

24.1 Introduction

Hypertext markup language (HTML) consists of a set of “tags” that tell a web browser how a web page should look and operate. For example the pair of tags `Introduction` tells a browser to display the word “Introduction” in a “strong” format (typically bold).

There are couple of important things to notice about this example:

1. The tags themselves are text. HTML is a text-only standard in which some of the text in a document is content and the rest of the text is used to tell the browser how to display the content. The angled braces “<” and “>” are use to differentiate HTML tags from the rest of the document.
2. It is up to the browser to interpret the tags and render the text accordingly. Although this approach minimizes the amount of information that must be transferred over the network, it leads to some inconsistencies. For example, a tag such as `` may mean one thing in NETSCAPE NAVIGATOR and something slightly different in MICROSOFT’S INTERNET EXPLORER. The result is that you, as the HTML author, have

incomplete control over how your page looks when viewed in different browsers.



HTML and HTML extensions (such as Dynamic HTML) are evolving in two areas: new functionality and greater control over how the document looks in the browser. Although HTML is nominally a standard, new tags and features (which may not be supported by all browsers) are continually being introduced. When in doubt, check the official standard at www.w3.org.

Although general purpose document authoring tools (like MICROSOFT WORD) and special WYSIWYG¹ HTML editors (like MICROSOFT FRONTPAGE) can write HTML for you, and although these tools are improving, there are still occasions when you must open the hood and deal with the HTML directly. In this lesson, we will use a simple text editor and write all our HTML from scratch.

¹ What You See Is What You Get



24.2 Learning objectives

- understand the structure of an HTML document
- use HTML tags to format web-based content
- create hyperlinks between two documents
- use HTML tables to display tabular data and format pages

24.3 Exercises

In this lesson, you will learn about a few basic tags and create some very simple HTML pages.



If you are interested in becoming a *real* web page designer, there are many HTML tutorials and resources on the Internet. We are merely skimming the surface here.

24.3.1 Tag basics

Most tags are found in pairs. For example, the tag `` means that all text which follows will be emphasized (italicized in most browsers). When the closing tag, ``, is encountered, the emphasis is turned off. The closing tag in the pair always starts with a slash.

It is possible to nest tags. For example, to get text that is strong *and* emphasized (bold and

italic in most browsers), you would nest the tags in the following manner:

```
NL <STRONG><EM>This text shows as bold
    italics is most browsers.
NL </EM></STRONG>
```



The amount of whitespace does not matter in HTML—there is either “no space” or “one or more spaces.” The latter is simply interpreted by browsers as a single space. As a result, the HTML below is equivalent to the previous example:

```
NL <STRONG>
NL   <EM> This text shows
NL       asbold italics in most
NL
NL   browsers.</EM>
NL </STRONG>
```



HTML tags are case-insensitive. Thus, the tags `` and `` are identical. Naturally, the capitalization of your content does matter.

24.3.2 HTML documents

An HTML document is simply an ASCII text file with a `htm` or `html` extension.¹ Within the file,



there should be three pairs of tags which define the following sections:

1. The overall HTML document – everything between the `<HTML>` and `</HTML>` tags.
2. The **header section** within the HTML document – everything between the `<HEAD>` and `</HEAD>` tags. The header section contains information about the document and is not visible in the browser.
3. The **body section** within the HTML document – everything between the `<BODY>` and `</BODY>` tags. The body section contains the visible content of the document.

Thus, the structure of an empty HTML document is:

```
NL <HTML>
NL   <HEAD>
NL   ...
NL   </HEAD>
NL   <BODY>
NL   ...
NL   </BODY>
NL </HTML>
```

¹ MICROSOFT DOS and WINDOWS 3.x can only handle three-letter extensions. As such, MICROSOFT continues to push the htm naming convention. However, since the Internet grew up on UNIX (which has always supported longer filenames), the four-letter html extension is more common.

24.3.2.1 Preliminaries

Web applications typically run on specialized web servers with high-speed fixed connections to the Internet. Web development, in contrast, typically takes place on the developer's lowly PC. In this scenario, you are the developer. In order to get your content and programs on to the web server, you have to transfer (or "publish") your HTML files over the network to your ISP's machine¹. The transfer is normally accomplished using a utility program based on the FTP (file transfer protocol) standard.



The constant transfers can become a bit tedious as you continually create, upload, test, and revise your content. For this reason, a good FTP client is a necessity. Alternatively, you may want to enable a web server on your desktop machine. In this way, you can create and test your application on a single machine and transfer the files to the "production" web server when the files are complete. Enabling a local web server is discussed in additional detail in [Section 24.4.2](#).

The important thing to keep in mind is that you will always have two copies of the files you

¹ ISP stands for Internet service provider. ISPs typically offer two broad class of services: access for dial-up users and hosting of content on their servers.



create: the **local copy** that you work on during development and the **published copy** on the production web server that the whole world can access.



Although maintaining two identical copies of every file in your web application can be tedious, at least you can use your local copy as a backup if something happens to your ISP's server (and vice-versa).

