# Teilnehmer/innen des Teams:

|  |  |
| --- | --- |
| Klasse:  AP17C | Team:  sudokuschwer.com |

# Anforderungsdefinition (Meilenstein A)

|  |  |
| --- | --- |
| Filemanager | |
| **Fachlicher Inhalt:**  (Allgemeine Beschreibung) | Nutzen: Mit dem Programm sollen Aufräumarbeiten automatisiert werden**Automation:**  Das Programm verwaltet Ordner, deren Struktur sowie auch Inhalt  **Details:**   * Konfiguration: Der User kann Konfigurieren, welcher Ordner überwacht werden soll sowie auch wo und wie die Daten sortiert werden sollen. * Integration: ------------------------------------------- * Administration: Unser Projekt wird via dem GUI administriert. * Sicherheitsaspekte: Die Berechtigungen werden via Windows User verwaltet. Daher müssen keine weiteren Sicherheitsaspekte beachtet werden.   **Erkenntnisse aus der Machbarkeitsabklärung in Windows Powershell:**  Filesystemwatcher ist vorhanden. Dieser überprüft Änderungen in einem Ordner. Ebenfalls kann man via Powershell Files auslesen. Das GUI ist ebenfalls mit diversen Tools realisierbar. |
| **MUSS**  **Kriterien:**  (Konkrete Features, die umzusetzen sind) | **Folgende Features sollen implementiert werden, um einen produktiven Ablauf sicherzustellen:**   * Skript starten / beenden via GUI * Config via GUI erstellen / bearbeiten * Skript läuft immer * Freie Auswahl des zu überwachenden Ordners * Sortierverhalten ist frei wählbar |

|  |  |
| --- | --- |
| **KANN**  **Kriterien:**  (Konkrete Features, die optional sind) | **Folgende Features können zusätzlich implementiert werden: (Varianten, Kreativität)**   * Ablageordner von verschiedenen Dateitypen analysieren |

## Planung Meilensteine (LB1 / LB2)

|  |  |  |  |
| --- | --- | --- | --- |
| *MS* | *Tätigkeit / Abgabe* | *Soll-Datum* | *Ist-Datum* |
| A | ProjektstartTeam Bildung, Kollaborationsplattform, GitHub Repos \*, LehrerzugangWahl / Ausarbeitung der Anforderungsdefinition Abnahme Anforderungsdefinition durch Lehrperson | 13.03.2020 | 09.03.2020 |
| B | Teamaufgabe 1:Abgabe: Lösungsdesign  (Funktionsmodell / GUI / PAP / Storyboard) | 20.03.2020 | 20.03.2020 |
| B2 | Teamaufgabe 2: (Nur LB2)Abgabe: Testvorschrift und Testfälle |  |  |
| C | Einzelaufgabe 2 (LB1) / 3 (LB2):Abgabe Programmcode und DokumentationFachgespräch Projektabnahme |  |  |
| C2 | Einzelaufgabe 4: (Nur LB2)Abgabe: Ausgefüllter Systemtest |  |  |

\*) Öffentliche GitHub-URLs im Ablageordner auf dem BSCW ablegen! (pro Team)

