Input Folder Scripting

April 29, 2010

Introduction

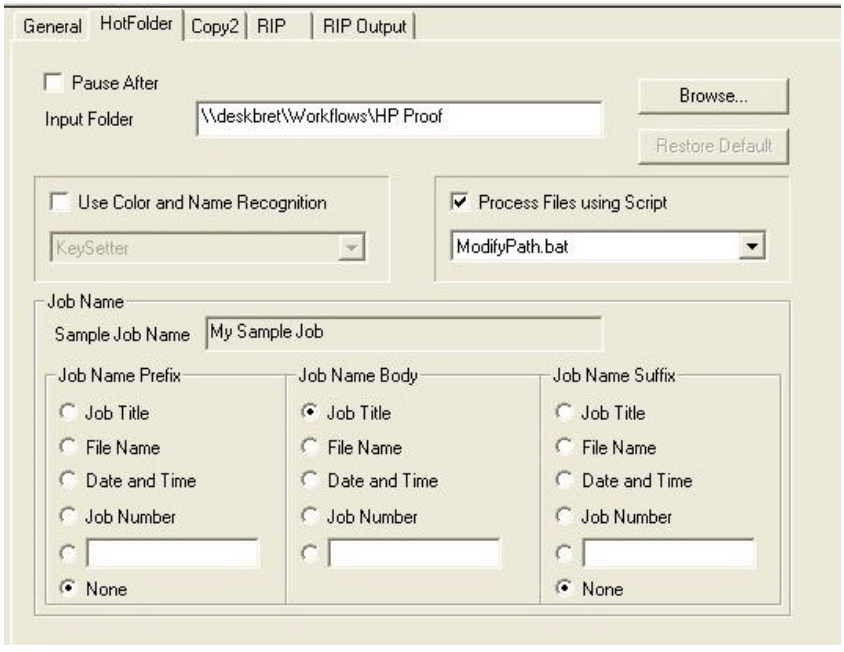
Navigator hot folders may be scripted to perform desired actions on incoming files. Possible uses for this ability include integrating third party applications, custom job sorting, or intelligent decision making on how to process jobs.

Useful scripts will be written and shipped via our web site.

A sample script is included with this documentation. Several sample scripts ship with the software.

Instructions

The script will require whatever scripting language you use to be running on the server. It does not require any scripting language in particular and can use batch files or Perl or any scripting language you prefer. In the "Hotfolder" configuration tab, check the "Process Files using Script" box to enable scripting. Choose a script from the list. To add more scripts to the list use the "Install Script" menu at the Server.
If your script has support files they must be copied to the Config/Scripts folder inside the Navigator Server install folder.



This script can run some DOS commands or run some other scripting language. If the script executes, then a job simply disappears into that script. You might have a workflow that never shows any jobs; but simply scripts them. If you wish the resulting file to re-enter

the workflow, have your script deliver it back to the hotfolder. The system will recognize that it has been scripted and then pass it to the next stage of the workflow for processing. Depending on the behavior of the script the input file will either:

1. Be moved out of the system, in which case the system will move on to the next file in the hot folder.
2. Be returned to the hotfolder, in which case the file will be passed to the next workflow action after the hotfolder
3. Be renamed in the script, in which case the renamed file should be moved back into the hotfolder for continued processing with the next workflow action.

Sample Script Details

We have included a Perl script that sorts jobs based on their physical size for intelligent routing to different sized CTP devices.

In order to run this script your Navigator server should have a Perl environment installed. Distributions of Perl are widely available.
You could look here:
http://www.activestate.com/activeperl/
and here:
http://www.perl.com/download.csp

The sample script reads the bounding box comment of a postscript or PDF file and copies jobs that are 12" wide to one hot folder and any others to another hot folder.
The Perl script, C2wf.cl", can be customized. See the final section for the actual Perl script. (important lines in blue).

Sample Perl script, "C2wf.pl"

```
#
# ToDo:
# Change this to that path where your workflows are, must use double backslashes:
#

$WFStem = "D:\\XiFlow\\WORKFLOWS\\";

#
# ToDo:
# set your default workflow here, where the file goes if your test
# conditions aren't met, e.g. BoundingBox not found, or width not matched, etc.
#

$WorkflowName = "Error";


$FirstLine = 1;

# Read in one line at a time
while(<PSFILEIN>)
{
    if($FirstLine)
    {
        $FirstLine = 0;
        # Check PS magic bytes here....
        if(!/^%!PS/)
        {
            printf "Sorry, didn't find PS signature, outta here...";
            last;
        }
    }

    print $_ if $dbg;
    # Search for the BB comment and extract the 4 values.
    if(/^%%BoundingBox: (\d+) (\d+) (\d+) (\d+)/)
    {
        # matched the %%BoundingBox comment the four parameters llx, lly, urx and
ury are in the
        # Perl variables: $1-$4.
        print "Found bounding box line. Values are: $1 $2 $3 $4\n" if $dbg;

        # This is where you would do your thing. In this case I simply direct
plates 12" wide
        # to a workflow named 12Inches. All other files go to the 10Inches
workflow.

        if($3 == 12 * 72)
        {
            $WorkflowName = "12Inches";
        }
        else
        {
            $WorkflowName = "10Inches";
        }
        # And exit
        last;
    }
}
```