Dokumentation Projekt FileAnalyzer

Severin Nauer

1 Teilnehmer/innen des Teams:

Klasse:	Team:
AP17C	sudokuschwer.com

2 Anforderungsdefinition (Meilenstein A)

Filemanager

Fachlicher Inhalt: (Allgemeine Beschrei-bung)	**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

2.1 Planung Meilensteine (LB1 / LB2)

MS	*Tätigkeit / Abgabe*	*Soll-Datum*	*Ist-Datum*
A	**Projektstart** ➢ Team Bildung, Kollaborationsplattform, GitHub Repos *, Lehrerzugang ➢ **Wahl / 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 Dokumentation ➢ **Fachgespräch Projektabnahme**		
C2	**Einzelaufgabe 4: (Nur LB2)** ➢ Abgabe: Ausgefüllter Systemtest		

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

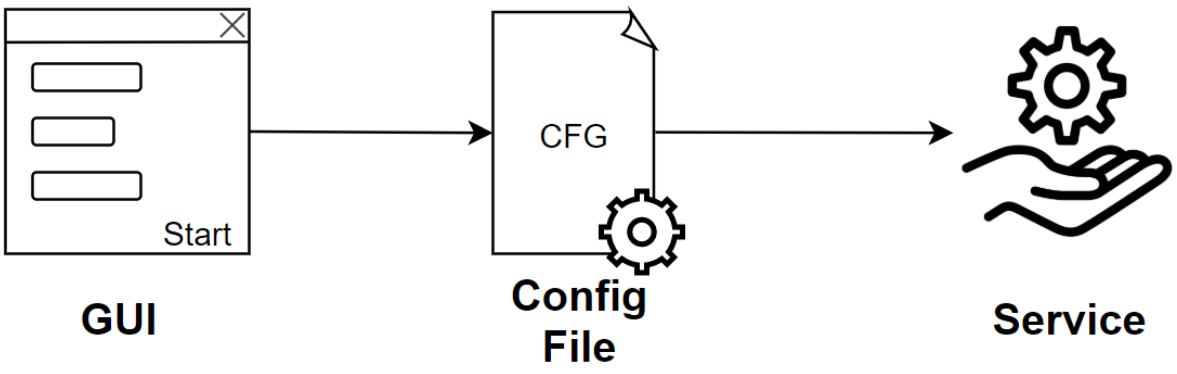
Namenskonvention URL: **M122_Klasse_Thema_Name_Name**

3 Lösungsdesign (Meilenstein B: Teamaufgabe 1)

Anhand der Analyse wurde folgendes Lösungsdesign entworfen:

3.1 Schematische Darstellung der Funktionalität, sog. Funktionsmodell

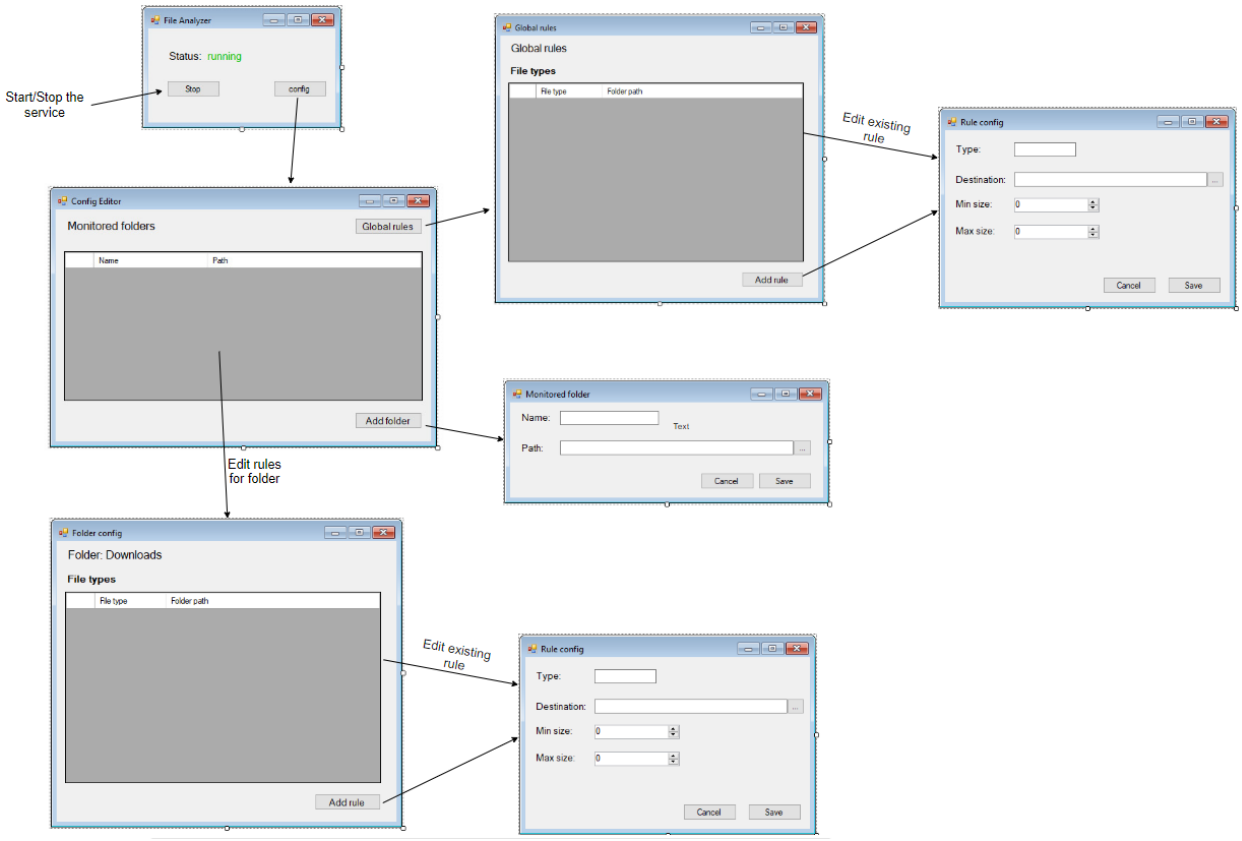
Via GUI kann unser Config-File bearbeitet werden. Dieses wird dann als Parameter an unseren Service weitergeleitet.



