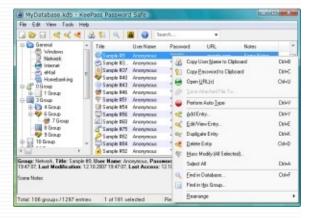


### **KeePass Password Safe**



KeePass: Copyright © 2003-2011 Dominik Reichl. The program is OSI Certified Open Source Software. OSI Certified is a certification mark of the Open Source Initiative. For more information see the License page.



### Introduction

Today you need to remember many passwords. You need a password for the Windows network logon, your e-mail account, your website's FTP password, online passwords (like website member account), etc. etc. The list is endless. Also, you should use different passwords for each account. Because if you use only one password everywhere and someone gets this password you have a problem... A serious problem. He would have access to your e-mail account, website, etc. Unimaginable.

But who can remember all those passwords? Nobody, but KeePass can. KeePass is a free, open source, light-weight and easy-to-use password manager for Windows.

The program stores your passwords in a highly encrypted database. This database consists of only one file, so it can be easily transferred from one computer to another.

KeePass supports password groups, you can sort your passwords (for example into Windows, Internet, My Website, etc.). You can drag&drop passwords into other windows. The powerful auto-type feature will type user names and passwords for you into other windows. The program can export the database to various formats. It can also import data from various other formats (more than 20 different formats of other password managers, a generic CSV importer, ...).

Of course, you can also print the password list or current view. Using the context menu of the password list you can quickly copy password or user name to the Windows clipboard. Searching in password database is possible.

The program ships with a strong random password generator (you can define the possible output characters, length, generation rules, etc.).

The program features a plugin architecture. Other people can write plugins for KeePass, extending its functionality: support for even more

data import/export formats, backup features, network features, etc. See the plugins page on the KeePass website for downloadable plugins and an introduction on how to write your own plugins.

And the best: it's free and you have full access to its source code!

KeePass is a Windows application. It has been developed using Microsoft Visual C++ with MFC classes. No .NET framework is required, nor any other special DLLs. It should run on all Windows operating systems.

KeePass is distributed under the GPL license. See the file *'License.txt'* in the downloadable KeePass package for details.

This documentation applies to KeePass 1.x.

-

### First Steps Tutorial A short tutorial showing you the basic usage of KeePass.

This short tutorial will show you how to actually use KeePass. It describes only the basic usage, advanced features are covered on separate pages.

### Creating a new database

The very first step is creating a new password database. KeePass will store all your passwords in such a database. To create one, click *'File - New...'* in the main menu or click the leftmost toolbar button. A window will appear, which prompts you for a master password and/or key file. The database will be encrypted with the password you enter here. The password you enter here will be the only password you'll ever have to remember from on now. It should be long and built up of mixed characters. Keep in mind that when someone gets your database file and guesses the password, he could access all passwords you stored in the database.

For this tutorial, we'll just use a password, not a key file. Click into the password edit field and enter a password of your choice. The password edit control isn't limited in length, so feel free to even enter a whole sentence (just keep in mind that you'll need to remember it).

After clicking [OK], a second dialog appears, in which you have to repeat the master password you just entered in the previous dialog. This is in order to avoid accidental typing errors.

Now you see the main window. On the left, you see the entry groups. On the right, you see the actual password entries. The password entries are grouped together into the password groups you see on the left. So, depending on which group on the left you selected, it'll show you the entries in this group in the right view. KeePass has created a few default groups for you, but you're totally free to delete them and create your own ones.

### Adding an entry

Time to store your very first password in the KeePass database! Right-click into the right password entry view and choose 'Add Entry...'. A window will pop up. In this window you can now edit your entry: enter some title for it, an username, an URL, the actual password, etc. If you don't need some of the fields, just leave them empty. When you're done, click [OK].

You'll see your new entry in the password list on the right now.

### **Using entries**

You got the new entry in the password list now. What can you actually do with it now? Right-click onto the entry.

You have several options now. You can for example copy the username of the entry to the Windows clipboard. When you've copied it, you can post it into any other program of your choice. The same works for copying passwords.

Alternatively, you can drag&drop fields into other windows. To see an example of how this works, see this page: Drag&Dropping Fields.

KeePass can open the URL you specified. To do this, just click 'URL(s) - Open URL(s)' in the context menu. KeePass will start the default browser and open the specified URL.

### Saving the database

It's time to save our database. Click onto the 'Save' toolbar button (looks like a disk; 3rd toolbar button). As you're saving the database the first time, you now have to specify a location where you want the database

file to be stored.

#### More

That's it! You've made the first steps in using KeePass! You can now have a look at the more advanced features of KeePass.

Passwords and Key Files: In the tutorial above we've encrypted the database using a password. But KeePass also supports key files, i.e. you can lock your database using a file (which you can carry around on your USB stick for example). It even supports combining those two methods for maximum security.

TAN Entries: TAN entries are one-time passwords. Many banks are using TANs for better security. KeePass supports TAN entries, by making them expire automatically when using them.

Auto-Type: The auto-typing functionality is a very powerful feature. In the tutorial above you've copied the username and password of an entry to the clipboard. Wouldn't it be nice if KeePass would just type those strings for you into other windows? Wouldn't it be nice if you could define whole sequences of keypresses that KeePass should type for you? That's exactly what the Auto-Typing feature does: it sends simulated keypresses for you to other windows!

URL Field Capabilities: The URL field supports URLs of course. In the tutorial, you've learned that you can enter simple URLs into this field and KeePass will open the browser window for you. But the URL field can do more! It actually supports many different protocols (not just http) and supports executing Windows command lines through the cmd:// virtual protocol. The field also features a powerful substitution engine, replacing codes by other fields (username, password, ...) of this entry.

Command Line Capabilities: You can open .kdb files by passing the filename to the KeePass executable file. But did you know that you can also send the password for the database and key file location over the

command line? You can also use the command line to pre-select a key file for you.

Plugins: KeePass features a powerful plugin architecture. If you miss some functionality, have a look at the plugins page to see if there are other people that have already written plugins for this. Many plugins exist to import/export data from/to other file formats.



# Acknowledgements / Credits Thanks to various people for contributions and/or work.

At this place I want to thank a lot of people very much for their help, source code, suggestions and other contributions (in no particular order).

- Donation Acknowledgements
- Source Code Acknowledgements
- Translation Acknowledgements
- Plugin Acknowledgements
- Tools Acknowledgements
- Hosting/Distribution Acknowledgements
- Suggestions and Forum Support Acknowledgements
- Special Acknowledgements
- Icons Acknowledgements

# **& Donation Acknowledgements**

Developing high-quality applications takes much time and resources. Donations make it possible to keep up the current development standard. Therefore, thanks a lot to all who donated to the project.

More information about donations and a list of people who donated can be found here: KeePass Donations.

## Source Code Acknowledgements

KeePass uses a lot of classes and libraries written by different people and given away for free. Here I want to thank them for writing these classes and libraries.

Author	Class / Library
Szymon Stefanek	C++ implementation of the AES/Rijndael encryption algorithm.
Brian Gladman	C implementation of the SHA-2 (256/384/512) hashing algorithms.
<b>Brent Corkum</b>	MFC class for XP-style menu (BCMenu).
<b>Davide Calabro</b>	MFC class for buttons with icons (CButtonST).
Peter Mares	MFC class for window side banners (CKCSideBannerWnd).
<b>Chris Maunder</b>	MFC class for system tray icons (CSystemTray).
Hans Dietrich	MFC class for hyperlinks in dialogs (XHyperLink).
PJ Naughter	MFC classes for checking for single instance and version information (CSingleInstance, CVersionInfo).
Lallous	Class for sending simulated keystrokes to other applications (CSendKeys).
Bouncy Castle Developers	Class for computing SHA-256 hashes (Bouncy Castle).
Mike Krueger, John Reilly	Library to handle GZip files (SharpZipLib).

# Translation Acknowledgements

Thanks a lot to all people who created translations for KeePass (see translations page).

# **8** Plugin Acknowledgements

Very many thanks to all people who wrote plugins for KeePass (see plugins page). Without you, KeePass would be a lot less powerful and useful!

### Tools Acknowledgements

Thanks to the **UPX Team** for the writing the Ultimate Packer for eXecutables (UPX). This tool is used to compress the final KeePass executables, in order to create a smaller distribution and making KeePass start up faster.

Thanks to **Jordan Russell** for creating Inno Setup. This tool is used to create the nice KeePass installation program.

### Hosting/Distribution Acknowledgements

Thanks to **SourceForge.net** for hosting the KeePass downloads / translations / plugins and for providing the project support platform (forums, feature requests / bug trackers, ...) for free.

Thanks to domain)FACTORY for hosting the KeePass website.

Thanks to **datensysteme-lenk** for hosting the German KeePass support forum in the past.

### Suggestions and Forum Support Acknowledgements

Thanks to all the people answering questions of others in the KeePass forums! A product is only as good as its support is, and I alone could never provide such an excellent individual help platform.

Two persons should be mentioned here, because of an extraordinary amount of suggestions (features, bug reports, ...) and helping others in the forums: **Paul Tannard** and **Michael Scheer**.

### Special Acknowledgements

Thanks to **Daniel Turini** for suggesting "KeePass" as the name of the project.

An especially *big* thanks to **Bill Rubin**. He not only contributed a lot of source code to KeePass, he also had an enormous amount of feature and improvement suggestions, helped people in the KeePass forums, and wrote a KeePass plugin for backing up databases. He's also the reason why many of the sections in the KeePass Help are very precise, helpful, clear and easy to understand. In our countless hours long IM chats, we not only discussed much about the design of KeePass, Bill also told me a lot about C++ and other stuff. Thanks!

### lcons Acknowledgements

Thanks a lot to **Christopher Bolin** for creating the main KeePass icon (see top left on this page) and its variations.

Thanks to **David Vignoni** for creating the nice 'Nuvola' icon theme. Most of the icons used in KeePass and on its website are icons of this theme. You can find the original images on the website of the author. Usage of the icons is allowed under the terms of the LGPL license (which you can find here: GNU Lesser General Public License), plus an addition. The original distribution notice can be found here: 'Nuvola' Icon Theme License.

-

## KeePass License License terms of KeePass 1.x.

KeePass: Copyright © 2003-2011 Dominik Reichl.

The program is distributed under the terms of the GNU General Public License version 2 or later. The complete source code can be obtained from the Official KeePass Website.

GNU General Public License (copy from http://www.gnu.org/licenses/gpl.html):

### **GNU GENERAL PUBLIC LICENSE**

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA Everyone is permitted to copy and distribute verbatim copies

of this license document, but changing it is not allowed.

### **Preamble**

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software-to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

# TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

**0.** This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to

say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

**1.** You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

- **2.** You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
  - **a)** You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
  - **b)** You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
  - c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a

warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

- **3.** You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
  - **a)** Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
  - **b)** Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of

Sections 1 and 2 above on a medium customarily used for software interchange; or,

**c)** Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

- **4.** You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
- **5.** You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your

acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

- **6.** Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
- **7.** If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a

consequence of the rest of this License.

- **8.** If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
- **9.** The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

**10.** If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### **NO WARRANTY**

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY

OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

### **END OF TERMS AND CONDITIONS**

### **How to Apply These Terms to Your New Programs**

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the program's name and an idea of what it does. Copyright (C) yyyy name of author

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

### End GNU General Public License

Icon Theme
The icon theme used in KeePass has been created and is copyrighted by <i>David Vignoni</i> . For more information, see the Acknowledgements page.



Installation / Portability KeePass 1.x installation, uninstallation, portability and updates.

- General information
- Installer program (KeePass-1.xx-Setup.exe file)
- Portable version (KeePass-1.xx.zip file)

### General information

When downloading KeePass, you have the choice between 3 different packages:

- KeePass-1.xx-Setup.exe: An installer program for Windows.
- **KeePass-1.xx.zip:** A KeePass ZIP package (portable version).
- **KeePass-1.xx-Src.zip:** The source code.

The installer and the portable version are described in detail below.

The source code package contains everything you need to compile KeePass. It includes the C++ source code and header files, resource files, sources for building the installer, etc.

In contrast to many other applications, KeePass does not *require* any functions of the latest Windows operating system, like theming on XP or like a .NET application that requires the .NET framework. For example, KeePass does support theming, but it's optional. KeePass runs perfectly on older systems without any feature limitations.

### **Updating KeePass:**

When a new KeePass version has been released, you can update your existing KeePass installation, without losing any configuration settings. The steps are depending on which package you are using (installer or portable), see below.

Translations should also be updated when you install a new KeePass version. You can find the latest translation files here: KeePass Translations.

### Installer program (KeePass-1.xx-Setup.exe file)

The KeePass development team provides an installer, which copies KeePass to your hard disk, creates shortcuts in the start menu and associates KDB files with KeePass, if desired.

Additionally, KeePass is automatically configured to store its settings in the application data directory of the current user. This way multiple users can use one KeePass installation without overwriting each other's settings (each user has his own configuration file). The setup program must run with administrative rights, however KeePass runs fine without administrative rights once it is installed.

#### Installation:

To install KeePass, run the KeePass-1.xx-Setup.exe file and follow the wizard.

### **Updating:**

Run the KeePass-1.xx-Setup.exe file. You do *not* need to uninstall the old version first. Your configuration options will not be lost.

#### **Uninstallation:**

In order to uninstall KeePass, run the uninstallation program, which is accessible by a shortcut in the start menu folder of KeePass, or in the program section of the system control panel. If you also want to remove your configuration settings, you need to delete the configuration file in the application data directory of your user profile, see Configuration.

#### **Silent Installation:**

The KeePass installer KeePass-1.xx-Setup.exe supports command line switches for silent installation, i.e. the program gets installed without asking the user for target directory or association options. The default settings of the installer are used.

The /SILENT command line switch performs a silent installation and shows a status dialog during the setup process. No questions will be asked though.

The /VERYSILENT command line switch performs a silent installation and does <i>not</i> show a status dialog during the setup process.		

### Portable version (KeePass-1.xx.zip file)

The portable version can be carried around on portable devices (like USB sticks) and runs on any computer directly from the device, without any installation. It doesn't store anything on your system (in contrast to the setup package, see above). KeePass doesn't create any new registry keys and it doesn't create any configuration files in your Windows or application data directory of your user profile.

Make sure that KeePass has write access to its application directory. Otherwise, if it doesn't have, it'll attempt to store the configuration options (nothing security-relevant though) into the application data directory of the currently logged on user. For more about that, see this page: Configuration.

### Installation:

KeePass does not need to be installed. Just download the ZIP package, unpack it with your favorite ZIP program and KeePass is ready to be used. Copy it to a location of your choice (for example onto your USB stick); no additional configuration or installation is needed.

### **Updating:**

Download the latest portable package of KeePass, unpack it and copy all new files over the old ones. Your configuration settings will not be lost (the settings are stored in the *KeePass.ini* file, which won't be overwritten, because KeePass ZIP packages don't include a *KeePass.ini* file).

### **Uninstallation:**

Simply delete the KeePass folder. This will leave no trace of KeePass on your system.



## Installing Translations How to install translations of KeePass 1.x. Installing User Interface TranslationsAdditional Localized Content

### Installing User Interface Translations

To install a user interface translation, follow these steps:

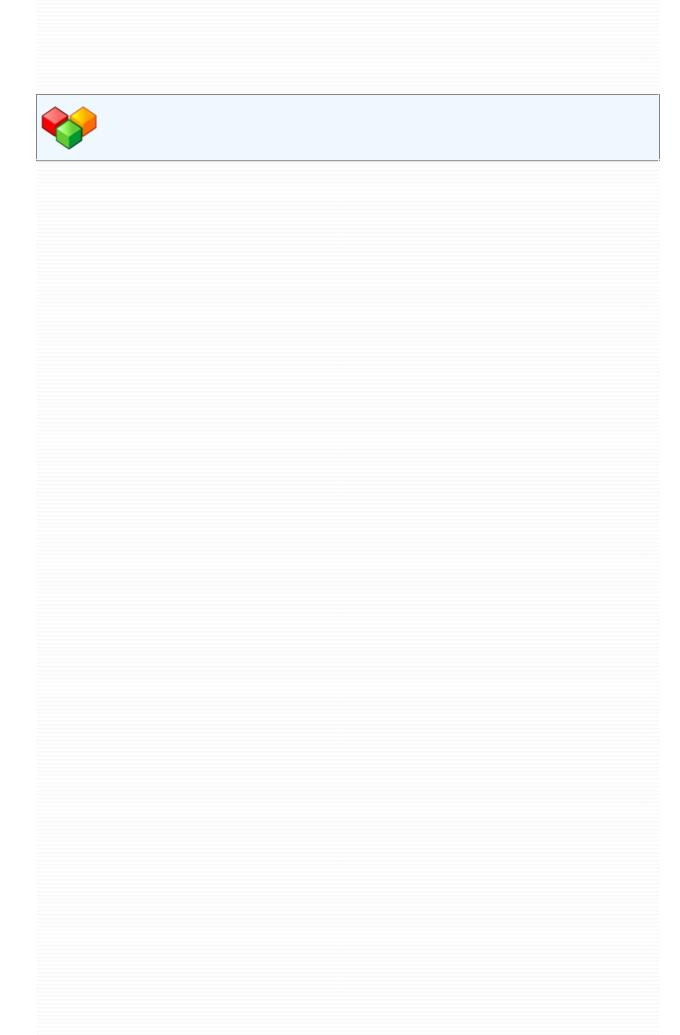
- 1. Download the translation file from the Translations page.
- 2. Unpack the ZIP archive you just downloaded to some directory.
- 3. Copy the <Language>.1ng file into the KeePass application directory (where KeePass.exe is located).
- 4. Start KeePass. In the main menu, click 'View' 'Change Language...'. In the dialog that opens, select your language. In the restarting prompt, click [Yes].

