

Fonts and Solid Framework

Author: Roger Dunham

Date: 12th July 2019

Updated 30th March 2021

Introduction

When a document is reconstructed from a PDF, an attempt is made to use the same font in the document as was used in the PDF.

There are times when this is not possible.

For example, if the PDF uses an obscure font (see Figure 1), then Solid Framework will need to substitute the original font for the one that it considers to be most similar. Other scenarios where substitution commonly occurs are when the PDF contains “Times” or other fonts which is generally not available on a Windows machine and must therefore be substituted with “Times New Roman”.



Figure 1- Example of an unusual font

Q: Why can't Solid Framework just extract the font from the original PDF?

A: Very often the PDF only contains the subset of the font for the characters that are actually used. That is very efficient in the PDF, but not very useful if you need to use a different character in the reconstructed document.

Furthermore, many fonts are subject to licenses which prevent them being copied from the original PDF.

How Font Substitution Works

The exact mechanism for how font substitution works is confidential, as it is one of the things that makes Solid Framework better than its competitors.

The general mechanism is as follows:

- Embedded vector (ttf) fonts are substituted based on various font metrics included in the PDF.
 - Non-embedded fonts are substituted using font name and glyph widths only. Twenty years of experience has taught us that other font metrics (e.g. ascent, descent and bbox) are unreliable in real world PDFs.
 - Metrics of the font in the PDF are compared with those of the available substitution candidates.
 - The objective is to minimize geometric difference between the font in the PDF and the one used in the generated document. This helps to retain document layout.
 - Substitution favours the same font family over fonts from a different family.
 - The font candidate must contain all of the Unicode codepoints used in the PDF if possible.
-
- Symbolic fonts and vertical fonts have specific handling
 - Embedded raster fonts cannot be used by Word and are resolved via OCR if that is available.

In previous versions of Solid Framework, the fonts that were considered as possible substitutes were just those that were installed on the machine on which Solid Framework was running. This worked well on Windows, but Linux® generally has far fewer and different fonts to those available on Windows machines. Some versions of Windows Server also have very limited fonts available.

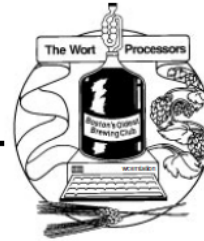
As such if a document is reconstructed on a Linux server then the result may be very different from the result that would be obtained by converting on a Windows desktop machine. Furthermore, there is no guarantee that the conversion would be the same on two different windows desktop machines, since different fonts could be installed on each.

The problem of differing reconstruction is shown below:

Brewprint

The Newsletter of the Boston Wort Processors
Volume IX, Number 12

December 1996



ELECTION RESULTS EDITION

THE PEOPLE SPEAK

Worts Just Say No!

The Boston Wort Processors once again voted against

Letter From The President

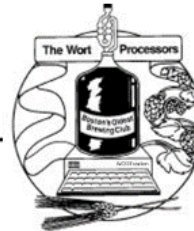
Usually, politicians say "Thanks for giving me the

Figure 2 - Original PDF

Brewprint

The Newsletter of the Boston Wort Processors
Volume IX, Number 12

December 1996



ELECTION RESULTS EDITION

THE PEOPLE SPEAK

Worts Just Say No!

The Boston Wort Processors once again voted against

Letter From The President

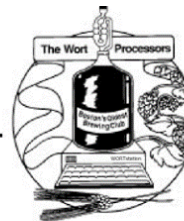
Usually, politicians say "Thanks for giving me the

Figure 3 - Document reconstructed on a Windows desktop machine.

Brewpri

The Newsletter of the Boston Wort Pr
Volume IX, Number

December 1996



ELECTION RESULTS EDITION

Worts Just Say No!

The Boston Wort Processors once again voted against politicians say "Thanks for giving me th

Letter From The President

Figure 4 Document reconstructed on a Linux machine with limited fonts.

