# Hex Password Generator

**Introduction:**

The Hex Password Generator is a small program used to create Random Passwords. In the Standard mode it uses Characters from A to Z, and lower case a to z, and numbers from 0 to 9. In Enhanced mode it adds 10 Special Characters, ! @ # $ % ( ) < > ? for sites that allow, or require them.

From what I have read, it's good to have Passwords based on Entropy. Entropy is, Length and Probability, that is the number of characters and how many different characters are involved.

The formula for Entropy is: H = L * log(N) / log(2)  Where H is Entropy, L is Length and N is the number of characters.

The higher the Entropy, the safer your password. Let's say you have a pool of 62 different characters, the same as the standard mode of this program, (26*2+10=62). For each characters you have an Entropy of 6 bits, (log(62) / log(2) = 5.95). You might ask if this is always true, and it is not. If you use a password of "Password" it would give you Entropy of 46 bits, but that password is still very easy to crack because of the way password crackers are deployed.

| Type | Pool Of Characters Possible |
|---|---|
| Lowercase | 26 |
| Lower & Upper Case | 52 |
| Alphanumeric | 36 |
| Alphanumeric & Upper Case | 62 |
| Common ASCII Characters | 30 |
| Diceware Words List | 7,776 |
| English Dictionary Words | 1,022,000 and counting |

As you can see, things change when you take into consideration the possibilities of how passwords are cracked. Let's take this for example. You create a password of JFKwasPresident. It has 15 upper and lower case characters, so figuring the Entropy would be: 15 * log(52) / log(2) = 85.5 bits.
BUT, looking at the Diceware Words List, that would be 3 words JFK was President. So to figure the Entropy for that would be: 3 * log(7776) / log(2) = 38.7 bits. As you can see, the Entropy dropped a lot because of how the password was put together, or built. Password crackers look for full words in passwords, and they also look for symbols and numbers in words too, such as Lo0k and $it, while better, they still look for things like that. So using Random Characters, Numbers and Symbols you get a stronger Password. With the computing power of today's CPU's, it wouldn't take as long to crack a password with just words and numbers and symbols than something that is totally random.
Something I came across while writing this program, and also looking over options while writing the code is this from Information Security at Stack Exchange.
"There are equations for when the password is chosen **randomly and uniformly** from a given set; namely, if the set has size *N* then the entropy is *N* (to express it in bits, take the base-2 logarithm of *N*). For instance, if the password is a sequence of exactly 8 lowercase letters, such that all sequences of 8 lowercase characters could have been chosen and no sequence was to be chosen with higher probability than any other, then entropy is $N = 26^8 = 208827064576$, i.e. about 37.6 bits (because this value is close to $2^{37.6}$).
Such a nice formula works only as long as uniform randomness occurs, and, let's face it, uniform randomness *cannot* occur in the average human brain. For human-chosen passwords, we can only do *estimates* based on *surveys* (have a look at that for some pointers).
What must be remembered is that entropy qualifies the **password generation process**, not the password itself. By definition, "password meter" applications and Web sites do not see the process, only the result, and uniformly return poor results (e.g. they will tell you that "BillClinton" is a good password). When the process is an in-brain one, anything goes.

(I generate my passwords with a computer, not with my head, and I encourage people to do the same.)”

That is the reason I wrote this program.

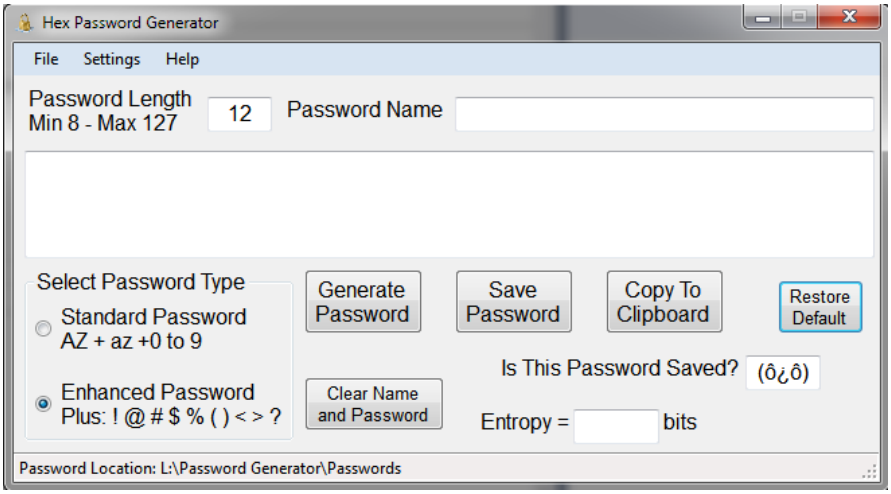
While there are other ways to come up with a Random Passwords, such as a Password Card found at: PasswordCard.org, I just found this a better way for me.
Since I didn’t encrypt the passwords generated, it’s best to store the passwords you generate on an external drive, such as a thumb drive or external hard drive that can be disconnected from your computer, and stored in a safe or another safe place.