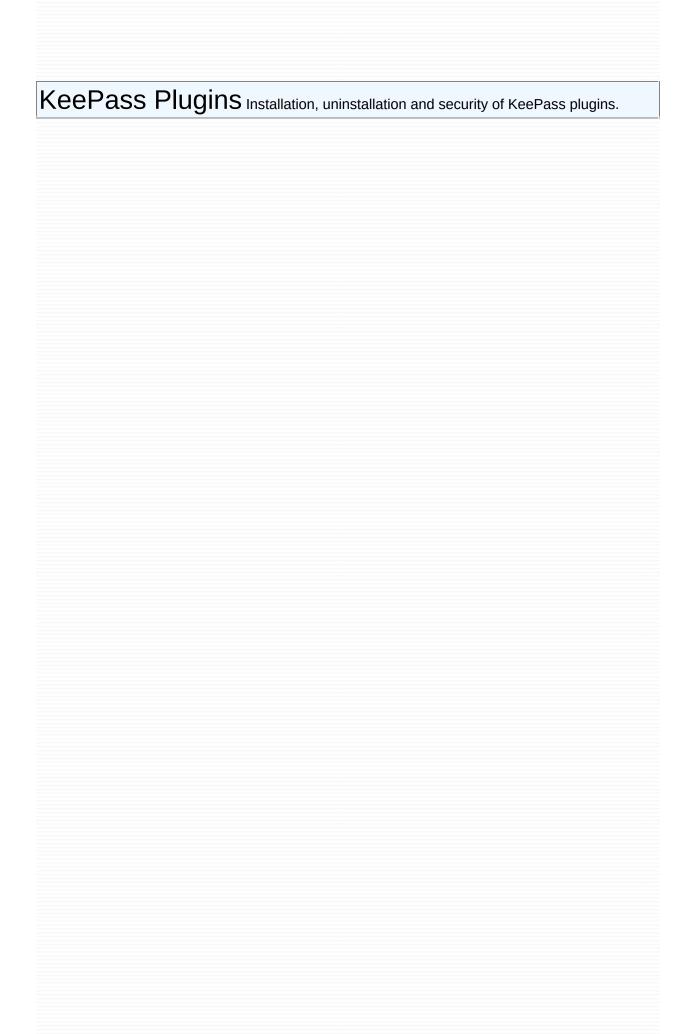
That's it! The user interface is now translated completely into your language!

### Additional Localized Content

For some languages (not for all) there is additional localized content available, like translated help files, tutorials, etc. All this content is available from the same page where the user interface translations are downloadable: Translations page.

If you'd like to create some translated content yourself, please first ask the KeePass team if the thing you're planning to create isn't in work already by someone else. If not, you'll make a lot of people very happy by creating translated content!





### Introduction

KeePass features a plugin framework. Plugins can provide additional functionality, like support of more file formats for import/export, network functionalities, backup features, etc.

## Online Resources You can download the latest KeePass plugins from http://keepass.info/plugins.html.

### Plugin Installation and Uninstallation

If the plugin doesn't provide explicit instructions how to install it, follow these steps:

- 1. Download the plugin from the page above and unpack the ZIP file to a directory of your choice.
- 2. Copy the unpacked plugin files into the KeePass directory (where the *KeePass.exe* is).
- 3. Restart KeePass in order to load the new plugin.

In other words, to "install" a plugin you simply need to copy it into the KeePass directory.

To "uninstall" a plugin, delete the plugin files.

### Security

What about the security of plugins? Can't malicious spyware plugins 'inject' themselves into KeePass?

If plugins can register themselves (i.e. have write access to the KeePass directory), they could also just replace the whole *KeePass.exe*. It's rather a problem of file access rights, not the plugin system.

If you worry about this, you can do the following:

- Install KeePass as administrator.
- 2. Write-protect the KeePass directory. Nobody must have write access.
- 3. Log on as normal user (with no administrator privileges).

This will solve the problem above. Since the KeePass directory is write-protected, no other program can copy files into it. KeePass requires the plugins to be in the application directory. Therefore, plugins cannot inject themselves anymore.



### Compatibility Compatibility between different KeePass versions, side-by-side usage.

This page lists the compatibility of **KeePass 1.20** with other KeePass versions.

Note that the latest version can load all database files created by any older KeePass version without any data loss.

If you were looking for instructions how to update to the latest version, please read the updating instructions.

### **Database File Compatibility (KDB Files):**

<b>Compatibility Type</b>	Versions
Full SxS	1.19b, 1.19, 1.18, 1.17, 1.16, 1.15, 1.14, 1.13, 1.12, 1.11, 1.10, 1.09, 1.08, 1.07, 1.06, 1.05, 1.04, 1.03, 1.02
DB SxS	1.01, 1.00, 0.99c, 0.99b, 0.99a, 0.98b, 0.98a
Lossy SxS	0.97c, 0.97b, 0.97a, 0.96b, 0.96a, 0.95b, 0.95a
No SxS	0.94a, 0.93b, 0.93a, 0.92a, 0.91, 0.90a, 0.89, 0.88a, 0.87, 0.86, 0.85, 0.84, 0.83b, 0.83, 0.82, 0.81, 0.80
Full Backward	1.19b, 1.19, 1.18, 1.17, 1.16, 1.15, 1.14, 1.13, 1.12, 1.11, 1.10, 1.09, 1.08, 1.07, 1.06, 1.05, 1.04, 1.03, 1.02, 1.01, 1.00, 0.99c, 0.99b, 0.99a, 0.98b, 0.98a, 0.97c, 0.97b, 0.97a, 0.96b, 0.96a, 0.95b, 0.95a, 0.94a, 0.93b, 0.93a, 0.92a, 0.91, 0.90a, 0.89, 0.88a, 0.87, 0.86, 0.85, 0.84, 0.83b, 0.83, 0.82, 0.81, 0.80

**Full SxS:** This compatibility type means that you can use the latest version with the listed ones side-by-side (SxS) without any data loss. These versions use exactly the same database format and are 100% compatible.

**DB SxS:** This type means that you can run these KeePass versions side-by-side, the database files are compatible. But there are some minor, *visual* incompatibilities. For example, from KeePass 1.01 to 1.02, the entry icons have changed. While this doesn't affect the database file format, creating a database with 1.01 will lead to different icons displayed in 1.02.

Lossy SxS: This type means that is possible to use the listed versions with the latest one side-by-side (database format is compatible), but some data of the newer version could be lost or displayed incorrectly in the old versions. For example, older versions don't know meta entries yet and display them as normal entries. Though no data will be lost, users may be confused when KeePass displays such meta entries. The listed versions can open databases of the latest one, but it is generally not recommended to use them side-by-side.

**No SxS:** This means that the database format of the latest version is incompatible with the listed KeePass versions. The listed ones are not able to load the database files created by the latest version. Note that the latest version can load *all* database versions (full backward compatibility).

**Full Backward:** The latest KeePass version can load all database files created by any older version. KeePass has been designed to be 100% backward compatible. No data is lost.



## $\begin{center} \textbf{Security} & \textbf{Detailed information about the security of KeePass.} \end{center}$

- Database Encryption
- Hashing and Key Derivation
- Random Number Generation
- Protection against Dictionary Attacks
- Process Memory Protection
- Enter Master Key on Secure Desktop (Protection against Keyloggers)
- Locking the Workspace
- Plugins
- Self-Tests
- Specialized Spyware
- References

### Database Encryption

KeePass database files are encrypted. KeePass encrypts the complete database, i.e. not only your passwords. The user names, notes, etc. are encrypted, too.

Databases are encrypted using one of the following block ciphers:

Cipher	<b>Block Size</b>	<b>Key Size</b>
Advanced Encryption Standard (AES / Rijndael)	128 bits	256 bits
Twofish	128 bits	256 bits

These algorithms are well-known, analyzed thoroughly and considered to be very secure (see [1] for comments by the NIST on AES for example). AES e.g. became effective as a U.S. Federal government standard and is approved by the National Security Agency (NSA) for top secret information.

The block ciphers are used in the CBC (cipher-block chaining) block cipher mode. In CBC mode, plaintext patterns are concealed.

For both algorithms, a 128-bit initialization vector (IV) is generated randomly each time you save the database. This allows multiple databases to be encrypted using the same key without observable patterns being revealed.

### Hashing and Key Derivation

In order to generate the 256-bit key for the block ciphers, the Secure Hash Algorithm SHA-256 is used. This algorithm compresses the *user key* provided by the user (consisting of password and/or key file) to a fixed-size key of 256 bits. This transformation is one-way, i.e. it is computationally infeasible to invert the hash function or find a second message that compresses to the same hash.

The recently discovered attack against SHA-1 [2] doesn't affect the security of SHA-256. SHA-256 is still considered as being very secure [3].

### **Key Derivation:**

If only a password is used (i.e. no key file), the password plus a 128-bit random *salt* are hashed using SHA-256 to form the final key (but note there is some preprocessing: Protection against Dictionary Attacks). The random salt prevents attacks that are based on pre-computed hashes.

When using both password *and* key file, the final key is derived as follows: SHA-256(SHA-256(password), key file contents), i.e. the hash of the master password is concatenated with the key file bytes and the resulting byte string is hashed with SHA-256 again. If the key file doesn't contain exactly 32 bytes (256 bits), they are hashed with SHA-256, too, to form a 256-bit key. The formula above then changes to: SHA-256(SHA-256(password), SHA-256(key file contents)).

### **Random Number Generation**

KeePass needs to generate several random bytes (for the IV, the master key salt, etc.). For this, several pseudo-random sources are used: current tick count, performance counter, system date/time, mouse cursor position, memory status (free virtual memory, etc.), active window, clipboard owner, various process and thread IDs, various window handles (active window, desktop, ...), window message stack, process heap status, process startup information and several system information structures. Additionally, KeePass uses random bytes provided by the system's default CSP RNG.

This pseudo-random data is collected in a random pool. To generate 16 random bytes, the pool is hashed (SHA-256) with a counter. The counter is increased after 16 generated bytes. This way, as many secure random bytes can be produced efficiently as needed.

### Protection against Dictionary Attacks

KeePass supports a protection against guessing and dictionary attacks.

You can't really prevent these attacks: nothing prevents an attacker to just try all possible keys and look if the database decrypts. But what we can do (and KeePass does) is to make it harder: by adding a constant work factor to the key initialization, we can make them as hard as we want.

To generate the final 256-bit key that is used for the block cipher, KeePass first hashes the user's password using SHA-256, encrypts the result *N* times using the Advanced Encryption Standard (AES) algorithm (called *key transformation rounds* from on now), and then hashes it again using SHA-256. For AES, a random 256-bit key is used, which is stored in the database file. As the AES transformations aren't pre-computable (key is random), an attacker has to perform all the encryptions, too, otherwise he cannot try and see if the current key is correct.

An attacker now needs much more time to try a key. If he can only try a few keys per second, a dictionary attack is not practical anymore. *N* is a work factor, only indirectly a time factor. A super computer can try a key a lot faster than a standard PC, but anyway testing one key with *N* transformation rounds will take *N* times longer than trying a key with no transformation rounds on the super computer.

By default, KeePass sets *N* to 6000 encryption rounds (full encryptions are meant; *N* has nothing to do with the internal encryption rounds of AES). This number has been chosen in order to provide compatibility with portable device versions (PocketPC processors are slower, therefore the key computation takes longer).

If you are using KeePass on PC only, it is highly recommended to increase the number of key transformation rounds. You can change the number in the database options dialog. Right of the field for the rounds, you'll find a button. When clicking this button, KeePass computes the rounds number that leads to a 1-second delay. Waiting 1 second at database opening isn't a problem, but for an attacker of course it is. But,

the number can be freely set to a number of your choice; the button only should give you a rough idea how many rounds can be computed in 1 second on your computer.

This protection feature is only useful for master passwords; key files are random anyway, there's no need to transform the key file contents. Guessing the key file contents is equally hard to a brute-force attack on the final key.

KeePass uses multithreading to compute the transformations (the master key is split up to two parts of 128 bits, which is the AES block size). On dual/multi core processors, the computation can be twice as fast as on a single core processor.

On Windows Vista and higher, KeePass can use Windows' CNG/BCrypt API for the key transformations, which is about 50% faster than the KeePass built-in key transformation code.

**KeePassX:** In contrast to KeePass, the Linux port project 'KeePassX' only partially supports protection against dictionary and guessing attacks.

### Process Memory Protection

While KeePass is running, sensitive data (like the hash of the master key and entry passwords) is stored encrypted in process memory.

This means that even if you would dump the KeePass process memory to disk, you couldn't find the passwords.

For example, when you are copying a password to the clipboard, KeePass first decrypts the password field, copies it to the clipboard and immediately re-encrypts it using the random key.

Additionally, KeePass erases all security-critical memory when it's not needed anymore, i.e. it overwrites these memory areas before releasing them (this applies to all security-critical memory, not only the password fields).

KeePass ≥ 1.15 and 2.x use the Windows DPAPI for in-memory encrypting the sensitive data. With DPAPI, the key for in-memory encryption is stored in a secure, non-swappable memory area managed by Windows. If DPAPI is not available or disabled (advanced KeePass options, by default using DPAPI is enabled), KeePass uses the ARC4 encryption algorithm with a random key. Note that this is less secure than DPAPI, mainly *not* because ARC4 cryptographically isn't that strong, but because the key for in-memory encryption is also stored in swappable process memory.

### Enter Master Key on Secure Desktop (Protection against **Keyloggers**)

Note: KeePass was one of the first (maybe even the first) password manager that allows entering the master key on a secure desktop!

KeePass ≥ 2.15 has an option (in 'Tools' -> 'Options' -> tab 'Security') to show the master key dialog on a secure desktop (supported on Windows ≥ 2000), similar to Windows' User Account Control (UAC). Almost no keylogger works on a secure desktop.

The option is disabled by default for compatibility reasons.

### Locking the Workspace

When locking the workspace, KeePass closes the database file and only remembers the last view settings (which group and entries were selected, list position, etc.).

This provides maximum security: unlocking the workspace is as hard as opening the database file the normal way. Also, it prevents data-loss (the computer can crash while KeePass is locked, without doing any damage to the database).

# Plugin Security A separate page exist about the security of plugins: Plugin Security.

### Self-Tests

Each time you start KeePass, the program performs a quick self-test to see whether the block ciphers and the hash algorithms work correctly and pass their test-vectors. If one of the algorithms doesn't pass its test vectors, KeePass shows a security exception message box.

### Specialized Spyware

This section gives answers to questions like the following:

- Would encrypting the configuration file increase security by preventing changes by a malicious program?
- Would encrypting the application (executable file, eventually together with the configuration file) increase security by preventing changes by a malicious program?
- Would an option to prevent plugins from being loaded increase security?
- Would storing security options in the database (to override the settings of the KeePass instance) increase security?
- Would locking the main window in such a way that only auto-type is allowed increase security?

The answer to all these questions is: no. Adding any of these features would not increase security.

All security features in KeePass protect against *generic* threats like keyloggers, clipboard monitors, password control monitors, etc. (and against non-runtime attacks on the database, memory dump analyzers, ...). However in all the questions above we're assuming that there's a spyware program running on the system that's specialized on attacking KeePass.

In this situation, the best security features will fail. This is law #1 of the 10 Immutable Laws of Security [4] [5]: "If a bad guy can persuade you to run his program on your computer, it's not your computer anymore".

For example, consider the following very simple spyware specialized for KeePass: an application that waits for KeePass to be started, then hides the started application and imitates KeePass itself. All interactions (like entering a password for decrypting the configuration, etc.) can be simulated. The only way to discover this spyware is to use a program that the spyware doesn't know about or can't manipulate (secure desktop); in any case it can't be KeePass.

### References and Further Reading

- [1] National Institute of Standards and Technology: Report on the Development of the Advanced Encryption Standard (AES) (PDF).
- [2] Bruce Schneier's blog: SHA-1 broken.
- [3] Bruce Schneier's blog: *Cryptanalysis of SHA-1*, with comments about the impact of that discovery and what to do now.
- [4] Scott Culp, Microsoft TechNet Essay, 2000: 10 Immutable Laws of Security.
- [5] Jesper M. Johansson, Microsoft TechNet Magazine, 2008: *Revisiting the 10 Immutable Laws of Security, Part 1*.

