SLAC-TN-96-3
September 1996

PC-based Web Authoring*

How to Learn as Little Unix as Possible While Getting on the Web**

L. Trindle Gennari, M. Breaux, S. Minton, R. McDunn
Stanford Linear Accelerator Center
Stanford University, Stanford, CA 94309

Abstract: *This document is a general guide for creating Web pages, using commonly available word processing and file transfer applications. It is not a full guide to HTML, nor does it provide an introduction to the many WYSIWYG HTML editors available. The viability of the authoring method it describes will not be affected by changes in the HTML specification or the rapid release-and-obsolescence cycles of commercial WYSIWYG HTML editors. This document provides a gentle introduction to HTML for the beginner, and as the user gains confidence and experience, encourages greater familiarity with HTML through continued exposure to and hands-on usage of HTML code.*

Chapter Outline **Page**

---

* This document is adapted from *Mac-based Web Authoring* by L. Trindle Gennari.
** Work supported by Department of Energy contract DE-AC03-76SF00515.

DISCLAIMER

1 Introduction

This document describes one set of methods and tools for creating and maintaining World Wide Web pages on an IBM-compatible PC, using either Windows 3.x or Windows NT. This is not the only set of tools, nor is it the only method, but it'll get you started. Note that some of the tools are shareware. The fees for shareware are generally small enough that they can be paid with petty cash. Please register if you use these wonderful tools.

This document assumes that you are already familiar with one or more Web browsers, and that the files you are working with are already in, will be created in, or can be pulled into Microsoft Word 6 or 7. Microsoft Word is a commercial product, and is not available as shareware.

2 Before You Begin

Before attempting to use this document, it's good to have an overall picture of the PC-based Web authoring process it describes. For example, you will need to take certain steps prior to diving in and creating Web pages, such as obtaining a Unix account (if you don't already have one), getting your Unix account AFS enabled, and understanding the basics of the Unix directory hierarchy system. This document will guide you through all of the above stages, offer step-by-step procedures necessary to produce the desired output (i.e., HTML Web pages), and point you toward the tools you will need to achieve this output.

3 Get the Tools to Use These Procedures

The first step in creating your Web pages is obtaining the necessary tools: Internet Assistant, Program Manager, WS_FTP or FTP, and Microsoft Word 6.x. (You do not need to obtain any applications you already have.) You will be making use of these tools as they appear in this document; be sure not to use a tool before it is mentioned.

Program Manager and WS_FTP (or FTP) are part of the Windows NT installation. Internet Assistant is available from the Microsoft Web site at the following site:
`http://www.microsoft.com/msword/internet/ia/`

Microsoft Word is a commercial product, and is not available as shareware.

4 The Necessity of a Unix Account

If you already have a Unix account that is AFS enabled, skip ahead to Section 5.1, "Get Space in the SLAC Web Hierarchy."

If you do not yet have a Unix account at SLAC, you will need to get one in order to use this document. Be aware that it may take a couple of days for your account to be activated. You will need to fill out a "SLAC Computer Account Form" (available

from the SLAC Computing Services Support Center or at *http://www.slac.stanford.edu/comp/form/acctform.ps)*, which will require the signature of your group czar. For further information, contact the SLAC Computing Services Help Desk (ext. 2406).

4.1 Get Your Unix Account AFS Enabled

Once you have a Unix account, you will need to have it AFS enabled, since the files you will be converting from Microsoft Word to HTML must reside in AFS space before they can be visible on the Web. (This will be covered later in this document.) Follow the steps below to enable your AFS account.

Step 1 Write your Unix password on a piece of paper, and next to it write "old." Under that, make up a new 8-character password and next to it write, "new." I know this sounds strange, but trust me....

Step 2 Log in to your Unix account. (See Section 5.3, Step 1.)

Step 3 At the prompt, type:

```
afsacct
```

Step 4 At the prompt, type your old password, and hit return.

Step 5 At the next prompt, type your new password, hit return, retype your new password, and hit return again.

Step 6 Make both passwords the same.

Note: In general, it is permissible to have different NFS and AFS passwords (your old and new passwords), but Telnet only gives you one password prompt per session. The combination of having both passwords be the same and logging in through Unixhub allows Telnet to get through both layers of password protection.

a. Type:

```
kpasswd
```

b. At the prompt, type your new password.

c. At the prompt, type your old password.

d. At the prompt, type your old password again. You are now back to your original password.

