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.

Available Converters

Password Manager	Converter	macOS Version	Windows Version	Notes
1Password	onepif	1Password 4	1Password 4	1PIF conversion to HTML, CSV
Chrome	chrome	65	65	Tested with 65.0.3325.146 - earliest supported version unknown
Clipperz	clipperz	NA	NA	Web program does not have version numbers; tested against clipperz as of Sept 2014
CSV	csv	√		Generic CSV conversion for Logins, Credit Cards, Memberships
Data Guardian	dataguardian	3.2	3.2	
DataVault	datavault	5.2.39	5.1.32	
EssentialPIM	essentialpim		6.04	
Enpass	enpass	5.6.3	5.6.3	CSV export required
eWallet	ewallet	7.3	7.6.4	
F-Secure KEY	fsecurekey	4.0.108	4.0.109	
Handy Safe	handysafe	1.02	Desktop Professional 3.01	
IronKey Identity Manager	ironkeyim		firmware: 4.0.5, software: 3.4.3.2	Export to XML available in v2.5.1.0
Keepass 2	keepass2		2.26	
KeePassX	keepassx	.0.4.3*		* See note in KeePassX section 3 below
Keeper Desktop	keeper	8.3.3	8.3.3	
Keychain (macOS)	keychain	10.9.5, 10.10	NA	
Key Finder	keyfinder	1.2.0.32		
LastPass	lastpass	√	√	Various versions of browser extension, tested 3.1.54 Firefox/Windows
LicenseKeeper	licensekeeper	1.8.4 (1702)		
mSecure	msecure	3.5.4*	3.5.4**	* Earlier versions improperly export CSV - converter compensates ** Improperly exports CSV; quoting / escaping issues
Norton Identity Safe	nortonis	*	2014.7.11.42	* Trial version unavailable, should work - please report your results
Passpack	passpack	7.7.14	7.7.14	Web version tested
Password Agent	passwordagent		2.6.3	
Password Depot	passworddepot		8.1.1	
Passwords Plus	passwordsplus	3.0020		
Password Safe	passwordsafe	3.xx	3.38	
PasswordWallet	passwordwallet	4.8.1 (6095)	4.8.1 (1147)	
RoboForm	roboform	1.95	6.9.99	See note in RoboForm section 3 below
Safe in Cloud	safeincloud	1.6	2.8	
SafeWallet	safewallet	1.2	2.4.1.2, 3.0.7	
SPB Wallet	spbwallet		2.1.2 (12118)	
SplashID	splashid	7.2.2	7.2.4	
Sticky Password	stickypassword		8.0.6.145	
True Key	truekey	1.12.0		
vCard	vcard	10.10.3		Tested vCard version 3.0; macOS & iCloud Contacts
Text files	txt2notes	√	√	Text file(s) to Secure Notes converter
Wallet 4	wallet4	TBD	4.5.0.5095	Known as Wallet 4X, Wallet 4W
Yojimbo	yojimbo	4.04		

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 use it for other purposes. Instructions specific to either macOS or Windows will be noted below. Specific instruction explaining how to export your password manager data follow in the **Exporting from your Password Manager** section later - be sure to read the section for your password manager!

Note: The conversion utility was written using a readable scripting language. This was done so that you and others can examine and audit the source code to feel confident that your private data is not transmitted, or used in any unintended way. You 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
macOS	1Password for Mac, version 4.0 or higher
Windows	1Password for Windows, version 4.1.0.520 or higher

The **Available 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 generally 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 macOS **Terminal** application or the Windows command shell **cmd.exe**.

On macOS, an AppleScript-based GUI helper is available which simplifies converting your exported data, eliminating several steps below.

On Windows, a Perl interpreter is required, and you will need the free Strawberry Perl. Download the latest **portable** version of Strawberry Perl, selecting the appropriate 32-bit or 64-bit version. It will be named **64bit PortableZIP edition** or **32bit PortableZIP edition** (do not use the download with extra PDL libs). 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. This is important!

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; in cases where attachments are exported, some converters will export the attachments into separate **1P4_Attachments** folder.

