Catherina Ackerman - u24076491

COS - 721: EXAMINATION 2024 —- 20 November - 22 November

Questions Answered = 2 & 4

Question 2

2A: Identifying EOF-Delimited Parts with Multiple /Producer Entries

An analysis was conducted using the script provided in Appendix A to determine the number of EOF-delimited sections within the dataset's files. The regular expression used to find the EOF markers was “b'%%EOF'”. The results revealed a total of 1550 EOF-delimited sections across all files. Among these, 333 sections contained contributions from more than one producer tag(s), while 1217 sections were associated with less than 2 producer tags. This was determined by using filters on the CSV output the script in Appendix A has provided. This script analyzes all PDF files in a specified folder, splits each file into parts based on the %%EOF marker, counts occurrences of /Producer and <pdf:Producer> tags in each part, and extracts the metadata (/Producer and /Creator) of each PDF. It then compiles the results into a CSV file for further analysis. Another script (Appendix B) was created to extract the parts of each document to have it on its own to be able to investigate each part separately and then in the whole document. The last mentioned script splits a given PDF into multiple parts based on the %%EOF marker (using the regular expression “b'%%EOF'”), and saves each part as a separate PDF file in a specified output directory.

According to the ISO 32000-1 standard, the PDF specification does not specifically forbid recurring metadata assertions, such as the /Producer or <pdf:Producer> keys. Such repetition, however, is not common practice and may result from compatibility issues or non-standard tool behaviors [1]. The program used to create or edit the PDF is identified by the /Producer key in the Document Information Dictionary. Keys in the metadata dictionary, such as /Producer, need to be distinct. The standard is broken when many /Producer keys are present in the same dictionary. Tools that insert both /Producer and <pdf:Producer> (for compatibility purposes, for example) may show up as components of distinct dictionaries or objects. A PDF may contain numerous entries if it goes via several tools, each of which may append its /Producer metadata. To make sure they work with older or non-standard readers, tools may duplicate metadata using both the /Producer and <pdf:Producer> syntax. As long as each key appears in a legitimate context, the unusual practice of repeating the same producer name throughout /Producer and <pdf:Producer> does not necessarily break the standard. However, the uniqueness constraint for dictionary keys would be broken if the same /Producer entries were repeated several times in the same dictionary [1].

Using a tool like pdftk the following 3 files were combined, 6RWOL, RUW5H, 7HBBX. After inspection, it was found that the Creator was changed to pdftk 2.02 - www.pdftk.com, and the producer of the tool changed to itext-paulo-155 \(itextpdf.sf.net-lowagie.com\). This occurs as a result of pdftk overwriting the metadata fields to reflect its part in creating the combined document when it generates a new PDF file during the input file merger.

/Creator (pdftk 2.02 - www.pdftk.com)

/Producer (itext-paulo-155 \(itextpdf.sf.net-lowagie.com\))

/ModDate (D:20241122113512Z)

/CreationDate (D:20241122113512Z)

Table 1: Compare /Producer and <pdf:Producer>

/Producer	<pdf:Producer>
The software used to produce or edit the file is indicated by the /Producer field, which is a component of the core PDF metadata in the PDF dictionary structure.	The XMP (Extensible Metadata Platform) metadata has <pdf:Producer> incorporated as an XML stream, providing a more contemporary and expandable method of storing metadata.
The software used to produce or edit the file is indicated by the /Producer field, which is a component of the core PDF metadata in the PDF dictionary structure.	The XMP (Extensible Metadata Platform) metadata has <pdf:Producer> incorporated as an XML stream, providing a more contemporary and expandable method of storing metadata.
Legacy compatibility guarantees that /Producer can be read by previous tools.	The depth and extensibility of <pdf:Producer> in XMP metadata are advantageous to modern systems.

Including both helps ensure compatibility across diverse tools and systems.

Their presence in separate sections confirms adherence to both legacy PDF standards and modern metadata practices. If more than one tool processed the PDF, each one might have included its own /Producer metadata. For instance, during various editing phases, programs like Apache FOP, iText, or Acrobat Distiller may leave their imprint.

Tools frequently replicate metadata using both /Producer and alternative syntax forms like <pdf:Producer> to preserve compatibility with older or non-standard PDF readers. There may be several references to /Producer under various scopes or contexts since some files store metadata in distinct dictionaries or objects within the file.

In this analysis at least 91 files with parts that have more than one /Producer and or <pdf:Producer> tag(s) were investigated, and a conclusion could be drawn that most files have both the /Producer and <pdf:Producer> syntax and mostly both reference the same producer.

Looking at file 4ANUUH, EOF-Section 2:

Line 462: <</CreationDate(D:20090303111607-10'00')/Creator(Adobe Acrobat 8.12)**/Producer**(Adobe Acrobat 8.12 Image Conversion Plug-in)/ModDate(D:20090303111607-10'00')>>
Line 487: **<pdf:Producer>**Adobe Acrobat 8.12 Image Conversion Plug-in</pdf:Producer>
Line 501: <</CreationDate(D:20090303111607-10'00')/Creator(Adobe Acrobat 8.12)**/Producer**(Adobe Acrobat 8.12 Image Conversion Plug-in)/ModDate(D:20090303111607-10'00')>>
Line 526: **<pdf:Producer>**Adobe Acrobat 8.12 Image Conversion Plug-in</pdf:Producer>

Line 540: <</CreationDate(D:20090303111606-10'00')/Creator(Adobe Acrobat 8.12)**/Producer**(Adobe Acrobat 8.12 Image Conversion Plug-in)/ModDate(D:20090303111606-10'00')>>
Line 565: **<pdf:Producer>**Adobe Acrobat 8.12 Image Conversion Plug-in</pdf:Producer>
Line 579: <</CreationDate(D:20090303111606-10'00')/Creator(Adobe Acrobat 8.12)**/Producer**(Adobe Acrobat 8.12 Image Conversion Plug-in)/ModDate(D:20090303111606-10'00')>>
Line 604: **<pdf:Producer>**Adobe Acrobat 8.12 Image Conversion Plug-in</pdf:Producer>
Line 637: **<pdf:Producer>**Adobe Acrobat 8.12 Image Conversion Plug-in</pdf:Producer>
Line 666: <</CreationDate(D:20090303160501-10'00')/Creator(Adobe Acrobat 8.12)**/Producer**(Adobe Acrobat 8.12 Image Conversion Plug-in)/ModDate(D:20090303160501-10'00')>>

We can see that all producer information is the same above, just given in a different syntax (/Producer or <pdf:Producer>). Each page or part of the PDF has its metadata section. The fact that each page has a unique information structure suggests that this duplication results from the way the document was produced or processed. The file contains both local (page or section level) and global (document level) metadata. There may be several references to the /Producer key as a result of this hierarchical arrangement.

