

File: .\doc\ext\eP-HF-Guide.pdf

## eProof HotFolder Guide

### 1. Introduction

Any file folder on your local or network drive can be declared as "Hot Folder" (HF). Such folder becomes sensitive to files placed into it. Any file "dropped" into Hot Folder, is processed immediately and automatically.

Hot Folders mechanics serves two purposes.

First, it simplifies adding proofs to eProof system. Being in your favorite publishing application, you just need to save your file in the dedicated place, the rest will be done automatically.

Second, HFs are basis of eProof integration in file processing workflow. One application saves or prints files to the dedicated HF, and these files are input for eProof.

### 2. "eProof Folder – Hot Folder" relationship

File system hierarchy: disk – folder - .. – subfolder - file.

eProof system hierarchy: root – folder - .. – subfolder – job - proof.

HF is the link between one of file system folder and one of eProof folder.

All subfolders of file system folder have corresponding subfolders in eProof hierarchy.

### 3. HF Types

There are two supported HF types – In/Out and In-Place.

In/Out HF is represented by pair of folders. Files are placed into input folder, during processing they are moved to output folder and after processing they are moved to eProof internal storage. Note that output folder is only temporary storage for processed files. In/Out HFs are convenient if you generate new proof files and submit these files to eProof.

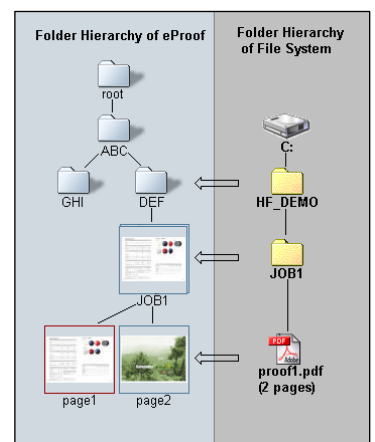
In-place HF has only one (input) folder where files are located (stored) even after their processing by eProof. In-place HF is preferable when you have some directory with files for editing and proofing. After you update a file in this directory, new version of this file is processed immediately.

### 4. Managing of Hot Folders

You must have administrative rights to manage HFs.

Newly installed eProof has one HF created automatically during the installation.

To manage HF's enter **Administration** menu, **Hot Folders** submenu.



### Create Hot Folder

<b>Name</b>	
Hot folder name	Demo
Comment	HF for Doc Example
<b>File input</b>	
Server name	. <a href="#">Edit Servers</a>
Input folder	E:\HF_Demo_In
Hot folder type	in/out
Move files in approved or rejected jobs	<input type="checkbox"/>
Delete empty subfolders	<input type="checkbox"/>
Compound images grouping timeout (seconds)	300
Compound index images grouping timeout (seconds)	1200
Delay before processing (seconds)	10
<b>File destination</b>	
Folder	Root
Move to server	. <a href="#">Edit Servers</a>
Move to folder (on the server)	E:\HF_Demo_Out
Job naming method	Use file name
Enable file processing feedback	<input type="checkbox"/>

[top of table](#)

Submit Reset Cancel

### Hot folder name

Specify unique HF name here. This name will be shown on the HF list page.

### Comment

Except for the name, any additional information can be added to the comment field.

### Server name

Choose one of connected eProof servers. Server must have at least 'list folder contents' access to the directory that will act as a HF and read/write permissions to all its files.

### Input folder

Specify full file path for HF input folder. The path must be visible to the server which will manage HF. It can be local file system path like `x:\my_hf\in` or local network path like `\\my_server\my_hf\in`. If no such folder exists, it will be created by eProof and will never be deleted.

### Hot folder type

Select desired HF type – In/Out or In-place.

### Move files in approved or rejected jobs (reserved option)

### Delete empty subfolders

This option is applicable to In/Out HF only. If the checkbox is marked, any subfolder that became empty after the last file had been moved from it to output folder, will be removed.

This option is useful for HFs with short term folder structure (create job, drop file, proof it and forget), and is not recommended for regular HFs with persistent subfolder structure.

### Compound images grouping timeout (seconds)

Specify a timeout for each subsequent file of compound image (\*) before HF concludes that image was completely copied. In other words, after first file of a potential compound image (e.g. high-resolution monochrome plate) was copied to HF the timeout starts. After second file with similar properties is copied to HF, it is grouped with the first one, and the grouping timeout is reset. Finally, if no file was copied to HF before timeout expired and no subsequent file(s) found, these grouped files are released for processing as one compound image.

Grouping of such index-less images is not predetermined because there is no information on how many files should be expected.

Another problem is that properties of one image file (dimensions, etc.) may interfere with properties of another image file and these files may be mistakenly grouped together. Fortunately, eProof has a mechanism of 'smart bundling'. This mechanism analyses grouped files before processing and splits them into different images by naming criteria. In such case resultant images will be released simultaneously after timeout elapsed.

(\*) Compound image – image that consists of several files (e.g. multi-plate separation files) and has no special index file with a list of all files of an image.

### **Compound index images grouping timeout (seconds)**

This parameter is similar to compound images timeout, but affects the processing of images that have index file with a list of all files of image. Examples of such images are TIFF-IT and DCS. Grouping of such files is determined because HF knows which next files to expect for an image. However some files of an image could be missing. In this case not completely grouped image will be released after timeout elapsed.

