Release Notes PDFDevice version 3.4.3

20 Dec 2019 04:13

PDFDevice Bug Fixes

ID : 1768 **Fixed in version :** 3.4.1

Short Description: PDF device 52.3.4.0.0 only seems to generate header on windows

Full Description: hi Michael

I have PDF device 3.4 working on studio 5 & 10 (OS X - thanks, changed studio 5 to posix pathname) and on windows studio 10.

on stuido 5.2.3.1 on windows 8.1 and 10 (which is my current test platforms), PDF device prints the pdf header out to file and then seems to ignore the rest of the file.

for the two PDF's I've enclosed, one is called studio 5.pdf and the other is called studio 10.pdf

studio 5.pdf was created with pdfddevice 3.4 on wind 32 as per the part of the header I've enclosed below. This shows:
1) that its the right version of PDF device
2) it is creating a file (so path name must be ok)
3) and so I'm guessing its just not adding the actual report.

Studio 10 version has the full file.

and if I manually move the PDF device version 3.3.0.0 for studio 5.2.3 into place in the xcomp folder, I will get Pdf devices again.

hope this helps

```
<<
/Creator (Theatre Manager)
/Producer (Arts Management Systems, PDFDevice v3.4.0 Win32)
/CreationDate (D:20190905124410-07'00')
/ModDate (D:20190905124410-07'00')
```

AttachedFile: Archive.zip

Comments : It appears that the standard-c function tmpfile() which I now use does not always work on windows. Has to do with permissions. If you run Omnis as Administrator it actually works. Incredible! When it fails, I have to resort to using some windows specific functions to create a temporary file name which I can then use with the standard-c fopen() function. The downside is, I have to maually delete the file when we are done with it.

I have attached a 3.4.1 build that hopfully fixes this. Creating temp files should not be this hard.

ID : 1773 **Fixed in version :** 3.4.2

Short Description: Print special characters

Full Description: Curious problem with PDFDevice.
Some special characters, such as the diameter symbol, do not print correctly, see for example page 64 of the pdfdevice1.pdf file. This happens if the printout is very long, if instead I print only a few records, the same descriptions with special characters, are printed correctly, see pdfdevice2.pdf files. With the OmnisPDF component, it prints correctly, see Omnispdf.pdf.

AttachedFile: pdf_example.zip

Comments : There was a situation in which a character failed to map to the standard unicode table within the font, causing the cached unicode mapping table to be exchanged for the OS based single byte character mapping table. This resulted in rare cases that subsequent characters failed to map correctly.

We have changed PDFDevice so it always maintains the preferred character mapping table throughout.

This effects all platforms.

ID : 1774 **Fixed in version :** 3.4.2

Short Description: Intermittent crash

Full Description: There was a rare crash which we discovered while testing for another issue. The crash was related to some code walking of the end of a buffer by just one byte in some rare circumstances while searching for web addresses within text for the purpose of annotating the link.

Comments :

ID : 1788 **Fixed in version :** 3.4.3

Short Description: Error in positioning of vertical text objects inside Omnis reports

Full Description: We have noticed that the current version (namely, v3.3.0 and v3.4.0) of PDFDevice does not handle correctly the POSITIONING OF OBJECTS CONTAINING VERTICAL TEXT inside Omnis reports. It somehow moves the objects a few millimetres to the left. Please notice that this issue did not occur with previous versions (for example, when printing the same reports with Omnis Studio 5.2 using a former version of PDFDevice, the positioning of vertical text is perfect.) We are using macOS 10.14 and the latest release of Omnis Studio (Studio 10.1).

We are enclosing three samples of the same invoice report from our
application, in PDF format, generated with 3 different engines:

* MacOS native PDF device
* Adobe Acrobat
* Brainy Data PDFDevice

Only the sample produced with PDFDevice seems to be wrong as far as the position of the vertical text on the right is concerned.

We are enclosing a sample test library, just in case. The library contains just a report with 3 adjacent text objects, all of them angled
at 270 degrees, with this basic definition:

[con(style(kEscAngle,kAngle270),iTextLine1)]

Should you require any further information or clarification, please let
us know. We'll be glad to help.

