

Forum:[Hack Forums](#)

Thread: [NetCheck simple SNMPc style network monitor](#)

b00st3d - October 7, 2014, 3:26 am

This is a script that monitors your network. It can also be used as a very simplified SNMPc if you want to monitor a device for connectivity. I made this one because my internet is not very reliable but I my wifi is always showing online. It is set up right now to poll www.google.com but as I said, it can poll any address that you want. What it does is if the device stops responding, it will wait 10 sec and try again. If it is still not responding then it will give a notification that the device is down. Same for if a device comes up.

I am using the popup notification that johnny cache posted [here](#)

Since this script runs invisible to the user, it will create a one-time-use kill script that will end the process then delete itself. If you don't use it then the next time the script is run it will just overwrite. This script is designed to run with the name NetCheck.bat If you try to name it something else, it will change it back to the correct name. Just try me.

Current script as of 10 April 2016

Code @echo off

cd %~dp0

cls

if not exist "%~dp0NetCheck.bat" (

ren "%~f0" NetCheck.bat

timeout /nobreak /t 2 >nul

call "%~dp0NetCheck.bat"

)

mkdir NClogs > NUL

cls

set address=www.google.com

::set address=10.150.0.1

call :getTime

set logfile=LOG_%time:~0,2%%time:~3,2% %day% %month% %year%.txt

<nul set /p ".=NetCheck is loading. Please wait"

:checkForFiles

if not exist invisible.vbs (

echo CreateObject^("Wscript.Shell"^).Run "***** ^& WScript.Arguments^(0^) ^& *****", 0, False >

invisible.vbs

<nul set /p ".=.

timeout /nobreak /t 1 >nul

)

REM Search for powershellBypass

```

if not exist powershellBypass.bat (
echo @echo off > powershellBypass.bat
echo cls >> powershellBypass.bat
echo ^:CheckPrivileges >> powershellBypass.bat
echo NET FILE 1^>NUL 2^>NUL >> powershellBypass.bat
echo if '%%errorlevel%%' == '0' ^( goto gotPrivileges ^) else ^( goto getPrivileges ^) >>
powershellBypass.bat
echo. >> powershellBypass.bat
echo ^:getPrivileges >> powershellBypass.bat
echo if '%%1'=='ELEV' ^(shift ^& goto gotPrivileges^) >> powershellBypass.bat
ECHO echo. >> powershellBypass.bat
echo ECHO ***** >> powershellBypass.bat
echo ECHO This program requires admin permission to enable popups >> powershellBypass.bat
echo ECHO ***** >> powershellBypass.bat
echo. >> powershellBypass.bat
echo setlocal DisableDelayedExpansion >> powershellBypass.bat
echo set "batchPath=%%~0" >> powershellBypass.bat
echo setlocal EnableDelayedExpansion >> powershellBypass.bat
echo ECHO Set UAC = CreateObject^("Shell.Application") ^> "%%temp%%\OEgetPrivileges.vbs"
>> powershellBypass.bat
echo ECHO UAC.ShellExecute "!batchPath!", "ELEV", "", "runas", 1 ^>^>
"%%temp%%\OEgetPrivileges.vbs" >> powershellBypass.bat
echo "%%temp%%\OEgetPrivileges.vbs" >> powershellBypass.bat
echo exit /B >> powershellBypass.bat
echo. >> powershellBypass.bat
echo ^:gotPrivileges >> powershellBypass.bat
echo ^:START >> powershellBypass.bat
echo setlocal ^& pushd . >> powershellBypass.bat
echo. >> powershellBypass.bat
echo powershell -command set-executionpolicy bypass >> powershellBypass.bat
<nul set /p .=.
timeout /nobreak /t 1 >nul

```

```

)
REM Check for start

```

```

if not exist start.bat (
echo @echo off > start.bat
echo if exist temp.ps1 del /f /q temp.ps1 >> start.bat
echo for /f "delims=" %%%%A in ^('powershell -command get-executionpolicy'^) do ^( >> start.bat
echo   set "GECP=%%%%A">> start.bat
echo ^) >> start.bat
echo echo Your current execution policy is: %%GECP%% >> start.bat
echo if not "%%GECP%%"=="Bypass" ^( >> start.bat
echo   call "powershellBypass.bat" >> start.bat
echo ^) >> start.bat
echo. >> start.bat
echo cls >> start.bat
echo cd %%~dp0 >> start.bat
echo cls >> start.bat
echo wscript.exe "invisible.vbs" "NetCheck.bat" >> start.bat
echo cls >> start.bat
echo exit >> start.bat