➔ Set up a directory on your machine for the local copy of your web application.



It is up to you how you organize your directory. However, it is customary to create a "images" subdirectory and a "private" subdirectory. The images subdirectory contains the files for graphic elements (pictures, icons, buttons, etc.). The private directory can contain resources (e.g., a database file) that outside users are prevented from accessing directly.

24.3.2.2 Creating a document

➔ Use NOTEPAD or some other text editor to create a document with the `<HTML>`, `<HEAD>`, and `<BODY>` tags discussed in [Section 24.3.1](#)



You cannot use a word processor like MICROSOFT WORD to edit HTML directly because word processing programs save documents in proprietary formats rather than plain ASCII text. Although you can use the **Save As** command to explicitly save the file as plain text, most people find it is easier to simply use NOTEPAD or some other text editor.

➔ Save the file as `MyFirst.html`, as shown in [Figure 24.1](#).

➔ In the document's header, add a title:

```
NL <HEAD>
NL <TITLE>Kitchen Supply Co.
   Extranet</TITLE>
NL </HEAD>
```



Although the header section is optional and does not show in the browser window, Internet search engines often use information in the header when indexing and display search results. As such, you should always provide a meaningful title for each of your pages.

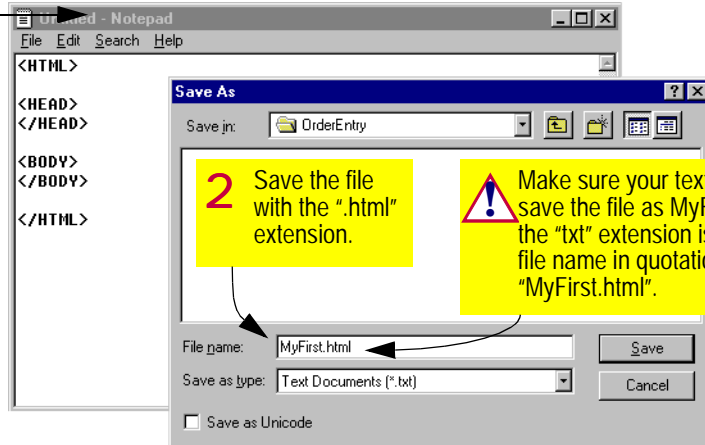
➔ In the document's body, add a heading and welcome message to users, for example:

```
NL <HTML>
NL <HEAD>
```



FIGURE 24.1: Save the skeleton HTML document created in NOTEPAD.

1 Use a plain text editor (such as MICROSOFT NOTEPAD) to write the core HTML commands.



```
NL <TITLE>Kitchen Supply Co.
  Extranet</TITLE>
NL </HEAD>
NL <BODY>
NL <H1>Welcome to the Kitchen Supply
  Co. Extranet</H1>
NL Please login to gain secure access
  to the system.
NL </BODY>
NL </HTML>
```

➔ Save the changes. Your file should resemble Figure 24.2.

24.3.2.3 Viewing the document

HTML pages are typically transferred to users via a web server. However, it is possible to bypass the web server and open local files directly with your browser.

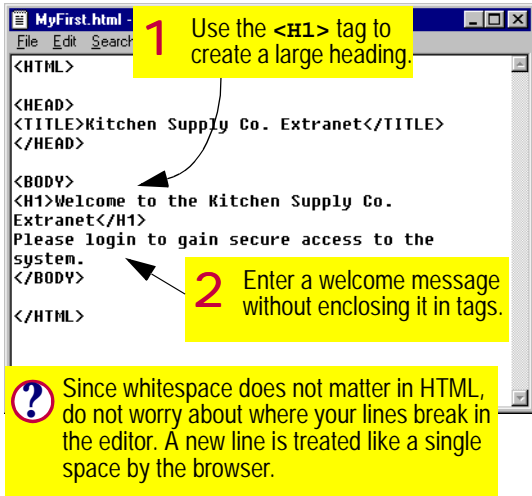
➔ Open your browser and select **File** → **Open** from its main menu.



Since there are many different makes and models of browsers in circulation, it is impossible to give specific menu and



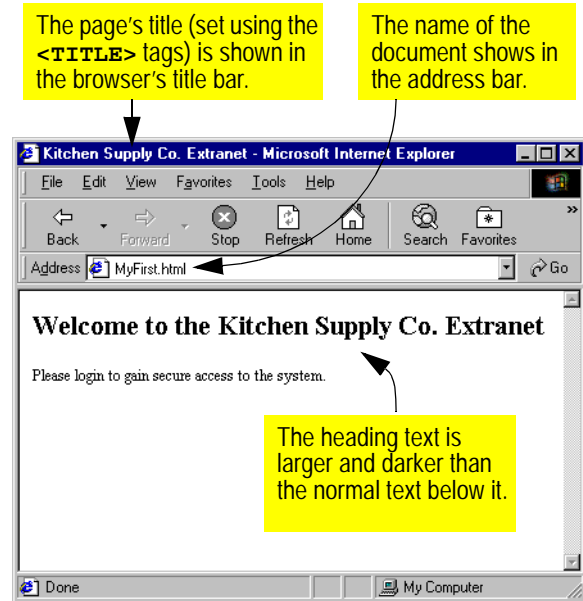
FIGURE 24.2: Add a welcome message to the HTML document.



keystroke commands for browser-based activities. In this lesson, MICROSOFT INTERNET EXPLORER version 5.0 is assumed. If you use a different browser, the command sequences and format of the output shown here may differ slightly from what you see on your computer. However, the underlying principles are the same regardless of the browser used.

