Batch Processing Images on a Mac

MARNIE MACLEAN DESIGNS

Table of Contents

What is covered in this document?

I had about 1000 images, all with the same name, stored in a series of numerically named folders within the same parent directory. I wanted to batch process them in Photoshop using the Save For Web functionality. However, Save For Web doesn't remember the images' parent directory, it remembers the last place it saved an image.

I asked around for solutions and Dr Ralph Sutherland provided me with the following instructions:

1. open the terminal,
2. `cd /path/to/where/sampledirectories/`
3. list all the files in their directories into a text file: `ls */Image.jpg > list.txt`
4. make a batch folder: `mkdir batch`
5. edit the list.txt to make a batch of rename commands ie change: `folder1/Image.jpg` to cp `folder1/Image.jpg batch/folder1_image.jpg` For each one (grep will help here if you have an editor like text wrangler, or just get typing)
6. execute the list: `source list.txt`
7. now copies are all in batch ready to process `batch/folder1_image.jpg batch/folder2_image.jpg` etc etc
8. edit the list.txt to reverse the renaming change: `cp folder1/Image.jpg batch/folder1_image.jpg` to: `cp batch/folder1_image.jpg folder1/Image.jpg`
9. run the commands: `source list.txt`

This document provide detailed instructions for executing these steps, for people are not well versed in scripting and grep. Any mistakes or errors are my own. Sorry about that. For those who are more advanced, the above instructions or the "Synopsis" on page 6, will provide you with a quick reference or refresher, as needed.

Please note that these instructions are written for Mac users. PC users will need to adapt these instructions for use with PC compatible software.

Note: This process cannot be undone. Be sure you back up your work before you begin.

Getting started

You will need the following to get complete this tutorial.

- Mac
- TextWrangler (free)
- Terminal (found in Applications | Utilities)
- Photoshop

I have provided sample files that you may download, if you would like to try this technique for yourself.

You may download a zip file, with sample images and test scripts, from my site. The page is here. Click the download link, to download the zip file.

Definitions

[ ] Brackets, when used within example code, indicate variable text that you must define. Neither the brackets nor the text contained within, should be used in your own script. All instances will include examples that you can use as a reference.

grep A search utility that can match regular expressions in text. This is more powerful than a standard search and replace.

shebang A computing term for the characters: #! Also, a Ricky Martin song that is most likely not about scripting.

parent directory The directory (folder) that holds all the subfolders with images you'd like to include in your batch processing.

Unix commands

cp	copy
ls	list
>	output to
cd	change directory
*	wildcard: stands in for a single or string of alphanumeric characters
#!/bin/sh	included at the start of scripts to indicate the type of script we'll be running

Grep search terms

You can search TextWrangler's help menu for, "grep", to get more information on this topic. These are some commonly used search terms used for grep searches.

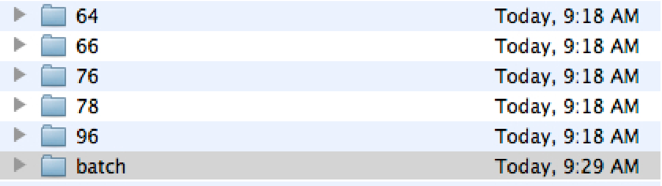
^	beginning of line
( )	defines a part of the string that can be referenced separately from other parts of the string
.	any character except for line breaks
\#	The # is a placeholder for any whole number and refers to items set off in parenthesis. The first item set off in parenthesis is `\1`, the second is `\2`, the third is `\3` and so on.
\d	wildcard for any single digit (0-9)
+	when combined with a wildcard, repeats that item. For instance `\d` will return a single digit, `\d+` will return any string of digits that are uninterrupted by any other characters.
$	end of line

What this documentation covers

- Creating a text file listing the desired images and their relative path within a parent directory
- Using a simple grep search and replace to create a shell script to copy each image to a batch folder and give it a unique identifying name
- Running the script
- Batch processing in Photoshop
- Creating a final script to copy the new images back to their original location with any name desired

Set up

Navigate to the directory where all your images are stored. Open the directory. Make a new folder into which your images will be copied temporarily. We'll call it "**batch**" for this tutorial but you may call it anything you like.



