Comprehensive Analysis

Obama LFBC PDF File

Garrett Papit

Table of Contents

I. Introduction

When the White House released a digital image of President Obama's long-form birth certificate last year, many were quick to point out what they considered to be anomalies. One of the major issues causing controversy is the existence of layers when the PDF file is opened within a graphics program such as Adobe Illustrator. Many state that the layers are the result of manual manipulation within a graphics editing application; while the opposing argument states that the layers are a natural attribute of optimization when saving the file as a PDF. Optimization would be applied to a file to make the file size smaller for easier download.

The only explanations provided for the existence of the layers, by either side of the debate, are optimization or manual manipulation (OCR having already been ruled out). Either someone built the composite image from many separate images of unknown sources, or an automated software application innocently segmented the file into many separate images.

Being a computer professional, I decided to investigate the facts involved to see if it was possible to determine if optimization created the layers in the White House PDF file. My qualifications include a Bachelor's in Computer Information Systems and a MBA with a concentration in Managerial Information Systems. I specialize in application development and systems analysis and work as a programmer of .com applications for a Fortune 500 company. The main focus of my work involves web and desktop application development utilizing various programming languages. I am also familiar with the methodology involved in PDF optimization and compression. Based on this experience, I proceeded to analyze the PDF file posted on the White House website, including the metadata and source code which had not been an area of major focus previously.

My goal was to try and determine whether the same type of layers on the WH PDF file could be created by optimization. Every effort was taken to be as objective and fair-minded as possible in attempting to replicate the characteristics of the layers on that file.

II. Optimization Overview

First, we will examine what PDF optimization does and why. It is important to understand how the functionality works and what it does to a scanned image. When a document is scanned the result is a flat image file. Flat, in this context, just means that it doesn't have layers. Layers refer to separate images that are stacked, like sheets of transparency, to make a composite image.

Optimization is simply the process of compressing a PDF file for easier use. Many large PDF files need to be optimized for quicker web view and download. The goal is to reduce the file size, increase the speed at which the image can be loaded and to save network bandwidth.

Many optimization routines include a step called segmentation. Segmentation is the process of separating the elements of a PDF image so that each can be compressed more effectively. Text and graphic elements, such as form lines, are separated from the background and images so that different compression algorithms can be applied to each layer. This results in a greater reduction in file size than if the entire image was compressed using the same method. Segmentation is the reason that optimization can create layers.

There are two major types of optimization that utilize segmentation. The first is Adaptive Compression which is a proprietary optimization method used by Adobe Acrobat Pro. The second is Mixed Raster Content (MRC) Compression, which is used by various other applications such as CVista, LizardTech, LuraTech, Nuance and many others. Both are similar in their functionality, but they handle color text differently.

Adaptive Compression

First we will look at an example of how Adaptive Compression works using the sample image to the right (Figure 1). Adobe Acrobat will attempt to separate the picture into black-and-white, color and grayscale layers. How successful the software is at segmenting the image depends on the complexity of the image and the quality of the print and scan. Once segmented, the individual layers will be compressed with the most appropriate methods.

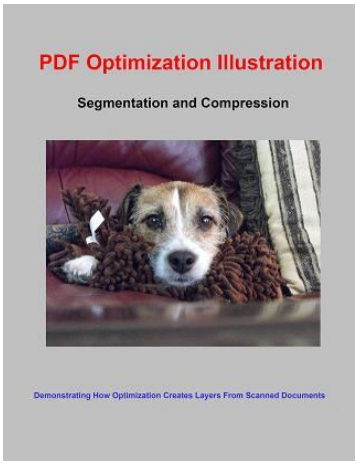


Figure 1 - Sample scan

- The foreground layer (Figure 2) is a transparent layer consisting of the text and near-black elements such as form lines.
- There is only 1 single transparent "text"* layer.

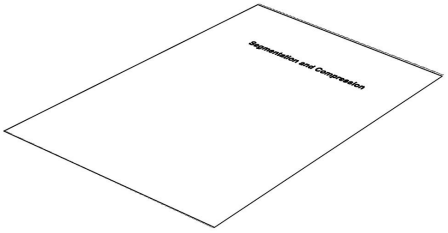


Figure 2 – Foreground layer

- The background layer (Figure 3) consists of background, images and color text.
- The background can be split into many separate layers.

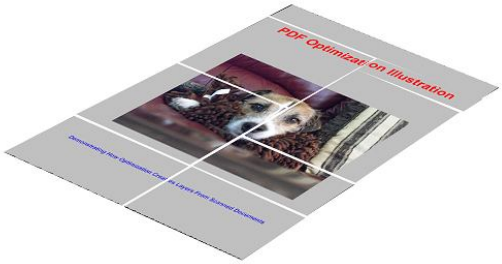


Figure 3 – Background layer

** Note: The "text" layer can contain other elements and may not always contain text if the text is not dark enough. However, for the sake of simplicity it will be referred to as the text layer going forward. Also, the term transparent refers to the fact that the layer is a 1-bit binary image mask isolated from the background.*

The image below (Figure 4) provides an overview of the Adaptive Compression process.

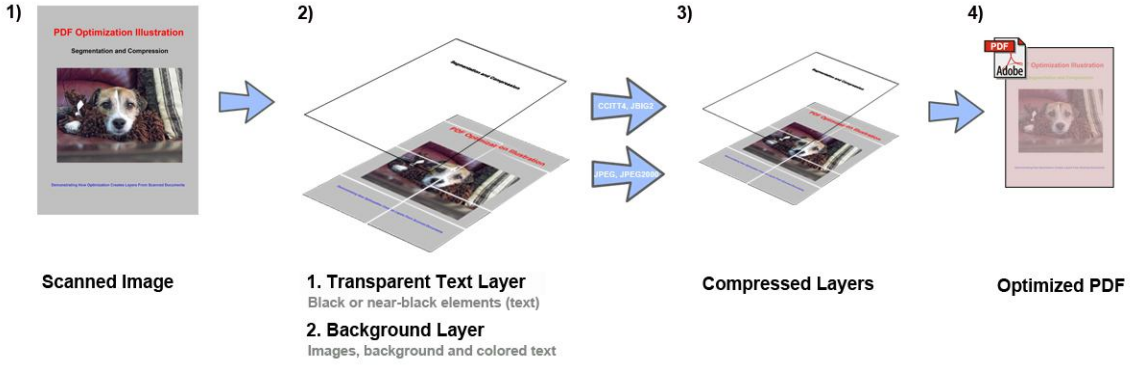


Figure 4 – Adaptive Compression overview

1. Start with a scanned document image consisting of background, images and text.

2. Segment the image based on content. The background layer will contain images, background and color text. It may be broken into many layers, as loading several small images is easier than loading one large one. The single foreground layer is a transparent layer consisting of text and near-black elements such as form lines.

3. Each layer is compressed using the most effective algorithm for the type of content it contains. This results in compressed layers that are smaller in file size.

4. The compressed layers are stacked and saved as an optimized PDF file.

MRC Compression

Next we will look at a sample of MRC Compression using the same sample image. Again, the image will be segmented based on content type but color text, and color graphic elements, will be handled differently.

- The foreground layer (Figure 5), or color layer, contains color information for the text and graphic elements, such as form lines.
- There is only 1 single color layer.

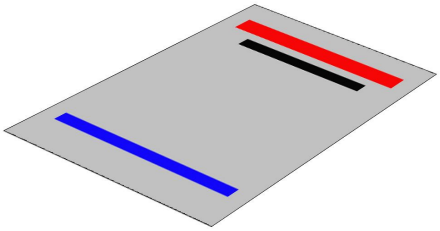


Figure 5 – Foreground layer

- The transparent text layer (Figure 6) contains the shapes of the text and graphic elements such as form lines. Unlike with Adaptive Compression, it also contains color text but the color is removed and stored on the foreground color layer
- There is only 1 single transparent text layer.

Figure 6 – Text layer

- The background layer (Figure 7) consists of the background and any images.
- Although Adaptive can have many background layers, MRC normally just has one single background layer.

