



**iamct**  
International Association of  
Microsoft Certified Trainers

## MCT Summit 2014 Bucharest



• July 3-5, 2014

*Platinum Sponsors*



*Sponsors*



International Association of  
Microsoft Certified Trainers



Please thank our sponsors

## Notes from the field: Serverless lab deployment



### Thorsten Butz

\$twitter = \$adn = @thorstenbutz

gplus.to/thorstenbutz

www.thorsten-butz.de (primarily German)

mctsummit.eu/agenda/thorsten-butz

Download the demo files:

[www.thorsten-butz.de/public/serverless\\_deployment.zip](http://www.thorsten-butz.de/public/serverless_deployment.zip)

### Session Description:

It's a permanent challenge to deploy test environments in a constantly evolving IT landscape. If you advance from pre-configured MOC-VMs, you win a lot, but there is plenty of work to do.

In this talk we will see three different approaches to accept these challenges: I'll introduce you to the famous "Hydration Kits" (based on MDT), the "PowerShell Deployment Toolkit (PDT)" by Rob Willis (MSFT) and I'll show how to develop your own toolkit, based on my personal experience with unattended files, PowerShell scripts and -of course- Hyper-V as the virtualization backend.

It's a challenge to deploy complex environments without a sophisticated server infrastructure (such as WDS, SCCM etc.). As an MCT you cannot rely on these tools, 'cause you simply don't have them in place, when you need them.

This talk will highlight a broad variety of techniques, from the creation of basic unattended files and "run once" scripts, to the complex deployment of the complete System Center 2012 (R2) suite.

Session Level: 300

## Notes from the field: Serverless lab deployment



### Thorsten Butz

\$twitter = \$adn = @thorstenbutz

gplus.to/thorstenbutz

www.thorsten-butz.de (primarily German)

mctsummit.eu/agenda/thorsten-butz

Download the demo files:

[www.thorsten-butz.de/public/serverless\\_deployment.zip](http://www.thorsten-butz.de/public/serverless_deployment.zip)

### Session Description:

It's a permanent challenge to deploy test environments in a constantly evolving IT landscape. If you advance from pre-configured MOC-VMs, you win a lot, but there is plenty of work to do.

In this talk we will see three different approaches to accept these challenges: I'll introduce you to the famous "Hydration Kits" (based on MDT), the "PowerShell Deployment Toolkit (PDT)" by Rob Willis (MSFT) and I'll show how to develop your own toolkit, based on my personal experience with unattended files, PowerShell scripts and -of course- Hyper-V as the virtualization backend.

It's a challenge to deploy complex environments without a sophisticated server infrastructure (such as WDS, SCCM etc.). As an MCT you cannot rely on these tools, 'cause you simply don't have them in place, when you need them.

This talk will highlight a broad variety of techniques, from the creation of basic unattended files and "run once" scripts, to the complex deployment of the complete System Center 2012 (R2) suite.

Session Level: 300



A **floppy disk**, or **diskette**, is a [disk storage](#) medium composed of a disk of thin and flexible [magnetic storage](#) medium, sealed in a rectangular plastic carrier lined with fabric that removes dust particles. Floppy disks are read and written by a **floppy disk drive** (FDD).

Floppy disks, initially as 8-inch (200 mm) media and later in 5¼-inch (133 mm) and 3½-inch (90 mm) sizes, were a ubiquitous form of data storage and exchange from the mid-1970s well into the 2000s.<sup>[1]</sup>

**Reference:**

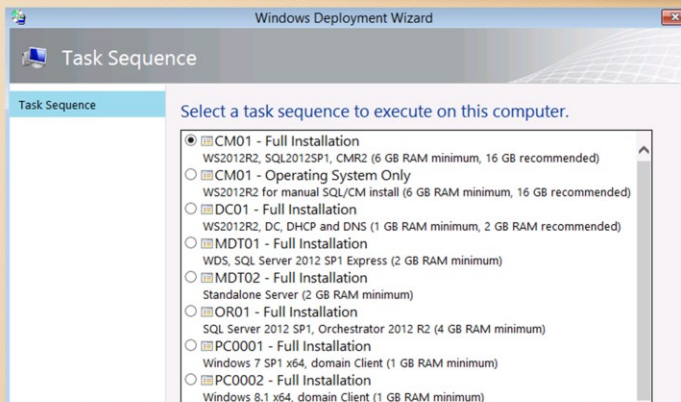
[http://en.wikipedia.org/wiki/Floppy\\_disk](http://en.wikipedia.org/wiki/Floppy_disk)

## Good old times?



These cities have been traditionally used by the MOC environment. The cities relate to 3 digits domain name, e.g. "van.contoso.com", "den.contoso.com".

## Part 1: Hydration kit(s)



The term "hydration kit" relates to a family of deployment solutions based on WADK, MDT and additional scripts.

The current example is called "The Hydration Kit for ConfigMgr 2012 R2", was created by (the famous!) Johan Arwidmark:

<http://www.deploymentresearch.com/Research/tabid/62/EntryId/113/The-Hydration-Kit-for-ConfigMgr-2012-R2-is-available-for-download.aspx>

Johan has detailed instructions on howto use the kit on his website (link above).

