

PasswordManager

By Joshua Stone

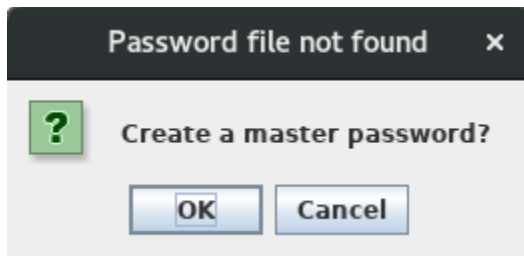
About:

This document is for demonstrating the functionality of the PasswordManager program, a graphical utility that acts as a username and password storage tool and protects data with a master password.

HOWTO:

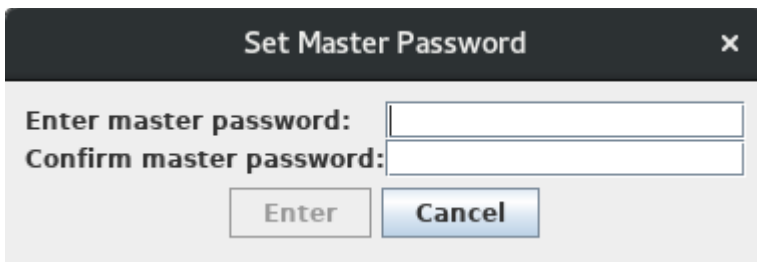
Step 1: Starting the application

On the first run, the program will check for the existence of a password files used for storing all user data. If no such file exists, then it should ask to create a new master password.

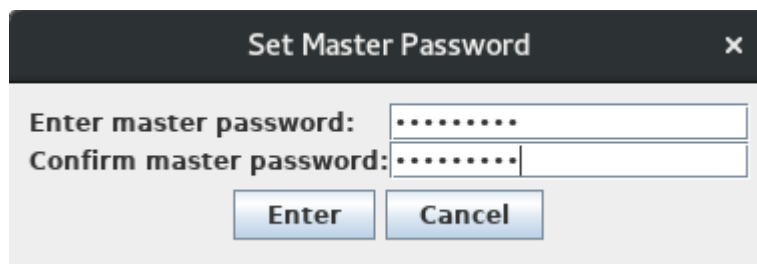


Step 2: Creating the master password

After clicking **Ok**, there should be a new window with two password input fields. The **Enter** button is disabled until both fields have a matching passwords for input.

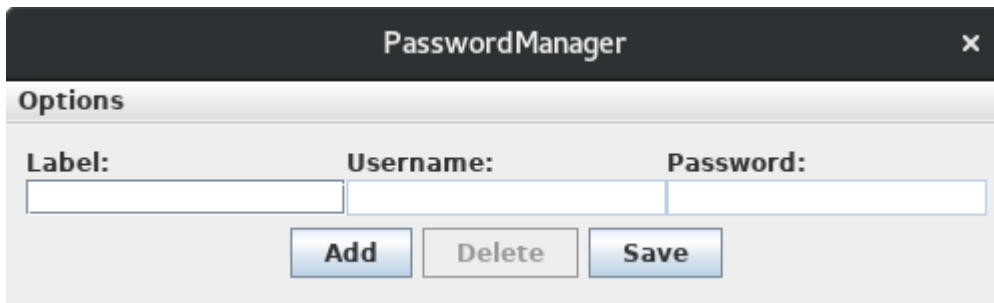


The **Enter** button is disabled until both fields have a matching passwords for input.



Step 3: Entering the main session

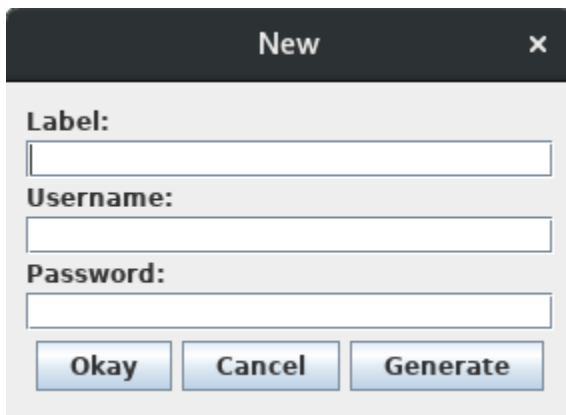
After a master password has been created, a new password store has been initiated and the main session window will open. All fields should be empty.