**The Program:**

The Hex Password Generator will allow you to have a password from 8 to 127 characters in length. You never want anything less than 8, and 127 is overkill for now.

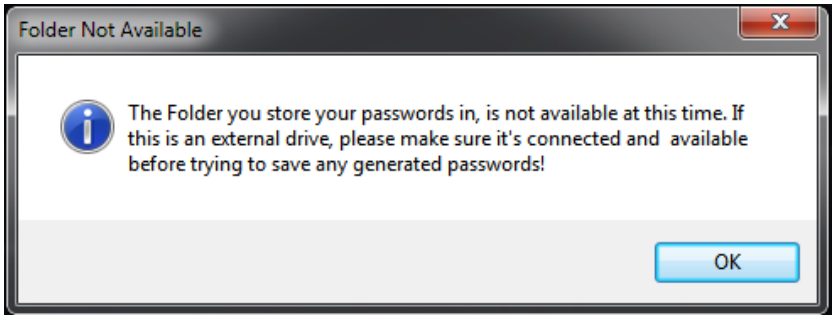
| # of Characters | Entropy Standard | Entropy Enhanced |
|---|---|---|
| 8 | 48 | 49 |
| 9 | 54 | 55 |
| 10 | 60 | 62 |
| 11 | 65 | 68 |
| 12 | 71 | 74 |
| 13 | 77 | 80 |
| 14 | 83 | 86 |
| 15 | 89 | 93 |
| 16 | 95 | 99 |
| 17 | 101 | 105 |
| 18 | 107 | 111 |
| 19 | 113 | 117 |
| 20 | 119 | 123 |

Here is a short list for this program that shows you the bits of Entropy for each mode. While the program itself has a built in Entropy meter, this gives you an idea of what that is all about.



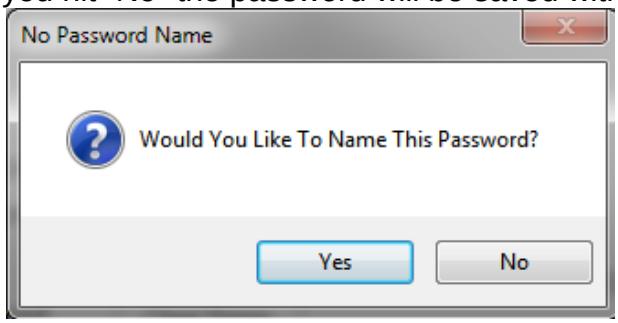

This is the Hex Password Generator. When you are ready to use the program, you start by giving the password a Name, as this is so you know what the password belongs too. Password Length can be changed manually, or set in the Settings Menu, (to be covered later in this document). You can select the Password Type either Standard or Enhanced, and then you can Generate your Password. You can Save your Password to the file, and also copy to your clipboard to paste it into the site you’re generating the password for. This will also tell you if the password has been saved or not, so that you know at a glance if you saved it.

