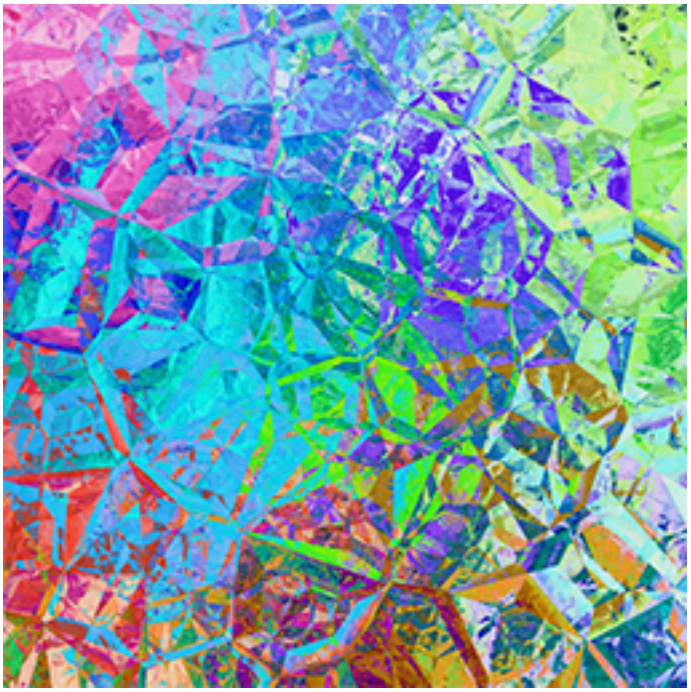


FilterForgeBatchWrapper / ffBatch.ahk/.exe

This makes it simpler to render many images with many filters and presets in one go, or batch, using the Filter Forge command-line renderer.

Hands-On Quick Start

First, double-clock *ffBatch.exe* and let it run until a notification box appears. Click "*Ok*" to close the notification box. This creates a default *[yourComputerName].ini* file.

Next, put a few smallish image failes you would like to see filter renders of in any folder. Drag and drop that folder onto *ffBatch.exe* and watch what happens; you'll see renders in progress in the newly created ff_output subfolder. Futz with the True/False settings in the automatically generated *[yourComputerName].ini* file to learn how it works. From that file you can change settings as follows.

You may alternately create a srcImg folder, copy image files to it, then run *ffBatch.exe*.

You will see output in either an ff_output folder or subfolder of the folder you drag onto *ffBatch.exe*, depending.

Features

In general, ***FilterForgeBatchWrapper (ffBatch.ahk/.exe) allows you to render, via Filter Forge's command-line renderer, many images, using many different filters, with presets you choose, and using random variations per preset***. (Alas, at the writing, it only uses the one preset for every filter you specify —plans are to choose presets per filter.)

It is designed for network rendering using more than one Filter Forge license; as it renders for a given file, it places an *imageFileName.jpg.rendering* empty placeholder file in the target directory for completed renders. If you run multiple copies of *ffBatch.exe*, it checks for the existence of such files before a render, and if the check file exists, it skips it and selects another file to render. You may therefore run the program on many workstations cooperating on source images from a server directory; theoretically by drag-and-drop from a network drive folder—which is not tested, but simultaneous renders with proper .ini [paths] section changes have been tested and they work, last I tested :)

There are two ways to start batch renders:

1. Drag-and-drop an image folder onto the *ffBatch.exe* file, or
2. Copy all the images you wish to render into the srcImg directory in the same folder as the executable, and run *ffBatch.exe*.

Note that it will not work if you select a group of images and drop them directly onto *ffBatch.exe* (this is a known issue; you would expect drag and drop to work from any context but it does not). You must have images in a folder, and drag and drop the folder onto *ffBatch.exe*.

All relevant path etc. settings to control this are in the .ini file mentioned in the "Quick Start" section; that .ini file with working defaults is created automatically by the program at first run.

Note: it may not render anything by drag-and-drop on the first run (as it reloads the program after making the defaul .ini file). Just drag-and-drop again.

The following feature descriptions also provide the .ini settings that control the respective features.

