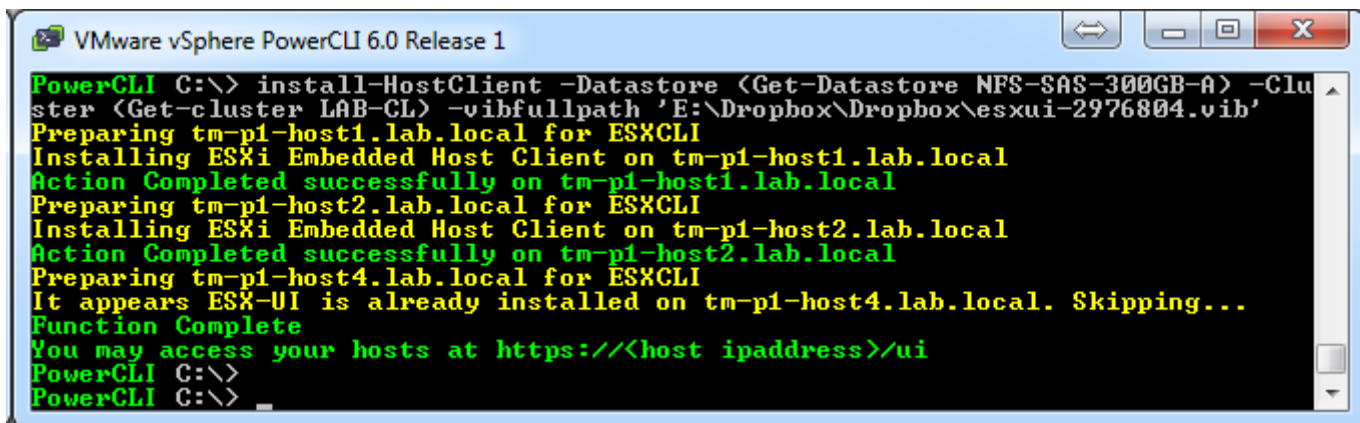


Automate the Install of Embedded Host Client for ESXi fling on all hosts in a cluster

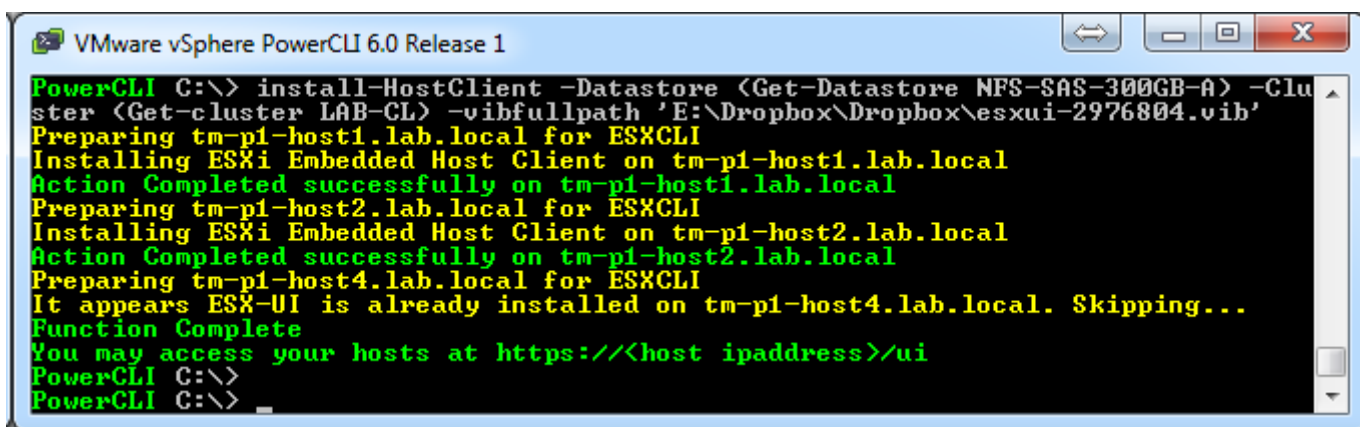
<https://www.brianjgraf.com/2015/08/13/automate-install-embedded-host-client-esxi-fling-hosts-cluster/#prettyPhoto>



```
VMware vSphere PowerCLI 6.0 Release 1
PowerCLI C:\> install-HostClient -Datastore (Get-Datastore NFS-SAS-300GB-A) -Cluster (Get-cluster LAB-CL) -vibfullpath 'E:\Dropbox\Dropbox\esxui-2976804.vib'
Preparing tm-p1-host1.lab.local for ESXCLI
Installing ESXi Embedded Host Client on tm-p1-host1.lab.local
Action Completed successfully on tm-p1-host1.lab.local
Preparing tm-p1-host2.lab.local for ESXCLI
Installing ESXi Embedded Host Client on tm-p1-host2.lab.local
Action Completed successfully on tm-p1-host2.lab.local
Preparing tm-p1-host4.lab.local for ESXCLI
It appears ESX-UI is already installed on tm-p1-host4.lab.local. Skipping...
Function Complete
You may access your hosts at https://<host ipaddress>/ui
PowerCLI C:\>
PowerCLI C:\>
```

