



**NAVIGATOR**

OPTIONS CHAPTER

---

**TIFF GROUPING**



**YOUR BUSINESS. OUR DRIVE.**

---

---

## INTRODUCTION

Most Navigator RIPs have the ability to interpret PostScript and PDF files and convert them into 1-bit TIFF format for consumption by other devices such as platesetters and proofing devices. In the case of platesetters, the RIP is most often set up to separate the jobs into individual process (CMYK) and spot color plates in preparation for mounting on a press.

When the RIP creates these TIFF files, they are usually named according to the color they represent but also carry the basic name of the original file. For example, a four-color job titled, Testjob.pdf might - after processing through the RIP - result in four TIFF files named Testjob(C).pdf.TIF, TestJob(M).pdf.TIF, Testjob(Y).pdf.TIF, and Testjob(K).pdf.TIF.

At this point, any system to which these TIFF files are sent (including RIP Manager) will recognize them – perhaps incorrectly - as four separate and un-related jobs.

With the addition of the TIFF Grouping option in Navigator GPS, operators can instruct the system to examine the names of TIFF files being introduced into the workflows and recognize TIFF files that belong together. This makes it possible to use Navigator GPS for:

- Integrating external proofing solutions that require 1-bit TIFF input
- Load balancing Harlequin RIPs used as TIFF catchers for high-res output devices
- Integrating any 3<sup>rd</sup> party TIFF-based applications into the workflow

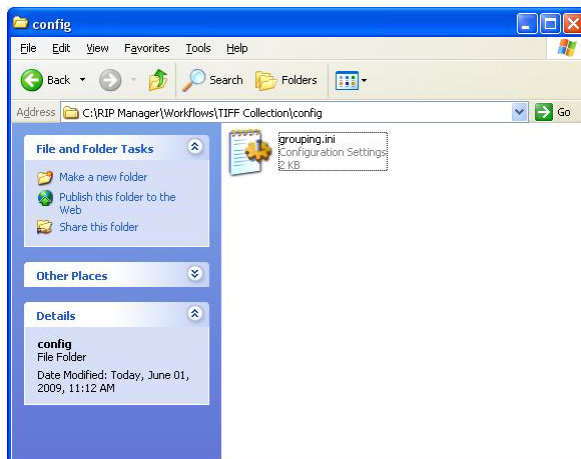
---

## CONFIGURATION

Configuring RIP Manager to group TIFF files requires only a few steps. Begin by locating a file labeled *grouping.ini* in the *Extras* folder of the installation CD. Place a copy of this file into a folder labeled *config* within the hot folder used by the workflow. For example, if a workflow named *TIFF Collection* has been designated to group TIFF files, the hot folder for that workflow might be found in *C:\RIP MANAGER\WORKFLOWS\TIFF Collection*. Properly set up, that folder would contain a folder labeled *config* and the *grouping.ini* file would reside there, as shown in Figure 1.

---

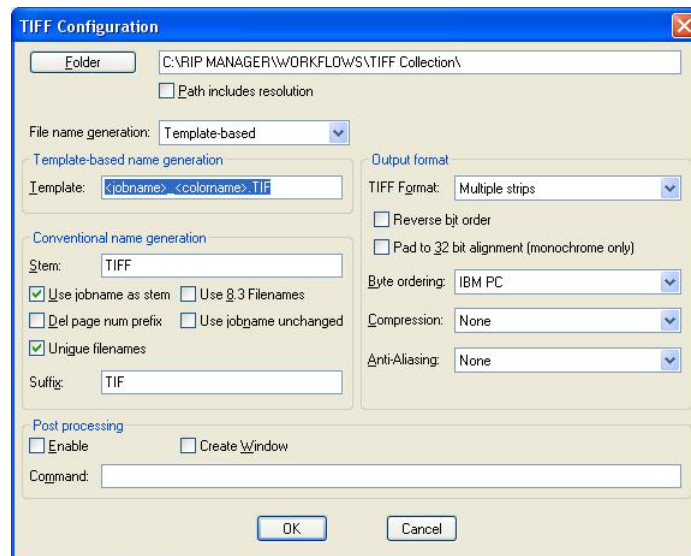
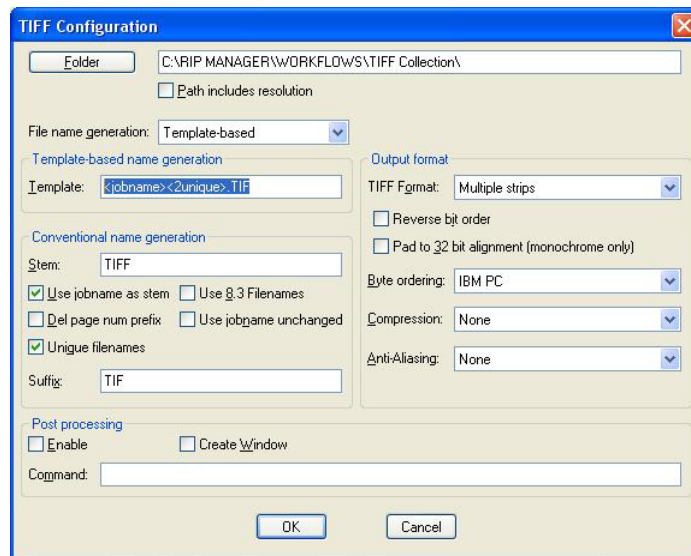
FIGURE 1: CONFIG FOLDER AND GROUPING.INI FILE



The *grouping.ini* file contains the programming used by the system to determine the proper grouping of files. It uses “regular expression parsing” to compare TIFF names inside a folder and recognize which files belong together. Consequently, *grouping.ini* is editable so that it can be adapted to whatever file naming standards are being used. (An example of the default *grouping.ini* file has been reproduced in the last section of this document.)

