

Wireless Password Recovery

USER MANUAL

Copyright (c) 2020 Passcape Software. All rights reserved.
Passcape Software

1.	Introduction	5
1.1	About the program	6
1.2	Features and benefits	6
2.	Program's interface	7
2.1	Overview	8
2.2	Project menu	9
2.2.1	Import	9
2.2.1.1	Import local hashes	9
2.2.1.2	Import hashes from external PC	10
2.2.1.3	Import from project/text files	11
2.2.2	Export	12
2.2.3	New	12
2.2.4	Open	12
2.2.5	Save	12
2.2.6	Save as	12
2.2.7	Close	12
2.3	Recovery menu	12
2.3.1	Run	13
2.3.2	Continue	13
2.3.3	Stop	13
2.4	Edit menu	13
2.4.1	Edit	13
2.4.2	Add	13
2.4.3	Delete	14
2.4.4	Reset passwords	14
2.4.5	Set Valid	14
2.4.6	Copy	14
2.4.7	Select	14
2.4.8	Search	14
2.5	Reports Menu	15
2.5.1	Password reports	16
2.5.2	Attack statistics	16
2.5.3	Miscellaneous statistics	17
2.5.4	Password-list analysis	17
2.6	Tools menu	18
2.6.1	Program access	19
2.6.2	Pass-o-meter	20
2.6.3	Password Checker	21

2.6.4	Hash Generator	21
2.6.5	Handshake constructor	23
2.6.6	PMKID constructor	24
2.6.7	MAC to vendor	25
2.7	Utils menu	25
2.7.1	Asterisk Password Revealer	26
2.7.2	Wordlist tools	26
2.7.2.1	Create new wordlist by indexing files	27
2.7.2.2	Merge wordlists	29
2.7.2.3	Wordlist statistics	30
2.7.2.4	Sort wordlist	32
2.7.2.5	Convert/compress wordlist	33
2.7.2.6	Compare wordlists	34
2.7.2.7	Additional operations	35
2.7.2.8	Index HDD sensitive areas	36
2.7.2.9	Extract HTML links	40
2.8	Settings menu	42
2.8.1	General settings	42
2.8.1.1	General options	42
2.8.1.2	Attack options	43
2.8.1.3	CPU settings	45
2.8.1.4	GPU settings	46
2.8.1.5	Password filters	47
2.8.1.6	GPU health monitor	48
2.8.1.7	GPU performance	49
2.8.1.8	Sound notifications	50
2.8.2	Attack Settings	50
2.8.2.1	Preliminary attack	50
2.8.2.2	Artificial intelligence attack	51
2.8.2.3	Fingerprint attack	53
2.8.2.4	Brute-force attack (exhaustive search)	56
2.8.2.5	Dictionary attack	57
2.8.2.6	Mask attack	61
2.8.2.7	Base-word attack	63
2.8.2.8	Combined dictionary attack	64
2.8.2.9	Pass-phrase attack	69
2.8.2.10	Hybrid dictionary attack	72
2.8.2.11	Batch attack	80
2.9	View menu	81
2.10	Themes menu	81
2.11	Help menu	82
2.12	Hardware Monitor	82
3.	Working with the program	84

3.1	Attacking WPA hashes	85
3.2	Attack comparison table	86
3.3	GPU FAQ	89
3.4	Online dictionaries	91
4.	License and registration	93
4.1	License agreement	94
4.2	Registration	95
4.3	Limitation of unregistered version	96
4.4	Editions of the program	96
5.	Technical support	99
5.1	Reporting problems	100
5.2	Suggesting features	100
5.3	Contacts	100
	Index	101

Introduction

1 Introduction

1.1 About the program

Welcome to **Wireless Password Recovery** - a utility for analyzing the security of your wireless networks and recovering WPA/WPA2 passwords. Wireless Password Recovery is the only software solution that employs the most advanced password recovery methods developed by our company.

Wireless Password Recovery is a powerful yet user-friendly utility designed primarily with security auditors, network administrators and forensic companies in mind although some home users will also likely find it very useful. The user interface is modern in design and there are no confusing features to get you bogged down with. This interface is also customizable in a variety of ways, allowing the program to more aptly cater to your individual requirements and preferences. If you want to recover lost passkeys, Wireless Password Recovery provides certain features which help to make this possible. It also takes advantage of your available hardware resources to recover passwords. Since recovering passwords can take a very long time, this is essential in order to make it practical. Wireless Password Recovery uses the full power of both your CPU and even your GPU to achieve this.

1.2 Features and benefits

Wireless Password Recovery has a number of advantages compared to similar solutions:

- Contemporary, customizable graphical interface.
- Built-in support for password search using both CPU and GPU power.
- Over 10 types of password recovery, many of which have been developed by and implemented in our company's products only.
- Advanced audit reports.
- Additional tools, including powerful utilities for creating and managing dictionaries. For example, you can create your own wordlists by indexing the files on your hard disk drive.
- 'On the fly' decryption of some WPA/WPA2 hashes.
- Loading password hashes from non-working operating systems, various network dumps, network sniffer logs, etc.
- Dictionary recovery supports text wordlists in ASCII, UNICODE, UTF8, PCD, RAR and ZIP.
- Great choice of online wordlists for dictionary attacks (near 2 GB).
- Some of the program's functions, such as word mutation, are unique. Total number of mutation rules exceeds one hundred and fifty. No similar application carries this!

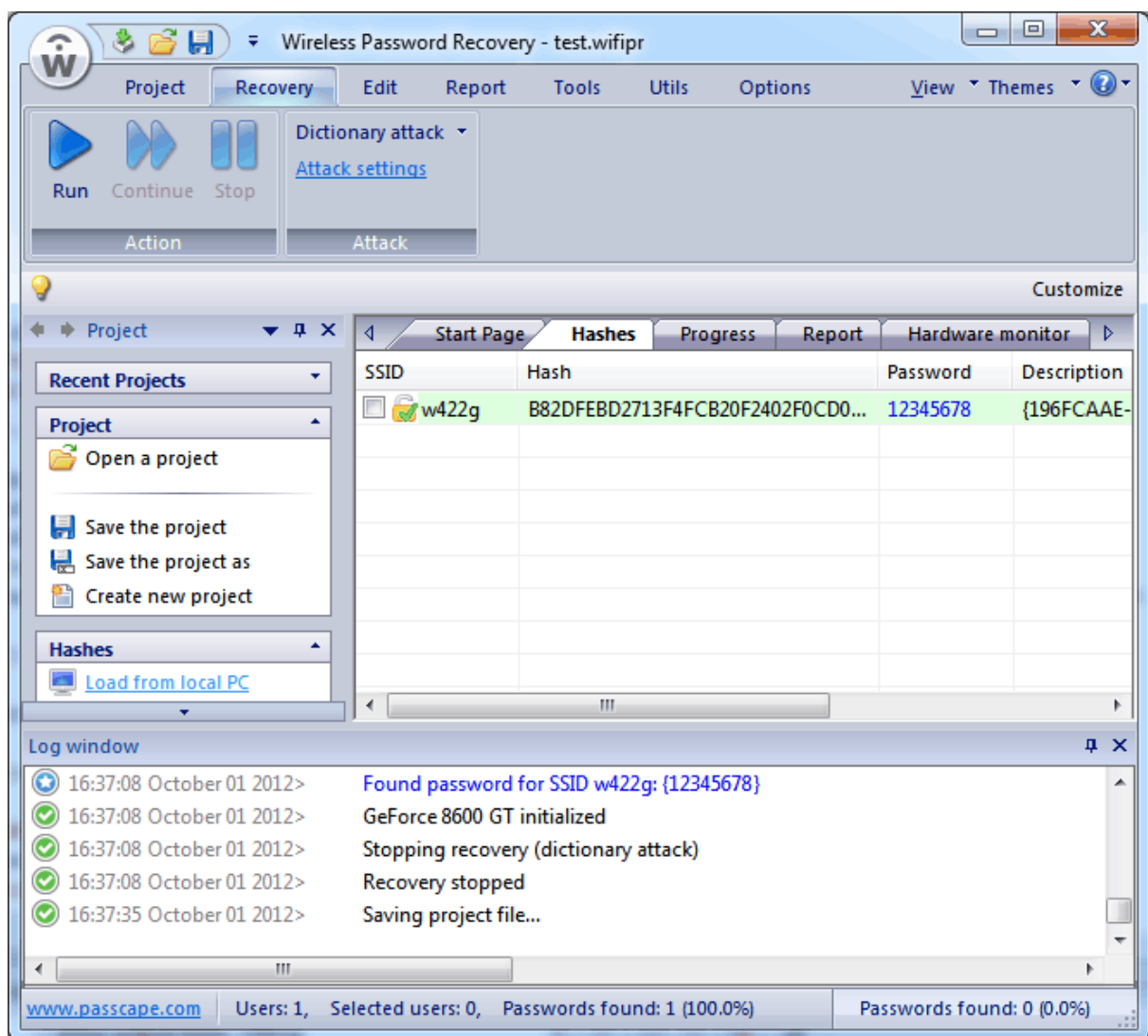
Program's interface

2 Program's interface

2.1 Overview

The program's interface is made in the form of the SDI architecture, i.e. it allows working with only one project at a time. The program's operation can be conventionally divided into 4 stages:

1. Creating a project
2. Importing (loading) password hashes into the project. Editing the hashes: deleting, adding, selecting, etc.
3. Recovering the hashes. Includes selecting, configuring and launching the selected one or several attacks.
4. Analyzing the results.



The entire interface can be conventionally divided into several components:

- Menu Bar
- Information Bar - for displaying brief information texts - like tips, warnings, etc.
- Task Bar - duplicates and compliments the menu bar, providing quick access to the most common operations. Consists of three parts:

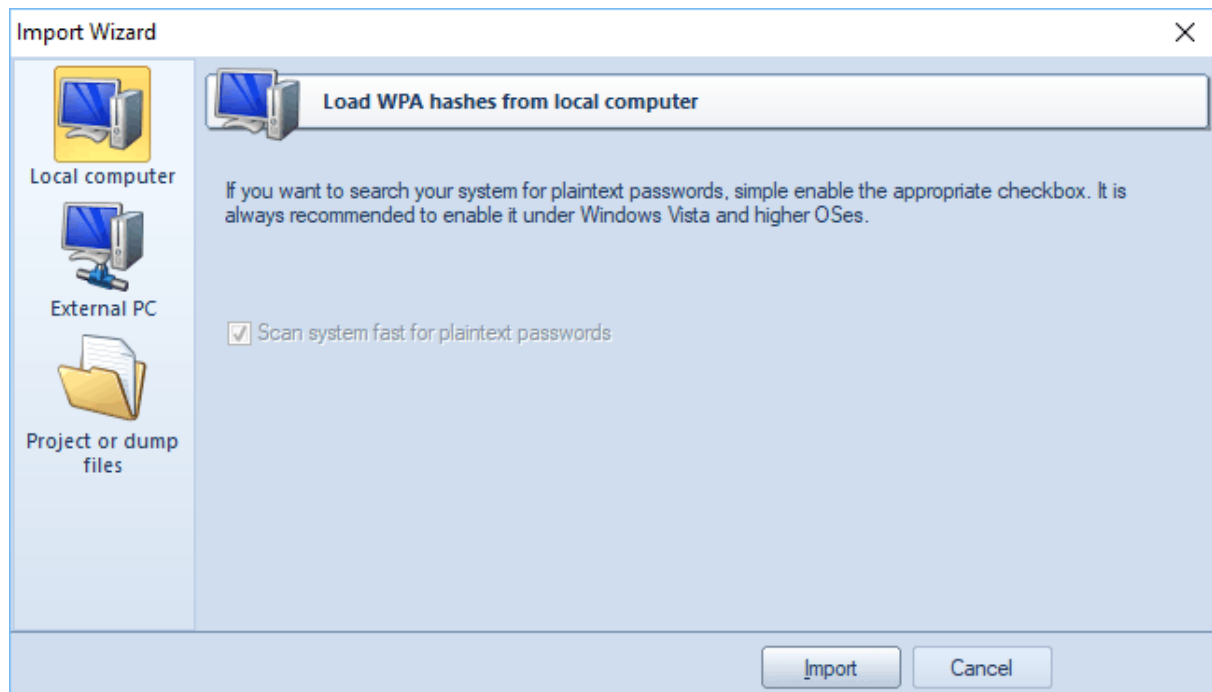
- Project - includes the main operations over project - like opening, closing, creating a new project, and importing hashes.
- Hash Editor. Duplicates the most common editing operations.
- Tools - includes a clock, calendar, and calculator.
- Main Window - bears the main burden and consists of 5 parts. The first tab is the welcome window. The second tab contains the list of hashes to be analyzed and recovered. Then there goes a tab with the current attack state (progress) indicator and a tab with the statistics and reports. And finally - a tab with the hardware monitor.
- Log Window - displays information on the current state of the application, current operation, etc. The program's log can be copied to clipboard or saved to a file (right-clicking opens the corresponding menu).
- Status Bar is designed for informational purposes.

2.2 Project menu

2.2.1 Import

Wireless Password Recovery offers a broad range of options for loading WPA-PSK hashes into the program.

2.2.1.1 Import local hashes



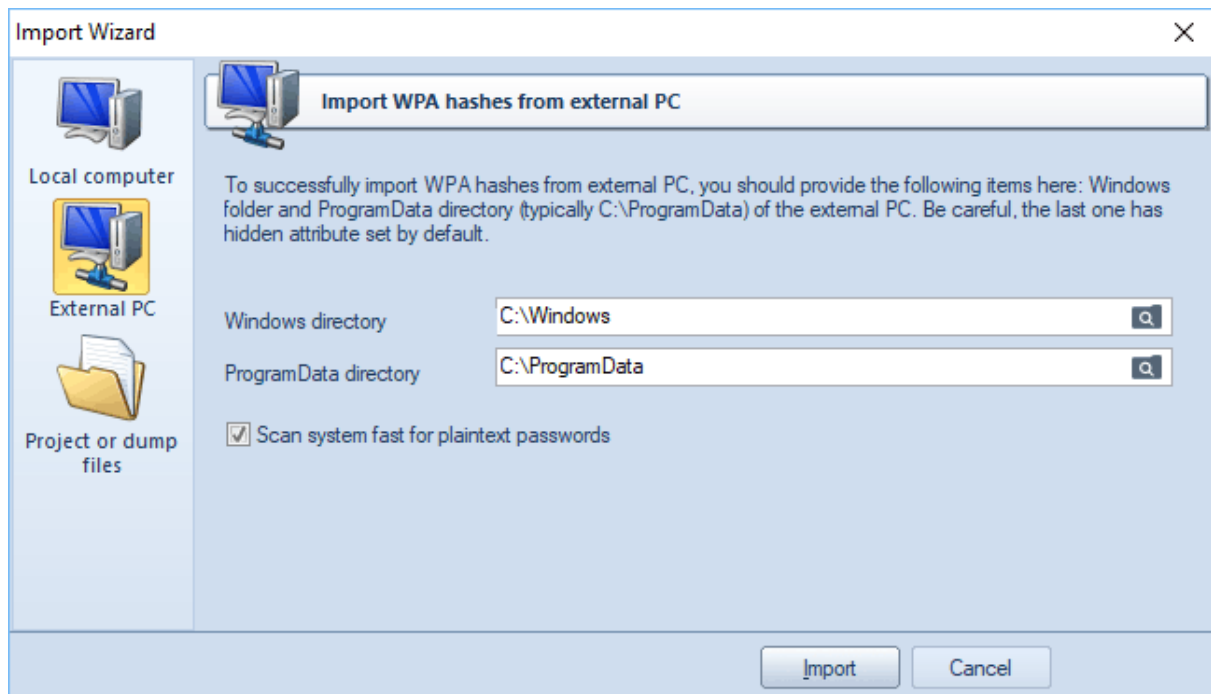
Import hashes from the local computer - the most preferable method to extract WPA passwords in Windows, as it implies the deepest overall analysis of the system and the passwords. Besides the hashes that are imported from the local computer can undergo the sophisticated *Intelligent attack*, which allows to relatively quickly recover the passwords to some WPA-PSK handshakes.

Importing local hashes runs well regardless to where the hashes are localized: in registry (Windows XP) or in the disk (Windows Vista and higher OSes). Note that in most cases the program can decrypt plain-text passwords for WPA-PSK accounts instantly (Windows Vista + only).

The only option here allows you to bypass the plain-text password recovery.

The local import functionality requires administrative privileges.

2.2.1.2 Import hashes from external PC



Importing hashes from external PC. Wireless Password Recovery can read WPA-PSK hashes from an external computer. For example, from other operating system's hard drive.

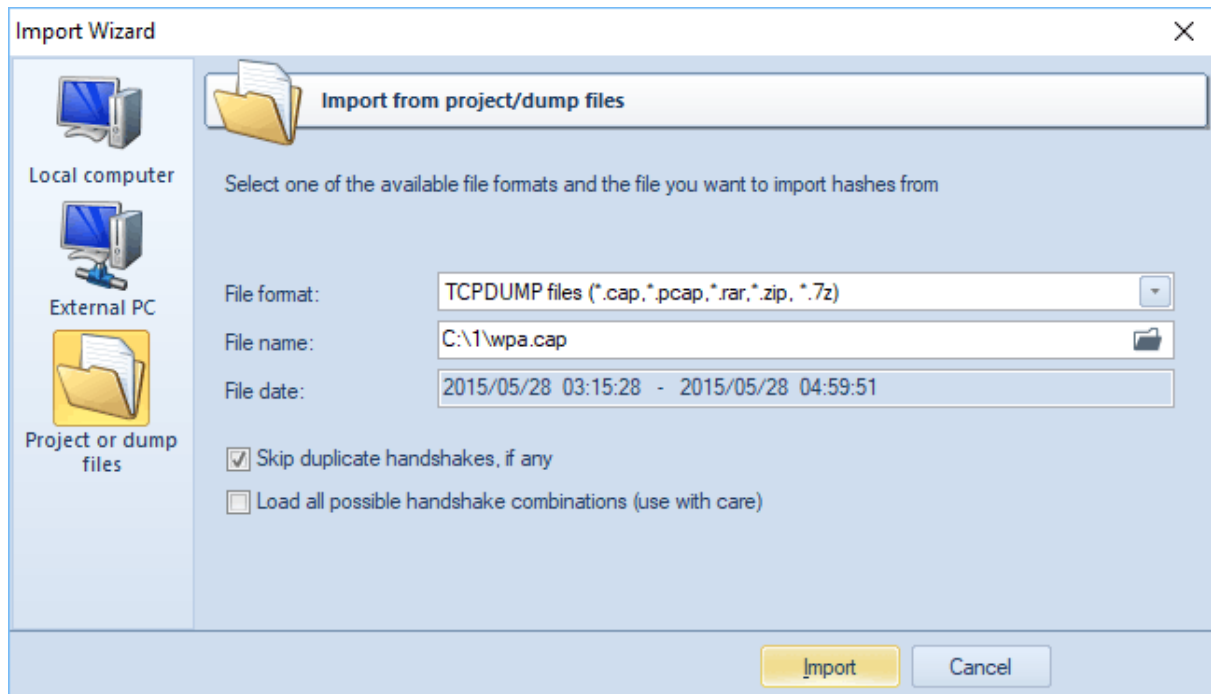
To do so, just specify the path to **Windows** and **ProgramData** folders on the external PC or on the connected HDD. Depending on the settings and type of the operating system, the program can automatically detect and use the following folders of the target OS:

- %WINDOWS%\System32\Config. The following registry files are processed here: SAM, SYSTEM, SECURITY and SOFTWARE.
- User profile for Master Key analysis.
- System account profile.
- System account with system Master Keys.
- System Vault storage.
- Other folders.

You can skip the system analysis and text password lookup by clearing the respective check box.

This option does not require administrative privileges.

2.2.1.3 Import from project/text files



Finally, you can load WPA-PSK hashes to project by importing them from other applications or from network traffic dumps. The following formats are supported:

- **TCPDUMP (*.cap, *.pcap, *.tcpdump, *.txt)** - de-facto standard among packet sniffers, designed for intercepting, parsing and storing TCP IP and other packets sent over network. The following popular solutions can export network packets to TCPDUMP file: airodump-ng, CommView for Wi-Fi, OmniPeek, Sniffer Global, Wireless Snif, WireShark, etc. Network packets to be loaded must contain authentication data (authentication handshake) between access point and client.
- **Tamosoft CommView dump file (*.ncf)** - this file format is used in a popular application CommView for Wi-Fi.
- **WPA-PSK hashes in the HashCat (*.hccap, *.hccapx)** format is used in Hashcat.
- **JohnTheRipper WPA-PSK dump (*.txt)**. This format is used by JohnTheRipper tool.
- **Elcomsoft EWSA project files (*.wkp)**. This format is used by Elcomsoft Wireless Security Auditor.
- **WPA-PSK PMK hash files by InsidePro (*.txt)**. This format is similar to the text file PWDUMP and used in InsidePro. It may have problems with some SSID names.
- **WIFIPR export files (*.wifiprdump)**. This text container is a native format of the Wireless Password Recovery application and used for exporting and importing WPA-PSK hashes. Properly handles non-standard SSID names.
- **Textual files with PMKID** - files in hashcat format that holds PMKID entries.

TCPDUMP and CommView dumps have also support for PMKID in addition to regular handshakes.

Additionally the program has 2 options (available for some formats only) for more accurate load:

- **Skip duplicate handshakes, if any.** If set, the option guarantees to load only unique handshakes in a file scope. Duplicate items will be ignored. It is recommended this option to be set on always.
- **Load all possible handshake combination.** Sometimes it is very difficult to extract valid handshakes out of a capture file, especially when the WPA-PSK 4-way handshake is suppressed by and massively alternate with deauthentication packets or when the source file is broken. Unfortunately,

there's no reliable way to determine whether a handshake is valid or not. When this option is set, the program guaranteed generates at least one valid handshake, if one exists. The backside of this option is that hundreds or even thousands trash handshakes might be created alone with valid items. Thus, it is highly recommended to use it in conjunction with [multimode](#) (simultaneous recovery all handshakes with the same SSID) only.

Upon importing hashes, the program automatically launches the [Preliminary recovery](#) on the first valid hash. This action is optional and can be disabled in the general settings. The default state is enabled.

2.2.2 Export

All project hashes, along with the settings, are stored in the project file (*.wifipr); however, for the sake of greater flexibility and compatibility with other software, the program can export hashes to a WIFIPRDUMP file.

2.2.3 New

Saves current project and creates a new one.

2.2.4 Open

Loads/opens a new project. The application's projects have the *.wifipr extension and contain program settings and hashes. However, for speeding up the search speed, the program stores the current state of the attack in a separate file progress.ini.

2.2.5 Save

Saves current project. It is recommended to save critical projects from time to time.

2.2.6 Save as

Saves current project under a different name (renames it).

2.2.7 Close

Closes current project.

2.3 Recovery menu

This menu item allows selecting and launching an attack. Take a note that before actually launching the recovery you must have selected/marked the necessary hash. You can do that through the **Edit-Select**

menu. Launching the recovery assumes that you have also made all the required settings (on the **Options-Attack Options** menu).

2.3.1 Run

Launches selected attack. When the attack is running, all other items on the menu are disabled. Please note that when the recovery is over, the program runs a special mutation and password analysis routine over the found passwords. This option is enabled by default, but it can be disabled in the general settings.

2.3.2 Continue

Resumes attack from the last stored point. Please remember that the last stored point is automatically erased when changes are made to the attack's options.

2.3.3 Stop

Pauses current attack.

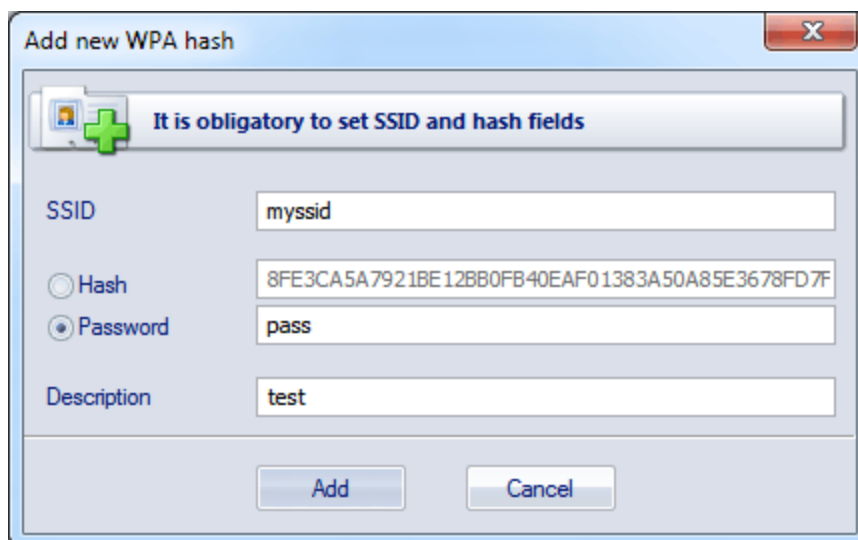
2.4 Edit menu

The Edit menu is available only when the 'Hashes' tab is active; it includes four items: Edit, Copy, Select, and Search.

2.4.1 Edit

Selecting this item opens the dialog where you can manually edit one of available fields for the selected hash.

2.4.2 Add



Add new WPA hash

It is obligatory to set SSID and hash fields

SSID: myssid

☐ Hash: 8FE3CA5A7921BE12BB0FB40EAF01383A50A85E3678FD7F

☒ Password: pass

Description: test

Add Cancel

This item allows adding items manually.

2.4.3 Delete

Deletes entries from list: highlighted (i.e. the one being under the cursor), marked or all at once.

2.4.4 Reset passwords

Drops all found passwords for list.

2.4.5 Set Valid

This option allows to set valid state for all hashes in the list.

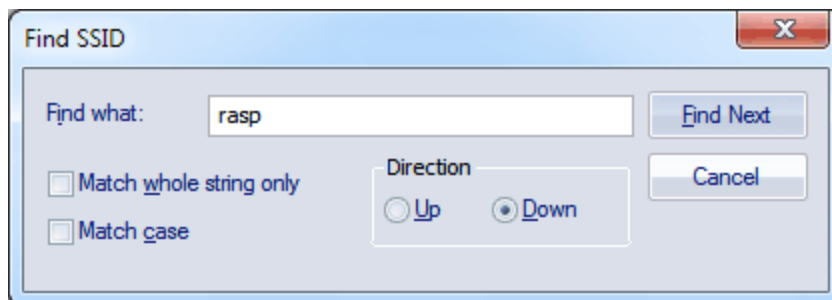
2.4.6 Copy

Copies current (highlighted) entry to Windows clipboard. Copies only the selected portion of the entry, not the entire entry. For example, SSID or the found password.

2.4.7 Select

Selects a hash to be recovered (ones with checkbox option is on). If during the attack the password for the selected hash is found, the checkbox will be automatically cleared, and the record will be marked green.

2.4.8 Search

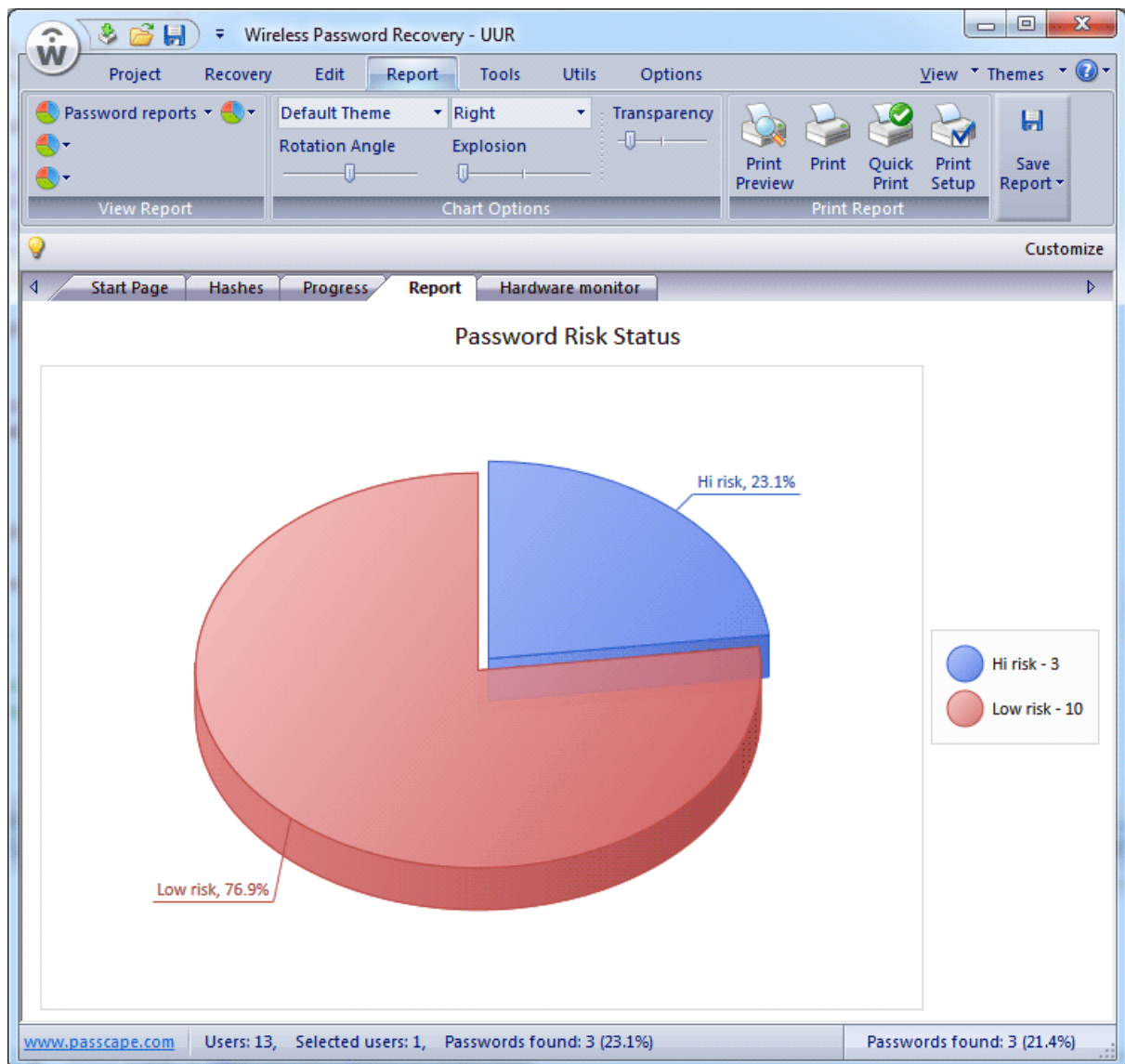


When the number of entries exceeds thousands, finding a specific entry often takes quite a bit of an effort. To make the job easier, the program offers the search of two types: searching a specific field - e.g., SSID - and quick-searching of serial entries. In the latter case, the program scans the entire entry, character by character.

2.5 Reports Menu

You can create, print or save one of the program's reports here. The following reports are available:

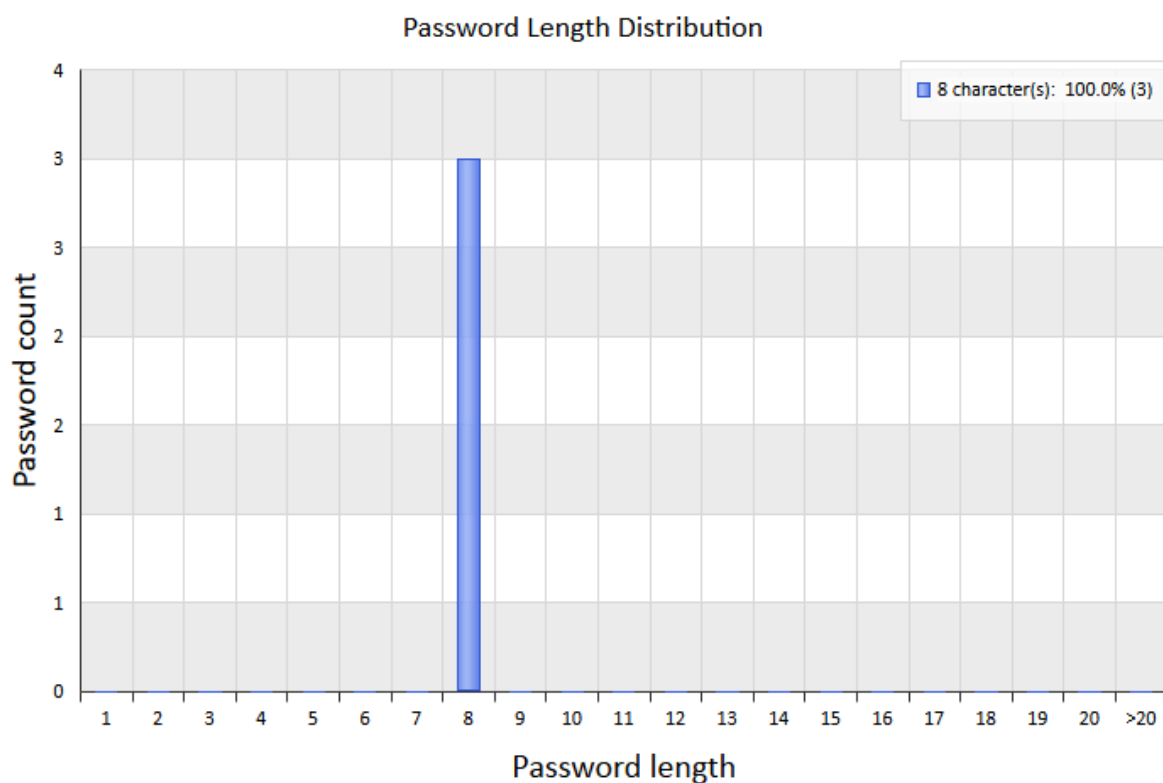
- [Password reports](#)
- [Attack statistics](#)
- [Miscellaneous statistics](#)
- [Password-list analysis](#)



2.5.1 Password reports

The following reports are available here:

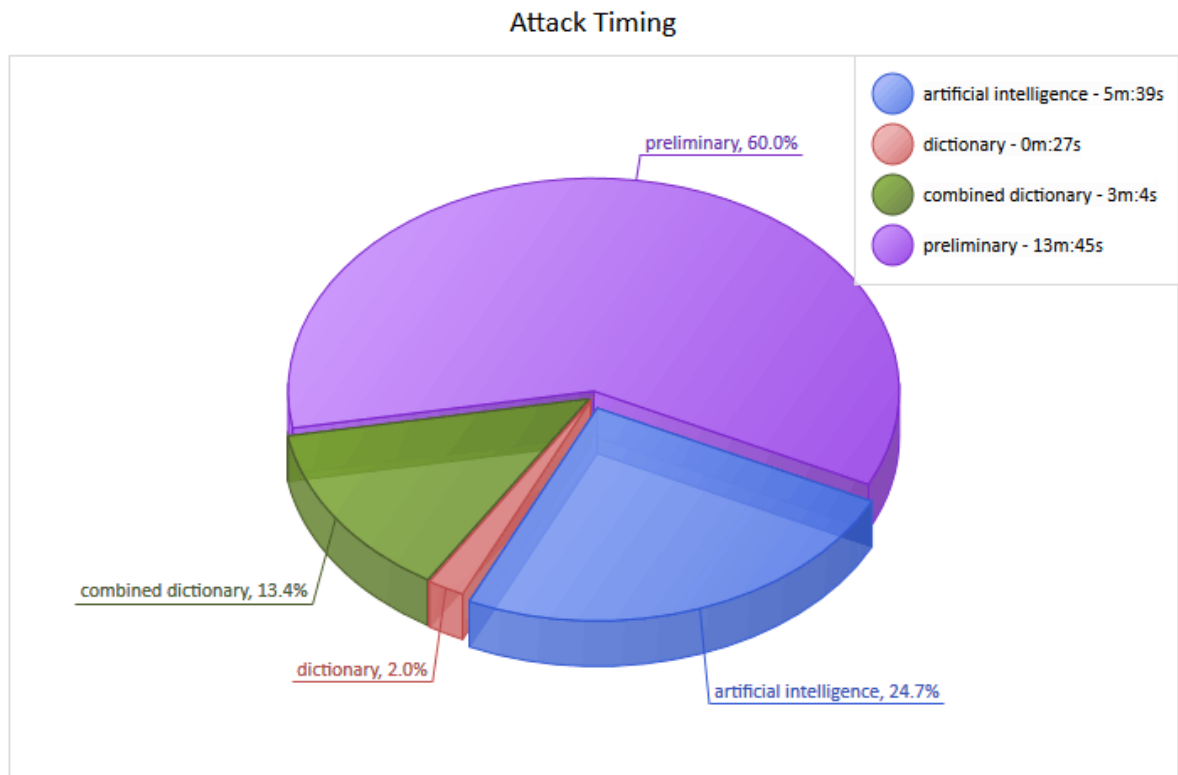
- **Password risk status** - displays empty, found, and not recovered passwords
- **Password complexity** - reports the number of passwords and various character sets being audited
- **Password length distribution** - shows overall length of the broken passwords
- **Password recovery time** - time took to crack a certain password(s). Most vulnerable passwords are marked in red palette.
- **Recovered vs unbroken passwords** - displays the number of discovered and not-found passwords
- **Passwords found** - shows a bit detailed report on found passwords



2.5.2 Attack statistics

Attack statistics includes the following items:

- **Preferred attack** - statistics on number and type of used attacks.
- **Attack time** - analysis of time spent on each attack.
- **Attack efficiency1** - efficiency analysis: time spent vs. passwords found during attack ratio.
- **Attack efficiency2** - efficiency analysis: overall efficiency for each attack.