The main issue here is that the "Arial" font in the original PDF has been substituted for "DejaVu Sans", since "Arial" was not available

A secondary issue is that when viewed on a Windows machine, Word has further substituted the font "DejaVu Sans" with "Verdana".

Specifying an Alternative Folder to Search for Fonts

Although the default location that is searched for font information is the system's fonts folder, it is possible to specify an alternative location where Solid Framework will search for fonts. This can be done using:

```
SolidFramework.Platform.Platform.SetFontDirectory([Path to folder containing fonts])
```

However this still requires the font files to be physically located within the specified directory.

Using a Fonts Database File

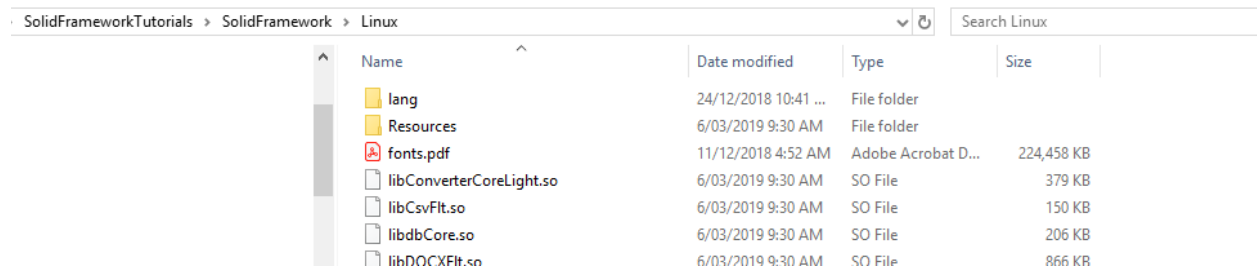
Solid Framework 9182 introduced the concept of a "fonts" database file. This file is normally called "fonts.pdf". The path to this folder should be specified using

```
SolidFramework.Platform.Platform.SetFontsDataBaseFile([Full path to fonts file])
```

If this value is set then Solid Framework will *only use the fonts embedded in this file* for substitution - fonts that are installed on the machine, but which are not present in the fonts database file will not be used in the reconstructed document.

Q: How do I create a fonts.pdf file?

A: You can create a version of this file to include or exclude fonts based on your specific requirements. Please email support@soliddocuments.com if you require further information about how to do this.



The screenshot shows a file explorer window with the path 'SolidFrameworkTutorials > SolidFramework > Linux'. The search bar contains 'Search Linux'. The file list is as follows:

Name	Date modified	Type	Size
lang	24/12/2018 10:41 ...	File folder	
Resources	6/03/2019 9:30 AM	File folder	
fonts.pdf	11/12/2018 4:52 AM	Adobe Acrobat D...	224,458 KB
libConverterCoreLight.so	6/03/2019 9:30 AM	SO File	379 KB
libCsvFit.so	6/03/2019 9:30 AM	SO File	150 KB
libdbCore.so	6/03/2019 9:30 AM	SO File	206 KB
libDOCXFt.so	6/03/2019 9:30 AM	SO File	866 KB

Figure 5- Typical location of fonts.pdf - in this case within a Linux installation.

Using the fonts file can result in significantly improved reconstruction even on a Linux machine with few installed fonts.

Brewprint

The Newsletter of the Boston Wort Processors
Volume IX, Number 12 December 1996



ELECTION RESULTS EDITION

THE PEOPLE SPEAK

Worts Just Say No!

The Boston Wort Processors once again voted against

Letter From The President

Usually, politicians say "Thanks for giving me the

Figure 6 - Document reconstructed on a Linux machine using fonts.pdf

Summary and Recommendation

If working on a platform which has limited fonts available, or where consistent reconstruction is required regardless of the machine on which conversion occurs, then it is recommended to generate a "fonts.pdf" file that contains the common fonts required for document reconstruction.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.