

Workflow Automation Begins with the Workflow Manager

For years PSL users have been automating their workflows by use of the robust programming language on which PSL Page Builder is based, called, strangely enough, PSL. That language, while being quite granular in the typesetting and word processing arenas so necessary for VDP work is, as well, quite robust in the Windows systems work area.

Many firms in the Jet Letter customer base run dozens of jobs a day (50 or 60 is common) and consequently such automation has paid off handsomely in time saving and error reduction. We have often been told that such automation was simply “mandatory” for the growth of their business. There was no other way – for a variety of reasons.

Consequently in 2017 the PSL Automation Control module was introduced. Not only was an easy to use interface provided but core level changes to PSL required for “bullet proof” automation were added.

Understanding automation in PSL is best if one understands the basic Workflow Manager first. So that is discussed in the next few pages and a look at the “Robot” interface is presented further along.

Workflow Manager

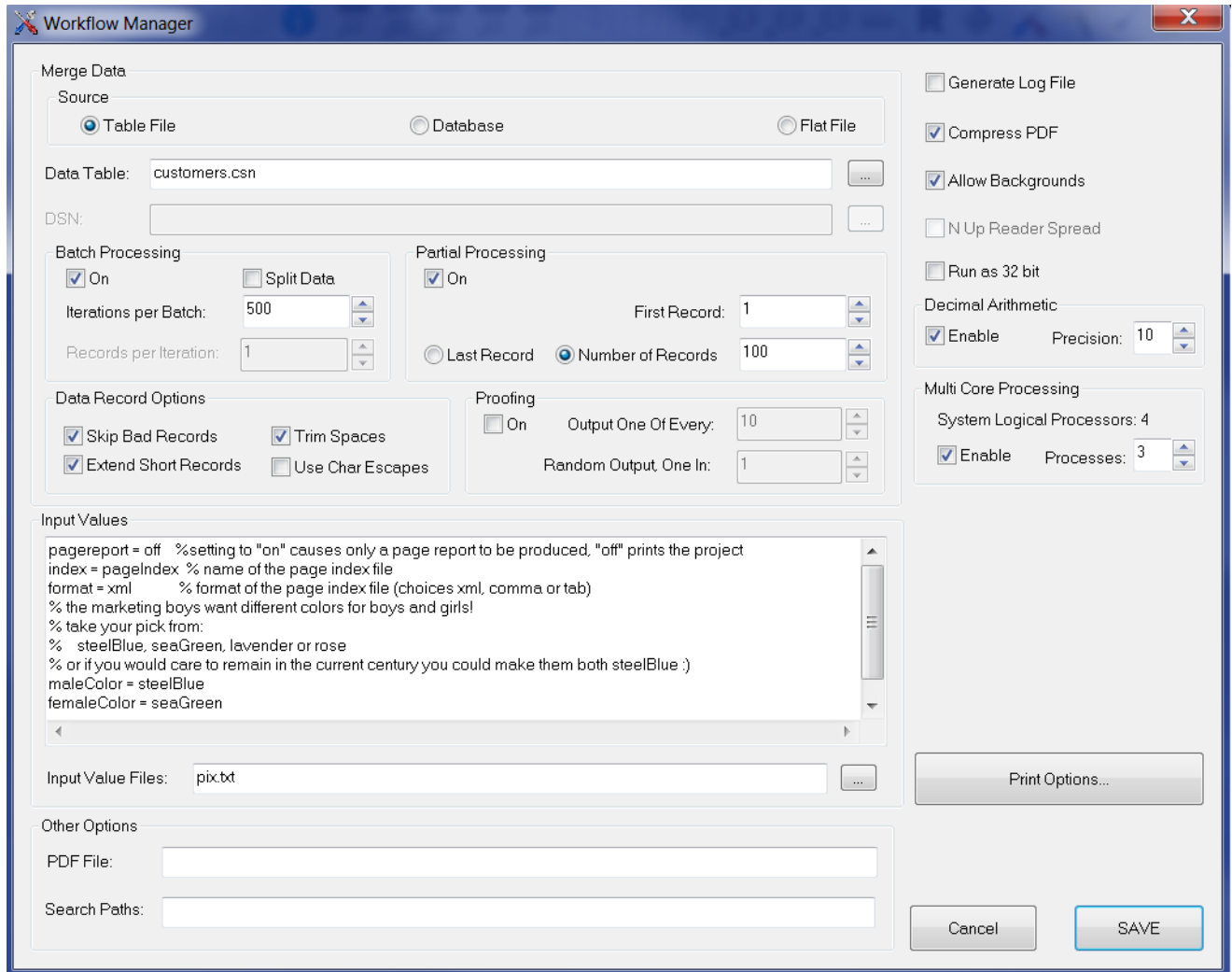
Not surprisingly, the Workflow manager allows users to control input files and output files, and a variety of job options as well as options for batches and printing. The use of it is covered in the general documentation, and is unremarkable as regards automation work, except for the Input Values section covered below. The Source section allows for selection of the Merge file from the potentially many files in the Data folder, by type.