For example if you copy DCS image plates without index file, that image will be correctly processed and released after timeout elapsed (of course if all required plates were copied). On the other hand, if all required files were copied it will be released immediately.

### **Delay before processing (seconds)**

During specified period of time eProof will not undertake any operation on file. This parameter is introduced to handle specific behavior of Mac OS Finder when copying files to network share: file is created, unlocked and then locked again for some purposes.

Mac OS resource forks (when they are represented as separated files instead of NTFS streams) are not processed by eProof in any way, but in case it is created in In-Place HF, it will not be deleted. In case of In/Out HF, resource fork will be deleted without copying into output folder.

### **Move to server**

This option is obsolete. Actually, eProof does not allow a file found in HF to be relocated between servers.

### **Move to folder (on the server)**

This option specifies the full path to output folder. This is required parameter of in/out HF. This parameter is required also for In-Place HF, if "Enable file processing feedback" option is ON, to set destination of **Feedback** folder.

Similarly to input folder, this folder must be visible to the server that manages HF.

If eProof stops for some reason (e.g. after a vast period of inactivity), this folder is cleaned in the following way: all empty subfolders except **Feedback** subfolder (if option "Enable file processing feedback" is ON), and output root folder will be deleted.

Optimization hint: for optimal performance output folder should reside on the same drive with input folder and that drive should be of NTFS type or another file-system type which supports hard-linkage of files.

### **Job naming method**

This option defines the name of a job that will be created in eProof for a proof uploaded via HF. Now there are four supported job naming methods:

#### **1. Use last subfolder name method.**

Job, created in eProof will have the name of the subfolder where your files are located after copying to HF input folder. For example, you have two files **1.psd** and **2.eps** in the folder **my\_images** on your PC. After copying this folder to HF input folder the job **my\_images** will be created in eProof. This job will have two proof files – **1.psd** and **2.eps**. If you copy some files straight to the root of input folder, job **nonameN** will be created.

## 2. Use file name method.

Job, created in eProof will have the name of a file without an extension. For example, if you copy two files **1.psd** and **2.eps** to input folder of such HF, two jobs will be created in eProof – 1 and 2).

The purpose of next two job naming methods is to simplify creation of multiple-proof jobs where proofs should have exactly defined page number. To use these naming methods your files should comply with the following requirements: **<page\_number>** – should be a numerical value and **<job\_name>** – should be the same for all uploading files. Both methods add a file to job **<job\_name>** as page with number **<page\_number>**.

## 3. Use file name method (<page\_number>-<job\_name>.ext).

For example, if you copy two files **1-test.psd** and **4-test.psd** to HF input folder, job **test** will be created in eProof. This job will have two proof files named **1-test.psd** and **4-test.psd**. The first page will have the number "1" and the second – "4".

## 4. Use file name method (<job\_name>-<page\_number>.ext).

For example, if you copy two files **test-1.psd** and **test-4.psd** to HF input folder, job **test** will be created in eProof. This job will have two proof files named **test-1.psd** and **test-4.psd**. The first page will have the number "1" and the second – "4".

Please note that that for both 3. and 4. you need to stick to file naming format mentioned, otherwise files will be treated according to job naming method 2.

## Enable file processing feedback

This option requires output folder to be specified even if In-Place HF type is chosen. If this checkbox is marked, eProof will generate simple report file about each file found in input folder, whether it was processed or rejected as invalid (unsupported, corrupt or encrypted). Report file has a unique name based on timestamp and is created in **Feedback** subfolder of output folder.

This option also influences on how invalid files are handled by eProof:

1. If enabled, eProof moves invalid files to HF output folder.
2. If disabled, eProof creates special subfolder **Invalid** in the folder where invalid file was found and moves that file there.

## 5. Functionality details

5.1. In/Out HF. Each file found in input folder will be represented in output folder with the same relative path. File name becomes the name of a subfolder that contains one or more versions of that file (e.g. file **my\_hf\_input\sub1\my\_file.tif** will be stored in output folder as **my\_hf\_output\sub1\my\_file.tif\00000001.tif**).

5.2. In-place HF. Originally, in-place HFs were designed for NTFS drives that support file streaming, however they can be also used with legacy streamless file system drives (FAT, FAT32, etc.). In latter case, for each file placed in input folder (e.g. **my\_file.jpg**) an additional hidden file **my\_file.jpg\_@\$STREAM@\$DIP\_Info** with auxiliary data will be created. To ensure proper HF operation this auxiliary file should not be removed from HF input folder. Otherwise the same file may be processed several times.

5.3. Processing feedback file format. Every file has one text line with comma separated parameters:

**DocumentID, proof\_file\_name, return\_code, return\_code\_description**

Return codes:

- 0 – all pages are processed (Success);
- 1 – processed with warnings (fonts or objects are missing) or error;
- 2 – severe error (Failure).

Proof file name is original file name for single file proof. For compound proof (image) "accumulative" file name is recorded, for example, for 4-files image consisting of **c:\hf\in\myFile\_c.tif**, **c:\hf\in\myFile\_m.tif**, **c:\hf\in\myFile\_y.tif** and **c:\hf\in\myFile\_k.tif**, proof file name will be **c:\hf\in\myFile\_<c,m,y,k>.tif** .