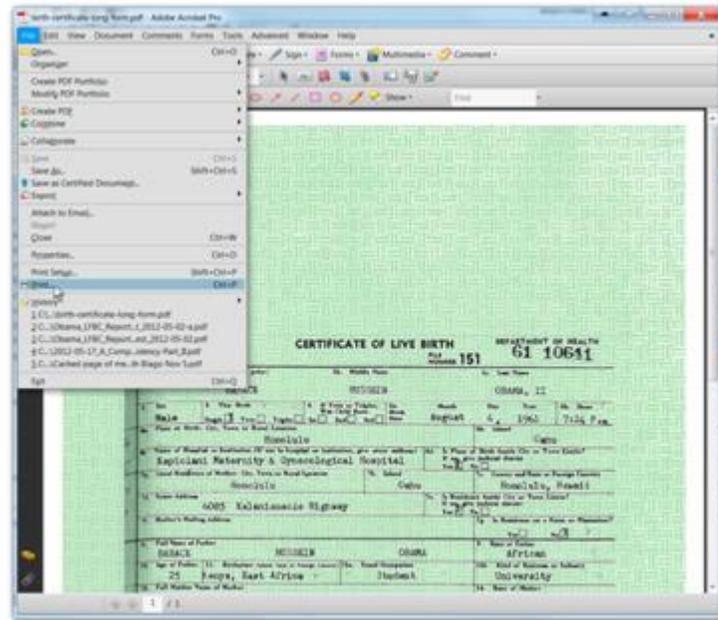


New Obama Birth Certificate Forgery Proof in the Layers

By Mara Zebest | July 18, 2012

My newly released report examines the metadata and object code of Obama's long-form birth certificate [PDF file](#) and explains how this information corroborates the claim that Obama's PDF file never originated as a paper document, but rather was born in cyberspace or was -- to put it another way -- *digitally manufactured*. The only time Obama's long-form birth certificate image exists as a paper document is when a computer user selects *Print* from the *File* menu. Obama's PDF file, like everything else from this administration, is a composite, a lie, an illusion -- even down to the deceptive colors presented in the file.

The focus of this report is centered on Obama's PDF file released on April 27, 2011; the file *[has been removed]* from [White House link](#). From the moment Obama's PDF file was posted, numerous professionals have been speaking out on the multitude of anomalies. The most notable indicator of malfeasance within Obama's PDF file is the presence of multiple *layers*. A document scanned into a computer should only contain *one* layer -- known as a flattened



file. However, Obama's PDF file not only contained more than one layer; it contained nine layers that *clearly* display recognizable signs of forgery, along with a host of other manipulative indicators. The problems range from [Tom Harrison's](#) observation of undisturbed safety-paper pattern beneath white dots to a clipping mask path that hides data and more.



Because *layers* are the most obvious symptom of foul play, it may be helpful to understand what layers are and what advantages layers provide. Layers are analogous to transparencies used on a projector in what will be referred to as *the analog world*. Content is printed on separate sheets (or layers), which allows the user to move, add, or remove a sheet element independently of the

remaining sheets. The total composite document comprises the combined stack of transparency sheets.

The digital equivalent allows a user to add or move digital image information (or text) onto separate layers. This, of course, would be a manual and manipulative choice. Like their analog parallel, layers can provide a means to maneuver individual elements within the image independently of surrounding image pieces.

The layers have been the most damning and problematic evidence of file-manipulation, and the defenders of Obama are quick to respond with a plethora of explanations to justify the presence of layers. The excuses range from OCR (Optical Character Recognition) software to the more predominant excuse of optimization -- both of which have been debunked in my previous report for the [Cold Case Posse press conference](#).

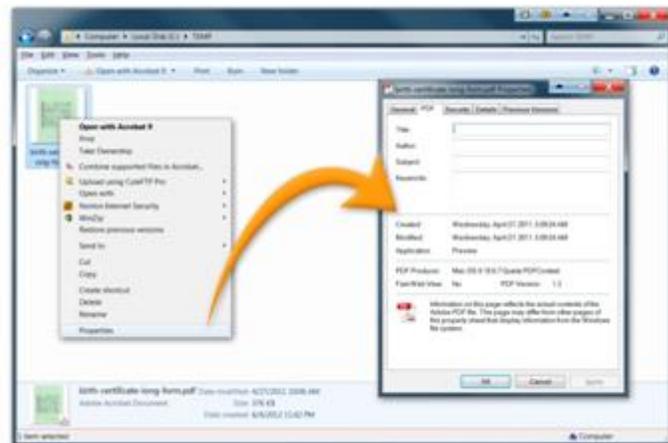
Many Obama defenders have conceded that OCR is *not* a factor and admit that OCR was never applied to the PDF file. However, arguments for optimization still persist. Optimization refers to a *file-saving* process in which the goal is to reduce the file size while maintaining (or optimizing) the quality of the image (as best as possible depending on the settings applied).