Batch Processing facilitates three major issues;

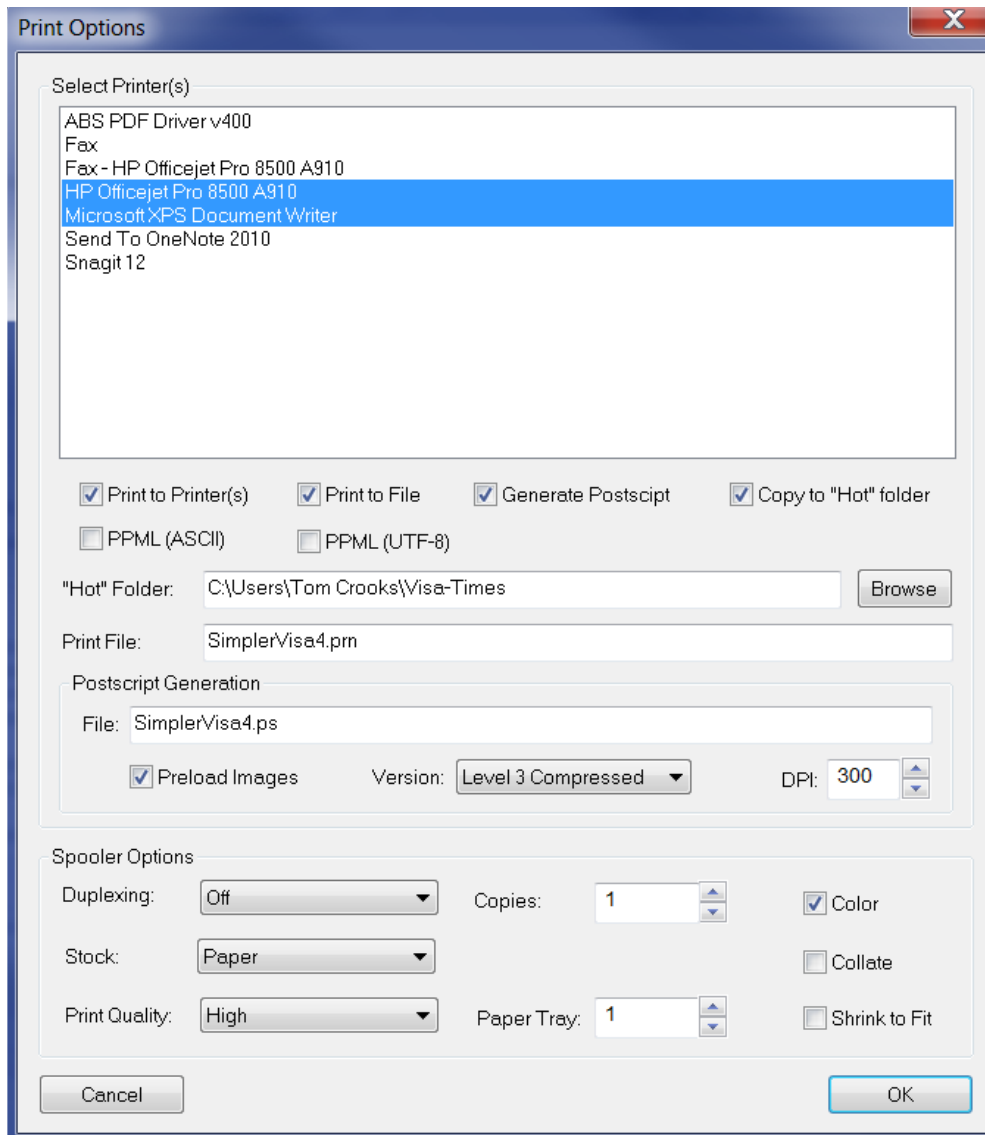
- 1) Allows one to print “round robin” to several printers, and,
- 2) To “work around” slow network connection problems to printers,
- 3) N-Up printing as the batch size affects cut and stack configurations.

