

Actions Tech. Note 2

Draft 1.0

Getting & Setting <div> Style Attributes.

Using JavaScript it is possible to get all or any of the <div> tags on your page. You would do this within your action in the same way as getting details of any tag, using `fwFindAll` or `fwFind`.

```
var allDivs = fwDocument.fwTags.fwFindAll("div");
```

The above call will yield an array of tags that you can then loop through one by one.

```
for (var i in divs)
{
    var div = divs[i];

    :
    :
}
```

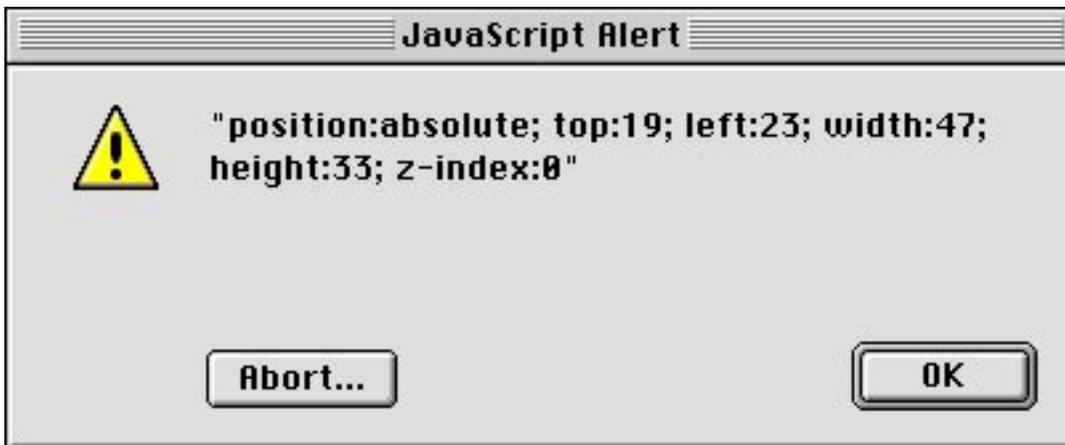
The <div> tag of a typical layer within Freeway is something like this:

```
<div id="item1" style="position:absolute; top:19; left:23; width:47; height:33; z-index:0">
```

You can access the style attribute of the tag as you can with any tag in the tag stream. So the following code fragment

```
alert(div.style);
```

would yield an alert something like:



Basically within FW3 all the attributes within the style are available only as a text string. This means that to manipulate the attributes you will have to take this text string and parse it. So to get the `left` attribute within the style you will have to get the style attribute and, find the text between the colon and the semicolon. Similarly to set this attribute you will have to cut out this portion of text and replace it with the new attribute.

JavaScript has very powerful string manipulation functions, and it is quite straightforward to write functions that perform this sort of manipulation. Below are two functions that do this - however you could very easily write your own.

```
// Gets a "CSS" attribute such as "position:absolute" from a tag value
// returns NULL if the attribute can not be found
function GetCSSAttribute(tag, fieldName, attributeName)
{
    if (tag==null)
        return null;

    var tagField = tag[fieldName];
    if (tagField == null)
        return null;
    else
    {
        var tagField = tagField.toString();
        var leftPos = tagField.indexOf(attributeName+":");
        if (leftPos > 1)
            leftPos = tagField.indexOf(" "+attributeName+":") + 1;
        if (leftPos > 0)
        {
            leftPos += attributeName.length+1;
            var rightPos = tagField.indexOf(";", leftPos);
            if (rightPos == -1)
                rightPos = tagField.length-1;

            return tagField.slice(leftPos, rightPos);
        }
        return null;
    }
}
```

```
// Sets a "CSS" attribute such as "position:absolute" in a tag value
function SetCSSAttribute(tag, fieldName, attributeName, attributeValue)
{
    if (tag==null)
        return;

    var tagField = tag[fieldName];
    if (tagField == null)
        tag[fieldName] = ''+attributeName+'='+attributeValue+'';
    else
    {
        var tagField = tagField.toString();
        var leftPos = tagField.indexOf(attributeName+":");

        if (leftPos > 1)
            leftPos = tagField.indexOf(" "+attributeName+":") + 1;
        if (leftPos > 0)
        {
            leftPos += attributeName.length+1;
            var leftStr = tagField.slice(0, leftPos);
            var rightStr = tagField.slice(leftPos, tagField.length);
            var rightPos = rightStr.indexOf(";");
            if (rightPos == -1)
                rightPos = rightStr.length-1;
            rightStr = rightStr.slice(rightPos, rightStr.length);
            tagField = leftStr+attributeValue+rightStr;
        }
    }
    else
}
```

```

        tagField = ''+tagField.slice(1,-1)+
            "; "+attributeName+": "+attributeValue+' ';
    }
    tag[fieldName]= tagField;
}

