

## Bug Fixes

**ID :** 1408**Fixed in version :** 3.2.4**Short Description:** PDF Device:print PDF document on printer, from OSX Preview**Full Description:** I create a PDF file from Omnis with component Device and open the PDF with OSX Preview; then I send to printer the PDF document and I get an unreadable document.

I get the same result if I export, from OSX Preview, on PDF.

I attach the printed PDF Document (printed\_pdf\_doc.pdf) and the exported PDF Document from OSX Preview (exported\_pdf\_doc.pdf).

I have tried to set font embedding property but the result is the same.

I have tried with your example library and with my software, on Omnis 5.1.0 with PDF Device 3.2.2 and PDF 3.2.1R.

**Comments :** Duplicate: see case 1402

## Bug Fixes

**ID :** 1402

**Fixed in version :** 3.2.4

**Short Description:** Problem with program Preview to display PDF

**Full Description:** If I display the PDF created with pdfdevice with the "Preview" program it's okay but if I print the PDF from the program or do "Export as PDF" file is wrong and displays squares instead of characters. Other PDF files not created with PDF Device haven't problems.

**Comments :** It appears that our PDF generation tool contains some legacy code that includes the font's POST table when embedding true-type fonts. This POST table had become obsolete when we changed our PDF generation tool from producing MacRomanEncoding or WinAnsiEncoding to specify "no encoding" to enforce a direct symbolic lookup of character indexes to Glyph data using the embedded CMAP table. According to PDF Reference version 1.4 (third edition) by Adobe Systems, this is a valid strategy which helped us to support and optimize the now mainly unicode based software implementations that use our tools internationally in markets where MacRomanEncoding or WinAnsiEncoding makes no sense.

Our interpretation of the PDF Reference suggests that when no encoding is specified, readers are to use primarily the CMAP table for mapping character indexes to Glyph data for painting operations and use the unicode mapping table associated with the embedded font for exporting text. I quote "If no Encoding entry is specified in the font dictionary, the 'cmap' subtable with platform ID 1 and encoding 0 will be used to map directly from character codes to glyph descriptions, without any consideration of character names." (PDF Reference version 1.4, p.334). This suggests that the POST table, although present, is to be ignored when no encoding is specified. This is confirmed by various operating systems and readers handling our embedded fonts as described above without adverse effects when printed.

We have forwarded these findings to Apple and we continue to hope that they will change their interpretation for the sake of legacy documents. However, legacy documents can be printed successfully using any Adobe Reader.

**ID :** 1411

**Fixed in version :** 3.2.3

**Short Description:** Error in generating PDF document (rotated text)

**Full Description:** We are trying to directly generate a PDF Invoice from an Omnis 5.2 application through your PDFDevice software. The invoice report contains two non-horizontal Omnis text objects, the first programmed to print at a 45-degree angle, the second at a 270-degree angle.

When we preview the document or print in on paper from the Preview page, everything works fine. Also, if we print the invoice from the Preview page through Acrobat 9.0, the PDF document so generated is perfect, too.

However, when we try to generate the same invoice directly through the PDFDevice XCOMP, both on Mac and Windows computers, the two text objects are truncated. The second object (two long vertical lines across the page, from top to bottom) seems to get truncated just at the same point the next section starts.

We are enclosing screen dumps (with marks) and real PDF documents, somehow self-explanatory, which exemplify the problem. We hope they will help you to find the origin of this incident!

Thanks in advance!

**Comments :** There were two independent issues. Firstly text that use kAngle270 are clipped to their page section

(i.e. when placed in the report or header section they are clipped to the section's boundary). When placed in a subtotal header or record section such text is clipped to the global page space which means they do not appear clipped in this example. Clipping of page headers was part of the original external specification and our investigation has shown that this still applies except for angled text. We have changed the clipping for angled fields in headers. Secondly, fields that use kEscAngle with a user value rather than a constant appear to be clipped to the boundary of the field itself. This was caused by inconsistent behaviour by Omnis as this clipping was specified by the object, but Omnis devices appear to ignore this clipping so we changed our code to fall in line.