Software Requirements Specification

for

BRIC

Requirements for Version 0.8.0

Prepared by Panagiotis Vasileiadis

Introduction to Software Engineering, Aristotle University

01/04/2014

Table of Contents

1. Introduction

1.1 Purpose

BRIC is a cross-platform multithreading batch image processing application whose features are Convert, Resize, Rotate, Watermark images. Multiple file types are supported for input and output. More specifically, supported input file types are jpg, jpeg, png, gif, tiff, tif, bmp, pdf, psd (photoshop), pnm (raw), pnm (ascii), pbm, ppm, pgm, wbmp and supported output file types are jpg, jpeg, png, pnm, pgm, pbm, ppm, gif, bmp, tif, tiff, pdf . Since it is a Java application, it runs on any platform where Java Runtime Environment is installed and uses image processing libraries Apache Sanselan, ImageIO and ImageJ. It is licensed under GNU General Public License version 3.0 (GPLv3) and the latest version until now is BRIC 0.8.0 .

1.2 Document Conventions

This document follows the IEEE standard for Software Requirements Specification documents. It describes the behavior and the requirements of BRIC, version number 0.8.0 and it is up to date on 01/04/2014. Any changes in the software requirements after this date cannot be reflected in this document. The document was developed in coordination with the project's developer and therefore, it is assumed to be as accurate as possible. Finally, in section 3 the software features are described in the same order as tabs are displayed from left to right in the main window of the program and parentheses are used to enclose the available values of the software features.

1.3 Intended Audience and Reading Suggestions

This document is written in plain, simple English so that even a newly introduced computer user can read it and comprehend it and it is intended for end users, developers, testers and documentation writers who are seeking a detailed documentation of the software. Each mentioned type of reader has his own needs and uses of the software. For a more thorough analysis on this matter, please refer to section 2.3 . Section 2 provides an overall description of the software and section 3 describes the functional requirements of the project, particularly useful to all of the aforementioned groups. Section 4 discusses the external interface requirements and finally, in section 5 you can find the nonfunctional requirements of the project. Each section is divided in subsections where different matters are discussed. It is suggested that experienced readers should skip the introductory section, as it is referring mostly to topics that inexperienced users would find useful and interesting.

1.4 Project Scope

BRIC is one of many programs used for converting, resizing, rotating, watermarking images and supports multiple file types for input and output. It uses batch processing and one of its main goals is to run on all operating systems. Therefore, it is written in Java and utilizes image processing libraries Apache Sanselan, ImageIO and ImageJ. Also, it aims at keeping the GUI as simple as possible, so it does not become a hindrance to the overall casual user experience. Furthermore, it manages to be fast because of being able to use multithreading. In the Figure 1.4.1 below, there is a screenshot of the software's main window:

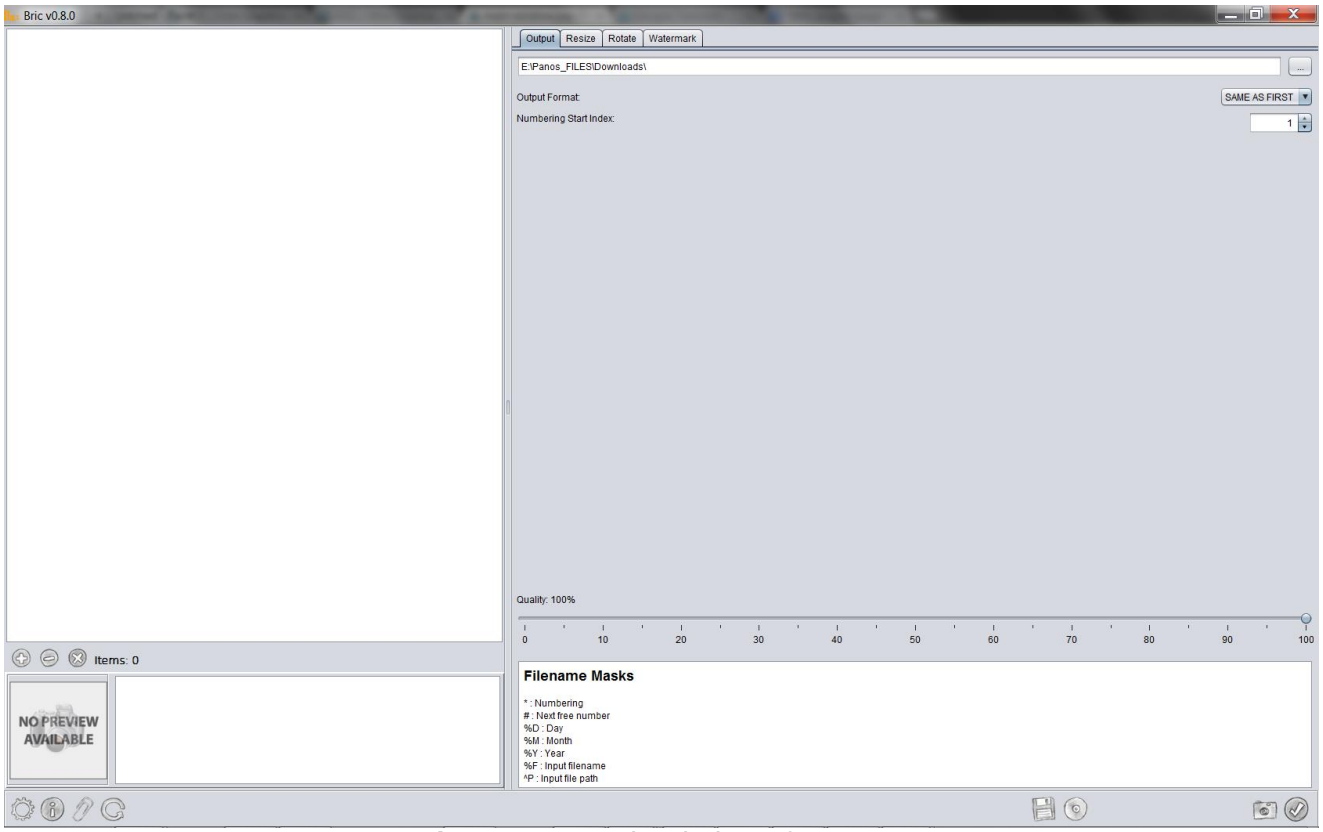

Figure 1.4.1 Main window of BRIC

1.5 References

- ✓ Official page – Home of BRIC:
  http://bric.sourceforge.net/
- ✓ Project BRIC page on Sourceforge:
  http://sourceforge.net/projects/bric/
- ✓ Download BRIC:
  http://sourceforge.net/projects/bric/files/latest/download
- ✓ Wiki BRIC page:
  http://sourceforge.net/p/bric/wiki/Home/
- ✓ Report Bugs or view issues with the project:
  http://sourceforge.net/p/bric/bugs/
- ✓ Feature Requests:
  http://sourceforge.net/p/bric/feature-requests/

2. Overall Description

2.1 Product Perspective

BRIC falls into graphics-editors software category. There are other similar programs which convert, resize, rotate and watermark images by using multithreading and batch processing, such as Image Tools, but BRIC has the advantage of supporting more file types for input/output and being able to run on all operating systems instead of only GNOME, Xfce (Linux Distributions) and Windows. Since BRIC is a Java application, it runs on any platform where Java Runtime Environment is installed and uses image processing libraries Apache Sanselan, ImageIO and ImageJ. Finally, it manages to be both simple and fast in its use.

2.2 Product Features

The major features of BRIC are summarized in the list below:

- ✓ reads jpg, jpeg, png, gif, tiff, tif, bmp, pdf, psd (photoshop), pnm (raw), pnm (ascii), pbm, ppm, pgm, wbmp
- ✓ writes jpg, jpeg, png, gif, tiff, tif, bmp, pdf, pnm, pbm, ppm, pgm
- ✓ batch processing
- ✓ multithreading (incredibly fast)
- ✓ convert, resize, rotate, watermark (image/text/html)
- ✓ easy watermark placement
- ✓ random rotation
- ✓ resize filters, rendering, sharpen
- ✓ cross-platform (Windows, Linux, Mac, etc...)

2.3 User Classes and Characteristics

This application is intended for use by 4 user classes:

- ✓ End Users of varying expertise who may want to convert, resize, rotate, watermark images. All users will use the program in English, since it is the only available language in software's preferences.
- ✓ Developers who are interested in expanding and making the application better, find and correct bugs and generally contribute to the BRIC community.
- ✓ Testers who use the beta versions of the program and test it in many ways for bugs and errors, then submit the data to the bug tracking system.
- ✓ Documentation Writers who can use this document for assistance in documenting the program's functions and features. This is important, documentation and help files are vital to any project's completeness.

2.4 Operating Environment

Since BRIC is a Java application, it runs on any platform (Windows, Linux, Mac, etc... - either 32 or 64 bit) where Java Runtime Environment is installed. In general, it does not consume many resources and almost every computer will be able to handle it properly.

2.5 Design and Implementation Constraints

Future developers should know about Java as BRIC is written in that language, Java Swing User Interface and image processing libraries Apache Sanselan, ImageIO and ImageJ which BRIC uses in order to successfully expand the program. Knowledge of multithreading programming is required too.

2.6 User Documentation

Online Documentation:

- ✓ Official page – Home of BRIC
  (How to launch application from terminal):
  http://bric.sourceforge.net/

- ✓ Wiki BRIC page
  (Program's Screenshots):
  http://sourceforge.net/p/bric/wiki/Home/

2.7 Assumptions and Dependencies

BRIC is written in Java, so the only requirement for it to work is Java Runtime Environment to be installed in any operating system. As for development requirements, these are Java programming language, Java Swing User Interface, multithreading programming and image processing libraries Apache Sanselan, ImageIO and ImageJ.

3. System Features

3.1 Manage Input / Output Files

In program's main window on the left side of the interface, there is:

- ✓ an "Add" option in order to add files to the input file list
- ✓ a "Remove" option which removes the selected files from the input file list
- ✓ a "Clear" option which removes all files from the input file list at once.

An example of how to add input files is shown below:

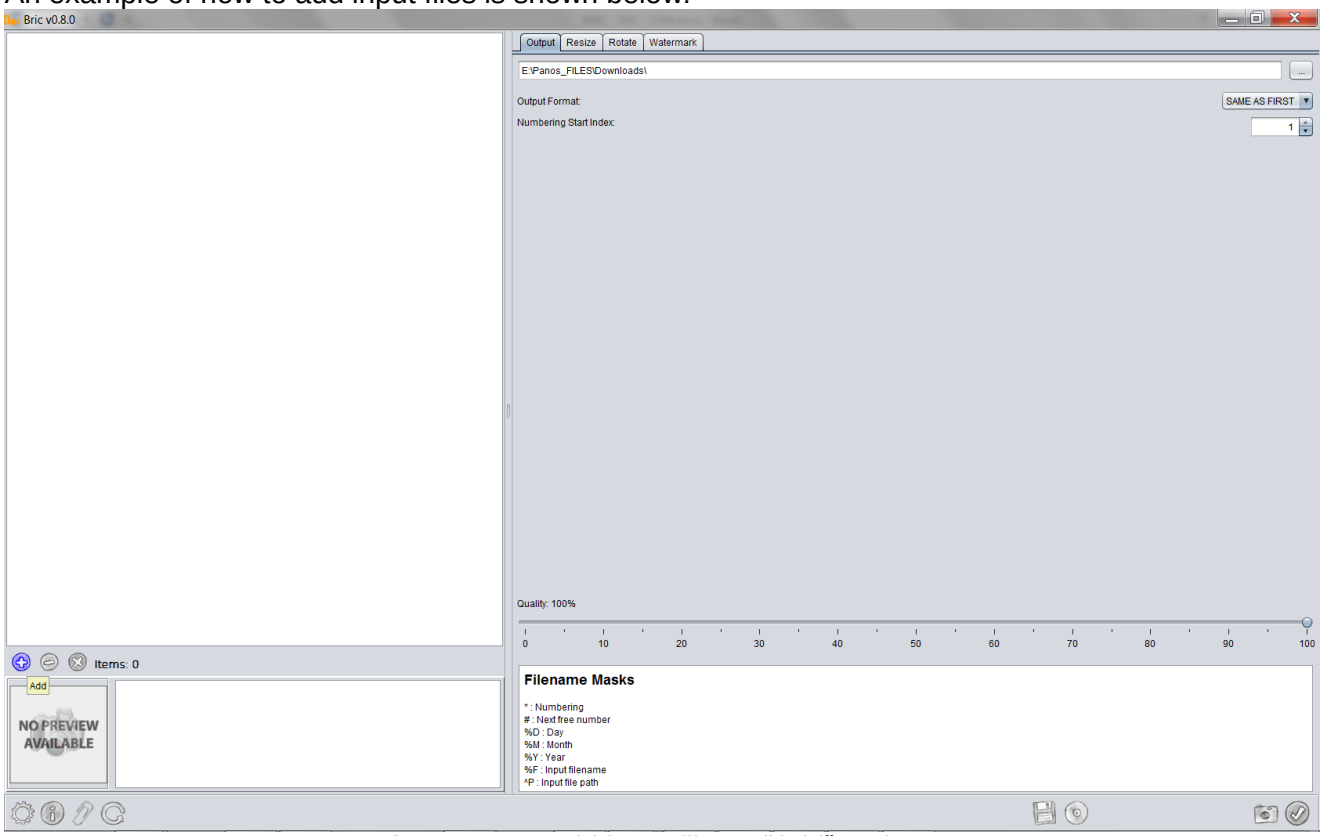

Figure 3.1.1 Add input files – "Add" option

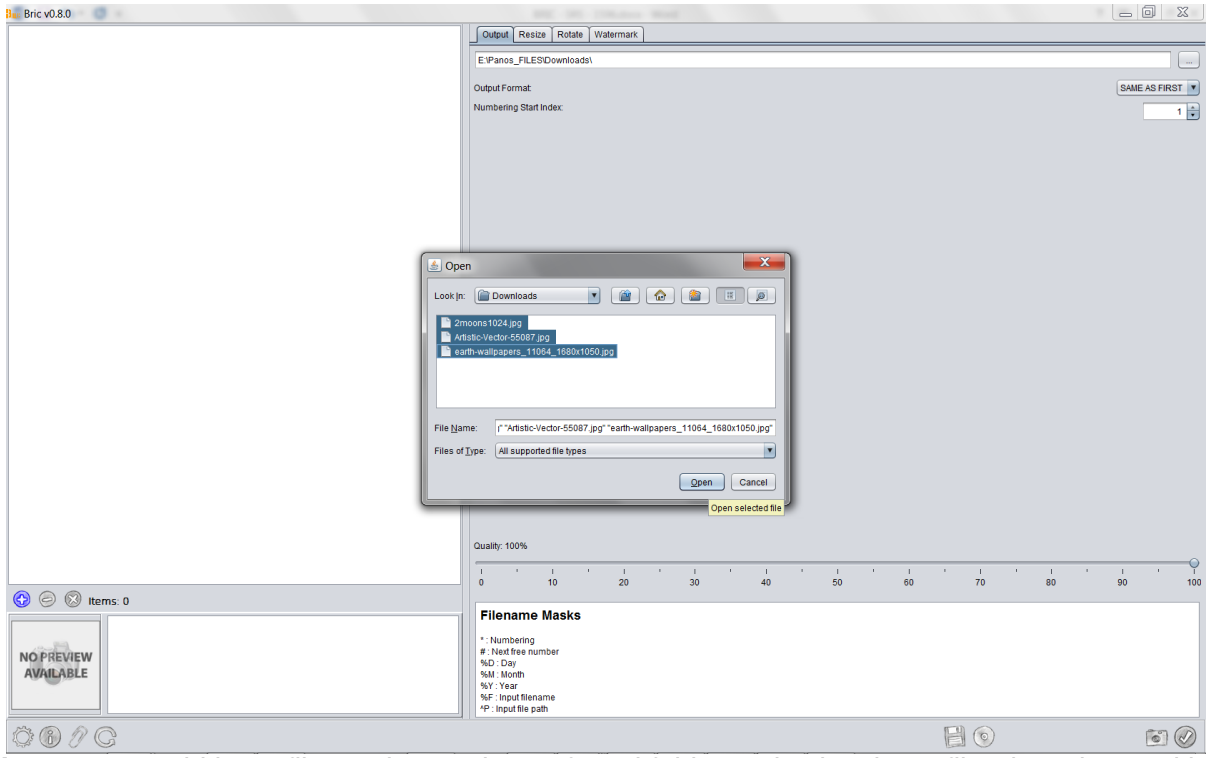

Figure 3.1.2 Add input files – choose the preferred folder and select input files (to select multiple inputs press ctrl while selecting) and click on “Open” button

At this point, all input files are displayed in the list above “Add” option and an input file counter is shown on the right of these buttons as seen below in Figure 3.1.3 .
A preview of the selected input file is displayed at the left-down corner, as well as input file’s information such as filename, image dimensions and filesize as seen below in Figure 3.1.3 .
Also, a “Hide Preview” option is available and it is located under the preview box as seen below in Figure 3.1.3 . By clicking once on this button, input file preview box is hidden as seen below in Figure 3.1.4 and by clicking twice on it, input file’s information are hidden too, as seen below in Figure 3.1.5 . By another click on it, the preview box is reshown as seen below in Figure 3.1.6 .

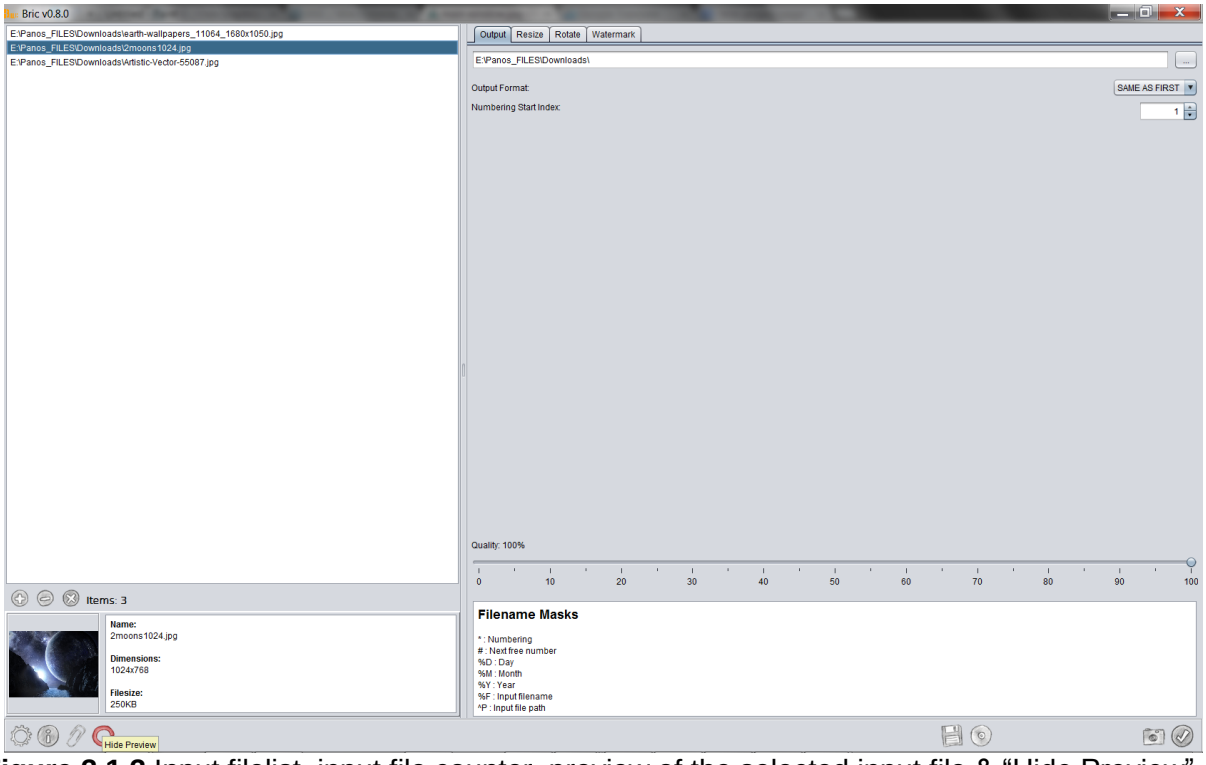

Figure 3.1.3 Input filelist, input file counter, preview of the selected input file & “Hide Preview” option

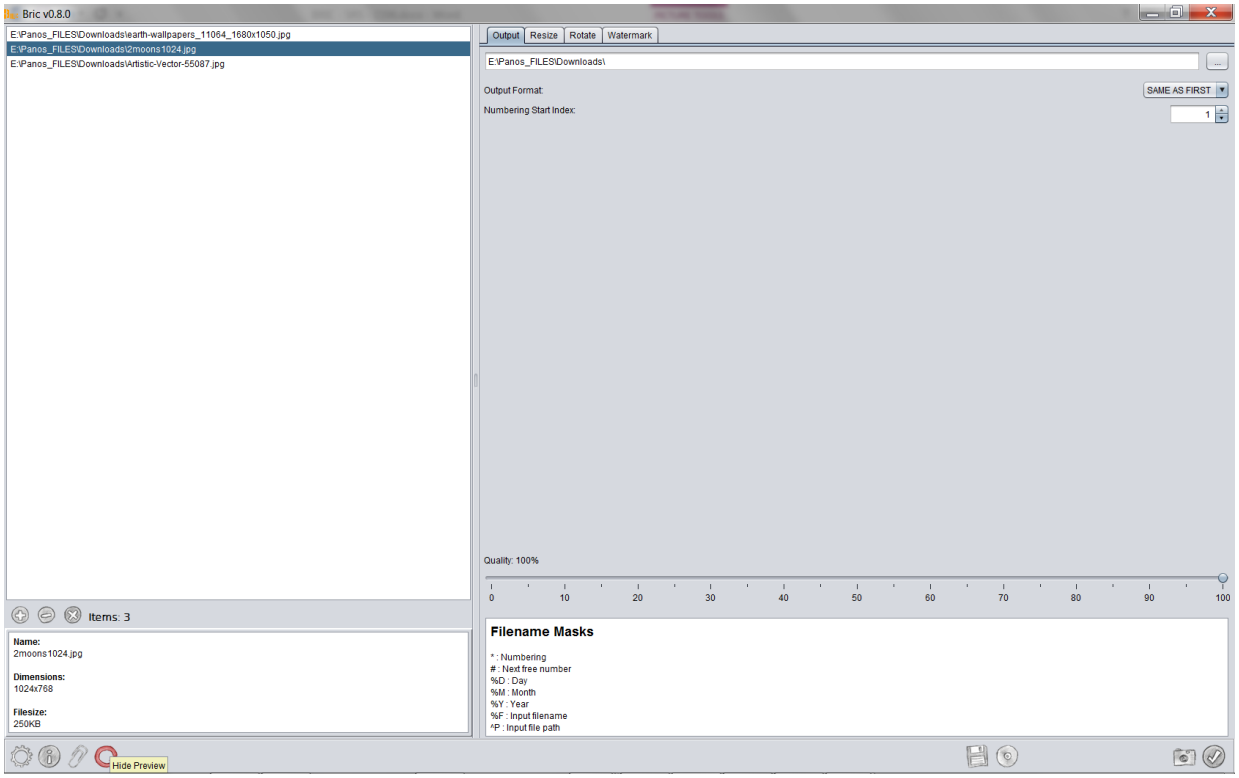

Figure 3.1.4 Input file preview box is hidden

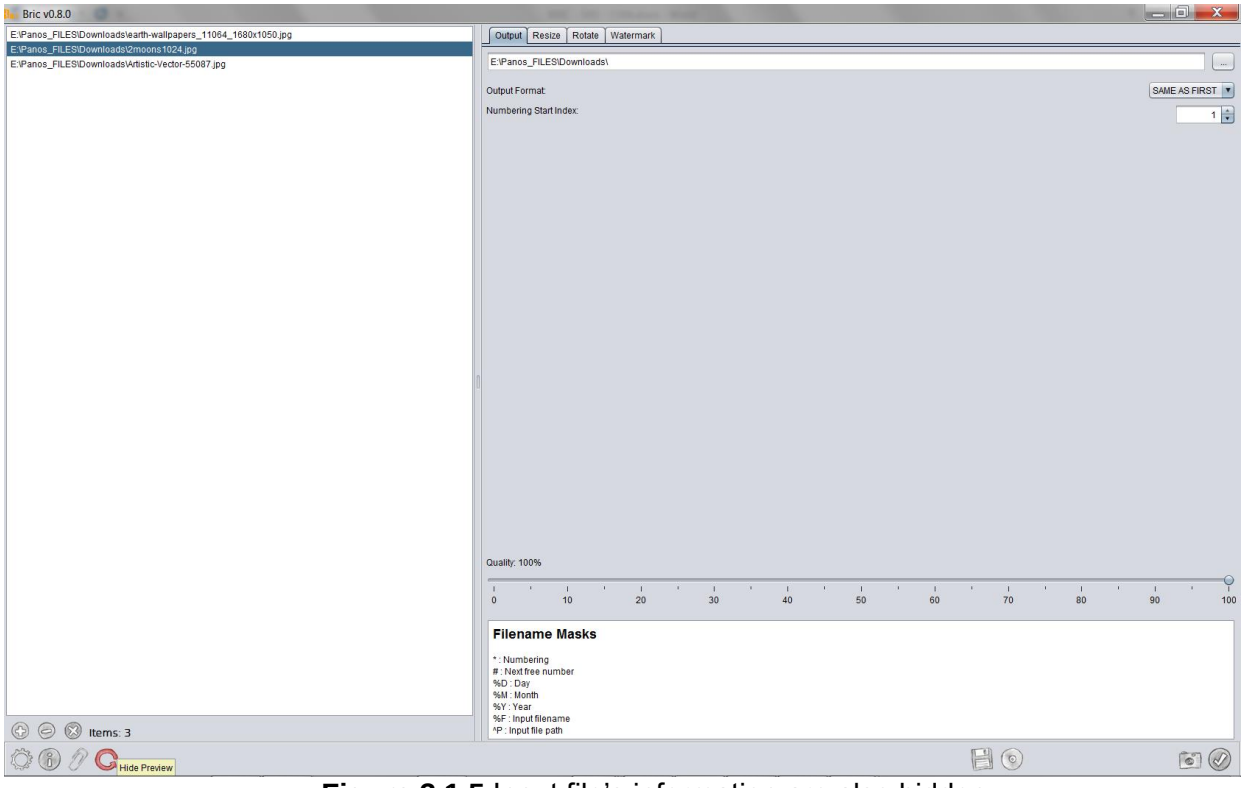

Figure 3.1.5 Input file's information are also hidden

In the Output tab, the user can select:

- ✓ the directory where the output files are to be saved
- ✓ the output format (SAME AS FIRST, jpg, jpeg, png, gif, bmp, tif, tiff, pgm, pbm, pnm, ppm, pdf)
- ✓ the numbering start index of the output files
- ✓ the quality of the output images (0 to 100 % – 100 % is recommended)

More specifically, selecting the output path is done by clicking on "…" button at the right-up corner, then choosing the preferred folder and finally by clicking on "Save" button as seen in the example below:

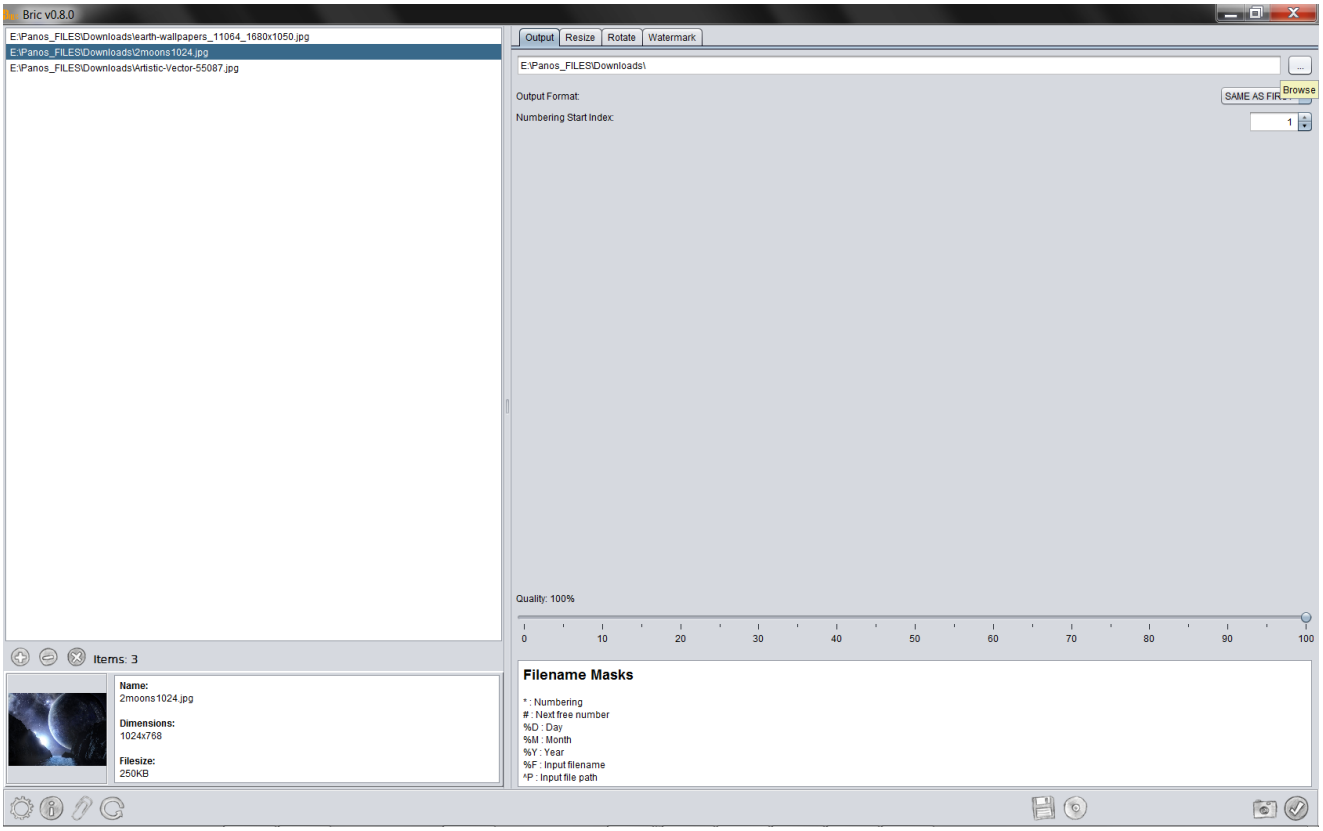

Figure 3.1.6 Input file preview box is reshown & output path selection – "…" button

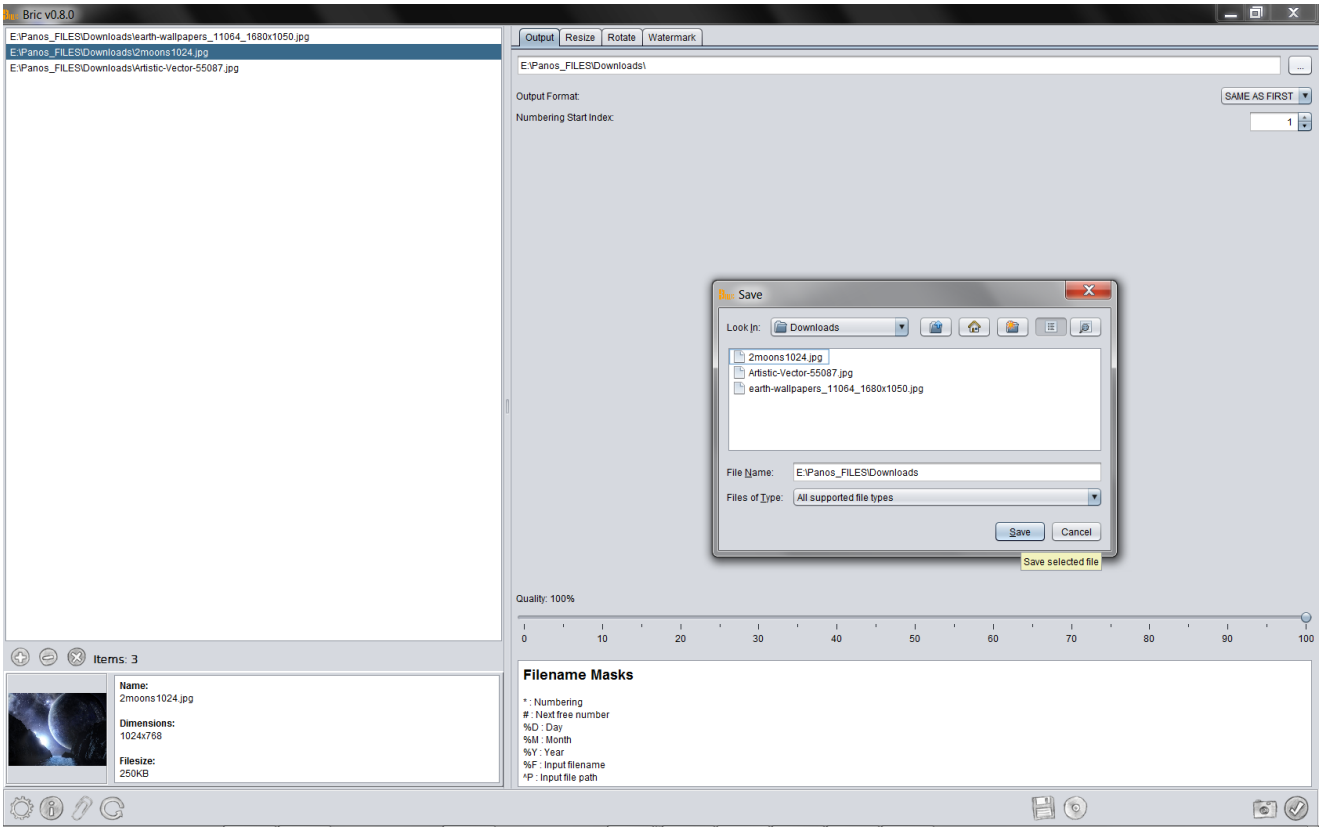

Figure 3.1.7 Output path selection – choose the preferred folder and click on "Save" button

An example of how to remove input files is shown below:

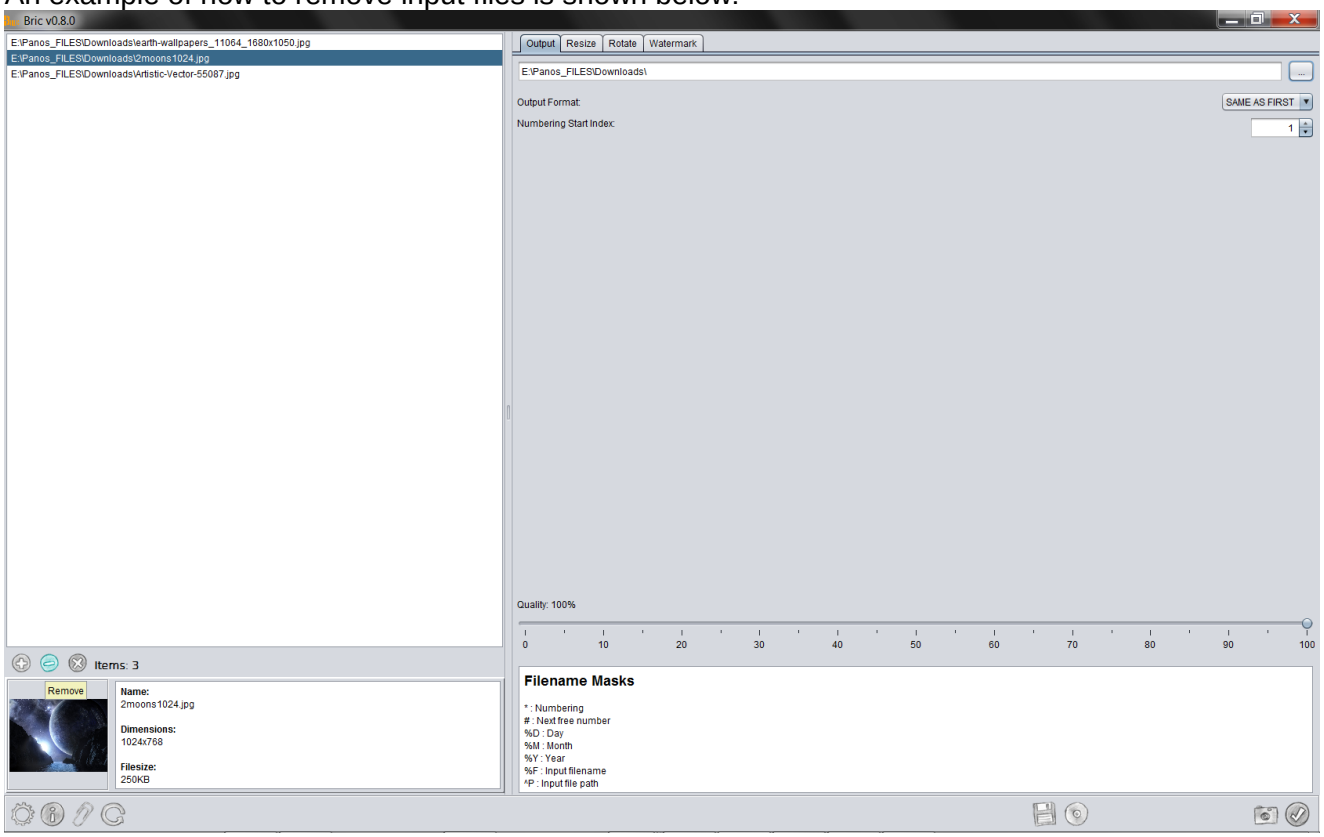

Figure 3.1.8 Remove input files – "Remove" option

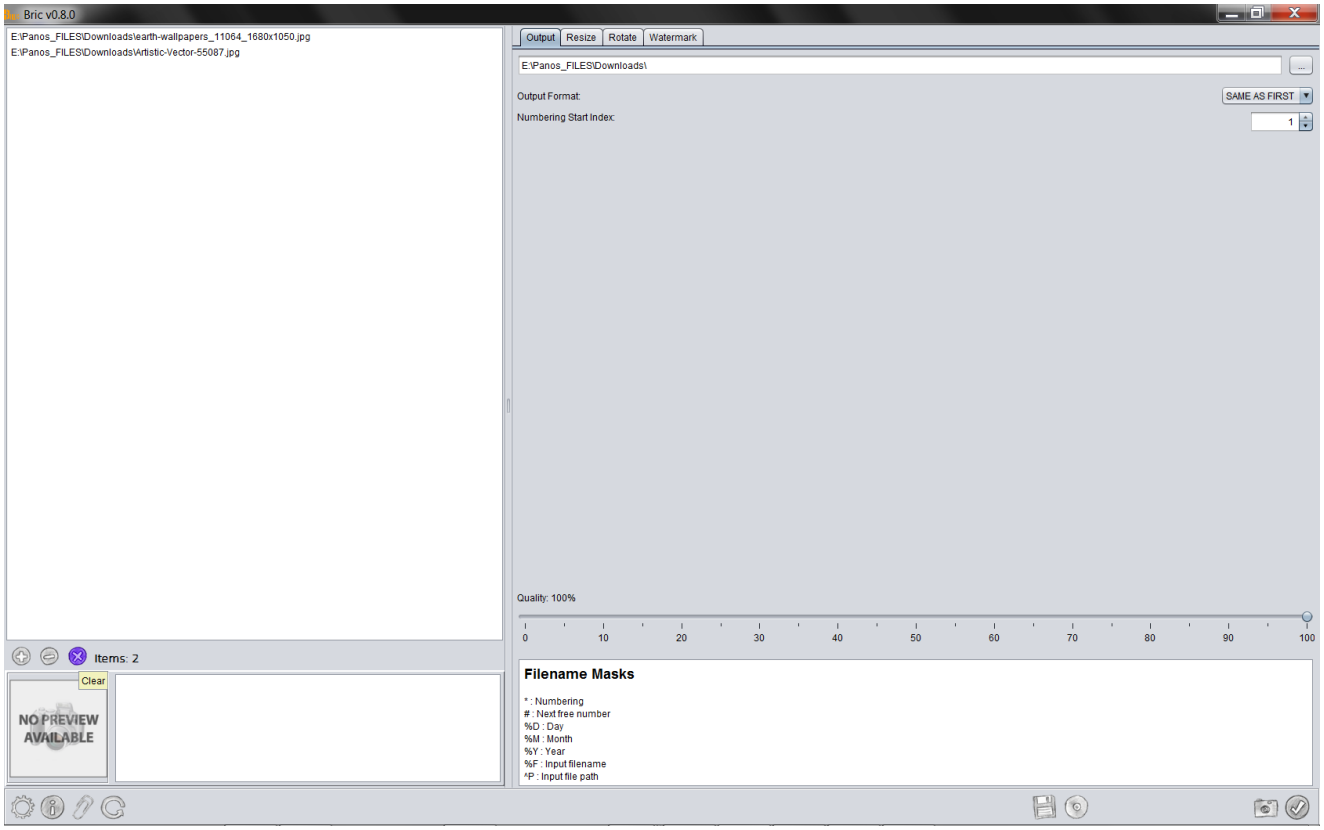

Figure 3.1.9 Remove input files – selected file was removed from the input file list & "Clear" option

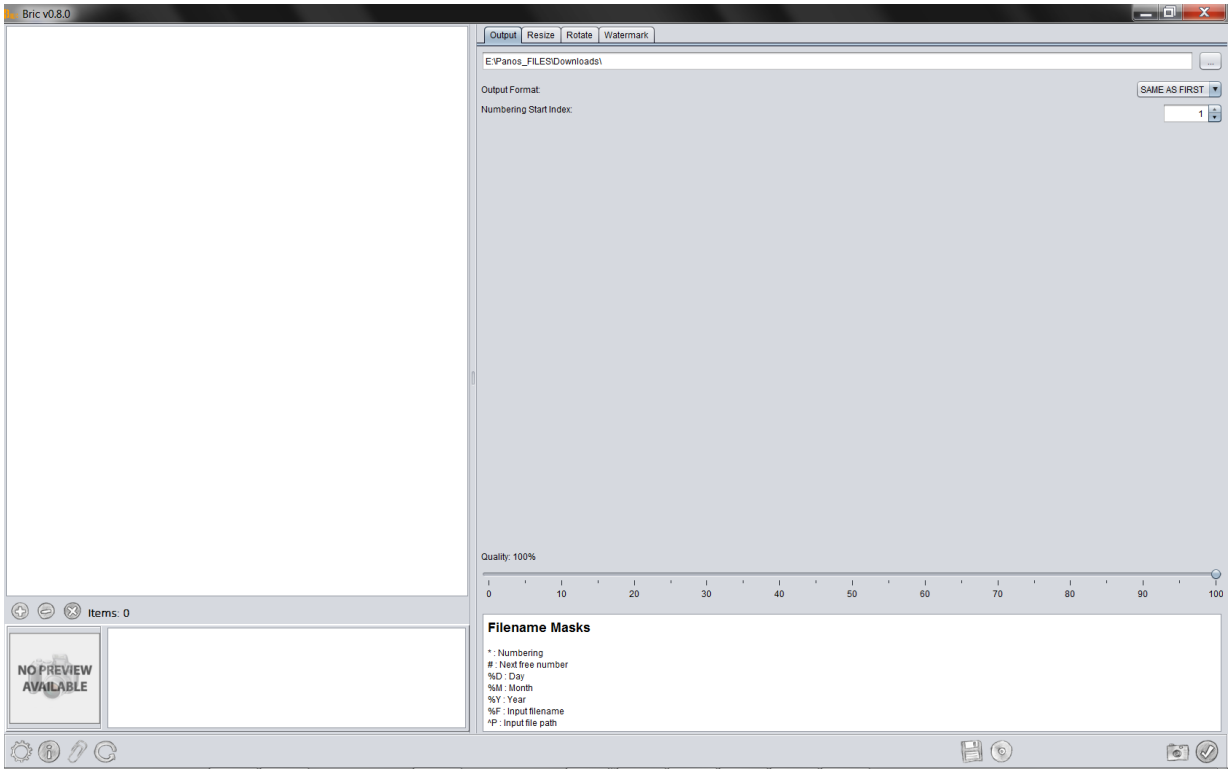

Figure 3.1.10 Remove input files – all files were removed from the input file list

3.2 Convert

For the converting process to commence, firstly an image needs to be inserted as input. Then, in the Output tab the user should select the output path, the output format, the numbering start index of the output files and the quality of the output images (the exact procedure is described in section "3.1 Manage Input / Output Files"). Finally by clicking on "Start" button at the right-down corner, the converting begins. An example of how to convert a png to bmp image is shown below:

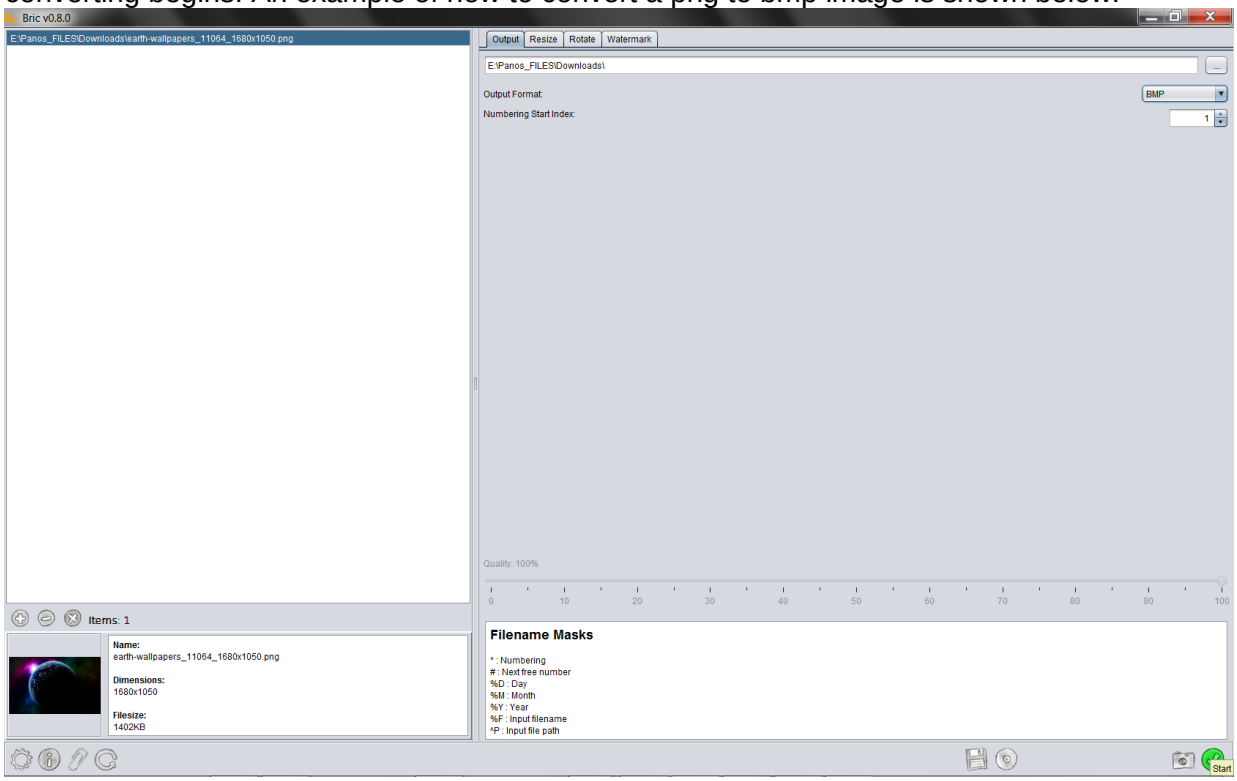

Figure 3.2.1 Convert png to bmp – "Start" option

Also, a “Preview” option is available which shows the output preview of the selected input image and it is located next to “Start” option as seen below in Figure 3.2.2 .

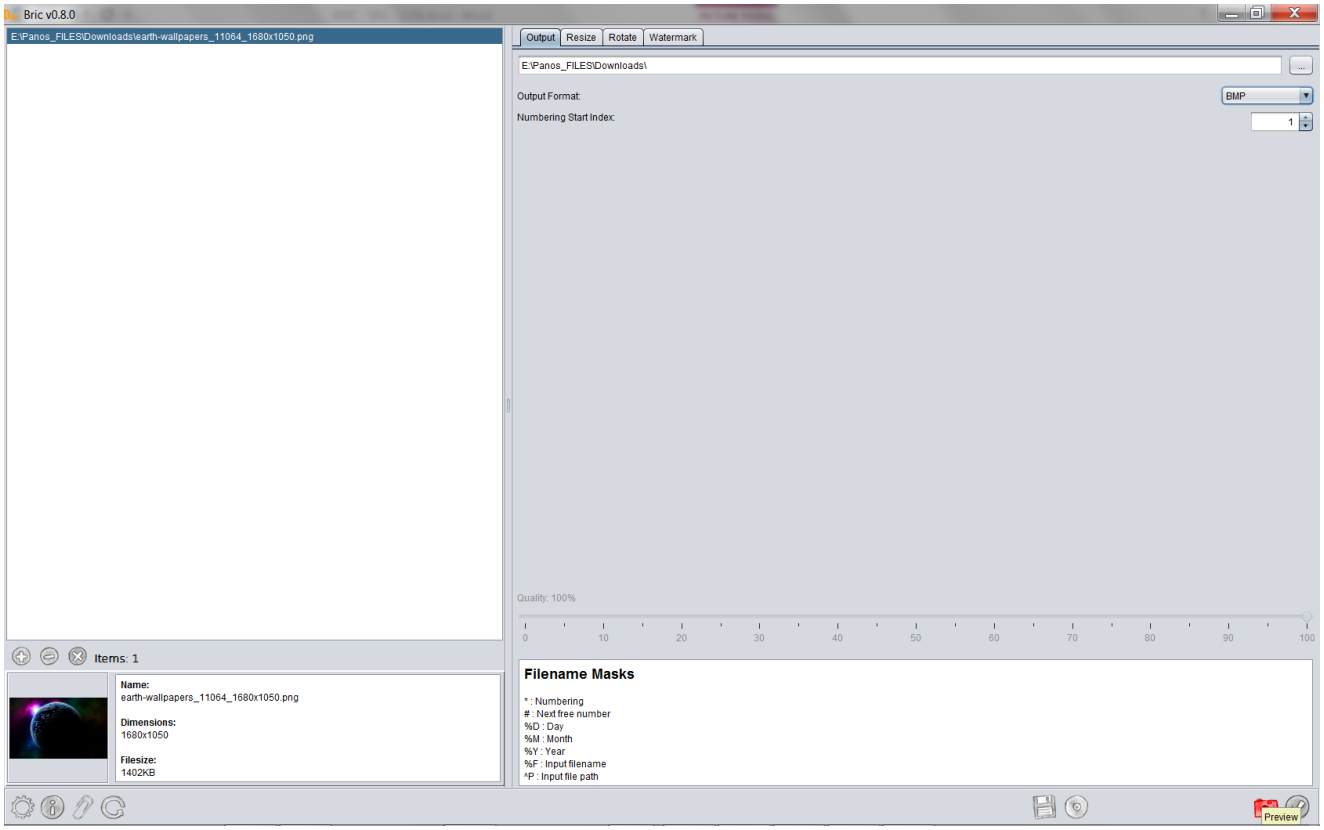

Figure 3.2.2 Convert png to bmp – “Preview” option

Output preview is shown in the Figure 3.2.3 below:

Figure 3.2.3 Convert png to bmp – output preview

3.3 Resize

For the resizing and converting process to commence, firstly an image needs to be inserted as input. Then, in the Output tab the user should select the output path, the output format, the numbering start index of the output files and the quality of the output images (the exact procedure is described in section “3.1 Manage Input / Output Files”).
Also, clicking in the Resize tab and then clicking on “Enable” option is needed as seen below in Figure 3.3.1 . In the Resize tab, the user can select:

- ✓ “Enable” option
- ✓ the width and height of the resized images either in pixels or in percentage
- ✓ “Maintain Aspect Ratio” option
- ✓ “Consider Orientation” option
- ✓ “Resize Filter” advanced option (auto, BiCubic, BiCubicHighFreqResponse, BiLinear, Lanczos3, MultiStep, Nearest_Neighbor, Bell, Box, Hermite, Mitchell, Triangle, Thumpnail)
- ✓ “Rendering” advanced option (Quality, Speed)
- ✓ “Sharpen” advanced option (none, Normal, Oversharpened, Soft, Verysharp)
- ✓ “Antialising” advanced option

Finally by clicking on “Start” button at the right-down corner, the resizing and converting begins.
An example of how to resize and convert a png to bmp image is shown below:

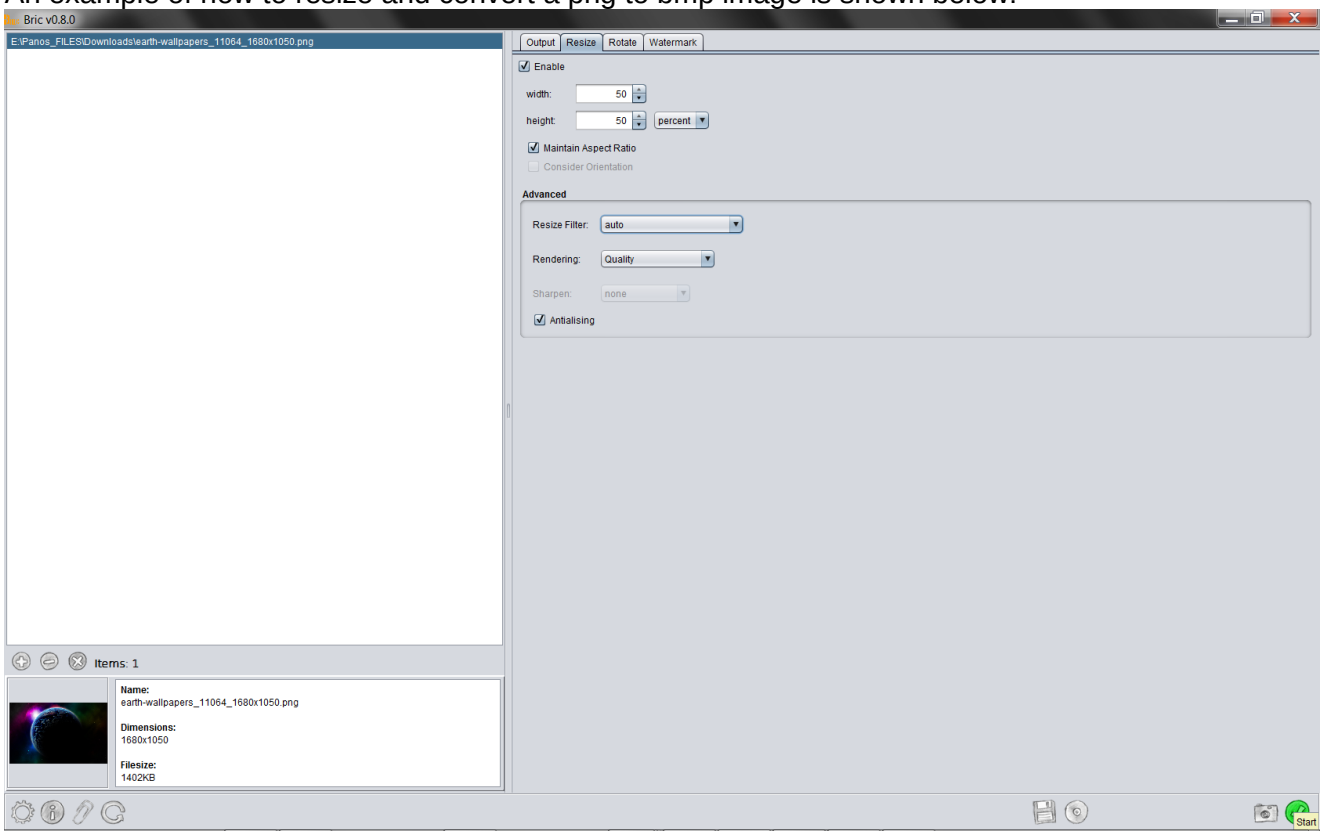

Figure 3.3.1 Resize images – “Enable” & “Start” options

Also, a “Preview” option is available which shows the output preview of the selected input image and it is located next to “Start” option as described in section “3.2 Convert”.

3.4 Rotate

For the rotating and converting process to commence, firstly an image needs to be inserted as input. Then, in the Output tab the user should select the output path, the output format, the numbering start index of the output files and the quality of the output images (the exact procedure is described in section “3.1 Manage Input / Output Files”).
Also, clicking in the Rotate tab and then clicking on “Enable” option is needed as seen below in Figure 3.4.1 . In the Rotate tab, the user can select:

- ✓ “Enable” option
- ✓ “Custom” option
  - ➢ the angle of rotation in degrees
  - ➢ “Random” option which randomly rotates the images
    - “Different value for each image” option
    - “Limit” option
      - “from” option which is the minimum value (in degrees) of the interval
      - “to” option which is the maximum value (in degrees) of the interval
- ✓ “Predefined” option
  - ➢ “Actions” option (180º Upside Down, 90º Counter Clockwise, 90º Clockwise, Horizontal Flip, Horizontal + Vertical Flip, Vertical Flip)

Finally by clicking on “Start” button at the right-down corner, the rotating and converting begins.
An example of how to rotate and convert a png to bmp image is shown below:

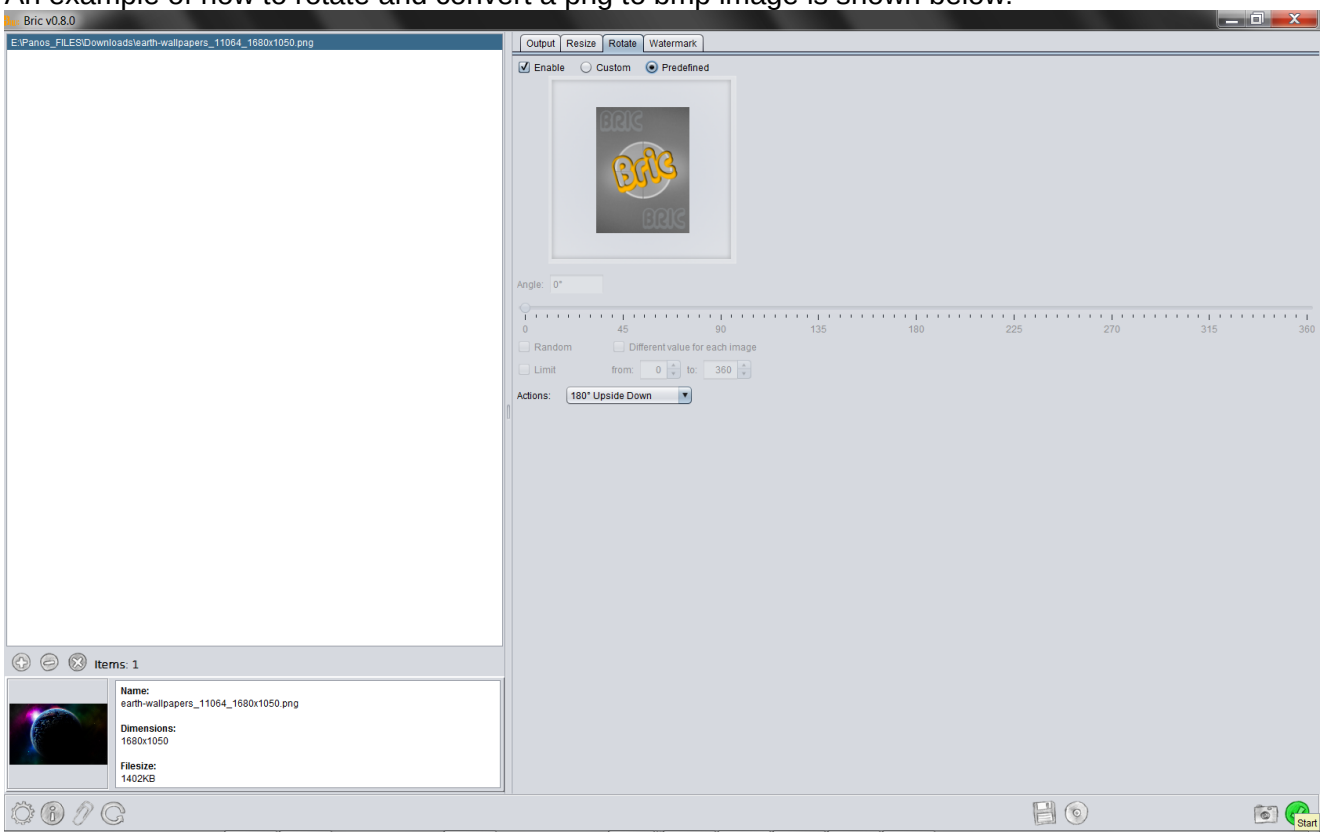

Figure 3.4.1 Rotate images – “Enable” & “Start” options

Also, a “Preview” option is available which shows the output preview of the selected input image and it is located next to “Start” option as described in section “3.2 Convert”.

3.5 Watermark

For the watermarking and converting process to commence, firstly an image needs to be inserted as input. Then, in the Output tab the user should select the output path, the output format, the numbering start index of the output files and the quality of the output images (the exact procedure is described in section “3.1 Manage Input / Output Files”).
Also, clicking in the Watermark tab and then clicking on “Enable” option is needed as seen below in Figure 3.5.1 . In the Watermark tab, the user can select:

- ✓ “Enable” option
- ✓ “Opacity” option (value in percentage)
- ✓ “Rotate” option (value in degrees)
- ✓ “Pattern” option
  - ➢ “Single” option
  - ➢ “Tiled” option
    - “R” option which adds multiple rows of watermarks (‘0’ for infinite copies)
    - “C” option which adds multiple columns of watermarks (‘0’ for infinite copies)
- ✓ “Text Editor” field which edits watermark text and shows its preview
  - ➢ “F” option which selects font type/name, font style and font size
  - ➢ “C” option which selects font color
  - ➢ mode option (Text Mode, HTML Mode, Image Mode → browse for watermark image)

Finally by clicking on “Start” button at the right-down corner, the watermarking and converting begins. An example of how to watermark and convert a png to bmp image is shown below:

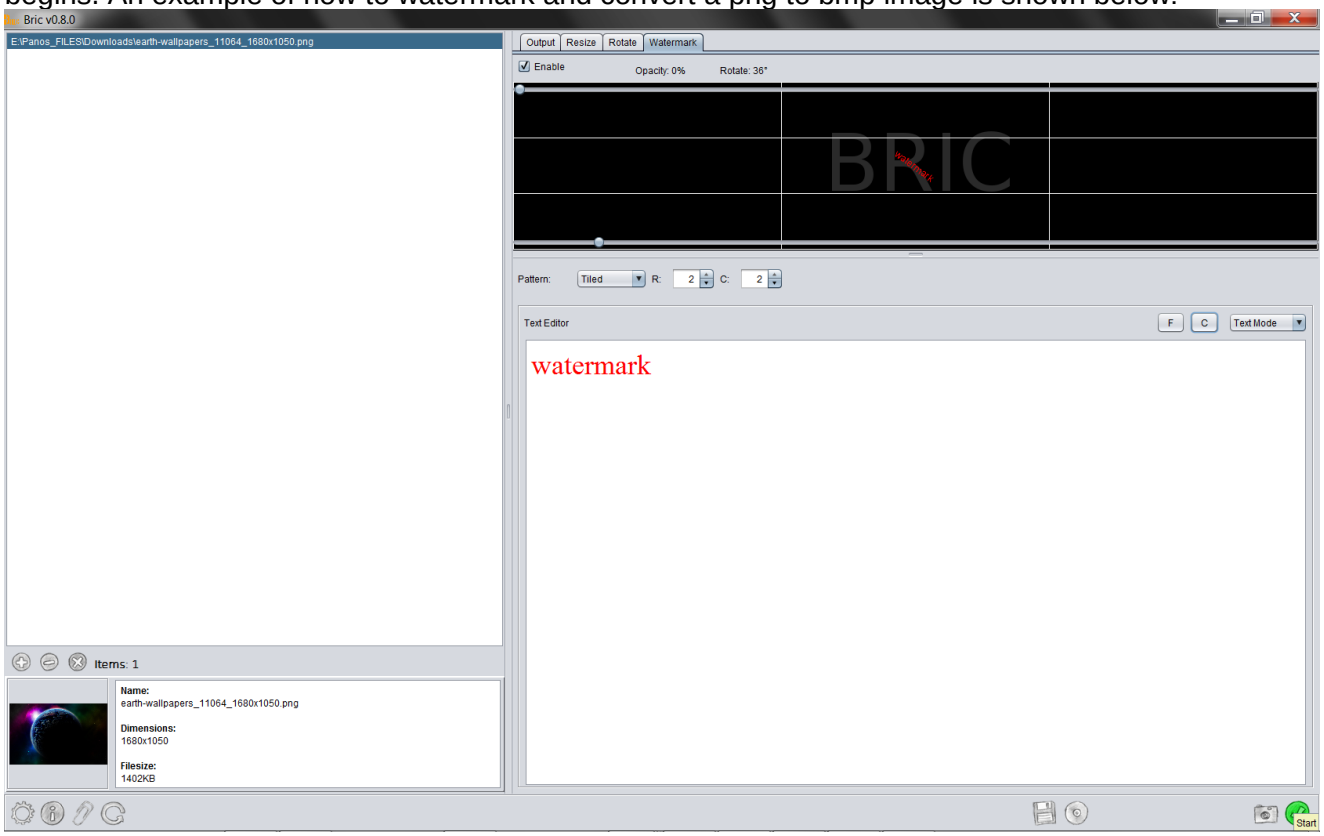

Figure 3.5.1 Watermark images – “Enable” & “Start” options

Also, a “Preview” option is available which shows the output preview of the selected input image and it is located next to “Start” option as described in section “3.2 Convert”.

3.6 Configure Preferences

The “Preferences” button is at the left-down corner in the program’s main window as seen below in Figure 3.6.1 . By clicking on it, the Preferences window opens as seen in Figure 3.6.1 too.

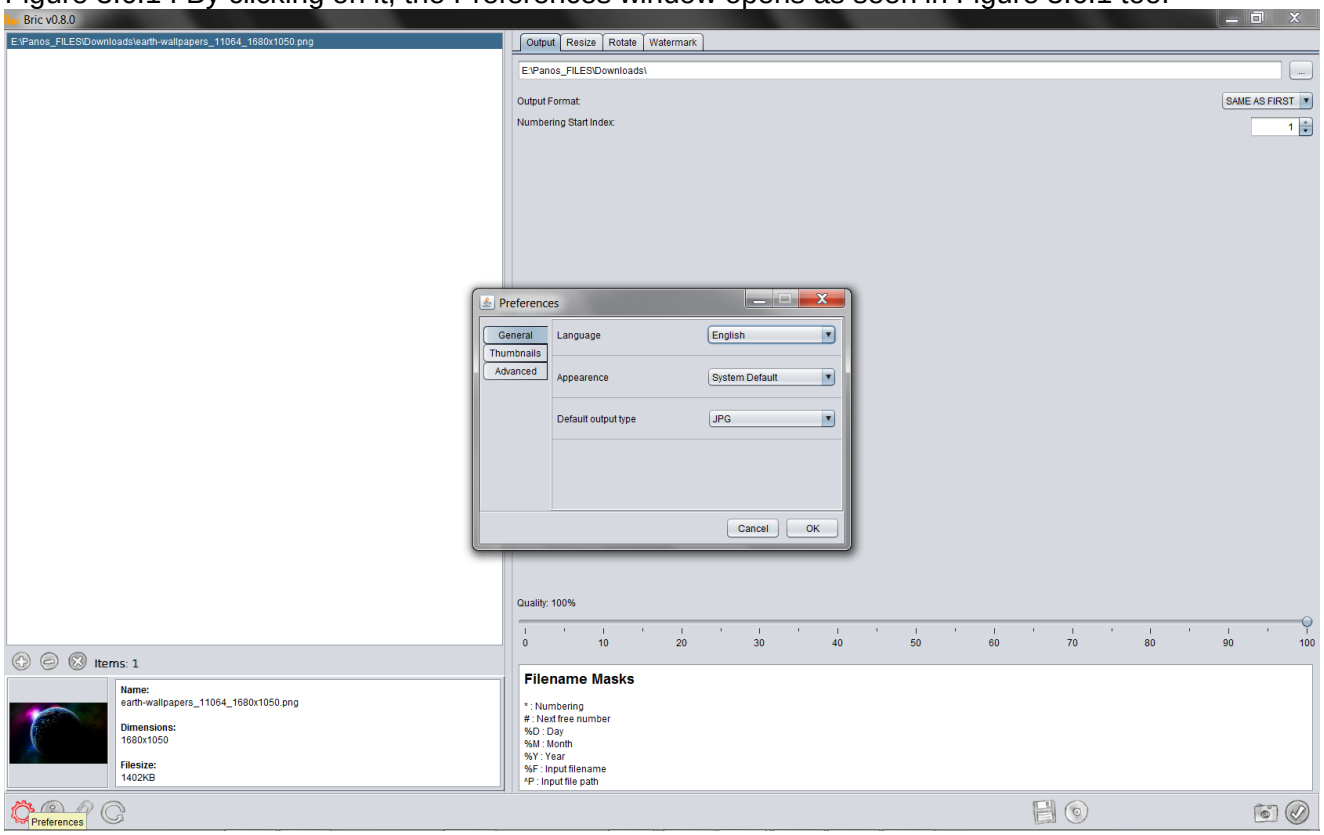

Figure 3.6.1 Configure Preferences – “Preferences” button & window

In the Preferences window, there are 3 tabs (General, Thumbnails, Advanced).

In the General tab, the user can select:

- ✓ “Language” option (English)
- ✓ “Appearance” option (System Default)
- ✓ “Default output type” option (jpg, jpeg, png, gif, bmp, tif, tiff, pgm, pbm, pnm, ppm)

In the Thumbnails tab, the user can select:

- ✓ “Generate Thumbnails” option (on import, on demand)
- ✓ “Collect Metadata” option (on import, on demand)
- ✓ “Import/Export window close after (ms)” field

In the Advanced tab, the user can select:

- ✓ “Threads per process on import” option (Auto, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
- ✓ “Threads per process on export” option (Auto, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

When the selections are done, the settings are applied by clicking the “OK” button in the Preferences window as seen in the Figure 3.6.1 .

In order to save preferences and selections, there is a “Save” option at the lower side of the interface in the program’s main window as seen below in Figure 3.6.2 .
An example of how to do it is shown below:

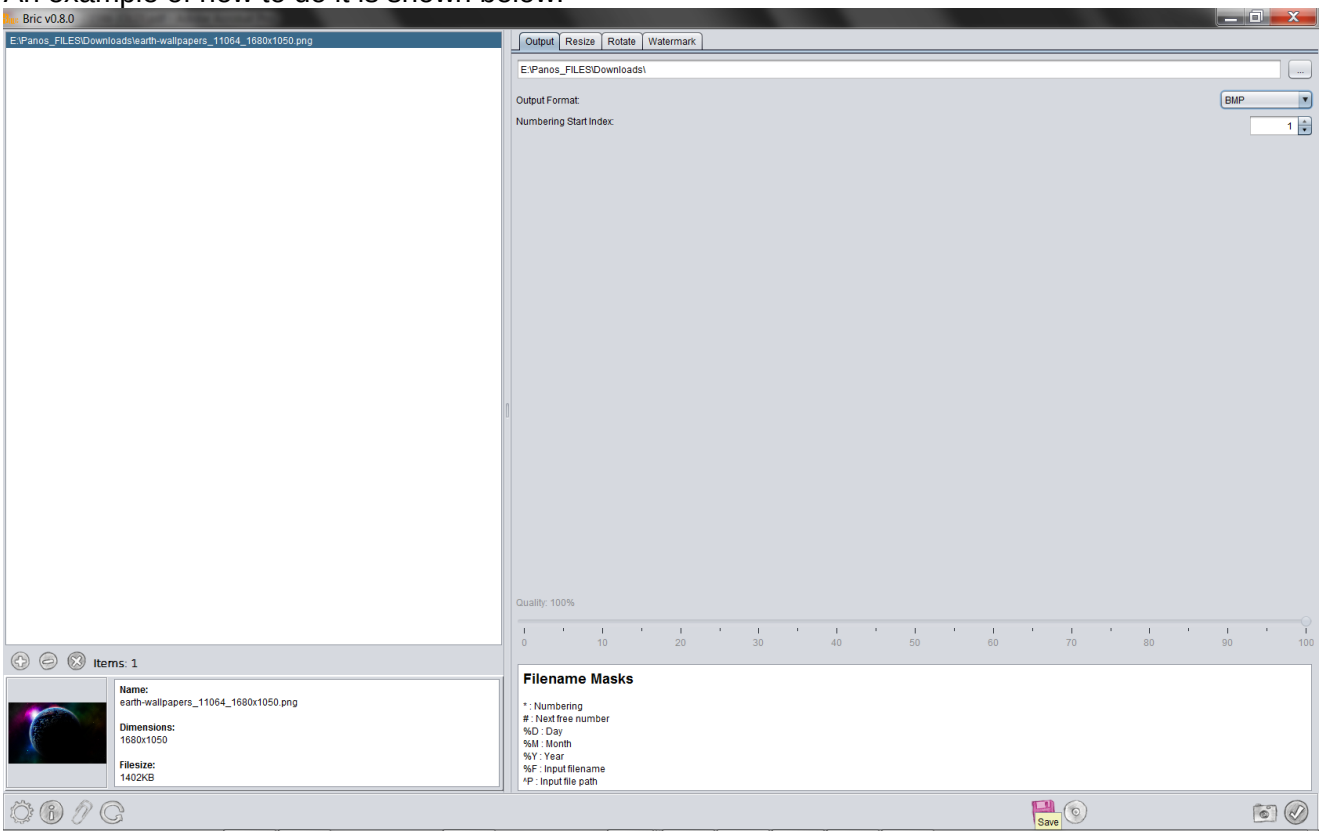

Figure 3.6.2 Save preferences & selections – “Save” option

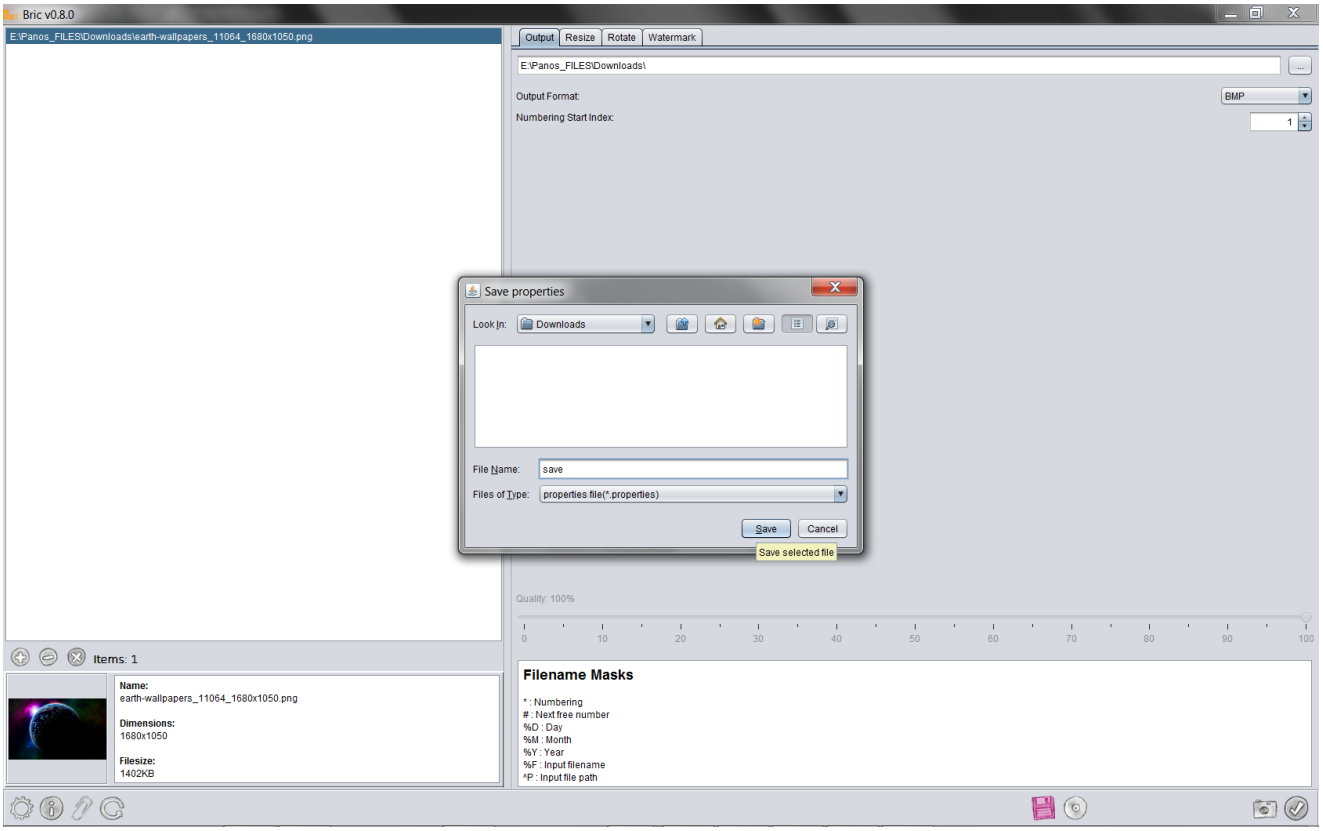

Figure 3.6.3 Save preferences & selections – choose the preferred folder, enter filename and click on “Save” button