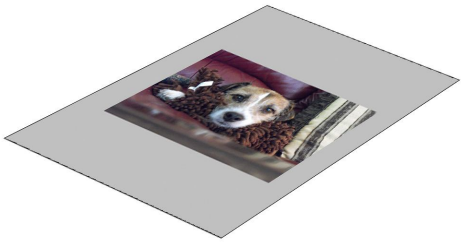


Figure 7 – Background layer

The image below (Figure 8) provides an overview of the MRC Compression process.

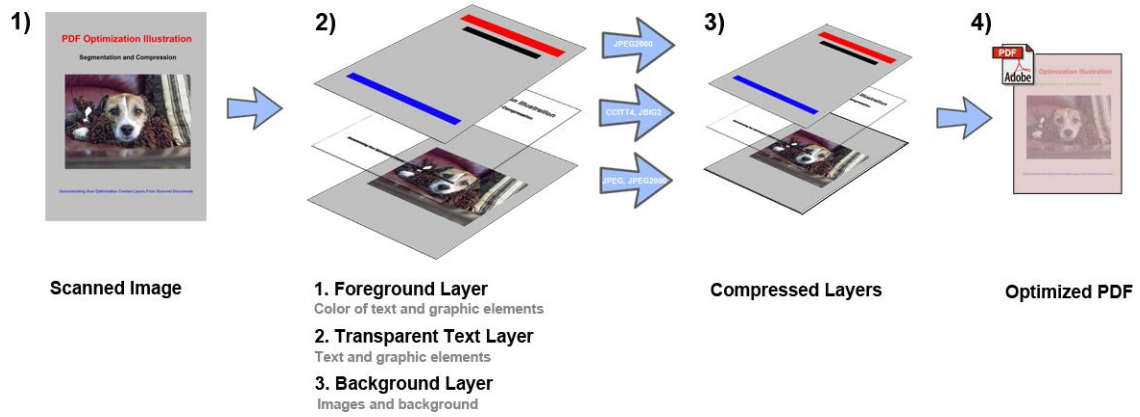


Figure 8 - MRC Compression Overview

1. Start with a scanned document image consisting of background, images and text.

2. Segment the image based on content. The background layer will contain images and background. The transparent text layer consists of text and graphic elements such as form lines. This layer includes color text, but the color is removed. The foreground layer contains the color information for the elements on the text layer. The text layer acts as a stencil. Anyplace where text shape exists, the color from the foreground layer will show through.

3. Each layer is compressed using the most effective algorithm for the type of content it contains. This results in three compressed layers that are smaller in file size.

4. The compressed layers are stacked and saved as an optimized PDF file.

Below is a list of similarities and differences between Adaptive and MRC Compression. Again, these are the two types of PDF optimization that can result in a file with layers.

Similarities

- Both try to place text on a separate layer.
- Both contain only 1 single transparent text layer.

Differences

- Adaptive only separates near-black text; MRC also handles color text.
- Adaptive converts all near-black text to black; MRC keeps original color using color layer.

Having provided an overview of how the process works, we will now further examine the attributes of the text layer and background layers. These facts will become important when performing a comparative analysis on the White House PDF file. Both Adaptive and MRC Compression share all of the following properties.

Text Layer

- The text layer is always 1-bit monochrome.
- 1-bit refers to the color depth and means there are only 2 colors, usually black and white.
- An example of 1-bit color depth is highlighted in red in the image to the right.
- The text layer is transparent, meaning it is isolated from the background. The text and graphic elements such as form lines are the only non-transparent elements.
- There is only 1 single transparent text layer.

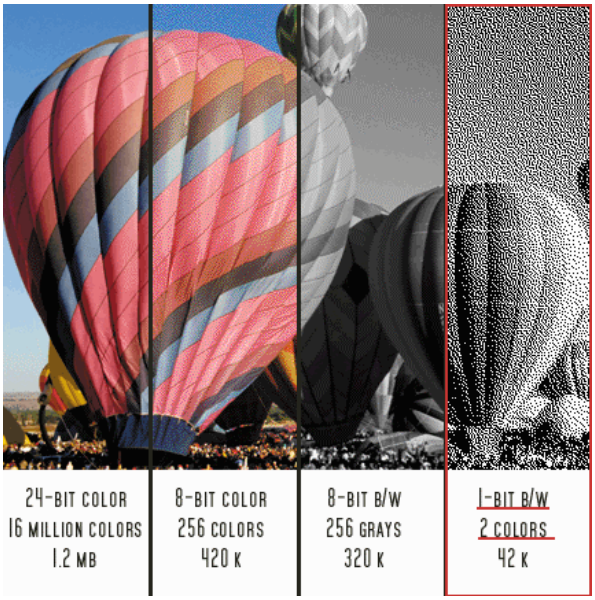


Figure 9 – 1-bit color example

Background Layer

- The background layer has 8-bit color, meaning 256 colors, as shown on the right.
- With MRC there is normally 1 single background layer.
- With Adaptive the background can be segmented into many layers depending on the settings used.

Figure 10 – 8-bit color example

III. Control Document Creation

In order to adequately test and analyze the effects of optimization, a control document was created that would be as close as possible to a birth certificate issued by the Hawaii Department of Health. The form data would have been printed onto green safety paper, and then the date stamp and registrar stamp would be manually added via a traditional ink stamp to certify the document.

At the March 1 press conference, the Cold Case Posse pointed out that the registrar signature stamp and date stamp were on separate isolated layers. It has been suggested that this occurred due to the fact that these two items were stamped separately with an inkpad and not generated by a printer (as was the rest of the form information). The theory that a computer can somehow pick up the differences in ink, or color, resulting in those areas being separated cleanly onto their own layers, needed to be adequately tested and examined.

To that end all of the form data, except for the registrar's date stamp, was printed onto a piece of green basket-weave safety paper. The date stamp was left off so that it could be manually applied with a traditional inkpad, as would be the normal process. The finalized control document, including manual date stamp, is shown below (Figure 4).

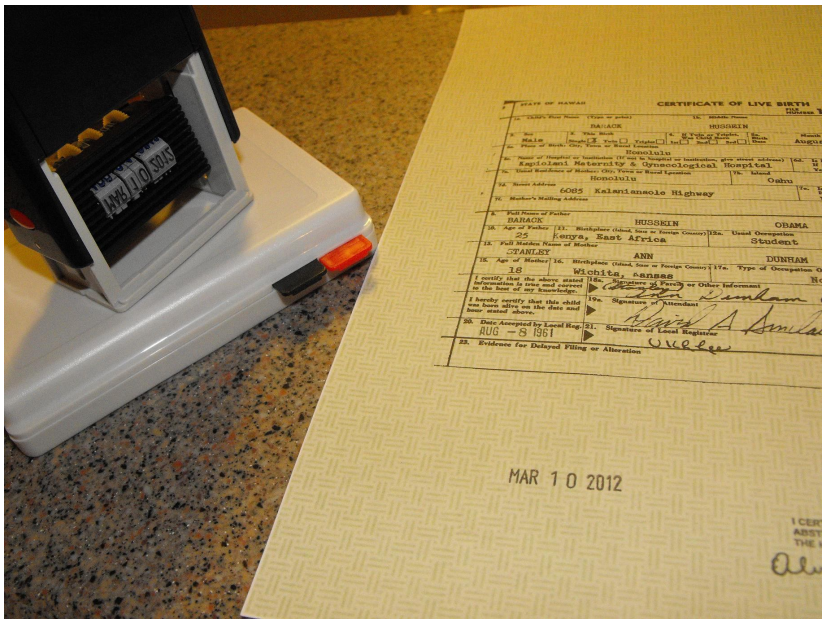


Figure 11 – Control document with manual stamp

In an effort to be as thorough as possible, the control document was printed and scanned using several different printers and scanners ranging from high quality to low quality. The goal was to see if the quality of the document altered the characteristics of the optimized file.

IV. Compare Optimized PDF with White House PDF

Just as in the case with the control document creation, every effort was made to be as thorough as possible in testing and comparing optimized files with the PDF released by the White House. A short description of the testing procedures will help convey the methodology employed for this task.

