

Contents

1. Overview:	2
2. Why You Need It:	3
3. Software Copyright Notice:	4
4. What Is Included:	7
5. Installation Procedure:	8
6. System Requirements:	10
7. GUI Description:	11
8. Program Logic:	13
9. Design Text File Validation:	19
10. PSD Template Structure:	20
11. Output File Structure:	23
12. Custom Specs Tab:	25
13. Sequential Editing Of All The Output PSD Files:	26
14. Swap Two Layers:	28
15. Convert The PSD Files To JPG And Subsequently To PDF Format:	30
16. Designing A New Template:	31
17. The Trial Program:	33
18. List Of All The Files Generated By All The Scripts:	34



1. Overview:

One Click Album Builder is the most user-friendly, intelligent, intuitive, customizable and realistic approach to design complex image albums within a very short time. This is an Adobe Photoshop script and can only be executed from Photoshop application. The designer can use his own design templates and this program will intelligently place the album images within those templates and create the final product within a very quick time. Everything can be done just by one click, and hence the name – *One Click Album Builder*.

2. Why You Need It:

Many album design softwares are available in the industry. Some softwares are very easy to use and design the full album within a very short period of time, but the designer cannot integrate his own design templates with those softwares. So, after those softwares complete their execution and generate their final designed product (PSD file for example), the designer has to again refine those PSD files. This is a very time-consuming process. Otherwise, the designer has to design the whole album using his own PSD templates. He has to decide and find the appropriate template for the corresponding images orientation and manually paste the images within the template layers. This is also equally time-consuming process! On average each spread takes 15 mins to complete, so a wedding album of 50 spreads will take approximately 12 hours to design properly. In reality it takes much more time. So the whole album design process is very monotonous, tiresome, tedious, repetitive and unexciting process. I am sure that all the album designers will agree on these points.

So how can we integrate our own design PSD templates with the album images and that's even in a very quick time? How can we give our own albums a very unique and nice design without much effort at all? That's the motivation behind creating this album designing software.

The script can support any type of album size, color profile, resolution and any number of pages. The only factor it depends on whether the corresponding PSD templates are present or not.

3. Software Copyright Notice:

Software: Copyright © 2023 Nirmalya Sinha. All Rights Reserved.

www.one-click-album-builder.com

Last Updated: July, 2023

This Software is licensed for use under an End User Software License Agreement.

END-USER SOFTWARE LICENSE AGREEMENT -

THIS IS A LICENSE AGREEMENT THAT YOU ARE REQUIRED TO ACCEPT BEFORE INSTALLING OR DOWNLOADING THIS SOFTWARE. BY DOWNLOADING, INSTALLING, OR OTHERWISE ACCESING OR USING THE SOFTWARE, YOU ACKNOWLEDGE THAT YOU HAVE REVIEWED AND ACCEPTED THIS LICENSE AGREEMENT, AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE AGREEMENT. IF YOU ARE USING THIS SOFTWARE FOR YOUR EMPLOYER, YOU ALSO REPRESENT AND WARRANT THAT YOU ARE DULY AUTHORIZED TO ACCEPT THE TERMS AND CONDITIONS OF THIS LICENSE AGREEMENT ON BEHALF OF YOUR EMPLOYER.

This End User License Agreement (“Agreement”) is made and entered into by and between Nirmalya Sinha (hereinafter “**Proprietor**”) and you as the end-user of the Software (hereinafter, “**Licensee**” or “you”). All computer programs, software, and associated media, printed materials and online and electronic documentation, including any updates or upgrades thereof are collectively referred to as “**Software**”. This Software is provided to Licensee is for use solely by the Licensee. By using the Software, the Licensee indicates his acceptance of these terms and conditions, and this Agreement becomes a legally binding agreement between the Licensee and the Proprietor.

1. License Grant. The Proprietor grants to Licensee, under the following terms and conditions, a personal, non-exclusive, revocable, nontransferable, non-sublicensable license for its internal and end-use purposes only. The software should be used solely at the Licensee’s facility, on a single (or multi) computer system accessible only by Licensee or its employees (and not on any public network or server), where the single (or multi) computer is owned, leased, or otherwise substantially controlled by the Licensee alone.

2. License Restrictions. The Software is licensed to the Licensee, not sold. The Licensee may not use, copy, reproduce, distribute, display, publish, sell, re-sell, lease, or sub-license the Software, in whole or in part under no circumstances. Licensee may not modify, improve, translate, reverse engineer, decompile, disassemble, or create derivative works of the Software or otherwise attempt to (a) defeat, avoid, by-pass, remove, deactivate, or otherwise circumvent any software protection mechanisms in the Software including, without limitation, any such mechanism used to restrict or control the functionality of the Software, or (b) derive the source code or the underlying ideas, algorithms, structure, or organization form of the Software.

Licensee will not allow, at any time, the Software or any portions or copies thereof in any form to become available to any third parties.

3. Ownership. The Software is protected by Nirmalya Sinha intellectual property laws. All right, title and interest in and to the Software (including associated intellectual property rights) are and will remain vested in Nirmalya Sinha. Licensee acknowledges that no rights, license or interest is granted to him hereunder. Licensee acknowledges that unauthorized reproduction or distribution of this Software, or any portion of it, may result in severe civil and criminal penalties.

4. Upgrades/Updates. Nirmalya Sinha may, at its sole discretion, provide updates or upgrades to the Software. In that case, Licensee shall have the same rights and obligations under such updates or upgrades as it has for the versions of the Software initially provided to Licensee hereunder. Licensee recognizes that Nirmalya Sinha is not obligated to provide any upgrades or updates to, or support for, the Software.

5. Data Integrity/Loss. The Software is not destructive and doesn't cause any data loss. Hence the Licensee agrees that Nirmalya Sinha is not responsible for any inability to access, loss or corruption of data as a result of using this Software and Nirmalya Sinha has no liability in connection with such inability to access, loss or corruption of data.

6. Term of License. This Agreement shall be in effect from the time Licensee expressly accepts the terms and conditions of this license and shall remain in effect until terminated. This license will otherwise terminate if Licensee fails to comply with any term or condition of this Agreement including failure to pay any applicable license fee. Licensee agrees upon termination of this Agreement for any reason to immediately discontinue use of and un-install the Software and destroy all copies of the Software as well as any associated documentation in its possession.

7. Limitation. Since the Software is an Adobe Photoshop script, it has to be executed only within the Adobe Photoshop host application. If Adobe Photoshop application has any defect, then the Software will not function as intended. The Licensee will not hold the Proprietor responsible in this circumstance. Adobe fixes their own defects in their next release and the Software will work again. So, in this situation the Licensee has to also upgrade his Photoshop version. The Software was tested in Windows OS with Photoshop version CS6, CC 2022 and CC 2023. The Software was also tested in Mac Catalina 10.15 with Photoshop version CC 2019. Moreover, a sample test software is provided with very limited functionality. Nirmalya Sinha guarantees that if this test software operates properly in Licensee's computer system within his Photoshop application, then the actual Software will also work as expected.

8. General. Licensee may not sublicense, assign, share, rent or transfer any of its rights under this Agreement. Nirmalya Sinha reserves the right to change this Agreement at any time. When Nirmalya Sinha makes any changes, Nirmalya Sinha will provide the updated Agreement, or a link to it, on www.one-click-album-builder.com and such updated Agreement shall become effective immediately. Licensee's continued access to or use of the Software represents Licensee's agreement to any revised Agreement.