- ➔ Navigate to the **MyFirst.html** file and open it. Your page should appear in the browser, as shown in Figure 24.3.

FIGURE 24.3: Preview the HTML document in a web browser.



- ? There are many ways to open a local HTML file in your browser, including double-clicking the file's icon in WINDOWS



EXPLORER or dragging the file's icon onto the browser.

When you compare [Figure 24.2](#) and [Figure 24.3](#), you should get the basic gist of HTML: plain text is rendered into formatted text via the use of special tags.

24.3.3 Adding hyperlinks

Of course, the real power of HTML is not its formatting, but rather its ability to create hyperlinks to other documents and resources on the World Wide Web (WWW). In this section, you are going to create a new page (a list of products) and create a hyperlink to **MyFirst.html**.

- ➔ Without closing the current instance of NOTEPAD, open a second instance in a different window (i.e., select **Start** → **Programs** → **Accessories** → **Notepad** from the WINDOWS taskbar).
- ➔ Cut and paste the HTML from **MyFirst.html** into the new document and save it as **ProductList.html**.
- ➔ Edit the heading to reflect the fact that this page is a list of products for KITCHEN SUPPLY Co.

- ➔ Delete the welcome message and enter the following without tags, as shown in [Figure 24.4](#).

```
NL <BODY>
NL <H1>Kitchen Supply Company: List of
   Products</H1>
NL Go to login page
NL </BODY>
```

To create a hyperlink, you need to know the Uniform Resource Locator (URL) of the target document. The URL consists of three items of information:

- the protocol to use to access the information (e.g., HTTP, FTP, Telnet, and so on);
- the Internet Protocol (IP) address of the machine on which the document or resource resides (e.g., **mis.bus.sfu.ca**); and,
- the name of the document or resource (e.g., **MyFirst.html**)

When the target of the hyperlink is located on the same server as the document containing the hyperlink, the URL can be replaced with the **relative path** and filename of the target.



IP addresses are numeric. For example, to access the DELL COMPUTER site, you type **http://143.166.82.178**. Fortunately, you can also use the easier-to-remember



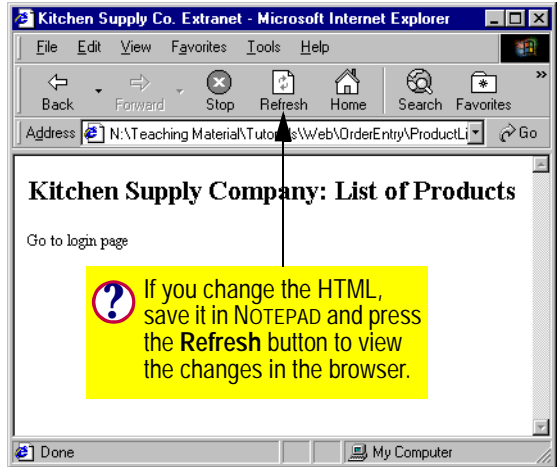
FIGURE 24.4: Create a new page for showing a list of products.

1 Create a new page by cutting and pasting from your existing page.

3 Preview the new page in your browser.

```
<HTML>
<HEAD>
<TITLE>Kitchen Supply Co. Extranet</TITLE>
</HEAD>
<BODY>
<H1>Kitchen Supply Company: List of
Products</H1>
Go to login page
</BODY>
</HTML>
```

2 Add text that can be converted into a hyperlink.



textual address (**www.dell.com**) thanks to a world-wide distributed database called the **Domain Name System** (DNS). A DNS lookup translates a textual IP address into its numeric equivalent before sending the request out on the network. The administrator of the **dell.com** domain is responsible for assigning textual names to

numeric IP addresses and storing this information in a local DNS database. For example, DELL has one IP address registered as **www.dell.com** and another registered as **support.dell.com**.



- Use the anchor tag and the **HREF** attribute to transform the plain text in Figure 24.4 into a hyperlink:

NL <BODY>

NL <H1>Kitchen Supply Company: List of Products</H1>

NL Go to login page

NL </BODY>

- Save the document and press the browser's refresh button to view and test the hyperlink, as shown in Figure 24.5.

FIGURE 24.5: Create a hyperlink to another HTML file in the same directory.

