

Form Builder framework

Contents

Form Design	2
Form Validation	5
Input Text fields	6
Checkbox fields	9
Radio button fields	13
Select drop-downs	17
Tying an Input Text field to an "Other" Selection.....	19
Textarea fields	22

Form Design

A form consists of multiple lines of information. Each line contains a set of data -- a label with an accompanying field (which could be a text field, one or more checkboxes, a drop-down menu, a select box, one or more radio buttons or a textarea field).

In the Form Builder this is represented by a set of DIVs with specific classes. For example:

```
<div class="formline">
  <div class="formset">
    <div class="overlabel">
      Enter a single line of information below</span>
    </div>
    <div class="underfield">
      <input type="text" value="" />
    </div>
  </div>
</div>
```

This displays a label above an input text field that spans the entire width of the column, as displayed here:



The "formline" div contains one or more "formset" DIVs. Each formset represents a pair of divs -- one for a label and one for its accompanying field. If you want the label directly to the left of the field, you would use "leftlabel" in place of "overlabel" and "rightfield" in place of the "underfield." Thus, "leftlabel" DIVs are always (and only) used with "rightfield" DIVs and "overlabel" DIVs are used with "underfield" DIVS in each formset.

Creating a formline with multiple sets of labels and fields is easy. Simply add more formsets within the formline DIV:

```
<div class="formline">
  <div class="formset">
```

```

        <div class="leftlabel">
            Email</span>
        </div>
        <div class="rightfield">
            <input type="text" value="" />
        </div>
    </div>
    <div class="formset noclr">
        <div class="leftlabel">
            Title</span>
        </div>
        <div class="rightfield">
            <input type="text" value="" />
        </div>
    </div>
    <div class="formset noclr">
        <div class="leftlabel">
            Dept.</span>
        </div>
        <div class="rightfield">
            <input type="text" value="" />
        </div>
    </div>
</div>

```

This displays the line as shown here:

Email	<input type="text"/>	Title	<input type="text"/>	Dept.	<input type="text"/>
-------	----------------------	-------	----------------------	-------	----------------------

Note that for the second and following formset DIVs, a class of "noclr" has been added. This allows the additional formsets to be placed to the right of each other. Otherwise, the additional formsets would be pushed below each other.

DIVs can be nested in the "...field" DIVs, giving you the ability to include several short fields within the same DIV. One example of this is the DIV for a phone number. Here is the basic construction of such a formset:

```

<div class="formset">
    <div class="leftlabel">
        Phone
    </div>
    <div class="rightfield">
        <input type="text" value="" />
    </div>
</div>

```

```
</div>
<div class="rightfield">
  <div class="midfield">
    <input type="text" value="" maxlength="3" />
  </div>
  <div class="midfield">
    <input type="text" value="" maxlength="3" />
  </div>
  <div class="midfield">
    <input type="text" value="" maxlength="4" />
  </div>
  <div class="midlabel">
    Ext.
  </div>
  <div class="midfield">
    <input type="text" value="" maxlength="5" />
  </div>
</div>
</div>
```

This is displayed as:



Note that the "rightfield" DIV contains several "midfield" and "midlabel" DIVs to give the user the ability to enter parts of a phone number in multiple fields for area code, exchange, number and extension.

Formlines with multiple formsets require the addition of the "noclr" class to the second and proceeding formsets to keep them from realigning along the left edge.

Each formline and formset must have an ID in order to allow the set to be highlighted when a validation error has been found (more on that in the Validation section). ID's are optional for formline, ...label and ...field DIVs.

Samples of a variety of different form elements can be found on the formTemplate.html page.

Form Validation

Proper validation of the form requires the addition of several established and custom attributes to the various form elements. The basic form elements are:

- Input Text fields
- Checkboxes
- Radio Buttons
- Select drop-downs and lists
- Textarea fields

The labels and fields within each formset are interdependent on each other in order to properly highlight formsets that have validation problems. In the following example, the Input Text field contains a "setid" attribute that is the same as the enclosing formset DIV id:

```
<div class="formline" id="line1" name="line1">
  <div id="set1101" name="set1101" class="formset">
    <div class="overlabel">
      <span class="required">Enter a big long number
    </div>
    <div class="underfield">
      <input type="text"
        id="text1101" name="text1101"
        value=""
        setid="set1101"
        label="Enter a big long number..."
        validate="number"
        required="yes"
        maxlength="120"
        tabindex="111"
        onfocus="changeBG(this.setid);"
        onblur="blurs(this);" />
    </div>
  </div>
</div>
```

Following is the basic construction of each different form element, along with a line-by-line description of the information that should be contained.

