

# Phosphor



divergent media, inc

# Contents

<b>1</b>	<b>1 Welcome to Phosphor</b>	<b>4</b>
	Welcome . . . . .	4
<b>2</b>	<b>2 Getting Started</b>	<b>5</b>
	Interface Basics . . . . .	5
	Supported Media . . . . .	6
	Exporting . . . . .	6
	PNG Export . . . . .	6
	JPG Export . . . . .	7
	TIFF Export . . . . .	7
	Interactive Mode . . . . .	7
	Loop . . . . .	7
	Zoom and Sizing . . . . .	7
	Rate . . . . .	7
	Background Color . . . . .	7
	The Export Process . . . . .	8
<b>3</b>	<b>3 Optimizing Media for Phosphor</b>	<b>9</b>
	Types of Source Material . . . . .	9
	Framerate . . . . .	9
	Duration . . . . .	9
	Compressing Phosphor Compositions . . . . .	9
<b>4</b>	<b>4 Working with Content on the Web</b>	<b>11</b>
	Previewing Content . . . . .	11
	Embedding Content . . . . .	11

<b>5</b>	<b>5 Javascript Framework</b>	<b>13</b>
	Callbacks . . . . .	13
	currentFrameCallback(frameCount, metadata) . . . . .	13
	onLoad . . . . .	13
	playbackFinishedCallback . . . . .	13
	Control functions . . . . .	13
	play . . . . .	14
	stop . . . . .	14
	debug(boolean) . . . . .	14
	loop(boolean) . . . . .	14
	(int)currentFrameNumber . . . . .	14
	setCurrentFrameNumber . . . . .	14
	Customization . . . . .	14
<b>6</b>	<b>6 Troubleshooting</b>	<b>15</b>
	Source Formats . . . . .	15
	Compatibility . . . . .	15
<b>7</b>	<b>A Appendix A</b>	<b>16</b>

# Chapter 1

## 1 Welcome to Phosphor

### Welcome

Phosphor is a new way of bringing animation and video to the web. Starting with a standard QuickTime file, Phosphor creates crossplatform, web-standards compliant assets which can be played on desktop and mobile devices without plugins or even video decoding capabilities. As long as a browser supports JavaScript and the ability to show images, it can display Phosphor content.

Using Phosphor is simple and fun. This manual will walk you through the basics, and will explore the situations in which Phosphor is most appropriate.

## Chapter 2

## 2 Getting Started

### Interface Basics

Upon launch, Phosphor displays a grid of previously accessed media, as well as an assortment of sample source material. If you're still in the process of getting familiar with Phosphor, the sample media gives you the opportunity to begin experimenting immediately. You can use the “open” button (or select open from the file menu) to open any other modern QuickTime movie on your computer.



Figure. 2.1: Phosphor File Browser

## Supported Media

Phosphor can open most modern QuickTime files, including H.264 and ProRes. In addition, Phosphor can open image sequences in any standard format (PNG, JPG, etc).

See chapter three for tips on how to prepare media for Phosphor.