Looking at file 3RCHL, EOF-Section 1-17 (Table 2) (1st column indicates the EOF-Section, 2nd column indicates the number of producer instances found in the corresponding section, 3rd column is the extracted lines with producer information):

Table 2: Producer Instances Found in Corresponding EOF-Sections 3RCHL

EOF section	# Producer indicators	Extracted lines with producer information
1	2	Line 19: <pdf:**Producer**>Acrobat Distiller 5.0 (Windows)</pdf:Producer> Line 45670: <</Author(JoAnne Butzerin)/CreationDate(D:20030604223311Z)/Creator(PScript5.dll Version 5.2)/ModDate(D:20100624100059-07'00')/**Producer**(Acrobat Distiller 5.0 \(Windows\))/Title(C:\\Documents and Settings\\JoAnne Butzerin.NWFSC\\Desktop\\JoAnne's Documents\\Williams\\Achord\\Wild2002\\Wild2002.wpd)>>
2	2	Line 102: <pdf:**Producer**>Acrobat Distiller 5.0 (Windows)</pdf:Producer> Line 157: <</Author(JoAnne Butzerin)/CreationDate(D:20030604223311Z)/Creator(PScript5.dll Version 5.2)/ModDate(D:20100624101616-07'00')/**Producer**(Acrobat Distiller 5.0 \(Windows\))/Title(C:\\Documents and Settings\\JoAnne Butzerin.NWFSC\\Desktop\\JoAnne's Documents\\Williams\\Achord\\Wild2002\\Wild2002.wpd)>>
3	2	Line 30: <pdf:**Producer**>Acrobat Distiller 5.0 (Windows)</pdf:Producer> Line 85: <</Author(JoAnne Butzerin)/CreationDate(D:20030604223311Z)/Creator(PScript5.dll Version 5.2)/ModDate(D:20100624102027-07'00')/**Producer**(Acrobat Distiller 5.0 \(Windows\))/Title(C:\\Documents and Settings\\JoAnne Butzerin.NWFSC\\Desktop\\JoAnne's Documents\\Williams\\Achord\\Wild2002\\Wild2002.wpd)>>

4	2	Line 177: <pdf:**Producer**>Acrobat Distiller 5.0 (Windows)</pdf:Producer> Line 232: <</Author(JoAnne Butzerin)/CreationDate(D:20030604223311Z)/Creator(PScript5.dll Version 5.2)/ModDate(D:20100624103315-07'00')/**Producer**(Acrobat Distiller 5.0 \(Windows\))/Title(C:\\Documents and Settings\\JoAnne Butzerin.NWFSC\\Desktop\\JoAnne's Documents\\Williams\\Achord\\Wild2002\\Wild2002.wpd)>>
5	2	Line 174: <pdf:**Producer**>Acrobat Distiller 5.0 (Windows)</pdf:Producer> Line 229: <</Author(JoAnne Butzerin)/CreationDate(D:20030604223311Z)/Creator(PScript5.dll Version 5.2)/ModDate(D:20100624103737-07'00')/**Producer**(Acrobat Distiller 5.0 \(Windows\))/Title(C:\\Documents and Settings\\JoAnne Butzerin.NWFSC\\Desktop\\JoAnne's Documents\\Williams\\Achord\\Wild2002\\Wild2002.wpd)>>
6	2	Line 12: <pdf:**Producer**>Acrobat Distiller 5.0 (Windows)</pdf:Producer> Line 67: <</Author(JoAnne Butzerin)/CreationDate(D:20030604223311Z)/Creator(PScript5.dll Version 5.2)/ModDate(D:20100624103746-07'00')/**Producer**(Acrobat Distiller 5.0 \(Windows\))/Title(C:\\Documents and Settings\\JoAnne Butzerin.NWFSC\\Desktop\\JoAnne's Documents\\Williams\\Achord\\Wild2002\\Wild2002.wpd)>>
7	2	Line 24: <pdf:**Producer**>Acrobat Distiller 5.0 (Windows)</pdf:Producer> Line 79: <</Author(JoAnne Butzerin)/CreationDate(D:20030604223311Z)/Creator(PScript5.dll Version 5.2)/ModDate(D:20100624104045-07'00')/**Producer**(Acrobat Distiller 5.0 \(Windows\))/Title(C:\\Documents and Settings\\JoAnne Butzerin.NWFSC\\Desktop\\JoAnne's Documents\\Williams\\Achord\\Wild2002\\Wild2002.wpd)>>
8	2	Line 78: <pdf:**Producer**>Acrobat Distiller 5.0 (Windows)</pdf:Producer> Line 133: <</Author(JoAnne Butzerin)/CreationDate(D:20030604223311Z)/Creator(PScript5.dll Version 5.2)/ModDate(D:20100624104435-07'00')/**Producer**(Acrobat Distiller 5.0 \(Windows\))/Title(C:\\Documents and Settings\\JoAnne Butzerin.NWFSC\\Desktop\\JoAnne's Documents\\Williams\\Achord\\Wild2002\\Wild2002.wpd)>>
9	2	Line 48: <pdf:**Producer**>Acrobat Distiller 5.0 (Windows)</pdf:Producer> Line 103: <</Author(JoAnne Butzerin)/CreationDate(D:20030604223311Z)/Creator(PScript5.dll Version 5.2)/ModDate(D:20100624104832-07'00')/**Producer**(Acrobat Distiller 5.0 \(Windows\))/Title(C:\\Documents and Settings\\JoAnne Butzerin.NWFSC\\Desktop\\JoAnne's Documents\\Williams\\Achord\\Wild2002\\Wild2002.wpd)>>
10	2	Line 21: <pdf:**Producer**>Acrobat Distiller 5.0 (Windows)</pdf:Producer> Line 76: <</Author(JoAnne Butzerin)/CreationDate(D:20030604223311Z)/Creator(PScript5.dll Version 5.2)/ModDate(D:20100624105308-07'00')/**Producer**(Acrobat Distiller 5.0 \(Windows\))/Title(C:\\Documents and Settings\\JoAnne Butzerin.NWFSC\\Desktop\\JoAnne's Documents\\Williams\\Achord\\Wild2002\\Wild2002.wpd)>>

11	2	Line 21: <pdf:**Producer**>Acrobat Distiller 5.0 (Windows)</pdf:Producer> Line 76: <</Author(JoAnne Butzerin)/CreationDate(D:20030604223311Z)/Creator(PScript5.dll Version 5.2)/ModDate(D:20100624105843-07'00')/**Producer**(Acrobat Distiller 5.0 \(Windows\))/Title(C:\\Documents and Settings\\JoAnne Butzerin.NWFSC\\Desktop\\JoAnne's Documents\\Williams\\Achord\\Wild2002\\Wild2002.wpd)>>
12	2	Line 15: <pdf:**Producer**>Acrobat Distiller 5.0 (Windows)</pdf:Producer> Line 70: <</Author(JoAnne Butzerin)/CreationDate(D:20030604223311Z)/Creator(PScript5.dll Version 5.2)/ModDate(D:20100624105933-07'00')/**Producer**(Acrobat Distiller 5.0 \(Windows\))/Title(C:\\Documents and Settings\\JoAnne Butzerin.NWFSC\\Desktop\\JoAnne's Documents\\Williams\\Achord\\Wild2002\\Wild2002.wpd)>>
13	2	Line 21: <pdf:**Producer**>Acrobat Distiller 5.0 (Windows)</pdf:Producer> Line 76: <</Author(JoAnne Butzerin)/CreationDate(D:20030604223311Z)/Creator(PScript5.dll Version 5.2)/ModDate(D:20100624110606-07'00')/**Producer**(Acrobat Distiller 5.0 \(Windows\))/Title(C:\\Documents and Settings\\JoAnne Butzerin.NWFSC\\Desktop\\JoAnne's Documents\\Williams\\Achord\\Wild2002\\Wild2002.wpd)>>
14	2	Line 12: <pdf:**Producer**>Acrobat Distiller 5.0 (Windows)</pdf:Producer> Line 67: <</Author(JoAnne Butzerin)/CreationDate(D:20030604223311Z)/Creator(PScript5.dll Version 5.2)/ModDate(D:20100624110636-07'00')/**Producer**(Acrobat Distiller 5.0 \(Windows\))/Title(C:\\Documents and Settings\\JoAnne Butzerin.NWFSC\\Desktop\\JoAnne's Documents\\Williams\\Achord\\Wild2002\\Wild2002.wpd)>>
15	2	Line 18: <pdf:**Producer**>Acrobat Distiller 5.0 (Windows)</pdf:Producer> Line 73: <</Author(JoAnne Butzerin)/CreationDate(D:20030604223311Z)/Creator(PScript5.dll Version 5.2)/ModDate(D:20100624114710-07'00')/**Producer**(Acrobat Distiller 5.0 \(Windows\))/Title(C:\\Documents and Settings\\JoAnne Butzerin.NWFSC\\Desktop\\JoAnne's Documents\\Williams\\Achord\\Wild2002\\Wild2002.wpd)>>
16	2	Line 4: 356 0 obj<</CreationDate(D:20030604223311Z)/Author(JoAnne Butzerin)/Creator(PScript5.dll Version 5.2)/**Producer**(Acrobat Distiller 5.0 \(Windows\))/ModDate(D:20100624115829-07'00')/Title(C:\\Documents and Settings\\JoAnne Butzerin.NWFSC\\Desktop\\JoAnne's Documents\\Williams\\Achord\\Wild2002\\Wild2002.wpd)>> Line 16917: <pdf:**Producer>**Acrobat Distiller 5.0 (Windows)</pdf:Producer>
17	2	Line 225: <</Author(JoAnne Butzerin)/CreationDate(D:20030604223311Z)/Creator(PScript5.dll Version 5.2)/ModDate(D:20100624121142-07'00')/**Producer**(Acrobat Distiller 5.0 \(Windows\))/Title(C:\\Documents and Settings\\JoAnne Butzerin.NWFSC\\Desktop\\JoAnne's Documents\\Williams\\Achord\\Wild2002\\Wild2002.wpd)>> Line 246: <pdf:**Producer**>Acrobat Distiller 5.0 (Windows)</pdf:Producer>