Testing Procedures

Each control document was painstakingly analyzed during the course of the testing. Every combination of settings was used in order to compare results from each change. Furthermore, a wide array of PDF software was used including ABBYY PDF Transformer, Adobe Acrobat Pro (Versions 6, 7, 8, 9 and X), CVista PDF Compressor, Lead Tools, LizardTech Document Express, LuraTech PDF Compressor, NovaPDF Professional Desktop, Nuance PDF Professional, Pdfforge PDFCreator, Primo PDF, Software602 Print2PDF and many more.

Layers in the White House PDF Document

The White House PDF image is a composite of many different layers, each containing an image element. When the individual images are all laid on top of one another, analogous to stacking transparencies onto a projector with each image printed on a separate transparency sheet, they create the composite birth certificate image. Below are screenshots of the most important layers.

Background Layer

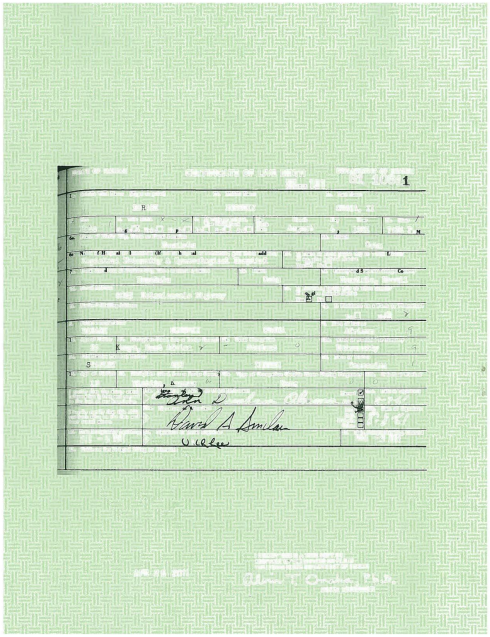
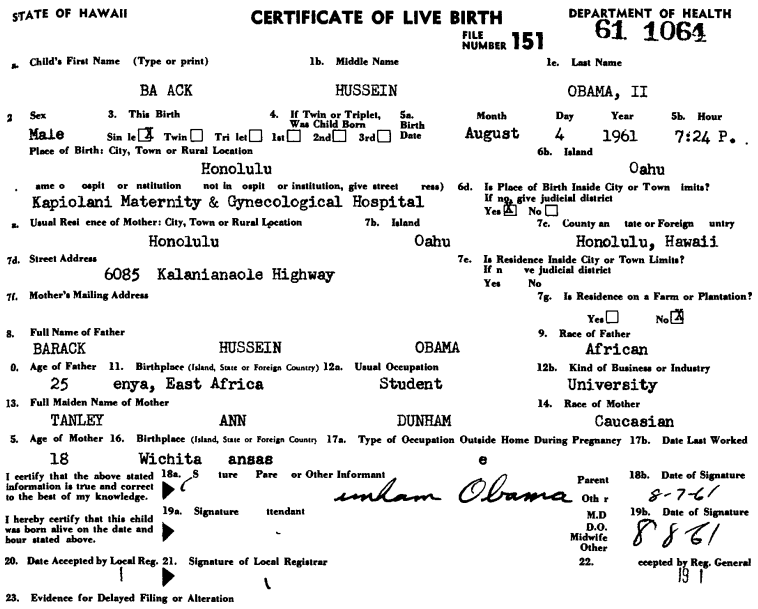


Figure 12 – Safety paper, form lines, doctor's signature and most of local registrar signature

Form data layer



STATE OF HAWAII

CERTIFICATE OF LIVE BIRTH

DEPARTMENT OF HEALTH

FILE NUMBER 151 — 61 1064

1a. Child's First Name (Type or print): BA ACK — 1b. Middle Name: HUSSEIN — 1c. Last Name: OBAMA, II

2 Sex: Male — 3. This Birth: Sin le [X] Twin [ ] Tri let [ ] — 4. If Twin or Triplet, Was Child Born: 1st [ ] 2nd [ ] 3rd [ ] — 5a. Birth Date: Month August, Day 4, Year 1961 — 5b. Hour 7:24 P.

Place of Birth: City, Town or Rural Location: Honolulu — 6b. Island: Oahu

6a. ame o ospit or natitution not in ospit or institution, give street ress): Kapiolani Maternity & Gynecological Hospital — 6d. Is Place of Birth Inside City or Town imits? If no, give judicial district: Yes [X] No [ ]

8a. Usual Resi ence of Mother: City, Town or Rural Location: Honolulu — 7b. Island: Oahu — 7c. County an tate or Foreign untry: Honolulu, Hawaii

7d. Street Address: 6085 Kalanianaole Highway — 7e. Is Residence Inside City or Town Limits? If n ve judicial district: Yes No

7f. Mother's Mailing Address — 7g. Is Residence on a Farm or Plantation? Yes [ ] No [X]

8. Full Name of Father: BARACK HUSSEIN OBAMA — 9. Race of Father: African

10. Age of Father: 25 — 11. Birthplace (Island, State or Foreign Country): enya, East Africa — 12a. Usual Occupation: Student — 12b. Kind of Business or Industry: University

13. Full Maiden Name of Mother: TANLEY ANN DUNHAM — 14. Race of Mother: Caucasian

15. Age of Mother: 18 — 16. Birthplace (Island, State or Foreign Country): Wichita ansas — 17a. Type of Occupation Outside Home During Pregnancy: e — 17b. Date Last Worked

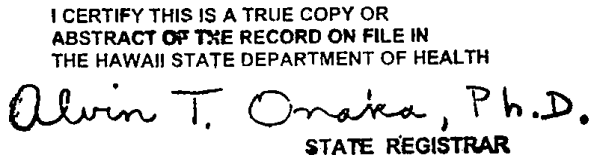
I certify that the above stated information is true and correct to the best of my knowledge. — 18a. S ture Pare or Other Informant: unham Obama — Parent / Oth r — 18b. Date of Signature: 8-7-61

I hereby certify that this child was born alive on the date and hour stated above. — 19a. Signature ttendant — M.D / D.O. / Midwife / Other — 19b. Date of Signature: 8 8 61

20. Date Accepted by Local Reg. — 21. Signature of Local Registrar — 22. ccepted by Reg. General 19 1

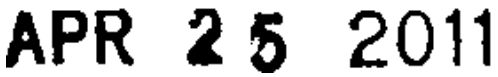
23. Evidence for Delayed Filing or Alteration

Figure 13 – Most of the form data, missing doctor's signature and last digit of BC#



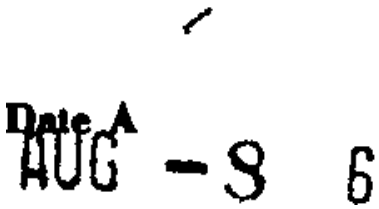
I CERTIFY THIS IS A TRUE COPY OR
ABSTRACT OF THE RECORD ON FILE IN
THE HAWAII STATE DEPARTMENT OF HEALTH
Alvin T. Onaka, Ph.D.
STATE REGISTRAR

Figure 14 - State Registrar's Signature Stamp



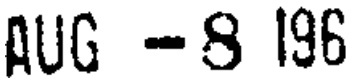
APR 25 2011

Figure 15 - State Registrar's Date Stamp



Date A
AUG -8 6

Figure 16 - Registrar General's Date Stamp



AUG -8 196

Figure 17 - Local Registrar's Date Stamp

Problems

A comprehensive analysis of the layers in the White House PDF file, along with a comparison of optimized PDF files, reveals several inconsistencies. The most significant problems are listed below:

1. There are multiple 1-bit text layers (two are the isolated date and registrar stamps).

2. The form lines are not on the same layer as the majority of the text.

3. The color properties of the text are not consistent with either optimization or scanning.

4. There is a white halo around all text and white space beneath each text layer.

1. Multiple Text Layers

- Recall that an optimized file has only 1 transparent text layer.
- Some text can be on the background layer, but not a 2nd transparent text layer (Figure 18).
- This is true for both Adaptive and MRC Compression.
- There are multiple transparent text layers in the WH PDF.

