

Problem

Code a Web Address Checker (WAC) program to check and parse the file format for the Netscape Navigator® bookmark file for a Web address (URL) management system.

Discussion

The WAC program must read the title for the bookmarks file, bookmark folder information and Web URLs. (See the input section for a discussion of the bookmark file contents.) Only required information from the bookmark file must be checked and validated. Any entry missing required data will not be stored, but will be reported to the user. The bookmarks folder and Web URL information must be stored in separate arrays of records. The interface for this program will be extremely simple, consisting of a query and answer dialog with the user. This program will later be incorporated into a Web address (URL) management system.

Execution

The WAC program should initially present the user with a startup screen displaying the name of the program, a brief 3-4 line explanation of the program and the programmer's name and email address. After the user hits the return key, the startup screen is to be cleared. A brief help screen should appear explaining the program, which is cleared upon the pressing of the return key.

Following the brief help screen the user is prompted for the name of the Navigator bookmark file. The name of the bookmark file is input and opened for reading. By default the program is to assume that the bookmark file is located within the current directory. (The user is responsible for entering a path name specification if the file is located in a different directory.) If the bookmark file does not exist, the program is to halt with an appropriate error message. If the bookmark file does exist, it is read, parsed and checked for format errors. For each entry of the bookmark file that is checked and found invalid, the WAC program will include a brief entry in the report file, and echo it to the screen. (See the Output section for the report file description.) After the bookmark file has been processed and all output performed, the program must issue a brief termination message before returning to the operating system.

Input

The format of the Netscape Navigator® bookmark file begins with the first four lines containing HTML comments that can be ignored. The fifth and sixth lines contains the bookmark file title, enclosed in title and header one tags respectively. The seventh line contains the first data list tag, <DL><p>, (followed by a paragraph tag), indicating the start of the unnamed root folder. All other lines will be either data term, <DT>, URL entries, data term folder entries, data list beginning tags, data list ending tags (data list tags must always be immediately followed by beginning paragraph tags, <p>, on the same line), data definition tags, <DD> (containing data term entry comments) or horizontal rule tags, <HR>. See the table below for a synopsis of the bookmark entries.

tag	contents	comments
<DL><p>	none	Data list paragraph tags always occur after a folder data term entry, (except for line 7).
</DL><p>	none	Data list end tag and paragraph tags are used to delimit the end of folder entries.
<DT><A ... > url 	URL (Web address)	Data term URL entry tag contains an anchor link tag with a Hypertext REFERENCE, (HREF).
<DT><H3 ...> folder </H3>	folder name	Data term folder entry tag contains a heading three tag with a FOLDED ADD_DATE timestamp followed by the folder name.
<DD>	comments	Data definition tags may follow any data term entry and contain user comments about the URL or folder. <DD> tags may also be empty.
<HR>	none	Horizontal rule tags are used as separators and can be ignored.

The HREF entry will be either a http, ftp, gopher, news, or mailto link. The data term entry tag will also contain an ADD_DATE, LAST_VISIT, and LAST_MODIFIED nine digit timestamp. The timestamp is in seconds and will be used in a later assignment. The LAST_VISIT, and LAST_MODIFIED timestamps may be zero in the bookmarks file.

Storage

The following information must be stored for each folder in the folder record array:

field	contents	comments
title	folder name	The title of the folder.
date	folder add date	Nine digit timestamp.
comment	optional	Possible <DD> tag user comment for folder.
findex	folder index	Integer index of the parent folder.
eindex	entry index	The integer index for all URLs stored in the folder.

The folder index will simply be an increasing integer index used to store the hierarchical folder relationship. As a new folder is encountered in the bookmarks file, the folder index of the containing folder that the new folder is located within, (i.e. the parent folder), is saved in the findex field. Also, each folder is assigned a unique entry integer index, eindex, that is attached to the URLs contained within the folder.

The information that must be stored for each data term URL entry in the Web URL information record array is given in the following table:

field	contents	comments
name	link name	title of link
url	link	Web address: (http, ftp, gopher, news, or mailto)
addDate	timestamp	Date link was added in the bookmarks.
lastVisit	timestamp	Date link was last visited.
modified	timestamp	Date link was last updated.
comment	optional	Possible <DD> tag user comment for URL.
eindex	entry index	Integer index for the URL's folder.

Output

The WAC program will produce two output files. The files must be echoed to both the screen and to output text files. The first file, the bookmarks report file, will contain error messages for bookmark entries that are missing data. The file name will be the same as the user entered bookmark file, but with a '.RPT' extension. Each Folder entry must contain a title, and date, (the optional comment is not required). Each URL entry must contain a name, Web address, and add/visit/modified date timestamps. Each line containing an error should be listed followed by a brief error message. Sample bookmarks error reporting is shown below:

```
Bookmark Error Report: BOOKMRKS.HTM
=====
<DT><H3 FOLDED ADD_DATE="885247584"></H3> --- Missing title

<DT><A ADD_DATE="822322878" LAST_VISIT="885240349"
LAST_MODIFIED="822322869">CS1704</A> --- Missing URL

<DT><H3 FOLDED>Misc URLs</H3> --- Missing add date

<DT><A HREF="http://ei.cs.vt.edu/~cs1704/"
ADD_DATE="822322878" LAST_VISIT="885240349"
LAST_MODIFIED="822322869"> </A> --- Missing name
```

The second output file will be a labeled tabular listing, neatly spaced and aligned, of all of the fields of both the folder and URL record arrays. The file name will be the same as the user entered bookmark file, but with a '.WAC' extension.

Assumptions

It may be assumed, for ease of initial implementation, that the maximum number of folder and URL entries will be ≤ 100 each. This limitation will be removed in later programs. Invalid folder entries result in their entries being included in the preceding folder. If `</DL><p>` folder ending tags are missing, it will result in the folder containing all of the remaining entries in the bookmarks file. Strings in the bookmarks file will be assumed to be no more than 100 characters in length. Characters beyond the 100th character can be ignored. This includes titles, address and comments. Data files will be provided shortly. Downloading instructions will be available from the course Web site: (<http://ei.cs.vt.edu/~cs1704>).

Structure Charts

A copy of the initial structure chart design must be submitted on **Monday Jan. 26th**. A final structure chart reflecting design changes and corresponding to the code must be produced and submitted with the executable. The initial chart will be compared against the final chart to determine the quality of the design.

Grading

The due date for this program is Wed., Feb. 4th. Arrays of struct, (records), must be used in this program. Separate compilation is also required for this program. Failure to produce a separately compiled program will result in grade penalties. Turn in hard copies of the following: source code, output files, initial and final structure charts. Submit a diskette (system labeled: 95 or NT), with files containing: ASCII source code, MS C project files, executable image, I/O files, and a brief ASCII *readme* file with execution instructions. The files on the disk may be zip compressed in a self extracting archive. The executable image must be called *wac.exe*. All deliverables must be placed in an sealed 9"x12" envelope, neatly labeled. Only 3 1/2" DOS (FAT 1.4 MB) disks are to be submitted. In addition, the GTAs may require you to demonstrate your program. To receive partial credit for programs that are non-working, or are not fully functional, a brief one or two paragraph description of the problem(s) must be included in the assignment folder. The location, routine minimum, must also be specified along with possible corrections that need to be made.