The screenshot shows a window titled "PasswordManager" with a close button (X) in the top right corner. Below the title bar is a tab labeled "Options". Under the "Options" tab, there are three input fields labeled "Label:", "Username:", and "Password:". All three fields are currently empty. Below the input fields are three buttons: "Add", "Delete", and "Save".

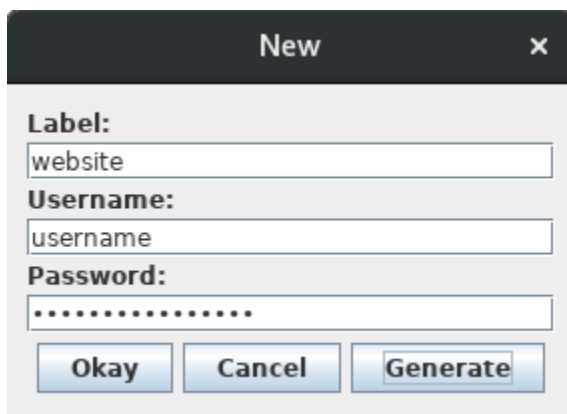
Step 4: Adding a new username and password

Pressing **Add** will create a new dialog that'll leave the main window disabled until the new window is closed.



The screenshot shows a dialog box titled "New" with a close button (X) in the top right corner. Below the title bar, there are three input fields labeled "Label:", "Username:", and "Password:". All three fields are currently empty. Below the input fields are three buttons: "Okay", "Cancel", and "Generate".

A *label* is for identifying which username and password is used for what. It can be the name of a website, email address, etc. Optionally, pressing **Generate** will generate a random combination of letters and numbers in the password field.

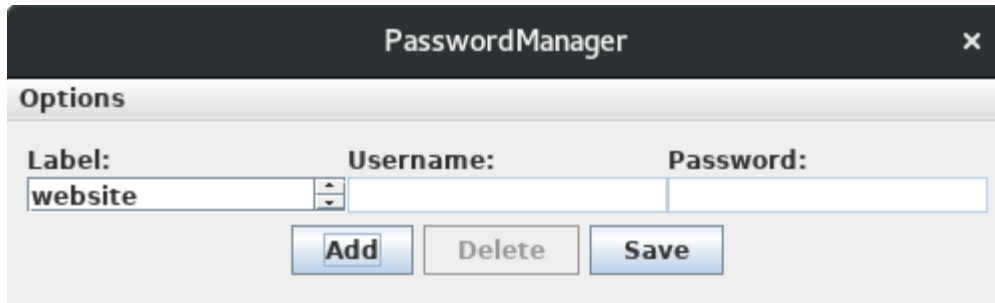


The screenshot shows the same "New" dialog box as before, but now the input fields are filled. The "Label:" field contains the text "website", the "Username:" field contains the text "username", and the "Password:" field contains a series of dots representing a generated password. The "Generate" button is highlighted with a blue border, indicating it is the active button.

Attempting to press **Okay** with one or more of the fields empty will result in an error message, so be sure to fill all three fields.

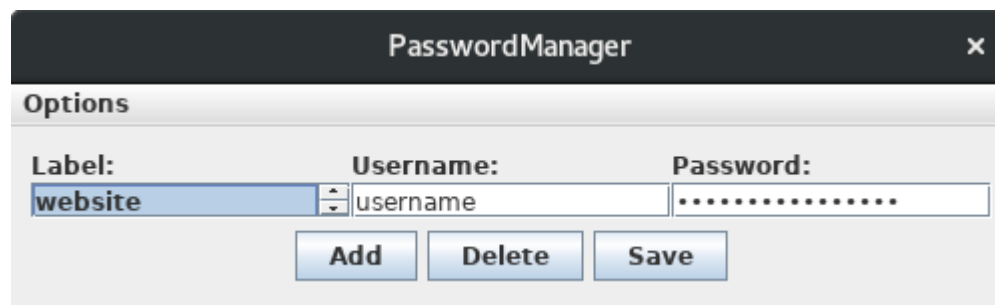
Step 5: Viewing user credentials

After pressing **Ok**, the the newly-entered label should be visible in the main window



The screenshot shows the PasswordManager application window. The title bar is dark gray with the text "PasswordManager" and a close button (X). Below the title bar is a tab labeled "Options". The main area contains three input fields: "Label:" with the text "website", "Username:" which is empty, and "Password:" which is empty. Below these fields are three buttons: "Add", "Delete", and "Save". The "Add" button is highlighted with a blue border.

