

*Protocol for Evocative Object
Emedia Publication*

Image Optimization

In order to decrease load time of images used for Web based PDF's, it is imperative to use Photoshop's 'Save for web and devices' functionality.

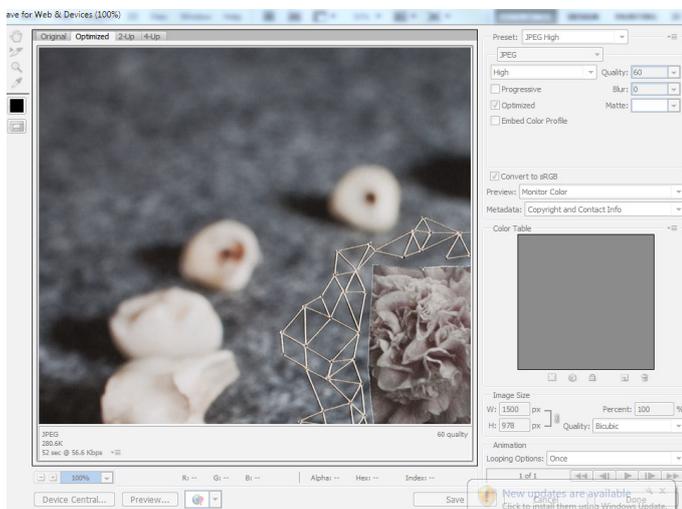
This tool can be found under, File>Save for Web and Devices.

The three file types that are commonly used are:

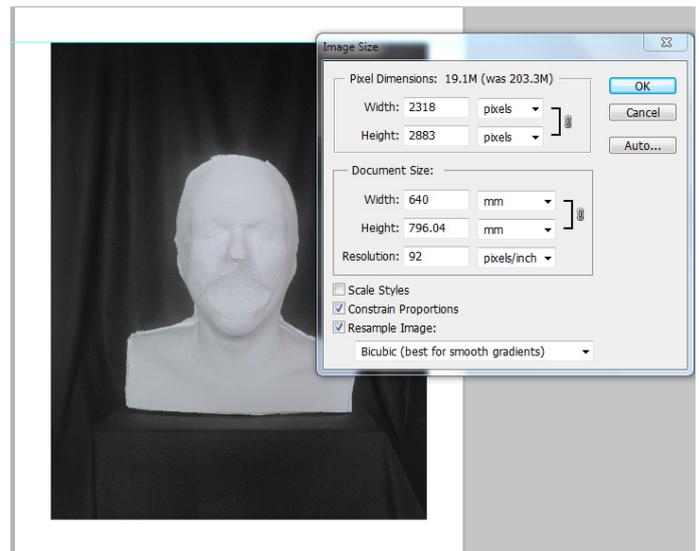
- .jpg – Lossy Format, great for photographs
- .png (8-bit) – Lossless Compression type, limited color range
- .gif – Lossless with limited color range like .png 8, but .png is usually the better choice.

jpg compression:

The .jpg image format is best suited to images that have a wide range of colours where there aren't sharp transitions or fine edges. Because of its compression algorithm, it works very well for Photographs. It is possible to significantly reduce the file size of a Photograph while keeping the image quality very high. With .jpg compression you have the ability to pretty freely adjust the quality in exchange for file size. With photographs, you'll probably find 60-70% quality to work just fine. If there is text in the image, you may notice more artefacts given the sharp edges, and may have to bump up the quality.



By keeping the quality set to 'High' the image retains quality but is taken from 4.19 mb down to 280 kb's which gives a load time of 52 secs. This is a compromise between keeping a certain level of depth to the original image while making the load time under a minute which streamlines the user experience. Another way to optimize the images was to take the dimensions down to around 1000 pixels wide, this gave uniformity to the size of the images and greatly reduced the file size. These images are saved as copies in a different folder so as to not destroy the original image in case of the need to use these images again in the future.



Since screen resolution is no higher than 92 pixels per inch, this is another way to reduce the file size before saving for web and device. If the resolution is dropped from 300 (print) to 92 (web) then the actual size of the image (scale) will drop as well. There are ways to undo this but it works well for what we are trying to achieve with our images in this instance.

.gif and .png compression:

The .gif and 8-bit .png image formats are better suited for images with limited color ranges (they can display up to 256 colors), where there are large areas of uniform color. In these instances, .png and .gif will usually produce a close-to-perfect representation of the original image while saving at a higher rate than .jpg. Typical uses of these formats include interface items, icons, and images where text needs to be represented accurately. More often than not, the 8-bit .png format will compress better than .gif. Both formats support transparent pixels (Not to be confused with alpha transparency which is supported by 24-bit .png).