Those who insist on the optimization argument either do not understand what attributes need to be present for this argument to hold water -- or they are hoping the general public does not understand. *It's probably a little of both*. The defenders certainly count on the ignorance of the average citizen when it comes to understanding the differences in layers produced from an automated process (such as optimization) compared to a manual choice to manipulate the file. One goal of the report is to offer a deeper understanding for recognizing the two patterns of layering (and to avoid being deceived or bamboozled). The report adds additional *proof* along the way that the *optimization* excuse fails *miserably* and can be completely ruled out as a justification for layers.

One particular and significant point to understand: *not all optimization processes are created equal*, and

only *certain programs* will create (automated) layers when optimizing. Adobe Acrobat Pro is the most commonly used program capable of creating layers from an automated process such as optimization, but even Acrobat Pro does not *always* produce layers when optimizing a file. The method and settings applied during the saving process play a factor in

whether layers will be automatically generated or not. To be clearer on this point, programs like Photoshop and Illustrator will *not* automatically generate layers when optimizing a file (regardless of the circumstances). Any layered files *generated* from these programs are reflective merely of layers *manufactured* by the user.

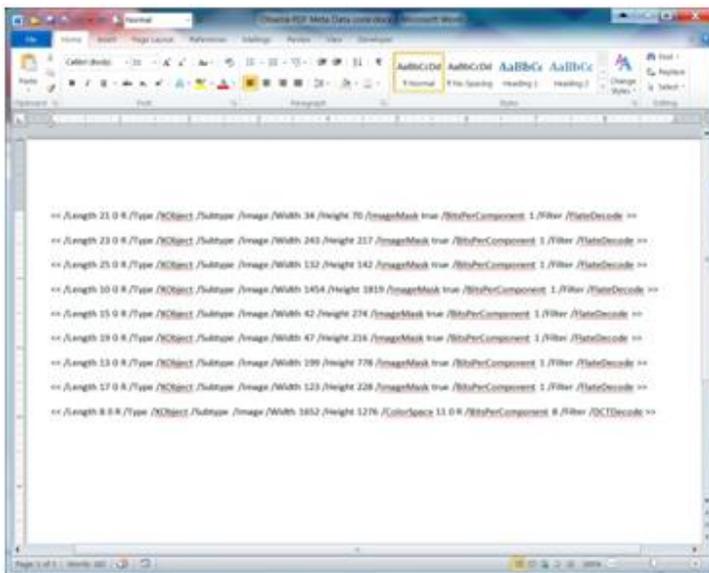


The problems for the opposition increases when the metadata is considered. So what is metadata? If you've ever examined a file and noticed property information such as the creation date, modified date, or the program used to create the file, this is known as *metadata*. The metadata for Obama's PDF reveals that a Mac based program called Preview was used to generate the optimized PDF file.

Preview is a program that allows a user to open and *view* a file (as the name suggests). Preview can view a range of file types created from other programs as well as allow a user to re-save the file in any file format available within Preview. Preview is limited in its abilities and certainly not a program used to create a layered file from scratch. The optimization argument becomes increasingly problematic when it is understood that Preview is incapable of producing or generating layers *automatically* from an optimization process (when saving a file). Preview will *only maintain* pre-existing layers produced from another program. Thus, Obama's PDF layers *had* to be generated from another program prior to being saved from Preview.

One more important point regarding Preview's role in creating Obama's PDF file: if a file is produced in another program, opened, and saved again within Preview, the metadata from any prior programs used is replaced with Preview metadata, thus eliminating any digital tracks for the previous programs utilized in the process (such as Photoshop or Illustrator, which might imply manipulation).

Along with metadata, *object code* is also buried within the PDF file -- which refers to the code that describes the attributes or properties present for each file *object* (and can be seen by opening the PDF file in WordPad or Word). *Layers* within the Obama PDF file would be examples of *objects* described in the *object code*. The [report](#) examines the layer object code, along with how the code properties can be applied in analyzing *automated* layers -- produced by an *Acrobat optimized* document -- against the *manufactured* layers -- produced in *Obama's PDF* file. Visible and consistent patterns in optimized layers do not exist in Obama's PDF file.



