

Bug Fixes

ID : 1199**Fixed in version :** 3.1.0**Short Description:** Omnis Crashes after \$redirect to PDF Device

Full Description: I can reproduce this issue in Omnis 4.2 and 4.3. I was able to do so with PDF Device 2.67 and 3.00. I have reproduced it on many different computers and several different printers. Both Runtime Omnis and Development versions. Redirecting to Disk, RTF and HTML do NOT reproduce the problem. In all cases I am using Write 3.15. All tests we done on the Non-Unicode versions of Omnis and your components.

I have a fairly complex issue going on here that involves Printing, PDF Device and OWrite. (Newest version of both.)

I can consistently crash Omnis through a series of events that don't „seem% to make sense to me.

The issue is that I cannot give you an easy to reproduce example because of the circumstances. You have to test with the full version of my code. I would like to schedule a time to show you this via a remote session and see if you have any idea what could be going on. I do not believe I can explain this well enough in a support ticket to make any sense.

The Cliff Notes version is this.

Print a report (Any Report) to Screen via an Omnis Report Field.
Print the report to the printer using \$redirect(ktrue)

Open a window with an OWrite field with PDF Device report using the following code.

```
Begin reversible block
  Set current list _List
  Calculate $cdevice as kDevPdf
End reversible block
Do $cdevice.$setparam(kDevPdfConvLinks,kTrue)
Do $cdevice.$setparam(kDevPdfFileName,ccPDFfile)
```

```
Do $cinst.$objs.ReportField.$redirect(kFalse)
```

Interact with an Owrite Field in a specific way then close the window.

Attempt to open the window a second time using the same code. Omnis Crashes at the \$redirect(kFalse) statement.

```
Begin reversible block
  Set current list _List
  Calculate $cdevice as kDevPdf
End reversible block
Do $cdevice.$setparam(kDevPdfConvLinks,kTrue)
Do $cdevice.$setparam(kDevPdfFileName,ccPDFfile)
```

```
Do $cinst.$objs.ReportField.$redirect(kFalse)
```

The keys are that you actually have to Print the Report to paper first, you have to interact with Owrite and you have to redirect the PDF twice to make this happen. But once you have the steps it is very reproducible.

But I would have no idea how to create the situation outside of my application.

Comments : This issue is caused by instabilities within Omnis created by using both the Printer destination and PDF destination alternately. It is an intermittent crash which can take some time (printing multiple times to Printer and PDF) to surface. It was caused by us using a specific callback to retrieve the scaling information from the Omnis page setup record.

Removing this call prevents these instabilities. However, this results in PDFDevice no longer respecting the \$scale property of the Omnis page setup data. As an alternative we have added a new device parameter kDevPdfPrintScale that can be set to a value between 25% to 400%.

ID : 1203

Fixed in version : 3.1.0

Short Description: PDFDevice crashes printing a large number of documents

Full Description: We started using PDFDevice 3 and though it works fine, we have functionality in our application to print of a multitude of documents and email them off to clients. After printing about 3 or 4 of these PDFDevice crashes.

It's not related to the emailing because just printing them in bulk also crashes Omnis.

We went back to PDFDevice 2 and all is fine. I'm not ruling out that we've overlooked doing something required in switching to PDFDevice 3.

Comments : This was a regression in version 3 caused by the font substitution changes. It only effects reports where non-true-type fonts are used, font substitution is enabled, and three or more reports are printed that use the same non-true-type font.

ID : 1208

Fixed in version : 3.1.0

Short Description: Thick Lines in PDF Device

Full Description: The Lines are very thick in the PDF.
If i Print the same Report with an other tool, the lines are smaller.
What can i do?

Comments : We have found a fractional problem caused with pixel lines and the internal report co-ordinate system used by Omnis. Omnis stores all report objects in 1000th of a millimetre. A single pixel line on Macintosh (72DPI) translates to 352.77 units. On Windows (96DPI) the same line translates to 264.58 units. Omnis however rounds to the nearest integer and stores this value. When PDFDevice receives this info, it was not fully considering this behaviour resulting in slightly thicker lines, made worse by the in-accurate scaling of Adobe Reader on screen. This is why this subtle difference almost vanishes when printed to a 600DPI printer.

We have now further improved this situation with lines produced by OWrite, but the device option kDevPdf72DPILines must be enabled for lines to appear correctly when generating documents on the Windows platform.

ID : 1210

Fixed in version : 3.1.0

Short Description: Can't print museo font

Full Description: I've attached an Omnis print file and a PDF generated with PDF Device 2.6.7 and PDF Device 3.0.1 I did sent you the font files before some time ago.

While both complain about the font not being a true type font the PDF created by PDF Device 2.6.7 seems to at least do something useful with the font.

Ideally we would like to be able to support these fonts to some degree because we get a lot of flack from our clients who say that "only" our software can't work with these fonts. That said, if we can't support non-true type fonts, we can't support non-true type fonts.

Comments : This problem is a regression in version 3 and was caused by the font substitution changes. It only effects text that is printed with non-true-type fonts when font substitution is turned off.

Although we have resolved this issue, using non-true-type fonts is not recommended as the documents may not display correctly on platforms that do not have these fonts installed.