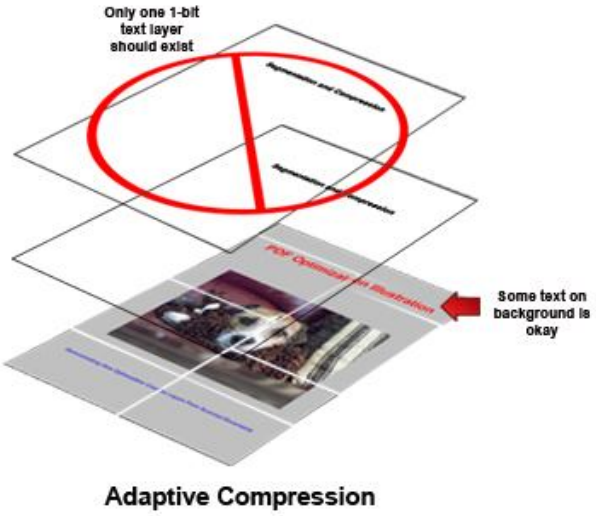


Figure 18 - Only one 1-bit text layer should exist

To better illustrate this point, here is a side-by-side view of the White House PDF and the sample optimized PDF file (Figure 19).

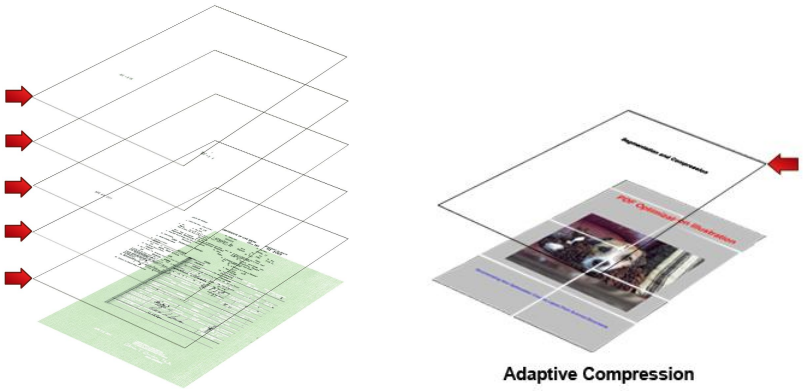


Figure 19 – WH PDF vs. Optimized PDF

The following control files demonstrate how optimization creates only one single 1-bit text layer.

Adaptive Compression Control 1

- This file was generated using a high quality printer and scanner and optimized with Adobe Acrobat.
- The background is in two separate layers (Figure 20) based on the settings used. The higher the quality chosen, the fewer background layers created.
- There is only one transparent text layer (Figure 21).
- The 3D view (Figure 22) illustrates how the layers are positioned.

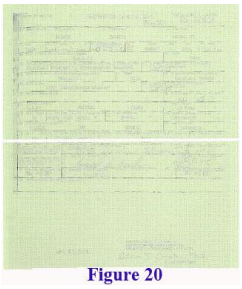


Figure 20

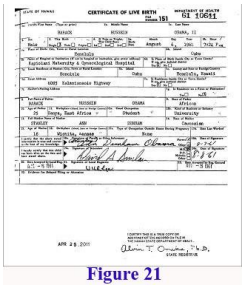


Figure 21

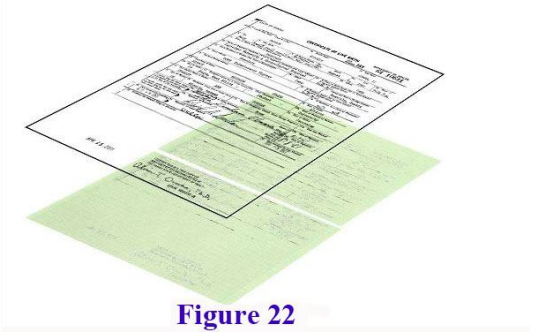


Figure 22

Adaptive Compression Control 2

- This file was generated using a low quality printer and scanner and optimized with Adobe Acrobat.
- Again, the background is in two separate layers (Figure 23). Due to the low quality of the printer, the text is not dark enough and much of it is left on the background layers.
- Again optimization creates only one transparent text layer (Figure 24). It contains the portions of the text that were dark enough. The manual date stamp is not on an isolated layer.
- A 3D view is included for clarification (Figure 25).

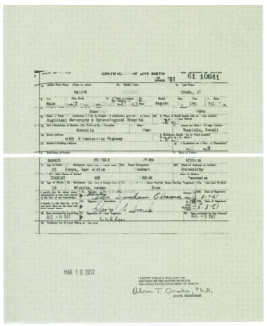


Figure 23

Figure 24

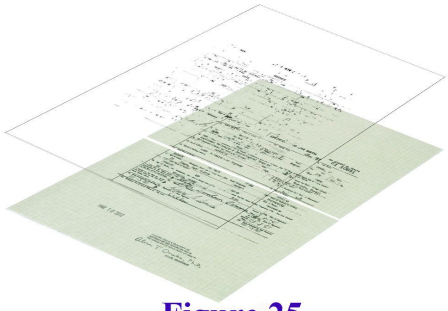


Figure 25

MRC Compression Control

- This file was optimized with an application that utilizes MRC compression techniques.
- The entire background is on one single layer (Figure 26).
- Just like Adaptive Compression, MRC Compression also creates only one transparent text layer (Figure 27). The manual date stamp, which was applied with an inkpad, is not on an isolated text layer.
- In addition, there is an extra layer that stores the color information for the text and form lines (Figure 28).
- The 3D overview shows how all three layers are stacked to make the final PDF file (Figure 29).

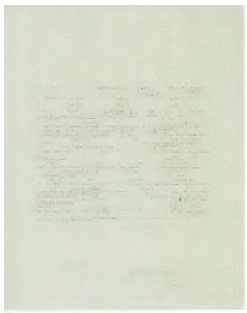


Figure 26

Figure 27

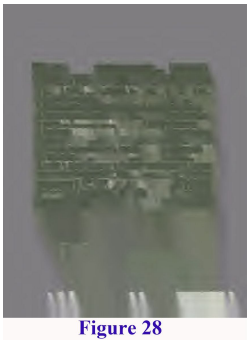


Figure 28

Figure 29

The text layer properties of the file released by the White House do not match the properties of an optimized PDF file. Specifically, an optimized PDF file has only one single 1-bit layer, referred to here as a transparent text layer. This is true of both Adaptive Compression, which is used by Adobe Acrobat, and MRC Compression, which is used by numerous other applications. By contrast, the file released by the White House has many separate isolated text layers. Among these layers are the state registrar's date stamp, state registrar's signature stamp and portions of both the registrar general and local registrar's date stamps. A comparative image of the control files and the White House file is provided below (Figure 30).