Input Text fields

An Input Text field is a single-line field that allows entry of text. Here is an example:



When to use:

Use an input text field when you need to offer a small number of characters to be available, such as allowing entry of a name, address or phone number.

When to avoid:

Avoid input text fields if a great number of characters are required, or multi-line input is necessary. Instead, use a textarea field.

This formset example has the following construction:

```
1. <div id="set0302" name="set0302" class="formset">
2.     <div class="leftlabel">
3.         Title
4.     </div>
5.     <div class="rightfield">
6.         <input type="text"
7.             id="text0302" name="text0302"
8.             value=""
9.             setid="set0302"
10.            label="Title"
11.            validate="name"
12.            required="no"
13.            maxlength="30"
14.            tabindex="32"
15.            onfocus="changeBG(this.setid);"
16.            onblur="blurs(this);" />
17.     </div>
18. </div>
```

Here is a line-by-line description of the attribute contents of this formset:

1. The id and name attributes must be the same. The class of "formset" is required. If it is the second or following formset on the same line, the class attribute should be "formset noclr".
2. The label DIV must be "leftlabel" if the label is to be placed to the left of the field or "overlabel" if the label is to be placed above the field.
3. This line contains the text that represents the label name. If the associated field is required, the text must be contained within a span that turns the color of the label red. In this case, line 3 should look like this:
`Title`
4. End the label DIV.
5. The field DIV class is dependent on the label DIV class. If the label DIV class is "leftlabel," the associated field DIV class must be "rightfield." If the label DIV class is "overlabel," the associated field DIV class must be "underfield."
6. All Input Text fields have a type attribute of "text."
7. The id and name attributes must be the same.
8. The value attribute represents any text that appears in the field when the page is first initiated. It is not required.
9. The setid attribute cross-references the containing formset DIV and allows the program to highlight the formset when there is an error. It must be the same value as the containing formset DIV id.
10. The label attribute represents how the field will be identified in the error message. If the identifying words in the accompanying label field (line 3) are short, simply repeat those words here. If the identifying phrase in the label field is long, shorten the phrase here to 3-5 words.
11. The validate attribute identifies the type of validation you want to perform on the field. For example, "name" means that you want the field to contain alphanumeric text. Other types of Input Text validation are: number (must contain only numeric characters), email (must be a proper email address), phone (must contain 10 numerals, with no additional punctuation),

- ph_structured (must contain a phone number, but allows certain punctuation, including parentheses and dashes), ph_areacode (allows only a three-digit area code number), ph_exchange (allows only a three-digit exchange number) and ph_number (allows only a four-digit phone number).
12. The required attribute indicates whether validation is required for the field. If validation is required, you must have a required value of "yes" and insert the required class span in the associated label DIV (see line 3).
 13. The maxlength attribute is required if the data is to be placed in a SQL-style database. Regardless of whether the field is required, each Input Text field and Textarea field will be checked for length and an error will be displayed if the number of characters in the field exceeds the maxlength value.
 14. The tabindex attribute is required to provide a smooth flow through the form for users who use the tab key to navigate the form. It is suggested to avoid applying numbers to this value until the form is otherwise completely developed.
 15. The onfocus attribute eliminates the error highlighting once the user click the field. This will be the same for each Input Text field.
 16. The onblur attribute shown here solves a problem that occurs when using Firefox. It should be included in every Input Text field.
 17. End the field DIV.
 18. End the formset DIV.

Checkbox fields

Checkboxes are a set of one or more Input Checkbox fields that let the user choose from among one or more options.

Here is an example:



Pick 1 or 2:

- Web Programming
- Web Administration
- This is a longer field than earlier
- Magazine Editing

When to use:

Use checkboxes when:

- Only a single option is given, such as in the case of "Check this box to indicate your acceptance of this license agreement."
- Multiple choices are OK or necessary, such as in the case of "Check one or more boxes..."

When to avoid:

Avoid checkboxes when one and only one selection is required. In this case, use a radio button or a select drop-down.

