Converting for 1Password

The **convert_to_1p4** utility converts exported data from various password management programs into a format that can be imported by 1Password, or in the case of 1Password data, suitable for printing. The table below lists the currently supported password management programs, along with the converter's name, and the purpose of the converter.

Supported Converters

Password Manager	Converter Name	Purpose	Minimal Version or Version Tested
1Password	onepif2html	printing	1Password 4
Clipperz	clipperz	importing	the clipperz web program does not have typical version numbers; tested against clipperz as of Sept 2014
CSV (generic)	csv	importing	OS X
Data Guardian	dataguardian	importing	OS X: 3.2; Windows: 3.2
DataVault	datavault	importing	OS X: 5.2.39; Windows: 5.1.32
EssentialPIM	essentialpim	importing	Windows: 6.04
eWallet	ewallet	importing	OS X: 7.3; Windows: 7.6.4
Handy Safe	handysafe	importing	OS X: 1.02; Windows: Handy Safe Desktop Professional 3.01
IronKey Identity Manager	ironkeyim	importing	Windows: firmware 4.0.5, software 3.4.3.2 (export to XML is available in software 2.5.1.0 onward)
Keepass 2	keepass2	importing	Windows: 2.26
KeepassX	keepassx	importing	OS X: .0.4.3 (version 2.0 alpha does not support exporting - use KeePass 2.x to read a KeePassX 2.0 database and export it as XML)
Keychain, OS X	keychain	importing	OS X 10.9.5, 10.10
LastPass	lastpass	importing	Browser-dependent extension; various versions exist (version 3.1.54 Firefox/Windows tested)
LicenseKeeper	licensekeeper	importing	OS X: 1.8.4 (1702)
mSecure	msecure	importing	OS X: 3.5.4 (previous versions do not properly export CSV, but the converter compensates); Windows: 3.5.4 (does not properly export CSV, has quoting / escaping issues)
Norton Identity Safe	nortonis	importing	Windows: tested with 2014.7.11.42; OS X: untested (no trial version available - should work - please report your results)
Passpack	passpack	importing	Web version 7.7.14 tested on OS X and Windows
Password Agent	passwordagent	importing	Windows: 2.6.3
Password Depot	passworddepot	importing	Windows: 8.1.1
PasswordWallet	passwordwallet	importing	OS X: 4.8.1 (6095); Windows: 4.8.1 (1147)
Safe in Cloud	safeincloud	importing	OS X: 1.6; Windows: 2.8
SafeWallet	safewallet	importing	OS X 1.2; Windows: 2.4.1.2, 3.0.7
SPB Wallet	spbwallet	importing	Windows: 2.1.2 (12118)
SplashID	splashid	importing	OS X: 7.2.2; Windows: 7.2.4
vCard, OS X Contacts	vcard	importing	OS X: 10.10.3 tested, vCard version 3.0

Instructions

The following instructions will guide you through the specific steps required to export your password manager's data, convert it, and import it into 1Password (or in the case of a 1Password export conversion, to print the data). Instructions specific to either OS X or Windows will be noted below. In addition, password manager-specific instructions follow in a separate section later.

Note: The conversion utility was written using a scripting language, and the source code is readable by anyone. This was done so that you and others can examine the source code to feel confident that your private data is not transmitted, or used in any unintended way. Yyou are free and welcome to examine it and ask any questions should you have any concerns.

Tip: Please don't let these instructions or the process intimidate you - the instructions are lengthy to be thorough, complete, and hand-holding. In most cases the export, conversion, and import will take just a few minutes.

1. Verify Requirements

Be sure you are running at least the required version of 1Password as mentioned in the requirements table below. If you are using an earlier version, you will need to update 1Password to properly import your data.

Platform	Required versions and software
OS X	1Password for Mac, version 4.0 or higher
Windows	1Password for Windows, version 4.1.0.520 or higher

The **Supported Converters** table above indicates the minimal software versions required or tested, and the platform used for export and conversion. It is not possible (nor would it be worthwhile) to test all versions on all the platform variations. Minor version differences generally should not matter (unless noted), so you can try the export and conversion.

The converter requires that both the export from your password manager and the conversion of that data are performed on the same OS platform. Example, if you export your password manager's data on Windows, then run the converter on Windows.

The converter (witten in the Perl scripting language) runs in a command shell using either the OS X **Terminal** application or the Windows command shell **cmd.exe**.

On OS X, an AppleScript is available which simplifies converting your exported data via drag-and-drop, eliminated several steps below.

On Windows, a Perl interpreter is required, and you will need the free Strawberry Perl, version 5.16.3.1 (*not* 5.18 or later). Download the **portable** version of Strawberry Perl, selecting the appropriate 32-bit or 64-bit version. When you are done with the conversion, you may delete the extracted portable version of Strawberry Perl and its zip file. Some additional modules are required, and installation instructions are noted in the sub-steps of Step 3 below.

2. Export Your Data

See your password manager's specific section under **Exporting from your Password Manager** below, and proceed to Step 3 when the data is exported. Note that some password managers support attachments and may or may not support exporting these attachments. Regardless, **convert_to_1p4** does *not* support importing attachments into 1Password.