9. Contact. Any questions regarding legal rights, duties, obligations, or restrictions associated with the software should be directed to Nirmalya Sinha – info@one-click-album-builder.com

10. Third Party Components. This Software does not include any third-party software or third-party program.

4. What Is Included:

As part of the delivered software, the following Photoshop scripts are included –

1. OneClick_Album_Builder.jsxbin
2. OneClick_Album_Builder_Swap.jsxbin
3. OneClick_Album_Builder_CheckValidity.jsxbin
4. Export_PSD.jsx
5. Sequentially_OpenFiles.jsx
6. CloseEventListener.jsx

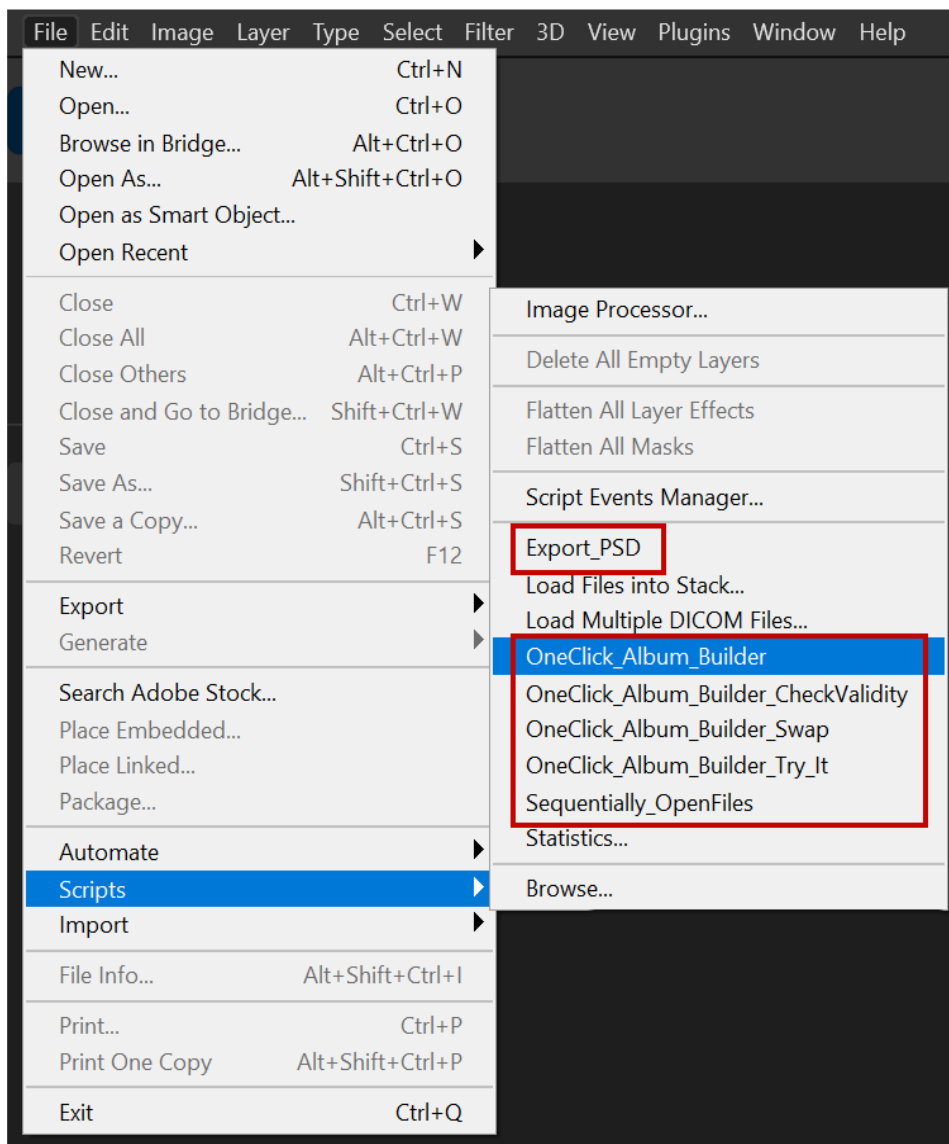
600 finely crafted PSD templates are also included.

This user guide is also included.

Also note that all the script files required for the album building are named with **OneClick_Album_Builder** word. Also these scripts are in binary format (jsxbin) to protect the copyright information. But **Sequentially_OpenFiles** and **Export_PSD** scripts can be used for any generic operation. Hence the names of these scripts are not prepended with **OneClick_Album_Builder** word and these files are in plain text format. These scripts are also provided as free programs.

5. Installation Procedure:

1. Close Photoshop if it is open.
2. Copy the first five files (except the last CloseEventListener.jsx) in the **Scripts** folder under the Photoshop installation folder.
Windows: C:\Program Files\Adobe\Adobe Photoshop [VERSION]\Presets\Scripts
MacOS: /Applications/Adobe Photoshop [VERSION]/Presets/Scripts/
3. Open Photoshop.
4. The script will now appear on the File > Scripts menu and can be launched from here every time Photoshop is launched.



5. Copy the CloseEventListener.jsx file in the **Event Scripts Only** folder under the Scripts folder.
Windows: C:\Program Files\Adobe\Adobe Photoshop [VERSION]\Presets\Scripts\Event Scripts Only
MacOS: /Applications/Adobe Photoshop [VERSION]/Presets/Scripts/Event Scripts Only

We will discuss more about CloseEventListener script in the section [***13. Sequential Editing Of All The Output PSD Files.***](#)

You can also allocate a keyboard shortcut to run the scripts.

1. Launch the keyboard shortcuts dialog by clicking Edit > Keyboard Shortcuts.
2. Under 'Application Menu Command' open the entry for 'File'.
3. Scroll down, past 'Scripts' until you see the name of your script.
4. Click the blank area for the script's shortcut.
5. Type the required shortcut. If there is a conflict between the chosen combination and an existing shortcut, a message will appear at the bottom of the dialog. When done, click OK.

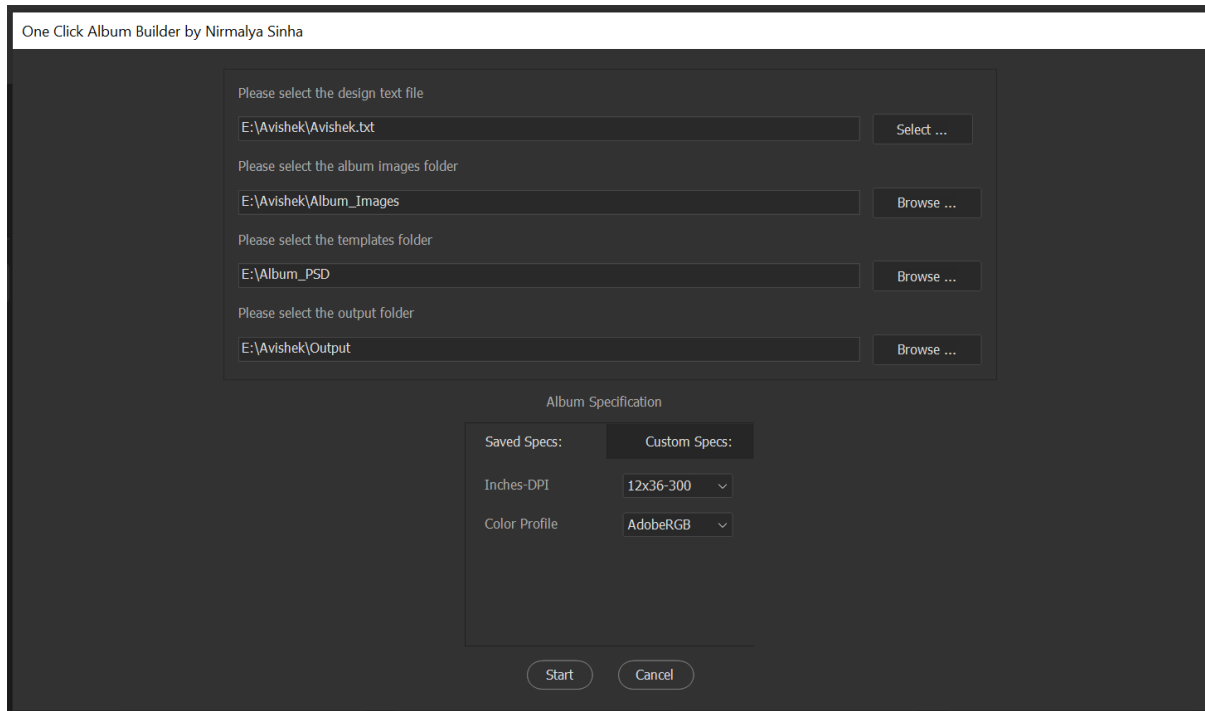
6. System Requirements:

If you are running Photoshop, then nothing else is required. This program is just a Photoshop script and should be kept under Photoshop Script directory. So no separate program installation is required. This program is tested in Photoshop CS3, CS6, Photoshop CC 2022 and Photoshop CC 2023 in Windows 10 OS. Similarly, this program is tested in Mac Catalina 10.15 with Photoshop CC 2019. So this program should work in all the operating systems with all the versions of Photoshop from CS3.

One additional requirement is that there should be enough space in the hard drive where the Photoshop scratch disk is assigned. Since this program will open many PSD files to search for the correct orientation and also open image files to paste into those PSD files, Photoshop may deplete space in the hard drive. So you may assign scratch disk in a hard drive where a lot of empty space is available.

7. GUI Description:

When the script is executed from Photoshop, the following GUI appears on the screen



The user has to select the design text file, album images folder, template folder and output folder.

- **Design text file:** This is a simple .txt or .csv (comma-separated value) file which contains the number of images to be accommodated in each side of the album spreads.

A simple example will be like the following –

Spread,Left,Right,PSD

1,,4,
2,2,5,
3,4,4,
4,4,3,
5,3,,

Here we are creating a 5 spread template. The first spread is a single right side and we are specifying 4 number of images to be placed on the right side. The second spread has both left and right sides, 2 images to be placed on the left side and 5 images to be placed on the right side. Similar explanations for third and fourth spreads. Fifth spread is the last spread and we are specifying that 3 images to be placed on the left side and right side is empty.

The fourth column is PSD and we will cover more about it in the next sections. We will also discuss about other valid formats.

- **Album images folder:** This is the folder where all the images selected for the album is kept together. The user has to keep all the **selected** images within this folder. Any image which should not be placed in the album, should not be kept in this folder.
- **Template folder:** This is the folder where all the Photoshop PSD templates are kept. These PSD templates are single side templates, not the templates for the whole spread. These templates have a specific structure so that the pictures can be automatically placed within the image layers. This will be covered in details in the next sections.
- **Output folder:** This is the folder where the output album PSD files will be stored by the program.
- The user can also specify the size, resolution and color profile of the final PSD files.

After selecting the above mentioned fields, the user has to click the **Start** button. The program will automatically find the templates corresponding to the number and orientation (landscape or portrait) of images specified in both sides of the spreads. Then the program will create the final output PSD file, copy the selected templates on both sides, paste the images within the image layers and build the final output PSD file. That's the intelligence of the program.

After successfully completion, the templates and images for each spread will be moved under **Done** folders. If any error occurs for any spread, then the templates and images will be moved under **Skipped** folder.

The user can also define his own album specifications in the **Custom Specs** tab. This is covered in section *13. Custom Specs tab*.

The program remembers the paths of the design file and respective folders. So when the program is launched again, the file and folder paths will be pre-populated for user's convenience.

8. Program Logic:

At first let's discuss about the design text file and the template naming convention.

Design text file: Within this design text file the user specifies the number of images to be placed in each page of the album. The program will select those many images from the **Album images folder** and paste them in the final PSD output file. Hence the user should not keep any unnecessary image within this **Album images folder**. Only the images which should be placed in the album should be kept in this folder.

The design file can be specified either in .txt or in .csv format. A comma-separated values (CSV) file is a delimited text file that uses a comma to separate values. The user can generate the csv file in Microsoft Excel which will automatically populate the spread numbers. Then the .csv file can be opened in any text editor like notepad. The file can be also renamed to .txt extension.

For easier operation the design file should have the same name as the client name. For example if the client name is Avishek, then the user can save the file as Avishek.txt or Avishek.csv.

In that case, the output file names will be Avishek_1.psd, Avishek_2.psd and so on for each and every spread number.

Template naming convention: The program is depending on selecting the user's PSD templates and for quicker and efficient selection, the templates have to be named properly.

If in the template all the image layers have landscape (L) orientation, then the template should be named with only L in correspondence to the number of image layers. For example, if there is only one L image layer, then the template should be named as L_1.psd, L_2.psd, L_3.psd and so on. If there are two image layers, then the template should be named as LL_1.psd, LL_2.psd, LL_3.psd and so on. So the template name should correspond to the number of image layers. Similarly for five image layers the names should be LLLLL_1.psd, LLLLL_2.psd, LLLLL_3.psd and so on. I will explain the reason behind all these naming convention in a while.

Similarly, if all the images layers within the template have portrait (P) orientation, then the template names should be P_1.psd, PP_1.psd, PPP_3.psd and so on.

If the image layers within the template have mixed orientation, some L and some P, then the name should be LP_1.psd, LPL_1.psd, LPLP_1.psd, LPLPL_1.psd and LPLPLP_1.psd. The logic is that irrespective of the actual orientation of the image layers (like how many L and how many P), if there are three image layers, then name should be LPL_1.psd. If all the image layers don't have the same orientations, then name the templates with alternate L and P. Since there are three image layers, there will be three characters in the name and hence LPL_1.psd. Successive templates for same three image layers will be named as LPL_2.psd, LPL_3.psd and so on. Same logic applies for any number of images layers with any combination of L and P. For example, for five image layers, for one L and four P, or four L and one P or whatever, name the templates as LPLPL_1.psd, LPLPL_2.psd and so on.

The logic behind all these naming conventions is as follows.

As stated earlier, the program is depending on picking up the right template as quickly as possible. The more quickly it will be able to pick up the right template, the more efficient it will be. Photoshop will also run smoothly if the program is able to process each record quicker. If Photoshop opens too many PSD templates while searching for the correct one, the space in the TEMP folder will drastically reduce and Photoshop may throw “Scratch disk full” error. So we need to segregate the templates into different groups so that the program searches the templates within only that group, instead of all the files present in the folder. At first the program will evaluate the orientation of the set of images for each side of the spread. If all the images have same orientation and let’s say there are four images, then the program knows that it has to select only the LLLL set of images and nothing else! If there are four images with L and P orientations, then the program knows that it has to search only the LPLP set of images! This way we are grouping the templates so that the search operates very efficiently.

