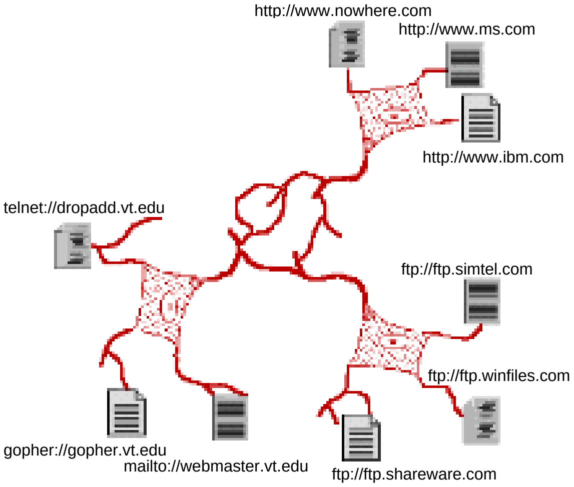
CS 2574 Spring 98
Project Specifications
Web Address Manager



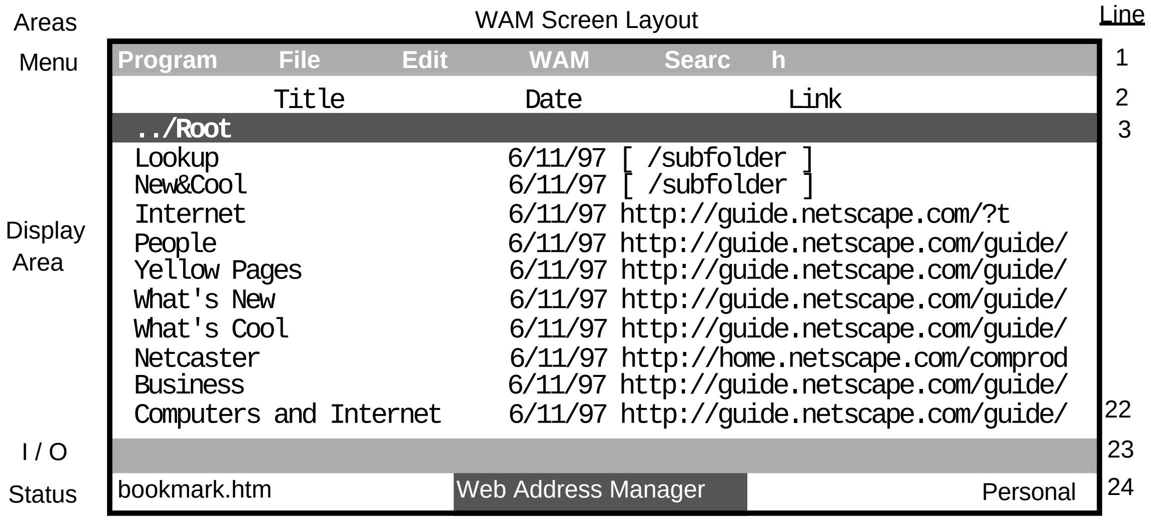
Web Address Manager (**WAM**) is an interactive software tool that would be used by Web surfers for the offline management of their URL database. The program is designed to allow Web document addresses to be browsed, created, deleted, modified and viewed hierarchically. WAM is menu-driven single-mode system. The user will scroll a selection bar to a desired data line and then choose a pull-down menu command to operate upon the current selection. The discussion that follows describes the WAM interface and operations in detail. Very minor deviations from these specifications may *not* result in grade penalties. However, developers would be well advised to receive prior approval before altering any of these requirements.

Discussion

WAM will begin execution with a startup screen giving the usual information about the system and developer, along with the current product version number. The startup screen may be presented in a clever manner in order to capture a viewer's attention. After the user hits the return key, the startup screen will clear and the WAM screen will be displayed, (see WAM screen layout, Figure 1). The command line parameters are then accessed to open the optional *bookmark.wam* file. The *bookmark.wam* file name, if present, follows the WAM program name on the operating system command line: "*wam bookmark*". A file existence check for any file opened is required and an error message is displayed if the file is not present in the current directory or specified file name path. The user may alternatively elect to start WAM without any command line parameters. In this case the user must first open an existing *bookmark.wam* or initial *bookmark.htm* files before most WAM menu commands become available. (For an complete explanation of the bookmark.htm file, wac file and wal file formats see: http://ei.cs.vt.edu/~cs1704/assignments.html). The remaining discussion will be made with regards to the areas of the WAM screen layouts and how the system execution affects them.

