

Scripts

Last Updated: 25-Aug-2025

Contents

Script Files.....	3
Script/Save Script	3
File Modified Date and Time	4
Color Management Settings.....	4
Save Script Options	4
Open Script... ..	5
Workspace Scripts	5
Snippets.....	5
Inserting a Snippet.....	6
Saving and Restoring Workspace Scripts between Sessions.....	6
Terminating Scripts	7
Autosaving Workspace Scripts.....	7
File Save.....	7
Save Script with Image Copies	7
Save Script with Image Names	8
Save Script without Image Names	8
Missing Files	8
Search List	8
Manually selecting a replacement file	10
Running Scripts on Startup	11
Organizing Script Files	11
Creating Generic Scripts with Convert to Copy	11
Generic Script Example	12

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 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.

By default, script images are saved as uncompressed TIFF files, but you can switch to compressed TIFF or JPEG or ask for the file type each time you save with image copies. To control how image files are saved, use the Save Script Images As setting in the

File/Preferences dialog box. Saving images as JPEGs can save a lot of disk space, but when you reload the script you will not get back exactly the images you started with due to compression artifacts.

Note: When using Save Script with Image Copies, if a File Open has more than one pathname in its file list, a warning is issued and only the current file is saved, to avoid potentially saving a large number of images in the folder. This situation typically occurs when you create a File Open using the File/Open List... command in preparation for performing a batch operation on a sequence of files. If you really need to save all the files, you need copy them to a folder and save it yourself.

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.

Color Management Settings

Various color management settings such as the working color space can affect the way images are processed. The color management settings in effect when a script is saved are recorded in the script file. When the script file is reloaded, the saved settings are compared with the current settings to see if anything has changed. If changes are detected, Picture Window lets you know and gives you the option of reverting to the old settings or continuing with the current settings. This warning is issued since, if the color management settings have changed, the script may not regenerate the same images as when it was originally saved. If you wish, you can disable these warning messages from the File Preferences dialog box.

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*.

Snippets

If you find yourself re-entering the same several transformations over and over in different situations, snippets can save you a lot of time and effort. Snippets are like fragments of a branch that you can insert after any transformation. Unlike scripts and workspace scripts, snippets are subject to the a few limitations:

- All the transformations in a snippet must be on the same branch, although they do not need to be contiguous.
- Auxiliary inputs and images referenced by masks are not restored when the snippet file is reloaded.
- Top level transformations, splitters and side branches are not allowed in snippets.

Creating a Snippet

To create a snippet, first use the image browser to select the transformations you want to include. This selection must comply with the limitations listed above. Next, click the  button at the top of the image browser to save copies of the selected transformations as a snippet file.



In this example there are three selected transformations. When you click , Picture Window asks you for the name of a snippet file to save and then copies the selected transformations to the snippet file.

If you are likely to modify the settings of or close any of the snipped transformations after you reload them, consider saving them bypassed. Then, when you later insert the snippet, its transformations will not all recalculate every time you edit one of them.

Inserting a Snippet

To insert a snippet, start by selecting the transformation you want to insert the snippet after. Next, select the **Script/Insert Snippet...** command from the main menu, or click the  button on the tool bar at the top of the main window.



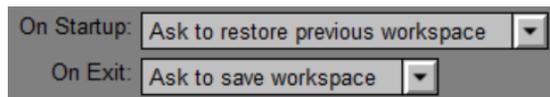
Pick the snippet file you want to insert, and Picture Window then inserts the transformations in the snippet file immediately after the current transformation.

If the transformations in the snippet file contain references to other transformations, either in the form of auxiliary inputs or masks, these references are cleared when the snippet file is loaded.

If you hold down the **Shift** key while selecting this command (or while clicking the button), instead of asking you which snippet file you want to open, Picture Window loads the last snippet file you used, assuming it exists. This is convenient if you are inserting the same snippet repeatedly.

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.:



Terminating Scripts

While a script, workspace script or snippet is being loaded, pressing and holding the Esc key causes the script to be terminated just before it starts executing the next transformation. This can be useful if you accidentally start loading a long script and don't want to wait for it to run all the way to completion. Once the script is terminated you have the option of clearing the workspace to get rid of any transformations already loaded.

Autosaving Workspace Scripts

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 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 or a workspace script to regenerate the image later. The script or workspace script file is stored in the same folder as the image file and with the same name as the image file with ".script" or ".workspace" added to the end. You can configure the option to save a script or workspace script file when saving image files via the File/Preferences command by clicking on the Default File Save Options... button.

A saved 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. Only the transformations necessary to regenerate the saved image are restored.

