

# Cfwindow Positioning Trial and Error

Posted At : November 6, 2007 11:17 AM | Posted By : Mark Kruger

Related Categories: Coldfusion MX 7, Coldfusion 8

I love the new widgety stuff in Coldfusion 8. The Cfwindow tag caught my attention right away. it seemed like a quick and easy way to frame content on a page and make it manageable. For example, I had a list of items that appear on the home page of our tracking system. It's the "hotlist" - a list of open and active work orders and projects. I decided to use Cfwindow to make it draggable and re-sizable. It was easy.

```
<cfwindow
    height="475"
    resizable="True"
    initshow="True"
    x="380"
    y="150"
    closable="False"
    draggable="True"
    title="Hot List"
    name="hotlist"
    headerstyle="background-color: ##A7B3C2;">
Some looping code that outputs a list of links.
</cfwindow>
```

This worked splendidly. I ended up with exactly what I wanted - a nice scrolling window that was easy to drag and resize to accommodate more entries.

Since I loved the little drop shadow thingy I decided I would use it on the right hand side menu as well - just to frame in the menu with a header. Oops... this turned out to be quite difficult. Why? Because there is no way to *anchor* the window to the layout. The layout is dynamic on the right. The right hand menu "clings" to the right side of the screen - meaning the X and Y coordinates are different for each screen size. But the Cfwindow appears *over* whatever is on the screen and the "x" and "y" coordinates make it's position absolute. I went hunting for an answer.

## First Try - Change To Relative Position.

**Russ Michaels** (a.k.a. "Snake") on the CF Guru list suggested changing the style "position" attribute to "relative". He came up with this sample code.

```
<div id=parentDiv>
<cfwindow name="foo"
    initShow="true"
    title="My Window"
    draggable="true"
    closable="true"
    modal="true">

<p>Hello World!</p>
<button onclick="doStuff()" ">Click Me</button>

</cfwindow>
</div>

<script>
```

```
function doStuff()

{
var cfWindow = ColdFusion.Window.getWindowObject('foo');
var myWin = document.getElementById(cfWindow.id);
myWin.style.position = "relative";
}
</script>
```

This *sort of* works. If you check out [this link](#) you will see that it does indeed reposition the window. The problem is that the drop shadow, which is a separate div tag that is aligned with the window and offset to create the affect, does not "move" when the window is repositioned. This leaves a love grey area on the screen - sort of like when you accidentally leave a tarp out on the back lawn for a day or 2 and the grass dies.

## Second Try - "alignTo"

I thought I'd try figuring out some of the native methods of the JavaScript Coldfusion Window object. I created some looping code that output the methods of the object and examined them (I know, there are utilities out there that can do this for me - blah blah blah). One of the methods I found was "alignTo". As arguments it took a string representing an object ID in the DOM, and an x/y relative to that object. So I tried this:

```
<div id=parentDiv style="margin-top: 400px; margin-right: 200px;">
<cfwindow name="foo"
  initShow="true"
  title="My Window"
  draggable="true"
  closable="true"
  modal="true">

<p>Hello World!</p>
<button onclick="doStuff()">Click Me</button>

</cfwindow>
</div>

<script>

function doStuff()

{
  var cfWindow = ColdFusion.Window.getWindowObject('foo');
  cfWindow.alignTo("parentDiv",0,0);
}
</script>
```

Actually, this worked splendidly as you can see by following [this link](#). The window positions itself correctly at the div location. Using onload( ) or some other creative JavaScript I could now make the windows fly around the screen and align themselves with layout divs.

I'm not particularly satisfied with this solution. It seems to me that having windows fly around the screen could be disconcerting, and I wonder if onload would suffer from synchronization problems since the JS files needed are external. It's possible my function for positioning could fire before the "Coldfusion.Window" object is ready to be manipulated.

My final solution was to try "cfpod" instead. It created nice windowy looking menu boxes for me. You can close and open them or move them around - but for the menu bar that suited me fine.