Moreover whenever the program will open any LPLP_x.psd file, it will create a map within a file in the template folder with the actual orientation of the image layers of the template. For example, suppose in LPLP_5.psd and LPLP_8.psd files, the image layers have the following orientation – LLPP. So the map will have the following entry –

LLPP: LPLP_5.psd,LPLP_8.psd

So next time when the images of any side of the spread has the same LLPP orientation, the program will find that they key value LLPP is already existing within the map and it will pickup LPLP_5.psd for that set of images.

When the program picks up LLLL_x.psd, then the search is very efficient; it has to just pickup the next file from the filesystem. But when the program picks up LPLP_x.psd, it has to verify whether the image layers have the same alignment with the album images. So the program has to open and close many templates one after another. This will fill up the OS TEMP folder. To reduce this overhead, the program is maintaining the map so that it can pick up the correct template in single operation. Next time when the program will scan the templates for a particular image orientation, at first it will check whether the exact orientation already exists within the map. If yes, it will pickup the template from the map itself, instead of file scanning. In this way the program will minimize the amount of scanning required. This logic makes the program very efficient and powerful.

We are naming all the three image layer templates with all the variations of L and P as LPL because otherwise it will be very cumbersome to complicate to name the templates. Following combinations are possible with L and P for three layers - LLP, LPL, PLL, PPL, PLP or LPP. So the best approach would be to name these templates according to their actual orientations, instead of the generic LPL. But for four, five and six images the combinations are huge in numbers and it’s totally impractical to name the templates with their actual orientations. Hence we are giving them a generic name and the program is intelligent to pick up the right template within that group.

So if the images mentioned in one side of the spread has LLP orientation (i.e. three images), the program will scan all the templates within the LPL group until it picks up a template whose image layers have the exact LLP orientation.

Also note that when templates are encoded for the whole spread, ie. left page and right page together, the templates can be named without any naming convention. Since these full spread templates contain image layers for both the pages, the above mentioned naming convention doesn't apply.

Not only the PSD template names, but also the template layers and layer groups should follow a specific structure so that the successive images can be automatically pasted in the corresponding image layers. This is covered in section - [10. PSD template structure](#).

Processing logic:

When the user specifies a certain number of images in the design text file for both sides of the spread, the program will process each record of the design text file according to the format of that particular spread.

So let's first discuss what are all the formats by which the user can customize his design.

This is one of the valid formats of a particular spread of the Avishek.txt file as already mentioned earlier –

Spread,Left,Right,PSD

1,,4,
2,2,5,
3,4,4,
4,4,3,
5,3,,

In the above case, the program will proceed as follows –

1. Check the orientation of the **4** images specified for the right side.
2. If they are all LLLL, pickup the first psd file, i.e. LLLL_1.psd.
 - a. If it has the valid format (will be explained later), then create the output Avishek_1.psd file; copy LLLL_1.psd file within Avishek_1.psd and then copy the images within the image layers. At last the program will move the LLLL_1.psd within the Done subfolder of the template folder. Similarly it will move the 4 images within the Done subfolder of album images folder.
 - b. If it has invalid format, then it will move LLLL_1.psd to the Skipped/1 subfolder within the templates folder. Here 1 corresponds to the spread number.
 - c. If all the files are scanned and all are invalid, then no more template file is found. The template file may not be found even if there are insufficient number of templates corresponding to the image orientation. In that case the images will be moved to the Skipped/1 subfolder within the images folder.
3. Similarly if they are all PPPP, pickup the first psd file, i.e. PPPP_1.psd and proceed as per the previous logic.

4. If the images have mixed orientation, i.e. one L, three P or two L, two P or three L, one P, then pickup the first LPLP file, ie. LPLP_1.psd.
 - a. If it has the valid format, then check the orientation of the image layers within the template. If the orientation of the image layers are same (or approximately same) as the album images, then create the output Avishek_1.psd file; copy LPLP_1.psd file within Avishek_1.psd and then copy the images within the image layers. At last the program will move the LPLP_1.psd within the Done subfolder of the template folder. Similarly it will move the 4 images within the Done subfolder of album images folder. The program will also save the actual orientation of the templates within the templates folder as a key-value pair (also called a map).
 - b. If it has invalid format, then it will move LPLP_1.psd to the Skipped/1 subfolder within the templates folder.
 - c. If the orientation is not matching, then it will search the next file within the template group i.e. LPLP_2.psd and the same logic will repeat.
 - d. If all the files are scanned and no match is found then the images will be moved to the Skipped/1 subfolder within the images folder.
5. Similarly if they are all PPPP, pickup the first psd file, i.e. PPPP_1.psd and proceed as per the previous logic.
6. After completing the first spread, the program will process the second spread. If it's full spread (i.e. both left and right side), the same logic for individual sides will be applied for both the sides. So same processing logic applies for each side of the spread - for **2** and **5** number of images. For full spreads, the additional task of the program is to align the second template to the right side of the output PSD file. Suppose for the left side, the program picks up LL_1.psd and for the right side it picks up LPLPL_1.psd. So Avishek_2.psd will comprise of LL_1.psd for the left side and LPLPL_1.psd for the right side. Photoshop can align the layers and layer groups to the left/right side of the canvas when they meet some specific conditions. So we will discuss about the valid template format in the next sections.

These are other possibilities for easy operation and customization –

Spread,Left,Right,PSD
2,2,5,X.psd
4,4,3,Y.psd

Here we are encoding the specific psd file to be picked up for designing the **whole spread**. For example we are specifying that for the second spread, X.psd file should be selected for the whole left side and right side. So the program will create the output Avishek_2.psd file, copy the X.psd file within it and paste the **2+5=7** number of images within it. Thus the designer is integrating his own design with the album images and with such ease.

Similarly for the fourth spread the program will pickup the Y.psd file for total $4+3=7$ number of images.

The user can also specify the individual PSD templates **for each left and right side of the spreads**. So this is the third format for the design text file –

Spread,Left,Right,PSD

1,,LLLL_10.psd,

2,LP_5.psd,5,

4,4,PPP_1.psd,

Here the user is specifying that in the first spread for the right side, the program should pick up **LLLL_10.psd** file, for the left side of the second spread **LP_5.psd** file and for the right side of the fourth spread **PPP_1.psd** file.

Please note that when the template name is encoded in the design text file (for either the whole spread or individual sides), the program will not apply any logic to search for the correct templates, but will blindly trust the designer and fit the images within the specified templates. For example, suppose the image has landscape orientation, but the corresponding image layer within the template has portrait orientation. In normal case the program will not select this template for these set of images. But if the template is encoded with this mismatch, then the program will paste the portrait image within the landscape image layer.

So the designer can specify the design text file with all these options and the final format can be a mix of all the options –

Spread,Left,Right,PSD

1,,5,

2,2,5,X.psd

3,4,LPLP_10.psd,

4,4,3,

5,PPP_1.psd,,

Program logic for the right side

X.psd for the full spread

Program logic for the left side, LPLP_10.psd for the right side

Program logic for the left side, Program logic for the right side

PPP_1.psd for the left side

Please note that when a template file is encoded for left or right page, that template file will not be picked up for any preceding page even if the orientation of the images matches in that particular page. For example, we are specifying **LPLP_10.psd** should be used for right side of the third page. But in the very first page, we have five images. Even if the orientation of these five images matches with the orientation of the **LPLP_10.psd**, this template will not be selected for the first page. Similarly, **PPP_1.psd** will not be selected for the right side of the fourth spread.

The designer should spend only this much time and effort for designing the whole album!

