SLAC-TN-96-2
June 1996

Mac-based Web Authoring

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

L. Trindle Gennari
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**

* Work supported by Department of Energy contract DE-AC03-76SF00515.

1 Introduction

This document describes one set of methods and tools for creating and maintaining World Wide Web pages on a Macintosh. 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. 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 Mac-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: GIFConverter, rtftohtml, MacSamson, Fetch, and Microsoft Word. (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.

GIFConverter and rtftohtml are available from the University of Texas Mac Archive at `http://wwwhost.ots.utexas.edu/mac/`. The archive is indexed by product to allow you to search by application name.

MacSamson and Fetch are available on Public Disk 1 via the AppleTalk network. In the **Chooser**, click **AppleShare** and choose the **Zone** and **Server** below.

Zone: Computer Center

Server: PC Hub

In the dialog box that appears, click **Guest**, and in the next box, choose **Public Disk 1**. On Public Disk 1, open the folder Things You Need. Create a folder on your hard drive and copy the entire folder corresponding to the tool you need. Fetch is located inside the Internet Tools folder; for MacSamson, go into the Networking Tools folder (which is also in Things You Need), then into the Terminal Emulators folder.

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/unix/unix.html/acctform.ps`), which will require the signature of your group czar. For further information, contact the SLAC Computing Services Help Desk at extension 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 (as 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 with MacSamson:

a. Open MacSamson.

b. The Telnet Sessions box will open. If it does not, then from the **Sessions** menu, choose **Telnet**.

c. In the boxes at the bottom of the screen, type:

Host: `unixhub`

Session: `unixhub`

d. Click **Open**.

Step 3 You will be asked to type in your username (after the "login" prompt), as well as your password.

Step 4 After you are logged in, you may enable your AFS account. Type:

```
afsacct
```

Step 5 At the prompt, type your old password, and hit return.

Step 6 At the next prompt, type your new password, hit return, retype your password, and hit return again.

Step 7 Make both passwords the same.

Note: In general, it is permissible to have different NFS and AFS passwords (your old and new passwords), but Fetch (an application you will be using later in this document to move your converted Word files to AFS space) only gives you one password prompt per session. The combination of having both passwords be the same and logging in through Unixhub allows Fetch 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 8 Eat the paper with your old and new passwords.

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	File Path	URL
Your public_html	~*userid*/public_html/*filename*.html	http://www.slac.stanford.edu/~*userid*/*filename*.html
Your group space	/afs/slac/www/grp/*deptcode*/*filename*.html	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, pages specific to your personal work, and 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 Mac 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 If you are not already logged in, log in to your Unix account with MacSamson. (If you are not, see Section 4.1, Steps 2 and 3.)

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 ~`.

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]
```

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 rtftohtml and GIFConverter to create your HTML files, so if you have source files that were not created with Microsoft Word, 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 rtftohtml") 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 rtftohtml

Follow the steps below to convert a Word file into an HTML file.

Step 1 Open Word.

Step 2 From the **File** menu, choose **Open** and navigate to the folder in which you placed your tools. Inside that folder:

a. Open the rtftohtml-mac folder.

b. Open the docs folder.

c. Open the file sample_styles.rtf.

Step 3 From the **File** menu choose **Save as...** and do the following, depending on which version of Microsoft Word you are using:

- If you are using Word 6.0:

  a. Name the file 4rtftohtml.

  b. From the **Save File as Type** pop-down list, choose the **Document Template** option.

  c. Save the file in the Templates folder inside your Microsoft Word 6.0 folder.

  d. Close the 4rtftohtml file.

- If you are using Word 5.1:

  a. Name the file 4rtftohtml.

  b. From the **Save File as Type** pop-down list, choose the **Stationery** option.

  c. Save the file in the Sample Documents folder inside your Microsoft Word 5.1 folder.

  d. Close the 4rtftohtml file.

Step 4 Open the Word file you want to convert.

Step 5 If you are using Word 5.1, skip to Step 8. If you are using Word 6, select **Format...Style Gallery...**

Step 6 Select 4rtftohtml from the list of template styles on the left, and click **OK**. This will copy the styles from the template to your document.

Note: If the listing of styles on the left only shows Normal, click Browse. Navigate to your Word folder, highlight the Templates folder, and click Use Selected Folder.

Hint: Setting up a style template is a great way to keep a collection of styles you want to use regularly and not re-create for each new document. You can also use this method to maintain a consistent look among a group of similar documents.

Step 7 Skip to Step 12.

Step 8 In Word 5.1, Choose **Style** from the **Format** menu.

Step 9 With the **Style** dialog box open, choose **Open** from the **File** menu.

Step 10 Navigate to the 4rtftohtml template in your Sample Documents folder, and click **Open**. This will copy the styles from the template to your document.

Hint: This is a great way to keep a collection of styles you want to use regularly and not re-create for each new document. You can also use this method to maintain a consistent look among a group of similar documents.

Step 11 Close the **Style** dialog box.

Step 12 If you look at the style pop-down menu in the toolbar (the default is Normal), you'll see the listing of styles that rtftohtml converts to corresponding HTML tags. For each paragraph, place the cursor anywhere in the paragraph, and pull down to the style corresponding to the HTML tag you want applied to that text. (See Section 7, "Basic Formatting for the Web.")

Hint: If you would like to include hypertext links in your HTML documents, this can only be done after processing all of your files with rtftohtml. This must also be done in HTML. For instructions on how to include links, refer to one of the HTML resources in Section 7, "Basic Formatting for the Web."

Step 13 Save your document as Rich Text Format (RTF) by choosing this option from the **Save File as Type** pop-down list, renaming before you save. The RTF file should be named with an RTF extension, no spaces in the name, and preferably no uppercase letters.

