

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 thorough 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 \$devices.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 :

www.brainydata.co.uk