Step 7 Eat the paper with your old and new passwords.

Step 8 Log out, and from the **Connect** menu, select **Exit**.

5 Make a Home in SLAC's Web

For the files you will be moving to AFS space to be visible to the World Wide Web, they will need to reside in a Web-visible directory. Your Web files will reside in one of the three places, summarized in the table below.

Location	Unix File Path	URL (Web location)
Your Unix public_html	*~userid*/public_html/*filename*.html **and/or** *~userid*/public_html/*filename*.gif	http://www.slac.stanford.edu/*~userid/filename*.html
Your AFS group space	/afs/slac/www/grp/*deptcode/filename*.html **and/or** /afs/slac/www/grp/*deptcode/filename*.gif	http://www.slac.stanford.edu/grp/*deptcode/filename*.html
Elsewhere in the SLAC Web	See your local Web Support Coordinator.	See your local Web Support Coordinator.

5.1 Get Space in the SLAC Web Hierarchy

To obtain space in the SLAC Web hierarchy, send e-mail to *www-admin@slac* asking for a directory for your group. This directory will typically be */afs/slac/www/grp/[groupcode]*, where *[groupcode]* is your group or department's two- or three-letter code. Be sure to specify in your e-mail the user names of anyone you want to be in the ACL (the group allowed to work in your new directory). It's best to have at least three people total, yourself and two others.

5.2 Organize Your Web Unix Space

Before your files are in Web-readable format, you need to think about how they will relate to each other and to the organization of the SLAC Web once published. Your *public_html* directory should be home to files such as your personal-professional page and pages specific to your personal work. It is also a good place to test pages destined for the */afs/slac/www/* hierarchy. This hierarchy contains SLAC's institutional information. You may have files in both places. For example, you may have a page of links to vendors you often deal with in your *public_html* directory and a departmental newsletter in the subdirectory */afs/slac/www/grp/yourdept/newsletter/*.

Think about organizing the directory space under your *public_html* or department directory before you publish your pages. Keeping all of your files at the top level is fine if you only have one cluster of pages, but this scheme quickly becomes unwieldy once you have many clusters of pages together in one directory. One option is to separate files into subdirectories by project or subject. (Subdirectories are equivalent to folders in the PC file

system.) For example, */afs/slac/www/grp/mygroup/projectx* is for all pages relating to Project X and */afs/slac/www/grp/mygroup/projecty* is for all pages relating to Project Y. Within these subdirectories you may wish to further separate text and figures. Remember there are no spaces in Unix file or directory names.

5.3 Create a *public_html* Directory and Subdirectories

If you have already created a *public_html* directory, skip ahead to Section 6, "Assemble and Organize Source Materials." If you have not created one, or would like to create subdirectories, follow the steps below.

Step 1 Log in to your Unix account with Program Manager for either Windows 3.x or Windows NT:

For Windows 3.x:

a. Open Program Manager.

b. Open **PC/TCP WinApps** group and choose **TNVT.**

c. Type in or select **Unixhub** as the host and **Connect.**

d. When the login command appears, type in your userid and password.

You are now in your Unix home directory. If you type *pwd* at the prompt, you will see the complete file location.

For Windows NT:

a. Open Program Manager.

b. From the **Accessories** folder, choose **Telnet.**

c. From the **Connect** menu, select **Remote System.**

d. In the "Connect" box that appears, type:

```
Host Name: unixhub
```

e. Leave the other settings as they are.

f. When the login command appears, type in your userid and password.

You are now in your Unix home directory. If you type *pwd* at the prompt, you will see the complete file location.

Step 2 To create a *public_html* directory, type:

```
mkdir public_html
```

Step 3 Create any subdirectories:

a. Make sure you are in your home directory by typing `cd ~` at the command line.

b. Change to your Web project directory. Type:

```
cd public_html or
cd /afs/slac/www/grp/[yourgroup]
```

c. Create a subdirectory. Type:

```
mkdir [subdirname]
```

Step 4 Exit Telnet/TNVT

a. Type `logout` or `exit` at the prompt or select **Exit** from the **Session** menu.

b. At exit, you may save the session for easier access in the future.