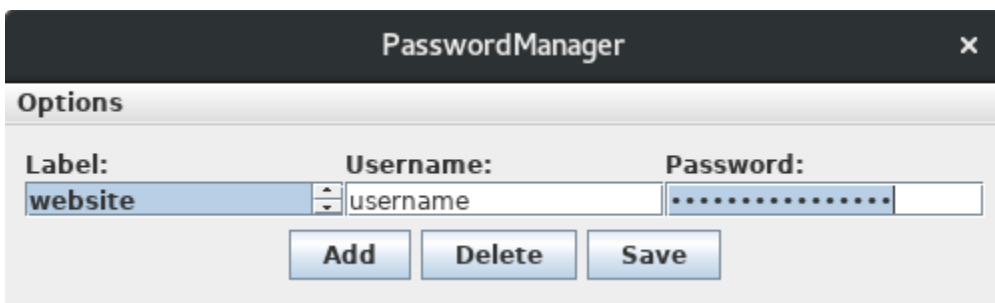
Selecting the label will show the username and password associated with it, as well as making both fields editable. The **Delete** button will also be enabled if one wants to remove credentials.



The screenshot shows the PasswordManager application window. The title bar is dark gray with the text "PasswordManager" and a close button (X). Below the title bar is a tab labeled "Options". The main area contains three input fields: "Label:" with the text "website", "Username:" with the text "username", and "Password:" with a series of dots. Below these fields are three buttons: "Add", "Delete", and "Save". The "Add" button is highlighted with a blue border.

Step 6: Retrieving a password

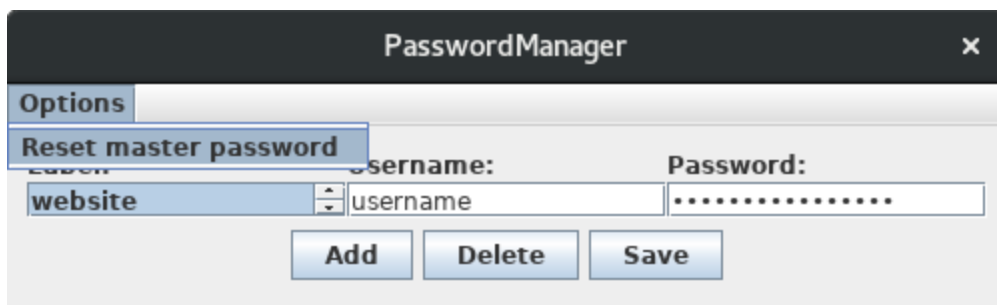
Right-click is disabled in the window, instead **Ctrl-C** will copy passwords to the clipboard.



The screenshot shows the PasswordManager application window. The title bar is dark gray with the text "PasswordManager" and a close button (X). Below the title bar is a tab labeled "Options". The main area contains three input fields: "Label:" with the text "website", "Username:" with the text "username", and "Password:" with a series of dots. Below these fields are three buttons: "Add", "Delete", and "Save". The "Add" button is highlighted with a blue border.

Step 7: Resetting the master password

If there's ever a need to reset a master password, then just look in the **Options** menu.

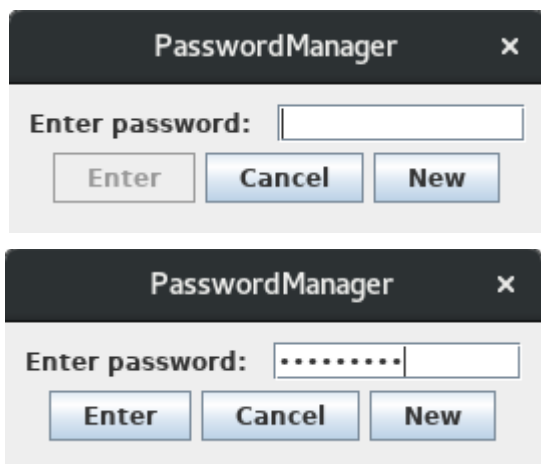


The screenshot shows the 'PasswordManager' application window. The 'Options' menu is open, and the 'Reset master password' option is selected. This opens a sub-window with the following elements: a 'Username:' label above a text input field containing 'username'; a 'Password:' label above a password input field filled with dots; a dropdown menu on the left showing 'website'; and three buttons at the bottom: 'Add', 'Delete', and 'Save'.