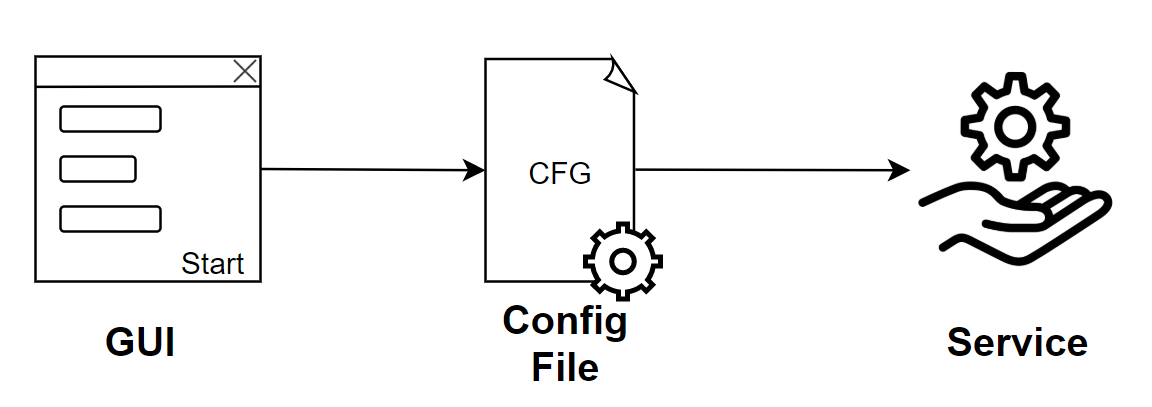
Namenskonvention URL: **M122\_Klasse\_Thema\_Name\_Name**

# Lösungsdesign (Meilenstein B: Teamaufgabe 1)

Anhand der Analyse wurde folgendes Lösungsdesign entworfen:

## Schematische Darstellung der Funktionalität, sog. Funktionsmodell

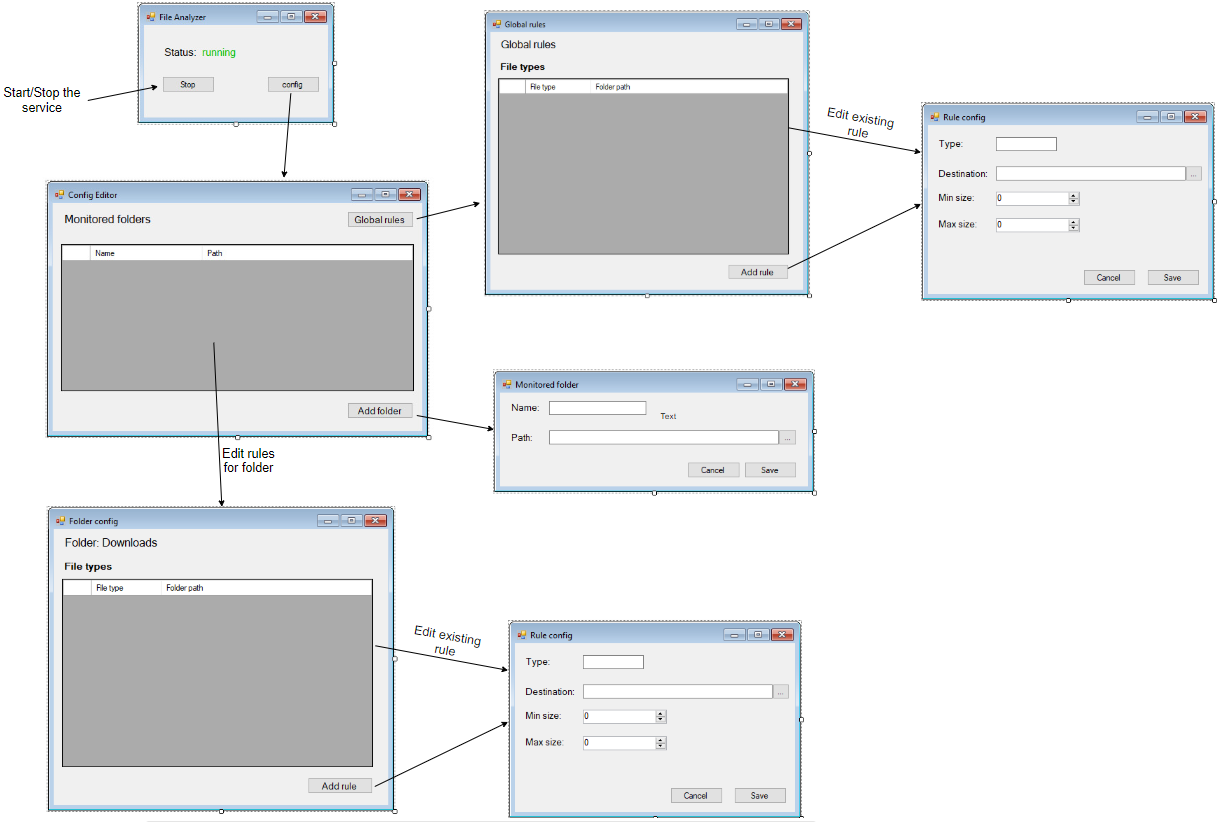
Via GUI kann unser Config-File bearbeitet werden. Dieses wird dann als Parameter an unseren Service weitergeleitet.