A saved workspace script file can be used later by the File Open or Open Workspace Script command to rebuild all the images in the workspace at the time the file was saved. This recovers not just the transformations necessary to regenerate the save file but also any other transformations present in the workspace at the time it was saved.

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.

Missing Files

When you open a script or workspace script file that was saved with Image Names, Picture Window attempts to re-load all the referenced image files. Within script files, Image references are recorded as the full pathname of the image file, so if all the image files remain in exactly the same locations as when the script was originally saved, there should be no problem locating them. However, there are several reasons why Picture Window may not be able to locate an image file:

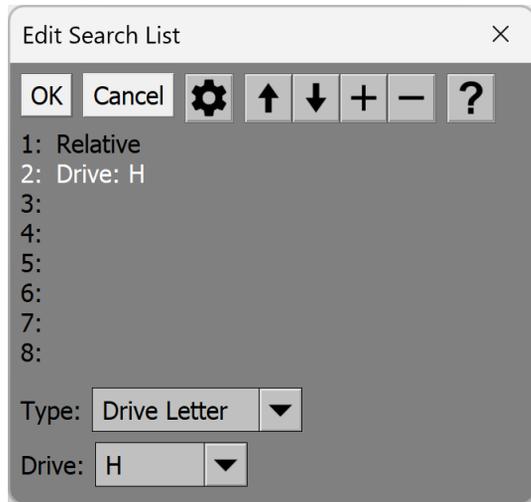
- **Deleted image file** – in this case, there is not much you can do other than possible substitute some other image file.
- **Renamed image file** – in this case the filename needs to be changed to the new filename,
- **Renamed folder** – if one or more of the folders containing the image file have been renamed or relocated, the pathname needs to be updated to reflect the new location.
- **Changed disk letter** – if, for example, you keep your image files on a separate disk drive and Windows assigns it a different drive letter, the drive letter part of the pathname need to be updated.

Search List

Some of these operations can be automated using a Search List which consists of a list of rules that are applied in sequence to find missing image files in scripts. Each rule consists of instructions on how to modify the stored image file pathname – if a file with the modified pathname exists, it is substituted for the original pathname and the script continues. Before performing the substitution, a warning is issued so you can decide if the new pathname is correct.

If you save the script file after successfully locating any missing image files, the new script file will contain the updated pathnames.

To create or modify the search list, use the **Script/Edit Search List...** command from the main menu. This displays a dialog box such as the following:



The search list consists of a numbered list of up to 8 rules which are displayed in the main body of the dialog box. The current rule is highlighted in white – to select a different rule, just click on it.

You can rearrange the order of the rules using the up/down arrow buttons. These buttons move the current rule up or down one position in the list.

You can delete the current rule using the – button. The + button inserts a new empty rule at the current rule location.

Type

The Type control lets you change the type of the current rule. The possible rule types are:

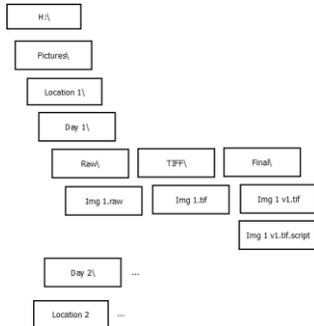
None – this rule type is ignored. You can use this to create a place holder for a future rule.

Drive Letter – this rule type lets you specify a different drive letter to substitute for the one in the image file pathname. A Drive Letter control is displayed to let you pick the desired letter. Use this rule type to deal with Windows changing the drive letter of the drive containing the image files.

Folder – this rule type lets you specify the pathname of a folder in which to find the image file. The folder name is displayed – you can either enter it manually or you can click the folder button to browse for the new folder. Use this rule type if the image files have been moved to a particular new folder.

Relative – this rule type is useful if you have renamed one of the folders that contains both the script file and the image file. It searches for the image file in a folder with the same relative location to the script folder that the original image file had to the original script folder.

For example, suppose your files are organized as follows:



All image files are in the folder: H:\Pictures. Within this folder, there are subfolders for each location or photo shoot. If it continues over multiple days, there is a subfolder for each day. Within this folder are subfolders for raw, TIFF and final images and there is a script file for each final image.

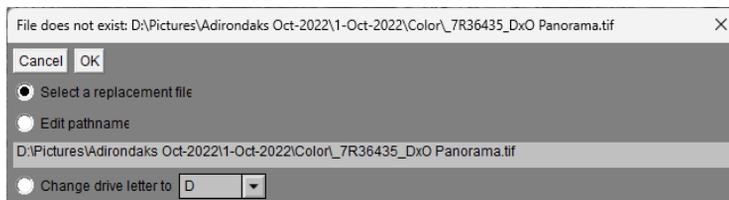
The script file loads `Img 1.tif` using the pathname:

H:\Pictures\Location 1\Day 1\TIFF\Img 1.tif

If you subsequently rename the Location 1 folder to something else such as Kansas, the script file will no longer be able to find `Img 1.tif` since its pathname has changed. However, if you include Relative as one of the search rules, the script file will correctly modify the pathname and be able to locate the file since it is still in the same location relative to the script file.