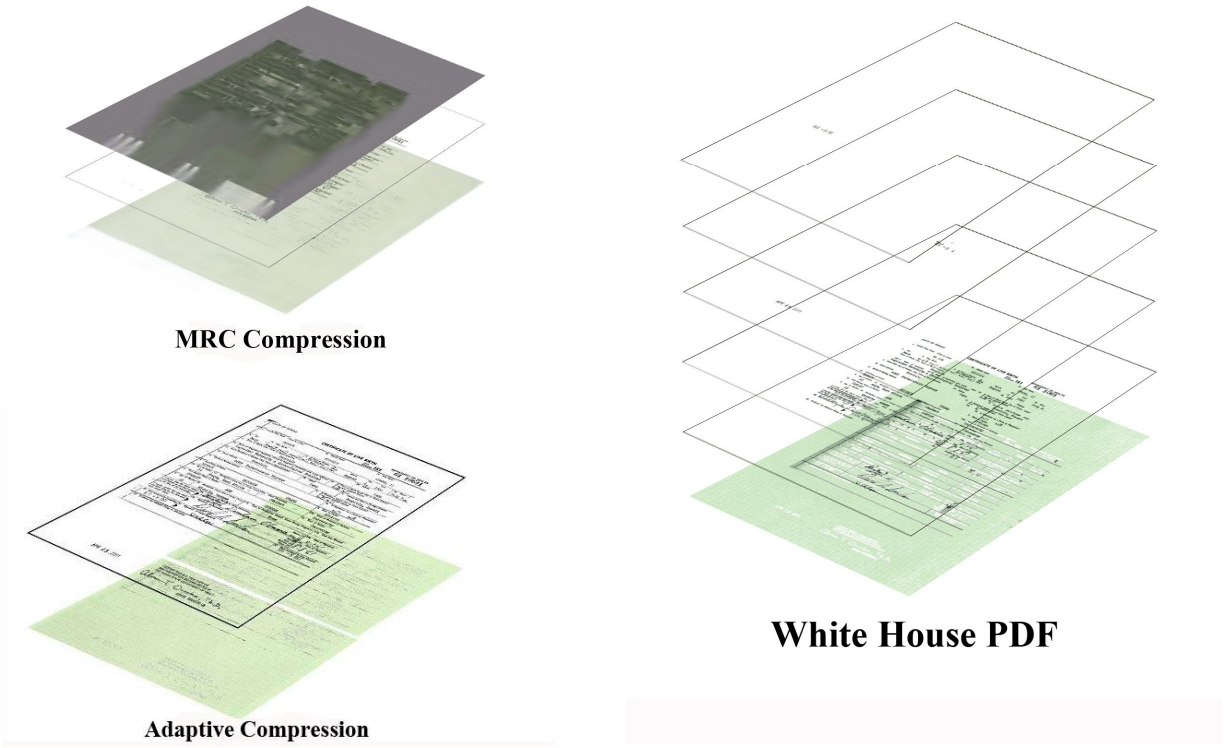


Figure 30 – Visual comparison of text layers

Note: More evidence related to the multiple text layer problem will be presented in the object code analysis section later in this document.

2. Form Lines

- Optimization treats black, or near-black, form lines like text.
- If most of the text is separated from the background, most of the form lines should be as well. If only some of the text is separated, only some of the form lines will be. The text and form lines should be evenly distributed between the layers.
- This is true for both Adaptive and MRC Compression.
- The form lines are entirely on the background layer in the WH PDF even though most text is not (Figure 31).

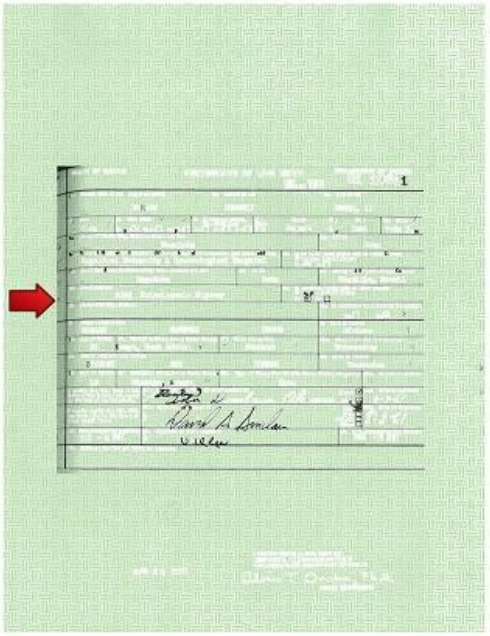


Figure 31 – Form lines on background

Again, the following control files will help illustrate the problem.

Adaptive Compression Control 1

- This file was generated on a high quality printer and scanner and optimized using Adobe Acrobat.
- Acrobat segments black or near-black portions of the image.
- Only remnants of the form lines are on the background layers (Figure 32). The same holds true for the text, where only remnants can be found on the background.
- Rather, the form lines are almost entirely on the single transparent text layer (Figure 33).

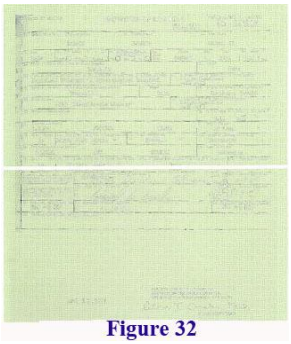


Figure 32

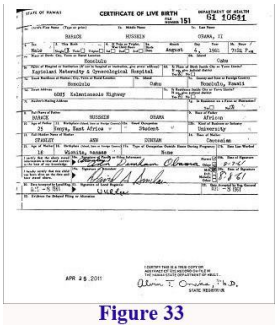


Figure 33

Adaptive Compression Control 2

- This file was also optimized with Adobe Acrobat, but was generated on a low quality printer/scanner.
- Due to the low quality print, much of the text was left on the background layer. Likewise, large portions of the form lines are on the background as well.
- Some bits of form lines are still on the text layer, just like random parts of the text are.
- The key is that text and form lines should be evenly distributed throughout the layers.

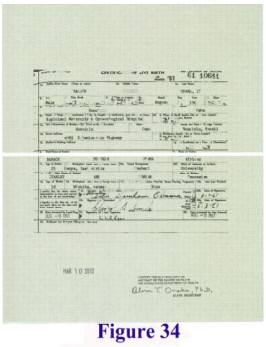


Figure 34

Figure 35

MRC Compression Control

- This file was optimized using MRC Compression.
- MRC Compression separates text and graphic elements, such as form lines, regardless of their color.
- As a result, there are no form lines on the background layer (Figure 36).
- Instead, the form lines are completely on the single text layer along with all text (Figure 37).

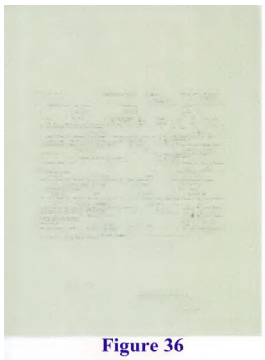


Figure 36

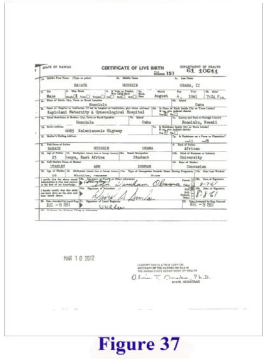


Figure 37

Much like the text layers properties, the location of the form lines on the PDF file released by the White House is not consistent with optimization. As the control files demonstrate, the form lines should be on the same layer as the majority of the text. Normally this is the 1-bit transparent text layer. However, if the document quality is poor the form lines can be on the background, but much of the text should be as well. Given that the majority of the text on the White House file is not on the background, the form lines should not be on the background in their entirety. A comparative image of the control files and the White House file is provided below (Figure 38).