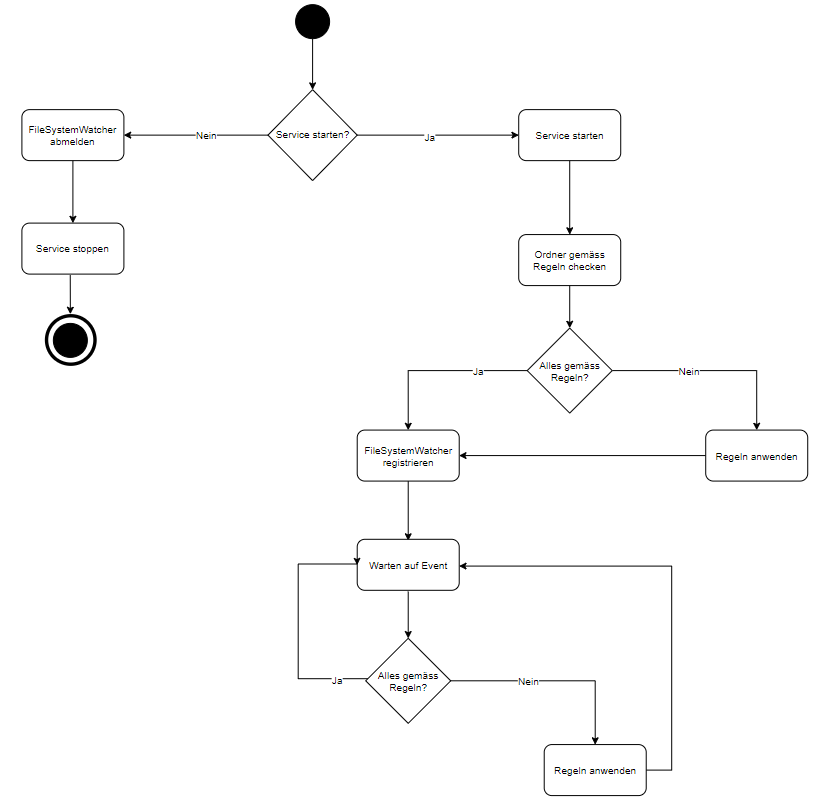
## Graphische Benutzer Schnittstelle (GUI) zur Konfiguration des Ablaufs

Mit unserem zentralen Einstiegspunkt kann unser Service gestartet sowie auch gestoppt werden. Ebenfalls kann man von diesem Einstiegspunkt auch das Config-File öffnen und bearbeiten.



## Ablauf der Automation

Aus Benutzersicht ist folgender Ablauf des Programms zu erwarten:





# Testvorschrift (LB2 Meilenstein B2: Teamaufgabe 2)

Testbeschrieb und vorbereitetes Testprotokoll siehe Dokument   
***M122\_LB2\_Testvorschrift\_MS-B2\_Namen.docx***

# Testprotokoll (LB2 Meilenstein C2: individuelle Aufgabe 4)

Ausgefülltes Testprotokoll siehe Dokument   
***M122\_LB2\_Testvorschrift\_MS-C2\_Namen.docx***