However, if left in its default state it will automatically group TIFF files based on the simple TIFF naming setup shown in Figure 2. To get started with this quick set up, create a new Page Setup on the RIP and select TIFF for the output device. Click the Configure Device Button and modify the parameters to match those shown in one of the examples in Figure 2. Either one will work with no editing of the *groupings.ini* file.

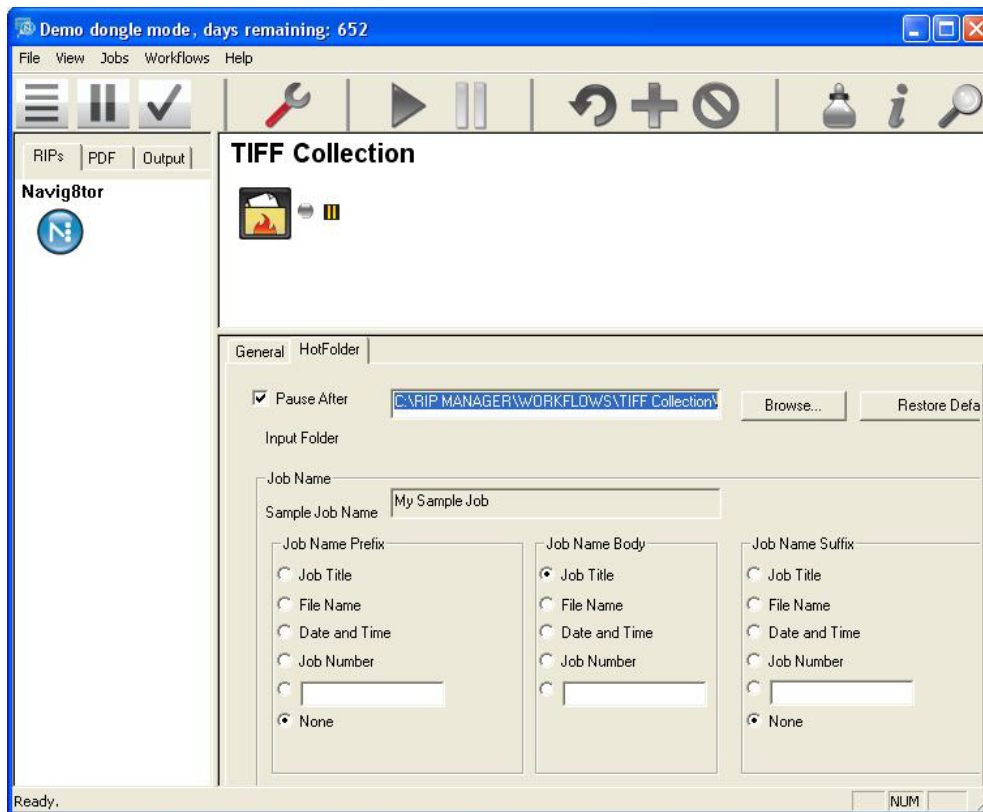
FIGURE 2: TIFF NAMING IN THE RIP



Note the location specified for the RIP to write the TIFF files is the same as in the example stated earlier: C:\RIP MANAGER\WORKFLOWS\TIFF Collection. This should be modified to match whatever the planned workflow location is. Also, TIFF file name generation is set to “Template based.” See the Navigator RIP User guide for additional TIFF naming template help.

Next, prepare a simple workflow like the one shown in Figure 3. Note the location of the hot folder matches the location designated in the RIP’s TIFF configuration.

FIGURE 3: SIMPLE WORKFLOW



Print a job to the RIP with the TIFF Page Setup designated. With the RIP properly configured, the test file should produce separate CMYK TIFF files that show up in the Tiff Collection workflow as a single group. An example is shown in Figure 4. This is the CORRECT result.

If the system has not been properly configured, jobs will appear as those shown in Figure 5. This is the INCORRECT result.