3. Prepare for Conversion

On OS X, you are encouraged to use the convenient drag-and-drop AppleScript named *AppleScript_Conversion_Helper* to simplify the conversion. Skip to Step 4 to use it, or continue following the instructions to perform manual conversion.

Note: The instructions that follow assume you have placed the **convert_to_1p4** folder complete with its contents onto your Desktop.

To perform the conversion manually on OS X, open **Terminal** (under `Applications > Utilities`, or type **Terminal.app** in Spotlight and select it under Applications). When a Terminal window opens, type (or better still, copy and paste) the command:

```
cd Desktop/convert_to_1p4
```

and hit Enter.

On Windows, perform the following steps:

- **3a.** unzip the Strawberry Perl archive and place the unzipped archive as the folder: `C:\myperl` (right click the archive, Extract All, and enter `C:\myperl` as the Destination path). When the unzip has completed, you can delete the zip file.
- **3b.** Enter the following command into Window's **Search programs and files** box `C:\myperl\portableshell.bat` and hit Enter. This will open a command window with the required PATH variables set.

The following one-time commands add required modules to the Strawberry Perl `C:\myperl` area, and will be entered into the just-opened command line window (copy and paste is the best method - to paste, right-click in the command line window). You can skip to step 3f if you've already installed the modules successfully. Be sure a command completes successfully (the last output from the command will end with `-- OK`).

- **3c.** Enter the command `cpan Text::CSV` and hit Enter.
- **3d.** Enter the command `cpan Date::Calc` and hit Enter.
- **3e.** Enter the command `cpan XML::XPath` and hit Enter.

The following command changes the working directory to the **convert_to_1p4** directory you placed on your Desktop:

- **3f.** Enter the command `cd %USERPROFILE%\Desktop\convert_to_1p4` and hit Enter.

4. Perform the Conversion

You are about to convert your data. The utility does not modify your password manager's data, nor does it transmit any data across a network.

On OS X, simply drag the exported data file you created in Step 2 *onto* the *AppleScript_Conversion_Helper* file and follow the on-screen instructions. If you get the OS X error message "*App can't be opened because it is from an unidentified developer*", click `OK`, and then right-click and `Open` the *AppleScript_Conversion_Helper*. The automatically-opened Terminal window will indicate the status of the conversion - you can `Quit` the Terminal application if the conversion was successful. Proceed to Step 5 when the conversion is complete. If you prefer not to use the *AppleScript_Conversion_Helper*, continue with the following steps.

Find the name of your converter from the list of **Supported Converters** at the top of this guide. You'll use that converter name in the command below, where you see *name_of_converter*. For example, if you were converting data exported from Password Depot, *name_of_converter* would be *passworddepot*.

Note: The list of supported converters can also be output by using the `--help` option, for example, the command:

```
perl convert_to_1p4.pl --help
```

outputs:

```
Usage: convert_to_1p4.pl <converter> <options> <export_text_file>

converters:
    clipperz dataguardian datavault essentialpim ewallet handysafe ironkeyim keepass2
    keepassx keychain lastpass licensekeeper msecure nortonis onepif2html passpack
    passwordagent passworddepot passwordwallet safeincloud safewallet spbwallet splashid
    vcard

specify one of the converters above on the command line to see complete options
```

Now, let's enter the command that performs the conversion. In the Terminal window (OS X) or the command window (Windows), enter the appropriate command below for your OS version, replacing *name_of_converter* with the relevant converter name for your password manager, and replacing *name_of_export_file* with the name of your password manager's export file:

- **OS X**

  ```
  perl5.16 convert_to_1p4.pl name_of_converter -v ../name_of_export_file
  ```

- **Windows**

```
perl convert_to_1p4.pl name_of_converter -v ..\name_of_export_file
```

Hit `Enter` after you've entered the correct command. See **Note** below.

The `-v` option (verbose) tells the script to state the number of records imported from the password manager's export file, and exported (per category) to the final output file.

Again, the command line above assumes that your Desktop contains:

- the text file exported from your password manager
- the folder `convert_to_1p4`

and the `convert_to_1p4` folder contains the `convert_to_1p4.pl` script and all of its accompanying modules. It also assumes you've replaced the generic placeholder terms *name_of_converter* with the specific converter you need to do the conversion, and *name_of_export_file* with the name of your password manager's export file.

5. Import into 1Password or Print

If the conversion was successful, there will be a file named **1P_import.1pif** on your Desktop, or if you used the **onepif2html** converter, a file named **1P_print.html**.

- Importing

  To Import the **1P_import.1pif** file into 1Password, use 1Password's `File > Import` menu and select the file **1P_import.1pif**.