	10011
	00111
	11010
_	

## Error Codes KeePass error codes and their meanings.

<b>Error Code</b>	Name	User?	Description
0x0000000	Unknown	*	An unknown error occured. KeePass uses this error code internally and the user should never see this error.
0x0000001	Success	<b>V</b>	No error occured. Everything worked perfectly.
0x0000002	Invalid Parameter	*	A function within KeePass has received an invalid parameter. This should never happen.
0x0000003	Out of Memory	<b>V</b>	The operating system didn't grant KeePass enough memory. The current operation was cancelled. The database maybe is in a broken state and should not be saved.
0x0000004	Invalid Key	<b>V</b>	The user-provided composite key (password / key file) for the database was incorrect, or the database is corrupted. Make sure the password and/or key file is correct.  If you are absolutely sure that the password / key

			file is correct, then probably the file is corrupted (maybe you accidently removed the USB stick from the computer without unmounting it). In this case, try the Repair Database functionality of KeePass.
0x0000005	File Access Error: Read File		The operating system (Windows) didn't grant KeePass read access to the specified database file. Note that this doesn't mean that the database is corrupted or something like that; KeePass just cannot get read access to the file.  Make sure that you can read the file.
0x0000006	File Access Error: Write File		The operating system (Windows) didn't grant KeePass write access to the specified database file.  Try saving the database to a different location, where the administrators of your network have granted you write access to.
		V	

0x0000007	File Error: Read		An error occured while reading the database or key file. This can happen if you for example open a big database from USB stick and remove the stick while KeePass is reading the file.  Make sure KeePass can access the complete file and don't remove the device before KeePass has read it completely.
0x0000008	File Error: Write		An error occured while writing the database or key file. This can happen if you for example save a big database to USB stick and remove the stick while KeePass is writing the file.  Make sure there is enough free space left on the target device and don't remove the device before KeePass has written the file completely. Make sure you correctly unmount the device (USB stick).
0x0000009	Invalid Random Source	×	This is an internal cryptographic error. The random source, that is used to generate random

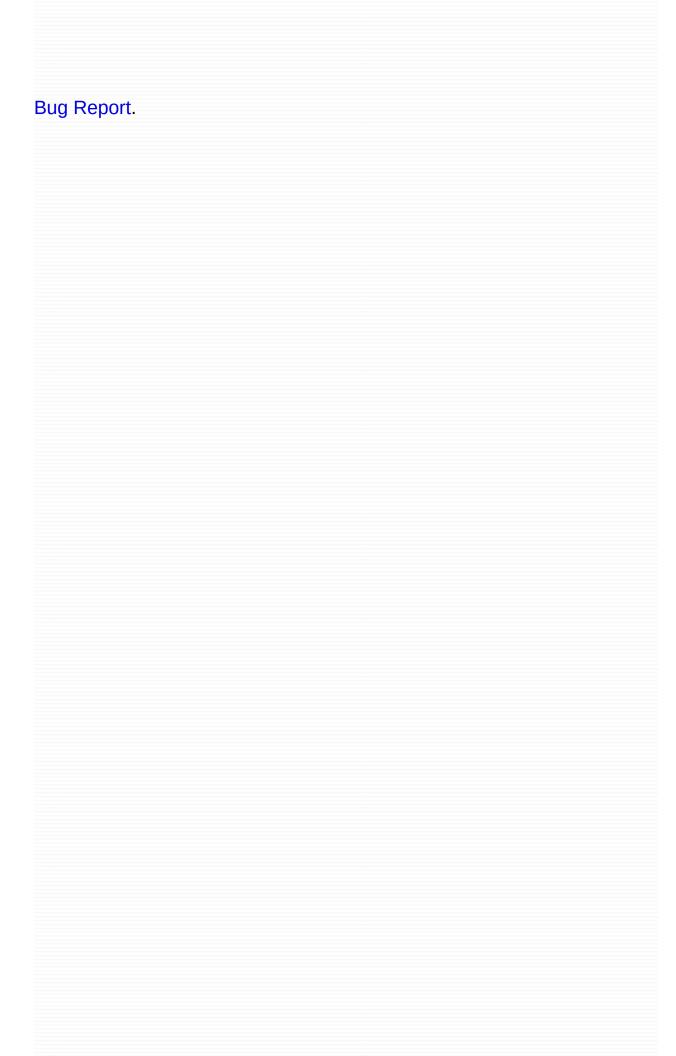
			numbers, is invalid. This means that it produces non-random values and therefore the security (secrecy) of your database cannot be guaranteed; therefore the operation was aborted.  This error should never occur. If you see this error code, please submit a bug report (see below).
0x000000A	Invalid File Structure		The structure of the current database file is invalid. This can either happen if the user-supplied composite key (password, key file) is invalid or if the file is corrupted.  If you are sure that the key is correct, try the Repair Database functionality.
0x000000B	Cryptographic Error	*	An internal cryptographic error occured. One of the cryptographic components cannot be initialized, tested and/or returned an error.
0x000000C	Invalid File Size	<b>V</b>	The file size isn't valid. This means that the file is corrupted.

0x0000000F File Access Erro Read Key	uei *		The file's header is invalid. The file could be corrupted, could have been created by a newer KeePass version, or could not be a KeePass file at all.  The operating system
	or:	<b>V</b>	
			(Windows) didn't grant KeePass read access to the specified key file or it doesn't exist. Note that this doesn't mean that the key file is corrupted, KeePass just cannot get read access to the file.  Make sure the media that contains the key file is inserted and that KeePass has the right to access it (check file access rights, ensure that no other application is blocking the file,).  In case you're selecting the key file's drive (i.e. not specifying it manually), have a look for the "pwsafe.key" file. See

			the composite key documentation for more information about using key files.
0x0000010	Prov: Invalid Key	<b>V</b>	The key provider plugin did not supply a valid key.
0x0000011	File Error: Verify		The data that has been sent to the disk/drive does not match the data that is now stored on the disk/drive, i.e. the file was corrupted by something. See error 0x00000008.
0x0000012	KDBX File	~	The file you tried to open is in KDBX format. KDBX files are only supported by KeePass 2.x. Either use KeePass 2.x, or use the 'Export' feature of KeePass 2.x to migrate the file to a KeePass 1.x KDB file.
0x00000013	System Error	~	A user-level error occured and the system provided the detailed description shown in the error message box.

### User?

- This error is not a bug. KeePass has failed to perform some operation and you can fix this "error" yourself. For example, when you entered an invalid key for your database, failing to open the database isn't a bug.
- X: This error is a bug. You should never see this error. Please submit a





# Auto-Type Powerful feature that sends simulated keypresses to other applications.

- Basic Auto-Type Information
- Invoking Auto-Type
  - Context Menu: 'Perform Auto-Type' Command
  - Global Auto-Type Hot Key
- Specifying Keystroke Sequences and Target Windows
  - Auto-Type Keystroke Sequences
  - Target Window Filters
  - Change Default Auto-Type Sequence
- Usage Example

#### Basic Auto-Type Information

KeePass features an "Auto-Type" functionality. This feature allows you to define a sequence of keypresses, which KeePass can automatically perform for you. The simulated keypresses can be sent to any other currently open window of your choice (browser windows, login dialogs, ...).

By default, the sent keystroke sequence is {USERNAME}{TAB}{PASSWORD} {ENTER}, i.e. it first types the user name of the selected entry, then presses the Tab key, then types the password of the entry and finally presses the Enter key.

For TAN entries, the default sequence is {PASSWORD}, i.e. it just types the TAN into the target window, without pressing Enter.

You can freely define your own Auto-Type sequence: simply write the sequence into the **notes field** of the entry, prefixed with "Auto-Type:". Your notes could look something like this:

```
You can write any notes here.
My e-mail I used to register: test@nowhere.com

Auto-Type: {USERNAME}{TAB}{TAB}Some fixed string{TAB}{PASSWORD}{ENTER}

Here you can continue with your notes if you wish...
```

As you can see, the only important thing is that the auto-type sequence is prefixed using "Auto-Type:" and is one single line. An auto-type sequence cannot be defined using two or more lines.

If you define two or more auto-type sequences, the first one is used.

Additionally, you can create customized window/sequence associations, which override the default sequence. You can specify different keystroke sequences for different windows for each entry. For example, imagine a webpage, to which you want to login, that has multiple pages where one can login. These pages could all look a bit different (on one you could additionally need to check some checkbox -- like often seen in forums). Here creating customized window/sequence associations solves the problems: you simply specify different auto-type sequences for each

windows (identified by their window titles).

## **Invoking Auto-Type:**

There are three different methods to invoke auto-type:

- Invoke auto-type for an entry by using the context menu command Perform Auto-Type while the entry is selected.
- Select the entry and press *Ctrl+V* (that's the menu shortcut for the context menu command above).
- Using the system-wide auto-type hot key. KeePass will search all entries in the currently opened database for matching sequences.

All methods are explained in detail below.

#### - Context Menu: 'Perform Auto-Type' Command

This method is the one that requires the least amount of configuration and is the simpler one, but it has the disadvantage that you need to select the entry in KeePass which you want to auto-type.

The method is simple: right-click on an entry of your currently opened database and click 'Perform Auto-Type' (or alternatively press the Ctrl+V shortcut for this command). The window that previously got the focus (i.e. the one in which you worked before switching to KeePass) will be brought to the foreground and KeePass auto-types into this window.

The sequence which is auto-typed depends on the window's title. If you didn't specify any custom window/sequence associations, the default sequence is sent. If you created associations, KeePass uses the sequence of the first matching association. If none of the associations match, the default sequence is used.

#### Global Auto-Type Hot Key

This is the more powerful method, but it also requires a little bit more work/knowledge, before it can be used.

#### **Simple Global Auto-Type Example:**

- 1. Create an entry in KeePass titled *Notepad* with values for user name and password.
- 2. Start Notepad under 'Programs' -> 'Accessories'.
- 3. Press *LCtrl+Alt+A* within Notepad. The user name and password will be typed into Notepad.

The KeePass entry title *Notepad* is matched with the window title of Notepad and the default Auto-Type sequence is typed.

#### **How It Works - Details:**

KeePass registers a system-wide hot key for auto-type. The advantage of this hot key is that you don't need to switch to the KeePass window and select the entry. You simply press the hot key while having the target window open (i.e. the window which will receive the simulated keypresses).

By default, the global hot key is *LCtrl+Alt+A* (i.e. hold the left Ctrl and Alt keys, press 'A' and release all keys). You can change this hot key in the options dialog (main menu - 'Tools' - 'Options', tab 'Advanced'): here, click into the textbox below "Global Auto-Type Hot Key Combination" and press the hot key that you wish to use. If the hot key is usable, it will appear in the textbox.

When you press the hot key, KeePass will look at the title of the currently opened window and search the currently opened database for usable entries. If KeePass finds multiple entries that can be used, it displays a selection dialog. In order to decide if an entry is usable for the current window title, at least one of the following conditions must be fulfilled:

• The title of the entry is a substring of the currently active window title.

 The entry has a window/sequence association, of which the window specifier matches the currently active window title.

The second condition has been mentioned already, but the first one is new. By using entry titles as filters for window titles, the configuration amount for auto-type is almost zero: you only need to make sure that the entry title is contained in the window title of the window into which you want the entry to be auto-typed. Of course, this is not always possible (for example, if a webpage has a very generic title like "Welcome"), here you need to use custom window/sequence associations.

Custom window/sequence associations can be specified using the Notes field of entries.

```
My e-mail I used to register: test@nowhere.com

Auto-Type: {USERNAME}{TAB}{TAB}Some fixed string{TAB}{PASSWORD}{ENTER}

Auto-Type-Window: Some Website - Welcome*

Here you can continue with your notes if you wish...
```

If you now have a window open that starts with "Some Website - Welcome" and press the global auto-type hot key combination, KeePass performs the auto-type sequence above.

Some sites, notably banks, use multi-page login schemes. You can use Auto-Type-Window strings to automate these sites. You can also use Auto-Type-Window strings to standardize your LAN login in one KeePass entry.

As many Auto-Type-Window strings per entry can be defined as you wish.

Also, one sequence can be used for multiple windows. For this, first define a window/sequence pair as normal, then continue by appending '-' and a number, starting with 1. Example:

```
Auto-Type: {USERNAME}{TAB}{PASSWORD}{ENTER}
Auto-Type-Window: Some Dialog - *
Auto-Type-1: {USERNAME}{ENTER}
Auto-Type-Window-1: * - Editor
Auto-Type-Window-1: * - Notepad
Auto-Type-Window-1: * - WordPad
Auto-Type-2: {PASSWORD}{ENTER}
Auto-Type-Window-2: Some Web Page - *
```

Here, the Auto-Type-1 sequence will be used for all Auto-Type-Window-1

windows.
Custom window associations override the KeePass entry title. If any custom window associations are specified, they will be the only item(s) used to determine a match and the KeePass entry title will be ignored.

#### Auto-Type Keystroke Sequences

An auto-type keystroke sequence is a one-line string that can contain placeholders and special key codes.

A complete list of all supported placeholders can be found on the page Placeholders. The special key codes can be found below.

Above you've seen already that the default auto-type is {USERNAME} {TAB} {PASSWORD} {ENTER}. Here, {USERNAME} and {PASSWORD} are placeholders: when auto-type is performed, these are replaced by the appropriate field values of the entry. {TAB} and {ENTER} are special key codes: these are replaced by the appropriate keypresses. Special key codes are the only way to specify special keys like Arrow-Down, Shift, Escape, etc.

Of course, keystroke sequences can also contain simple characters to be sent. For example, the following string is perfectly valid as keystroke sequence string:

{USERNAME}{TAB}Some text to be sent!{ENTER}.

Special key codes are case-sensitive.

#### **Special Keys:**

The following codes for special keys are supported:

Special Key	Code
Tab	{TAB}
Enter	{ENTER} or ~
Arrow Up	{UP}
Arrow Down	{DOWN}
Arrow Left	{LEFT}
Arrow Right	{RIGHT}
Insert	{INSERT} or {INS}
Delete	{DELETE} or {DEL}

Home	{HOME}
End	{END}
Page Up	{PGUP}
Page Down	{PGDN}
Backspace	{BACKSPACE}, {BS} or {BKSP}
Break	{BREAK}
Caps-Lock	{CAPSLOCK}
Escape	{ESC}
Help	{HELP}
Numlock	{NUMLOCK}
Print Screen	{PRTSC}
Scroll Lock	{SCROLLLOCK}
F1 - F16	{F1} - {F16}
Keypad +	{ADD}
Keypad -	{SUBTRACT}
Keypad *	{MULTIPLY}
Keypad /	{DIVIDE}
Shift	+
Ctrl	۸
Alt	%

Special Key	Code
Space	{SPACE}
+	{PLUS}
@	{AT}

%	{PERCENT}
Apps	{APPS}
۸	{CARET}
~	{TILDE}
{, }	{LEFTBRACE}, {RIGHTBRACE}
(, )	{LEFTPAREN}, {RIGHTPAREN}
Windows Key: left, right	{LWIN}, {RWIN}
Windows Key	{WIN} (equ. to {LWIN})
Windows Key	@
Numpad 0 to 9	{NUMPAD0} to {NUMPAD9}

Additionally, some special commands are supported:

<b>Command Syntax</b>	Action
{DELAY X}	Delays X milliseconds.
{CLEARFIELD}	Clears the contents of the edit control that currently has the focus (only single-line edit controls).
{VKEY <i>X</i> }	Sends the virtual key of value $X$ .

<b>Command Syntax</b>	Action
{BEEP X Y}	Beeps with a frequency of <i>X</i> and a duration of <i>Y</i> milliseconds.
	Sets the default delay to

·	X milliseconds for all following keypresses.
{APPACTIVATE WindowTitle}	Activates the window "WindowTitle".

#### Finally, some examples:

{TITLE}{TAB}{USERNAME}{TAB}{PASSWORD}{ENTER}

Types the entry's title, a Tab, the user name, a Tab, the password of the currently selected entry, and presses Enter.

{TAB}{PASSWORD}{ENTER}

Presses the Tab key, enters the entry's password and presses Enter.

{USERNAME}{TAB}^v{ENTER}

Types the user name, presses Tab, presses Ctrl-v (which pastes data from the Windows clipboard in most applications), and presses Enter.

#### **Toggling Checkboxes:**

Sometimes you find checkboxes on websites ("Stay logged in on this computer" for example). You can toggle these checkboxes by sending a space character (' ') when auto-typing. For example: {USERNAME}{TAB}{PASSWORD}{TAB} {TAB}{ENTER}

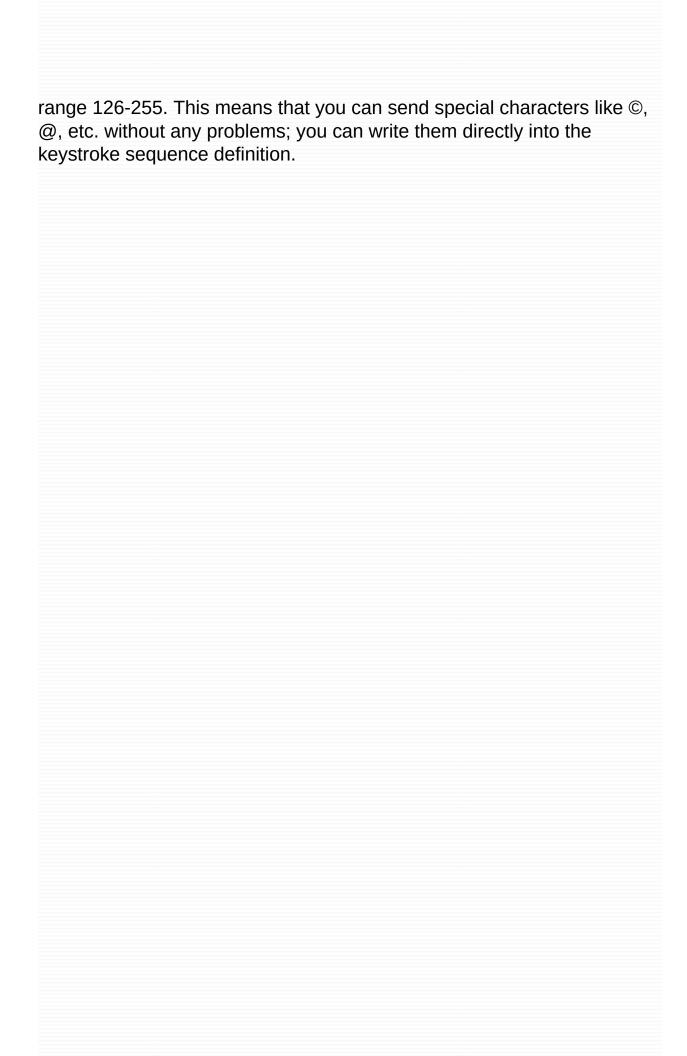
If there is a webform with a user name field, password field and a checkbox, this sequence would enter the user name, the password and toggle the checkbox that follows the password control.

### **Pressing Non-Default Buttons:**

Pressing non-default buttons works the same as toggling checkboxes: send a space character (' '). Note that this should only be used for non-default buttons; for default buttons, {ENTER} should be sent instead.

#### **Higher ANSI Characters:**

The auto-type function supports sending of higher ANSI characters in



#### Target Window Filters

When creating a custom window/sequence association, you need to tell KeePass how the matching window titles look like. Here, KeePass supports simple wildcards:

<b>String with Wildcard</b>	Meaning
STRING	Matches all window titles that are named exactly "STRING".
STRING*	Matches all window titles that start with "STRING".
*STRING	Matches all window titles that end with "STRING".
*STRING*	Matches all window titles that have "STRING" somewhere in the window title. This includes the string being directly at the start or at the end of the window title.

Other wildcards aren't supported. The \* wildcard must not be in the middle of a string.

For example, \*windows\*Explorer\* won't match windows Explorer, it will only match windows\*Explorer, i.e. the middle \* character is treated as '\*' text character instead of a wildcard.

By using wildcards, you can make your auto-type associations browser-independent. See the usage examples for more information.

#### Change Default Auto-Type Sequence

The default auto-type sequence (i.e. the one which is used when you don't specify a custom one) is {username}{tab}{password}{enter}. KeePass allows you to change this default sequence. Normally you won't need to change it (use custom window/sequence definitions instead!), but it is quite useful when some other application is interfering KeePass (for example a security software that always asks you for permission before allowing KeePass to auto-type).