1 Surround the visible text with an anchor tag.

2 Preview the hyperlink in the browser.

The image shows a Notepad window on the left and a Microsoft Internet Explorer window on the right. The Notepad window displays the HTML code for 'ProductList.html'. The code includes a title 'Kitchen Supply Co. Extranet' and a body with an h1 heading 'Kitchen Supply Company: List of Products'. An anchor tag is added: `Go to login page`. A yellow circle highlights the text 'Go to login page' within the anchor tag. A yellow callout box with a question mark icon explains: 'Many HTML tags support the use of attributes within the tag. In this case, the anchor tag <A> has an HREF attribute/value pair that creates a hyperlink.' The Internet Explorer window shows the rendered page with the title 'Kitchen Supply Company: List of Products' and the text 'Go to login page' underlined in purple. A yellow callout box points to this text, stating: 'The text within the anchor tags becomes highlighted as the hyperlink.'

If the target document is in a subdirectory relative to the current document, then the name of the subdirectory has to be included:

NL



In UNIX, directories are separated by slashes (“/”) rather than back-slashes (“\”) as in MS-DOS and WINDOWS. The convention in HTML is to use UNIX-style directory names for URLs, regardless of the platform you are using.

If the target document is in the current document’s parent directory, the double-dot notation has to be used to ascend one level in the directory tree:

```
NL <A HREF="../MyFirst.html">
```



The notation “..” is shorthand for the parent of the current directory.

If the target document is on a different machine, then the full URL (including the machine name and directory) has to be used:

```
NL <A HREF="http://mis.bus.sfu.ca/
pages/MyFirst.html">
```

24.3.4 The paragraph tag

Since whitespace is ignored in HTML, the only way to create space between paragraphs is to use the paragraph tag, <P>. In the original HTML standard, one could use the <P> tag without a closing tag. However, the use of opening and closing tag is consistent with other HTML tags and is therefore preferred.

➔ Nest a <P> tag inside the hyperlink tag:

```
NL <BODY>
```

```
NL <H1>Kitchen Supply Company: List of
Products</H1>
```

```
NL <A HREF="MyFirst.html"><P>Go to
login page</P></A>
```

```
NL </BODY>
```

24.3.5 Using HTML tables

Tables are used within HTML to

- format tabular data, and
- provide page designers with additional control over the layout of the page.

At this early stage, you should not concern yourself with the finer points of page layout. Instead, our focus is on using tables to display lists of data.

➔ Add a <TABLE>... </TABLE> pair underneath the hyperlink.

24.3.5.1 Rows and headings

A table consists of one or more rows and each row consists of one or more cells. Rows are designated using the <TR> tag and cells are designated using the <TD> tag. Special heading cells (typically bold and centered) are designated using <TH> tags.



- Add a row of headings to the table:

```
NL <BODY>
NL ...
NL <TABLE>
NL   <TR>
NL     <TH>Product ID</TH>
NL     <TH>Description</TH>
NL     <TH>Unit</TH>
NL     <TH>Price</TH>
NL   </TR>
NL </TABLE>
NL </BODY>
```

- Add a second “detail” row that contains product information:

```
NL <BODY>
NL ...
NL <TABLE>
NL   <TR>
NL     ...
NL   </TR>
NL   <TR>
NL     <TD>51 5012</TD>
NL     <TD>Water jug, s.s. w/ice guard,
NL       2 litre</TD>
NL     <TD>EA</TD>
NL     <TD>$23.50</TD>
NL   </TR>
NL </TABLE>
NL </BODY>
```

- Save the document and preview it in your browser, as shown in [Figure 24.6](#).

24.3.5.2 Adding tag attributes

It is possible to use tag attributes to control the format of the table and elements within the table:

- Change the format of the table so that it has solid borders and inserts some padding (space) between cells:

```
NL <BODY>
NL ...
NL <TABLE BORDER="1" CELLPADDING="5">
NL   ...
NL </TABLE>
NL </BODY>
```

- Add `ALIGN="right"` to the `<TH>` tag used for the “price” heading. Add the same attribute to the `<TD>` tag used for the price body rows.



If the attribute value is a single word or number (i.e., if the value contains no spaces), then quotation marks are not required. Thus, it is possible to use `ALIGN=right` in your `TABLE` tag without having bad things happen. However—as you will see in [Lesson 25](#)—it is possible to have attributes with spaces (e.g.,



FIGURE 24.6: A basic table to show product information.

