2023

EasySave

Prosoft

23/02/2023

Documentation technique

**Une image contenant outil

Description générée automatiquement**

Table des matières

[1. Introduction 2](#_Toc128051712)

[2. Systèmes d’exploitation disponibles : 2](#_Toc128051713)

[3. Configuration des langues 3](#_Toc128051714)

[4. Technologies utilisées 4](#_Toc128051715)

[5. Structure du programme 4](#_Toc128051716)

[6. Les fichiers de logs 5](#_Toc128051717)

[7. Fenêtre Client 6](#_Toc128051718)

# Introduction

Cette documentation est destinée à répertorier l'ensemble des détails techniques de la V3 d'EasySave.

Ce document va aborder des détails comme les systèmes d’exploitation sur lesquels EasySave fonctionne, la structure de l'application ou encore la configuration des langages.

# Systèmes d’exploitation disponibles :

Pour ce qu'il est des systèmes d'exploitation sur lesquels EasySave peut fonctionner c'est assez général, on entend par "général" c'est que l'on peut générer une version de l'application pour chaque système d'exploitation.

De base l’application a été générée pour un système windows 64 bits mais on aurait bien pu générer une version pour des systèmes Linux ou Mac OS. Pour accéder à une autre version (autre que windows) il faudra faire une demande auprès de notre service commercial pour que l’on génère une version pour un autre système d’exploitation.

Une image contenant texte

Description générée automatiquementVoici la liste des systèmes d’exploitation disponibles :

# Configuration des langues

Pour ce qu’il est de la traduction de l’application elle est gérée par des fichiers .xaml organisé de la manière suivante :

Une image contenant texte, orange

Description générée automatiquement

Chaque fichier "Language" modifie l’entièreté des langues dans l’application

Une image contenant texte

Description générée automatiquementVoici un exemple du fichier "LanguageFr" :

Pour rajouter de nouvelles traductions il suffit de rajouter de nouveau .xaml, par exemple pour l’Allemand on aurait : LanguageDe.xaml.

# Technologies utilisées

Pour le développement de cette application les technologies utilisées sont :

* Le framework .NET 6.0
* Le langage C#
* Le format JSON
* Le format XML
* Interface en WPF
* Github pour le versionning et le travail collaboratif
* L’IDE Visual Studio 2022

# Structure du programme

La structure du programme est basée sur le modèle MVVM.

Voici l’architecture de la solution :

Vous pouvez retrouver la View et les ViewModels dans le projet easy\_save.Desktop qui contient tout l’interface WPF de l’application.

Et vous retrouverez les Models et les Services dans easy\_save.Lib.

Une image contenant texte

Description générée automatiquement

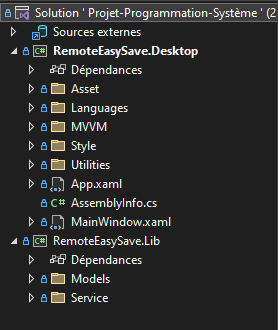
Pour ce qui est des détails de la structure du code merci de vous référer aux diagrammes UML.

# Les fichiers de logs

Voici l’ensemble des différents fichiers de log.

|  |  |
| --- | --- |
| Fichier de log d’état json | Une image contenant texte  Description générée automatiquement |
| Fichier de log d’état xml |  |
| Fichier de log daily  json |  |
| Fichier de log daily  xml |  |

# Fenêtre Client

L’application EasySave comprend aussi un projet client qui permet de déporté une fenêtre pour accéder seulement à la partie contrôle des processus.

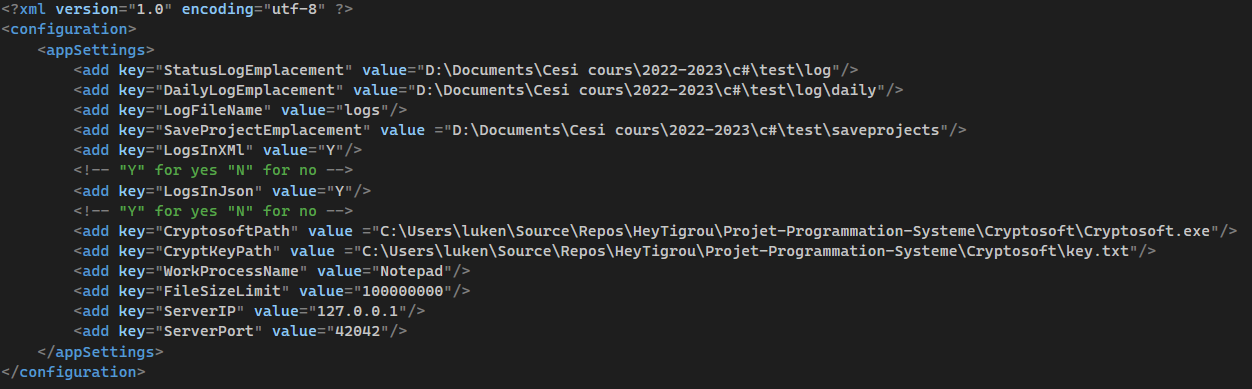
Pour ce faire on retrouve une structure de la solution similaire à celle du projet initial.

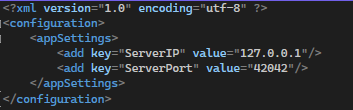
Avec RomoteEasySave.Desktop qui contient toutes l’interface en WPF, puis on la partie RemoteEasySave.Lib contenant les modèles et servie pour le client.

A savoir que nous utilisons des sockets pour réaliser cette fenêtre déportée.

Vous pouvez aussi modifier les IP et Port de connexion entre le serveur et le client depuis les fichiers app.config du Client et du Serveur :

* App.Config Serveur :



* App.Config Client :