The default auto-type sequence can be changed in the Auto-Type configuration dialog. This dialog can be found in 'Tools' -> 'Options' -> 'Advanced' -> 'Auto-Type'.

#### Usage Example

Now let's have a look at a real-world example: logging into a website. In this example, will we use the global auto-type hot key to fill out the login webpage. First open the test page, and afterwards create a new entry in KeePass with title *Test Form* and a user name and password of your choice.

Let's assume the global auto-type hot key is set to *LCtrl+Alt+A* (the default). KeePass is running in the background, you have opened your database and the workspace is unlocked.

When you now navigate to the test page and are being prompted for your user name and password, just click into the user name field and press *LCtrl+Alt+A*. KeePass enters the user name and password for you!

Why did this work? The window title of your browser window was "Test Form - KeePass - Internet Explorer" or "Test Form - KeePass - Mozilla Firefox", depending on the browser you are using. Because we gave the entry in KeePass the title Test Form, the entry title is contained in the window title, therefore KeePass uses this entry.

Here you see the huge advantages of auto-type: it not only doesn't require any additional browser software (the browser knows nothing of KeePass -- there are no helper browser plugins required), it is also browser-independent: the one entry that you created within KeePass works for Internet Explorer *and* Mozilla Firefox (and other browsers) without requiring any modifications or definitions.

When you would use window/sequence associations (instead of entry title matching), you can achieve the same browser-independent effect using wildcards: you could for example have used Test Form - KeePass - \* as window filter. This filter matches both the Internet Explorer and the Firefox window.



## Command Line Options Command line options to automate KeePass tasks.

You can pass a file path in the command line in order to tell KeePass to open this file immediately after startup.

Switches can be either prefixed using a minus (-) or two minus characters (--). On Windows, a slash (/) is another alternative. The prefixes are equivalent; it doesn't matter which one you use.

**Database file.** The database file location is passed as argument. Only one database file is allowed. If the path contains a space, it must be enclosed in quotes (").

**Password.** Passwords are passed using the -pw: option. In order to pass 'abc' as password, you would add the following argument to the command line: -pw:abc. Note that there must be no space between the ':' and the password. If your password contains a space, you must enclose it in quotes. For example: -pw:"my secret password".

**Key file.** For supplying the key file location, the <code>-keyfile:</code> switch exists. The same rules as above apply, just that you specify the key file location: <code>-keyfile:D:\pwsafe.key</code>. You also need to quote the value, if it contains a space, tab or other whitespace characters.

**Pre-selection.** In order to just pre-select a key file, use the -preselect: option. For example, if you lock your database with a password *and* a key file, but just want to type in the password (so, without selecting the key file manually), your command line would look like this:

KeePass.exe "C:\My Documents\MyDatabase.kdb" -preselect:C:\pwsafe.key

KeePass would then show a prompt for the password for the database, but in the key file list, the C:\pwsafe.key file is selected already. When using the preselect switch, KeePass by default activates the key file switch and sets the focus to the password edit window.

Note the difference! The preselect switch just pre-selects the key file for you and displays the login prompt. In contrast, the keyfile switch doesn't

prompt you for the (maybe missing) password.

The -minimize command line argument makes KeePass start up minimized.

The -auto-type command line argument makes all previous KeePass instances perform a global auto-type.

If the -readonly switch is present in the command line, KeePass will open the database in read-only mode.

The -lock switch forces KeePass to open in locked mode (i.e. you will not be asked immediately for the master key when also passing a database path).

The order of the arguments is arbitrary.

#### Usage Examples

Open the database file 'C:\My Documents\MyDatabase.kdb' (KeePass will prompt you for the password and/or key file location):

KeePass.exe "C:\My Documents\MyDatabase.kdb"

If you got a database that is locked with a password 'abc', you could open it like this:

KeePass.exe "C:\My Documents\MyDatabaseWithPw.kdb" -pw:abc

If your USB stick always mounts to drive F: and you've locked your database with a key file on the USB stick, you could open your database as follows:

KeePass.exe "C:\My Documents\MyDatabaseWithFile.kdb" -keyfile:F:\pwsafe.key

If you've locked your database using a password *and* a key file, you can combine the two switches and open your database as follows:

KeePass.exe "C:\My Documents\MyDatabaseWithTwo.kdb" -pw:abc -keyfile:F:\pwsafe.key

You have locked your database using a password *and* a key file, but only want to have the key file pre-selected (i.e. you want to get prompted for the password), your command line would look like this:

KeePass.exe "C:\My Documents\MyDatabaseWithTwo.kdb" -preselect:F:\pwsafe.key

#### Starting KeePass using a Batch File

Batch files can be used to start KeePass. Mostly you want to specify some of the parameters listed above. You can theoretically simply put the command line (i.e. application path and parameters) into the batch file, but this is not recommended as the command window will stay open until KeePass is closed. The following method is recommended instead:

```
START "" KeePass.exe ..\MyDb.kdb -pw:MySecretPw
```

This START command will run KeePass (which opens the ..\myDb.kdb file using MySecretPw as password). KeePass is assumed to be in the same directory (working directory) as the batch file, otherwise you need to specify a different path.

START executes the given command line and immediately exits, i.e. it doesn't wait until the application is terminated. Consequently, the command window will disappear after KeePass has been started.

Please note the two quotes (") after the START command. These quotes are required if the application path contains quotes (in the example above, the quotes could also be removed). If you want to learn more about the START command syntax, type START /? into the command window.

#### Closing/Locking KeePass using a Batch File

To close all currently running KeePass instances, call KeePass.exe with the '--exit-all' parameter:

KeePass.exe --exit-all

All KeePass windows will attempt to close. If a database has been modified, KeePass will ask you whether you want to save or not. If you wish to save in any case (i.e. a forced exit without any confirmation dialog), enable the 'Automatically save database on exit and workspace locking' option in 'Tools' - 'Options...' - 'Advanced'.

The KeePass instance that has been created by the command above is not visible (i.e. it does not show a main window) and will immediately terminate after sending close requests to the other instances.

KeePass  $\geq$  1.17 and  $\geq$  2.10 support the --lock-all and --unlock-all command line options to lock/unlock the workspaces of all other KeePass instances.



# Composite Master Key This document details how KeePass locks its databases.

- Master Passwords
- Key Files
- Windows User Account
- For Administrators: Specifying Minimum Properties of Master Keys

KeePass stores your passwords securely in an encrypted file (database). This database is locked with a master password, a key file and/or the current Windows account details. To open a database, **all** key sources (password, key file, ...) are **required**. Together, these key sources form the *Composite Master Key*.

KeePass does not support keys being used alternatively, i.e. it's not possible that you can open your database using a password **or** a key file. Either use a password, a key file, or both at once (both required), but not interchangeably.

#### Master Passwords

If you use a master password, you only have to remember one password or passphrase (which should be good!) to open your database. KeePass features protection against brute-force and dictionary attacks on the master password, read the security information page for more about this.

If you forget this master password, all your other passwords in the database are lost, too. There isn't any backdoor or a key which can open all databases. There is no way of recovering your passwords.

#### Key Files

You don't even have to remember a long, complicated master passphrase. The database can also be locked using a key file. A key file is basically a master password in a file. Key files are typically stronger than master passwords, because the key can be a lot more complicated; however it's also harder to keep them secret.

- A key file can be used *instead* of a password, or in *addition* to a password (and the Windows user account in KeePass 2.x).
- A key file can be any file you choose; although you should choose one with lots of random data.
- A key file must not be modified, this will stop you opening the database. If you want to use a different key file, you can change the master key and use a new/different key file.
- Key files must be backed up or you won't be able to open the database after a hard disk crash/re-build. It's just the same as forgetting the master password. There is no backdoor.

Do not backup the key file to the same location as the database, use a different directory or disk. Test opening your database on another machine to confirm your backup works. For a detailed discussion on the difference between backing up the key file and the database, see the ABP FAQ.

The point of a key file is that you *have* something to authenticate with (in contrast to master passwords, where you *know* something), for example a file on a USB stick. The key file content (i.e. the key data contained within the key file) needs to be kept secret. The point is *not* to keep the location of the key file secret — selecting a file out of thousands existing on your hard disk basically doesn't increase security at all (it's very easy for malware/attackers to find out the correct file, for example by observing the last access times of files). Trying to keep the key file location secret is security by obscurity, i.e. not really effective.

A 'key disk' is just a normal disk which holds a file (called 'pwsafe.key') with key bytes (KeePass can generate such disks for you). If you want, you can also select the key file (which is stored on the key disk)

manually, i.e. one disk can then store multiple keys for multiple databases. In this case, you have to tell KeePass which file it should use; you cannot simply select a drive anymore (when you just select a drive, KeePass assumes that it should load the 'pwsafe.key' file in the root directory of the disk).

KeePass can generate key files for you, however you can also use any other, already existing file (like JPG image, DOC document, etc.).

In order to use an existing file as key file, click the button with the 'Save' image in the master key creation dialog and select the existing file. After accepting the dialog, KeePass will ask you whether to overwrite or reuse the file (see screenshot).

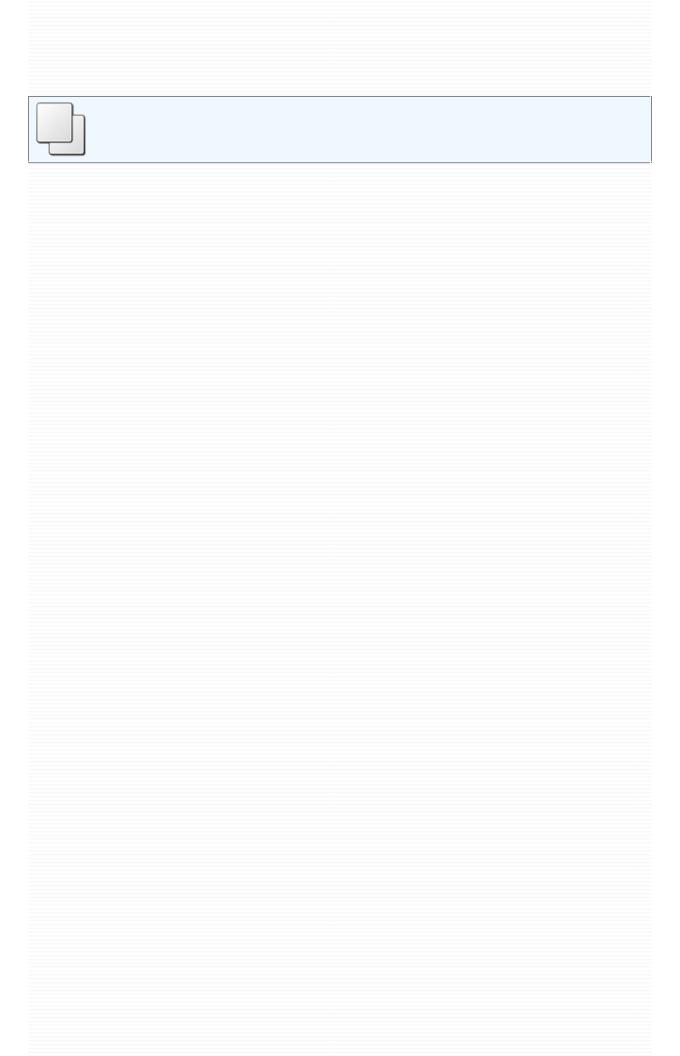
<b>&amp; Windows User Account</b>		
KeePass 1.x does not support encrypting databases using Windows use account credentials. Only 2.x and higher support this.		

# **Example 2** For Administrators: Specifying Minimum Properties of Master Keys

Administrators can specify a minimum length and/or the minimum estimated quality that master passwords must have in order to be accepted. You can tell KeePass to check these two minimum requirements by adding/editing appropriate definitions in the INI/XML configuration file.

The value of the KeeMasterPasswordMinLength key can contain the minimum master password length in characters. For example, by specifying KeeMasterPasswordMinLength=10, KeePass will only accept master passwords that have at least 10 characters.

The value of the KeeMasterPasswordMinQuality key can contain the minimum estimated quality in bits that master passwords must have. For example, by specifying KeeMasterPasswordMinQuality=64, only master passwords with an estimated quality of at least 64 bits will be accepted.



# Configuration Details about how and where KeePass stores its configuration.

KeePass supports multiple locations for storing configuration information: the *global* configuration file in the KeePass application directory, a *local* user-dependent one in the user's private configuration folder, and an *enforced* configuration file in the KeePass application directory. The first one is called *global*, because everyone using this KeePass installation will write to the same configuration file (and possibly overwriting settings of other users). The second one is called *local*, because changes made to this configuration file only affect the current user.

Configuration files are stored in INI format.

Configuration	Location	Typical File Path
Global	Application Directory	C:\Program Files\KeePass Password Safe\KeePass.ini
Global (Virtualized)	Windows Vista/7 Virtual Store	C:\Users\User Name\AppData\Local\VirtualStore\Program Files\KeePass Password Safe\KeePass.ini
Local	User Application Data	C:\Documents and Settings\ <i>User</i> Name\Application  Data\KeePass\KeePass.ini
Enforced	Application Directory	C:\Program Files\KeePass Password Safe\KeePass.enforced.ini

### Installation by Administrator, Usage by User

If you use the KeePass installer and install the program with administrator rights, the program directory will be write-protected when working as a normal/limited user. KeePass will use local configuration files, i.e. save and load the configuration from a file in your user directory.

Multiple users can use the locally installed KeePass. Configuration settings will not be shared and can be configured individually by each user.

#### Portable Version

If you downloaded the portable version of KeePass (ZIP package), KeePass will try to store its configuration in the application directory. No configuration settings will be stored in the user directory (if the global configuration file is writable).

#### Create Portable Version of Installed KeePass

If you are currently using a locally installed version of KeePass (installed by the KeePass installer) and want to create a portable version of it, first copy all files of KeePass to the portable device. Then get the configuration file from your user directory (application data, see above) and copy it over the configuration file on the portable device.

#### For Network Administrators: Enforced Configuration

KeePass can be forced to load specific configuration settings. Enforced configuration settings are loaded from KeePass.enforced.ini (KeePass 1.x) and KeePass.config.enforced.xml (KeePass 2.x) files in the application directory (where KeePass.exe is stored).

Configuration items that are not present in the enforced configuration file are loaded normally from global/local configuration files.

This method of course only is effective as long as your users run the KeePass installation on the network drive. If they copy KeePass to their hard drives and run it from there, the options you set are not enforced (the local KeePass installation doesn't know anything of the enforced configuration file on the network drive in this case).

#### Technical Details

This section explains in detail how loading and saving the configuration works.

When KeePass starts up and finds both global and local configuration files, it must decide the order in which KeePass tries to get the configuration items. This is controlled by the (Kee)PreferUserConfiguration flag in the global configuration file. If it is not present, it defaults to false.

The flag is set to *true* in the global configuration file of the KeePass installer package. The portable ZIP package does not contain a configuration file, consequently the flag defaults to *false*.

#### Loading:

- Try to get the configuration item from the enforced configuration file. If found, use this one.
- If the item is neither present in the global configuration file nor in the local one: use default value.
- If the item is present in the global configuration file, but not in the local one: use item from global configuration.
- If the item is present in the local configuration file, but not in the global one: use item from local configuration.
- If the item is present in the global and the local configuration file:
  - If the KeePreferUserConfiguration flag is True, use the item from the local configuration file, otherwise use the one of the global one.

#### Saving:

- If the KeePreferuserConfiguration flag is True, try to store the configuration item into the local configuration file. If this fails, try to store the item into the global configuration file. If this fails, report error.
- If the KeePreferUserConfiguration flag is *False*, try to store the item into the global configuration file. If this fails try to store the item into

the local configuration file. If this fails, report error.



# Field References How to put references to data in fields of other entries.

- Introduction
- Placeholder Syntax
- Example

#### Introduction

KeePass can insert data stored in different entries into fields of an entry. This means that multiple entries can share a common field (user name, password, ...), and by changing the actual data entry, all other entries will also use the new value.

To create a field reference, you can either use the convenient field references wizard (in the entry editing window, click the 'Tools' button at the bottom left and select 'Insert Field Reference'), or insert the placeholder manually (see the syntax below).

Note that field references are intended for referencing data stored in *different* entries. If you want to insert data from the *same/current* entry, you should use local placeholders, like {TITLE} and {S:FieldName}; see Placeholders.

#### Placeholder Syntax

The placeholder syntax for field references is the following:

{REF:<WantedField>@<SearchIn>:<Text>}

The *WantedField* and *SearchIn* parts need to be replaced by 1-letter codes identifying the field:

Code	Field
Т	Title
U	User name
Р	Password
A	URL
N	Notes
1	UUID
0	Other custom strings (KeePass 2.x only)

The *Text* part is the search string, i.e. this text must occur in the specified field of an entry to match.

If multiple entries match the specified search criterion, the first entry will be used. To avoid ambiguity, entries can be identified by their UUIDs, which are unique. Example:

{REF:P@I:46C9B1FFBD4ABC4BBB260C6190BAD20C} would insert the password of the entry having 46C9B1FFBD4ABC4BBB260C6190BAD20C as UUID.

#### Example

Let's assume you have two entries: one with title "Mozilla Website" and one with "Mozilla Forums", and want to insert the user name of the website account into the URL of the forums entry. Within the forum entry's URL, you could reference the user name like this: http://fictitious-forum.mozilla.org/?user={REF:U@T:Mozilla Website}



Import / Export KeePass supports importing/exporting data from/to various file formats.

KeePass 1.x supports importing data from **CSV files** (special form), **CodeWallet**, **Password Safe**, and **Personal Vault**.

KeePass 2.x supports importing data from CSV files (all), KeePass 1.x (KDB, XML and CSV), KeePass 2.x XML, 1Password Pro, 1PW, Alle meine Passworte, Any Password, CodeWallet, DataVault, FlexWallet, Handy Safe, Handy Safe Pro, Kaspersky Password Manager, KeePassX, Mozilla Bookmarks, PassKeeper, Passphrase Keeper, Password Agent, Password Depot, Password Exporter, Password Gorilla, Password Keeper, Password Memory, Password Prompter, Password Safe, Passwords Plus, Passwort.Tresor, Personal Vault, PINs, Revelation, RoboForm, Security TXT, SplashID CSV, Steganos Password Manager 2007, Sticky Password, Whisper 32, ZDNet's Password Pro, and Spamex.com.