# Systemdokumentation (Meilenstein C: individuelle Aufgabe 3)

Die erstellten Projekt-WPS-Scripts sind hier abgelegt und für Entwickler dokumentiert:

Öffentliche GitHub-URLs im Ablageordner auf dem BSCW ablegen! (pro Team)

Namenskonvention URL: **M122\_Klasse\_Thema\_Name\_Name**

*🡪 Ein* ***Branch*** *und separater Doku-Ordner pro Teammitglied erstellen*

## Umfang / Abgrenzung / Änderungen gegenüber Design

Aufgrund unten beschriebener Umstände sind Anpassungen des ursprünglichen Lösungsdesigns gemacht worden:

### Allgemein

Sämtlicher Umfang des geplanten Designs konnte umgesetzt werden.

### GUI Erweiterungen

Bei der Planung des GUIs wurden einige Buttons vergessen. Im Endprodukt wurden noch Delete Buttons auf dem «Monitored Folder» und dem «Rule Config» Dialog hinzugefügt. Ausserdem kam auf dem «Monitored Folders» Dialog noch ein Button zum Editieren hinzu.

## Funktionalität der Implementation.

Zusätzlich zu der Inline-Dokumentation sind hier folgende Funktionen / Eventhandler detailliert beschrieben:

### Main.ps1

Einstiegspunkt der Applikation. Öffnet den Menü Dialog und definiert einige globale Variablen.

#### Inhalt

Variablen:

|  |  |
| --- | --- |
| Name | Zweck |
| configPath | Pfad des Config files. Wird später beim Lesen der daten verwendet. |
| servicePath | Pfad des Service Scripts das beim Starten des Service vom Menü aus ausgeführt wird. |
| running | Status des Service. Boolean ob der Service im Hintergrund gerade läuft oder nicht. |

### Menu.psm1

GUI des Hauptmenüs der Applikation

#### Inhalt

Die GenerateMenu Funktion initialisiert das GUI des Hauptmenüs und öffnet diese als Dialog.

Inhalt der «GenerateMenu» Funktion:

|  |  |
| --- | --- |
| Funktion | Zweck |
| $handler\_configButton\_Click | Öffnet den Config Editor. |
| $handler\_runButton\_Click | Ändert den Status des Service auf «running» und umgekehrt. Verändert das GUI entsprechend des aktuellen Status. Startet oder beendet den Prozess des Service. |
| $getCurrentState | Überprüft ob der Prozess des Service läuft und setzt die global running Variable dementsprechend. |

### Folders\_config.psm1

GUI für den Config Editor Start Bildschirm

#### Inhalt

Inhalt der «GenerateConfig» Funktion:

|  |  |
| --- | --- |
| Funktion | Zweck |
| $fillGrid | Füllt die GridView mit den definierten folders im config.json. |
| $handler\_editButton\_Click | Öffnet den «FolderConfig» Dialog mit dem ausgewählten folder vom Grid. |
| $handler\_addButton\_Click | Öffnet den «MonitoredFolder» Dialog mit dem «new» parameter. |
| $open\_Global\_Config | Öffnet den «GlobalConfig» Dialog. |
| $CellMouseDoubleClick | Öffnet den «MonitoredFolder» mit dem doppelgeklickten folder als Parameter. |

### Folder\_config.psm1

Config Fenster für einen Spezifischen Ordner

#### Inhalt

Inhalt der «GenerateFolderConfig» Funktion:

Parameter: $index -> Index des ausgewählten Ordners im Json

|  |  |
| --- | --- |
| Funktion | Zweck |
| $fillGrid | Füllt die GridView mit den definierten rules für den ausgewählten folder im config.json. |
| $addButton\_OnClick | Öffnet den «RuleConfig» Dialog mit dem Parameter isNew = $true |
| $CellMouseDoubleClick | Öffnet den «RuleConfig» Dialog mit dem doppelgeklickten Item des Grids. |

