2023

EasySave

Prosoft

13/02/2023

Technical documentation

**Une image contenant outil

Description générée automatiquement**

Table of contents

[1. Introduction 2](#_Toc127225166)

[2. Available operating systems 2](#_Toc127225167)

[3. Langages configuration 2](#_Toc127225168)

[4. Technologies used 3](#_Toc127225169)

[5. Program structure 3](#_Toc127225170)

[6. Log files 4](#_Toc127225171)

# Introduction

This documentation is intended to list all the technical details of EasySave V1.

This document will cover details such as the operating systems on which EasySave runs, the structure of the application and the configuration of the languages.

# Available operating systems

As for the operating systems on which EasySave can run, it is quite general, by "general" we mean that we can generate a version of the application for each operating system.

The application has been generated for a 64-bit Windows system, but we could have generated a version for Linux or Mac OS systems. To access another version (other than Windows) you will have to ask our sales department to generate a version for another operating system.

Une image contenant texte

Description générée automatiquementHere is the list of available operating systems:

# Langages configuration

Une image contenant texte

Description générée automatiquementAs for the translation of the application, it is managed by .resx files organised as follows:

The String.resx will be the default resource used and then we can manually change the culture of the program. If we change the culture of the program to "fr" the program will use String.fr.resx instead of String.resx for translation.

To add new translations you just need to add a new .resx, for example for German you would have: String.de.resx.

# Technologies used

For the development of this application the technologies used are :

* The .NET 6.0 framework
* The C# language
* The JSON format
* The XML format
* Github for versioning and collaborative work
* Visual Studio 2022 IDE

# Program structure

The structure of the program is based on the MVVM model.

Here is the architecture of the solution :

Une image contenant texte, équipement électronique, écran, fermer

Description générée automatiquement

You can find the View and the ViewModels in the easy\_save.Cmd project and you will find the Models and the Services in easy\_save.Lib

For the details of the code structure please refer to the UML diagrams.

# Log files

Here are all the different log files.

|  |  |
| --- | --- |
| State log file json | State log file xml |
| Une image contenant texte  Description générée automatiquement | Une image contenant texte  Description générée automatiquement |

|  |  |
| --- | --- |
| Daily log file json | Une image contenant texte  Description générée automatiquement |
| Daily log file xml | Une image contenant texte  Description générée automatiquement |