6 Assemble and Organize Source Materials

Using Microsoft Word, create the pages you'd like to move to the Web, and place them all in the same folder, if you have not already done so. (Okay, maybe this part sounds obvious, but it really makes the next step, organization, a lot easier.) This procedure is built on using Internet Assistant to create your HTML files, so if you have source files that were not created with Microsoft Word 6 or 7, then pull everything into Word, one file per Web page. Your files should be straight text and figures with no formatting other than paragraph breaks. Once you see all of the material in one place, you'll have a much better idea what shape it should take in your Web space.

Organization is arguably the most important step. Figure out how many Web pages your material will become, what pictures or graphics will go on each page, and the relationship between the pages. Do you have one central page with a bunch of equally important subordinates? Do your pages lend themselves to an upside-down tree structure, with one top page, a few second-level pages, third-level pages off of those, and so on? Having a clear idea of organization from the beginning will make things easier once you're working with your files in Unix space.

7 Basic Formatting for the Web

Although a discussion of Web formatting would require more information than is within the scope of this document, there are basic hints that will help you along in preparing a Word document for HTML conversion. The steps in the next section (Section 8, "Convert Your Word Files Using Internet Assistant") guide you through the conversion process. You will be instructed to choose from a list of styles, and to apply these styles to each paragraph in your Word document. The way your Web page looks will depend on which styles you apply; the formatted Word document

displayed on your screen is a good approximation of what the HTML version will look like.

For a more exhaustive discussion of HTML tags and Web publishing, check one of the resources below.

- "A Beginner's Guide to HTML":

  `http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html`

- The SLAC World Wide Web User's Group (SWUG) has prepared a mini-course in HTML which shows you how to create basic HTML tags:

  `http://www.slac.stanford.edu/slac/www/resource/swug/htmlclass/class.html`

- Books on Web publishing may be found in the SLAC Library by searching in the SLAC Library Book Catalog:

  `http://www.slac.stanford.edu/spires/form/bookspif.html`

8 Convert Your Word Files Using Internet Assistant

Follow the steps below to either convert text to HTML (Section 8.1) or convert graphics to GIF files (Section 8.2).

8.1 The Text

Step 1 Open the Word file you want to convert.

Step 2 From the **File** menu, select **Save As**.

Step 3 In the **Save as Type** pop-down list, select the **HTML** option (*.htm). The Word (*.doc) file should be named with a .htm extension (added automatically when you save as an HTML Document), no spaces in the name, and preferably no uppercase letters.

Example: *filename.htm*

Step 4 Click **OK**.

Step 5 If you look at the style pop-down menu in the toolbar (the default is Normal), you'll see a listing of styles corresponding to HTML tags. To format each paragraph, place the cursor in the paragraph, and pull down the style corresponding to the HTML tag you want applied to that text (i.e., Address, Heading2, etc.).

Note: Internet Assistant is a plug-in, which means having it on your machine allows Word to automatically find it and supply a collection of HTML tags.

Step 6 Save your HTML document.

Note: There is no need to keep a separate copy in regular Word format, since Word can open the HTML files it creates.

Step 7 Repeat Steps 1–5 for each Word document you want to convert to HTML.

Step 8 Close your documents.

8.2 Adding Graphics to your HTML Document

If you want to add graphics to your Web page (i.e., the BSD logo), the graphics must be saved separately as *.gif files.

Step 1 Open Word.

Step 2 From the **File** menu, select **Open** and choose the *.htm file to which you want to add graphics.

*Note: If the file does not appear in the list of choices, type .htm in the **File name** box and click on **Find Now**.*

Step 3 Position the cursor in the document where you want the graphic(s) to be inserted.

Step 4 From the **Insert** menu, select **Picture**, or click on the **Picture** button on the toolbar.

Step 5 In the **Image Source** or **File Name** box, type `/filename.gif`. The / indicates that it is a relative file located in the same directory as the HTML document.

Note: If the graphic is not located in the same directory as the HTML document, you will have to type in the relative URL.

Example: `/directory/subdirectory/filename.gif`

Step 6 In the **Alternative Text** box, type in a description of the GIF image.

Example: `[Department Name] Logo`

Step 7 Click **OK**. A box with the alternative text description may appear on the screen at the point where the graphic should be. This is okay.

Step 8 Save your document.