  If the 1PIF import is successful, all of your password manager's records will now be available in 1Password. These records may require some clean-up, as some or your password manager's fields may not safely map into some 1Password fields, or the data may be problematic (certain ambiguous date fields, for example). Any unmapped fields will be pushed to an entry's Notes area, so the data will be available for you within the 1Password entry. In addition, a single entry from your password manager may convert into multiple 1Password entries. The converter tries to place data in the right place inside of 1Password, and when it is unable to, the notes section of an entry or the Secure Notes category become the catch-all locations. See **Additional Notes** below for more information.
- Printing

  To Print the file named **1P_print.html**, open and view the file with your favorite browser, and use its Print command. You may want to configure various print options before printing out the document.

6. Securely Remove Exported Data

As soon as you have completed the import into 1Password, or printing of your exported data, be sure to securely delete the exported file(s) you created in Step 2, as well as the file created by the converter script in Step 4 (e.g. 1P_import.1pif, 1P_print.html), since these contain your unencrypted data.

Miscellaneous Notes

Command Line Options

Usage help and several command line options are available to influence the behavior of the conversion script.

Option: `--help`

For usage help, enter the command:

```
perl convert_to_1p4.pl --help
```

More specific help is also available when you've specified a converter on the command line. For example:

```
perl convert_to_1p4.pl ewallet --help
```

would result in the output:

```
$ perl convert_to_1p4.pl ewallet --help

Usage: convert_to_1p4.pl <converter> <options> <export_text_file>

converters:
    clipperz dataguardian datavault essentialpim ewallet handysafe ironkeyim keepass2
    keepassx keychain lastpass licensekeeper msecure nortonis onepif2html passpack
    passwordagent passworddepot passwordwallet safeincloud safewallet spbwallet splashid
    vcard

options:
    -d or --debug              # enable debug output
    -e or --exptypes <list>    # comma separated list of one or more export types from list below
    -f or --folders            # create and assign items to folders
    -h or --help               # output help and usage text
    -i or --imptypes <list>    # comma separated list of one or more import types from list below
    -o or --outfile <ofile>    # use file named ofile.1pif as the output file
    -v or --verbose            # output operations more verbosely
          --nowatchtower       # do not set creation date for logins to trigger Watchtower checks

supported import types:
    bankacct callingcard carinfo cellphone clothes combolock contact contactlens creditcard
    driverslicense email emergency general health idcard insurance internet lens librarycard
    membership note passport password prescription serialnum socialsecurity software
    voicemail votercard website
supported export types:
    bankacct creditcard driverslicense email login membership note passport server
    socialsecurity software
```

Options: `--imptypes` and `--exptypes`

By default, all exported entries will be processed and converted to types that 1Password can import. The options `--imptypes` and `--exptypes` allow you to selectively choose which entry types to process on import and which entry types to export to the 1Password 1PIF file, respectively.

For example, if you only wanted eWallet's types `bankacct` and `votercard` converted, the option `--imptypes bankacct,votercard` would be added to the command line. And if you only wanted `note` types to be exported to the 1Password 1PIF file, then the option `--exptypes note` would be added. This may result in more entries than you expect, as some password manager entry types will be split into two or more different 1Password entry types (aka Categories).

For example, a single LastPass entry of type *Identity* may be split into four separate 1Password entries, one each the types *Identity*, *Login*, *Secure Note*, and *Social Security*. The list of supported types is available via the `--help` option when a converter is specified on the command line (see the example `--help` output above). Take care when using `--exptypes` with types that split.

Option: `--folders`

The `--folders` option supports the creation of Folders in 1Password, and places your records into the same folder hierarchy as supported in your password manager. This feature is disabled by default, because the converter is unaware of existing folders in your vault. If you use this option, all Folder names existing in the vault are ignored, and the converter will create new Folders, possibly with names identical to those already in your vault. In addition, re-running the converter and re-importing will duplicate the Folder names, since new unique folder identifiers are created each time the converter is run. For best results, import converted data only into a fresh vault.

Option: `--modified`

Some password managers will export a date of modification, and this date can be used to set 1Password's `last modified` field for an entry. For converters that support this option, you may include the `--modified` option on the command line (it is disabled by default) . See the output of the `--help` option; converters that support the `--modified` option will show the option in the `--help` output.

Note: This feature requires at least 1Password for Mac 5.3.BETA-17.

Note: The modified date used by various password managers may reflect the date the record was last updated, or the date the password field was last updated. The converters can only set the `last modified` value for the entire 1Password entry, and not for any specific field within the entry.

Option: `--nowatchtower`

This converter sets the Created date for each *Login* item to 1/1/2000. This sufficiently old date in the past allows 1Password's Watchtower service the ability to flag potentially vulnerabile sites (i.e. for the HeartBleed security issue). Unfortunately, there is no way for the converter, and hence 1Password, to know if you had already changed your password for a given site. This converter errors on the side of allowing 1Password's Watchtower service the ability to at least warn you of a site's previous vulnerability so that you can act accordingly. If you had already changed all of the potentially vulnerable passwords for your logins, you can include the `--nowatchtower` option on the command line to cause the converter to not set the record's Created time, and the Watchtower vulnerability banner will not be shown in a *Login* entry.

If you have not recently changed the passwords for your imported items, AgileBits recommends visiting the Watchtower page and entering the URLs, one at a time, to check for vulnerabilities.

