

Design and Implementation of a File Splitter and Merger Software

تصميم وتنفيذ برنامج لتقطيع وتجميع الملفات

Muhanad Hayder

University of Kerbala - College of Science

Department of Computer Science

E-mail : muhanadeng2005@gmail.com

Abstract

This paper presents a file splitter tool that is based on Java for making our distribution of files easier in removable media like floppies, CD, ...etc. For this it can split a large file that cannot be fit into a single USB flash memory or CD or any other media to one or more files that is easily or perfectly fit into them. Also it is possible to merge them back in destination machine.

For splitting, user will choose the file which he wants to split [for e.g. .txt, .doc, .mp3, .bmp, .xsl, .dat].then utility will ask for attributes of splitted files, location and size. Then reads the inputted file byte by byte write it into sequence of split files all these newly created files are stored either in default folder or in user define folder. All the above information related to splitted file is stored in 'original file name.jfs'. The splitting folder will be used to rebuild original file as it is. User have to give path of '*jfs' file to utility for all files to be merged.

The setting like (default split folder, default merge folder and default split) can be configured and saved by using File Splitter and Merger configuration under options tab.

This paper used object oriented programming with the Java2 Standard Edition (J2SE) programming language.

Keywords: file splitter, file merger, object-oriented programming and J2SE.

// الخلاصة

مقطع ومجمع الملفات هو عبارة عن تطبيق بلغة جافا اصدار (J2SE) يسهل عملية توزيع ونقل الملفات في وسائط خزن المعلومات كـ (floppy, CD and DVD) يستخدم التطبيق لتقطيع الملفات كبيرة الحجم التي لا يمكن ان تخزن في وسائل الخزن الشائعة لكبر حجمها الى قطع صغيرة تتلائم وحجم الاقراص المدمجة والمرنة وشرائح الذاكرة صغيرة الحجم وباستخدام نفس التطبيق يمكن تجميع هذه القطع لاستعادة الملف الاصيلي .

يمكن استخدام هذا التطبيق لتقطيع جميع انواع الملفات وبدون استثناء حيث يحدد مستخدم التطبيق عدد او حجم القطع التي سوف تنتج عن تقطيع الملف الاصيلي وموقع هذه القطع بعد الضغط على زر (Split) تتم عملية قراءة الملف بايت بايت ومن ثم تكتب بالقطع التي تم تحديدها من قبل المستخدم وبالذليل المحدد هذه العملية تتزامن مع بناء الملف (.jfs) الذي يحتوي على عدد وحجم وموقع جميع القطع الناتجة عن تقطيع الملف الاصيلي والتي يحتاجها التطبيق في عملية تجميع الملف (Merge) حيث يحدد المستخدم الدليل للملف (.jfs) والمسمى بنفس اسم الملف الاصيلي فتتم قراءة قطع الملف واحدة تلو الاخرى (byte by byte) لبناء الملف الاصيلي .

1. Introduction

File splitting and merging is a fundamental concept in many aspect of computer science, as it is involved in many applications. Even if you don't actually feel, a lot of operations inside the computer have some type of splitting and merging things, silence tracking in sound recognition applications, extracting ROI (region of interest) by masking an image and a lot of other things are all different types of file splitting operation. The development of a software system is usually just a part of finding a solution to a larger problem. You may need to upload a file to the Internet, but with our current network speed it's really hard to do so with large files. You may need to extract a small part of a music or video file, but it's hard to work with software available now as they are for professional people. Furthermore we have a real problem in dealing with the miss compatibility from one operating system's software to another operating system [1].

This paper has made these problems easily solved as it's a useful splitter/merger, platform independent software with non complex user interface which makes you enjoy splitting and merging different type of files easily. This paper contains three fundamental operations Split, Merge and Option

2. Split and Merge

In the Windows environment, a designation of the operational or structural characteristics of a file is called file type. The file type identifies the program, such as Microsoft Word, that is used to open the file. File types are associated with a file name extension. For example, files that have the .txt or .log extension are of the Text Document type and can be opened using any text editor. In the Macintosh environment, a four-character sequence that identifies the type of a Macintosh file. The Macintosh Finder uses the file type and file creator to determine the appropriate desktop icon for that file [1].

For handling all types of files in any system we have to work on the basic concepts of a computer on which files are constructed, which is a stream of bytes, as every file regardless of its type, is a set of bytes combined together forming a special format in the system [2].

There is a lot of benefit of using file splitting but the most important one is to break a large file into chunks, these chunks can be easily fit in the removable media as well as can be easily transmitted over the Internet [3].

3. Object Oriented Programming

The object-oriented paradigm grew out of a need to meet the challenges of best practices using standard structured programming. By being clear about these challenges, we can better see the advantages of object-oriented programming, as well as gain a better understanding of this mechanism. To some, the object-oriented programming (OOP) technique is merely a way of organizing programs, and it can be accomplished using any language. Working with a real object-oriented language and programming environment, however, enables you to take full advantage of object-oriented methodology and its capabilities for creating flexible, modular programs and reusing code [4]

Most object-oriented programming languages include a set of class libraries that provide basic data types, system input and output capabilities, and other utility functions. These basic libraries are part of the standard language environment, which also includes simple libraries, form networking, common Internet protocols and user interface toolkit functions [5].

In this paper we make a heavy use of object oriented programming by using this rich class libraries. In fact whatever model implemented here is nothing but class, because java is pure object oriented programming language where write every thing inside a class even main function.

4. Platform Independence

Platform independence-that is, the ability of a program to move easily from one computer system to another. It is one of the most significant advantages that Java has over other programming languages (because java has both compiler and interpreter), particularly if your software needs to run on many different platforms. If you're writing software for the World Wide Web, being able to run the same program on many different systems is crucial to that program's success. Java is platform independent at both the source and object code (bytecode) [6].

Platform independence means that a program can run on any computer system. At the source level, Java's primitive data types have consistent sizes across all development platforms. Java's foundation class libraries make it easy to write code that can be moved from one platform to another without the need to rewrite it to work with that platform. Platform independence at the source level means that you can move Java source files from system to system and have them compile and run cleanly on any system [7].

Normally, when you compile a program written in C or in most other languages, the compiler translates your program into machine code or processor instructions. Those instructions are specific to the processor your computer is running, so for example, if you compile your code on an Intel-

based system, the resulting program will run only on other Intel-based systems. If you want to use the same program on another system, you have to go back to your original source code, get a compiler for that system, and recompile your code so that you have a program specific to that system. The Java bytecode interpreter is often also called the Java virtual machine or the Java runtime is used to avoid such problems [8].

The disadvantage of using bytecode is in execution speed. Because system-specific programs run directly on the hardware for which they are compiled, they run significantly faster than Java bytecode, which must be processed by the interpreter. For many basic programs, speed may not be an issue. Java support enterprise application as well mobile application as depicts in figure (1).

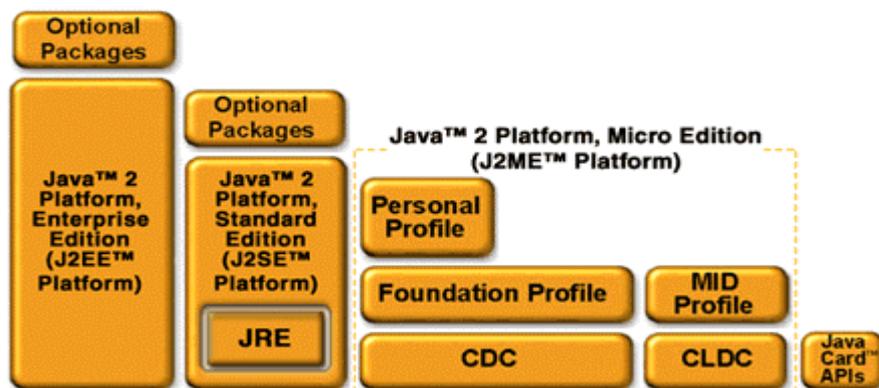


Fig. (1): java platforms [9].

6. File Structure

There is one important distinction that must be made at the outset when discussing file structures. And that is the difference between the logical and physical organization of the data. In the whole a file structure will specify the logical structure of the data that is the relationships exist between data items independent of the way in which these relationships may actually be realized within any computer. It is a logical aspect that we will concentrate on. The physical organization is much more concerned with optimizing the use of the storage medium when a particular logical structure is stored on, or in it. Typically for every unit of physical store there will be a number of units of the logical structure (probably records) to be stored in it. For example, if we were to store a tree structure on a magnetic disk, the physical organization would be concerned with the best way of packing the nodes of the tree on the disk given the access characteristics of the disk. [10]

7. System Design

The design steps of this paper are shown in figures 2,3,4,5,6,7, and 8.