For both KeePass 1.x and 2.x, there are importer plugins available, which add more import capabilities/formats.

- For KeePass 1.x:
  - File Format: CSV
  - File Format: XML
- For KeePass 2.x:
  - Generic CSV Importer
  - Formats that require custom options/steps in order to be imported:
    - How to Import CodeWallet TXT 6.05
    - How to Import Password Gorilla CSV 1.42
    - How to Import PINs TXT 4.50
    - How to Import Data from RoboForm
    - How to Import Data from Steganos Password Manager 2007
    - How to Import Data from PassKeeper 1.2
    - How to Import 1Password Pro CSV

Unfortunately there isn't any standard password database format. Every password manager uses its own file format. Anyway, almost all support exporting to CSV or XML files. This sounds good at first glance, but CSV and XML files aren't specialized password database formats, they only specify a low-level layout of the stored data (for CSV: data fields are separated by commas; for XML: hierarchical form using tags). These formats do not specify the high-level arrangement of the data (for CSV: order/meaning of the fields; for XML: tag names and structure). Because of this, many users are confused when application #1 exports data to CSV/XML and application #2 can't read the CSV/XML file, although it claims that it can read those files.

This help page details the expected CSV and XML file formats. Knowing the formats which KeePass expects, you can reformat CSV and XML files exported by other password managers to match the KeePass formats. CSV files can be reformatted using e.g. *OpenOffice Calc* (see below). XML files can be reformatted using an XML editor.

KeePass can import many password database formats directly (see top of this page). Additionally, there are specialized KeePass Plugins available for importing more formats (like AnyPassword CSV, Oubliette files, PINs TXT, ZSafe files, and many more...). Using these plugins, you don't need to manually reformat the output of other password managers; you can directly import the exported files.

If no import plugin exists for importing data from your previous password manager, feel free to post a request for it in the KeePass Feature Requests Tracker and/or in the Open Discussion forum.

#### File Format: CSV (KeePass 1.x)

KeePass imports and exports data from/to CSV files in the following format:

"Account", "Login Name", "Password", "Web Site", "Comments"

For a detailed example, download this file: FileSample\_CSV.zip. This file is zipped only in order to ensure correct encoding (if not zipped, browsers or download managers could automatically convert the file to a different encoding). When importing a CSV file, it must *not* be zipped!

Important notes about the format:

- The file must be encoded using UTF-8 (Unicode). Other encodings are not supported.
- CSV files only support the following fields: title, user name, password, URL and notes. Other fields like last entry modification time, expiration time, icon, entry file attachments, etc. are not supported. If you want to transfer such information, you have to use a different format (like XML).
- All fields must be enclosed in quotes ("). These quotes are mandatory, unquoted fields are not allowed.
- Quotes (") in strings are encoded as \" (two characters).
   Backslashes (\) are encoded as \\.
- Multiline comments are realized through normal line breaks.
   Encoding line breaks by \n is not supported.

Microsoft Excel by default does not enclose fields in quotes ("). It is recommended that you use OpenOffice Calc to create a correct CSV file (see below), or use the Generic CSV Importer of KeePass 2.x (import your CSV file into KeePass 2.x, then export the data to a KeePass 1.x KDB file), or fix the CSV file by manually adding the quotes using a text editor.

If you want to transfer data between KeePass 1.x databases, you must not change the default export options of KeePass. Do not export additional fields or uncheck any options, otherwise KeePass will not be able to re-import the CSV file, because it does not comply to the specification above any more.

#### Using OpenOffice Calc to create a CSV file:

OpenOffice Calc can be used to create CSV files that can be imported correctly into KeePass. Follow these steps:

- Make sure you got 5 columns as described above.
- Select everything, right-click and select 'Format Cells'. In the dialog, choose Text as category. Click [OK].
- Go 'File' -> 'Save As...', choose a location and make sure that the check box 'Edit Filter Settings' is enabled. Click the 'Save' button.
- Choose 'Unicode (UTF-8)' as character set. The field separator must be set to a comma. The text separator must be ". Make sure that the Fixed Column Width box is not checked. Click [OK].

#### File Format: XML (KeePass 1.x)

This section describes the KeePass 1.x XML format. Note that this format is different from the XML format used by KeePass 2.x (anyway, KeePass 2.x can import KeePass 1.x XML files).

You can download a detailed XML sample file here: FileSample\_XML.zip. This file is zipped only in order to ensure correct encoding (if not zipped, browsers or download managers could automatically convert the file to a different encoding). When importing a XML file, it of course must *not* be zipped!

Important notes about the format:

- The files must be encoded using UTF-8 (Unicode). Other encodings are not supported.
- The following five entities must be encoded: < > & " '. They are encoded by &1t; &gt; &amp; &quot; &apos;.
- The UUID is a hex-encoded 16-byte string (i.e. an 32 ANSI hex character string in the XML file). It is unique (also across multiple databases) and can be used to identify entries.
- Dates/times are encoded in the standard date/time XML format (YYYY-MM-DDTHH:mm:ss): first the date in form YYYY-MM-DD, a 'T' character, and the time in form HH:mm:ss.

#### Generic CSV Importer

KeePass 2.x ships with a generic CSV importer. This tool can import almost all CSV formats out there. The CSV files are loaded and you can manually specify the encoding / character set, assign columns to data fields, and specify how the low-level structure looks like (usage of quotes, etc.).

To start the generic CSV file importer, click 'File' -> 'Import' and choose 'Generic CSV Importer'.

#### How to Import CodeWallet TXT 6.05

CodeWallet is a password manager that supports different card types (fields). KeePass cannot know which of the CodeWallet fields correspond to the KeePass standard fields (title, user name, ...), because they don't have fixed names (language-dependent, user-customizable, ...). Therefore all fields from the CodeWallet file are imported into custom string fields of KeePass entries. After importing the file, you can move some of the strings to the correct standard fields (by clicking the 'Move' button on the second tab page of the entries dialog.

#### How to Import Password Gorilla CSV 1.42

In order to successfully import a Password Gorilla CSV file, you need to configure the following settings in 'File - Preferences - Export' in Password Gorilla:

- Enable 'Include password field'.
- Enable 'Include notes field'.
- Enable 'Save as Unicode text file'.
- Set the field separator to 'μ' (mu, press AltGr+M).
- Make sure you didn't use the 'μ' character anywhere in your password database.

When you carefully set these options, export the data to a CSV file and import it into KeePass using 'File -> Import' in KeePass 2.x.

### **■ How to Import PINs TXT 4.50**

In order to successfully import a PINs TXT file, you need to do the following:

- Switch PINs language to 'English'.
- In PINs export dialog: Enable all fields.
- In PINs export dialog: Set separator to 'tab'.
- In PINs export dialog: Enable 'Quote texts'.

After exporting a TXT file using the settings above, import it using 'File -> Import' in KeePass 2.x.

#### How to Import Data from RoboForm

First you need to export your passcards to a HTML file. To do this, open RoboForm's *Passcard Editor* (*'Edit Passcards'* or *'RoboForm Editor'* in the Windows start menu) and in the editor's main menu go *'Passcard'* -> *'Print List'* (in newer versions you have to click the *'RoboForm'* button and go *'Print List'* -> *'Passcards'*). In the dialog that opens, click the *'Save'* button. Choose a location and file name, and click *'Save'*.

Open your KeePass 2.x database file and go 'File' -> 'Import'. Choose 'RoboForm HTML' as format and select the HTML file you just exported, then click 'OK'.

#### How to Import Data from Steganos Password Manager 2007

**Warning!** It is possible that the transfer fails and that KeePass accidently overwrites your existing passwords in Steganos Password Manager. Therefore backup your SEF file before starting the import! In any case you should restore your passwords by restoring the backup you just created after the import process! Even if you think KeePass hasn't changed anything, restore from the backup!

Unfortunately Steganos Password Manager (SPM) lacks any form of export functionality. As the SEF file format (in which the data is stored) is proprietary and no specification is available, KeePass needs to try to get all the data out of the windows of SPM.

The import process works as follows. First you start SPM and open your password database. The main password management window should be open (i.e. the one which lists your items in the middle of the screen, and got toolbar-like buttons at the top). Make sure that *all* your items are displayed in the list (select the correct filter in the combobox above the item list).

Now switch to KeePass 2.x and open your KeePass database. Go *File -> Import* and choose *Steganos Password Manager 2007*. Click [OK]. Now read the rest before continuing.

After pressing the [Yes] button in the KeePass import confirmation dialog, you got 10 seconds to switch to the SPM window. Select the very first entry within the SPM window (but do not open it, just select it). This is important! The first entry must have the keyboard focus and must be selected.

When the 10 seconds are over, KeePass will start importing. You will see how KeePass opens the SPM items, copies the data, closes the item's window, select the next item, etc. Everything goes automatic now and you can just sit back and watch. Sometimes Windows playes a *ding* sound, this is normal.

Note that it can take quite some time to import your items. Do **not** do

anything while KeePass is importing! One single mouse click or keypress can ruin the complete import process.

The last item will be scanned twice. When completed, KeePass will show a message "The import process has finished!".

It is possible that KeePass failed to import some items (mainly caused by SPM's unpredictable slow response times). It is highly recommended that you compare each of the entries.

How to Import Data from PassKeeper 1.2		
The import process works visually, exactly like the import method for Steganos Password Manager data. Please read all instructions in How to Import Data from Steganos Password Manager.		

## How to Import 1Password Pro CSV

KeePass can import CSV files exported by 1Password Pro. When exporting the data, make sure:

- All fields must be exported, in the original order.
- Choose the tabulator (Tab) as field separator.
- The option for enclosing fields in quotes must be enabled.



Integration How KeePass integrates into your operating system environment.		
<ul> <li>Global Hot Key to Restore KeePass Window</li> <li>Limit to Single Instance Option</li> </ul>		

#### Global Hot Key to Restore KeePass Window

To quickly switch back from an application to KeePass, you can use the global hot key that restores the KeePass main window.

If you have multiple instances of KeePass running, pressing the global hot key will restore the first instance that has been started.

The global hot key is **Ctrl+Alt+K**.

The hot key cannot be changed, however it can be disabled in the advanced options.

#### Limit to Single Instance Option

If you enable the 'Limit to Single Instance' option, at most one KeePass instance can be running at a time. If you try to start a second KeePass instance, it is immediately terminated, and the first instance is brought to the foreground.

If the second instance was started using a command line specifying a database to open, and the first instance doesn't have a sub-dialog open, the first instance attempts to close the current database. If this attempt is successful (or if no database was open in the first place), the first instance opens the database specified by the second instance's command line, using any <code>-pw</code> or <code>-keyfile</code> options specified by the second instance. All other command line options in the second instance are ignored.



# Multi-User Details about multi-user features of KeePass.

### • Shared Database Editing:

- General Information about Shared Databases
- KeePass 1.x: Office-Style Locking
- KeePass 2.x: Synchronize or Overwrite

#### **&** General Information about Shared Databases

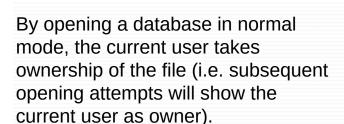
Both KeePass 1.x and 2.x allow multiple users working with one database, which is typically stored on a shared network drive or a file server.

All users use the same master password and/or key file to open the database. There are no per-group or per-entry access control lists (ACLs).

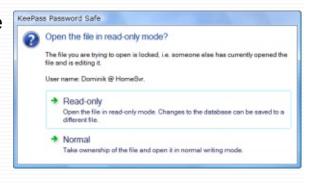
In order to restrict write access to the database file (i.e. only a select set of users may change the stored data), use file system access rights.

#### KeePass 1.x: Office-Style Locking

With KeePass 1.x, a database can be stored on a shared network drive and used by multiple users. When a user tries to open a database that is already opened by someone else, a prompt asks whether to open the database in read-only or normal mode (see image on the right).



KeePass 1.x does not provide synchronization, i.e. by saving the database you are saving your current data to disk. If another user has changed an entry in the meanwhile (i.e. since you loaded the database), these changes are overwritten.



If you want to use KeePass 1.x with a database on a shared network drive, it is recommended to let an administrator write to the database and let users only read it (ensure this using file system access rights). By using the -readonly command line switch, KeePass will automatically open a given database in read-only mode (i.e. not show the mode prompt). Users would open the database using a shortcut that contains this command line switch.

If there is no central administrator managing the database, users need to be careful to not overwrite each others changes.

#### KeePass 2.x: Synchronize or Overwrite

With KeePass 2.x, a database can be stored on a shared network drive and used by multiple users. When attempting to save, KeePass first checks whether the file on disk has been modified since it was loaded. If yes, KeePass asks whether to synchronize or overwrite the file (see image on the right).



By synchronizing, changes made by other users (file on disk) and changes made by the current user are merged. After the synchronization process has finished, the current user also sees the changes made by others (i.e. the data in the current KeePass instance is up-to-date).

If there is a conflict (multiple users edited the same entry), KeePass uses the latest version of the entry based on the last modification time.

Note: the synchronize prompt is only triggered by the 'Save' command, not by the 'Save As' command. When executing the 'Save As' command and manually selecting a file, this file will always be overwritten.



# Password Generator Details about the built-in password generator of KeePass.

- Generation Based on Character Sets
- Generation Based on Patterns
- Generating Passwords that Follow Rules
- Security-Reducing Options
- Creating and Using Password Generator Profiles
- Configuring Settings of Automatically Generated Passwords for New Entries

#### **@** Generation Based on Character Sets

This password generation method is the recommended way to generate random passwords. Other methods (pattern-based generation, ...) should only be used if passwords must follow special rules or fulfill certain conditions.

Generation based on a character set is very simple. You simply let KeePass know which characters can be used (i.e. upper-case letters, digits, ...) and KeePass will randomly pick characters out of the set.

#### **Defining a Character Set:**

The character set can be defined directly in the password generator window. For convenience, KeePass offers adding commonly used ranges of characters to the set. This is done by ticking the appropriate check box. Additionally to these predefined character ranges, you can specify characters manually: all characters that you enter in the 'Also include the following characters' edit box will be directly added to the character set.

#### **Character Sets are Sets:**

In mathematical terms, character sets are sets, not vectors. This means that characters can not be added twice to the set. Either a character is in the set or it is not.

For example, if you enter 'AAAAB' into the additional characters box, this is exactly the same set as 'AB'. 'A' will not be 4 times as likely as 'B'! If you need to follow rules like 'character A is more likely than B', you must use pattern-based generation + permuting password characters.

KeePass will 'optimize' your character set by removing all duplicate characters. If you'd enter the character set 'AAAAB' into the additional characters box, close and reopen the password generator, it'll show the shorter character set 'AB'. Similarly, if you tick the *Digits* check box and enter '3' into the additional box, the '3' will be ignored because it is already included in the *Digits* character range.

### **@** Generation Based on Patterns

The password generator can create passwords using patterns. A pattern is a string defining the layout of the new password. The following placeholders are supported:

Placeholder	Туре	Character Set
a	Lower-Case Alphanumeric	abcdefghijklmnopqrstuvwxyz 0123456789
A	Mixed-Case Alphanumeric	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789
U	Upper-Case Alphanumeric	ABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456789
С	Lower-Case Consonant	bcdfghjklmnpqrstvwxyz
С	Mixed-Case Consonant	BCDFGHJKLMNPQRSTVWXYZ bcdfghjklmnpqrstvwxyz
Z	Upper-Case Consonant	BCDFGHJKLMNPQRSTVWXYZ
d	Digit	0123456789
h	Lower-Case Hex Character	0123456789 abcdef
Н	Upper-Case Hex Character	0123456789 ABCDEF
1	Lower-Case Letter	abcdefghijklmnopqrstuvwxyz
L	Mixed-Case Letter	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz
u	Upper-Case Letter	ABCDEFGHIJKLMNOPQRSTUVWXYZ

р	Punctuation	··· ···
b	Bracket	0[[{}<>
S	Printable 7-Bit Special Character	!"#\$%&'()*+,/:;<=>?[\]^_{ }~
S	Printable 7-Bit ASCII	A-Z, a-z, 0-9, !"#\$%&'()*+,/:;<=>? [\]^_{ }~
V	Lower-Case Vowel	aeiou
V	Mixed-Case Vowel	AEIOU aeiou
Z	Upper-Case Vowel	AEIOU
x	High ANSI	From '~' to U255 (excluding U255)
\	Escape (Fixed Char)	Use following character as is
{n}	Escape (Repeat)	Repeat the previous character <i>n</i> times
[]	Custom Char Set	Define a custom character set

The \ \ placeholder is special: it's an escape character. The next character that follows the \ \ \ is written directly into the generated password. If you want a \ \ \ in your password at a specific place, you have to write \\.

Using the  $\{n\}$  code you can define how many times the previous placeholder should occur. The  $\{\ \}$  operator duplicates placeholders, not generated characters. Examples:

- » d{4} is equivalent to dddd,
- » dH{4}a is equivalent to dнннна and
- » нda{1}dн is equivalent to нdadн.

The [...] notation can be used to define a custom character set, from

which the password generator will pick one character randomly. All characters between the '[' and ']' brackets follow the same rules as the placeholders above. Examples:

- » [dp] generates exactly 1 random character out of the set digits + punctuation,
- » [d\m\@]{5} generates 5 characters out of the set "0123456789m@",
- »  $[u \ ][u \ ]$  generates 2 characters out of the set upper case + '\_'.

### More examples:

ddddd

Generates for example: 41922, 12733, 43960, 07660, 12390, 74680, ...

\H\e\x\:\ HHHHHH

Generates for example: 'Hex: 13567A', 'Hex: A6B99D', 'Hex: 02243C', ...

### **Common Password Patterns:**

Name	Pattern
WEP Hex Key - 40 Bit	h{10}
WEP Hex Key - 128 Bit	h{32}
WEP Hex Key - 256 Bit	h{64}
Random MAC Address	HH\-HH\-HH\-HH\-HH

# Generating Passwords that Follow Rules

Below are a few examples how the pattern generation feature can be used to generate passwords that follow certain rules.