2.5.3 Miscellaneous statistics

Some additional stuff like:

- **CPU speed** - password recovery speed comparison (for brute-force attack).
- **GPU speed** - shows and compares password recovery speed for your GPU device.
- **Cracked users** - displays the number of cracked entries. The full list of cracked entries can be saved to text file additionally.
- **Cracked users and passwords** - displays the list of cracked entries and passwords.

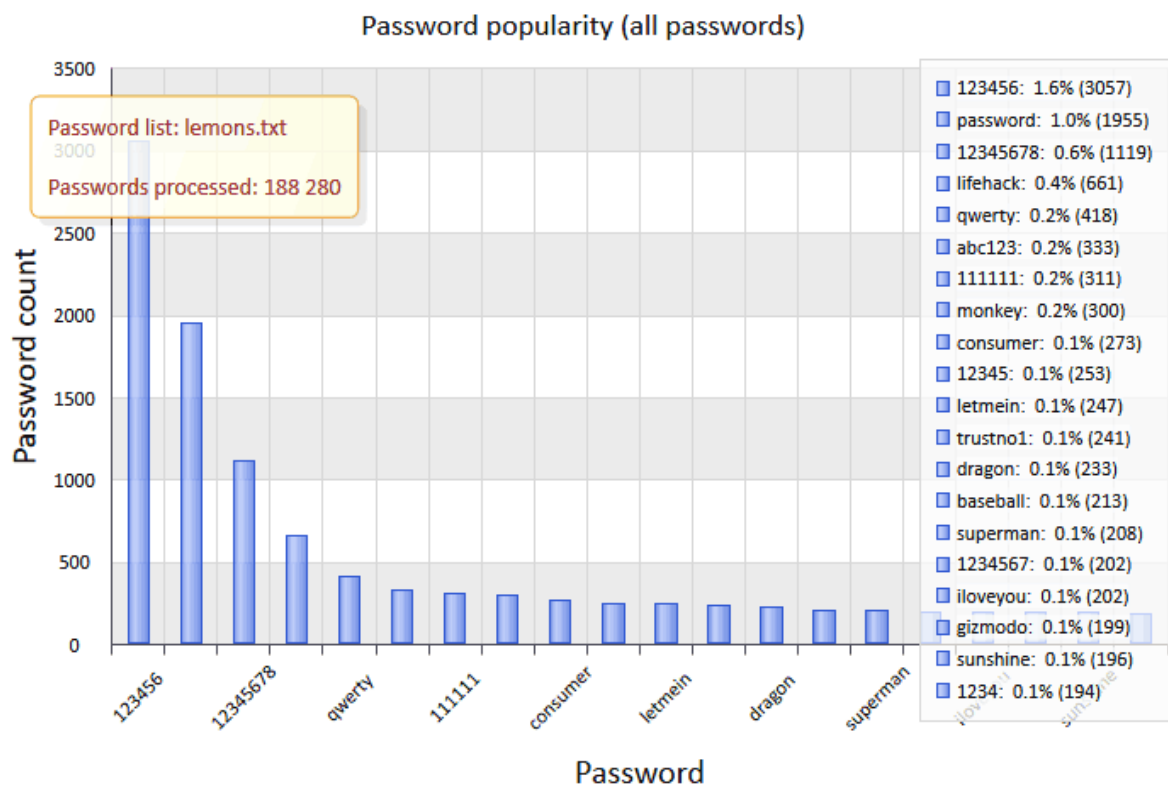
2.5.4 Password-list analysis

Password-list reports display various statistics and perform a deep analysis for input wordlists. As a source wordlists you can use, for example, the list of passwords recovered by the program. You can generate reports for all words of the input list as well as for passwords with a certain length only. The following reports are available here:

- **Password length distribution** - displays the overall length of the password in a given wordlist.
- **Password uniqueness** - this report shows unique against identical passwords chart.
- **Password popularity** - displays the most popular passwords and their percentage of the total number of passwords.
- **Password format** - statistics on the 20 most popular formats. The password format is defined by a character mask. For example, the DDUUUUDD mask corresponds to passwords consisting of two

leading and two trailing digits, with four capital letters in the middle. You can save popular password masks into a file so that you can easily use them in a mask-based attack later.

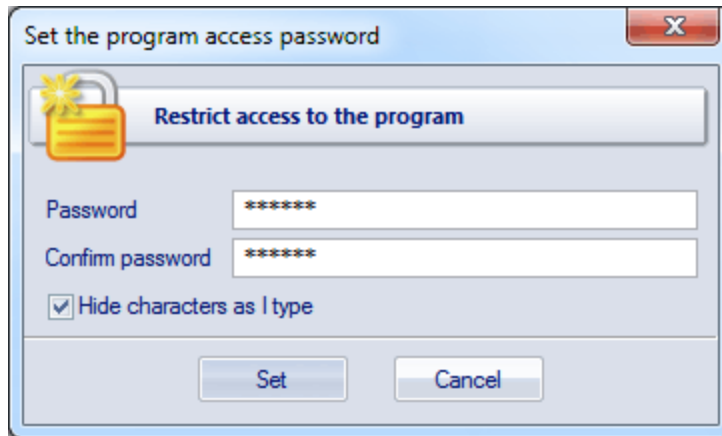
- **Character set exclusivity** - this report displays the number of passwords consisting of one unique character set and the percentage of these passwords to those consisting of several ones.
- **Character set diversity** - the percentage ratio of passwords consisting of one, two, or more character sets.
- **Character sets** - lists all charsets the input passwords are made of.
- **Character set ordering** - the most popular password templates corresponding to the character set order. For example, the *digit-string-special* template includes the following passwords: 123password!@#, 1ove****, and 12monkey^, etc.
- **Character frequency** - statistics on the frequency of characters in the input words. The 20 most frequent characters are displayed.
- **Unique characters** - the 20 least frequent characters.
- **Frequently used leading characters** - statistics on the most frequent combinations of 1 to 3 characters in the beginning of words.
- **Frequently used trailing characters** - statistics on the most frequent combinations of 1 to 5 characters in the end of words.
- **Frequent combinations** - the 20 most frequently used combinations of 4 to 8 characters.



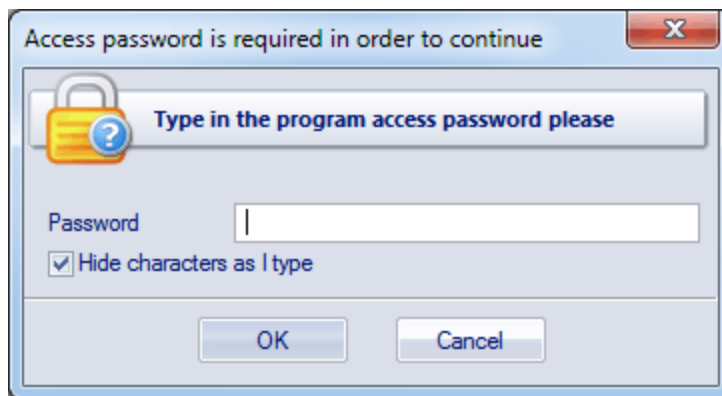
2.6 Tools menu

The Tools consists of two parts: tools for controlling access to the application and tools for working with passwords.

2.6.1 Program access



If anyone besides you can access your computer or account, you can password-protect the application. In this case, when starting the program, user will be prompted for the password, and the application will fail to continue unless the valid password is supplied.



2.6.2 Pass-o-meter

Pass-o-meter

Check the quality of your password

The password quality depends on its length and complexity. The most of WPA passwords consist of at least 8 characters. Thus cracking such passwords may take days or even years. Note however that if the password contains a common word or a phrase, it should be always considered as weak.

Source password

Type in your password:

Charset length:

Recovery configuration

Recovery speed: passwords per second

Hardware:

Password quality

Password quality:

Time to crack: 168d 13h:35m:19s

[Share your benchmarks](#)

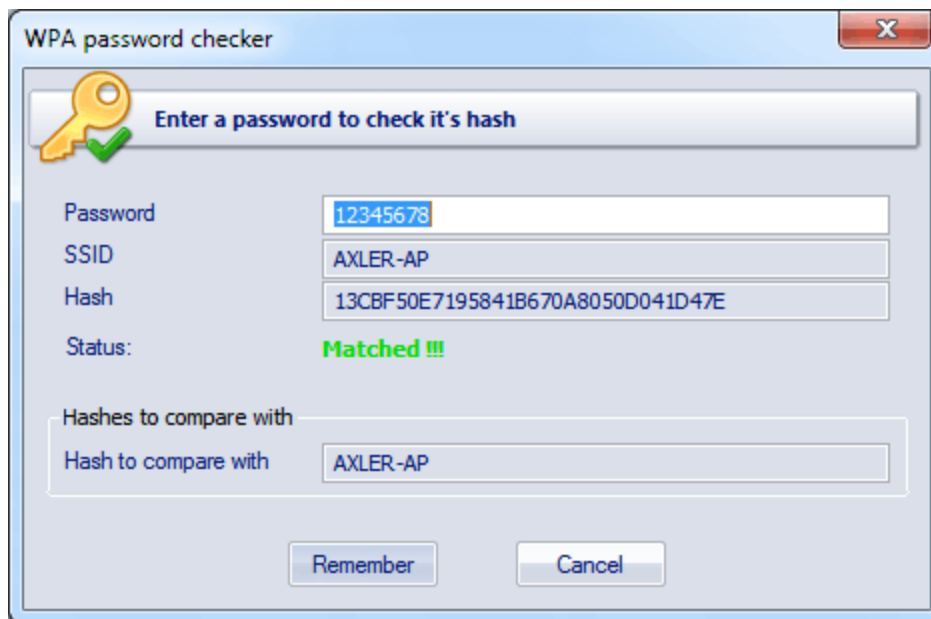
A tool for measuring password strength. During its first start, the program asks you to test your computer's performance. To check the quality of a password:

- Enter the password in the corresponding field.
- Select the computer type. *'This computer'* indicates your computer's search speed.
- If you want to test the speed of your GPU device, select *'This computer (GPU)'* from *'Recovery speed'* combo box and click *Adjust* button. Note, that you can do it from *Reports* menu as well.

You will see the quality of your password and time required to break it.

We would be grateful if you let us know the speed you've reached on your PC.

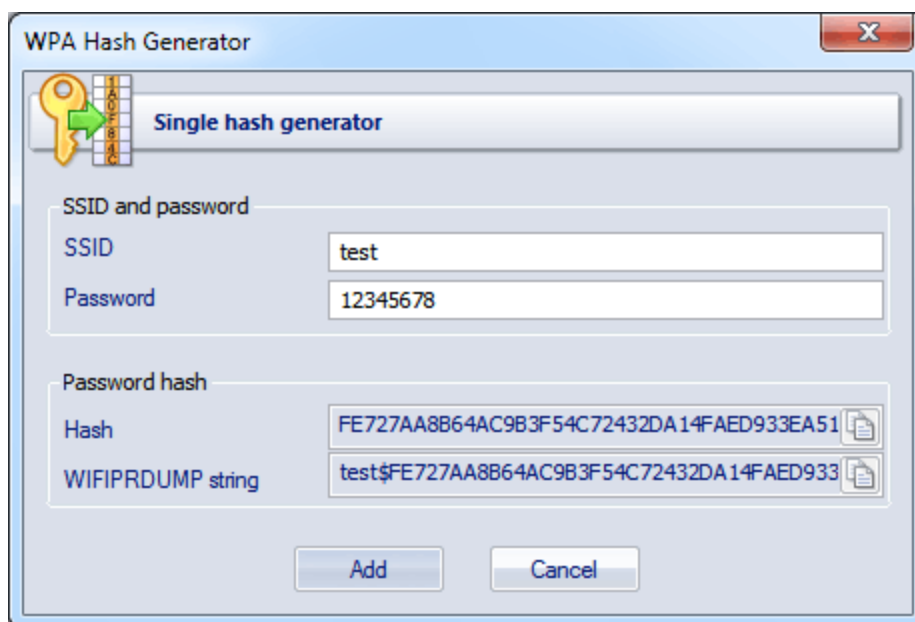
2.6.3 Password Checker



The screenshot shows a dialog box titled "WPA password checker". It features a key icon and a green checkmark. The main instruction is "Enter a password to check it's hash". Below this, there are four input fields: "Password" (containing "12345678"), "SSID" (containing "AXLER-AP"), "Hash" (containing "13CBF50E7195841B670A8050D041D47E"), and "Status:" (displaying "Matched !!!" in green). At the bottom, there is a section "Hashes to compare with" with a sub-field "Hash to compare with" (containing "AXLER-AP"). Two buttons, "Remember" and "Cancel", are at the bottom right.

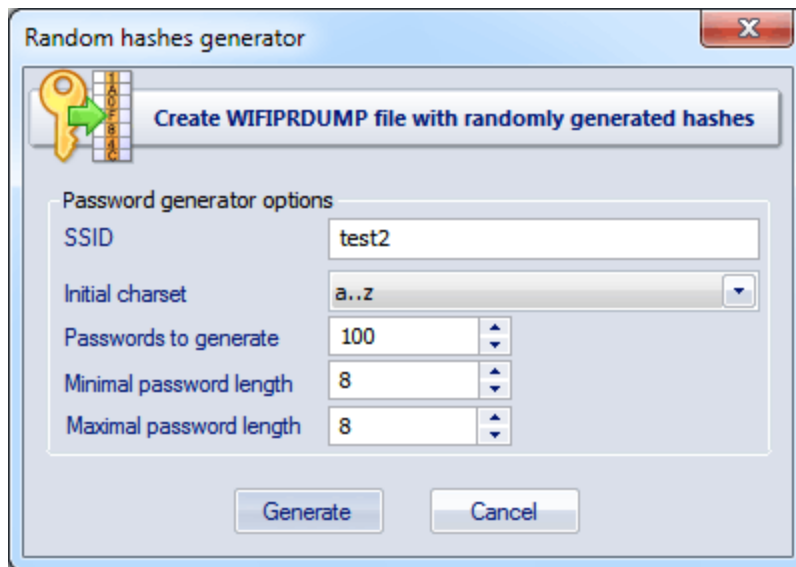
This tool allows checking the password of a selected hash manually. The tool is often necessary for validating certain hashes.

2.6.4 Hash Generator

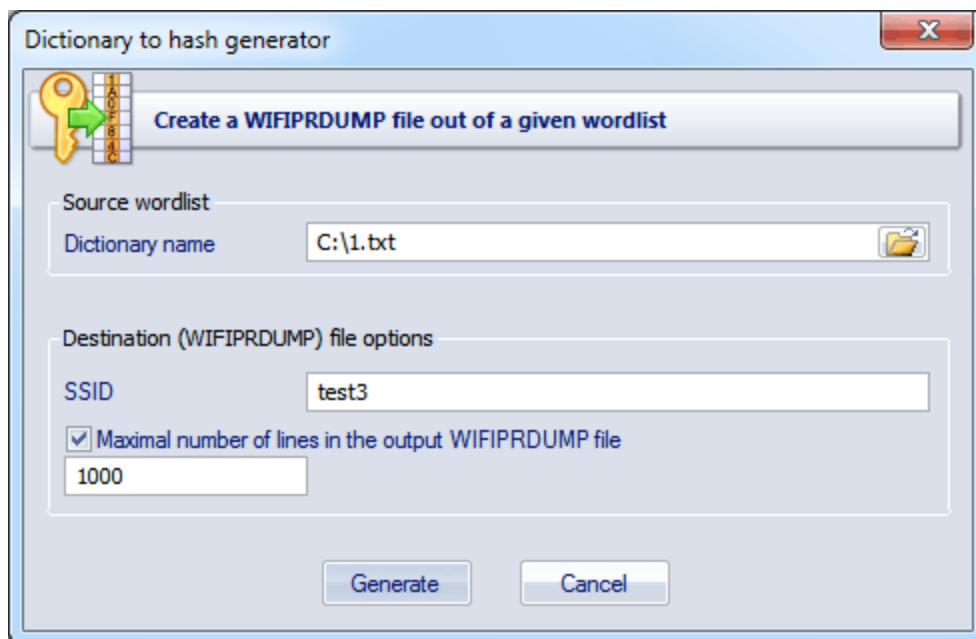


The screenshot shows a dialog box titled "WPA Hash Generator". It features a key icon and a green arrow. The main instruction is "Single hash generator". Below this, there are two sections: "SSID and password" with fields for "SSID" (containing "test") and "Password" (containing "12345678"), and "Password hash" with fields for "Hash" (containing "FE727AA8B64AC9B3F54C72432DA14FAED933EA51") and "WIFIPRDUMP string" (containing "test\$FE727AA8B64AC9B3F54C72432DA14FAED933"). Each hash field has a copy icon. Two buttons, "Add" and "Cancel", are at the bottom right.

The single-hash generator allows to quickly generate a test entry for a specified passwords and add it to the hash list.

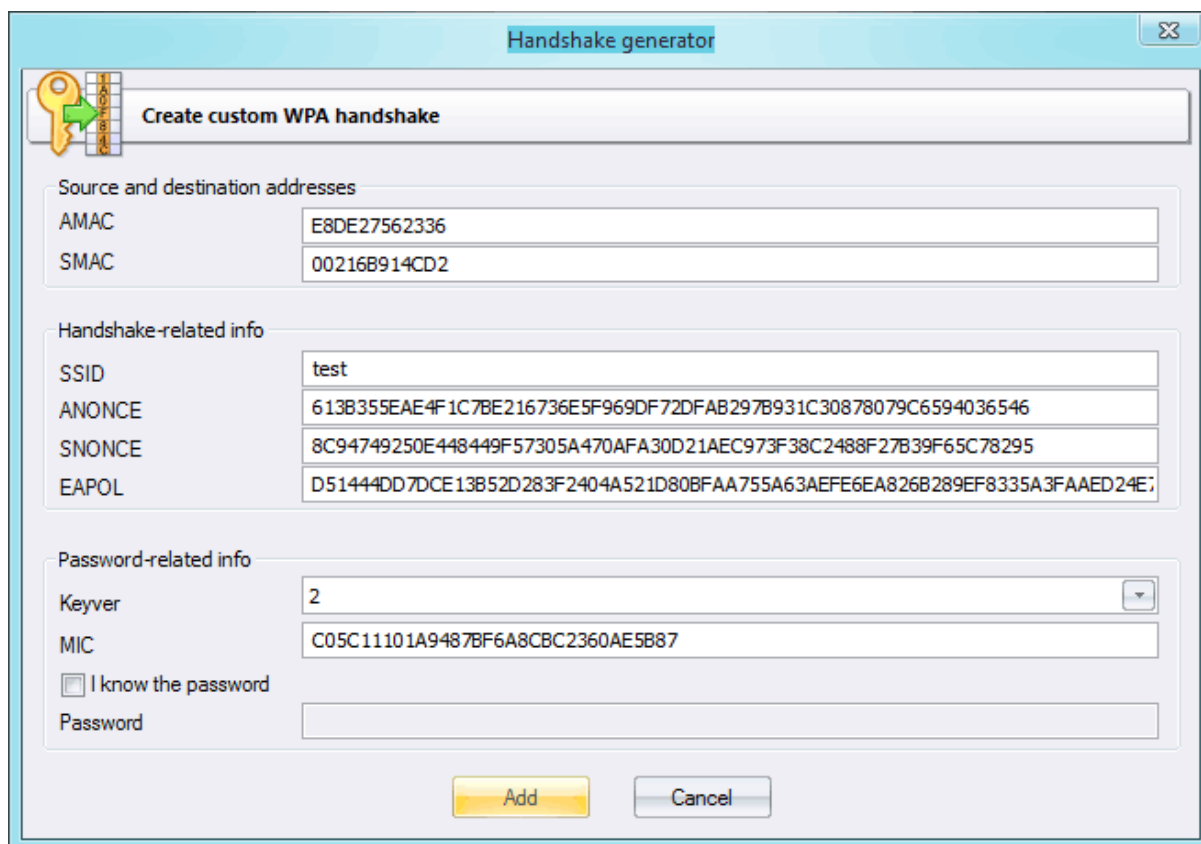


If you want to create a WIFIPRDUMP file with a certain number of randomly generated hashes, you will need to use this tool. In the hash creation dialog, select the minimum and maximum length, character range, and the total number of passwords to be generated.



Sometimes it might be necessary to create a test WIFIPRDUMP file with hashes from existing wordlist. In that case, very convenient is to use the dictionary hash generator. The interface of the tool is rather simple and consists of a dialog, at the top of which you should specify the list of source words for the generator. Using those words, the program will generate the hashes. At the bottom, specify the SSID and limit the number of hash records you need.

2.6.5 Handshake constructor



The image shows a software window titled "Handshake generator" with a close button in the top right corner. Inside the window, there is a section titled "Create custom WPA handshake" with a key icon. Below this, there are three main sections: "Source and destination addresses", "Handshake-related info", and "Password-related info".

Source and destination addresses

AMAC	E8DE27562336
SMAC	00216B914CD2

Handshake-related info

SSID	test
ANONCE	613B355EAE4F1C7BE216736E5F969DF72DFAB297B931C30878079C6594036546
SNONCE	8C94749250E448449F57305A470AFA30D21AEC973F38C2488F27B39F65C78295
EAPOL	D51444DD7DCE13B52D283F2404A521D80BFAA755A63AEFE6EA826B289EF8335A3FAAED24E

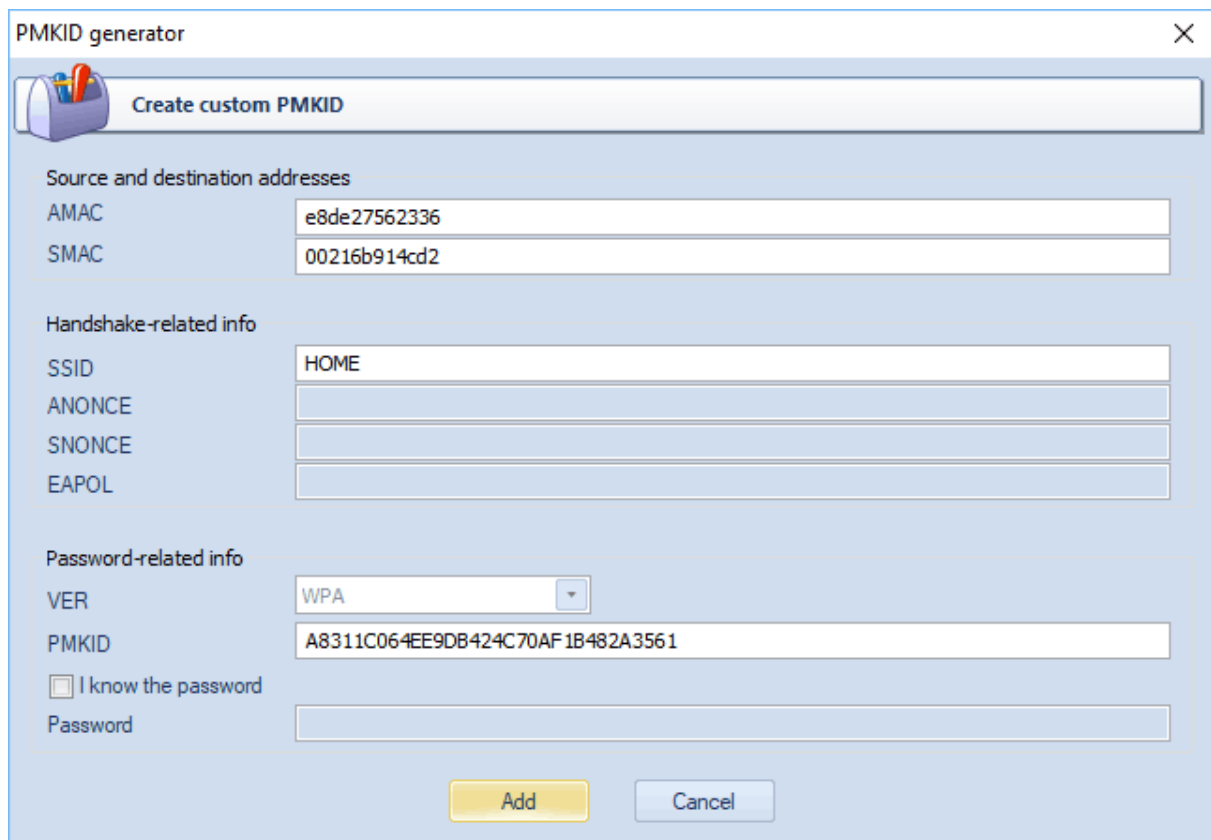
Password-related info

Keyver	2
MIC	C05C11101A9487BF6A8CBC2360AE5B87
<input type="checkbox"/> I know the password	
Password	

At the bottom of the window, there are two buttons: "Add" and "Cancel".

The handshake generator tool is used to create new recovery item out of raw handshake data. For example, you can use it to analyze WPA-PSK data caught with wireless sniffer or to construct new handshake out of a damaged dump file.

2.6.6 PMKID constructor



The screenshot shows a window titled "PMKID generator" with a close button (X) in the top right corner. Below the title bar is a toolbar with a folder icon and a button labeled "Create custom PMKID". The main area is divided into three sections: "Source and destination addresses", "Handshake-related info", and "Password-related info".

Source and destination addresses

AMAC	e8de27562336
SMAC	00216b914cd2

Handshake-related info

SSID	HOME
ANONCE	
SNONCE	
EAPOL	

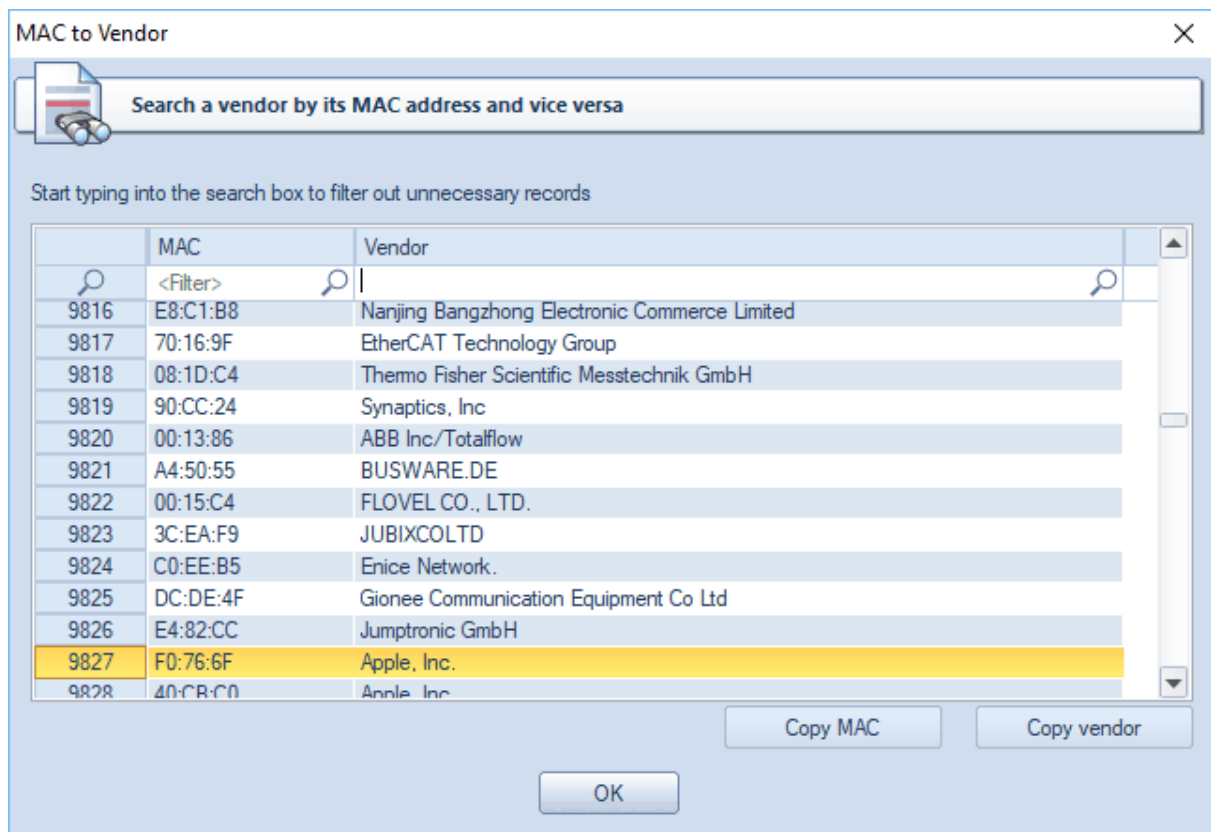
Password-related info

VER	WPA
PMKID	A8311C064EE9DB424C70AF1B482A3561
<input type="checkbox"/> I know the password	
Password	

At the bottom of the window are two buttons: "Add" (yellow) and "Cancel" (blue).

This tool is used for manual PMKID creation out of custom data. You can then load the item into the program, for example, for further password recovery, for checking password validity, etc.

2.6.7 MAC to vendor

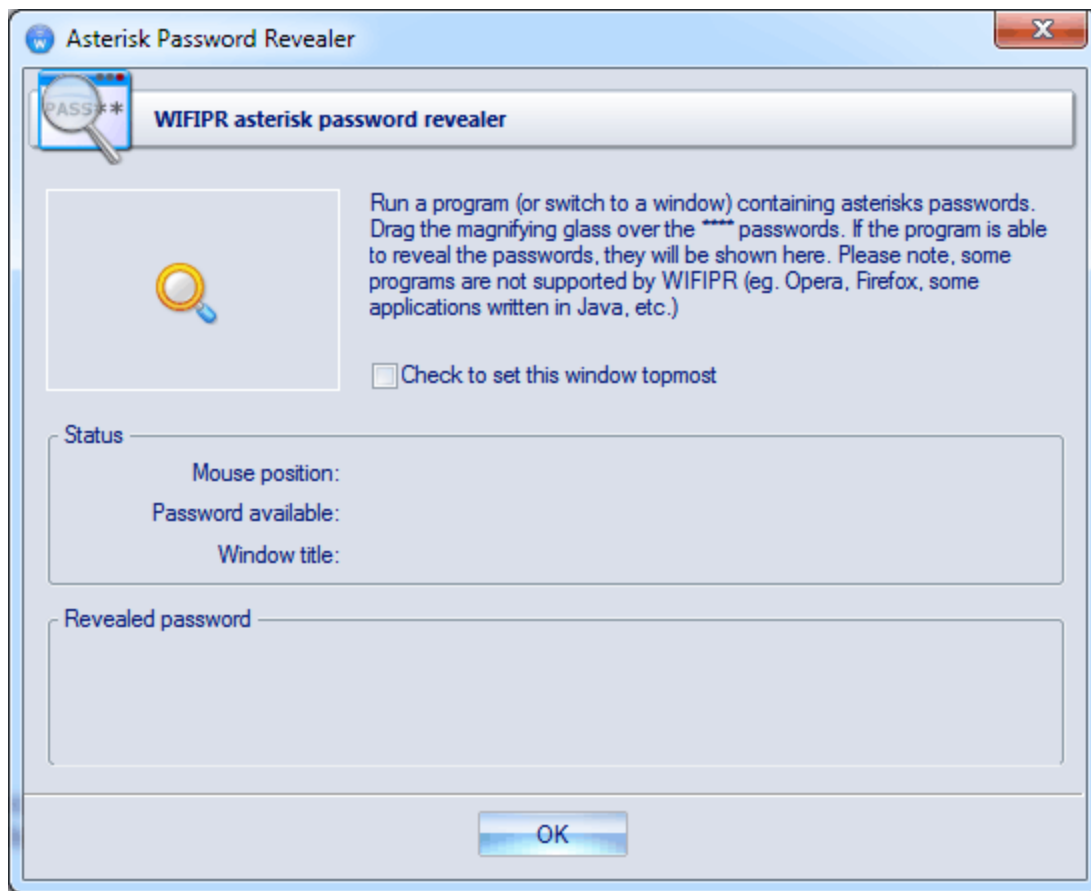


This simple tool is aimed to search for vendor name by its MAC address and vice versa. To find a vendor, start typing its MAC address into the appropriate (<Filter>) search box. The same way, you can view all MAC addressed that belong to a certain vendor.

2.7 Utils menu

Utilities menu consists of additional addons aimed mainly for advanced users.

2.7.1 Asterisk Password Revealer



This tool allows to recover passwords hidden behind asterisks. It is often helpful when you need to quickly recall a **** password and don't have the necessary recovery tools handy. In order to get the *** password visible, you should have to drag the magic magnifier from the program's window to the field with asterisks.

This method works both for Windows controls and Internet Explorer windows. It has a number of restrictions though:

- Some applications have their own GUI, and therefore Asterisks Revealer may be unable to interact with such applications. Those include Opera, Mozilla, Firefox, etc.
- Some websites have a built-in protection, which hides either the garbage or the actual asterisks behind the asterisk characters * (asterisks hidden behind asterisks!).
- In some Windows system dialogs asterisks also hide the * character and not the real password.

To ensure the proper operation of this tool, you are to have the administrator privileges.

2.7.2 Wordlist tools

Rather a scant number of acceptable tools for working with specialized password dictionaries has inspired the developers of this software to create their own toolkit. With this toolkit, you can easily create new and edit existing wordlists, as well as use them with any password recovery applications.

2.7.2.1 Create new wordlist by indexing files

This tool is designed for creating a new wordlist by selecting (indexing) words from local files on your computer. For example, those could be *.html, *.xml, *.txt, *.doc files, as well as *.mdb, *.pdf, *.exe files, etc.

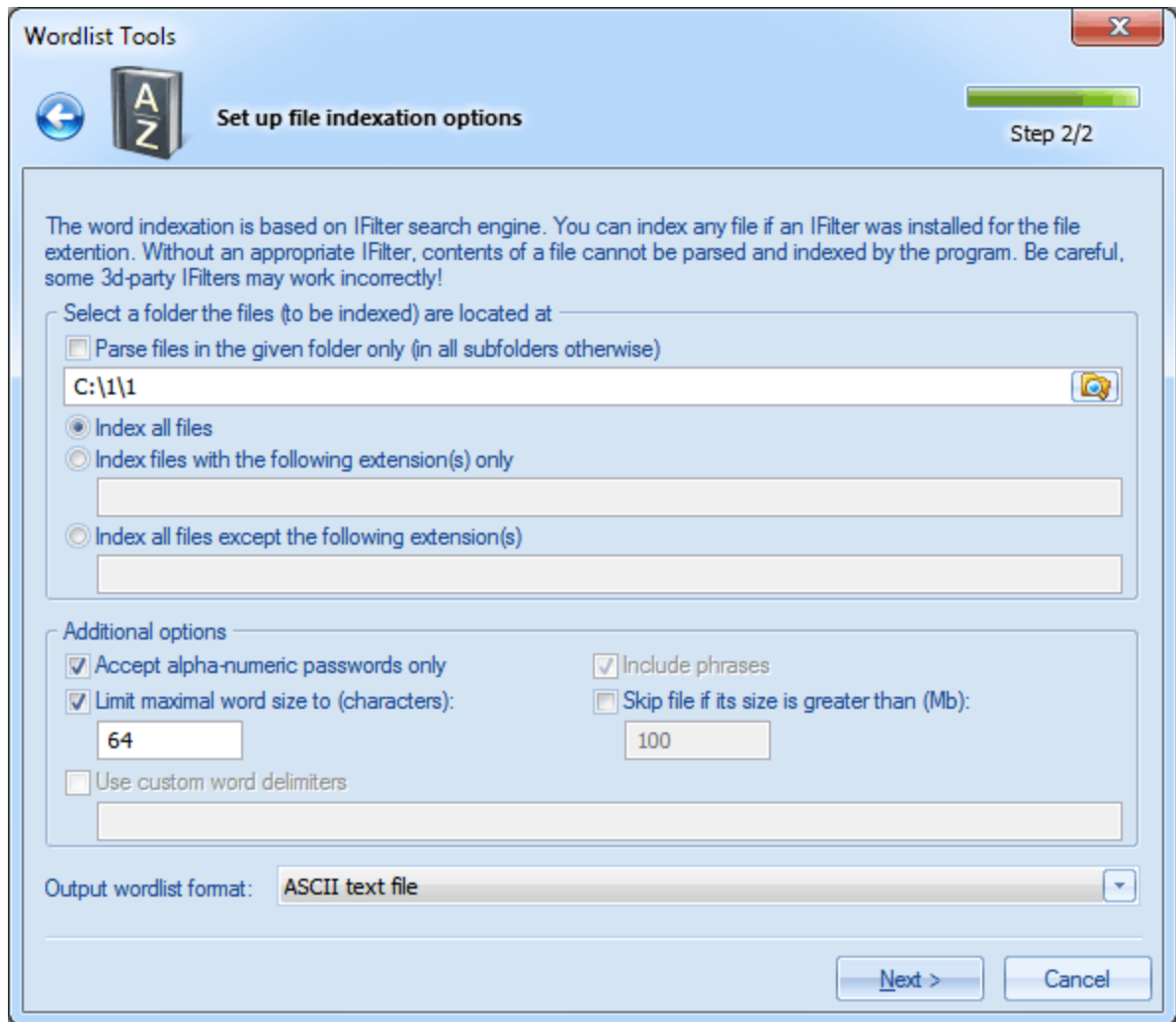
The indexing is based on the **IFilter** technology, which you can read about in [Wikipedia](https://en.wikipedia.org/wiki/IFilter). The idea of the technology, developed by Microsoft, comes down to the possibility of indexing the text of any file, which an appropriate IFilter plugin is installed for. This way, you could access the text contained, for example, inside *.exe or *.dll files, e-mail client's database, etc.

Despite the fact that numerous IFilter plugins, both commercial and free, can be found on the Internet, Wireless Password Recovery has internal support for the following types of files:

- Archives: *.zip, *.cab, *.rar
- Programs: *.exe, *.dll, *.cpl, *.ocx, *.sys, *.scr, *.drv
- Text: *.txt, *.dic
- Internet: *.html, *.htm

In other words, files with these extensions can be parsed by the program even without a single IFilter installed on the computer.