Comments : This issue has been resolved. You should test your vertical/diagonal text positioning within all of your reports to ensure they have not been adversely affected.

Release Notes PDFDevice version 3.4.0

3 Sep 2019 23:26

PDFDevice Bug Fixes

ID : 1725 **Fixed in version :** 3.3.1

Short Description: Characters printed on top of each other (was Crash in printing)

Full Description: Attached is an example library that only crashes for us in runtime, it works perfectly in design time. It simply takes an Omnis print file and prints it to PDFDevice. Nothing too special.

It crashes with a memory allocation error in the nanov2 library. The crash log does not name PDFDevice by name but printing to any other destination does work. It may thus be a problem with Omnis (I have send tech support a copy too) but it may be a problem with PDFDevice.

Could this be related to locking logic that you're adding into the next version of PDFDevice? Or is that an unrelated new feature?

Note that we started experiencing these crashes ever since we went up to Studio 8.1 even though we are now on the latest version of 10. All works fine in Studio 6.1.3

AttachedFile: print_crash.zip

Comments : As discussed, we were unable to reproduce the crash, but we did notice that for some report fields all the characters were drawn on top of each other. This is the problem we have investigated and resolved.

The field in question was using a font (New York) that did not exist on our system. PDFDevice did the right thing and mapped it to some standard default font (in this case Arial) so it would be able to embed the text correctly. Unfortunately, it appeared that Omnis did not map the font to something suitable leaving the system to map the font, which resulted in a disparity between Omnis and PDFDevice when measuring text. If both chosen fonts had been true-type fonts, there would have been no problem. However, the system, in this case, appeared to have picked a non-true-type font which caused problems when mapping measurements as PDFDevice assumed we are using a true-type font. This may well also work around the crash on the Omnis side that you are experiencing.

PDFDevice now ensures that Omnis also uses Arial in such cases. However, it is strongly advised to avoid using fonts that may not exist on your client machines or on machines that print report files, as PDFDevice cannot know what font Omnis will end up using to measure and position text before passing the report objects to PDFDevice.

ID : 1726 **Fixed in version :** 3.3.1

Short Description: Incorrect version number displayed in producer details

Full Description: The PDF producer details show a strange version number when using Studio 10 to produce the PDF file. This works correctly in Studio 8.1.x or earlier.

Comments : As discussed, we were unable to reproduce the crash, but we did notice that for some report fields all the characters were drawn on top of each other. This is the problem we have investigated and resolved.

The field in question was using a font (New York) that did not exist on our system. PDFDevice did the right thing and mapped it to some standard default font (in this case Arial) so it would be able to embed the text correctly. Unfortunately, it appeared that Omnis did not map the font to something suitable leaving the system to map the font, which resulted in a disparity between Omnis and PDFDevice when measuring text. If both chosen fonts had been true-type fonts, there would have been no problem. However, the system, in this case, appeared to have picked a non-true-type font which caused problems when mapping measurements as PDFDevice assumed we are using a true-type font. This may well also work around the crash on the Omnis side that you are experiencing.

PDFDevice now ensures that Omnis also uses Arial in such cases. However, it is strongly advised to avoid using fonts that may not exist on your client machines or on machines that print report files, as PDFDevice cannot know what font Omnis will end up using to measure and position text before passing the report objects to PDFDevice.

ID : 1732 **Implemented in version :** 3.4.0

Short Description: Writing error when printing to PDFDevice

Full Description: With your recent fix and a bunch of fixes from Omnis Support resulting in a Studio 10.0.0.3 (you might want to talk to those guys, they are aware of a bunch of things around this whole thing) we're no longer crashing.

This however has been replaced with an even more annoying issue which we can only reproduce on SOME machines but consistently produce on those machines (my old 2012 Mac being one of them while my brand new 2018 Mac works great). It also ONLY happens in runtime, there is no issue in a developer build.

I've included the same print example I gave you some time ago with a new print file. It will likely work fine on your computer as it does on mine but maybe you can tell something from the print file or from the PDF file I included.

