

```

#!/usr/bin/python
import dbif
import cfgif
import sys
import os
import time

#开始处理
def startaction(msg):
    actiontbl = cfgif.getactiontable()
    if not actiontbl:
        print "Can't find Action table name in config file"
        return None
    filetbl = cfgif.getfiletable()
    if not filetbl:
        print "Can't find DataFile table name in config file"
        return None
    datatbl = cfgif.getdatatable()
    if not datatbl:
        print "Can't find NMRData table name in config file"
        return None
    insertsqls = cfgif.getinsertdatasqls(datatbl)
    if not insertsqls:
        print "Can't prepare insert data sql with config file"
        return None
    (result, conn) = dbif.connect()
    if result and conn:
        cursor = conn.cursor()
        cursor.execute("INSERT INTO \""+actiontbl+"\" (message,createtime) VALUES (%s,%s)
RETURNING id;",
            (msg,time.strftime("%Y-%m-%d %H:%M:%S")))
        actid = cursor.fetchone()[0]
        conn.commit()
        cursor.close()

    return
{"actid":actid,"conn":conn,"actiontbl":actiontbl,"filetbl":filetbl,"datatbl":datatbl,"insertsqls":insertsqls}
    else:
        print "Failed to connect database"
        return None

#処理終了です
def finishaction(conn):

```

```

if not dbif.disconnect(conn):
    print "Failed to disconnect database"

def importfile(path,params):
    fn = os.path.split(path)
    print "Importing '"+fn[1]+"...",
    sys.stdout.flush()
    cursor = params["conn"].cursor()
    cursor.execute("INSERT INTO \'" + params["filetbl"] + "\" (name,action,directory) VALUES
(%s,%s,%s) RETURNING id;",
                    (fn[1],params["actid"],fn[0]))
    fid = cursor.fetchone()[0]
    rslt = -1
    with open(path) as f:
        sqls = []
        for ln in f:
            sqls.append([fid]+ln.split())
        rslt = len(sqls)
        if rslt > 0:
            cursor.executemany("INSERT INTO \'" + params["datatbl"] + "\" " + params["insertsqls"] + ":",
sqls)
    params["conn"].commit()
    cursor.close()
    print "OK(%d)"%(rslt)
    return rslt

def importdir(path,recursive,params):
    print "Enter "+path
    flist = os.listdir(path)
    for f in flist:
        fn = os.path.join(path,f)
        if os.path.isfile(fn):
            importfile(fn,params)
        if os.path.isdir(fn):
            if recursive:
                importdir(fn,recursive,params)

def processdir(path,msg,recursive):
    params = startaction(msg)
    if params:
        importdir(path,recursive,params)
    if params["conn"]:
        finishaction(params["conn"])

```

```

def processfile(path,msg):
    params = startaction(msg)
    if params:
        importfile(path,params)
    if params["conn"]:
        finishaction(params["conn"])

def main(args):
    msg = None
    if "-m" in args and len(args) > args.index("-m")+1:
        msg = args[args.index("-m")+1]
        if msg.startswith("-"):
            print "Message can't begin with '-'"
            return
    if "-d" in args and "-f" in args:
        print "\nCan't specify directory(-d) and file(-f) at the same time.\n"
        return
    if "-d" not in args and "-f" not in args:
        print "\nPlease specify a directory(-d) or file(-f) to import to database.\n"
        return
    dir_path = None
    file_path = None
    if "-d" in args:
        if len(args) > args.index("-d")+1:
            dir_path = args[args.index("-d")+1]
            if os.path.isdir(dir_path):
                processdir(dir_path,msg,("-r" in args))
            else:
                print dir_path+" is not a directory."
                return
        else:
            print "Please specify directory path."
            return
    if "-f" in args:
        if len(args) > args.index("-f")+1:
            file_path = args[args.index("-f")+1]
            if os.path.isfile(file_path):
                processfile(file_path,msg)
            else:
                print file_path+" is not a file."
                return
        else:
            print "Please specify file path."

```

```
    return
```

```
if __name__ == "__main__":  
    global Incnt  
    Incnt = 0  
    main(sys.argv)
```