScript problems to the support group, try simulating or reproducing the problem by following the procedures described in *Application Note #4* on page 11. If you still cannot isolate the problem, prepare the following information before contacting your local Creo Care Center.

The following details are necessary:

- Platform
- Operating system
- Application Version
- LaserWriter/Printer driver
- RIP version
- RIP parameters (resolution, overprint, spot or process color, and so on)
- A short description of the workflow
- If APR/OPI is used, a description of how the **.e** and/or **.lay** files were created.
- Displayed error messages and other symptoms.

In addition to the above, the PostScript and application files including all fonts, images, and graphics are necessary. Also the Job Ticket, if possible. This enables a simulation of the problems and a thorough investigation of the finer details of the intended workflow.

For more information, see also *Application Note #1* on page 2.

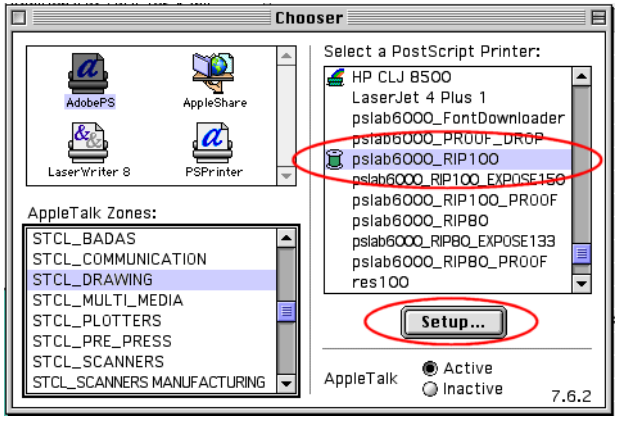
Application Note #3

Creating PostScript Files on a Macintosh

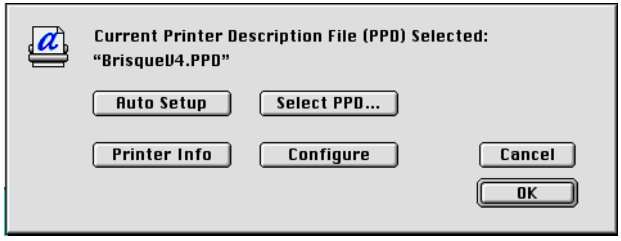
Creating PostScript files is generally a straight forward process, as long as you know which parameters to pay attention to. Below are general guidelines for printing PostScript files, and for printing to disk. These procedures should be used across the board with Macintosh desktop publishing applications. Look for other technical notes concerning specific applications.

To print a PostScript file:

1. On the Macintosh, open the **Chooser**.
2. Select a published printer and click **Setup**.



The following dialog box appears.



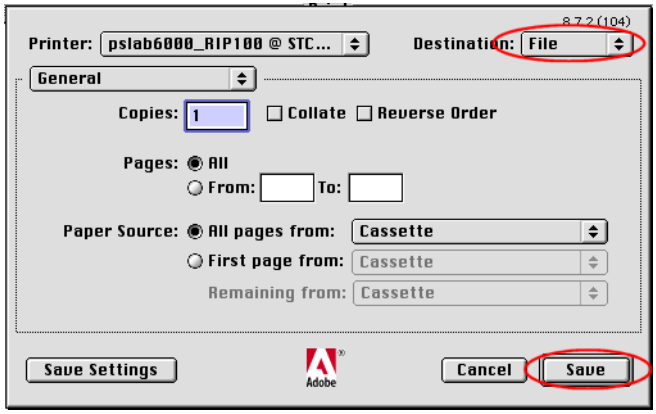
3. Click **Select PPD**; a browser appears, displaying the contents of the **Printer Description** folder.

Note: If the browser does not automatically display the contents of the **Printer Description** folder, navigate to it. It is located in the **Extensions** folder found in the **System** folder.