M122-Funktionsdiagram.drawio

3.2 Graphische Benutzer Schnittstelle (GUI) zur Konfiguration des Ablaufs

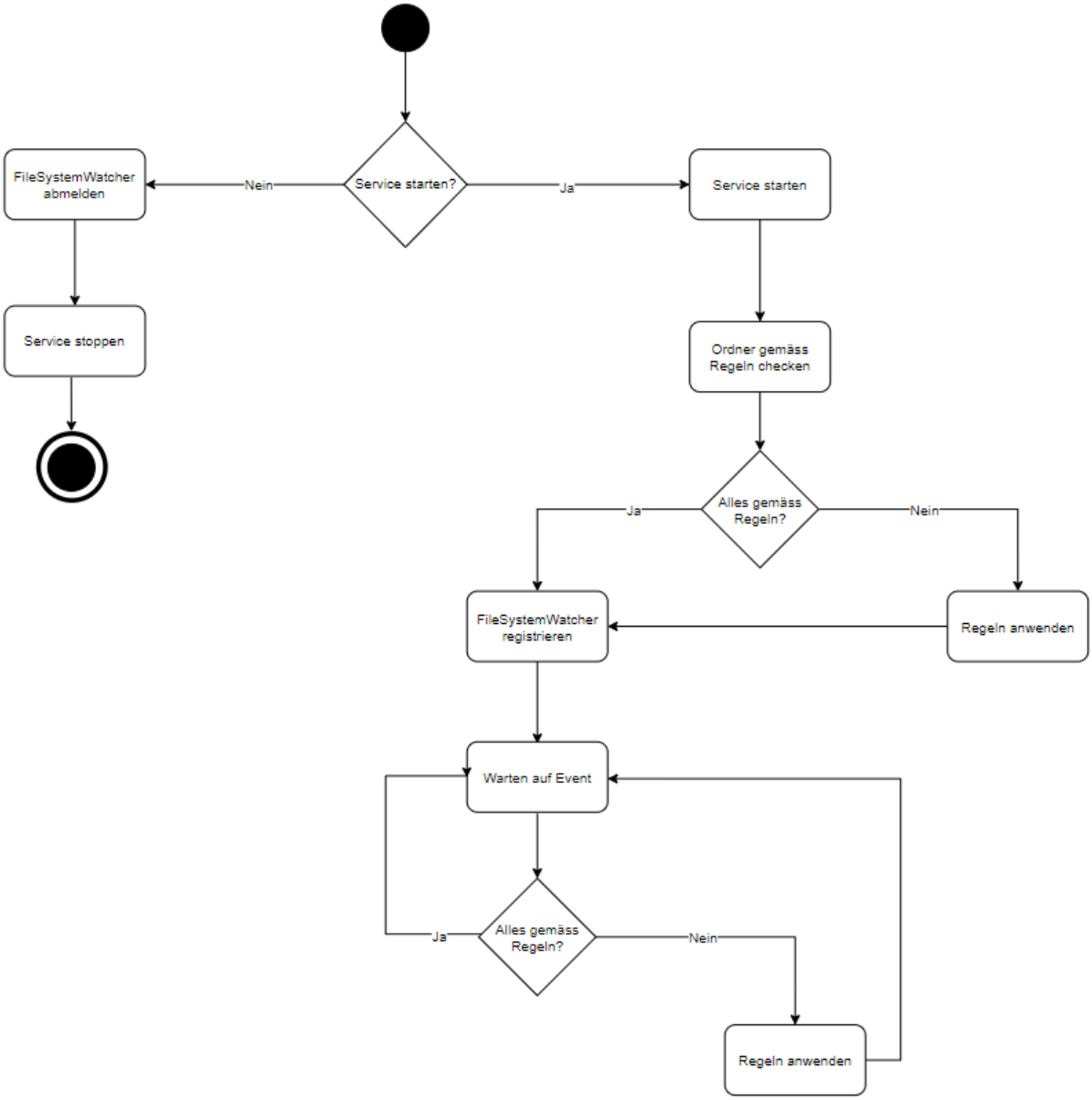
Mit unserem zentralen Einstiegspunkt kann unser Service gestartet sowie auch gestoppt werden. Ebenfalls kann man von diesem Einstiegspunkt auch das Config-File öffnen und bearbeiten.



M122-Mock.drawio

3.3 Ablauf der Automation

Aus Benutzersicht ist folgender Ablauf des Programms zu erwarten:



M122-Aktivitätsdiagram.drawio

4 Testvorschrift (LB2 Meilenstein B2: Teamaufgabe 2)

Testbeschrieb und vorbereitetes Testprotokoll siehe Dokument
M122_LB2_Testvorschrift_MS-B2_Namen.docx

5 Testprotokoll (LB2 Meilenstein C2: individuelle Aufgabe 4)

Ausgefülltes Testprotokoll siehe Dokument
M122_LB2_Testvorschrift_MS-C2_Namen.docx

6 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*

6.1 Umfang / Abgrenzung / Änderungen gegenüber Design

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

[redacted]Allgemein

Sämtlicher Umfang des geplanten Designs konnte umgesetzt werden.

[redacted]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.

6.2 Funktionalität der Implementation.

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

[redacted]Main.ps1

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

[redacted]*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.

[redacted]Menu.psm1
GUI des Hauptmenüs der Applikation
[redacted]*Inhalt*
Die GenerateMenu Funktion initialisiert das GUI des Hauptmenüs und öffnet diese als Dialog.
Inhalt der «GenerateMenu» Funktion:

Funktion	Zweck
$handler_config-Button_Click	Öffnet den Config Editor.
$handler_runBut-ton_Click	Ändert den Status des Service auf «running» und umgekehrt. Verändert das GUI entsprechend des aktuellen Status. Startet o-der beendet den Prozess des Service.
$getCurrentState	Überprüft ob der Prozess des Service läuft und setzt die global running Variable dementsprechend.

