逗比云监控客户端windows版

2018年08月25日 · 2785点热度 · 7人点赞 · 0条评论

[title]说明[/title]

逗比云监控(https://github.com/ToyoDAdoubi/ServerStatus-Toyo)的windows客户端好久没更新，测试在win上也无法使用，所以就改了一下。

主要改了socket通信和流量统计。

本文还会介绍win中怎样开机启动。

[title]win客户端代码[/title]

status-win.py

```
# -*- coding: utf-8 -*-
# Update by : https://github.com/tenyue/ServerStatus
# 依赖于psutil跨平台库:
# 支持Python版本：2.6 to 3.5 (users of Python 2.4 and 2.5 may use 2.1.3 version)
# 支持操作系统：  Linux, Windows, OSX, Sun Solaris, FreeBSD, OpenBSD and NetBSD, both 32-bit and 64-bit architectures
# update by kamino 2018/08/25

SERVER = "127.0.0.1"
PORT = 35601
USER = "username"
PASSWORD = "password"
INTERVAL = 1  # 更新间隔

import socket
import time
import string
import math
import os
import json
import collections
import psutil


def get_uptime():
    return int(time.time() - psutil.boot_time())


def get_memory():
    Mem = psutil.virtual_memory()
    try:
        MemUsed = Mem.total - (Mem.cached + Mem.free)
    except:
        MemUsed = Mem.total - Mem.free
    return int(Mem.total / 1024.0), int(MemUsed / 1024.0)


def get_swap():
    Mem = psutil.swap_memory()
    return int(Mem.total / 1024.0), int(Mem.used / 1024.0)


def get_hdd():
    valid_fs = ["ext4", "ext3", "ext2", "reiserfs", "jfs", "btrfs", "fuseblk", "zfs", "simfs", "ntfs", "fat32", "exfat",
                "xfs"]
    disks = dict()
    size = 0
    used = 0
    for disk in psutil.disk_partitions():
        if not disk.device in disks and disk.fstype.lower() in valid_fs:
            disks[disk.device] = disk.mountpoint
    for disk in disks.values():
        usage = psutil.disk_usage(disk)
        size += usage.total
        used += usage.used
    return int(size / 1024.0 / 1024.0), int(used / 1024.0 / 1024.0)


def get_load():
    try:
        return os.getloadavg()[0]
    except:
        return -1.0


def get_cpu():
    return psutil.cpu_percent(interval=INTERVAL)
```

```
class Traffic:
    def __init__(self):
        self.rx = collections.deque(maxlen=10)
        self.tx = collections.deque(maxlen=10)

    def get(self):
        avgrx = 0
        avgtx = 0
        for name, stats in psutil.net_io_counters(pernic=True).items():
            if name == "lo" or name.find("tun") > -1:
                continue
            avgrx += stats.bytes_recv
            avgtx += stats.bytes_sent

        self.rx.append(avgrx)
        self.tx.append(avgtx)
        avgrx = 0
        avgtx = 0

        l = len(self.rx)
        for x in range(l - 1):
            avgrx += self.rx[x + 1] - self.rx[x]
            avgtx += self.tx[x + 1] - self.tx[x]

        avgrx = int(avgrx / l / INTERVAL)
        avgtx = int(avgtx / l / INTERVAL)

        return avgrx, avgtx


def liuliang():
    NET_IN = 0
    NET_OUT = 0
    vnstat = os.popen('netstat -e').readlines()
    for line in vnstat:
        if line[0:2] == "字节":
            mdata = line.split()
            NET_IN = int(mdata[1])
            NET_OUT = int(mdata[2])
            break
    return NET_IN, NET_OUT


def get_network(ip_version):
    if (ip_version == 4):
        HOST = "ipv4.google.com"
    elif (ip_version == 6):
        HOST = "ipv6.google.com"
    try:
        s = socket.create_connection((HOST, 80), 2)
        return True
    except:
        pass
    return False


if __name__ == '__main__':
    socket.setdefaulttimeout(30)
    while 1:
        try:
            print("Connecting...")
            s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            s.connect((SERVER, PORT))
            data = s.recv(1024)
            if data.find("Authentication required".encode('utf-8')) > -1:
                s.send((USER + ':' + PASSWORD + '\n').encode('utf-8'))
                data = s.recv(1024)
                if data.find("Authentication successful".encode('utf-8')) < 0:
```

```
                print(data.decode())
                raise socket.error
        else:
            print(data.decode())
            raise socket.error

        print(data.decode())
        data = s.recv(1024)
        print(data.decode())

        timer = 0
        check_ip = 0
        if data.find("IPv4".encode('utf-8')) > -1:
            check_ip = 6
        elif data.find("IPv6".encode('utf-8')) > -1:
            check_ip = 4
        else:
            print(data.decode())
            raise socket.error

        traffic = Traffic()
        traffic.get()
        while 1:
            CPU = get_cpu()
            NetRx, NetTx = traffic.get()
            NET_IN, NET_OUT = liuliang()
            Uptime = get_uptime()
            Load = get_load()
            MemoryTotal, MemoryUsed = get_memory()
            SwapTotal, SwapUsed = get_swap()
            HDDTotal, HDDUsed = get_hdd()

            array = {}
            if not timer:
                array['online' + str(check_ip)] = get_network(check_ip)
                timer = 10
            else:
                timer -= 1 * INTERVAL

            array['uptime'] = Uptime
            array['load'] = Load
            array['memory_total'] = MemoryTotal
            array['memory_used'] = MemoryUsed
            array['swap_total'] = SwapTotal
            array['swap_used'] = SwapUsed
            array['hdd_total'] = HDDTotal
            array['hdd_used'] = HDDUsed
            array['cpu'] = CPU
            array['network_rx'] = NetRx
            array['network_tx'] = NetTx
            array['network_in'] = NET_IN
            array['network_out'] = NET_OUT

            s.send(("update " + json.dumps(array) + "\n").encode('utf-8'))
    except KeyboardInterrupt:
        raise
    except socket.error:
        print("Disconnected...")
        # keep on trying after a disconnect
        s.close()
        time.sleep(3)
    except Exception as e:
        print("Caught Exception:", repr(e))
        s.close()
        time.sleep(3)
```