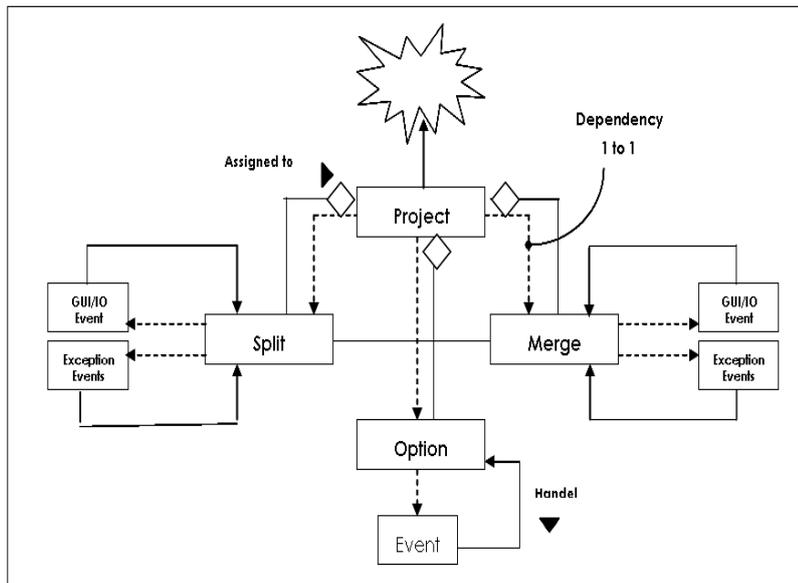


Fig.(2):Class Diagram

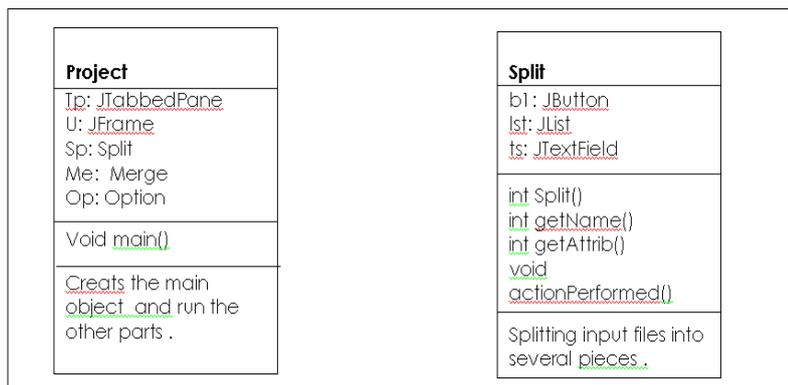


Fig. (3): Project and Split classes

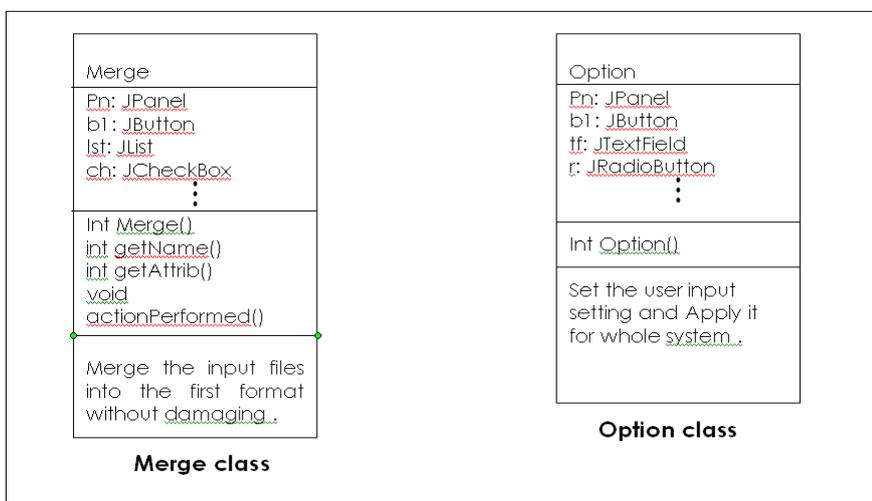


Fig. (4): Merge and Option Classes

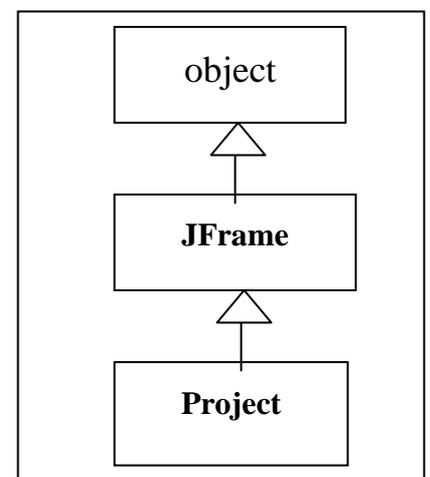


Fig. (5): Inheritance

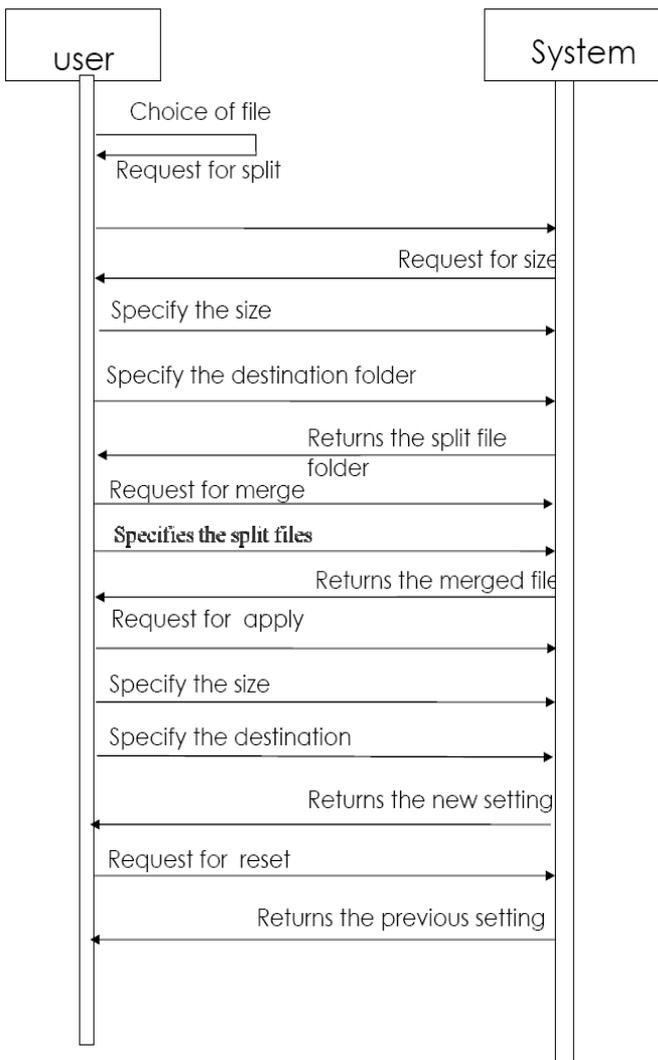


Fig. (6): Sequence Diagram

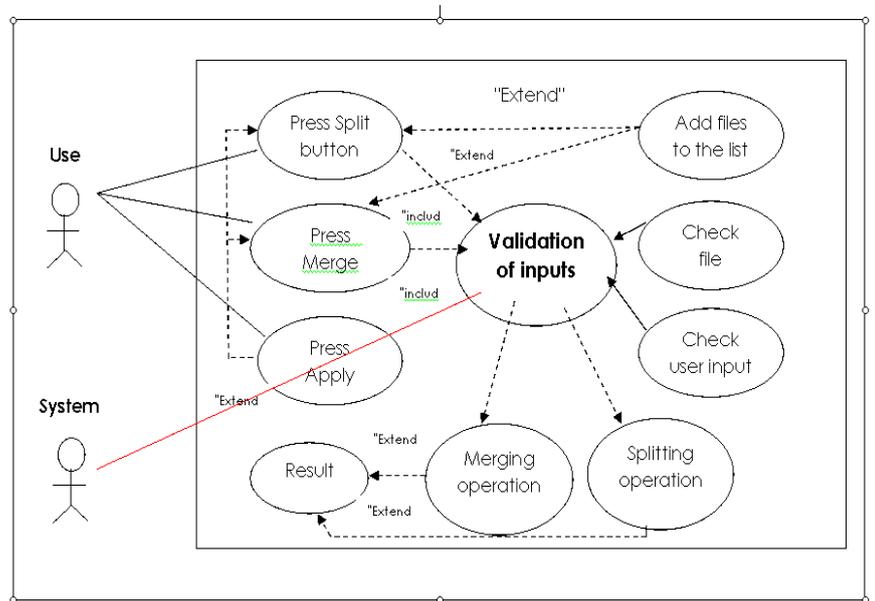
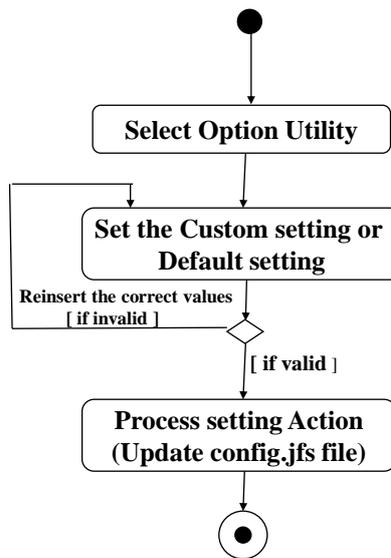
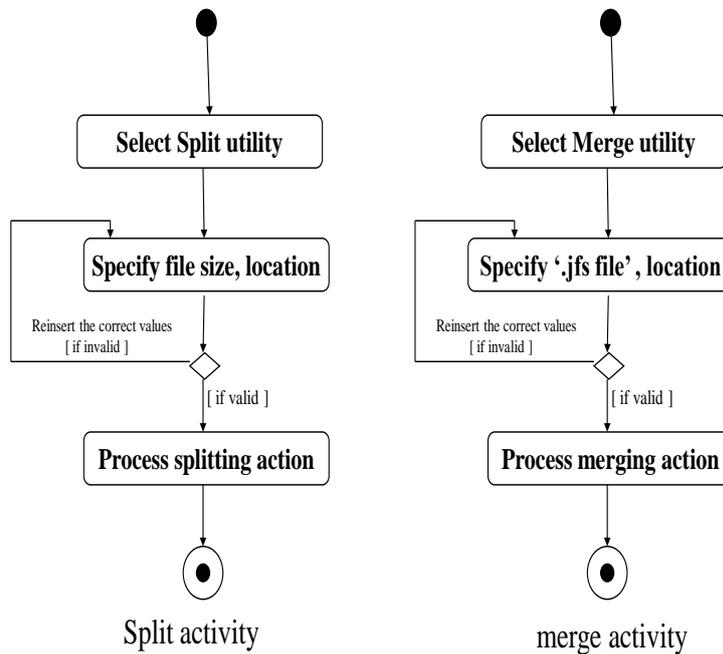


Fig. (7): Use case Diagram

ACTIVITY DIAGRAM



Set Option Activity

Fig. (8) : Activity Diagram

8. System Implementation of Split Tab

When user runs the system the following window will appear, this is the main form of the system as shown in figure 9. The splitting operation comes in two forms, as it is available in both size and number of each piece, the user can choose which way he wants to split the file, numbers or sizes.

If he chooses to split in size-based way then the splitting can be done base on Bytes, Kilobytes or Megabytes. In both ways the user can specify number for pieces size or their number of piece themself. The user can Add, Remove or Remove all items in the list , he can also select item from

list to show there source path, he can choose the destination path to save the pieces after Split operation and can show the number of pieces and size of each piece and the size of last pieces. For adding files to the list the user press the "Add" button, to show the open dialog as depicts in figure 10.

User can choose as many files as he wants, and of any type (can split up to 100 files in one split operation and there is no restriction on file type). The open dialog is multi-selection such that the user can add all files one add operation as shown in figure 11. After choosing files user should press "Open" button to put all files in the list (figure 11).

The user can know the source path of every file in the list by selecting file name from the list of added files, the source path will appear on the text field under the list (figure 12). If the user interesting on changing the destination path he can press "Brows" button, the open dialog get opened then he can select destination folder to save the resulting file pieces after splitting the original file (figure 13).

If the user need to remove any file from the list , he can select its name (or its size) and press " Remove " button to remove it , if the user press "Remove" button without any selection , Error Message will appear to notify the user to select a file name from the list to remove it (figure 14). If the user press "Split Now" button without adding any file to the list , he will see message box that notify "No source file is selected" (figure 15). When press the "Split Now" button first input files which were added to the list are checked to ensure that they exist, if they don't exist an exception will occur which is handled by showing an error message to the user (figure 16). If the user write invalid number in text field when he press "Split Now" button the software check whether the number is valid or not , if its not, message will be displayed to notify the user about this error (figure 17). After checking all possible error conditions the actual work begins. Each file in the list is being read as an array of Bytes, and it can be handled as a contiguous chunk.

Determining the number of resulting pieces was an effort consuming operation as it had to take all the possibilities. We wrote a method (*getAttribute()*) to specify the number of pieces as well as the size of each piece. This method will divide size of the file by size of requested piece (Byte, kilobyte or Megabytes) to determine the number of pieces if the size didn't fit, the number will be increased by one to determine the size of last (smallest) piece. If the splitting was based on number of pieces then this method will determine the size of pieces by dividing the total size by the number of pieces. The other problem in this operation was determining the name, and more complicatedly the extension of output pieces. This problem was solved by writing a method (*getName()*) which will take the responsibility of naming all the pieces. The resulting files will have the same name as the input file with index appended to the name and the same extension by saving its type in the input array to be merged with the file name after naming operation.

Finally all have to do is checking the output path where we have to write the pieces, with adding a file of type ".jfs" (java file splitting) it is our own extension naming that contain the information needed by the software to get all the pieces back together in merging operation. The user will be notified that the operation is done by showing message of "Splitted Successfully" at the end as shown in figure (18).