This formset example has the following construction:

```
1. <div id="set0801" name="set0801" class="formset">
2.     <div class="leftlabel">
3.         <span class="required">Pick 1 or 2:</span>
4.     </div>
5.     <div class="rightfield">
6.         <label for="check0801">
7.             <input type="checkbox"
8.                 id="check0801" name="check0801"
9.                 value="webprogramming"
10.                setid="set0801"
11.                label="Pick 1 or 2"
12.                required="yes">
```

```
13.         min="1" max="2"
14.         tabindex="81"
15.         onfocus="changeBG(this.setid);" />
16.         Web Programming
17.     </label><br />
18.     <label for="check0802">
19.         <input type="checkbox"
20.             id="check0802" name="check0801"
                value="webadministration"
                setid="set0801"
                label="Pick 1 or 2"
                required="yes"
                min="1" max="2"
                tabindex="82"
                onfocus="changeBG(this.setid);" />
                Web Administration
    </label><br />
    <label for="check0803">
        <input type="checkbox"
            id="check0803" name="check0801"
            value="newswriting"
            setid="set0801"
            label="Pick 1 or 2"
            required="yes"
            min="1" max="2"
            tabindex="83"
            onfocus="changeBG(this.setid);" />
            News Writing
    </label><br />
    <label for="check0804">
        <input type="checkbox"
            id="check0804" name="check0801"
            value="editingmagazine"
            setid="set0801"
            label="Pick 1 or 2"
            required="yes"
            min="1" max="2"
            tabindex="84"
            onfocus="changeBG(this.setid);" />
            Magazine Editing
```

```
        </label><br />
    </div>
</div>
```

Here is a line-by-line description of the attribute contents of this formset:

1. The id and name attributes of the formset DIV must be the same. The class of "formset" is required. If it is the second or following formset on the same line, the class attribute should be "formset noclr".
2. The label DIV must be "leftlabel" if the label is to be placed to the left of the field or "overlabel" if the label is to be placed above the field.
3. This line contains the text that represents the label name. If the associated field is required, the text must be contained within a span that turns the color of the label red, as shown here.
4. End the label DIV.
5. The field DIV class is dependent on the label DIV class. If the label DIV class is "leftlabel," the associated field DIV class must be "rightfield." If the label DIV class is "overlabel," the associated field DIV class must be "underfield."
6. By containing the input element inside a label tag, the user can click on either the checkbox or its associated label to select or deselect it.
7. All Checkbox input fields have a type attribute of "checkbox."
8. The id attribute must be unique for each checkbox, but the name attribute should be the same for each checkbox in the set.
9. The value attribute represents the value that will be passed along in the next step of processing, such as passing along the data to a database.
10. The setid attribute cross-references the containing formset DIV and allows the program to highlight the formset when there is an error. It must be the same value as the containing formset DIV id.

11. The label attribute represents how the field will be identified in the error message. If the identifying words in the accompanying label field (line 3) are short, simply repeat those words here. If the identifying phrase in the label field is long, shorten the phrase here to 3-5 words.
12. The required attribute indicates whether validation is required for the field. If validation is required, you must have a required value of "yes" and insert the required class span in the associated label DIV (see line 3).
13. The min and max attributes indicate the least number of checkboxes that must be checked (min) and the most number of checkboxes that can be checked (max).
14. The tabindex attribute is required to provide a smooth flow through the form for users who use the tab key to navigate the form. It is suggested to avoid applying numbers to this value until the form is otherwise completely developed.
15. The onfocus attribute eliminates the error highlighting once the user click the field. This will be the same for each Input Text field.
16. The visible label associated with the individual checkbox.
17. End the label tag. If the next checkbox is to fall below this checkbox, add a break tag.
18. Additional checkboxes in this set follow with the same construction as that which is given to lines 6-17. Note that in line 20, the id attribute is given a unique value, but the name attribute is the same as the name attribute in line 8.

Of course, if any checkboxes need to be preselected at the time the page is first displayed, just add the attribute:

```
selected="selected"
```

Radio button fields

Radio buttons are similar to checkboxes, but allow only one selection to be picked.

Here is an example:

Pick just one from all of the radio buttons listed below this very long label:

- Web Programming
- Web Administration
- This is a much much longer field than above
- Magazine Editing

When to use:

Use radio buttons when one and only one selection is required or permitted.

When to avoid:

Avoid radio buttons if space limitations make radio button usage unwieldy, such as the case when selecting one option out of dozens. Use a select drop-down instead.

