Adobe Bugs, affecting LaTeX for scientific typesetting

TFAA working group, TeX Users Group

Latest version: May 5, 2004

If you encounter a bug using Adobe software that affects how you need to prepare images for use with TeX, LaTeX or related software, then please do the following:

- check whether it is listed below; follow the link to obtain a more detailed description, with examples;
- if your bug is similar to one listed already, but occurs on a different platform or with a version of Adobe software not stated in the description, then please report it stating: 'Same as bug ⟨bug-ID⟩' quoting the identifier found on the description page.
- if it is a completely different 'bug', then report it giving both a short description and a more detailed description. The latter can be a link to a website or archive where sufficient details, with examples, can be obtained that characterize the 'bug' and how it affects your work. Please describe also any methods that you have found to 'work around' the bug/deficiency.

Bug report site: http://www.cds.caltech.edu/WARM/adobe/tfaa/ (temporary)
(This site will move soon to a location hosted at: http://www.tug.org/.)

Bug List:

Bug: *Illustrator* puts wrong MediaBox into saved PDF 2
Bug: *Acrobat 5* views fine, but substitutes fonts upon printing 4
Status: fixed 4
Suggested Fix/Workaround 4
Bug: *Illustrator 10* freezes on GnuPlot figures with labels 5
Bug: *Illustrator 10* incorrectly rotates text labels on GnuPlot figures 6
Bug: *Illustrator 8–10* puts wrong byte count for hex thumbnail 7
Suggested Fix/Workaround 8

Bug: *Illustrator* puts wrong MediaBox into saved PDF

When *Illustrator 9 & 10* save artwork as PDF, the `/MediaBox` is set to the size of the canvas (i.e. to the full page) rather than the natural bounding-box for the artwork.

Software Versions

- *Illustrator 9* and *Illustrator 10* and *Illustrator CS*
- MacOS 9.1, MacOS X, and other platforms.

Reported by: Ross Moore, 11 January 2002; revision and work-arounds added, 5 May 2004.

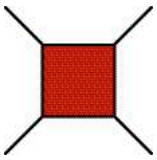
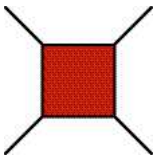
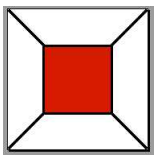
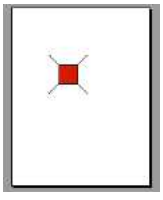
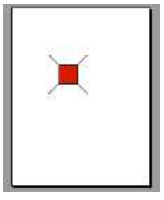
This is arguably not a bug at all, since a correct bounding-box is deducible from the `/ArtBox` stored in the PDF. Yet we report it as a bug, because:

- the PDF created by Save As Acrobat PDF is visibly quite different from that created by the sequence: Save As Illustrator EPS followed by *Acrobat Distiller*;
- it renders useless a document created using Save As PDF for importing as an image into other word-processing documents, when the importing application is unaware of the `/ArtBox` information, or cannot access it.

Certainly a user cannot get the desired result in the usual way for importing images, but must search for extra import options, or pre-process the image using other software (see the work-arounds belowBug:).

Example

(Click on the file-name to download the corresponding file.)

Illustrator artwork	...saved as EPS...	...then 'distilled' to PDF
 `bugtest1.ai` (size = 236k)	 `bugtest1.eps` (size = 352k)	 `bugtest1_dist5.pdf` (size = 4.8k)
	...saved as PDF (Acrobat 4)	...saved as PDF (Acrobat 5)
	 `bugtest1_ar4.pdf` (size = 5.0k)	 `bugtest1_ar5.pdf` (size = 5.1k)

Note that the two ways to get a PDF file give different results.

Further Information

- For File Compatibility, the options Acrobat 4 (PDF 1.3), Acrobat 5 (PDF 1.4) Acrobat 6 (PDF 1.5) show no difference.

Suggested Fix

Either . . .