The PDF file is the resulting file, you can thus see that PDFDevice opens the file and does successfully start writing it. But it only writes some of the header data and then fails. I'm hoping you can pinpoint where about there may be a problem.

Also Omnis PDF prints the file correctly but doesn't have alot of your features that we use so it's not a solution.

As we've got a few clients who've not been able to print PDFs since the start of the year first due to the crash issues and now this follow up issue you can image we're under a bit of pressure to get this resolved. Any help you can offer is greatly appreciated.

Comments : PDFDevice was using the Omnis SDK callback FILEcreateTemp to create a temporary file during PDF generation. Unfortunately, depending on how Omnis was installed, FILEcreateTemp may fail due to lack of permissions in the locations where Omnis tries to create temp files.

We have changed PDFDevice to directly call the system to create all files during PDF generation, so this should no longer pose a problem. Because this is an important change, you should thoroughly test PDF generation in your application. If you are using the memory destination feature of PDFDevice, no files are created.

Important Note for Macintosh Developers: PDFDevice now uses the Standard-C library for all file IO. Consequently, when using PDFDevice on Macintosh with Studio 5 or earlier, file paths must be converted from HFS paths to POSIX paths using the FileOps function $converthfstoposixpath().

We have included below the response from Omnis engineering, which may provide useful information for developers who may encounter similar issues with Omnis temp files.

------ quote start
Our engineer ... has confirmed that calling FILEcreateTemp will try to create a temporary file local to the Omnis.app file which is usually in /Applications/Omnis Studio 10.0.app/Contents/MacOS.

If the Omnis.app package and its sub-directories are owned by root and only the owner has read and write permissions, then this problem you are experiencing with BrainyData's xcomp may occur as it would be failing to create the temporary file.

I have tried to do a sudo chown +R aaugustin on the ... app and I can confirm that it works fine now, so this is a possible a work around for the moment or alternatively BrainyData could use an explicit user based path for the cache/temp files.

The engineer also wanted to point out that, for a code-signed install, modifying the app package with temporary files is not desirable (although if the files are deleted then it may be ok).

He will make changes to Omnis so that temporary files are written to the user support file directory rather than the program directory, however, it is good to bear in mind for the future that if the tree is not using the firstruninstall mechanism, then the program and support file directory will be essentially the same.
------ quote end

Bug Fixes

ID :	1633	**Fixed in version :**	3.3.0
Short Description:	Problems generating PDF files - Error 109		

Full Description: Problems generating PDF files - Error 109

Some customers have problem with PDF files generated using pdfdevice.dll component.
When they try open the PDF file, the error 109 is generated.

Not all system configuration are affected, but in the last month several users reported us this problem.

This is the configuration of the last ticket:
Windows 10 Pro 1709 Build 16299.248
Adobe Acrobat Pro DC classic version 2015.006.30413

We are unable to reproduce the problem.
I attached an example of a corrupted file sended by the user.

Thanks in advance.

AttachedFile: DocMatrixPDF.pdf

Comments : PDFDevice automatically removes empty pages during the file writing process. This feature conflicted with changes made for font substitution, causing corruption of PDF object indexes and cross references.

Release Notes PDFDevice version 3.2.9

22 Dec 2017 08:26

Bug Fixes

ID : 1625 **Fixed in version :** 3.2.9

Short Description: overlap issue using PDFDevice

Full Description: there is an issue using pdf device 3.3.7 and 3.3.8.

In the attachment you can see two pdf reports. The first one it's ok and it is had done printing on PDF from screen device on macOS. The second one it is had done using pdf device.

Please take a look at page no. 7 to see the difference.

What do you think? Can it be a font issue?

AttachedFile: 1.pdf_sample.zip

Comments : This issue was caused by changes related to the new justified text feature in Studio 8 reports. Word spacing set in PDFDevice for fully justified text was not cleared correctly, causing non-justified text that follows spaces to be positioned incorrectly.