This formset example has the following construction:

```
1.     <div id="set1001" name="set1001" class="formset">
2.         <div class="overlabel">
3.             <span class="required">Pick one:</span>
4.         </div>
5.         <div class="underfield">
6.             <label for="radio1001">
7.                 <input type="radio"
8.                     id="radio1001" name="radio1001"
9.                     value="webprogramming"
10.                    setid="set1001"
11.                    label="Pick one"
12.                    required="yes"
13.                    tabindex="101"
14.                    onfocus="changeBG(this.setid);" />
15.                 Web Programming
```

```

16.         </label><br />
17.         <label for="radio1002">
18.             <input type="radio"
19.                 id="radio1002" name="radio1001"
                    value="webadministration"
                    setid="set1001"
                    label="Pick one"
                    required="yes"
                    tabindex="102"
                    onfocus="changeBG(this.setid);" />
                    Web Administration
        </label><br />
        <label for="radio1003">
            <input type="radio"
                id="radio1003" name="radio1001"
                value="newswriting"
                setid="set1001"
                label="Pick one"
                required="yes"
                tabindex="103"
                onfocus="changeBG(this.setid);" />
                News Writing
        </label><br />
        <label for="radio1004">
            <input type="radio"
                id="radio1004" name="radio1001"
                value="editingmagazine"
                setid="set1001"
                label="Pick one"
                required="yes"
                tabindex="104"
                onfocus="changeBG(this.setid);" />
                Magazine Editing
        </label><br />
    </div>
</div>
</div>
</div>

```

Here is a line-by-line description of the attribute contents of this formset:

1. The id and name attributes of the formset DIV must be the same. The class of "formset" is required. If it is the second or following formset on the same line, the class attribute should be "formset noclr".
2. The label DIV must be "leftlabel" if the label is to be placed to the left of the field or "overlabel" if the label is to be placed above the field.
3. This line contains the text that represents the label name. If the associated field is required, the text must be contained within a span that turns the color of the label red, as shown here.
4. End the label DIV.
5. The field DIV class is dependent on the label DIV class. If the label DIV class is "leftlabel," the associated field DIV class must be "rightfield." If the label DIV class is "overlabel," the associated field DIV class must be "underfield."
6. By containing the input element inside a label tag, the user can click on either the radio button or its associated label to select or deselect it.
7. All radio button input fields have a type attribute of "radio."
8. The id attribute must be unique for each radio button, but the name attribute should be the same for each radio button in the set.
9. The value attribute represents the value that will be passed along in the next step of processing, such as passing along the data to a database.
10. The setid attribute cross-references the containing formset DIV and allows the program to highlight the formset when there is an error. It must be the same value as the containing formset DIV id.
11. The label attribute represents how the field will be identified in the error message. If the identifying words in the accompanying label field (line 3) are short, simply repeat those words here. If the identifying phrase in the label field is long, shorten the phrase here to 3-5 words.

12. The required attribute indicates whether validation is required for the field. If validation is required, you must have a required value of "yes" and insert the required class span in the associated label DIV (see line 3).
13. The tabindex attribute is required to provide a smooth flow through the form for users who use the tab key to navigate the form. It is suggested to avoid applying numbers to this value until the form is otherwise completely developed.
14. The onfocus attribute eliminates the error highlighting once the user click the field. This will be the same for each Input Text field.
15. The visible label associated with the individual checkbox.
16. End the label tag. If the next checkbox is to fall below this checkbox, add a break tag.
17. Additional checkboxes in this set follow with the same construction as that which is given to lines 6-16. Note that in line 19, the id attribute is given a unique value, but the name attribute is the same as the name attribute in line 8.

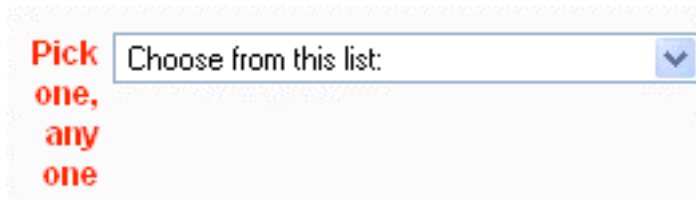
Of course, if any radio button needs to be preselected at the time the page is first displayed, just add the following attribute to that radio button:

```
selected="selected"
```


Select drop-downs

Select drop-downs are similar to radio buttons in that they allow only one selection of a range of options.

