

# File to dict

---

*Source:* <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2007-12/msg00658.html>

---

- *From:* [mrkafk@xxxxxxxxxx](mailto:mrkafk@xxxxxxxxxx)
  - *Date:* Fri, 7 Dec 2007 03:31:10 -0800 (PST)
- 

Hello everyone,

I have written this small utility function for transforming legacy file to Python dict:

```
def lookupdmo(domain):
    lines = open('/etc/virtual/domainowners','r').readlines()
    lines = [ [y.lstrip().rstrip() for y in x.split(':')] for x in
    lines]
    lines = [ x for x in lines if len(x) == 2 ]
    d = dict()
    for line in lines:
        d[line[0]]=line[1]
    return d[domain]
```

The /etc/virtual/domainowners file contains double-colon separated entries:

```
domain1.tld: owner1
domain2.tld: own2
domain3.another: somebody
....
```

Now, the above lookupdmo function works. However, it's rather tedious to transform files into dicts this way and I have quite a lot of such files to transform (like custom 'passwd' files for virtual email accounts etc).

Is there any more clever / more pythonic way of parsing files like this? Say, I would like to transform a file containing entries like the following into a list of lists with doublecolon treated as separators, i.e. this:

```
tm:$1$aaaa$bbbb:1010:6::/home/owner1/imap/domain1.tld/tm:/sbin/nologin
```

would get transformed into this:

```
[ ['tm', '$1$aaaa$bbbb', '1010', '6', , '/home/owner1/imap/domain1.tld/
tm', '/sbin/nologin'] [...] [...] ]
```

File to dict