



Software-Development and Travel Technology

jQuery Background Canvas Plugin API

Version 0.8.5.b

Thomas Maierhofer

Content

Content.....	2
Introduction.....	3
What is the JQuery Background Canvas Plugin?	3
Supported Browsers	3
Installation.....	3
Download the newest Release	3
JavaScript Files.....	3
Code Block with the necessary scripts:	3
excanvas.js.....	3
JQuery.js	3
JQuery.mseitutilities.js	3
JQuery.backgroundCanvas.js	3
Using the jQuery Background Canvas Plugin	4
General Building Block	4
Injection of the Canvas Element	4
Serving the “load” and “resize” events	4
Providing a user defined paint function.....	4
Advanced	5
Delayed Event Handling	5
Injection Process	5
Single Background Canvas.....	6
API Reference	7
Canvas Injection and Drawing.....	7
\$(Element).backgroundCanvas(makeElementTransparent, transparentColorIE6)	7
\$(Element).backgroundCanvasPaint(paintFkt)	7
paintFkt(context, width, height, elementInfo).....	7
\$(Element).makeElementTransparent(transparentColorIE6).....	8
Contact and Imprint	9

Introduction

What is the JQuery Background Canvas Plugin?

The HTML5 standard introduces the <CANVAS> element, a resolution dependent bitmap canvas on which 2D graphic objects can be rendered. With JavaScript one can draw shapes like circles, rectangles and lines on this element. The jQuery background canvas plugin injects such a canvas element behind a div and provides functions to draw on it. When the background of the div is made transparent, this canvas element can be seen and it will provide the new background for the div.

Supported Browsers

The jQuery background canvas is designed to run on every browser that supports the HTML5 <CANVAS> tag. It is supported by Firefox, Opera, Safari, Google chrome. With the Explorer Canvas emulation it can also be used on Internet Explorer.

Download Explorer Canvas: <http://code.google.com/p/explorercanvas/>

Installation

Download the newest Release

The project is hosted on CodePlex, and the newest Release can be downloaded from there.

Download Background Canvas: <http://jqueryplugins.codeplex.com>

JavaScript Files

Code Block with the necessary scripts:

```
<!--[if IE]>
<script type="text/javascript" src="/Script/excanvas.js" ></script>
<![endif]-->
<script src="/Script/JQuery.js" type="text/javascript" ></script>
<script src="/Script/JQuery.mseutilities.js" type="text/javascript" ></script>
<script src="/Script/JQuery.backgroundCanvas.js" type="text/javascript"></script>
```

excanvas.js

Explorer Canvas Script to support Internet Explorer. Don't forget the conditional comments to prevent the inclusion on other browsers than Internet explorer.

JQuery.js

The jQuery Library itself.

JQuery.mseutilities.js

Some utilities needed for background canvas (included in the package)

JQuery.backgroundCanvas.js

The jQuery background canvas plugin itself

Using the jQuery Background Canvas Plugin

General Building Block

Injection of the Canvas Element

The Canvas element can be injected in the `$(document).ready()` function. In this code sample all elements with class "Test" will get an canvas object injected:

```
$(document).ready(function()
{
    $(".Test").backgroundCanvas();
});
```

Serving the "load" and "resize" events

After the DOM is constructed, the canvas content can be drawn. If the user resizes the browser window, the content must be drawn again. We can provide a `DrawBackground` Function and call it from the "load" and "resize" events:

```
function DrawBackground() {
    $(".Test").backgroundCanvasPaint(TestBackgroundPaintFkt);
}

// Draw the background on load and resize
$(window).load(function () { DrawBackground(); });
$(window).resize(function() { DrawBackground(); });
```

Providing a user defined paint function

To draw on the canvas, we must provide a paint function to do the job:

```
function TestBackgroundPaintFkt(context, width, height, elementInfo)
{
    var options = {x:0, height: height, width: width,
        radius:14, border: 0 };

    // Draw the red border rectangle
    context.fillStyle = "#FF0000";
    $.canvasPaint.roundedRect(context,options);

    // Draw the gradient filled inner rectangle
    var backgroundGradient = context.createLinearGradient(0, 0,
        0, height - 10);

    backgroundGradient.addColorStop(0 , '#AAAAFF');
    backgroundGradient.addColorStop(1, '#AAFFAA');

    options.border = 5;
    context.fillStyle = backgroundGradient;
    $.canvasPaint.roundedRect(context,options);
}
```

Advanced

This Section describes more advanced techniques to get the best rendering performance and visual appearance.