"Acrobat Distiller 5.0 (Windows)" is declared by both the /Producer syntax in the document information dictionary and the <pdf:Producer> syntax in the XMP metadata. This shows that the tool was used consistently throughout the creation of the paper.

Because of possible redundancies between metadata layers (XMP vs. document info dictionary) or updates made at different points in the document lifecycle, the /Producer information appears in the metadata more than once. Despite numerous changes, the Author and Title fields stay the same, confirming that the metadata relates to the same document.

The document's (3RCHL & 4ANUUH) revision history and the development of PDF metadata handling are shown in several instances of /Producer and <pdf:Producer> parts. This redundancy might lead to inefficiencies even while it guarantees greater compatibility and traceability. Such problems can be streamlined and fixed with the use of appropriate metadata management and an understanding of the document's production chain.

Looking at file 46KXL, EOF-Section 2:

Line 2: 11 0 obj<</CreationDate(D:20081210124205-08'00')/Subject(CalPERS Agenda Item)/Author(claffin)/Creator(PScript5.dll Version 5.2.2)**/Producer(Acrobat Distiller 7.0 \(Windows\))/**ModDate(D:20081210124205-08'00')/Title(Item 9 - Attachment T)>>
Line 6: 33 0 obj<</CreationDate(D:20081210124205-08'00')/Subject(CalPERS Agenda Item)/Creator(PScript5.dll Version 5.2.2)/**Producer(Acrobat Distiller 7.0 \(Windows\))/**ModDate(D:20081210124230-08'00')/Title(Item 9 - Attachment T)>>
Line 14: <pdf:Producer>Acrobat Distiller 7.0 (Windows)</pdf:Producer>

The purpose of both /Producer and <pdf:Producer> is to balance data robustness, increased functionality, and backward compatibility in PDF files. However, version-tracking procedures, tool constraints, and editing workflows can all lead to duplicate instances. Document clarity and integrity are enhanced by ensuring metadata consistency and reducing redundancy, particularly for preservation or standardized use cases.

Looking at file 7AUNH, EOF-Section 2:

Line 2486: /**Producer** (Acrobat PDFWriter 4.05 for Windows NT) {117 0 obj}
Line 2503: /**Producer** (APSetDocInfo 1.9 Solaris SPDF_1086 Aug 7 2003) {Part of 119 0 obj}

It is shown in the above that the producer info is different. This implies that the document was first produced using Acrobat PDFWriter and then processed or altered using APSetDocInfo. The purpose of this dual representation is probably to keep track of the instruments and procedures that went into creating the document, improving traceability and ensuring adherence to archiving or auditing requirements. It is not recommended to have two /Producer fields with different values that are not clearly connected or explained because this could cause uncertainty [1].

Looking at file 23JCE, EOF-Section 1:

Line 607: /Producer (Acrobat Distiller 8.0.0 \(Windows\)) {27 0 obj}
Line 1228: <pdf:Producer>Adobe PDF library 6.66</pdf:Producer> {79 0 obj}
Line 1806: <pdf:Producer>Acrobat Distiller 8.0.0 (Windows)</pdf:Producer> {59 0 obj}

Line 1228 producer information, corresponds to the document dictionary's producer (Object 27).

Tools that read structured XMP data should have redundant but consistent metadata in XML format. Line 1806 producer information, shows that Adobe's PDF Library was used for additional processing or modification of the PDF.

This investigation emphasizes the significance of strong metadata practices to improve document traceability, uphold compliance, and minimize inefficiencies when managing PDF files. In addition to enhancing tool interoperability, these actions will facilitate more efficient document production and processing procedures.

2B: Toolmarks and Consistency

The five toolmarks found in the file sections that shared the /Producer declaration "Acrobat Distiller 8.1.0 (Windows)" were analyzed, and the results strongly suggest that the toolmarks in the various sections are comparable.

The 1st toolmark was the Carriage Return (CR) and this was investigated by using VIM and identifying if each part does contain the Carriage Return as indicated in VIM as ^M. It was found that all parts do contain the Carriage Return.

The 2nd toolmark that was investigated was the tree structure, and there was observed that all lines except the XML part were indented.

Example:

```
41 0 obj
<</Subtype/XML/Length 4031/Type/Metadata>>stream
<?xpacket begin="ï»¿" id="W5M0MpCehiHzreSzNTczkc9d"?>
<x:xmpmeta xmlns:x="adobe:ns:meta/" x:xmptk="Adobe XMP Core 4.0-c316 44.253921, Sun Oct
01 2006 17:14:39">
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <rdf:Description rdf:about=""
        xmlns:pdf="http://ns.adobe.com/pdf/1.3/">
      <pdf:Producer>Acrobat Distiller 8.1.0 (Windows)</pdf:Producer>
    </rdf:Description>
    <rdf:Description rdf:about=""
        xmlns:pdfx="http://ns.adobe.com/pdfx/1.3/">
      <pdfx:Company>B&amp;McD</pdfx:Company>
      <pdfx:SourceModified>D:20090608145838</pdfx:SourceModified>
    </rdf:Description>
    <rdf:Description rdf:about=""
        xmlns:xap="http://ns.adobe.com/xap/1.0/">
      <xap:CreatorTool>Acrobat PDFMaker 8.1 for Word</xap:CreatorTool>
      <xap:ModifyDate>2009-06-08T18:27:55-06:00</xap:ModifyDate>
      <xap:CreateDate>2009-06-08T18:27:51-06:00</xap:CreateDate>
      <xap:MetadataDate>2009-06-08T18:27:55-06:00</xap:MetadataDate>
    </rdf:Description>
    <rdf:Description rdf:about=""
        xmlns:xapMM="http://ns.adobe.com/xap/1.0/mm/">

<xapMM:DocumentID>uuid:f88669b8-1569-4784-bc13-c7177882d6c3</xapMM:DocumentID>
      <xapMM:InstanceID>uuid:0dc7eb0c-816e-42b2-9ed8-df8d70ee69ca</xapMM:InstanceID>
      <xapMM:subject>
        <rdf:Seq>
          <rdf:li>24</rdf:li>
        </rdf:Seq>
      </xapMM:subject>
    </rdf:Description>
    <rdf:Description rdf:about=""
        xmlns:dc="http://purl.org/dc/elements/1.1/">
      <dc:format>application/pdf</dc:format>
      <dc:creator>
        <rdf:Seq>
          <rdf:li>K. Brunkhorst</rdf:li>
        </rdf:Seq>
      </dc:creator>
      <dc:title>
        <rdf:Alt>
          <rdf:li xml:lang="x-default">PART 1 -  REFERENCES</rdf:li>
        </rdf:Alt>
      </dc:title>
    </rdf:Description>
  </rdf:RDF>
</x:xmpmeta>
```

The object numbering toolmark was the 3rd toolmark. This toolmark focused on the consistency of the item numbering across the file sections rather than the precise numbers or the order in which the number appeared. For example, one part has the object numbering 1526 0 obj and another 8 0 obj (SRBFX - Part 2).