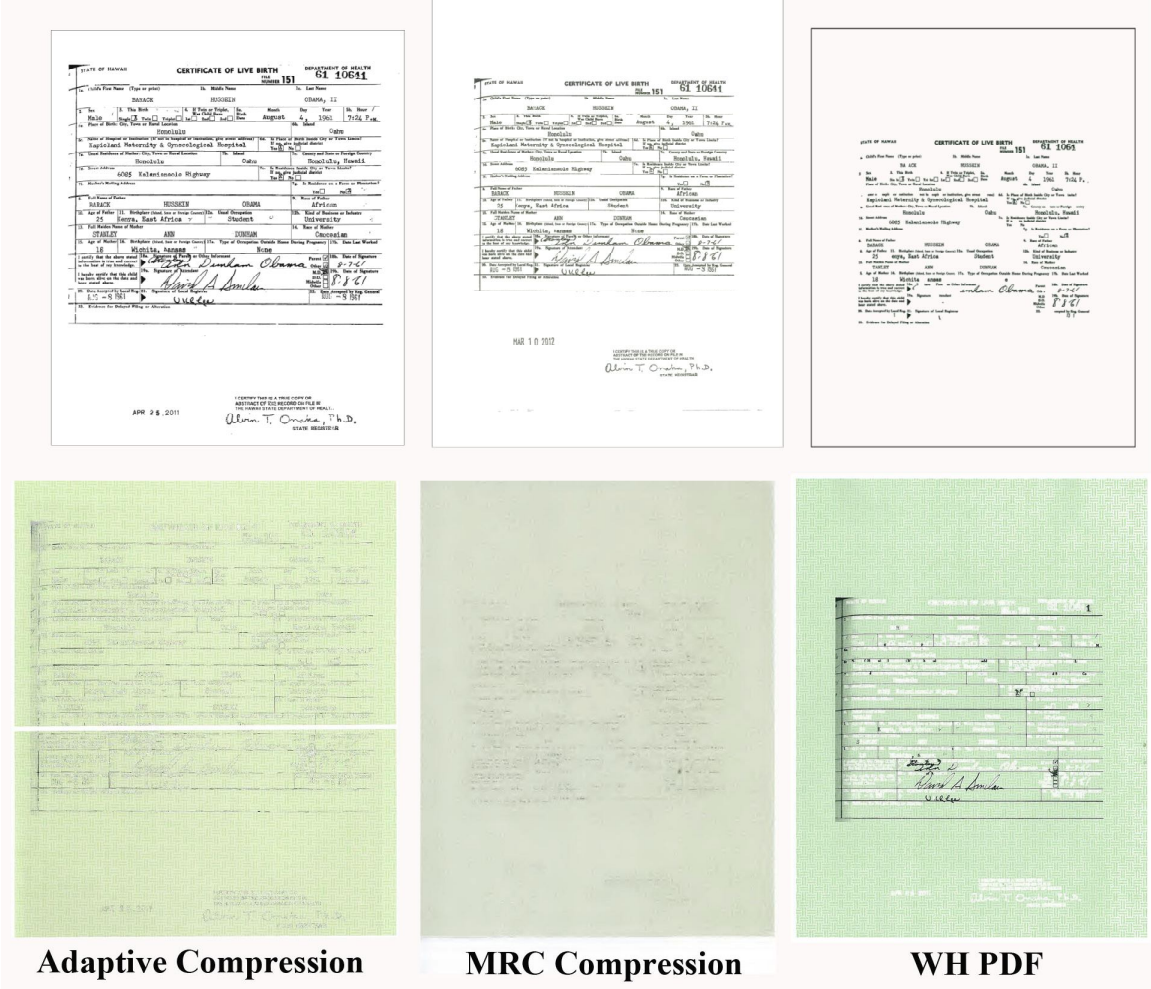


Figure 38 – Comparison of form lines

Notice that the text layers for both Adaptive and MRC Compression contain the form lines almost in their entirety. By contrast, the layer with most of the form text on the WH PDF contains no form lines at all. In addition, that layer is missing the state registrar's date stamp, the state registrar's signature stamp, most of the doctor's signature, most of the local registrar's signature, portions of the registrar general's date stamp and most of the local registrar's date stamp.

Furthermore, the background layers for both Adaptive and MRC Compression normally only contain remnants of the form lines, if any form lines at all. However, the form lines are completely on the background layer of the file released by the White House. In addition, the background layer contains most of the doctor and local registrars' signatures.

3. Text Color Properties

- Text on a color scan will vary in color from pixel to pixel, even within the same letter. This is due to the reflection of the light as it passes over the document. Furthermore a document with black text on a color background will result in a scanned image with text that is not completely black. Some of the color will be reflected from the background and result in text that has red, green and blue (RGB) color values that vary slightly from place to place within a letter.

- Adaptive Compression will convert any text that is segmented onto the 1-bit text layer to grayscale black, removing all RGB color values. The text will be the same uniform black throughout without any variation. Each pixel in the 1-bit text layer will have the same exact grayscale black color value.

- MRC Compression will retain the original varied color through the use of a color layer. The text will be the same color as it was in the scan and will still vary from place to place. In other words, each pixel can potentially have different RGB color values, even within a single letter.

- The text on the WH PDF does not vary from pixel to pixel within any individual 1-bit text layer, as would be expected with MRC Compression. Instead, each text layer has a uniform color value throughout. However, the text is not grayscale black as would be expected from a document optimized with Adaptive Compression. Rather, each text layer has a unique RGB color value and is dark green in appearance.

- In addition, two of the date stamps have visibly different colors from digit to digit despite being uniform in color from pixel to pixel. This is because the dates are broken onto separate 1-bit text layers and each layer has a unique color value.

The following control file text samples will help clarify the points above.

Adaptive Compression Control

- This sample text was optimized with Adobe Acrobat's Adaptive Compression.

- Any variance in color has been removed along with any RGB color values.

- The color is uniform throughout the entire text layer and is 92% grayscale black. Also, the color is the same from pixel to pixel within the same letter and among all letters (Figure 39).

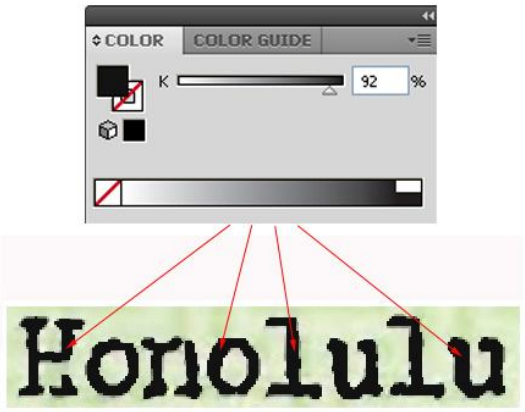


Figure 39 - Adaptive Compression

MRC Compression Control

- This sample text was optimized with MRC Compression.
- The color variance from the original scan is retained. The text has RGB color values and varies in color from pixel to pixel within each letter.
- Pixels in different locations of the text will have slightly different RGB values (Figure 40). This means that even a single letter will vary in color from place to place.

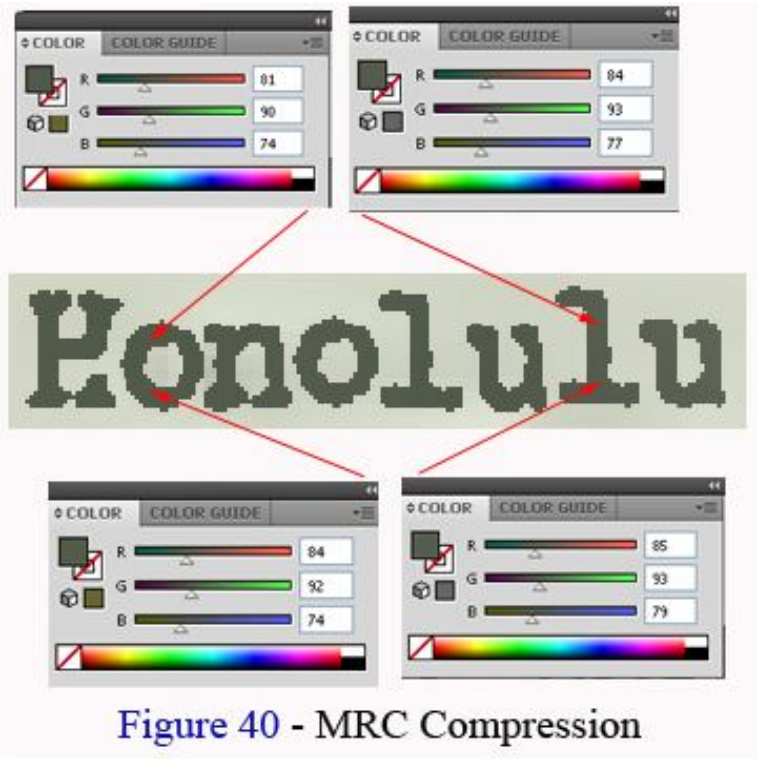


Figure 40 - MRC Compression

By contrast, the text on the WH PDF file is uniform from pixel to pixel but is not grayscale black. This means that the text has the same RGB color value throughout each text layer, regardless of location (Figure 41).

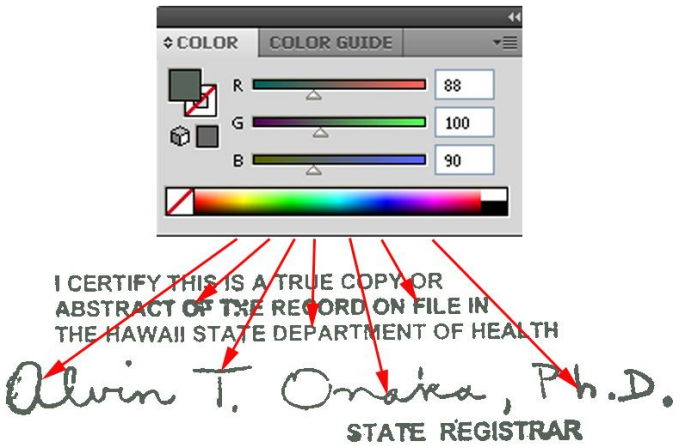


Figure 41- WH PDF

As with the previously analyzed items, the color properties of the text on the WH PDF file are not consistent with scanning or optimization with either Adaptive or MRC Compression. A color scan, of a document that has black text on a color background, will result in text that has varied RGB color values from place to place. An image optimized with Adobe's Adaptive Compression will have no color variation on the text layer and no RGB color values, but instead will be grayscale black throughout. Finally, an image optimized with MRC compression will retain the varied RGB color values that existed from the color scan. Unlike any of the above scenarios, the WH PDF has no color variation within a text layer yet the text is not grayscale black. Instead each text layer has a single, unique and uniform RGB color value.