> When selecting Save As Acrobat PDF set the `/MediaBox` to be the size of the minimal rectangle surrounding the visible artwork.

Or . . .

> Include an option to allow the user to specify whether to set the `/MediaBox` to include just the visible artwork (`/ArtBox`) or leave it as the whole page. Use a check-box for this choice.

Work-arounds

Here are some ways to work with PDF images that exhibit this problem.

Crop the image. With a program such as *Acrobat* (Full version) or *Acrobat 6 Professional*, there is an option under the Document menu (check the submenus!) which allows a page to be cropped.

... for pdf-TEX: Use coding in the document preample:

```
\expandafter\ifx\csname pdfoptionalwaysusepdfpagebox\endcsname\relax
   \pdfoptionalwaysusepdfpagebox 5 %
\fi
```

This works since pdf-TEX detects and records various boxes: `/MediaBox`, `/CropBox`, `/BleedBox`, `/TrimBox`, `/ArtBox` that may have been recorded in the PDF file. The `/ArtBox` is #5 in this list.

Bug: *Acrobat 5* views fine, but substitutes fonts upon printing

Some PDF documents created by some versions of pdf-TEX view correctly on-screen in *Acrobat 5.00* and *Acrobat Reader 5.00*, but Courier is substituted for some of the Computer Modern fonts when the job is printed.

Software Versions

- *Acrobat 5.00* and *Acrobat Reader 5.00*
- all platforms

Reported by: Tom Kacvinsky, April 2001.

This bug has been known since early 2001, when *Acrobat 5* was released. Earlier versions of *Acrobat* and *Acrobat Reader* did not exhibit such font substitutions. The cause was diagnosed, by Tom Kacvinsky and others, as being due to 'gaps' in the Subrs array of the subsetted fonts for which substitution occurs; the CoolType font technology, introduced for version 5 of the *Acrobat* software, rejects fonts with these gaps.

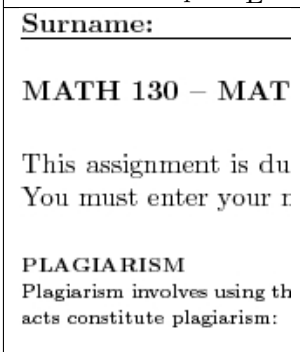
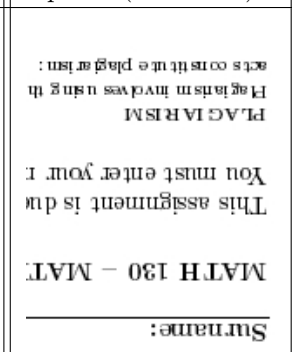
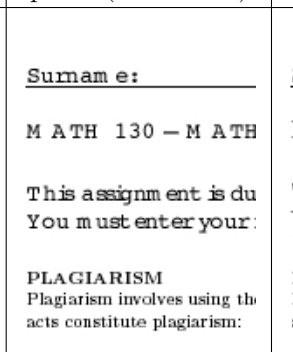
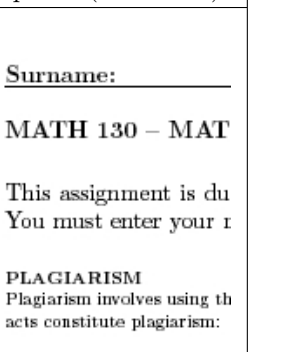
Tom K. devised a patch to pdf-TEX so as to avoid this problem, yet many legacy documents remain.

Status: fixed

Adobe fixed the CoolType engine, for *Acrobat 5.05* and *Acrobat Reader 5.05*.

Example

(Click on the file-name to download the corresponding file.)

created with pdf-TEX	print: (Reader 4.05)	print: (Acrobat 5.00)	print: (Reader 5.05)
			
Questions.pdf (size = 80k)	Questions_405.pdf (size = 56k)	Questions_500.pdf (size = 67k)	Questions_505.pdf (size = 53k)

The first file here was created using pdf-TEX. Others come from capturing a (PostScript) print-job into a file, then processing with *Acrobat Distiller* (version 5.0) on a Macintosh.

