

Re: Dependent dropdowns

Source: <http://www.tech-archive.net/Archive/Scripting/microsoft.public.scripting.jscript/2004-08/0114.html>

From: Dave Anderson (*GTSPXOESSGOQ_at_spammotel.com*)

Date: 08/07/04

Date: Sat, 7 Aug 2004 11:49:47 -0500

"James Baker" wrote:

> *Looking for a simple javascript solution to this and it's annoying me*
> *to no end.*

>

> *Long story short, I have two drop downs on a form (using VBScript as*
> *my server side language). Drop down 2 will have nothing in it, until*
> *you select an item from Drop down 1. I can't handle a submit here, I*
> *need to have it all be preloaded into Javascript (most likely arrays)*
> *that I can just access from the client side. The first drop down is*
> *"PropertyType" and the second is "LoanType". Essentially, each*
> *PropertyType has a list of loan types that are acceptable for it.*
> *Right now I keep getting lost in trying to design this array. What's*
> *the best javascript approach to a problem like this?*

I find that one simple solution is to use multiple SELECT elements, displaying only one at a time. To the user, it simply looks like one SELECT object.

```
<SELECT NAME="PropertyType" ONCHANGE="doStuff()"> ... </SELECT>
```

```
<SELECT NAME="LoanType1" STYLE="display:none"> ... </SELECT>
```

```
<SELECT NAME="LoanType2" STYLE="display:none"> ... </SELECT>
```

```
...
```

```
<SELECT NAME="LoanTypeN" STYLE="display:none"> ... </SELECT>
```

```
var dependentList =
```

```
  new Array(document.myform.LoanType1, ..., document.myform.LoanTypeN)
```

```
function doStuff() {
```

```
  for (var i=0; i<dependentList.length; i++)
```

```
    dependentList[i].style.display = "none"
```

```
  { additional logic for deciding which element to reveal.
```

```
    To do so, set the .style.display property to "block" }
```

```
}
```

The advantages of this approach are (a) it is easier to reliably build a set

microsoft.public.scripting.jscript: Re: Dependent dropdowns

of OPTION tags from arbitrary lists of data available on the server than to build JS arrays, and (b) the demands on the client are smaller. This second point only really matters when the secondary lists (and subsequently the arrays) are long. Deconstruction and rebuilding of the dependent list can bog down lesser machines.

--

Dave Anderson

Unsolicited commercial email will be read at a cost of \$500 per message. Use of this email address implies consent to these terms. Please do not contact me directly or ask me to contact you directly for assistance. If your question is worth asking, it's worth posting.