

CS4405

Extending JPEG

Transparent JPEG?

▶ The JPEG Image Compression FAQ

- "No. JPEG does not support transparency and is not likely to do so any time soon. ... The only real solution is to combine lossy JPEG storage of the image with lossless storage of a transparency mask using some other algorithm. Developing, standardizing, and popularizing a file format capable of doing that is not a small task. As far as I know, no serious work is being done on it; transparency doesn't seem worth that much effort."
- Last-modified: 28 March 1999

Web Pages Containing Images

▶ Use the best file format

- For detailed artwork or photographs JPG and 24-bit PNG are typically used because they have a much larger colour palette
- A 24-bit PNG results in superior image quality, this comes at the price of a larger file size
- When you can, use JPG instead and adjust the quality setting so you can compress the image as much as possible within your desired tolerance for image quality

PNG and JPEG



Compositing



W3C CSS Masking

- ▶ CSS Masking provides two means for partially or fully hiding portions of visual elements: masking and clipping
- ▶ Masking describes how to use another graphical element or image as a luminance or alpha mask

`<mask-image> = [<image> [alpha | luminance]?] | none`

W3C CSS Masking

- ▶ A value of **alpha** indicates that the alpha values of the mask image should be used as the mask values
- ▶ A value of **luminance** indicates that the luminance values of the mask image should be used as the mask values
- ▶ A value of **none** counts as an image layer but does not mask the element

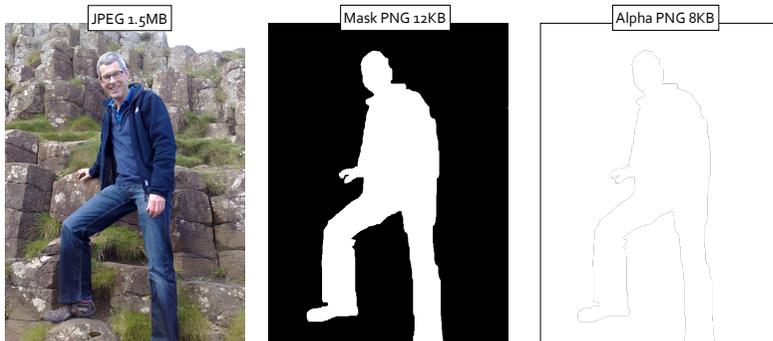
Default value is **alpha**

Examples

- ▶ The procedure for calculating mask values assumes the content of the mask is a four-channel RGBA graphics object
 - If a three-channel RGB graphics object that is used in a mask the effect is as if the object were converted into a four-channel RGBA image with the alpha channel uniformly set to 1

```
body { mask-image: linear-gradient(black 0%, transparent 100%) }  
p { mask-image: none }  
div { mask-image: url(t1.png), url(tr.png) }
```

Mask



CSS Masking

```
<html>
<head>
<title>CSS Mask Test</title>
</head>
<body>

</body>
</html>
```

```
img#example { mask-image: url(example-mask-full-alpha.png) }
```

```

```

CSS Masking
W3C Working Draft 15 November 2012

Extend JPEG

- ▶ The JPEG image format makes provision for application specific data elements
 - There are 16 available APPx markers
 - JPEG APPn segments are limited to just under 64k bytes
- ▶ Use one of these to contain a mask image
 - The embedded image have the data in the alpha channel (i.e. the embedded image must be in a format like PNG)

Transparent JPEG

```
$ jpegapp -h
Usage: jpegapp [options] [input file]
e.g.: jpegapp -i 7=alpha0content-type:image/png -i 7#mask.embed image.jpg

Options:
-i --insert seg#file insert the contents of file as APPx where x=seg
--insert seg=string insert the argument as APPx where x=seg
-r --remove seg remove all APPx segments where x=seg
-o --output file send output to file instead of stdout
-v --verbose Print a lot of debug messages
-h --help This help message
```

<http://jim.studt.net/jpeg-alpha/>



```
jpegapp -i 7=alpha0content-type:image/png \
-i 7=alpha0data:example-mask-full-alpha.png \
example.jpg -o example-embed.jpg
```

```
( echo -n "alpha0data:"; cat example-mask-full-alpha.png ) > example-mask-full-alpha.embed
```

```
jpegapp -i 7=alpha0content-type:image/png \
-i 7#example-mask-full-alpha.embed \
example.jpg -o example-embed.jpg
```

Original Size: 1,481,466 bytes
Embed Size: 1,488,229 bytes

JavaScript

```
jpeg = new JPEG( new ByteString(data));  
var contentType, contentData;  
for (;;) {  
  var m = jpeg.nextMarker();  
  
  if ( m.tag == 0xe7 ) {  
    var magic = m.value.nextBytes(6).toString();  
  
    if ( magic == 'alpha0' ) {  
      var kind = '';  
      for (;;) {  
        var c = m.value.next();  
        if ( c == undefined || c == '.'.charCodeAt(0) ) break;  
        kind = kind + String.fromCharCode(c);  
      }  
      if ( kind == 'content-type' ) {  
        contentType = m.value.nextBytes().toString();  
      } else if ( kind == 'data' ) {  
        contentData = m.value.nextBytes().toString();  
      }  
    }  
  }  
  if ( m.tag == 0xd9 ) break; // end of image tag  
}
```

```
if ( contentType && contentData ) {  
  var url = "data:" + contentType + ";base64," + encode64( contentData );  
  if ( !useImageMask ) {  
    $(val).css( "webkit-mask-box-image", "url(" + url + ") 0 0 0 stretch stretch" );  
  } else {  
    val.setAttribute( "data-alpha-src", url );  
    create_alpha_jpeg(8, val);  
  }  
}
```

Look for APP7 segments that begin 'alpha0' – should be two

- content-type: followed by mime type
- data: followed by name or image data

<http://jim.studt.net/jpeg-alpha/>

JavaScript

```
function create_alpha_jpeg(idx, img) {  
  var alpha_path = img.getAttribute('data-alpha-src');  
  
  img.style.visibility = 'hidden';  
  
  var image = document.createElement('img');  
  image.src = img.src;  
  image.onload = function () {  
  
    var alpha = document.createElement('img');  
    alpha.src = alpha_path;  
    alpha.onload = function () {  
      var canvas = document.createElement('canvas');  
      canvas.width = img.width;  
      canvas.height = img.height;  
      img.parentNode.replaceChild(canvas, img);  
  
      var context = canvas.getContext('2d');  
      context.clearRect(0, 0, canvas.width, canvas.height);  
      context.drawImage(alpha, 0, 0, canvas.width, canvas.height);  
      context.globalCompositeOperation = 'source-in';  
      context.drawImage(image, 0, 0, canvas.width, canvas.height);  
    }  
  }  
}
```

Replace the image with a canvas element
Draw the composite of the JPEG image
and the mask image