Release Notes PDFDevice version 3.2.7

13 Sep 2016 09:57

Bug Fixes

ID : 1513 **Fixed in version :** 3.2.6

Short Description: All text becomes bold

Full Description: We have a client using the 'Baskerville Old Face' font. We recently updated their pdfdevice and they have now reported that their layouts are now in bold.
It appears that if you set some text in the OWrite field to bold the PDFDevice makes all the text bold.
Changing the font to Arial for example cures the problem.

Comments : We think we have fixed the issue. Could you try the attached build 3.2.6 and let us know if this resolves your problem.

ID : 1514 **Fixed in version :** 3.2.6

Short Description: Text bold using RepJText component

Full Description: We use in our reports the RepJText component.
Using the latest version of PDF Device 3.2.5, the text printed with RepJText component is turned to bold.
I attached two PDF files created with PDF Device version 3.2.4 and 3.2.5 so you can see the difference.

Comments : We think we may have fixed the issue, could you please try the attached build 3.2.6 and let us know if this resolves your issue.

ID : 1519 **Fixed in version :** 3.2.7

Short Description: Loss of performance

Full Description: In rare cases, loss of performance may be experienced as a consequence of a change to memory handling that was introduced in version 3.2.4.

Comments :

Release Notes PDFDevice version 3.2.5

15 Jun 2016 23:58

Bug Fixes

ID : 1447 **Fixed in version :** 3.2.5

Short Description: PDF Doc produced displays error when opened with Adobe

Full Description: We have a Omnis studio library that is running correctly
in both Studio 6.0.3 and Studio 6.1.2.1 using
PDFDevice 3.1.5.

We are testing PDFDevice 3.2.4 with a view to going to
the 64bit version of Studio in the future.

Having updated the PDFDevice.dll with the relevant 3.2.4
version, for both 6.0.3 and 6.1.2.1 and restarting.
The PDFs are still being created without issue, however:-
When the PDFs are opened/viewed using either Adobe
Acrobat Reader or Acrobat Writer, a warning message is displayed
advising of a document error (see attached .jpg in zip)

The documents are still viewable after clearing the
warning message, and look correct.
The documents appear to open correctly with the
"MS-Edge viewer".

Docs created by both versions of PDFdevice are also
attached in the ZIP.

Comments : We believe we have resolved the issue which turned out to be a lot more complex than anticipated. Consequently it was not a simple fix and carries some risk. If we provide you with a pre-release build, can you test it with your library to make sure it functions now as expected?

ID : 1450 **Fixed in version :** 3.2.5

Short Description: Vertical text not positioned correctly

Full Description: When using the style() function to print text at different angles, some angles are not positioned correctly in the resulting PDF.

Comments :

ID : 1463 **Fixed in version :** 3.2.5

Short Description: PDF Device files with Error when open in Acrobat

Full Description: Been having a issue lately with PDF files created in PDF
Device that flag an error or cannot be opened in Acrobat.

I am attaching a sample of one of the files.

I can repeatedly create files with errors. It does not
however happen on all reports.

I have other examples if you need them.

Comments : We have fixed such an error in version 3.2.5 which we have not released yet. We can sent a patch release for Windows but the reported Omnis Studio version appears to say 3.2.2.1. I assume it is meant to say 4.3.x.x!

Release Notes PDFDevice version 3.2.4

30 Nov 2015 14:09

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 int his 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.

Bug Fixes

ID :	1374	**Fixed in version :**	3.2.2
Short Description:	Text truncation and invalid PDF error		

Full Description:

We have an odd situation with a report we use for hazard labels for laboratory samples. The text of the report appears in both English and French. The report prints fine within Omnis using the Printer or Screen destinations, but the BrainyPDF truncates one of the report fields. Also when opening the saved PDF, Acrobat complains: "An error exists on this page. Acrobat may not display the page correctly. Please contact the person who created the PDF document to correct the problem."