```

```
<nul set /p .=.  
timeout /nobreak /t 1 >nul  
start start.bat  
exit  
)
```

```
mode con:cols=40 lines=5  
set /A change = 2  
echo new log started: %dtg% >> "%~dp0\NClogs\%logfile%"  
:main  
cls
```

```
title monitoring Network Connectivity  
timeout /nobreak /t 2>nul  
del /Q "%~dp0invisible.vbs"  
del /Q "%~dp0powershellBypass.bat"  
del /Q "%~dp0start.bat"
```

```
REM CREATE Script to kill this process  
rem Note: Session Name for privileged Administrative consoles is sometimes blank.  
if not defined SESSIONNAME set SESSIONNAME=Console
```

```
setlocal
```

```
rem Instance Set  
set instance=%DATE% %TIME% %RANDOM%  
::echo Instance: "%instance%"  
title %instance%
```

```
rem PID Find  
for /f "usebackq tokens=2" %%a in (`tasklist /FO list /FI "SESSIONNAME eq %SESSIONNAME%"  
/FI "USERNAME eq %USERNAME%" /FI "WINDOWTITLE eq %instance%" ^| find /i "PID:") do set  
PID=%%a  
if not defined PID for /f "usebackq tokens=2" %%a in (`tasklist /FO list /FI "SESSIONNAME eq  
%SESSIONNAME%" /FI "USERNAME eq %USERNAME%" /FI "WINDOWTITLE eq  
Administrator: %instance%" ^| find /i "PID:") do set PID=%%a  
if not defined PID echo !Error: Could not determine the Process ID of the current script. Exiting.&  
exit /b 1
```

```
cd %~dp0  
echo taskkill /pid %pid% > kill.bat  
echo del "%~f0" >> kill.bat  
endlocal
```

```
:start  
ping %address% -w 2000 -n 1 | FIND "TTL" > nul  
IF NOT ERRORLEVEL 1 (
```

```

    if %change% == 2 (
REM verify it's up
timeout /nobreak /t 10 > nul
    ping %address% -w 5000 -n 1 | FIND "TTL" > nul
    IF NOT ERRORLEVEL 1 (
        set state=up
        set /a change = 1
        call :log
        call :message
    )
    )
cls
color 2
echo Ping Successful!
) else (
if %change% == 1 (
REM verify it's down
timeout /nobreak /t 10 > nul
    ping %address% -w 5000 -n 1 | FIND "TTL" > nul
    IF ERRORLEVEL 1 (
        set state=down
        set /a change = 2
        call :log
        call :message
    ) else (

        timeout 2 > nul
        color
        goto start
    )

)
cls
color 4
echo Ping Failed!
)
timeout /nobreak /t 2 > nul
color
goto start

:message
echo [System.Reflection.Assembly]::LoadWithPartialName^("System.Windows.Forms"^) ^| out-null
> "temp.ps1"
echo [System.Reflection.Assembly]::LoadWithPartialName^("System.Drawing"^) ^| out-null >>
"temp.ps1"
echo $Balloon = new-object System.Windows.Forms.NotifyIcon >> "temp.ps1"
echo $Balloon.Icon = [System.Drawing.SystemIcons]::WinLogo >> "temp.ps1"
echo $Balloon.Visible = $true; >> "temp.ps1"
echo $Balloon.ShowBalloonTip^(1, "Network", "Network changed state to %state%", "None"^); >>
"temp.ps1"
echo. >> "temp.ps1"
echo Unregister-Event -SourceIdentifier click_event -ErrorAction SilentlyContinue >> "temp.ps1"
echo Register-ObjectEvent $Balloon Balloon_TipClicked -sourceIdentifier click_event -Action { >>

```

```
"temp.ps1"  
echo. >> "temp.ps1"
```

REM THIS NAVIGATES TO A WEBSITE ON CLICK

```
echo $ie = New-Object -com InternetExplorer.Application >> "temp.ps1"  
echo #$ie.navigate2^("https://intranet.mydomain.com"^) >> "temp.ps1"
```

```
echo $ie.visible = $true >> "temp.ps1"  
echo. >> "temp.ps1"  
echo } ^| Out-Null >> "temp.ps1"  
echo. >> "temp.ps1"  
echo Wait-Event -timeout 5 -sourceIdentifier click_event > $null >> "temp.ps1"  
echo Remove-Event click_event -ea SilentlyContinue >> "temp.ps1"  
echo Unregister-Event -SourceIdentifier click_event -ErrorAction SilentlyContinue >> "temp.ps1"  
echo $Balloon.Dispose^(^) >> "temp.ps1"
```

```
powershell -file temp.ps1 >nul  
del temp.ps1 >nul  
exit /b
```

```
:log  
call :getTime  
echo. >> "%~dp0\NClogs\%logfile%"  
echo %address% %state%: %dtg% >> "%~dp0\NClogs\%logfile%"  
exit /b
```

```
:getTime  
set year=%date:~-4,4%  
set month=%date:~-10,2%  
set day=%date:~-7,2%  
set clock=%time:~0,3%%time:~3,2%  
if %day%==01 set day=1  
if %day%==02 set day=2  
if %day%==03 set day=3  
if %day%==04 set day=4  
if %day%==05 set day=5  
if %day%==06 set day=6  
if %day%==07 set day=7  
if %day%==08 set day=8  
if %day%==09 set day=9  
if %month%==01 set month=Jan  
if %month%==02 set month=Feb  
if %month%==03 set month=March  
if %month%==04 set month=April  
if %month%==05 set month=May  
if %month%==06 set month=June  
if %month%==07 set month=July  
if %month%==08 set month=Aug  
if %month%==09 set month=Sep  
if %month%==10 set month=Oct  
if %month%==11 set month=Nov  
if %month%==12 set month=Dec  
set dtg=%clock% %day% %month% %year%
```

exit /b

b00st3d - October 8, 2014, 1:17 pm

updated balloon popup to be more reliable. Added logging.

