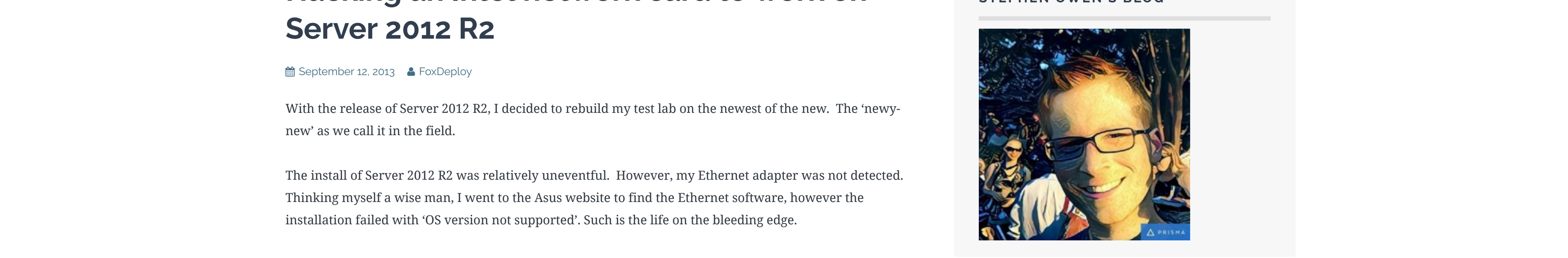


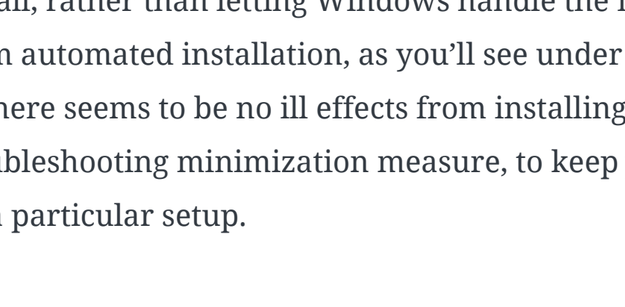
Hacking an Intel network card to work on Server 2012 R2

September 12, 2013 FoxDeploy

With the release of Server 2012 R2, I decided to rebuild my test lab on the newest of the new. The 'newy-new' as we call it in the field.

The install of Server 2012 R2 was relatively uneventful. However, my Ethernet adapter was not detected. Thinking myself a wise man, I went to the Asus website to find the Ethernet software, however the installation failed with 'OS version not supported'. Such is the life on the bleeding edge.

I then went to the Intel site to download the NIC directly, but this install failed with 'Cannot Install Drivers. No Intel Adapters are present in this computer'. Huh? Well, being resourceful, I began Googling furiously. I found a number of threads of chatter around related issues for other components (all cited below), and after some experimentation, I got it all working.



This message was not enough to stop me. I know there is an Intel NIC inside!

The thing to note here is that the problem is cause when Intel's software attempts to automate the device install, rather than letting Windows handle the installation. They go so far as to exclude certain device types from automated installation, as you'll see under the [ControlFlags] section. Who knows why this was done, as there seems to be no ill effects from installing this on a Server 2012 R2 system. It is likely a troubleshooting minimization measure, to keep people from installing drivers for devices that won't work on a particular setup.

We will not be so easily swayed. :). Here is how to alter the .inf files for a device and then put Windows into a test mode which will allow us to force this modified driver into being installed. You will need:

- The device drivers for your particular device.
- Willingness to tamper with and seek a deeper understanding of your computer system.
- To take ownership of the fact that things could go wrong in an unexpected manner, and it is not my fault or duty to help you fix it.

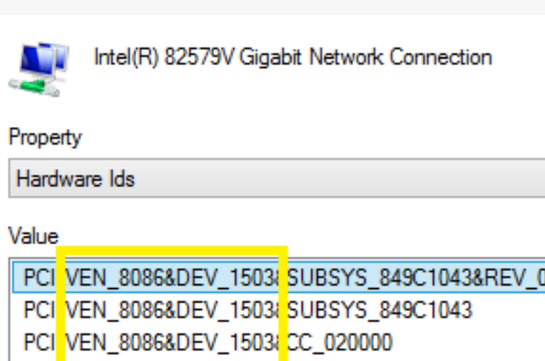
First, enable test mode and disable driver signing in order to install these hacked drivers. We'll need to do this because Windows requires signed drivers and will note that our changes to the driver files that we are about to make.