The WAM screen will be divided up into several areas: the menu area, display area, I/O area and the status area. The contents of these areas may change, but the four areas will be maintained on the screen at all times, after the startup screen has cleared and until the WAM quit operation executes. The menu area will hold the names of the pull-down menus available to the user, containing all WAM system commands. The display area will be used to show folder/URL information dependent upon the user's request. The Input/Output (I/O) area is employed by WAM to give error/ warning or prompt messages and to accept keyboard input. (All input from the I/O line, unless explicitly noted, will be buffered, (i.e. cin >>). All other input, [menu selection and scrolling actions], will be unbuffered. The status line will contain 3 fields: 1. the name of the current open WAM bookmark file, left justified; 2. the WAM system name, centered and **inversely displayed** ; and 3. the current folder, (dependent upon the display area contents), almost right justified. Note: most terminals will automatically scroll one line whenever a character is displayed at the lower right corner, (row 25 column 80), so care must be taken to ensure that this auto-scrolling does not affect the WAM system display if line 25 is used.

When the display area contains folder/URL data, (i.e. abbreviated folder and URL data), a **inversely displayed selection bar** will be provided, that the user can position over to highlight the desired data line that they wish to perform a menu command upon. They may change the “current highlighted data line” by moving the display area inverse video selection bar, vertically using the ↑ ↓ cursor keys, one line vertically up/down at a time. When the selection bar is located at the top or bottom of the display area and the user depresses the ↑ ↓ cursor key respectively, the display area contents will scroll one line. The line scrolled into the display area becomes the current highlighted data line. If the inverse selection bar is positioned at the top or bottom of the data list, (i.e. at the head or the tail of the current folder data list), and the user attempts to scroll beyond the list bounds, this should cause a bell to sound or the screen to flash and an appropriate error message to be displayed on the I/O line. In addition to the single line scrolling just described, WAM will provide two other display area ‘browsing’ methods. The user may also scroll up/down a 'page', [a page in this context is defined to be the number of data list elements shown in the display area, approx. 10-20], at a time by pressing 'U' or 'u' / 'D' or 'd' respectively, (the PgUp and PgDn keys may be used instead). WAM will also allow the top or bottom of the data list to be moved to directly, by the user depressing either the ‘T’ or ‘t’ / ‘B’ or ‘b’ keys, respectively, (the Home and End keys may also be used). The display area should never contain blank lines for data lists with ≥ 20 elements. Consider a folder with a data list of 100 elements, if currently elements 11 through 30 are displayed and page up is hit, the first 20 elements in the list should now be on the screen. Similarly, if elements 71 through 90 are displayed and page down is hit, the last 20 elements (81..100) are then displayed. For folders with <20 elements, page up/down would receive an error message and line up/down should not cause any line/element to scroll off the display. Display speed should not be an over-riding factor for implementation of these browsing actions.


WAM Screen Layout

```
              Title                    Date                Link
 ../Root
 Lookup                           6/11/97 [ /subfolder ]
 New&Cool                         6/11/97 [ /subfolder ]
 Internet                         6/11/97 http://guide.netscape.com/?t
 People                           6/11/97 http://guide.netscape.com/guide/
 Yellow Pages                     6/11/97 http://guide.netscape.com/guide/
 What's New                       6/11/97 http://guide.netscape.com/guide/
 What's Cool                      6/11/97 http://guide.netscape.com/guide/
 Netcaster                        6/11/97 http://home.netscape.com/comprod
 Business                         6/11/97 http://guide.netscape.com/guide/
 Computers and Internet           6/11/97 http://guide.netscape.com/guide/

bookmark.htm               Web Address Manager                 Personal
```

Figure 1.

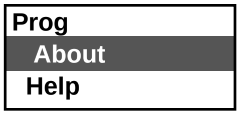
The display area shall contain the current’s folders parent’s folder title at the top, (except in the case of the root), followed by the current folder’s subfolder information, which in turn is followed by the current folder’s URLs. When the selection bar is positioned over a folder and a user hits the return key WAM will navigate into the selected folder. It will become the current folder causing an update of the status line. The display area contents will be replaced by the selected subfolder’s information.

Menu Commands

It is now time to explain how the pull down menus are to be accessed and how they will behave in WAM. Following this menu operation explanation, the rest of this section will discuss each pull down menu and each of the commands that they contain. A user will activate a desired menu by hitting the first letter of the menu, [**P** , **F** , **E** , **W** , **S**]. This will cause the corresponding menu and its commands to be displayed with the first command inversely displayed on a selection bar. The user may move the selection bar up/down the menu using the ↑ ↓ cursor keys, with the bar wrapping around from top to bottom and vice-versa. The user may choose a command by hitting the return/enter key when the bar is positioned over the desired command, which will of course cause the command to be performed. The user may also elect to NOT choose a command and exit the menu without executing any action by hitting the delete/backspace key. After a user has chosen, or decided not to choose, a menu command the menu should be erased from the screen and the display area restored to its previous or updated contents.