Code @echo off

cd %~dp0

if not exist "%~dp0NetCheck.bat" (

ren "%~f0" NetCheck.bat

timeout /nobreak /t 2 >nul

call "%~dp0NetCheck.bat"

)

mkdir NClogs > NUL

cls

set address=www.google.com

::set address=10.150.0.1

set year=%date:~-4,4%

set month=%date:~-10,2%

set day=%date:~-7,2%

set clock=%time:~0,2%%time:~3,2%

if %day%==01 set day=1

if %day%==02 set day=2

if %day%==03 set day=3

if %day%==04 set day=4

if %day%==05 set day=5

if %day%==06 set day=6

if %day%==07 set day=7

if %day%==08 set day=8

if %day%==09 set day=9

if %month%==01 set month=Jan

if %month%==02 set month=Feb

if %month%==03 set month=March

if %month%==04 set month=April

if %month%==05 set month=May

if %month%==06 set month=June

if %month%==07 set month=July

if %month%==08 set month=Aug

if %month%==09 set month=Sep

if %month%==10 set month=Oct

if %month%==11 set month=Nov

if %month%==12 set month=Dec

set dtg=%clock% %day% %month% %year%

set logfile=LOG_%dtg%.txt

<nul set /p ".=NetCheck is loading. Please wait"

:checkForFiles

if not exist invisible.vbs (

echo CreateObject^("Wscript.Shell"^).Run "***** ^& WScript.Arguments^(0^) ^& *****", 0, False >

invisible.vbs

<nul set /p .=.

timeout /nobreak /t 1 >nul

)

REM Search for powershellBypass

```

if not exist powershellBypass.bat (
echo @echo off > powershellBypass.bat
echo cls >> powershellBypass.bat
echo ^:CheckPrivileges >> powershellBypass.bat
echo NET FILE 1^>NUL 2^>NUL >> powershellBypass.bat
echo if '%errorlevel%' == '0' ^ ( goto gotPrivileges ^) else ^ ( goto getPrivileges ^) >>
powershellBypass.bat
echo. >> powershellBypass.bat
echo ^:getPrivileges >> powershellBypass.bat
echo if '%1'=='ELEV' ^ (shift ^& goto gotPrivileges^ ) >> powershellBypass.bat
ECHO echo. >> powershellBypass.bat
echo ECHO ***** >> powershellBypass.bat
echo ECHO This program requires admin permission to enable popups >> powershellBypass.bat
echo ECHO ***** >> powershellBypass.bat
echo. >> powershellBypass.bat
echo setlocal DisableDelayedExpansion >> powershellBypass.bat
echo set "batchPath=%~0" >> powershellBypass.bat
echo setlocal EnableDelayedExpansion >> powershellBypass.bat
echo ECHO Set UAC = CreateObject^("Shell.Application") ^> "%temp%\OEgetPrivileges.vbs"
>> powershellBypass.bat
echo ECHO UAC.ShellExecute "!batchPath!", "ELEV", "", "runas", 1 ^>^>
"%temp%\OEgetPrivileges.vbs" >> powershellBypass.bat
echo "%temp%\OEgetPrivileges.vbs" >> powershellBypass.bat
echo exit /B >> powershellBypass.bat
echo. >> powershellBypass.bat
echo ^:gotPrivileges >> powershellBypass.bat
echo ^:START >> powershellBypass.bat
echo setlocal ^& pushd . >> powershellBypass.bat
echo. >> powershellBypass.bat
echo powershell -command set-executionpolicy bypass >> powershellBypass.bat
<nul set /p .=.
timeout /nobreak /t 1 >nul

)
REM Check for start

if not exist start.bat (
echo @echo off > start.bat
echo if exist temp.ps1 del /f /q temp.ps1 >> start.bat
echo for /f "delims=" %%%%A in ^('powershell -command get-executionpolicy'^) do ^ ( >> start.bat
echo   set "GECP=%%%%A">> start.bat
echo ^) >> start.bat
echo echo Your current execution policy is: %GECP% >> start.bat
echo if not "%GECP%"=="Bypass" ^ ( >> start.bat
echo   call "powershellBypass.bat" >> start.bat
echo ^) >> start.bat
echo. >> start.bat
echo cls >> start.bat
echo cd %~dp0 >> start.bat
echo cls >> start.bat
echo wscript.exe "invisible.vbs" "NetCheck.bat" >> start.bat
echo cls >> start.bat