Another problem is that two individual date stamps on the WH file have two visibly different colors within them. This is not the same as the color variation that you see from pixel to pixel in a normal scan. Each of these colors is uniform throughout the text layer that contains it, without any variation from place to place. Instead, the dates are actually on two separate layers, each of which has a unique RGB color. The result is two individual date stamps made up of two different solid colors (Figure 42). The letters that are underlined in red have a completely different RGB color value than the other characters in the same date. They are visibly darker even to the naked eye.

Figure 42 - Date stamps with 2 distinct colors

4. White Halo

- All text on the White House file is surrounded by a white halo (Figure 43).

Figure 43 – White halo

- There is also white space beneath the text layer on the background (Figure 44).

Figure 44 – White space

- The safety paper pattern is obscured by the white space on the background layer.
- Optimization does not randomly add a white halo or white space beneath the text.

Again, the following control samples will help illustrate the differences.

Adaptive Compression Control

- This sample is the manual ink stamp from a control document optimized with Adaptive Compression.
- There is no white halo surrounding the stamp (Figure 45).

Figure 45 – No white halo around text

- There is no white space beneath the stamp. Instead, a remnant of pixels is left behind forming the border of the text that was removed (Figure 46).

Figure 46 – No white space beneath text

MRC Compression Control

- This sample is the manual ink stamp from a control document optimized with MRC compression.
- Again, note there is no white halo surrounding the stamp (Figure 47).

Figure 47 – No white halo around text

- In addition, there is no white space left behind under the stamp. In fact, there is no visible residue beneath the text at all (Figure 48).

Figure 48 – No white space beneath text

The following side-by-side comparison image (Figure 49) clearly shows the difference between optimized files and the WH PDF in terms of the white halo and white space issue.

Figure 49 – Side by side halo comparison

The presence of a white halo around the text, and white space beneath the text, on the White House file is inconsistent with optimization or a simple scan. Under normal circumstances, a scanned document will not result in an image that has a white halo around the lettering. Similarly, optimization will not arbitrarily add a white halo or insert white space beneath the text layer. In order to generate such a halo, a user would have to take deliberate action, such as applying an unsharp mask, within a graphics editing program such as Photoshop. Manual editing with such an application would dramatically erode the provenance of this document.

V. Object Code Analysis

Each object within a PDF file is defined within the code. In this context, each layer, or image, would be considered an object. To better demonstrate the composition of the PDF file released by the White House, the image below displays the object code for each layer (Figure 50).

```
<< /Length 21 0 R /Type /XObject /Subtype /Image /Width 34 /Height 70
/ImageMask
true /BitsPerComponent 1 /Filter /FlateDecode >>

<< /Length 23 0 R /Type /XObject /Subtype /Image /Width 243 /Height 217
/ImageMask
true /BitsPerComponent 1 /Filter /FlateDecode >>

<< /Length 25 0 R /Type /XObject /Subtype /Image /Width 132 /Height 142
/ImageMask
true /BitsPerComponent 1 /Filter /FlateDecode >>

<< /Length 10 0 R /Type /XObject /Subtype /Image /Width 1454 /Height 1819
/ImageMask true /BitsPerComponent 1 /Filter /FlateDecode >>

<< /Length 15 0 R /Type /XObject /Subtype /Image /Width 42 /Height 274
/ImageMask
true /BitsPerComponent 1 /Filter /FlateDecode >>

<< /Length 8 0 R /Type /XObject /Subtype /Image /Width 1652 /Height 1276
/ColorSpace
11 0 R /BitsPerComponent 8 /Filter /DCTDecode >>

<< /Length 19 0 R /Type /XObject /Subtype /Image /Width 47 /Height 216
/ImageMask
true /BitsPerComponent 1 /Filter /FlateDecode >>

<< /Length 13 0 R /Type /XObject /Subtype /Image /Width 199 /Height 778
/ImageMask
true /BitsPerComponent 1 /Filter /FlateDecode >>

<< /Length 17 0 R /Type /XObject /Subtype /Image /Width 123 /Height 228
/ImageMask
true /BitsPerComponent 1 /Filter /FlateDecode >>
```

Figure 50 – Object code from the WH PDF file

The code defines the object type, the image dimensions, BitsPerComponent and compression method used. BitsPerComponent refers to the amount of data within the color component. 1 bit per component refers to the transparent text layer that has only 1-bit color depth. 8 bits per component defines the background layer, which has 8-bit color depth. See the optimization section of this analysis for further clarification of the difference between 1-bit layers and 8-bit layers. Notice that the White House file has many 1-bit layers and only one 8-bit layer.

By contrast, look at the code from an optimized PDF shown below (Figure 51). This code was taken from a control document that was optimized by Adobe Acrobat using Adaptive Compression.

```
<< /Length 58 0 R /Type /XObject /Subtype /Image /Width 1920 /Height
2192
/ImageMask true /BitsPerComponent 1 /Filter /FlateDecode >>

<< /Length 16 0 R /Type /XObject /Subtype /Image /Width 168 /Height 20
/ColorSpace
59 0 R /BitsPerComponent 8 /Filter /DCTDecode >>

<< /Length 48 0 R /Type /XObject /Subtype /Image /Width 56 /Height 24
/ColorSpace
59 0 R /BitsPerComponent 8 /Filter /DCTDecode >>

<< /Length 34 0 R /Type /XObject /Subtype /Image /Width 336 /Height 408
/ColorSpace
59 0 R /BitsPerComponent 8 /Filter /DCTDecode >>

<< /Length 26 0 R /Type /XObject /Subtype /Image /Width 8 /Height 16
/ColorSpace
59 0 R /BitsPerComponent 8 /Filter /DCTDecode >>

<< /Length 24 0 R /Type /XObject /Subtype /Image /Width 464 /Height 20
/ColorSpace
59 0 R /BitsPerComponent 8 /Filter /DCTDecode >>

<< /Length 8 0 R /Type /XObject /Subtype /Image /Width 636 /Height 212
/ColorSpace
59 0 R /BitsPerComponent 8 /Filter /DCTDecode >>

<< /Length 14 0 R /Type /XObject /Subtype /Image /Width 192 /Height 24
/ColorSpace
59 0 R /BitsPerComponent 8 /Filter /DCTDecode >>
```

Figure 51 – Object code from an optimized PDF file

The important fact to note is that there is only one layer object with BitsPerComponent set to 1. This is the single 1-bit layer that contains all of the text that was segmented from the background. As previously discussed, optimization routines attempt to pull all text and graphic elements onto a separate layer from the background so that they can be compressed using more efficient algorithms. As shown by the above object code, optimization only creates a single 1-bit transparent text layer containing all of the text and graphic elements that were detected by the software. This again reinforces the difference between an optimized PDF and the one released by the White House.