3. Prepare for Conversion

On macOS, you are encouraged to use the convenient 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 macOS, 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`.

For converting on a Windows platform, perform steps **3a**, **3b**, **3c**, and **3d**:

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.

3c. Some additional modules need to be installed into the Strawberry Perl `C:\myperl` area. Enter the command below into the already open command line window (copy and paste is the best method - to paste, right-click in the command line window). After you've entered the command, hit `Enter`.

```
%USERPROFILE%\Desktop\convert_to_1p4\install_modules.bat
```

Note: If you are re-running the converter, you don't need to do this step again. The modules only need to be installed once.

3d. The working directory needs to be set - enter the command below into the command line window and hit Enter.

```
cd %USERPROFILE%\Desktop\convert_to_1p4
```

Note: If your user folder is redirected to a non C: drive, you'll need to change the working drive in the command line window before doing the `cd` command above. Just type your drive letter followed by a colon in the command line window, and hit Enter. For example, if our drive is X:, type `X:` and hit Enter. Then do the appropriate `cd` command to reflect where you've put the convert_to_1p4 folder.

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 macOS, simply double-click the *AppleScript_Conversion_Helper* app and follow the on-screen instructions. If you already have your exported data file from Step 2, you may drag it *onto* the *AppleScript_Conversion_Helper* app.

If you get the macOS 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>

Select a converter:
    clipperz csv dataguardian datavault essentialpim ewallet fsecurekey handysafe ironkeyim
    keepass2 keepassx keeper keychain keyfinder lastpass licensekeeper msecure nortonis
    onepif passpack passwordagent passworddepot passwordsafe passwordsplus passwordwallet
    roboform roboform_de safeincloud safewallet spbwallet splashid stickypassword truekey
    txt2notes vcard wallet4

Select one of the converters above and add it to the command line to see more
complete options.  Example:
```

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

Now, let's enter the command that performs the conversion. In the Terminal window (macOS) 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:

- **macOS**

  ```
  perl 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` *directly on the Desktop*

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 Using the Converted File

If the conversion was successful, there will be a file named **1P_import.1pif** on your Desktop, or if you used the **onepif** converter, a file named **1P_converted.html** or **1P_converted.csv** (depending on which export formatter you used).

- **Import a** 1P_import.1pif **file into 1Password**

  To import a **1P_import.1pif** file into 1Password, use 1Password's `File > Import` menu. When the *Import* dialog appears, select `Other` and then select `Import a 1PIF file`. From the pull down menu, select the desired vault to use for import, press the `Select File...` button, and navigate to your Desktop to select the file named **1P_import.1pif**, and click the `Open` button.

  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.

- **Print a** 1P_converted.html **File**

  To print a **1P_converted.html** file, 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.

- **Open a** 1P_converted.csv **File with a Spreadsheet Program**

  To view the CSV file **1P_converted.csv** in a spreadsheet, use your spreadsheets Open or Import function. The details depend on the spreadsheet you use.

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_converted.html, 1P_converted.csv), 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. These are described below. In addition, some converters may have additional options - see the `--help` output for a converter to see its list of supported options.

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:

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

Select a converter:
    clipperz csv dataguardian datavault essentialpim ewallet fsecurekey handysafe ironkeyim
    keepass2 keepassx keeper keychain keyfinder lastpass licensekeeper msecure msecure5
    nortonis onepif passpack passwordagent passworddepot passwordsafe passwordsplus
    passwordwallet roboform safeincloud safewallet spbwallet splashid stickypassword truekey
    txt2notes vcard wallet4 yojimbo

options:
    -a or --addfields          # add non-stock fields as custom fields
    -c or --checkpass          # check for known breached passwords
    -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
          --notimestamps       # do not set record modified/creation timestamps
    -o or --outfile <ofile>    # use file named ofile.1pif as the output file
    -t or --tags <list>        # add one or more comma-separated tags to the record
    -v or --verbose            # output operations more verbosely