64	Today, 9:18 AM
66	Today, 9:18 AM
76	Today, 9:18 AM
78	Today, 9:18 AM
96	Today, 9:18 AM
batch	Today, 9:29 AM

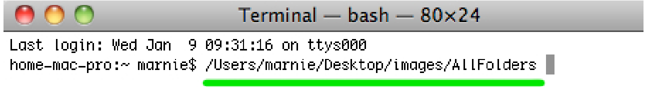
Create a list of images

For this step, you will need to know the path to the parent directory for all the images and the name or names of images you will be searching for.

Determine the path to the directory

Open Terminal.

Drag your parent directory into the terminal window. The string of text, starting with a slash, is the path to your directory. Copy this string.



```
Last login: Wed Jan  9 09:31:16 on ttys000
home-mac-pro:~ marnie$ /Users/marnie/Desktop/images/AllFolders
```

Start a script

In TextWrangler insert the following information:

```
#!/bin/sh
cd [path to directory]
```

e.g.: `cd /Users/marnie/Desktop/images/AllFolders`

Line 1 indicates the type of scrip being run, in this case, a Bourne shell. *Note, there are no spaces in this string.*

Line 2 uses the unix command "change directory" and the path to tell the script where to run the commands.

Define search string

In your own workflow, you'll need to do some problem solving to determine what your search string will be. Use the example given as a jumping off point.

To search for all the image.jpg files within the AllFolders directory, you need only go down one level, so your search string would look something like this:

```
*/image.jpg
```

The asterisk is a wildcard and will allow us to look in any directory, regardless of its name. The slash indicates that we are then looking inside these directories and the "image.jpg" indicates that this is the file name we are looking for.

In another case, you might want to find all the PNG files you've created for products on your website. You might have a search that looks like:

```
*/images/*.png
```

As long as you have a consistent hierarchy for the location of your images, you should be able to create an appropriate search string.

Once you have determined your search query, you'll add a third line to your script that looks like this:

```
ls [search string] > [filename].text
```

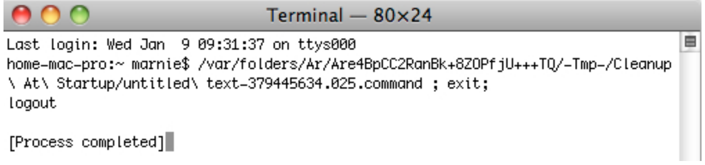
e.g.: `ls */image.jpg > list.text`

This will create a text file that lists the path to and name of all the images you want to batch process.

Run the script

Your script is now 3 lines long. In TextWrangler, go to the shebang menu [#!] and choose, Run in Terminal.

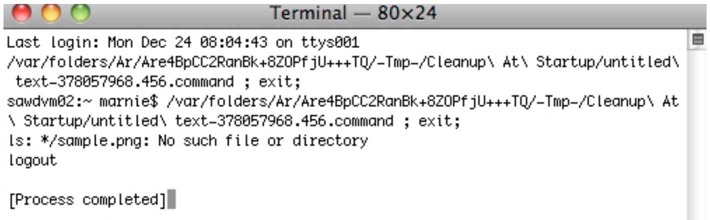
This will open a Terminal window. If all three lines were correctly defined, you should see the message that looks something like this:



```
Last login: Wed Jan  9 09:31:37 on ttys000
home-mac-pro:~ marnie$ /var/folders/Ar/Are4BpCC2RanBk+8Z0PfjU+++TQ/-Tmp-/Cleanup
\ At\ Startup/untitled\ text-379445634.025.command ; exit;
logout

[Process completed]
```

If you didn't define the correct search parameters in "Define search string" on page 3, you may get an error that looks like this:

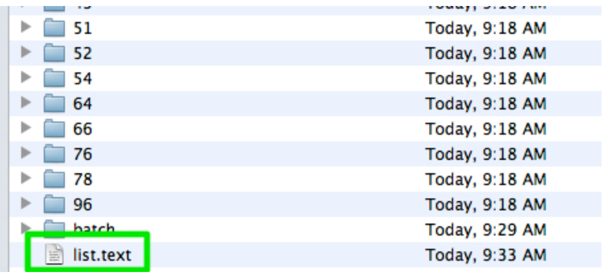


```
Last login: Mon Dec 24 08:04:43 on ttys001
/var/folders/Ar/Are4BpCC2RanBk+8Z0PfjU+++TQ/-Tmp-/Cleanup\ At\ Startup/untitled\
 text-378057968.456.command ; exit;
sawdvm02:~ marnie$ /var/folders/Ar/Are4BpCC2RanBk+8Z0PfjU+++TQ/-Tmp-/Cleanup\ At
\ Startup/untitled\ text-378057968.456.command ; exit;
ls: */sample.png: No such file or directory
logout

[Process completed]
```

You'll need to review your search string and determine what went awry.

Once your script has run successfully, the text file will appear in the parent directory you defined in line 2 of the script.



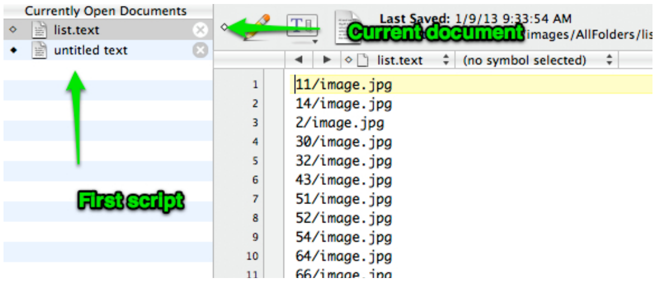
51	Today, 9:18 AM
52	Today, 9:18 AM
54	Today, 9:18 AM
64	Today, 9:18 AM
66	Today, 9:18 AM
76	Today, 9:18 AM
78	Today, 9:18 AM
96	Today, 9:18 AM
batch	Today, 9:29 AM
list.text	Today, 9:33 AM

Copy and name images

Now, we'll use the information in the text file to copy all the images to a folder for batch processing. We'll also give each image a name that coincides with its location so the images can be put back when we are done. To do this, we'll use a grep search and replace.

Create a new script

In TextWrangler, open the text file you just created. It should look something like this.



```
11/image.jpg
14/image.jpg
2/image.jpg
30/image.jpg
32/image.jpg
43/image.jpg
51/image.jpg
52/image.jpg
54/image.jpg
64/image.jpg
66/image.jpg
```

Note that you can access your first script by clicking on it in the left panel.

We want to copy each file from its current location into the batch folder with the name of its parent directory as part of the file name. The format will look like this:

```
cp [folder]/image.jpg batch/[foldername]_image.jpg
```

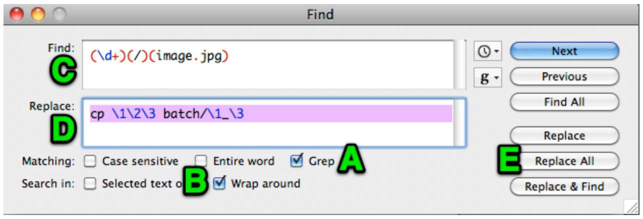
e.g.: `cp 1/image.jpg batch/1_image.jpg`

Note: the only spaces in this string are after the "cp" command and before "batch".

Still in TextWrangler, go to your Search menu and choose, Find [cmnd+f].

Use the same logic that you used in "Define search string" on page 3 to define your grep find and replace. We'll use the easiest example and you can extrapolate out from there.

Our current document lists a directory name which can be any whole number, followed by a slash and then, "image.jpg"



Find: (\d+)(/)(image.jpg)

Replace: cp \1\2\3 batch/\1_\3

Matching: □ Case sensitive □ Entire word ☑ Grep

Search in: □ Selected text o ☑ Wrap around

A. Turn on the option for Grep

B. Turn on the Wrap around option

C. Find: `(\d+)(/)(image.jpg)`

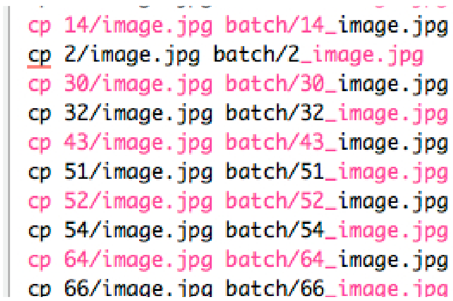
D. Replace: `cp \1\2\3 batch/\1_\3`

E. When you are done, click Replace All to apply the search and replace.

The find command is looking for any string of numbers. The parenthesis around it sets it off as a part of the string we want to be able to retrieve on its own, the plus sign ensure we catch folders numbers greater than 9. Next it finds the slash and sets it off as its own section of the string, and finally it does the same with "image.jpg". You can click the "Next" button to test if your search is working. It should select the entire contents of a single line.

