

Site Management

Nadalia Yuehong Liu
IT Lab
School of Information
University of Texas at Austin
Fall 2003

Introduction

Web site structure and management is one of the most important and overlooked aspects of web design. If you begin your site with a clear structure, adding more pages will be easy.

A webpage is merely an index to different files, whether they are fill-in forms, photos, large clickable images, hyperlinks, PDF files, sounds, videos for downloading, etc.

A Webpage is an index to different files

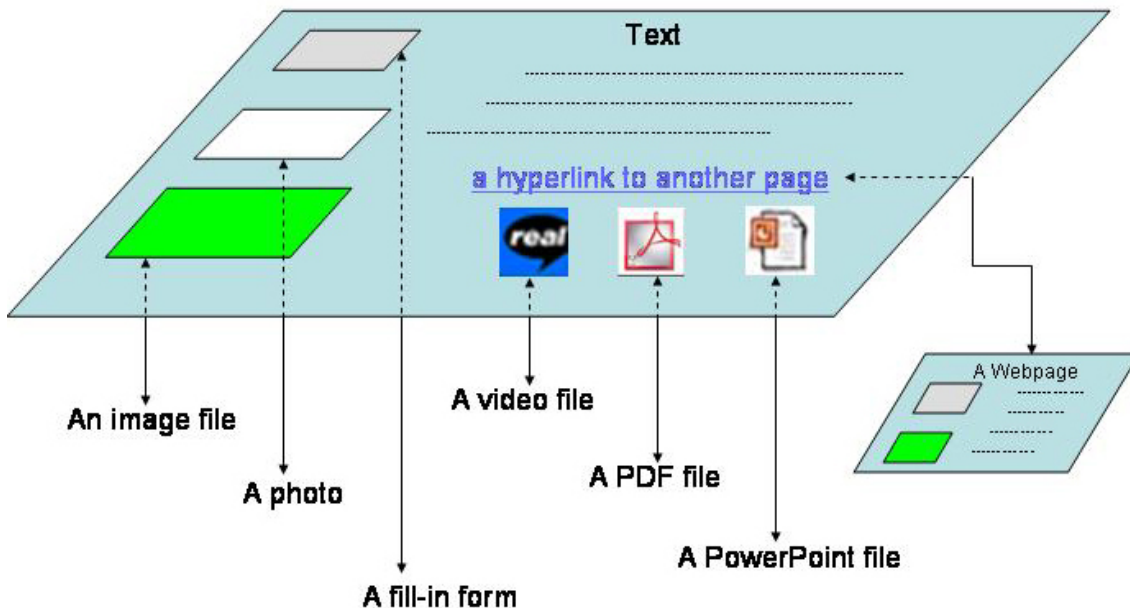


Figure 1

Since one webpage may have several files linked to it, a website may have hundreds. Making sure that you, as the designer, know where all the links live on your server and that they are functional is a daunting task. That is why a good site structure is very important to maintain good organization of your site.

This tutorial will introduce how to logically organize and manage difference files for Web publishing and maintaining.

