precisely

EngageOne Enrichment

Version 7.4.0

What's New

Enrichment uses a different method for processing PDF files. This offers users additional PDF-specific features, as well as some potential differences. Enrichment can now also process PDF on Red Hat Linux and CentOS.

Contents:

New features

This section summarizes the new features available in Enrichment 7.4.0 when processing PDFs, and the requirements for PDF processing on Red Hat Linux.

Creating bookmarks

Enrichment can now create PDF bookmarks. You can use this for further post-processing, or for conveying information to a user.

Control file fragment example: Applying a bookmark to the first page of each document

```
<rule>
  <content>
    %%bookmark = "Document: " | %%DOCUMENT_NO
  </content>
</rule>

<add>
 <addtype> B
 <addpart> %%bookmark 15
 <onpage> F
</add>
```

Reading bookmarks

Enrichment can now use existing bookmarks as document breaks. Enrichment can also extract the bookmark label to be used in rule file processing.

Control file fragment example: Using existing bookmarks to determine the top of each document, and write the text contained in the bookmark to a WRITE file

```
<INPUT>
  <NAME> INPUT1
  <FILE> inputfile.pdf
  <TYPE> PDF
  <doc> BOOKMARK
</INPUT>

<rule>
  <content>
```

```
      WRITE("write.txt", "Bookmark text <" | %%BOOKMARK_TEXT | ">")
  </content>
</rule>
```

Adding watermarks

Enrichment can now add watermark images to PDF files. You can rotate, size and set the opacity of the specified image.

Control file fragment example: Adding a watermark image to each page of a PDF file

```
<add>
  <addtype> I
  <addpart> "images\apple.png"
  <watermark> 30 45 25 //30% opacity, rotated 45 degrees, resized to 25%
 of page width
</add>
```

Password protected output files

Enrichment can now apply a password to output files. You can place passwords in an Enrichment variable and construct them using any Enrichment string handling function.

Control file fragment example: Applying a dynamic password to each document

```
<rule>
  <content>
     %%filename = "output" | %%DOCUMENT_NO | ".PDF"
     %%password = READ("accountPasswords.txt")
     <FILEBREAK>
  </content>
</rule>

<output>
  <name>out1
  <dynafile> %%filename
  <pdfuserpassword> %%password
</output>
```

Replace and reposition

Enrichment can now replace the contents of a <FIELD> that uses a <REPOSITION> tag.

For example, the following control file fragment is now valid:

```
  <field>%%addressLines R3
      <window>0.990 1.311 2.591 1.971 in
      <reposition> 1 -1 in
  </field>
<rule>
  <content>
     %%addressLines[0] = "New address line 1"
     %%addressLines[1] = "New address line 2"
     %%addressLines[2] = "New address line 3"
  </content>
</rule>
```

Replacing fonts

Enrichment can now use a font map to replace fonts in PDF files. For example to change a subsetted font to a full embedded font. You specify the font map file using the <TTFONTSMAP> tag in an <INPUT> group.

The following is an example of a valid font map:

```
% Top of font map file
Arial C:\Windows\Fonts\Arial.ttf
"Custom Font Name" C:\Repository\CustomFont.ttf
% End of font map file
```

Windows support

To process PDF on Windows:

- Using the 32-bit engine, fsdk_win32.dll must be in the PATH.
- Using the 64 bit engine, fsdk_win64.dll must be in the PATH.

Linux support

Enrichment can now process PDF files on Red Hat Linux platforms. To process PDF using the:

- 32-bit engine, include libfsdk_linux32.so in a directory in LD_LIBRARY_PATH
- 64-bit engine, include libfsdk_linux64.so in a directory in LD_LIBRARY_PATH

Potential differences

This section summarizes the differences between Enrichment 7.4.0 and the previous release when processing PDFs.

Linux

For Linux, glibc 2.17 or higher is required and gcc 3.14.15 or higher is required.

Large PDF files

For large PDF files, we recommend at least 16GB of memory.

Performance

Using the <THREADS> tag in <ENVIRONMENT> group can improve performance.

<PDF> tag group

Enrichment no longer utilizes a transformation in processing PDF files. Because of this, the <PDF> tag group is no longer necessary.

Character measurement

Enrichment now treats PDF streams as 72 pels/inch, rather than 14400 pels/inch

Hyperlink changes

When creating underlined hyperlinks, the underline will be black, regardless of <COLOR>.

Two-dimensional barcode changes

The default error correction value may be different for some two-dimensional barcodes. The desired error correction can be specified using <QRCODE>, <DATAMATRIX> , or <PDF417> depending on which two-dimensional barcode is being specified.

4 State barcode changes

When adding a 4 state barcode, the FOURSTATE function must be used along with `<ADDTYPE> 4`.

Character placement

Due to subtle changes in text processing, fields whose <WINDOW> value ends in the middle of a word, phrase or sentence may need to be adjusted.

Scaling changes

Previously, to scale a document, the `PDFIN_SCALE` command was used in an options file. Now, the <SCALE> tag in the <INPUT> or <OUTPUT> group will be used to scale pages before <FIELD> processing, if used in the <INPUT> group, or after <FIELD> processing if used in the <OUTPUT> group.

Virtual objects (VDF/VDI files)

Enrichment no longer uses virtual objects such as VDF or VDI files. You now need to add fonts as TrueType Fonts, and other virtual objects as unconverted images in JPG or PNG format.

Font handling

Text <ADD>'s will point to TrueType Font files rather than VDF files. The default font for PDF will be Arial, rather than P0612$.

For PDF files using embedded Type 3 fonts, <INFONTNAME> may not return the font name. Instead of using the embedded name to add text, use a known True Type font.

2 Blue Hill Plaza, #1563
Pearl River, NY 10965
USA

www.precisely.com