From an Administrative Command prompt:

bcdedit -set loadoptions DISABLE_INTEGRITY_CHECKS

bcdedit -set TESTSIGNING ON

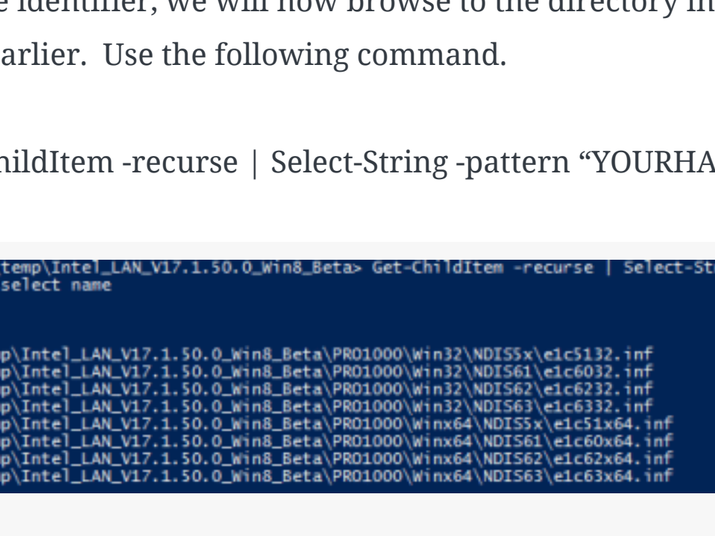
Now, restart the system. As you see, we are now 'Flying Dirty'



This is Microsoft's way of reminding you that you're doing something that you shouldn't be doing.

Open your driver download and extract it somewhere (in my case for an Asus p8z77-v with an Intel 82579V Gigabit NIC). I've placed my files at C:\temp\Intel_LAN_V17.1.50.0_Win8_Beta\PRO1000\Winx64\NDIS63. You'll want to open the various .inf files until you find the one that contains your Hardware IDs.

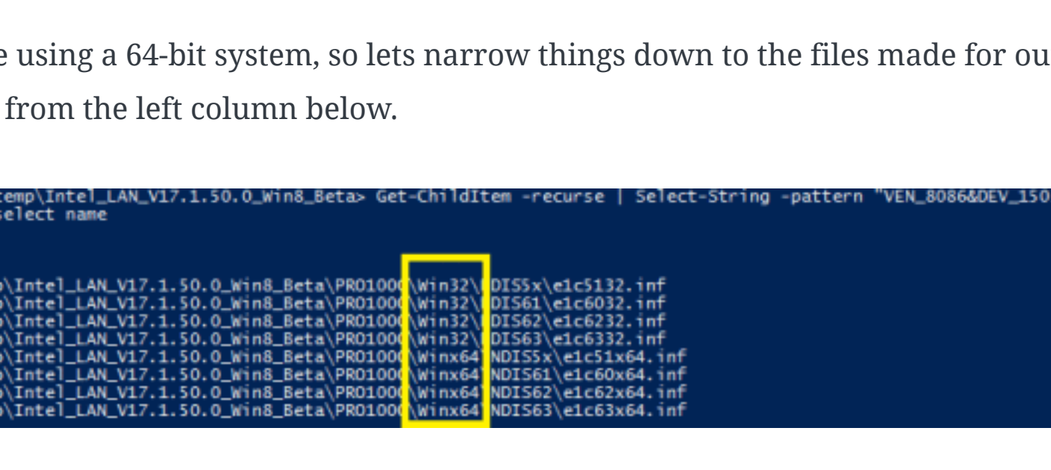
If you'd like to automate tracking down the desired driver(wouldn't you!), open up device manager and select the malfunctioning device. Go to the Details Pane and Select Hardware IDs.



Please ignore the fact that the name is detected successfully, I forgot to take a screen shot before I fixed the issue. If you're reading this, your screen will generally look much more jacked up, with frowning device faces everywhere.

We'll use 'VEN_8086&DEV_1503'. This may likely be different on your system. Armed with our Vendor and device identifier, we will now browse to the directory in PowerShell where we placed the unzipped install files earlier. Use the following command.