Note that the page was rotated by 180° when printed from *Acrobat Reader 4.05a*; this is due to an unrelated bug!

Further Information Versions of pdf-TEX that produce suspect PDF files are: 0.14e, 0.14f .

Suggested Fix/Workaround

Adobe: No further action required.
Users: upgrade to Acrobat 10.0, or later.

Bug: *Illustrator 10* freezes on GnuPlot figures with labels

Some EPS documents created by GnuPlot, containing text labels, open fine in *Illustrator 9.02*. However, when attempting to open such files with *Illustrator 10* (Macintosh, MacOS 9.1) the application appears to freeze; no picture is drawn, and there is not even a progress-bar. Now 'force-quit' fails (separate bug, reported elsewhere!) so a reboot is required.

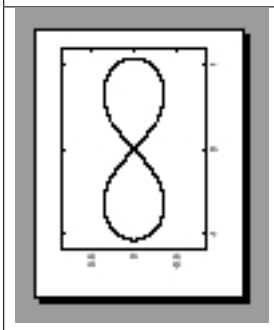
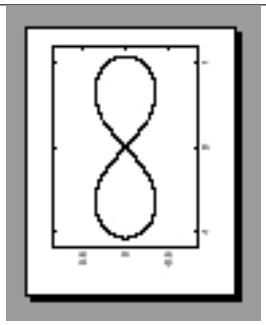
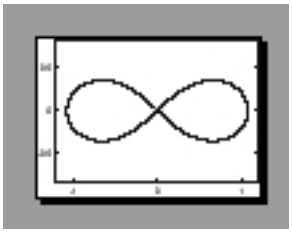
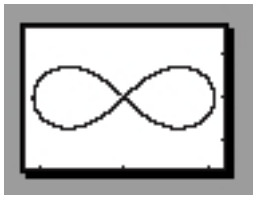
Software Versions

- *Illustrator 10*
- MacOS 9.1

Reported by: Ross Moore, 24 January 2002.

In fact, on a subsequent attempt to open the file, *Illustrator 10* was finally successfully, after 12–15 minutes. Subsequent opening, in the same session, of this and similar files were successful immediately. This errant behaviour still constitutes a 'bug'. (See also Bug: ILL10-2002-01-24b.)

Example (Click on the file-name to download the corresponding file.)

created with GnuPlot	distilled to PDF	resaved: (Illus 9.02)	resaved: (Illus 10)
			
`fig1.ps` (size = 12.5k)	`fig1.pdf` (size = 12.2k)	`fig1-ill9.eps` (size = 192k)	`fig1-ill10.eps` (size = 364k)

The file could be opened by *Illustrator 10* only after removing the lines placing text-labels.
The original EPS file has Bounding-Box and Orientation comments as follows:

```
%%BoundingBox: 50 50 554 770
%%Orientation: Landscape
```

Both *Illustrator 9* and *Illustrator 10* rotate the artwork, but the Page Setup has to be adjusted to suit. (Note that *Acrobat Distiller* cannot respect the orientation comment.)

Further Information The GnuPlot-produced EPS file handles axis-ticks with labels using PostScript coding such as the following, taken from the example above:

```
/M {moveto} bind def  /L {lineto} bind def
/R {rmoveto} bind def /V {rlineto} bind def
/vshift -80 def
/Lshow { currentpoint stroke M 0 vshift R show } def
/Rshow { currentpoint stroke M dup stringwidth pop neg vshift R show } def
/Cshow { currentpoint stroke M dup stringwidth pop -2 div vshift R show } def

(Helvetica) findfont 240 scalefont setfont
-63 0 V  -6129 0 R
(-0.5) Rshow
```

and usages of `Lshow` and `Cshow`, similar to that of `Rshow`. When all lines using these operators are commented-out, *Illustrator 10* opens the EPS file successfully. (Perhaps the problem is in finding the appropriate font, when it needs to be used.)