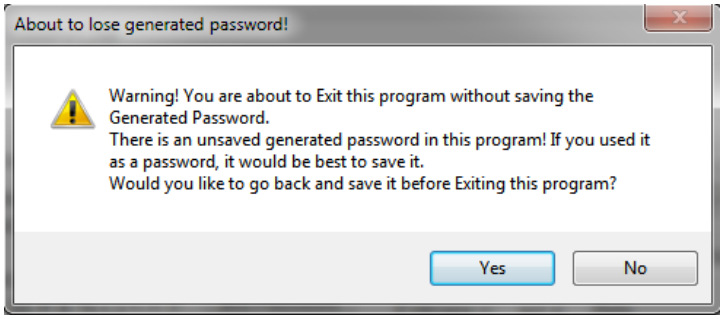
If by chance you have changed the password location to an external drive, and you do not connect the drive before you start the program, you will first get this informational popup.




When you hit OK, the program will start up as normal. You will still be able to generate passwords, but you can't save them until the external drive is connected and ready for use. The Password Location is listed at the bottom of the program in the Status Bar. (See Program picture above)

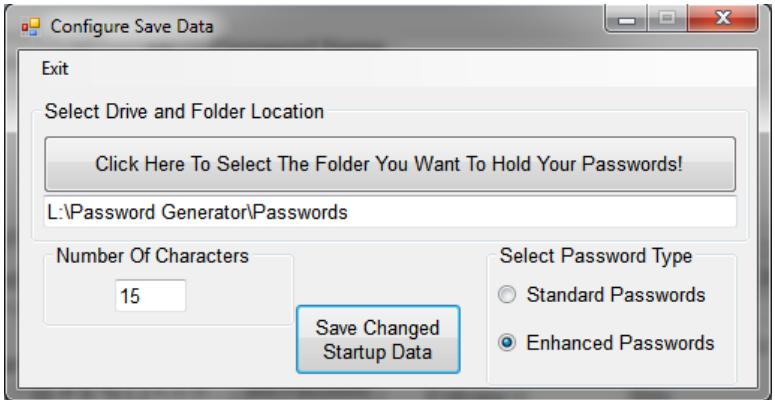
Because Passwords are of such importance, if you didn't name your password, the program will remind you, but you can still save the password without a name. If you hit "Yes", you will be taken back to the program without saving the password, and you can enter a name for the password, and if you hit "No" the password will be saved without a Name.



