# File Ninja (Admin)

File Ninja helps you manage your files and automate tasks that are too repetitive to do manually. The Admin version includes everything from the Lite version, plus more features.

# Opening File Ninja

* Download and open the File Ninja p7m file.
* OR download XlsxWriter (pip install XlsxWriter) and run the Python code directly.

# Explanation of Basic Elements

## Lite

| Element | Description |
| --- | --- |
| Browse to Select | Browse to select the directory to perform operations on. |
| Browse to Exclude | Select subdirectories of the main directory to exclude from operation. Hidden directories — and their subdirectories — are automatically excluded. To remove a previously added excluded directory, either double-click or right click it. (If *Include Subdirectories* is off, this feature does nothing.) |
| Find procedures | Flags any errant files based on the selection. |
| Include Subdirectories | If on, File Ninja will peruse through all subdirectories as well. Otherwise, will only traverse the currently selected directory. Related: *Browse to Exclude*. |
| Add Recommendations~ | If on, procedures that end in “~” will show helpful recommendations. Red highlight indicates a stronger warning: those files should perhaps be deleted. Yellow highlight indicates a weaker warning: those files should be looked at. |
| README | Open the README docx file if it is available. |
| Execute | Commence execution. |
| Results | Open results directory, containing Excel files of previous executions. File names follow the convention “SelectedDirectoryName-YY-mm-DD-HH-MM-SS.xlsx”. |

## Admin

| Element | Description |
| --- | --- |
| Fix procedures | If *Allow Modify* is off, flags any errant files based on the selection, showing would-be fixes. If *Allow Modify* is on, executes said fixes. |
| Parameter# | Fix procedures that end in *#* require an argument. Input their arguments here. If running multiple Fix procedures, separate arguments using “/”, in order from top to bottom of the list of Fix procedures selected.  Example 1: If running “Replace Character (DIR)#” and “Replace Character (FILE)#”, you may input (without the quotes) “&>and / @>at”.  NOTE: All Find procedures with arguments have usable a default value. To use its default value, either input nothing in the parameter field, or separate by slashes and leave its value blank.  Example 2: If running "Old File Find#~", "Empty Directory#~", and "Replace Character (FILE)#", you may input (without the quotes) "/ / &>and". |
| Allow Modify | If on, executes all changes irreversibly. Can not run multiple modifier Fix procedures simultaneously. |
| Include Hidden Files | If on, Find procedures will include hidden files.  NOTE: Fix procedures will always ignore hidden files. |
| Save Settings | Save current settings to a JSON file. After saving settings, the user is asked if they would like to create a corresponding batch file. Execute these batch files to automatically run File-Ninja-Control using the JSON's settings.  NOTE: The JSON and corresponding batch file must have the same name and be in the same directory.  NOTE: File-Ninja-Control must not be renamed and must be in the same location as File-Ninja-Admin’s original location. If the batch file suddenly stops working, load the JSON's settings and re-save it, overwriting the old batch file.  NOTE: Even if set to true in the JSON, *Allow Modify* will always be set to false when running the program via a batch file. |
| Load Settings | Load settings from a JSON file. |

# Find Procedures

| Name | Description |
| --- | --- |
| List All Files | Lists all files within their respective directories. |
| List All Files (Owner) | Lists all files within their respective directories. Includes an owner column and a last modified column. Owner is displayed in the format "DOMAIN\NAME (SID\_TYPE)". |
| Identical File | Flags duplicate files. Error count is incremented for each duplicate found. Most GIS extensions are omitted from output, except for .shp and .dbf. (Includes owner and last modified columns.)  Example 1: if a group of 5 identical files are found, the error count is incremented by 4. |
| Large File Size# | Details a summary of each file extension found, as well as listing all items and their respective sizes, grouped by extension. Any file over 100MB count as an error. |
| Old File#~ | Flags any file that has not been accessed in over 1095 days (3 years). (Includes owner and last modified columns.)  NOTE: Windows has a glitch regarding the “last accessed” attribute for a file, in that a file may be considered accessed even if it was not explicitly opened. Therefore, run this Find procedure first and foremost before perusing through your files. |
| Empty Directory | Flags any directory that holds 0 folders and 0 files within. |
| Empty File~ | Flags any file that is 0 bytes. (Includes owner and last modified columns.)  NOTE: Some file types may be empty but are not 0 bytes, such as most Microsoft files. For instance, an empty excel file is still roughly 6kb in size. |
| Space Error (DIR) | Flags directory names with spaces. |
| Space Error (FILE) | Flags file names with spaces. |
| Bad Character (DIR) | Flags directory names with bad characters. A bad character is any character that is either not alphanumeric, nor a hyphen (-). |
| Bad Character (FILE) | Flags file names with bad characters. |
| Exceed Character Limit | Flags file paths over 200 characters. These files are not backed up. |

# Fix Procedures

| Name | Description |
| --- | --- |
| Delete Empty File~ | Deletes any file that is 0 bytes in size. Read “Empty File Error” find procedure for more information. (Includes owner and last modified columns.) |
| Replace Space w Hyphen (DIR) | Same as the file version, except for directory names. |
| Replace Space w Hyphen (FILE) | Replaces all instances of spaces within file names with a hyphen and fixes bad hyphen usage.  Example 1: “Engagement Tracker.txt” -> “Engagement-Tracker.txt”.  Example 2: “- Engagement - - Tracker -.txt” -> “Engagement-Tracker.txt”.  Example 3: “Engagement–Tracker.txt” -> “Engagement-Tracker.txt”. |
| Replace Character (DIR)# | Same as the file version, except for directory names. |
| Replace Character (FILE)# | For file names, replaces substring with another substring, using a “>” as separator between the replacer and replace pair, and “\*” as a separator between pairs.  WARNING: If running multiple pairs of arguments, one pair’s replacer can undo the work of an earlier pair’s replacee (not a typo!); order can matter if not careful. Always double-check your output if unsure.  Example 1: With the argument set to “& > -and-”, all instances of “&” will be replaced with ”-and-”.  Example 2: With the argument set to “@>-at- \* &-and-”, all instances of “@” will become “-at-”, and all instances of “&” will become “-and-”. |

# Hints

## Scheduling Executions with Task Scheduler

NOTE: Ensure the exe file exists, not just the p7m file.

* Open **Task Scheduler**.
* Select **Create Basic Task…**.
* Create a name.
* Specify a trigger.
  + Specify trigger details further if applicable. These can always be changed later.
* Select **Start a program**.
* Specify program details.
  + **Program/script** = File-Ninja-Admin.exe
  + **Add arguments** = The location of the JSON settings you want to run. (ex: "C:\Users\FirstName LastName\routineRunSettings.json"). **MUST** be in quotes.
  + **Start in** = The directory in which File-Ninja-Admin.exe resides. (ex: C:\Users\FirstName LastName\). Must **NOT** be in quotes.
* Press **Finish**.
* Right-click the newly created task, then press **Run** to ensure you set your program details correctly.

## Miscellaneous

* Middle-click anywhere in the window to alternate between light and dark mode.
* ctrl+w to close the window.