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 picturecard prescription serialnum socialsecurity
    software voicemail votercard website
supported export types:
    bankacct creditcard driverslicense email login membership note passport password server
    socialsecurity software
```

Options: `--addfields`

By default, password manager fields that do not naturally map into 1Password fields are placed into an entry's notes section as key:value pairs. The `--addfields` option will instead generate custom fields, and place them in a section named **Original Fields**.

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: `--checkpass`

The `--checkpass` option will enable safely testing the passwords found in your export against a list of over 500 million passwords known to be compromised. It uses the Troy Hunt Pwned Passwords facility (1)(2), and your password will **not** be transmitted over the network. This facility has been endorsed by Agilebits (3).

If a password in an item is found to be compromised, the item is tagged with the tag **Password Compromised** so that you may review these items after you have imported the 1PIF file into 1Password. Information about these items will also be output when using the `--verbose` option, including a summary of the total number of discovered compromised passwords.

When the `--checkpass` option is not provided, the code used to perform these checks is **not** loaded, so those who do not wish to use this facility can feel an extra level of comfort and security.

For specific implementation details, see **Searching by range** under the **Pwned Passwords overview** in (2). In a nutshell, a password is hashed using the SHA1 algorithm, and the first 5 hexidecimal values of that hash - the prefix - is provided to the API. A list is returned of all the SHA1 prefix values that match, and the converter checks your password's SHA1 to see if it is contained in that list. If it is found, the item is tagged.

- 1: https://www.troyhunt.com/ive-just-launched-pwned-passwords-version-2/
- 2: https://haveibeenpwned.com/API/v2#PwnedPasswords
- 3: https://blog.agilebits.com/2018/02/22/finding-pwned-passwords-with-1password/

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: –notimestamps

The converter will use your password manager's time stamps, if they exist, for record creation and record modification. The `--notimestamps` option disables this, allowing 1Password to set the time stamps as it creates the records (i.e. they will be set to the time you import the 1PIF file).

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.

Note: Setting time stamps requires at least 1Password for Mac 5.3.BETA-17.

Option: `--tags`

The `--tags` option will assign one or more tags to the converted items. Each of the tags supplied in the comma-separated list of tag values will be assigned to each entry.

Source Files and Folders

The conversion suite consists of the main driver Perl script and several Perl modules, as described in the table below:

Folder	Purpose
convert_to_1p4.pl	The main conversion script.
Converters	Contains the individual converter modules.
Utils	Common code modules used by the main converter script and the converter modules.
JSON **Text** **UUID**	Code modules used by the conversion suite. 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.
Formatters	Conversion formatters used by the **onepif** converter.

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

The **onepif** converter is a generic 1PIF 1Password export converter. It is used, for example, to convert a 1PIF file to HTML (for printing), or CSV (for opening in a spreadsheet). There are several formatters available. Look inside the Formatters folder for their names.

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 export file is controlled by a formatter specified using the `--format` option along with your choice of format processor.

• Chrome

This converter does not require an export file; you do not need to supply one on the command line. The converter finds your Chrome data and decrypts it directly. You will need to quit Chrome before running the converter.

On macOS, the converter will cause two password dialog boxes to be presented, requesting access to your login keychain and the Chrome-specific password record that protects your Chrome storage (this key is required to decrypt your encrypted Chrome autofill entries). Supply the passwords when requested.

• Clipperz

Go to the Clipperz web site to export its database. When logged in, select the menu button in the upper right of the interface (the 3 stacked bars). Select `Data`, then select `Export`, and then click the `download HTML+JSON` button. This will save the export file to your desktop. You may rename it **pm_export.txt** if you wish, or can use the file name as it is. Use that name as your conversion file name.

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 logout of the main Clipperz page.

The Clipperz converter currently supports only English field names.

• CSV

This is a generic CSV converter which currently handles the the following categories:

- Bank Account
- Credit Card
- Identity
- Login

- Membership
- Password
- Social Security

Only one category per CSV file is currently supported.

Construct the CSV in a spreadsheet program. The first row must be a header row that contains the category-specific field names, and possibly other columns, as mentioned below. The letter case of the field names does not matter, nor does the order of the columns. A value must be present in the record for at least one of the field names shown in **bold** below in order to cause the category to be auto-detected. The supported column fields labels for each category are listed below, along with other notes about the data requirements:

Category	Fields	Notes
Bank Account	*Title*, **Bank Name**, **Owner**, **Account Type**, **Routing Number**, **Account Number**, **SWIFT**, **IBAN**, **PIN**, *Phone*, *Address*, *Notes*	If the **Account Type** is not one of the case-insensitive values **Checking**, **Savings**, **LOC** or **Line of Credit**, **ATM**, **Money Market** or **MM**, or **Other**, the Account Type will be placed in the Notes section of the entry.
Credit Card	*Title*, **Card Number**, **Expires**, **Cardholder**, *PIN*, **Bank**, **CVV**, *Notes*	If the date data in the Expires column is not formatted as mm/yyyy or mmyyyy, or is not a valid month/year, the expiration data will be placed in the Notes section of the entry.