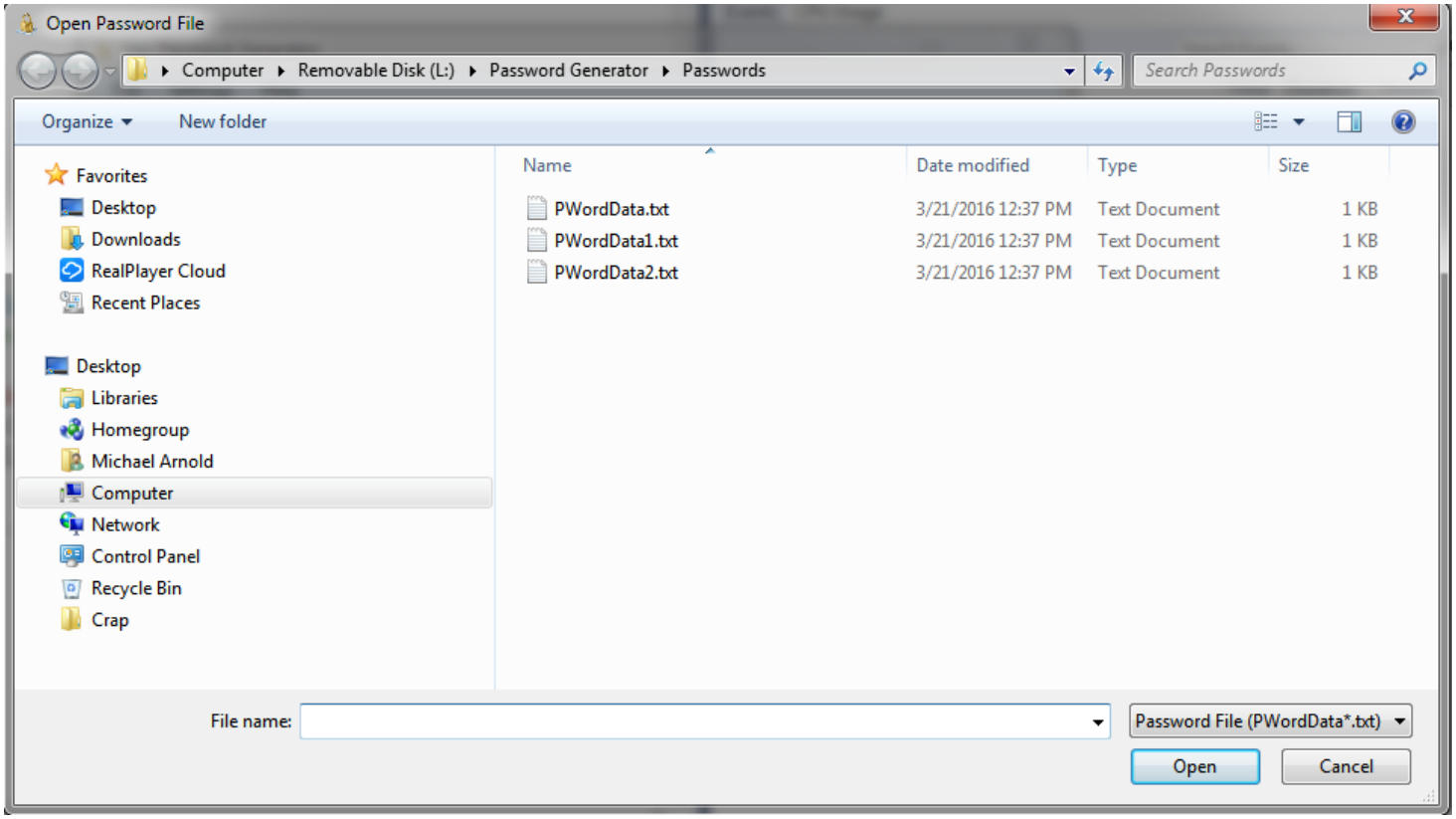

Likewise, if you generated a password, and didn't save it and try to Exit the program, you will get a warning letting you know you are about to lose your password. Hitting "Yes will take you back to the program and cancel the closing so you can save your password, and hitting "No" will Exit the program without saving your password.
It may seem that there are a lot of Question and Warnings popup windows in this program, but they are there only for the protection of the generated passwords, and so you don't lose a password you created and used with this program.

Configure:




Under the "Settings" on the menu bar, you will find "Configure". Clicking on Configure will bring up this Window. Here you can set your Password Folder location, and the number of characters, and what password type you want to start the program with. Each time you start up the program, it will use these settings. In the program itself, you can always change the password type and number of characters, but not the location of where you store the saved passwords. You don't have to worry about your password file when you change the location as your password file is moved to the new location when you hit the Save Changed Startup Data. If the current folder is on an external drive, and you try to change the location without having the drive connected and ready for use, you will get a warning letting you know about it. It will give you the option of changing the location without moving your old password file. That is not a good idea, and if you chose to change the location without moving the old password file, then you start a new password file with your next saved password. The main thing to remember is that if you use an external drive, you always want your password drive connected and ready to use when starting up this program. Things go much smoother when everything is in place and ready for new passwords.

**Open File:**




Last by not least, is the Open Password File. You will find that under File on the menu bar, and under File you will see “Open”. That brings up this Window. The only reason I had it open up this window instead of just opening up the PWordData.txt straight away, is that I tend to Number or Date old copies of my password files. That is, PWordData1.txt, PWordData-3-21-2016.txt and so on. You can add anything to the file name as long as the file name starts out with PWordData. Just remember that the program will ALWAYS save the new passwords to the PWordData.txt file. When I open things this way, all the PWordData files are visible for me to open as shown above. There is a reason I set the password data file name set in stone, and one of them is that when it moves your password file to a new location, it moves all the text files that start out with PWordData. You can have your password file in with a bunch of other text files, but in only moves the ones that start with PWordData. If by chance you misplace the file, it makes is much easier to search for a name you know, and not ones you made up to store your passwords in.
As a last minute change, I added a Restore Default button that changes things back to what you have stored for Number Length and Password Type.

I have always liked programs that are straight forward, and easy to use. If you keep your password file on an external drive, and remember to connect it first, and name your password, you will never see a popup window, but they are there just in case. ☺

If you find a bug, need help, or have a suggestion, you can email me at: hexsoft@hexappeal.net

Michael Arnold