Windows 7 has an internal Windows Desktop Search tool, which has a wide range of filters for supporting the majority of popular documents. Under other operating systems, Windows Desktop Search can be installed manually; the setup file can be downloaded from the official website of Microsoft.



The configuration options for this tool consist of two groups. In the first group, you set path to the initial folder, where you need to index the files, and select a file parsing method, namely:

- Parse files in the specified folder only. If this option is not set, the program recursively analyzes all the subfolders and files inside them.
- Index all files
- Index files with certain extensions only
- Index all files except certain extensions

File extensions are to be typed without the dot and to be separated by a comma. Example:
txt,dic,xml,htm,html

The additional options group allows to customize file parsing methods, namely:

- Accept alpha-numeric passwords only. If set, this option will skip all special characters. Only alpha-numeric passwords will be extracted.
- Include phrases. This option also allows putting phrases into destination wordlist. A phrase is considered as a string of characters (of up to 256 symbols) with at least one space character in it.
- Limit maximum word size. It is recommended to always set this option. The best maximum word length in a wordlist is 16-64 characters. Cutting the maximum length sometimes radically speeds up the file parsing process. It wouldn't be worthless to remind that the maximum allowed password length in Windows is 128 characters.

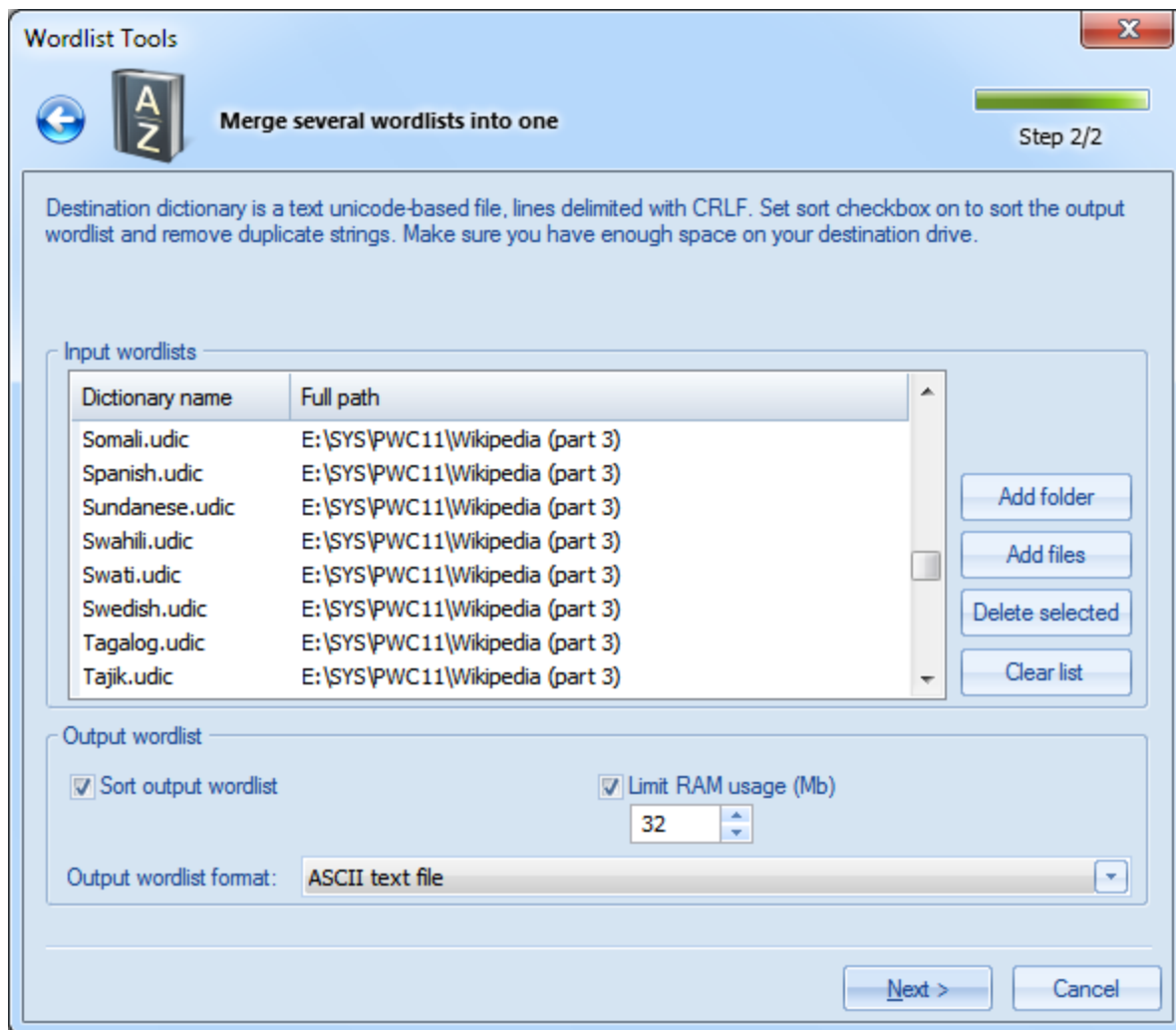
- Skip files with size greater than specified. Some IFilters take very long to parse large files; that can cause the program to "hang".
- Use custom word delimiters. You can set your own word delimiters for parsing files. For example, you could use characters like: !"#\$%&'()*+,-./:;<=>?@{}[]_ and, of course, space.

Clicking the **Next** button launches the actual indexing, which may take considerable time. For the sake of speeding up the process, the list of words found during the indexing is created and maintained in the computer memory; that requires significant resources. So, if you get a runtime error of lacking the memory, try decreasing the maximum word length or limiting the number of files being parsed and then try running it over again. Once the operation is completed, and the found words are saved to disk, sort them out to get a truly valuable wordlist. Found words are guaranteed to be unique, i.e. they do not contain duplicates.

Be careful though, some third-party filters could fail to run properly and cause the application to "hang", fail or abnormally terminate. For example, some filters for parsing PDF in Windows XP are known to generate errors.

2.7.2.2 Merge wordlists

A wordlist merging tool is used when you need to combine two or more wordlists in one.

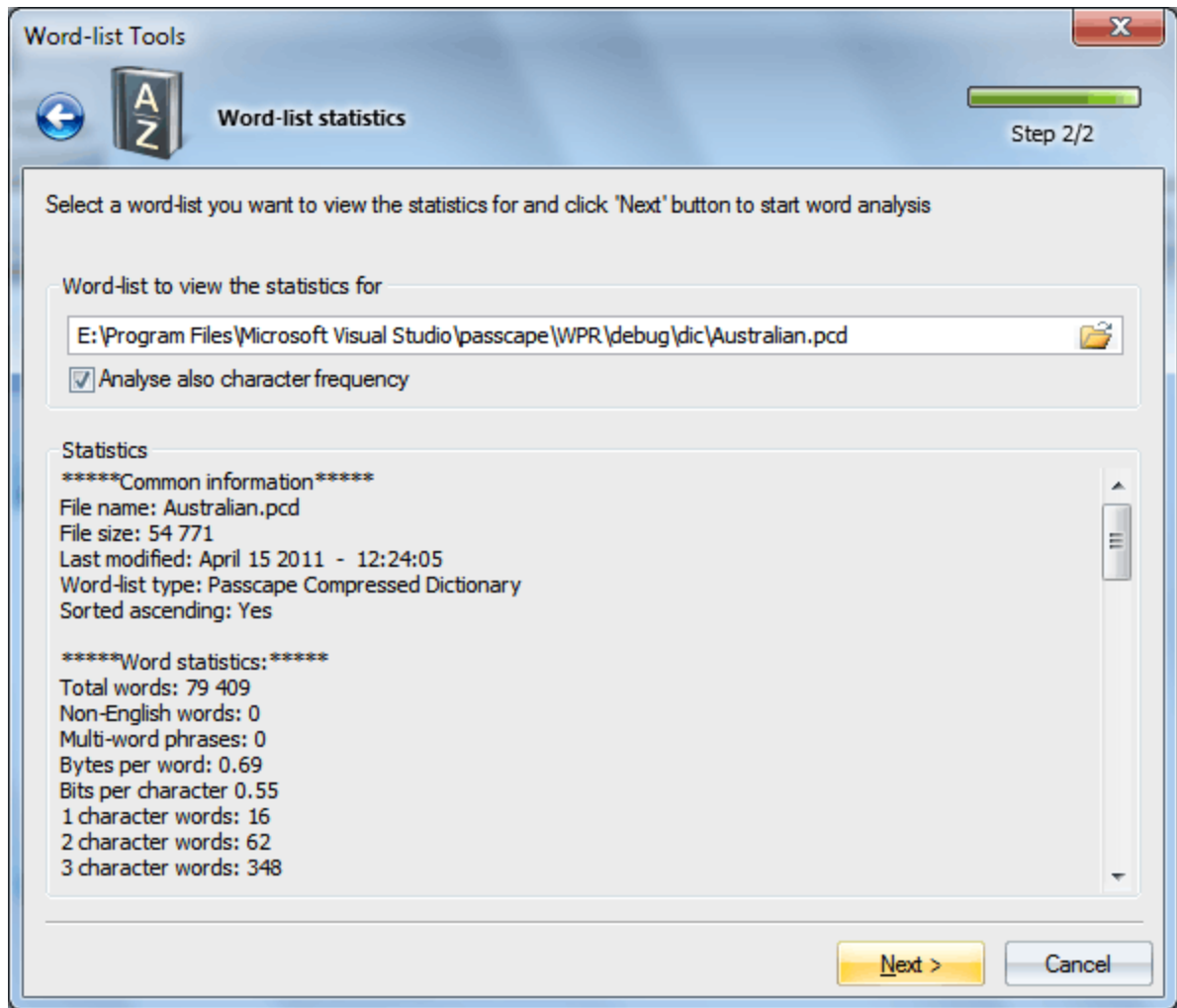


If the **'Sort output word-list'** option is not set, merging comes down to simply adding new words, without sorting or checking for duplicates. In practice, however, more common is merging with sorting; it ensures that all the words in the output wordlist are alphabetically sorted and duplicate-free.

Sorting may take a considerable amount of memory; therefore, it is appropriate to set a limit for the amount of memory that can be used by the process (at the expense of a little downgrade of the operation speed).

2.7.2.3 Wordlist statistics

Wordlist analyzer gathers and shows the following statistics:



Common information

- Dictionary name
- Size in bytes
- File type
- Last modified date and time
- Whether or not alphabetically sorted (the check takes place only if the file is sorted ascending)

Word statistics

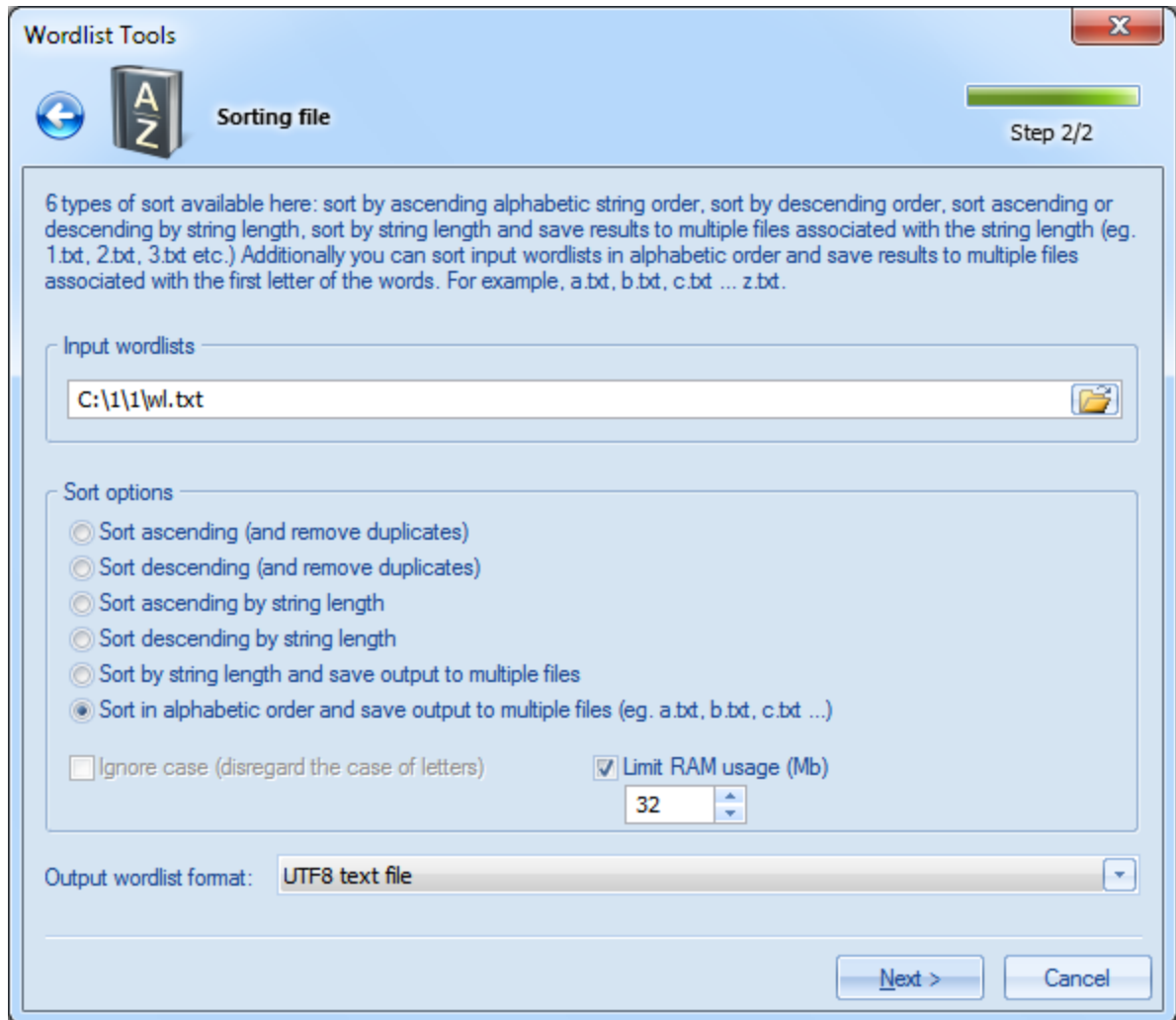
- Total words
- Non-English words
- Multi-word phrases, i.e. words separated with space
- Bytes per word, less word delimiter. Shows average wordlist compression ratio.
- Bits per character. Shows real wordlist compression ratio. For example, in UNICODE the bits per character value tends to 16 (not counting word delimiter), in regular ASCII wordlists - to 8. In certain compressed PCD wordlists one letter can be coded by less than 1 bit (see the screenshot).
- Word statistics - how many words consist of 1, 2, 3, etc. characters.

Character frequency analysis (if the respective option is set)

- Indicates how frequently a certain character appears in a wordlist

2.7.2.4 Sort wordlist

The toolkit offers 6 modes of wordlist sorting; 4 of them are common, and 2 are extended. The common sorting modes include sorting wordlists in the alphabetical order (both ascending and descending) and by word length. When sorting alphabetically or by word length, the program automatically removes word duplicates.



Additionally, you can sort a wordlist by length and save the results in multiple files, associated with word length. For example, file 1.txt would contain 1-character words, 2.txt - two-character, etc.

The sixth sorting mode works similarly. At the same time, the program sorts the source wordlist in the alphabetical order and creates several target wordlists that correspond with the first letter of the word. For example, all words beginning with letter A would be written to file A.txt, words beginning with B - to B.txt, etc. You should keep in mind that certain words may begin with characters that cannot be used in a file name. In this case, the program automatically suggests a replacement by issuing an appropriate warning in the messages window.

If the 'Ignore case' option is set, the sorting is carried out regardless of letter case; i.e., the words *bad*, *Bad* or *BAD* are considered identical, with all the ensuing consequences.

Target wordlist name may be the same as the source; however, that is not recommended.

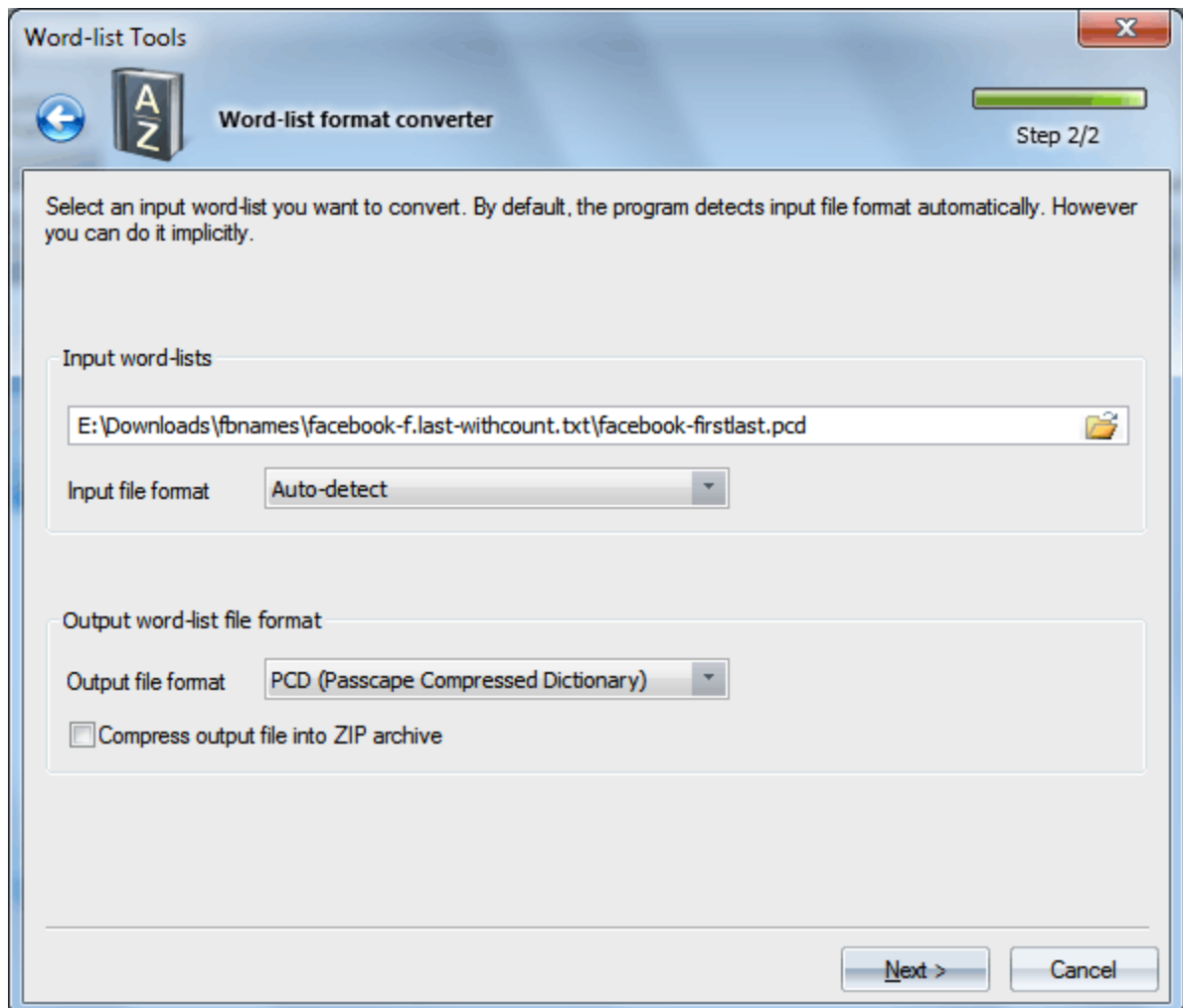
Sorting large files (supports files larger than 4 GB) involves intensive use of RAM; the amount of it can be limited by the respective option. For large files, it is not recommended to set the memory limit less than 16 MB, as that can affect the speed of sorting.

While sorting, the program may create auxiliary files in the application's temporary folder. Make sure that the disk with the temporary folder has enough room for the swap files.

2.7.2.5 Convert/compress wordlist

Numerous wordlists that can be found on the Internet are usually represented by three major formats: **ASCII**, **UTF16** (Unicode) and **UTF8**. With this tool, you can convert a wordlist from one format to another and optionally compress wordlists to ZIP files. Besides the three above mentioned formats, the program supports its own format **PCD** (Passcape Compressed Dictionary), which, in the majority of cases, gives a greater gain in size even compared to a compressed ZIP archive.

Creating large PCD files may take considerable time!



This tool's user interface is pretty easy. In the upper group, select the source wordlist and its format. By default, the program detects the format of the file automatically, but you can also specify it by hand.

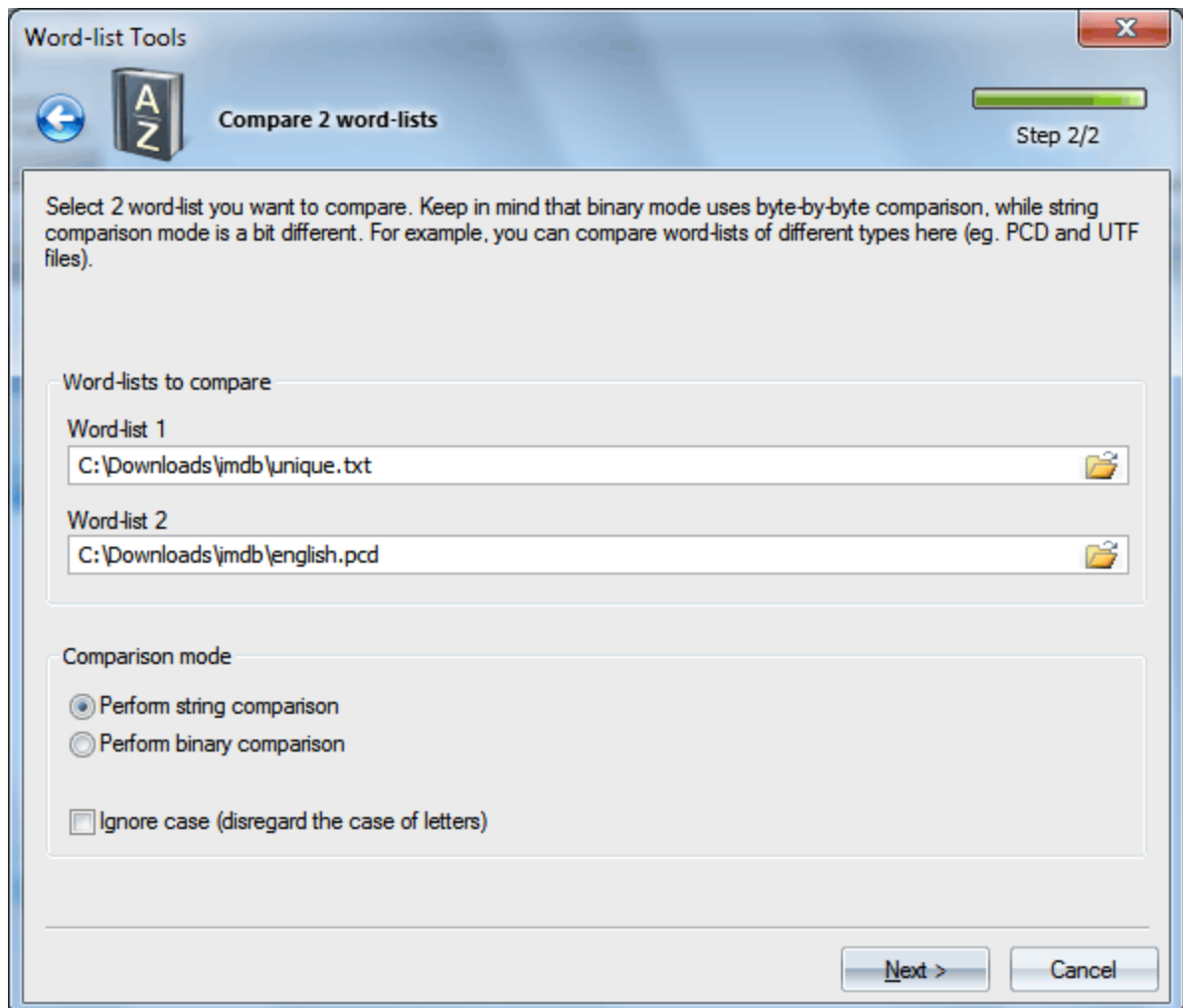
While the format of a PCD can be clearly recognized, with text files it's not that easy. As a rule, text files/wordlists in UTF16 or UTF8 begin with a two- or three-byte marker that describes the type of the file. However, there are Unicode wordlists that do not have any identifying markers. For such "hard" cases, you need to set the type of the source file manually. Otherwise, the program, being unable to see an appropriate identifier, improperly recognizes the file as ASCII.

Target wordlist, similarly, is defined by one of the four above mentioned formats. With the compression option set, the program additionally compresses the file to a ZIP archive.

Target wordlist name may be the same as the source; however, that is not recommended.

2.7.2.6 Compare wordlists

Sometimes, it is necessary to determine whether two wordlists are identical. That is what the wordlist comparison tool for.



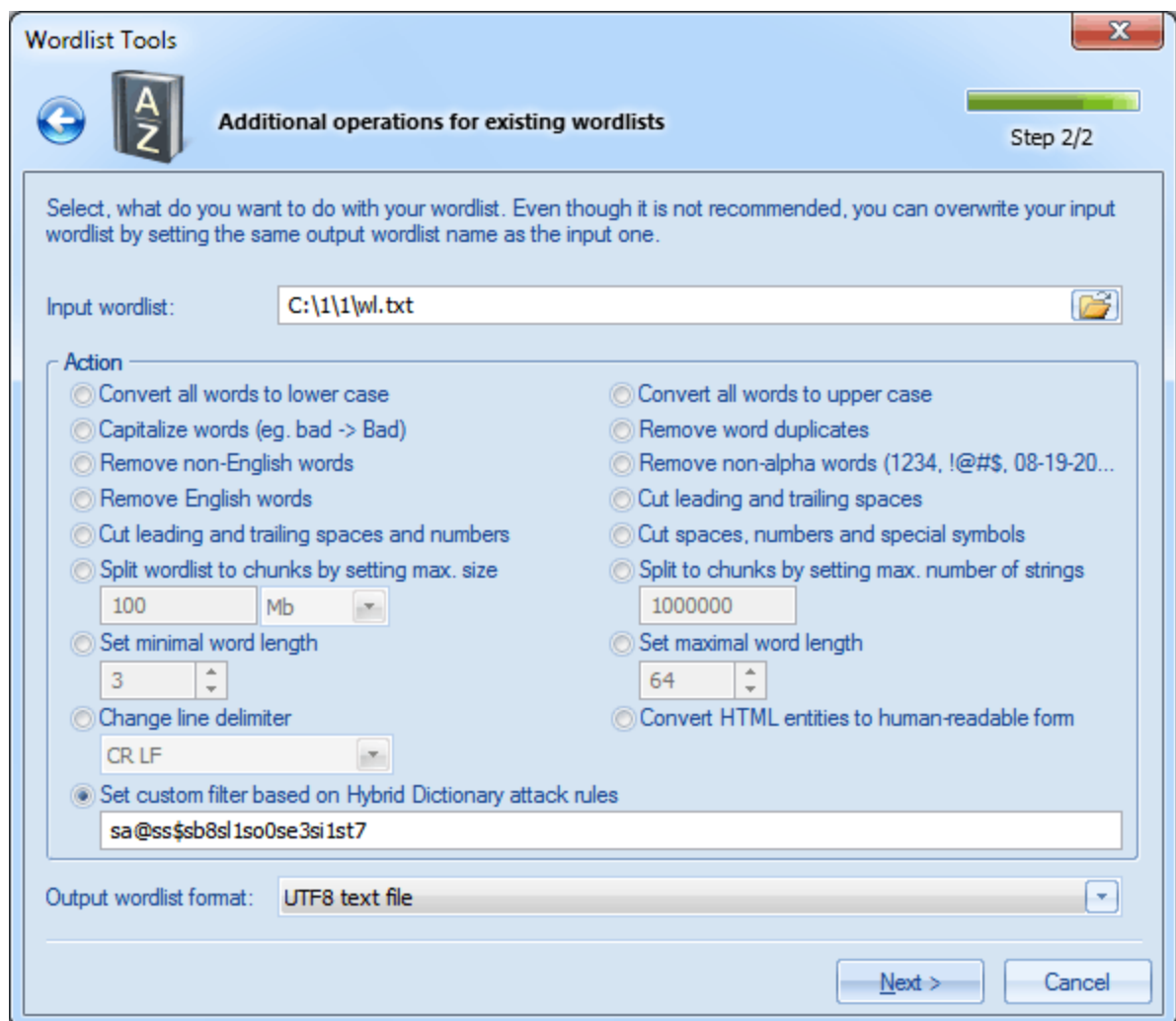
This tool offers two operating modes:

1. Binary comparison, for comparing files by-byte
2. String comparison, which compares words rather than bytes. This mode is noteworthy for its ability to compare wordlists of different formats. For example, PCD and UNICODE, or UNICODE and ASCII.

If the ignore case option is set (string comparison mode only), then, for example, the words *bad* and *Bad* will be considered identical.

2.7.2.7 Additional operations

The additional tools are designed primarily for editing and tuning up existing wordlists.



The tools include the following operations:

- Convert all words in wordlist to lower case. For example, BAD -> bad.
- Convert all words to upper case. For example, Bad -> BAD.
- Capitalize words - upper-case first letter, lower-case all others. For example, bad -> Bad.
- Remove word duplicates.

- Remove non-English words.
- Remove words that entirely consist of numbers and/or special characters. For example, 12345, !@#\$%, 08-19-10, etc.
- Remove English words.
- Cut/remove leading and trailing spaces.
- Cut/remove leading and trailing spaces and numbers.
- Cut/remove leading and trailing spaces, numbers and special characters.
- Split wordlist to chunks by maximum size.
- Split wordlist to chunks by maximum word count.
- Remove words of length smaller than specified.
- Remove words of length greater than specified.
- Change line delimiter.
- Convert HTML entities to human-readable form. For example, **&** -> **&**, **@** -> **@**
- Set your own filter based on [Hybrid Dictionary rules](#)

For source wordlist, the program takes ASCII, UTF16, UTF8 and PCD files. Target wordlist can be a text of ASCII, UTF16 or UTF8.

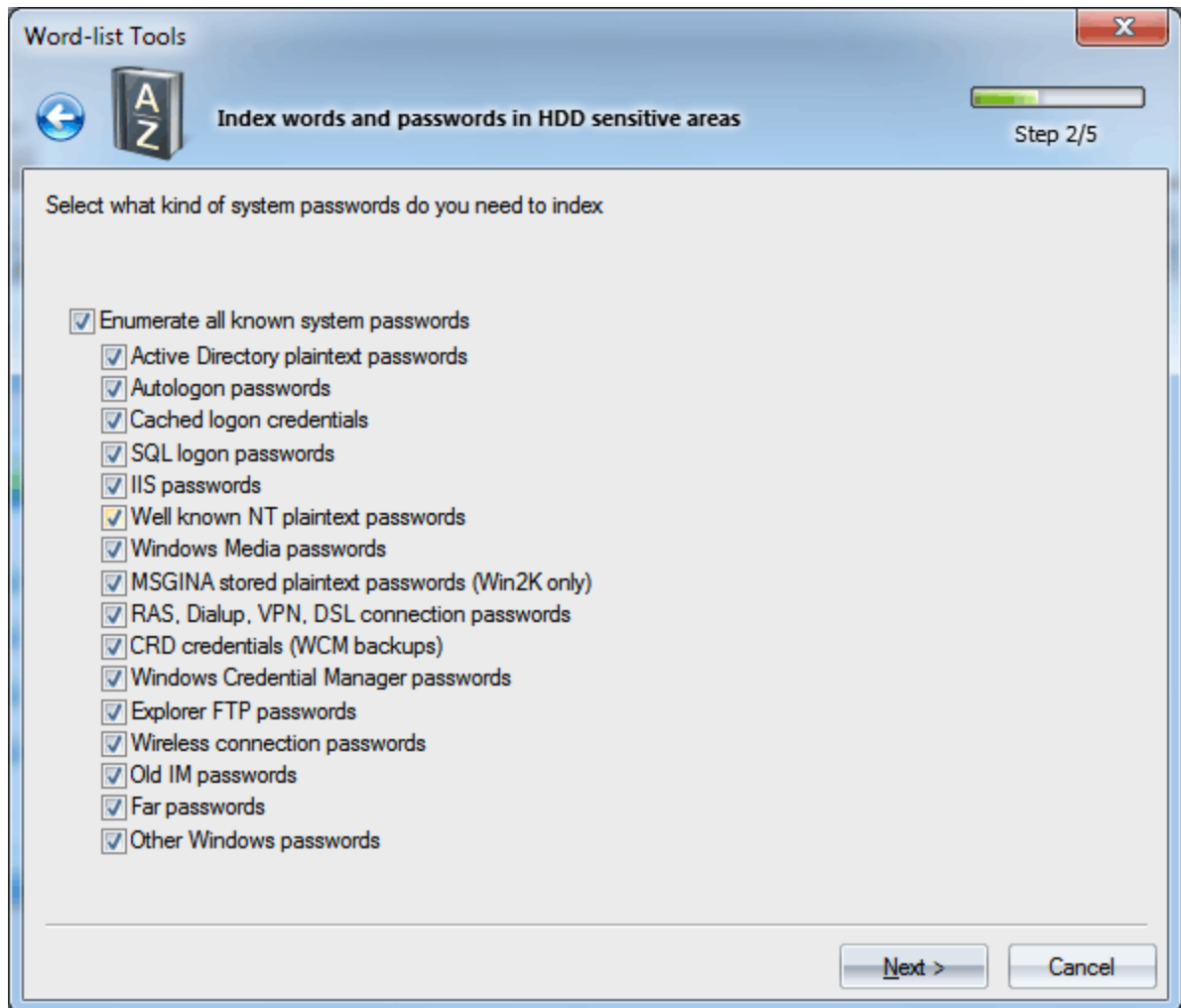
Source and target wordlist name may be identical (not recommended). In this case, the source wordlist will be overwritten.

2.7.2.8 Index HDD sensitive areas

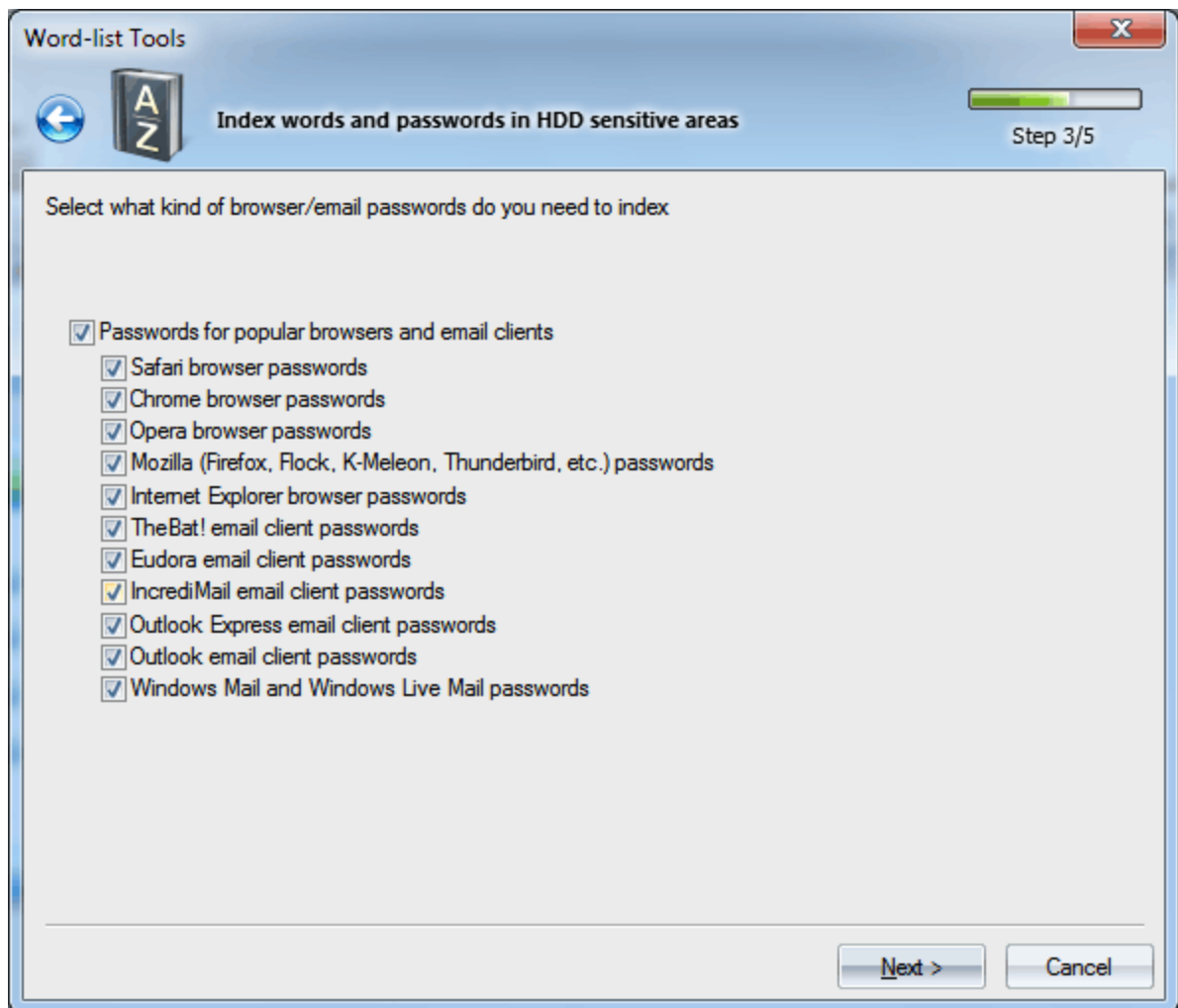
Creating a wordlist by indexing the hard disk (followed by an attack using this wordlist) is a pretty useful and sophisticated tool for decrypting passwords to local Windows accounts.