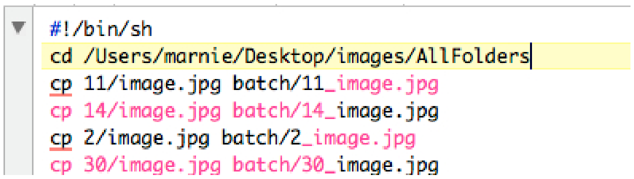
The replace is going to add the copy command [cp], places the entire original string of text, then adds a space, "batch," a slash and takes the number-slash-image.jpg string and changes it to number-underscore-image.jpg.

When you apply the find and replace, a confirmation message will tell you how many instances were changed. Your results will look something like this:



```
cp 14/image.jpg batch/14_image.jpg
cp 2/image.jpg batch/2_image.jpg
cp 30/image.jpg batch/30_image.jpg
cp 32/image.jpg batch/32_image.jpg
cp 43/image.jpg batch/43_image.jpg
cp 51/image.jpg batch/51_image.jpg
cp 52/image.jpg batch/52_image.jpg
cp 54/image.jpg batch/54_image.jpg
cp 64/image.jpg batch/64_image.jpg
cp 66/image.jpg batch/66_image.jpg
```

To complete the script, add the same first two lines you added in "Start a script" on page 3 so that your script looks something like this:

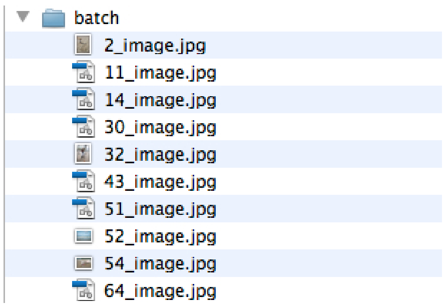


```
#!/bin/sh
cd /Users/marnie/Desktop/images/AllFolders
cp 11/image.jpg batch/11_image.jpg
cp 14/image.jpg batch/14_image.jpg
cp 2/image.jpg batch/2_image.jpg
cp 30/image.jpg batch/30_image.jpg
```

Now you can run your script, just as you did for "Run the script" on page 3.

Batch process images

You should now have a copy of each image, stored in the batch folder you created in "Set up" on page 3.



```
batch
    2_image.jpg
    11_image.jpg
    14_image.jpg
    30_image.jpg
    32_image.jpg
    43_image.jpg
    51_image.jpg
    52_image.jpg
    54_image.jpg
    64_image.jpg
```

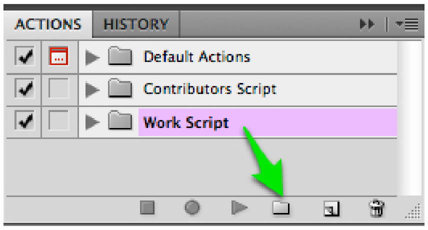
Batch Processing Images on a Mac

MARNIE MACLEAN DESIGNS

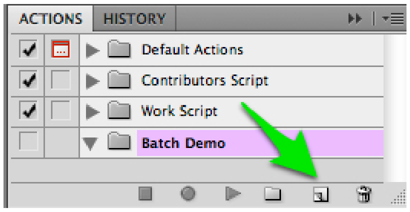
Create an action

Open one of your images in Photoshop.

From the Windows menu, choose, Action. All actions must be part of a set. There is a Default Actions set that comes built into Photoshop, which you may use, or you may define a new set by clicking the icon for a new set, at the bottom of the actions palette.