The software allows user to see all details he need to know about the splitted files (size of each piece , number of pieces and size of last piece) by selecting any file from the list (figure 19). There is an option which let user choose to delete or keep the source file. If user chose to delete it the splitter will delete the source file after finishing all processes.

Figure 26,27 and 28 depicts the splitting example for video, audio and executable files, as its is obvious the aggregate size of all pieces is equal to the size of original files. Furthermore you can play audio and video pieces. The software is able to split any type of file but it make no sense to split documents (text) files or an HTML files base on there size.

9. Implementation of Merge Tab

This software is able to merge files that were splitted, back together without damaging the original format of the file. Splitter/Merger software takes “.jfs” files as input, and merges all the pieces which belong to the same file which are mentioned in the “.jfs” file. When the user click on "Merge" tab, the following screen will appear (figure 20). Where he can add , remove or remove all files from the list , select the destination path , select whether delete file's pieces after merge operation or not and show or hide the details of files by press " Show Details " button. This software gives the user all the details he need to know about the files to be merged, such information include: files names, size, number of pieces, size of each piece, and for special cases (when the last piece size is different) the size of last piece, making the user interactivity as simple as possible, by reading this information from the “.jfs” file.

When the user press “Show Details” the details will appear (figure 21). If the user press "Merge Now" button without any selected (.jfs) files (pieces of splitted files) , error message will appear as follow in figure (22). When press the “Merge Now” button first system ensures that the input file is correct and the file doesn't have problems, and then ensure that all pieces do exist at the same location to be merged. Otherwise the exceptions are handled by showing messages to the user asking him to reenter its request in proper way, handling the exception preventing the software from any possible fault.

The merger function reads each piece of the file by taking its name from “.jfs” file and places it in a Byte array then combines the different arrays to form one contiguous file, put it in a file writer format. Merge function checks the output path to ensure that it is correct and if the directory doesn't exist, it will create one with the given name, place the output file there, and if the user chose to delete the pieces, it will delete the pieces and finish its work.

After finishing merge operation of each file the message of "One file was Merged successfully" will appear in figure (23).

10. Implementation of Option Tab

In standard software, there is a part concerned with the whole system's setting, which allow the user to change the setting of the system manually. This software also contain such part, through which user can control the system. Option tab is responsible for this activity.

User can save the setting like default split folder default merge folder and default split configuration by using File Splitter and Merger configuration under options tab (figure 24). To set default folder for split and merge file click the browse button right to the text box and choose the folder you want to set as default. File splitter will add a subfolder split files or Merge Files to your path. You can choose the different split and Merge folder.

Also user can choose the type of split, either base on the size of each piece or the numbers of pieces per file. For this you have to select the appropriated radio buttons. If you select the size radio button you have to select the size of each pieces in Byte or kilo Bytes or Mega Bytes radio buttons. After doing the above said changes, click Apply button to save it for future use. On the applying process the software will create a file (if doesn't exist or rewrite if it does exist) which name is “configure.jfs”, include all information needed in the system load, and then apply the changes to all the system without restarting it.

The software will take all setting from “configure.jfs” next time it is loaded. If there is any problem about the “configure.jfs” file, the exception operation will take care of the initial values preventing the software from crashing and thus terminating by Operating systems mechanism.

If the user wants to repair the system by retrieving the basic setting of the system, he can use the reset option which will reset the entire setting back to first default one and then apply it to system.

If the number entered in the test box is not a correct one (i.e. letter or negative number) an error message will occur (figure 25).