Bug: *Illustrator 10* incorrectly rotates text labels on GnuPlot figures

Some EPS documents created by GnuPlot, containing text labels, display correctly in *Illustrator 9.02*. However, when opening such files with *Illustrator 10* (Macintosh, MacOS 9.1) the line-art portions of the artwork is correctly rotated to landscape, but positions for text-labels, and the labels themselves, are rotated twice, hence through 180°.

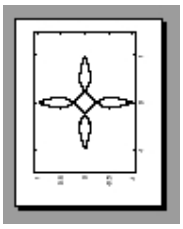
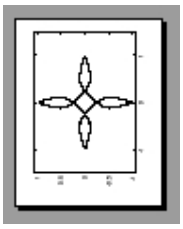
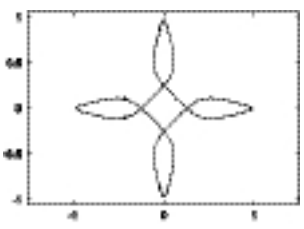
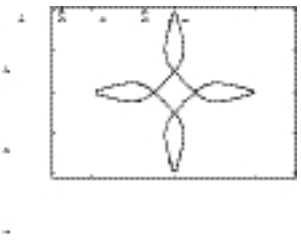
Software Versions

- *Illustrator 10*
- MacOS 9.1

Reported by: Ross Moore, 24 January 2002.

EPS files exhibiting this bug may also be subject to excessive time taken to open the file, or failure to open at all. (See Bug: ILL10-2002-01-24a.)

Example (Click on the file-name to download the corresponding file.)

created with GnuPlot	distilled to PDF	resaved: (Illus 9.02)	resaved: (Illus 10)
			
`fig2.ps` (size = 13.9k)	`fig2.pdf` (size = 13.1k)	`fig2-ill9.eps` (size = 204k)	`fig2-ill10.eps` (size = 392k)

Note how with *Illustrator 10* the non-text portion of the artwork has been rotated to landscape, but the text-labels and their relative positions have been rotated by 180°. The original EPS file has Bounding-Box and Orientation comments as follows:

```
%%BoundingBox: 50 50 554 770
%%Orientation: Landscape
```

Both *Illustrator 9* and *Illustrator 10* rotate the artwork, but the Page Setup has to be adjusted to suit. (Note that *Acrobat Distiller* cannot respect the orientation comment.)

Further Information GnuPlot-produced EPS files handle axis-ticks with labels using PostScript coding such as the following, taken from the example above:

```
/M {moveto} bind def  /L {lineto} bind def
/R {rmoveto} bind def /V {rlineto} bind def
/vshift -80 def
/Lshow { currentpoint stroke M 0 vshift R show } def
/Rshow { currentpoint stroke M dup stringwidth pop neg vshift R show } def
/Cshow { currentpoint stroke M dup stringwidth pop -2 div vshift R show } def

864 1599 M 63 0 V 5985 0 R
-63 0 V -6129 0 R
(-0.5) Rshow
```

and usages of `Lshow` and `Cshow`, similar to that of `Rshow`.

Bug: *Illustrator 8–10* puts wrong byte count for hex thumbnail

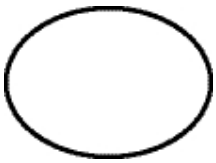
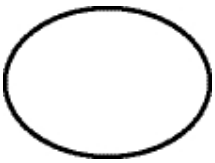
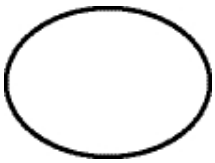
Illustrator 8.0 & 9.x & 10.0 generate an invalid DSC comment at the top of EPS files; this causes dvips to reject these files when *comment elision* (-K) is turned on. Essentially, the byte count for the hex thumbnail (as expressed in the %%BeginData comment) is incorrect, so does not conform to the DSC specification.