Important! For all of the following examples you must enable the 'Randomly permute characters of password' option!

Rule	Pattern
Must consist of 2 upper case, 2 lower case characters and 2 digits	uulldd
Must consist of 9 digits and 1 letter	d{9}L
Must consist of 10 alphanumeric characters, where at least 1 is a letter and at least 1 is a digit	LdA{8}
Must consist of 10 alphanumeric characters, where at least 2 are upper case and at least are 2 lower case characters	uullA{6}
Must consist of 9 characters out of the set "ABCDEF" and an '@' symbol somewhere in it	\@[\A\B\C\D\E\F]{9}

### Security-Reducing Options

The password generator supports several options like 'Each character must occur at most once', 'Exclude look-alike characters', and a field to explicitly specify characters that should not appear in generated passwords.

These options are reducing the security of generated passwords. You should only enable them if you are forced to follow such rules by the website/application, for which you are generating the password.

The options can be found in the 'Advanced' dialog / tab page.

If you enable a security-reducing option, the 'Advanced' button in the password generator window is shown in red.



### Creating and Using Password Generator Profiles

Password generator options (character set, length, pattern, ...) can be saved as password generator profiles.

### **Creating/Modifying a Profile:**

- 1. Open the Password Generator window.
- 2. Specify all options of the new profile.
- 3. Click the **Save** as **Profile** button.
- 4. Enter the name of the new profile, or select an existing profile name from the drop-down list to overwrite it. Close the dialog with *OK*.
- 5. If you want to immediately create a password using the new profile, click *OK/Accept*. Otherwise click *Cancel/Close* (the profile is not lost; profile management is independent of password generation).

### **Using a Profile:**

To use a profile, simply select it from the drop-down profiles list in the password generator window. All settings of this profile will be restored accordingly.

# **②** Configuring Settings of Automatically Generated Passwords for New Entries

When you create a new entry, KeePass will automatically generate a random password for it. The properties of these generated passwords can be configured in the password generator dialog.

To configure, specify the options of your choice and overwrite the '(Automatically generated passwords for new entries)' profile (see section above).

### **Disabling Automatically Generated Passwords:**

To disable automatically generated passwords for new entries, select 'Generate using character set' and specify 0 as password length.

Overwrite the appropriate profile (see above).



# Placeholders KeePass supports various placeholders.

In many places in KeePass (auto-type, URL field, triggers, ...), placeholders can be used.

- Entry Field Placeholders
- Entry Field References
- Paths and Date/Time Placeholders
- Environment Variables
- Other Placeholders

Placeholders are case-sensitive.

# Entry Field Placeholders

Field	Placeholder
Title	{TITLE}
User Name	{USERNAME}
URL	{URL}
Password	{PASSWORD}
Notes	{NOTES}

# **8** Entry Field References Fields of *other* entries can be inserted using Field References.

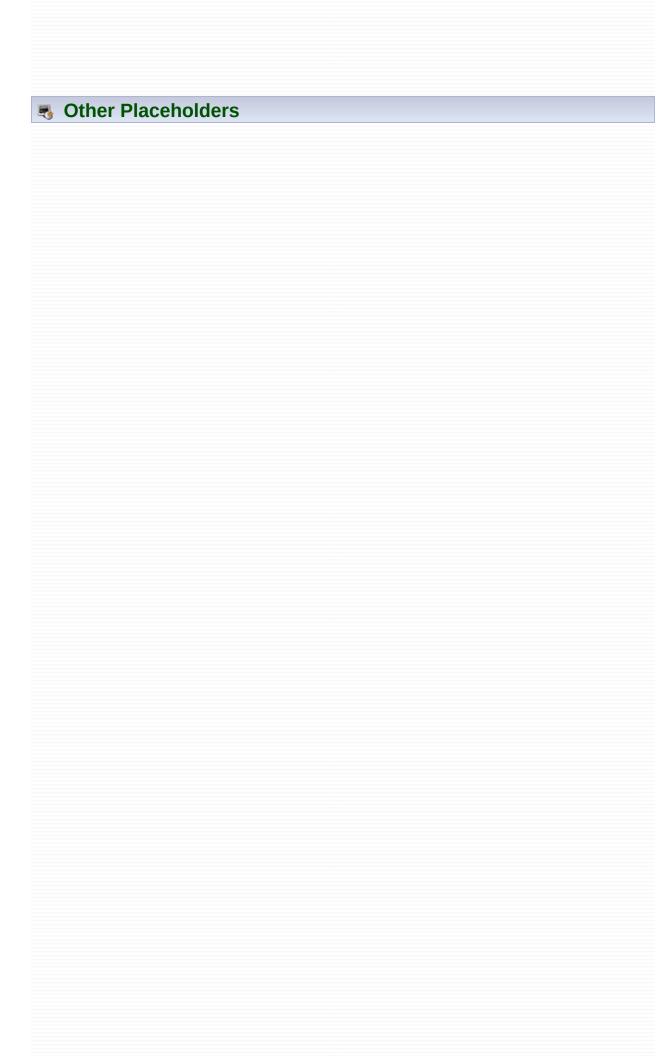
# Paths and Date/Time Placeholders

Placeholder	Is Replaced By
{INTERNETEXPLORER}	Path of Internet Explorer, if installed.
{FIREFOX}	Path of Mozilla Firefox, if installed.
{OPERA}	Path of Opera, if installed.
{GOOGLECHROME}	Path of Google Chrome, if installed.

Placeholder	Is Replaced By
JAPPINRU	KeePass application directory path.

# **B** Environment Variables

System environment variables are supported. The name of the variable must be enclosed in '%' characters. For example %TEMP% is replaced by the user's temporary path.





# Repairing Databases KeePass can repair corrupted databases in some cases.

KeePass has quite some features to avoid database file corruption (transacted database writing, device buffer flushing, ...). However, data corruption can still be caused by other programs, the system or broken storage devices (note that KeePass by default is verifying the integrity of database files immediately after writing them, i.e. at this point of time, KeePass guarantees file integrity; however, KeePass of course can't do anything when the data becomes corrupted/unreadable at a later point of time).

In these cases, the database repair functionality might help you. Here, KeePass tries to read as much data as possible from the corrupted file, without any integrity checks.

In KeePass 1.x, the repair functionality can be found in 'Tools' -> 'Repair KeePass Database File...'.

Anyway, if you've lost the master key for the database, the repair functionality cannot help you. Also, if the header of the database (first few bytes) is corrupted, you're out of luck too: the repair functionality won't be able to restore any entries (because the header contains information required to decrypt the database).

The repair functionality should be seen as last hope. Regularly making backups of your databases is much better and has to be preferred. Backups have *no* cryptographic security implications. There are plugins that automate the backup process, see the KeePass plugins page.



# Secure Edit Controls KeePass supports security-enhanced edit controls.

KeePass was one of the first password managers featuring secure edit controls. The edit controls used in KeePass are resistant to password revealers and password control spies. Additionally, the entered passwords are protected against memory dumping attacks: the passwords aren't even visible in the process memory space of KeePass!

KeePass uses secure edit controls only when the hiding behind asterisks option is turned on! If you show the passwords in plaintext, they won't be protected (secure edit controls are just disabled then, replaced by standard Windows edit controls).

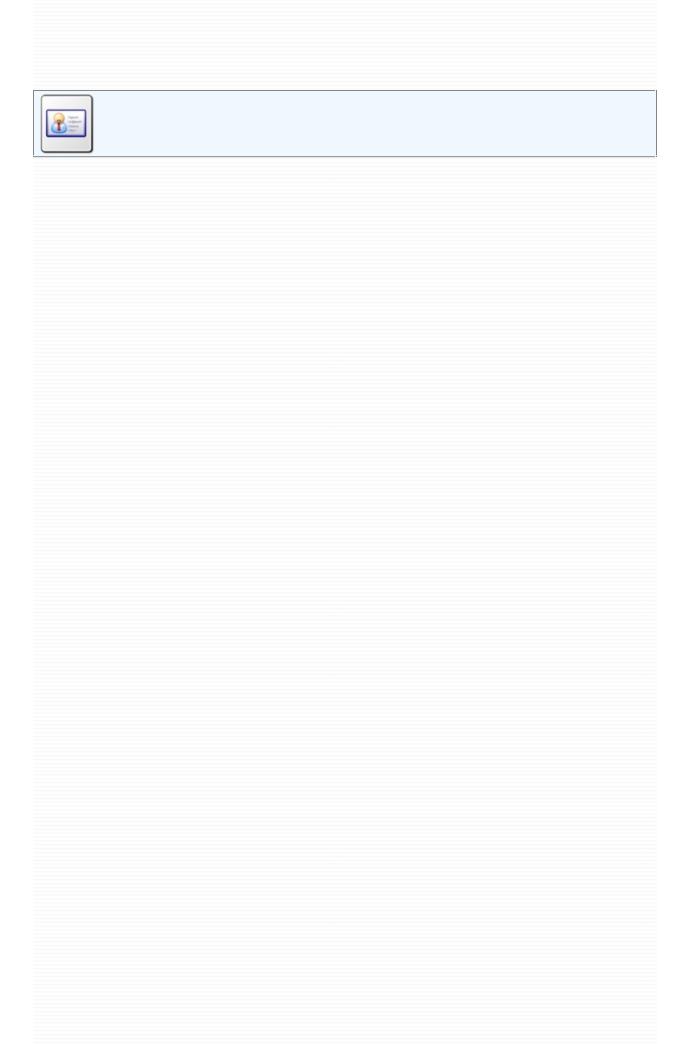
### **Selection Limitation**

A limitation of these secure edit controls is that you can't select a range of characters. You for example cannot select 3 characters and replace them by the current contents of the clipboard using the paste command.

If you want to delete all the contents of a secure edit control, press *Shift-Home* or *Shift-End*. This will remove all entered characters.

### **Technical background**

If you want to know more about how it's working internally, read this article (by the KeePass author) about secure edit controls: http://www.codeproject.com/KB/edit/SecEditEx.aspx.



# TAN Support KeePass supports Transaction Authentication Numbers (TANs).

- Using the *TAN Wizard* to add TANs
- Using TANs

KeePass supports TANs, i.e. passwords that can be used only once. These special passwords are used by some banks: you need to confirm transactions using such TANs. This provides additional security, as a spy cannot perform transactions, even if he knows the password of your banking account.

### Using the TAN Wizard to add TANs

You can use the KeePass **TAN Wizard** to add several TANs at once to your database. Just open the TAN wizard dialog (menu *Tools - TAN Wizard*) and enter all your TANs. The formatting doesn't really matter, KeePass just uses all alphanumerical strings, i.e. characters like line breaks, tabs, spaces, dots, etc. are interpreted as separators.

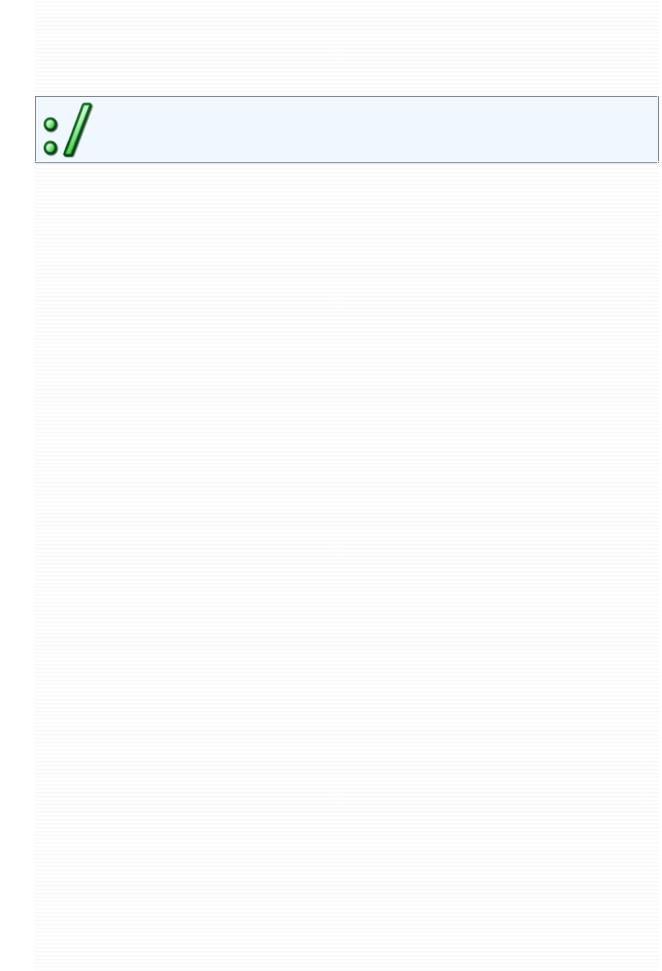
The wizard will then generate several TAN entries from the data you entered into the dialog. Each TAN is a standard KeePass entry. The title of a TAN entry always is set to "<TAN>". This tells KeePass that the entry is a TAN entry. You cannot change the title, user name and URL of a TAN. But you can freely add notes to a TAN entry, if you wish.

### Using TANs

When you use the TAN (e.g. execute the "Copy Password" command on it), its expiration date will be set to the current time, which expires the entry. It will get a red  $\mathbf{X}$  as icon. If you later want to know when you used a specific TAN, you can just have a look at its expiration date.

When copying a TAN to the clipboard, the database is marked as modified. You must save the file in order to remember the usage of a TAN.

If you accidently used a TAN without needing it, you can reset it (i.e. remove the red X and show it as valid TAN again). To do this, open the TAN entry (right-click it and choose 'Edit/View Entry...'). Here, uncheck the 'Expires' checkbox. Click [OK] to close the dialog.



# URL Field Capabilities The URL field supports various special protocols and placeholders.

### URL Field Capabibilities:

- Standard Capabilities
- Executing Command Lines
- Placeholders
- Changing the URL Handler (URL Override)

### Usage Tips & Tricks:

- Starting RDP/TS Sessions (Remote Desktop / Terminal Server Connection)
- Executing Built-In Shell Commands

### Standard Capabilities

The URL field can execute any valid URL, for which you have defined a protocol handler. On most systems at least the http://, ftp:// and mailto: protocols are defined. KeePass supports all registered protocols that Internet Explorer supports.

For example, if you globally (i.e. using the Windows Explorer) register PuTTY for ssh:// URLs, KeePass will automatically use PuTTY for ssh:// URLs, too.

### Executing Command Lines

Instead of an URL, you can also execute command lines using the URL field. To tell KeePass that the line you entered is a command line, prefix it using cmd://. For example if you would like to execute Notepad, your URL could look like this:

cmd://C:\Windows\Notepad.exe C:\Test\MyTestFile.txt

The virtual cmd:// protocol also supports parameters for executable files, in contrast to the file:// protocol. This was the main reason why cmd:// was introduced; with file:// you aren't able to pass any parameters to started applications. Use the cmd:// protocol instead.

The paths for the cmd:// protocol don't need to be encoded. For example, you do not have to replace space characters by %20, as it is normally required for other URLs. KeePass just cuts away the cmd:// virtual protocol prefix and passes the remaining command line to the system.

If the file path contains spaces, you must enclose it in quotes (").

### **Environment Variables:**

System environment variables are supported. The name of the variable must be enclosed in '%' characters. For example %TEMP% is replaced by the user's temporary path.

### **UNC Paths:**

Windows-style UNC paths (starting with \\) are directly supported, i.e. do not need to be prefixed with cmd://.

### Placeholders

In the URL field, you can use several placeholders that will get automatically replaced when the URL is executed. For example:

http://www.yoursite.com/default.php?user={USERNAME}&pass={PASSWORD}

For this entry, KeePass will replace {USERNAME} by the data of the username field and {PASSWORD} by the data in the password field when you execute the link.

For a complete list of supported placeholders, see the page Placeholders.

Also note that the special placeholders are supported, too. For example, the {APPDIR} placeholder is replaced by the application directory path of the currently running KeePass instance. It's the absolute path of the directory containing the KeePass executable, without a trailing backslash. If you would like to start a new KeePass instance, you could set the URL to:

cmd://"{APPDIR}\KeePass.exe"

To use different browsers for entries, you can use URLs like the following:

cmd://{INTERNETEXPLORER} "http://www.yoursite.com"

cmd://{FIREFOX} "http://www.yoursite.com"

cmd://{OPERA} "http://www.yoursite.com"

cmd://{GOOGLECHROME} "http://www.yoursite.com"

The browser placeholder will be replaced by the browser's executable path (if the browser is installed).

### Changing the URL Handler (URL Override)

The URL field behavior can be overridden individually for each entry using the *Notes* field. This allows you to execute a specific URL, while still using the URL field to (only) store data.

Simply enter url-override: followed by the command line you require into the notes field. When double-clicking the URL field of the entry in the main window, the specified command line (in the notes field) will be run.

### Using a different browser:

If your default browser is Firefox and you want to open a specific site with Internet Explorer, add the following to the Notes field:

```
Url-Override: cmd://{INTERNETEXPLORER} "{URL}"
```

KeePass will open Internet Explorer and pass the data from the URL field as the parameter. This uses a placeholder to find Internet Explorer.

### Globally changing the URL behavior:

If you want to change the *default* URL action (i.e. for *all* URLs), you can add a KeeUrloverride line to the KeePass.ini file.

### Starting RDP/TS Sessions

You can use the URL field of entries and the virtual cmd:// protocol to start remote desktop connections.

For this, enter the following in the URL field of an entry:

cmd://mstsc.exe

When you now double-click the URL field of the entry in the main window, a Windows remote desktop connection is initiated.

MSTSC is the Windows terminal server connection program (remote desktop connection). You can pass a path to an existing RDP file to the program to open it. For example, the following URL opens the specified RDP file:

cmd://mstsc.exe "C:\My Files\Connection.rdp"

MSTSC also supports several command line options:

### /v:<Server[:Port]>

Defines the terminal server to connect to.

### /console

Connects to the terminal session of the server.

• /f

Starts the client in full screen mode.

/w:<Width>

Defines the width of the remote desktop screen.

