

Devolutions Password Server Command Line Interface Table



MANY END USERS LIKE HAVING COMMAND LINE OPTIONS AVAILABLE

A long, long, long time ago, before graphical user interfaces (GUI), there was the humble command line. Or maybe we should call it the hated **command line**, because so many developers dreaded them (and still do).

However, many end users like having command line options available, because it can be **faster** and **more flexible** — especially when entering multiple commands. Plus, it's nice to have the ability to astonish your non-technical colleagues every now and then with some ultra-mega-super-geek command line wizardry.

With this in mind, **here is a table that highlights some basic command line options for Devolutions Password Server:**

CONFIG

Parameters		Comment	Example
-V	--vault	The default vault id to use for the other commands: 0000000-0000-0000-0000-0000-000000000000	DPS> config --server http://localhost/dvls/api DPS> config -v 12345678 DPS> config --json [{"server":"http://localhost/dvls/api","vault":"00000000-0000-0000-0000-0000-000000000000"}]
-s	--server	The address of the server to connect to. If you call config without parameters, it will return the current settings.	
Options			
-j	--json	The command output will be formatted in JSON format.	

GET

The GET command is used to **return the credentials**

Parameters		Comment	Example
		The entry's ID can be found in its advance property or in the last part of its URL.	DPS> get 4574725f-0d24-4cbc-a116-a5270179e251 domain: dddd
Options			
-j	--json	The command output will be formatted in JSON format.	username: asdf password: 12345 DPS> get 4574725f-0d24-4cbc-a116-a5270179e251 --json { "domain": "dddd", "username": "asdf" "password": "12345" }
-d	--domain	Only returns the domain	
-u	--username	Only returns the username	
-p	--password	Only returns the password	DPS> get 4574725f-0d24-4cbc-a116-a5270179e251 -u asdf DPS> get 4574725f-0d24-4cbc-a116-a5270179e251 -p 12345

```
DPS> get 4574725f-0d24-4cbc-a116-a5270179e251
domain: dddd
username: asdf
password: 12345

DPS> get 4574725f-0d24-4cbc-a116-a5270179e251 --json
{
  "domain": "dddd",
  "username": "asdf",
  "password": "12345"
}

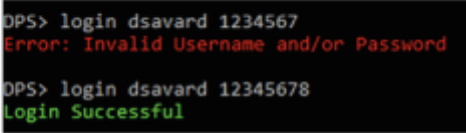
DPS> get 4574725f-0d24-4cbc-a116-a5270179e251 --domain --json
{"domain":"dddd"}

DPS> get 4574725f-0d24-4cbc-a116-a5270179e251 --username --json
{"username":"asdf"}

DPS> get 4574725f-0d24-4cbc-a116-a5270179e251 --password --json
{"password":"12345"}

DPS>
```

LOGIN

Parameters		Comment	Example
<p>First parameter: the username (or the key for an application)</p> <p>Second parameter: the password (or the secret for an application)</p>			<pre>DPS> login dsavard 12345</pre>  <pre>DPS> login dsavard 1234567 Error: Invalid Username and/or Password DPS> login dsavard 12345678 Login Successful</pre>
Options			
-j	--json	The command output will be formatted in JSON format.	

LOGOUT

The LOGOUT command is used to **logout the current user**.

```
DPS> logout
```

```
DPS> logout
Logout Successful
```

SET

The SET command line is used to **update the credentials of an entry.**

Parameters		Comment	Example
		The entry's ID can be found in its advance property or in the last part of its URL.	DPS> set 4574725f-0d24-4cbc-a116-a5270179e251 --domain newdomain
Options			
-j	--json	The command output will be formatted in JSON format.	
-d	--domain new-value	Update the domain	
-u	--username new-value	Update the username	
-d	--password new-value	Update the password	

```
DPS> set 4574725f-0d24-4cbc-a116-a5270179e251 --domain newdomain --username newuser --password newpassword
Entry has been successfully updated

DPS> get 4574725f-0d24-4cbc-a116-a5270179e251
domain: newdomain
username: newuser
password: newpassword
```

I hope you find this information useful. You may not plan on using command lines in DPS, but it's nice to know that you can "just in case."

Also, if you use any other DPS command lines that aren't in the table, **then please share them below.**