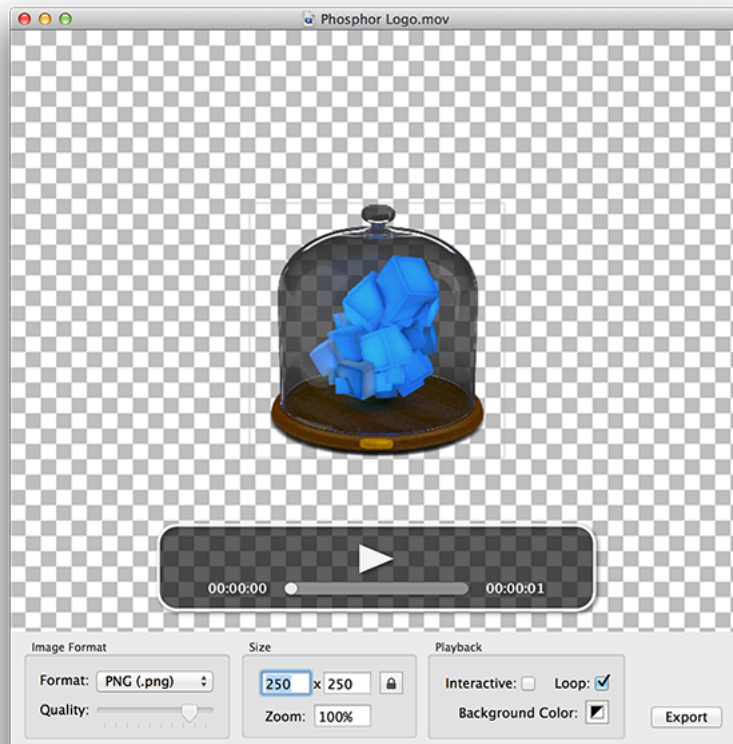


Figure. 2.2: Phosphor Document

## Exporting

Once you've loaded your movie into Phosphor, you can adjust a variety of settings, including format, quality, and background color.

### PNG Export

PNG files are a good choice for movies which include an alpha layer. The transparency will be preserved, allowing you to integrate your Phosphor composition with an existing background design on your website.

## JPG Export

JPEG files are the best choice for most types of content, as the files are highly compressed and easily worked with. When working with JPEG files, the quality slider adjusts how heavily compressed the resulting files will be.

## TIFF Export

TIFF files are uncompressed, and are not suitable for use directly within a webpage. Use this option if you'll be compressing your files separately, using an external tool.

## Interactive Mode

Interactive Mode will allow users to scrub back and forth through your composition using their mouse or a touch interface. This is ideal for product demonstrations and 3d visualizations. The compositions generated in interactive mode will be larger than non-interactive mode.

## Loop

The loop setting will cause your composition to loop when played in a webbrowser. You can adjust this after exporting by editing the HTML generated by Phosphor.

## Zoom and Sizing

If you're working with content captured by a traditional video camera, it's very large. You can resize your video right within Phosphor, using either the "zoom" adjustment, or by entering height and width values. The lock icon will allow you to adjust the aspect ratio of your image.

## Rate

The rate control allows you to adjust how fast your Phosphor composition plays in the browser. The default is 100%, meaning the composition will play at the same rate as the source file.

## Background Color

If your source media is in the ProRes 4444 format, and contains alpha (transparency) information, you can use the adjust the background color using the "Set Background Color" menu option (under the "encode" menu). This color will be composited into your image when exporting in the JPEG format, which does not support transparency. If your target format supports transparency (PNG or TIFF), the transparency in your source video will be preserved.

## The Export Process

Once you've made your selections, press the export process to generate your Phosphor output. Phosphor will analyze your content and figure out the most efficient way to prepare it for delivery. With a longer movie, this can be time consuming.

You can work with additional movies within Phosphor during an export - simply select "open" from the "file" menu to begin working on another composition.



# Chapter 3

## 3 Optimizing Media for Phosphor

### Types of Source Material

Phosphor works best with short movies which do not contain a lot of visual noise. Visual noise includes things like static on a television screen, a snow storm, or ripping water.

Sources can be either QuickTime files (H.264, ProRes) or image sequences. Image sequences should be in their own folder, and the files should be numbered sequentially. By default, image sequences play at a rate of 20 frames per second, but this can be altered using the “rate” control.

### Framerate

Video typically runs at 30 frames per second. In many cases, you can achieve fluid motion with a lower framerate, and thus benefit from greater efficiency. Using your video editing software to convert your media to 15 frames per second will substantially reduce the size of your composition.

The “Rate” control in Phosphor allows you to control the rate at which the final composition plays, but will not alter the overall filesize.

### Duration

Videos longer than approximately 30 seconds will generally result in large compositions, which may not work well on mobile devices or devices with limited memory. For these types of content, a more traditional video embed is more appropriate.

### Compressing Phosphor Compositions

Phosphor outputs standard image files, in either JPG, PNG or TIFF format. These files can be modified with traditional image manipulation tools. For example, the free [ImageOptim](#) application can often substantially reduce the size of PNG and JPG files, without an additional reduction in quality.

When editing Phosphor images, it's important that you don't resize the images. If the overall resolution of the file changes, the Phosphor library will no longer be able to display the composition.

