

UsingForms: HTML and Word

Margie Metzler

margie@netapp.com; **x3185**

<http://web.netapp.com/engineering/webresources/>

March 15, 2001

Table of Contents

Introduction	1
Introduction to HTML forms	1
Example: Basic Form Template for mail.pl	3
Here is the HTML code that created this form:	3
Creating a Form: Basics	5
Collecting User Information with Text Fields	6
Collecting User Information with Text Fields	7
Radio Buttons	8
Menus	9
Submit, Reset, Hidden Variables and Hidden Text	10
Or...Create your Form Using FrontPage, Netscape Composer, Dreamweaver, or	11
Modifying your form to use mail.pl	11
Detailed Description of Derek's Forms Template (HTTP://WEB.NETAPP.COM/~DEREK/MAILPL/) Overview	13
Features	13
Field Descriptions and Usage	14
Creating Word Tables and Forms	19
Tables	19
Why Use Tables?	19
Creating Tables	19
Fast and Easy.....	19
Draw a Table	20
AutoFormat a Table	21
Selecting Cells, Rows or Columns	21
Merging and Splitting Cells	22
Insert and Delete Cells, Columns and Rows	22
Changing Row and Column Dimensions	22
Navigating in a Table	23
Selecting Text in a Table	23
Applying Formatting to Table Cells	23
Inserting Tabs in a Table	24
Cutting or Copying Table Information	24

Splitting Tables	24
Table Page Breaks	24
Converting Text to Tables	25
Converting Tables to Text	25
Using Calculations	25
Sorting	27
Taming Unruly Tables	27
Putting Tables Within Margins.....	27
Distributing Columns Evenly	28
Spacing Table Rows Vertically.....	28
Drawing a Table	28
<i>Forms</i>	<i>29</i>
Using the Forms Toolbar	29
The Forms Process	29
Quick Forms: Creating a Questionnaire	30
Printing Data Only for Forms.....	30
<i>Using Word Documents on the Web.....</i>	<i>30</i>
<i>References.....</i>	<i>32</i>
<i>Index</i>	<i>33</i>

Using Forms

Introduction

Forms are a quick way to get an document to a user, have them gill out the spaces you've set up, and send the data back to you. We're all familiar with paper forms, which you fill out manually and then fax, hand deliver or mail back to whoever needs the information. But more and more, people are using electronic forms, which are somewhat more complex to create but MUCH easier for people to use.... And *vastly* quicker!

Forms on the Web are essentially HTML documents you create however you create your HTML: by writing native HTML, or by using an editor. The problem is, after a form has been created on an HTML page, there must be a mechanism to deliver the data to someone. That mechanism is a program, called a CGI script (Common Gateway Interface), which is called by the HTML document by the browser, but which lives on the Web server. That program is typically a PERL script, and that means you need a programmer to write it. The script usually either e-mails the data to someone, or uses it in a database of some kind. We haven't resolved the database part, but we do have a workaround solution to the people who want the data emailed.

A Word form is created using a y, usually, and has form fields within the cells of the table. The form is than protected, or locked, and sent electronically to the people who need it. Those people can open the document and print it, but they can type only within the form fields.

Then they have to send it back to whoever needs the in formation, typically via e-mail. You can link to these forms from a web page, but like all rod documents, they can only be seen by Internet Explorer users. Netscape users can download the files, but only MS Office users can open them.

Introduction to HTML forms

HTML **Forms** are a way to collect information from a user via a web page and send that data to a program for further processing. It is not the intent of this class to teach programming or the handling of form data, but a basic mail program has been provided on our server by Derek Ryan to display the form data for class use.

In practice, you place special input fields along with regular HTML markup. A user inputs data into the fields when browsing that page. When the forms are activated using the **submit** button, the browser will place the form data into variables and send that information to the program specified in the form. (See Derek's page at <http://web.netapp.com/~derek/>):

“I noticed that much of the cgi programming needed by folks who wish to use forms on the web involved simply mailing some data from the form to some specified recipient. Unfortunately, that usually means new scripting/programming for each new form. I decided to make it much easier, and wrote a generalized script that will allow web authors to generate new web forms without having to do any new scripting. It's called mail.pl, and I think it's relatively cool. All you have to know is 1) how to make a web form, and 2) how to alter a few hidden input variables, and you can make your own arbitrary form and get the data sent to any address(es) you specify. (You can even specify which fields are required for form submission and mail.pl will check those fields and require them to be filled out before the form is submitted.)”

-Derek Ryan

Many forms enhancements have been added to HTML 4.0, but since these are not widely deployed, we will concentrate on the form elements available in HTML 3.2.

A form basically works like this: Your viewer enters appropriate data, then submits the form. Or, the form can be cleared if the user makes a mistake. After the page has been submitted, the browser calls the *cgi script*, which has been requested in your HTML document, which is mail.pl in our case. (**CGI stands for Common Gateway Interface.** CGI scripts are programs that can be written in C++, PERL, or other applications. Typically, HTML coders are not programmers so we have to work around this limitation.)

Example: Basic Form Template for mail.pl

(<http://web.netapp.com/~derek/mailpl/template.html>)

Required Information