Source Files and Folders

The main converter script convert_to_1p4.pl and each of the converter modules use the shared, common code modules **PIF.pm** and **Utils.pm** stored in the **Utils** folder.

The **Converters** folder contains the individual converter modules.

The folders **JSON**, **Text** and **UUID** contain code modules used by the conversion script and modules. These are included for your convenience in case they are not installed on your system. These modules are commonly used or bundled Perl modules, and are available on CPAN.

Alternate Download Locations

This script package and its updates are available on the 1Password Discussions forum and from other download locations. AgileBits reviews the code posted on the GitHub repository referenced in AgileBit's guide Import your data, and therefore recommends only downloading from that site.

Exporting from your Password Manager

Reminder: these instructions assume that you take note of the name you use to create your export data file, and use that name in the commands above in Step 4, and that you place that file onto your Desktop. On Windows, file extensions for common file types are hidden by default, so be sure to check the file name ultimately created after exporting the data. You will need to adjust the commands above to reflect the actual name of the export file created, specifically the *name_of_export_file* file in the conversion command. Some suggested file names will be provided, but you are free to choose the export file's name.

Find the section below for your password manager, and follow the instructions. Proceed to Step 4 above when the export is complete.

• 1Password (for printing)

The **onepif2html** converter is used for printing out your 1Password entries. Launch 1Password, and export its database as a 1PIF export file using the `File > Export > All Items...` menu item (or chose the `Selected Items` sub-menu if you only want the selected items to be exported). Enter your Master Password when requested, and click `OK`. Navigate to your **Desktop** folder in the *Export* dialog, and in the *File name* area, enter the name **UNENCRYPTED_DATA**. Set the File Format to `1Password Interchange Format (.1pif)` if it is not already selected. Click `Save`, and you should now have your data exported as as a 1PIF file by the name above on your Desktop. You may now quit 1Password.

Note: After export, 1Password for Mac places the 1PIF file into a folder with the name placed in the *File name* area - it will open this folder after the export has completed. Inside will be a file named **data.1pif**. This is the data file you will be working with. You might want to drag this to your Desktop for convenience.

Note: The formatting of the data in the resulting html file is controlled by a combination of an XSLT XML stylesheet and CSS. This data is

embedded at the end of the **onepif2html** converter, and can be customized to suit your needs.

• Clipperz

Launch and log into the clipperz web interface, and export its database to the JSON format. Select the **data** tab, in the upper right of the web interface, select `Export` from the left sidebar, and then click the `Export to JSON` link. This will open a new page, containing your data. Select and Copy all of the text in the box.

On OS X, open TextEdit and paste the text into the new document. Save the file with the name **pm_export.txt** (the `Save As` pulldown) and select your Desktop as the destination (the `Where` pulldown). Select `Unicode (UTF-8)` from the `Plain Text Encoding` pulldown.

On Windows, open Notepad, and paste the text into the new document. Save the file with the name **pm_export.txt** (`File > Save As`), select the Desktop folder as the destination, and in the `Encoding` pulldown, select `UTF-8`.

You should close the JSON export Clipperz web page, since it contains your unencrypted data. Also, since your data is on the clipboard, you should copy some other text to clear the clipboard, and remove any clipboard manager entries if you have a clipboard manager program installed. It is possible that your browser has also cached the exported data. You may now close the main Clipperz page.

The Clipperz converter currently supports only English field names.

• CSV

This is a generic CSV converter which currently handles Logins. Construct the CSV in a spreadsheet program - it must have the following columns: Title, Username, Password, URL, Notes. Ensure the first row is a column header with those names (case does not matter). Additional columns may follow, and their titles will be used to create custom fields in the entry, and the corresponding values will be stored in these fields.

• Data Guardian

Launch Data Guardian and export its database to a CSV text file using the `Record > Export...` menu item. From the left side bar, select **Text**. Under the list of fields that can be exported, click the `Check All` button. In the **Field Delimiter** pulldown, select the *Comma (CSV)* choice. Under **Options**, select the *Include header row* checkbox. Click the `Export` button, and in the *Export Database* dialog, navigate to your **Desktop** folder, and save the file with the name **pm_export.txt** to your Desktop. When the dialog appears asking what suffix to use, select the `use .txt` button. You may now quit Data Guardian.

Note: Data Guardian does not construct a reliable CSV file. In certain cases, it is impossible to determine one field from the next field (the field's value will contain one or more commas and/or double quotes). When this occurs, the converter will skip the record, and report an error indicating the name of the record that could not be converter.

• DataVault

Launch DataVault and export its database to a text file using the `Tools > Export` menu item. Select the **All Items (DVX, CSV)** choice, and click `OK`. In the *Save As* dialog, navigate to your **Desktop** folder, and save the file with the name **pm_export.csv** to your Desktop, leaving the **Save as type** set to *CSV files (.csv files)*. You may now quit DataVault.