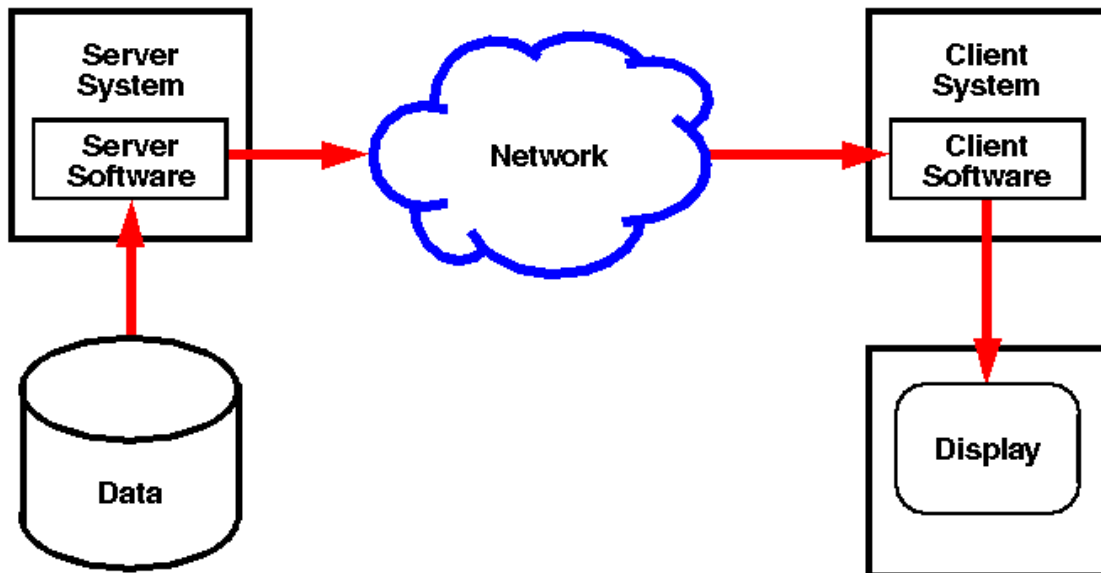
Definitions

Host: A host is any computer directly connected to a network that acts as a repository for services (such as e-mail, FTP, or World Wide Web) available for other computers on the network. The host name in our school is **login.ischool.utexas.edu**.

Server: A server is usually a computer that provides the information, files, Web pages, and other services to the client that logs on to it. A web server accommodates requests from users, handles requests for data, e-mail, file transfers, and other network services from other computers. In order to make all the WebPages viewable by the browsers, you must transfer the pages to the server by using client software, such as SSH, FTP, or Fugo.

Client: A software application that works on your behalf to access a service from a server somewhere on the network. A client is usually on your desktop computer or on a host that you access via Telnet. In the context of Internet connection sharing, client computers rely on a server computer or a router to provide ("serve") their Internet connection.

Client-server relationship (Figure 2)



Naming Files

When you create a Webpage, you should get into the habit of naming your files in a way that makes them easy to find and group similar files in a directory listing. These are the golden rules:

- Always use lowercase.
- Always use short and descriptive file names if you can.
- Never use weird characters in file and anchor/bookmark names.
- You can use a dash (-) or underscore (_) within file names instead of spaces (spaces will turn into “%” in browser windows and will cause problems on the Internet and other platforms).
- Try not to arbitrarily change the name of your page; other web sites may link to your old file name.

File Name Extensions

A Website may contain many files in different formats. Knowing different file extensions will help you manage your files.

.html – A basic web page. View using your web browser.

.jpg / .gif - Graphics file formats that use compression standards.

.doc - A Microsoft Word document; to view or print these files, you will need to use Microsoft Word in your local computer to open or save the file.

.pdf – Adobe Portable Document Format, for printing. Can be viewed in your browser or downloaded to your local computer, depending on your browser settings. You will need a PDF viewer to use these documents. Converting your word document into a .pdf essentially makes it a picture, so it is more difficult for people viewing your page to copy your work.

.ppt - A Microsoft PowerPoint document; to print these files, you will need to save the file to your local computer and then use Microsoft PowerPoint.