VI. Metadata Analysis

Metadata is commonly defined as data about data. Metadata describes how and when and by whom a particular set of data was collected, and how the data is formatted. In the case of PDF documents, the metadata usually contains any or all of the following information: Author, Producer, Creator, Title, Creation Date and Modified Date. Below are the relevant sections of metadata from the PDF released by the White House (Figure 52):

```
%PDF-1.3
%Derek's PDF
4 0 obj
<< /Length 5 0 R /Filter /FlateDecode >>

<< /Type /Pages /MediaBox [0 0 612 792] /Count 1 /Kids [ 2 0 R ] >>
endobj
31 0 obj
<< /Type /Catalog /Pages 3 0 R >>
endobj
32 0 obj
()
endobj
33 0 obj
(Mac OS X 10.6.7 Quartz PDFContext)
endobj
34 0 obj
()
endobj
35 0 obj
()
endobj
36 0 obj
(Preview)
endobj
37 0 obj
(D:20110427120924Z00'00')
endobj
38 0 obj
()
endobj
39 0 obj
[ () ]
endobj
1 0 obj
<< /Title 32 0 R /Author 34 0 R /Subject 35 0 R /Producer 33 0 R /Creator
36 0 R /CreationDate 37 0 R /ModDate 37 0 R /Keywords 38 0 R
/AAPL:Keywords
39 0 R >>
```

Figure 52 – Metadata from the WH PDF file

The metadata shows the machine that generated the final PDF was a Macintosh running OS X 10.6.7 and that the PDF itself was generated by Quartz PDFContext. Quartz PDFContext is a PDF generation tool that is built into the Mac OS X line of operating systems. It can be trigerred from just about any program by using the 'Print – Save as PDF' option. The final red line above shows that the application used to do this was Mac Preview. Preview is the default image viewer in Mac OS X and is similar to MS Paint or Windows Picture and Fax Viewer within Windows.

Through the use of metadata, many file types contain information about how they were created. The PDF format goes one step further and is capable of tracking the history of a particular file. PDF files are actually able to retain previous changes within a newer version of the file. In essence, this creates a running history of the changes made to the document over time.

However, Mac Preview actually removes all traces of existing metadata as well as any change history that may have existed. It does this because it actually rebuilds the PDF document from scratch, using the code for the individual objects that the file contains.

Preview Lacks Layering Capability

Mac Preview is capable of compressing PDF files but doesn't have more advanced optimization logic like Adobe Acrobat. It includes the ability to apply Quartz filters to a file when saving it as a PDF. According to the Apple support website for Mac OS X, you can use Quartz filters to modify a file by adding effects, changing the color space, and even reducing the size of the file by recompressing graphics. The bottom line is that Preview does not utilize segmentation during compression and therefore is incapable of creating layers within a PDF file.

However, Preview does have the ability to retain any existing layers that were in a PDF file previously. In addition, Preview is capable of opening native Adobe Illustrator and Adobe Photoshop files that contain layers, and saving them as PDF files.

Digital Chain of Custody

Since Preview is unable to create layers yet layers exist within the PDF released by the White House, it can be stated definitively that some other unknown application created these layers before the file was saved within Preview. Because of that fact, the 'digital chain of custody' of this document is unknown and the integrity of the data is unknowable from a technical perspective. Any number of processes and applications could have been used on this document between the time it was scanned, assuming it was scanned, and the time it was converted to the final PDF file.

The file could have been manipulated, or even built from scratch, within Photoshop, Illustrator or any graphics editing software application. The other possibility is that the file was optimized by an application such as Adobe Acrobat. Optimization routines can create layers, but it has been demonstrated, in the optimization section of this document, that the characteristics of such layers don't match what is seen on the WH PDF. Let's put that aside for a moment and examine the PDF optimization argument from a purely logical standpoint.

Layers of Logic

First it is important to understand that the output of PDF optimization is a PDF file. This seems obvious enough to not need mentioning, however it is very significant from a logical standpoint. Let's assume that the birth certificate posted by the White House was scanned and then run through PDF optimization in Adobe Acrobat, resulting in a PDF file with layers. Again, let's temporarily forget that the layers in the WH PDF file are dissimilar from optimized layers from a technical perspective. The output of Acrobat's optimization routine would already be a PDF file. Take a second to digest that last statement. What the user would have at that point is a PDF that has already been optimized and is ready to post for download. If the layers were generated by Adobe Acrobat's PDF optimization routine, there would be no logical need to open the PDF in Preview to generate a second-generation "refried" PDF file. Yet the metadata clearly testifies to the fact that the final PDF was generated by Preview.

Suppose these layers were created through manual manipulation within Adobe Illustrator or Photoshop, would there be a logical reason to reopen the file in Preview and "refry" the PDF? There seem to be at least two potentially logical reasons.

First, the user might have assumed that Preview would remove any layers and flatten the document when converting it to PDF. Again, Preview can open native Illustrator (.AI) and Photoshop (.PSD) files so it is possible the user didn't flatten it within the editing app, but instead just saved it as an .AI or .PSD file assuming that it would be flattened by Preview during the PDF conversion process.

Second, there would be a need to cover the digital tracks if this document was tampered with in an Adobe product. Simply exporting it, as a PDF from Photoshop or Illustrator, would result in metadata that would show the application used to create it. If Photoshop or Illustrator metadata existed within this file it would be strong evidence that the document was altered or digitally compiled. Converting the final product to PDF within Mac Preview would be a perfect way of 'erasing' the 'digital chain of custody' that exists within the file's metadata.

VII. Conclusion

In conclusion, it is not disputed that the PDF file released to the public on April 27, 2011 contains multiple layers. The only way that layers could exist within a PDF file is by optimization or manual manipulation. The third option, Optical Character Recognition (OCR), has been previously ruled out because the text within the PDF is not searchable. Comprehensive analysis shows that the characteristics of the layers found in the White House file do not match the attributes of layers created by optimization.

1. The layers are more logical than would be expected from a computerized optimization routine. In particular the date stamp and registrar stamp, the very items that give validity to the document, are on their own isolated layers separate from the background.

2. Optimization only creates a single 1-bit transparent text layer consisting of the text and graphic elements of the image. By contrast the file posted on the White House website has several 1-bit transparent text layers. These extra text layers include the state registrar's date stamp, state registrar's signature stamp, portions of the registrar general's date stamp and most of the local registrar's date stamp.

3. The lines of the form itself are on the background image layer instead of being included on one of the transparent text layers. Optimization attempts to segment the image and put any text and graphic elements on a separate layer from the background and images. This is so different compression algorithms can be applied to each layer to achieve the best results in terms of smaller file size. Not only are there too many text layers on the White House PDF file, but they also do not contain the correct portions of the image, namely the form lines.

4. The color properties of the text layers are not what would be expected from optimization. Adaptive Compression used by Adobe Acrobat Pro, converts any color text to grayscale black when it segments the image. With this method all text is the same grayscale color value. On the other hand MRC compression, which is utilized by many other applications, actually retains the full color variance of the text. With this method the color can vary from pixel to pixel and contains RGB color values. By contrast, the PDF released by the White House has only one single, non-grayscale, color value for each text layer. This differs from adaptive compression in that the text is not grayscale black, and differs from MRC compression in that the text is only one color with no variance. In addition, two of the stamps on the document contain two different colors despite having supposedly been created from one stamp with one ink color.

5. The existence of a white halo around all of the text on the White House PDF file cannot be explained by optimization. Neither adaptive compression nor MRC compression yields these results.

6. The object code within the file released by the White House does not match the object code from an optimized PDF file. An optimized file contains only one layer that has 1 bit per component. That is the single transparent text layer that contains the text and graphic elements of the image. In comparison, the White House PDF has multiple layers with 1 bit per component. The object code reinforces the second point above, that this file has too many transparent text layers.

In addition Mac Preview, which is the application that generated the White House PDF file, erases any previous metadata within a file. Because of this, the digital chain of custody of this document is unknown and any number of processes and applications could have been used on the file before it was finally resaved in Preview.

Furthermore, extensive testing and research determined that Mac Preview is not capable of creating layers within a PDF file itself. This means that the layers were created by some other application before the file was resaved using Preview. This again underscores the lack of a digital chain of custody of this document and raises a logical conundrum for those claiming the layers are due to optimization. The output of an optimization routine would be a PDF file that was reduced in file size and ready to be posted online for download. There would be no logical need to resave the PDF within Preview. However, if the document was altered, or totally fabricated, by some other application there would be a potential reason. Saving the file again in Mac Preview would erase any metadata that might show information about the application that altered or created the document.

For the above-mentioned reasons, it is my opinion that scanning and optimization alone cannot explain the anomalies on the PDF released by the White House. The evidence suggests that this file has been tampered with in some other manner.