Often users, instinctively, set same passwords to their Windows accounts, Web, ICQ, etc. The idea of this tool is to create a wordlist of all found formerly used passwords, user's messages, words from recently opened files, etc. and then use the accumulated wordlist for looking up passwords to the local accounts. This technique is engaged in the Artificial Intelligence attack.

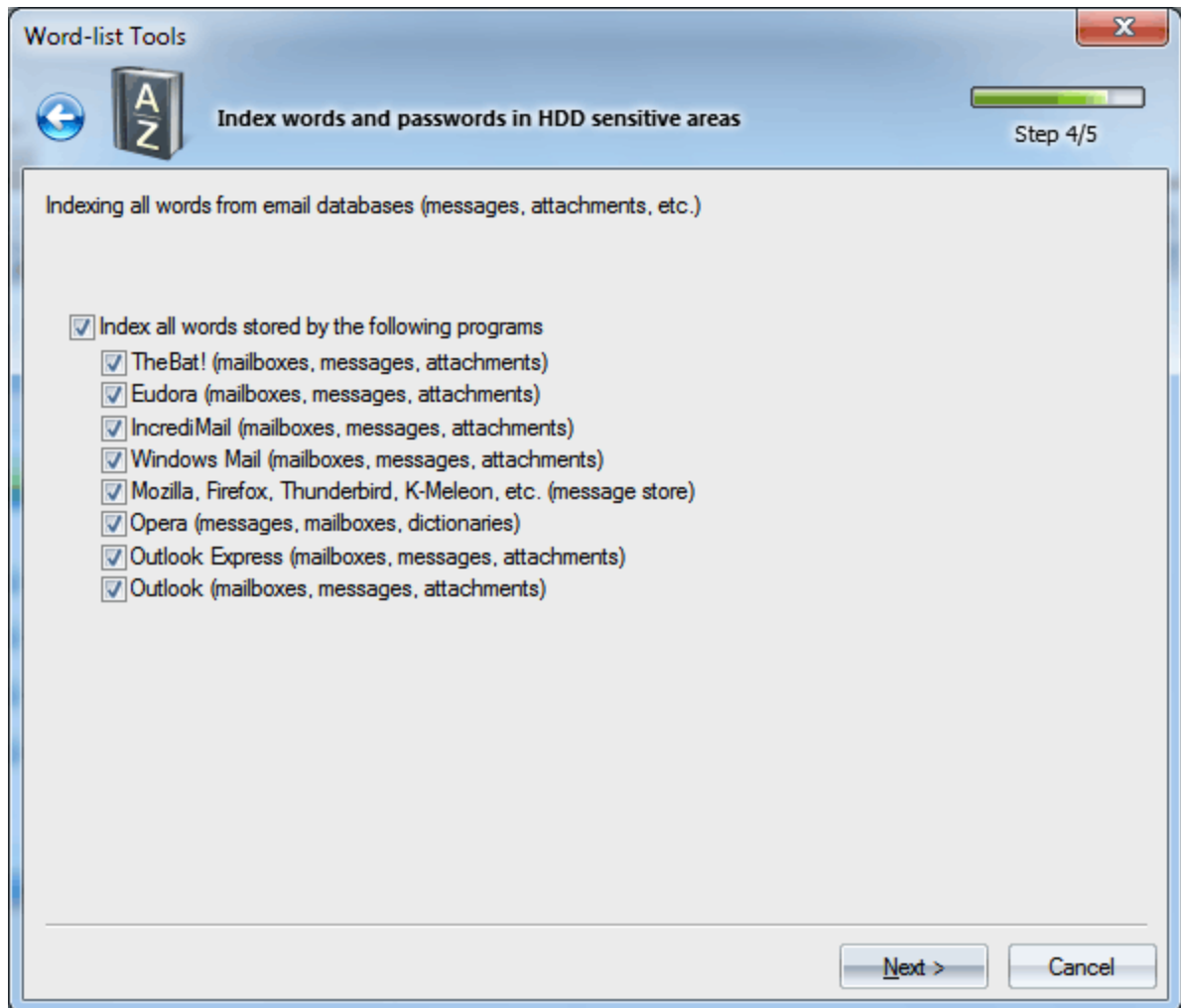
The configuration of the tool conventionally consists of four parts:



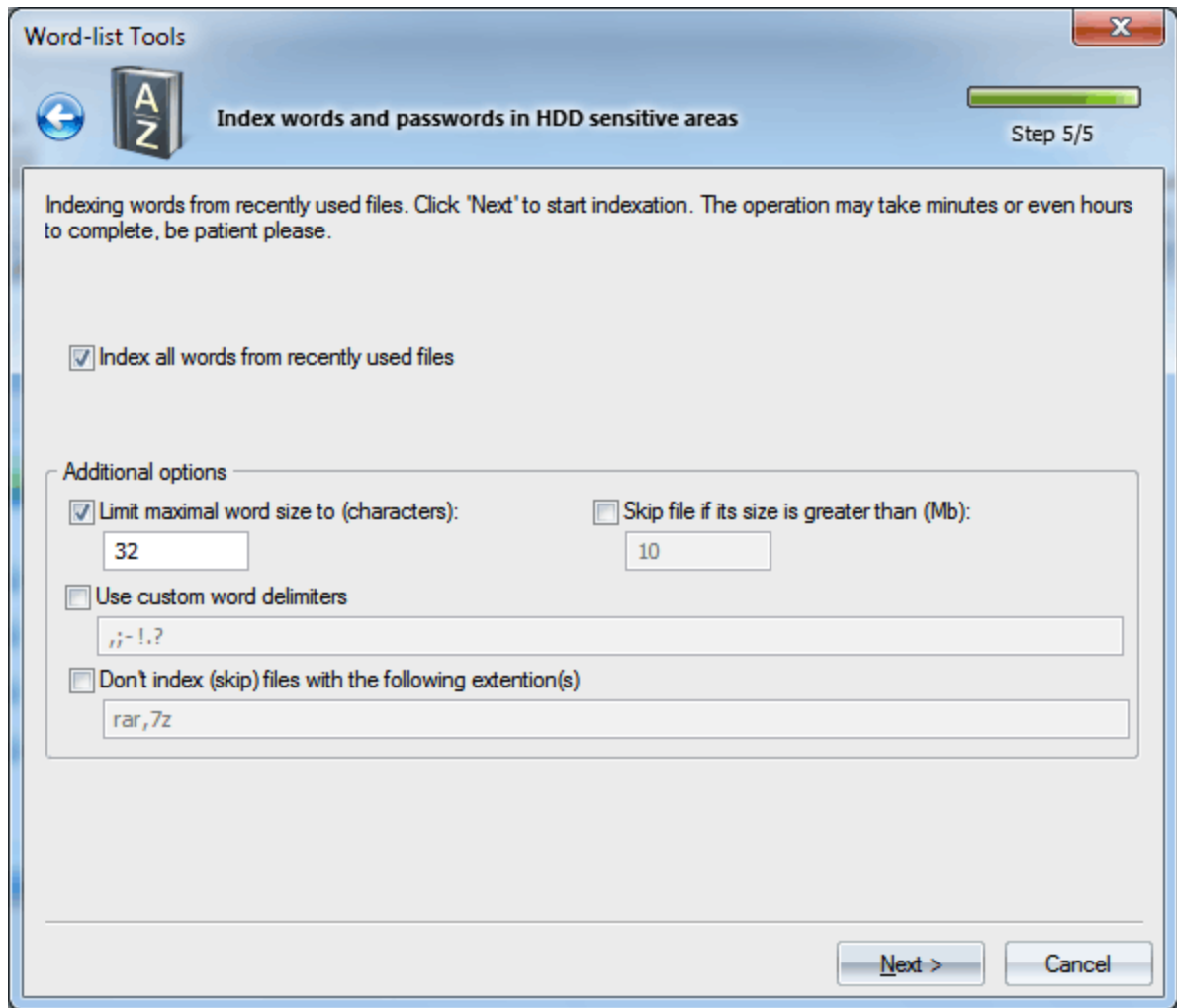
First, select the system modules to be used when generating the wordlist. These modules find and index the following types of passwords on your computer's hard disk: Active Directory plaintext passwords, startup passwords and cached startup passwords, SQL, IIS, Windows Media, Win2K text passwords, RAS, Dialup, VPN, DSL, WEP, WPA, FTP connection passwords, Windows Credential Manager passwords, Instant Messengers, etc. passwords.



In the second part of the configuration, select the browsers and e-mail clients, passwords from which are also to be found and added to the wordlist being created. The program supports the following major web browsers: Safari, Chrome, Opera, Mozilla-based browsers (Firefox, K-Meleon, Flock, etc.), Internet Explorer. E-mail clients are represented by: TheBat!, Eudora, IncrediMail, Outlook Express, Outlook, Windows Mail, and Windows Live Mail.



Besides merely gathering passwords, the program can index user's e-mail communication, scanning all found mailboxes, messages, attachments, etc. The hard disk search is performed for all accounts in a system, so the process may take considerable time, especially when the system hosts many users or when e-mail clients' databases are large. One way or the other, you can enable/disable each module individually.



Finally, in the last dialog, you can set the options for indexing words from all files, recently opened by current user. Available options include:

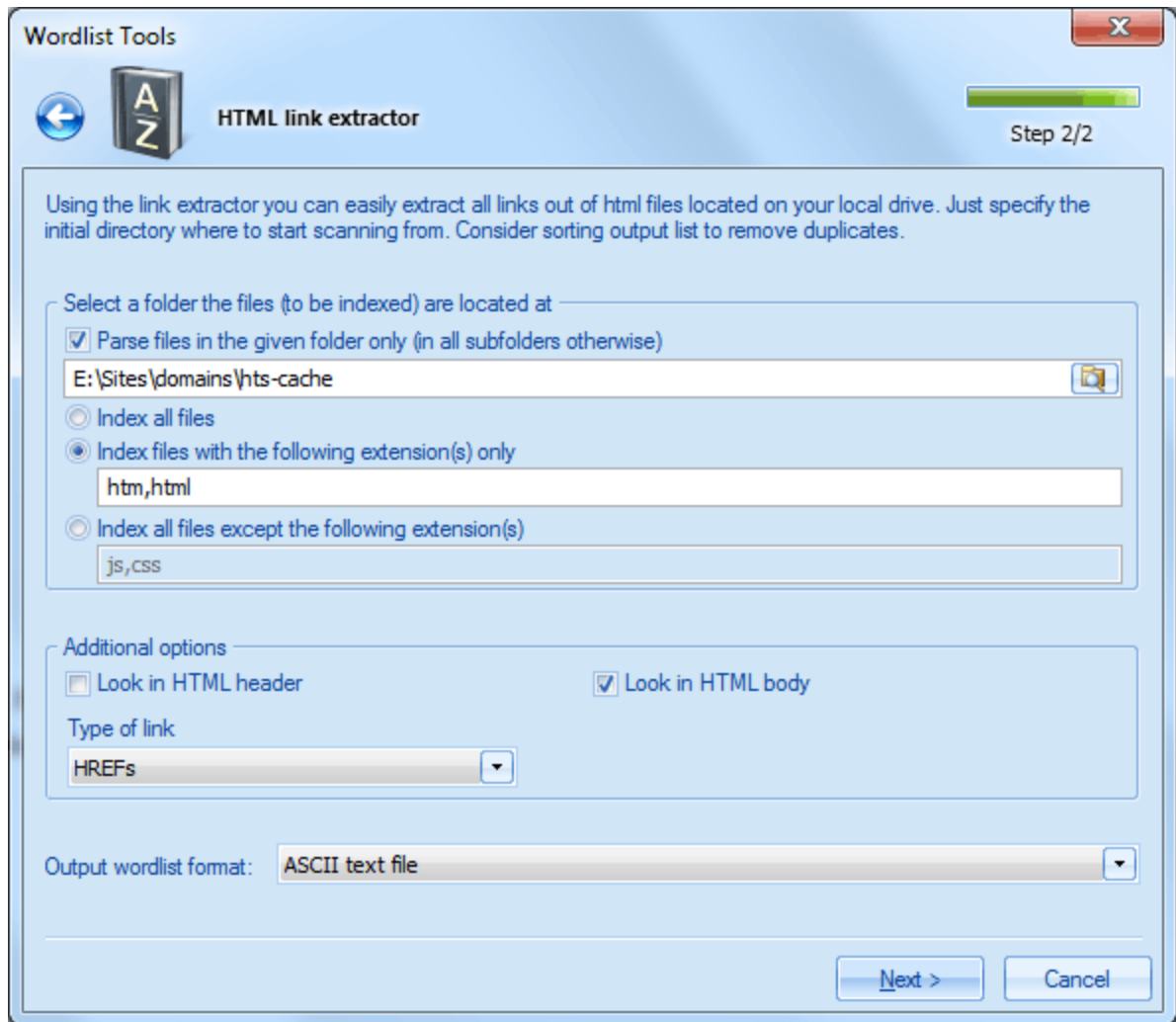
- Set the maximum length of words that can be added to the wordlist. All words with length greater than the specified limit will be skipped.
- Skip files with size greater than specified. The size is specified in MB.
- Use custom word delimiters. By default, word delimiters are all non-alphabetic characters.
- Do not index files with specified extensions. Use this option to skip files that you consider unnecessary.

Clicking the **Next>** button starts the indexing process.

Keep in mind that it can take considerable time!

2.7.2.9 Extract HTML links

This tool is designed for extracting HTML hyperlinks from HTML files.



The configuration options for this tool consist of two groups. In the first group, you should set a path to the initial folder, where the HTML files are located, and select a file parsing method, namely:

- Parse files in the specified folder only. If this option is not set, the program recursively analyzes all the sub-folders and files inside them.
- Index all files
- Index files with certain extensions only
- Index all files except certain extensions

By default, the tool checks *.htm and *.html files only.

The additional options group allows to set the type of links, as well as where to look for them:

- Look in HTML header
- Look in HTML body
- Look links in HREF tag, SRC tag or in both tags.

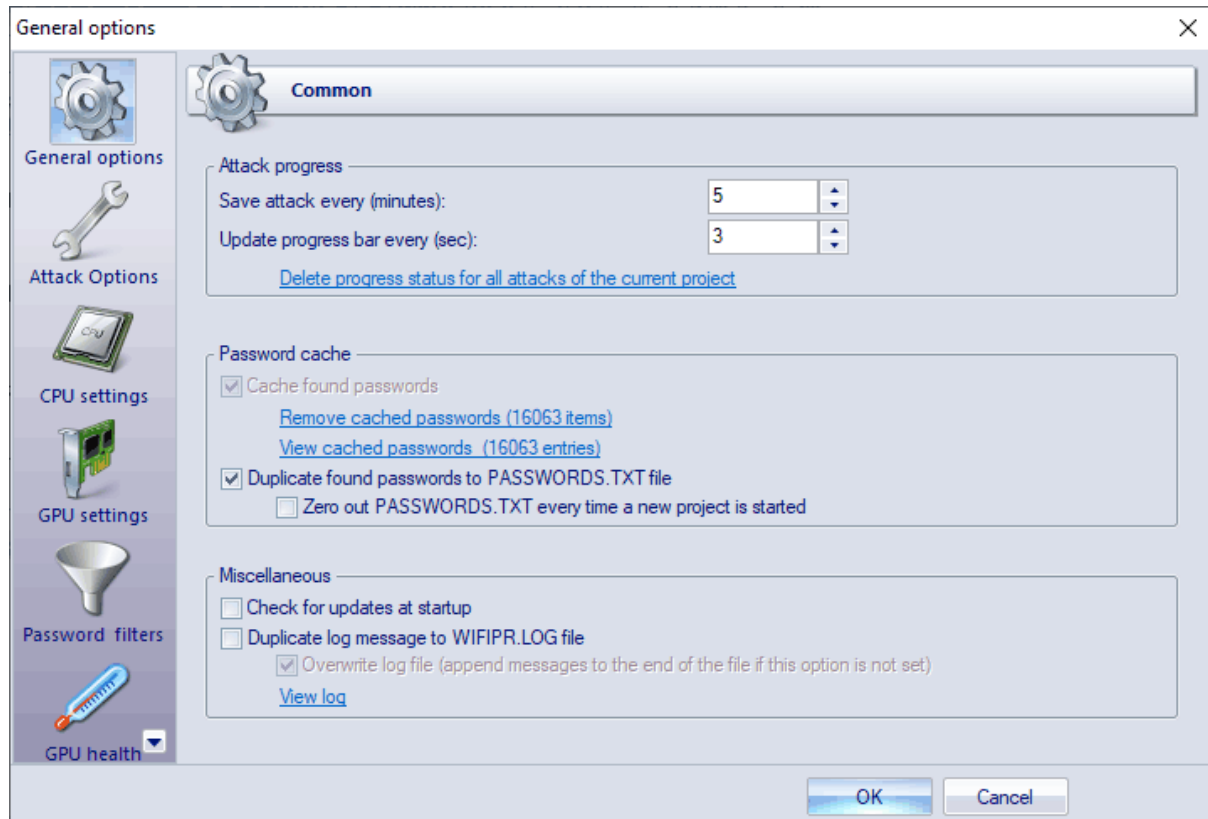
Clicking the **Next>** button launches the search, which may take considerable time. Once the operation is completed, and the found links are saved to disk, consider sort them out to get ride of duplicates.

2.8 Settings menu

2.8.1 General settings

The general settings are divided into several tabs.

2.8.1.1 General options



Progress

The first group of settings allows setting the save and update intervals for the current state of an attack. By default, an attack saves its state every 5 minutes (further on, you can resume the attack from the last saved point) and updates the screen every 3 seconds.

Password Cache

All passwords found by the program are cached by default. A very helpful thing that is engaged in many subsystems. For example, in the AI or preliminary attack. Deleting password cache is recommended in cases of the extreme need only. For example, when their number exceeded ten thousand. In this case, the search speed for some attacks can drop significantly.

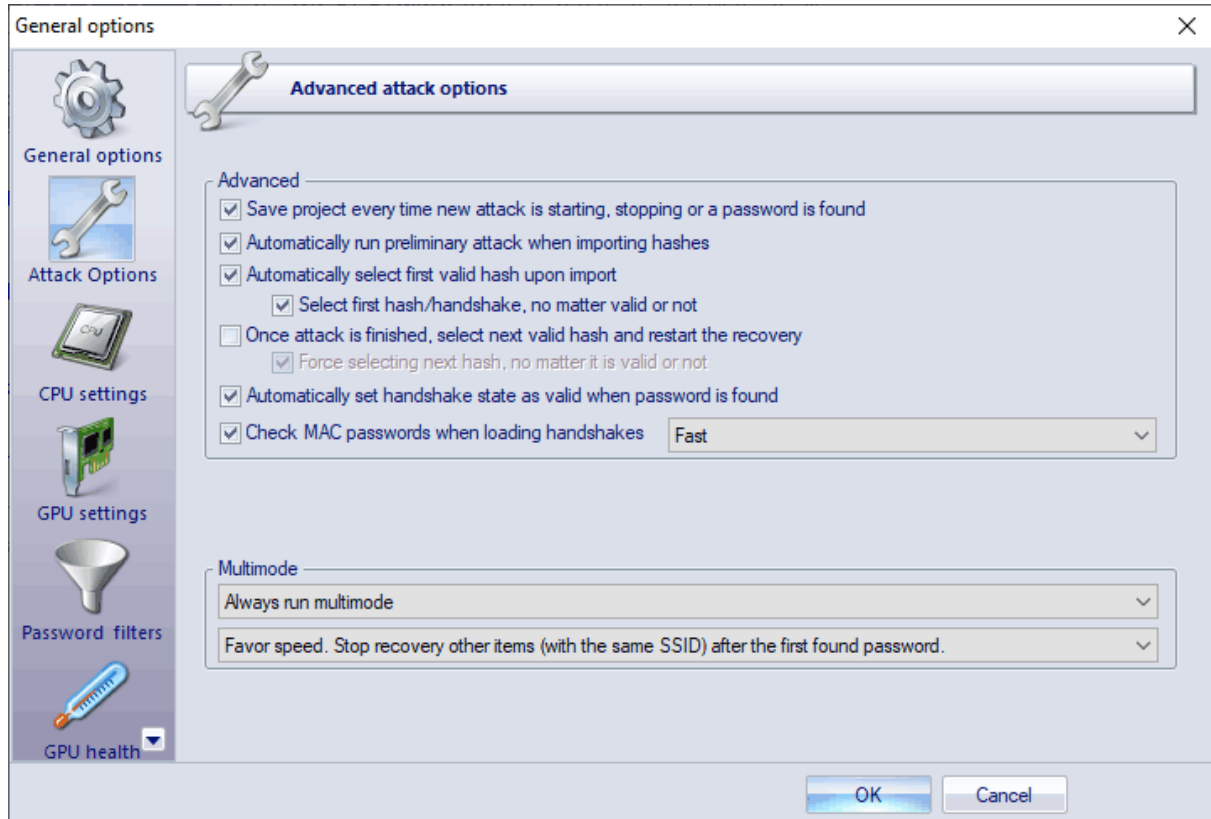
Additionally, you can duplicate found passwords to text file. So even if the program fail unexpectedly or in case of sudden power failure, the found passwords guaranteed to be written to file.

Check for updates at startup - check if an update is available every time the program starts. The option works only if PC is connected to internet.

Duplicate log messages to wifipr.log file - this option when set, writes all messages the log window holds to WIFIPR.LOG file. It can be helpful when the program stalls or works unstable. WIFIPR.LOG is located in the program's installation directory.

Overwrite log file - overwrite the log file every time the program starts. Otherwise new messages will be appended to the end of the log file.

2.8.1.2 Attack options



Advanced

- *'Save project every time a new attack is starting or stopping'*. Autosave project file when a new attack is started.
- *'Automatically run preliminary attack upon import'*. Automatically run preliminary attack every time hashes are imported. This attack can reveal particularly weak passwords in a matter of seconds.
- *'Automatically select first valid hash upon import'*. Automatically select first valid hash for further analysis when loading hashes into the program.
- *'Once an attack is finished, select next valid hash and restart the recovery'*. With this option set, once the selected attack is finished, regardless of its success, the program will select next valid hash and automatically restarts the attack.
- *'Set handshake state to valid when the password is found'*. Some users prefer to have a full control over the process of password recovery. This option, if set, automatically changes the handshake's state to 'Valid' when the password is found.
- Check MAC passwords when loading handshakes. Check passwords based on MAC address when loading handshakes. Two algorithms are available for fast and deep checkout.

Multimode

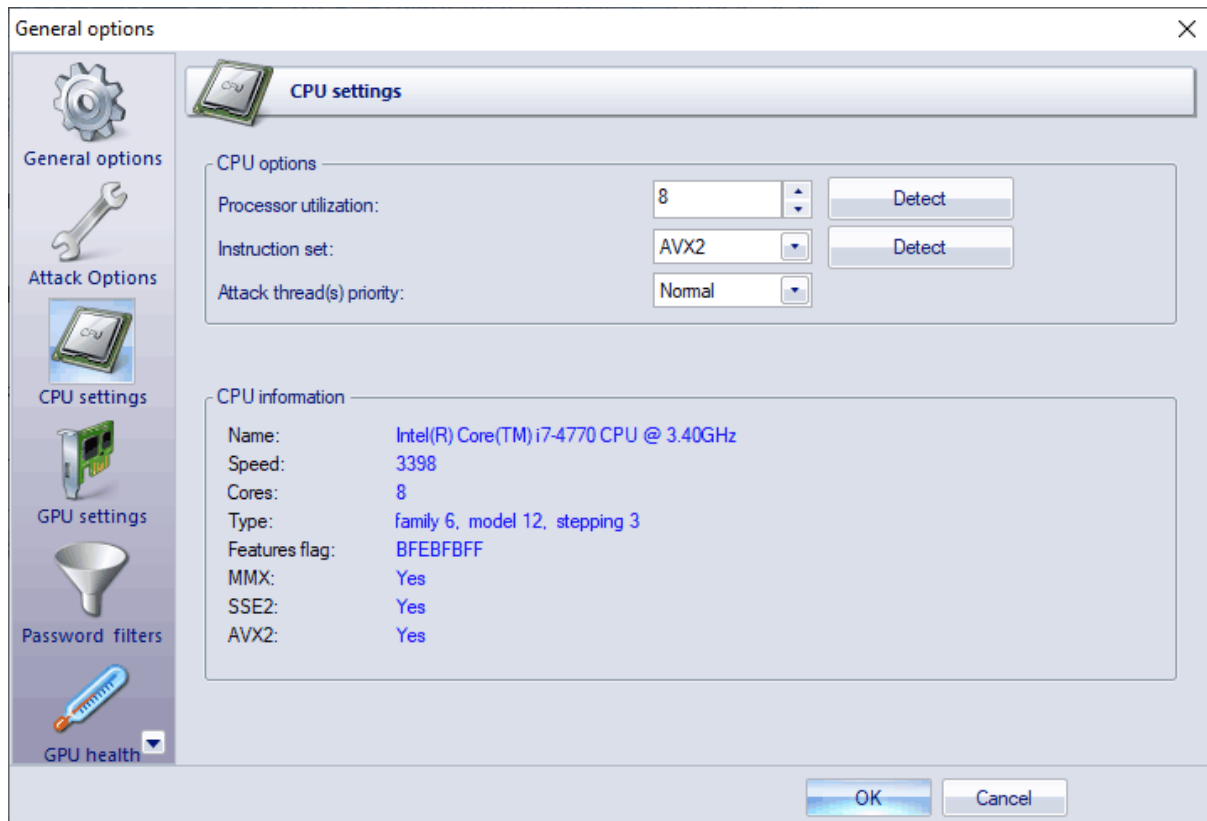
In multimode, the program can crack multiple handshakes with the same ESSID simultaneously with almost no speed loss. Wireless dump files may contain broken handshakes and often there's no reliable way to figure out whether a handshake is valid or not. Furthermore, some AP models use default ESSIDs. The program can combine all handshakes with similar SSID and recover them simultaneously. That is what the multimode is for.

- Select what to do when the program detects that it can run in multimode:
 - Prompt for multimode every time
 - Do not prompt, always run in multimode (recommended)
 - Do not prompt, never switch to multimode

Note that if the *'handshake auto increment'* option is set and the *'Prompt for multimode'* is selected, the program will prompt every time it switches to the next entry.

- Multimode simple logic:
 - Full recovery. The program tries to find as many passwords as possible. The multimode will continue working until an attack is over or until all passwords of the same SSID are found. If the recovery entries consist of several types (PMK hash, handshake or PMKID), the multimode will guaranteed to be launched for each type.
 - Smart recovery. Once a password is found for an SSID entry, the program will switch to the next entry with different BSSID. If a password for a PMKID entry is found, the program will skip other PMKID entries with the same SSID and all other handshake entries with the same SSID and BSSID.
 - Fast recovery. Once a password is found, the program ignores other entries with the same SSID regardless of BSSID and entry type. Be careful using this mode. For example, if you have multiple PMK, handshake and PMKID entries and the program is able to find a PMK password, all other PMKIDs and handshakes (with the same SSID) will be ignored and so the multimode will be switched to another SSID entries..

2.8.1.3 CPU settings

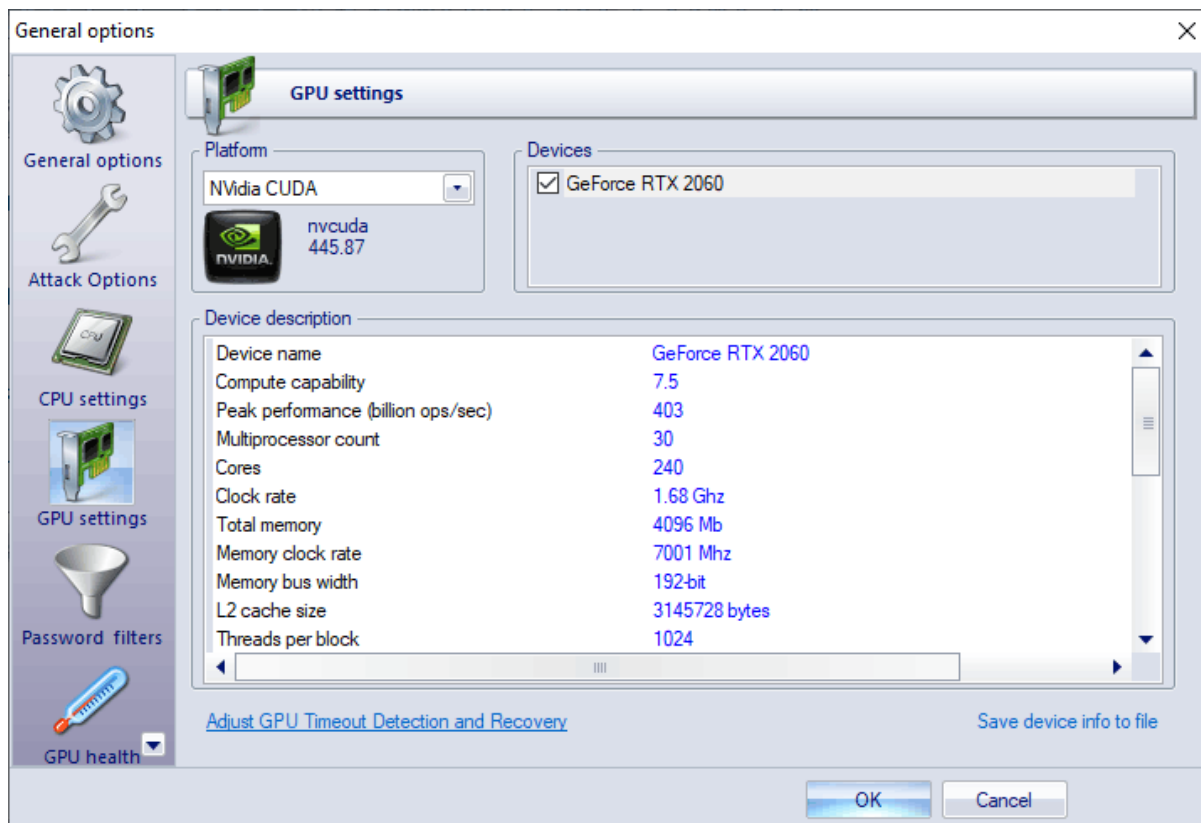


Since the majority of the attacks support multithreading, you can manually set the number of search threads to be run simultaneously. In the majority of cases, it should match the number of cores you've got in your CPU. However, if you are searching for passwords using GPU, that value should be less.

Password lookup algorithms in Wireless Password Recovery are optimized for certain CPU architectures: X86, MMX, SSE2 or AVX2. Typically, CPU that support newer architecture, performs runs faster.

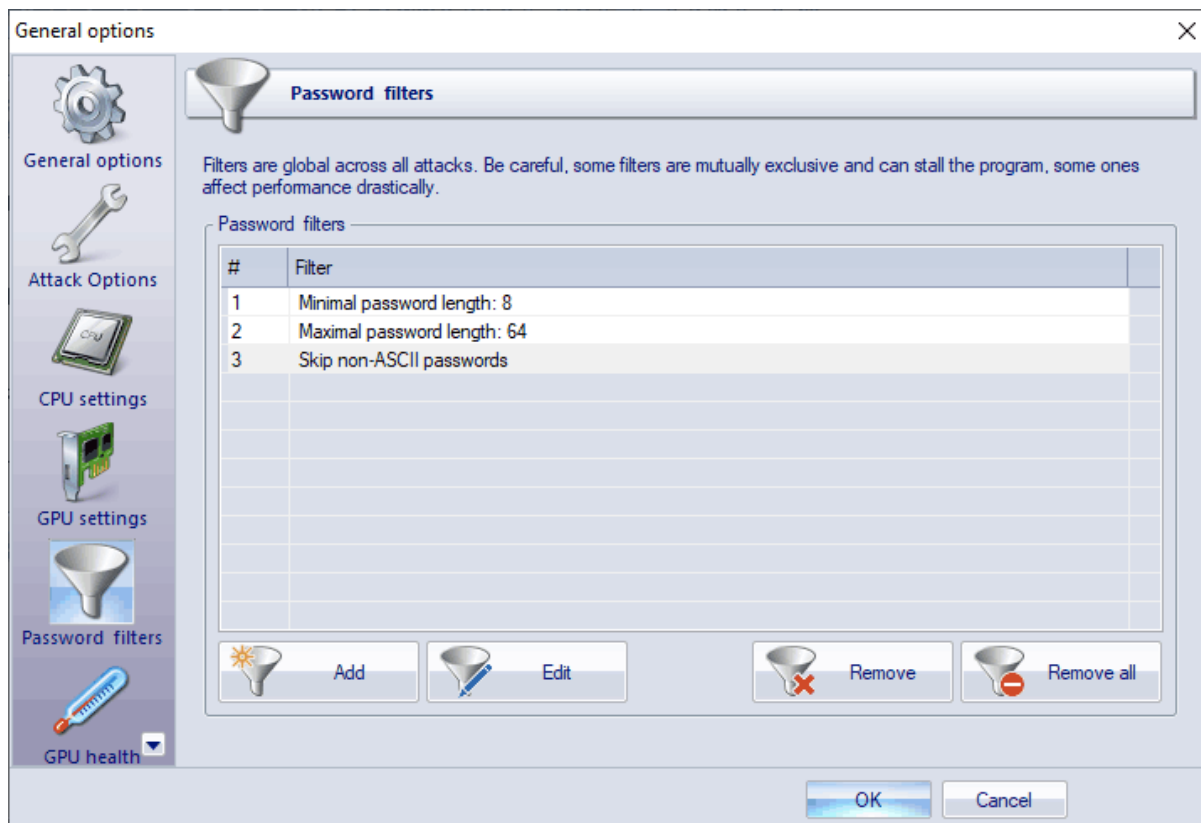
It is not recommended to set the attack priority above normal; otherwise, you may observe a considerable reduction of performance of the entire system.

2.8.1.4 GPU settings



Before launching a GPU attack, make sure to select it in the application's general settings. Simply check on the box against the respective GPU name. All the basic features of the device are shown in the description table. The software supports up to 255 devices.

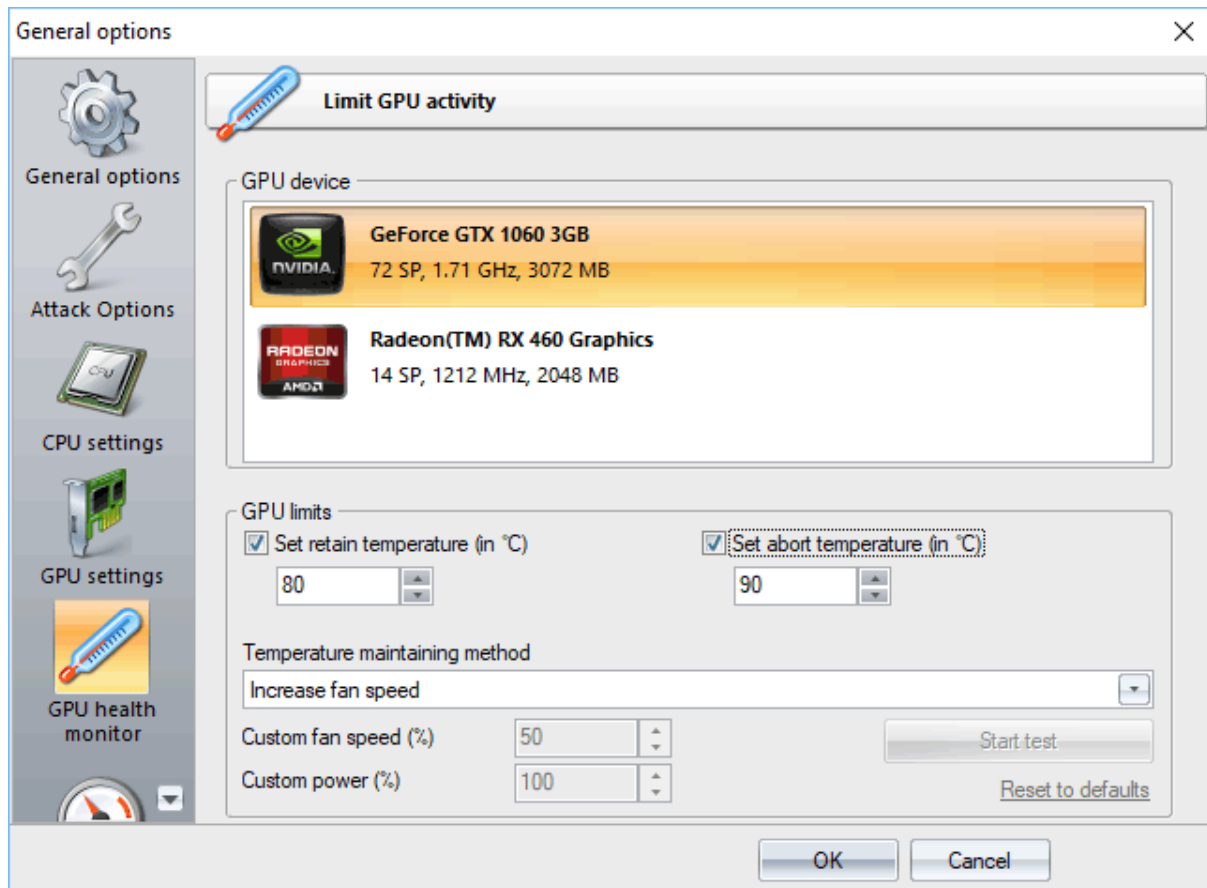
2.8.1.5 Password filters



Filters are used to skip/ignore passwords considered to be needless. For example, you can limit the minimum and maximum length, bypass passwords with only Latin characters, and so on.