Note: DataVault has several essentially identical templates, which are indistinguishable in the CSV export. These are treated as a single DataVault template, and are mapped into 1Password categories. These are:

- bank account, checking account –> bankacct
- business contact, personal contact –> contact
- credit card, mastercard, visa –> creditcard
- business, financial –> business

• EssentialPIM

Launch EssentialPIM and export its password database to a text file using the `File > Export > Password Entries > Comma Separated Values (*.csv)...` menu. Select All entries from the *Entries to be exported* dialog. You may optionally select the fields you want exported as well by selecting the `Fields...` button (but you should leave selected the fields that correspond to the stock fields: Title, User Name, Password, URL, and Notes). Click the `OK` button, and navigate to your **Desktop** folder, and save the file with the name **pm_export.txt** to your Desktop. You may now quit EssentialPIM.

Note: EssentialPIM does not properly handle the exported field names if the names contain any comma characters. Before you export, edit a single password entry record in EssentialPIM, and examine each of your field names. Replace any commas in the field names some other character. Editing the field names inside a single record will globally change the field names for all records. Once the commas are removed from the field names, you may now export your data.

• eWallet

Launch eWallet and export its database to a text file using the `File > Save As > Text File...` menu. Save the file with the name **pm_export.txt** to your Desktop. You may now quit eWallet.

The eWallet type of Picture Card will be exported as Secure Notes; no pictures will be exported.

Note: eWallet's export format is ambiguous and not well-defined. The converter attempts to determine proper record boundaries, but can fail if a card's notes contain a certain pattern. The pattern is a blank line line followed by the word **Card** followed by a space and then anything else. For example, notes containing the following text would confound the converter's ability to detect the card's boundaries:

```
My good notes.

Card anything

Final thoughts - that "Card " pattern above will cause a problem.
```

If the conversion indicates the same number of records as contained in your eWallet, this problem did not occur. If you have extra records, the problem may have occurred, and you should examine the data imported into 1Password. To remedy the problem, be sure no notes in eWallet contain the pattern {blank line}Card{space}{anything else} - it is sufficient to simply lowercase the word Card, or add some other character in front of the C, such as **x**Card.

• Handy Safe

OS X: Launch Handy Safe and select the HANDY SAFE (topmost) grouping in the Handy Safe left sidebar. To export the database as an XML export file, select the `File > Export` menu item.

Navigate to your **Desktop** folder in the Export File dialog, and in the *File name* area, enter the name **pm_export.txt**. Click `Save`, and you should now have your data exported as an XML file by the name above on your Desktop. You may now quit Handy Safe.

• IronKey Identity Manager

Insert and unlock your IronKey device. From the IronKey Control Panel, launch Identity Manager. From the `Identity Manager` menu, select the `Backup > Export as Text or XML` item. Click `OK` when export warning dialog appears (e.g. *The information in exported file will not be encrypted. Would you like to export your database to a file?*). Navigate to your **Desktop** folder, and save the file with the name **pm_export.xml** to your Desktop. Ensure that **Xml files** is set as the *Save as type:* and click `Save`. You should receive a final cautionary dialog indicating that the database has been successfully exported and that its contents are unencrypted. Click `OK` to continue. You may now close Identity Manager.

• KeePass 2

Launch KeePass 2, and export its database to an XML export file using the `File > Export ...` menu item, and select the KeePass XML (2.x) format. In the `File: Export to:` section at the bottom of the dialog, click the floppy disk icon to select the location. Select your **Desktop** folder, and in the *File name* area, enter the name **pm_export.txt**. Click `Save`, and you should now have your data exported as an XML file by the name above on your Desktop. You may now quit KeePass 2.

The converter will decode and convert an entry's attachments. They are placed in a folder named **1P4_Attachments** in the same location that the **1P4_import.1pif** file will be created. An entry's attachments are placed in a sub-directory named with the entry's Title.

• KeePassX

Launch KeePassX, and export its database to a text file using the `File > Export to > KeePassX XML File...` menu. Navigate to your **Desktop** folder, and save the file with the name **pm_export.txt** to your Desktop. You may now quit KeePassX.

• Keychain

The **Keychain Access** app allows you to view your OS X keychains. It will show both local OS X keychains and the iCloud Keychain. However, **Keychain Access** does does not support exporting the contents of keychains; instead, the OS X command line `security` tool provides this capability. To further complicate the issue, the `security` tool only supports dumping entries from *local* keychains; iCloud keychains require copying entries to a new local keychain.

When run, the `security` tool will present many confirmation dialogs, one confirmation dialog per item in your Keychain. So you will have to press `Allow` many times, or you can use an AppleScript to help automate this. To see how many entries you have, lanuch `Applications > Utilties > Keychain Access`, and select the **login** item under **Keychains**, and **All Items** under **Categories**. The list of items that will be exported (and hence require confirmation) is in the area at the right.