The dictionary referencing style, such as whether the file section styled the dictionary as /Types/Pages or as /Types /Pages, was the 4th toolmark and it could be found that in all parts investigated, there was no spacing found between the dictionary referencing style.

The 5th toolmark looked at and which was consistent was the Font identifier /TT. Searching through the document parts, only the /TT font identifier was found and not other common ones like /F.

Table 3: Toolmarks and consistency (No-0/Yes-1)

File name	Carriage return present in part	Tree structure remains same for all compared parts	Object # stays consistent	Spacing between /Type & /Page	Font identifier	Creator_tool (According to Excel file)	Producer_tool (According to Excel file)
SRBFX - Part 2	1	1	0	None	/TT0 1512 0 R/TT1 1513 0 R	Acrobat PDFMaker 8.1 for Word	Acrobat Distiller 8.1.0 (Windows)
SRBFX - Part 3	1	1	0	None	/TT2 1530 0 R	Acrobat PDFMaker 8.1 for Word	Acrobat Distiller 8.1.0 (Windows)
SCOED - Part 2	1	1	0	None	/TT0 63 0 R	Acrobat PDFMaker 8.1	Acrobat Distiller 8.1.0

						for Word	(Windows)
X2TE6 - Part 2	1	1	0	None	/TT0 53 0 R/TT1 270 0 R/TT2 44 0 R...	Acrobat PDFMaker 8.1 for Word	Acrobat Distiller 8.1.0 (Windows)
HE2GQ - Part 2	1	1	0	None	/TT0 1194 0 R/TT1 1195 0 R	Acrobat PDFMaker 8.1 for Excel	Acrobat Distiller 8.1.0 (Windows)
M3UMH - Part 2	1	1	0	None	/TT0 51 0 R/TT1 52 0 R	Acrobat PDFMaker 8.1 for Word	Acrobat Distiller 8.1.0 (Windows)

Toolmarks are consistent across sections of the same file that have corresponding /Producer declarations. It is highly supported by the uniformity found in the five toolmarks under investigation, Carriage Return, tree structure, object numbering, dictionary reference style, and font identifiers, that the components were processed by the same manufacturer, "Acrobat Distiller 8.1.0 (Windows)".

This study emphasizes how crucial consistent toolmark and metadata standards are to preserving document integrity and traceability. The results highlight how crucial it is to handle metadata carefully to optimize workflows and guarantee tool and standard compliance.

Question 4

4A: Font Creation Date vs. Document Creation Date

Attesting to Calibri's creation in 2007 is avoided for the following reasons:
According to primary evidence, it was created earlier (between 2002 and 2004).
Calibri is a family of digital sans-serif fonts in the modern or humanist style. It was created by Luc(as) de Groot between 2002 and 2004 and made available to the public in 2007 together with Windows Vista and Microsoft Office 2007 [2]. Calibri was designed by De Groot in the early 2000s, as one of several fonts for improved screen reading [4], and was released to the public in 2007. [2]. This is according to Lucas de Groot's claims and Microsoft's documents.

Steve Matteson created the sans-serif typeface Aptos [5]. It was first presented as Bierstadt in 2021 after work started in 2019. It was renamed Aptos and formally published in July 2023 as the new default font for Microsoft Office programs, replacing Calibri, following user input and other improvements [6].

Since Aptos recognizes the typeface's provenance, its development year should typically correspond to 2019 for historical and design-focused contexts. Nonetheless, 2023 might be utilized in branding or release-specific circumstances. To prevent misunderstandings the dual history of Aptos must be made clear as both a rebranding and an improvement of a previous typeface without invalidating the preceding design effort, this distinction must be made clear.

The following fonts were examined and researched to get their creation dates to compare with a document's creation date, Verdana, Helvetica, ArialMT, Times-Roman and Courier. The document Modification date was included to ensure that it is always after the creation date or on the same date and time.

Table 4: Font creation vs creation and modification date in a file

Font file group	File name	Font name	Font Creation Date	Document Creation Date	Document Last Modification Date	Years Between Font Creation and Document Date
1	QFKWG	Verdana	1996 [7]	2009-12-17 11:18:21 (Converted from hexadecimal-encoded Unicode)	NA	13
	5M56W			2013-10-16 09:33:19	2013-10-16 09:33:19	17
2	6C4F5	Helvetica	1957[8]	2013-08-05 9:16:00	2015-06-10 0:47:41	56
	QNKRV			2005-02-28 16:36:27	2005-02-28 16:36:27	48
3	GGGXC	ArialMT	1982[9]	2017-04-21 1:15:31	2017-04-21 1:15:31	35
	64PJR			2002-05-13 19:51:35	2002-05-28 09:31:05	26
4	OAAZZ	Times-Roman	1932[10]	2006-03-31 15:18:11	NA	74
	DT6DE			2005-04-28 12:48:04	2005-04-28 12:48:04	73
5	Z2PBT	Courier	1955[11]	2006-06-26 14:46:45	2006-06-26 14:46:53	51
	37RZ6			2010-01-25 14:46:21	2010-01-25 14:46:21	55

NA indicates the modification date was unavailable in the document metadata.

Document generation times are exact to the second or millisecond, whereas fonts have a more expansive, frequently "fuzzy" creation timeline. Truncate the document creation time to the closest year in order to align the font creation date with the document creation date. For instance: Verdana: There is a 13 or 17-year time difference between the document date of 2009 or 2017 and the font development year of 1996. For Verdana, created in 1996, and a document generated in 2009, the 13-year difference highlights its legitimate use within the document's timeline. By comparing the years, the impractical goal of measuring font production with sub-second accuracy is avoided.

In 1996, Matthew Carter and Tom Rickner collaborated to design Verdana for Microsoft as part of the ClearType initiative [7].

Max Miedinger and Eduard Hoffmann collaborated to produce Helvetica in 1957 for the Haas Type Foundry in Switzerland. In 1960, it changed its name from Neue Haas Grotesk to Helvetica to better reflect its global appeal [8]. Additional Source: Meggs, Philip B. Meggs' History of Graphic Design. Wiley, 2016. The history and cultural influence of Helvetica are covered in this textbook.

Robin Nicholas and Patricia Saunders created Arial for Monotype Typography in 1982. Arial was created as Helvetica's metrically equivalent substitute. "Monotype" (abbreviated "MT") is the variant that is frequently included in PDFs [9]. Additional Source: Bringhurst, Robert. The Elements of Typographic Style. Hartley & Marks Publishers, 2013, discusses the technical function and design history of Arial.

Stanley Morison and Victor Lardent created Times New Roman on commission for The Times in 1931. Its digital version is called Times-Roman [10]. Additional Source: Morison, Stanley. A Tally of Types. Cambridge University Press, 1973, the author of Times New Roman official overview of Times New Roman.

Howard "Bud" Kettler created Courier for IBM typewriters in 1955 [11]. Additional Source: Eisenstein, Elizabeth L. The Printing Revolution in Early Modern Europe. Cambridge University Press, 1983, explains typewriter fonts and how they have influenced the history of type.