Filters are global across all attacks and are applied when generating dictionaries as well (refer to attack settings for more info on dictionary generators).

2.8.1.6 GPU health monitor



You can use the GPU health monitor to retain a GPU temperature at a certain value, as well as to abort password search in case the temperature reaches a certain critical value.

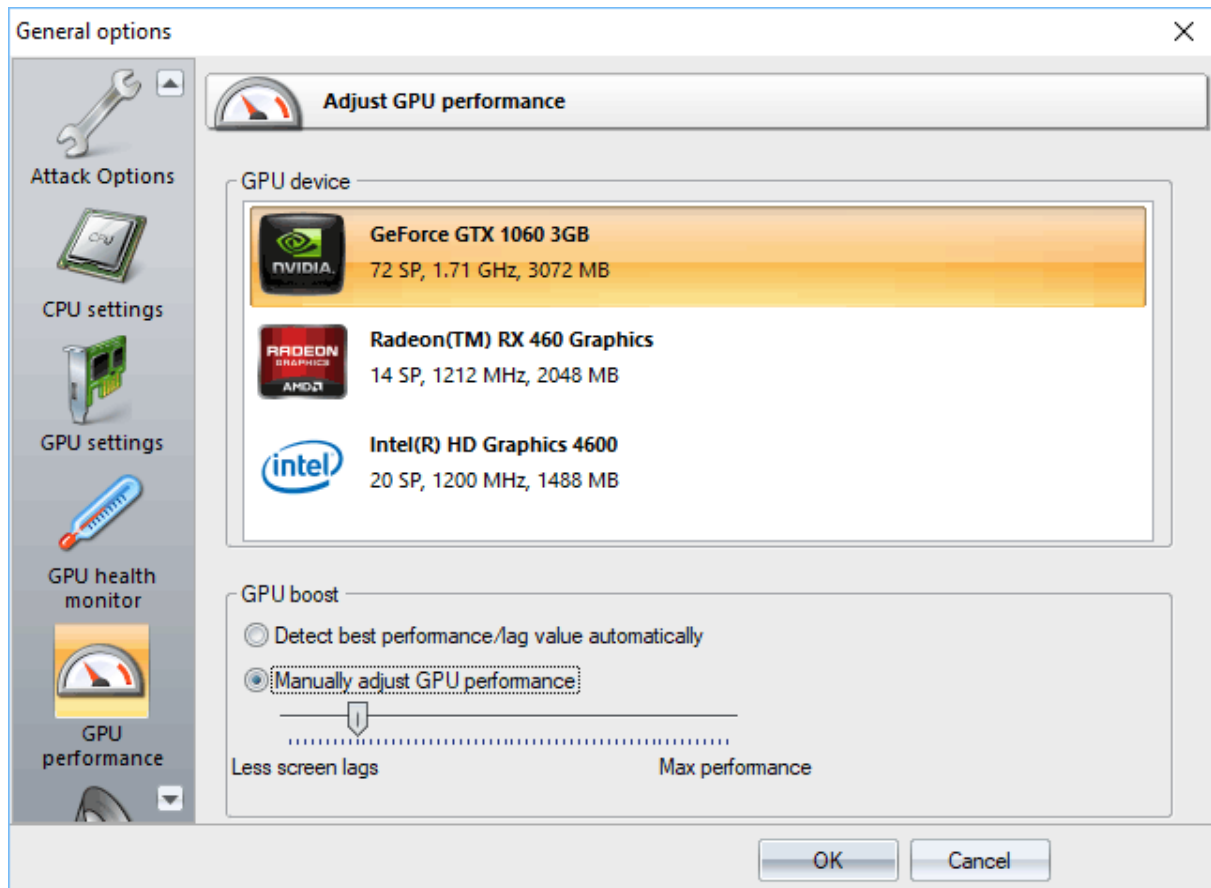
To set the retaining temperature, select your GPU from the list of available devices, check the appropriate option on and set up your value at degrees Celsius. Then select one of retaining methods:

- **Simultaneously increase fan speed and decrease GPU power.** Use it if you want to reduce the fan noise, but a slight decrease in the performance is not a problem for you.
- **Increase fan speed.** When this method is selected, the program increases fan speed when the temperature exceeds the limit and decreases when it is cooling down. Use it when you need maximal performance. It is also recommended to set up the abort temperature to avoid GPU overheating.
- **Decrease GPU power.** It is quite effective to cool down a GPU temperature. The GPU processor frequency automatically adjusts to the retaining temperature while the fan speed will be determined by your system settings. The disadvantage includes a significant performance drop down in certain cases.
- **Custom settings** include user-defined fan speed and/or GPU power.

Use the abort temperature to stop searching for passwords once the GPU temperature reaches the limit.

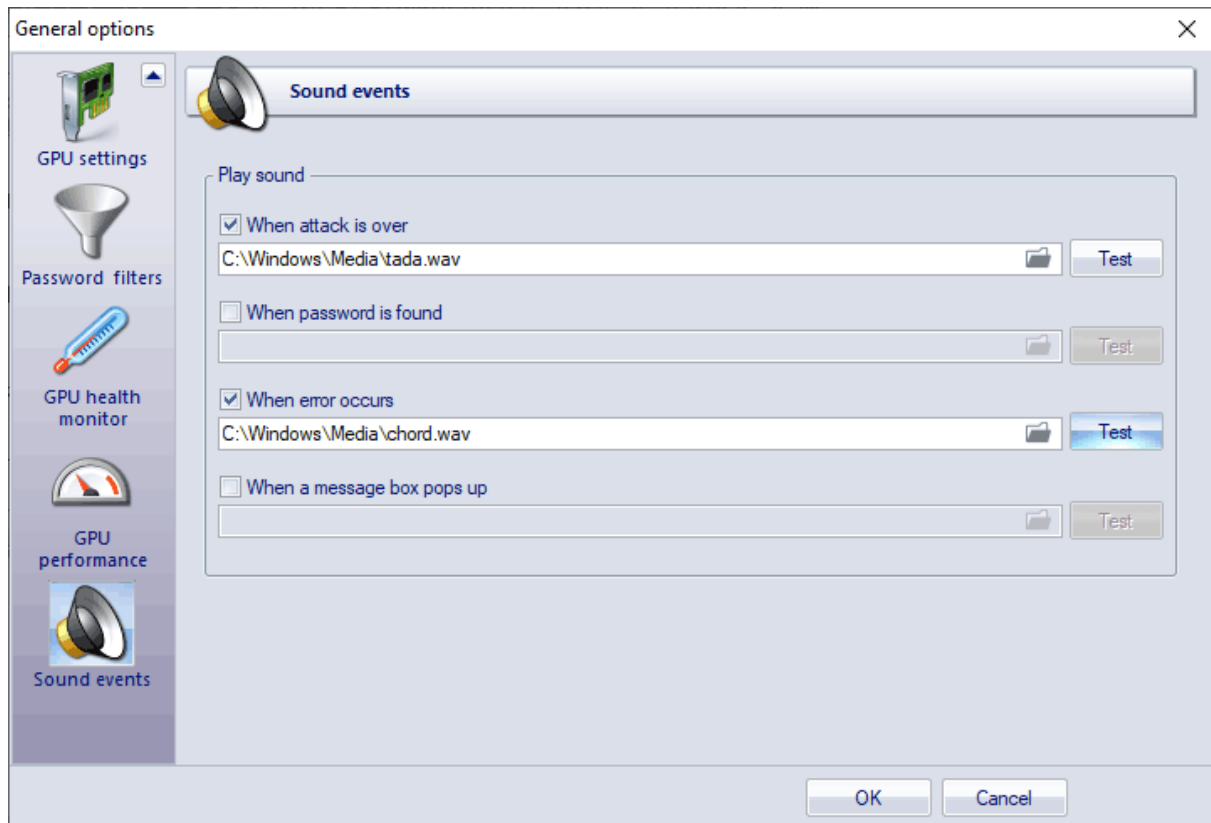
Warning! Once an AMD device is selected, the program supports for abort temperature option only. AMD drivers do not allow overriding some values manually for most cards and have numerous bugs that may cause system instability or even BSOD. Consider using the Radeon built-it tool to set up your own performance profile instead.

2.8.1.7 GPU performance



New versions of the program come with the GPU performance adjustment option now. By default, the program automatically detects best GPU load value. You can, however, adjust this value by switching to the manual mode. For example, if you want to watch a video while the program is brute-forcing WiFi passwords, reduce the GPU performance to minimum possible.

2.8.1.8 Sound notifications

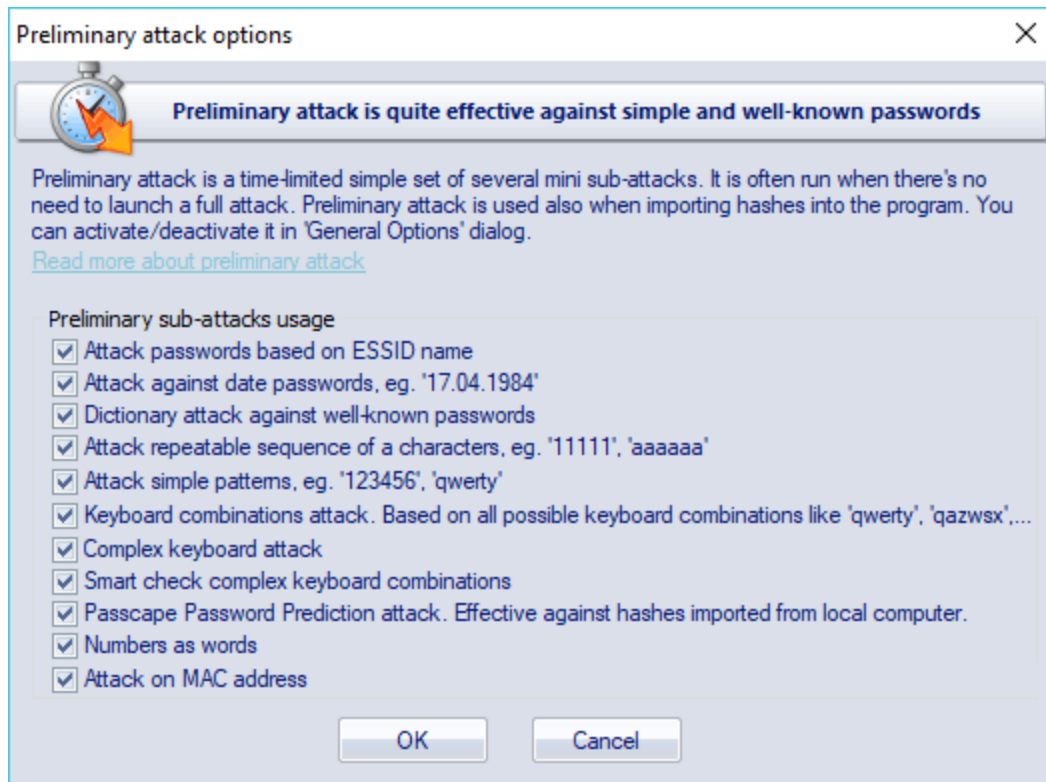


The software allows setting up sound notifications for certain events. For example, when the attack is over or when a password is found.

2.8.2 Attack Settings

2.8.2.1 Preliminary attack

Preliminary attack (developed in Passcape) is quite effective against short, simple, dictionary, repetitive, keyboard, etc. passwords and consists of several mini-attacks. Each mini-attack can be enabled/disabled individually.



Preliminary attack run about 5-6 minutes or even faster. It consists of at least the following sub-attacks:

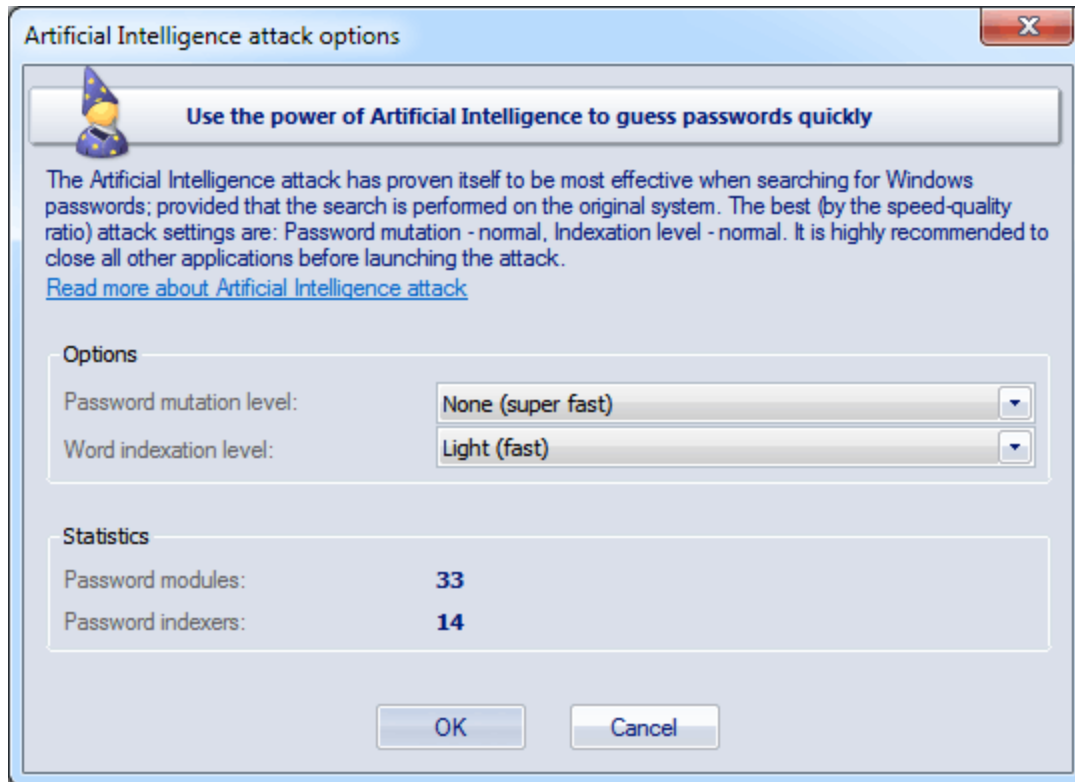
- Attack against SSID name. Verify various combinations of the wireless network name.
- Dates attack. Checks passwords that were based on date. For example, 12031994.
- Simple dictionary attack. Fast check the password by verifying all words from a given dictionary.
- Attack on repeatables. Checking passwords as a repeatable sequence of a character. Eg. '1111111' or 'xxxxxxx'.
- Attack on simple patterns, like '123456' or 'qwerty'.
- Keyboard attack checks for keyboard passwords and all possible combinations. Eg. 'qwer', 'qazwsx', 'asdzxc', etc.
- Complex keyboard attack is the same as previous 2 attacks, for compound keyboard patterns.
- Smart keyboard attack checks complicated keyboard combinations.
- Passcape Password Prediction attack is the most complicated and state-of-art password prediction tool.
- Attack passwords based on numbers (as words)
- Guess passwords based on MAC address

2.8.2.2 Artificial intelligence attack

Artificial Intelligence Attack is a new type of recovery developed in our company. It is based upon a social engineering method and has never been implemented in password recovery applications yet.

This one is mostly used when the hashes are imported from the local computer. Intellectual attack scans the local computer, indexes and creates the list of found words and passwords, analyzes them, upon the results of the analysis produces user's preferences, performs the mutation of the found words and, based on all that, attempts to recover the passwords.

This attack allows, without resort to time-consuming and costly computations, to almost instantly recover certain passwords encrypted with hash functions. The basic idea behind the Artificial Intelligence attack is that an average user very often chooses similar words and word combinations or follows the same password generation rule when creating one's passwords. With that in mind, we could attempt to figure that rule out and pick the original password.



Although this sounds somewhat abstractive, in the reality the attack clearly splits into four successive steps.

1. Initiating the collection of private data. Here comes into action the password retrieval and indexation module, which looks for all available and hidden in the system passwords entered by user at any moment of time. Those include network access passwords, ICQ, email, FTP, Windows account passwords, server passwords, LSA Secrets, etc.
2. Launches the data collection and indexation module. During the execution of this step, we analyze the activity of the user (or all users, if the indexation module selected is different than Light) in the system. Next, basing upon that, we generate the list of words - potential passwords selected from the text files, archives, internet browsers' history, email correspondence, etc.
3. Includes the semantic analysis module for the database of found passwords and the list of potential passwords.
4. On the final stage, the data analysis module will perform the mutation of the words and attempt to pick the passwords.

In the beginning of the attack, the program will search the system for all passwords it knows of. For that purpose, there are currently 38 mini modules for decrypting system, mail, browser, messenger, archive and other passwords. Then there goes the file and data indexation, along the course of which the program generates a potential attack dictionary. The third module breaks the passwords and words into pieces, out of which in the last module it will assemble new combinations for picking and guessing the original password.

Naturally, the more complex is the mutation and indexation level, the more efficient will be the search. However, reaching the topmost indexation and analysis level may take hours and even days, depending on the speed of the password validation algorithm and the number of users in the system.

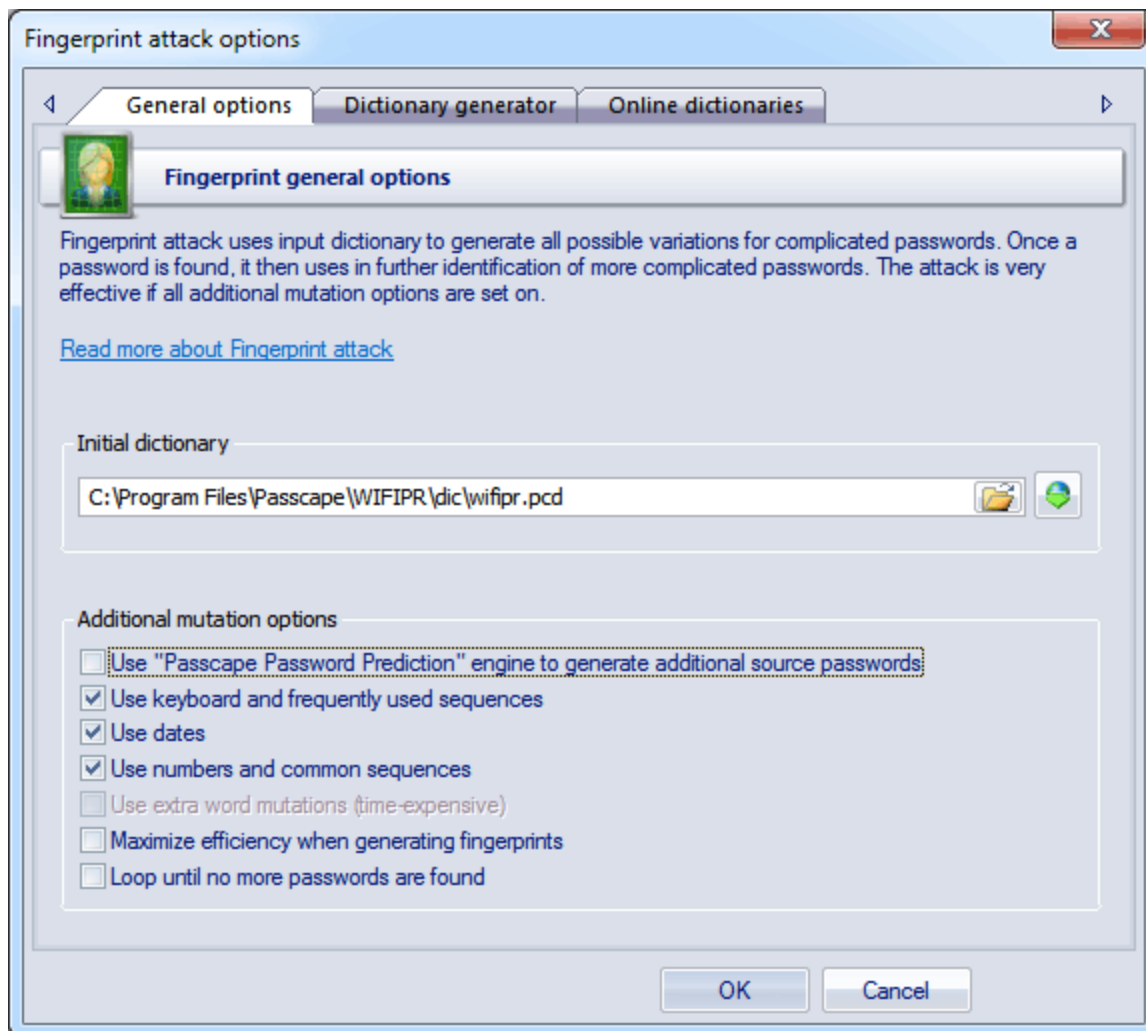
The Artificial Intelligence attack has proven itself to be most effective when the search is performed on the original system. Only two options are available here: password mutation depth and word indexing level. The most preferred options for running a speedy attack are *Light:Light*. For a deeper (and at the same time slower) search, set these options to *Normal* or even *Deep*. The duration of an intellectual attack also depends on the configuration of your system, and other factors.

It is highly recommended to shut down all other programs before launching the attack. If your Artificial Intelligence attack runs very slow, you may need to remove your program's cached password (for example, when the total amount of cached passwords exceeds 10000).

2.8.2.3 Fingerprint attack

Fingerprint attack is a brand-new tool for recovering complex passwords, which could not be decrypted by other attacks. The idea of the attack is that here, to recover a password, we take neither individual words from the source dictionary, like in the dictionary attack, nor even word combinations, like in the combined attack, but so-called "fingerprints". Now, every source word from the dictionary is used for generating several fingerprints. If some password is found during the attack, it participates in generating new fingerprints, and the attack goes another round.

Before launching the attack, specify the source dictionary to be used for creating the fingerprint bank. The software comes with a dictionary, `common.pcd`, optimized for this attack, but you can use yours or download one off the Internet ('Online dictionaries' tab). There are no certain requirements to the dictionary, except one: the source dictionary must not be too large; otherwise, the attack will take significant time. You can use dictionaries with national passwords, if you suspect that the sought password contains characters in a national encoding.



Here is the way to generate fingerprints: first, break each word from the source dictionary into one-character passwords, then - into 2-character, etc. For instance, break the source word **crazy** into one-character fingerprints. We get:

c
r
a
z
y

Now, two-character:

cr
ra
az
zy

Next, three-character:

cra
raz
azy

And, finally, four-character:

craz
razy

We have got $5+4+3+2=14$ fingerprints, not counting the source word.

Repeat this for each word of the source dictionary. After this, all the fingerprints are dumped into a single database, naturally, discarding duplicates. We have got a database of fingerprints that would be used for checking passwords by gluing all the fingerprints with each other and finding the match.

The real fingerprint generation algorithm is much more sophisticated. Moreover, there is an option in the attack settings, **Maximize efficiency when generating fingerprints**, which uses a more sophisticated algorithm, which maximizes the efficiency (at the expense of speed) by generating additional fingerprints.

Let's take a look at the remaining options.

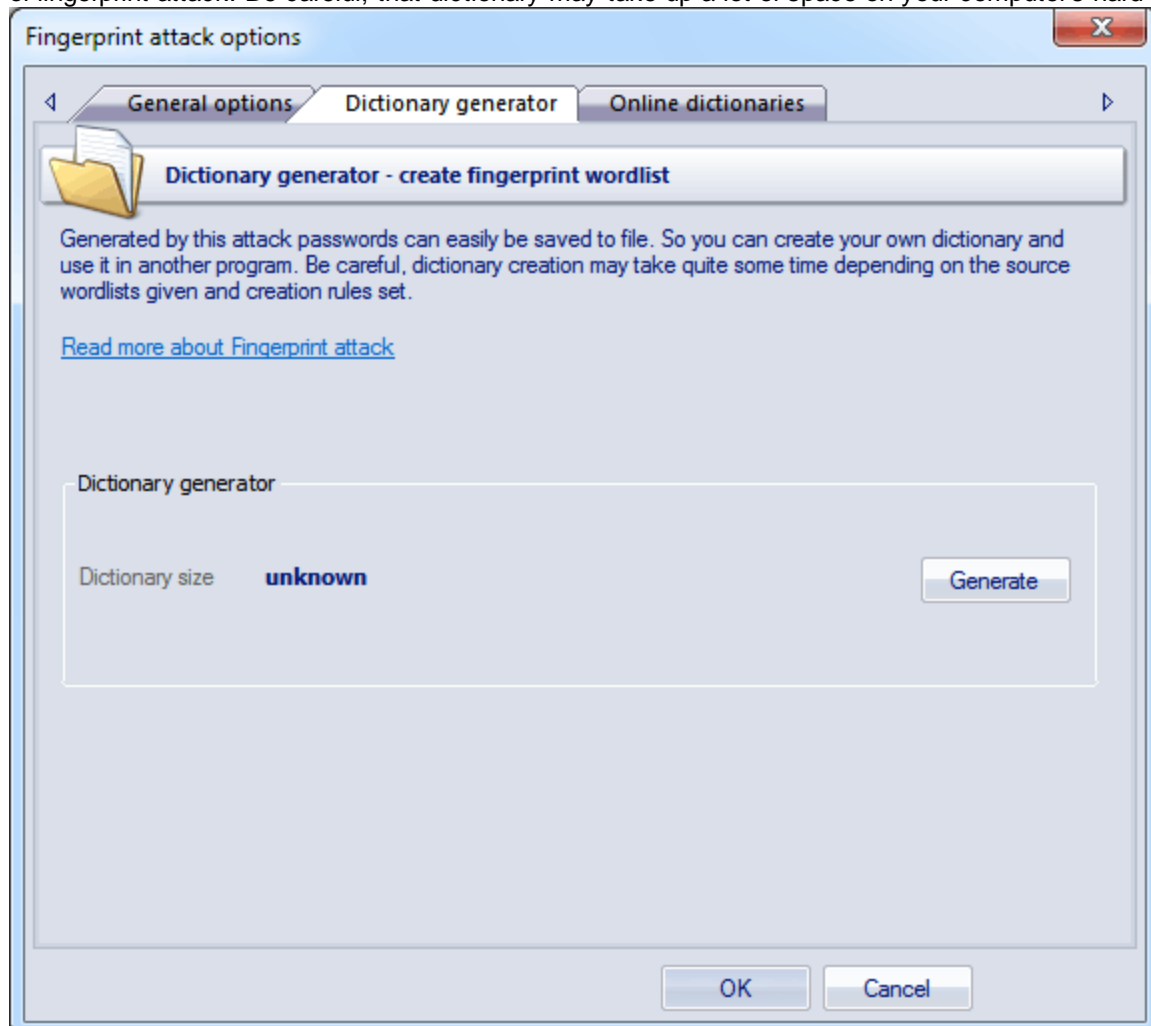
Use PPP engine to generate additional passwords - use passwords found in other attacks when generating fingerprints.

Use keyboard and frequently use sequences - add keyboard combinations and common sequences to fingerprint bank.

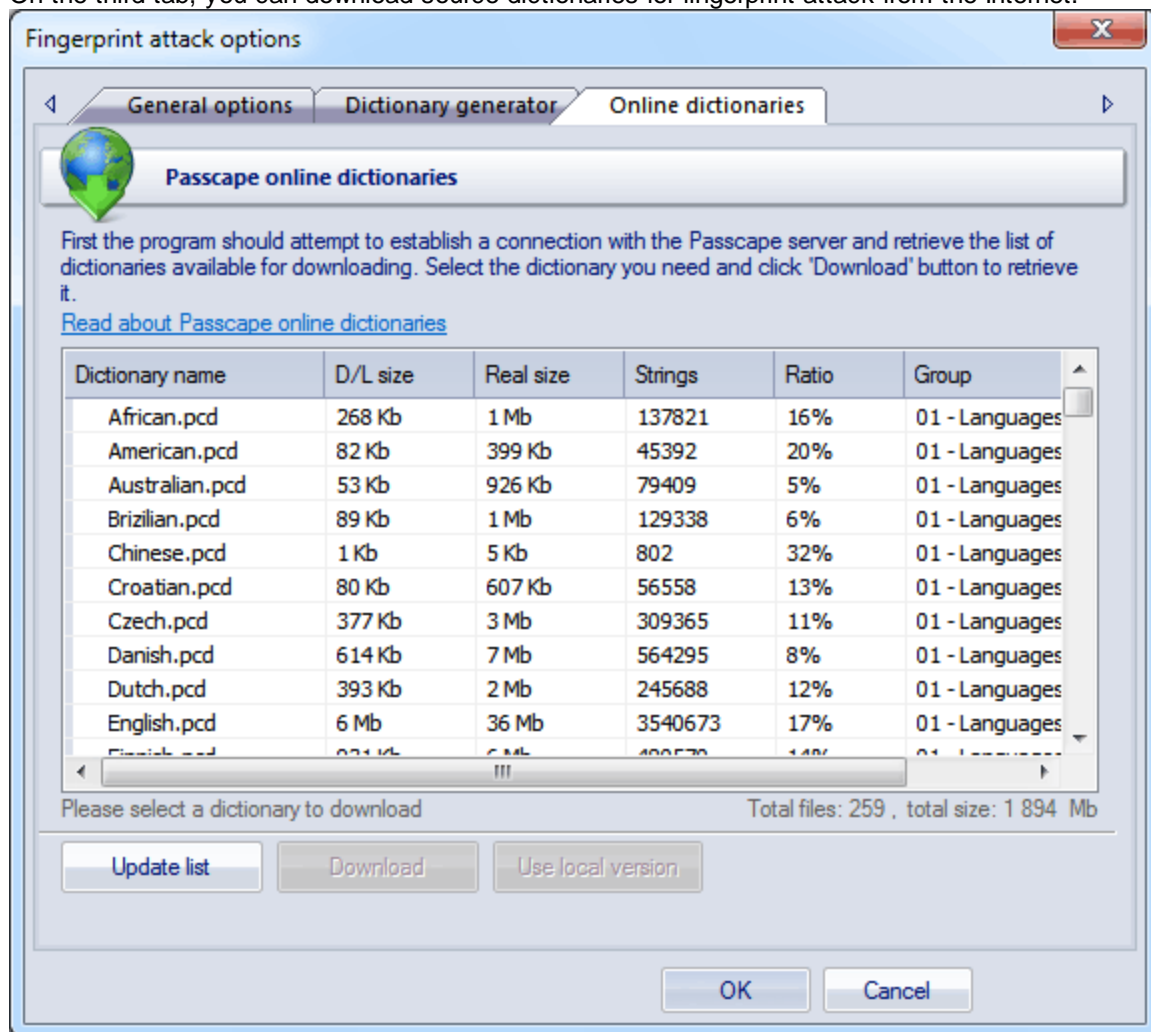
Use dates - add dates to fingerprints.

Use numbers and common sequences - use digits and simple combinations of letters.

The second tab with the settings allows to create and record a custom dictionary using current options of fingerprint attack. Be careful; that dictionary may take up a lot of space on your computer's hard disk.



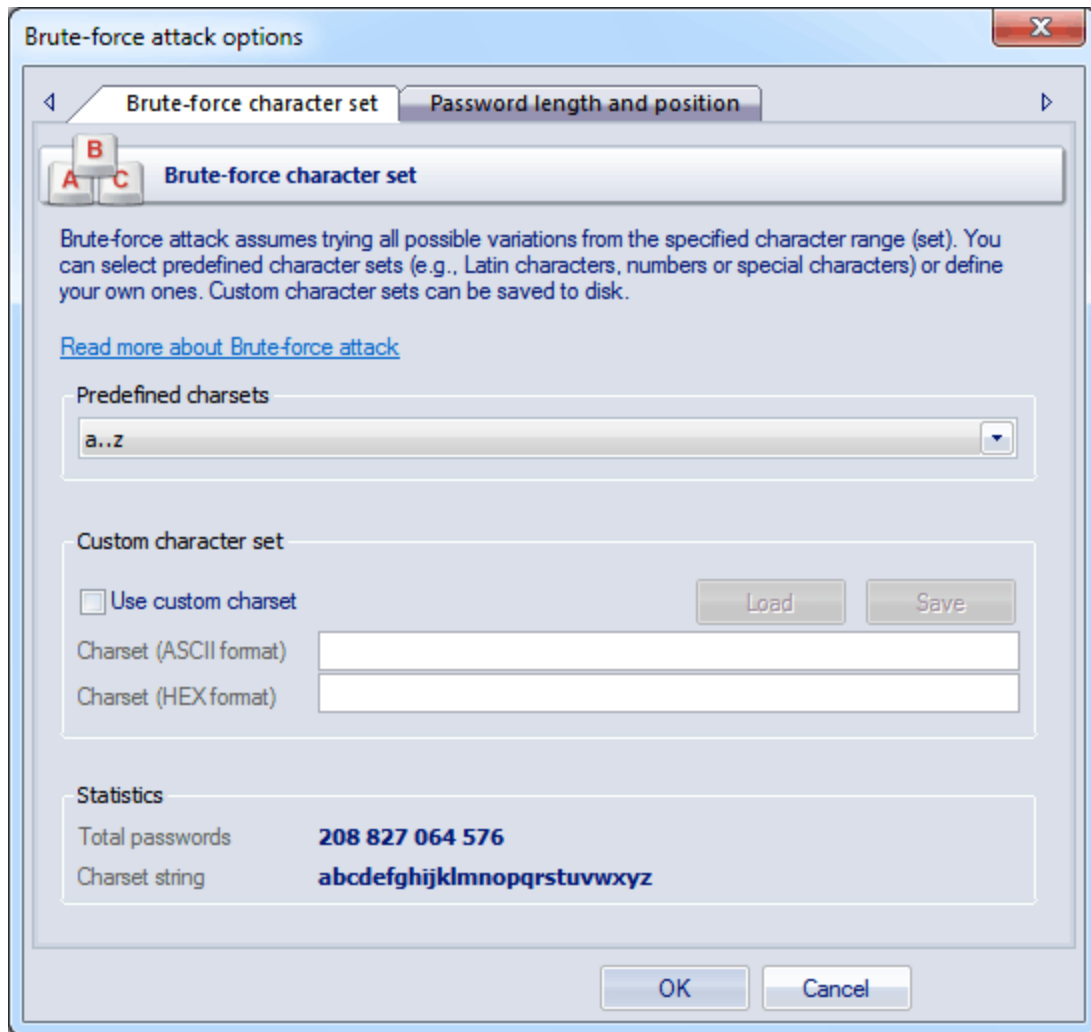
On the third tab, you can download source dictionaries for fingerprint attack from the Internet.



2.8.2.4 Brute-force attack (exhaustive search)

In cryptanalysis, a brute force attack is a method of defeating a cryptographic scheme by trying a large number of possibilities; for example, exhaustively working through all possible keys in order to decrypt a message. This definition was taken from [Wikipedia site](#).

Well, to put it in simple words, brute-force attack guess a password by trying all probable variants by given character set. Eg. checking all combination in lower Latin character set, that is 'abcdefghijklmnopqrstuvwxyz'. Brute-force attack is very slow. For example, once you set lower Latin charset for your brute-force attack, you'll have to look through 208 827 064 576 variants for 8 symbol password. Brute-force attack is rarely used in WPA hashes recovery. It must be used only if other attacks have failed to recover your password.



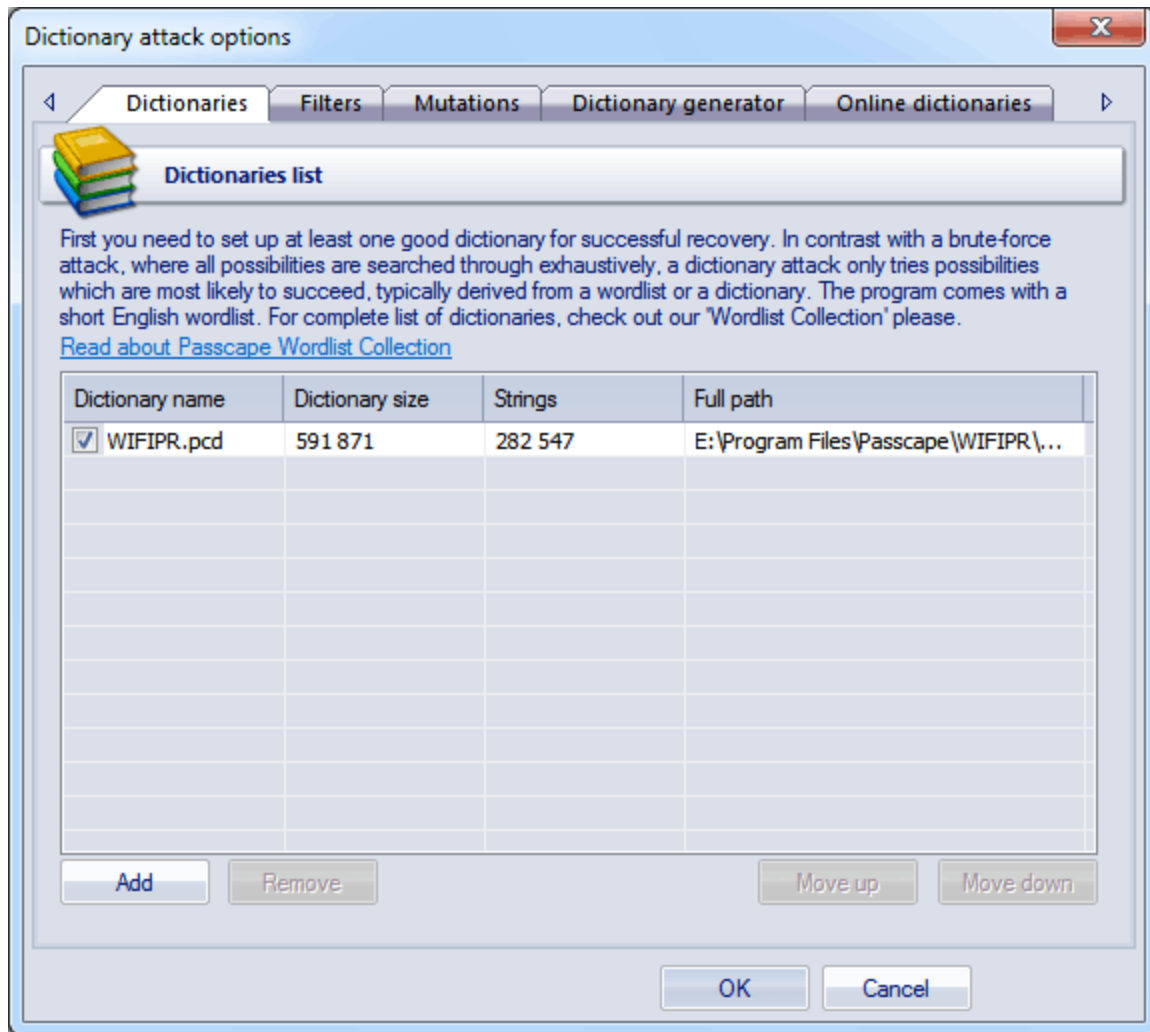
The brute-force attack options consist of two tabs.