Specify the preset number to use (currently, the same preset number will be used for all filters; vary preset(s) per filter planned). Ini setting:

```
Global Filter Preset Number Override=n
```

—where *n* is any existing preset number in the filter.

By default, this exploits the variation (randomization) option built into many Filter Forge filters, by using a random variation number between 1-30,000 (where applicable) for every image in the folder; the range of numbers to draw from is controlled thusly in the [options] section of the .ini:

```
Variation Randomization Range=1-30000
```

To make that variation always be e.g. the number 7, change that to:

```
Variation Randomization Range=7-7
```

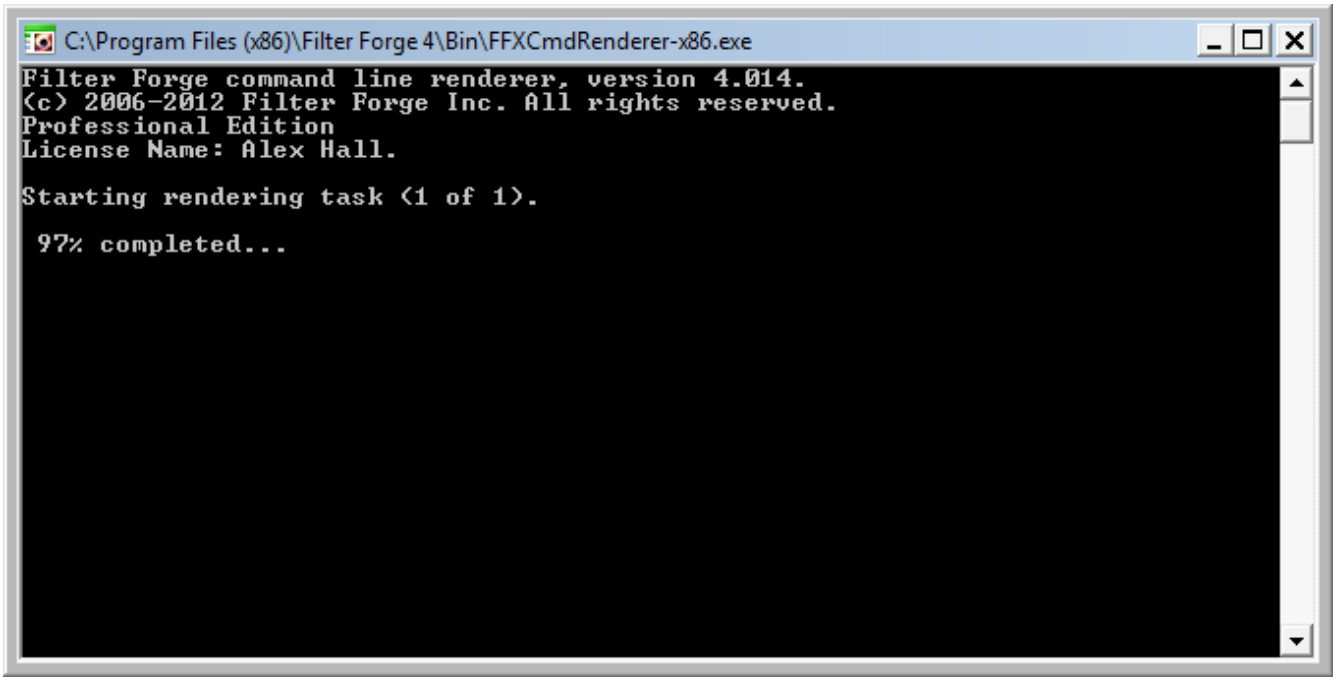
Likewise, to randomly pick a number between e.g. 7,000 to 8,000 for every image, change that to:

```
Variation Randomization Range=7000-8000
```

Show or hide the render console during the batch run. Ini setting:

```
Rendering Console Visibility=Hide
```

Change *Hide* to *Show* to make the console visible. I suggest you want that setting at *Hide* (the default), unless you're keen on seeing the render console repeatedly pop up and interrupt other work you may be doing:



Save the render settings of each image into an ff_output/XML_FFXMLbatches subfolder:

```
Keep Generated XML And FFXML Batch And Filter Settings Files=True
```

Change *True* to *False* to not save those.