In order to load saved preferences and selections, there is a “Load” option at the lower side of the interface in the program’s main window as seen below in Figure 3.6.4 .
An example of how to do it is shown below:

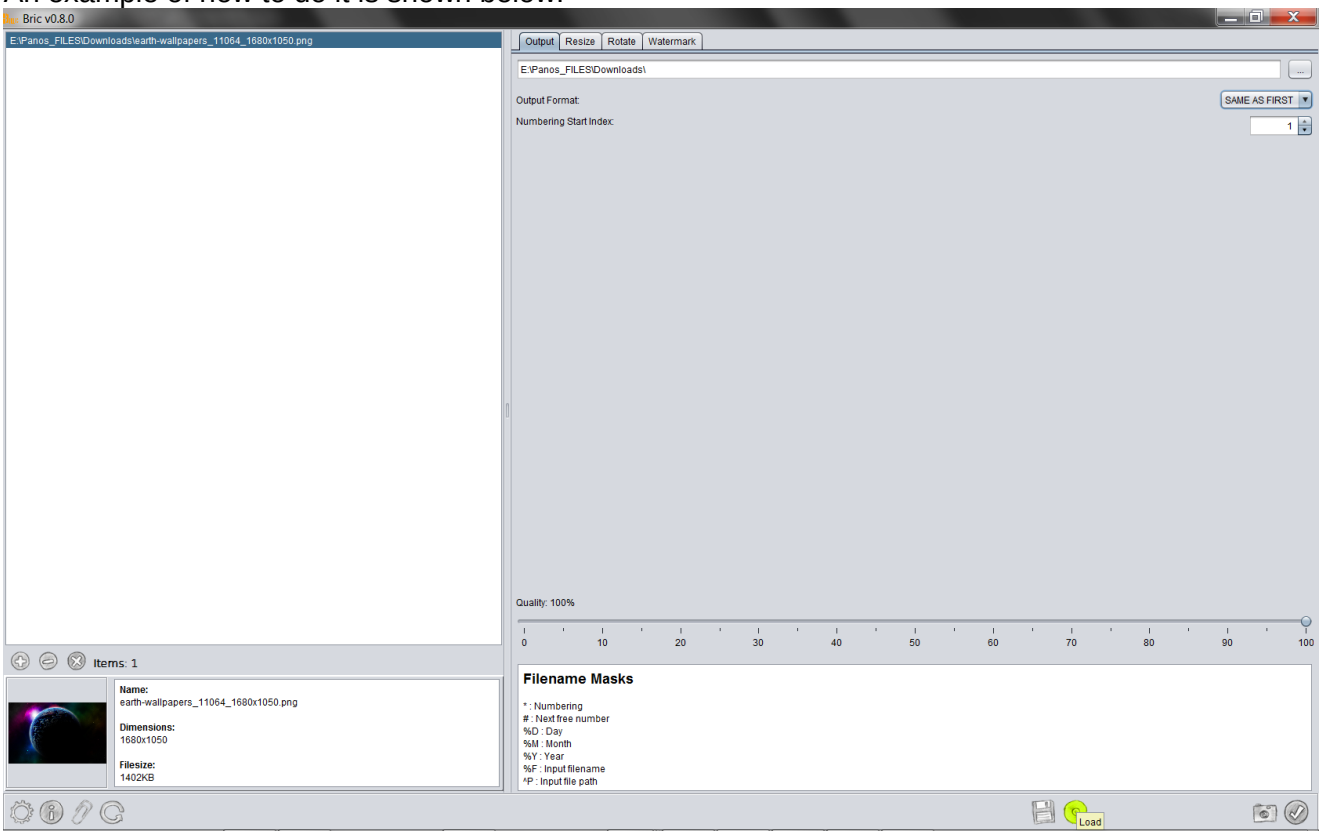

Figure 3.6.4 Load preferences & selections – “Load” option

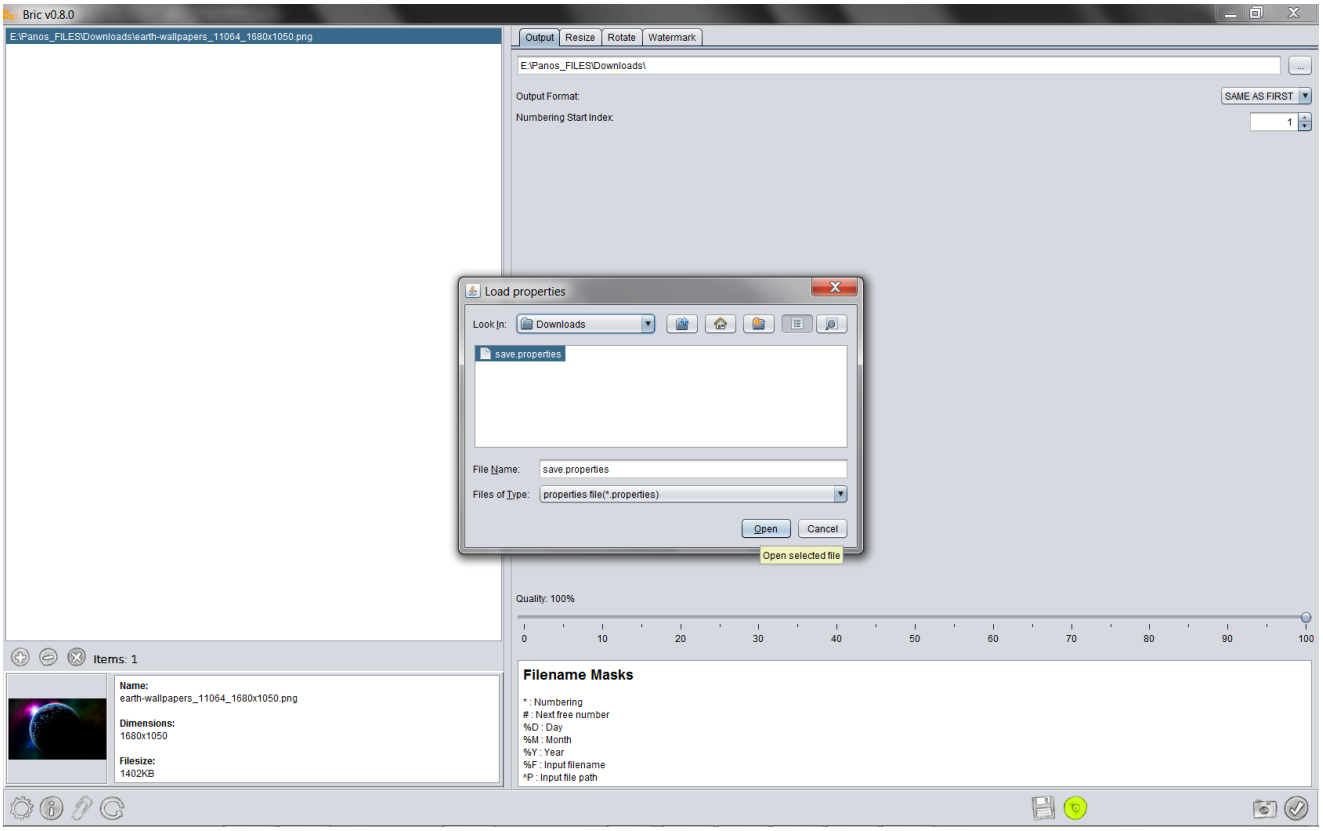

Figure 3.6.5 Load preferences & selections – choose the preferred folder, select saved-properties-file and click on “Open” button

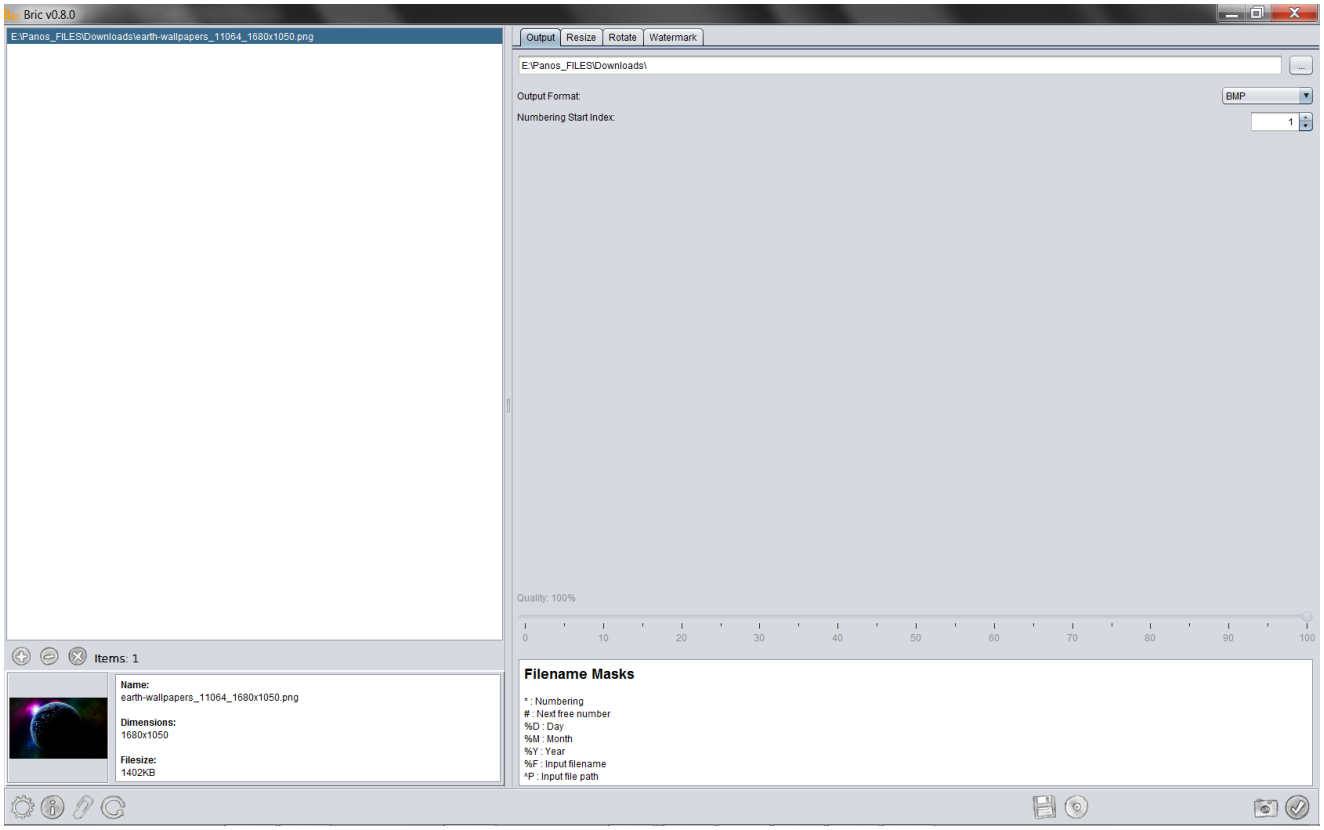

Figure 3.6.6 Load preferences & selections – properties are loaded

4. External Interface Requirements

4.1 User Interfaces

The developers of BRIC used Java to build the application and the user interface of the software is Java Swing. Also, the GUI is as simple as possible, so it does not become a hindrance to the overall casual user experience. So, program's operations take place with a few clicks and exit is done like any other window by clicking on "X" on the upper right corner (as seen below).
The first screen of the program that the user sees is the following:

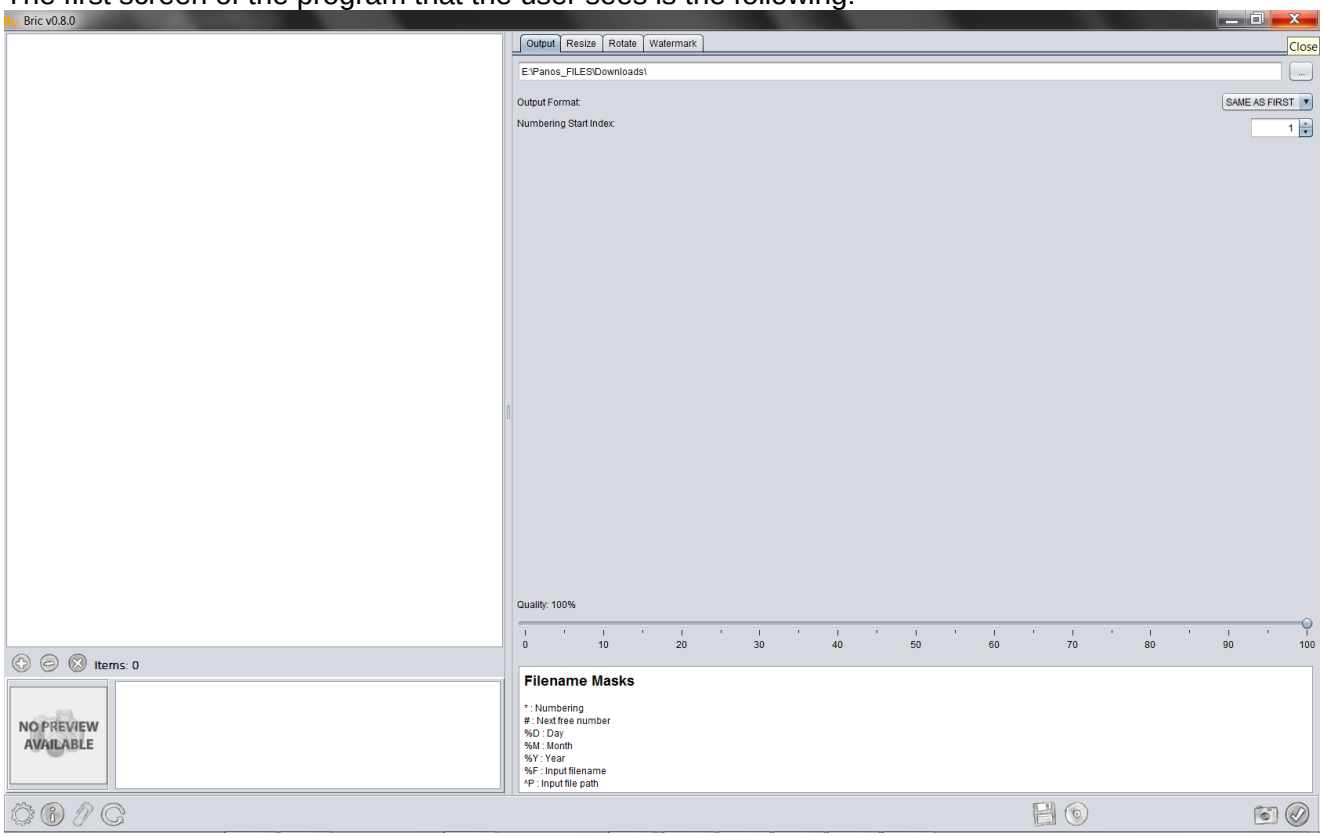

Figure 4.1.1 User Interface – Main Window & "Close"/Exit Program button

When program's operations take place like input-image addition or image resizing, a progress bar appears as seen below in Figure 4.1.2 .

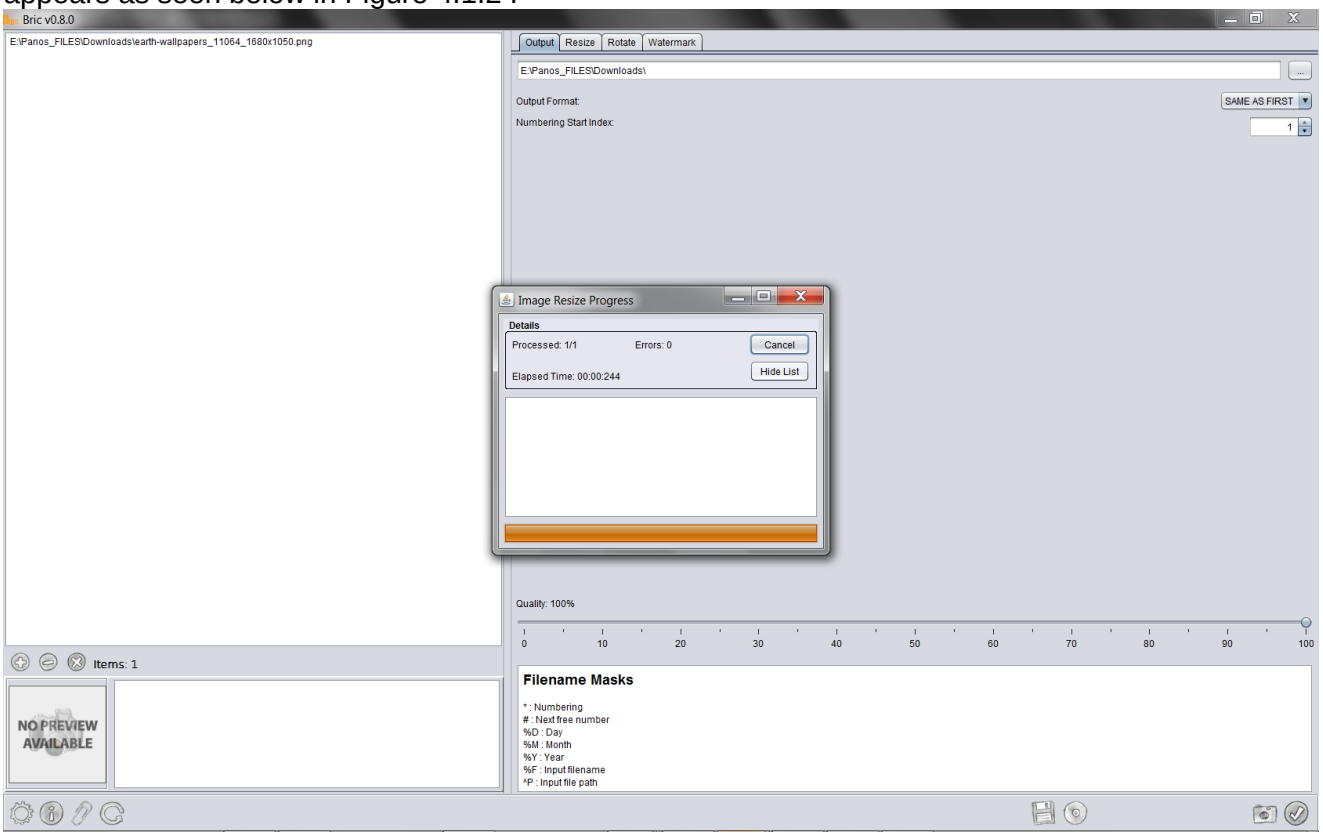

Figure 4.1.2 User Interface – Progress Bar

4.2 Hardware Interfaces

In order for BRIC to run, a PC, Mac or laptop (regardless of its operating system) is needed. Since it is not a resource "hungry" program, it will run on most systems without a problem. Also, a functional standard mouse is required and a functional standard keyboard is recommended, too.

Supported device types (used for input / output files)

- ✓ Hard disks
- ✓ USB Flash drives
- ✓ External hard drives (USB – eSATA enclosures etc)

4.3 Software Interfaces

As it has already been stated, BRIC runs on every system where Java Runtime Environment is installed and uses image processing libraries Apache Sanselan, ImageIO and ImageJ.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

As it has already been mentioned, BRIC is not a resource hog and will run on every computer. Its functions and features are not computationally intensive. It does not require a powerful processor or graphics card, much RAM or much disk space and the program loads fairly quickly.

5.2 Safety Requirements

The application is programmed in such a way that it should not cause any problems on any machine where it is run while it is executed within the guidelines of the requirements stated in this document. The only notable case is when the name and filetype of the output image is the same as another file's in the output directory; in this case "Add" option should be preferred if the user does not want to overwrite that image (as seen below in Figure 5.2.1).
However, if the use of the program creates any problems, it would be wise to contact the developers in order for them to address and solve these issues.

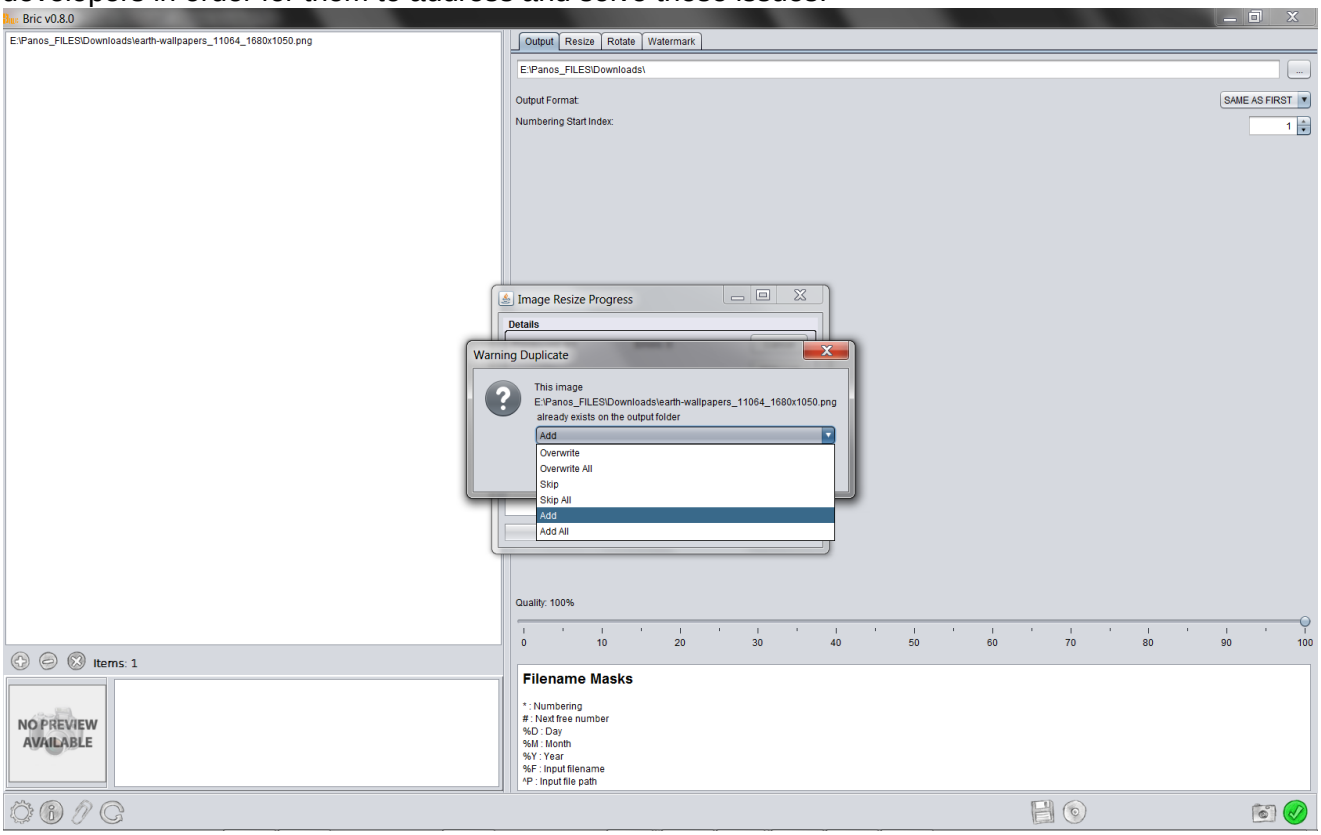

Figure 5.2.1 Warning Duplicate – "Add" option

5.3 License

BRIC is free, open source software, so it is licensed under the widely-known and famous GNU General Public License version 3.0 (GPLv3). This specific type of license gives everyone the freedom of owning a copy of the source code free of charge. Also, it gives the permission to share, modify or redistribute the source code which accompanies the software.
Furthermore, because the software is licensed free of charge, there is no warranty for it.

More information can be found in the following link:
http://www.gnu.org/licenses/gpl.html