Order of precedence: The program picks up the templates in the following order –

1. Checks whether the full spread template is encoded

2. Checks whether one side template is encoded (for any spread, the user cannot specify the full spread template and one side template at the same time)
3. If all the images have same L or P orientation, then picks up LL or PP templates
4. If the images have mixed orientations, then checks whether the orientation is existing in the map
5. At last searches for the correct template with matching alignment by file scan (creates the map as well)

9. Design Text File Validation:

Some possibilities of **incorrect** values of the design text file –

Spread,Left,Right,PSD

- 1,0,4, - 0 is not allowed. If the user don't want to put any picture in any side, he should put it blank instead of 0
- 2,2,LPLPL, - Single side, the value should be either numeric or LPLPL.psd, not just LPLPL
- 3,4,4,Y - Full spread, the value should be Y.psd, not just Y
- 4,7,3, - The number of images should be less than or equal to **six**
- 5,9,3,Z.psd - When the full spread template is encoded, the number of images for each page should be less than or equal to **eight**
- 6,3, - Please note that one comma (corresponds to the right most column) is missing
- 7,3,4 - Same reason, comma is missing
- 7,4,4, - Nothing is wrong here, but just observe that the spread number is the same as the previous spread, so the previous spread output PSD file will be overwritten

The user can also skip the very first line, i.e. **Spread,Left,Right,PSD** and mention the subsequent lines from the very first line of the design text file. So this is also a valid format –

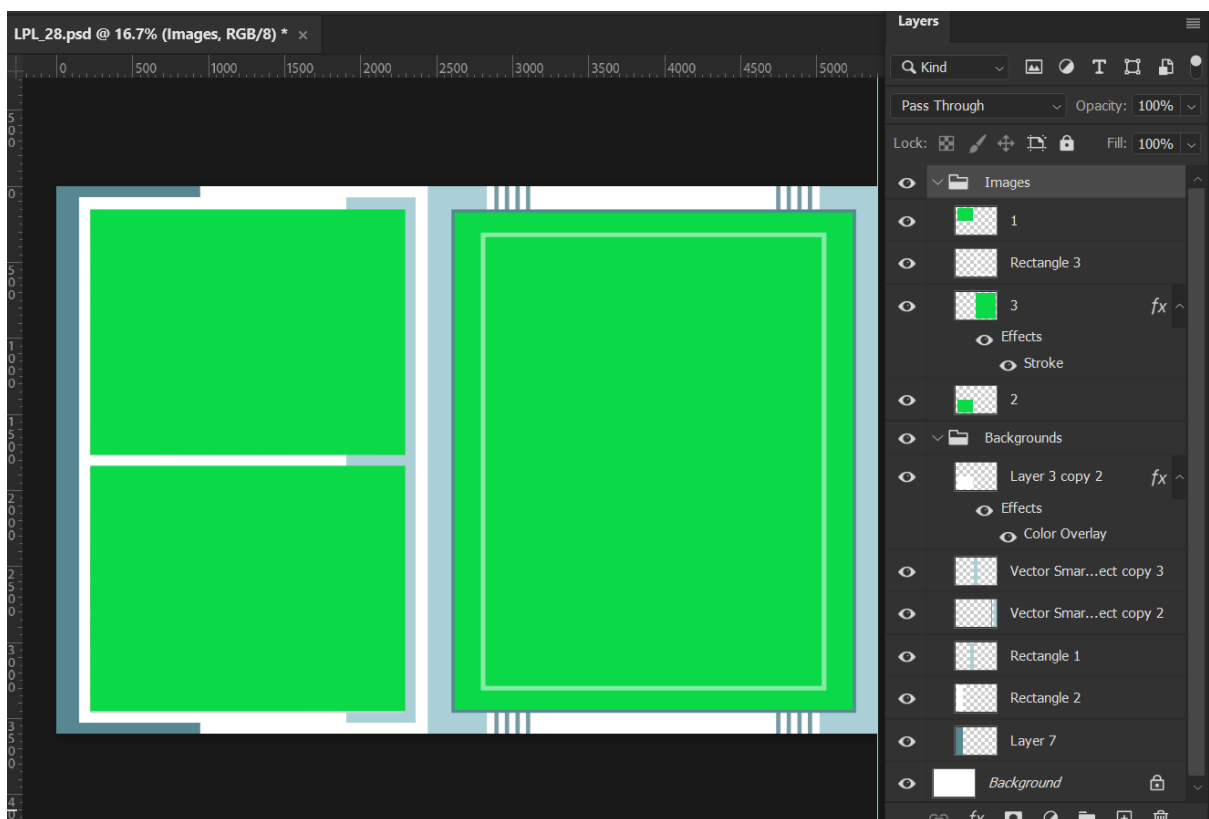
- 1,,4,
- 2,2,5,
- 3,4,4,
- 4,4,3,
- 5,3,,

10. PSD Template Structure:

There are two main logics upon which this program is based.

1. The templates should have the specific naming convention. Only then the program will be able to select it.
2. The layers and layer groups of the templates should have the proper structure and naming pattern. Only then the program will be able to paste the images in the consecutive layers.

This is the correct layer structure of any PSD template -



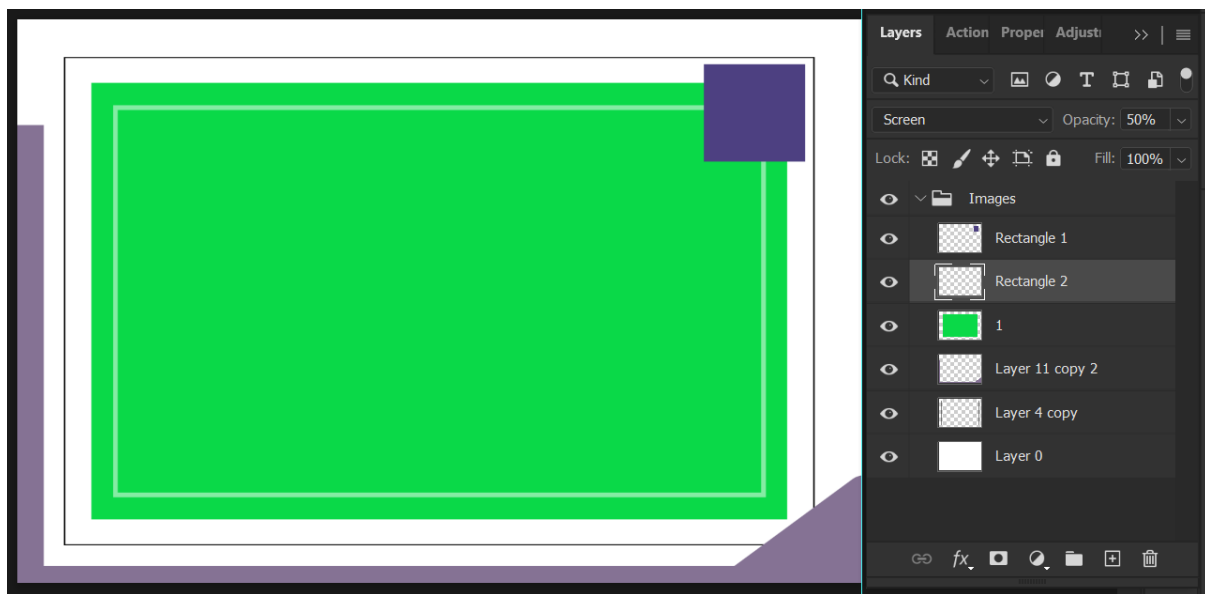
As you can notice that all the layers are contained in two different groups – **Images** and **Backgrounds**.