[redacted]Folders_config.psm1
GUI für den Config Editor Start Bildschirm
[redacted] *Inhalt*
Inhalt der «GenerateConfig» Funktion:

Funktion	Zweck
$fillGrid	Füllt die GridView mit den definierten folders im config.json.
$handler_editBut-ton_Click	Öffnet den «FolderConfig» Dialog mit dem ausgewählten folder vom Grid.
$hand-ler_addBut-ton_Click	Öffnet den «MonitoredFolder» Dialog mit dem «new» parameter.
$open_Glo-bal_Config	Öffnet den «GlobalConfig» Dialog.
$CellMouse-DoubleClick	Öffnet den «MonitoredFolder» mit dem doppelgeklickten folder als Parameter.

[redacted]Folder_config.psm1
Config Fenster für einen Spezifischen Ordner
[redacted]*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_On-Click	Öffnet den «RuleConfig» Dialog mit dem Parameter isNew = $true
$CellMouse-DoubleClick	Öffnet den «RuleConfig» Dialog mit dem doppelgeklickten Item des Grids.

[redacted]Global_config.psm1
Config Fenster für die Globalen Regeln
[redacted]*Inhalt*
Inhalt der "GenerateGlobalConfig" Funktion:

Funktion	**Zweck**
$fillGrid	Füllt die GridView mit den definierten globalRules
$addButton_On-Click	Öffnet den «RuleConfig» Dialog mit dem Parameter isNew = $true und isGlobal = $true
$CellMouse-DoubleClick	Öffnet den «RuleConfig» Dialog mit dem doppelgeklickten Item des Grids und dem parameter isGlobal = $true

[redacted]Monitored_folder.psm1
Dialog zum bearbeiten oder erstellen eines überwachten Ordners
[redacted]*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
$cancelBut-ton_onClick	Schliesst den Dialog ohne Änderungen an der Config vorzuneh-men
$deleteBut-ton_OnClick	Löscht den Ordner aus der Config anhand des Indexes der als Parameter überreicht wurde.
$handler_save-Butoon_Click	Fügt einen neuen Ordner in die Config hinzu, wenn das parame-ter $new = $true überreicht wurde. Ansonsten wird der vorhan-dene Ordner in der Config Editiert.
$handler_pathpi-cker_Click	Öffnet den PathPicker und fügt den Ausgewählten Wert in eine Textbox.

[redacted]Rule_config.psm1
Dialog zum bearbeiten oder erstellen einer Regel.
[redacted]*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.

[redacted]Config.psm1
Funktionen die das Config File verwalten.
[redacted]*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
DeleteGlobal-Rule	$index	Löscht eine bestehende Regel
SetLastId	$lastId	Setzt das "lastId" Property im Config.json

[redacted]Pathpicker.psm1

Modul mit der Pathpicker Funktion

[redacted]*Inhalt*

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

[redacted]serviceFunctions.psm1

Funktionen die im service.ps1 benutzt werden.

[redacted]*Inhalt*

Funktion	**Param**	**Zweck**
applyRules	$folder	Setzt die definierten Regeln für einen Ordner durch
applyGlobal-RulesToFolder	$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.

[redacted]Service.ps1

Der eigentliche Service. In ihm werden die definierten Regeln angewendet und die File-SystemWatcher registriert.