To create an AppleScript to perform the Allow action for you, launch `Applications > Utilities > Script Editor`. Copy the AppleScript code below, and paste it into the upper area of the open script edit entry area.

```
tell application "System Events"
    repeat while exists (processes where name is "SecurityAgent")
        tell process "SecurityAgent"
            click button "Allow" of window 1
        end tell
        delay 0.2
    end repeat
end tell
```

Now, open `System Preferences > Security & Privacy`, and select the **Accessibility** section. Unlock the System Preferences dialog (if it is locked) by clicking the lock icon in the lower left of the dialog and entering your system password. Once unlocked, click the add (plus) button below the section named *Allow the apps below to control your computer*, and add `Applications > Utilities > Script Editor` to the list of applications that may control your computer (you can disable this later).

To export the OS X Keychain, copy the command below and paste it into the Terminal app:

```
security dump-keychain -d login.keychain > ~/Desktop/pm_export.txt
```

and press `Enter`. When OS X prompts you to allow the action - repeatedly press `Allow`, or if you've created the AppleScript mentioned above and have `Script Editor` open with the script entered, press the run (right-facing triangle) button. The AppleScript will run until there are no more entries, and your keychain data should now be on your Desktop in the file named **pm_export.txt**.

To export your iCloud Keychain, you need to create a new local keychain and copy the contents from the iCloud Keychain. Launch the `Keychain Access` app, and select the `File > New Keychain` menu item. Name the keychain **local-icloud**, and click the `Create` button. Enter a short, easy to type password and click `OK`. Under the **Keychains** section, select the **iCloud** keychain. Now select all of the items in that keychain on the right, and copy them with ⌘-C or `Edit > Copy`. Now select the **local-icloud** keychain, and paste the entries with ⌘-V or `Edit > Paste`. You will be presented with many authorization dialogs. Enter your passwords as requested, or open a new script document in the **Script Editor**, and copy/paste the following script as above:

```
tell application "System Events"
    repeat while exists (processes where name is "SecurityAgent")
        tell process "SecurityAgent"
            set value of text field 1 of window 1 to "MYPASSWORD"
            click button "OK" of window 1
        end tell
        delay 0.2
    end repeat
end tell
```

replacing the word **MYPASSWORD** with the password you just set for the keychain. Run the AppleScript to automate performing the password entry.

. To export this local copy of your iCloud Keychain, copy the command below and paste it into the Terminal app:

```
security dump-keychain -d local-icloud > ~/Desktop/pm_export-icloud.txt
```

and press `Enter`. As above, repeatedly press `Allow`, or use the AppleScript method to automate the tedious process. Once the export is complete, you may delete the **local-icloud** keychain in `Keychain Access` if you wish (delete both References and Files).

You may now quit **Keychain Access**, **Script Editor**, and **System Preferences** if you wish.

You will now have one or two exported files - convert and import each of these into 1Password independently, one at a time.

Note: Copying items from an **iCloud Keychain** to a local keychain sets the modified and created dates of the pasted entries to the time of the paste. The original dates are not retained in the copies.

• LastPass

Launch the browser you normally use with LastPass. From the LastPass browser extension, select `Tools > Advanced Tools > Export To > LastPass CSV File`, and when prompted, enter your LastPass vault password. Navigate to your **Desktop** folder in the *Select a file to export to* dialog, and in the *File name* area, enter the name **pm_export.txt**, and click `Save`.

Note: In some cases, the LastPass exported data will open in a separate browser window, showing the contents of your LastPass data. If this happens, select all of the data on the page (Cmd-A) and Copy it (Cmd-C). Open TextEdit, and go to the menu `TextEdit > Preferences`. In the New Document tab, under Format, select `Plain Text` and close that dialog. Open a new document (Cmd-N). Paste your data now (Cmd-V), and save it to your Desktop with the file name `pm_export.txt` and selecting `Unicode (UTF-8)` as the Plain Text Encoding.

Note: The LastPass extension for Chrome incorrectly opens the export as an HTML document. If you are using Chrome as your browser, do not use the extension to start the export. Instead, either use the LastPass extension in another browser, or use the `Tools > Advanced > Export` item in the left sidebar when your vault is open in the browser (you will have to use the copy / paste method mentioned in the note above).

Note: LastPass exports Form Fill Profiles separately, using the `Tools > Advanced Tools > Export To > Form Fill Profiles` menu. If you want to convert and import your Form Fill Profiles, export these to a separate export file, for example, **pm_export_ff.txt**. When you have completed converting and importing your primary data, repeat from Step 4 onward, this time converting the file **pm_export_ff.txt** and and importing the resulting .1pif file.

• LicenseKeeper

Launch LicenseKeeper, and export its database as an XML export file using the `File > Export > Library to XML...` menu item. When the folder save dialog appears, select **Desktop** from the sidebar, and then click the `Choose Export Folder` button. This will store the XML file inside a folder named **LicenseKeeper Export**. The file named **LicenseKeeper.xml** inside that folder is your exported XML data. After the export has completed, you may quit LicenseKeeper.

The converter will rename the exported, name-encoded attachments. They will be prefixed with the record's Title, and can be found in the **Attachments** folder within the **LicenseKeeper Export** folder.

