

Remote calls to a windows maschine

Last Updated Saturday, 13 December 2008

Here <http://coreygoldberg.blogspot.com> is another script to remotely get

- Uptime
- CPU Utilization
- Available Memory
- Memory Used
- Ping

with Python and the wmi module. Simply drop the wmi module and the script into a directory and import it from a separate script or i.e. add a `get_cpu("maschine_name")`.

Here is the source code:

```
import re
import wmi
from subprocess import Popen, PIPE
```

```
def get_uptime(computer, user, password):
    c = wmi.WMI(find_classes=False, computer=computer, user=user, password=password)
    secs_up = int([uptime.SystemUpTime for uptime in c.Win32_PerfFormattedData_PerfOS_System()][0])
    hours_up = secs_up / 3600
    return hours_up
```

```
def get_cpu(computer, user, password):
    c = wmi.WMI(find_classes=False, computer=computer, user=user, password=password)
    utilizations = [cpu.LoadPercentage for cpu in c.Win32_Processor()]
    utilization = int(sum(utilizations) / len(utilizations)) # avg all cores/processors
    return utilization
```

```
def get_mem_mbytes(computer, user, password):
    c = wmi.WMI(find_classes=False, computer=computer, user=user, password=password)
    available_mbytes = int([mem.AvailableMBytes for mem in c.Win32_PerfFormattedData_PerfOS_Memory()][0])
    return available_mbytes
```

```
def get_mem_pct(computer, user, password):
    c = wmi.WMI(find_classes=False, computer=computer, user=user, password=password)
    pct_in_use = int([mem.PercentCommittedBytesInUse for mem in c.Win32_PerfFormattedData_PerfOS_Memory()][0])
    return pct_in_use
```

```
def ping(host_name):
    p = Popen('ping -n 1 ' + host_name, stdout=PIPE)
    m = re.search('Average = (.*)ms', p.stdout.read())
    if m:
        return True
    else:
        raise Exception
```