I've attached a sample PDF. We use the Text Block commands in Omnis to dynamically construct the text, interpersed with conditions and a few style() functions so that the "signal word" appears in bold, and the text of the "hazard phrases" appear in plain text. The PDF seems to drop the text just at the first carriage return, which is added from the text block command. The French text which should follow contains some accented characters, so I also wondered if perhaps we may have a code page compatibility problem somewhere...? Method text follows.

```
Selected methods for Class 'Shared.rGHS_sample' as at 31 Jul 2015  13:44:36
                                                          Page 1

##### Method 'formatPhrases' #####
No. Method text
1   Calculate hTxt as ghsObjRef.$getHtxt()
2   Calculate pTxt as ghsObjRef.$getPtxt(kTrue)     ;; label only
3   Calculate hTxtFr as ghsObjRef.$getHtxtFr()
4   Calculate pTxtFr as ghsObjRef.$getPtxtFr(kTrue)     ;; label only
5
6   Begin text block
7   If len(sWord)>0
8     Text: [style(kEscStyle,kBold)][sWord][style(kEscStyle,kPlain)]
9   End If
10  If len(hTxt)>0
11    Text: [hTxt] (Carriage return)
12  End If
13  If len(sWordFr)>0|len(hTxtFr)>0
14    Text: ------------------------------------------------------------ (Carriage return)
15    If len(sWord)>0
16      Text: [style(kEscStyle,kBold)][sWordFr][style(kEscStyle,kPlain)]
17    End If
18    If len(hTxtFr)>0
19      Text: [hTxtFr] (Carriage return)
20    End If
21  End If
22  End text block
23  Get text block hTxtBlock

No method errors found
```

Sample text:
 00000001#DANGER 00000000#Highly flammable liquid and vapour. Causes skin irritation. Causes serious eye damage. Suspected of damaging fertility or the unborn child.

 00000001#DANGER 00000000# Liquide et vapeurs très inflammables. Provoque une irritation cutanée. Provoque des lésions oculaires graves. Susceptible de nuire à la fertilité ou au foetus.

Seeing as the weekend has started in the UK, we can pick this up on Monday!

Comments : The PDF error message was the result of a calculation issue when there were no spaces in a line which resulted in the standard c libraries inserting an “inf” instead of “0.000” for the extra word spacing. It was the line with the repeating = signs that caused it.

We were also able to reproduce an issue when fully justified text (kJustifiedJst) includes text escapes that were inserted using the style() function. The result was that due to style changes within the text run some text would overlap. For example bold text and plain text within the same line would become justified independently and overlap.

Both these issues have been resolved.

Release Notes PDFDevice version 3.2.1

22 Jan 2015 10:12

Bug Fixes

ID :	1321	**Fixed in version :**	3.2.1
Short Description:	Black boxes instead of lines		

Full Description:

First of course the best wishes for 2015. Hopefully it will be a year in good health, lots of new customers and less bug messages.

A customer bought our software because of our mail option of PDF Invoices by email. However they have a problem on their iMac when the PDF's are created. On the places where lines are on the report, they get black boxes (see attachment). They are using OS X 10.9.5. When i try the same on my MacBook Air with OS X 10.9.5 it shows right. (also in attachment)
We use PDF Device 3.1.4R. But i also tried the latest download 3.2.0.
Which gave the same problem.

The method we use is printing the report with PDFDevice to disk and attach this to the Omnis mail that is sent with SMTPSend.
The file on disk already has the problem.

I cannot see the difference between their Mac and mine.
I hope this is a know issue or you know in what kind of combination this occurs.

Comments :

There was an issue with setting kDevPdfPrintScale to anything other than 100% . PDFDevice failed to scale the line thickness which resulted in these large black boxes.

Bug Fixes

ID : 1268 **Fixed in version :** 3.1.8

Short Description: Unable to print PDF

Full Description:
A customer of ours says his customers can't print the PDF documents he sends.
The PDF's are generated with PDFDevice 3.0.0
Is there anything you notice with these document which may explain this behaviour ?
The security settings allow for printing.
Maybe some font used ?

Comments : This problem is related to the kDevPdfConvLinks option and occurred as the report contained a web site address followed by a closing bracket. PDFDevice mistakenly included the closing bracket ')' as part of the link when it annotated the link. PDFDevice has been changed to consider punctuation characters such as brackets and excludes them from any links that are encountered.