```

```
echo exit >> start.bat
<nul set /p .=.
timeout /nobreak /t 1 >nul
start start.bat
exit
)
```

```
mode con:cols=40 lines=5
set /A change = 2
:main
cls
```

```
title monitoring Network Connectivity
timeout /nobreak /t 2>nul
del /Q "%~dp0invisible.vbs"
del /Q "%~dp0powershellBypass.bat"
del /Q "%~dp0start.bat"
```

```
REM CREATE Script to kill this process
rem Note: Session Name for privileged Administrative consoles is sometimes blank.
if not defined SESSIONNAME set SESSIONNAME=Console
```

```
setlocal
```

```
rem Instance Set
set instance=%DATE% %TIME% %RANDOM%
::echo Instance: "%instance%"
title %instance%
```

```
rem PID Find
for /f "usebackq tokens=2" %%a in (`tasklist /FO list /FI "SESSIONNAME eq %SESSIONNAME%"
/FI "USERNAME eq %USERNAME%" /FI "WINDOWTITLE eq %instance%" ^| find /i "PID:") do set
PID=%%a
if not defined PID for /f "usebackq tokens=2" %%a in (`tasklist /FO list /FI "SESSIONNAME eq
%SESSIONNAME%" /FI "USERNAME eq %USERNAME%" /FI "WINDOWTITLE eq
Administrator: %instance%" ^| find /i "PID:") do set PID=%%a
if not defined PID echo !Error: Could not determine the Process ID of the current script. Exiting.&
exit /b 1
```

```
cd %~dp0
echo taskkill /pid %pid% > kill.bat
echo del "%~f0" >> kill.bat
endlocal
```

```
:start
ping %address% -w 2000 -n 1 | FIND "TTL" > nul
IF NOT ERRORLEVEL 1 (
    if %change% == 2 (
        REM verify it's up
        timeout /nobreak /t 10 > nul
        ping %address% -w 5000 -n 1 | FIND "TTL" > nul
        IF NOT ERRORLEVEL 1 (
            set state=up
```

```

        set /a change = 1
        call :log
        call :message
    )
)
cls
color 2
echo Ping Successful!
) else (
if %change% == 1 (
REM verify it's down
timeout /nobreak /t 10 > nul
ping %address% -w 5000 -n 1 | FIND "TTL" > nul
IF ERRORLEVEL 1 (
    set state=down
    set /a change = 2
    call :log
    call :message
) else (

        timeout 2 > nul
        color
        goto start
    )
)
cls
color 4
echo Ping Failed!
)
timeout /nobreak /t 2 > nul
color
goto start

:message
echo [System.Reflection.Assembly]::LoadWithPartialName^("System.Windows.Forms"^) ^| out-null
> "temp.ps1"
echo [System.Reflection.Assembly]::LoadWithPartialName^("System.Drawing"^) ^| out-null >>
"temp.ps1"
echo $Balloon = new-object System.Windows.Forms.NotifyIcon >> "temp.ps1"
echo $Balloon.Icon = [System.Drawing.SystemIcons]::WinLogo >> "temp.ps1"
echo $Balloon.Visible = $true; >> "temp.ps1"
echo $Balloon.ShowBalloonTip^(1, "Network", "Network changed state to %state%", "None"^); >>
"temp.ps1"
echo. >> "temp.ps1"
echo Unregister-Event -SourceIdentifier click_event -ErrorAction SilentlyContinue >> "temp.ps1"
echo Register-ObjectEvent $Balloon BalloonTipClicked -sourceIdentifier click_event -Action { >>
"temp.ps1"
echo. >> "temp.ps1"

REM THIS NAVIGATES TO A WEBSITE ON CLICK
echo $ie = New-Object -com InternetExplorer.Application >> "temp.ps1"
echo #$ie.navigate2^("https://intranet.mydomain.com"^) >> "temp.ps1"

```

```

echo $ie.visible = $true >> "temp.ps1"
echo. >> "temp.ps1"
echo } ^| Out-Null >> "temp.ps1"
echo. >> "temp.ps1"
echo Wait-Event -timeout 5 -sourceIdentifier click_event > $null >> "temp.ps1"
echo Remove-Event click_event -ea SilentlyContinue >> "temp.ps1"
echo Unregister-Event -SourceIdentifier click_event -ErrorAction SilentlyContinue >> "temp.ps1"
echo $Balloon.Dispose^(^)>> "temp.ps1"

```

```

powershell -file temp.ps1 >nul
del temp.ps1 >nul
exit /b

```

```

:log
echo %address% %state%: %dtg% >> "%~dp0\NClogs\%logfile%"
exit /b

```

b00st3d - November 3, 2014, 5:39 pm

Updated.

changes

When the balloon is clicked it will open the log. (Thanks Only3)

Removed some of the unnecessary spaces in the temp files (I forgot to remove them after testing)

Code @echo off

cd %~dp0

cls

if not exist "%~dp0NetCheck.bat" (

ren "%~f0" NetCheck.bat

timeout /nobreak /t 2 >nul

call "%~dp0NetCheck.bat"

)

mkdir NClogs > NUL

cls

set address=www.google.com

::set address=10.150.0.1

call :getTime

set logfile=LOG_%time:~0,2%%time:~3,2% %day% %month% %year%.txt

<nul set /p ".=NetCheck is loading. Please wait"