[redacted]*Inhalt*

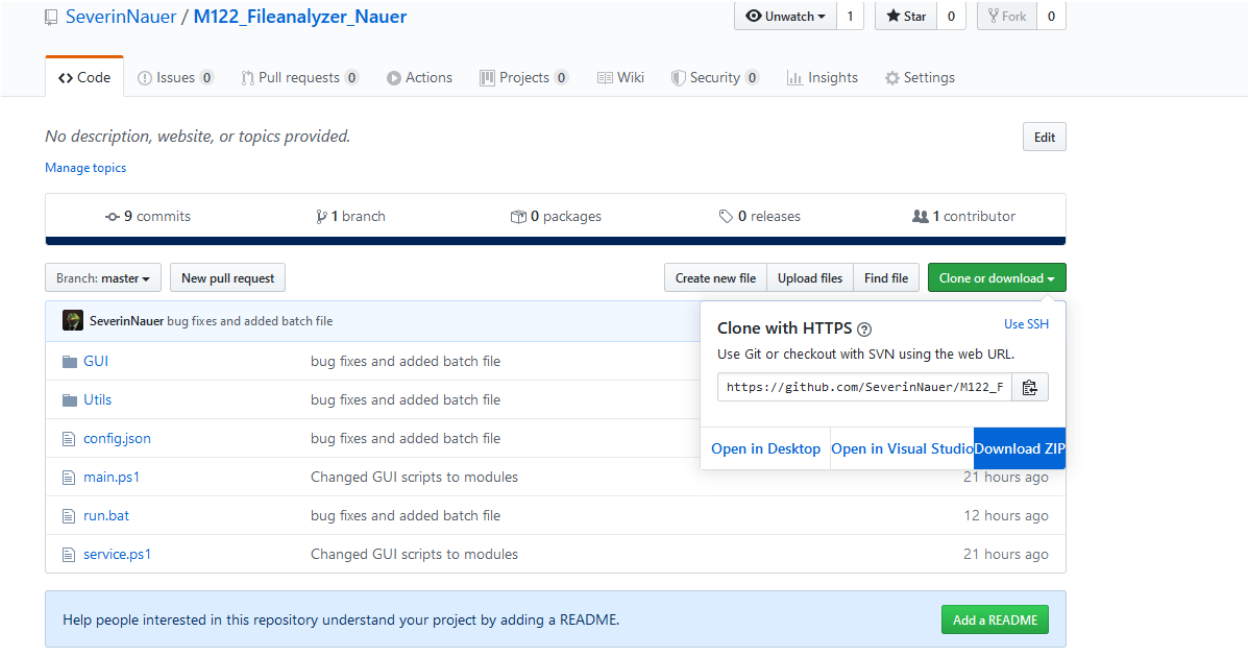
Funktion	**Param**	**Zweck**
run	$folder	Wendet die definierten Regeln im Config json an.
registerFile-Watcher	$folder	Registriert FileSystemWatcher auf einem bestimmten Ordner

7 Betriebsdokumentation (Meilenstein C: individuelle Aufgabe 3)

Für Benutzer werden folgende Anleitungen ausgeliefert ...