The first tab is for setting the range of characters to be searched. You can use the predefined sets or create your own ones. To define your own character set, select the option '*Custom charset*'. This will enable two fields for defining a custom character set: the first one - for entering ASCII characters, second one - for entering non-printable characters. You can save your custom character set on disk. The program comes with several examples of user-defined character sets.

On the second tab, set the minimum and maximum length of the passwords to be searched. You can also set a starting password, which would start the search.

2.8.2.5 Dictionary attack

In contrast with a brute-force attack, where all possibilities are searched through exhaustively, a dictionary attack only tries possibilities which are most likely to succeed, typically derived from a wordlist or a dictionary. Generally, dictionary attacks succeed because many people have a tendency to choose passwords which are short, single words in a dictionary, or are simple variations that are easy to predict.



On the *'Dictionaries'* tab, set up the list of dictionaries to be used in the attack. Supported are plain-text dictionaries in the formats ASCII, UNICODE and UTF8, as well as encrypted/compressed dictionaries in the native PCD format, developed in Passcape Software. ZIP and RAR packed wordlist are supported as well with some restrictions. To deactivate a dictionary, simply clear the checkbox by its name. In this case, the dictionary, although it remains on the list, will be skipped during an attack. The software comes with a 280000-word dictionary. For complete list of dictionaries, check out our [wordlist collection](#) please. Or you can use our [online dictionaries](#) as an alternative.

The *'Filters'* tab filters the words from a dictionary by the include/exclude principle. If the first, inclusive, filter is enabled, the attack will accept only the words that contain at least one of the characters entered in the filter. If the second, exclusive, filter is set, the program will skip the words that contain at least one of the entered characters.

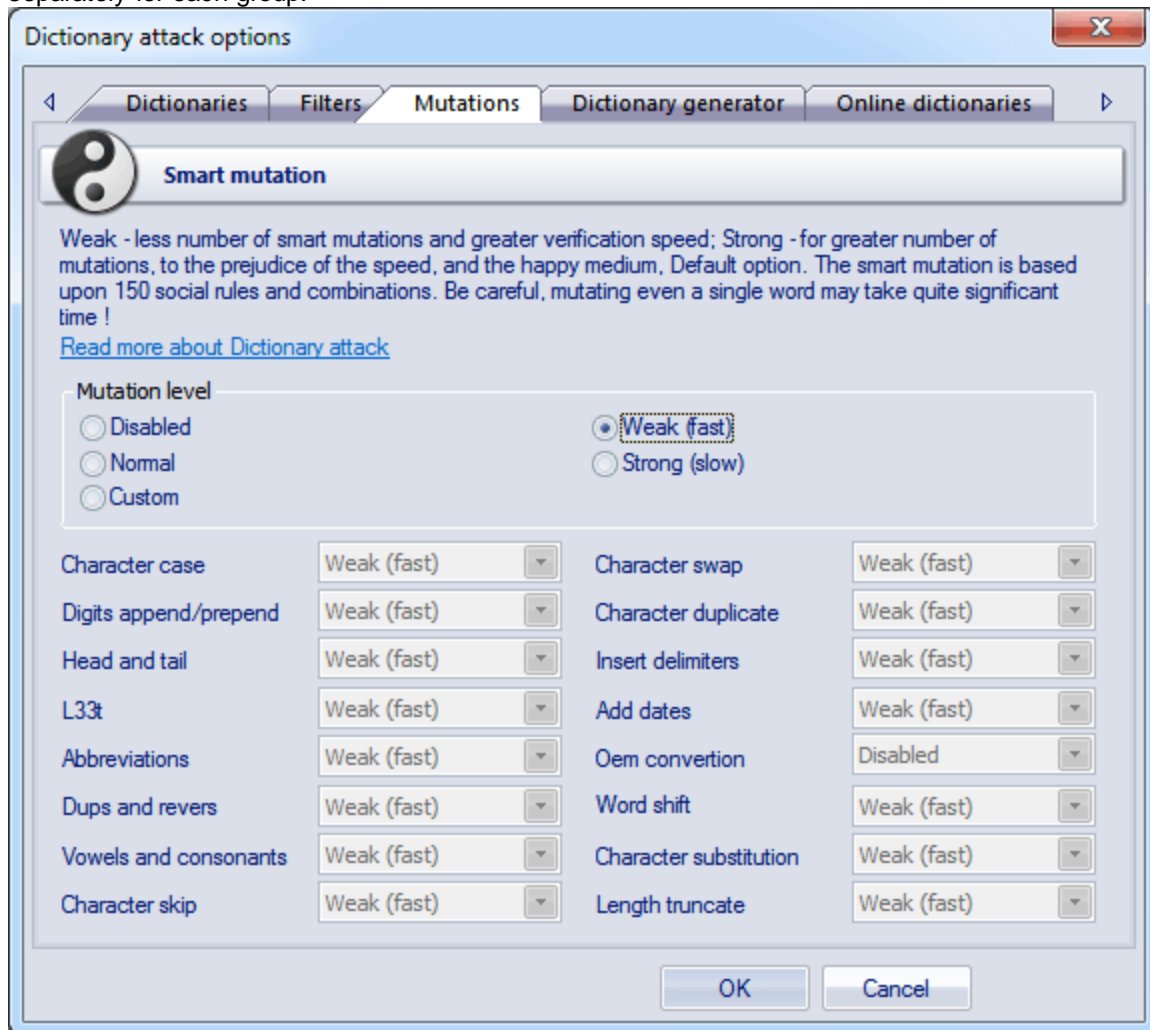
The *'Mutation'* tab allows setting all kinds of possible combinations of the words to be searched. For example, if you set a strong mutation, the program will create several hundreds of analogs for each word from the dictionary. For example, secret - Secret - s3cr3t - secret123, and so on. You can set up to three mutation rules: *Weak* - less number of mutations and, in its turn, greater verification speed; *Strong* - for greater number of mutations, to the prejudice of the speed, and the happy medium, *Default* option.

You can use *Dictionary Generator* to create your own wordlists based on options of the first three tabs.

Online dictionaries. The program has a great feature that allows downloading and using existing dictionaries available on the Passcape website. We have accumulated quite a large dictionary collection - over 250 items. That should get you rid from the extra hassle on finding the required content on the Net.

Customizing mutations

The program has ability to customize the smart mutation of the Dictionary attack. All mutation rules are clustered into 16 primary groups. You can set one of three mutation levels or disable mutation separately for each group.



Simple description of what all these mutation groups mean is given below:

Group name	Description	Examples (for word 'password')	Comments
Character case	Checks case combinations of the input word.	Password, PassworD, PaSsWoRd	Maximal (Strong) level of the mutation group DOES NOT generate all possible case combinations of input words. To check all possible case variants,

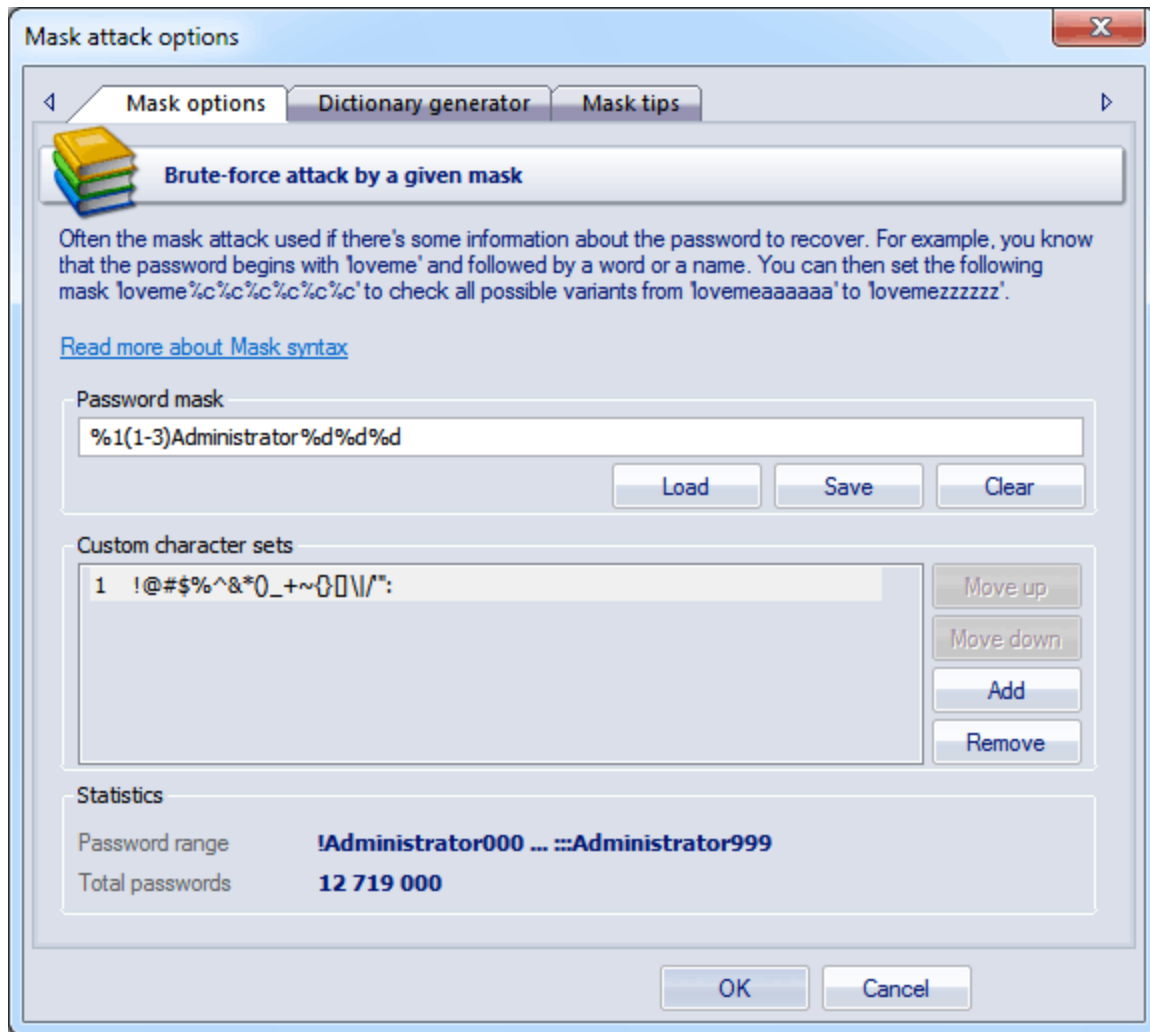
Group name	Description	Examples (for word 'password')	Comments
Digits append/prepend	Adds digits to the beginning or to the end of the word.	password99, 2Password, PASSWORD3	consider using Hybrid dictionary attack (aN rule) Maximal level adds 2 digits.
Head and tail	Almost the same as previous one, but appends or prepends words, abbreviations, characters, keyboard combinations, etc.	#Password#, password12345, 4everPASSWORD, Passwordqwerty	
l33t	Creates different combinations using leet language .	p@ssword, P@\$w0rd, P@\$\$W0RD	
Abbreviation	Converts several character combinations (if the initial word contains any) into abbreviations.	ihateyou -> ih8you, lh8u	
Dups and revers	Revers, duplicates the word, etc.	drowssap, passwordpassword, PasswordDrowssap	
Vowels and consonants	Mutates vowels and consonants (English characters only).	Psswrđ, PaSSWoRD, pAsswOrd	
Character skip	Skips a single character of the original word.	assword, Passwrđ, Pasword	
Character swap	Exchanges two adjacent characters.	apssword, Passowrd	
Character duplicate	Duplicates characters.	ppassword, ppaasswwoorrd, Passworddddd	
Delimiters	Separates characters with delimiters.	p.a.s.s.w.o.r.d, P-a-s-s-w-o-r-d	Maximal level uses 10 delimiters.
Dates	Adds dates to the end of the word.	Password2010, password1980	Even though the mutation engine can generate more complicated variations (for example, password03171998 or Password19710830), this feature if turned off here even in maximal mutation level.
Oem conversion	Converts English word into another language and vice-versa using alternative keyboard layout (second language of the OS).	If your OS has 2 languages installed (let it be English and Russian), the program will convert initial word password into Russian and Russian will be converted into gfhjkm .	The program works correctly for 2 or even more languages. So if you have 5 languages installed locally (including English one), there will be 4 different combinations of the input word.
Word shift	Stupidly shifts all characters of the word to the right or to the left.	asswordp, dpasswor	
Character substitution	Replaces a character of the initial word.	oassword, passqord	This is quite helpful rule assuming the fact that the

Group name	Description	Examples (for word 'password')	Comments
			characters for substitution are taken from a special table. For example, the character 's' will be replaced with the following ones: 'a', 'w', 'e', 'd', 'x', 'z'. You can notice that all of these characters are located near 's' on any qwerty keyboard.
Length truncate	Truncates word length to probe all possible length combinations.	passwor, passwo, Pass	

2.8.2.6 Mask attack

Mask attack is an irreplaceable tool when you know a fragment of the password or have any specific details about it. For example, when you know that the password consists of 12 characters and ends with the *qwerty*, it is obvious that searching the entire 12-character range of passwords is unreasonable. All what would be required in this case is to pick the first 6 characters of the sought password. That is what mask attack is for.

In our case, we could define the following mask: **%c%c%c%c%c%cqwerty**. That means that the program would serially check the following combinations: aaaaaaqwerty .. zzzzzqwerty. If the original password is 'secretqwerty', it perfectly hits our range.



The mask entry field is used for setting the rule, by which the program will try to recover the password. If the mask is set correctly, below you will see the range of characters generated by the mask. User-defined masks can be saved to disk. You can also use the mask tool to generate a dictionary (may not be available in some editions).

The mask syntax is quite trivial and consists of static (unmodifiable) and dynamic (modifiable) characters or sets. Dynamic characters/sets always have a leading %. For example, if you set the mask `secret%d(1-100)`, the program will generate 100 passwords (`secret1`, `secret2`: `secret100`).

Wireless Password Recovery supports the following dynamic mask sets:

- %c lower-case Latin characters (a..z), 26 symbols
- %C upper-case Latin characters (A..Z), 26 symbols
- %# full set of special characters (!..~ space), total 33 symbols
- %@ small set of special characters (!@#\$%^&*()-_+= space), 15 symbols
- %? all printable characters with ASCII codes of 32..127
- %d one digit (0..9)
- %d(x-y) numbers between x and y inclusive
- %r(x-y) user-defined characters with serial ASCII codes between x and y

- %r(x1-y1,x2-y2...xn-yn) set of several non-overlapping sequences of ASCII characters. Useful for defining custom character sets; e.g., of OEM characters.
- %[1..9] a character from user defined charset 1..9
- %[1..9](min-max) user-defined range of variable length (from min to max). You can set up to 9 your own custom character sets.
- %% standalone static character %

When setting %r, keep in mind that the range of defined OEM characters (with character code greater than 127) is generated using the DOS encoding.

Examples:

```
test%d           - will generate password range test0..test9, 10 passwords total
test%d(1980-2007) - test1980 .. test2007, 28 passwords
test%r(48-57,97-122) - test0 .. testz, 36 passwords
%#test%#        - _test_ .. ~test~, 1089 passwords
admin%1(1-5)     - admina .. adminzzzzz, where %1 is user defined charset 1 (a..z)
%1%1%1pin%2%2%2 - aaapin000 .. zzzpin999, %1 is user character set a..z and %2 is second user-defined charset which contains characters 0..9
```

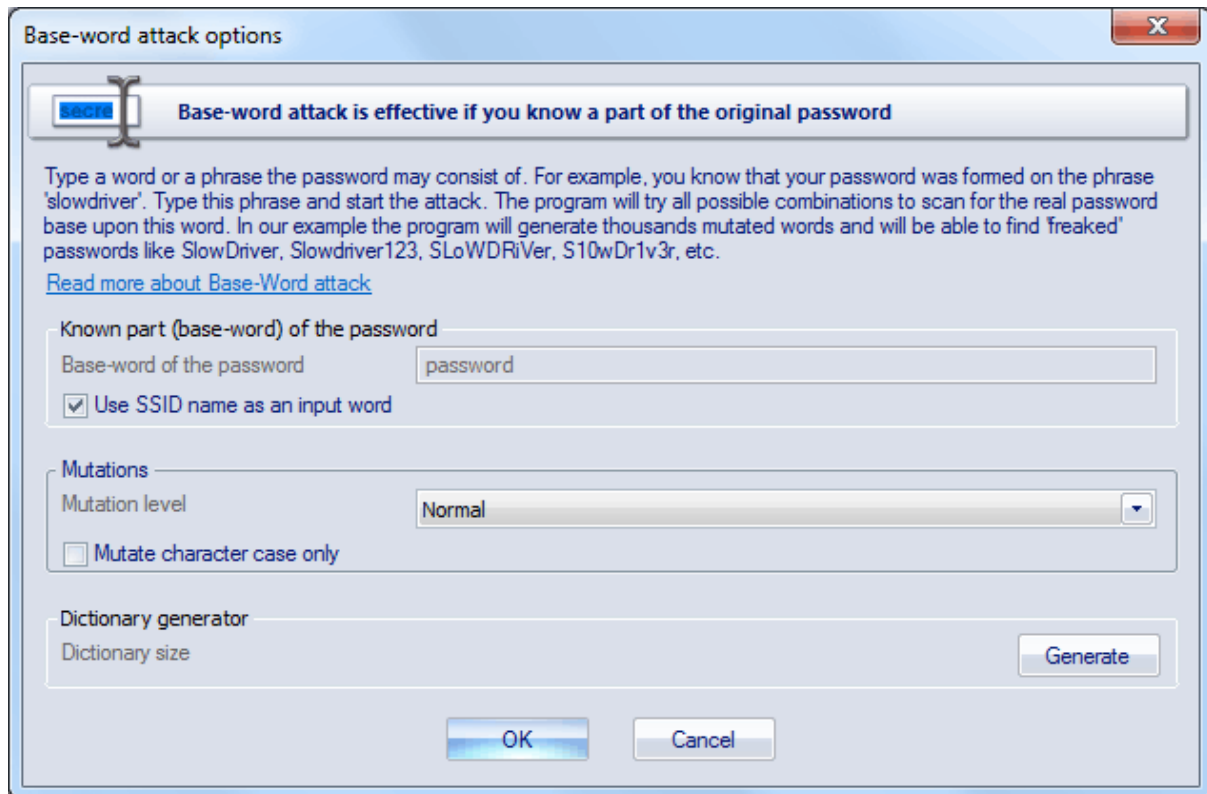
By switching to **Dictionary generator tab**, you can generate your own dictionary by a given mask, and save it to disk. This feature available in Advanced edition of the program only.

Third tab of the mask options contains a short description of the mask syntax and a couple of simple examples.

2.8.2.7 Base-word attack

Base-word attack (developed by Passcape) is in many ways similar to mask attack. However, here you don't need to set up the syntax; simply enter the keyword, which supposedly was the base word for the password. It is an irreplaceable recovery tool when you know a portion of the password or its basic component. Normally, such cases dispose to using mask attack; however, it does not always allow coping with the task set forth. Suppose our password was 'S10wDr1v3r'. Trying to recover such a complicated password using brute-force attack would be an ungrateful job, even if you are quite sure that it is based upon the 'slowdriver' word. These are the cases when the base-word attack will rescue you.

With this tool, the program will attempt to recover the original password, trying all possible combinations founded upon 15 groups of rules (total over 150 rules.) If you enter 'slowdriver' in the field, you will see that the program has generated several thousands of different combinations upon this phrase, and one of those combinations could match our password.



If the length of the phrase exceeds 8-10 characters, the mutation may take significant time. If you remember the original password precisely and simply have forgotten the sequence of the upper-case and lower-case characters in it, you can select the option 'Use only case mutations'. With this option selected, the program will generate passwords with all possible combinations of upper-case and lower-case characters, total 2^n passwords, where n - is password length. For example, for the password 'slowdriver' the program will generate $2^{10}=1024$ different combinations for each keyboard layout installed on your computer. You can also generate a dictionary on those mutations and save it on a disk (available not in all editions).

Turn on the 'Use document name as an input word' option to let the program automatically select the base-word phrase based on SSID name.

Note, if your password length exceeds 15-16 characters, it may take quite some time to prepare (mutate) the password for the attack.

2.8.2.8 Combined dictionary attack

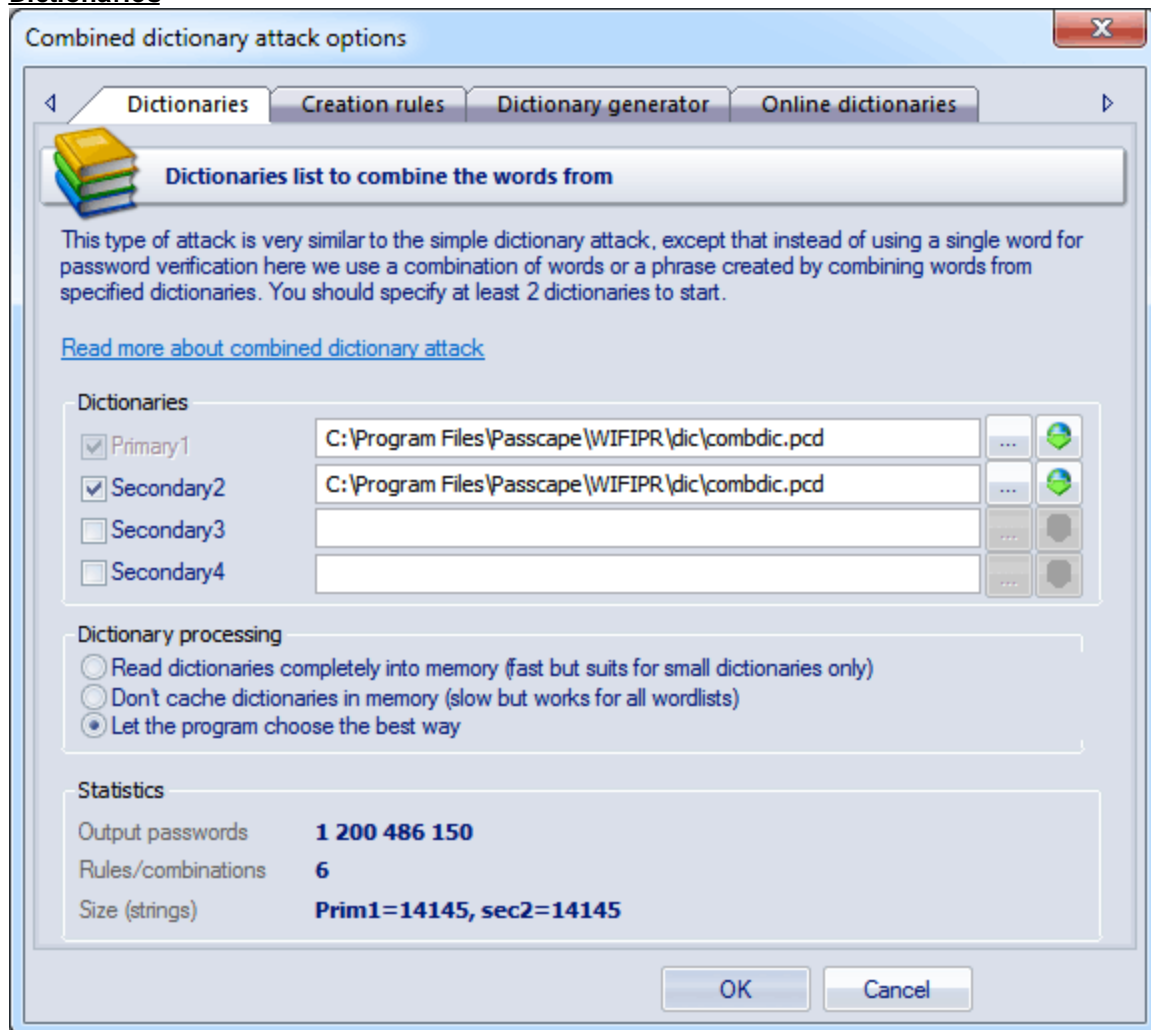
Combined dictionary attack (developed by Passcape Software) is great at recovering passwords that consist of 2,3 and even 4 words. This type of attack on difficult and compound passwords is very similar to the simple dictionary attack, except that instead of using a single word for password verification here we use a combination of words or a phrase created by combining words from specified dictionaries. To successfully utilize this attack, set at least two dictionaries and the rules for generating passwords. You can set the regular dictionaries used in the simple dictionary attack, but it is recommended to use rather small dictionaries with the most common words. Perfect dictionaries for the combined pass phrase attack are those that have different forms of words in them; e.g. jump, jumper, jumped, jumping.

Combined attack sets a certain limit to the number of dictionaries that can be used; that's not more than 4. Thus, the general limitation of this attack is that only password phrases of not more than 4 words can be recovered using this attack.

Another essential drawback is the wide range of phrases generated. And, as the consequence, the proportional increase of the time spent on the validation of a password. Keep in mind that when generating passwords that consist of 3 or 4 words, the generation process takes considerable time

If finding the right dictionary is difficult, don't worry. The software comes with a special dictionary for the combined attack. You can also take advantage of the [Online Dictionaries](#) tab or the corresponding button to download such dictionaries from the Passcape website.

Dictionaryes



The way the combined attack works is really simple. For example, if you have set two dictionaries, the program will generate the passwords as follows: it will take the first word from the first dictionary and glue it with the first word from the second dictionary, then with the second word, and so on until the end. Then it checks the second word from the first dictionary and goes the same route, and so on.

To understand how the combined attack works, let's take a look at a couple of password generation examples that involve, in the first case, the same dictionary and in the second case - two different ones.

1. Suppose we've got a single dictionary with three words: action, bad, and computer. We will set this dictionary as two original sources: primary dictionary & secondary dictionary2 (see the figure). After these dictionaries have been processed, at the output we have the following phrases (they will be used when checking the password sought):

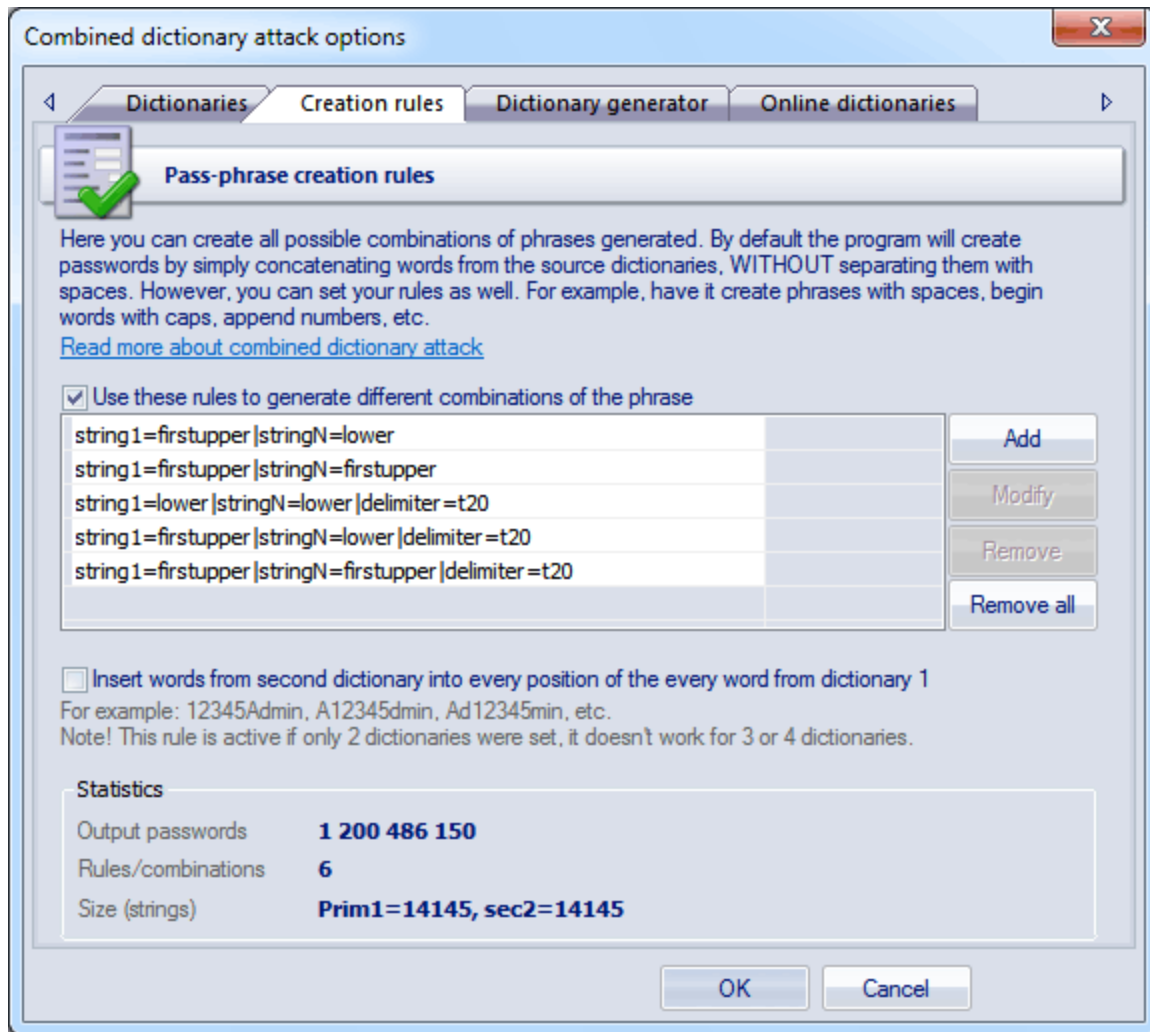
'actionaction', 'actionbad', 'actioncomputer'
'badaction', 'badbad', 'badcomputer'
'computeractio', 'computerbad', 'computercomputer'.
9 phrases total.

2. In the second case, we have got two different dictionaries. For example, the first dictionary consists of three words: action, bad, and computer. The second one also has three words: date, eagle, fail. In this case, we are going to have the following phrases:

'actiondate', 'actioneagle', 'actionfail'
'baddate', 'badeagle', 'badfail'
'computerdate', 'computereagle', 'computerfail'.

The example is plain but demonstrative. The idea is that for multiple sources you can successfully use both a single dictionary and multiple ones. It all depends on your imagination. The last example shows that a special attention should be paid to the order of the dictionaries if they are different. The order of the words in the phrases to be created depends directly on the order of the source dictionaries. In our second example, if we swap the primary and the secondary dictionaries, at the output we will obtain a completely different set of phrases.

Mutation rules



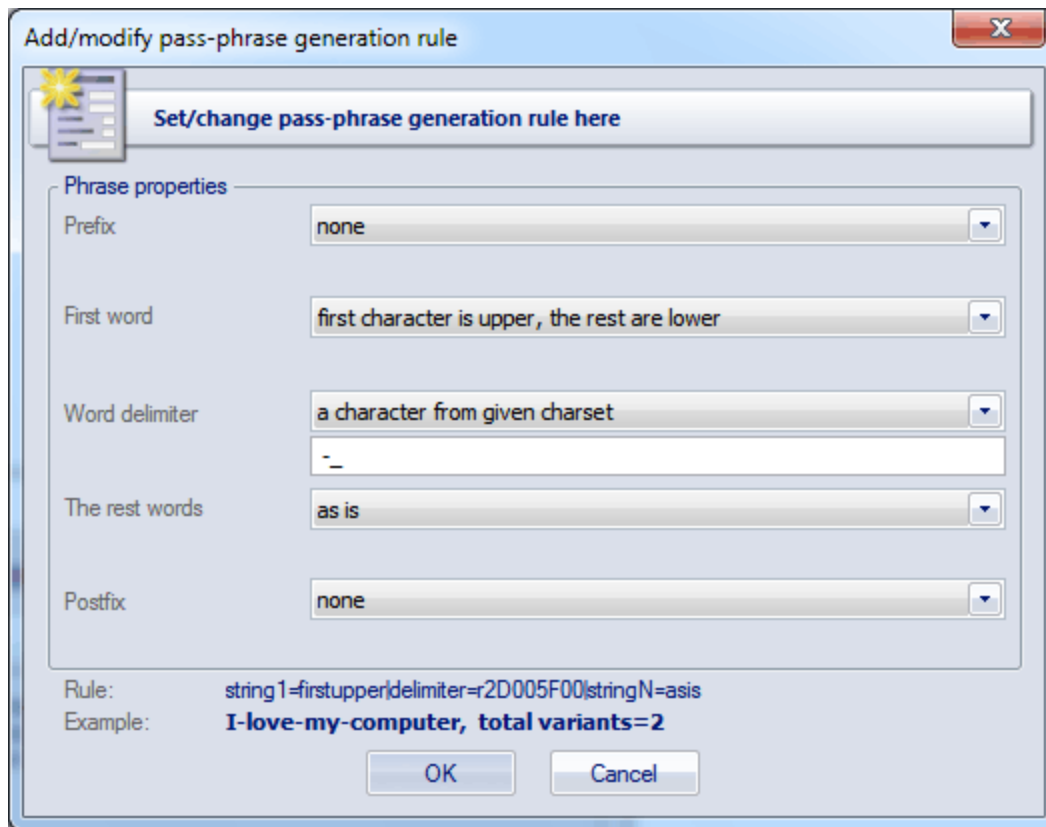
Passwords created by the combined attack are generated according to special rules that are to be set on the second tab. By default, when password generation rules are disabled, the program generates passwords by simply gluing up the words from the dictionaries, without separating them with a space. For example, of the two words are 'my' and 'computer', you will get 'mycomputer'.

If word insertion option is set, the program additionally creates passwords by inserting words from second dictionary into every position of the word from dictionary 1. For example, if the first dictionary's word is **Admin**, and the word from the second dictionary is **12345**, the program will generate the following passwords:

12345Admin
A12345dmin
Ad12345min
Adm12345in
Admi12345n

And so on for all words of the second dictionary. Then goes another word from dictionary 1, etc. The option is active if only 2 dictionaries were set.

The generation rules are made to extend the password search options. For example: Mycomputer, MyComputer, MY COMPUTER, my-computer, etc. There are special rules available for this purpose; you don't have to know the syntax of them, for the mutation rule creation dialog is simple and intuitive.



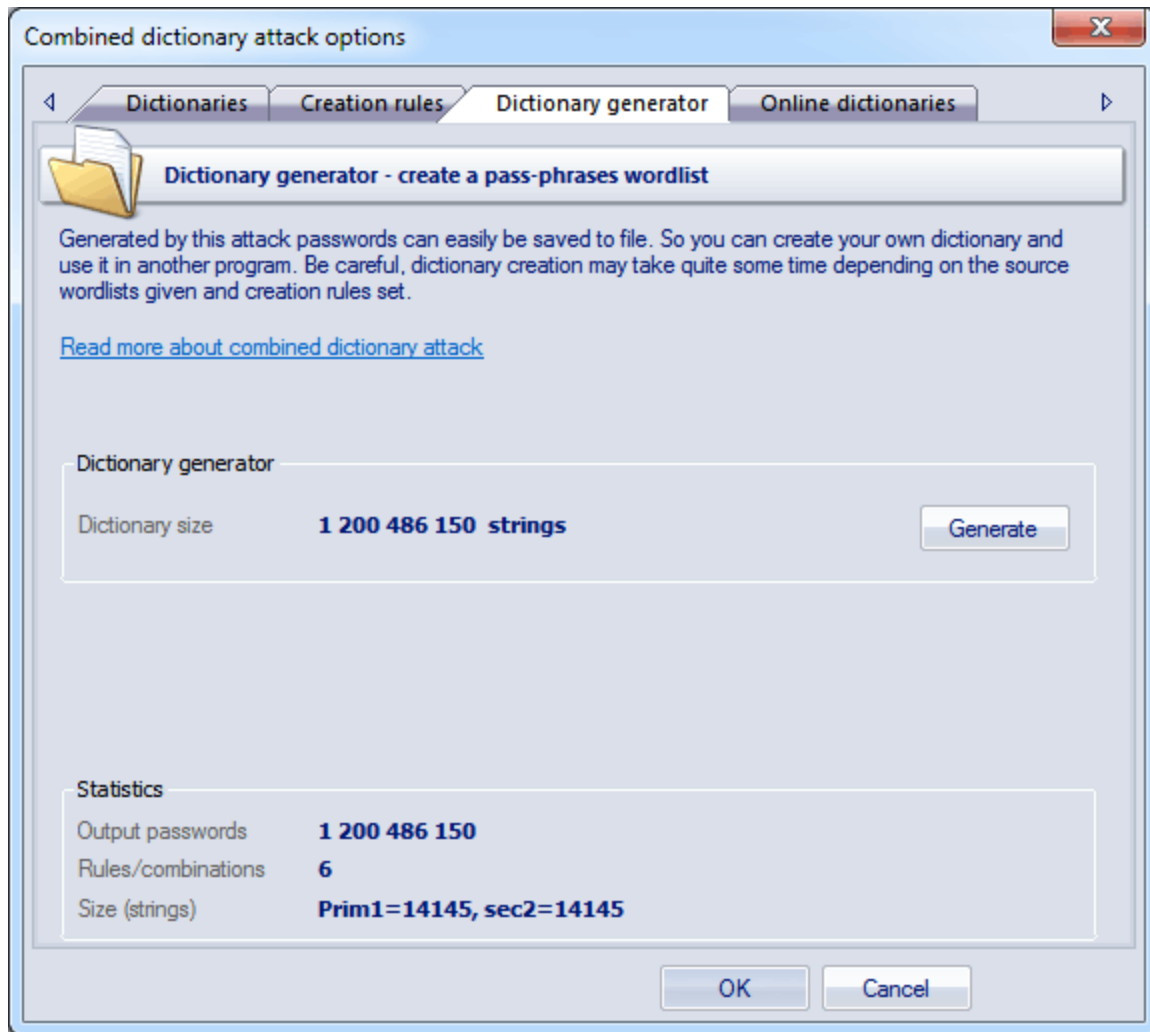
Each mutation rule consists of five elements:

1. *Prefix* - text that will appear before each phrase. This element can be a character from given charset, plain text string, one digit between 0 and 9 or a number. For instance, if you set a one-digit prefix, the phrases created with this rules will look as follows: '0 aaa bbb', '1 aaa bbb' : '9 aaa bbb'.
2. *First word* - the action to be performed over the first word of each phrase. There are only four options. Namely: leave intact as is in dictionary, convert all characters to lowercase, convert all characters to uppercase or capitalize only the first letter of the word.
3. *Word separator*. It may be absent. Then all the words will be concatenated. Example: 'aaabbb', 'aaaccc', 'aaadd', etc. You can otherwise set a custom separator; e.g. the '-' character: 'aaa-bbb', 'aaa-ccc', 'aaa-ddd'. Or you can set a range of characters.
4. *Other words*. With this attribute, similarly to 2., you can set rules for the other words of a phrase.
5. *Postfix* - text that will finalize each phrase. For example, if you set Postfix to the '?' or ' ?', all phrases created with this rule will have the question mark at the end.