:checkForFiles

if not exist invisible.vbs (

echo CreateObject^("Wscript.Shell"^).Run "***** ^& WScript.Arguments^(0^) ^& *****", 0, False >

invisible.vbs

<nul set /p .=.

timeout /nobreak /t 1 >nul

)

REM Search for powershellBypass

```

if not exist powershellBypass.bat (
echo @echo off > powershellBypass.bat
echo cls >> powershellBypass.bat
echo ^:CheckPrivileges >> powershellBypass.bat
echo NET FILE 1^>NUL 2^>NUL >> powershellBypass.bat
echo if '%errorlevel%' == '0' ^& goto gotPrivileges ^) else ^& goto getPrivileges ^) >>
powershellBypass.bat
echo ^:getPrivileges >> powershellBypass.bat
echo if '%1'=='ELEV' ^&(shift ^& goto gotPrivileges^&) >> powershellBypass.bat
echo ECHO ***** >> powershellBypass.bat
echo ECHO This program requires admin permission to enable popups >> powershellBypass.bat
echo ECHO ***** >> powershellBypass.bat
echo setlocal DisableDelayedExpansion >> powershellBypass.bat
echo set "batchPath=%~0" >> powershellBypass.bat
echo setlocal EnableDelayedExpansion >> powershellBypass.bat
echo ECHO Set UAC = CreateObject^("Shell.Application") ^& "%temp%\OEgetPrivileges.vbs"
>> powershellBypass.bat
echo ECHO UAC.ShellExecute "!batchPath!", "ELEV", "", "runas", 1 ^&^&
"%temp%\OEgetPrivileges.vbs" >> powershellBypass.bat
echo "%temp%\OEgetPrivileges.vbs" >> powershellBypass.bat
echo exit /B >> powershellBypass.bat
echo ^:gotPrivileges >> powershellBypass.bat
echo ^:START >> powershellBypass.bat
echo setlocal ^& pushd . >> powershellBypass.bat
echo powershell -command set-executionpolicy bypass >> powershellBypass.bat
<nul set /p .=.
timeout /nobreak /t 1 >nul

)
REM Check for start

if not exist start.bat (
echo @echo off > start.bat
echo if exist temp.ps1 del /f /q temp.ps1 >> start.bat
echo for /f "delims=" %%%%A in ^('powershell -command get-executionpolicy'^) do ^& (>> start.bat
echo set "GECP=%%%%A">> start.bat
echo ^) >> start.bat
echo echo Your current execution policy is: %GECP% >> start.bat
echo if not "%GECP%"=="Bypass" ^& (>> start.bat
echo call "powershellBypass.bat" >> start.bat
echo ^) >> start.bat
echo cls >> start.bat
echo cd %~dp0 >> start.bat
echo cls >> start.bat
echo wscript.exe "invisible.vbs" "NetCheck.bat" >> start.bat
echo cls >> start.bat
echo exit >> start.bat
<nul set /p .=.
timeout /nobreak /t 1 >nul
start start.bat
exit
)