### Prerequisites:

a) WADK 8.1

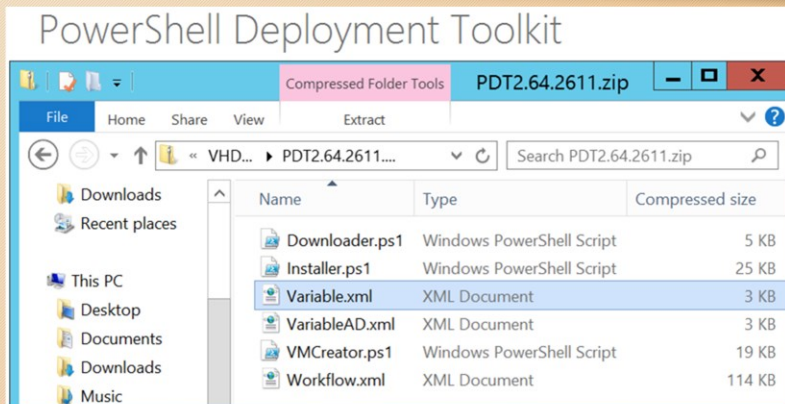
Windows Assessment and Deployment Kit (Windows ADK)

<http://www.microsoft.com/en-us/download/details.aspx?id=39982>

b) MDT 2013

<http://www.microsoft.com/en-us/download/details.aspx?id=40796>

## Part 2: PDT (PowerShell Deployment Toolkit)



The PDT is a set of PowerShell scripts developed by Rob Willis, a Microsoft employee who was formerly in charge of developing the "Unified installer" for the System Center product family.

The PDT targets (primarily) the complete lab setup of all the different System Center products. It's capable of deploying the software on different machines. You have the choice between deploying into an existing environment (may or may not be virtualized), or a complete "from scratch" virtual lab deployment.

PDT Download (Scripts only):

<http://gallery.technet.microsoft.com/PowerShell-Deployment-f20bb605>

PDT "Documentation":

<http://blogs.technet.com/b/privatecloud/archive/tags/deployment+track/>

### Prerequisites:

a) WinRAR

<http://www.rarlab.com/download.htm>

b) WPI (Windows platform installer) # optional

<http://www.microsoft.com/web/downloads/platform.aspx>




## Part 3: "Eat your own dog food"



### Download the demo files:

[http://www.thorsten-butz.de/public/serverless\\_deployment.zip](http://www.thorsten-butz.de/public/serverless_deployment.zip)

 serverless\_deployment.zip

### Useful tools:

- a) All of the previously mentioned solutions: Hydration kits, PDT, ...
- b) WADK, including SIM (System Image Manager)
- c) PowerShell
- d) Chocolatey (<https://chocolatey.org/>)
- e) The internet

## Example: Customize Hyper-V server settings



```
$vmhost = Get-VMHost

Set-VMHost -VirtualHardDiskPath `
'C:\Hyper-V\Virtual Hard Disks' -VirtualMachinePath 'C:\Hyper-V,

$vmhost.VirtualHardDiskPath
$vmhost.VirtualMachinePath

1..3 | ForEach-Object {
    if (Get-Vmswitch -Name "HV_Private$_" -ea:0) {"HV_Private$_ exists" } else {
        New-VMSwitch -Name HV_Private$_ -SwitchType Private
    }
}
```

## Example: Create a VM



```
Import-Module Hyper-V
```

```
$vmhost = '.'
$timestamp = Get-Date -UFormat '%y%m%d'
$vm = $timestamp + '_' + (Read-Host 'VM name')
$cpucount = 2
$startupram = 4GB
$vhdzsize = 300GB
$switch = 'HV_Private2'
$iso1 = 'C:\depot\iso\HydrationCM2012R2_ADATUM_DE.iso'
```

```
New-VM `
-ComputerName:$vmhost `
-Name:$vm `
-MemoryStartupBytes:$startupram `
-NewVHDPath:"$vm.vhdx" `
-NewvhdSizeBytes:$vhdzsize
```

```
Set-VM `
-ComputerName:$vmhost `
[...]
```

*Mind the notes for the complete example!*

```
Import-Module Hyper-V
```

```
$vmhost = '.'
$timestamp = Get-Date -UFormat '%y%m%d'
$vm = $timestamp + '_' + (Read-Host 'VM name')
$cpucount = 2
$startupram = 4GB
$vhdzsize = 300GB
$switch = 'HV_Private2'
$iso1 = 'C:\depot\iso\HydrationCM2012R2_ADATUM_DE.iso'
```

```
New-VM `
-ComputerName:$vmhost -Name:$vm -MemoryStartupBytes:$startupram -NewVHDPath:"$vm.vhdx" `
-NewvhdSizeBytes:$vhdzsize
```

```
Set-VM `
-ComputerName:$vmhost -name:$vm -ProcessorCount:$cpucount -DynamicMemory
```

```
Set-VMbios `
-ComputerName:$vmhost -vmname:$vm -StartupOrder:@("IDE", "CD", "Floppy", "LegacyNetworkAdapter")
```

```
Connect-VMNetworkAdapter -ComputerName:$vmhost -VMName:$vm -SwitchName:$switch
```

```
if ($iso1) {
    Set-VMdvdDrive -ComputerName:$vmhost -VMName:$vm -Path:$iso1
}
```

```
Start-VM -ComputerName $vmhost -VMName $vm
```