11. The Features of the System

1. System could split and merge any type of files, compare to other one which can split only specific type of file like MP3 splitter merger 1.5, PDF Merger Splitter 4.0.1.5. even some of them could either split or merge file like JR splitter and Twin File Merger.
2. the software could preserve the type of split pieces so that the resulting pieces have same name and extension of original file, where most other software they couldn't so user could play the pieces without manually changing the extension of the file. Example of such one WinMend File Splitter 1.2 .
3. the software give the user full control over the process of splitting and merging files. Like splitting and merging files base on size or number of pieces which is not available on JR Splitter (split using size only), selecting list of file for splitting or merging, deleting original file after splitting or deleting the pieces after merging, changing and set the directories and configurations of the software using option tab. All this controls done using very friendly user interface.
4. Most important features which made this software unique is being running across platforms so that it can run on windows, linux, Macintosh, and Solaris and any other operating System.

12. Conclusions

1. The system runs on all existing operating systems, because Java is platform independent.
2. System implementation, make the task easier for users to deal with such kind of functionality wasn't proposed in user friendly GUI before.
3. The system is almost error free as we handled all exceptions that will catch every possible error.
4. The result split files have the same extension and name with index as the input ones; this facility is not available in many equivalent systems.
5. Working with IDE integrated developers environment IDE which support drag and drop technique is much easier than IDEs which don't support it, as we had to write code for each and every component in the system, and to change small details we had to do a huge work just to make the GUI suitable one, while in IDEs with drag and drop facility all of these are automatically available to the programmer.

References

- [1] Mike Cohn, Bryan Morgan, Michael Morrison, Michael T. Nygard, Dan Joshi and TomTrinko., **JAVA Developers Reference** , 1996.
- [2] http://www.cryer.co.uk/file_types/undex.htm
- [3] <http://www.methods.co.nz/splitfile/splitfiles.htm>
- [4] <http://java.sun.com/j2se/documents>
- [5] Bruce Eckel, President, MindView, Inc., **Thinking in Java, 2nd Edition**, Prentice-Hall, **Release 11** mid-June, 2000
- [6] Herbert Schildt, **The JAVA 2 Complete Reference, Fifth Edition** McGraw-Hill/Osborne, Copyright © 2002
- [7] Dennis de Champeaux, Doug Lea, Penelope Faure, **Object-Oriented System Development** copyright ©1993
- [8] Alan Shalloway, James Trott, **Design Pattern Explained - A New Perspective on Object Oriented Design**. December, 2000
- [9] <http://www.fastgraph.com/help>
- [10] Grady Booch, James Rumbaugh, Lvar Jacobson, **The Unified Modeling Language User Guide**, Rational Software Corporation. October 20, 1998