FIGURE 4: CORRECT RESULT WITH TIFF COLLECTION ACTIVE

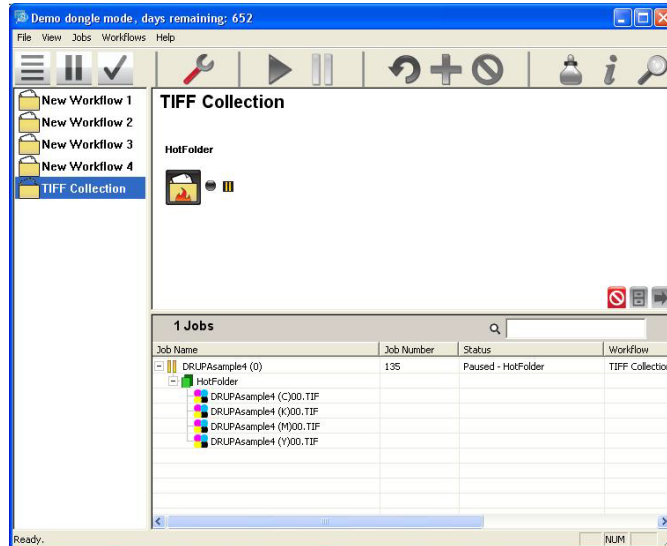
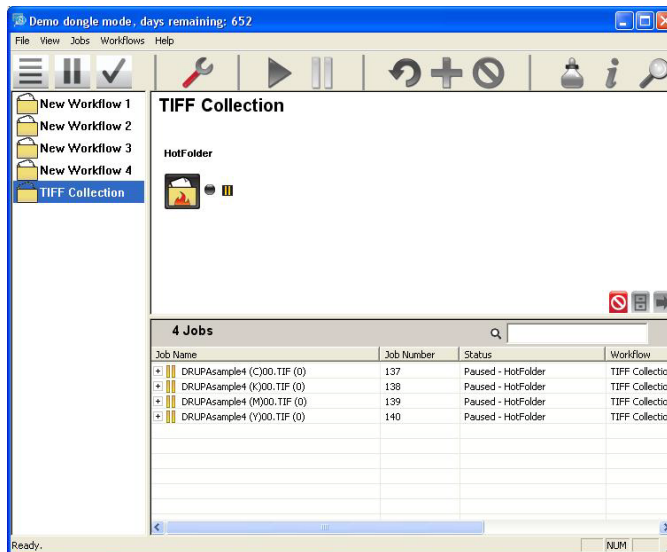


FIGURE 5: INCORRECT RESULT - TIFF COLLECTION NOT ACTIVE



---

## EDITING THE GROUPING.INI FILE

As mentioned earlier, the system uses “regular expression parsing” to compare TIFF names and recognize which files belong together. The file “*grouping.ini*” controls the regular expression that is used as well as other parameters like timeouts, hard counts of files in group, etc.

While editing the *grouping.ini* file, there are two keys for configuration that change how a file name is recognized. Locate these entries in the file:

**BasePartIndex=1**

**RegEx=[\(\)\.]+**

The BasePartIndex tells which part, after examining the file name, on which to match. The *regular expression* (RegEX) is contained within square brackets followed by a plus sign. Some characters in the string are also used in other parts of programming so the system has to be told that these are actual characters to search for and not part of the program. This is done by inserting a backslash before that character.

For example, the regular expression being used here (what we are searching for in the file name) is an open parenthesis, a closed parenthesis and a period. Since the parenthesis and period characters are also used in programming, a backslash is inserted prior to each. Hence, the example shows `[\(\)\.]+`

When the system sees a parenthesis in a file name, it has been told that anything before the parenthesis is the job name. Therefore, a file called “TestJob (c).tif” will be recognized as having the name “TestJob”, and “TestJob (m).tif” will be recognized as being part of that same job.

Depending on your source for TIFF files you may want to edit the regular expression to add an underscore or a dash. Both will require a backslash. The contents of the default *groupings.ini* file are reproduced on the following page.

---

## GROUPINGS.INI FILE CONTENTS

```
[SettingsInfo]

; Section Index that determines the 'base' name for the group.
; Valid values are: 0 - number of sections in name
; Default is 2
BasePartIndex=1

; Section Index that determines the page number for the file.
; Valid values are: 0 - number of sections in name
; Default is 4
OrdinalPartIndex=3

; Section Index that determines the separation colour for the file.
; Valid values are: 0 - number of sections in name
; Default is 5
ColourPartIndex=4

; String containing the delimiters used to break a file name into sections
; Default is ( -_.)
Delimiters=( -_.)

; Regular expression string used to break file name into sections
; Default is [\\(-\\.\\) _]+
Regex=[\\(-\\.\\) _]+

; FileCount trigger.
; Closes active group if a specific number of files has been added to the group
; Valid values are 0 (disabled) or 1 (enabled).
; Default is 0
FileCountTrigger=0

; Trigger value for 'FileCount' trigger
; Default is 4
FileCount=4

; Timeout trigger.
; Always active. This closes the group after a number of seconds of inactivity.
; Adding a file to a group resets the timeout.
; Default is 30
TimeoutSeconds=10

; SerialMode trigger.
; Flag to control single or multiple active groups in a folder.
; Valid values are 0 (Multiple groups allowed) or 1 (Only one group at a time).
; Default is 1
SerialMode=1

; MatchFail trigger
; Always active.
; In SerialMode, trigger closes the current group if the match fails.
; A new group is then created for the new 'grouping' criteria.
```