ID : 1293 **Fixed in version :** 3.1.6

Short Description: Printing a Report Omnis Server Crashes

Full Description: pdfdevice v3.1.5 causes Omnis Server to crash. The reason for that isn't to much data. It seems that it is one or more non visual characters.

Here is an example of the text which causes problems:

"&#65279;&#65279;Ziegeleistrasse 17/A"

In front of the character Z are two non visual characters. I hope they don't get lost when submitting the request.

I attached two Omnis Disk images (server_print.rep and dev_print.rep) in the print_to_disk.zip which Michael asked for. server_print.rep was printed with a Omnis Server and dev_print.rep with Omnis Develop.

Comments : Unfortunately, we have been unable to reproduce a crash. So the only immediate solution that we can offer is that we handle the characters that are causing the issue. From now on PDFDevice will strip out byte order characters U+FFFE and U+FEFF as these character do not perform any useful function within the context of a PDF file.

This issue has now been identified, please see release notes for case 1299.

ID : 1299 **Fixed in version :** 3.1.7

Short Description: When printing to PDF Device Omnis crashes

Full Description: this problem is like the one of "Case ID 1293" except it seems to be a different character which causes the problem.
The character
" &#61514; " U+F04A
seems to cause the problem. When i remove the character it prints fine.

Like Case ID 1293 it only crashes when Omnis runs as a service and it also crashes when Omnis Server runs in window mode.

Comments : It appears that when Omnis runs as a windowless service, Omnis does not initialise the window's COM object library. Without this initialisation PDFDevice fails to fetch the multi language support object which allows PDFDevice to map characters that are not supported in one font to a font where they are supported. This interface was assumed to always be accessible and a vital test was missing which resulted in a crash in a windowless service application when unusual characters were encountered.

We believe the reason the COM library is not initialised in this case is because Omnis does not directly initialise the COM library and it is only initialised in a windowed application when Omnis loads OLE support which initialises the COM library indirectly. There may also be other windows functionality that Omnis utilises which will initialise the COM library when called.

Having studied the requirements for initialising the COM object library we have found it somewhat ambiguous in relation to what has to be done from within a plug in external DLL when the main application code does not initialise the COM library by default but may attempt so later. It is made ambiguous in two ways. Firstly, there are two ways of initialising the library and choosing the correct one is vital for the operation of the entire application. Secondly, our DLL may or may not be executed on the main thread. If the COM library is initialised incorrectly within the context of Omnis some parts of Omnis that require the COM interface may stop working.

Based on the information available to us, we have implemented a mode of initialisation that is compatible with OLE and which is executed when Omnis failed to initialise the COM library first. We were unable to fully resolve potential ambiguities relating to different thread execution and the impact on COM library initialisation. We therefore recommend that, if PDFDevice is used in a web server and is run as a service, PDFDevice is stress tested by forcing multiple thread execution, i.e. at least two simultaneous requests that produce PDF files. Unfortunately, COM object initialisation appears to be very library dependent (i.e. what functionality your library utilises and in what order) so we were unable to devise a simple test that would cover all eventualities.

However a failure by Omnis or PDFDevice to initialise the COM library will no longer result in a crash. At worst, there will be a small performance overhead when PDFDevice attempts to manually map a suitable font or if that is not possible characters that cannot be mapped to a font may appear as invalid characters within the PDF output. This can take on the appearance of a square where the invalid character is located.

Release Notes PDFDevice version 3.1.5

16 Jul 2014 19:43

Bug Fixes

ID :	1273	**Fixed in version :**	3.1.5
Short Description:	Crash pdfdevice		

Full Description: PDFDevice crashes with some prints, I suppose it depends to some strange characters

Comments : ~~~27Jun14 10:15 Brainy Data Response~~~
The issue was caused by Windows 7 not having installed a suitable true-type font for displaying some of the 'special' chinese characters within the report. As a substitution font, the system returned a non-true-type font which for some reason failed to map to any of the fonts known by PDFDevice (it was this that caused the crash). We have now changed our code to reject non-true-type substitution fonts which ultimately will mean that the 'special' characters will not be displayed correctly, unless the client installs a true-type font that supports them, such as "Arial Unicode MS". We have also made save the code that caused the crash when PDFDevice fails to map an unknown substitution font that was created by the system.

~~~27Jun14 07:45 Brainy Data Response~~~
We have tested the provided report and were unable to make it crash on Windows XP. We tried Omnis Studio 6.0.2, 5.2.2 and 5.1.

However, we were able to make it crash on Windows 7, indicating it is a platform specific issue which we are currently investigating. We will contact you as soon as we know more.
~~~