Clicking **Reset master password** will open a new window that looks like to the one in **Step 2**.

Step 8: Saving a session

Once all data has been entered, click **Save** and click the **Close** button. Restarting the application should open a different window prompting for a password.



The first screenshot shows the 'PasswordManager' window with the prompt 'Enter password:' and an empty text input field. Below the input field are three buttons: 'Enter', 'Cancel', and 'New'. The second screenshot shows the same window, but the input field now contains seven dots, indicating a password has been entered. The 'Enter', 'Cancel', and 'New' buttons remain at the bottom.

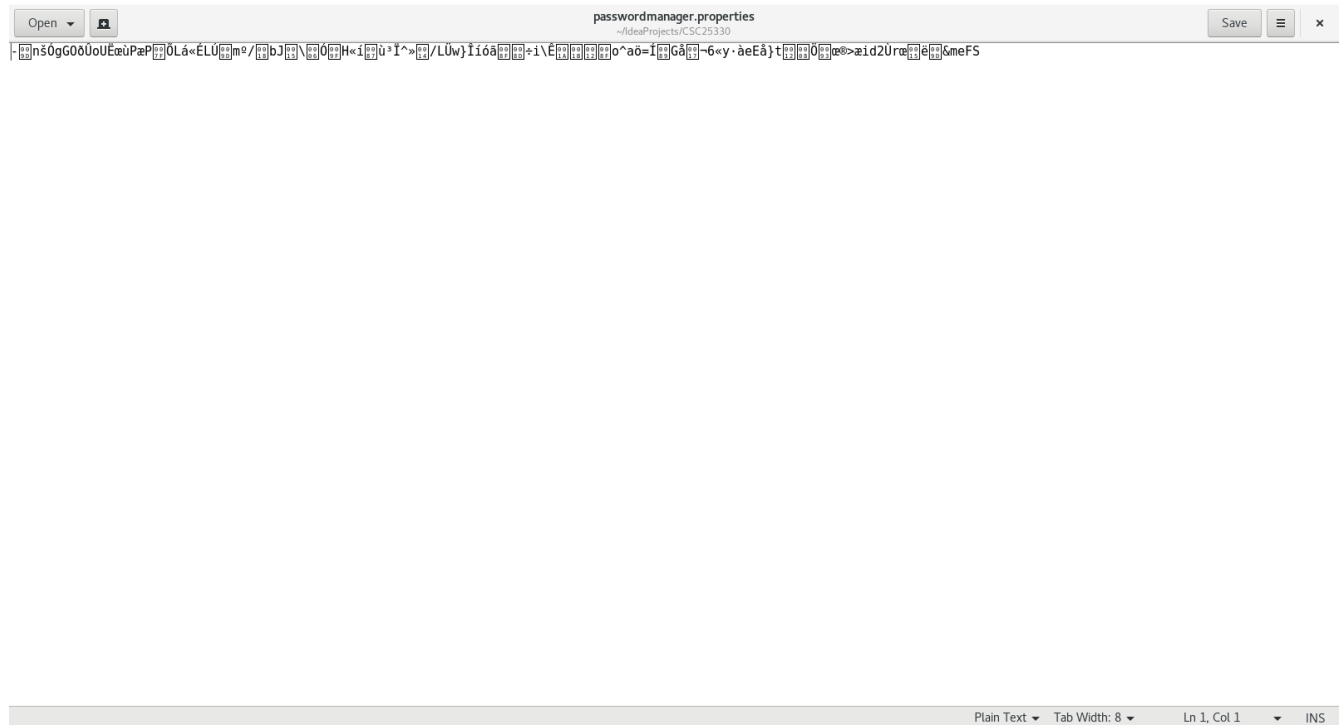
Pressing the **New** button will start a brand new session, although it won't overwrite the previous password file unless **Save** has been pressed.

Step 9: Verifying that the passwords are secure

The PasswordManager program is built around standard encryption algorithms for file encryption, where a master password acts as a *key* to unlock the password file and read its contents.

The use of a randomly generated *salt* acts as an additional input of random data, and an *initialization vector* (iv) is used for ensuring randomization when using the same key. This has the property of producing a completely random sequence of bytes every time PasswordManager writes to the disk, even if none of the data changed and the same password is being used.

Data is stored in *passwordmanager.properties* found in the same directory where the program is being run, and attempting to open the file will show completely random data.



Repeatedly pressing **Save** in the application will keep writing a random byte sequence, but they all contain the exact same data.