Delayed Event Handling

Especially the “resize” event will be triggered very often. To get a better performance, we could delay the execution of these events. The following code delays the event 50mS and eats up multiple events in this 50mS to a single call of DrawBackground():

```
var drawBackground = false;

$(window).load(function()
{
    drawBackground = true;
    window.setTimeout(function()
    {
        if (drawBackground)
        {
            drawBackground = false;
            DrawBackground();
        }
    }, 50);
});

$(window).resize(function()
{
    drawBackground = true;
    window.setTimeout(function()
    {
        if (drawBackground)
        {
            drawBackground = false;
            DrawBackground();
        }
    }, 50);
});
```

Injection Process

The canvas is injected by wrapping the inner content of the element into a child div with the class “jbgContentDiv”. It is prepended with a div(class “jbgCanvasDiv”) containing the canvas(class “jbgCanvas”).

Element -> Content

will be after injection:

Element -> DIV{jbgCanvasDiv} -> CANVAS{ jbgCanvas }

-> DIV{ jbgContentDiv } -> Content

The Canvas DIV and Content DIV are overlaid by relative/absolute positioning.

Single Background Canvas

It is possible to use one single background canvas for the complete background drawing. This helps to keep the DOM small and renders faster than painting on a vast amount of injected canvas objects.

API Reference

Canvas Injection and Drawing

\$(Element).backgroundCanvas(makeElementTransparent, transparentColorIE6)

Injects the canvas behind the given element. By default, the background and border of the element is set to “transparent” to show the canvas. This can be prevented by setting the “makeElementTransparent” flag to false. On IE6 a transparent color is needed to set the border transparent. If the standard backgroundCanvas color (#fac2f5) is used by the content of the element, a different color can be provided in the “transparentColorIE6” parameter.

```
// Inserting the canvas in all elements with the class “Test”
$(".Test").backgroundCanvas();

// preventing to set the border and background to transparent
$(".Test").backgroundCanvas(false);

// Setting a different transparency color for IE6
$(".Test").backgroundCanvas(true, "#ffaaff");
```

\$(Element).backgroundCanvasPaint(paintFkt)

Calls the provided paint function “paintFkt” for the given element.

paintFkt(context, width, height, elementInfo)

The paint function (name it as you want) allows painting on the canvas context provided in the parameter “context”. The width and height of the canvas is also provided by the according parameters. Additional Information is provided by the “elementInfo” parameter, which is an object. It provides this members:

- elementInfo.**Canvas**: The canvas object itself
- elementInfo.**\$canvas**: The canvas object as a jQuery object.
- elementInfo.**\$canvasDiv**: The canvas DIV as a jQuery object.
- elementInfo.**\$content**: the content DIV as a jQuery object.
- elementInfo.**\$this**: The element itself as a jQuery object.
- elementInfo.**borderLeft**: The left border of the element (CSS).
- elementInfo.**borderRight**: The right border of the element (CSS).
- elementInfo.**borderTop**: The top border of the element (CSS).
- elementInfo.**borderBottom**: The bottom border of the element (CSS).
- elementInfo.**paddingTop**: The top padding of the element (CSS).
- elementInfo.**paddingBottom**: The bottom padding of the element (CSS).
- elementInfo.**paddingLeft**: The left padding of the element (CSS).
- elementInfo.**paddingRight**: The right padding of the element (CSS).

\$(Element).makeElementTransparent(transparentColorIE6)

This function makes the border and the background of the element transparent, so that the canvas behind the element can be seen. On IE6 a transparent color is needed to set the border transparent. If the standard backgroundCanvas color (#fac2f5) is used by the content of the element, a different color can be provided in the “transparentColorIE6” parameter.

```
// Make all elemets with class “Test” transparent
$(".Test").makeElementTransparent();

// Setting a different transparency color for IE6
$(".Test").makeElementTransparent("#ffaaaff");
```


Contact and Imprint

Homepage: <http://www.maierhofer.de/EN/>

<http://www.maierhofer.de/en/open-source/jquery-background-canvas-plugin.aspx>

Maierhofer Software

Thomas Maierhofer

Hauptstrasse 32

88427 Bad Schussenried

Tel: 07525-911000

Fax: 07525-911001

E-Mail: Info@maierhofer.de