Release Notes PDFDevice version 3.1.4

2 Apr 2014 19:43

Bug Fixes

ID :	1230	**Fixed in version :**	3.1.2
Short Description:	Pdfdevice crash		

Full Description: From our customer with OSX 10.9 (mavericks), whenever he's printing PDF with PdfDevice (3.0 and 3.1), Omnis crashes.

Comments : The problem was related to some special unicode characters that are used in the report in question, specifically U+2028 and U+0823. One of these characters does not exist in any of the installed fonts so the Mac system resorts to the "LastResort" font. The "LastResort" font is a special font that encodes all unicode characters and displays them as a box with the unicode character ID inside it. Although the "LastResort" font is a true-type font, it uses an indexing table that PDFDevice did not support (all standard fonts usually provide alternative indexing tables that are supported by PDFDevice). We have now implemented the latest indexing tables as specified by the true-type specification.

This problem also highlighted some issues in relation to font-substitution in general which also effects the windows platform in some manner.
1) a potential crash if a report contains two characters not supported by the intended font and if they appear in an order that causes PDFDevice two refer to two different substitution fonts.
2) when a font is substituted some subsequent characters that are supported by a more suitable alternative substitution font may always map to the current substitution font instead.

Both of these problems are rare intermittent problem and only occur during language substitution, when unusual characters are used.

ID :	1239	**Fixed in version :**	3.1.4
Short Description:	Pdfdevice crash		

Full Description: With another DB the latest version of PdfDevice (v3.1.2) you sent crashes.

Comments : This is a Mac OS X only issue.

An invalid character code caused a crash in the Mac OS X CFString handling. It appears there is some corruption in the Omnis data in the last character of a string. More suitable safety checks have now been put into place.

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.

Bug Fixes

ID : 1118 **Fixed in version :** 3.0.0

Short Description: Unicode characters not rendering

Full Description: PDFDevice does not handle "pseudo fonts". Arial Unicode MS does work but simple Arial does not render Thai, Korean, Chinese characters. Omnis Studio does render the characters correctly. This means that a report sent to screen or printer comes out fine but to PDFDevice it comes out with squares. To try it out go to www.caliach.com and download one of Thai, Korean or Chinese Language Edition demo - any report fails with PDFDevice.

Comments : When using fonts such as Arial with languages such as Thai, the system has to use a different font to render the language glyphs that are not supported in the chosen font. Unfortunately, as PDFDevice was directly accessing the chosen fonts data to embed glyphs in PDF files, it failed to consider language font substitution.

We have resolved this by implementing font substitution whenever an unsupported font-glyph combination is encountered. On both Macintosh and MS Windows the system provides specific functions which PDFDevice now utilises.

ID : 1189 **Fixed in version :** 3.0.0

Short Description: Report does not print graphic to PDF

Full Description: Attached is a library with a report. Printing to screen or the printer the background picture prints fine along with the text. Printing to PDF the text prints fine but the graphic does not print.

Comments : This issue is caused by the report overlaying two platform specific background images. One image that only prints on the Macintosh, the other only prints on Windows. Unfortunately, external devices do not have official documented access to the information that is required to distinguish these type of images resulting in external output devices such as PDFDevice, HTML and RTF output destinations producing output for both images. This results in one empty white image and one correct image to be output. In the case of PDFDevice on windows, the white image produced from the Macintosh background picture is placed on top of the windows' background picture.

We have managed to correct this by extracting Omnis picture header information and establishing the actual type of the image data so that platform specific images not intended for the current platform are skipped.

This fix will be released as part of the version 3 release which is imminent.

A better work-around would be to convert the background image to a cross platform format. This would mean that one of the images can be removed from the report which will reduce the overall size of the report. Please refer to the Omnis documentation how to convert background report pictures to an Omnis color shared format.