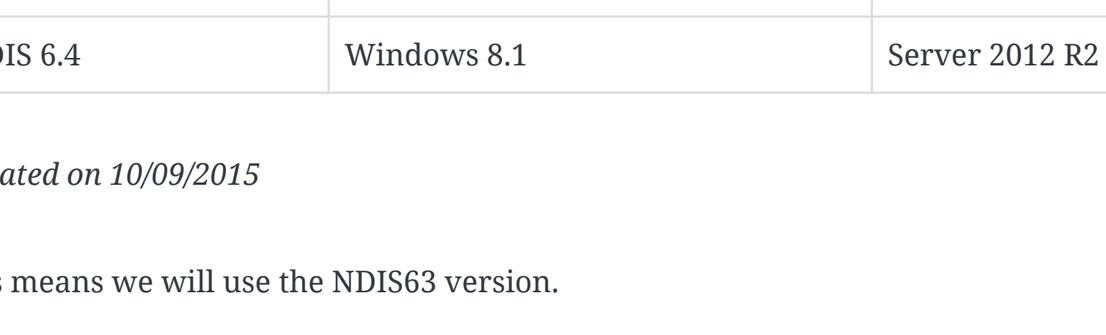
Get-ChildItem -recurse | Select-String -pattern "YOURHARDWAREID" | group path | select name



*So nicely sorted. Imagine if we didn't know Powershell and had to manually do all of this *shudder**

This is giving us an output of every .inf file which mentions our device ID by name. This is very powerful and is saving us a ton of time finding the right file for our situation. The next few bits just depend on us knowing a thing or two about the system we're using and a bit about Driver Specification history (everyone's favorite topic!)

We're using a 64-bit system, so lets narrow things down to the files made for our architecture, as seen in the third from the left column below.

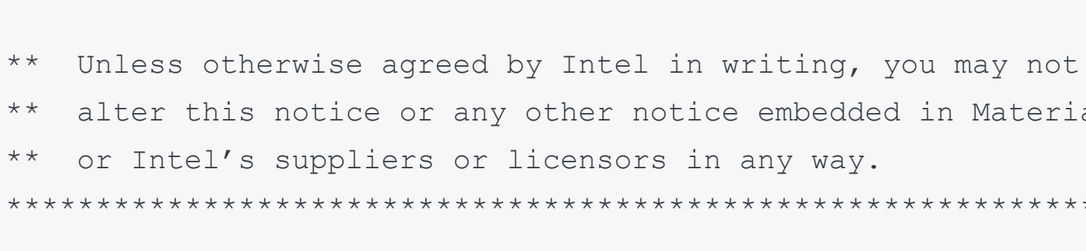


Then the Network Driver Interface Specification (NDIS) that matches our OS, for reference:

Version	Desktop OS	Server OS
NDIS 6.0	Vista	*
NDIS 6.1	Vista SP 1	Server 2008
NDIS 6.2	Windows 7	Server 2008 R2
NDIS 6.3	Windows 8	Server 2012
NDIS 6.4	Windows 8.1	Server 2012 R2

Updated on 10/09/2015

This means we will use the NDIS63 version.



This has narrowed it down to only one file, the e1c63x64.inf file. If you're a do-it-yourselfer, you can just paste my entry below, taking care to match the section heading. if you want to know more, check the links below for more detail as to why this works. The mile-high view is that we're removing the exclusions for our device from Windows Auto Detection, and then copying some bits around. Or something like that. Seriously check the links if you have a burning curiosity.