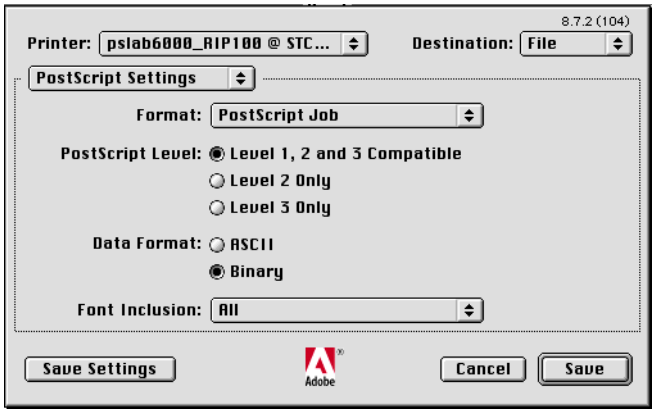
4. From the **Printer Description** folder, select the PPD that best matches your output device, then click **Select**; you are returned to the previous dialog box.
5. Click **OK** to confirm the selection and return to the **Chooser**.

To print to disk:

1. From the **File** menu, select **Print**; the Print dialog box appears.
2. In the Print dialog box, from the **Destination** menu, select **File**; the **Print** button becomes the **Save** button.



3. From the **General** menu, select **PostScript Settings**; the following parameters appear.



4. From the **Format** menu, select **PostScript Job**.
5. In the **PostScript Level** section, select **Level 1, 2 and 3 Compatible**.
6. In the **Data Format** section, select **Binary**.

Note: Binary files are smaller than ASCII files, take up less disk space, and cause less traffic on networks.

7. From the **Font Inclusion** menu, select **All**. All printer fonts will be included in the PostScript file.

Note: Including the printer fonts results in a slightly bigger PostScript file (average font size is 30K), but prevents the problems that could occur during RIP because of missing fonts.

8. Click **Save**.

Application Note #4

Reproducing PostScript Errors

Below is a set of procedures that should be performed before sending the file to your local Creo Care Center. Use these procedures to determine whether the PostScript error is a bug (which only happens on a Brisque RIP), and if so, to try to isolate and identify the error.

To try to resolve a PostScript error, try the following:

- RIP the file on a non-Creo RIP such as a PostScript Laser Printer. You can try to distill the file in Adobe Distiller and then try to open it with Acrobat Reader. If this does not work then the problem is with the file itself, and not with the RIP.
- Try RIPping the file on a different Creo RIP (such as, PSM).
- Try and preview the file on screen. The Brisque **Preview** is an excellent trouble shooting tool. An error message prompts you when **Preview** reaches an erroneous object. Remove this object from the file, recreate the PostScript file and try to RIP it again.
- Try to RIP the file at different resolutions and different page sizes. Also try to RIP it using the **SetPage** parameters then try to RIP it using the **Bounding Box** parameters.
- Try to save the PostScript file to disk using different parameters (ASCII/Binary, Level 1,2,3, and EPS.)

For more information, see *Application Note #1* on page 2, and *Application Note #2* on page 7.

B The PS/PDF Utils Folder

The **PS/PDF Utils** folder of the **Brisque 4.0 Accessories CD** contains helpful utilities that can assist you in improving your productivity.

Following is a list of some of the utilities provided in the PS/PDF Utils folder:

- **For Acrobat:**
  - ❒ Enfocus PitStop
  - ❒ Creo FAF Plug-In for Acrobat
  - ❒ Creo Spot Colors Plug-In for Acrobat.
- **For Photoshop:**
  - ❒ PSImage Exporter
  - ❒ Creo NewCT Plug-In.
- **For QuarkXPress:**
  - ❒ Creo Launchpad XT
  - ❒ Creo Spot Colors XT
  - ❒ Creo Print XT
  - ❒ Creo TIFFixer XT
  - ❒ Creo Versioning XT.
- **PPDS:**
  - ❒ 4 PPDs for Macintosh
  - ❒ 2 PPDs for PC.