Identity	*Title*, **First Name**, **Initial**, **Last Name**, **Sex**, **Birth Date**, **Occupation**, **Company**, **Department**, **Job Title**, **Address**, **Default Phone**, **Home Phone**, **Cell Phone**, **Business Phone**, **Default Username**, **Reminder Question**, **Reminder Answer**, **Email**, *Website*, **ICQ**, **Skype**, **AIM**, **Yahoo**, **MSN**, **Forum Signature**, **Notes**	The **Address** field will go to notes, since there is no universal single line address format that can be reliably parsed into 1Password's required internal form.
Login	*Title*, **Login Username**, **Login Password**, **Login URL**, *Notes*	
Membership	*Title*, **Group**, **Member Name**, **Member ID**, **Expiration Date**, **Member Since**, *PIN*, *Telephone*, **Membership Username**, **Membership Password**, **Membership URL**, *Notes*	If the date data in the Expiry Date or Member Since columns are not formatted as mm/yyyy or mmyyyy, or are not valid month/year values, the date data will be placed in the Notes section of the entry. The Username, Password and Website data if present will be placed into a Login record.
Password	*Title*, **Password URL**, **Password**, *Notes*	
Social Security	*Title*, *Name*, **SS Number**, *Notes*	

Other Columns

You may add a column named *Tags*. If there is a Tags column, the cells should contain any number of comma-separated values to be stored into 1Password's Tags.

You may add columns named *Modified* and/or *Created*. These columns will be used to set the record's modified and created dates, respectively. These values are Unix epoch integers (seconds since 1/1/1970). Invalid values will cause the data to be stored in the record's Notes.

You may add additional columns, and their titles will be used to create custom fields in the entry, and the corresponding values will be stored in these fields (do not use the reserved column names mentioned above).

Note: The file should be encoded as UTF-8. No BOM is required, but will be handled if it exists.

• 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: The DataVault CSV export is lossy - it does not export the Category, Type, or Template information. The DataVault CSV export on Windows is lossy. The user interface accepts Unicode characters, however, only latin1 characters are exported to the CSV file. Non-latin1 characters are transliterated. For example, the character *ş* becomes *s*. .

Note: The CSV export for DataVault for macOS does not properly CSV-quote its data. This will cause the converter to skip the problematic record and generate a warning. You will need to manually enter any of these noted records.

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

• Enpass

Launch Enpass and set the language to English under `Tools > Settings > Advanced > Language`. Quit Enpass completely and relaunch it. The language must be set to English for the field names in the export to be able to match what the converter expects.

Export your Enpass data to a CSV text file using the `File > Export > as CSV` menu. Provide your master password when the Export dialog appears, and click the `OK` button. Next click the `OK` button in the dialog that warns you about the data export being unprotected. In the Save dialog, navigate to your **Desktop** folder, and save the file with the name **pm_export.txt** to your Desktop. You may now quit Enpass.

Note: There are several generic Other categories in Enpass, and these contain only two stock fields Field 1 and Field 2. These records are entirely indistinguishable in the export - the converter maps these to a single **other** category.

Note: Enpass' export is problematic and ambiguous; the converter does the best it can given this.

• 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.

• F-Secure KEY

Launch F-Secure KEY and export its database to a text file. Click `Settings` in the sidebar, and then click `Export passwords` in the Settings pane. Click the `Export` button, and save the file with the name **pm_export.txt** to your Desktop. You may now quit F-Secure KEY.

• Handy Safe

macOS: 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.

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

Note: KeePassX version 2.0 does not support an XML export. However, its database can be read by KeePass 2. If you can install Keepass 2, you can use the Keepass 2 instructions above to perform the export and conversion using the **keepass2** converter. Unfortunately Keepass 2 installation on an macOS system is non-trivial, so if you happen to have a PC, do the export there. KeePassX version 2 can export to CSV, so you may also use the **csv** converter to perform the conversion.

• Keeper Desktop

Launch Keeper Desktop, and export its database to a file using the `Backup` button on the upper right side of the window.

For Keeper 10, press the last `Export Now` button, labeled *Export to Text File*. Keeper will open a Finder folder for you, with the exported file (it is named to include the export date, as in 20170815210313-keeper.txt). Drag the file to your **Desktop** folder. You may rename it **pm_export.txt** if you wish, or can use the file name as it is. You may now quit Keeper Desktop.

For versions prior to version 10, press the last `Export Now` button, labeled *Export to Excel (not encrypted)*. Navigate to your **Desktop** folder, and save the file with the name **pm_export.txt** to your Desktop. You may now quit Keeper Desktop.

• Key Finder

Launch Key Finder, and export its database as an XML export file using the `File > Save > Save to XML (*.xml)` menu item. When the Save dialog appears, enter the name **pm_export.xml**. Next, expand the dialog by pressing the downward triangle at the right of the `Save As` field, and select **Desktop** from the sidebar, and finally, press the `Save` button. You may now quit Key Finder.

• Keychain

There is no need to export any data for this converter. The converter will read and decrypt the keychain data directly from the specified keychain file. The simplest method to convert a keychain is to use the *AppleScript_Conversion_Helper*. Otherwise, you will need to supply on the command line the full path to the keychain file you wish to export. These paths can be obtained by running the command:

```
security list-keychains
```

in the Terminal app. A list will be output and you can just copy the entire line for the desired keychain, and enter that on the command line.

In either case, the converter will ask you to supply the password for the specified keychain. Enter it when asked, and hit Enter.

The converter has no access to iCloud-based keychains - it can only work on local keychains. iCloud keychains require copying entries to a new local keychain.

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 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:

```
set keychainPassword to "MYPASSWORD"

tell application "System Events"
    repeat while exists (processes where name is "SecurityAgent")
        tell process "SecurityAgent"
            set frontmost to true
            if (count of windows) > 0 then
                set window_name to name of front window
            end if
            try
                # keystroke "password"
                keystroke keychainPassword
                delay 0.1
                keystroke return
                delay 0.1
            on error
                -- do nothing to skip the error
            end try
        end tell
        delay 0.4
    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.

You can now run the *AppleScript_Conversion_Helper* and select the **local-icloud** you just created.

Once the export and conversion has completed, you may delete the **local-icloud** keychain in `Keychain Access` if you wish (delete both References and Files).

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

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.

Note: Some items from the iCloud keychain can not or will not be copied. You may have to work in batches to figure out which can be copied and which cannot. Resolving the reason for these failures is beyond the scope of this document.

• 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

There are two versions of the mSecure converter currently, one for version 5, and one for the previous versions of mSecure. The mSecure export format and capabilities of mSecure 5 have changed dramatically. Use the **msecure5** converter for version 5, and the **msecure** converter for older versions. mSecure 5 eliminated older, less-often used Types, and changed the names of the common Types. It is not yet clear to me what happens to user's data who have imported their mSecure 3 vaults into mSecure 5, in terms of the old Type names, etc., as I've been unable to import my old backup file. Please let me know if you have such a vault, and have a mixture of old and new Types.