### Global\_config.psm1

Config Fenster für die Globalen Regeln

#### Inhalt

Inhalt der "GenerateGlobalConfig" Funktion:

|  |  |
| --- | --- |
| Funktion | Zweck |
| $fillGrid | Füllt die GridView mit den definierten globalRules |
| $addButton\_OnClick | Öffnet den «RuleConfig» Dialog mit dem Parameter isNew = $true und isGlobal = $true |
| $CellMouseDoubleClick | Öffnet den «RuleConfig» Dialog mit dem doppelgeklickten Item des Grids und dem parameter isGlobal = $true |

### Monitored\_folder.psm1

Dialog zum bearbeiten oder erstellen eines überwachten Ordners

#### Inhalt

Inhalt der "GenerateMonitoredFolder" Funktion:

|  |  |  |
| --- | --- | --- |
| Parameter | Typ | Beschreibung |
| $name | String | Name des überwachten Ordners |
| $path | String | Pfad des überwachten Ordners |
| $new | Boolean | Flag ob ein neuer Ordner hinzugefügt wird oder ein bereits bestehender bearbeitet werden soll. |
| $index | Int | Index des zu bearbeitenden Ordners in der Config |

|  |  |
| --- | --- |
| Funktion | Zweck |
| $cancelButton\_onClick | Schliesst den Dialog ohne Änderungen an der Config vorzunehmen |
| $deleteButton\_OnClick | Löscht den Ordner aus der Config anhand des Indexes der als Parameter überreicht wurde. |
| $handler\_saveButoon\_Click | Fügt einen neuen Ordner in die Config hinzu, wenn das parameter $new = $true überreicht wurde. Ansonsten wird der vorhandene Ordner in der Config Editiert. |
| $handler\_pathpicker\_Click | Öffnet den PathPicker und fügt den Ausgewählten Wert in eine Textbox. |

### Rule\_config.psm1

### Dialog zum bearbeiten oder erstellen einer Regel.

#### Inhalt

Inhalt der "GenerateRuleConfig" Funktion:

|  |  |  |
| --- | --- | --- |
| Parameter | Typ | Beschreibung |
| $folderIndex | Int | Index des Ordners auf dem die Regel besteht. Ist leer wenn die Regel Global erstellt wurde |
| $index | Int | Index der Regel auf dem Folder oder in den globalen Regeln |
| $isGlobal | Boolean | Flag ob die Regel Global gilt. Wird später benötigt beim Bearbeiten des Config Files |
| $isNew | Boolean | Flag ob der Eintrag eine neue Regel ist. |

|  |  |
| --- | --- |
| Funktion | Zweck |
| $getRuleFromData | Erstellt ein Objekt mit den Daten des Dialogs |
| $saveButton\_OnClick | Speichert den Inhalt des Dialogs in das Config File |
| $handler\_pathPicker\_Click | Öffnet den PathPicker und fügt den Ausgewählten Wert in eine Textbox. |
| $handler\_deleteButton\_Click | Löscht die Regel aus der Config Datei |
| $ruleCancel\_onClick | Schliesst den Dialog ohne Änderungen vor zu nehmen. |
| $fillData | Füllt die Daten des Dialogs mit den Daten der ausgewählten Regel der Config ab. |

### Config.psm1

Funktionen die das Config File verwalten.