## autounattend.xml (partial)



```
<?xml version="1.0" encoding="utf-8"?>

<unattend xmlns="urn:schemas-microsoft-com:unattend">
  <settings pass="windowsPE">
    <component name="Microsoft-Windows-Setup" ...>

      <UserData>
        <AcceptEula>true</AcceptEula>
        <ProductKey>
          <!-- AVMA PID: -->
          <Key>DBGBW-NPF86-BJVTX-K3WKJ-MTB6V</Key>
          <!-- gVLK PID:
          <Key>D2N9P-3P6X9-2R39C-7RTCD-MDVJX</Key> -->
          <WillShowUI>OnError</WillShowUI>
        </ProductKey>
      </UserData>

    <!-- ...
  </component>
</settings>
</unattend>
```

"autounattend.xml" files can be created with the SIM (System Image Manager), which is part of the (previously mentioned) Windows ADK/WADK.

The code above is a small part of a more complete/complex file you will find in the demofiles archive.

### Automated virtual machine Activation (AVMA):

Mind the example product IDs. The so called AVMA PIDs can be used as a placeholder during deployment. The superior advantage compared to the "generic volume license keys (gVLK)" is the fact that your VMs show up activated, as long as two conditions are met:

- a) WS 2012 R2 Datacenter Edition as Hyper-V host, already activated  
(the Hyper-V host may also be a KMS client)
- b) WS 2012 R2 as VM

### Reference:

<http://blogs.technet.com/b/askcore/archive/2013/11/13/activating-virtual-machines-via-automatic-virtual-machine-activation-in-windows-server-2012-r2.aspx>

## Example: Change hostname in "autounattend.xml"



```
# TEMPLATE AUTOUNATTEND.XML FILE
$file = "c:\depot\unattended\ws2012r2\template-www1\autounattend.xml"

# GRAB THE XML FILE
[xml]$content = Get-Content $file

$component = $content.unattend.settings.component |
  where-object {$_.computername -like "template*" }
$component computername

# ASK FOR A NEW NAME
[string]$vm = Read-Host 'New computername'

# NEW AUTOUNATTEND.XML FILE
$newfile = "c:\depot\unattended\ws2012r2\template-www1\autounattend_demo.xml"

$component computername = $vm
$content.Save($newfile)
```

The example above is strictly intended as a DEMO. It shows the procedure how to handle the "autounattend.xml" file, how to grab the computername out of the XML data and how to change it.

You find a more complex script attached to this slide:  
"D\_CreateISO\_from\_TemplateFolder.ps1"

The script allows to change the hostname by script and create ISO files from a "template" folder structure (recursively).

**Prerequisites:** # To create ISO files

OSCDIMG.EXE

Part of the WADK (mentioned previously)

## Some useful snippets



```
# ENABLE PS-REMOTING FOR WORKGROUPS
Set-Item WSMAN:\localhost\Client\TrustedHosts * -Force

# DISABLE DHCP-SERVER ROQUE DETECTION
REG ADD "HKLM\SYSTEM\CurrentControlSet\Services\DHCPServer\Parameters" `
  /v DisableRogueDetection /t REG_DWORD /d 1 /f

# CREATE A SHARE
New-SmbShare -name depot -path "C:\Depot" -FullAccess:Everyone

# CONFIGURE FW
netsh.exe advfirewall firewall set rule `
  group="windows management instrumentation (wmi)" new enable=yes

# INSTALLAING CHOCOLATEY
iex ((new-object net.webclient).DownloadString `
  ('https://chocolatey.org/install.ps1'))
```

### # Example 1: PS-Remoting in workgroups

Authentication will fail by default in workgroup environments; to enable remoting you have to configure WSMAN on the management system (your own server, where you run the script).

### # Example 2: DHCP roque detection

Without the registry key modified, you must authorize your DHCP server in AD - which is extremely "difficult" if you do not have any AD.

You can also use this PS command:

```
Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\`
  DHCPServer\Parameters" -Name DisableRogueDetection -Value 1 -Force
```

But the PS command has a disadvantage: it runs only if the path exists. The REG.EXE does not bother, so I prefer the "classic" style.

### # Example 5: Chocolatey

If you do not know, what chocolatey is, you should check their website asap. You will love it: <https://chocolatey.org>

## Wrap up



	Feature rich	Data handling	Versatile
Hydration Kit(s)	😊	😞	😐
PDT	😐	😊	😐
Your work	?	😊	😊

Your mileage may vary.

It would be helpful, if you take the time to fill out the evaluation survey:

<http://mctsummit.eu/agenda/>

Click the on the session link, below you find the the survey link.

Or directly:

<https://www.surveymonkey.com/s/YTMFGDY>

Any other feedback is also very welcome!

Thanks for your interest.

Thorsten