Randomly select images to render, e.g. if you are working with a lot of source images and want to see varied results quickly:

```
Randomly Shuffle Source Images List=True
```

Randomize the order of filters used for rendering—to the same end:

```
Randomly Shuffle Filter List File Names=True
```

Use one randomly selected filter (from the filter list) only once for each image, to see many randomly selected filter effects only one per image:

```
Render Only One Random Filter Per Image=True
```

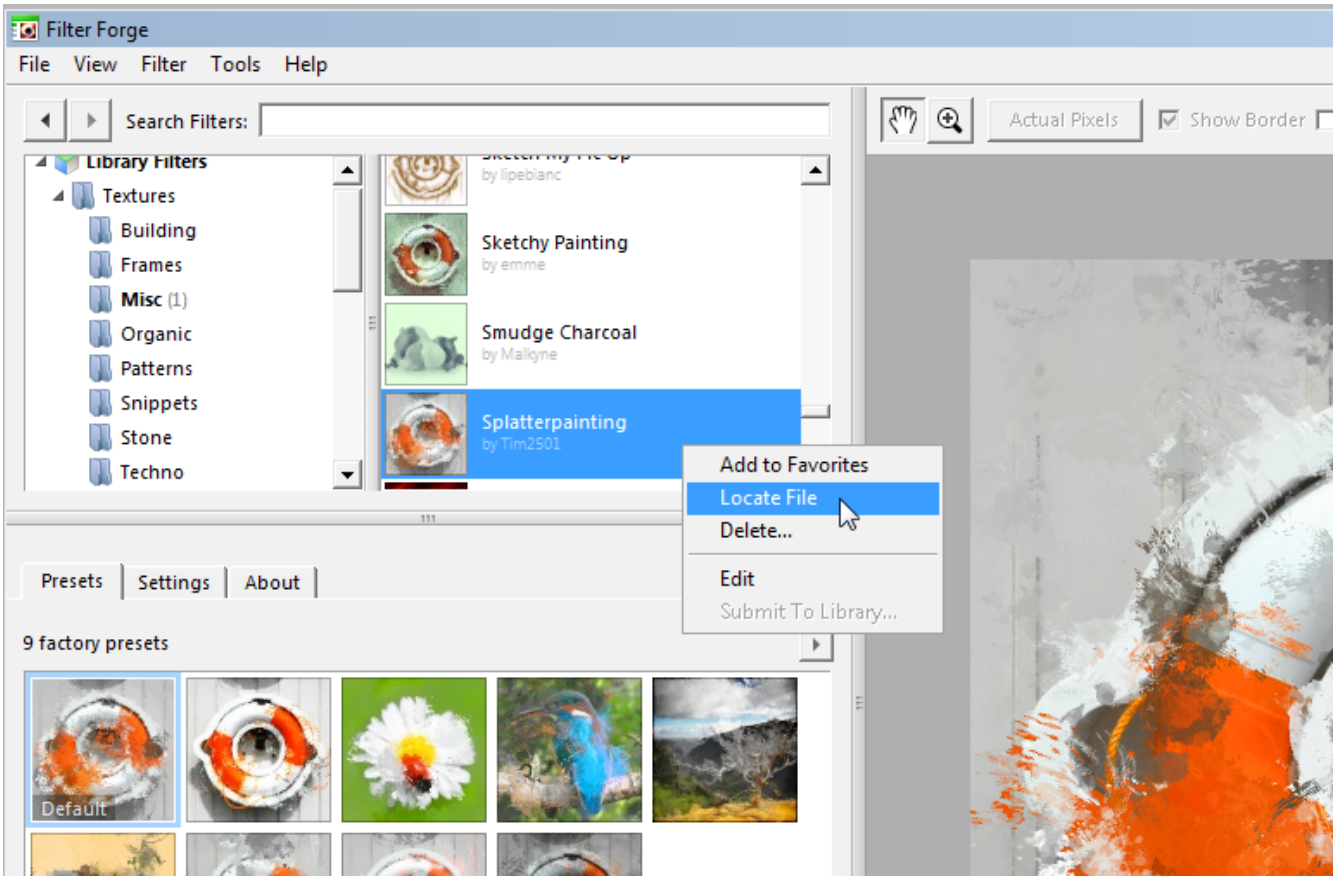
Sort source image into a _rendered subfolder (of the images source folder)

after render, to prevent duplicate renders of it (e.g. in future runs or if you change .ini settings) :

```
Sort Source After Render=True
```