```

So a call to:

```
alert(GetCSSAttribute(div, "style", "top"));
```

would yield an alert with "19". Similarly a call to:

```
SetCSSAttribute(div, "style", "top", 120);
```

Would set this property to 120.

The only caveat here is that the `GetCSSAttribute` function will return a string - and to use it as a number you will have to convert it to one. JavaScript provides a `parseInt` function that does this.

The following action is designed to go through all the `<div>`s on a page and renumber their 'z' index. the actual purpose if the action was a part of something that would create partial HTML files as includes - however it demonstrates the possibilities of reading and writing these properties.

```

<page-action name="Renumber z-indices">
<action-number name="Start" default=0 minimum=0/>
<action-checkbox name="offset" default="No"/>
<action-javascript>

// Gets a "CSS" attribute such as "position:absolute" from a tag value
// returns NULL if the attribute can not be found
function GetCSSAttribute(tag, fieldName, attributeName)
{
    if (tag==null)
        return null;

    var tagField = tag[fieldName];
    if (tagField == null)
        return null;
    else
    {
        var tagField = tagField.toString();
        var leftPos = tagField.indexOf(attributeName+":");
        if (leftPos > 1)
            leftPos = tagField.indexOf(" "+attributeName+":") + 1;
        if (leftPos > 0)
        {
            leftPos += attributeName.length+1;
            var rightPos = tagField.indexOf(";", leftPos);
            if (rightPos == -1)
                rightPos = tagField.length-1;

            return tagField.slice(leftPos, rightPos);
        }
        return null;
    }
}

```

```

// Sets a "CSS" attribute such as "position:absolute" in a tag value
function SetCSSAttribute(tag, fieldName, attributeName, attributeValue)
{
    if (tag==null)
        return;

    var tagField = tag[fieldName];
    if (tagField == null)
        tag[fieldName] = ''+attributeName+':'+attributeValue+'';
    else
    {
        var tagField = tagField.toString();
        var leftPos = tagField.indexOf(attributeName+":");

        if (leftPos > 1)
            leftPos = tagField.indexOf(" "+attributeName+":") + 1;
        if (leftPos > 0)
        {
            leftPos += attributeName.length+1;
            var leftStr = tagField.slice(0, leftPos);
            var rightStr = tagField.slice(leftPos, tagField.length);
            var rightPos = rightStr.indexOf(";");
            if (rightPos == -1)
                rightPos = rightStr.length-1;
            rightStr = rightStr.slice(rightPos, rightStr.length);
            tagField = leftStr+attributeValue+rightStr;
        }
        else
            tagField = ''+tagField.slice(1,-1)+"; "+attributeName+": "+attributeValue+'';

        tag[fieldName]= tagField;
    }
}

function fwBeforeEndBody()
{
    var divs = fwDocument.fwTags.fwFindAll("div");
    var zIndices = new Array();
    var minZindex = 32000;
    var firstZIndex = parseInt(fwParameters["Start"]);
    var offset = fwParameters["offset"].fwBoolValue;

    // loop through all the div's, and record their z-index,
    // also find the minimum z-index that is set.
    for (var i in divs)
    {
        var div = divs[i];

        // get the z index
        var zIndex = GetCSSAttribute(div, "style", "z-index");

        if (zIndex != null)
        {
            // convert the zIndex to a string
            zIndex = parseInt(zIndex);

            // find the maximum
            if (minZindex > zIndex)
                minZindex = zIndex;
        }

        zIndices.push(zIndex);
    }
}

```

```
if (offset)
  minZindex = 0;

// loop through all the div's that have a z-index and renumber them
for (var i in divs)
{
  var div = divs[i];
  var zIndex = zIndices[i];

  if (zIndex != null)
  {
    var newzIndex = zIndex-minZindex+firstZIndex;
    SetCSSAttribute(div, "style", "z-index", newzIndex);
  }
}
}
```

```
</action-javascript>
</page-action>
```