One of the object code properties examined will be **BitsPerComponent** (seen in the figure), which is typically followed by a numeric value of 1 or 8. To simplify, this refers to either a 1-bit layer or an 8-bit layer, respectively. In layman's language, a 1-bit layer that also contains an *ImageMask true* property (in the object code) equates to a layer that is capable of containing only one single color -- *black*. Conversely, an 8-bit (*ColorSpace*) layer is capable of

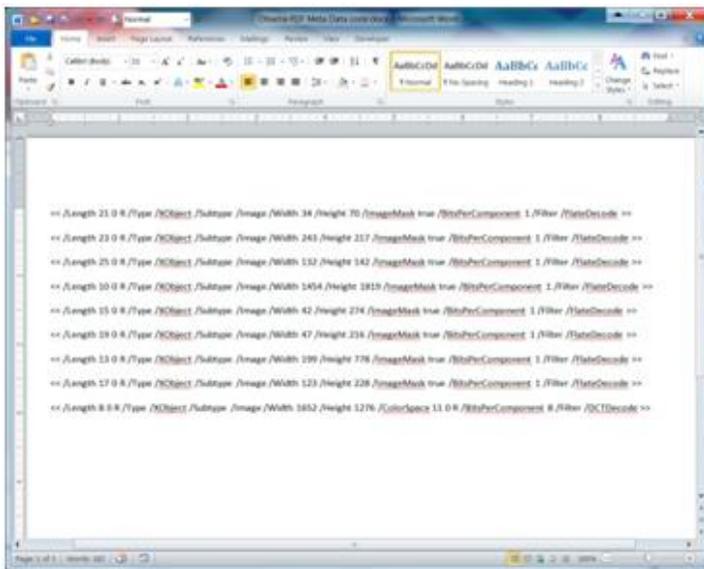
containing more than one color -- indeed, of containing any color value (not restricted to just black). If this concept is clear, then the rest is easy, and the [report](#) will explain these concepts further.

The bottom line, documented in detail within the report, is that an optimized layer pattern will contain only a single layer of 1-bit quality that displays black only. All remaining layers (in an optimized file) are 8-bit -- containing multiple colors. In other words, the optimization process pulls out all the black color onto one (1-bit) layer -- not two (1-bit) layers, not three (1-bit) layers...just one layer, with this 1-bit object code attribute.

Obama's PDF file (as seen in the figure) contains EIGHT 1-bit layers -- all of which fail to reflect a black color. However, the colors presented are a deception. The black is there, but it's hidden in plain sight (as outlined in the report).

Taking into account the multiple 1-bit layers embedded in Obama's file, combined with the associated color-manipulation defined in the report, it is *impossible* to duplicate these file attributes through scanning or any other argument thrown at the debate, which currently includes:

- Optimization
- Adaptive compression
- MRC (Mixed Raster Content) compression
- JBIG2
- Adobe Acrobat Pro, and...
- OCR



Here's a challenge to anyone who would like to defend the list of excuses or to supplement any additional excuses. It is advised (before doing so) to carefully read the report, examining the provided figure of object code for the nine layers found in Obama's PDF (with important properties highlighted). Then run tests to see if any of the described *suppositions* proposed in the above bulleted list (or any new excuses being considered) can produce object code exactly

as seen in the capture (without manipulation). The final result must include eight *1-bit Image Mask true* layers (with **FlateDecode**, as seen in the figure), and one *8-bit ColorSpace* layer (with **DCTDecode**, as seen for this last layer). Oh, and as an added bonus, the 1-bit layers (all eight of them) need to have colors displayed that are *not* within a pure (grayscale color value) range of black.

When these displayed attributes from Obama's PDF anomalies are reproduced via scanning and optimization settings, please publish the winning recipe settings for all to verify. And don't forget to include a reflow from Preview as the last step. If you cannot document the winning recipe, then *supposition* does not count, and

therefore you have nothing -- which falls into the category of misinformation, misunderstanding, misrepresenting, and misdirecting.

Obama's PDF object code properties cannot be reproduced without *manufacturing* and *manipulating layers*. Any layers produced through optimizing will result in a single 1-bit layer. Repeat: *just one* 1-bit layer -- not *eight*. In the case of adaptive compression, the color on the 1-bit layer will reflect a black color value -- not off-white, not greenish-black, not greenish-gray (as displayed in Obama's PDF file).

The report will detail these concepts and provide a documented recipe on how the layers in Obama's PDF file *can* be replicated down to the metadata, object code, and low file size. Here's a hint: the only way it was possible was through complete image manipulation and manufacturing of layers. The winning recipe included Photoshop, Illustrator, and Preview (in that order).

Mara Zebest is a graphic artist and co-author for a number of Adobe product books including the Inside Photoshop series, which typically exceeded 1,000 pages and was published in at least ten different languages around the world. Ms. Zebest is also tech editor for numerous books for both Adobe and Microsoft products and has worked closely with the Cold Case Posse (CCP) for the Maricopa County Sheriff's Office (MCSO) in providing evidence on Obama's forged birth certificate.

[\[https://www.americanthinker.com/articles/2012/07/new_obama_birth_certificate_forgery_proof_in_the_layers.html\]](https://www.americanthinker.com/articles/2012/07/new_obama_birth_certificate_forgery_proof_in_the_layers.html)