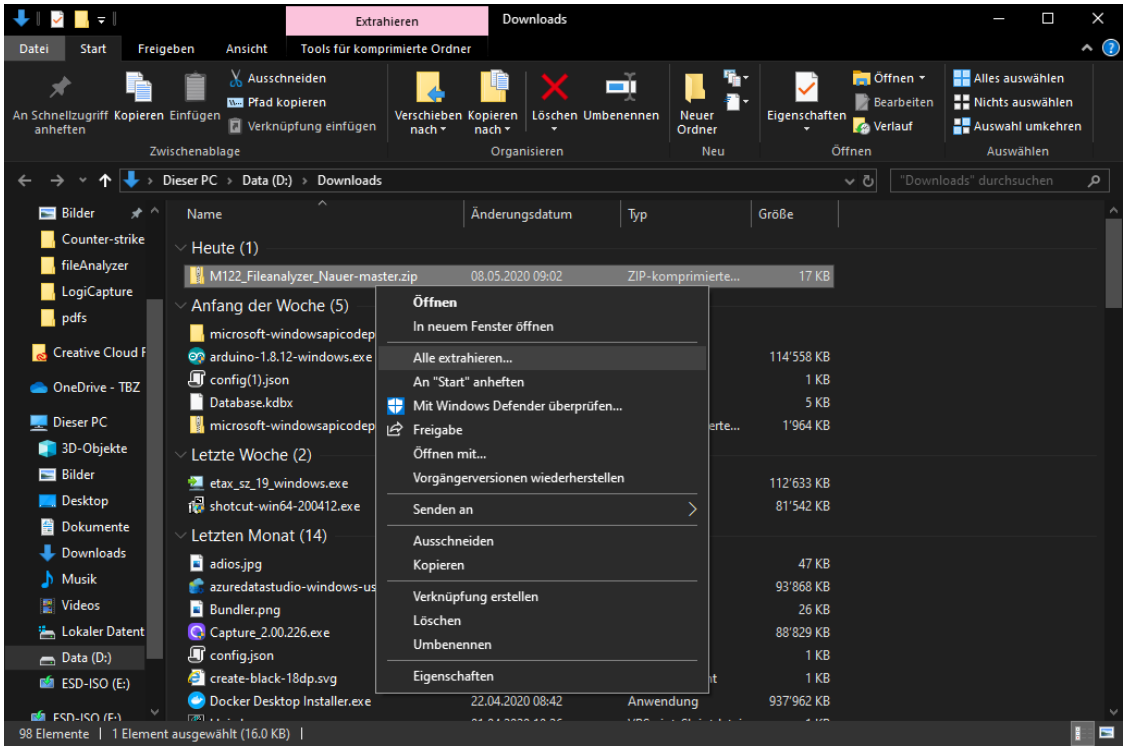
7.1 Installationsanleitung

Das Programm ist folgendermassen zu installieren und konfiguriert:
[redacted]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.



[redacted]Entpacken

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



[redacted]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.

[redacted]Verknüpfung

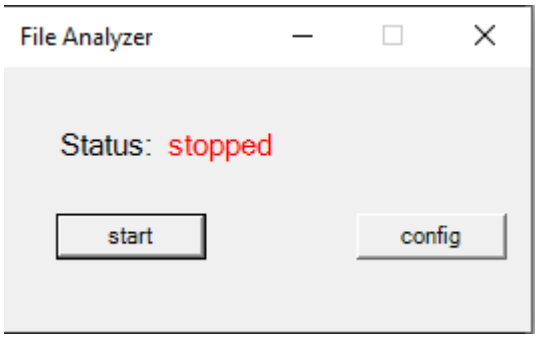
Damit der Ordner nicht immer gesucht und geöffnet werden muss, kann eine Verknüpfung der batch Datei an einem beliebigen Ort (z.B. Desktop) erstellt werden.

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

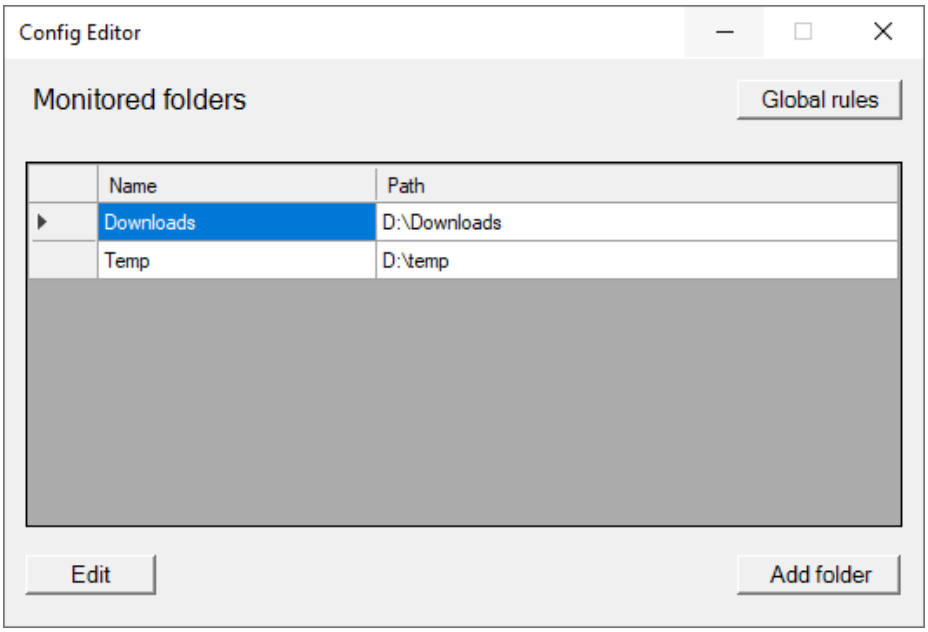
7.2 Bedienungsanleitung für Benutzer

Das Programm ist folgendermassen zu bedienen:
Beim Start öffnet sich das Hauptmenü.