The image layers are the layers where the images will be placed by the program. They are kept within the **Images** layer group. So when this PSD template is picked up by the program, the first image will be placed within the layer named 1, the second image will be placed within the layer named 2 and so on.

And as already stated earlier, there should be only three image layers within the LLL, PPP and LPL set of templates. You can notice in the above picture that the PSD file's name is LPL_28.psd and there are three image layers within the Images layer set (colored as green).

Any number of additional layers can be present within the Images layer group, but their names should not follow the 1,2,3... naming pattern. They can have any arbitrary name apart from the numeric names.

The template designer can put all the other layers within the **Backgrounds** layer group. This will keep the templates neat and tidy. If there are no other layers except the **Images** layer group, then the user can skip the **Backgrounds** layer group altogether. For example the following is also a valid template structure –



You can notice that there are no **Backgrounds** layer group, all the layers are within the **Images** layer group. Since *Rectangle 1* and *Rectangle 2* layers are top of Image layer *1*, *Rectangle 1* and *Rectangle 2* layers are also kept within the **Images** layer. Since only a few background layers are present, there is no separate **Backgrounds** layer group and all layers are kept inside the same **Images** layer group.

The image layers within the Images layer group need not to be arranged sequentially in the layer panel. That is not required, the program is picking up the image layers by their names and not by their arrangements. But these layers should be named as the human eye moves across them. So top left layer is 1, bottom left is 2, top middle is 3 and so on.

When a template file is selected for the right set of images, it's moved to the right side of the output psd file. Photoshop restricts that layer groups can not be transformed or realigned if there is no content within the layer group. So if the designer keeps only the Background layer without any layer underneath, then the program will throw error. For example this is an invalid structure -



You can notice that the **Backgrounds** layer group is empty – *this is invalid*.

In this case, the program will delete the whole **Backgrounds** layer group to make the PSD a valid template.

Moreover if there is no normal or text layer within the **Backgrounds** layer group, even then it cannot be transformed. For this purpose the program rasterizes the color fill, gradient and pattern fill layers and saves the PSD templates if there is no other normal layer within the **Backgrounds** layer group.

When these incorrect PSD files will be first encountered, the program will make the necessary changes and save the PSD files so that the same issue won't come up again.

Also note that if a layer is outside of the Images or Backgrounds layer, then this layer is not copied to the output PSD file. In the output PSD file, the Background layer is white, but if in the individual template file the Background layer has any different color, then move that layer within the **Backgrounds** layer so that it's copied to the output PSD file.

Moreover Photoshop mandates that if any layer with name as "**Background**" is present within a layer set, then the layer set can not be moved to any side. So the designer should not keep any layer with name as **Background** within the layer groups.

As you can notice in the flowchart that at the very beginning the program validates the structure of the selected template. If the program finds that the template is invalid, then it moves the template within the **Skipped/spreadnumber** subfolder under the templates folder.

The user can make the required corrections and use the template again in the future.

11. Output File Structure:

When the program completes the execution of one spread, it generates the output PSD file. Its layer structure will be as follows –



As you can notice that in the output PSD file, the left side Images layer group will be renamed as **Images_1** and right side Images layer group will be renamed as **Images_2**. Similarly left side Backgrounds layer will be renamed as **Backgrounds_1** and right side Backgrounds layer will be renamed as **Backgrounds_2**.

In the output PSD file, all the image layers will be renamed with their relative sizes compared to their original picture sizes. So if a picture is resized to 80% within the PSD file image layer, then the layer will be appended with **Size- 80%**. This information will help the user to resize the layer appropriately if requires.



The program completely fits the image within the image layers. If the image layers don't have the exact 2:3 ratio like the normal images, then the parts of the images will be hidden behind. The user can adjust the desired position of the images within the image layer very easily. The user can also zoom in, crop or any other final adjustment in the output PSD file before delivering the output to the client.

The program is designed so intelligently that the pasted image will have all the layer properties of the corresponding image layers. Border, opacity and all other properties of the image layers will be preserved within the pasted images.

The program will also create a report file in user's desktop folder and record all the operations for all the spreads for the particular album. The report file name will be **Avishek_OneClickAlbumBuilder_OutputLog.txt**.

This will be the sample content of the report file –

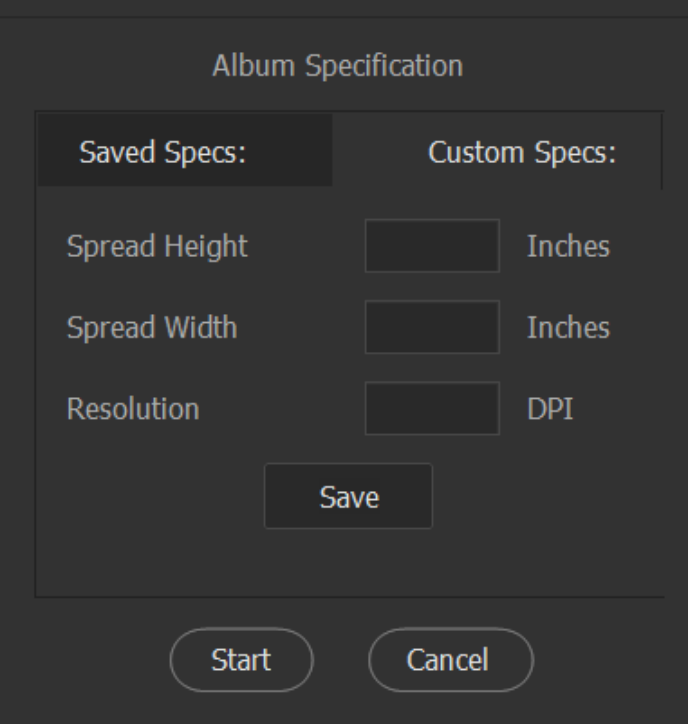
Done: Spread:42 Left:Encoded Templatefile:LPLPL_36.psd Moved to the done folder
Done: Spread:42 Right:Filescan Templatefile:LL_1.psd Moved to the done folder

Done: Spread:43 Left:Map Templatefile:LPLP_3.psd Moved to the done folder
Done: Spread:43 Right:Filescan Templatefile:LPLP_20.psd Moved to the done folder

Done: Spread:44 Left:Filescan Templatefile:LLLL_19.psd Moved to the done folder
Done: Spread:44 Right:Filescan Templatefile:PPP_10.psd Moved to the done folder

12. Custom Specs Tab:

The program is preconfigured with two album specifications, 12 inches spread height by 36 inches spread width and 12 inches spread height by 30 inches spread width. DPI for both the cases is 300. Since these are the most common album specifications, they are pre-configured in the program. But the user can specify his own album sizes and DPI in the Custom Specs tab.



The image shows a dialog box titled "Album Specification". It has two tabs: "Saved Specs:" and "Custom Specs:". The "Custom Specs:" tab is active. It contains three input fields: "Spread Height" (with "Inches" to its right), "Spread Width" (with "Inches" to its right), and "Resolution" (with "DPI" to its right). Each input field is currently empty. Below the input fields is a "Save" button. At the bottom of the dialog are two buttons: "Start" and "Cancel".

The user can input the spread height (inches), spread width (inches) and resolution. When the user clicks the **Save** button, the specification is permanently saved in the system for future operations.

The program permanently saves these specifications in a text file, named **OneClickAlbumBuilderConfig.txt** in the Photoshop's preference folder.