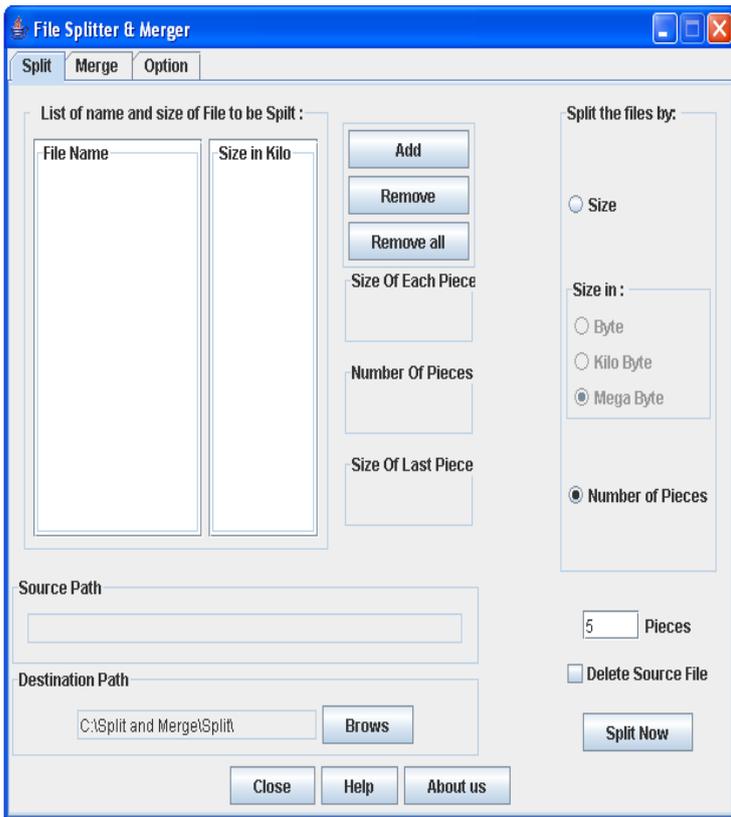


Figure (9) Split tab window

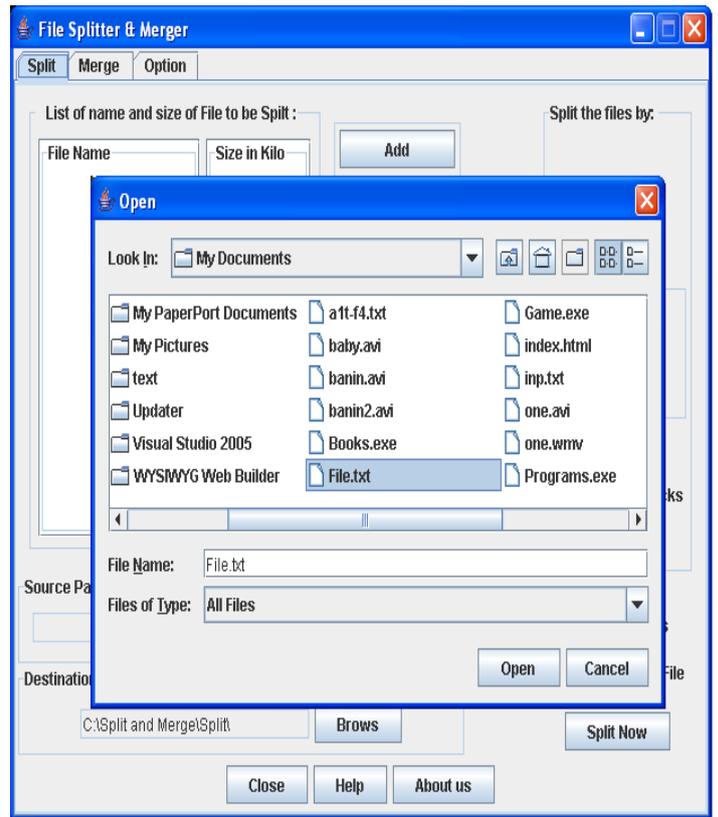


Figure (10) Open dialog of split tab

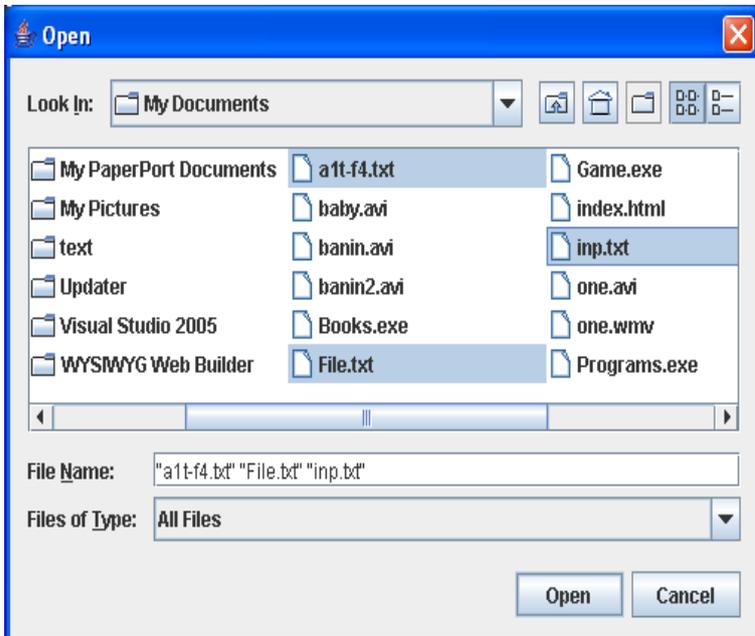


Figure (11) Multi selection property

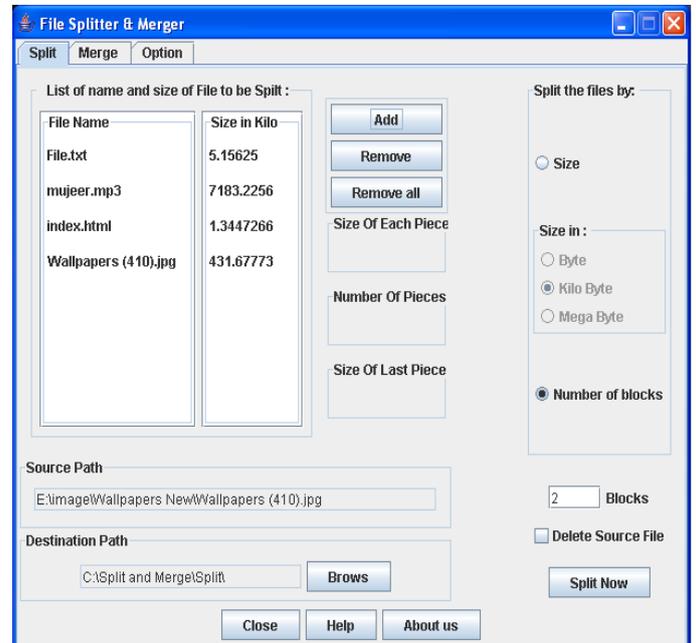


Figure (12) Files added to list

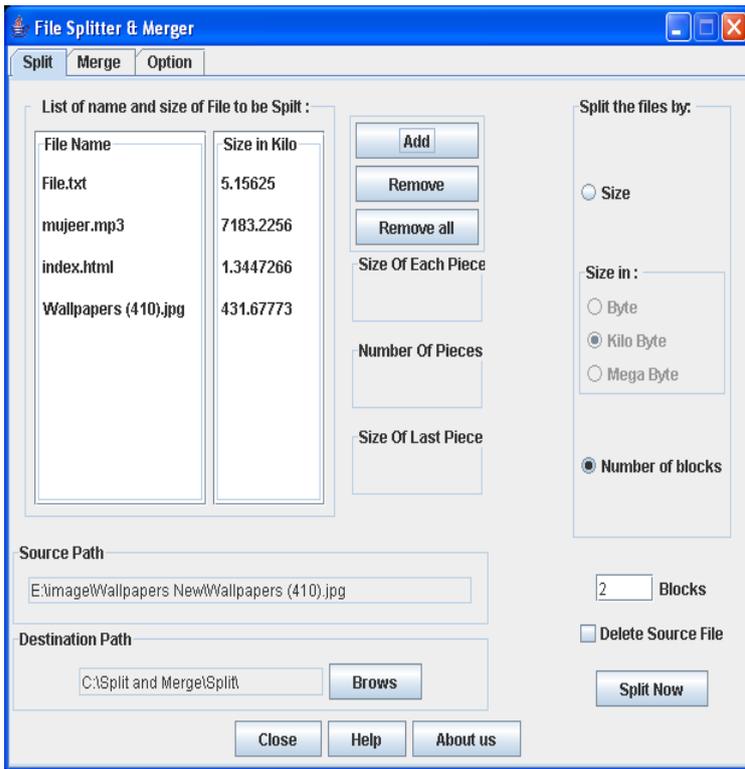


Figure (13) Open dialog of Brows button

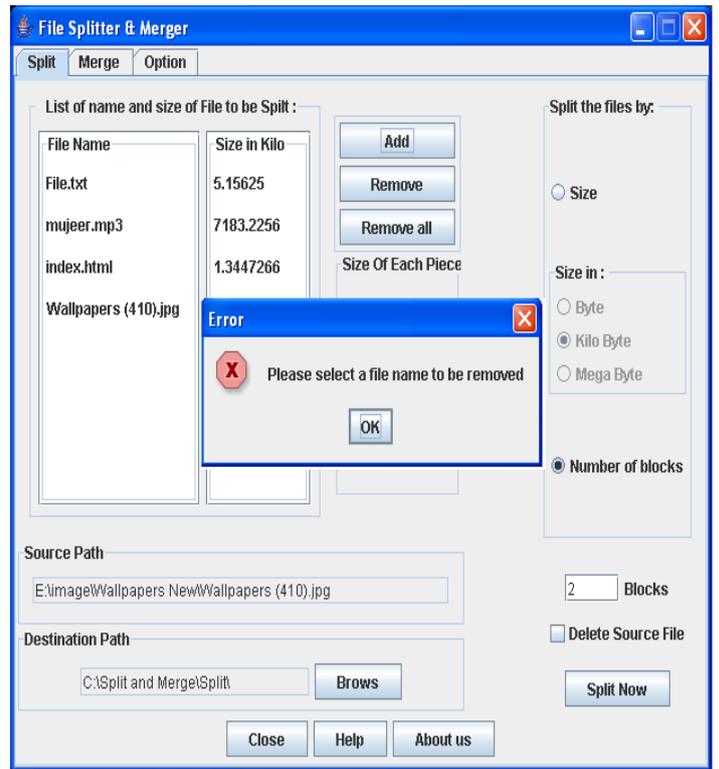


Figure (14) Error Message for non selected file

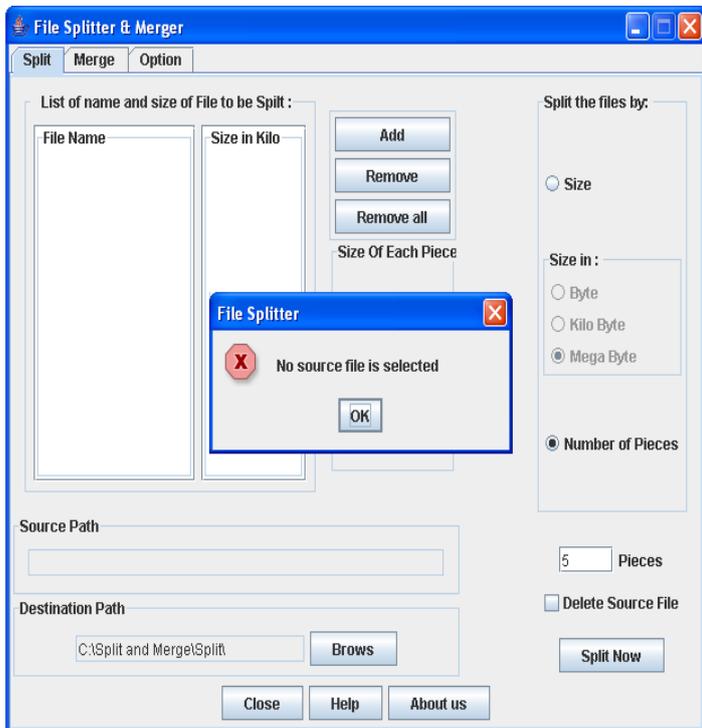


