

Using a Calculation to Hide/Show a Field

By Thom Parker, www.pdfscripting.com
(For Acrobat/PDF forms)

Acrobat provides easy, non-scripting methods for adding many types of calculations to a form, such as summing entries in a column or adding tax into a total. But what do you do if you want a calculation that does something that's not just math? For example, maybe you have a form on which you want to show a field only when the field's calculation result is greater than zero. While the calculation itself might be straightforward, the total result of the calculation is not. For this we need to use JavaScript.

How it's done

The following script is a tax calculation that multiplies two numbers together, the "OrderTotal" and the "StateTaxRate." The result is the amount of tax on the order and it's displayed in the "Tax" field, the same field that holds the calculation. But in this script, the "Tax" field is going to be hidden if there is zero tax.

```
event.value = this.getField("OrderTotal").value * this.getField("StateTaxRate").value;  
event.target.hidden = (event.value <= 0);
```

The first line of the script is a standard Acrobat Forms calculation. It acquires field values using the "`this.getField()`" function, multiplies them together, and then assigns the result of the calculation to "`event.value`." In Acrobat Forms, as opposed to LiveCycle Forms, the result of a JavaScript calculation is always assigned to "`event.value`."

The next line of the script is our main topic. This line is used to hide or show the "Tax" field. In this code the "Tax" field is referenced with "`event.target`." This looks a bit similar to the first line. Both use the "`event`" keyword. This keyword is an object that contains all the properties of the current event, which is a calculation on the "Tax" field. So, "`event.value`" refers to the value of the "Tax" field, "`event.target`" refers to the "Tax" field itself, and "`event.target.hidden`" refers to the "hidden" property of the "Tax" field. Setting this property to true hides the field, and setting it to false shows the field.

The code on this line creates a true/false (or Boolean) value by using a comparison. The code "`(event.value <= 0)`" returns true if the calculation result is less than or equal to zero, and false if the result is greater than zero.

It's that simple, and this same technique (sometimes with a little massaging) can be used to control all kinds of form field properties through calculation results.

Order Total:

State Tax Rate:

State Tax:

