LESSON 2

Creating Forms

OBJECTIVES

You will be able to:

- describe HTML forms and their function.
- identify methods for processing a form.
- create a HTML form.
- identify how to use the Forms Wizard to create a HTML form.
- retrieve information collected on a form.

This lesson provides you with information on HTML forms. You will learn what an HTML form is and how they are used on the Web. You will then learn how to create forms using the forms toolbar, templates and the Forms Wizard. Finally, you will retrieve information collected on a form.

What is an HTML Form?

Up until now, you have learned how to create HTML documents to present information to the user. HTML also offers a mechanism called a "form" that allows the user to send information by interactively entering data into an HTML page. The user-entered data is then submitted to the server where it is stored or processed. In this lesson, you will learn how to create forms and collect the data inputted by users.

HTML forms may contain several elements, including:

- Text entry fields
- Radio buttons
- Checkboxes
- List boxes and Scrolling lists
- Submit and Reset buttons

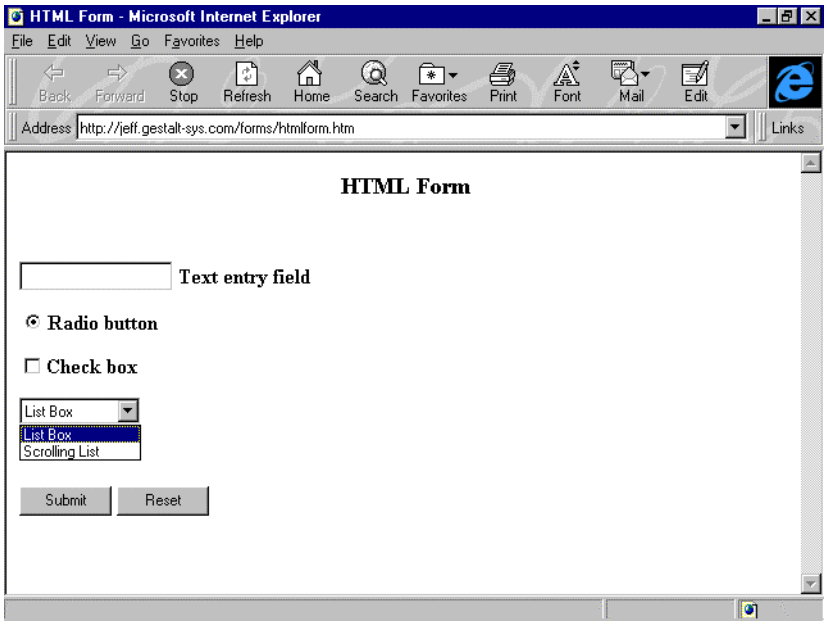

Figure 2-1: HTML Form Elements

These user interface elements allow customized forms to be constructed. FrontPage also comes with a form Wizard and form templates.

Processing a Form

The data input into a form by the user may be processed by a *Common Gateway Interface (CGI)* script residing on the server. Those scripts can be quite complicated to construct. Fortunately, FrontPage includes FrontPage Components that create the CGI script for you. If you have experience with CGI, however, you can use CGI scripts with FrontPage. We will explore this method in more depth later in this lesson.

When form data is processed, the data will appear as a string, similar to the example below:

FirstName=Jane&LastName=Student&Email=JStudent@abc.com

Although the message may appear somewhat cryptic at first glance, it is quite simple to use a search and replace function to translate the data to a more useable form. The first task for a CGI script is to *parse* this information. The data might be translated to appear as the sample below:

FirstName: Jane

LastName: Student

Email: JStudent@abc.com

Form Tag

The FORM tag is a closed tag used within the body of an HTML document. It has two attributes associated with it.

Attribute	***Description***
METHOD	Either GET or POST. GET sends the form data as part of the URL in the header of the transmission. POST sends the form data as part of the body of the transmission. Choice is determined by the requirements of the recipient.
ACTION	Specifies where the information is to be sent when the user submits the form.

Figure 2-2: Form Attributes

The ACTION will refer to the URL where the data is to be submitted. This is normally a FrontPage Component/CGI script.

<FORM METHOD=POST ACTION= "http://www.form.com/cgi/response">

A form field is any area of the form where the user is able to input data. Some of these fields require the user to type in information, other fields allow the user to make selections from a list of alternatives.

There are several attributes that can be associated with a form field. Most of these can be set in the Form Field Properties dialogue box (except, of course, the type, which is established when you click on one of the form field buttons).

Attribute	*Description*
TYPE (required)	Indicates the display of the entry field. Possible values are TEXT, RADIO, CHECKBOX, SUBMIT and RESET.
NAME (required for some types)	The name for the data field. Required for text fields, radio buttons and checkbox elements. If the returned data is to be submitted to a script or entered into a database, it may be necessary to use a name corresponding to that script or database.
SIZE (optional)	Specifies the displayed width of a text field, specified as a number of characters.
VALUE (required for some types)	Indicates the content of the selection submitted by the user. Required for radio buttons and checkboxes, optional for other elements.

Figure 2-3: Field Attributes

Here is an example of an HTML form that includes a text box:

```
<FORM METHOD=POST ACTION= “http://www.form.com/cgi/response”>
 <P>
<INPUT TYPE=TEXT SIZE=27 NAME=textbox1>
</P>
</FORM>
```

Creating Forms with FrontPage

To begin a form you simply click on one of the form field buttons or you can choose **Insert** then **Form** and **Form**. This will insert the beginning and ending form tags, indicated by a dotted box surrounding the form boundary. Getting to the Form toolbar is different from many of the other toolbars. When you choose Insert and Form, you can make the menu that appears into a toolbar by dragging it into the workspace. Here is a quick look at the toolbar.

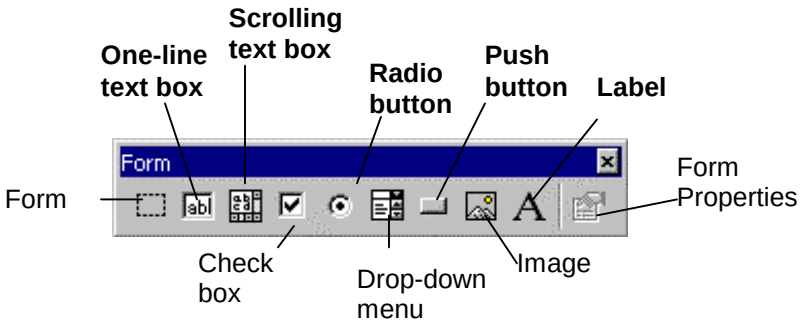

Figure 2-4: Forms Toolbar

Every time you begin a form, the Submit and Reset buttons are automatically inserted. This is to prevent a form from being created that can not be submitted. There is much more information about those buttons later in the lesson.

Form Field Properties

For each form field there are a set of properties that can be accessed by double-clicking the form field or right-clicking and choosing **Form Field Properties**. The properties determine how a field will look to the user and how it will operate once the form is posted. The exercises contain specific information on each field's properties.

Validation

Except for the Check Box, each one of the form fields can be further modified by what FrontPage calls *validation*. This is essentially a filter for the amount and types of information the form will collect. It restricts the answers to form fields so that unwanted data is not passed on to the server and so that all required fields in the form contain data. You may have run across this on the Web if you have ever tried to fill out a form where the name was required. If you do not fill in the name, you may receive a message saying, "Sorry, you must fill in your name to complete your submission." Because each form field serves a different purpose, there are different validation parameters you can set for each field.

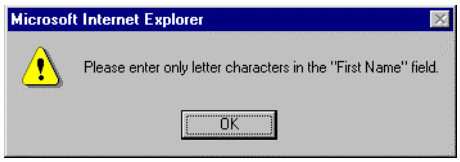

Figure 2-5: Form Error Message

Validation helps to reduce the amount of work the server must perform. By cutting down the types of answers at the client level, the server will not have to sift through as much data. This system is not completely foolproof, but should assist the server. There are occasions when validation will not work. When you choose a validation option, FrontPage inserts the Validation FrontPage Component. That Component creates a script in JavaScript that handles the validation. In order for the Component to work on various Web servers, you will need to have the FrontPage server extensions installed. For the JavaScript to work, the user's browser must be Java-enabled. The latest browsers are Java-enabled, but many older browsers are not.

One-Line Text Box

The One-Line Text Box is probably the most basic form field. The user clicks in the box and types in information. This box is normally used for contact information, such as name and address. Once the box is inserted you will want to name the box. When the form is posted, the name appears before the information the user typed in. For example, if you name the box "Address", the posted data would read something like: Address=123 Main St. By default, FrontPage inserts "T1" (Text box 1) for the first box you create, "T2" for the second and so on. Although you could leave those as the name of the box, it certainly isn't descriptive and could cause confusion when trying to decipher the posted data.

You are also able to set an initial value for the box. This is text that you place in the box to prompt a particular user response or otherwise assist the user in filling out that field quickly. The width of the box can be changed by a setting in the Form Field Properties dialogue box or selecting the box, then clicking and dragging the edges. The final adjustment you can make is whether the box is going to be used for typing in passwords. If you answer yes, the text the user types in will appear as asterisks or dots, so the password is not revealed to others.

Exercise 2-1: Inserting a One-Line Text Box

In this exercise, you will begin building a form by opening a Web containing what will become a survey and inserting several one-line text boxes.

1. In FrontPage, open the **survey** Web.
2. Open **mountain.htm** within the "survey" Web.
3. From the Insert menu, choose Form then drag the Form menu onto the workspace.

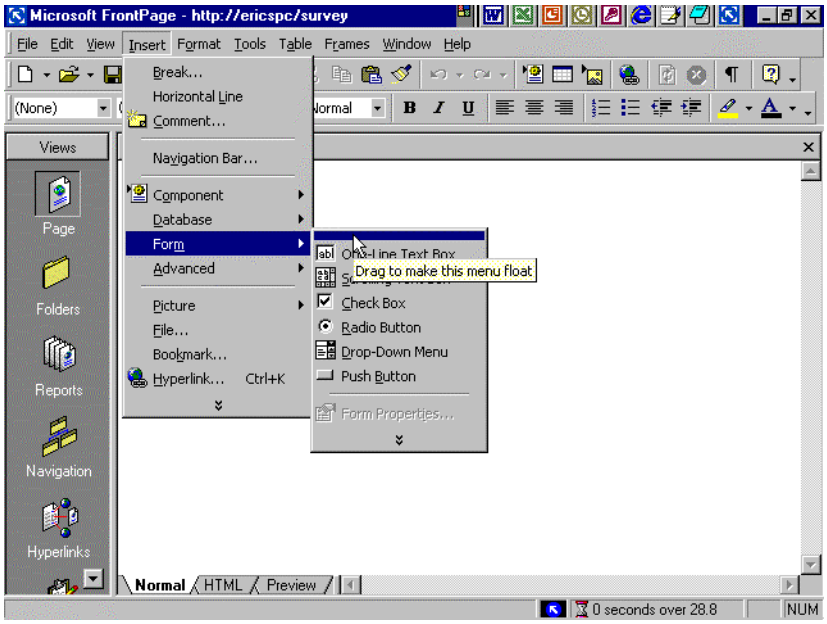

Figure 2-6: Making the Form Toolbar Float

You can drag it up with the other toolbars if you like.

4. Skip a few lines after the paragraph and click on the **One-Line Text Box** button. (See Figure 2-4)

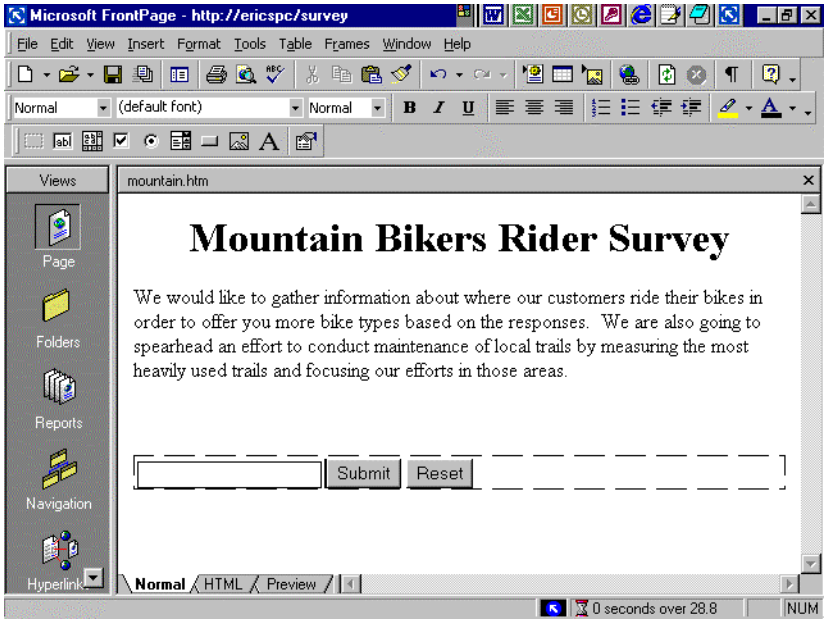

Figure 2-7: One-Line Text Box Inserted

Notice the dotted line indicating the form boundary and that it inserted the "Submit" and "Reset" buttons.

4. Insert the cursor before the "Submit" button and press **<Enter>** several times to make room for the other fields you will be inserting.

5. Click before the text box and press **<Shift>-<Enter>** to create a line above the box.

6. Insert the cursor at the beginning of the line and type **Name:**

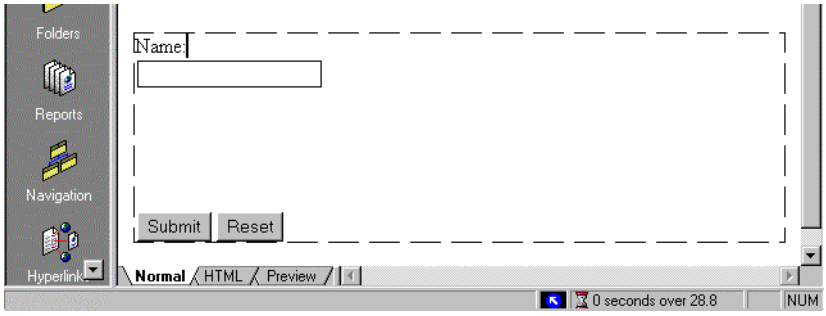

Figure 2-8: Text Inserted with Box

7. Click after the box and press **<Enter>**

8. Type **Address**:, then press <**Shift>-<Enter>** and click on the **One-Line Text Box** button.

9. Repeat steps 7 and 8 for these other items: **City:**, **State:** and **Zip:**

10. Right-click on the box underneath **Name:** and choose **Form Field Properties…**

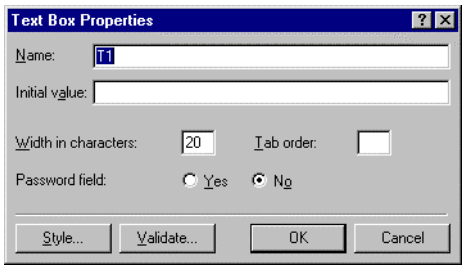

Figure 2-9: Form Field Properties Dialogue Box

*The "Tab order" field allows you to specify the order the user will travel down the form as they keep pressing Tab to get to the next field. It appears on all of the field properties dialogue boxes. Tab order will only work in Internet Explorer 4.0 and is not necessary as the browser will simply take users from top to bottom as they press Tab. Making it a password field will turn any characters the user types into *.*

11. In the **Name:** field, type **Name**, then click **OK**.

12. Repeat steps 10 and 11 for the other boxes, naming them per the text that appears above each box as shown in Figure 2-10.

13. Right-click on the box below **Name:**, choose **Form Field Properties…** and change **Width in characters:** to **32**, then click **OK**.

Next, you will use the other method of sizing the text box.

14. Select the **Address** box and click and drag the right edge to the right, making the width approximately 55 characters.

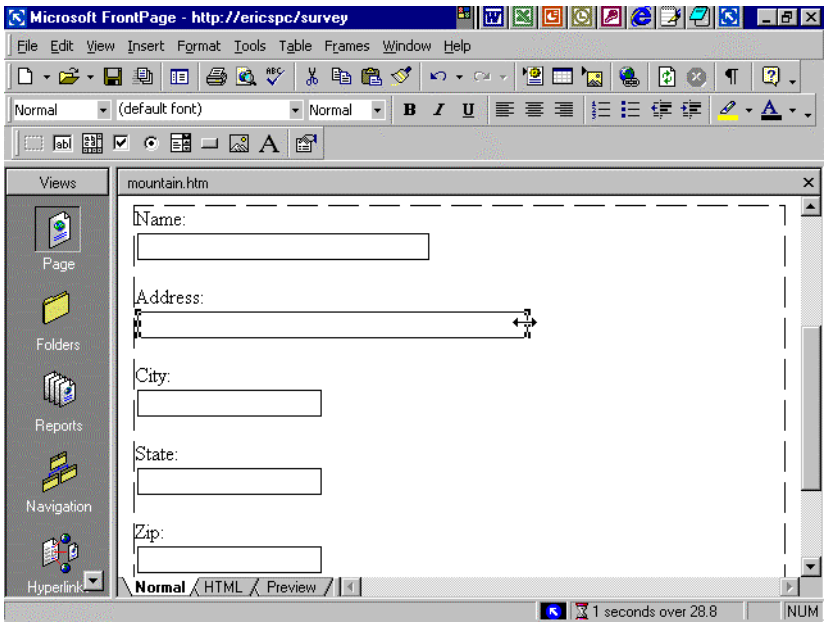

Figure 2-10: Resizing Text Box

15. Make the “City” box bigger and the “State” and “Zip” boxes smaller.

16. Choose the Form Field Properties for the State box and in the **Initial value:** field type **ST**, then click **OK**.

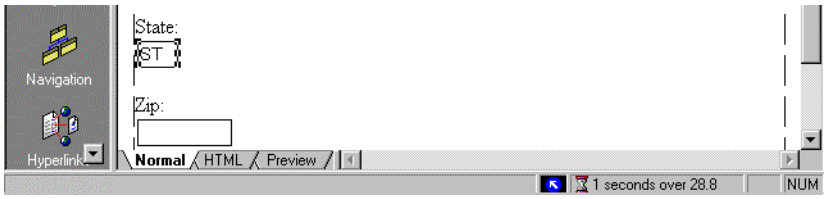

Figure 2-11: Initial Value

This is meant to prompt users to use the two letter state abbreviation.

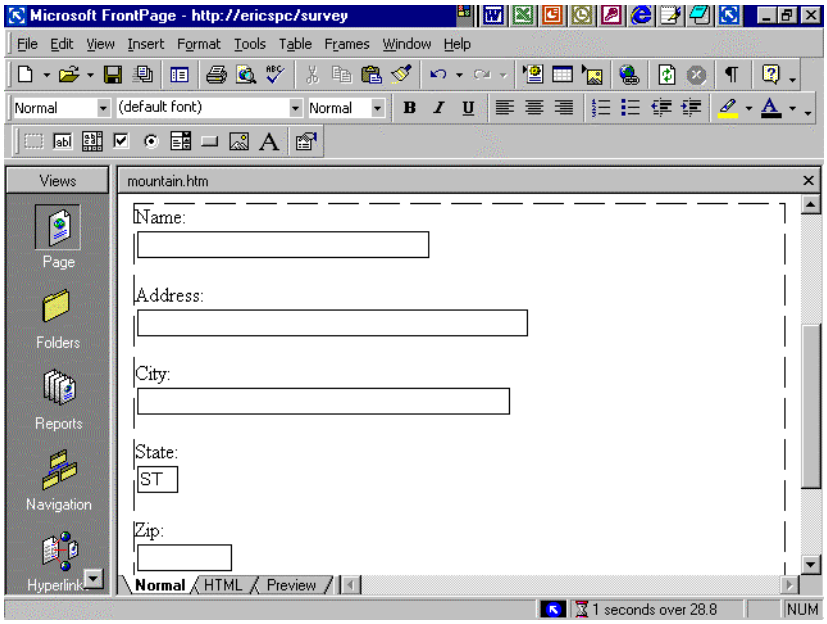

Figure 2-12: Form with Text Boxes

Web Page Design Tip

As shown in the opening paragraph of the page used in this exercise (See Figure 2-7), try to explain what the purpose of the form is and what is in it for the user to fill it out. If users do not have a reason to fill out a form, they won't.

Exercise 2-2: Validating the One-Line Text Box

In this exercise, you will set the validation parameters for a One-Line Text Box you created in the previous exercise.

1. Right-click on the **State** box and choose **Form Field Validation…**

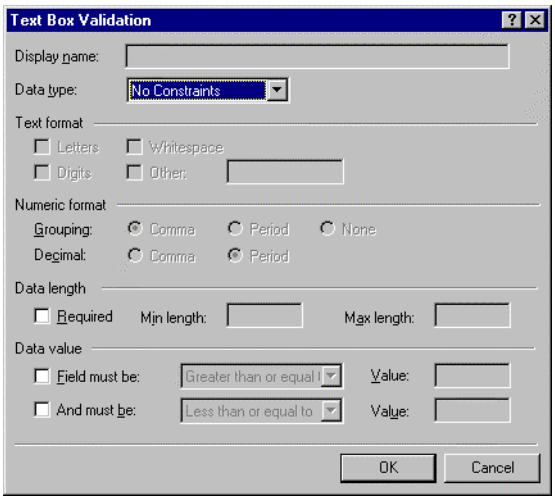

Figure 2-13: Form Field Validation Dialogue Box

You can also reach this dialogue box by clicking on the Validate… button in the Form Field Properties for the currently selected field.

2. In the **Data Type:** drop down box, select **Text**

3. Click on **Letters** for the Text Format.

   This says that the user can only type in letters, not numbers and that whitespace is also not allowed (e.g., the user can not type in "O H", only "OH").

4. Click on **Required** for "Data Length" and type in a **Min Length:** of **0** and **Max Length:** of **2**, then click **OK**.

   If you type in 2 for the minimum length, then the user must enter at least two characters in the field for the form to be processed.

5. Choose Form Field Validation for the Zip box and set the following parameters:

   - **Data Type: Number**
   - **Min Length: 0**
   - **Max Length: 5**

   There is a place on the validation dialogue box to input a "Display Name," which is how the field will be called in the warning message. By default, the name you assign a field in the Form Field Properties will appear.

Sometimes, though, the name you give the field may make sense for data collection purposes, but may be confusing for the user in a warning message. Thus, the "Display Name" substitutes for the form field name.

6. Click on the **Preview** view.

7. Click in the State field and try to type in more than two letters, then try to type more than five numbers into the Zip field.

   You may hear an error beep. By the way, you will be able to enter in numbers in the State field despite having restricted that field to letters. The field would not be validated until you tried to post the form.

Scrolling Text Box

The Scrolling Text Box is very similar to the One-Line Text Box, except the user can type in multiple lines of text. This form field is often used for questions that require a more narrative response, such as "What other comments do you have on our service?"

As with the other fields, you should name this field. The default is S1, for Scrolling Text Box 1. You are also able to set the number of lines (rows) and width (columns) of the text box by clicking and dragging the edges or entering amounts in the form field properties. The number of lines is not how many lines the user can type – that is set in the validation – it is how many lines appear on the screen (i.e., the height of the box). The validation options are the same as those for the One-Line Text Box.

Exercise 2-3: Inserting a Scrolling Text Box

In this exercise, you will insert a Scrolling Text Box form field.

1. Return to **Normal** view.

2. Below the Zip text box, type

   Any other background information you would like to tell us about?

   Make sure you are still within the form boundary. If you put the Scrolling Text box outside of the boundary, you are creating a completely separate form. All of the form fields that make up the form must appear within the same form boundary for the form to work.

3. Press **<Shift>-<Enter**>.

4. Click on the **Scrolling Text Box** button.

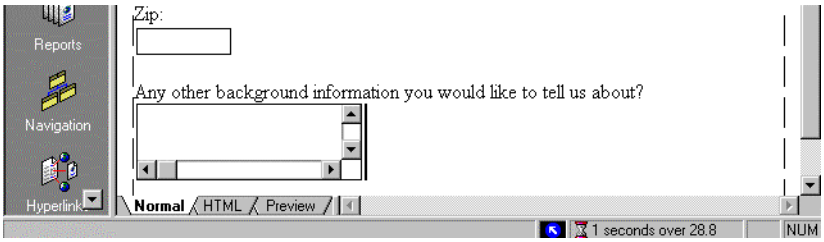

Figure 2-14: Scrolling Text Box

5. Double-click on the Scrolling Text Box to open the Properties dialogue box.

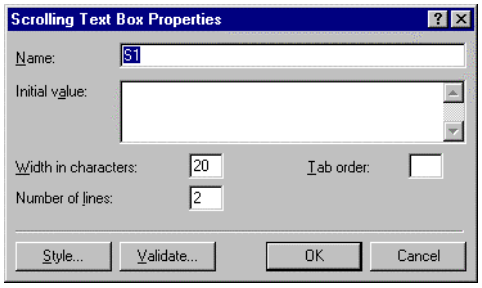

Figure 2-15: Scrolling Text Box Properties

6. In the **Name:** field, type **BackgroundInfo**, then click **OK**.

Web Page Design Tip

Although even a very small Scrolling Text Box can accept large amounts of text, users will normally rely on the size of the box as an indication of how much text you expect them to enter. If you want users to enter in a large amount of text, make the box larger.

Check Box

A check box lets a user make a choice, or several choices, by clicking on a box next to an option. If the box is empty, a check will appear in the box once the user selects an option. Use a check box for occasions when there is a list of choices that can have multiple selections or none. You can choose to have the box appear in the form with a check already in the box. The user can then keep the box checked or click on the box to uncheck it. There is no validation because the Check Box will only return either ON or NULL as the response. In the value field, the default is set for ON, which means that if you mark "Not checked" for the Initial State, when the box is checked by the user a response of ON will be returned. If the box is left unchecked a response of NULL is returned. The reverse is true if you mark "Checked" for the initial state.

You do not have to use the ON or NULL system. The other option is to name all of the checkboxes the same and change the value for each checkbox. This groups together the checkboxes under one question.

For example, if you asked users to check which products they are interested in, you could name every box “products” and give the first box a value of “product1”, the second a value of “product2” and so on.

Which method you choose will depend on personal preferences as well as how the software handling the results file normally accepts data. Using the products example again, should the user select products 1 and 3 the results from the ON/NULL method would look like: &product1=ON&product3=ON. Using the same name method, the results would look like: &products=product1&products=product3.

Exercise 2-4: Inserting a Check Box

In this exercise, you will create several Check Boxes and name each one.

1. Create an empty line after the Scrolling Text Box.
2. Below the Scrolling Text Box, type

   Which of the following bike types are you riding or have you recently ridden?

3. Press **<Enter>**.
4. Click on the **Check Box** button, then type **Mountain Bike**.

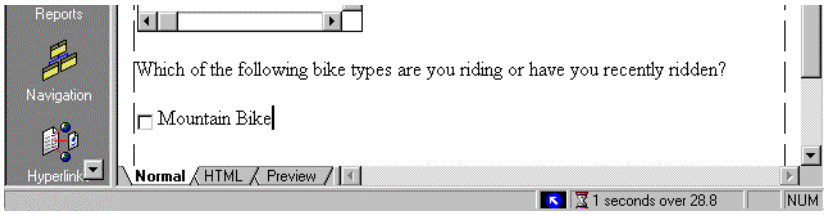

Figure 2-16: Check Box

5. Press Shift Enter after “Mountain Bike” and click on the **Check Box** button, then type **Ten Speed**.
6. Repeat step #5 for: **Hybrid**, **Tandem** and **Other**.
7. Double-click on the check box next to “Mountain Bike.”

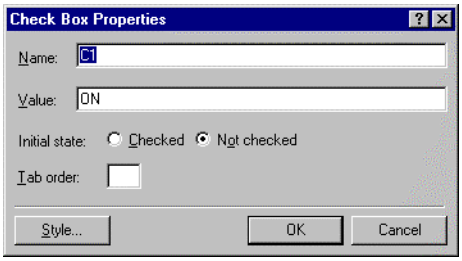

Figure 2-17: Check Box Properties

8. In the **Name**: field, type **MountainBike**, then click **OK**.

9. Repeat steps 7 and 8 for the other check boxes, naming each one the label that is beside it (e.g., name the box “Hybrid” that is next to the “Hybrid” text).

   The result of naming each box different will be that if Hybrid is checked by the user, it will appear as Hybrid=ON.

   You now decide that most of the users will probably be mountain bike riders, so you will check that option for them.

10. Double click on the **MountainBike** check box and click on the **Checked** option, then click **OK**.

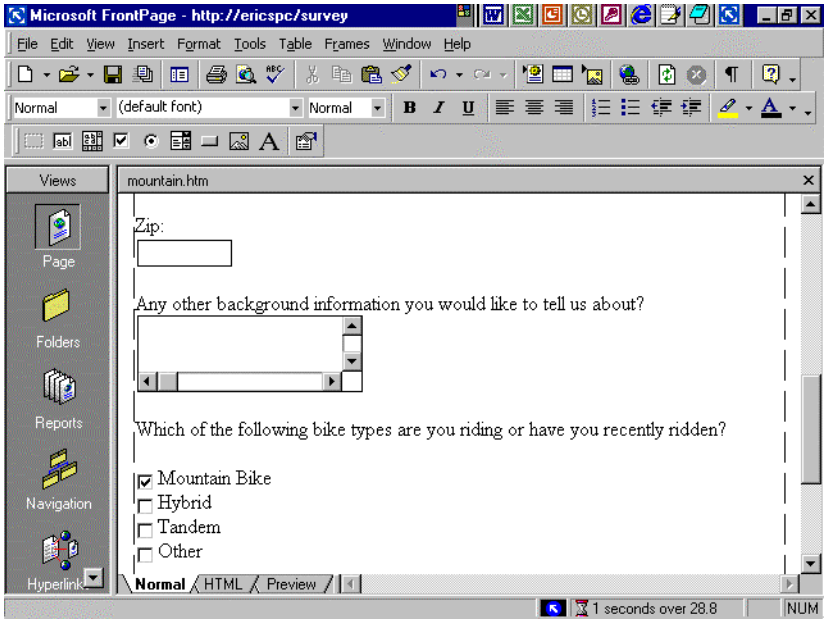

Figure 2-18: Check Boxes

Web Page Design Tip

You want to make your forms as painless as possible to complete. By having the most popular responses already checked, the user has less to fill out.

Radio Buttons

Radio buttons are open circles that appear filled-in when clicked on. They are used when the user must make a choice from a list of options, such as male or female. When the user clicks one radio button, the others become not selected. By default, FrontPage sets the first button you create as selected. You can choose to have it not selected, if you do not want to force users to answer the question.

The Name/Value pair for radio buttons is a little different than the other form fields. Because the radio buttons always work in tandem, they are given a group name and then each button is assigned a value. The only validation that is necessary for radio buttons is whether data is required in order for the form to be processed. Along with that is the option for entering in the descriptive name that will appear in any warning messages.

Exercise 2-5: Inserting Radio Buttons

In this exercise, you will insert several radio buttons and create group and value names.

1. Return to **Normal** view.
2. On the empty line after the check boxes, type **What is your age?** and press **<Enter>**.
3. Click on the **Radio Button** button (See Figure 2-4 for its location), then type **20 or Under** and press **<Shift>-<Enter>**.

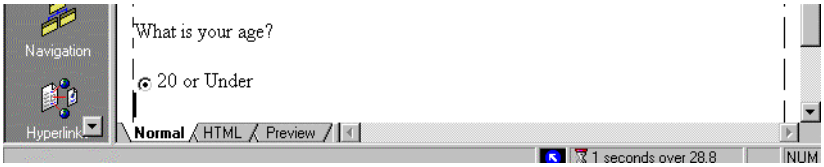

Figure 2-19: Radio Button

Notice that this first one is already selected.

4. Repeat step #3 for these other items:
   - **21 to 30**
   - **31 to 40**
   - **Over 40**
5. Double click on the “20 and Under” radio button.

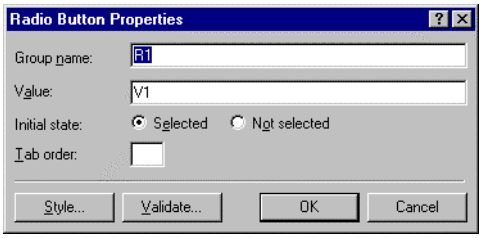

Figure 2-20: Radio Button Properties

6. In the **Group Name:** field, type **Age** and in the **Value**: field, type **0-20**, then click **OK**.

This will create a result of "Age=0-20" if this option is chosen.

7. Repeat step #6 for the other items, typing in the following values:

   - 21 to 30 = **21-30**
   - 31 to 40 = **31-40**
   - Over 40 = **40+**

   Be sure you type "Age" the same in every name field for the radio buttons. It is case sensitive. For example, if you type "Age" for two of the buttons and "age" for the others, there will be two groups of radio buttons that will not work in tandem.

8. Click on the **Preview** view and click on a few of the radio buttons to see how it only allows one option.

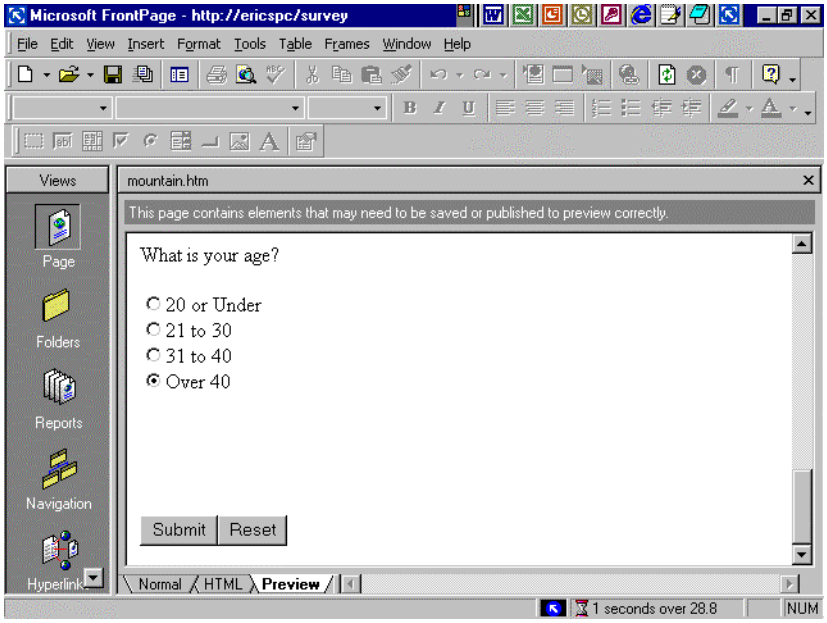

Figure 2-21: Radio Buttons in Preview

Drop-Down Menu

Creating a Drop-Down Menu form field is a little more complex than the other form fields. The user will see either a box with a down arrow that, when clicked on, reveals the choices or a box with the choices listed. There are options for the selection of one or multiple items. Drop-down menus make nice alternatives to check boxes and radio buttons because they save space. Instead of listing all of the items directly on the form, with the drop-down box you can have just one of the selections showing. Only when the user clicks on the box are the other choices revealed.

Creating a Drop-Down Menu is basically a two step process. First, you insert the box and open the Drop-Down Menu Properties box, where you enter the name for the field.

Second, in the properties dialogue box, you click the **Add…** button to view another dialogue box where you input the items that will appear in the menu. In that dialogue box, you create the list items and their value. For example, if the user choice is "Windows95 or Windows 3.1/3.0", you could specify the value to be "Win95/3.1/3.0" to shorten the response for your own purposes.

Once you have a list of items, you can modify or delete items and change the order in which the items appear. You can also adjust the height, which is a measure of how many items will appear in the box. Finally, you can choose whether to allow multiple selections (i.e., allowing the user to choose more than one item by holding down the Control/Command key while clicking selections).

For the validation, you can choose whether data must be entered and, if so, the minimum and maximum number of choices allowed. One other adjustment you can make is to disallow the first item on the list. This is often advantageous for creating a first item that reads something like "Click here to make selection." It is a direction for the user, not a choice; thus it is disallowed.

Exercise 2-6: Inserting a Drop-Down Menu

In this exercise, you will insert a Drop-Down Menu and make adjustments to how the menu interacts with the user.

1. Return to **Normal** view.
2. On the line after the radio buttons, type

   Which trail do you use most often?

3. Press **<Shift>-<Enter>**.
4. Click on the **Drop-Down Menu** button. (See Figure 2-4 for its location)

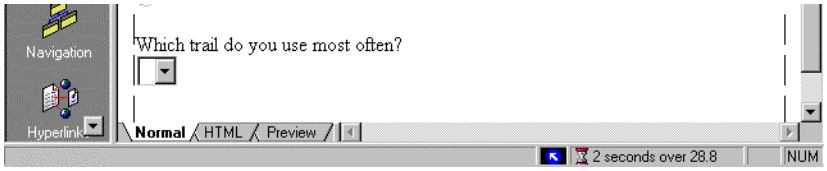

Figure 2-22: Drop-Down Menu

The drop-down menu is very small at this point because we have yet to create items for the list.

5. Double-click on the Drop-Down Menu field.

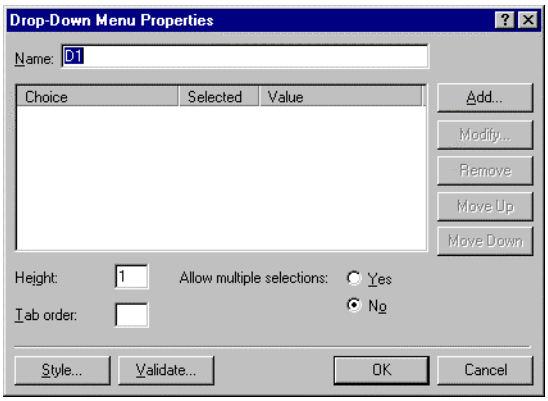

Figure 2-23: Drop-Down Menu Properties

6. In the **Name:** field, type **Trail**

7. Click on the **Add…** button.

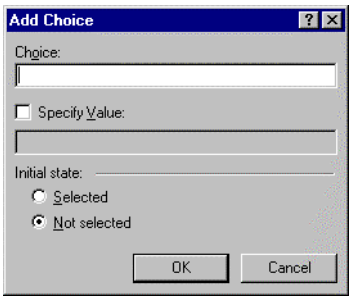

Figure 2-24: Add Choice Dialogue Box

8. In the **Choice:** field, type **Clear Water Creek**, then click **OK**.

9. Repeat steps 7 and 8 for these other choices:

   Hudson's Gulch

   Greenbriar

   Railroad Bridge

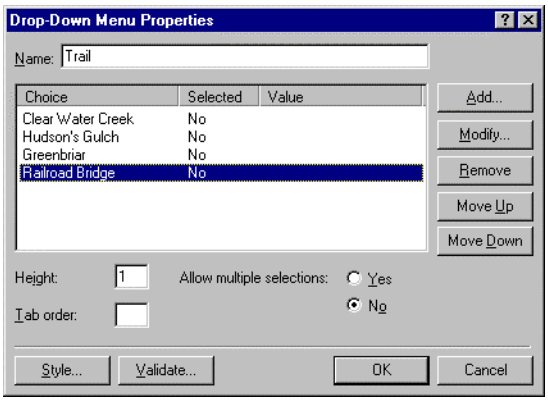

Figure 2-25: Choices Filled In

Now that you have the choices entered, you can begin to re-order them.

10. Select **Railroad Bridge** and click on the **Move Up** button twice. (See Figure 2-24.)

11. Select **Hudson's Gulch** and click on the **Move Down** button once, then click **OK**.

12. Click on **Preview** and click on the Drop-Down Menu to observe how it operates, then return to **Normal** view.

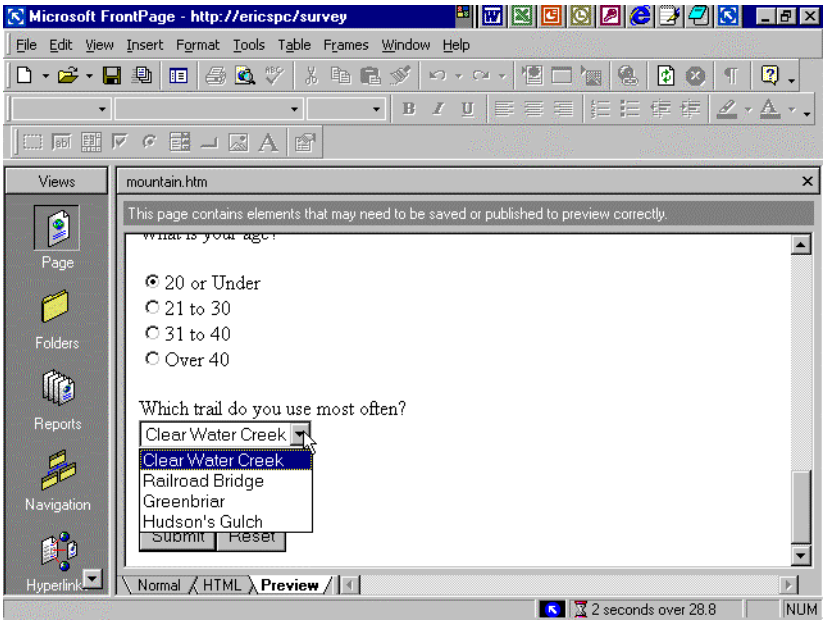

Figure 2-26: Drop Down in Preview

Next, you will change the appearance of the Drop-Down Menu.

13. Double-click on the Drop-Down Menu and click on **Yes** for allowing multiple selections and change the **Height**: to **3**, then click **OK**.

14. Click in after the sentence “Which trail do you most often use?” and press **<Shift>-<Enter>**.

15. Type the following sentence:

    (Hold down the Control Key (PC) or the Command Key (Mac) while clicking to make multiple selections.)

16. Click on **Preview** and observe how the menu now operates, then return to **Normal** view.

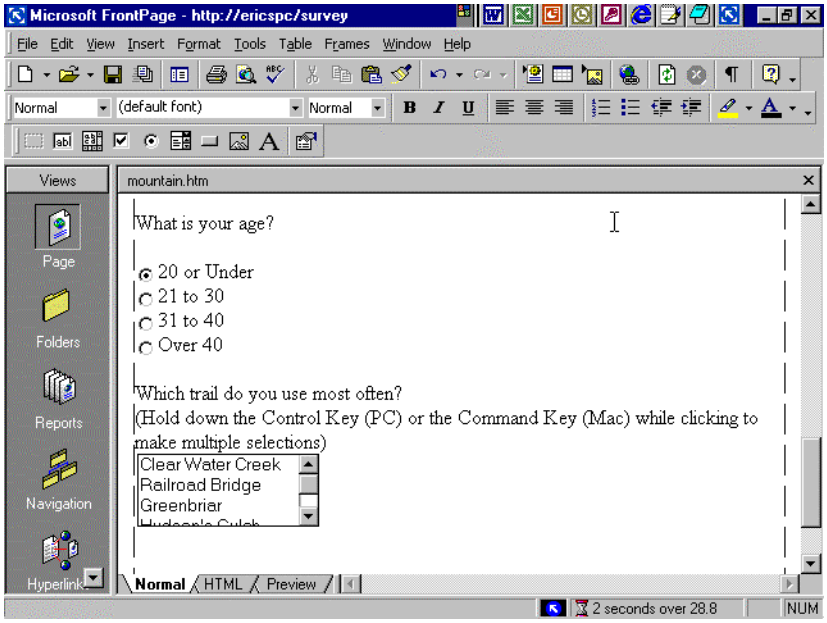

Figure 2-27: Multiple Selection Drop-Down Menu

Web Page Design Tip

Whether you put in the directions about holding down the Control or Command key will depend on your audience for the page. If they are fairly sophisticated computer users, they should know how to make multiple selections, but the opposite is true for novice computer users. Also, you will most likely want to increase the height for multiple selections because it makes it easier for users to click on the choices.

Form Field Labels

Using form field labels allows you to click on the text next to a form field to invoke the action of the field, such as clicking “Tandem” next to the checkbox will cause the box to be checked. This is similar to how Windows operates. If you have ever noticed, you do not have to click exactly on a check box or radio button, as long as you click anywhere on the field label the box or button will be clicked.

Exercise 2-7: Making Labels

In this exercise, you will change the check boxes you created earlier so that the user of your form can click the text next to the box to also have the box checked.

1. Scroll back up to the check boxes.

2. Select Mountain Bike and the check box next to it.

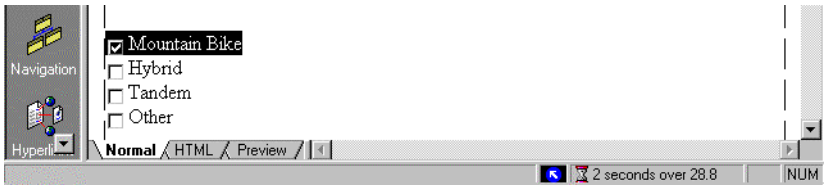

Figure 2-28: Selecting Field and Label

3. On the Forms toolbar, click **Label**.

4. Click on **Preview**.

5. Click on the text “Mountain Bike.”

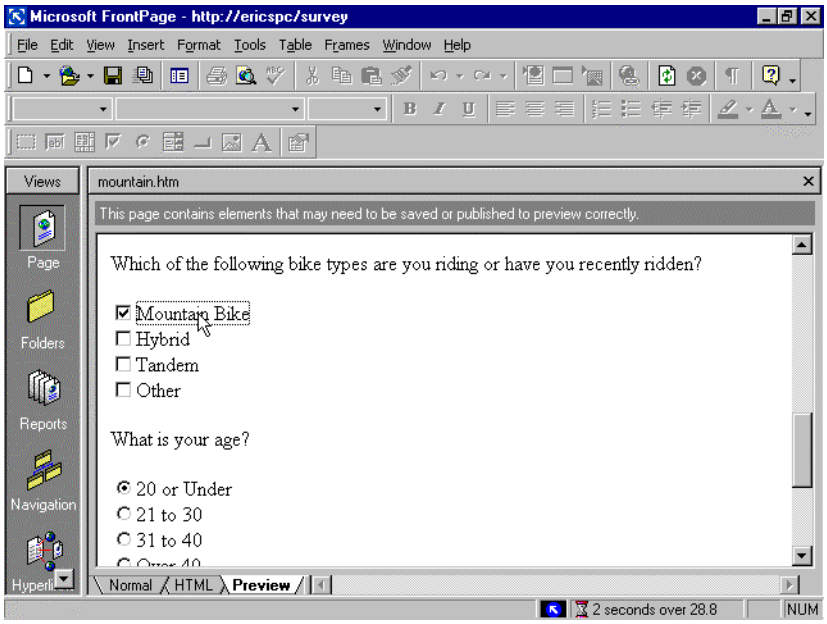

Figure 2-29: Form Label Implemented

Notice that is does check and uncheck the box and that the other boxes do not operate the same (you have to check in the box for them, but you will fix that in the next step).

6. Repeat steps 2 and 3 for the other check boxes.

Push Button

In order for the form to be processed, you must include a button for submitting the form contents, which is why FrontPage automatically inserts the "Submit" button. When the user is finished, he clicks on the submit button, invoking a script or FrontPage Component. FrontPage also inserts a "Reset" button, which clears any data entered by the user. The other type of button you can choose to insert is "Normal" button, which can be assigned a script.

The default label for the buttons are "Submit," "Reset" and "Button" (for the Normal button). You can change the label to anything you wish. You can also assign a name to anyone of the buttons.

There is a **Form…** button on the Push Button Properties dialogue box that takes you to another dialogue box where you can choose how the form will be handled (i.e., FrontPage Component, CGI, etc.). You can get to this same dialogue box by right-clicking anywhere on the form and choosing the **Form Properties…** option or click **Form Properties** on the Form toolbar. In this next exercise, you are going to save the results to a text file using a FrontPage Component.

Saving the Form Results with FrontPage

FrontPage allows you to take the user entered data and post it to a file in many different formats. This type of form handling is normally conducted by complex CGI scripts, but FrontPage has a simple dialogue box for setting up the handler options. If you select "Send to:" FrontPage will use a Component to handle the results, either one that uses a text file or E-mail.

The default is to send the results to a text file. The default file name is "_private/form_results.txt". That would place the text file in your current Web. The "_private" folder is a hidden folder, meaning that it will not be found by outside users of your Web. You can also choose to send the file anywhere you want, such as your hard drive, by simply typing in a new location and filename.

The E-mail option under "Send to:" will send the results as an E-mail message. The FrontPage Component handles the mailing, once the mailing option has been configured. If you try to use this option, you are likely to receive the message shown in Figure 2-27. Check with your server administrator at this point, because it requires modification of a FrontPage initialization file.

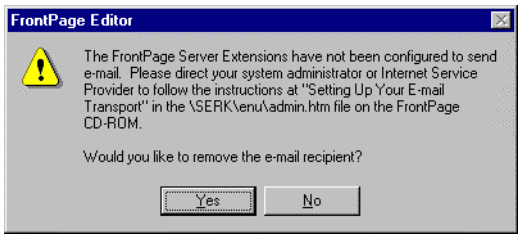

Figure 2-30: E-mail Option Message

You can have the results posted to both the E-mail address and a file by filling in both fields. If you choose "Send to other:" the form can be handled by another means, such as your own CGI script.

The only drawback to FrontPage Component handlers is that the Web server where the FrontPage Component is executed must have the proper FrontPage server extensions. Check with your systems administrator about whether the server extensions are installed before using this option.

The **Options…** button will allow you to choose preferences for each of the major options. When you click on that button with the "Send to:" option selected, you receive a dialogue box with four tabs: File Results, E-mail Results, Confirmation Page and Saved Fields.

On the File Results tab you can set the format of the results file. There are basically three major file types for you to choose from:

- HTML
- Text (ASCII Plain Text)
- Formatted Text

There are several different formats available for those file types. For example, you can choose a tab delimited text file, which can be easily imported into most modern databases. You can also create a second results page to send the results to two different departments at the same time. They could then manipulate the data for their own purposes. For example, the marketing department might only want the name and phone number to import into a contact database. You can specify the page name and file format for the second results page.

On the E-mail Results tab you can choose the same file formats for the E-mail message as well as the subject and reply-to lines of the message. The default for the subject is "Form Results," but you might want to make it more descriptive by typing in your own subject in that field. You can also make the subject one of your field names by checking that option, which will include the user's input for the field in the subject line. If you ask users to fill in a field containing their E-mail address, you can then include the name of that field in the reply-to section and be able to reply to their form submission.

On the Confirmation Page tab you can choose files that will appear when the submission is successful and when it fails. This is not necessary as FrontPage will automatically put these pages in for you, but if you want to create custom pages you can create links to them on this tab. For example, a confirmation page might read, "Thank you for your submission. Please press the 'Back' button on your browser to return to our site."

The Saved Fields tab allows you to choose which fields you want to collect and the order of their appearance on the results page(s) – these settings will affect both pages should you designate a second results page. By default, all of your form fields appear on the list. You are able to re-order or remove fields. Also on the Saved Field tab are several other items to include, such as time, date and even browser type.

That information is picked up by the FrontPage Component and appears in the results page or E-mail message.

There is one other option, which is to send the results to a database. You can only use this option, however, if you have both the FrontPage Server Extensions and Active Server Pages Server Extensions installed. If both are not installed, the option is grayed out. FrontPage will assist you in creating your own database in Microsoft Access format (.mdb). Alternatively, you can map the form fields to the tables in an existing database.

Exercise 2-8: Submitting the Form

In this exercise, you will create submit and reset buttons for the form and post the form using the Save Results FrontPage Component.

1. Scroll to the bottom of the form boundary.

2. Double-click on the **Submit** button.

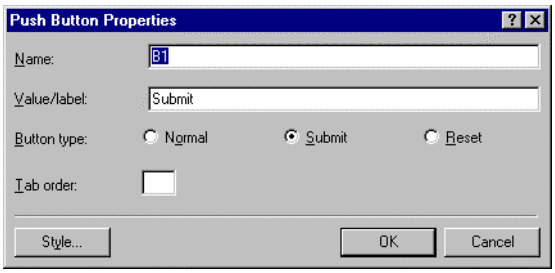

Figure 2-31: Push Button Properties

3. In the **Name**: field, type **Submit** and in the **Value/Label:** field, type **OK, Submit It!**, then click **OK**.

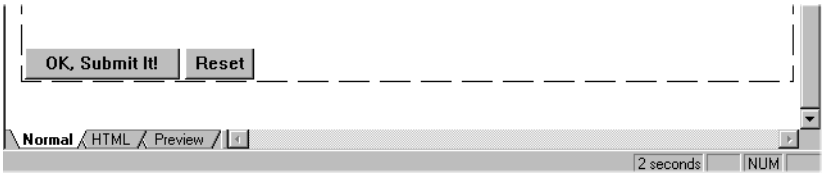

Figure 2-32: Edited Push Button

4. Click on the **Form Properties** button on the Form toolbar.

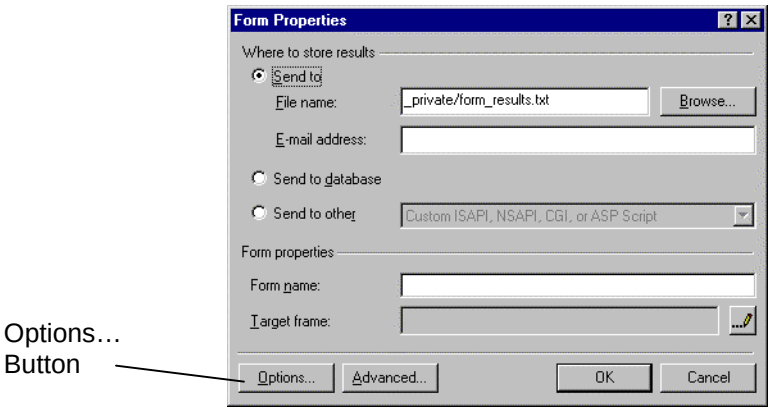

Figure 2-33: Form Properties

The Form Name can be used to create a label for the overall form, such as "Mountain Bike Survey 1." When the form is processed that name will appear in the posted data. This can be useful if, for example, you have several forms being posted.

5. In the **File Name**: field, select **_private/form_results.txt** and type **C:\results.txt**.

6. Click on the **Options…** button.

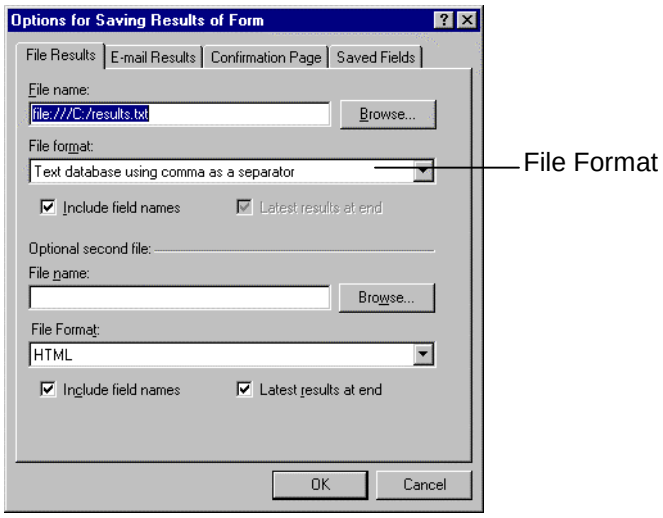

Figure 2-34: Form Posting Properties

7. From the top **File Format** drop down, choose **Text database using comma as a separator**

8. Click on the **Saved Fields** tab.

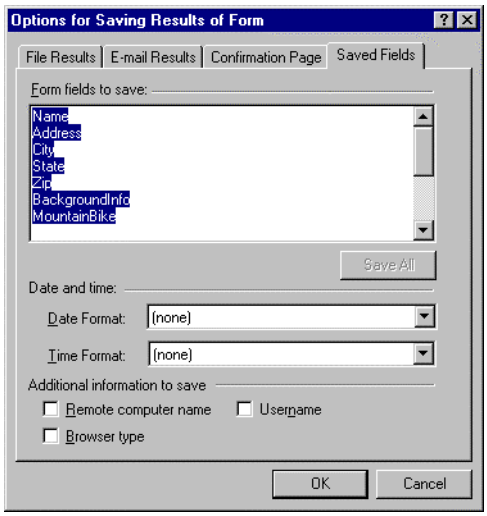

Figure 2-35: Saved Fields

9. Click on **Time**, **Date** and **Browser type**.

10. Click **OK** on all three open dialogue boxes to return to the form.

11. Click on the **Preview in Browser** button.

12. Fill out the form correctly and submit it.

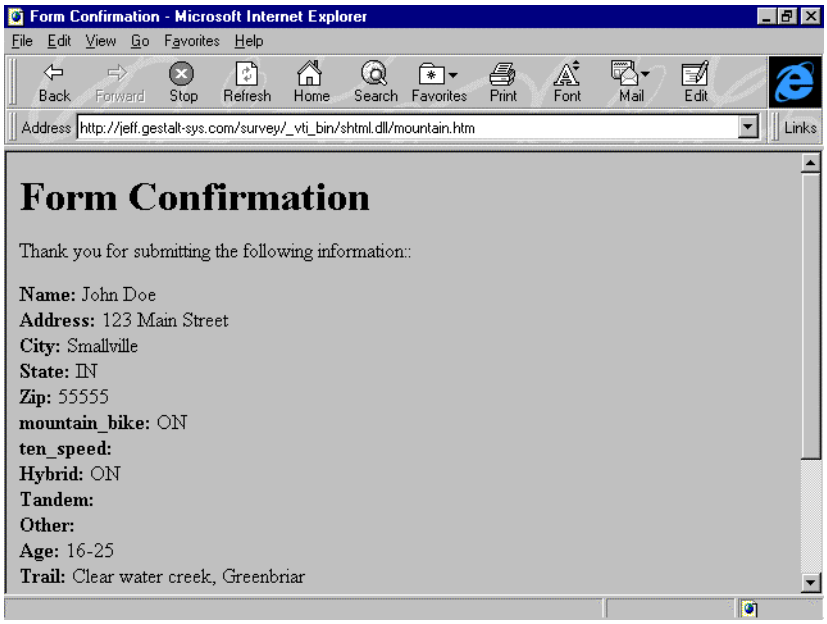

Figure 2-36: Form Confirmation

This page is automatically generated by the FrontPage Component. It will appear unless you specify your own confirmation page.

14. Open a text editor or word processor and open the file **C:\results.txt** to view your submission.

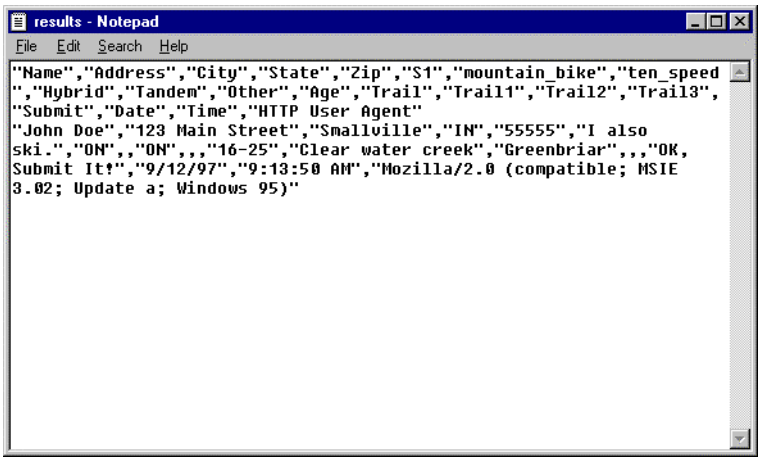

Figure 2-37: Results Page

This file may not be easy to read, but you if you created a database for the results each form field would have its own field in the database. The results could then be searched, counted, sorted, etc.

Form Templates

The form templates create many of the most common forms found on the Web. Although these templates help to create forms quickly, you need to do some customization to make sure the form is handled properly. At the least, you will need to edit the properties of the form fields to your liking.

The forms templates include: Confirmation Form, Feedback Form, and Guest Book. An example of a form template is shown in Figure 2-38.

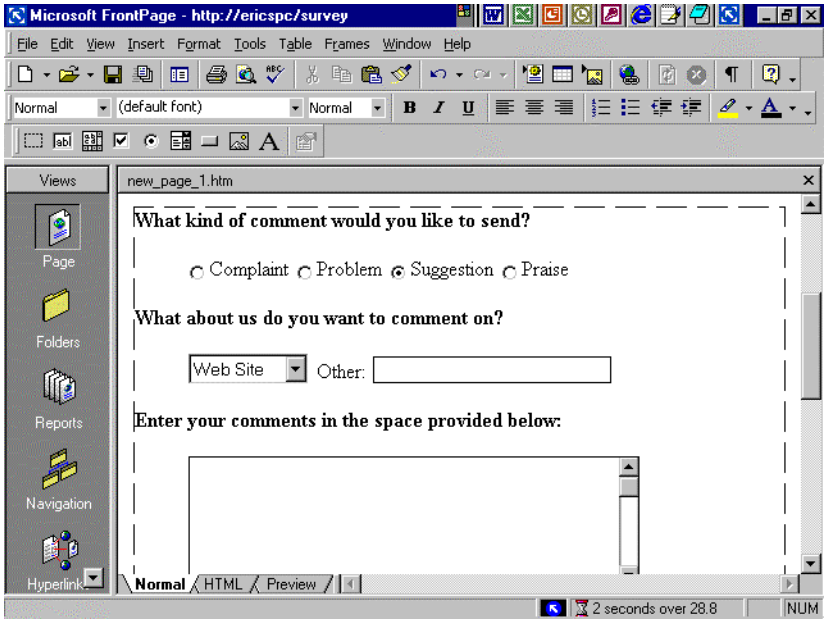

Figure 2-38: Feedback Form Template

Form Wizard

The Form Wizard is probably the quickest way to create a customized form. It walks you through the creation of a form, allowing you to choose a form filename, the types of questions you want to ask, the presentation style of the form and the output options.

Once you put in the page title and filename, you can add a series of questions to the form by clicking the Add button and choosing a question topic, such as “contact information.” Once you select a question topic, you can choose a variety of options for the topic. After selecting those options, you are taken back to the screen where you click on the Add button, should you want to include more questions. After adding questions you can choose from several presentation options, such as regular paragraphs or a list. Finally, you choose the format for the form output (either web page, text file or CGI) and the filename for the results file.

You will still have to customize the output from the Wizard, but it should be fairly close to what you had in mind. Most of the editing will be creating names for each of the form fields. Figure 2-39 shows one of the Form Wizard screens.

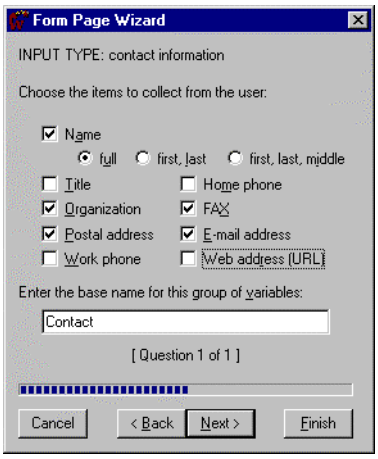

Figure 2-39: Setting Variables for Contact Information Question

Review

In this lesson, you learned the purpose and function of forms on HTML pages. You learned how to insert each one of the form fields and edit their properties. You also saw that forms can be quickly created using templates and wizards.