1 Insert a paragraph tag around the hyperlink.

2 Add a table row containing headings.

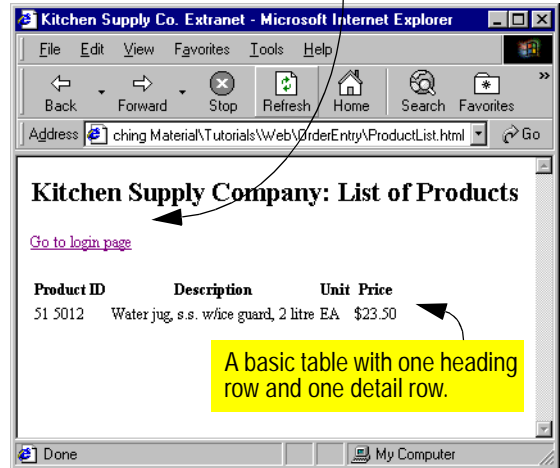
The paragraph tag puts some space between the hyperlink and the table.

```

ProductList.html - Notepad
File Edit Search Help
<A HREF="MyFirst.html"><P>Go to login
page</P></A>
<TABLE>
  <TR>
    <TH>Product ID</TH>
    <TH>Description</TH>
    <TH>Unit</TH>
    <TH ALIGN="Right">Price</TH>
  </TR>
  <TR>
    <TD>51 5012</TD>
    <TD>Water jug, s.s. w/ice guard,
      2 litre</TD>
    <TD>EA</TD>
    <TD ALIGN="Right">$23.50</TD>
  </TR>
</TABLE>
</BODY>

```

3 Add a second table row containing product details.



A basic table with one heading row and one detail row.

VALUE=51 5012). Without quotation marks, the browser will assume the attribute **VALUE** is equal to "51" and will try to interpret "5012" as a different attribute. To avoid such problems, it is considered good practice to put *all* attribute values in quotation marks.

➔ Save the changes and view the document in the browser, as shown in Figure 24.7.



FIGURE 24.7: The table with some formatting attributes specified.

A visible border has been added to the table and the amount of space within cells has been increased.



different tools that can do most of the slug work for you.

24.4.1.1 Dedicated HTML editors

There are many commercial and shareware WYSIWYG HTML editors that are about as easy to use as modern word processors. The editors allow you to use the mouse to format text, create tables, and so on. In addition, some allow you to “round trip”—that is switch back and forth between WYSIWYG and raw HTML modes. Examples of commercial products in this group include MICROSOFT FRONTPAGE, ADOBE PAGEMILL, MACROMEDIA DREAMWEAVER, and ALLAIRE HOMESITE.

24.4.1.2 Word processors

In addition to dedicated HTML editors, word processors themselves are getting better at saving documents in HTML format. The advantage of using a word processor is that you can continue to work within a familiar environment and exploit features that tend to be weak in dedicated HTML editors (for example, spell checking and table editing).

However, it is important to realize that word processors typically support much richer formatting and page layout options than HTML is currently capable of expressing. As such, the limitations of HTML should be kept in mind

24.4 Discussion

24.4.1 Authoring options

Clearly, writing HTML by hand is tedious and error-prone. Fortunately, there are many



when authoring documents. Even common enhancements, such as embedded graphics, footnotes, and special formatting can overwhelm the HTML translator and result in an HTML document that bears little resemblance to its source. In addition, most word processors are not very good at round tripping. Adding non-standard HTML tags or scripting code in HTML mode can cause problems when you switch back to word processing mode.



You have to be careful to select an authoring tool that suits your purposes. Many of the tools designed for beginners or casual users take it upon themselves to “fix” your code. If you are a developer and are using scripts and advanced tags, you may find that the tool obliterates your work in its attempt to be helpful.

24.4.1.3 Application development suites

A third way to create web content is to use an “integrated application development suite”. This is an ill-defined, emerging class of tools that includes MICROSOFT VISUAL INTERDEV and NETOBJECTS FUSION (MICROSOFT FRONTPAGE and ALLAIRE HOMESITE/COLDFUSION might also be included). Application development suites are much more than HTML editors; they include tools for site management, creation of dynamic content and scripting, database integration,

and support for team-based design and implementation. Although these tools are very powerful, they are also complex. Using them effectively requires an organization-wide commitment to the development methodology advocated by the suite.

24.4.2 Setting up a local web server

A web server is simply a program that runs on a computer and “listens” for requests from other programs. In subsequent lessons, access to a MICROSOFT web server is required in order to execute server-side scripts. Fortunately, a full-featured web server may be sitting on your desktop right now.

All versions of WINDOWS since WINDOWS 95 include a scaled-down version of MICROSOFT’s flagship web server—INTERNET INFORMATION SERVER (IIS)—on the installation disk. Depending on your version of WINDOWS, the bundled copy of IIS may be referred to using an alias such as “peer web services” or “personal web server”.



The terms “personal” and “peer” are intended to drive home the point that the web server included with WINDOWS is meant for development work or as a very small server for a network of colleagues and friends. For industrial-scale web applications, MICROSOFT sells its full



version of IIS as part of the BACKOFFICE suite.

24.4.2.1 Advantages of using a local web server

If you are using these tutorials as part of a course and your instructor has set up a web server for you, there is no strict requirement for you to set up a local web server. However, you might want to consider the advantages below before skipping the remainder of this section. For those who do not have an instructor taking care of you, read on.