By compressing images across any web based presentation, load time becomes quicker and server storage is reduced. This allows for better user experience and keeps page sizes consistent with one another.

Interactive Galleries:

The images needed to have an animated quality in order to make each individual gallery work in conjunction with sound. These were made in Adobe Flash with a simple animation of each image for 220 frames with a fade of

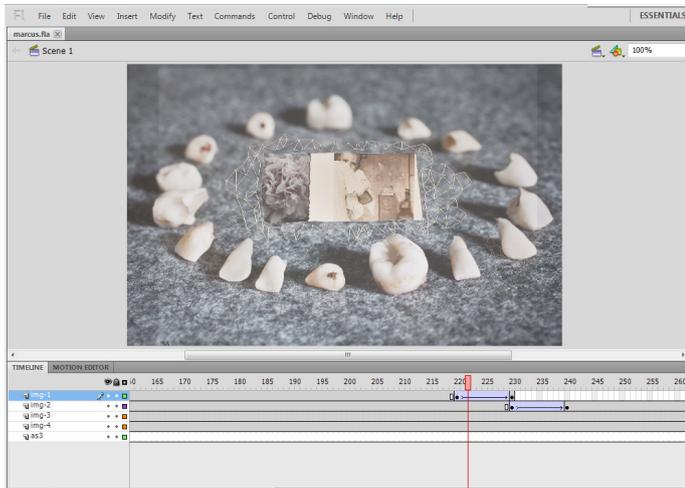
10 frames applied at the end of each as a transition into the next image. This then had a basic "got to and play 1" action script applied to it that essentially loops the animation allowing for the images to keep a transnational quality throughout the duration of the sound files.

Indesign Workflow

By creating the image slide show and converting the audio to .flv, the workflow in Indesign becomes one of placement, hyper linking and file organization. Firstly is the consideration of layout and navigation, there were a few iterations of this as the need for familiarity for the user coupled with functionality became concerns that needed to be weighed against the content.

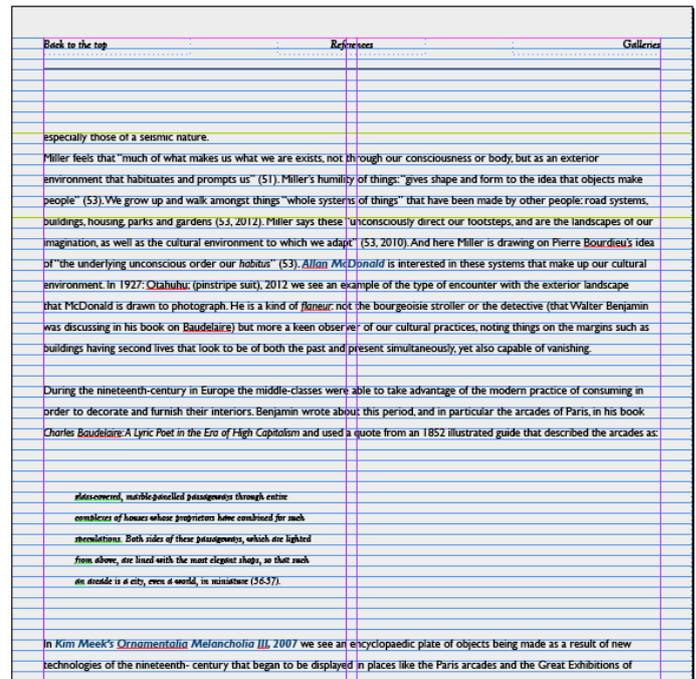
Layout

As the content was an essay, it seemed imperative to give the layout a book page composition.



The animated content is shown here as the first frame transitions into the second. Visible in the Flash timeline are the controls for the fade transition and the start position for the second frame of imagery. Visible to the left of the timeline are the layers that show the images and the bottom layer which is where the action script is written.

What pre-composing the images allows for is less work within the Indesign document in terms of functionality. By using flash we are able to make compatible files (.swf) that can be nested within the document and allow for full compatibility within the PDF file structure.



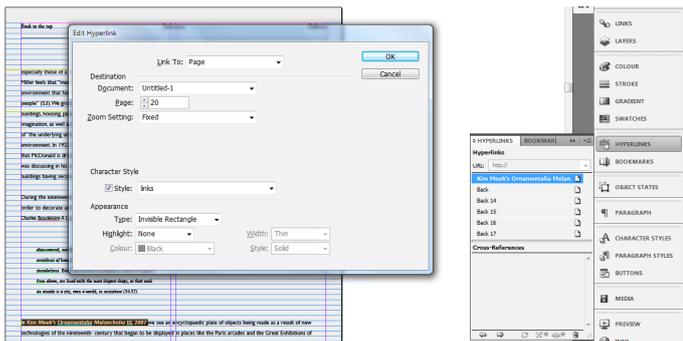
Text was divided into the body text and the hyperlinks (which are in blue) this gave a distinct visual cue to hover over certain text in order to access the interactive media embedded in the document.

