

# Scripts

Last Updated: 29-Jan-2019

# Script Files

Script files define one or more transformations and the way they are connected to each other in the image tree; they perform several useful functions:

- Saving a record of your work on one or more images so you can pick up editing later right where you left off or make changes to any stage of the workflow.
- Applying the same sequence of operations to multiple images.
- Sharing workflows you have developed.

## Script/Save Script

To use the **Save Script** command, first select an image whose workflow you want to save—typically the last image in a branch of the image tree. Saving a script file based on this image records all the transformations you used to create the image, all the way back to when you opened an original file(s). Script files are usually very compact as they just list the commands and their settings for each transformation and tool. If you performed a lot of free-hand retouching or masking, the script file may be somewhat larger as each stroke you made is captured in the script file.

There are three forms of the **Save Script** command. While mostly the same, they deal differently with the original image files on which the image depends.

### Save Script with Image Names

This option saves the pathnames of the original files, but it does not save the files themselves. This is the option to use when working on an image you want to refine later.

### Save Script without Image Names

This option omits the pathnames of the original files. This lets you save the workflow you used to create an image and apply it to other image files.

When you open a script without image names, Picture Window asks you for the name of the file you want to open. When asking for pathnames, the starting folder is set to your default folder if the original file was opened using the File/Open from Default command.

### Save Script with Image Copies

This option saves TIFF copies of all the original files in a folder with the same name as the script file. Use this option to archive images you might want to edit later, when you don't want to worry about one of the original images being deleted, edited or renamed in the interim.

Files are saved with their original filenames to make it easier to tell where each one came from. If two or more files have the same filename (presumably because they had different extensions or came from different folders originally), subsequent filenames are made unique by appending a file number.

If a File Open has more than one pathname in its file list, a warning is issued and only the current file is saved.

## **File Modified Date and Time**

In addition to saving all the commands, script files also include the time and date each input file was last modified. When a script file is reloaded, the saved time and date is compared to the current modification date to make sure the file has not been changed since it was used as part of the script. If there is a discrepancy, you have the option of proceeding with the modified file or canceling. If you don't get a warning, the input files are the same as the ones you originally used.

## **Save Script Options**

The advantage of Save Script with Image Names is that you use less disk space and, if the original file is reworked, those changes will be reflected in the result image when the script file is subsequently opened. The disadvantage is that if you happen to delete, rename, move or modify one or more of the files from which the result image was created, you will no longer be able to use the script file to recover the result image. This situation can also arise if the original file was obtained by pasting it from the clipboard or from some other non-repeatable source.

If you use Save Script with Image Copies, the original images from which the result image was created are saved along with the script file and you should be able to recover the result image even if one of the original files was deleted or if one or more of the original images came from the clipboard. On the other hand, if you rework one of the original files after saving the script and images, those updated files will not automatically update the result image since Save Script and Image Copies effectively creates a snapshot in time and the result image will never change. However, if you wish to update it, after loading the script you can manually reload whatever input images you want to refresh and then you can re-save the script with the updated files.

## **Open Script...**

The Open Script command first asks for a script filename to open. If the open is successful the commands contained in the script file are re-played, restoring the final image and all the intermediate images from which it was created. If the script was saved without image names, you will be asked to open the files to which you want to apply the script.

You can also use the File Open command to open script or workspace script files, but only if you select just the script file. Attempting to include script files in a multiple file selection will result in an error message. This method of opening script files is more convenient if the script file is stored along with image files since the Open Script command always starts up in the default script folder, forcing you to navigate to the image folder.

To help you keep track of which script is currently active (if any), the filename of the last loaded script or workspace script file is displayed in the main window caption.

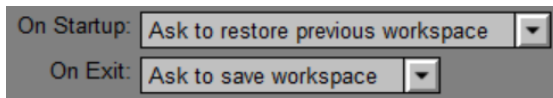
## Workspace Scripts

In addition to the save and open script commands described above, there is a parallel set of commands for saving and loading workspace scripts. Workspace scripts are like regular scripts except:

- Regular scripts save only the information needed to reconstruct the current image at the time the script was saved. Images downstream from the current image or unrelated to the current image are not saved. Regular scripts have the file extension: *.script*.
- Workspace scripts save all the images in the image tree, including those that are unrelated to the current image. Workspace scripts have the file extension: *.workspace*.

