Find and replace string in all files in a directory (Python recipe)

by Anthony Barker

ActiveState Code (http://code.activestate.com/recipes/277753/)

1

A friend of mine needed to change the IP address in his DNS hosting bind configuration. Here is a quick script I wrote during lunch it interates all files in a directory and finds and replaces as string. In this case an IP address.

Python, 27 lines

```
#!/usr/bin/python2
import fileinput, glob, string, sys, os
from os.path import join
# replace a string in multiple files
#filesearch.py

if len(sys.argv) < 2:
    print "usage: %s search_text replace_text directory" % os.path.basename(sys.argv[0])
    sys.exit(0)

stext = sys.argv[1]
rtext = sys.argv[2]
if len(sys.argv) ==4:
    path = join(sys.argv[3],"*")
else:
    path = "*"

print "finding: " + stext + " replacing with: " + rtext + " in: " + path

files = glob.glob(path)
for line in fileinput.input(files,inplace=1):
  lineno = 0
  lineno = string.find(line, stext)
  if lineno >0:
        line =line.replace(stext, rtext)

  sys.stdout.write(line)
```

Tags: text

4 comments

Paddy McCarthy 10 years, 2 months ago

WHy I have a toe in 'the dark side'. I too had the problem at work, several times in fact. The Unix tools that I knew made it fairly easy to write a one-liner that could make the changes to derived file names, but I didn't know enough ed or ex etc, to make my changes and save them to the same file.

On discussing this with collegues, one piped up "just use perl dash pie". "You what"?? said I. "It's easy", he said, and went on to explain "use perl -p -i -e 's/change this/..to this/g' ".

It worked, and looking up the perl I found out why it worked. But then came my dilemma, I'm the sites Python advocate (with not much to show at the moment). This one liner is very useful to me, but written in perl. What do I do?

I decided that I should be practical, perl-dash-pie is easier for me to remember than the syntax of the cut command, and all the sites Solaris and Linux boxes have perl so now I have added perl-dash-pie to my toolbag and squared it with my conscience.

But I'm still only a reluctant Perl programmer for the multi line stuff.

Kevin Martin 9 years, 9 months ago

Very Nice Trick... I LOVE the "Perl Dash PIE" trick!! Can it be made to recursively "find and replace" as well?

Sean Kane 9 years ago

Updated Code. I made some changes to this code to make sure that it does not try to do search and replace on directories and allows the user to submit a list of extensions that should searched, so that not all files are checked. I'd like to also adjust this to make and keep a backup of the changed files. Unfortunely, if you use fileinput.input's build in ability to save a backup, then you end up with a nackup of every file checked.

```python
def searchreplace(req,path,search,replace,exts=None):

    import fileinput, glob, string, sys, os
    from os.path import join
    # replace a string in multiple files
    #filesearch.py

    files = glob.glob(path + "/*")
    if files is not []:
        for file in files:
            if os.path.isfile(file):
                if exts is None or exts.count(os.path.splitext(file)[1]) is not 0:
                    for line in fileinput.input(file,inplace=1):
                        lineno = 0
                        lineno = string.find(line, search)
                        if lineno >0:
                            line =line.replace(search, replace)
                        sys.stdout.write(line)
```

mark 8 years, 3 months ago

if lineno >0: should be if lineno >=0:

Also, how would you make this recursively check directories?