Sound Optimization

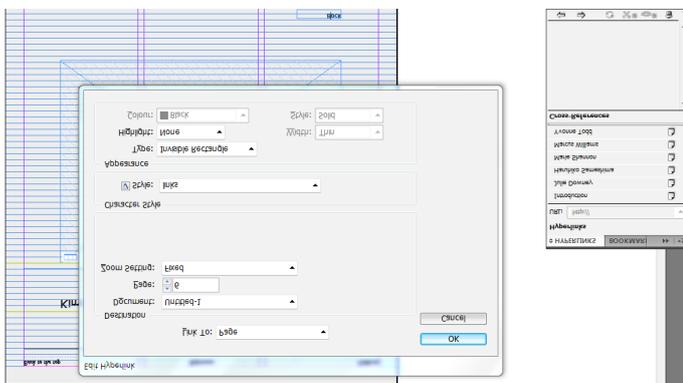
The sound files that were originally provided were much too large to be embedded directly into the Indesign document. In order to maximize functionality, usability and download time, there needed to be a certain amount of compression applied to the audio in much the same way that the images were saved in a web friendly format. There is 2 ways to go about this, one is to have both After Effects and Flash installed on your computer and export the audio from After Effects as .flv (.flv being one of the most compatible formats for video and audio in an interactive PDF) the other is do do the same process with Adobe Media Encoder, upload audio - change output module to .flv and then render.

Hyper-links

The text acts as a trigger, linking to pages further in the document that hold the interactive media. There was the choice to either keep everything in one document or to split the media from the essay and make them separate documents that would feed off of each other. In the end it became the logical option to house everything in one document as this helps with stability of the links.



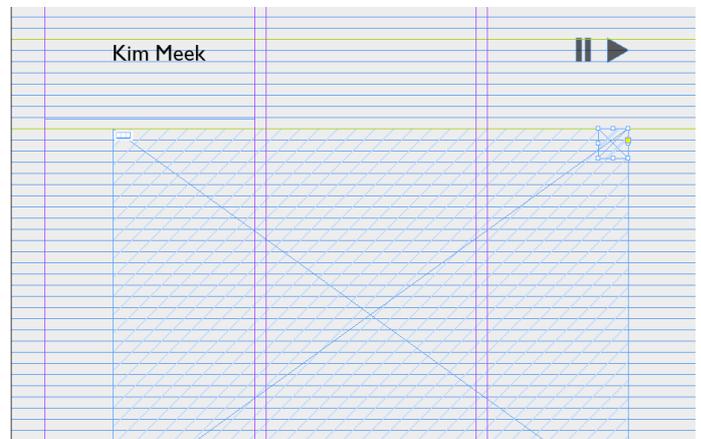
This image displays how the navigation occurs. The highlighted text is converted to a hyperlink which is connected to the page containing the media for that particular link. By hovering the mouse cursor over the text a hand appears indicating that this is an active link.



Additional functionality of being able to navigate back to the point of the document where the user was before activating the link was also achieved through the use of hyper links. Each 'back' button was linked to the specific point of the page where the initial hyperlink existed, allowing for effortless navigation throughout the document.

Media Pages

Placing the media was down to the initial size of the .swf slideshows made in Flash. This meant that the image resolution would be retained, whereas scaling the .swf could possibly cause deterioration or image distortion. The audio elements were placed in the corner of the slideshow as an invisible object. Controls (button states) were added as a feature that would allow the user to access the audio interactively rather than passively. This feature was added as there were initially problems having the page turn causing an overlap of sound within the document, i.e. if the user accidentally turned the page the sound of the previous page would overlap the sound of the new page.



The audio sits as an invisible object on the same page as the visual media

Controls were added with the simple functionality of pausing and playing the audio. This did not impact overly on the visual media as they were created to be seamless loops of images, giving the user the ability to play the audio when ready. Simple identifiable icons that relate to play and pause symbols made identification of the use of these buttons easily read.

Considerations for use

This document may not be all device friendly. I have not been able to test on either Android or iPad/iPhone devices. Initially to get around known device issues such as a lack of flash players on Apple products, there could be video files that mimic the slideshow .swf files and contain the audio within them. This however would have made a much larger final document and may have caused deterioration of image quality. It was therefore decided that PDF existing protocols would be followed even at the expense of cross device usability. There are paid apps for Apple that allow interactive PDF's to be read with differing degrees of success.