Fig (15) Error Message for no source file

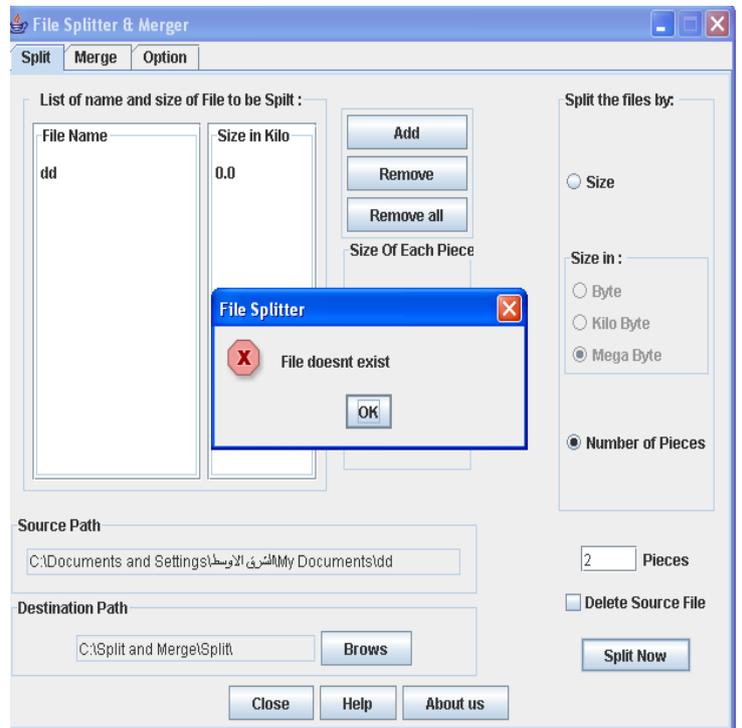


Fig (16) Error Message for file doesn't exist

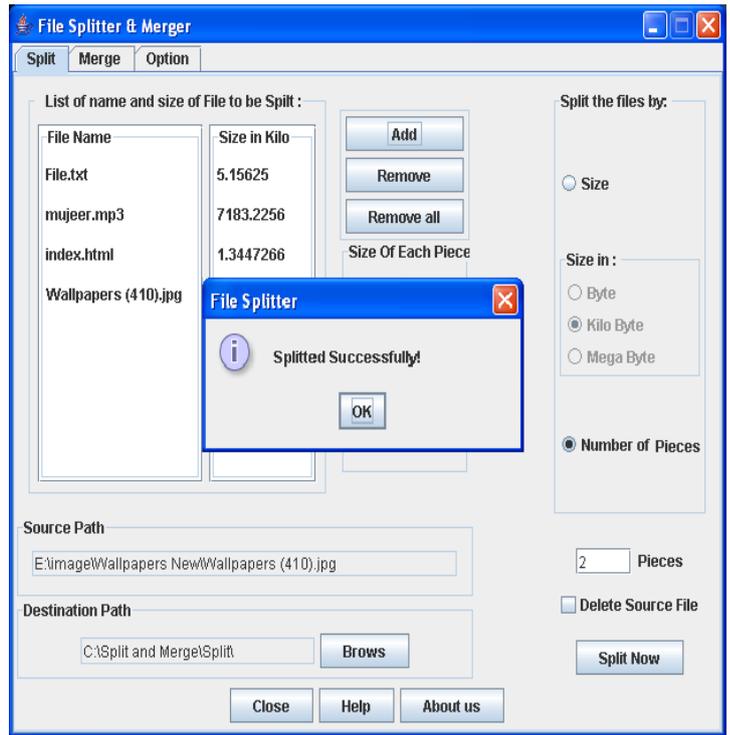
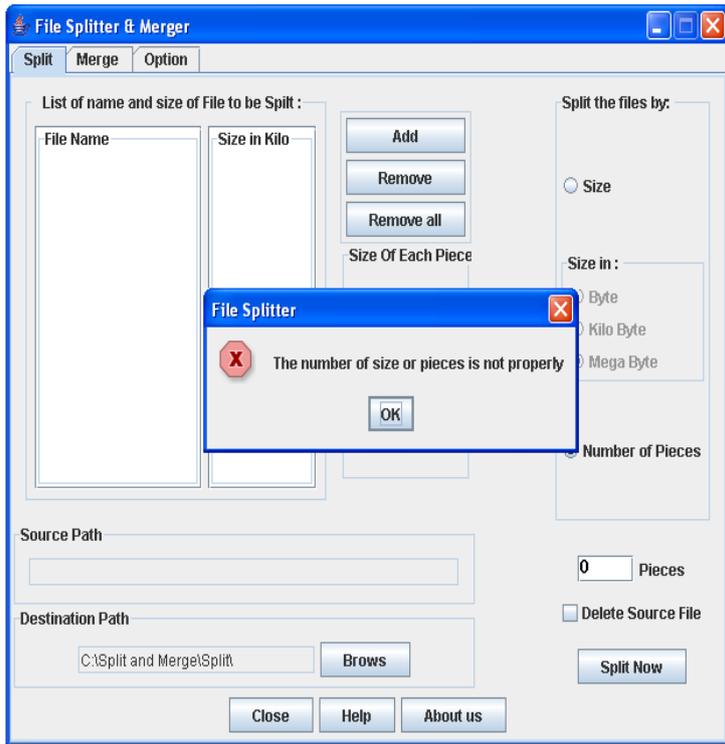


Fig (17) Error Message for invalid input number

Fig (18) final message for successful split

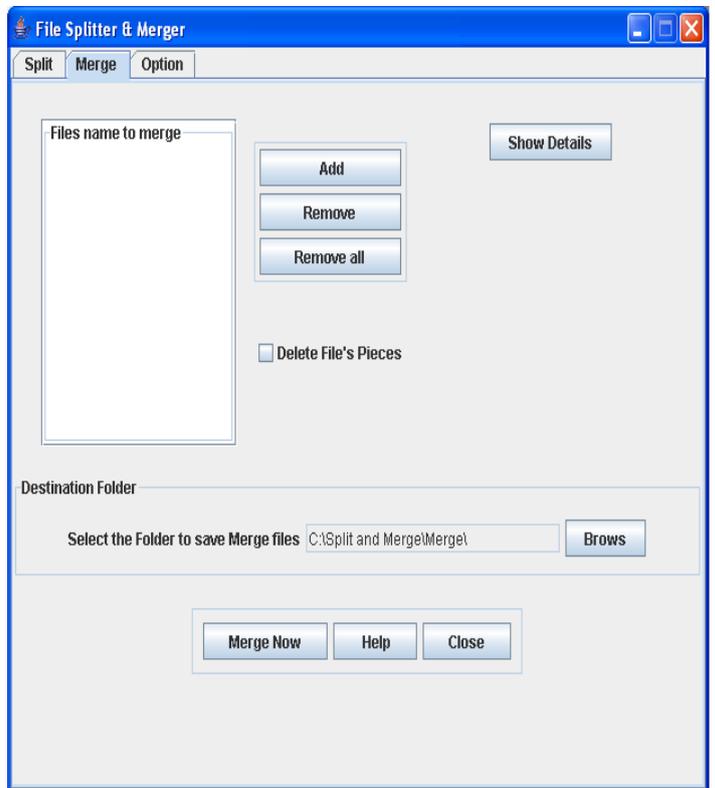
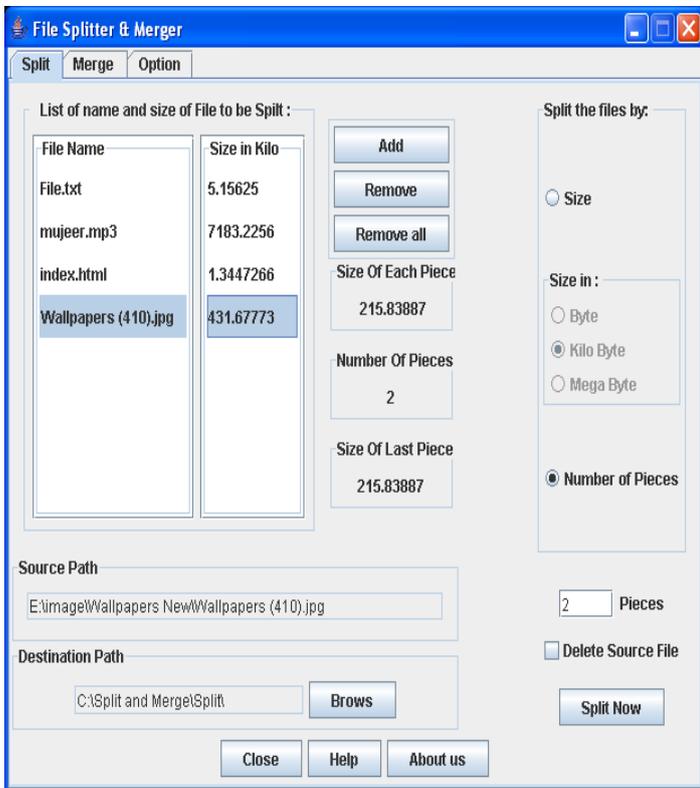


Fig (19) Window after split

Fig (20) Merge tab window

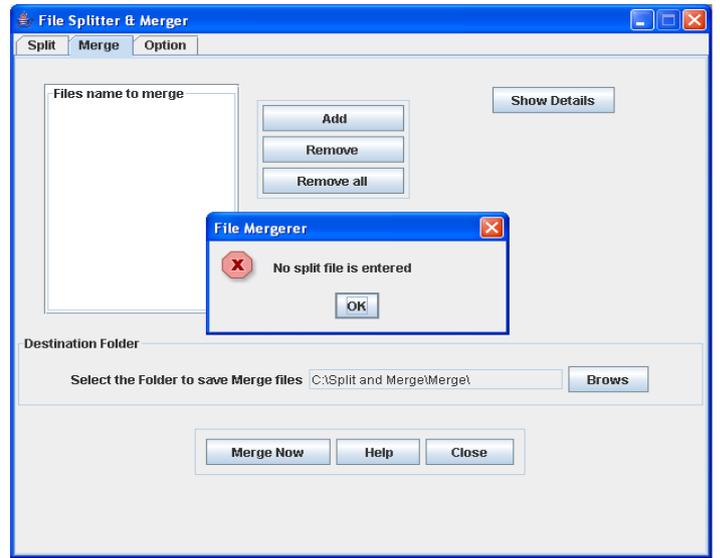
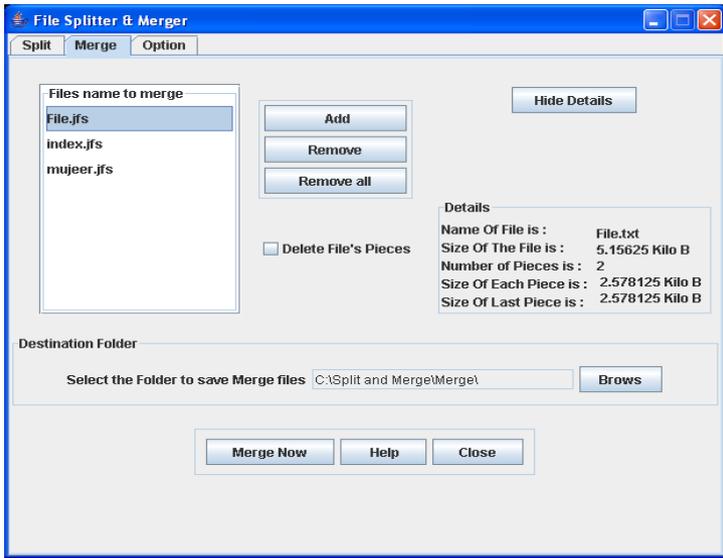