13. Sequential Editing Of All The Output PSD Files:

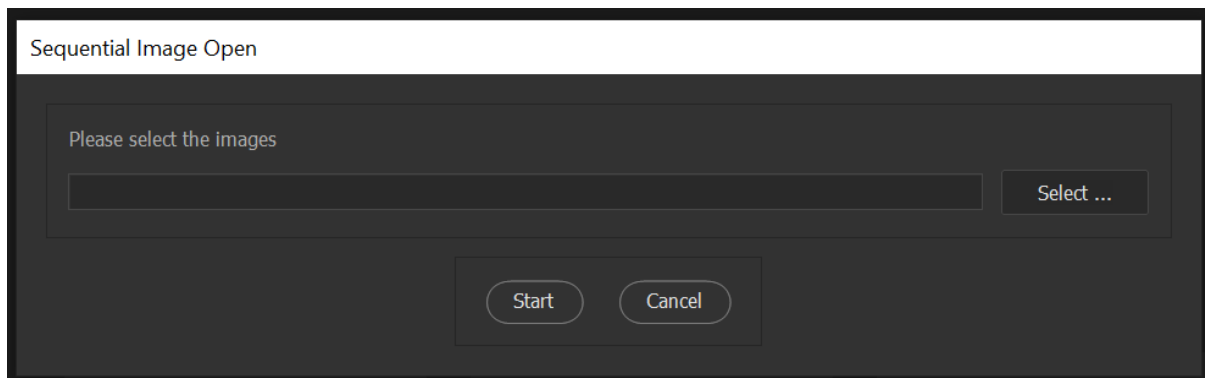
When all the output PSD files are generated, the user can fine tune the files according to his own choice. He may reposition the images if required and change the background layer colors.

But manually opening so many files one by one is a tedious task. When we open a file manually by double-clicking the file, the operating system's device driver software interacts with the file, checks the file type and then opens it within the corresponding host application. This takes a certain amount of time and I assume that everyone has experienced this delay.

But if photoshop opens a file from a script, then this overhead is avoided. First of all, the user don't have to sequentially double-click the files and moreover, since the files are directly opened by Photoshop the overhead of device driver software is eliminated. So, opening all the files one by one and capability to edit them is a very desirable feature. But it's a very tedious and tricky program. A lot of programmers have tried to design this - how to open a number of images, but one at a time? The user will edit the image and only when he saves and closes the image, the second image will be automatically opened. And the same thing will continue for the rest number of images. But as I told, it's a very tricky.

I have created one program which is actually a composite of two scripts – **Sequentially_OpenFiles** and **CloseEventListener**.

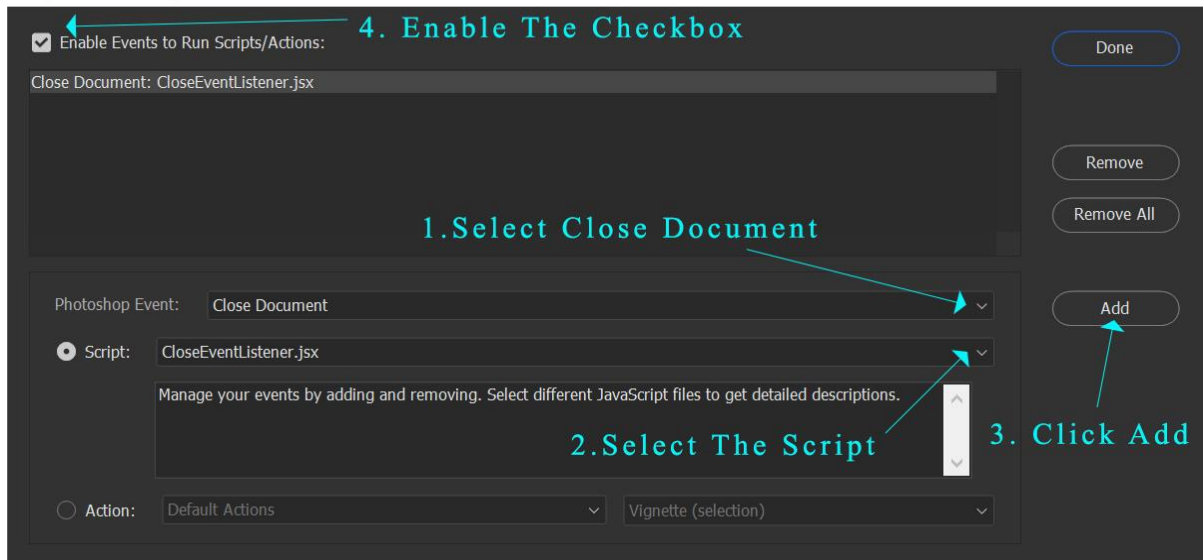
Sequentially_OpenFiles – when this script is launched, it asks for the input folder.



The user can select all the images that he wants to edit one at a time. When the user clicks the **Start** button, the script saves the list of files within **OpenClose_FileList.txt** text file in the Photoshop's preference folder. The script also opens the very first image form the list.

Successive images will be opened by the second script, i.e. **CloseEventListener**. For this, it has to be registered under Photoshop's **Script Events Manager**. The procedure is depicted in the following diagram –

Script Events Manager



The user has to select the **CloseEventListener.jsx** file in the **Scripts/ Event Scripts Only** folder in the second step – **2. Select The Script**. Then this script will be registered for the **Close Document** Photoshop Event. So whenever any image will be closed, this script will open the successive file stored within the **OpenClose_FileList.txt** text file. The user can perform all the editing operations. The user can also swap two images to exchange their positions (this is explained in more details in the next section). After the due editing, whenever the image will be closed, this script will delete the first line from the text file and open the next image. And this operation will continue until all the images are processed.

At any time if the user wants to break the operation, i.e. doesn't want to open the next file from the list, then he has to simply uncheck the Enable ... checkbox.

This **Sequentially_OpenFiles** script supports any type of image files, i.e. jpg, psd, png etc. So the user can use this script for any other batch editing purpose. This script also remembers the paths of the images folders between successive launches.

14. Swap Two Layers:

During the editing of the output PSD files, the user may want to swap two image layers. For example, the user may want to replace one small image with a bigger image. For this another script – **OneClick_Album_Builder_Swap** is provided herewith. The user needs to click two image layers in the layers panel and execute the script to swap the layers. This is clearly shown in the following picture.