#### Inhalt

|  |  |  |
| --- | --- | --- |
| Funktion | Param | Zweck |
| ImportConfig | $Configpath | Importiert die Daten des Config files in ein Objekt und gibt dieses zurück. |
| SaveConfig | $config | Speichert ein Config objekt in im config.json |
| AddFolder | $folder | Fügt ein Ordner in der Config hinzu |
| EditFolder | $folder, $index, $withTypes | Editiert einen bestehenden Ordner in der Config |
| DeleteFolder | $index | Löscht einen bestehenden Ordner in der Config |
| AddGlobalRule | $rule | Fügt eine Globale Regel in die Config hinzu |
| EditGlobalRule | $rule,  $index | Editiert eine bestehende Globale Regel |
| DeleteGlobalRule | $index | Löscht eine bestehende Regel |
| SetLastId | $lastId | Setzt das "lastId" Property im Config.json |

### Pathpicker.psm1

Modul mit der Pathpicker Funktion

#### Inhalt

|  |  |
| --- | --- |
| Funktion | Zweck |
| PickPath | Öffnet den nativen Ordner Picker und gibt den ausgewählten Ordnerpfad als String zurück. |

### serviceFunctions.psm1

Funktionen die im service.ps1 benutzt werden.

#### Inhalt

|  |  |  |
| --- | --- | --- |
| Funktion | Param | Zweck |
| applyRules | $folder | Setzt die definierten Regeln für einen Ordner durch |
| applyGlobalRulesToFolder | $folder,  $config | Fügt die Globalen Regeln zu den Regeln eines Ordners hinzu, falls dieser noch keine Regel für diesen Dateityp enthält. Gibt den Ordner mit den neuen Regeln dazu. |

### Service.ps1

Der eigentliche Service. In ihm werden die definierten Regeln angewendet und die FileSystemWatcher registriert.

#### Inhalt

|  |  |  |
| --- | --- | --- |
| Funktion | Param | Zweck |
| run | $folder | Wendet die definierten Regeln im Config json an. |
| registerFileWatcher | $folder | Registriert FileSystemWatcher auf einem bestimmten Ordner |

# Betriebsdokumentation (Meilenstein C: individuelle Aufgabe 3)

Für Benutzer werden folgende Anleitungen ausgeliefert ...

## Installationsanleitung

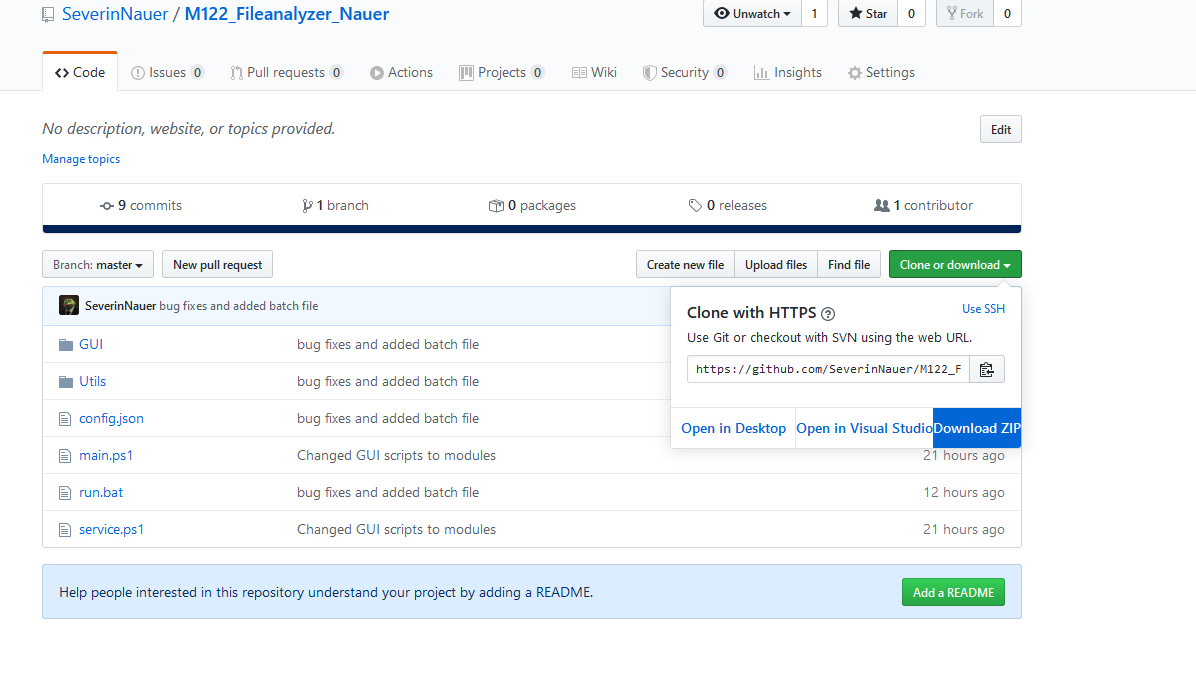
Das Programm ist folgendermassen zu installieren und konfiguriert:

### Herunterladen

Das Tool befindet sich unter folgendem GitHub link:

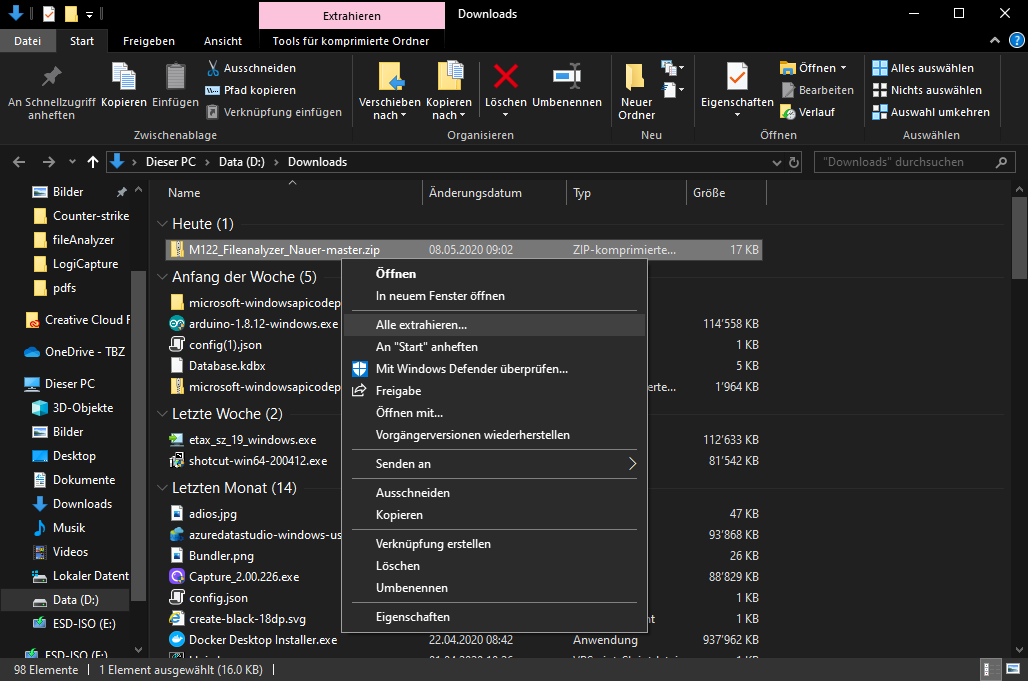
<https://github.com/SeverinNauer/M122_Fileanalyzer_Nauer>

Unter diesem Link kann ein Zip mit den Dateien heruntergeladen werden.



### Entpacken

Die heruntergeladen ZIP Datei ist im Download Ordner zu finden.  
Diese kann in einen beliebigen Ordner entpackt werden.



### Ausführen

Der entpackte Ordner beinhaltet ein batch File («run.bat»). Dieses dient als Einstiegspunkt des Programms. Mit einem Doppelklick auf diese Batch Datei öffnet sich der Einstiegsdialog.

### Verknüpfung

Damit der Ordner nicht immer gesucht und geöffnet werden muss, kann eine Verknüpfung der batch Datei an einem beliebigen Ort (z.B. Desktop) erstellt werden.