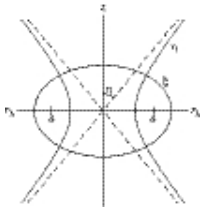
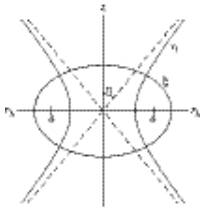
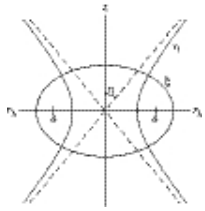
Software Versions

- *Illustrator 8* and *Illustrator 9*
- all platforms

Reported by: Tomas Rokicki, 1 February 2002.

Example (Click on the file-name to download the corresponding file.)

created with (Illus 8.0)	resaved: (Illus 9.02)	resaved: (Illus 10)
		
rokicki/sample.eps	rokicki/sample-ill9.eps	rokicki/sample-ill10.eps
%%BeginData: 4312 Hex Bytes	%%BeginData: 4314 Hex Bytes	%%BeginData: 4466 Hex Bytes

created with (Illus 9.0)	resaved: (Illus 9.02)	resaved: (Illus 10)
		
rokicki/sphcoor.eps	rokicki/sphcoor-ill9.eps	rokicki/sphcoor-ill10.eps
%%BeginData: 6956 Hex Bytes	%%BeginData: 7244 Hex Bytes	%%BeginData: 7052 Hex Bytes

The changes in the number of Hex Bytes, when the original EPS file is resaved with a later version of *Illustrator* , indicates actual differences in the thumbnails produced by these different versions.

In each case, the number stated is 2× the count of 'virtual' data (see below); hence agrees with the count of 'physical' data, but only after ignoring the white-space (line-ends) and 'comment' characters (%) preceding each line.

Further Information Tomas writes as follows:

This note is to report a bug in *Illustrator 8.0 and 9.0*; its use of %%BeginData does not conform to the DSC specification.

Specifically, in the thumbnail section at the top of the document comments section, it includes comments such as %%BeginData: 4312 Hex Bytes
According to the specification (PostScript Language Reference Manual, Second Edition, page 648, describing the %%BeginData DSC):

> "Note that ⟨numberof⟩ indicates the bytes of *physical* data, which vary from the bytes of *virtual* data in some cases. With hex, each byte of *virtual* data is represented by two ASCII characters (two bytes of *physical* data). Although the PostScript interpreter ignores white space in hex data, these count toward the byte count.

> "For example, `FD 10 2A 05` is 11 bytes of *physical* data (8 bytes hex, 3 spaces) and 4 binary bytes of *virtual* data."

I am enclosing two such broken files sent to me by two of my users (of dvips). In both cases, the byte count seems to be the total number of valid hex characters, not including the extraneous % character that *Illustrator* inserts at the beginning of the line, nor the line terminators. A simple analysis of the file shows that, in the data section, we have:

Total bytes: 4531
Hex: 4312 White: 146 Misc: 73

So the DSC comment *should* be: `%%BeginData: 4531 Hex Bytes`

The example on the following page (649) clearly indicates that the byte count, even for hex, should include *all* characters (the example they give includes an image operator, and they specifically mention that all five characters of "image", plus the newline following it, must be included in the byte count).

The impact on my users is that, for a very widespread version of dvips , they cannot include *Illustrator* files when they are using certain combinations of options. I am currently testing a workaround for the next release of dvips , but clearly *Illustrator* must be fixed.

I am always happy to help in any way, so contact me at rokicki@cs.stanford.edu if you have any further questions, or if there is any way I can be of help.

Thanks!

Tomas Rokicki

Suggested Fix/Workaround

Adobe: either follow the published DSC specification, or rewrite this specification to agree with what actually is done by the software.

Users:
Tomas has a patch to dvips 5.86e that works around this problem, for the next release of dvips .

If you have broken EPSF files, Tomas has written a Perl script that may fix them. If your system insists on executing scripts with the PL extension, you can also get it as text.