• mSecure

Launch mSecure, and export its database as a CSV export file using the `File > Export > CSV File...` menu item. Click `OK` to the warning dialog that advises caution since your data will be exported in an unencrypted format. Navigate to your **Desktop** folder in the *Export File* dialog, and in the *File name* area, enter the name **pm_export.txt**. Click `Save`, and you should now have your data exported as an CSV file by the name above on your Desktop. You may now quit mSecure.

Note: mSecure does not output the labels for custom fields. Before you uninstall mSecure, you should verify that you understand the meaning of your imported data. Take a screenshot or write down the field names and orders of your customized fields and cards, and compare these against the data imported into 1Password.

Note: mSecure on Windows 3.5.4 (and probably earlier) does not properly export CSV data. There are issues with the characters backslash \ and double-quotes “. There may be additional issues due to the incorrect and ambiguous handling of these escape and quoting characters. Before you uninstall mSecure, you should verify your data against the data imported into 1Password.

• Norton Identity Safe

Launch Norton Identity Safe, and export its database as a CSV export file using the Settings (gear) icon, and selecting the `Import/Export` tab. Select the *Plain Text - CSV file (Logins and Notes only)* radio button, and click `Export`. Enter your vault password when the *Password Protected Item* dialog appears, and click `OK`. Navigate to your **Desktop** folder in the *Save As* dialog, and in the *File name* area, enter the name **pm_export.csv**, and click `Save`. Click `OK` when the successful export dialog appears. You may now quit Norton Identity Safe.

Note: Norton Identity Safe does not export Wallet items (Credit Card, Bank Account, or Identity) - it only exports Login and Note items. Also, it does not export Tags.

• Passpack

Launch the browser you normally use with Passpack and unlock your vault. From the Passpack toolbar menu, select `Tools` and then on the next screen, select `Export`. Select the option `Comma Separated Values`, and the other option `All entries`. Now select the columns to export under *Click on the name of the first field to export* - select these one at a time in the same order as presented on the screen: **Name**, **User ID**, **Password**, **Link**, **Tags**, **Notes** and **Email**. Now click the `Continue` button. A new window will appear with your exported data. Select all the text data in the text box and copy it. You will save it with either Notepad (on Windows) or TextEdit (on OS X) as follows:

On Windows, create and open a new text document by right-clicking the Desktop and selecting New > Text Document. Name the document `pm_export.txt`. Right-click that document, select `Edit`, and Paste your copied data (Ctrl-V). Select Notepad’s `File > Save As...` menu, set the *Encoding* to **UTF-8**, and click `Save` to save the document.

On OS X, open TextEdit, and go to the menu `TextEdit > Preferences`. In the New Document tab, under Format, select `Plain Text` and close that dialog. Open a new document (Cmd-N). Paste your copied data (Cmd-V), and save the document to your Desktop with the file name `pm_export.txt`, selecting `Unicode (UTF-8)` as the Plain Text Encoding.

• Password Agent

Launch Password Agent, and export its database as an XML file using the `File > Print & Export` menu item. When the *Print & Export Wizard* dialog opens, click the **all** selector above the *Groups* listing, and click the **all** selector above the *Fields* listing. Under the *Output format*, select the **XML - export database** option. You can ignore the *Sort by* pulldown. Click `Next` to continue. When the the *Save As* dialog appears, click the `Browse` button, navigate to your Destkop, and save the file to your Desktop with the name **pm_export.xml**. When the export has completed, close the wizard. You may now quit Password Agent.

• Password Depot

Launch Password Depot. On the left side, select your password file’s name (or the highest level folder you want to export). Select the `Tools` ribbon menu item, click the `Export` button, and select the `Export Wizard` item. Enter your Password Depot password when the dialog appears, and press `OK`. The *Export format* should be left as *Extensible Markup Language format (**.xml)*. Click the `Browse` button, navigate to your **Desktop** folder and in the *File name* area, enter the name **pm_export.xml**. Be sure the *Original folder* pulldown has the top-level folder you want exported (the top level is the entire contents of the password file). Click `Next` to complete the export and then click `Finish` to close out the wizard. You should now have your data exported as an XML text file by the name above on your Desktop. You may now quit Password Depot.

Note: Password Depot supports formatted text in Information items. It does not, however, export this formatting information - only the raw text is exported.

• PasswordWallet

Launch PasswordWallet, and export its database as a text file using the `File > Export > Visible entries to text file...` menu. Enter the wallet's password when the password dialog appears. Navigate to your **Desktop** folder, and save the file with the name **pm_export.txt** to your Desktop. You may now quit PasswordWallet.

• Safe in Cloud

Launch Safe in Cloud, and export its database to an XML file using the `File > Export > As XML` menu. Navigate to your **Desktop** folder, and save the file with the name **pm_export.txt** to your Desktop. You may now quit Safe in Cloud.

• SafeWallet

Launch SafeWallet, and export its database to an XML file. On OS X, use the `File > Export...` menu. Select the XML tab, and click `Export`. On Windows, use the `File > Export Wallet` menu, select the `Export to XML file` button, and click Next. Click the `Browse button`.