[title]开机自启动[/title]

把运行脚本放到startup文件夹中需要登录用户才会启动脚本，我们需要让脚本在不登录时就启动。

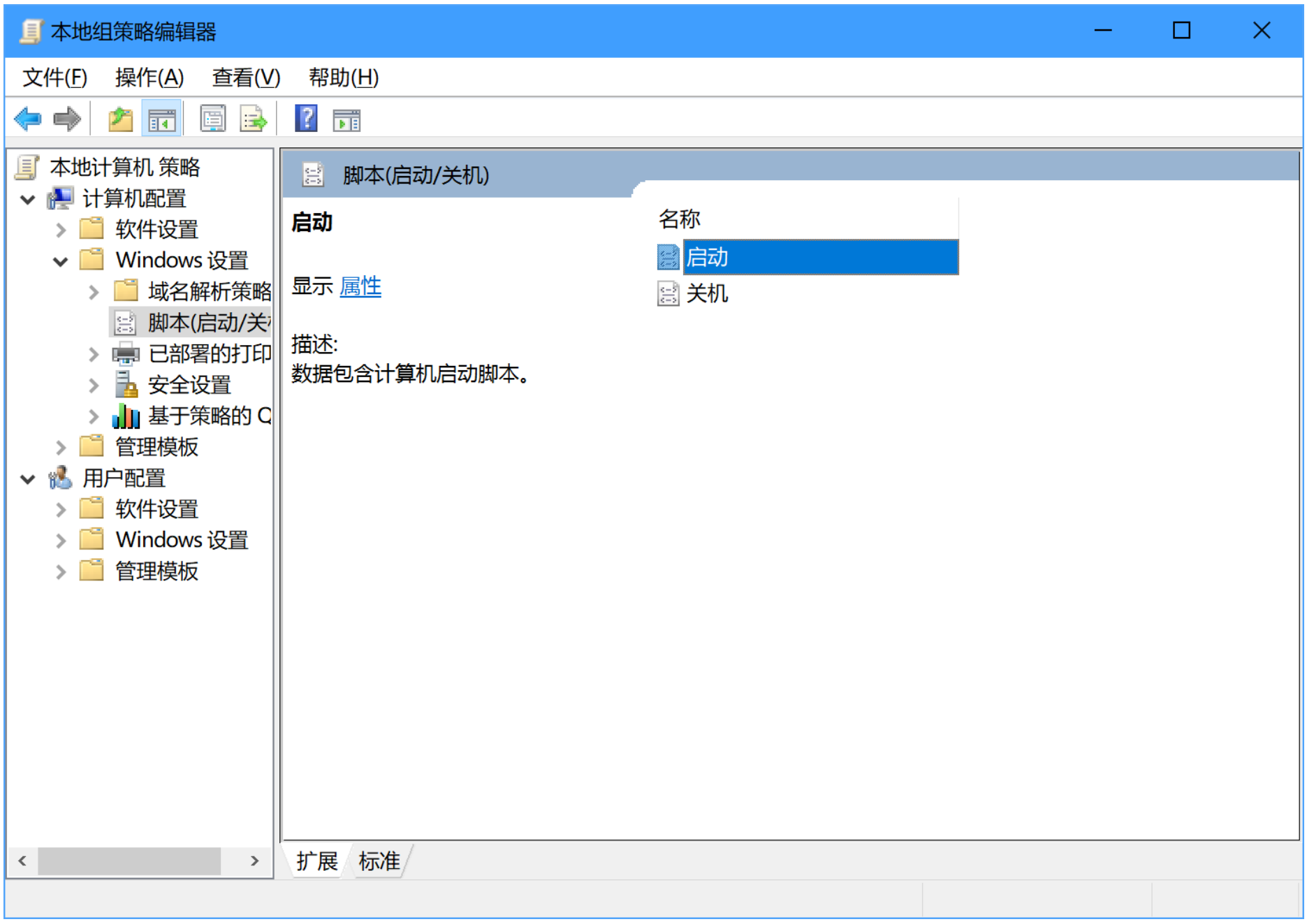
这篇文章可以解决这个问题：Windows中实现不依赖账户登录的开机启动程序

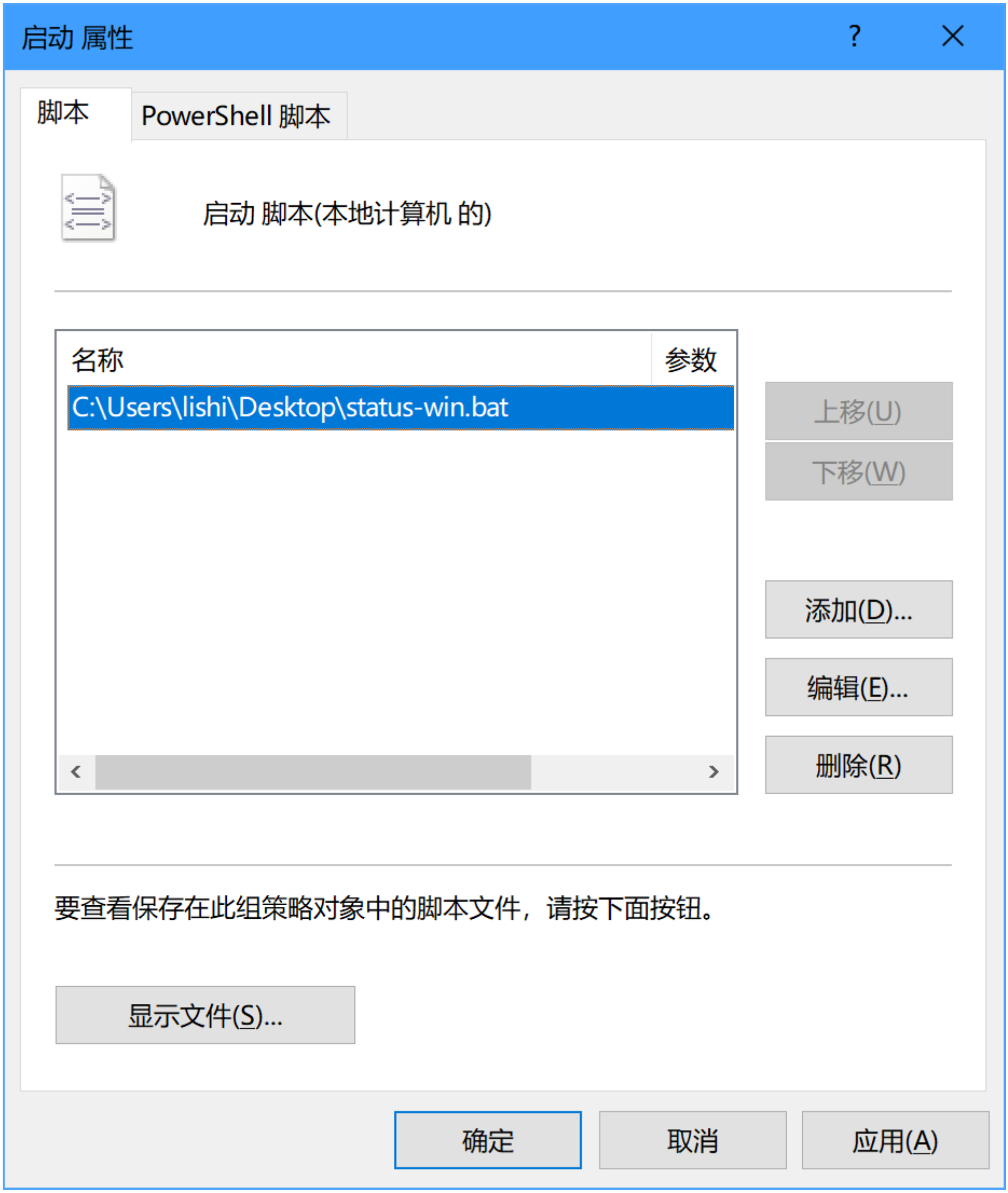
这里采用了Windows Startup Script使脚本开机自启

首先创建bat文件，内容为 `python status-win.py的绝对路径`

依此点击 运行 -> gpedit.msc -> 计算机配置 -> Window设置 -> 脚本(启动/关机) -> 启动 -> 添加

选择你创建的bat文件，确定，应用





标签： Python web

KAMINO

暂无

打赏 点赞

< 上一篇

下一篇 >