

```

import os
import sys
import subprocess
import argparse
import re

def run_command(command, env=None):
    #print " Subprocess Call : %s" % subprocess.list2cmdline(command)
    try:
        if env:
            output = subprocess.check_output(command, stderr=subprocess.STDOUT, env=env)
        else:
            output = subprocess.check_output(command, stderr=subprocess.STDOUT)
        return output
    except subprocess.CalledProcessError:
        print " Error running the command %s " % str(command)
        sys.exit(-1)
    except Exception:
        print " Unknown error running the command %s " % str(command)
        sys.exit(-1)

if __name__ == '__main__':
    """ This tool is a wrapper around the isql command line tool provided by Sybase
    We used TeamCity as our continuous Build tool
    TeamCity has an option of terminating the build process when Task exits with a
    non zero exit code

    However, when sql script that is run by isql has an error, then isql just prints
    out an error message
    and exits with a zero status code.

    Hence we used this wrapper to catch the Errors and make the isql exit with a
    non-zero exit code
    """

    parser = argparse.ArgumentParser()
    parser.add_argument("-I", "--interfaces_file", type=str, required=True, help=
    "Interfaces File")
    parser.add_argument("-S", "--server", type=str, required=True, help="Server Name")
    parser.add_argument("-D", "--database", type=str, required=True, help="Database Name")
    parser.add_argument("-U", "--username", type=str, required=True, help="Username")
    parser.add_argument("-P", "--password", type=str, required=True, help="Password")
    parser.add_argument("-i", "--input_sql_file", type=str, required=True, help="Input
    sql file")

    args = parser.parse_args()

    if not os.path.isfile(args.input_sql_file):
        print "Input sql file : {0} is INVALID".format(args.input_sql_file)
        sys.exit(-1)

    if not os.path.isfile(args.interfaces_file):
        print "Interfaces file : {0} is INVALID".format(args.interfaces_file)

```

```
sys.exit(-1)

cmd = [r'isql',
       '-I', args.interfaces_file,
       '-S', args.server,
       '-D', args.database,
       '-U', args.username, '-P', args.password,
       '-i', args.input_sql_file]

output = run_command(cmd)

print output

# Check if any failure message is given by the isql output
if re.search(r'Msg [0-9]+, Level [0-9]+, State [0-9]+:', output) is not None:
    print "The sql script {0} failed to deploy successfully".format(args.
        input_sql_file)
    sys.exit(-1)
```