Partial Processing is often set smaller than the whole merge file to save time during development. It's also useful for re-prints. Data record options were discussed above and proofing should be obvious.

Of the options on the right, only Multi-Core Processing deserves comment as the best merge speeds are often obtained by enabling one less than the maximum number of processes available (leaving one for Windows to handle disk access). PSL handles such function completely automatically.



With few exceptions, most of the Print Options are as well unremarkable regarding transactional work. One option that is of interest, however, is the Hot Folder section. Modern printers generally have "Hot Folder" setup options that monitor a folder and begin printing as soon as an appropriate file (such as PDF) appears in that folder. Under automation procedures this may be altered from run to run. As well, the name of the print file is subject change with each run.



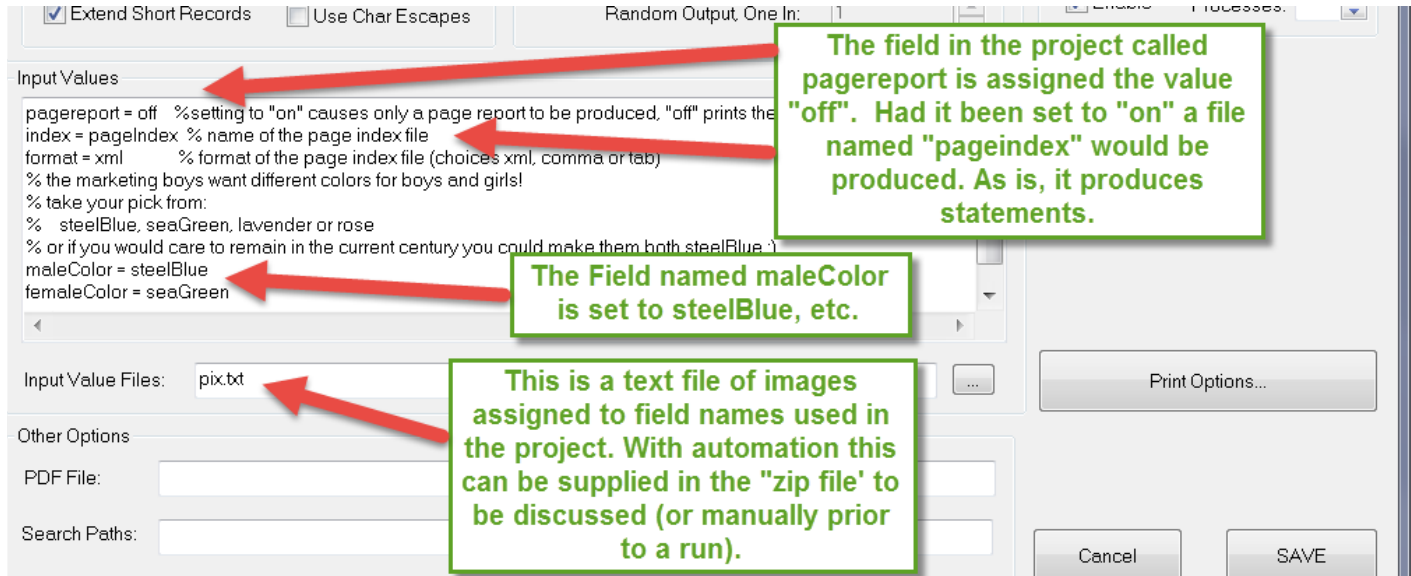
Print Options

Input Values

The portion of the Workflow Manager of particular interest in automation work is the Input Values section. As indicated below a user can assign values to field names used in the project in this section. A simple advantage to this is that changes from run to run do not involve opening the development system – just the Workflow Manager where it's pretty difficult to “destroy” the project.

As one may see in the visa example, simply changing the “pagereport” value from “yes” to “no” completely changes the output of the project – from a simple file necessary to

accompany the presort to an actual run of statements using that file with the presorted merge file. Use of these values is particularly relevant to automation.



