

Handling JavaScript event model differences in Internet Explorer and Firefox

Posted At : 14 April 2010 15:15 | Posted By : Shaun McCran

Related Categories: Internet Explorer, Browsers, Javascript

A few weeks ago I created an inline editing system for a Content Management System. It allowed a user to click on a page element and edit it inline. The updated data was then submitted through an AJAX request.

I have recently discovered an issue with some of the JavaScript code that I wrote in the event handling routine.

The original article is here:

<http://www.mccran.co.uk/index.cfm/2010/2/10/Dynamically-editing-web-content-inline-using-JavaScript-and-AJAX>

The event handling section of the script is repeated below. The problem stems from Internet Explorer and Firefox handling click events differently.

```
function getId(e) {
    var targ;
    if (!e) var e = window.event;
    if (e.target) targ = e.target;

    else if (e.srcElement) targ = e.srcElement;

    if (targ.nodeType == 3) // defeat Safari bug
        targ = targ.parentNode;

    //thisTarget = e.target.id; // FF
    //thisTarget = e.srcElement.id; //IE

    thisTarget = clickHandler(e)
    //alert(thisTarget);
}
```

Examining the code above I can see that to track event handling in Internet Explorer I need to use:

```
e.srcElement.id
```

This is because IE uses the window.event model, whereas all other browsers intrinsically handle the event model. This means that you can just say:

```
e.target.id
```

As Firefox will automatically resolve the 'target' value.

The Fix

To create a cross browser script that will work in both Internet Explorer and Firefox we can use the returned value from a function like this:

```
function clickHandler(e) {  
  return (window.event) ? window.event.srcElement.id : e.target.id;  
}
```

This function will evaluate the response of two arguments from a switch statement. Each browser will only understand *one of each of these statements*, so the browser will only return the value of the statement it understands (Both statements return the same calculated value).

And that is a cross browser Event handling script.