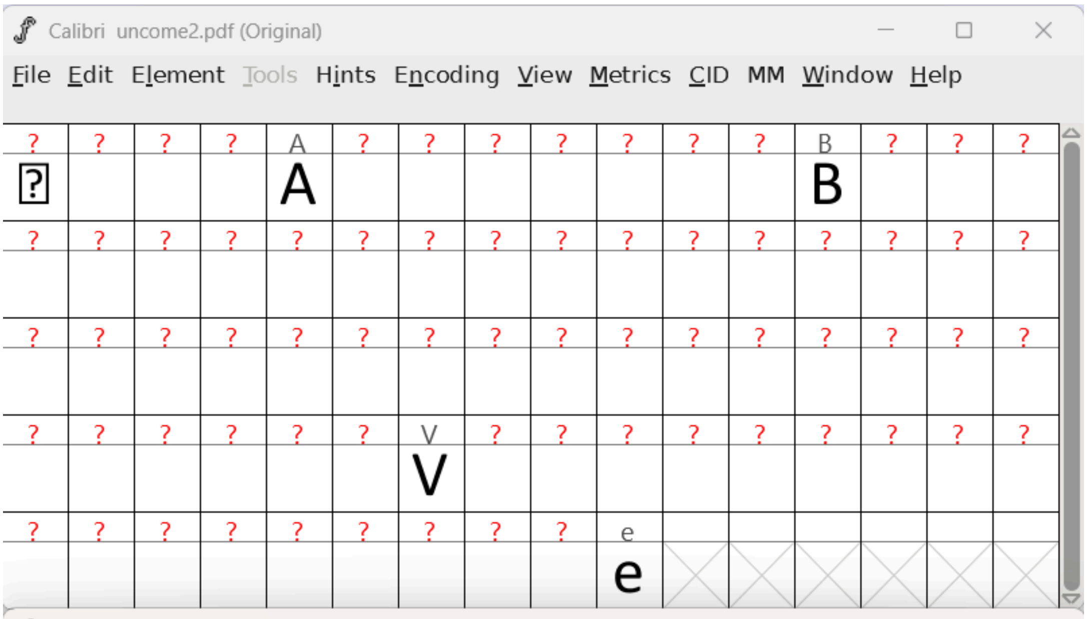
The dates of creation listed in the table match confirmed historical documents for these typefaces. The data is based on verified dates of development and release from reliable sources, including respectable organizations closely linked to these typefaces' past. This guarantees the data's dependability and correctness, enabling it to be used for scholarly purposes. Articles on Wikipedia can be edited by anybody with an internet connection. This implies that people with prejudices or a lack of experience may alter the information, adding errors. This is why other reliable sources or font providers should be referenced or used in court cases to verify dates or information given on Wikipedia.

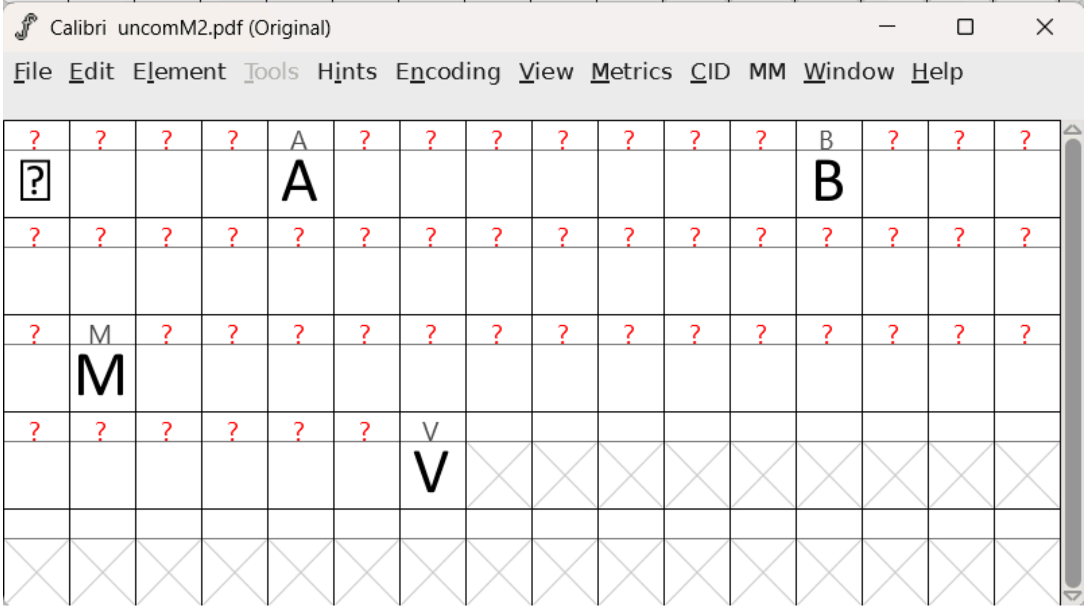
4B: Comparison of Glyph Definitions

Created 2 files one with the text “ABVe” and another with the text “ABVM”, both with the TTF font Calibri
Extracted fonts from the created files using a tool called pdffonts:

```
C:\Users\catac\Downloads>pdffonts uncome2.pdf
name                          type            encoding        emb sub uni object ID
------------------------------ ---------------- ----------------  --- --- ---  ---------
AAAAAA+Calibri        CID TrueType      Identity-H        yes yes yes      8  0
C:\Users\catac\Downloads>pdffonts uncomM2.pdf
name                          type            encoding        emb sub uni object ID
------------------------------ ---------------- ----------------  --- --- ---  ---------
AAAAAA+Calibri        CID TrueType      Identity-H        yes yes yes      8  0
```

Using a tool called FontForge, I compared the 2 files created character’s glyphs.


Calibri uncome2.pdf (Original)
File Edit Element Tools Hints Encoding View Metrics CID MM Window Help
A
B
V
e


Calibri uncomM2.pdf (Original)
File Edit Element Tools Hints Encoding View Metrics CID MM Window Help
A
B
M
V

In the image above, it can be seen that the glyphs look similar, and comparing it to the files created, the character glyphs look the same as well.

Table 5: Embedded Font Glymphs Comparison

Font	File 1	File 2	Identical Glyphs (Yes/No)
Verdana	QFKWG (Referenced at least 3 times in document metadata)	5M56W (Referenced at least 2 times in document metadata)	YES
Helvetica	6C4F5 (Referenced at least 2 times in document metadata)	QNKRV (Referenced atleast 2 times in document metadata)	YES
ArialMT	GGGXC	64PJR	YES

	(Referenced at least 2 times in document metadata)	(Referenced at least 11 times in document metadata)	
Time-Roman	OAAZZ (Referenced at least 3 times in document metadata)	DT6DE (Referenced at least 3 times in document metadata)	YES
Courier	Z2PBT (Referenced at least 3 times in document metadata)	37RZ6 (Referenced at least 4 times in document metadata)	YES

All comparisons are attached in Appendix C.

An overview of the results

Table 5 contrasts the representations of typeface glyphs in two files. We looked at fonts like Courier, Helvetica, ArialMT, Time-Roman, and Verdana. File 1 and File 2 list the glyph representations of each font (in encoded or hexadecimal format). Identical (Yes/No) specifies if each font's glyph representations are the same in the two files. According to the results, the glyph representations for every font under analysis appear to be the same, suggesting that the fonts' representations are constant across the PDF files under investigation.

Extraction of Fonts:

Using programs like pdffonts, and a tool called FontForge, fonts were taken out of two PDF documents.

To ensure comparability, the emphasis was on certain fonts that were used in both documents.

Comparison of Glyph:

Character glyphs (such as 'A') were taken out of the two typefaces. The fonts were compared using the tool FontForge and the results are attached in Appendix C. The fonts were also manually compared in the document themselves taking the 1st 3 characters of each FontForge result that matches each other (starting with alphabetic characters and then numerical characters), attached in Appendix C, to compare the character glyphs to the document itself.

The glyph representations for the chosen characters in every font under examination were the same, indicating that there were no variations in font rendering or embedding between the two PDF files. This would suggest that the font versions used in the two PDFs are identical or that the glyphs were substituted in the same manner when the PDF was created. The consistency in glyphs suggests that both PDFs used the same font version. This supports the integrity of font embedding during PDF creation.