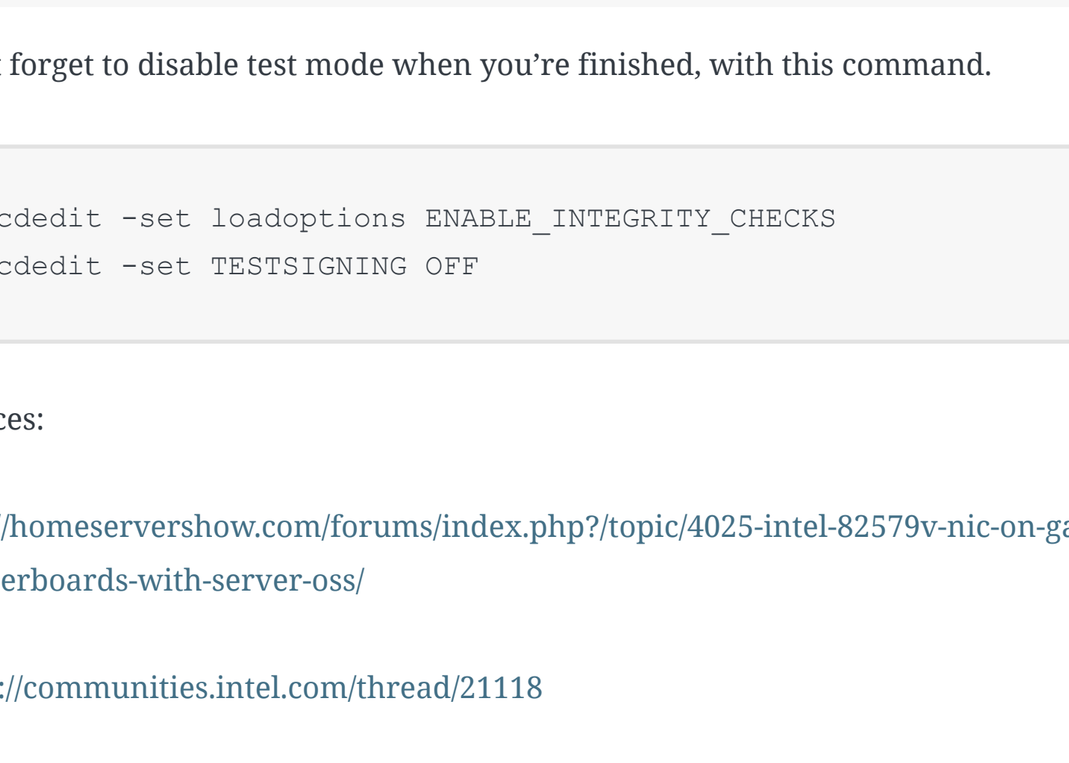
<——BEGIN COPY————>

```
;**  Unless otherwise agreed by Intel in writing, you may not remove or      **
;**  alter this notice or any other notice embedded in Materials by Intel    **
;**  or Intel's suppliers or licensors in any way.                            **
;*****************************************************************************
;
;*****************************************************************************

; e1c63x64.INF (Intel 64 bit extension Platform Only,
; Windows 8 64 bit extension)

;
; Intel(R) Gigabit Network connections
;*****************************************************************************
;

[Version]
Signature   = "$Windows NT$"
Class       = Net
ClassGUID   = {4d36e972-e325-11ce-bfc1-08002be10318}
Provider    = %Intel%
CatalogFile = e1c63x64.cat
DriverVer   = 03/29/2012,12.1.10.0

[Manufacturer]
%Intel%     = Intel, NTamd64.6.2, NTamd64.6.2.1

[ControlFlags]
;ExcludeFromSelect = \
;    PCI\VEN_8086&DEV_1502,\
;    PCI\VEN_8086&DEV_1503

[Intel]

[Intel.NTamd64.6.2.1]
; DisplayName                   Section              DeviceID
; -----------                   -------              --------
%E1502NC.DeviceDesc%            = E1502.6.2.1,       PCI\VEN_8086&DEV_1502
%E1502NC.DeviceDesc%            = E1502.6.2.1,
PCI\VEN_8086&DEV_1502&SUBSYS_00011179
%E1502NC.DeviceDesc%            = E1502.6.2.1,
PCI\VEN_8086&DEV_1502&SUBSYS_00021179
%E1502NC.DeviceDesc%            = E1502.6.2.1,
PCI\VEN_8086&DEV_1502&SUBSYS_80001025
%E1503NC.DeviceDesc%            = E1503.6.2.1,       PCI\VEN_8086&DEV_1503
%E1503NC.DeviceDesc%            = E1503.6.2.1,
PCI\VEN_8086&DEV_1503&SUBSYS_00011179
%E1503NC.DeviceDesc%            = E1503.6.2.1,
PCI\VEN_8086&DEV_1503&SUBSYS_00021179
%E1503NC.DeviceDesc%            = E1503.6.2.1,
PCI\VEN_8086&DEV_1503&SUBSYS_80001025
%E1503NC.DeviceDesc%            = E1503.6.2.1,
PCI\VEN_8086&DEV_1503&SUBSYS_04911025

[Intel.NTamd64.6.2]
; DisplayName                   Section        DeviceID
; -----------                   -------        --------
%E1502NC.DeviceDesc%            = E1502,       PCI\VEN_8086&DEV_1502
%E1502NC.DeviceDesc%            = E1502,
PCI\VEN_8086&DEV_1502&SUBSYS_00011179
%E1502NC.DeviceDesc%            = E1502,
PCI\VEN_8086&DEV_1502&SUBSYS_00021179
%E1502NC.DeviceDesc%            = E1502,
PCI\VEN_8086&DEV_1502&SUBSYS_80001025
%E1503NC.DeviceDesc%            = E1503.6.2.1,       PCI\VEN_8086&DEV_1503
%E1503NC.DeviceDesc%            = E1503.6.2.1,
PCI\VEN_8086&DEV_1503&SUBSYS_00011179
%E1503NC.DeviceDesc%            = E1503.6.2.1,
PCI\VEN_8086&DEV_1503&SUBSYS_00021179
%E1503NC.DeviceDesc%            = E1503.6.2.1,
PCI\VEN_8086&DEV_1503&SUBSYS_80001025
%E1503NC.DeviceDesc%            = E1503.6.2.1,
PCI\VEN_8086&DEV_1503&SUBSYS_04911025

;=============================================================================

;                 WINDOWS 8 for 64-bit EXTENDED PLATFORMS

;
;=============================================================================
```