```
Example: myfile.rtf
```

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

Step 14 Repeat Steps 4 to 13 for each Word document you want to convert.

Step 15 Close your documents.

Step 16 Drag your RTF file onto the rtftohtml icon (inside the rtftohtml-mac folder) to convert to HTML. If your file is named test.rtf and contains text and one picture, your output will be three files, test.html, testToC.html, and test1.pict; these files will all be placed in the same folder as your RTF file(s). You may also have a file with a .err extension. This will list any problems rtftohtml found with your file. Since rtftohtml is pretty forgiving, you can usually ignore this file.

Step 17 Drag any other RTF files you'd like to convert onto the rtftohtml icon. If you have any PICT (or other graphic) files, go to Step 18.

Step 18 Drag your PICT file onto the GIFConverter application icon. GIFConverter will open your picture in a window. (GIFConverter will work with GIF, TIFF, EPS, RIFF, PICT, JPEG [JFIF], MacPaint, and Thunderscan files). After you have dragged your file onto the GIFConverter icon:

a. Choose **Save As...** from the **File** menu.

b. Choose **GIF** from the pop-down menu. The .pict extension will automatically be replaced by .gif. Don't change the name of your GIF file, or the links to it in the HTML file created by rtftohtml will not work.

9 Test Your Files Locally

Before moving your files to Unix space, you should test them locally using your Web browser. Simply drag each HTML file onto your browser icon to make sure the file opens and looks they way you want.

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

Step 1 Open the RTF file corresponding to the HTML file you'd like to change.

Step 2 Re-tag any paragraphs as necessary.

Step 3 Save the file.

Step 4 Process again with rtftohtml.

Step 5 Test again by dragging the file onto your browser icon.

10 Clean Up

At this point in the Mac-based Web authoring process, you will have a bunch of files with the .html extension, some with .rtf, and others that you will either need to save or can be thrown 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 Mac 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, RTF, 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 to copy your converted files to the appropriate AFS space in Unix.

Step 1 Open Fetch.

Step 2 In the **New Connection** window, type:

Host: unixhub

User ID: [youruserid]

Password: [yourpassword]

Step 3 In the **Directory** box, do one of the following:

a. If you want to put files in your `public_html` directory, type:

`~/public_html`

b. If you want to put files in the `/afs/slac/www/` hierarchy, type:

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

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 through the Web.

Step 4 Click **OK**.

Hint: Once you are in Fetch, you may change to another directory by pressing Command-D and typing in the path of the desired directory.

Step 5 Click **Put File...**

Step 6 Navigate to the .html file you created, and click **Open**.

Step 7 In the dialog box that appears, rename your file and change the uploading format if necessary. Do not rename your GIF files or the links to them will break.

Note: Text files must be uploaded as Text, and GIF files must be uploaded as Raw Data, or they will not be visible through the Web.

Hint: If you want to upload many files at once, choose ***Put Folders and Files...*** *from the Remote directory, select a file, click* ***Add*** *(repeat for all files), then click* ***Done.*** *Remember not to upload GIF files this way, since you need to use the Raw Data upload format rather than the default. Some versions of Fetch will let you choose the upload formats for groups of files. In this case, choose Text for text files and Raw Data for other file types.*

Step 8 Click **OK**.

Step 9 Quit Fetch when you are finished uploading.

12 Set Permissions

The files you've moved into Unix Space should have the correct permissions so that they are readable through the Web. To check this, and change the permissions if necessary, follow the steps below.

Step 1 Log in to your Unix account with MacSamson, if you are not already logged in. (If you are not, see Section 4.1, Steps 2 and 3.)

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 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 `http://www.charm.net/~web/Vlib/Providers/Validation.html`.

14 Making Changes

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

This procedure requires basic knowledge of HTML.

Step 1 Use Fetch to download (**Get**) the HTML file you want to edit.

Step 2 Open the file in Word.

Note: When you open the file in Word, you will see HTML code on your screen. If you are only adding text to your document, insert the text in the desired location, ignoring the code. However, if you add paragraphs requiring HTML tags, a new paragraph, heading, etc., you will have to put these tags in by hand. See "A Beginner's Guide to HTML" at http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html.

Step 3 Make the necessary changes.

Step 4 Save the file as **Text Only** with the same name.

Step 5 Use Fetch to upload (**Put**) the file. Uploading a new file with the same name automatically replaces the old with the new.

Step 6 Clean up.