Here is an example:



When to use:

Use select drop-downs:

- when space is limited or when the range of options numbers in the dozens, such as selecting a U.S. state of residence.
- when you need to construct a list of options interactively at the time the page is constructed or as a result of other select options being chosen.
- when using "Other" and tying it to a corresponding input text field is required (see instructions below).

When to avoid:

Avoid select drop-downs if you feel the end-user may not be competent enough to click and select a drop-down option, such as young children, elderly or disabled persons.

Also, note that the select object allows for the creation of multi-line boxes that allow multiple choices to be made. Avoid these whenever possible. The method of multiple selection differs between Windows and the Macintosh, causing customer assistance headaches. Use checkboxes instead.

This formset example has the following construction:

```
1. <div id="set0701" name="set0701" class="formset">
2.     <div class="leftlabel">
3.         <span class="required">Pick one, any one</span>
4.     </div>
5.     <div class="rightfield">
```

```

6.         <select
7.             id="select0701" name="select0701"
8.             size="1"
9.             setid="set0701"
10.            label="Pick one, any one"
11.            required="yes"
12.            tabindex="71"
13.            onfocus="changeBG(this.setid);">
14.                <option value="">Choose one:</option>
15.                <option value="1">One</option>
16.                <option value="2">Two</option>
17.                <option value="3">Three</option>
18.                <option value="set0702">Other</option>
19.            </select>
20.        </div>
21. </div>

```

Here is a line-by-line description of the attribute contents of this formset:

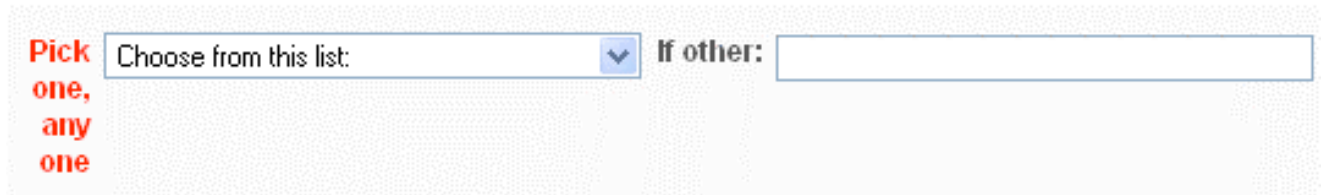
1. The id and name attributes of the formset DIV must be the same. The class of "formset" is required. If it is the second or following formset on the same line, the class attribute should be "formset noclr".
2. The label DIV must be "leftlabel" if the label is to be placed to the left of the field or "overlabel" if the label is to be placed above the field.
3. This line contains the text that represents the label name. If the associated field is required, the text must be contained within a span that turns the color of the label red, as shown here.
4. End the label DIV.
5. The field DIV class is dependent on the label DIV class. If the label DIV class is "leftlabel," the associated field DIV class must be "rightfield." If the label DIV class is "overlabel," the associated field DIV class must be "underfield."
6. Start the SELECT tag.

7. The id attribute must be unique for each radio button, but the name attribute should be the same for each radio button in the set.
8. To make it a drop-down, it must have a size of 1. Avoid multi-line select boxes.
9. The setid attribute cross-references the containing formset DIV and allows the program to highlight the formset when there is an error. It must be the same value as the containing formset DIV id.
10. The label attribute represents how the field will be identified in the error message. If the identifying words in the accompanying label field (line 3) are short, simply repeat those words here. If the identifying phrase in the label field is long, shorten the phrase here to 3-5 words.
11. The required attribute indicates whether validation is required for the field. If validation is required, you must have a required value of "yes" and insert the required class span in the associated label DIV (see line 3).
12. The tabindex attribute is required to provide a smooth flow through the form for users who use the tab key to navigate the form. It is suggested to avoid applying numbers to this value until the form is otherwise completely developed.
13. The onfocus attribute eliminates the error highlighting once the user click the field. This will be the same for each Input Text field.
14. The first option should have a blank value and should offer some type of indication of what kind of information appears in the list, such as "Choose a state." Avoid displaying only a list without a label -- this results in validation errors (such as in the case of typing "Los Angeles" in a city field but leaving "Alabama" unchanged as the initial state in the state select drop-down).
15. The second and following options should have a unique value.

Tying an Input Text field to an "Other" Selection

Often it will be necessary to include an "Other" selection in a select drop-down in the case when choosing "Other" requires an explanation in a corresponding input text field.