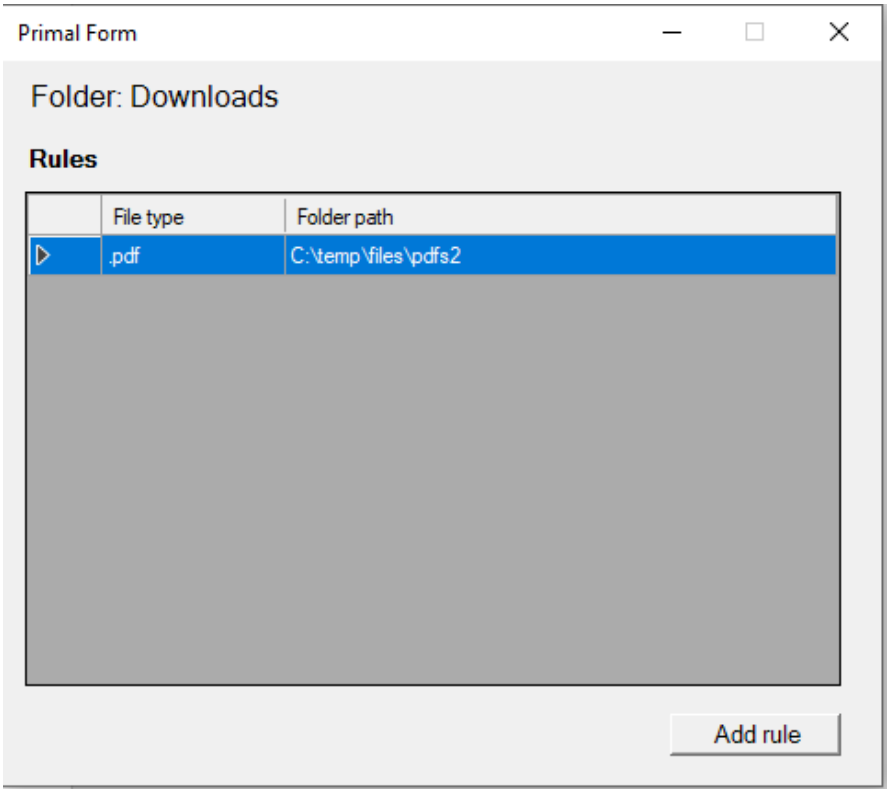


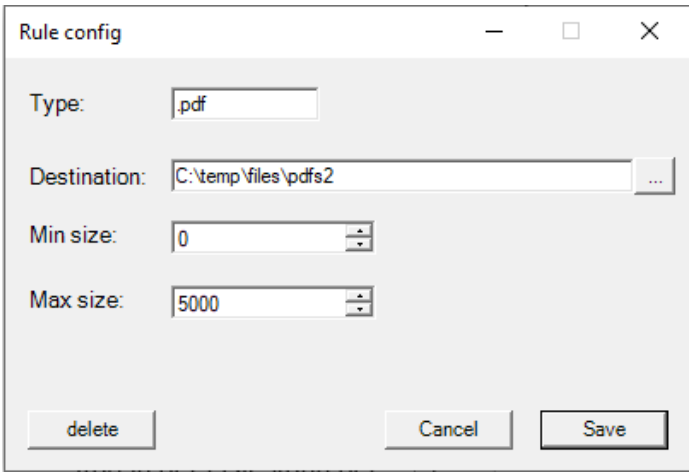
Die Status anzeige indiziert ob der Service momentan aktiv oder gestoppt ist. Mit dem Start/Stop Button kann der Service gestartet oder gestoppt werden. Der Config Knopf dient zum Öffnen des Config Dialogs.

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.