In order to use this document computers need to have both flash player and Adobe acrobat, in addition to this both of these programs to be up to date in order for the functionality to work properly.

Issues with creation

I have included a list of issues that were encountered in the creation of this document for the purpose of awareness and as a means to explain why the final output looks, behaves and has had certain formats used in it's creation.

Issues

sound overlap due to lack of a stop when page is turned option

Solved by forcing user to start sound

Must be played with an up to date pdf viewer.

Workflow Programs

layout - indesign, photoshop

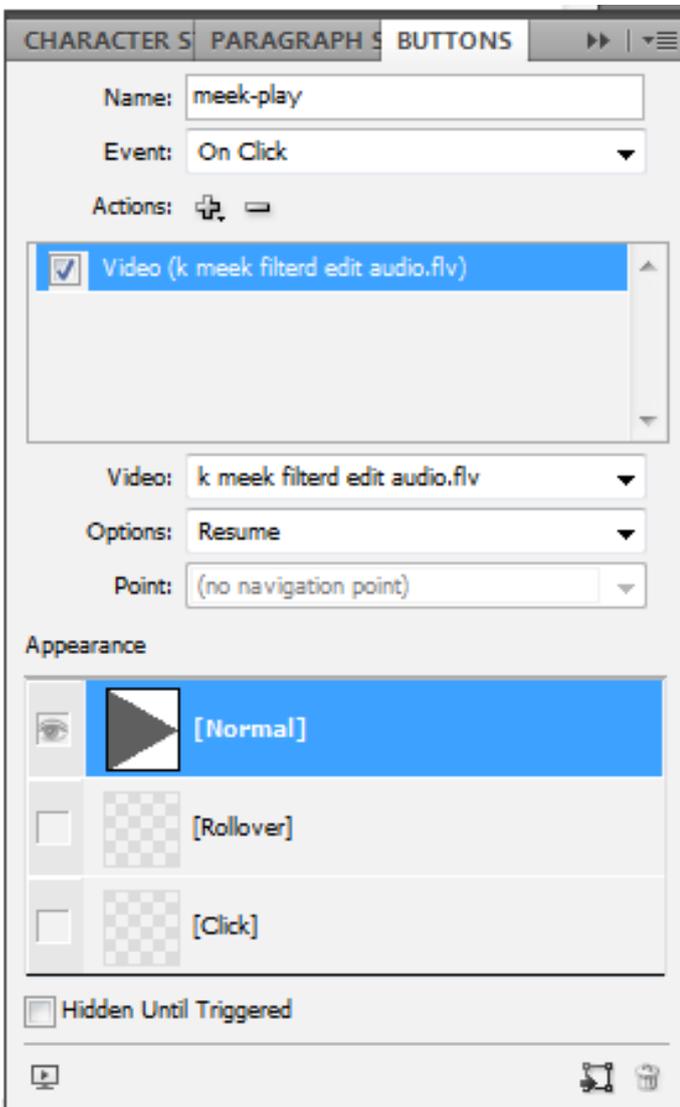
interactivity (hyperlinks) - indesign

sound to flv- After effects

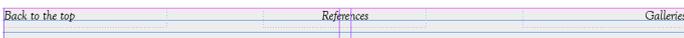
image slideshows - photoshop, flash (action script)

embedded images back into indesign

linked sound to buttons in indesign



This illustrates how the control buttons work, what they are linked to on an individual basis and what the commands are. It displays that the button is calling for the converted .flv video to be played once clicked on, allowing for the audio to begin as the image slideshow plays on the page.



An additional function was to add hyperlinks that took the user to either the title page of the document, the references list or the Gallery index which contains a list of all the interactive pages by the artists name, giving additional access to this content. The reasoning behind this index was to help organize the content into a comprehensive list that could be accessed in a more passive way.

Overview

As this was an initial excursion into creating and outputting an emedia publication, this protocol should be seen as an overview on a particular style of document, namely, PDF.

Outlined within this document is an overview of the finished articles creation and the tools used in order to maximise functionality while minimizing file size.

There are newer technologies that could improve upon this template, one option would be to make an html or web based document that could be hosted on a server allowing for high resolution videos to take place of the flash based files that were used in this document. The user could then stream the files straight from the server and device compatibility would not become an issue.

Another way would be to explore Indesigns newer epub output modules which would allow the document to be used on a range of devices including ereaders.

This document however is a fully realised, functional emedia publication that has been maximized to have as low a file size as possible to make it more attractive as a download. The functionality mimics existing practices in terms of both navigation and usability.

Document co-created by Martin Tomars & Justin Barter.