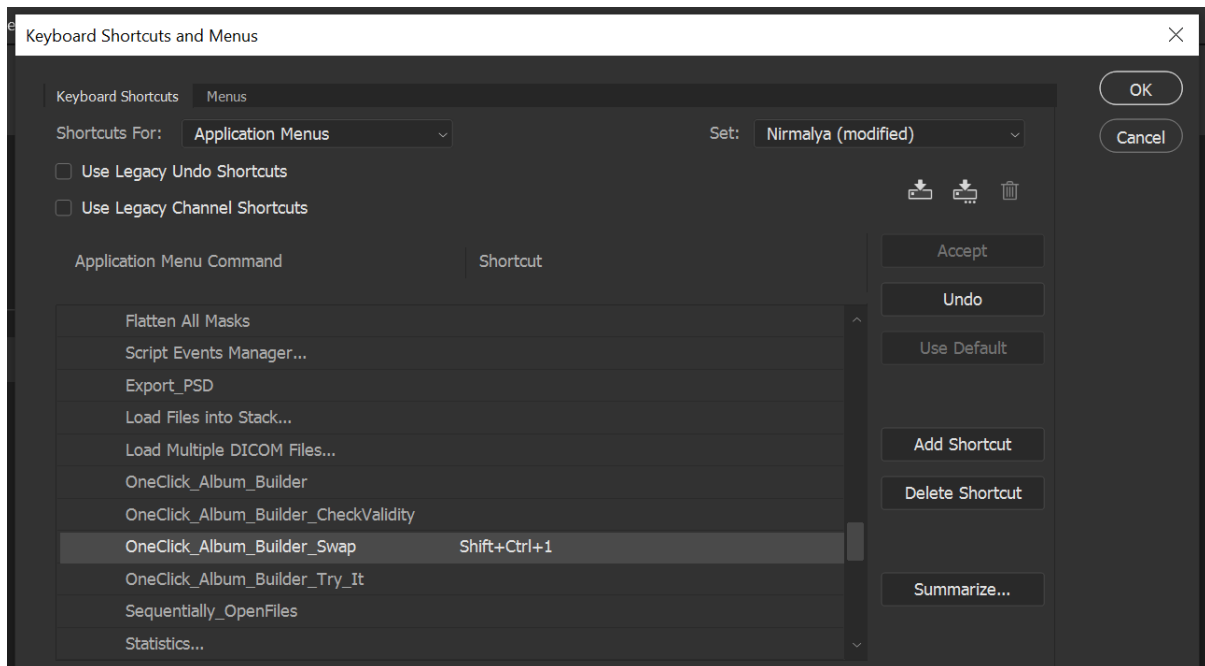


So the user has selected the layer 1 and 2 of the **Images_1** layer group and then executed the **OneClick_Album_Builder_Swap** script. So the layers will be swapped like this –



Please notice that the new layer names will also change according to their new sizes.

The user can also assign a Keyboard Shortcut to the frequently accessed script. This can be done by Edit > Keyboard Shortcuts menu and assigning an unallocated keyboard combination to the particular script. For example, we can assign **Shift+Ctrl+1** to the **OneClick_Album_Builder_Swap** script as shown in the following picture.



So if the user has to swap two layers, he has to just press **Shift+Ctrl+1**.

Please note that this script will work only for output PSD files generated using the **OneClick_Album_Builder** script. Also note that only the pasted image layers having names like **1:Size- 50%** can be swapped with this script. If the user selects any other two layers and execute this script, then this script will just exit by doing no change.

15. Convert The PSD Files To JPG And Subsequently To PDF Format:

After all the fine tuning of the output PSD files are completed, the user can convert the PSD files to JPG files for album printing. The user can also reduce the size of the JPG files and put them all together within a pdf file for client proofing.

This operation can be done by the third script – **Export_PSD**. This script takes the list of PSD files and the output JPG folder as it's input. It then converts all the PSD files into best quality JPG files.

After the conversion, it asks the user whether he wants to convert the JPG files into a PSD file. If the user selects Yes, then the script will create a folder named **Small** under the output JPG folder. Then the script will reduce the size, resolution and quality of the JPG images and save them under the **Small** folder. Then the script will convert all the small JPG files to a PDF file and save it within the **Small** folder.

The user can send this PDF file to the client for reviewing.

16. Designing A New Template:

The program comes with a bunch of PSD templates, but the album designer can also integrate his own templates with this program. He has to just name the PSD files and **Images** layers appropriately. Then the program will automatically pick up the PSD files and use them as and when required.

For easier operation, the designer may create different Photoshop actions for grouping the image layers within the **Images** group and background layers within the **Backgrounds** group.

Please note that if all the layers are in L (landscape) or P (portrait) orientation, then the PSD file can be renamed to an existing old file. For example, suppose LLL_3.psd file is already existing. Now the designer has created a new PSD file and he feels it looks better than the old LLL_3.psd file. So the designer can rename the old LLL_3.psd file to LLL_4.psd and the new file to LLL_3.psd. So during template searching, the new LLL_3.psd will be chosen ahead of LLL_4.psd.

But if the layers have mixed orientations (some L and some P), then the designer should not rename the old files. The reason is that whenever those files are scanned during an album designing process, the map is generated with the actual orientation as the key and file names as the values.

For example, suppose the map already contains -

LLL: LPLP_3.psd, LPLP_7.psd, LPLP_20.psd

Now if the user creates a new PSD file whose actual orientation is LLPP and renames it to LPLP_3.psd, then the map is distorted. So when the program is executed again for a new client, then the program will wrongly place the third P orientation image to the third L orientation **Images** layer. So the designer should not rename the existing mixed orientation PSD files.

He can rename the files if the entries are not present within the map file present in the template PSD folder. The user can also edit the map file to place his own entries.

When the designer creates a new PSD template, he can validate the name and structure of the file by using the **OneClick_Album_Builder_CheckValidity** script. This script can be used in two ways, with a single file or with multiple files.

If a PSD template file is opened in Photoshop and the script is executed, then the script will validate the particular PSD template. The script will display the appropriate valid or invalid messages on the screen. Height and width in inches as well as the resolution in DPI will also be displayed.

But if no document is already opened in Photoshop and then the script is executed, then a dialog window will appear. The user can select any number of PSD templates for validation. The script will generate the report within the **OneClick_Album_Builder_CheckValidity.txt** file

in the user's desktop folder. Height and width in inches as well as the resolution in DPI will also be reported.

Please note that the CheckValidity script validates the template name with the combination of the image layer orientations. For single side templates, if they don't match, then the script will throw error and the designer has to fix the error. But for the full spread templates the filename is obviously different than the combination of the image layer orientations. So CheckValidity script will throw invalid filename error for full spread templates. But this condition is checked at the very end. At first the CheckValidity script checks whether the **Images** and **Backgrounds** groups are present and whether the layers names are in proper sequence. So for the full spread templates, if *Layer names are not correct* error is **not** thrown, then that means that the template has passed that validity check and is a valid template. At the very end it will throw *Filename doesn't match with number of layers*, but that error can be ignored.

Also note that this is not mandatory that the user has to validate the PSD templates before usage. He may not. But if the PSD templates are invalid, then during the album building, the PSD files will be moved to the **Skipped/spreadnumber** folder.

17. The Trial Program:

This is the abridged version of the full program. It will provide the same GUI as the full featured program. But in the input design text file the user can provide only one image for each side of the spread. The program will determine the image orientation, either L or P; select the appropriate PSD templates, paste the images in the corresponding layers and generate the final output PSD file. The encoded single page template can have only one image layer and the encoded full spread template can have only two image layers. So one valid design text file can have the following format –

Spread,Left,Right,PSD

1,1,1,

2,L_1.psd,P_1.psd,

3,1,1,X.psd

(Only one image layer is allowed for each page)

(Here X.psd can have only two image layers)

18. List Of All The Files Generated By All The Scripts:

It's better to list down all the files generated by all the above mentioned scripts in one place –

- **OneClick_Album_Builder**

Name	Location	Purpose
OneClickAlbumBuilderConfig.txt	Photoshop's preference folder	Saves the custom album specification
OneClickAlbumBuilder_Template_Map.txt	Template folder	Maps the actual orientation and file names of all the PSD templates as a key:value pair
<Client>_OneClickAlbumBuilder_OutputLog.txt	Desktop folder	Stores the log of the entire album building operation

- **Sequentially_OpenFiles**

Name	Location	Purpose
OpenClose_FileList.txt	Photoshop preference folder	Saves the list of all the files selected for the operation

- **OneClick_Album_Builder_CheckValidity**

Name	Location	Purpose
OneClickAlbumBuilder_CheckValidity.txt	Desktop folder	Stores the validity or reason for invalidity of all the selected PSD templates