There are a number of important advantages to having a local web server up and running when you are developing a web application:

1. You do not need a persistent, high-speed connection to the Internet to test your pages.
2. You can edit your files directly on the web server, thereby eliminating the upload step from the create → upload → test → revise cycle.
3. If, during testing, your web application does something so bad that you need to restart the server (not unheard of, unfortunately), you do not impact the other users of the server.

24.4.2.2 Is IIS running already?

Some programs, like MICROSOFT FRONTPAGE, configure the local web server behind the scenes when they install. To see whether IIS is already running on your machine, type the following into the URL window of your browser:

http://localhost



The IP address of the local machine is always 127.0.0.1. A small file called **hosts** in your Windows folder maps this numerical address to “localhost”.

If a web page, a blank page, or any page that does not contain errors pops up, IIS is already installed and configured on your computer. If you get an error or get shunted off to a search engine looking for “localhost”, then IIS is not setup on your computer.

24.4.2.3 If IIS is not running

If the results of the localhost test are negative, then you have to have to install IIS. The good news is that it is a very simple procedure. The bad news is that the procedure itself depends greatly on which version of WINDOWS you are running. As such, the first place to start is a search for terms like “web browser” and “peer web services” in the WINDOWS help system



(Start → Help). At a very general level, the install procedure involves the following:

1. Ensure you have the TCP/IP networking protocol installed (which you do if you use your computer to access the Internet).
2. Install the web server from your WINDOWS installation media. The sequence for installing IIS on a WINDOWS 2000 machine is shown in Figure 24.8 and Figure 24.9.

FIGURE 24.8: Install and administer the IIS web server on your desktop machine (Part 1).

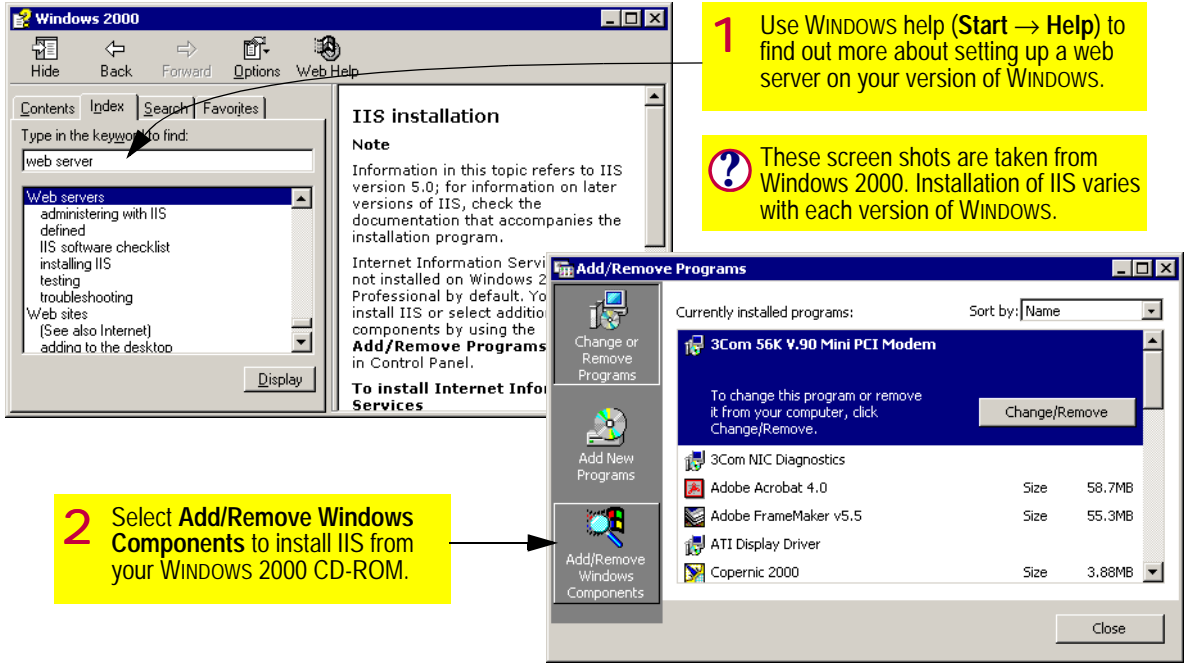
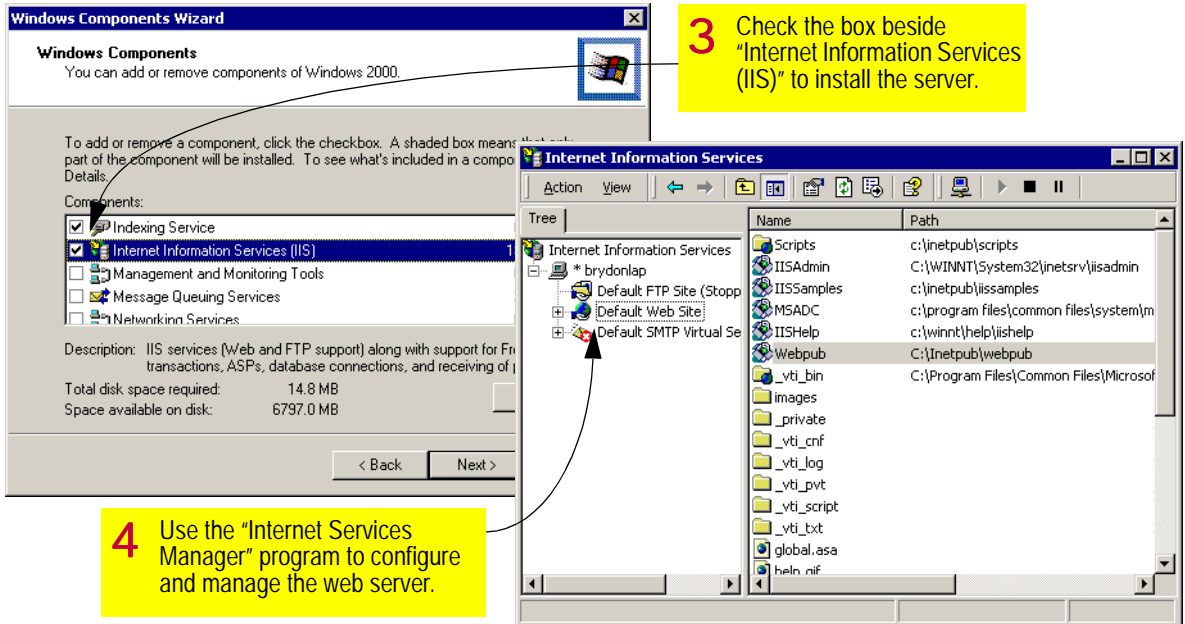