# Chapter 4

## 4 Working with Content on the Web

### Previewing Content

After exporting, you will have access to a folder containing a mix of file types - HTML, JavaScript, JSON and images. In most cases, double-clicking the index.html file will allow you to preview your composition within your local webbrowser. Some security configurations or browser settings may prevent you from loading local content within your browser. In these cases, you may need to adjust a setting or load your content from a traditional webserver.

### Embedding Content

The index.html file generated by Phosphor contains a basic scaffolding for your composition, including basic controls. If you'd like to embed the composition in an existing website, there are a few important items to include. First, you'll need to include the "jquery" and "phosphorframework" javascript files. Then, you'll need the "img" tag with the "anim\_target" id attached to it. Finally, you'll need the first few lines of the "script" block:

```
$(document).ready(function(){
    player_test = new PhosphorPlayer('anim_target_test');
    phosphorCallback_test = function(data) {

        /**
         * Instantiate the player. The player supports a variate of callbacks for deeper
         * integration into your site.
         */

        framecount = data.frames.length;
        player_test.load_animation({
            imageArray:["test_atlas000.jpg"],
            animationData: data,
            loop: true,
            onLoad: function() {
                player_test.play();
            }
        })
    }
})
```

```
    });  
  }  
  var jsonpScript = document.createElement("script");  
  jsonpScript.type = "text/javascript";  
  jsonpScript.id = "jsonpInclude_test";  
  jsonpScript.src = "test_animationData.json" + "p";  
  document.getElementsByTagName("head")[0].appendChild(jsonpScript);  
  
  });
```

The other code within the script block enables playback controls and interactivity - you may choose to leave those out, depending on your particular needs.

# Chapter 5

## 5 Javascript Framework

The sample HTML generated by Phosphor shows an example of a basic phosphor composition (see Chapter 4). However, there are further opportunities for customization for users familiar with Javascript development.

### Callbacks

The Phosphor Framework has three callbacks which may be specified as parameters on the `load_animation` function. These allow you to trigger custom actions based on events which occur within the Phosphor framework.

#### `currentFrameCallback(frameCount, metadata)`

This callback will be triggered each time a new frame is drawn. Your function declaration should include two parameters, which will be passed the current framecount as well as any metadata associated with that frame. This count will reset to zero each time the composition repeats. You may manually add metadata to a frame by adding an “m” key to the json object for each frame containing arbitrary metadata.

#### `onLoad`

This function will be triggered after all of the images for your composition have successfully been loaded from the server. Generally, this is used to trigger the `play()` function.

#### `playbackFinishedCallback`

This callback will be triggered each time the last frame in your composition is displayed. If your composition is set to loop, this will be triggered multiple times.

### Control functions

Phosphor exposes a variety of functions for playback control, which you may choose to trigger from your own code.

## **play**

Begins playback of your composition.

## **stop**

Stops playback of your composition.

## **debug(boolean)**

Enables or disable debug overlay.

## **loop(boolean)**

Enables or disables looped playback of the composition.

## **(int)currentFrameNumber**

Get the current frame number.

## **setCurrentFrameNumber**

Jump to a particular frame within the composition.

## **Customization**

The Phosphor Framework is relatively straightforward, and should be easy to customize by developers looking for functionality not available out of the box. We'd love to see what you build - send us your results at [support@divergentmedia.com](mailto:support@divergentmedia.com) or fork the [framework on GitHub](#).

# Chapter 6

## 6 Troubleshooting

### Source Formats

Phosphor supports QuickTime videos which are encoded using H.264 and Apple ProRes. Future versions of Phosphor will support loading image sequences.

### Compatibility

Phosphor compositions should work in any browser that supports the canvas tag. In some cases, older browsers which do not natively support the canvas tag can gain that support via the [excanvas javascript library](#).

Running multiple Phosphor compositions on one page may lead to performance problems on older mobile devices (smartphones) and should be avoided for sites which are likely to be heavily trafficked by mobile browsers.

## Chapter 7

## A Appendix A

Phosphor makes use of the libjpeg library from the [Independent JPEG Group](#).