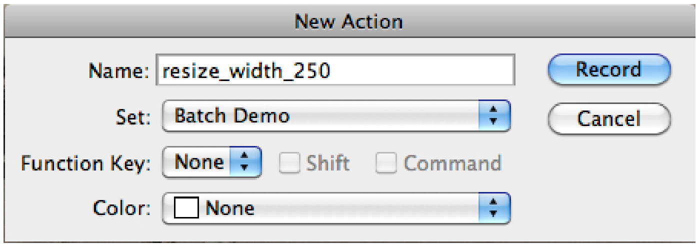


With your desired set selected, click the new action icon.



Name and Record Action

A dialog box will appear. Give your new action a meaningful name, then click the Record button.



Name: resize_width_250

Set: Batch Demo

Function Key: None ☐ Shift ☐ Command

Color: None

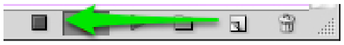
For this example, I will be resizing all images to be 250 pixels wide, saving for web into the batch folder and then closing the image without saving.

With the recording running, go to the Image menu and choose, Image Size. Set the desired width of 250 pixels and apply the change.

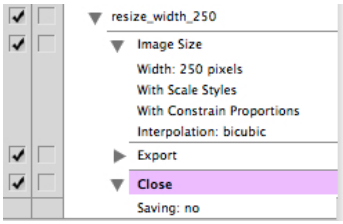
Go to the File menu and choose, Save for Web & Devices and set the correct file format and quality desired. Click save when you are ready. Choose the batch folder as your destination. When you confirm this step, you'll be asked if you want to overwrite the existing copy. Say, Yes.

Lastly, close the document **without** saving.

Now stop recording your action.



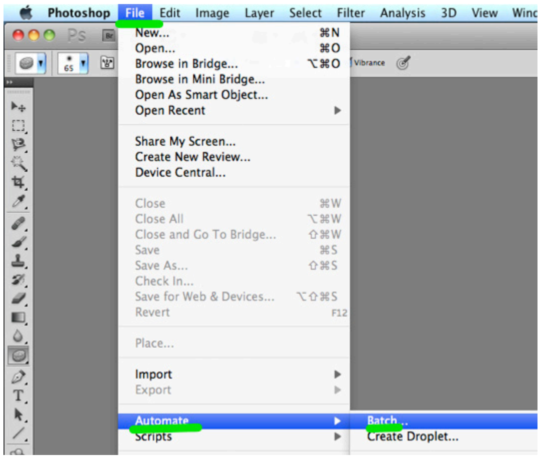
You will be able to review the specifications for your action, within the actions palette.



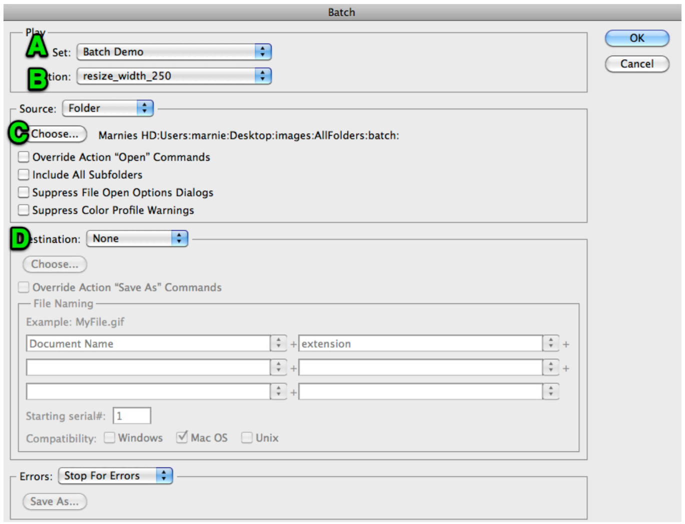
resize_width_250
- Image Size
  - Width: 250 pixels
  - With Scale Styles
  - With Constrain Proportions
  - Interpolation: bicubic
- Export
- Close
  - Saving: no

Run the batch

From the File menu, choose: Automate | Batch



The Batch dialog box will appear.



Play

Set: Batch Demo

Action: resize_width_250

Source: Folder

Choose... Marnies HD:Users:marnie:Desktop:images:AllFolders:batch:

☐ Override Action "Open" Commands

☐ Include All Subfolders

☐ Suppress File Open Options Dialogs

☐ Suppress Color Profile Warnings

Destination: None

Choose...