```

```
mode con:cols=40 lines=5
set /A change = 2
echo new log started: %dtg% >> "%~dp0\NClogs\%logfile%"
:main
cls
```

```
title monitoring Network Connectivity
timeout /nobreak /t 2>nul
del /Q "%~dp0invisible.vbs"
del /Q "%~dp0powershellBypass.bat"
del /Q "%~dp0start.bat"
```

```
REM CREATE Script to kill this process
rem Note: Session Name for privileged Administrative consoles is sometimes blank.
if not defined SESSIONNAME set SESSIONNAME=Console
```

```
setlocal
```

```
rem Instance Set
set instance=%DATE% %TIME% %RANDOM%
::echo Instance: "%instance%"
title %instance%
```

```
rem PID Find
for /f "usebackq tokens=2" %%a in (`tasklist /FO list /FI "SESSIONNAME eq %SESSIONNAME%"
/FI "USERNAME eq %USERNAME%" /FI "WINDOWTITLE eq %instance%" ^| find /i "PID:") do set
PID=%%a
if not defined PID for /f "usebackq tokens=2" %%a in (`tasklist /FO list /FI "SESSIONNAME eq
%SESSIONNAME%" /FI "USERNAME eq %USERNAME%" /FI "WINDOWTITLE eq
Administrator: %instance%" ^| find /i "PID:") do set PID=%%a
if not defined PID echo !Error: Could not determine the Process ID of the current script. Exiting.&
exit /b 1
```

```
cd %~dp0
echo taskkill /pid %pid% > kill.bat
echo del "%~f0" >> kill.bat
endlocal
```

```
:start
ping %address% -w 2000 -n 1 | FIND "TTL" > nul
IF NOT ERRORLEVEL 1 (
    if %change% == 2 (
        REM verify it's up
        timeout /nobreak /t 10 > nul
        ping %address% -w 5000 -n 1 | FIND "TTL" > nul
        IF NOT ERRORLEVEL 1 (
            set state=up
            set /a change = 1
            call :log
            call :message
        )
    )
)
```

```

cls
color 2
echo Ping Successful!
) else (
if %change% == 1 (
REM verify it's down
timeout /nobreak /t 10 > nul
ping %address% -w 5000 -n 1 | FIND "TTL" > nul
IF ERRORLEVEL 1 (
set state=down
set /a change = 2
call :log
call :message
) else (

        timeout 2 > nul
        color
        goto start
    )
)
cls
color 4
echo Ping Failed!
)
timeout /nobreak /t 2 > nul
color
goto start

:message
echo [System.Reflection.Assembly]::LoadWithPartialName^("System.Windows.Forms"^) ^| out-null
> "temp.ps1"
echo [System.Reflection.Assembly]::LoadWithPartialName^("System.Drawing"^) ^| out-null >>
"temp.ps1"
echo $Balloon = new-object System.Windows.Forms.NotifyIcon >> "temp.ps1"
echo $Balloon.Icon = [System.Drawing.SystemIcons]::WinLogo >> "temp.ps1"
echo $Balloon.Visible = $true; >> "temp.ps1"
echo $Balloon.ShowBalloonTip(1, "Network", "Network changed state to %state%", "None"^); >>
"temp.ps1"
echo Unregister-Event -SourceIdentifier click_event -ErrorAction SilentlyContinue >> "temp.ps1"
echo Register-ObjectEvent $Balloon BalloonTipClicked -sourceIdentifier click_event -Action
{Invoke-Item "%~dp0\NClogs\%logfile%"} ^| Out-Null >> "temp.ps1"

REM THIS NAVIGATES TO A WEBSITE ON CLICK
REM echo $ie = New-Object -com InternetExplorer.Application >> "temp.ps1"
REM echo $ie.navigate2^("http://www.google.com"^) >> "temp.ps1"
REM echo $ie.visible = $true >> "temp.ps1"
REM echo } ^| Out-Null >> "temp.ps1"

echo Wait-Event -timeout 5 -sourceIdentifier click_event > $null >> "temp.ps1"
echo Remove-Event click_event -ea SilentlyContinue >> "temp.ps1"
echo Unregister-Event -SourceIdentifier click_event -ErrorAction SilentlyContinue >> "temp.ps1"
echo $Balloon.Dispose(^) >> "temp.ps1"

```

```
powershell -sta -file temp.ps1 >nul
del temp.ps1 >nul
exit /b
```

```
:log
call :getTime
echo. >> "%~dp0\NClogs\%logfile%"
echo %address% %state%: %dtg% >> "%~dp0\NClogs\%logfile%"
exit /b
```

```
:getTime
set year=%date:~-4,4%
set month=%date:~-10,2%
set day=%date:~-7,2%
set clock=%time:~0,3%%time:~3,2%
if %day%==01 set day=1
if %day%==02 set day=2
if %day%==03 set day=3
if %day%==04 set day=4
if %day%==05 set day=5
if %day%==06 set day=6
if %day%==07 set day=7
if %day%==08 set day=8
if %day%==09 set day=9
if %month%==01 set month=Jan
if %month%==02 set month=Feb
if %month%==03 set month=March
if %month%==04 set month=April
if %month%==05 set month=May
if %month%==06 set month=June
if %month%==07 set month=July
if %month%==08 set month=Aug
if %month%==09 set month=Sep
if %month%==10 set month=Oct
if %month%==11 set month=Nov
if %month%==12 set month=Dec
set dtg=%clock% %day% %month% %year%
exit /b
```

gogglebot - November 4, 2014, 6:33 am

This is looking pretty cool Boost3d! I might keep a copy on a memory stick!
Good work. :D

Not sure about getting rid of the icon being left behind, it seems to be something to do with the attached process stopping and not removing it's icon on the notifications bar. I've seen it happen with other things. Hovering over it seems to update the bar. Unless there's a way to update the bar by other means I'm not sure what to do.

b00st3d - November 4, 2014, 5:14 pm

(Nov 4, 2014 07:33 AM)gogglebot Wrote: [\[CITE\]](#) This is looking pretty cool Boost3d! I might keep a copy on a memory stick!
Good work. :D

Not sure about getting rid of the icon being left behind, it seems to be something to do with the attached process stopping and not removing it's icon on the notifications bar. I've seen it happen with other things. Hovering over it seems to update the bar. Unless there's a way to update the bar by other means I'm not sure what to do.

The icon should go away in this version. It wasn't going away because the powershell process wasn't being closed properly.

notM9 - November 4, 2014, 6:45 pm

This seems really nice actually. I have no use for it, but wow. Incredible work on the coding. I used to love batch when i was 13/14. I loved it.

b00st3d - November 4, 2014, 9:09 pm

(Nov 4, 2014 07:45 PM)Mulato Wrote: [\[CITE\]](#) This seems really nice actually. I have no use for it, but wow. Incredible work on the coding. I used to love batch when i was 13/14. I loved it.

Glad you like it. I work with batch mostly out of necessity at work but sometimes a need arises at home so I will make things like this.

b00st3d - April 12, 2015, 9:22 pm

I just finished making some updates to this script.