.xls – Excel document

.txt – plain text file – unlike a word processor these files have no formatting

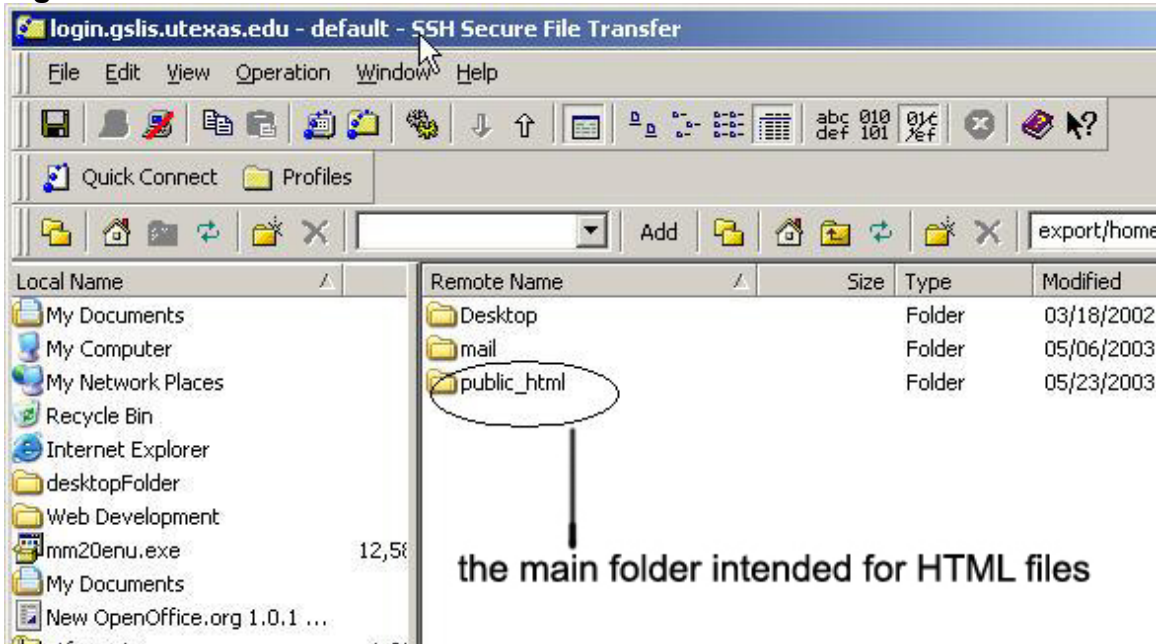
Directories and Files

Once you've created all your Web pages, you are ready to transfer your files to the Web server. Each account on the Web server has a home directory (or folder). When you login, you will be in your home directory (please reference our tutorials on SSH or Fugu to learn how to transfer your files to the Web server). Within your UNIX account you should use (or create if there is no such folder) the following:

public_html

Within your home directory you need a sub-directory called public_html. It will contain your Web pages. If you login for the first time, you need to create this directory (please reference our tutorial on how to publish Web Pages). This should be the main folder intended for HTML files. This public_html folder should contain all of your HTML files. You will create those files either with an HTML editor (such as Dreamweaver or Mozilla composer) or writing the HTML codes yourself.

Figure 3



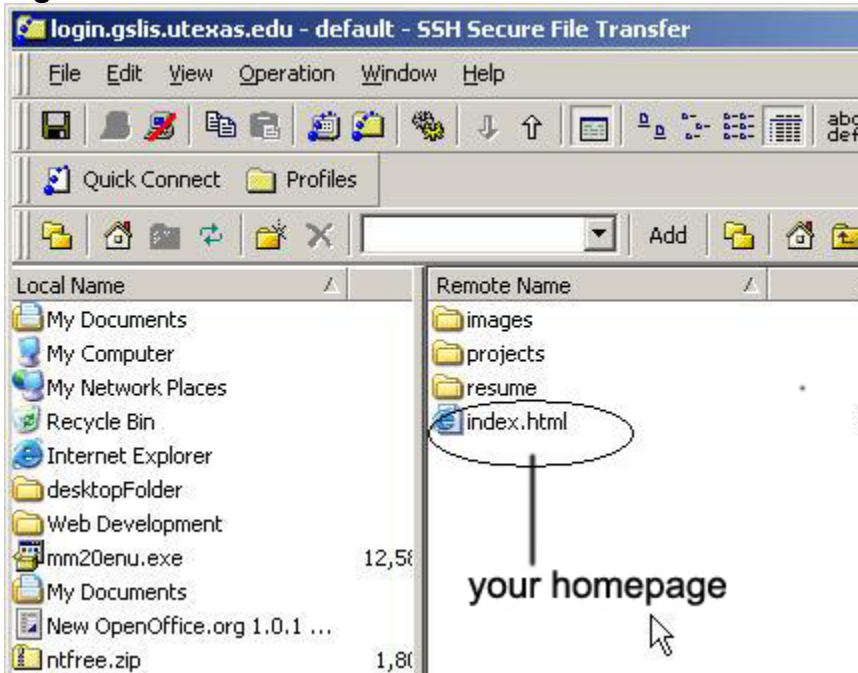
index.html

Within public_html you should make a file called **index.html**. This will be your home page and the URL will be:
www.ischool.utexas.edu/~yourusername.

Images

You need to have a sub-directory named **images** in your **public_html** directory that contains all the images referenced in **index.html** and other Web pages contained in **public_html**. You can put all of your general graphics files, the buttons, logos, background images, photos, etc. in the images folder. Linking to the images in these folders has a different method. To link to a folder inside your main folder (or, in other words, to link "forward") your link must start with the name of the folder and a slash. A link to an image in your images folder, then, would look like this: `images/picture.jpg` (or `images/picture.gif`). We'll discuss how to link files in detail at the following part of the tutorial.