The analysis's main objectives were to compare the glyph representations of certain fonts between two PDF files and investigate the connection between font creation dates and document creation/modification dates. The study affirms that the analyzed PDF files exhibit consistent font and glyph representations and emphasizes the significance of comprehending font history in the context of document generation. This uniformity increases trust in the dependability of the font-handling systems in the PDFs. These results guarantee validity and comparability in font representation research and offer a strong foundation for further document analysis.

References

[1] ISO (2008). Document management – Portable document format – Part 1: PDF 1.7 (ISO 32000-1:2008). Geneva: International Organization for Standardization.

[2] Wikipedia (n.d.) *Calibri*. Available at: https://en.wikipedia.org/wiki/Calibri (Accessed: 21 November 2024).

[3] Pardes, A. (2021) 'Even Calibri’s Creator Is Glad That Microsoft Is Moving On', *WIRED*, 1 May. Available at: https://www.wired.com/story/calibri-default-font-microsoft-moving-on/ (Accessed: 21 November 2024).

[4] Daniels, S. (2023) 'A change of typeface: Microsoft’s new default font has arrived', *Microsoft Design*, 19 July. Available at: https://microsoft.design/articles/a-change-of-typeface-microsoft-s-new-default-font-has-arrived/ (Accessed: 21 November 2024).

[5] Warren, T. (2023) 'Meet Microsoft Office’s new default font: Aptos', *The Verge*, 13 July. Available at: https://www.theverge.com/2023/7/13/23793428/microsoft-aptos-new-default-font-office-365 (Accessed: 21 November 2024).

[6] Cunningham, A. (2023) 'So long, Calibri: Microsoft has settled on a new font for its Office apps', *Ars Technica*, 14 July. Available at: https://arstechnica.com/gadgets/2023/07/microsoft-changes-default-font-in-word-and-other-apps-for-the-first-time-since-2007/ (Accessed: 21 November 2024).

[7] Microsoft (2022) 'Verdana font family', *Microsoft Learn*, 30 March. Available at: https://learn.microsoft.com/en-us/typography/font-list/verdana (Accessed: 21 November 2024).

[8] Linotype (n.d.) 'Helvetica: The world's most popular font', *Linotype Helvetica Page*. Available at: https://www.linotype.com/48344/helvetica-world-family.html (Accessed: 21 November 2024).

[9] Microsoft (2022) 'Arial font family', *Microsoft Learn*, 30 March. Available at: https://learn.microsoft.com/en-us/typography/font-list/arial (Accessed: 21 November 2024).

[10] Microsoft (2022) 'Times New Roman font family', *Microsoft Learn*, 30 March. Available at: https://learn.microsoft.com/en-us/typography/font-list/times-new-roman (Accessed: 21 November 2024).

[11] Prepressure (n.d.) 'Courier', *Prepressure*. Available at: https://www.prepressure.com/fonts/interesting/courier#:~:text=The%20history%20of%20Courier,the%20standard%20typeface%20for%20typewriters (Accessed: 21 November 2024).

Appendix A

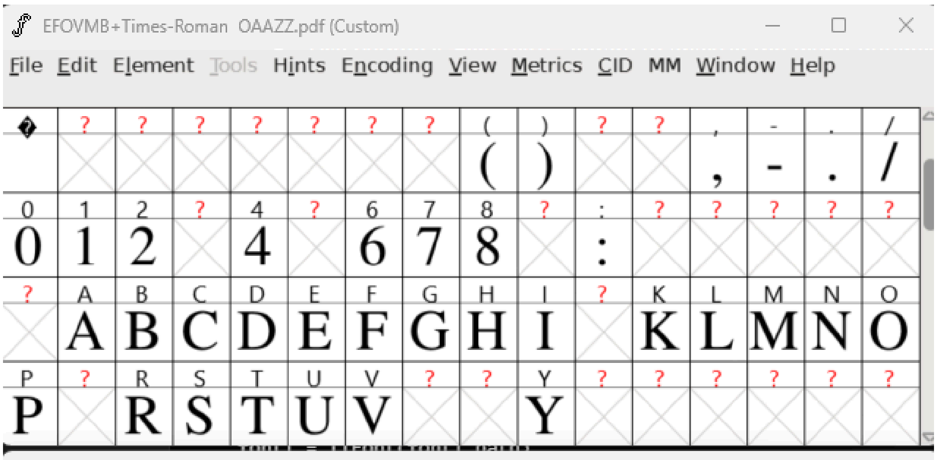
```
import os
def split_pdf_into_parts(file_path):
    with open(file_path, 'rb') as file:
        content = file.read()
    eof_marker = b'%%EOF'
    parts = []
    eof_positions = [pos for pos in range(len(content)) if content.startswith(eof_marker, pos)]
    start = 0
    for pos in eof_positions:
        parts.append(content[start:pos + len(eof_marker)])
        start = pos + len(eof_marker)

    return parts

def count_producer_tags(content):
    count_producer = content.count(b'/Producer')
    count_pdf_producer = content.count(b'<pdf:Producer>')
    return count_producer + count_pdf_producer

def analyze_pdf_files(folder_path, output_csv="producer_analysis.csv"):
    results = []
    for file_name in os.listdir(folder_path):
        if file_name.endswith(".pdf"):
            file_path = os.path.join(folder_path, file_name)
            parts = split_pdf_into_parts(file_path)
            for i, part in enumerate(parts):
                eof_marker_count = part.count(b'%%EOF')
                producer_count = count_producer_tags(part)
                results.append({
                    "Original File Name": file_name,
                    "Part Number": i + 1,
                    "EOF Marker Count": eof_marker_count,
                    "Producer Count": producer_count,
                })
    import pandas as pd
    results_df = pd.DataFrame(results)
    results_df.to_csv(output_csv, index=False)
    return results_df

folder_path = 'C:/data/'
output_csv = 'C:/outputs_exam/producer_analysis3.csv'
results_df = analyze_pdf_files(folder_path, output_csv)
results_df
```

Appendix B

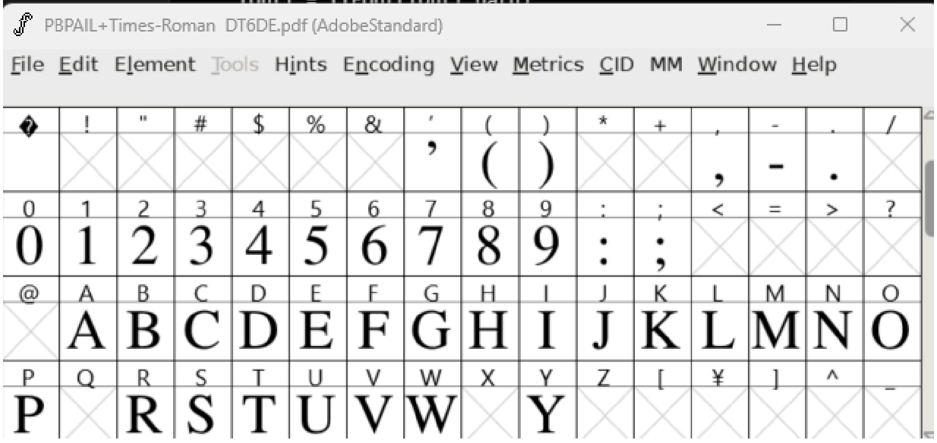
```
from PyPDF2 import PdfReader
def split_pdf_into_parts(file_path):
    with open(file_path, 'rb') as file:
        content = file.read()
    eof_marker = b'%%EOF'
    parts = []
    eof_positions = [pos for pos in range(len(content)) if content.startswith(eof_marker, pos)]
    start = 0
    for pos in eof_positions:
        parts.append(content[start:pos + len(eof_marker)])
        start = pos + len(eof_marker)
    return parts

def save_parts(parts, output_directory="output_parts"):
    import os
    os.makedirs(output_directory, exist_ok=True)
    for i, part in enumerate(parts):
        file_name = f"{output_directory}/part_{i+1}.pdf"
        with open(file_name, 'wb') as output_file:
            output_file.write(part)
        print(f"Saved: {file_name}")

def analyze_metadata(parts, output_directory="output_parts"):
    for i, part in enumerate(parts):
        temp_file = f"{output_directory}/part_{i+1}.pdf"
        with open(temp_file, 'wb') as output_file:
            output_file.write(part)
        try:
            reader = PdfReader(temp_file)
            metadata = reader.metadata
            print(f"Metadata for part {i+1}: {metadata}")
        except Exception as e:
            print(f"Could not read metadata for part {i+1}: {e}")

file_path = 'C:/Users/catac/Documents/TUKS/4. 721 -Security
2/data/22ZOCVPAF2GSGXR357RM7UT4Z22RS2LH.pdf'
parts = split_pdf_into_parts(file_path)
print(f"Number of parts found: {len(parts)}")
save_parts(parts)
analyze_metadata(parts)
```