Here's an example:



The screenshot shows a form with a dropdown menu on the left and a text input field on the right. The dropdown menu is labeled 'Choose from this list:' and has a blue arrow pointing down. To the right of the dropdown is the text 'If other:' followed by a text input field. On the far left, the text 'Pick one, any one' is displayed in red.

To accommodate this, note that line 18 of the select drop-down example above displays a value of "set0702". This corresponds to the accompanying input text field formset:

```
22. <div id="set0702" name="set0702" class="formset noclr">
23.     <div class="leftlabel">
24.         If other:
25.     </div>
26.     <div class="rightfield">
27.         <input type="text"
28.             id="text0702" name="text0702"
29.             value=""
30.             setid="set0702"
31.             label="Pick one, any one/Other"
32.             validate="name"
33.             required="if-other"
34.             maxlength="60"
35.             tabindex="72"
36.             onfocus="changeBG(this.setid);" />
37.     </div>
38. </div>
```

The following things are required to tie an "Other" selection to a corresponding text input field:

- The option for the select drop-down must be called "Other". No derivations are allowed. (See sample, line 18.)
- The "Other" option must have the same value as the id for the corresponding input text formset. (Compare line 18 with line 22.)
- The required attribute of the input text formset must be "if-other". (See sample, line 33.)

Thus, if "Other" is selected in the drop-down, an error will occur if the "If other" text field is empty. Likewise, if "Other" is not selected, an error will occur if the "If other" text field contains any characters.

Note that, at this time, no other validation is performed on the "Other" text input field, such as checking for only numeric characters. This might change in the future.

Textarea fields

Textarea fields are similar to text input fields, but allow for the easy inclusion of a great deal of data.

Here's an example:



When to use:

Use textarea fields when the inclusion of a large amount of data is necessary, as well as when the data needs to include linebreaks.

When to avoid:

Avoid textarea fields when the data requirements can be accommodated in a text input field.

This formset example has the following construction:

```
1. <div id="set0401" name="set0401" class="formset">
2.     <div class="leftlabel">
3.         <span class="required">Describe it</span>
4.     </div>
5.     <div class="rightfield">
6.         <textarea
7.             id="text0401" name="text0401"
8.             rows="4"
9.             setid="set0401"
10.            label="Describe it"
11.            required="yes"
12.            maxlength="200"
13.            tabindex="41"
14.            onfocus="changeBG(this.setid);"></textarea>
15.     </div>
16. </div>
```

1. The id and name attributes must be the same. The class of "formset" is required. If it is the second or following formset on the same line, the class attribute should be "formset noclr".
2. The label DIV must be "leftlabel" if the label is to be placed to the left of the field or "overlabel" if the label is to be placed above the field.
3. This line contains the text that represents the label name. If the associated field is required, the text must be contained within a span that turns the color of the label red.
4. End the label DIV.
5. The field DIV class is dependent on the label DIV class. If the label DIV class is "leftlabel," the associated field DIV class must be "rightfield." If the label DIV class is "overlabel," the associated field DIV class must be "underfield."
6. Start the TEXTAREA tag.
7. The id and name attributes must be unique, but the same as each other.
8. The number of rows (height) must be included, but cols (width) may be omitted, since the width of textarea fields are generated when the page is constructed.
9. The setid attribute cross-references the containing formset DIV and allows the program to highlight the formset when there is an error. It must be the same value as the containing formset DIV id.
10. The label attribute represents how the field will be identified in the error message. If the identifying words in the accompanying label field (line 3) are short, simply repeat those words here. If the identifying phrase in the label field is long, shorten the phrase here to 3-5 words.
11. The required attribute indicates whether validation is required for the field. If validation is required, you must have a required value of "yes" and insert the required class SPAN in the associated label DIV (see line 3).

12. Although maxlength is not normally an attribute of textareas, it is used here to restrict the data to be placed in a SQL-style database. Regardless of whether the field is required, each textarea field will be checked for length and an error will be displayed if the number of characters in the field exceeds the maxlength value.
13. The tabindex attribute is required to provide a smooth flow through the form for users who use the tab key to navigate the form. It is suggested to avoid applying numbers to this value until the form is otherwise completely developed.
14. The onfocus attribute eliminates the error highlighting once the user click the field. This will be the same for each Input Text field. Note that the ending TEXTAREA tag immediately follows -- this is to avoid putting errant spaces in the field unknowingly.