<————Please stop copying————>

If you're the trusting sort, you can just click this link for the Complete inf https://dl.dropboxusercontent.com/u/6268163/e1c63x64.inf

Now, go back to device manager, select the errant device and Update Driver, pointing it to the directory that contains our modified driver. If all went well, you should see the device begin to install automatically!



Yay, Happy Device Manager. 'Whats up with that missing 'e-home transceiver?' Um... this is more of a do as I say, not as I do situation.

Don't forget to disable test mode when you're finished, with this command.

```
bcdedit -set loadoptions ENABLE_INTEGRITY_CHECKS
bcdedit -set TESTSIGNING OFF
```

Sources:

http://homeservershow.com/forums/index.php?/topic/4025-intel-82579v-nic-on-ga-z77x-ud5h-and-other-motherboards-with-server-oss/

https://communities.intel.com/thread/21118

http://laslow.net/2012/03/14/disable-driver-signature-enforcement-in-windows-8-cp/

sixcolumns.com/t/how-to-disable-and-bypass-driver-signature-enforcement-in-windows-8/151/2

Related

Solving the DSC Pull Chicken and Egg problem
In "Scripting"

Recovering your Dedeuped Files on Windows 10
In "Scripting"

SCCM 1606 Cloud Proxy Guide
In "SCCM"

Other 2012R2, PowerShell

PREVIOUS ARTICLE
SCCM – Dealing with updates that won't download

NEXT ARTICLE
SCCM Report – All Packages, Programs and their command lines

91 THOUGHTS ON "HACKING AN INTEL NETWORK CARD TO WORK ON SERVER 2012 R2"

KIRAN
May 24, 2017 / 11:16 am

I just wanted to thank you for this info. Trying to run Windows Server 2012 or 2016 on an AsRock Z270M Pro4 motherboard with the I219-V and no driver would install. Followed your instructions and it worked perfectly!

Like

Reply

FOXDEPLOY
May 25, 2017 / 1:09 pm

Awesome! Happy to hear it! I was really upset when I heard Microsoft would pass WHQL anymore for some configurations of hardware. Happy these steps still work!

Like

Reply

DENVER
July 27, 2017 / 10:55 am

Would this method work on the Asus X99-A II with Windows Server 2016 Essentials?

Like

Reply

FOXDEPLOY
July 27, 2017 / 10:40 am

Absolutely, when you add new drivers, Windows scans through the inf files to see if the driver says it supports that OS version. Normally, there is little distinction between a desktop or a server OS so our steps here assume it will work, and we simply add a new entry for our OS. There is no reason it would not work for you

Like

Reply

ALEXANDER
September 7, 2017 / 6:22 pm

Hello!

Thanks for your information, the same problem: winserver 2012R2 + z270 chipset.
process a bit differs from yours one but it works now!

have a good time!

Like

Reply

Pingback: دانلود Windows Server 2012 R2 – مجله کلیک

THUAN
February 8, 2018 / 10:22 am

Hi,

I've been so frustrated unable to see the "Cannot install.." error while installing Intel I219-V onto a Prime H270M-Plus motherboard running Windows Server 2016 Standard in Core mode. Just read your article and hopefully will be able to "hack" it to work. Will keep you posted, many thanks.

Liked by 2 people

Reply

FOXDEPLOY
February 12, 2018 / 5:03 pm

Happy it helped!