/h:<Height>

Defines the height of the remote desktop screen.

/edit

Opens the specified RDP file for editing.

• /migrate

Migrates old connection files to new RDP files.

### Executing Built-In Shell Commands

The URL field can be used to start applications/documents and URLs. If you want to execute a built-in shell command, like copy for example, this however doesn't work directly, because there is no copy. EXE (in Windows 9x times there actually was one, but on all modern Windows operating systems these commands are built-in to the command line window).

In order to execute built-in shell commands, you need to pass them to the command line interpreter cmd.exe.

For the copy command you would specify cmd.exe as executable file and /c copy from to as arguments (where 'from' and 'to' are paths). The /c parameter tells cmd.exe to execute the command line that follows.

In the URL field, your URL would look like the following:

cmd://cmd.exe /C COPY from to

In other locations, like command lines in the trigger system, you can leave out the cmd:// URL prefix.

-

Using Stored Passwords How to transfer passwords stored in KeePass to other applications.

There are many different methods to copy passwords stored in KeePass to other applications:

- Context-Sensitive Entry List
- Drag&Drop
- Auto-Type
- KeeForm and other Plugins

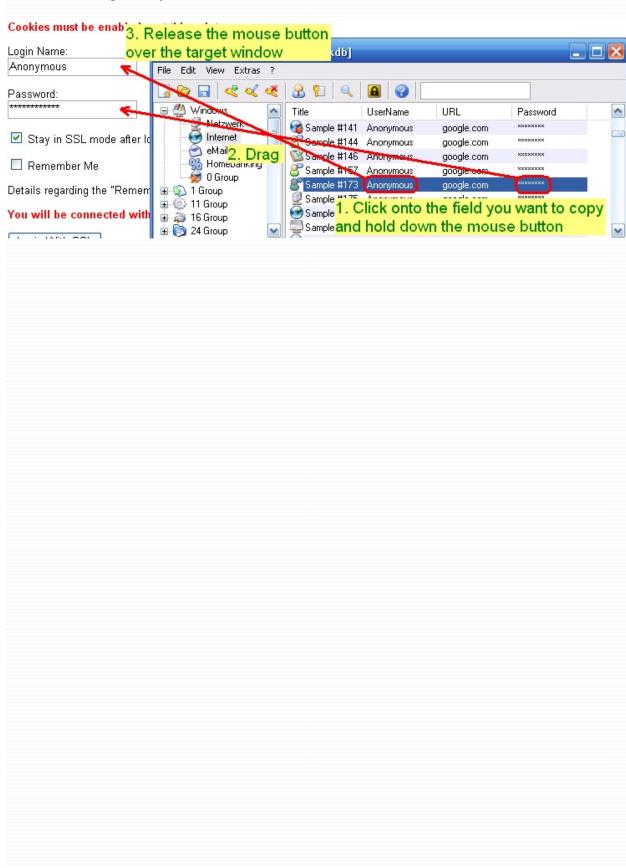
### Context-Sensitive Entry List

Depending on which field you double-click in the entry list (main window), different actions are performed:

- **Title field:** open the entry editing dialog for this entry.
- User Name field: copy user name to clipboard.
- **URL field:** open (in browser) or copy the URL to clipboard (behavior configurable in the options).
- Password field: copy password to the clipboard.
- Notes field: notes are copied to the clipboard.
- Attachment field: [1.x] copy to clipboard, [2.x] open in internal editor / viewer.
- Other fields (like time and UUID fields): copy the contents of that field to the clipboard.

# Drag&Drop

You can drag&drop all fields of KeePass entries into other windows:



**Auto-Type** Auto-Type is a powerful feature that sends simulated keypresses to other applications. You can find more details about it here: Auto-Type documentation page.

## KeeForm and other Plugins



# Administrative FAQ Frequently Asked Questions about the project, licensing, ...

- How can I help you? (Supporting the KeePass project)
- May KeePass be used in a company?
- What about a centralized KeePass Internet server?

### How can I help you?

If you like KeePass and would like to help the developers in some way:

### Donate

This is the best way of helping, if you don't have that much time or experience in application development.

### Make a translation

If you have some free time, you could make a translation of KeePass (of course only if you're language isn't offered already).

### • Test new releases and report bugs

KeePass is under constant development, new features get implemented, bugs get fixed. If you have some free time, you could test new releases thoroughly and report bugs. If you're a programmer, look through the sources to find bugs and maybe even submit fixes.

### Spread the word

If you like KeePass, tell all your friends how great KeePass is, publish articles about it, press it on CDs/DVDs, ship USB sticks preinstalled with it, submit it to software archives, talk in forums about it, etc.!

### May KeePass be used in a company?

Yes. KeePass is free software and you don't have to pay any fees. You may freely use KeePass under the terms of its license.

But of course, if you like KeePass, donations are always greatly appreciated.

You might be interested in this page: Customizing KeePass 1.x.

### What about a centralized KeePass Internet server?

The idea on the first glance sounds simple and useful: there should be a centralized KeePass Internet server, on which all users can store their passwords. By having Internet connection, you'd have access to all your passwords.

Note that this idea is different from simply providing webspace. KeePass 2.x already supports storing databases on servers using HTTP/FTP. The point is having one server for all users.

When creating such a server, there are several difficulties:

- A fairly complex synchronization and caching mechanism will be required. You won't want to transfer the complete database, otherwise the service will be unusable for everyone storing attachments, etc.
- Directly related to the previous point: in order to do synchronization, the server needs to be able to read and understand databases, i.e. some dedicated KeePass server would need to be written. While the transport way could be secure HTTPS, the server certainly has some user data as plain text in memory for some time. One needs to be very careful here. What to do if the server gets compromised? The security implications would be horrible, if an attacker could read any user data.
- How to avoid server compromises? If a normal Internet server is compromised, the security implications are minimal: in the worst case all user accounts and data for this website are lost. But with KeePass server, whole identities would be lost. An attacker couldn't only impersonate someone on this particular server, but on the complete Internet and real world, depending on what is stored in the databases.

Therefore, banking-level security systems would be required for a KeePass server. Keeping PHP / ASP / Linux / Windows (or whatever will be used) up-to-date definitely is not enough here.

Basically you offer people webspace for their databases, therefore
the service obviously will cost something. By charging people, they
expect reliability and you need to make up-time guarantees.
Therefore, at least 2 servers are required (by different hosters),
which need to be synchronized.

Summary: a centralized Internet server currently is out of range. If someone wants to start a company providing such a service, feel free to use KeePass as base application (of course respect the Open Source terms).

But what can and probably will be done later is a local intranet KeePass server (for companies for example). Employees could log in to the company's password server and use it. But a centralized Internet server - no chance.



### Technical FAQ Frequently Asked Questions about the usage of KeePass.

### Configuration:

 I've saved my options, but when I reopen KeePass I get the old options. What's wrong?

### Installation / Integration:

- Why doesn't the CHM help file work?
- Where can I find more application icons for Windows shortcuts?
- How can I add more *client icons* for password entries?
- Does KeePass support a mini mode?
- Why doesn't KeePass lock after Auto-Type?
- Why doesn't Auto-Type work correctly on Polish systems?
- Why doesn't printing work in KeePass 1.x?

### Security:

- Is Auto-Type keylogger-safe?
- Can Auto-Type locate child controls?
- Could you add the ... encryption algorithm to KeePass?
- Why doesn't KeePass lock when Windows locks and a KeePass sub-dialog is open?
- Printing creates a temporary file. Will it be erased securely?
- Why KeePass 2.x resets in-memory protection options?
- Why the estimated quality of a password suddenly drops?

### Usage:

- Can I directly edit file attachments?
- How to store and work with large amounts of (formatted) text?

### I've saved my options, but when I reopen KeePass I get the old options. What's wrong?

KeePass supports two different locations for storing configuration information: the global configuration file in the KeePass directory and a

local, user-dependent one in the user's private configuration folder. Most likely you do not have write access to your global configuration file.
For more details, see 🗎 Configuration.

### Why doesn't the CHM help file work?

**Symptoms:** When trying to open the KeePass CHM help file from a remote computer or shared network drive, it's not displayed correctly (navigation aborted, ...).

**Solution:** See Microsoft Security Bulletin MS05-026.

### Where can I find more application icons for Windows shortcuts?

Application icons are icons in Windows ICO format. They can be used in Windows shortcuts and/or as file association icons. The KeePass executable contains various application icons which can be used for these purposes.



Additional application icons are available from the "Ext/Icons" directory of the KeePass source code download. Most of them, shown at right, are slight variations of the main KeePass icon.

Even more, contributed icons (by users) can be found on the plugins page.

If you have multiple KeePass databases, you can use differently colored KeePass application icons in order to distinguish them.

These icons are not included in the binary distribution because this would make the application file too large.

### How can I add more client icons for password entries?

Client icons are the icons used for password entries and groups within KeePass. Each entry can be assigned its own icon.



These icons are built-in. You cannot add/import your own icons.

# O Does KeePass support a mini mode? Yes, see KeePass 1.x Mini Mode.

### Why doesn't KeePass lock after Auto-Type?

I have enabled the "Use alternative auto-type method (minimize window)" and "Lock workspace when minimizing the main window" options. Why doesn't KeePass lock after auto-typing?

In this very special case, the window minimization only is a way to lose the focus, i.e. the window below comes to the foreground. The minimization is not user-initiated (it's only a side-effect of auto-type), nor a consequence of an external minimization command, therefore it is not (and should not be) affected by the automatic workspace locking handler.

If you worry about having KeePass minimized and unlocked, enable the "Automatically lock workspace after the following number of seconds" option and specify a reasonable amount.

### Why doesn't Auto-Type work correctly on Polish systems?

On Polish systems, the default auto-type hot key *Ctrl+Alt+A* conflicts with a system command and is frequently used in typing. Therefore, auto-type is often executed accidentally.

The global auto-type hot key can be changed to a different key combination in the KeePass options (see Auto-Type for details).

### # Is Auto-Type keylogger-safe?

Is the Auto-Type feature resistant to keyloggers?

No. The Auto-Type feature has been designed in a way that it's impossible for target applications to distinguish real keys from auto-typed ones. This on the one hand has the advantage that the feature is really compatible with *all* applications out there. On the other hand, the auto-typed keys can of course be logged by keyloggers. If you worry about keyloggers, you have to use one of the other methods (drag&drop, copying to clipboard, KeeForm, ...).

### Can Auto-Type locate child controls?

No. Auto-Type only checks whether the title of the currently active top level window matches.

Browsers like Mozilla Firefox completely draw the window (all controls) themselves, without using standard Windows controls. Consequently it is technically impossible for KeePass to check whether an URL matches (methods like creating a screenshot and using optical character recognition are not reliable and secure). Also, it's impossible to check which child control currently has the focus. These problems can only be avoided by using browser integration plugins, i.e. not using auto-type at all.

The user must make sure that the focus is placed in the correct control before starting auto-type.

### **②** Could you add the ... encryption algorithm to KeePass?

Only Rijndael (AES) and Twofish are supported. There are no plans to add additional algorithms for the following reasons:

- **Backward Compatibility:** If new algorithms are implemented and used, older KeePass versions will not be able to read files encrypted with the new algorithms.
- Security: Some people are not well informed about encryption algorithms and may choose a weak algorithm such as TEA, if implemented. This would compromise the security of passwords managed by KeePass.
- Size & Functionality: KeePass is a secure password manager, not a Swiss army knife of algorithms.

# Why doesn't KeePass lock when Windows locks and a KeePass sub-dialog is open?

KeePass automatically tries to lock its workspace when Windows is locked, with one exception: when a KeePass sub-dialog (like the 'Edit Entry' window) is currently opened, the workspace is not locked.

To understand why this behavior makes sense, it is first important to know what happens when the workspace is locked. When locking, KeePass completely closes the database and only remembers several view parameters, like the last selected group, the top visible entry, selected entries, etc. From a security point of view, this achieves best security possible: breaking a locked workspace is equal to breaking the database itself.

Now back to the original question. Let's assume an edit dialog is open and the workstation locks. What should KeePass do now? Obviously, it's too late to ask the user what to do (the workstation is locked already and no window can't be displayed), consequently KeePass must make an automatic decision. There are several possibilities:

- Do not save the database and lock.
   In this case, all unsaved data of the database would be lost. This not only applies to the data entered in the current dialog, but to all other entries that have been modified previously.
- Save the database and lock. In this case, possibly unwanted changes are saved. Often you open files, try something, having in mind that you can just close the file without saving the changes. KeePass has an option 'Automatically save database when KeePass closes or the workspace is locked'. If this option is enabled and no sub-dialog is open, it's clear what to do: try to save the database and if successful: lock the workspace. But what to do with the unsaved changes in the sub-dialog? Should it be saved automatically, taking away the possibility of pressing the 'Cancel' button?
- Save to a temporary location and lock.
   While this sounds the best alternative at first glance, there are several problems with it, too. First of all, saving to a temporary

location could fail (for example there could be too few disk space or some other program like virus scanner could have blocked it). Secondly, saving to a temporary location isn't uncritical from a security point of view. When having to choose such a location, mostly the user's temporary directory on the hard disk is chosen (because it likely has enough free space, required rights for access, etc.). Therefore, KeePass databases could be leaked and accumulated there. It's not clear what should happen if the computer is shutdown or crashes while being locked. When the database is opened the next time, should it use the database stored in the temporary directory instead? What should happen if the 'real' database has been modified in the meanwhile (quite a realistic situation if you're carrying your database on an USB stick)?

Obviously, none of these alternatives is satisfactory. Therefore, KeePass implements the following simple and easy to understand behavior:

When Windows is locked and a KeePass sub-dialog is opened, the KeePass workspace is not locked.

This simple concept avoids all the problems above. The user is responsible for the state of the program.

Security consequence: the database is left open when Windows locks. Normally, you are the only one who can log back in to Windows. When someone else logs in (like administrator), he can't use your programs anyway. By default, KeePass keeps in-memory passwords encrypted, therefore it does not matter if Windows caches the process to disk at some time. So, your passwords are pretty safe anyway.

**Note.** On Windows  $\leq$  XP, the 'Terminal Services' Windows service should be enabled. If this service is disabled, locking KeePass when Windows locks might not work. This service isn't required on newer operating systems.

### Printing creates a temporary file. Will it be erased securely?

KeePass creates a temporary HTML file when printing password lists and showing print previews. This file is securely erased (i.e. overwritten multiple times before being removed from the file system tree) when closing the database.

You must wait for the file being printed completely before closing KeePass (and close the print preview before closing KeePass), otherwise it could happen that the printing application blocks KeePass from deleting the file.

There is no way around the temporary file in the current printing system. If you want to write a plugin that directly sends the data to the printer, you can find a plugin development tutorial here: KeePass 2.x Plugin Development.

### Why KeePass 2.x resets in-memory protection options?

Important: the following describes a behavior related to in-memory protection, which has nothing to do with database encryption. All database fields are encrypted; see the Security page for details.

For all fields except the password field, in-memory protection is turned off automatically when the fields are searched or displayed as plain text.

If you want data to be in-memory protected, you additionally must turn on hiding by asterisks for the column and never search this field type. In-memory protection is useless otherwise. If you don't hide by asterisks or search, your data is clear in memory, no matter whether the in-memory protection is on or off.

### Why the estimated quality of a password suddenly drops?

For estimating the quality/strength of a password, KeePass not only uses statistical methods (like checking which character ranges are used, repeating characters and differences), it also has a built-in list of about 1500 most common passwords. Such most common passwords are rated down to 1/8th of their statistical rating. Thus, the estimated quality can drop to a lower value while entering a password.

### Can I directly edit file attachments?

No, this is not possible (exception: text files, see working with large texts). In order to edit attachments, you need to save them to a (temporary) file, edit it using an external application, import it back to KeePass as attachment, and finally delete the temporary file.

There will no feature be implemented that automates these steps, because of security problems. To see the problems, let's assume that KeePass would support editing attachments. When you click a button, KeePass would save the attachment to a file and open it using its associated external application. When the external application is closed, KeePass would import the temporary file and delete it securely. But what happens when KeePass is closed before the external application? KeePass cannot delete the file because it's eventually locked by the external application. Theoretically KeePass could tell the user this fact before closing, but what to do when the computer shuts down? Here, there's no time left to ask the user what to do. The temporary file would have been leaked, i.e. left unencryptedly on disk, which is obviously very bad.

One could argue that the leakage would only be temporary: at the next start, KeePass could scan the temporary directory for remaining files and delete them. Anyway, the files would be freely accessible (unencrypted) by all other applications during a complete computer shutdown and boot process. If you don't start KeePass on this computer ever again, the file is leaked forever. As KeePass is designed to be portable, i.e. may be securely used on many computers, this temporary leakage is unacceptable.

# We have to store and work with large amounts of (formatted) text? There is no direct support for storing and working with large formatted texts.



**Customize** KeePass features various options for network administrators to customize the program's appearance and behavior.

- Preliminaries
- Mini Mode
- Unsafe Operations
- Groups in Newly Created Databases
- Window Title Suffix
- Dialog Banner Appearance
- Minimum Master Password Requirements
- More Options

### Preliminaries

Most options below are configured by directly editing the KeePass.ini configuration file. If you're planning to deploy a customized KeePass version, you should fully understand the KeePass configuration system, especially how to enforce some settings and leave others up to users.

Note that KeePass features a rich plugin framework. If there's no item in the INI file to configure what you're thinking about, you might want to write a plugin.

### Mini Mode

KeePass supports a mini mode. By specifying KeeMiniMode=True in the KeePass.ini file, KeePass will run in a minimal functionality mode.

In this mode, the following functionality is hidden:

- Configuration and administration options (database settings, global program options, plugins, ...).
- Functionality that is not interesting to users in a company (like checking for updates, ...).
- Import / export.
- Auto-Type.
- Key files (only a master password can be entered).
- File operations (only the Save command is shown).

If the specified database does not exist, KeePass automatically creates and opens one. To specify the base path of a newly created database, use the KeeAutoNewDbBasePath configuration option (must not end with a terminating '\'; KeePass will create the specified directory tree, if it doesn't exist already). To specify the base name of the database file, use the KeeAutoNewDbBaseName option (without ".kdb").