Step 9 Repeat Steps 3–8 for each graphic you insert.

Step 10 Close your document.

9 Test Your Files Locally

Before moving your files to Unix space, you should test them locally by opening them through your Web browser.

If you would like to change a paragraph tag, follow the steps below.

Step 1 In Word, open the .htm file you'd like to change.

Step 2 Re-tag any paragraphs as necessary.

Step 3 Save the file.

Step 4 Test again by opening the file through your Web browser.

10 Clean Up

At this point in the PC-based Web authoring process, you will have a bunch of files that you will either need to save or throw away. Since it is very easy to accidentally delete files once you're in Unix, it is very important to keep a local backup copy of each file. Create a folder on your PC called WebProjects, and create folders inside it for each Web project you do. (If your internal hard drive is small, you can put your backup folder on an external drive, or keep your backup files on a collection of diskettes.) Keep in mind that these are backup files. The master copies are the ones in Unix Web space. This will discussed further in Section 14, "Making Changes." For now, follow the steps below in making backup copies.

Step 1 Put your HTML and GIF files into the backup folder(s).

Step 2 Throw away all other files (Word, EPS, PICT, etc.).

Note: Make sure all of your local testing is done before throwing away any files, as once these files are trashed, all subsequent fixing must be done in HTML.

11 Upload Your HTML and GIF Files to Unix Space

Follow the steps below for either Windows 3.x (Section 11.1) or Windows NT (Section 11.2) to copy your converted files to the appropriate AFS space in Unix.

11.1 For Windows 3.x Users

Step 1 Log in to your Unix account from Program Manager.

a. Open Program Manager.

b. From the **PC/TCP WinApps** group, choose **FTP**. (It's a file drawer icon.)

c. If the **Open** box does not automatically appear, then from the **Session** menu, select **New**.

d. In the box, type:

```
Host Name: unixhub
USERID: username
Password: [your password]
```

Step 2 Click **Connect**.

Step 3 On the **Local System** side (left), highlight your HTML files:

a. Double click on .. to get to the `c:\` directory.

b. Double click on the directory where your HTML files are saved.

c. Select the proper subdirectory until your HTML files are listed.

Note: You must put your files in your `public_html` directory or one of the directories under `/afs/slac/www` for them to be visible on the Web. Be sure the mode selection is `Binary`.

Step 4 Move files from your `c:\` directory to your `public_html` directory:

a. If you want to put files in your `public_html` directory, from **Remote System** (right side of screen) select **Change Directory** and type:

```
public_html
```

or simply double-click on `public_html`.

b. Select your *.htm file in the local directory (left side of screen) and select the <<copy>> button. If you are given the option of renaming, rename the file with a .html extension.

```
Example: filename.html
```

c. From **Remote System** (right side of screen), select the **Refresh** button. (The transferred file will appear as `filename.htm`.)

d. Select the **Rename** button and rename the file:

```
filename.html
```

Step 5 Move files from your `c:\` directory to your group directory.

a. If you want to put files in the `/afs/slac/www/` hierarchy, from **Remote System** (right side of screen) select **Change Directory** under the **Commands** menu and type:

```
/afs/slac/www/grp/[groupcode]/[etc.]
```

b. Select your *.htm file in the local directory (left side of screen) and select the <<copy>> button.

c. From **Remote System** (right side of screen), select the **Refresh** button. (The transferred file will appear as *filename.htm*.)

d. Select the **Rename** button and rename the file:

filename.html

Step 6 Log out and from the **Session** menu, select **Exit**. (You can save the session for future use.)

Step 7 Go to Section 12, "Set Permissions."

11.2 For Windows NT Users

Step 1 From the **Applications** folder, select **WS_FTP**.

Step 2 In **Session Profile**, type:

Host Name: *unixhub*

Host Type: *Automatic detect*

User ID: *userid*

Remote Host: *your /a/juno directory*

Step 3 Select **Save** (to save the session profile).

Step 4 Type in your password.

Step 5 Click **OK**.

- Local system should be your *c:\WS_FTP* directory.
- Remote system should be your *~userid* directory (Unix).

Step 6 On the **Local System** side (left), highlight your HTML files:

a. Double click on .. at the top of the file to get to the *c:* directory.

b. Double click on the directory where your HTML files are saved.

c. Select the proper subdirectory until your HTML files are listed in the lower section.