Fig (21) Show details frame

Fig (22) Error Message for no split file entered

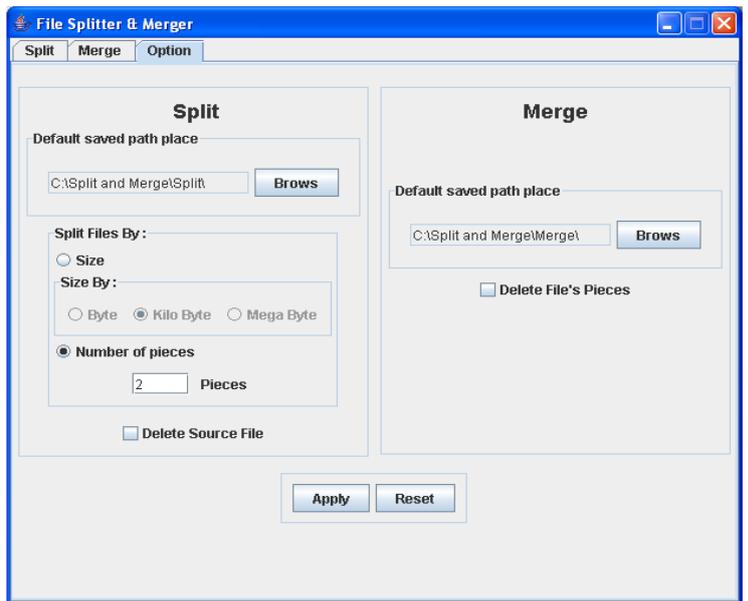
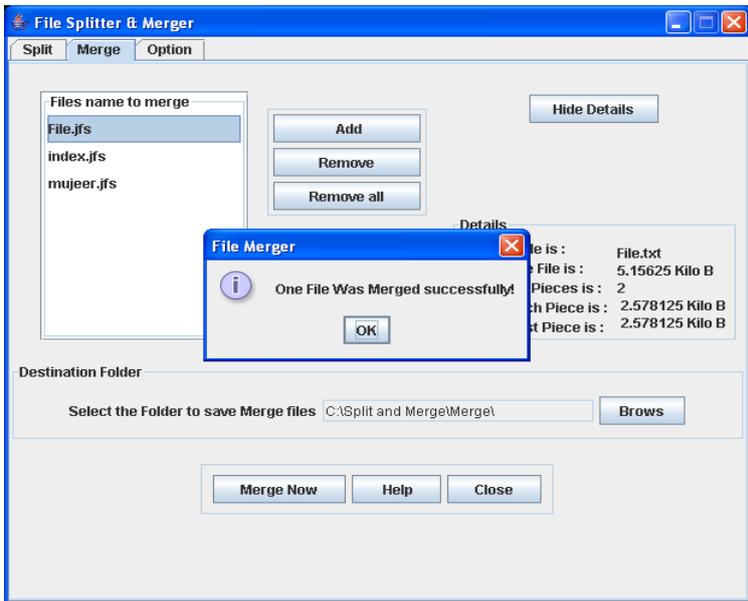