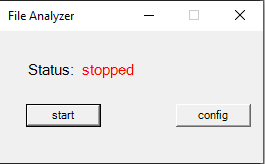
Erstellen der Verknüpfung:

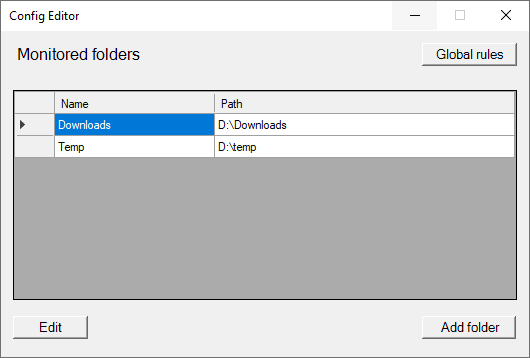
1. Rechtsklick auf das batch File
2. Klick auf „Verknüpfung erstellen“
3. Die Verknüpfung wird im gleichen Ordner erstellt
4. Die erstellte Verknüpfung in belieben Ordner verschieben

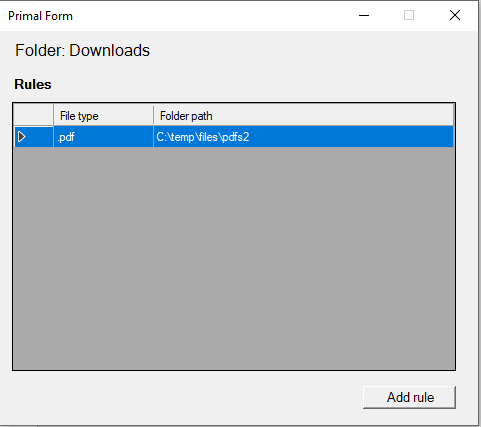
## Bedienungsanleitung für Benutzer

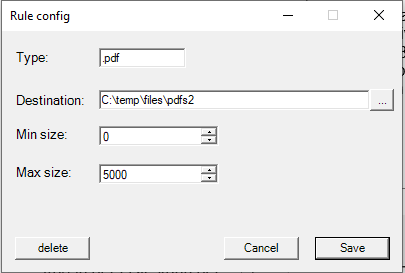
Das Programm ist folgendermassen zu bedienen:

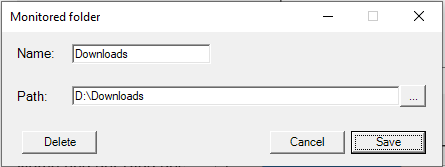
Beim Start öffnet sich das Hauptmenü.

  
Die Status anzeige indiziert ob der Service momentan aktiv oder gestoppt ist. Mit dem Start/Stop Button kann der Service gestartet oder gestoppt werden. Der Config Knopf dient zum Öffnen des Config Dialogs.

Im Config Editor werden die Überwachten Ordner angezeigt. Mit einem Doppelklick auf einen Eintrag in der Liste kann der Name und der Pfad des Ordners angepasst werden. Der "Montiored Folder" Dialog wird geöffnet. Der selbe Dialog wird mit dem "Add Folder" Button geöffnet. Der Edit Button öffnet den "Rules Dialog" mit dem Ausgewählten Listenelement. Der Global Rules Button öffnet den "Rules Dialog" mit den globalen Regeln.

Dieser Dialog zeigt eine Übersicht der Regeln auf dem bestimmten Ordner oder der globalen Regeln an. Mit einem Doppelklick auf eine Regel, kann diese Bearbeitet werden. Mit "Add rule" wird ein Dialog geöffnet in dem eine neue Regel hinzugefügt werden kann.



Im "Rule Config" Dialog können einzelne Regeln editiert, hinzugefügt oder gelöscht werden.

Im «Monitored Folder» Dialog kann ein einzelner Ordner editiert, hinzugefügt oder gelöscht werden.