Step 7 On the **Remote System** side (right), open the subdirectory into which you want to upload HTML files.

a. If you want to move files into your *public_html* directory, double click on **public_html**, and continue until you reach the appropriate subdirectory.

b. If you want to move files into your group directory, select the **ChgDir** button and type:

cd /afs/slac/www/grp/[groupcode]/[etc.]

Step 8 Move files from `c:\`

a. Highlight the `c:\` htm file you want to move.

b. Make sure **Binary** is checked at the bottom. (All WWW files must be moved as binary files.)

c. Select the right arrow button (-->) to move the file.

d. Select the **Refresh** button on the **Remote System** side to see if the file was transferred (file will have a .htm extension).

Step 9 Rename the AFS file as an HTML file:

a. Select the *`filename.htm`* file on the **Remote System** side and click the **Rename** button.

b. Rename the file with an HTML tag (i.e., *`filename.html`*).

c. Click **OK**.

d. Repeat as needed for all files moved.

Step 10 Select the **Close** button.

Step 11 Select the **Exit** button and close the **Applications** folder.

Hint: Windows NT users also have the option of transferring files to AFS space using the "F" drive from File Manager. See your system administrator for setup and instructions.

12 Set Permissions

The files you've moved into Unix space must have correct permissions to be readable through the Web. To check this, and to change the permissions if necessary, follow the steps below.

Step 1 Log in to your Unix account with Program Manager, if you are not already logged in. (If you are not, see Section 5.3, Step 1.)

Step 2 Go to the directory containing your files by typing *`cd`* then the directory name. For example, if your files are in your *`public_html`* directory, type:

```
cd ~/public_html
```

Note: If you are already in your home directory, you can simply type `cd public_html`.

Step 3 List the files and their permissions. Type:

```
ls -l
```

The output will look something like this:

```
-rw-r--r--   1 owner   is      55602 Apr  1 10:30  myfile.html
```

Step 4 The filename is at the far right; the first column entry shows the permissions of the file or directory and should be *-rw-r--r--* for a file, or *drwxr-xr-x* for a directory. If the first column does not look like this, set the permissions to make your files Web-visible:

a. For a file, type:

```
chmod 644 [filename]
```

b. For a directory, type:

```
chmod 755 [directoryname]
```

Note: All files and directories inside under `public_html` must have proper permissions in order to be Web-visible.

13 Test and Validate Your Web-visible HTML

Once you have your files installed in their new, Web-visible home, they're ready to be tested on the Web. Open your Web browser and enter the URL based on the naming scheme from the table in Section 5. (If you encounter errors, check the URL to make sure you have typed it correctly and in full.)

It's also a good idea to run your Web page(s) through one of the Web page analysis tools. These tools check your Web document for such things as correct spelling and valid hypertext links. The following two are good ones, and are available over the Web. Be prepared to enter the URL of the Web page(s) you'd like to test.

- Doctor HTML:

  `http://imagiware.com/RxHTML.cgi`

- Weblint:

  `http://www.unipress.com/cgi-bin/WWWeblint`

A list of other such Web page analysis tools is available at the following address: `http://www.charm.net/~web/Vlib/Providers/Validation.html`

14 Making Changes

When you need to make changes to your pages, follow the steps below, which entail downloading the HTML files to your PC, making the changes, and replacing the Web-visible files with your corrected ones. This method prevents versioning problems between the backup files on your PC and the files on the Web.

This procedure requires basic knowledge of HTML.

Step 1 To download the HTML file you want to edit, use WS_FTP (using the left arrow button [<--]) or FTP. (See Section 11.1 on page 9 for FTP or Section 11.2 on page 11 for WS_FTP.)

Step 2 Open the file in Word.

Step 3 Make the necessary changes.

Step 4 Save the file as a .htm file.

Step 5 Test the file locally (see Section 9 on page 9). If the page doesn't load properly, try clicking on **Reload**.

Step 6 Use FTP or WS_FTP to upload the file (see Section 11 on page 9). Follow the directions in the uploading section (Section 11). When you rename the file with a .html extension, it automatically replaces the old file.

Step 7 Test the file on the Web. If the page doesn't load properly, try clicking on **Reload**.

Step 8 Clean up. (See Section 10 on page 9.)