Certainly, the more password generation rules you set, the more chances you have to pick the right password. But, on the other hand, the more time you will have spent on the attack.

The 'Statistics' group shows the average and recommended average size of a dictionary, number of words in source dictionaries, total number of passwords being generated and other helpful information.

Dictionary generator



The third tab of options serves for creating combined attack-based dictionaries (available not for all editions).

You can also [download additional dictionary modules](#) from the Passcape Software Web site.

2.8.2.9 Pass-phrase attack

More and more users choose to make up their passwords of entire phrases, passages from poems, movie aphorisms, Latin aphorisms, etc. Attempting to recover such passwords using the traditional techniques is unthinkable, even with the reference to the advancement of the computing power of modern computers. Therefore, the recovery help comes with the predefined and known phrase attack.

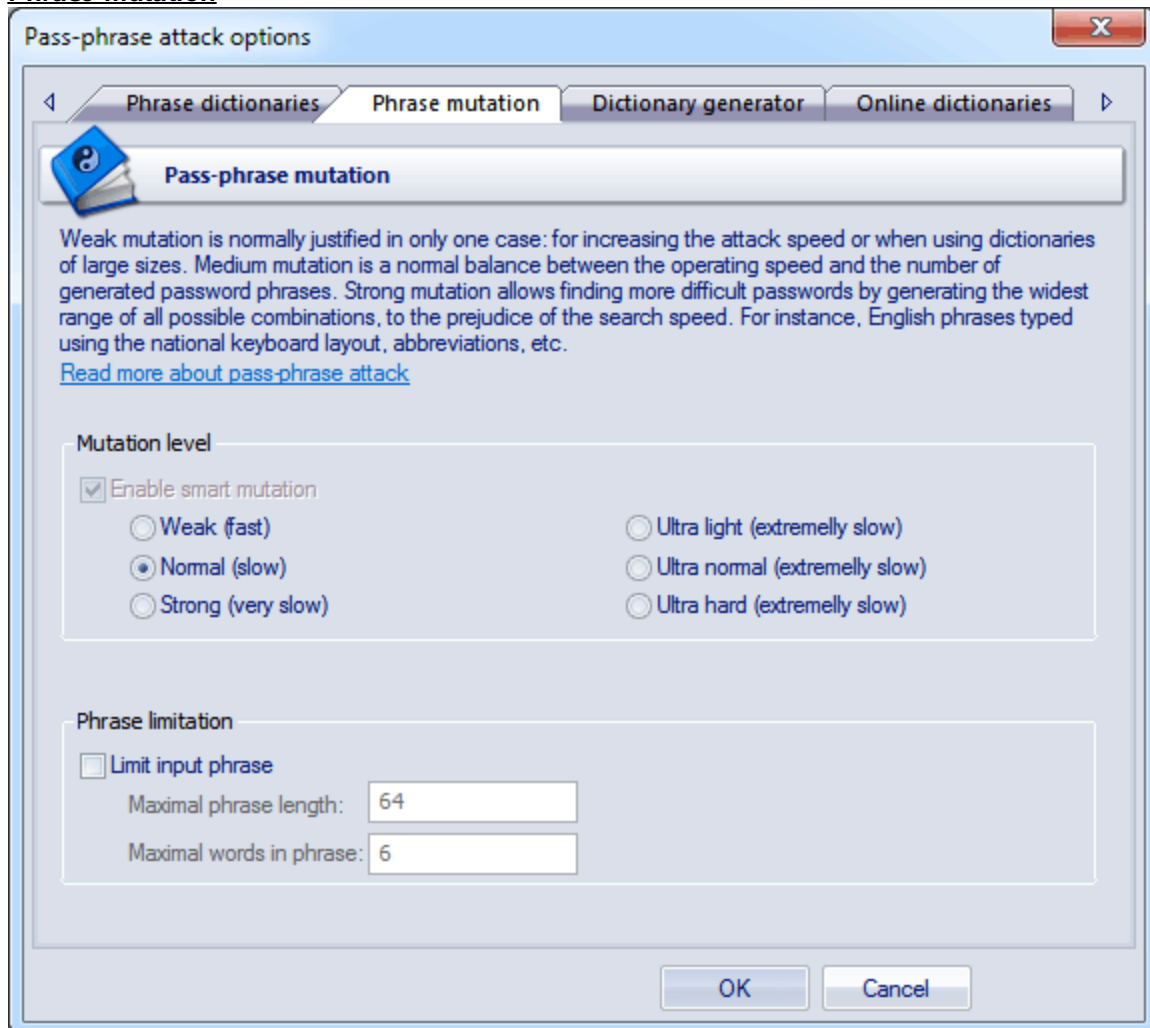
Pass-phrase attack is by much similar to the simple dictionary attack, except that here the password search goes phrase by phrase instead of going word by word. The main idea of the attack is to guess the right password by searching through predefined frequently used expressions, phrases and word combinations.

For example, if the sought password is made of the widespread phrase 'To be or not to be', it is obvious that this is the only attack that has the virtue to cope with such a password. In order to do that, you are

Wireless Password Recovery comes with a short dictionary of phrases and aphorisms.

[illegible]

Copyright (c) 2020 Passcape Software. All rights reserved.

Phrase mutation

Mutation is worth saying more, since as you should have known strong mutation significantly raises chances for the successful recovery. Weak mutation is normally justified in only one case: for increasing the attack speed or when using dictionaries of large sizes. Medium mutation is a normal balance between the operating speed and the number of generated password phrases. Strong mutation allows finding more difficult passwords by generating the widest range of all possible combinations, to the prejudice of the search speed. The greater is the mutation level, the more passwords the attack will cover.

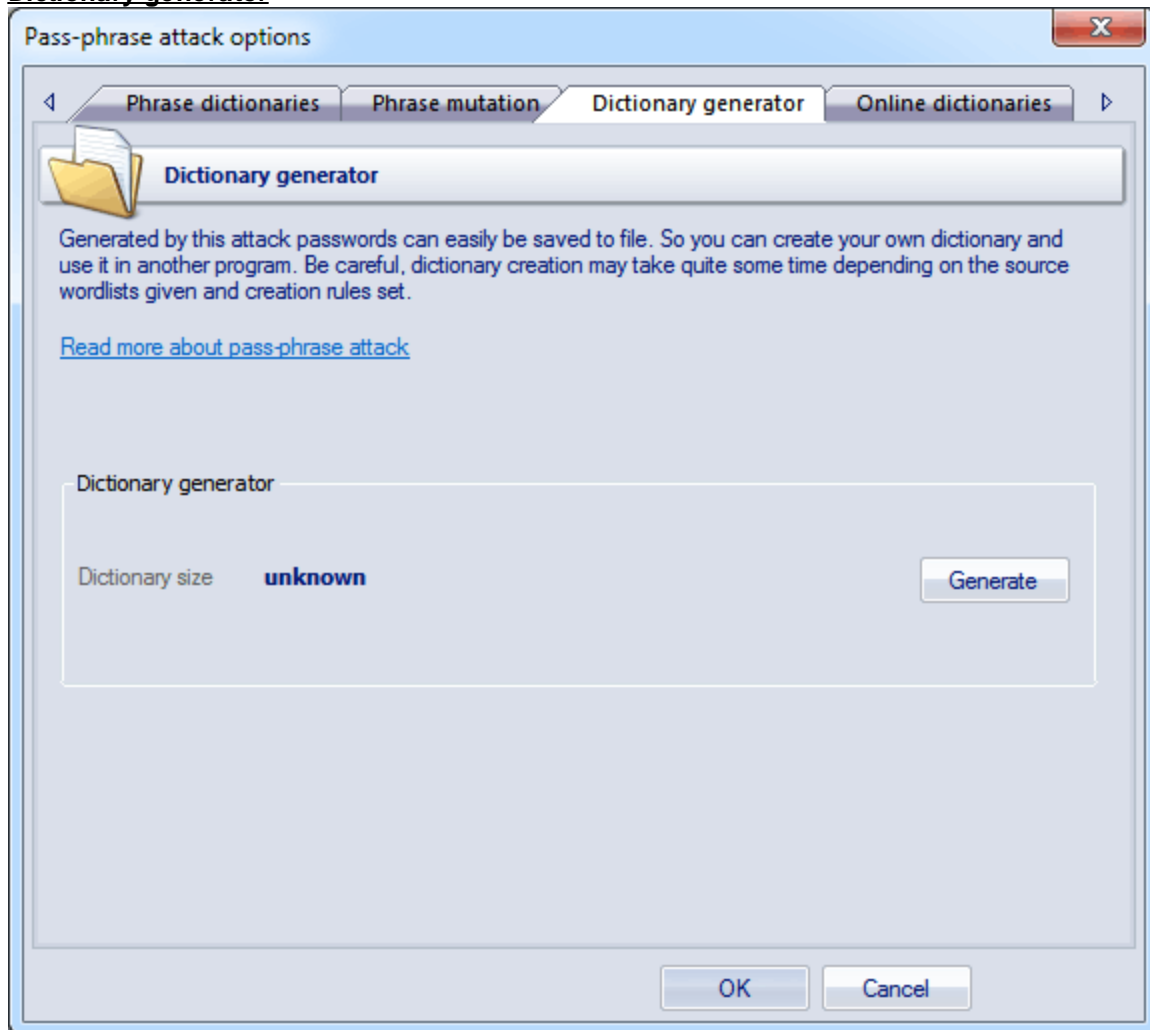
Major difference in mutation levels:

- Weak - simplest thus fastest mutations.
- Normal - the same as Weak, but generates several additional mutations and case combinations.
- Strong - the same as normal plus more mutations and national passwords (according to the installed keyboard layouts, if any).
- Ultra light - this is a 2-step mutation because every generated in Weak mode password goes through the second mutation round (one used in Weak mode of the simple dictionary attack).
- Ultra normal - 2-step mutation. Every password generated in Normal mode is used as a source to generate additional combinations by implementing additional Normal mutation level.

- Ultra hard - every password generated in Strong mode is used as a source to generate additional combinations by using additional Strong mutation level.

Be careful! Ultra modes generate a great number of passwords, thus the attack may be ran extremely slow. To speed up the attack, consider setting up input phrase limits. For example, you can limit input phrases to 6 words and 27 characters.

Dictionary generator



The third tab uses for creating pass-phrase dictionaries.

2.8.2.10 Hybrid dictionary attack

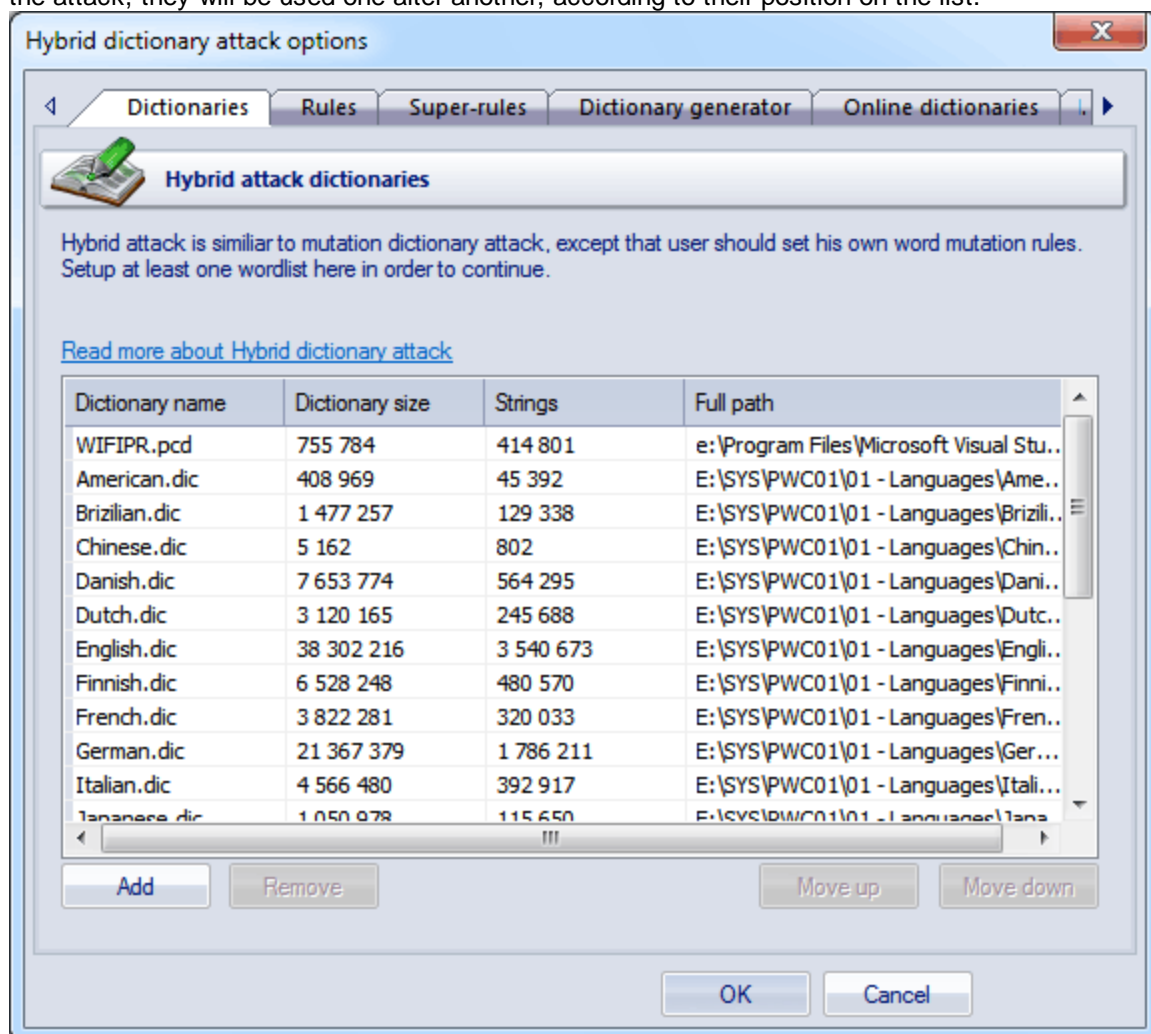
Hybrid dictionary attack is a form of [simple dictionary attack](#). However, unlike the latter, hybrid attack allows user to set his own word mutation (variation) rules and attempt to validate the modified words as source passwords. For example, user could capitalize the first letter of a password being validated, append '2' to it, replace the number 8 in it with the letter B, O with 0, etc.

Actions, performed on source words from the dictionary, are called rules. Multiple rules can be applied to each source word.

Hybrid dictionary attack settings are grouped in 7 tabs:

1. **Dictionaries** - for setting up source dictionaries.
2. **Rules** - files with set of rules.
3. **Super-rules** - ones to be applied over the top of regular rules
4. **Dictionary generator**, where you can create files of words obtained from the hybrid attack.
5. **Online dictionaries** - for downloading new dictionaries to the application.
6. **Attack syntax** - complete description of all rules with examples.
7. **Rule tester**, where you can test your rules.

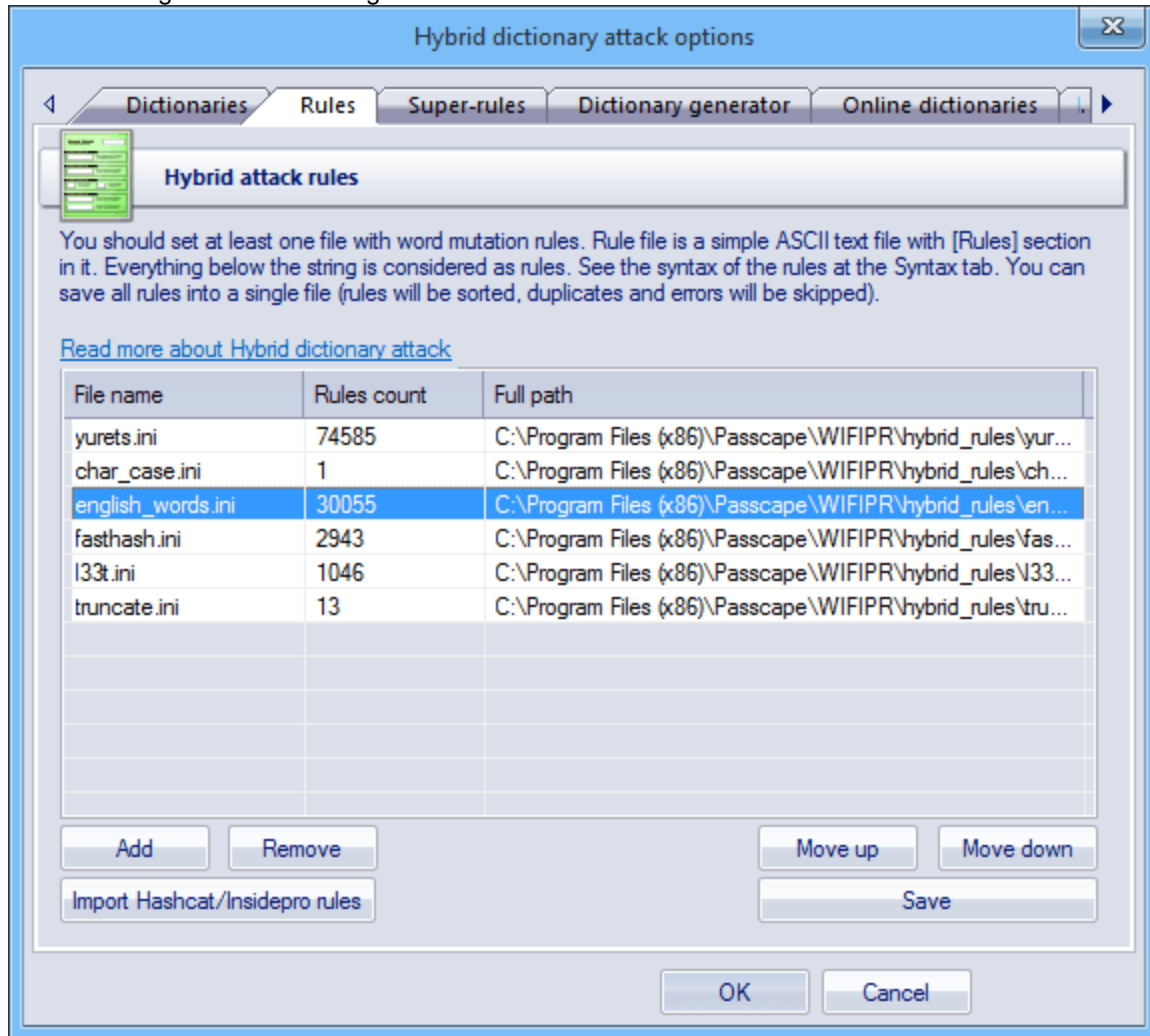
Wordlists to be used in the attack are set on the first tab. Traditionally, the application supports wordlists in ASCII, UTF8, UNICODE, PCD, RAR and ZIP format. The position of the files on the list can be altered. For example, you may want to move smaller dictionaries up the list or the other way. During the attack, they will be used one after another, according to their position on the list.



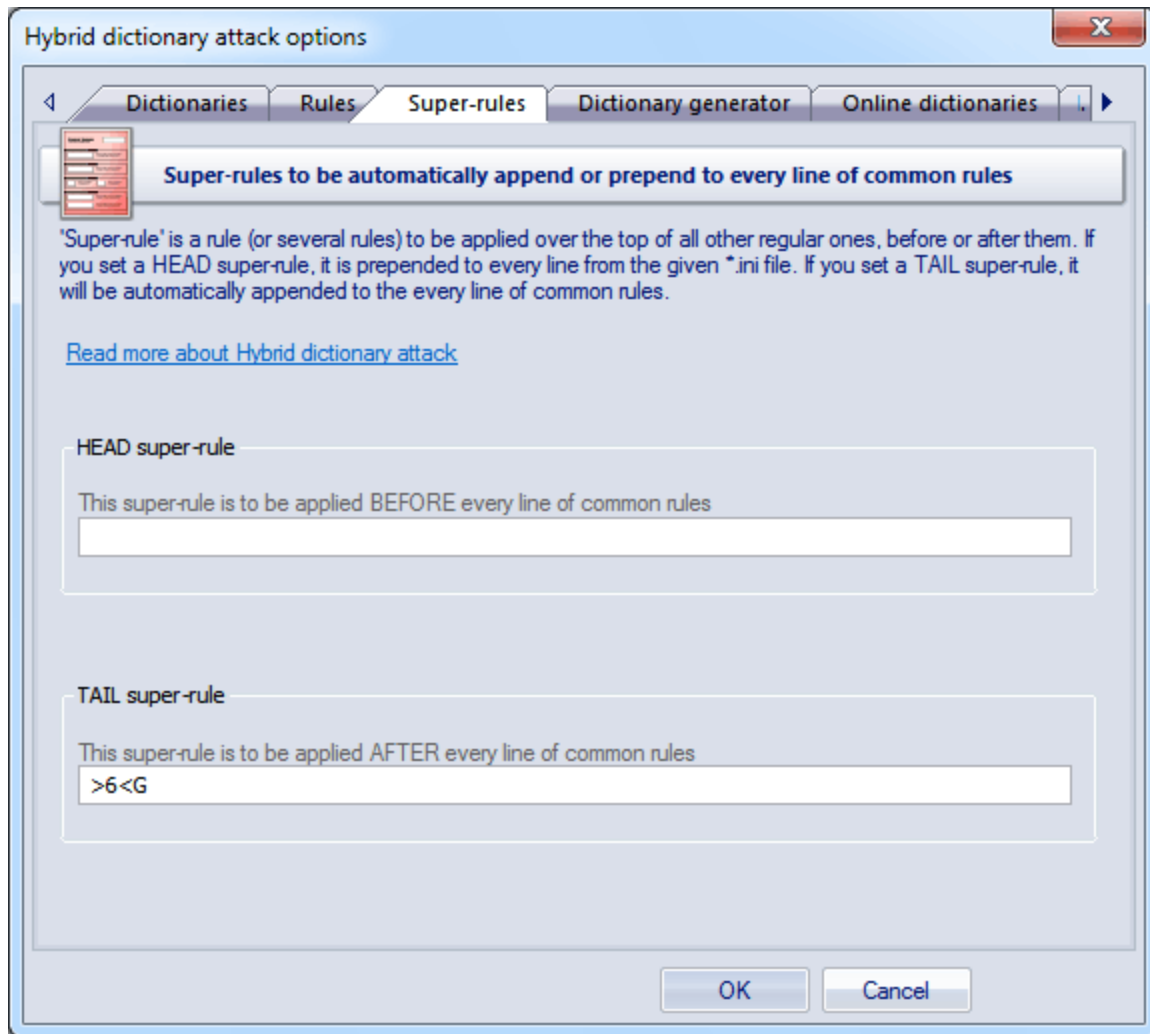
On the **'Rules'** tab, define at least one file with password mutation rules. The format of the rules file is quite trivial; it is a plain-text ASCII file with the **'[Rules]'** string. Anything above this string is considered comments and ignored by the program. Whatever goes below this string is considered rules. Each string can contain several rules, applicable to a source word. If a string contains multiple rules per word, those

rules are parsed left to right. For example, if you apply the rule '@pc\$ab\$bc' to the source word 'password', at the output you will get 'Asswordabc'. The maximum length of an output word may not exceed **256** characters.

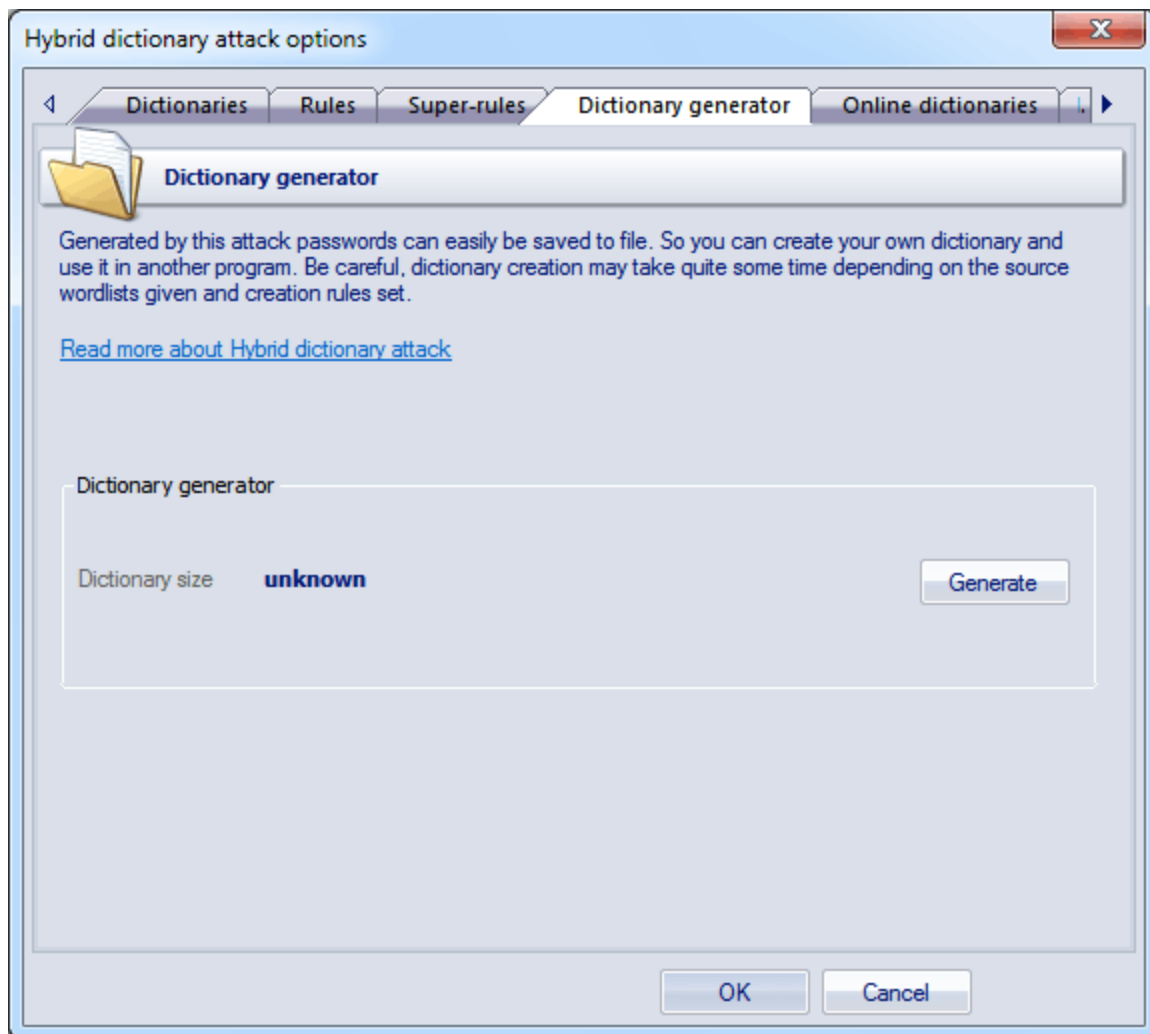
The RWP's mutation rules are similar pretty much to those used by many password recovery software. You can import your favorite rules from HashCat, InsidePro or JohnTheRipper here, as well as to combine/merge them into a single file.



'Super-rule' is a rule (or several rules) to be applied over the top of all other regular ones, before or after them. For example, you can set 'a8' tail super-rule to create all possible case combinations after a common mutation has been done. So '/asa4' rule from l33t.ini file will become '/asa4a8', '/csc(' will become '/csc(a8', etc. Yet another one example: setting the '>6<G' head rule allows you to skip all words of less than 6 or greater than 16 characters, before starting a common mutation. This is a helpful feature once you decide to add the same rule to all text lines of the selected *.ini files. There's no need to modify them all. Be careful though, the 'aN' super-rule may increase the total number of generated passwords drastically.

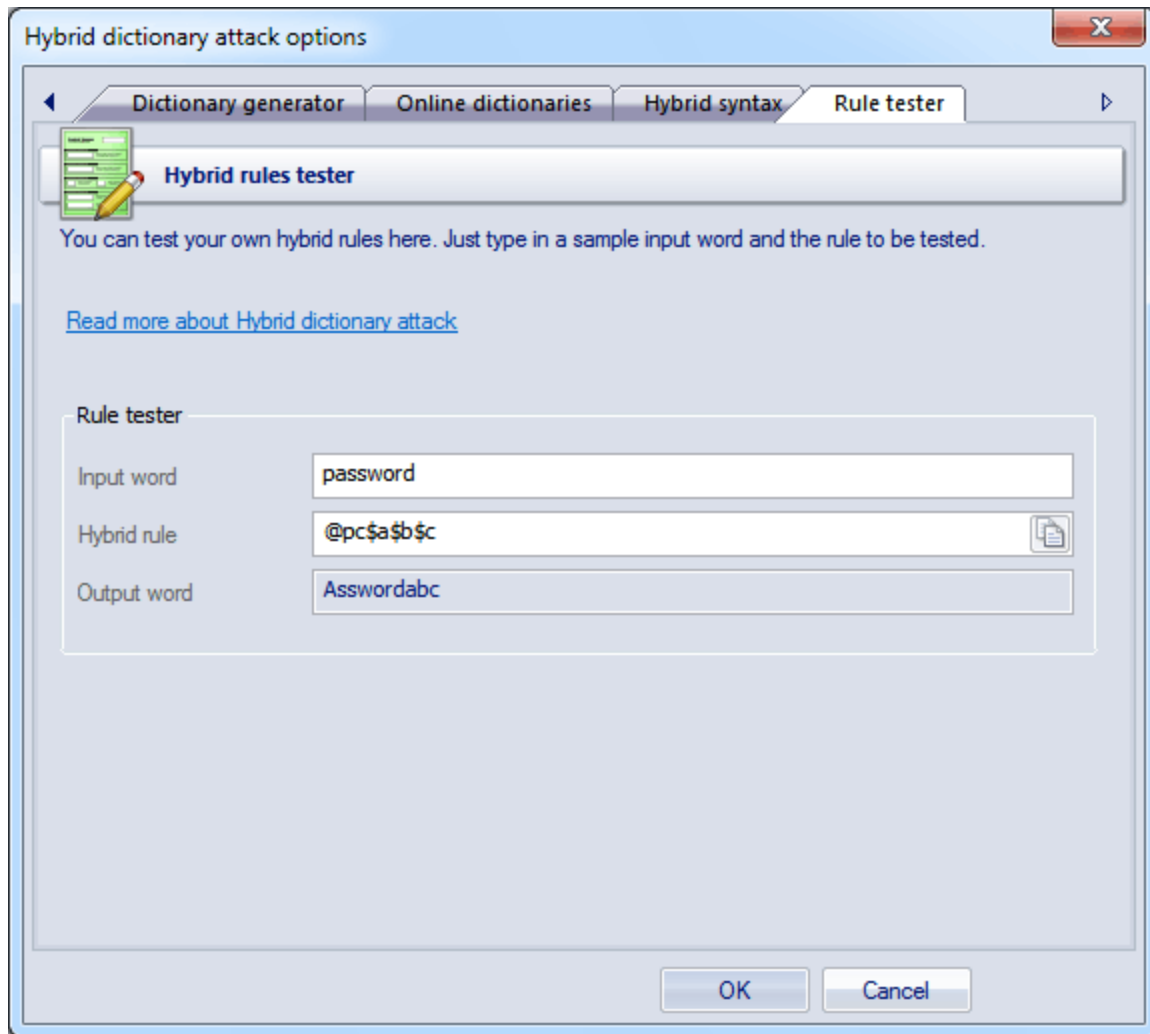


The '**Dictionary generator**' tab is designed for generating dictionaries obtained from an attack. Further on, those dictionaries could be used, for example, in other applications. To generate a dictionary, specify a source dictionary and a set of mutation rules for it. The size of a target file may exceed 2 GB. Be careful, the dictionary generation process may take considerable time!



You can download additional wordlists for the attack using '[Online dictionaries](#)' tab.

If you want to create your own set of rules, you can use the last two tabs as sources of help. While the '**Syntax**' tab gives mere descriptions of available rules, on the last tab you can actually test them by specifying a source word and a rule for the hybrid attack. Forward your rule sets to us; if we find them interesting/useful, we will include them in the default distribution of the program.



Rules description for the hybrid dictionary attack

Several rules at a line are allowed to be set.

Rules (if any) are processed from the left to the right.

Maximal line length is limited to **256** characters.

Maximal output word length is limited to **256** characters.

White space is ignored as long as it is not used as a parameter.

A line started with # character considered as a comment

All text before the **[Rules]** line is considered as comment.

N and M always start at 0. For values greater than 9 use A..Z (A=10, B=11, etc.)

The following rules should be at the last position of a line: aN, ?iN[C], ?i[C], ?oN[C], ?o[C]

?iN[C], ?i[C], ?oN[C], ?o[C] rules use the following predefined charsets:

digits - 0123456789

loweralpha - abcdefghijklmnopqrstuvwxyz

upperalpha - ABCDEFGHIJKLMNOPQRSTUVWXYZ

alpha - abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

special - !@#\$%^&*()-_+=~`[]{}|;\:;'"<>.,?/ "

loweralphanumeric - abcdefghijklmnopqrstuvwxyz0123456789

upperalphanumeric - ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

alphanumeric - abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

printable - abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#\$%^&*()-
_+=~`[]{}|;:"'<>.,? /

Rules

Rule	Example	Input	Output	Description
:	:	password	password	Do nothing to the input word
{	{	password	asswordp	Rotate the word left
}	}	password	dpassword	Rotate the word right
[[password	assword	Delete the first character
]]	password	passwor	Delete the last character
c	c	password	Password	Capitalize
C	C	password	pASSWORD	Anti-capitalize (lowercase the first character, uppercase the rest)
d	d	password	passwordp assword	Duplicate word
f	f	password	passworddr owssap	Reflect word
k	k	password	gfhjkm	Convert word using alternative (first after default) keyboard layout. The rule works in both directions. For example, if there's Russian keyboard layout installed previously in the system, the rule should convert word 'password' to Russian 'пассворд', and Russian word 'пассворд' to 'gfhjkm'. This is very helpful when looking for non-English passwords. If only one language is installed in the system, the rule does nothing.
K	K	password	passwodr	Swap last two characters
l	l	password	password	Convert all characters to lowercase
q	q	password	ppaassssw woorrrd	Duplicate all symbols
r	r	password	drowssap	Reverse word
t	t	PassWord	pASSWORD	Toggle case of all characters
u	u	password	PASSWORD	Convert all characters to uppercase
U	U	my own password	My Own Password	Capitalize all words delimited with space (upper-case the first character and every character after a space)
V	V	password	PaSSWoR D	Vowels elite
v	v	password	pASSWoR D	Vowels noelite
'N	'4	password	pass	Truncate the word to N character(s) length
+N	+1	password	pbssword	Increment character at position N by 1 ASCII value
-N	-0	password	oassword	Decrement character at position N by 1
.N	.4	password	passoord	Replace character at position N with character at position N+1
,N	,1	password	ppssword	Replace character at position N with character at position N-1. Where N > 0.
<N				Reject (skip) the word if it is greater than N characters long
>N				Reject (skip) the word if it is less than N characters long
aN				Check all possible symbol cases for the word. N is a maximal length of the word to apply this rule for.

