File Manipulator - requirements specification

# Introduction

The utility application dedicated to the MS Windows platform, used to mass management the names, attributes and content of files in selected directories, as well as to monitor changes such as adding, deleting, renaming a file or changing the attributes and content of a file/folder.

# Product characteristics

The program is addressed to people who are forced to work on large sets of files, where sometimes you need to automate certain management processes. So they can be software engineers, photographers, image and audio editors. The target environment is MS Windows and FAT, FAT32, exFAT, NTFS, ReFS file systems, as well as network file systems like SMB.  
You plan to use MS Visual Studio Community for non-commercial use and WPF.NET using C# technology.  
Any data needed to build the program will be stored in a public Git repository.  
The project will be managed according to the Scrum methodology on the public MS Azure DevOps boards.  
Version management is also introduced in accordance with the Semantic Versioning v.2.0.0[[1]](#footnote-1).  
The application will use open source libraries with binary files available on NuGet platform, which will be included in the program.

# Requirements:

## Functional:

* Create as many Watcher (change tracking) and Manipulator (file/directory management) tasks as you want.
* Write the task body to the preset that will be stored in the task library and load the task from the library.
* Write/read task library to XML/JSON file.
* Block manipulation in the folders listed below if the application does not have administrative privileges:
  + System folder (mostly C:\Windows)
  + Application Data Folder (C:\ProgramData, C:\Program Files (x86))
  + Folder containing profiles of all users (C:\Users)
* If the selected files and folders are unavailable or if you are not able to save them, the job will stop.

## User:

* Clear interface in the tabbed form (similarity to a web browser)
* Aid and hint system – in particular as regards irreversible information

## Non-functional:

* Stability and resistance to I/O problems
* Creating code based on the current principles of so-called purity (S.O.L.I.D. and using well-known design and architectural patterns like MVVM) to improve readability and accelerate further development

# Data collection and processing

The Software does not collect or send any data about you. It does not require network access (the exception is read/write the job to network media).

1. https://semver.org [↑](#footnote-ref-1)