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.

Note: The mSecure CSV export on Windows is lossy. The user interface accepts Unicode characters, however, only latin1 characters are exported to the CSV file. Non-latin1 characters are transliterated. For example, the character *ş* becomes *s*.

Note: The mSecure 5 CSV export sidesteps correctly encoding its CSV export. It converts double-quotes into Unicode left and right quotes for some fields. The **msecure5** converter must re-code these back to double quotes, but this leaves the possibility that intentional left and right Unicode quotes (0x201c and 0x201d) may get converter to ASCII double quotes.

• 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 macOS) 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 macOS, 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.

• Passwords Plus

Launch Passwords Plus, and export its database to a file using the `Tools > Export` menu item. Click the `Select Location` button, and enter the name **pm_export.csv** in the *Save As* area of the dialog, and click the `Where` button to navigate to your **Desktop** folder. Click the `Save` button, and then click the `Export` button in the original dialog, and click the `OK` button when the final dialog appears. The name of your *name_of_export_file* will be **pm_export.csv**. You may now quit Passwords Plus.

• Password Safe

Launch Password Safe, and export its database to an XML file using the `File > Export To... > XML Format...` menu item. Enter your password in the *Current Safe Combination* area. Click the **Export file** button to the right of the path entry area, navigate to your **Desktop** folder, and enter the name **pm_export.xml** in the *File name* field in the dialog, and click the `Save` button, and the finally, click the `Export XML` button to create the export file on your Desktop. Click the `Finish` button to close the export dialog. The name of your *name_of_export_file* will be **pm_export.xml**. You may now quit Password Safe.

• 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.

• RoboForm

RoboForm's Passcards/Logins, Identities, and Safenotes can be exported using its Print List feature. The converter currently supports Passcards/Logins and Safenotes conversions.

Launch RoboForm for macOS or for Windows, and export your passcards/logins, identities, and safenotes as follows:

For RoboForm 6 for Windows, use the Passcard Editor's `Passcard > Print List...` menu item. When the *Roboform Passcards List* dialog appears, select *Passcards* from the pulldown at the top of the dialog, and enable the *Full URL* checkbox. Now click the Save button. Navigate to your **Desktop** folder, and save the file with the name **pm_export_logins.html** to your Desktop. This file name will be the name you use as your *name_of_export_file*.

For RoboForm 7 for Windows, launch the RoboForm Editor, and click the green RoboForm button in the upper left of the dialog, and select `Print List... > Logins`. Enter your master password when prompted. When the print dialog appears, click the Save button. Navigate to your **Desktop** folder, and save the file with the name **pm_export_logins.html** to your Desktop. This file name will be the name you use as your *name_of_export_file*. See Note below.

For macOS, use the `File > Print List...` menu and select **Logins**. RoboForm will save a file to your Desktop that begins with the name *RoboForm Logins* and ends with the current date. This is your conversions file - you may want to rename it to a simpler **pm_export_logins.html**, and use that name as your *name_of_export_file*.

You may also want to export your Identities and Safenotes now, using a similar procedure, but using the file name **pm_export_identities.html** and **pm_export_safenotes.html**, respectively.

You may now quit RoboForm. Perform conversions and imports on each of the files you exported. The **roboform** converter will convert one or more export files at once, so you may specify each of the files you exported above on the same command line. Example:

```
perl convert_to_1p4.pl roboform -v pm_export_logins.html pm_export_identities.html pm_export_safenotes.html
```

The command above will convert all three export files into a single 1PIF file containing your converted logins, identity items, and safenotes.

Note: RoboForm version 7 and above does not export the full original URL for Login items. You will probably need to edit some login entries within 1Password to make Open and Fill work correctly.

• 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 macOS, 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, your *name_of_export_file* will be **..\name_of_your_spbwallet_file.swl**.

If would prefer not to move the .swl file, you'll have to specify a path to your wallet file as your *name_of_export_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.