### Unsafe Operations

Security-critical operations (like changing the master key, printing, exporting, unhiding passwords, etc.) can be disabled by specifying KeeDisableUnsafe=True in the INI file.

Some operations can be re-enabled by explicitly specifying additional configuration items. Note that because of security reasons these additional items are only loaded from the enforced and global INI files, not from the user's local INI file. They override the unsafe operations and mini mode settings. The following boolean items are supported:

KeeForceAllowChangeMasterKey, KeeForceAllowPrinting,

KeeForceAllowImport, KeeForceAllowExport.

To disallow printing passwords, specify KeeDisallowPrintingPasswords=True.

### Groups in Newly Created Databases

Using the KeeRootInNewDb and KeeGroupInNewDb# items in the INI file, you can specify the groups that are automatically created when a user creates a new database file.

Example. If you'd like the initial tree to look like the one on Windows the right, your INI file would contain the following lines:

KeeRootInNewDb=@My Company Name@35 KeeGroupInNewDb0=@Windows@38 KeeGroupInNewDb1=@Network@3 KeeGroupInNewDb2=@Internet@1 KeeGroupInNewDb3=@eMail@19 KeeGroupInNewDb4=@Applications@32



Items must be numbered consecutively in ascending order starting from 0. The first character of the item's value specifies a separator character. In the example above, '@' is used as separator, but you could pick any other character (in case the group name contains an '@'). The group name follows the separator character and is terminated by the separator character. The number that follows is the ID of a built-in KeePass client icon (see the icon picker dialog of KeePass for a list of icons). Creating tree structures (i.e. subgroups of the groups above) is not supported.

### Window Title Suffix

A main window title suffix can be specified using the KeeWindowTitleSuffix item in the INI file.

### Dialog Banner Appearance

The appearance of dialog banners of KeePass can be configured using the following items in the INI file:

- KeeBannerColorStart: Specifies the starting color of the background gradient. Set it to a BGR byte triple (decimal).
- **KeeBannerColorEnd**: Specifies the end color of the background gradient. Set it to a BGR byte triple (decimal).
- **KeeBannerColorText**: Specifies the foreground text color. Set it to a BGR byte triple (decimal).
- **KeeBannerFlip**: Boolean value that specifies whether to swap the natural gradient direction (i.e. from horizontal to vertical or from vertical to horizontal). The natural direction depends on the KeePass version you're using.

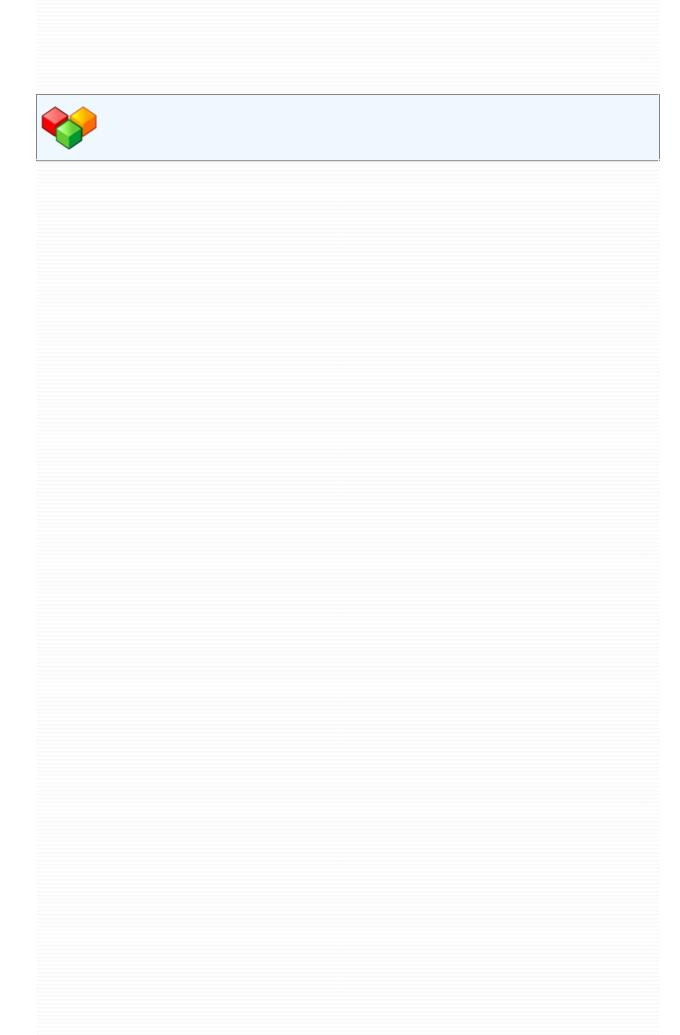
### **Minimum Master Password Requirements**

You can specify several properties that master passwords must have in order to be accepted (length, estimated quality, ...). See Specifying Minimum Master Password Properties.

### More Options

The number of days below which entries are treated as soon to expire can be specified using the **KeeSoonToExpireDays** configuration item.

In KeePass  $\geq$  1.17, the locations of the configuration files can be overridden using the **KeeConfigFileOverrideGlobal** and **KeeConfigFileOverrideUser** settings (in the enforced configuration file). Complete, absolute file paths should be specified, and the two overrides should *not* point to the same file.



### Plugin Development How to develop plugins for KeePass 1.x.

This documentation applies to KeePass 1.x plugins (all versions  $\geq$  1.15). 1.x plugins are fundamentally different from 2.x plugins. 2.x plugins cannot be loaded by KeePass 1.x.

A detailed SDK documentation is available here: Plugin SDK Documentation.

- Requirements
- Step-by-Step Tutorial
- Plugin Conventions and Recommendations
- Upgrading Plugins from  $\leq 1.14$  to  $\geq 1.15$
- Detailed Documentation of All Interfaces
- Plugin Framework for C++

### Requirements

Before you can start developing a KeePass plugin, you need the following prerequisites:

- Latest KeePass source code package. You can get it from the KeePass website.
- A C++ development IDE / compiler.
- Windows Platform SDK.

The KeePass plugin API uses some concepts of the Component Object Model (COM) standard. If you don't have experience with COM, the following pages are recommended for reading:

- Wikipedia: IUnknown.
- Reference counting section on Wikipedia: Component Object Model.
- MSDN: COM Interfaces and Interface Implementations.
- MSDN: COM Interface Pointers and Interfaces.
- MSDN: COM QueryInterface: Navigating in an Object.

### Step-by-Step Tutorial

Start your favorite IDE and create a new *Win32 Project* (application type *DLL*, empty project). In this tutorial the example plugin we're developing is called TestPlugin. Create two files in the new project: a C++ source file (*TestPlugin.cpp*) and a header file (*TestPlugin.h*).

In order to access the KeePass interfaces, you have to include a header file from the KeePass SDK: put an #include statement in the *TestPlugin.h* file, which includes the file KeePassLibCpp/SDK/KpSDK.h from the KeePass source code.

Windows DLLs may optionally implement a DllMain function. So if you want one (not required by KeePass though), implement a default one now in the *TestPlugin.cpp* file (just always return TRUE):

```
BOOL WINAPI DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)

{
    UNREFERENCED_PARAMETER(hinstDLL);
    UNREFERENCED_PARAMETER(fdwReason);
    UNREFERENCED_PARAMETER(lpvReserved);
    return TRUE;
}
```

The next step is to create a plugin class, which must implement the IKpPlugin interface. So, lookup the IKpPlugin interface (abstract C++ class) and design a class that implements all these methods. Details about the methods can be found in the Plugin SDK Documentation.

Now export a function, which KeePass will use to create an instance of your plugin class:

```
KP_EXPORT HRESULT KP_API KP_I_CREATEINSTANCE_DECL(REFIID riid,
void** ppv0bject, IKpUnknown* pAPI);
```

In this function, you'll need to create an instance of your plugin class and store an interface pointer of the type requested by KeePass (riid) in the ppv0bject parameter. The papi parameter is an interface pointer to the KeePass API, which you should store for later use in case you're able to return a valid plugin interface.

KeePass is currently offered only as ANSI application, not Unicode. Therefore, go to *Project -> Test Plugin Properties* and choose *Multi Byte*  as character set.

It is recommended (but not required) to statically link with the runtime library (and MFC, if you're using it). To do this, go to *Project -> Test Plugin Properties -> C/C++ -> Code Generation* and choose a runtime library not ending with '-DLL'.

The last step before building your plugin is to add a version information resource. So, go to the 'Resources' tab of the plugin project and add a resource of type 'Version'. Here, set the product name to Keepass Plugin. All other fields can be freely set to strings of your choice.

**Example.** You can find a fully documented and extended version of this simple plugin on the KeePass plugins web page ("Test Plugin").

### Plugin Conventions and Recommendations

### Conventions:

- The DLL file must have a version information resource, in which the product name is set to KeePass Plugin. KeePass uses this to determine whether the DLL is a KeePass plugin or not (i.e. if you don't create a version information resource with this string, KeePass will not load your DLL file).
- A KeePass plugin must export the following function:
   KP\_EXPORT HRESULT KP\_API KP\_I\_CREATEINSTANCE\_DECL(REFIID riid, void\*\* ppv0bject, IKpUnknown\* pAPI);

KeePass will call this function to create an instance of your plugin class. You have to return an interface of type riid in the ppv0bject parameter, if your plugin class supports this interface (return s\_ok). Otherwise set ppv0bject to NULL and return E\_NOINTERFACE. You may store the papi interface pointer for later use. KeePass guarantees that the pointer is valid as long as it has a pointer to your plugin class instance.

Important: explicitly check for which interface KeePass is asking (riid), otherwise your plugin is not upward-compatible and might crash in future KeePass versions.

A KeePass plugin may optionally export the following functions:
 KP\_EXPORT HRESULT KP\_API KP\_I\_INITIALIZELIB\_DECL(IKpUnknown\* pAPI);
 KP\_EXPORT HRESULT KP\_API KP\_I\_RELEASELIB\_DECL(IKpUnknown\* pAPI);

KeePass will call the first function after the DLL is loaded, and the second one shortly before the DLL is unloaded.

You should not store the papi interface pointer for later use. Consider the pointers as temporary; they might become invalid as soon as you

return from the KP\_I\_INITIALIZELIB\_DECL or KP\_I\_RELEASELIB\_DECL function. The papi interface pointer values passed to KP\_I\_INITIALIZELIB\_DECL and KP\_I\_RELEASELIB\_DECL are not guaranteed to be the same as each other, or as the pointer value passed to KP\_I\_CREATEINSTANCE\_DECL.

- The protocol is DllMain (if present), KP\_I\_INITIALIZELIB\_DECL (if present), KP\_I\_CREATEINSTANCE\_DECL, plugin interface methods, KP\_I\_RELEASELIB\_DECL (if present).
- KeePass is using the multi-byte character set. Therefore, make sure that you're also compiling your plugin in multi-byte character set mode, not Unicode.

### Recommendations:

- All plugin files should begin with a common prefix. For example, if your plugin is called *VariousImport*, the DLL file might be named *VariousImport.dll* and its help file *VariousImport.html*. If you don't use a common prefix, users might run into overwriting problems when installing multiple plugins, because all plugins must be copied into the KeePass application directory. For example, if there's a plugin that ships with a *ReadMe.txt* file and another plugin that also ships with such a file, the latter overwrites the readme file of the first, or the user chooses not to overwrite and the readme file of the second plugin is not available. Using a common prefix avoids this problem.
- The version information block should at least be available in English (USA) language.
- There are two implementations of the Ikpconfig interface. One implementation is identified by CLSID\_KpConfig, the other by CLSID\_KpConfig\_ReadOnly. The first one supports both reading and writing, the second one only reading. It is highly recommended that you use the second implementation, if you only want to read configuration items.

Trying to write using the <code>clsid\_KpConfig\_ReadOnly</code> implementation will assert if KeePass is compiled in Debug mode, and will fail in Release mode (and might possibly destroy parts of the current configuration).

### **②** Upgrading Plugins from ≤ 1.14 to ≥ 1.15

When upgrading a plugin from KeePass  $\leq 1.14$  to  $\geq 1.15$ , it is highly recommended that you create a new project file, start from scratch and copy / fill out the interface methods with the old code.

### Notes:

• There is no κeepass.lib file anymore. The new plugin architecture is based on interfaces. Including the κpspk.h header file is all you need to do.

Do not compile with any of the files in the KeePass source code or include any other header file than kpsdk.h.

 Previously a command line prefix was registered by setting the cmdLineArgPrefix member of the plugin information structure. In the new architecture, the command line prefix must be returned by the GetProperty member method of the plugin interface when being called with the KPPS\_COMMANDLINEARGPREFIX parameter.

Your GetProperty might look like this:

```
STDMETHODIMP_(LPCTSTR) CYourPluginImpl::GetProperty(LPCTSTR lpName)
{
    if(lpName == NULL) return NULL;

    if(_tcscmp(lpName, KPPS_COMMANDLINEARGPREFIX) == 0)
        return _T("mypluginprefix.");

    return NULL;
}
```

The plugin should not access the KeePass command line until after all plugins have been loaded. This is because at that time KeePass of course can't call the <code>GetProperty</code> method of your plugin yet and consequently doesn't know the prefix yet (and this will lead to 'unknown command line option' warnings). Instead, perform command line dependent initialization when KeePass calls your <code>OnMessage</code> handler method with the <code>KPM\_DELAYED\_INIT</code> code.

# Operation of All Interfaces See the Plugin SDK Documentation.

### Plugin Framework for C++

Thanks to Bill Rubin, there's an optional *Plugin Framework* available, facilitating the development of KeePass plugins in C++. The features in detail:

- 1. Complete implementation of the IID\_IKpUnknown interface.
- 2. Implementation of the IID\_IKpPlugin interface, except for the onMessage member function, which is always application-specific.
- 3. Ability to get smart pointers to any other KeePass interface, including IKpAPI, IKPCommandLine, IKpCommandLineOption, IKpConfig, IKpDatabase, IKpFullPathName, and IKpUtilities.
- 4. Implementation of the κpcreateInstance function, which the plugin must export from its DLL.
- 5. Comprehensive error checking of COM handshaking for Items 3 and 4. If an error is found, PFK displays a message box containing all information about the error. Without this feature, a plugin will, in most cases, silently fail to load. In other cases, it will fail to perform its function.
- 6. Convenience utilities to display a message box, translate a Windows error code into a natural language string, and declare a standard string independently of character type.
- 7. Compile-time checking of constructor invocations for smart COM pointers. Without this feature, it is difficult for a developer to interpret compiler error messages caused by using the wrong smart COM pointer constructor.
- 8. The PFK code avoids defining macros. Instead templates, inline functions, typedefs, and other C++ constructs maintain safe design practices, with no runtime penalty.

There's also a Test Plugin available using the Plugin Framework.

-

### LGPL License GNU Lesser General Public License

### GNU LESSER GENERAL PUBLIC LICENSE Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

- 2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
  - a) The modified work must itself be a software library.
  - b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
  - c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
  - d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must

be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
- b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

- 7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:
  - a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
  - b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.
- 8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
- 9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.
- 10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.
- 11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot

distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

- 12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
- 13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

### END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!



### Nuvola Icons License of the 'Nuvola' icon theme by David Vignoni.

Usage of the 'Nuvola' icons is allowed under the terms of the LGPL license (which you can find here: GNU Lesser General Public License), plus an addition. Here's the original distribution notice:

```
This copyright and license notice covers the images in this directory.
Note the license notice contains an add-on.
TITLE: NUVOLA ICON THEME for KDE 3.x
AUTHOR: David Vignoni | ICON KING
       http://www.icon-king.com
MAILING LIST: http://mail.icon-king.com/mailman/listinfo/nuvola_icon-king.com
Copyright (c) 2003-2004 David Vignoni.
This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation,
version 2.1 of the License.
This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.
You should have received a copy of the GNU Lesser General Public
License along with this library (see the the license.txt file); if not, write to the
Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA #######**** NOTE THIS ADD-ON ****#######
The GNU Lesser General Public License or LGPL is written for software libraries
in the first place. The LGPL has to be considered valid for this artwork
library too.
Nuvola icon theme for KDE 3.x is a special kind of software library, it is an
artwork library, it's elements can be used in a Graphical User Interface, or
Source code, for this library means:
- raster png image*
The LGPL in some sections obliges you to make the files carry
notices. With images this is in some cases impossible or hardly usefull.
With this library a notice is placed at a prominent place in the directory
containing the elements. You may follow this practice.
The exception in section 6 of the GNU Lesser General Public License covers
the use of elements of this art library in a GUI.
dave [at] icon-king.com
Date:
           15 october 2004
           1.0
Version:
DESCRIPTION:
Icon theme for KDE 3.x.
Icons where designed using Adobe Illustrator, and then exported to PNG format.
Icons shadows and minor corrections were done using Adobe Photoshop.
Kiconedit was used to correct some 16x16 and 22x22 icons.
Released under GNU Lesser General Public License (LGPL)
```

Look at the license.txt file.

CONTACT

David Vignoni

e-mail : ICQ : david [at] icon-king.com 117761009

http: http://www.icon-king.com



## Übersetzungen installieren Wie man Übersetzungen für KeePass 1.x installiert.

- Übersetzung der Benutzeroberfläche installieren
- Zusätzliches übersetztes Material

### Übersetzung der Benutzeroberfläche installieren

Die deutsche Übersetzung der Benutzeroberfläche lässt sich wie folgt installieren:

- 1. Herunterladen der Übersetzungsdatei von Translations-Seite.
- 2. Entpacken des heruntergeladenen ZIP-Archivs.
- 3. Kopieren der Datei **German.1ng** in das KeePass-Anwendungsverzeichnis (in dem sich die Datei **KeePass.exe** befindet).
- 4. Starten von KeePass. Im Hauptmenü 'View' 'Change Language...' auswählen. In dem Dialog, der sich daraufhin öffnet, **German** auswählen. Den Dialog zum Neustarten von KeePass mit [Ja] bestätigen.

Die komplette Benutzeroberfläche ist nun Deutsch.

## Zusätzliches übersetztes Material Zusätzliches deutsches Material finden Sie auf der Translations-Seite.