

# HWDB Python software

## Location

SVN: `tool/hwdb/src/linux`

Public installation: `/afs/slac/g/cci/hwdb`

In order to use the software add the directory to the front of your `PATH` and `PYTHONPATH` environment variables.

## Return codes

All of these commands yield a return code of zero upon successful termination and yield a non-zero code upon failure.

## Confluence login

Most of these scripts will prompt you for your Confluence password so that they can log in to Confluence. Some scripts will take a login token on the command line instead; run `hwdblogin.py` to get a token. Each such token is good for 60 minutes but will also expire if not used for anything for 20 minutes.

Commands will that ask for a password log you in to Confluence just once and will terminate when the login expires. Commands that take a login token will fail if the token is illegal or has expired.

There are several different types of authentication failures that you may see:

Error code 253

Unable to log in to server: <https://confluence.slac.stanford.edu/rpc/soap-axis/confluenceservice-v1> with user: `<username>`.

Cause: `com.atlassian.confluence.rpc.AuthenticationFailedException: Attempt to log in user '<username>' failed - incorrect username/password combination.`

You used the wrong user name or password. This increments the failed login count for the account, assuming that the user name is actually a valid one.

Error code 253

Remote error: `com.atlassian.confluence.rpc.InvalidSessionException: User not authenticated or session expired.`  
Call `login()` to open a new session.

This is what you'll see when you try to use an illegal or expired token.

Error code 253

Unable to log in to server: <https://confluence.slac.stanford.edu/rpc/soap-axis/confluenceservice-v1> with user: `<username>`.

Cause: `com.atlassian.confluence.rpc.AuthenticationFailedException: Attempt to log in user '<username>' failed. The maximum \ number of failed login attempts has been reached. Please log into the web application through the web interface to reset the nu\ mber of failed login attempts.`

Confluence allows five consecutive login failures for a given account. After that you'll be required to log in to that account via the web interface and type in a CAPTCHA phrase in order to prove that you're not a robot.

## Creating blank pages: `makeblanks.py`

Blank pages are all children of the page "Unused HWDB pages". To add more pages and get their URLs for the QR code stickers run `makeblanks.py`. It will prompt you for the number of new pages to create. `makeblanks.py` will print the URLs of the newly created pages. Save these so that stickers may be printed from them.

```
bash-3.2$ makeblanks.py
```

```
Number of new pages: 10
```

```
Confluence password:
```

```
New page URLs:
```

```
https://confluence.slac.stanford.edu/pages/viewpage.action?pageId=165088610  
https://confluence.slac.stanford.edu/pages/viewpage.action?pageId=165088611  
https://confluence.slac.stanford.edu/pages/viewpage.action?pageId=165088612  
https://confluence.slac.stanford.edu/pages/viewpage.action?pageId=165088613  
https://confluence.slac.stanford.edu/pages/viewpage.action?pageId=165088614  
https://confluence.slac.stanford.edu/pages/viewpage.action?pageId=165088615  
https://confluence.slac.stanford.edu/pages/viewpage.action?pageId=165088617  
https://confluence.slac.stanford.edu/pages/viewpage.action?pageId=165088618  
https://confluence.slac.stanford.edu/pages/viewpage.action?pageId=165088619  
https://confluence.slac.stanford.edu/pages/viewpage.action?pageId=165088620
```

## Listing blank pages: listblanks.py

Just in case, you can use listblanks.py to list the URLs for all the children of the "Unused HWDB pages" page. The line with the page count is written to stderr while the URLs are written to stdout.

```
bash-3.2$ listblanks.py
Confluence password for tether:
30 pages
https://confluence.slac.stanford.edu/pages/viewpage.action?pageId=163485551
https://confluence.slac.stanford.edu/pages/viewpage.action?pageId=163485553
https://confluence.slac.stanford.edu/pages/viewpage.action?pageId=163485556
...
```

## Getting a login token: hwdblogin.py

Some commands can't prompt for a Confluence password because they have to be non-interactive. For such commands you need to obtain a login token by running hwdblogin.py. It will prompt for a password, log in to Confluence and print the resulting token on stdout. You'll need to capture the token for later use.

```
bash-3.2$ token=$(hwdblogin.py)
Confluence password for tether:
bash-3.2$ echo $token
49eb5131ce
```

## Allocating a MAC address: allocmac.py

The file "macfile" in the same directory as this software shows which of the DAT group's MAC address have been allocated and which are still free. allocmac.py reads the file, allocates one MAC address, updates the page and prints the allocated address on stdout. For testing purposes allocmac.py can take an option --macfile which specifies a MAC address file different from the default.

```
bash-3.2$ mac=$(allocmac.py --macfile mymacs)
bash-3.2$ echo $mac
08:00:56:00:43:38
```

The MAC file read by allocmac.py must consist of exactly three lines. The first line contains the first address in the block assigned to the DAT group (or your test block). The second line contains the first free address. The third line contains the last address in the assigned block. For example:

```
08:00:56:00:40:00
08:00:56:00:43:31
08:00:56:00:7f:ff
```

## Checking pages for validity: checkpage.py

This command prompts repeatedly for page identifiers. These may be URLs of the page ID form used for the QR code stickers or just the page IDs themselves. checkpage.py performs the following checks on each page:

- The title contains two words: "HWDB" followed by the page ID.
- The Type field is present and valid.
- The Serial field is present and valid.

```
bash-3.2$ checkpage.py
Check device pages in the hardware database in Confluence.
Type control-D or control-C at a prompt to terminate.
```

```
Confluence password for tether:
```

```
Enter page ID (or URL with Page ID in it): 165089846
Title is 'HWDB 165089846'.
OK
```

```
Enter page ID or URL from sticker: https://confluence.slac.stanford.edu/pages/viewpage.action?pageId=165089846
Title is 'HWDB 165089846'.
OK
```

```
Enter page ID or URL from sticker: ^D
```

## Filling in a device page: initpage.py

Although you can edit a device page in the hardware database just as you can any other Confluence page, you may find it more convenient to use initpage.py. This command repeatedly prompts for a page ID or sticker URL. It checks whether the Type and Serial fields are present and valid, prompting for the value of any missing or invalid field. You can enter a null value for the serial number if you don't know it yet. The page is updated accordingly. You're then asked to where you want to move the page.

initpage.py will add a comment to the page giving the "date of birth".

bash-3.2\$ initpage.py

Fill in incomplete hardware database pages in Confluence.

Type control-D or control-C at a prompt to terminate.

While making a page type a period to abort the page.

Confluence password for tether:

Enter page ID (or URL with Page ID in it): 165089846

Type field is missing or blank.

Device class?

0 COB

1 RTM

2 FLASH

3 DPM

4 DTM

? 0

Type field (hw revision)?

0 PC\_248\_101\_01\_C05

1 PC\_248\_101\_01\_C06

2 PC\_248\_101\_01\_C07

? 2

Serial field is missing or blank.

Serial number? 690000007e910c70

Move page to?

0 Site SLAC Lab 1

1 Spares SLAC Lab 1

? 1

Enter page ID or URL from sticker: ^D