Navigate to your **Desktop** folder, and save the file with the name **pm_export.txt** to your Desktop. On Windows, click the `Finish` button and press `OK` when the successful export dialog appears. You may now quit SafeWallet.

• SPB Wallet

There is no need to export any data from SPB Wallet. The converter will read and decrypt the data directly from the SPB Wallet .swl file. It might be easiest to move your .swl wallet file to your Desktop. When you enter your command line, use your .swl wallet file's name as *name_of_export_file*.

If would prefer not to move the .swl file, you'll have to specify a path to your wallet file in place of the relative path ..*name_of_your_spbwallet_file*.

• SplashID Safe

Launch SplashID Safe, and export its database to a vID file using the `File > Export > SplashID vID` menu (be sure not to choose SplashID vID3). Navigate to your **Desktop** folder, and in the `Save As` area, enter the file name **pm_export** (note: your file will be named **pm_export.vld** after exporting the data). At the bottom of the dialog, select `Export All Records` and deselect `Export Attachments`. Click `Save`. When the *Set Password* dialog appears, just click `OK` (do not enter a password). You may now quit SplashID Safe.

• vCard (from OS X Contacts)

Launch the OS X Contacts app, select the desired contacts, and export them to a vCard file using the `File > Export... > Export vCard...` menu. Navigate to your **Desktop** folder, and in the `Save As` area, enter the file name **pm_export** (note: your file will be named **pm_export.vcf** after exporting the data). Click `Save`. You may now quit Contacts.

Additional Notes

Customized Fields and Types

Many password management programs have limitations with exported data. Some do not export the meaning of fields, and use only simple, generic labels, often the same label across all entry types (making type detection difficult or impossible). And some do not output the field label at all. When no type information is present, converters attempt to determine the type of entry based on matching certain key field labels

against a table of stock, known entries. When no field labels exist, the specific number of fields may be used. And some converters may use other information present in the export file to hint at the entry type.

Some password management programs provide support for customized templates or for customizing the fields in an entry. Customized field names which are not recognized by the converter will be included in the standard *notes* section of the entry as Field:Value pairs. When no field labels exist, generic field labels such as **Field**_*n* (where *n* is 1, 2, …) are used. And customized entries which cannot be matched against the known types will converted into 1Password *Secure Notes*.

Customizing Converters

Each converter module employs a table of types, field definitions and special instructions or processing code. In many cases, the field names are regular expression patterns, which are used by the script for field recognition. These may be modified to accomodate special needs. New, custom types may be added to the table, and this may be useful if you have many entries based on custom types. In addition, your custom fields can be mapped to new 1Password label/fields within an existing section or a new one. Should you have such a need, please ask for more details on how this this can be accomplished.

Entry Splitting

Some password managers have entry types that more naturally work with 1Password when split into multiple, specific entries (i.e. 1Password Categories). For example, a single form fill profile entry from one password management program can result in four separate entries in 1Password (with the categories *Identity*, *Credit Card*, *Bank Account*, and *Social Security Number*). Converters place the field values in the appropriate locations, and place supplementary information such as extra or unmapped fields into the primary entry (*Identity* in this case).

Date Fields

Some password management programs provide no input validation for date fields, and the date values are ultimately just simple text fields, where it is possible, for example, to enter nonsense dates such as *Feb 31, 2015* or even the relative term *Tomorrow*. Because 1Password dates are more strict, date validation is performed by a converter where sensible, and when the date field can be arbitrary text, date fields are pushed to the *notes* section.

Some 1Password date fields are only stored as month / year values. When conversion of a date would result in loss of information (e.g. when the date value also contains a day component), a converter will both store the 1Password appropriate date, and also store the original value in an entry's *notes* section.

When an invalid date is detected, it is stored in the *notes* section.

Some password management programs export no century component to their date's year values. The converters employ a strategy, depending upon the particular date field, to use a sensible century value. For example, a *Valid From* date of 12/03 is almost guaranteed to be 12/2003 as opposed to 12/1903, and with today being Oct 9th, 2014, a birthdate of 11/14 cannot be 11/2014.

Groups, Tags, and Other Indicators

Some password management programs allow placement of items into groups or folders (flat or hierarchical). When available in the export, these will be recreated as colon-separated lists of 1Password Tags (note: these are not currently shown in the main UI of the Windows version of 1Password, however, the Tags do exist within the database file, and can be seen in the Tags section when editing an entry). With the `--folders` option, items will also be placed into the appropriately named Folder within 1Password.

Likewise, any tags, favorite or star markers, when available for an entry, will become 1Password tags.

Finally, other specific indicators may be placed into the 1Password entries notes section (e.g. Color: Red).

If the `--folders` option was not used, you can place a set of tagged 1Password entries into a 1Password folder by selecting a given tag and creating an appropriately named folder for that tag. With the Folder list expanded, and the tagged items selected, drag the items onto the folder name.

Questions, Help and Requests

If you have questions, issues, or requests, feel free to ask for assistance in the forum thread:
https://discussions.agilebits.com/discussion/30286/mrcs-convert-to-1password-utility/p1

-MrC