Rul e	Exa mpl e	Input	Output	Description
DN	D2D 2	password	paword	Delete the character at position N
pN	p3	key	keykeykey	Copy word N times
TN	T1T 5	password	pAsswOrd	Toggle case of the character at position N
yN	y3	password	paspaswor d	Duplicate first N characters
YN	Y3	password	passwordord	Duplicate last N characters
zN	z3	password	ppppasswo rd	Duplicate the first character of the word N times
ZN	Z3	password	passwordd dd	Duplicate the last character of the word N times
\$X	\$0\$ 0\$7	password	password0 07	Add character X to the end of the word
^X	^3^2 ^1	password	123passwo rd	Insert character X at the beginning of the word
@X	@s	password	paword	Remove all characters X from the word
!X				Reject (skip) the word if it contains at least one character X
/X				Reject (skip) the word if it does not contain character X
(X				Reject (skip) the word if the first character is not X
)X				Reject (skip) the word if the last character is not X
eX	e@	mike@ya hoo.com	mike	Extract a substring starting at position 0 and ending up before first occurrence of X character (do nothing if X is not found)
EX	E@ e.	mike@ya hoo.com	yahoo	Extract a substring starting right after first found X character and till the end of the string (do nothing if X is not found)
% MX				Reject (skip) the word if it does not contain at least M instances of the character X
*XY	*15	password	possward	Swap characters at positions X and Y
=N X				Reject (skip) the word if the character at position N is not equal to the X
iNX	i4ai 5bi6 c	password	passabcwo rd	Insert the character X in position N
oN X	o4*o 5*	password	pass**rd	Overwrite a character in position N with the character X
sXY	ss\$ so0	password	pa\$\$w0rd	Replace all characters X with Y
xN M	x4Z	password	word	Extract a substring of up to M characters length, starting from position N.
INX -Y	rl0/- r	google.co m	google.com /	Insert the character X at position N if previous character at position N is not Y.
INX +Y	rl0. +.r	password	password.. .	Insert the character X at position N if previous character at position N is Y.
ON X-Y	O0- +p	password	-assword	If the character at position N is not Y, overwrite it with X character.

Rule	Example	Input	Output	Description
ON X+ Y	OOP +p	password	Password	If the character at position N is Y, overwrite it with X character.
? iN[C]	? iO[di gits]	password	0password .. 9password	Insert a character from a charset [C] into position N of the word. Where C should be either a predefined charset name or a custom character set itself.
? i[C]	? i[sp ecia l]	password	~password .. password+	Insert a character from a charset [C] into every position of the word. Where C should be either a predefined charset name or a custom character set itself.
? oN[C]	? o1[u pper alph a]	password	pAssword .. pZssword	Overwrite a character at position N with a character taken from a charset [C]. Where C should be either a predefined charset name or a custom character set itself.
? o[C]	?o[- =.]	password	-assword .. password.	Overwrite a character at every position of the word with a character taken from a charset [C]. Where C should be either a predefined charset name or a custom character set itself. The given example (?o[C] rule for the word password) will generate the following combinations: -password, =password, .password, -assword, =asswords, .assword, p-ssword, p=ssword ... password.

Additional

Wireless Password Recovery distribution kit comes with extended sets of password mutation rules:

hybrid_rules/english_words.ini file contains basic rules for English passwords.

hybrid_rules/simple_dates.ini - a lot of rules with dates, months, seasons, etc.

hybrid_rules/I33t.ini - rules to freak words (based on leet dictionary). For example, password-

>p@\$w0rd

hybrid_rules/dotcom.ini - different internet domains rules

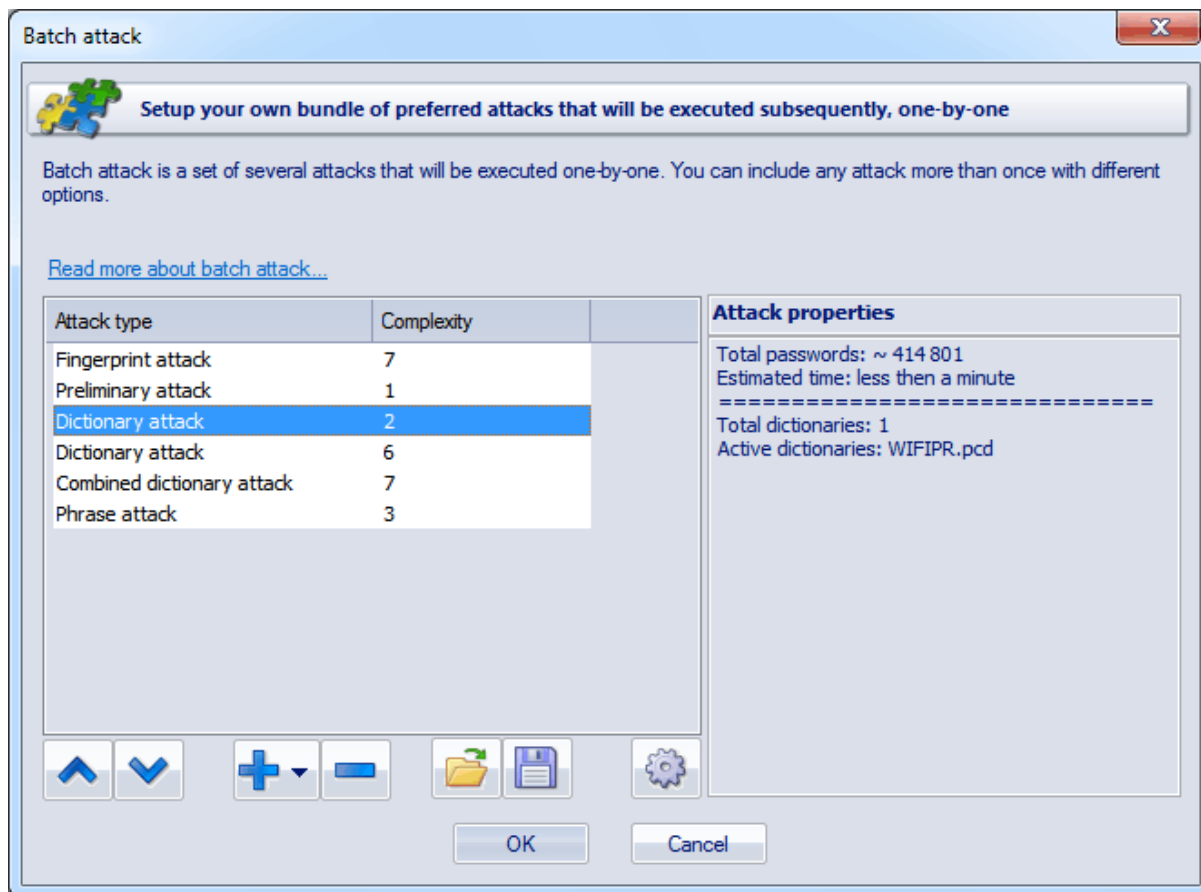
hybrid_rules/numbers.ini - append, prepend numbers, etc.

hybrid_rules/overwrite.ini - extended rules for English words

Looking for a convenient way to handle as much passwords as possible? Downloading the [full set of more than 180000 sorted and duplicate-free rules](#).

2.8.2.11 Batch attack

Since each attack covers its own password range, sometimes, in order to fully recover password hashes, you have to run several different attacks one after another. The basic idea behind the batch attack (developed by Passcape Software) is to create a list/batch of attacks to be run one after another, so that you could launch all those attacks with a single click of the mouse and not hassle with configuring each of them individually every time you need them.



The batch attack options are available as a list that you can extend or cut (buttons [+] and [-]). Each attack on the list can be moved up or down (buttons [^] and [v]), and its settings can be edited. A batch can include several attacks of the same kind, but the attacks can have different settings. The pane to the right of the selected entry displays the properties of the selected entry; brief specifications of the attack and the estimated time the attack will take to complete.

2.9 View menu

The View menu enables/disables the auxiliary elements of the interface, change the interface language, minimize the application to the tray or run it in the invisible mode.

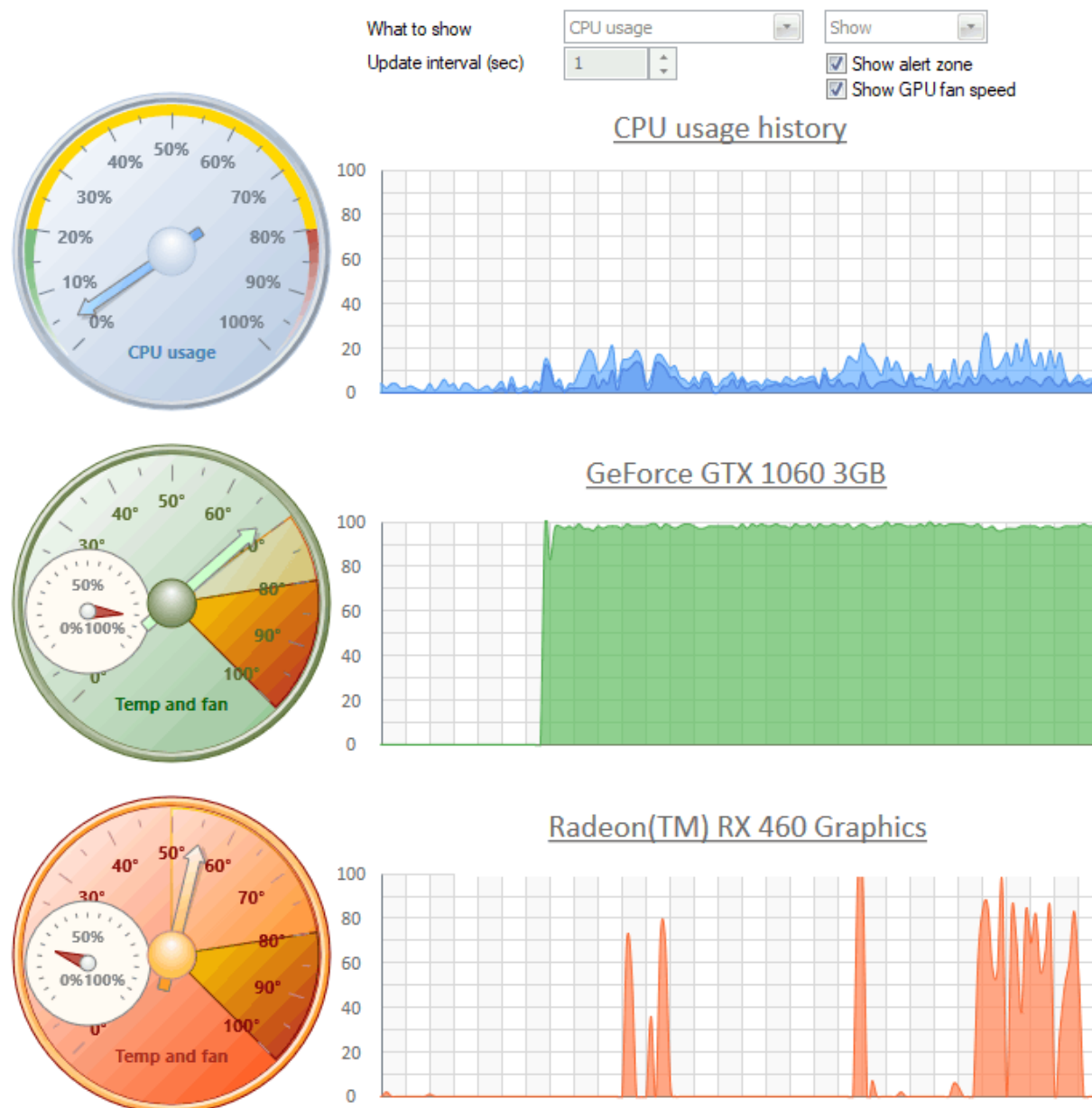
2.10 Themes menu

You can select here one of the themes you've liked or create your own theme.

2.11 Help menu

In this section of the menu, you can access the help articles on using the software, visit the program's home on the Web, check availability of updates, submit a bug report, register your copy of Wireless Password Recovery, etc.

2.12 Hardware Monitor



On this tab, you can view current CPU load and RAM utilization, each GPU load and temperature. By default, the refresh interval is set to 3 seconds. Be careful: gathering these statistics also takes CPU time.

Working with the program

3 Working with the program

3.1 Attacking WPA hashes

Currently the program can decrypt WPA-PSK hashes in several ways:

Preliminary attack (developed by Passcape Software) is based upon a social engineering method and consists of several sub attacks. Preliminary attack is very fast and often it is used for guessing simple and short passwords when there's no need to launch a fully scalable attack.

Artificial Intelligence attack - is a brand-new type of attack developed in our company. It is based upon a social engineering method and allows, without resort to time-consuming and costly computations, to almost instantly and painless recover certain passwords.

Dictionary attack. It is the most efficient recovery method, when the program tries each word from the dictionary (or dictionaries if there are several dictionaries) you specify until it finds the original password or until the wordlist is out of words. This method is very efficient since many people use regular words or phrases for password. Besides this type of recovery is performed quite fast compared to brute-force attack, for instance. Additional dictionaries and word-lists can be [downloaded from our site](#) or can be [ordered on CDs](#).

Brute-force attack tries all possible combinations from specified range of characters. This is the slowest attack, so it is really great for short passwords and used rarely in WPA hashes recovery.

Mask attack is a variation of the brute-force attack, except that some characters for finding the password remain unchanged, and only a portion of the password may change. The special syntax is used for setting a mask or rule for finding a password.

Base-word attack (developed by Passcape). At the first glance, this type of attack reminds the one we just described. It is just as efficient if a portion of the password to be recovered is known to us. However, unlike in the previous attack, here you do not have to set a mask - just provide a basic word. The program will take care of the rest. The base-word attack is based upon the experience of the social engineering to generate a great number of possible combinations of the given password.

Combined dictionary attack (developed by Passcape) uses to find compound passwords. For example, 'nothingtodo' or 'I give up'. It is very similar to the dictionary attack, except that instead of using a single word for password verification it uses a combination of words created by combining words from several dictionaries. You can create your own password generation rules.

Phrase attack (developed by Passcape) is very efficient against complex passwords. The idea of it is to guess the right password by searching through frequently used phrases and combinations. You can download pass-phrase wordlists and dictionaries from our site only.

Fingerprint Attack. Developed by Passcape. The attack parses input wordlist to generate so-called "fingerprints" used to recover the password. The attack is quite effective in finding difficult passwords.

Hybrid dictionary attack is like a simple dictionary attack, but allows user to customize word mutation and set his own password mutation rules. The rule definition syntax is compatible with some other password recovery software.

Batch attack (developed in Passcape Software) creates a list/batch of attacks to be run one-by-one, so that you could launch all those attacks with a single mouse-click instead of configuring each of them individually.

3.2 Attack comparison table

Which attack is the best? How do you choose the attack? The answers to these questions should be found in the attack comparison table.

Attack	Description	Time required	Guaranteed	Pros	Contras	Limitations
Preliminary	A set of light and speedy mini-attacks for finding simple, short or common combinations	Usually several minutes	No	Great quick-find tool for quick recovery of common, simple, short passwords, keyboard combinations, repetitive sequences, etc. Good for finding weak passwords quickly; doesn't require additional settings	Practically useless for serious analysis, when recovering the majority of complex passwords	Finds mainly simple passwords
Artificial Intelligence	The most advanced way of recovering passwords, based on the methods of social engineering.	Min: 5-6 minutes, Max: several days	No	The best tool for finding complex passwords, which other methods cannot cope with. Works great for passwords, words and combinations that the user stored in the system any time in the past.	During the most efficient analysis, when all the options are set to the maximum performance, the attack takes considerable time. Finds not all passwords.	Efficient only when run on the original system (where the passwords were taken)
Brute-force	Searches all possible combinations within a specified character set	Depends on options, several days min.	Yes	The only attack (along with the mask attack) that is guaranteed to recover a completely unknown password. Good for any short and medium passwords	Takes considerable time. Hard to guess the right range of characters to be searched.	May take centuries to search long passwords. Does not find passwords when uses wrong character set or password length exceeds the one specified.
Dictionary	Finds password by searching	A couple of minutes	No	Good and speedy tool for recovering common	Requires having good dictionaries,	Finds only common

	words from predefined dictionaries (word-lists)			passwords	does not take into account letter case	passwords
Dictionary with smart mutation	Same as dictionary attack, except here each word from the dictionary undergoes all kinds of mutations. For instance, appending numbers, changing letter case, deforming (displacing) letters, etc.	Up to 1000000 times slower than a simple dictionary attack	No	Good for all sorts of variations of common passwords	The maximum (most effective) mutation takes considerable time	Fails to find strong (non-dictionary) passwords, mutation takes considerable time
Mask	Finds passwords by specified mask (password generation rule)	Depends on options	Yes	Guaranteed to recover the remaining portion of a password. Good option when some portion of the original password is known.	Requires having the exact known portion of the password and its length and specifying the right character set to be searched	Password will not be found if a wrong character set, incorrect password length or incorrect known portion of the source password is specified
Combined dictionary	Checks complex passwords (composed of two or more words) by gluing words from several dictionaries	Depends on options	No	The only attack that finds long and complex passwords	Limited set of field-specific dictionaries. With a large source dictionary, the attack takes considerable time.	Requires to know in advance that the password being searched for consists of two or more words; very slow
Combined dictionary with smart mutation	Same as combined attack, plus mutations	Depends on options	No	Same as the previous attack	Same as the previous attack. Requires setting additional mutation rules for the passwords to be generated	Same as the previous attack; mutations require

						considerable time
Base-word	Takes advantage of a known base word used for making up the password	Usually couple of hours if the base-word length is not exceeds 16 characters	No	Good for the cases when you had known the original password but have forgotten its variations, e.g., letter case or trailing numbers	Mutation for long passwords (over 16 characters) may take some time	Does not always work
Phrase	Same as dictionary attack, except that instead of a word this one checks a phrase, popular expression, excerpts from songs, books, etc.	Min several minutes and up to several days	No	The only attack against password phrases.	Only a small percentage of users use pass-phrases as passwords. The mutation and analysis take considerable time. Insufficient number of relevant dictionaries.	Limited choice of mutations. Difficulty in the creation of specialized dictionaries.
Fingerprint	Based on fingerprints that were generated out of the given wordlist	Min several hours and up to several years (depends on the initial dictionary)	No	Finds complex passwords that were impossible to recover in other attacks	Big input dictionary may generate too much fingerprints. The success depends on the input dictionary.	The attack take too much time to complete when setting a big input wordlist.
Hybrid dictionary	It is much similar to simple dictionary attack, except that the password mutation rules are fully customizable and should be set by user.	Depend on the source wordlist and rules counter. Usually up to several hours for	No	Good for all sorts of variations of common passwords	Cannot recover complex passwords.	Fails to find strong (non-dictionary) passwords

a small
wordlist.

3.3 GPU FAQ

Q: What are the system requirements for the program?

A: Currently the program supports NVidia video cards with CUDA compute capability 2.0 or higher, AMD Radeon 7xxx+ GPUs and Intel HD Graphics 4xxx and higher. The full list of CUDA supported devices can be found at <http://developer.nvidia.com/cuda-gpus>. Compatible AMD Radeon cards are shown here: http://en.wikipedia.org/wiki/Comparison_of_AMD_graphics_processing_units.

Q: What versions of Windows the program supports?

A: GPU acceleration is supported starting up with Windows XP (NVidia GPUs) and Windows Vista (AMD GPUs) on both 32-bit and 64-bit systems.

Q: How do I know which architecture does my video card support?

A: For NVidia devices:

Launch the program, open the menu 'Options - General Options,' select the 'GPU Settings' tab, select 'NVidia CUDA' platform and choose your video card here. The 'Compute capability' field in the description section should display your GPU architecture.

For AMD devices and Intel Graphics:

Launch the program, open the menu 'Options - General Options,' select the 'GPU Settings' tab, select 'AMD OpenCL' platform and choose your video card here. The 'CL_DEVICE_VERSION' and 'CL_DEVICE_OPENCL_C_VERSION' fields should display your GPU architecture supported.

Q: Where can I get the latest video drivers?

A: You can download the latest drivers from NVidia (<http://www.nvidia.ru/drivers>) and AMD (<http://support.amd.com/us/gpudownload/Pages/index.aspx>) web sites.

Q: Where can I read more info about CUDA?

A: [Wikipedia site](#) is a good starting point to start from.

Q: Where can I read more info about AMD/ATI Radeon cards?

A: http://en.wikipedia.org/wiki/Comparison_of_AMD_graphics_processing_units

Q: Where can I read more info about Intel Graphics?

A: https://en.wikipedia.org/wiki/Intel_HD_and_Iris_Graphics

Q: After I launch a GPU-based attack, my computer freezes or crashes into BSOD. What's the problem?

A: The problem may be caused by the following reasons:

- Your video card had been overclocked, and it was malfunctioning at high load. If that's the case, bring the frequencies of the video memory/cores to its defaults.
- Insufficient or ineffective cooling of your card. When you launch a GPU-based attack, the program utilizes the most of the GPU power, and the GPU temperature rises to a critical level. Make sure that your video card is well cooled, the GPU slot and your system unit are free from dirt and dust. An unwise use of some video settings may have a negative impact on the video card's temperature and its stability under high load conditions. For example, some applications reduce the fan speed to minimize the noise, which does result in noise reduction, but also increases the core temperature.

- Power supply problem. Your card can consume a lot of energy at full load, and the power supply unit may be unable to handle such a high demand for power. If the video card has additional 6-pin or 8-pin power connectors, make sure they are all properly connected.
- There's yet another issue with GPU kernel execution timeout. The program should ask you to adjust the settings upon first call to GPU recovery. Otherwise you should change the WDDM registry settings yourself, for example, by disabling the kernel timeout.

Q: Does the PCI-Express bus have any impact on the performance?

A: Actually, this impact is negligible. It's usually masked by other factors. So the generation of your PCI-Express bus and its performance don't matter much.

Q: Does the amount of video memory matter?

A: No, it doesn't. However in most cases, your GPU should have at least 256 Mb of video memory.

Q: A GPU-based attack slows down my PC so I can barely use it. How can I fix it?

A: As a permanent fix, install a second video device, provided that you have a second slot on your motherboard and that your power supply unit can handle the additional load. For example, you can use some cheap card as the primary one (for displaying information on your monitor), and a second, more powerful one, for brute-forcing passwords.

Q: I have more than video cards in my computer. Can I use them all for brute-forcing?

A: Yes. You can use all or some of them. Just open general settings and specify the GPU device(s) to be used by the program.

Q: What's the maximal number of GPU devices does your program support?

A: It depends on your hardware. Even though the program supports up to 255 devices, typically, up to 8 devices can be installed into a 4 PCI-E slot motherboard (4 double-GPU cards).

Q: Can I brute-force passwords on devices which performance varies a lot?

A: Yes, you can.

Q: The program can not detect my video card. What can I do?

A: Update your video drivers. If it didn't help, try to extend your desktop to all devices (if you have more than one device). Re-plug your device into another PCI-Express slot.

Q: Your application will not work with all of my GPUs.

A: You will have to disable SLI in order to be able to use all devices.

Q: Can I use both NVidia and ATI devices simultaneously?

A: Yes, you can use NVidia, AMD and Intel HD graphics simultaneously.

Q: How can I check my GPU utilization?

A: Open 'Hardware Monitor' tab. In 'What to show' drop-box choose the device you need and select 'Show' to display it. You can then click 'Start' or 'Stop' buttons to manage the hardware monitoring. The GPU monitor shows device load (utilization), temperature and fan speed.

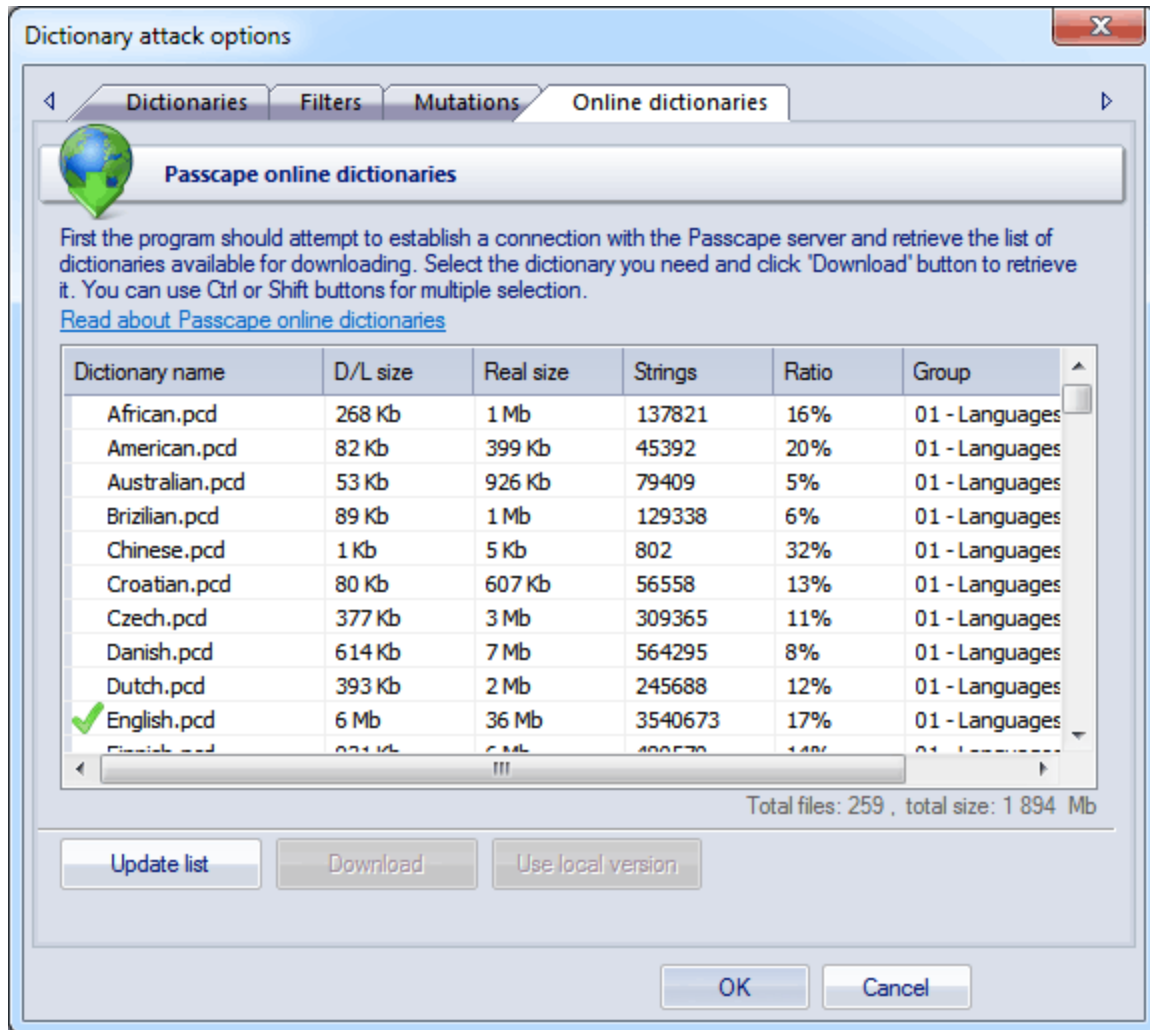
Q: My NVidia GPU is absent in hardware monitor.

A: You should install/reinstall NVAPI library. Download the library at <https://developer.nvidia.com/nvapi>

Q: Password recovery speed is much lower than expected.

A: You should have at least one free CPU core to handle one GPU core. Otherwise performance may drop drastically. Consider also trying to turn off CPU usage completely: menu *Options - General Options - CPU Settings*, set *Processor Utilization* field to 0.

3.4 Online dictionaries



The online dictionary selection dialog is extremely simple. When it opens up, the program attempts to establish a connection with the Passcape server and then retrieves and displays the list of dictionaries available for downloading.

Select the dictionary you need and then click on the 'Download' button to retrieve it and use in the program.

Some of the dictionaries are large. For instance, the size of 'music_songs.pcd' is more than 59 MB in the compressed format. Naturally, retrieving such a large amount of data may take some time, which depends upon file size, bandwidth of your Internet connection and net load.

All online (and some additional) dictionaries can be [ordered on CD](#). The total size of all the dictionaries is over 1GB. You can also share your own dictionary with us by e-mailing us the dictionary or the link where it can be downloaded.

The word-list are used in common dictionary attack, combined dictionary and pass-phrase attacks.

License and registration

4 License and registration

4.1 License agreement

=====

SOFTWARE LICENSE AGREEMENT

=====

IMPORTANT-READ CAREFULLY: This is the End User License Agreement (the "Agreement") is a legal agreement between you, the end-user, and Passcape Software, the manufacturer and the copyright owner, for the use of the "Wireless Password Recovery" software product ("SOFTWARE").

All copyrights to SOFTWARE are exclusively owned by Passcape Software.

The SOFTWARE and any documentation included in the distribution package are protected by national copyright laws and international treaties. Any unauthorized use of the SOFTWARE shall result in immediate and automatic termination of this license and may result in criminal and/or civil prosecution.

You are granted a non-exclusive license to use the SOFTWARE as set forth herein.

You can use trial version of SOFTWARE as long as you want, but to access all functions you must purchase the fully functional version. Upon payment we provide the registration code to you.

Once registered, the user is granted a non-exclusive license to use the SOFTWARE on one computer at a time (for every single-user license purchased).

With the personal license, you can use the SOFTWARE as set forth in this Agreement for non-commercial purposes in non-business, non-commercial environment. To use the SOFTWARE in a corporate, government or business environment, you should purchase a business license. With the business license you can run the SOFTWARE on multiple computers belonging to your organization - no matter where they are located.

The registered SOFTWARE may not be rented or leased, but may be permanently transferred together with the accompanying documentation, if the person receiving it agrees to terms of this license. If the software is an update, the transfer must include the update and all previous versions.

You may not create any copy of the SOFTWARE. You can make one (1) copy the SOFTWARE for backup and archival purposes, provided, however, that the original and each copy is kept in your possession or control, and that your use of the SOFTWARE does not exceed that which is allowed in this Agreement.

The SOFTWARE unregistered (trial) version may be freely distributed, provided that the distribution package is not modified. No person or company may charge a fee for the distribution of the SOFTWARE without written permission from the copyright holder.

You agree not modify, decompile, disassemble, otherwise reverse engineer the SOFTWARE, unless such activity is expressly permitted by applicable law.

Passcape Software does not warrant that the software is fit for any particular purpose. Passcape Software disclaims all other warranties with respect to the SOFTWARE, either express or implied. Some jurisdictions do not allow the exclusion of implied warranties or limitations on how long an implied warranty may last, do the above limitations or exclusions may not apply to you.

The program that is licensed to you is absolutely legal and you can use it provided that you are the legal owner of all files or data you are going to recover through the use of our SOFTWARE or have permission from the legitimate owner to perform these acts. Any illegal use of our SOFTWARE will be solely your responsibility. Accordingly, you affirm that you have the legal right to access all data, information and files that have been hidden.

You further attest that the recovered data, passwords and/or files will not be used for any illegal purpose. Be aware password recovery and the subsequencial data decryption of unauthorized or otherwise illegally obtained files may constitute theft or another wrongful action and may result in your civil and (or) criminal prosecution.

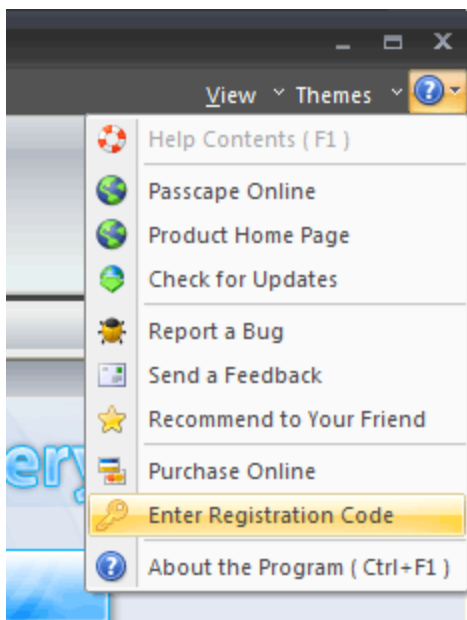
All rights not expressly granted here are reserved by Passcape Software.

4.2 Registration

The software is available in two editions: Standard and Professional. The detailed instructions for all kinds of orders are available online at [WIFIPR's order page](#). Online orders are fulfilled in just a few minutes 24 hours a day 7 days a week. If you purchase our products online, you will receive an automatically generated e-mail message with registration details within several minutes (if the order passes the fraud check system). However some orders can be marked for manual checkout or as 'suspicious'. This may increase order time up to several hours.

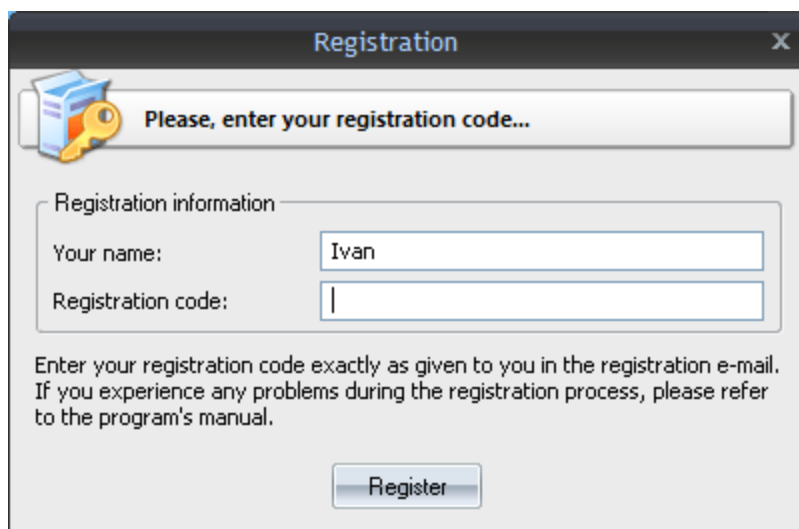
Important: when completing the order form, please double-check that your e-mail address is correct. If it will not, we'll be unable to send you the registration code

To complete the registration,



- Open the registration message and copy the registration code to the Windows clipboard.
- Run the program, select **Help - Enter Registration Code**

- Type in your registration name and paste the code here
- Click **Register** button to confirm

A screenshot of a software registration dialog box titled "Registration". It features a key icon and the text "Please, enter your registration code...". Below this, there is a section for "Registration information" with two input fields: "Your name:" containing the text "Ivan" and "Registration code:" which is empty. A paragraph of text below the fields reads: "Enter your registration code exactly as given to you in the registration e-mail. If you experience any problems during the registration process, please refer to the program's manual." At the bottom center is a "Register" button.

Registration

Please, enter your registration code...

Registration information

Your name: Ivan

Registration code:

Enter your registration code exactly as given to you in the registration e-mail. If you experience any problems during the registration process, please refer to the program's manual.

Register

4.3 Limitation of unregistered version

An unregistered version of **Wireless Password Recovery** shows only first 3 characters of recovered passwords and has some functional limitations. Registered version of the program eliminates all restrictions. Please refer to [this page](#) to view restrictions of certain edition.

4.4 Editions of the program

Wireless Password Recovery comes in two editions: Standard and Professional. The detailed list of features and compatibility chart is shown below.

FEATURE	Standard	Professional
Support for all versions of Windows starting with Windows XP	+	+
Windows 64-bit support	+	+
Multithreaded recovery	+	+
Interface themes support	+	+
Load hashes from local computer	+	+
Load hashes from external PC	-	+
Search for plaintext passwords	+	+
Import hashes from other programs	+	+
Export hashes to WIFIPRDUMP file	+	+

Export found passwords to text file	+	+
Support for wifi dump files with broken packets	+	+
Support for RAR/ZIP packet dumps	+	+
Common attacks	+	+
Advanced attacks	+	+
Smart attacks	+	+
Number of CPU cores supported	2	32
Number of GPU devices supported	1	255
Batch attack	-	+
View AI password cache	-	+
Online dictionaries	+	+
Generate dictionaries by mask	-	+
Generate dictionaries by given base-word	-	+
Combined dictionaries generator	-	+
Pass-phrase dictionary generator	-	+
Fingerprint dictionaries generator	-	+
Create wordlists based on hybrid attack	-	+
Create wordlists based on simple dictionary attack	-	+
Restrict access to the program	+	+
Password strength measurement	+	+
Hash checker	+	+
Random hash generator	+	+
MAC to vendor converter	+	+
Handshake generator	+	+
PMKID generator	+	+
Multiple hashes generator	-	+
Dictionary to hash generator	-	+
Asterisk password viewer tool	+	+
Wordlist tools: create a wordlist by indexing files	-	+
Wordlist tools: merge wordlists	+	+
Wordlist tools: wordlist statistics	+	+
Wordlist tools: sorting	+	+
Wordlist tools: conversion/compression	+	+
Wordlist tools: wordlist comparison	+	+
Wordlist tools: additional operations	+	+
Wordlist tools: indexing words/passwords of HDD sensitive areas	-	+
Wordlist tools: HTML links extractor	+	+
Hardware monitor	+	+
GPU health monitor	+	+
GPU performance adjustment	+	+
Password filters	+	+

Password reports	-	+
Run in hidden mode	+	+
Multimode	+	+
14-days money back guarantee	+	+
License	personal	business

* - uses some restrictions

Technical support

5 Technical support

5.1 Reporting problems

If you have a problem, please contact us at support@passcape.com. Please inform us about the following:

- Full name and version of the program
- Windows version including service pack, OEM and language information, etc.
- Registration information if any
- Detailed description of the problem, whether it is a constant or spontaneous error
- If you're reporting a critical error, please attach Crash.log file that was saved during an unhandled exception session.

5.2 Suggesting features

If you have any questions, comments or suggestions about the program or would like more information, email us at info@passcape.com. Please don't forget to mention the program name and version. Also make sure you have the latest program version installed. Your feedback helps us to improve our products and work more effective.

5.3 Contacts

Please don't hesitate to send your questions regarding our products to e-mail support@passcape.com. You will get reply during one or two days. Note, that registered users have priority in technical support.

If you experience any problems during registration process, please send a letter to sales@passcape.com

We will be happy to assist you with the registration.

Please write in English!

You can find other password recovery utilities at <https://www.passcape.com>

© 2020 Passcape Software. All rights reserved.

- | -

interface, program 8