Fig (23) Message for Merge success

Fig (24) Option tab window

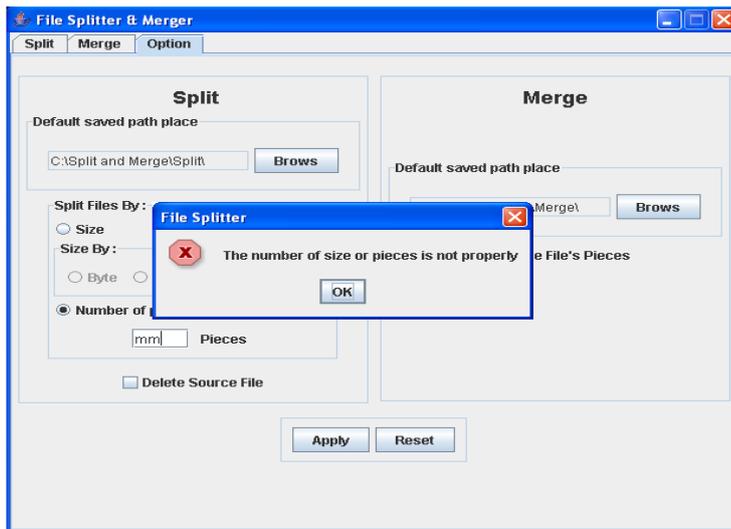


Fig (25) Error Message for invalid input to text field

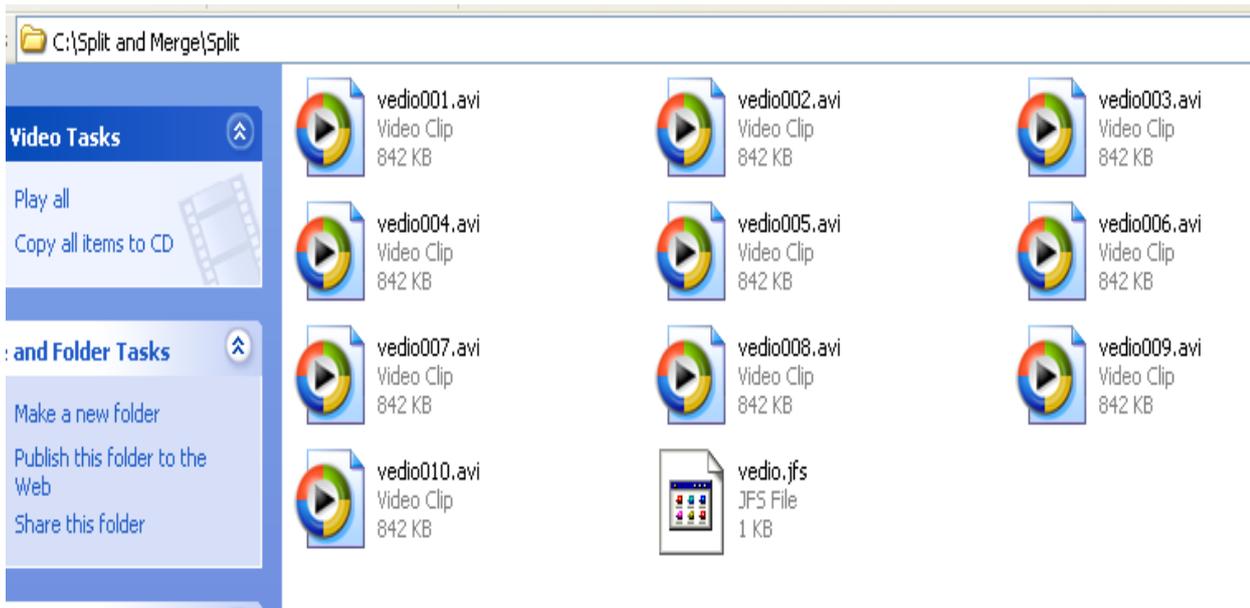


Fig (26) Splitting video file with avi extension

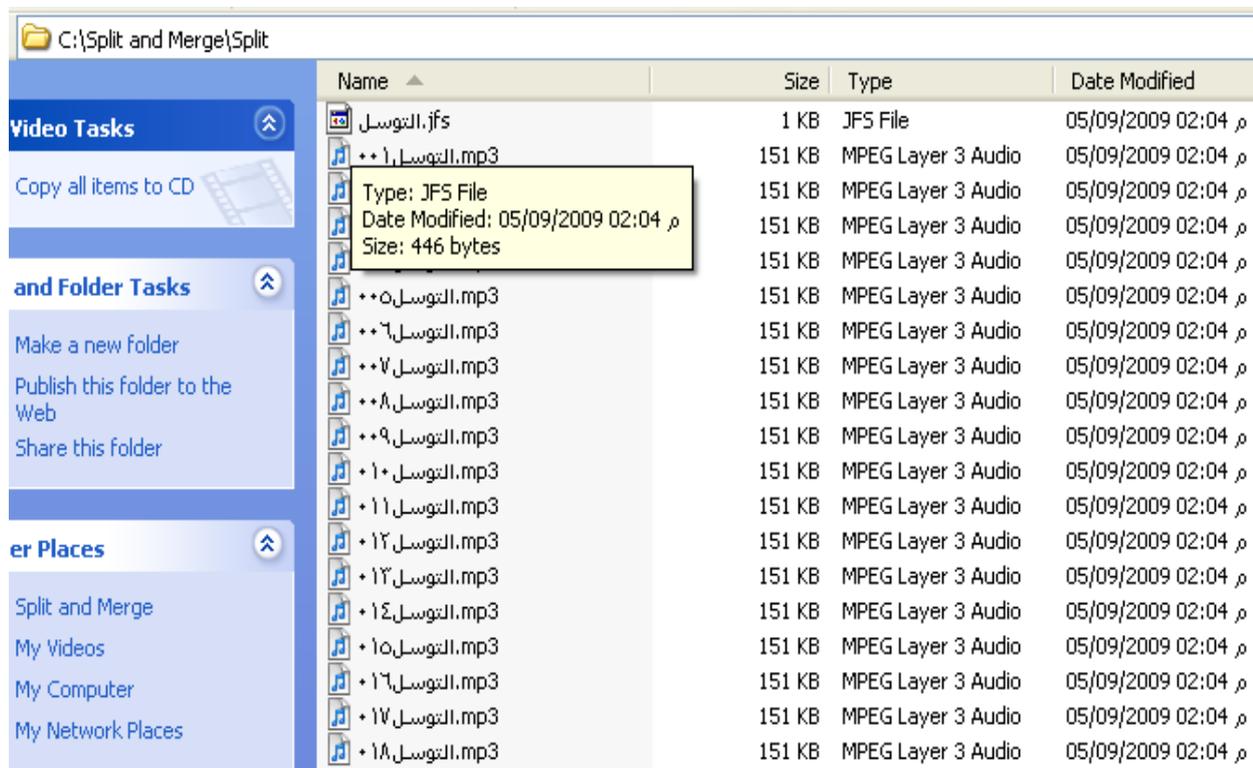
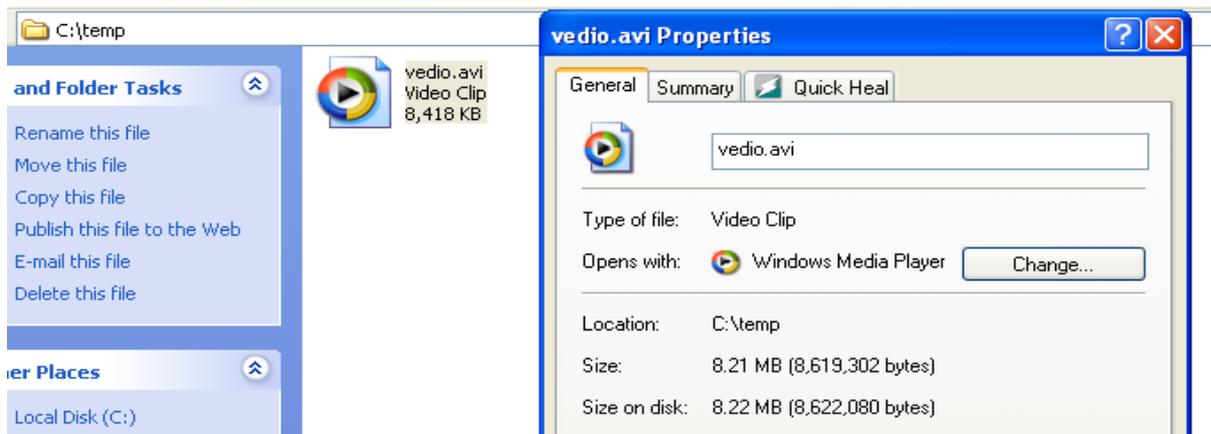


Fig (27) Splitting audio file with mp3 extension

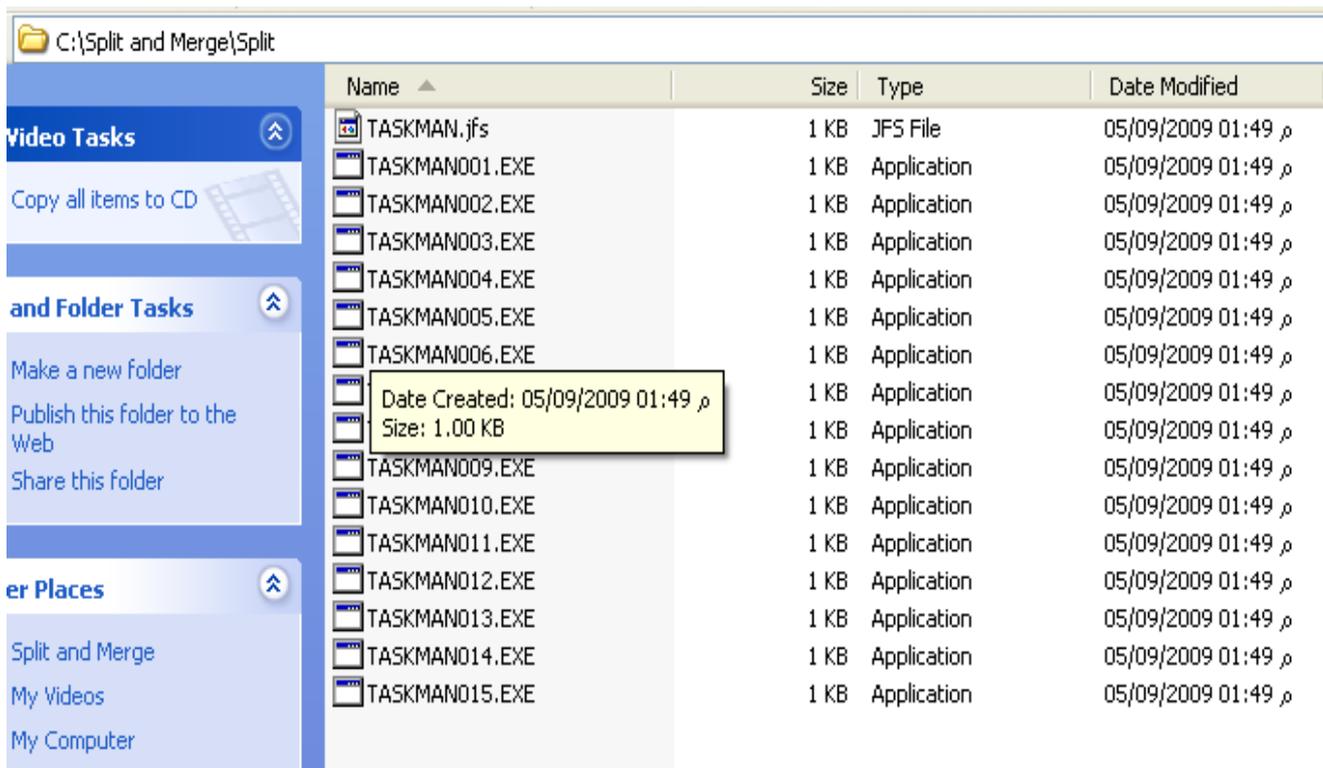
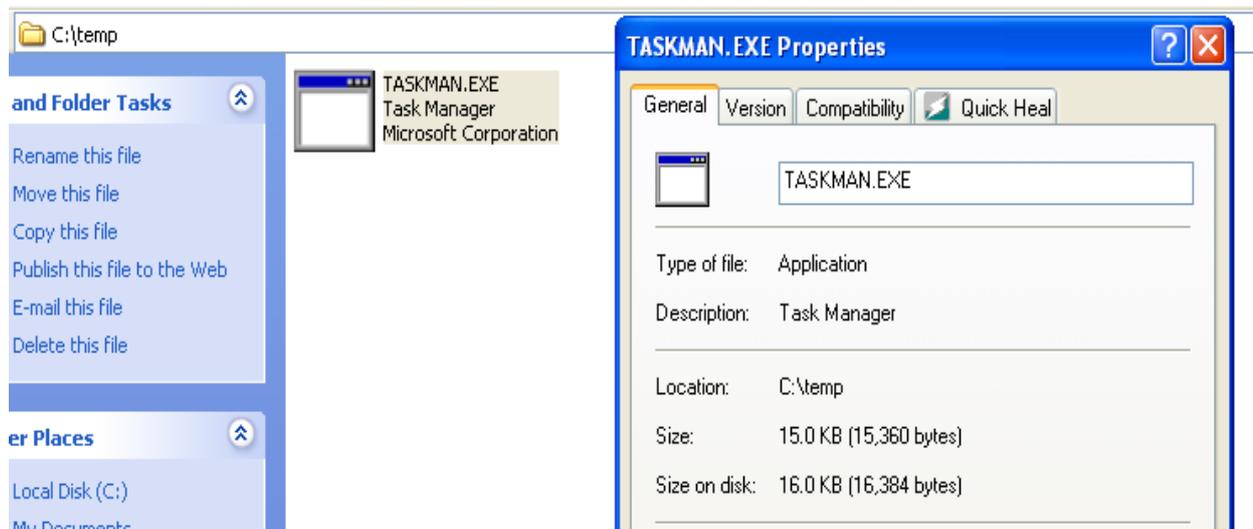


Fig (28) Splitting executable file with exe extension