Changes:

- *I didn't like that it was hard coded to monitor a single address so I changed it to ask for user input on what you would like to monitor. To do this I added a pretty cool vbs input box that returns a string.

- *Each instance of the program will have an easily identifiable batch file to kill the process.

- *NetCheck_cleanup.bat will also be generated to automatically run all the kill scripts at the same time.

- *changed the naming scheme of the logs generated so each device writes it's own log. I may change this in the future but don't quote me on that.

Code @echo off

cd %~dp0

cls

if not exist "%~dp0NetCheck.bat" (

ren "%~f0" NetCheck.bat

timeout /nobreak /t 2 >nul

call "%~dp0NetCheck.bat"

)

mkdir NClogs 2> NUL

cls

<nul set /p ".=NetCheck is loading. Please wait"

:checkForFiles

REM Search for invisible

if not exist invisible.vbs (

echo CreateObject^("Wscript.Shell"^).Run "'''''' ^& WScript.Arguments^(0^) ^& ''''''", 0, False >

invisible.vbs

<nul set /p ".=.

timeout /nobreak /t 1 >nul

)

```

REM Search for powershellBypass
if not exist powershellBypass.bat (
(
echo @echo off
echo cls
echo ^:CheckPrivileges
echo NET FILE 1^>NUL 2^>NUL
echo if '%errorlevel%' == '0' ^( goto gotPrivileges ^) else ^( goto getPrivileges ^)
echo ^:getPrivileges
echo if '%1'=='ELEV' ^(shift ^& goto gotPrivileges^)
echo setlocal DisableDelayedExpansion
echo set "batchPath=%~0"
echo setlocal EnableDelayedExpansion
echo ECHO Set UAC = CreateObject^("Shell.Application") ^> "%temp%\OEgetPrivileges.vbs"
echo ECHO UAC.ShellExecute "!batchPath!", "ELEV", "", "runas", 1 ^>^>
"%temp%\OEgetPrivileges.vbs"
echo "%temp%\OEgetPrivileges.vbs"
echo exit /B
echo ^:gotPrivileges
echo ^:START
echo setlocal ^& pushd .
echo powershell -command set-executionpolicy bypass
) > powershellBypass.bat
<nul set /p .=.
timeout /nobreak /t 1 >nul

```

)

```

REM Search for cleanup
if not exist NetCheck_Cleanup.bat (
(
echo @echo off
echo cd %~dp0
echo if not exist "%~dp0NetCheck_cleanup.bat" ^(
echo ren "%~f0" NetCheck_cleanup.bat
echo timeout /nobreak /t 2 ^>nul
echo call "%~dp0NetCheck_cleanup.bat"
echo ^)
echo for /f %%%a in ^(dir /b/a/d ^^^| find "kill"^) DO ^(
echo call "%%~f0" /q
echo ^)
echo del "%~f0" /q
) >NetCheck_Cleanup.bat
<nul set /p .=.
timeout /nobreak /t 1 >nul
)

```

REM Check for start

```

if not exist start.bat (
(
echo @echo off

```

```

echo if exist temp.ps1 del /f /q temp.ps1
echo for /f "delims=" %%%%A in ^('powershell -command get-executionpolicy'^) do ^(
echo   set "GECP=%%%%%%A"
echo ^)
echo echo Your current execution policy is: %%GECP%%
echo if not "%%GECP%%"=="Bypass" ^(
echo   call "powershellBypass.bat"
echo ^)
echo cls
echo cd %%~dp0
echo cls
echo wscript.exe "invisible.vbs" "NetCheck.bat"
echo cls
echo exit
) > start.bat
<nul set /p .=.
timeout /nobreak /t 1 >nul
start start.bat
exit
)
call :inputbox "Please enter a website or computername to monitor" "Network Monitor"
set address=%input%

call :getTime
set logfile=LOG_%%address%.txt

mode con:cols=40 lines=5

set /A change = 2
(
echo.
echo=====
echo new log started: %dtg%
) >> "%~dp0\NClogs\%logfile%"
:main
cls

title Monitoring Network Connectivity
timeout /nobreak /t 2>nul
del /Q "%~dp0invisible.vbs"
del /Q "%~dp0powershellBypass.bat"
del /Q "%~dp0start.bat"

REM CREATE Script to kill this process
rem Note: Session Name for privileged Administrative consoles is sometimes blank.
if not defined SESSIONNAME set SESSIONNAME=Console

setlocal

rem Instance Set
set instance=%DATE% %TIME% %RANDOM%
title %instance%

