Source Projects
Presents

SCKF Editor

Written in Java
https://www.java.com

Table of contents

2. About the project

The idea of this project is made to generate custom SCKF files for the “Source Cipher”, you can check it out at GitHUB website, just type in my name and surname. In this project I introduced new features like custom language and so on.

I created this project to create something unusual that no one created, it was meant as an add-on to source cipher while being a standalone application.

While the project was successful, the first build was a total BS, lots of features were buggy and not working, but I managed to fix them. The obvious feature is that the program has multilingual feature and has the language editor.

This project was made under Linux Mint 19.2 with Eclipse IDE for Java and the Jar file was exported from this IDE. The project was made to test out some features that weren’t included in “Source Cipher”.

This was a challenge to create this projects, but I took it and succeeded. The main challenge was to make the program to support multiple languages and able to edit the language files and ability to easilly to install them. The amounts of bugs that were fixed and tested in the application the most common ways as possible.

Chapter II: How to use it

1. Creating an SCKF

Just type in anything in the “Primary password:” field, this will create a single key SCKF encryption key, if you check the “Randomize” check box, this will create a double key SCKF encryption key.

It’s easy actually to create it, just tick “Randomize” check box and you’ll see that the password will shuffle in the “Secondary password:” field. To save the progress simply click “File” menu and select “Save” or “Save as” to save the progress. The key limit is 512 symbols, the limit is described in F.A.Q. section.

The UNICODE symbol filter wasn’t implemented so avoid using UNICODE symbols or you might encounter errors in the application, use ANSI letters (English letters and symbols).

2. Adding language file

If you want to add a language file, simply put “<language code>.properties” for example “ru.properties” file into “langs” directory, be sure that the application is closed. Once you putted the “ru.properties” into “langs” directory simply start the application, it’ll import the language file and it’ll be available in the “Language > Languages” menu.

Be sure that the language file start with 2 letters before “.properties” extension for example “ru”, “lt”, “en” or else the application will ignore the third letter.

To set a new language simply go to “Language > Languages” menu and then select your new language, if everything went OK you’ll see a message box to restart the application, then close and reopen the application and you’ll see that the application started using the new language that you selected.

If the “langs” directory is empty, you’ll see an error. To resolve it, simply delete the “config.properties” file and “langs” directory and then start the application, it’ll revert into default settings.

3. Creating or editing existing language file

Well it’s not that difficult to create a language file but at the same time it’s not easy, I made as simple as possible to do it.

To make it easier to create a language file, simply click “Open” button and select “en.properties”, this will load all the variables and the fields will get populated.

Note: The first field “Language code” only accepts 2 letters only, this is where application takes it and names the “.properties” file, for example the “Language code” field is “ru”, then the language file will be “ru.properties”.

If you don’t get anything after clicking “Save” button then that means that the language file was generated successfully. The same goes for when your editing existing language file. If there was an error, you’ll see the message box indicating the error.

Chapter III: F.A.Q.

1. General

Q: Who is the developer of this application?

A: Edgars Zabroda, you can visit my GitHUB page where I put all of my project. Click here to visit it.

Q: How can I contact you?

A: You can contact me through my GitHUB page.

2. GitHUB related questions

Q: Why do you post your code on GitHUB?

A: So that people have freedom to contribute to my projects and enhance or add/remove features from them and so that newbie programmers could learn from the code that I post.

Q: I know how to code, can I contribute to your project?

A: Of course you can, the repository isn't archived. You can select any branch that you want to edit. Each branch of this project is a specific Java package where if you don't want to download the entire code. If you pull the "master" branch, you'll get the entire source code of this project. After you made changes you can push to my GIT repository.

Q: Can I take credits for what I improved in the application?

A: Of course, why even ask that. People who contributed to my project will be listed in "Credits" branch once I'll get the push request.

3. Application specific

Q: Why can’t I make a password bigger than 512 symbols?

A: Well this limit was made because the cipher multiplies the password length by 1 megabyte and the cipher was made take up to 512 megabytes of RAM and that’s why this limit is imposed.

Q: I found a bug, where can I report it?

A: Post them here and I’ll try to fix them.