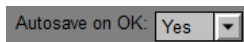
## Saving and Restoring Workspace Scripts between Sessions

When you shut down Picture Window, if the workspace is not empty, it saves a workspace script (with images) under the name *Previous.workspace* so you can pick up where you left off in a future session. Similarly, when Picture Window starts up, it can set to ask to load a saved workspace file. The process of saving and restoring the workspace file can be turned on or off via settings in the File/Preferences dialog box.:



## Autosaving Workspace Scripts

If Autosave is enabled, each time you click OK or Apply in a transformation dialog box, or just before you close an image or clear the workspace, the current workspace is saved under the name *Autosave.workspace*. This lets you recover your work up to the last completed transformation in the event of a crash. The process of autosaving the workspace file can be turned on or off via settings in the File/Preferences dialog box.:



The four most recent versions of the *Autosave.workspace* file are retained, should you need to retrace your steps further back in time. These additional files are named *Autosave1.workspace* ... *Autosave4.workspace* (most recent to least recent). To restore an autosaved backup, use the Script/Open Workspace Script command and select the workspace file you want to restore.

## File Save

At the time image files are saved, Picture Window lets you save a script (either with image names or with image copies) to regenerate the image later. The script file is stored in the same folder as the image file and with the same name as the image file with ".script" added to the end.

This script file can be used later by the File Open or Open Script command to rebuild the image from scratch. This lets you recover and modify all the work you did to create an image.

## Save Script with Image Copies

Use this to archive work you want to come back to later without having to worry about some of the original files being modified, deleted or unavailable.

For example, say you create an image you print and offer for sale. You can use **Save Script and Images** to preserve a copy of the original image(s) and your workflow. This documents exactly how the final image was created and lets you restore both the original images and the workflow later. You can then modify the workflow to refine the image later if you wish to tweak the image or simply reload it to make another copy of the image to print.

To be extra safe, make sure you save the final image as well as the script that produces it in case there is some problem running the script under a newer version of Picture Window.

Saving image copies facilitates moving your work to a different computer. If you transfer the script file and its image folder, you should be able to continue your work on the other computer by opening the script file.

When saving with images, the script file and image folder must be within the same folder, although you can move them both to a new folder if you wish. If you move the script file and its image folder to different folders, the script file will not be able to find the images and the **Open Script** command will fail.

## Save Script with Image Names

Use this to save a workflow when you know all the original files will remain available.

## Save Script without Image Names

Use this if you create a workflow that applies a sequence of operations to an image to get a specific effect and want to apply them to different images later.

## Running Scripts on Startup

You can open Picture Window and run a script when it starts up by specifying the name of a script or workspace script file in its command line or by using the Open With... command in Windows Explorer to register Picture Window as the application to open files with the extensions: .script and/or .workspace.

## Organizing Script Files

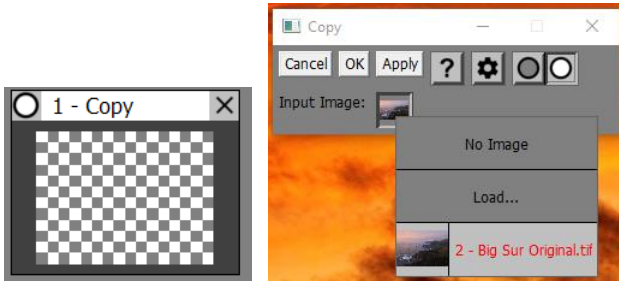
Workspace scripts are saved by default in *My Documents/Picture Window Pro/Workspace Scripts*.

Script files created when image files are saved are stored in the same folder as their associated image file. Otherwise script files are saved by default in the last folder in which a script or image file was saved.

## Creating Generic Scripts with Convert to Copy

One way to use scripts is to record what operations you applied to a starting image to get one or more result images. Sometimes these operations are very specific to a given image, but you may find yourself applying the same sequence of operations over and over to different images.

One way is to use the Save Script without Image Names command. This works well if you want to apply the same operations to one or more image files, but not if you want to apply the script to an image in the middle of an existing branch. To get around this problem, use the Edit/Convert to Copy command to replace the File Open with a Copy transformation just before saving the script. The Copy transformation acts as a placeholder for the input image for the script. When you open the script, instead of asking you for the name a file to open, you will get an empty Copy transformation at the top of the branch:



All that remains is to connect the Copy transformation to the image you want to process -- double click on the Copy transformation and use the Input Image control to select an image. Finally, click OK to close the Copy transformation and recalculate.

If your workflow has multiple input images, convert each one to a Copy transformation and connect them one at a time to the images you want to process. With multiple Copy transformations, it is a good idea to edit the caption of each one to help you remember what each one is used for.