Converting and Importing Data into 1Password

The script **convert_to_1p4.pl** can convert exported data from several password management programs into a 1PIF format that can be imported by 1Password. The following password management programs are currently supported:

- Clipperz
- eWallet
- Handy Safe
- Keepass 2
- KeepassX
- OS X Keychain
- LastPass
- mSecure
- Safe in Cloud
- SafeWallet
- SplashID

Requirements

The script is a Perl script. You will run it in a command shell under OS X using the **Terminal** application or under Windows using the **cmd.exe** command shell.

OS X

- 1Password for Mac, version 4.0 or higher

Windows

- 1Password for Windows, version 4.1.0.520 or higher
- ActivePerl version 5.16 (*not* 5.18)

Password Manager Version Requirements

The following list of password managers indicate the software version and platform used for export and conversion testing. It is not possible to test all versions on all the platform variations,

nor would it be a worthwhile venture. Minor version differences generally should not matter (unless noted), so you can try the export and conversion. The converters assume the export and conversion are to be done on the same platform. You are welcome to report any issues you find.

- Clipperz: the clipperz web program does not have typical version numbers; tested against clipperz as of Sept 2014.
- eWallet: OS X: 7.3; Windows: 7.6.4
- Handy Safe: OS X: 1.02; Windows: Handy Safe Desktop Professional 3.01
- KeePass 2: Windows: 2.26
- KeepassX: 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): OS X 10.9.5, 10.10
- LastPass: LastPass is a browser-dependent extension, and various versions exist (version 3.1.54 Firefox/Windows tested)
- mSecure: OS X: 3.5.4 (previous versions do not properly export CSV, but the converter compensates); Windows: 3.5.4 (do not properly export CSV, has quoting / escaping issues)
- Safe in Cloud: OS X: 1.6; Windows: 2.8
- SafeWallet: OS X 1.2; Windows: 2.4.1.2
- SplashID Safe: OS X: 7.2.2; Windows: 7.2.4

Instructions

The following instructions will guide you through the specifics required to export your wallet data, convert it, and import it into 1Password. Instructions specific to either OS X or Windows will be noted below. In addition, instructions specific to a given password manager will be noted in a separate section below. Finally, these instructions assume you have placed the onepassword-utilities folder onto your Desktop.

1. Verify Requirements

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

OS X includes Perl, so no additional software is required.

Windows users will need to download and install ActivePerl for the OS type. Select either the 32-bit or the 64-bit version of ActivePerl. You do not need to install the documentation or example scripts. Allow the installer to modify your PATH (otherwise you will need to specify the full path to the perl program below). When you are done with the conversion, you may uninstall ActivePerl. Some additional modules are required, and installations instructions are noted in Step 2 below.

2. Open the Command Line Shell

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):

```
cd Desktop/onepassword-utilities/convert_to_1p4
```

and hit Enter.

On Windows, start the command shell by going to the Start menu, and entering **cmd.exe** in the *Search programs and files box*, and hit Enter. When the cmd.exe command line window opens, type:

```
cd Desktop\onepassword-utilities\convert_to_1p4
```

and hit Enter. You are advised to increase the width and height of the *cmd.exe* window - it is very small by default. Right-click its titlebar, select Properties, and set the Window Size to 100 x 90 (width x height).

Some additional modules are required for Windows users - the commands below only need to be run once, after you've installed ActivePerl. In the **cmd.exe** window, type the following command and hit Enter:

```
ppm install xml-xpath
```

3. Export Your Data

See your password manager's specific section under **Exporting from your Password Manager** below, and proceed to Step 4 when the data is exported.

4. Execute the Perl Script

You are about to convert your data. The script only reads the file you exported from your

password manager, and does not touch your password manager's original data file, nor does it transmit any data across a network. Since the script is readable by anyone, you are free and welcome to examine it and ask any questions should you have any concerns.

On OS X, in the Terminal window, enter the command:

```
perl convert_to_1p4.pl converter -v ../../pm_export.txt
```

Yosemite note: replace the `perl` command above with `perl5.16` (Yosemite ships with a newer version of Perl).

On Windows, in the command shell, enter the command:

```
perl convert_to_1p4.pl converter -v ..\..\pm_export.txt
```

where **converter** is the name of the appropriate converter (from the list at the top of these instructions, or as shown in the output when using `--help`). Hit `Enter` after you've typed or copy/pasted the command above. See **Note** below.

The `-v` option will cause the script to indicate the number of records imported and converted (per type) to the 1PIF file.

Note: The command line above assumes that your Desktop contains:

- the exported password manager text file **pm_export.txt**
- the folder `onepassword-utilities/convert_to_1p4` on OS X or
- the folder `onepassword-utilities\convert_to_1p4` on Windows

and the `convert_to_1p4` folder contains the `convert_to_1p4.pl` script and all of its accompanying modules.

5. Import 1PIF into 1Password

If the conversion was successful, there will be a file named **1P4_import.1pif** on your Desktop. To Import this file, use the 1Password `File > Import` menu and select the file 1P4_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.

6. Securely Remove Exported Data

As soon as you are completed the import into 1Password, be sure to securely delete the exported pm_export.txt file you created in Step 3, as well as the import file created by the converter script in Step 4, 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 <export_text_file>

converters: clipperz ewallet handysafe keepass2 keepassx keychain lastpass msecure safeincloud splashid

options: –debug | -d # enable debug output –exptypes | -e # comma separated list of one or more export types from list below –help | -h # output help and usage text –imptypes | -i # comma separated list of one or more import types from list below –outfile | -o # use file named outfile.1pif as the output file –verbose | -v # output operations more verbosely –[no]watchtower | -w # set each card's creation date to trigger Watchtower checks (default: on)

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: `--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 save your export data file to your Desktop using the name **pm_export.txt**. 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 may have to adjust commands above to reflect the actual name of the export file created.

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

● 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.

● 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.
```

• 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.

• 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

Copy the command below and paste it into the Terminal app:

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

and press Enter.

OS X will prompt you to allow the action - press `Allow`. You will have to do this repeatedly until all the entries have been exported. If you have many entries, a Click Allow AppleScript will press `Allow` automatically. To see how many entries you have, open the `Keychain Access` application under `Applications > Utilties`. Select the **login** item under **Keychains**, and **All Items** under **Categories**.

• 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: LastPass exports Form Fill Profiles separately. When you have completed converting and importing the data you exported above, repeat the process from Step 3 onward, this time exporting your Form Fill Profiles using the `Tools > Advanced Tools > Export To > Form Fill Profiles` menu.

● 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.

● 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.

● 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: you cannot use the .txt suffix - your file will be named **pm_export.vld** after exporting the data - please make adjustments to the commands above, replacing the **.txt** suffix with **.vld** where appropriate). 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.

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).

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).

Questions, Help and Requests

If you have questions, issues, or requests, feel free to ask for assistance in the specific forum thread: TBD

-MrC