Figure 4

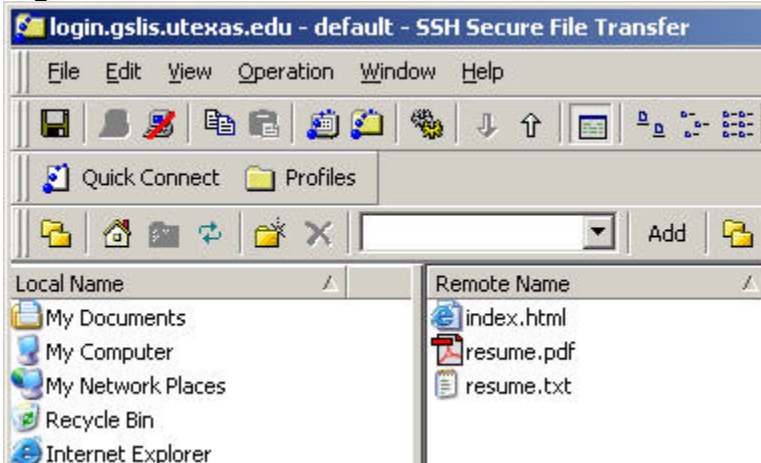


sub-directories

If you're building pages on your resume and the class projects, these should be kept in their appropriate sub-directories.

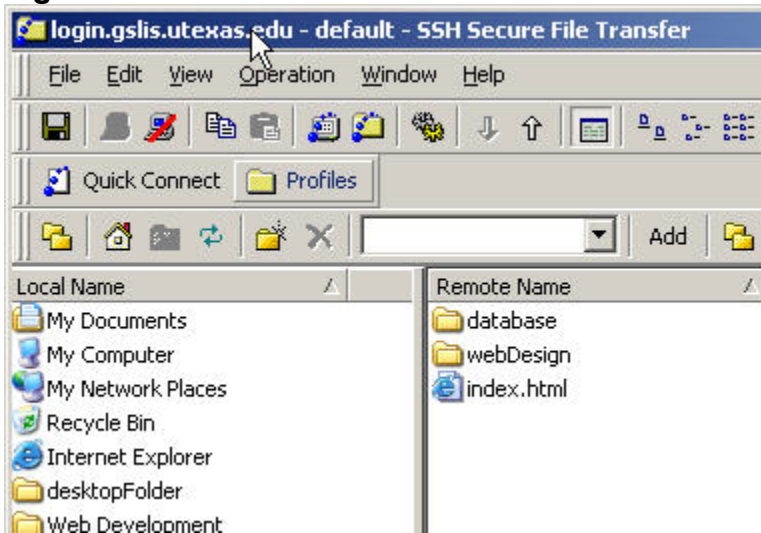
1. You should have a sub-directory within your `public_html` directory called **resume**, and **projects** (see Figure 4).
2. You will put different formats of your resume, for example, HTML (.html), PDF (.pdf), and straight text (.txt) in **resume** sub-directory.

Figure 5



3. Within the **projects** sub-directory, you should create an **index.html** page that presents information on projects and you should put all your projects into this folder.
4. If you have many projects and you want to categorize them, within **projects** sub-directory, you may create several sub-directories. For example, you can create **webDesign** sub-directory within **projects** sub-directory to save all your projects related to Web Design; or you can create a sub-directory called **database** within **projects** sub-directory and you can put all your database projects into this sub-directory.

Figure 6



The whole structure, displayed graphically, would look like this:

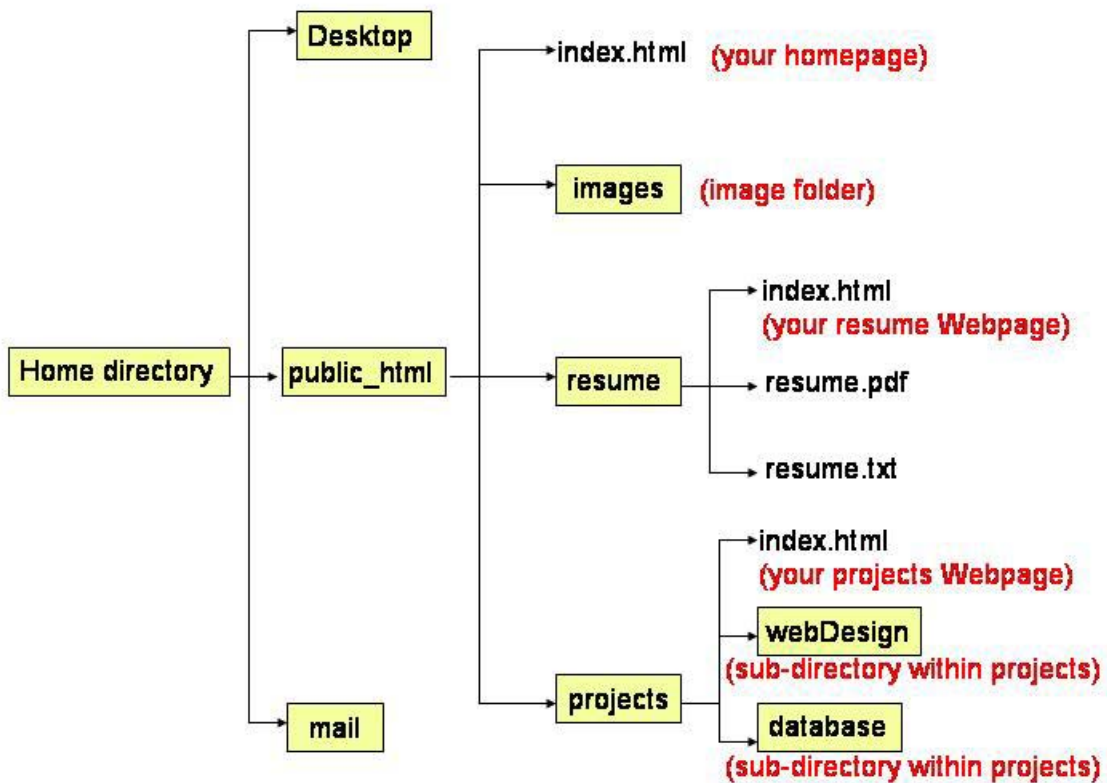


Figure 7

Linking Folders

When you design web pages, you need to know how to make links to other files located in different folders. Web designers usually use **relative links** when linking to other sections of the Web site that are under their control, and **absolute links** when they cannot control the site. For example, the relative link from your homepage to your resume page will be **resume/index.html**; the absolute link to our iSchool homepage is **http://www.ischool.utexas.edu/**

Here we will mainly focus on making relative links to WebPages designed by user self.

Linking Inside a Folder

To link to the folders from the main folder, the relative link would look like this: **"foldername/filename "**.

For example, if you want to link to a graphic file in the images folder from homepage, the link should be “images/filename.jpg or images/filename.gif” (depends on what file format it has). If you have a folder called resume which contains PDF format of your resume, and you want to link to it from the main folder, the link should be “resume/resume.pdf”

Linking Outside a Folder

Lets say your main folder is inside another folder and you want to link to a file outside. Your link would look like this: **../index.html**.

For example, if you want to provide a link on your resume page to the homepage, the link should read “**../index.html**”

Notice the “../” That's a code that tells the browser it should get out of your folder one time and then access files in the folder where your folder is located. Every time you type “../” you're telling the browser to go outside one folder. So lets say your link looks like this:

../../index.html

That means you're telling the browser to look for the file index.htm two folders outside yours (or “backwards.”)

This is important to remember because if your page is part of a larger project and you need to access a main interface you'll need the “../” to get out of your folder.

* Please notice that different file formats have different file suffix. You need to pay attention to file suffix when you link different files. Please reference **File names extensions** in this tutorial.

Linking Files in Same Folder

If you want to link files that are located in the same directory, the link is simply the file name itself. For example, if you want to link from resume homepage to resume.pdf file, the link should read “**resume.pdf**”, because the resume page and resume.pdf file both are in resume directory.

Your file URL

To make all the files viewable by the browsers, you must transfer them to the public_html directory first. Then, what is the URL for a particular file in your public_html directory? Let's use **Figure 7** as an example:

- The URL for your homepage (index.html) under public_html folder will be **www.ischool.utexas.edu/~yourusername/**
- The URL for your resume page (index.html) under resume sub-directory will be **www.ischool.utexas.edu/~yourusername/resume/**
- The URL for your projects page (index.html) under projects sub-directory will be **www.ischool.utexas.edu/~yourusername/projects/**

Warning:

You will have several files called index.html. Be careful not to mix them up and overwrite one with the other.

- The URL for any file in the sub-directory within public_html directory will be **www.ischool.utexas.edu/~yourusername/sub-directoryname/filename**. For example, if you uploaded a picture called flower.jpg into images folder, the URL will be **www.ischool.utexas.edu/~yourusername/images/flower.jpg**; same as the files in your resume sub-directory. The URL for the file - **resume.pdf** will be **www.ischool.utexas.edu/~yourusername/resume/resume.pdf**.

Cheat Sheet

Confused again? Remember the following:

- Webpages are a series of independent files put together by a web browser.
- A web browser puts the files together by reading commands from HTML files.
- HTML files are simple text files with commands telling the browser how to organize text and image files in a page.
- You need to have a main folder for HTML files
- Inside that folder you need an images folder for your GIFs and JPEGs
- To link to the folders from the main folder the relative link should read "foldername/filename "
- To link outside the main folder (to "get out") your links should read "../filename.htm."
- Each "../" indicates one level back in the folder structure.