

---

## Name

passwdmancli — CLI for storing, generating and updating passwords and honeypots.

## Synopsis

```
passwdmancli -i ---interactive -B ---single-backup infile
```

## DESCRIPTION

*infile* is the file **passwdmancli** will read its commands from, if *infile* is omitted **passwdmancli** will read from standard input. `-i` or `--interactive` must be specified if you want the commands **undo** and **redo**, or if you want **passwdmancli** to stay alive after an error.

If you specify the option `-B` or `--single-backup`, every change that has been done, will be removed by TWO undos.

**passwdmancli** reads its input as colon separated values. Quotes are either forbidden or optional (look at `unquote()` in `passwdmanapi(3)`). Integers are created with the python built-in `int()`. All commands are listed in **COMMANDS**. If the line begins with a number-sign, the line will be ignored. If the line begins with a pipe-sign, it will write the rest of the line to standard output.

All outputs that are prefixed with an exclamation mark are error messages. All outputs that are prefixed with a number sign are messages for a human user. All other outputs are prefixed with their command (not their arguments), using dots instead of colons, followed by a question mark.

Beware of whitespace. The whitespace comes after colons and/or question marks and before quotes.

The passwords have meta-data. They have a minimum required length and a maximum allowed length. They also have a type which is one of:

- 10        The password may only use digits.
- 64        The password can use big letters, small letters, digits, underscores and exclamation marks.
- human    The password is human generated.

The honeypots are weak passwords intended to be used as traps.

WARNING: If a command accepts a `name` and `index` argument, any string of digits will be treated as an index.

## COMMANDS

<code>bells</code>	Show message about the bells and whistles.
<code>bugs</code>	Show message about wasps and mosquitoes.
<code>hell</code>	Synonym for <b>verbose:on</b> .
<code>help</code>	Help message.
<code>honeypot:add:'value'</code>	Add a new honeypot with the value <code>value</code> .
<code>honeypot:help</code>	Help message.
<code>honeypot:list</code>	List all honeypots. Returns multiple lines of: "honeypot.list?", the index, a colon and the value surrounded by single quotes.
<code>honeypot:pick</code>	Pick a honeypot. Returns a line of: "honeypot.pick?" followed by the honeypot surrounded by single quotes.

honeypot:pick:n:'sep'	Pick n honeypots and separate them with <code>sep</code> . Returns a line of: "honeypot.pick?" followed by the separator-separated honeypots surrounded by single quotes.
honeypot:pickl:n	Pick n honey-pots, escape and quote (single quotes). Returns a line of "honeypot.pickl?" followed by the quoted honey-pots separated with commas.
honeypot:remove:'value'	Remove the honeypot with the value <code>value</code> .
honeypot:remove:index	Remove the honeypot at <code>index</code> . WARNING: The indexes of the following honeypots will be decreased by one!
passwd:add:type:minlength:maxlength:and:ch	Add a new password for <code>name</code> , with at least <code>minlength</code> letters, but no more than <code>maxlength</code> . If <code>type</code> is "10", the password will only contain digits. If <code>type</code> is "64", the password will contain big letters, small letters, digits, underscores and exclamation marks. It will not give any output.
passwd:add_human:'name'	Add a human generated password for <code>name</code> . It will read the password on a new line. It will not give any output.
passwd:get:'name'	Returns a line of: "passwd.get?" followed by the value of the password for <code>name</code> surrounded by single quotes.
passwd:get:index	Returns a line of: "passwd.get?" followed by the value of the password at <code>index</code> surrounded by single quotes.
passwd:help	Help message.
passwd:list	List all passwords. Returns multiple lines of: "passwd.list?", the index, a colon and the name surrounded by single quotes.
passwd:meta:'name'	Get the meta-data for the password for <code>name</code> . Returns a line of: "passwd.meta?", the type surrounded by single quotes, a colon, the minimum length for the password surrounded by single quotes, a colon and the maximum length for the password surrounded by single quotes.
passwd:meta:index	Get the meta-data for the password at <code>index</code> . Returns a line of: "passwd.meta?", the type surrounded by single quotes, a colon, the minimum length for the password surrounded by single quotes, a colon and the maximum length for the password surrounded by single quotes.
passwd:remove:'name'	Remove the password for <code>name</code> .
passwd:remove:index	Remove the password at <code>index</code> . WARNING: The indexes of the following passwords will be decreased by one!
passwd:update:'name'	Update the password for <code>name</code> using its meta-data. Returns a line of: "passwd.update?", the old password, a colon and the new password.
passwd:update:index	Update the password at <code>index</code> using its meta-data. Returns a line of: "passwd.update?", the old password, a colon and the new password.
passwd:update_meta:type:min:max:'name'	Do the same thing as <b>passwd:update:'name'</b> , but use <code>type</code> , <code>min</code> , and <code>max</code> as meta-data, instead of the passwords own (which might not even exist).

<code>passwd:update_meta:type:min:max:index</code>	Do the same thing as <b><code>passwd:update:index</code></b> , but use <code>type</code> , <code>min</code> , and <code>max</code> as meta-data, instead of the passwords own (which might not even exist).
<code>redo</code>	Opposite of <b><code>undo</code></b> .
<code>undo</code>	CLI undo button; restore from latest auto-generated backup.
<code>verbose:on</code>	Enable verbose mode; release mosquitoes from hell that will be noisy and give you a headache.
<code>verbose:off</code>	Disable verbose mode; kill the mosquitoes.
<code>whistles</code>	Show message about the bells and whistles.

## FILES

`~/.passwdman/passwords` is a XML file containing all passwords.

`~/.passwdman/honeypots` is a XML file containing all honeypots.

`~/.passwdman/undoable` is a directory full of backups.

`~/.passwdman/redoable` is a directory full of backups.

## DIAGNOSIS

All error messages are sent to standard output and are prefixed with an exclamation mark.

<code>!syntax error</code>	Bad speling or incorrect usage. The next line of output will be the offending input prefixed with an exclamation mark.
<code>!no meta</code>	The command requires meta-data, but the password doesn't have any.
<code>!not found</code>	Not found or index out of range

## SEE ALSO

`passwdmanapi(3)`, `random(4)`, `urandom(4)`, `passwdmanfiles(5)`

## BUGS

Mosquitoes:

- If verbose mode is enabled the message "#READY" is written too many times.

## SECURITY

**`passwdmancli`** uses `random(4)` or `urandom(4)` for everything.

## AUTHOR

Written by Oskar Skog ([oskar.skog.finland@gmail.com](mailto:oskar.skog.finland@gmail.com)).

Please send patches, questions, bug reports and wish-lists.