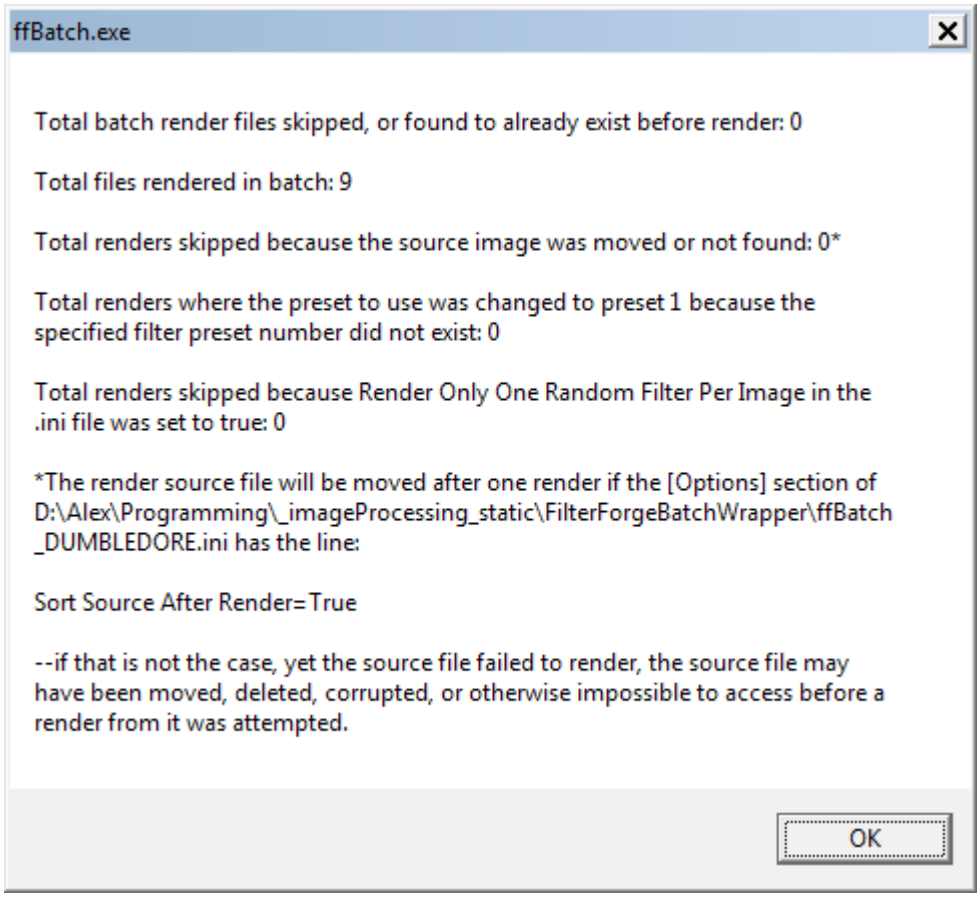
Specify Filters to render with (multiple allowed). Future plans are to make this simpler and automated. For now, to specify the filters to use, follow these steps:

Finding the filter file name of the desired filter in Filter Forge (right-click any filter name, then left-click "Locate File" to find the file name:



—this should open the filter folder in Explorer, selecting the file for the specified filter. Press F2, then CTRL+A, CTRL+C, then ENTER to copy the file name. Open FffilterList.txt in the *ffBatch.exe* folder, then paste the filter file name into that text file and save. When you run *ffBatch.exe*, it will use that filter. Note that it will also search for and use .ffxml filters in the "My Filters" folder for your user profile, and in the same folder as *ffBatch.exe*.

After a run of this program, you will see an information dialog like the following, with render statistics:



License, Contact, Known Issues

See README.md and LICENSE.txt for additional information (including contact info for bug reports).