

PROGRAM FINAL

This C++ program constructs a large html table out of all the small html files previously created. It writes the name of each nuclide at the beginning of every table row, next to which displays all three analyzed libraries. If a library includes the nuclide, a link to the small html file appears under the library name. In case the library does not contain the nuclide a »No data« is written.

At the beginning of the code the name of the output html file is specified, after which the program creates its header. The following lines include 502 nuclide codes. In the next step the program opens the folder where all the small html files are stored. Strings (previous=" ", ly=" ") and numbers (ENDF=0, JEFF=0, JENDL=0) which will be useful in further processing are defined. The while loop reads all the single file names in the folder. Inside it we first define where the nuclide code will be stored, which sign ('n' or '_') appears after the number code, where the library name will be stored and the string »name=" "«, which will be later useful. The »if« that follows extracts the number code from the file name. The program then tests if the code, which has been just extracted, is equal to the one extracted in the previous iteration. This means that we have the same nuclide. Its data must be written in the same row of the large table. In case of the same nuclide, the library will be extracted from the file name by the following »if«. The next three loops increase one of the numbers (ENDF=0, JEFF=0, JENDL=0) to 1 in order to check which library has been just read and its name is written in the string »name«. The »e« sign checks if the small html name involves 'n' or '_' in order to write »New evaluation« or not under the library name. At the end the code is assigned to the string »previous«. The »else« following the first loop is entered when the code we read is not equal to any one before, because this means that we have a new nuclide. When a new code is extracted, the program first checks if the previous nuclide was included in all three libraries. A »No data« is written under the library name where it is missing. The table row is then closed, the numbers are set to 0 and it starts a new table row. The extracted code is then compared with all codes included in the program using the next »if«, »else« lines. The procedure that follows is the same as before: it checks to which library falls, if it has »n«, sets the link and assigns the code to the string »previous«. After the last file in the folder has been analyzed, it checks if that nuclide is included in all libraries (when all the numbers are 1) and closes the html table.

The conditional loops used for reading nuclide codes, libraries and checking for 'n' or '_' have to be doubled, because some codes have an additional character or character and number after the 7 digit number. Because of this the position of the other marks in the file name changes.

Conditional loops checking newly extracted codes must have a restriction in the condition in order to ignore the '.' and '..' files which always occur in folders.

The table we have just created must be moved to the folder where all the small html files are stored in order to make work properly the links.

The output html must be then opened in Text-pad in order to correct the double quotes in all the lines which construct the links to the small html files, otherwise the links from the main table to the data files for each nuclide will not work.