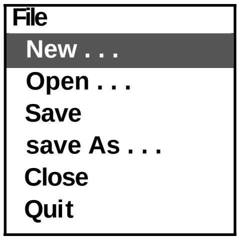
Not all menu options are functional all the time. Whether a particular option is functional is dependent upon the current contents of the display area. Edit options affecting URLs, (see figure 1), should not be operational when a folder is currently selected in the display area and vice-versa. For example, the Edit menu Visit Date... command would not be functional when the ../Root parent folder entry shown in figure 1 is selected. If a user selects a non-functional menu option at any time a bell and/or error message on the I/O line should occur. (Optionally the non-functional options could be disabled, not allowing the user to select them. Or non-functional options could simply not be displayed on the menus when they are inactive. Both of these approaches are optional and require a more sophisticated menu implementation than required.)

Prog



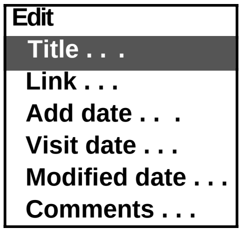
The **P**rogram menu contains only two commands, About and Help. The About command will display the initial startup screen and will be removed after the user hits the return key, restoring the display area contents. The Help screen will present a brief one screen overview of the WAM system and will also be removed after the user hits the return key, restoring the previous display area contents.

File



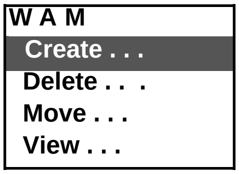
New will prompt the user for the name of a Netscape Navigator® bookmark file. The Navigator® BOOKMARK.HTM file is opened and processed. This involves building a double-linked list of all of the folder and URL data derived from the BOOKMARK.HTM file, ordered upon the entry (parent folder) indexes. This list is identical to the list built in the Web Address List, (WAL), program. (For specifications of the WAL program see: http://ei.cs.vt.edu/~cs1704/WAL.pdf .) As the Netscape Navigator® bookmark file is imported, a report file containing any encountered errors is built. If the file does contains errors, a warning message should be given instructing the user to view the report file externally. The Open command will prompt the user first to see if they wish to close any current open WAM file. (If the user decides not to close the current WAM file then the file menu is exited with no action taking place). If the user does wish to close a current WAM file then the close command is automatically executed and the user is prompted for the name of an existing WAM file to open. At this point the WAM file is read and the list data structure is created. The Save command simply writes the current contents of the internal WAM list structure onto the disk in the WAM file format, (identical to the WAL format), in the current/default directory, over-writing any other version of the file that might exist in the directory. The save As command operates almost the same as the save command, except that the user is prompted for the name for a new WAM file, which is then created and written to the disk, becoming the current open WAM file, (causing a status line update). The Close command will simply clear the display area and destroy the WAM list structure, (ideally it would be nice to inform the user if they have not saved the file if changes have been made and give the option for doing so, but this is not required). The Quit command will prompt the user to confirm that they wish to exit WAM, (☺), destroy the list structure and close any open files, before clearing the screen and returning to the operating system.

Edit



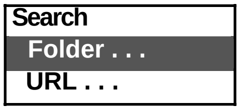
The **E**dit menu will provide a simple mechanism for a user to change the folder/URL information. Changes will require an update of the list contents and the display area. The Title option prompts the user for a title name to replace the current highlighted folder/URL, in the display area. Likewise the Link option prompts the user for the a URL to replace the current highlighted Web address entry. (This command may generate error message if the user selects it with a folder highlighted.) The Add date option allows a user to enter a new add date to replace the current highlighted add date of the folder/URL. The Visit date and Modified date options operate the same as the Add date option, but only are valid for highlighted URL entries. Both of these options must display the current visit/modified date in the user prompt. The Comments option should show the stored comment, if any, for the current highlighted entry in the display area. It should also query the user if they wish to enter a new comment. If the user decides to change the comment they should be allowed to enter a one line replacement comment. The display area must be updated after the comments option completes.