Appendix C

Font: Times-Roman

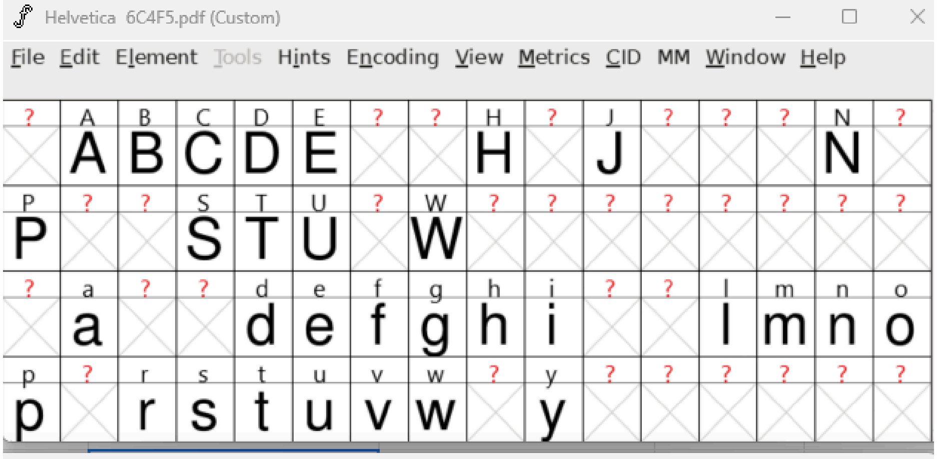

EFOVMB+Times-Roman OAAZZ.pdf (Custom)
File Edit Element Tools Hints Encoding View Metrics CID MM Window Help

�	?	?	?	?	?	?	?	(	)	?	?	,	-	.	/
								(	)			,	-	.	/
0	1	2	?	4	?	6	7	8	?	:	?	?	?	?	?
0	1	2		4		6	7	8		:					
?	A	B	C	D	E	F	G	H	I	?	K	L	M	N	O
	A	B	C	D	E	F	G	H	I		K	L	M	N	O
P	?	R	S	T	U	V	?	?	Y	?	?	?	?	?	?
P		R	S	T	U	V			Y						

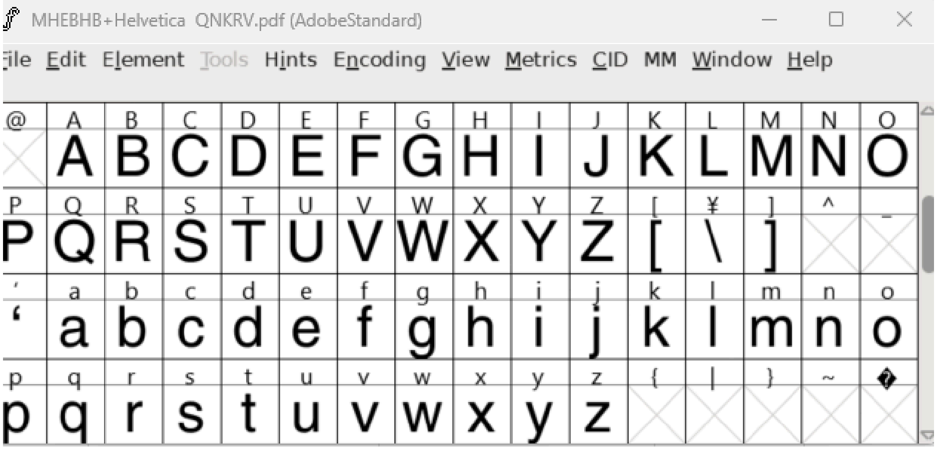

PBPAIL+Times-Roman DT6DE.pdf (AdobeStandard)
File Edit Element Tools Hints Encoding View Metrics CID MM Window Help

�	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
							’	(	)			,	-	.	
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0	1	2	3	4	5	6	7	8	9	:	;				
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W	X	Y	Z	[	¥	]	^	_
P		R	S	T	U	V	W		Y						

Font: Helvetica

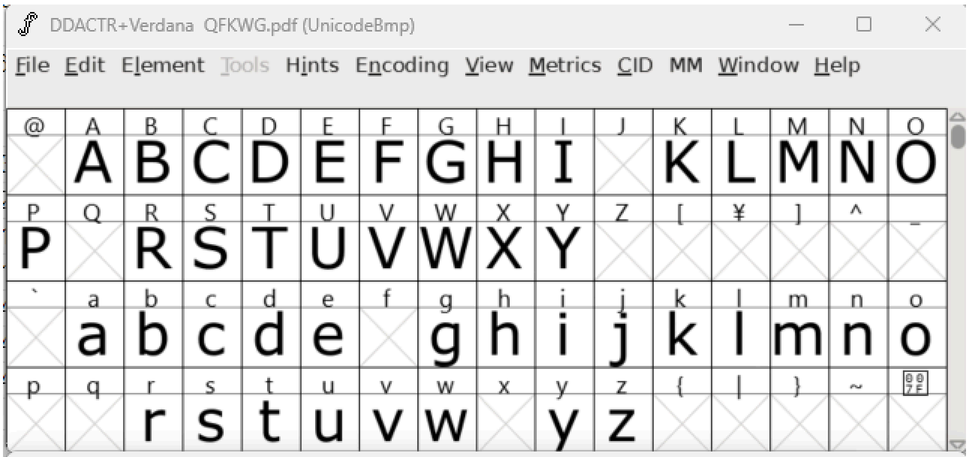

Helvetica 6C4F5.pdf (Custom)
File Edit Element Tools Hints Encoding View Metrics CID MM Window Help

?	A	B	C	D	E	?	?	H	?	J	?	?	?	N	?
	A	B	C	D	E			H		J				N	
P	?	?	S	T	U	?	W	?	?	?	?	?	?	?	?
P			S	T	U		W								
?	a	?	?	d	e	f	g	h	i	?	?	l	m	n	o
	a			d	e	f	g	h	i			l	m	n	o
p	?	r	s	t	u	v	w	?	y	?	?	?	?	?	?
p		r	s	t	u	v	w		y						


MHEBHB+Helvetica QNKRV.pdf (AdobeStandard)
ile Edit Element Tools Hints Encoding View Metrics CID MM Window Help

@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W	X	Y	Z	[	¥	]	^	_
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]		
'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
‘	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	�
p	q	r	s	t	u	v	w	x	y	z					

