**User Simulation Script Documentation**

**How to use the script for a local run:**

-Copy the folder contaning the relevant files into your test system.

-Modify settings at your convenience by Right-Click on 'Config.ps1', 'Open with' and 'Wordpad'.

-Right-click "RunUserSimulation.ps1". "Run with powershell"

-The logs are in the Logs folder, named by date/time.

**How to use the script for a multisession run:**

-See "setup the environment" first of all.

-Right-click "RunUserSimulation.ps1". "Run with powershell", and accept the UAC prompt

**Setup the environment (for remote/multi-users only)**

**(if you plan to clone VMs, do this first. If you already have many VMs, you have to follow these steps for each one)**

**Activate script execution:**

-To authorize script execution (needed if you are not yourself manually executing the script), you need to enter "Set-ExecutionPolicy -Scope CurrentUser Bypass" in a powershell prompt , and answer A (yes for all) to the question

-(you can open such a prompt via 'Windows + X' then 'windows powershell'

**Setup Virtual Machines (Network):**

- Connect VM(s) on a common network

- (In guests)Disable the firewall for public network

- turn on network discovery ()public networks)

- If you already have many VMs, ensure each machine has a different name, IP (enable DCHP or see 'static IP attribution') and MAC address (!)

- In case you choose to manually name your PCs, do NOT name them "Local", "All","First" or "Random" (or any case variation), that will conflict will the schedule feature. (see 'schedule')

- If you feel tinkering and changing precise firewall rules, it is known that powershell remoting uses TCP ports 5985 and 5986.

**Setup Network (static IP attribution):**

- In guest OS control panel / network&sharing center / change adapter settings / ethernet / properties / IPv4 / properties, check "use the following IP" and fill the field with an unique IP, (common) network mask and default gateway

Eg:

IP: 192.168.0.10

mask : 255.255.255.0

default gateway : 192.168.0.1

**Setup Network (DCHP with VirtualBox):**

- From the host OS, open a shell, go to (change directory to) the virtualbox software directory (usually in "C\programm files x86\Oracle\Virtualbox\" in Windows)

- Run the following command (one-line): (if you know what you are doing you can change these parameters)

.\VBoxManage dhcpserver add -netname intnet -ip 10.13.13.100 -netmask 255.255.255.0 -lowerip 10.13.13.101 -upperip 10.13.13.254 -enable

enter-pssesion

**Enable needed rules (for powershell remoting):**

- Run twice the EnableEverything.ps1 script (right-click / run in powershell, then accept the UAC prompt)

See "Configure the VMs" if the defaults settings for the modules are not what you want, as modules options have to be configured separately

**(you can clone VMs now. Be sure that in the end VMs do NOT have the same PC NAME, IP ADDRESS and MAC ADDRESS)**

**Booting the VMs:**

- Boot one VM first (if you have static IPs, choose the one with the lowest IP just in case).This will be our master VM

- Once the master VM has finished to boot and you have logged in, boot the other VMs.

- From the master VM, run the "RunUserSimulation" script as administrator.

**Enable multisession in configuration file:**

- In the master VM only: Open the Config.ps1 file and in Global Config change the value of : "global:global\_localonly =" from '$true' to '$false'

**Configure the VMs:**

You MUST have a copy of the 'user simulation script' folder in each VM at the same path.

The global configuration option are general, that means you can change the behavior of all PCs from the master VM.

The module configuration options only affects the local PC which means you can have PCs with different browsing, mapping shares etc... options.

Modify these to suit your needs

Note:

The script can manage to connect to an infinity of network computers. You should check that the hardware specs of the master VM / host PC are enough for your network topology. 32 networks PCs is the "safe" limit

**You are done!**

**Enable Use the scheduling feature:**

- You must enable the $global\_schedule in config

- Modify the schedule.txt file:

The first line MUST always be this one, then write what you want line by line:

**Stop** **All** activity on **All** **after** **00:01**

(Keywords in bold, not case-sensitive)

Use the same structure to construct orders

**Stop** : Accepted values 'Start' and 'Stop'

**All** : What kind of activity to enable/disable

Values: 'All', 'IE', 'MapShare', 'Type' ....

activity

on

**All** : Which PC to start activity on

Values: 'Local' (for local simulation), 'All', 'First' (only one PC), 'Random' (random PCs),

or any of your computer names (eg DESKTOP-MUJ56)

**after** : start this activity after the given time

Values: 'after', 'at' (give the clock hour/min at which to start activity)

**09:15 : HH:mm** is the time (24h format) after/at which it do something, here it's 9h15

- You must not enter twice the same line

- Due to the way it works, the program may need up to 1 minute to start executing your orders

**What this script do?**

This script repeatedly run 'rounds'. Rounds are made of several activities (such as IE browsing or network shares mapping when enabled). Those activities run one after the other, until the round completes.

Before starting the next round, the script wait for an 'un-activity' time specified in the configuration file.

**Run at startup: #not working**

To run the programm at user login you should:

-Create a shortcut of the RunUserSimulation.ps1 file

-cut/paste the shortcut into the startup folder, which you can access by 'Windows + R' , 'shell:startup'

-Be sure to have activated script execution as mentionned below

This script should be the only thing running on the computer. Else it can break stuff as it is made to interact with environmemt and open/close things such as Internet Explorer, by force if necessary (eg to prevent clutter if some windows ask for confirmation before closing).