

CS 1704: Program #1
Format Validation Checker
Due: Wednesday Oct. 2, 1996

Problem

This date has changed from the hardcopy handed out in class.
 Code a program to check the file formats for on-line code library documentation.

Discussion

There are two types of file formats to be validated (checked): 1. category file; 2. documentation file. A category file will have a .gor extension. There will be one root category file that will contain references (file names) to all of the other system category files. The category files will contain links to all of the documentation files. Documentation files will have a .dr extension. See the "Input" section for a discussion of the category and documentation file formats.

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 code library documentation system.

Execution

The 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, address and telephone number. 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 root category file. The name of the root category file is input and opened for reading. By default the program is to assume that the root category 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 root category file does not exist, the program is to halt with an appropriate error message. If the root category file does exist, it is read, and checked for format errors. All category or documentation files referenced by the root category file, (that do not contain format errors), are checked for existence, opened, validated and processed in the same manner as the root category file. For each file that is checked and validated the program should include a brief entry in the report file (and echo it to the screen). See the Output section for the report file description. After all category and documentation files have been processed, the program should issue a brief termination message before returning to the operating system.

Structure Charts

An initial structure chart design must be submitted on **Monday Sept. 16**. 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.

Input

The format of a category file requires that lines at the beginning of the file be organized as follows. Line 1 will contain a brief description of the category. Lines 2-15 will contain a full description of the category. However, all 14 may not be used, (i.e., present). When less than 14 lines are used for the full description, the last description line must be followed by a 'termination' line that contains a period in column 1. Thus if any line of the full description begins with a period in column 1, it will cause the remaining lines of the description to be processed incorrectly. Note that when all 14 lines are used for the full description a termination line is not required. All remaining lines in a category file must adhere to the following column format:

columns	contents	comments
1	'C' or 'D'	File type entry code: C-category, D-documentation.
5-15	Label	Label for file.
20-80	file name	File name of entry with extension (.gor, .dr). May contain relative or absolute pathname of the file.

Format Validation Check

Due Date: Wed. Oct. 2, 1996

This date has changed from the hardcopy handed out in class.

The format for a documentation file requires lines organized as follows. Lines omitted in the following specification should be blank, and will be ignored by the program.

<i>line</i>	<i>contents</i>	<i>comments</i>
1	name	The name of a function.
3	syntax	Function heading.
5	include	Required file include headers.
7	parameters	Brief parameter description.
9	return	Return value description.
11	see also	Related functions.

The remaining lines, (starting at 13), will contain a full description of the function followed by lines giving an example usage of the function. The full description and example usage sections must be separated by a termination line, (containing a period in column 1). The program need not check the contents of each line of a documentation file. It need only determine the existence of each line.

Output

The output report must be echoed to both the screen and the output text file "xxx.rpt", where xxx is replaced by the root category file name. Each category and documentation file processed/validated must have entries similar to the following:

```

curses.gor
=====
File is correctly formatted.

window.gor
=====
Line 1: Invalid entry code.

scrollok.dr
=====
File id correctly formatted.

```

Assumptions

It may be assumed, for ease of initial implementation, that a maximum of no more than **100** total files, (category and documentation), will be checked and validated. (This maximum file limitation will be removed later.) Lines in all files should be no more than 80 characters in length. Characters beyond the 80th column can be ignored. Data files will be provided. Downloading instructions will be available from the course Web site: (<http://ei.cs.vt.edu/~cs1704>).

Grading

Separate compilation is required for this program. Failure to produce a separately compiled program will result in grade penalties. Turn in hard copies of the source code, input/output files, structure chart and a diskette (system labeled: Win 3.1, 95 or NT), with files containing: ASCII source code, executable image, I/O files, and a brief ASCII *readme* file with execution instructions. The executable image must be called *validator.exe*. The files on the disk should not be compressed and file protection should be set to allow anyone access. All deliverables must be placed in an envelope folder, neatly labeled. In addition, your GTA may require you to demonstrate your program. Do NOT submit any 5 1/4" MSDOS disks. Only 3 1/2 inch DOS (FAT) disks (either 720K or 1.4 MB) 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.