Enhancements

ID : 456 **Implemented in version :** 3.0.0

Short Description: Ability to print to memory.

Full Description: We would like the "print to memory" feature, so we would process a document once and then output it to paper and then PDF, instead of processing it twice (what we do now).

Comments : see details for enhancement 1172

ID : 713 **Implemented in version :** 3.0.0

Short Description: Font Embedding

Full Description: How difficult would it be to support Open Type font

Comments : The request is for Type 1 Postscript fonts.

We have decided at this time not to support type 1 font embedding but instead we have added a feature that allows the substitution of type 1 fonts with a closely matching true-type font. The device parameter kDevPdfSubstituteFonts, if set to kTrue, enables type 1 font substitution.

WARNING: This will only work with type 1 fonts that implement standard mac-roman/ANSI character sets. It may not produce the desired response for type 1 bar code or other symbolic fonts. Alternative true-type fonts should be used instead.

ID : 1153 **Implemented in version :** 3.0.0

Short Description: Bold or Italic not shown

Full Description: When using fonts that do not support the bold or italic typefaces in their font data, the system synthesises the bold and italic style from suitable font data provided by that font. It typically takes the regular typeface and adjusts the rendering process accordingly.

PDFDevice will now simulate this behaviour by utilising a combination of PDF matrix and rendering features in order to synthesize the required typeface in the PDF file.

Comments :

ID : 1154 **Implemented in version :** 3.0.0

Short Description: OS X Core Text Support

Full Description: The entire OS X platform dependent code base of PDFDevice has been updated to use the OS X version 10.6 core text API for all font related work. PDFDevice will still be compatible with prior versions of OS X for which it will revert to the now obsolete MAC ATSUI SDK.

As these changes were substantial, PDFDevice will require thourough testing prior to releasing version 3 to end users.

Comments :

ID : 1172 **Implemented in version :** 3.0.0

Short Description: Printing to memory

Full Description: We have added the ability to print PDF output directly to memory without any disk access. This should improve printing performance in situations where developers require PDF output in memory for storing in a DB or using FTP to upload to another server.

To use this feature one must provide the full notation string to a method of a class instance in the kPdfFileName device parameter. For example:

```
Do $cdevice.$setparam(kDevPdfFileName,con($cinst().$fullname,'.$myMethod(#)'))
```

When the print job is complete and the specified method is called. In the place of the '#' symbol, PDFDevice will insert the unique ID of the current memory output stream. The method $myMethod must specify a parameter of type Long Integer.

When the method is called, the static function "PDF Device.$getmemoryoutput" (see Catalog->Functions->PDF Device) can be called to fetch the binary PDF data. For example:

```
Do PDF Device.$getmemoryoutput(pOutputId,ivMemoryOuput)
```

Note: The notation method is called prior to Omnis returning from the print command.

Please inspect the OWrite Plus examples for example code. Search for marker CHANGE_PDF_MEM_DEST.

Comments :

ID : 1182 **Implemented in version :** 3.0.0

Short Description: PDF/A Support

Full Description: Version 3 implements support for full PDF/A-1b compliance. The new device parameter kDevPdfaEnabled turns on PDF/A generation. For full details please refer to technical note TN0023 at http://www.brainydata.co.uk/supportpublic/technotes.htm.

Comments :

ID : 1183 **Implemented in version :** 3.0.0

Short Description: New notation name

Full Description: Due to conflicting external device names in Studio version 6, we have changed the name that is used when the device is referred to using notation. Prior to this change one could refer to PDFDevice using the notation $devices.PDF. This has now been changed to $deviceds.BrainyPDF. Please refer to technical note TN0021 at http://www.brainydata.co.uk/supportpublic/technotes.htm.

Comments :

ID : 1184 **Implemented in version :** 3.0.0

Short Description: Studio 6 compatibility

Full Description: Version 3 includes DLLs that were build with the Studio version 6 external SDK and developers should use these builds with Studio 6. These builds also include support for the new jstJustified option for report text objects.

Comments :