• Sticky Password

Launch Sticky Password, and export its database to an XML file using the `Menu` pulldown in the upper right corner of the user interface. Select the `Export` menu item, and then select the `Export All` item. In the next dialog screen, select `Sticky Password 6 XML` and click `Next`. Select `Save to file`. In the next dialog screen, navigate to your **Desktop** folder, and in the `File name` area, enter the file name **pm_export**, and click the `Save` button. Your file will be named **pm_export.xml** after exporting the data. You may now quit Sticky Password.

• True Key

Launch True Key, and export its login entries as text file using, using the settings gear icon in the upper right corner of the user interface. Select `App Settings` below that, and click the `Export` button in the `Export Data` section of the settings at the bottom of the list. Click `Continue` when the export warning dialog is presented, enter your master password when prompted, and click the `Unlock` button. When the Save dialog appears, navigate to your **Desktop** folder, and in the `Save As` area, enter the file name **pm_export**, and click the `Save` button. After exporting the data, your file will be named **pm_export.csv** or **pm_export.json**, depending upon your version of True Key. You may now quit True Key.

• vCard

The vCard converter will import contacts into the Identity category by default. Include the `--securenote` option to place them into the Secure Note category instead.

If an appropriate Perl graphics libraries is installed, and the `--icon` option is included, the vCard converter will import the vCard's icon. The currently supported graphics library is: GD.

macOS Contacts

To export from macOS Contacts, launch the macOS 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.

iCloud Contacts

To export from Contacts on iCloud, launch Safari, and log into your iCloud account. Click the **Contacts** icon. Select one or more contacts from the list (you can click on a single contact and then use ⌘-A to select all contacts). Then, click the gear icon at the bottom left of the broswer window, and select `Export vCard...`. Safari will create a new .vcf file in your Downloads folder. It will also prompt you to import your items into Contacts - you probably want to deny this by hitting the `Cancel` button. You may now log out of your browser's iCloud, or just quit Safari, or close the page. Move the newly created .vcf file onto your Desktop, and use this file's name as your *name_of_export_file*.

Other vCard Exports

The converter has not been tested with vCard exports from other applications. Please contact me with any requests you have.

• Text files to notes

This converter will import text files from the file system into Secure Notes. You may supply one or more file paths on the command line. If you supply a directory name, its contents will be converted. Sub-directories will be ignored unless you supply the `-r` or `--recurse` option.

You will need to install an additional Perl module to use this. Issue the command:

```
cpan File::Type
```

in the command shell. This only needs to be done once, prior to using this converter.

• Wallet 4 (4X / 4W)

Launch Wallet 4, and export its database as an HTML file using the `Wallet > Export HTML...` 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.html** after exporting the data). Click `Save`. You may now quit Wallet 4.

Note: Wallet 4 does not export attached files or images, so they are not available to the converter. Be sure to export your files from Wallet 4 before you remove it from your system. A Files or Pictures item in Wallet 4 will be imported as a Secure Note in 1Password, and it contains a reference to the original file name in the notes section.

Note: Favorites will not be retained. Wallet 4 does not link the Favorites entries to the actual entries. Only the name of the item is referenced in the favorites list, but this name may not be unique, so it is not possible to tag an item as a favorite.

• Yojimbo

Launch Yojimbo, select the Library sidebar area, and select all of the entries you want to export. Export the selected files using the `File > Export...` menu. In the `Open` dialog, navigate to your **Desktop** folder, and click the `New Folder` button at the bottom left of the dialog. Enter the folder name **pm_export**, and click `Create`, and then click the `Choose` button. You may now quit Yojimo. Supply the folder name **pm_export** to the command line when running the converter via command line.

By default, Serial Number entries are assigned to 1Password's Software License category. You may use the option `--serial2note` to force them to the Secure Note category.

Note: The converter will ignore **image** and **web archive** types. Formatted notes will be converted into plain text (since 1Password only supports plain text in notes).

Note: The converter cannot decrypt encrypted items. Be sure your items are decrypted prior to export.

Note: Yojimbo does not export a record's Tags, Flagged status, Labels, or Collections. And for some types, it does not export the entry's Comments.

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