W A M



The **WAM** menu contains options for a user to manage Web address entries. The operation of the WAM options will depend upon whether a folder or URL is currently highlighted in the display area. The Create option will prompt the user for folder information for a new folder to be added to the current folder if a folder is highlighted in the display area. Likewise it will prompt for URL information for a new URL to be added to the current folder if a URL is highlighted in the display area. The Delete option remove the highlighted URL from the current folder. If a folder is highlighted the Delete option will query the user if they wish to perform a single folder removal or a recursive folder removal. If the user selects a single removal, only the highlighted folder from the current folder is deleted. All of the folder's contained URLs, are removed, and all of the sub-folders are *moved* to the current folder. (A single removal functions the same as in the WAL program.) If the user selects a recursive removal then all of the highlighted folder's URLs and subfolders are deleted. Effectively erasing an entire branch in the folder hierarchy. This *must be* accomplished by the implementation and use of a stack ADT. The stack will be used to store the subfolders for continued deletion. Highlighted parent root folder entries will generate an error message when the Delete option is selected. The Move option will allow a user to transfer the current highlighted folder/URL to the current folder's parent folder or one of its sub-folders. After selecting the options the user should highlight the destination parent entry or desired sub-folder entry and hit the return key. The user must be allowed to cancel the option by hitting the delete/backspace key. The View option will temporarily clear the display area in order to show a labeled, semi-hierarchical view of all of the stored WAM bookmark folders and URLs. It begins by showing the user the name of root folder, followed by its contents indented. The contents of nested folders will also be indented, with the indentation increasing and decreasing depending upon the level of nesting encountered. Only ten to twenty folder/URL entries are displayed on the screen at a time, allowing the user to hit the return key to list the next ten/twenty lines. Truncation will occur on the right as the indentation increases. After the end of the data has been shown the user should be prompted to hit the return key to redisplay the current folder's contents. No other input, menu or otherwise will be allowed until the View option completes. The user should be also be given the opportunity to exit the View option at any time during the screen pausing.

Search



The **S**earch menu contains only two options. Each option will allow the user to enter a corresponding folder/URL title. WAM will then hunt for the title in all of the stored WAM bookmark folders and URLs correspondingly. If the entered target title has a string length of N characters, this will involve a string comparison to match the first N characters of corresponding folder/URL list entries. If a match is encountered then the folder containing the matched entry becomes the current folder, resulting in an update of the display area. Non-matched titles should result in a warning message. (Ideally the matched entry would become the current highlighted item in the possibly updated current folder, but this is not required.) WAM is only responsible for transferring to the first located match. If multiple matches are possible WAM is not responsible for determining that they exist.

Whenever an WAM action will take more than a couple of seconds to execute, a brief specific message should be displayed, followed by "Please Wait...". An attempt to activate an illegal or invalid operation should cause a bell to sound or the screen to flash and an appropriate error message to be displayed on the message line. All dates, displayed and entered must be represented in a normal month day year format. Dates may be stored internally in any format desired, (i.e., Universal Coordinate Time, UTC), but the user must be unaware of the internal representation.

Text Windowing

The use of a text windowing package is NOT required for this program, but is recommended. The ability to define multiple windows to refer to the same screen area will greatly aid the task of screen content management. The restoring of the display area's contents after displaying pull-down menus and switching between different display area contents will be made simple. The text windowing facilities will greatly enhance the professional look-and-feel of the project.

For students coding a non-text windowing interface, the interface specifications described above must still be adhered, with the following modifications. The display area will be lines 1-22, the menu bar on line 23, the status area on line 24 and the I/O area on line 25. When a user selects a menu, the complete screen should be redrawn with the menu area line, (23), replaced by the menu commands. The user will be allowed to select a menu command by hitting the first letter of the command. The lines in the display area should be numbered to allow a user to select a line's contents for operation upon by a selected menu operation. A non-text windowing interface will require the creation of a window ADT and menu ADT for controlling the screen contents.

A specification addendum for this project may be forthcoming shortly if required. The addendum will describe minor enhancements or changes to WAM.

PROJECT GRADING and DOCUMENTATION

Successful completion of this software development and implementation project will result in the following external and internal specification documents for submission:

1. A complete **STRUCTURE CHART** of the system's design, including all interface specifications. The structure chart must be of sufficient detail to communicate to others WHAT functions are needed to implement this system and how the various parts are interconnected. The structure chart is complete if the algorithms for each module could be developed from it, along with a Data Dictionary, although no data dictionary is required. Use only 8 1/2" X 11" sheets of paper to record the chart, DO NOT use larger sized pages. Pages must be bound or stapled together. Use the symbols described in class for the components and the interfaces. Legibility counts and points will be deducted for sloppily presented work. Careful and CORRECT notation is also required and points will be deducted for a poorly annotated chart. The final chart will be compared with the initial chart to determine the amount of modification made to the original system design. Penalties will be incurred according to the amount of modification. Note — calls to standard C/C++ system routines: cin, cout, etc. need not be annotated on the chart.