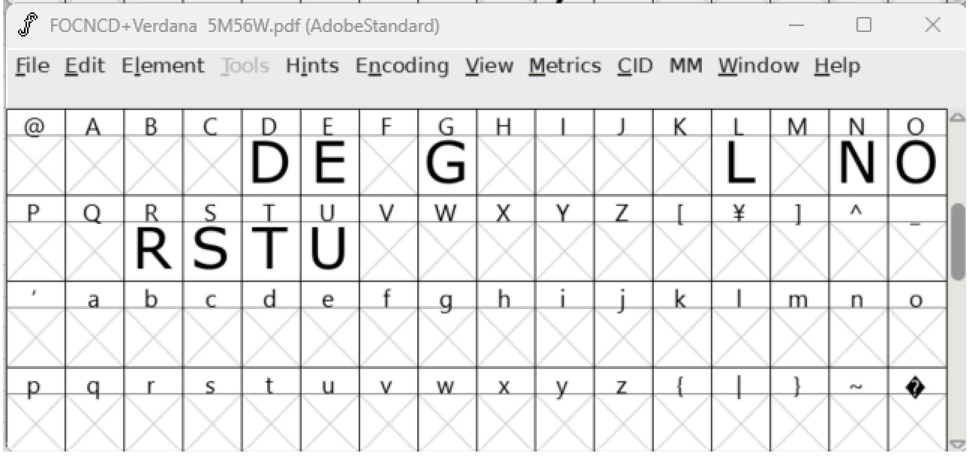
Font: Verdana



DDACTR+Verdana QFKWG.pdf (UnicodeBmp)

File Edit Element Tools Hints Encoding View Metrics CID MM Window Help

@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	A	B	C	D	E	F	G	H	I		K	L	M	N	O
P	Q	R	S	T	U	V	W	X	Y	Z	[	¥	]	^	_
P		R	S	T	U	V	W	X	Y						
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
	a	b	c	d	e		g	h	i	j	k	l	m	n	o
p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	00 7F
		r	s	t	u	v	w		y	z					

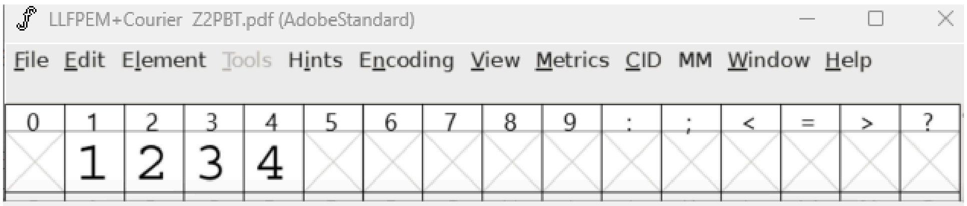


FOCNCD+Verdana 5M56W.pdf (AdobeStandard)

File Edit Element Tools Hints Encoding View Metrics CID MM Window Help

@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
				D	E		G					L		N	O
P	Q	R	S	T	U	V	W	X	Y	Z	[	¥	]	^	_
		R	S	T	U										
'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	�

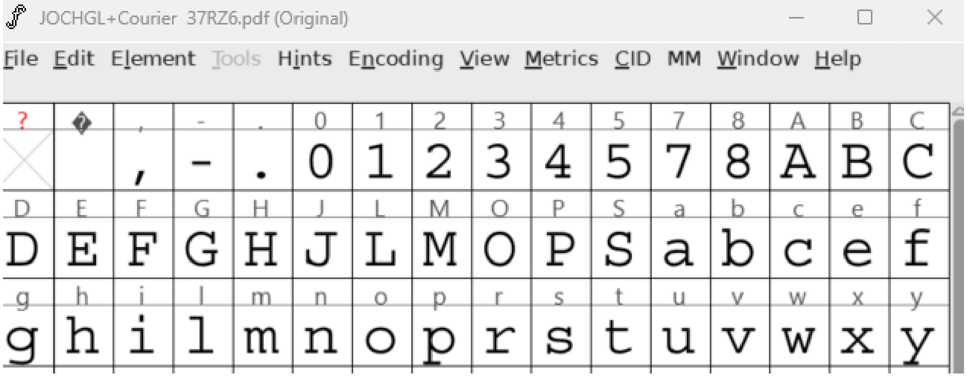
Font: Courier



LLFPEM+Courier Z2PBT.pdf (AdobeStandard)

File Edit Element Tools Hints Encoding View Metrics CID MM Window Help

0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
	1	2	3	4											

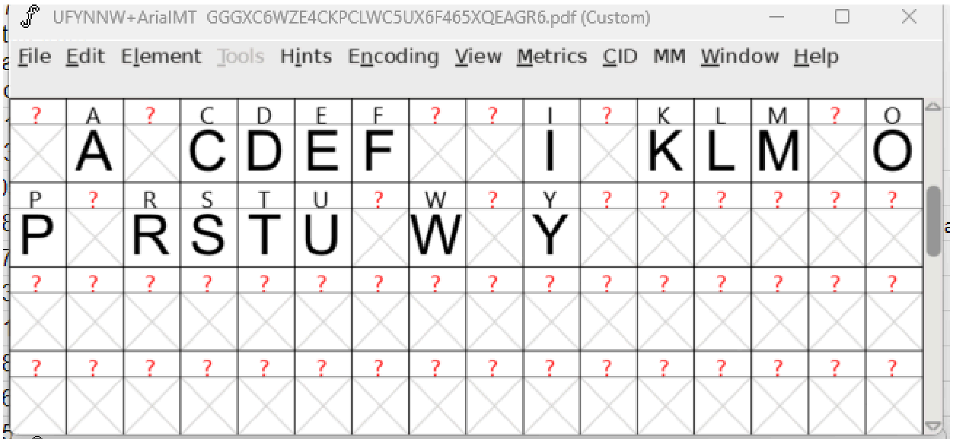


JOCHGL+Courier 37RZ6.pdf (Original)

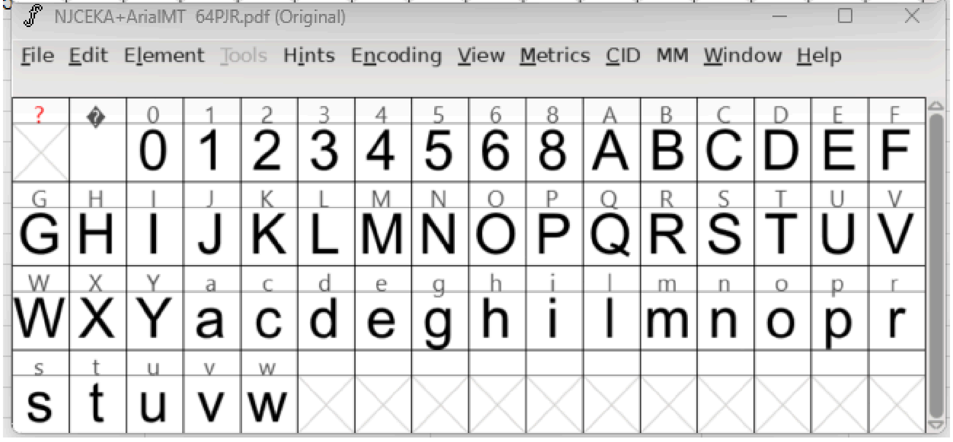
File Edit Element Tools Hints Encoding View Metrics CID MM Window Help

?	�	,	-	.	0	1	2	3	4	5	7	8	A	B	C
		,	-	.	0	1	2	3	4	5	7	8	A	B	C
D	E	F	G	H	J	L	M	O	P	S	a	b	c	e	f
D	E	F	G	H	J	L	M	O	P	S	a	b	c	e	f
g	h	i	l	m	n	o	p	r	s	t	u	v	w	x	y
g	h	i	l	m	n	o	p	r	s	t	u	v	w	x	y

Font: ArialMT


UFYNNW+ArialMT GGGXC6WZE4CKPCLWC5UX6F465XQEAGR6.pdf (Custom)
File Edit Element Tools Hints Encoding View Metrics CID MM Window Help

?	A	?	C	D	E	F	?	?	I	?	K	L	M	?	O
P	?	R	S	T	U	?	W	?	Y	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?


NJCEKA+ArialMT 64PJR.pdf (Original)
File Edit Element Tools Hints Encoding View Metrics CID MM Window Help

?	�	0	1	2	3	4	5	6	8	A	B	C	D	E	F
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
W	X	Y	a	c	d	e	g	h	i	l	m	n	o	p	r
s	t	u	v	w											