Liked by 1 person

Reply

← OLDER COMMENTS

Have a code issue? Share your code by going to Gist.github.com and pasting your code there, then post the link here!

Enter your comment here...

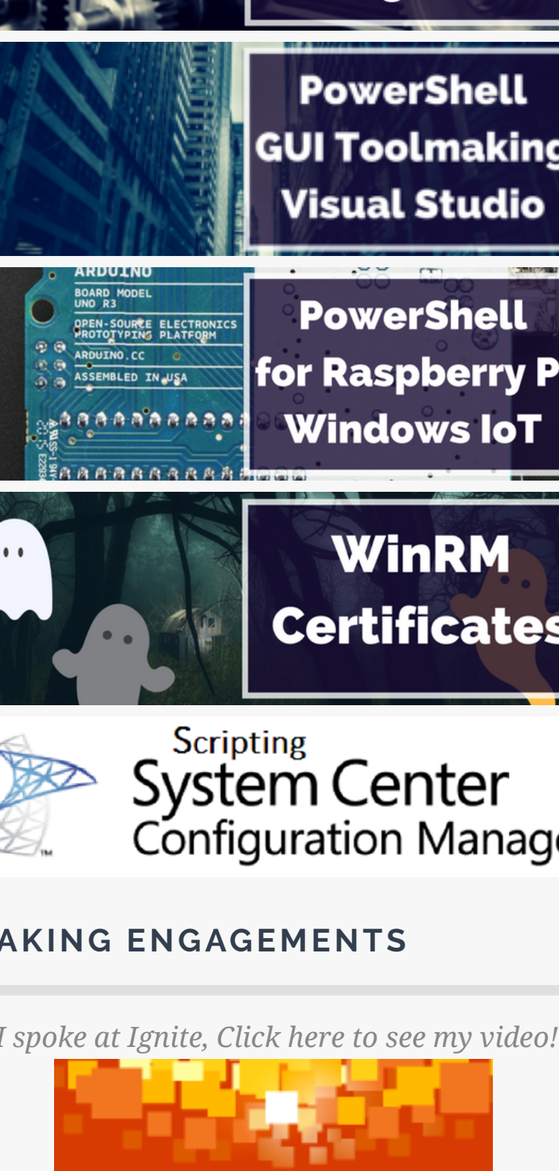
This site uses Akismet to reduce spam. Learn how your comment data is processed.

STEPHEN OWEN'S BLOG

MICROSOFT MVP

Microsoft Most Valuable Professional

BLOG SERIES



SPEAKING ENGAGEMENTS

I spoke at Ignite, Click here to see my video!

ATLANTA POWERSHELL USER'S GROUP

Our group has a new home! Sign up here to stay up to date with Meetup.com
Atlanta PowerShell User's Group on MeetUp.com

TOP POSTS & PAGES

- Part I - Creating PowerShell GUIs in Minutes using Visual Studio - A New Hope
- SOLVED! Windows 10 Reset - There was a problem resetting your PC
- Part II - Deploying PowerShell GUIs in Minutes using Visual Studio
- Windows vs Intel Raid Performance Smackdown
- Learning GUI Toolmaking Series

BLOG STATS

- 1,700,340 hits

NEED HELP?

Get help much faster (and help each other) on our new dedicated Subreddit!
Click the fox to join us!

Reddit.com/R/FoxDeploy

STACKEXCHANGE

FoxDeploy 6k •2 •15 •34

FORK MY REPOS

- 1RedOne opened a pull request in Microsoft/WPF-Samples
  December 30, 2018
- 1RedOne pushed to master in 1RedOne/WPF-Samples
  December 30, 2018

RECENT POSTS

- Quickie – Join video files with PowerShell and FFMPEG
- Excursion: Model View Controller Programming – Part I
- Faster: ConfigMgr Collection Manipulation Speed Test
- ClientFaux – the fastest way to fill ConfigMgr with Clients
- Hard to test cases in Pester

BLOGS I READ

Jeff Renfroe's Deploy OS SCCM Blog
Chris Nacker's System Center Blog
Jarvis Davis' System Center Blog
Troy Hunt's Management Blog
Ben Armstrong's Virtual PC guy blog
Hyper-V.NU
Francois Cat's PowerShell blog
Boe Prox's PowerShell blog

ARCHIVES

Select Month

RSS

- RSS - Posts