2. An **INTEGRATION PLAN**, that is a plan for HOW and WHEN the various phases of the project will be developed and which modules will be added at each integration point.

a. THREE integration points must be given. The points reflect a semi-equal division of the project work. (Specifying only 2 modules to be completed at the first 2 points, while delegating the remaining, approximately 90% of the project for the last point would be unacceptable.)

b. For each integration point, you must name the TASKS that will be completed by the identified date and clearly define WHAT BEHAVIOR can be expected to be exhibited by your program at that point.

c. The three integration points for the system are: *1. 2/6/98 2. 3/2/98 3. 4/13/98*. Each student will be expected to demonstrate to a GTA the behavior of their project at the second and last integration points. Failure to meet with the GTA for a demo will result in point penalties.

d. At any integration point or when the program is submitted, any non-functioning commands, actions, etc. of the system are not expected to bomb the system. A message should be displayed informing the user that this operation has not currently been implemented or is still under development. On the final submission, all non functioning aspects of the system MUST be accompanied by a short explanation describing the suspected problem(s).

3. A **USER MANUAL** generated by the word processor of your choice must be written. The manual should enable the novice WAM user to access all the features of the system. This document should be at least 8-10 pages in length, excluding the title page, table of contents, index, etc. The user manual must be written from the perspective of a novice WAM user, not from a programming implementation point of view.

<u>QUALITY OF CODE</u>

It is expected that the code will be WELL-DOCUMENTED, appropriately indented and VERY READABLE. Points will be deducted for poorly presented code. Individual functions, excluding their documentation and declarations may be no longer than one page. Each compilation module should contain only related functions. Function headers should resemble the following template:

```
/*******************************************************)
(*FUNCTION NAME                                        *)
(*DESCRIPTION OF FUNCTION                              *)
(*DESCRIPTION OF ALGORITHM : FUNCTION IMPLEMENTATION   *)
(*                                                     *)
(*CALLED BY: (LIST OF FUNCTIONS)                       *)
(*CALLS: (LIST OF FUNCTIONS)                           *)
(*                                                     *)
(*PARAMETERS: NAME AND ROLE IN ALGORITHM OF EACH       *)
(*                                                     *)
(*AUTHOR: name of author                               *)
(*REVISIONS: DATE, REASON                              *)
(*            AUTHOR FOR EACH (if different)           *)
(*VERSION: x.xx                                        *)
(*******************************************************/
```

Each of these documents must be placed in ONE binder or folder. Each separate document must be clearly labeled, bound and separated from the others. The diskette(s) containing the code must be inside and securely fastened. ALL WRITTEN MATERIAL MUST BE EASILY READABLE AND THE PAGES SHOULD BE EASY TO TURN IN THE BINDER. ILLEGIBLE WORK WILL NOT BE GRADED AND YOU WILL RECEIVE A "ZERO" FOR ANY SUCH PRODUCT SO REJECTED. Failure to conform with these standards will result in an immediate loss of points.

The following scale will be used for grading:

PRODUCT	PERCENTAGE OF PROJECT GRADE	PERCENTAGE OF FINAL GRADE
Structure Charts:	20	5.00
Integration Plan:	10	2.50
Code/Execution:	60	15.00
User's Manual:	10	2.50
Total:	100%	25.00%

Due Date schedule:

There will be NO extensions or late submissions for this project! Any project or portion thereof NOT submitted on time will be rejected!

PRODUCT	DATE
Structure Chart:	03 - 20 - 98 (Initial Version)
Structure Chart:	04 - 13 - 98 (Final Version)
Integration Plan:	04 - 13 - 98
Code/Execution:	ALL code is due by **5:00PM** on **04-13-98**. **• D-Day •**
User Document:	04 - 13 - 98.

The following deliverables are due on **April 13th**:

a. Hardcopy and softcopy LISTING for all code.

b. Diskette(s), (system labeled: WIN 95 or NT), with all source code, project files and a single executable image. The executable image must run by typing **wam**. The files on the disk may be zip compressed in a self extracting archive. Only 31/2 inch DOS (FAT)1.4 MB disks are to be submitted. No "extra" files (other than project source files, I/O, data files, structure chart and user manual files) should be on the diskette(s).

c. A hardcopy and softcopy of the user's manual. The softcopies (2) should be in the word processor format used to create the manual and also in standard ASCII text format.

d. Modified final up-to-date Structure Chart + the initial design version. (If a softcopy is available please include it on the disk.)