FIGURE 24.9: Install and administer the IIS web server on your desktop machine (Part 2).



3 Check the box beside "Internet Information Services (IIS)" to install the server.

4 Use the "Internet Services Manager" program to configure and manage the web server.



If you are running an older version of Windows (pre 98), you may want to search on the MICROSOFT site for and upgraded version of IIS.

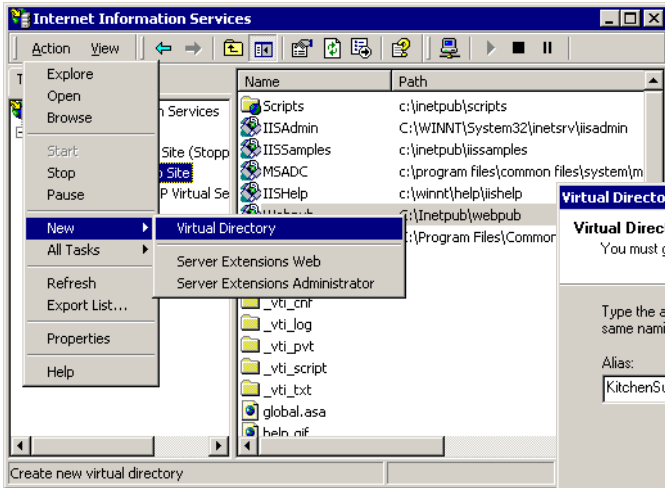
3. Find the web server's administration tool. It is normally named "Internet Services

Manager" or something similar. The tool will allow you to set up virtual directories, set permissions on the directories, start the web service, and so on. Figure 24.10 and Figure 24.11 show the procedure for creating a virtual directory called `http://localhost/KitchenSupply` in a physical

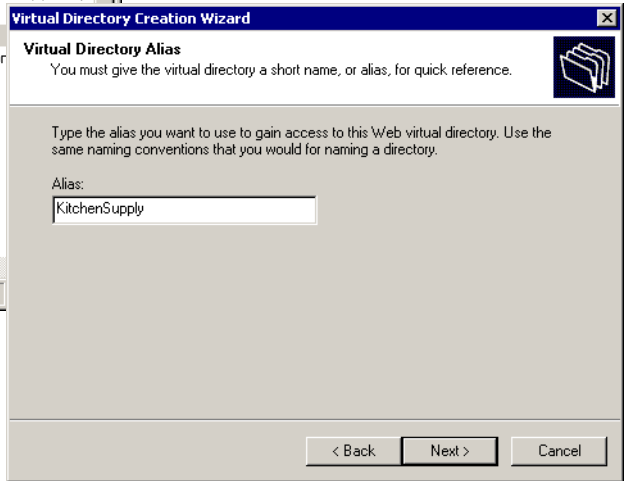


directory called **C:\Documents and Settings\My Documents\KitchenSupply**.

FIGURE 24.10: Create a virtual directory on the web server (Part 1).