☐ Override Action "Save As" Commands

File Naming

Example: MyFile.gif

Document Name + extension +

Starting serial#: 1

Compatibility: ☐ Windows ☑ Mac OS ☐ Unix

Errors: Stop For Errors

Save As...

A. Choose the correct Action set.

B. Choose the correct Action

C. Click the button to navigate to the batch folder you created in "Set up" on page 3.

D. Make sure Destination is set to, None.

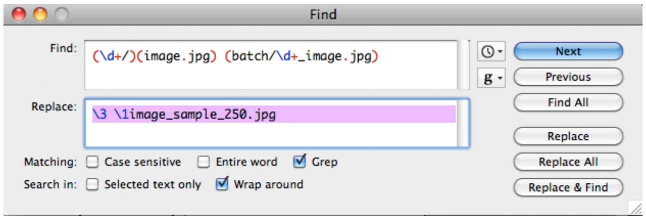
Click OK when you are ready and Photoshop will start opening each image, and running the action. You'll be alerted of any problems or errors. If you realize that your action is not running correction, you can press, escape, to interrupt the batch processing.

Copy images back

Use the same method outlined in "Copy and name images" on page 4, to create a script to copy all the images back to their original directories with the proper naming convention. In this case, we want these new images to be named image_250.jpg, since they are all 250px wide, though you can save them back with their original name and overwrite, if you prefer.

Go back to your script in TextWrangler. Using the same script we used to copy all the images to the batch folder, bring up the Find dialog box.

Use the following find and replace criteria:



Find: (\d+/)(image.jpg) (batch/\d+_image.jpg)

Replace: \3 \1image_sample_250.jpg

Matching: □ Case sensitive □ Entire word ☑ Grep

Search in: □ Selected text only ☑ Wrap around

Find: `(\d+/)(image.jpg) (batch/\d+_image.jpg)`

Replace: `\3 \1image_250.jpg`

This will reverse the direction of the copy and rename the files so they are all called image_250.jpg.

If you didn't plan to rename the files from their original naming convention, you could simply search for the first half of the string, then the second half of the string and reverse their position, so that your find and replace might read:

Find: `(\d+/image.jpg) (batch/\d+_image.jpg)`

Replace: `\2 \1`

Run your script as indicated in "Run the script" on page 3, and you're done.

That's it. Do a spot check on your files to make sure it worked as expected.

Synopsis

These are the bare bones instructions, for the already proficient and those who just need a quick refresher.

1. Navigate to parent directory
2. Create a folder to which images will be copied
3. Create a text file listing all the images and their path, adjusting search string as necessary to include full path in relationship to parent directory:

   e.g.: `ls */image.jpg > list.text`

4. Open text file
5. Find and replace using grep to create command: `cp [sourcepath/filename] batch/[source path_filename]`

   find e.g.: `(\d+)(/)(image.jpg)`
   replace e.g.: `cp \1\2\3 batch/\1_\3`

6. Run script
7. Batch process images using Photoshop. Save to same folder
8. In text file, find and replace to copy back to source folder, adjusting for desired final file name: `cp: [batch/sourcepath_filename] [sourcepath]/finalname]`

   find e.g.: `(\d+/image.jpg) (batch/\d+_image.jpg)`
   replace e.g.: `\2 \1`

Copyright and Disclaimer

This document is copyright Marnie MacLean, 2013 and is provided for your own personal, non commercial use. You are welcome to pass it along, but please keep it whole and in its original format without modification. The images in the sample files are also copyright protected, but they have my URL plastered across them so it'd be really weird if you decided to repurpose them. But if you were thinking about it, please don't.

Marnie MacLean has no idea what she's talking about and takes no responsibility for anything that blows up, stops working, or disappears because you made the questionable decision to trust some random person on the internet. She sincerely hopes she happened to get most of this stuff right, despite her short comings, but she welcomes feedback and corrections if you have them.

She does a bunch of stuff that's entirely unrelated to this documentation. You can find out all about it at her website: http://MarnieMacLean.com