The Input Value Section of the Workflow Manager

Workflow Automation

Until quite recently, firms had automated production of VDP projects using the PSL programming language. With the use of Workflow Automation now that number is increasing. Automation is the key to profitability for many reasons;

- Customers love it – they are in control
- It saves money for all concerned
- Human errors are reduced – if not eliminated
- Response can be quite quick – such as middle of the night
- All manner of complexities can be dealt with
- Changes from run to run can be quite robust just based on the zip file submitted

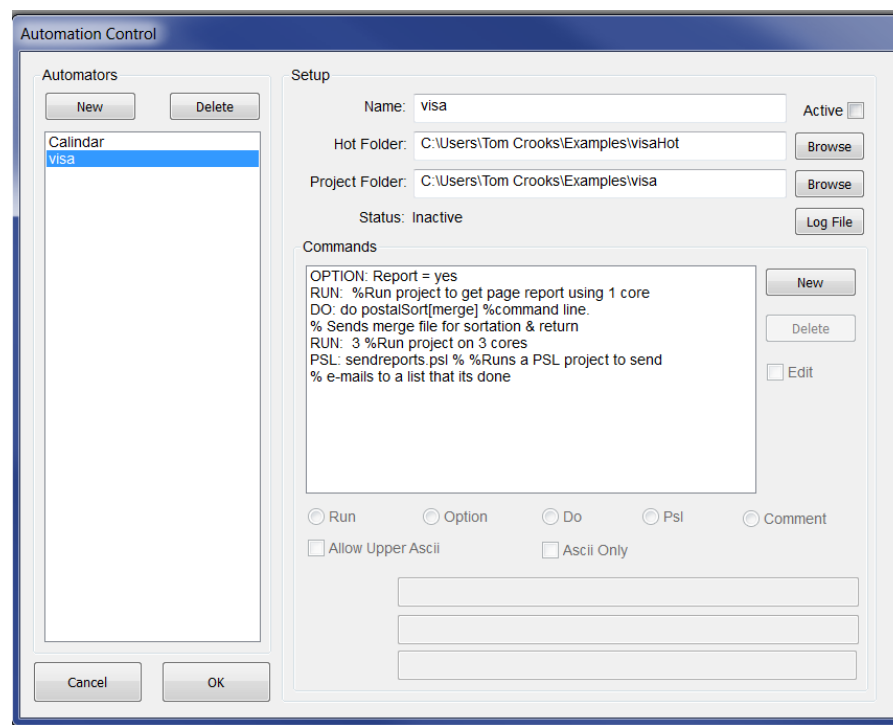
Although customers have used programming to implement automated systems in the past, automated systems can be complex. Issues such as concurrency, shut down-restart, and time outs were difficult if not impossible to deal with. **We consider the automation issue of overriding importance.** As a result, we have introduced an automatic automation system, the PSL “Robot”.

In a general case the “Robot” for automation allows the PSL user, or service provider, to define a sequence of commands that will execute when a “zip” file is dropped into a “hot folder” assigned to a particular project. These commands can run the project, change

its attributes and run any arbitrary program (such as postal sortation, a page count report to go with sortation, a separate file/report summarizing the transactions just performed). A typical example based on the visa project discussed above, might be:

- Submit a change in Resources
- Select which printer(s) to use
- Prepare supporting files and send them where desired
- Monitor various hot folders
- Provide selected job options for the “current” run
- Tell people (or programs) what will happen or has happened and when?
- Run a sequence of events;
 - Run page count program (such as in visa by saying “yes”)
 - Run presort (local or in the cloud)
 - Trigger on return of file and run project or command line sequence
 - Send output to selected folders
 - Run a PSL program to send results where needed or an e-mail to the manager
 - Run a program in the language of your choice to do what it does best (The Do command)

The interface looks like this:



To review, users may build a list of commands to be run for any number of automated projects. Then to cause the job to run a zip file may be dropped into the project's hot folder. The zip file typically includes reference to a new merge file of course, but a user may also override any job parameters, set new Input Values, and replace resources.

It is important to note the distinction between the end user (your customer) and the service provider (you). The normal assumption is that the "zip" file is dropped into the hot folder by the end user (e.g. via ftp). The contents of this file override the original files in the project so that the end user can change resources (such as fonts and images) as well as data as required. However, the "robot" will not process any files that are deemed security risks, or that may change the functionality of the project. The service provider, if desired, can design the project to extend control to the end user of whatever aspects needed. **It is the service provider that remains in full control at all times.**