1 Use the Internet Services Manager to create a new virtual directory called (for example) "KitchenSupply"



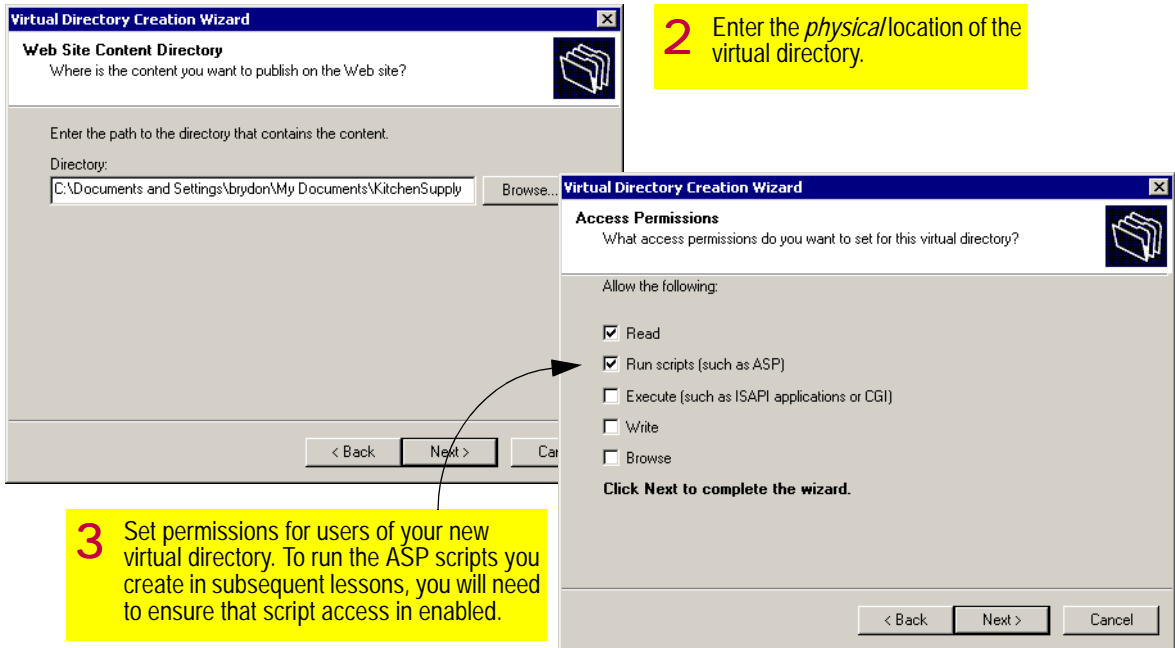
? A number of virtual directories are set up by default when you install the web server. For example, full documentation for IIS can be viewed by typing **http://localhost/IIShelp** into your browser.

? You can use any directory names you like; these are simply examples.

4. Put your content in a physical web server directory and create a virtual directory to access the files.



FIGURE 24.11: Create a virtual directory on the web server (Part 2).



24.5 Application to the project

24.5.1 Application structure

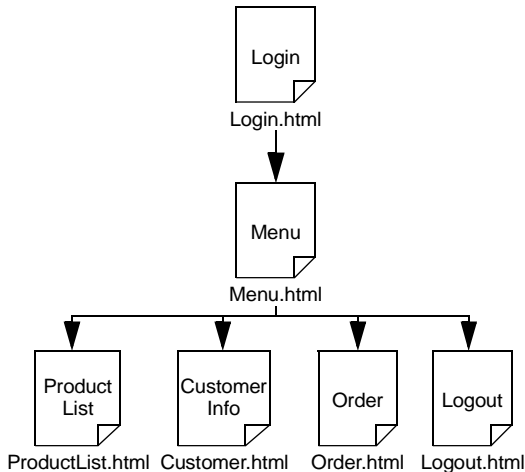
For the simple web-based order entry system you will create in subsequent lessons, you will

need to start with six pages with HTML content. The structure of the application and HTML pages is shown in [Figure 24.12](#).

Since only authorized users are permitted to place orders, the point of entry is a login page. The login page requires the user to enter a



FIGURE 24.12: The structure of the web-based order entry system.



username and password. If the username/password pair are valid, then the user is transferred to a main menu. From the menu, the user can

- view a list of products,
- update her customer information (e.g., contact person, shipping address, and so on),
- place or view an order, and

- logout.

At this early stage, you should concentrate on creating the basic documents. Dynamic content, hyperlinks, and form elements will come later.

- ➔ Create the HTML documents shown in [Figure 24.12](#). Each document should have a header and body section.
- ➔ Add meaningful titles (using the `<TITLE>` tag) to the header sections of your documents.
- ➔ Use the heading tags (e.g., `<H1>`) to add visible headings to all your pages.



It is possible to spend an enormous amount of time formatting your pages. At this point, you are *not* encouraged to invest much effort in making your pages look good by adding additional tags, backgrounds, images, and so on.

24.5.2 Local web server

- ➔ Review [Section 24.4.2](#) to determine whether there is a web server installed and configured on your machine.



➔ If no web server is installed and it is practical to install one, consult the WINDOWS documentation for instructions on how to install a scaled down version of IIS.

⚠ If a web server has not been set up for you, you will need to install a local server before you can continue with [Lesson 25](#).