```

```

rem PID Find
for /f "usebackq tokens=2" %%a in (`tasklist /FO list /FI "SESSIONNAME eq %SESSIONNAME%"
/FI "USERNAME eq %USERNAME%" /FI "WINDOWTITLE eq %instance%" ^| find /i "PID:") do set
PID=%%a
if not defined PID for /f "usebackq tokens=2" %%a in (`tasklist /FO list /FI "SESSIONNAME eq
%SESSIONNAME%" /FI "USERNAME eq %USERNAME%" /FI "WINDOWTITLE eq
Administrator: %instance%" ^| find /i

"PID:") do set PID=%%a
if not defined PID echo !Error: Could not determine the Process ID of the current script. Exiting.&
exit /b 1

cd %~dp0
echo taskkill /pid %pid% > kill_%address%.bat
echo del "%~f0" >> kill_%address%.bat
endlocal

:start
ping -4 %address% -w 2000 -n 1 | FIND "TTL" > nul
IF NOT ERRORLEVEL 1 (
    if %change% == 2 (
        REM verify it's up
        timeout /nobreak /t 10 > nul
        ping %address% -w 5000 -n 1 | FIND "TTL" > nul
        IF NOT %ERRORLEVEL%==1 (
            set state=is online
            set /a change = 1
            call :log
            call :message
        )
    )
    cls
    color 2
    echo Ping Successful!
) else (
    if %change% == 1 (
        REM verify it's down
        timeout /nobreak /t 15 > nul
        ping -4 %address% -w 5000 -n 1 | FIND "TTL" > nul
        IF %ERRORLEVEL%==1 (
            set state=could not be reached
            set /a change = 2
            call :log
            call :message
        ) else (
            timeout 2 > nul
            color
            goto start
        )
    )
    cls

```

```
color 4
echo Ping Failed!
)
timeout /nobreak /t 2 > nul
color
goto start
```

```
:message
```

REM This section originally posted to hackforums.net by Johnny Cache and modified for my purposes here

```
echo [System.Reflection.Assembly]::LoadWithPartialName^("System.Windows.Forms"^) ^| out-null
> "temp.ps1"
echo [System.Reflection.Assembly]::LoadWithPartialName^("System.Drawing"^) ^| out-null >>
"temp.ps1"
echo $Balloon = new-object System.Windows.Forms.NotifyIcon >> "temp.ps1"
echo $Balloon.Icon = [System.Drawing.SystemIcons]::WinLogo >> "temp.ps1"
echo $Balloon.Visible = $true; >> "temp.ps1"
echo $Balloon.ShowBalloonTip^(1, "Network", "%address% %state%", "None"^); >> "temp.ps1"
echo Unregister-Event -SourceIdentifier click_event -ErrorAction SilentlyContinue >> "temp.ps1"
echo Register-ObjectEvent $Balloon BalloonTipClicked -sourceIdentifier click_event -Action
{Invoke-Item "%~dp0\NClogs\%logfile%"} ^| Out-Null >> "temp.ps1"
```

REM THIS NAVIGATES TO A WEBSITE ON CLICK I LEFT IT HERE FOR FUTURE EXPANSION

```
REM echo $ie = New-Object -com InternetExplorer.Application >> "temp.ps1"
REM echo $ie.navigate2^("http://www.google.com"^) >> "temp.ps1"
REM echo $ie.visible = $true >> "temp.ps1"
REM echo } ^| Out-Null >> "temp.ps1"
```

```
echo Wait-Event -timeout 5 -sourceIdentifier click_event > $null >> "temp.ps1"
echo Remove-Event click_event -ea SilentlyContinue >> "temp.ps1"
echo Unregister-Event -SourceIdentifier click_event -ErrorAction SilentlyContinue >> "temp.ps1"
echo $Balloon.Dispose^(^) >> "temp.ps1"
```

```
powershell -sta -file temp.ps1 >nul
del temp.ps1 >nul
exit /b
```

```
:log
call :getTime
echo. >> "%~dp0\NClogs\%logfile%"
echo %address% %state%: %dtg% >> "%~dp0\NClogs\%logfile%"
exit /b
```

```
:getTime
set year=%date:~-4,4%
set month=%date:~-10,2%
set day=%date:~-7,2%
set clock=%time:~0,3%%time:~3,2%
if %day%==01 set day=1
if %day%==02 set day=2
if %day%==03 set day=3
if %day%==04 set day=4
```

```
if %day%==05 set day=5
if %day%==06 set day=6
if %day%==07 set day=7
if %day%==08 set day=8
if %day%==09 set day=9
if %month%==01 set month=Jan
if %month%==02 set month=Feb
if %month%==03 set month=March
if %month%==04 set month=April
if %month%==05 set month=May
if %month%==06 set month=June
if %month%==07 set month=July
if %month%==08 set month=Aug
if %month%==09 set month=Sep
if %month%==10 set month=Oct
if %month%==11 set month=Nov
if %month%==12 set month=Dec
set dtg=%clock% %day% %month% %year%
exit /b
```

:InputBox

REM Input routine for batch using VBScript to provide input box

REM Stephen Knight, October 2009, <http://www.dragon-it.co.uk/>

set input=

set heading=%~2

set message=%~1

echo wscript.echo inputbox(WScript.Arguments(0),WScript.Arguments(1)) >"%temp%\input.vbs"

for /f "tokens=* delims=" %%%a in ('cscript //nologo "%temp%\input.vbs" "%message%"

"%heading%") do set input=%%a

exit /b