Manually selecting a replacement file

If the search list is empty or if none of its rules result in a matching image file, the following dialog box is displayed to let you specify the substitute pathname manually:



Select a replacement file

Select this option and click OK if you want to select a replacement for the missing file using the standard Windows File Open dialog box.

Edit pathname

Select this option if you want to enter the pathname of the replacement file directly. This is useful, for example, if you have renamed a folder. Just make the necessary changes to the pathname displayed just below and click OK.

Change drive letter to

Use this option if the drive containing the missing file has been assigned a different drive letter. Then select the new drive letter and click OK.

If you cancel this dialog box, a default checkerboard image is substituted for the image file and the rest of the script is loaded:



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

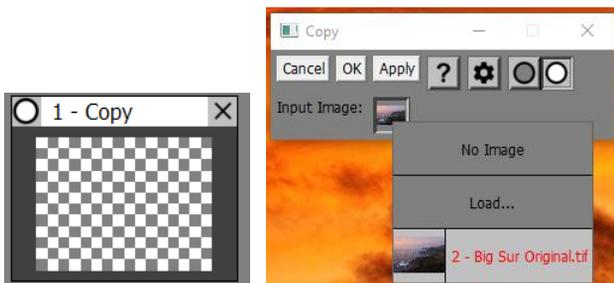
Workspace scripts are saved by default in *My Documents/Picture Window Pro/Workspace Scripts*.

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

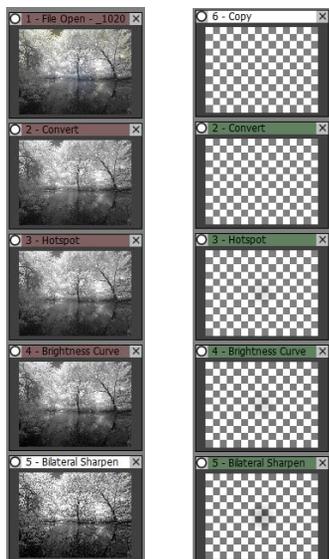
transformations, it is a good idea to edit the caption of each one to help you remember what each one is used for.

Generic Script Example

Here is an example of a simple generic script with a single input (see below). It is designed to process infrared images by applying a standard sequence of transformations. First it opens an image, then it converts it from color to black and white, then the central hotspot is removed, then the brightness is adjusted, and finally it is sharpened.

Once you create the workspace, convert the File Open to a Copy transformation by right clicking on the File Open thumbnail and selecting Convert to Copy from the popup menu. At this point, since there is no image to copy defined yet, all the thumbnails show up as checkerboards (see below).

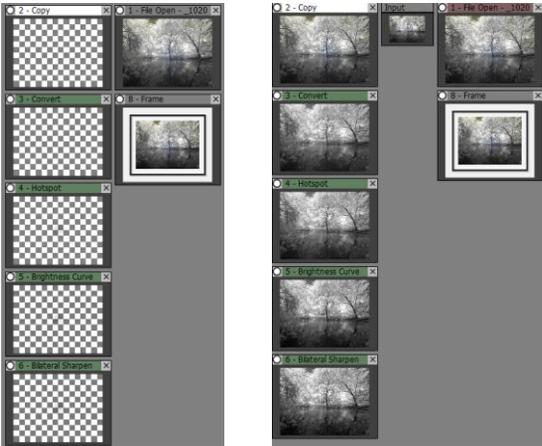
To create the generic script, first select the last transformation (Bilateral Sharpen in this example) and then use the Script/Save Script without Image Names command. Since there are no File Opens in the workspace so it doesn't actually matter if you save with or without image names.



Now, say you have another workspace into which you want to incorporate the saved generic script (see below):



To do this, open the generic script with the Script/Open Script command, but do not close all images before loading the script when Picture Window asks you. This places the saved transformations from the script file in a new branch alongside the existing transformations.



To apply the generic script to the File Open transformation (or to any other transformation in the workspace), double click the Copy transformation thumbnail to open its dialog box and select the image to which you want to apply the script and then click OK. Then as shown above, the transformations in the generic script are performed on the image you selected.

If you want to insert the transformations from the script between the File Open and the Frame transformation, you can select the four transformations (not including the Copy transformation) and copy them to the other branch. At this point you could delete the branch containing the script transformations or leave it around in case you want to copy its transformations to another branch.