Automate the Install of Embedded Host Client for ESXi fling on all hosts in a cluster

Aug 13, 2015 | [Blog](#), [PowerCLI](#), [Virtualization](#)



```
VMware vSphere PowerCLI 6.0 Release 1
PowerCLI C:\> install-HostClient -Datastore (Get-Datastore NFS-SAS-300GB-A) -Cluster (Get-cluster LAB-CL) -vibfullpath 'E:\Dropbox\Dropbox\esxui-2976804.vib'
Preparing tm-p1-host1.lab.local for ESXCLI
Installing ESXi Embedded Host Client on tm-p1-host1.lab.local
Action Completed successfully on tm-p1-host1.lab.local
Preparing tm-p1-host2.lab.local for ESXCLI
Installing ESXi Embedded Host Client on tm-p1-host2.lab.local
Action Completed successfully on tm-p1-host2.lab.local
Preparing tm-p1-host4.lab.local for ESXCLI
It appears ESX-UI is already installed on tm-p1-host4.lab.local. Skipping...
Function Complete
You may access your hosts at https://<host ipaddress>/ui
PowerCLI C:\>
PowerCLI C:\>
```

It's true! If you are reading this then you already know that there is a new VMware fling that rocks this world! You can read more about it on William Lam's blog [HERE](#):

Now, to be able to use the Embedded Host Client for ESXi fling, you will need to:

- download the VIB from the VMware Flings site [HERE](#):
- WinSCP the VIB to each esx host (or place on a datastore)
- Enable SSH on each host
- Putty/SSH each host and run the vib-install command
- Disable SSH

ok, for the functionality gained by this fling, it's worth it to do those steps above. **But what if you could do it all with PowerCLI in one command?**

I've created two advanced functions that I think you will like. The first is Install-HostClient, which will copy the VIB to shared storage and automatically install the VIB on all hosts in the specified cluster, all in a single command. you ready??

Install-HostClient

you can copy and paste the code below, or you can also find this script on my Github repo [HERE](#).

Install-HostClient requires three parameters to be set (as well as be connected to vCenter):

- Datastore (must be shared storage)
- Cluster (the cluster where you will install the vib)
- vibfullpath (the full path to the vib on your local machine)

The command would look like this:



```
1 install-HostClient -Datastore (Get-Datastore NFS-SAS-300GB-A) -Cluster (Get-Cluster LAB-CL) -vibfullpath c:\temp\esxui-2976804.vib
```

or you could also pre-define the datastore and cluster with variables and then place the variables in the command:



```
1 install-HostClient -Datastore $ds -Cluster $cluster -vibfullpath c:\temp\esxui-2976804.vib
```

You will notice that the script requires the datastore and cluster objects, not just a string/name for each. Once the command is executed it will upload the vib to the parent directory of the designated datastore, then loop through each host.

The script checks to see if the vib is already installed or not. If it is, the script will skip that host, otherwise it will install.

It will also verify that the vib installed correctly, if it does not, it will give you an error message and move on.

As you can see in the screenshot:

```
VMware vSphere PowerCLI 6.0 Release 1
PowerCLI C:\> install-HostClient -Datastore (Get-Datastore NFS-SAS-300GB-A) -Cluster (Get-cluster LAB-CL) -vibfullpath 'E:\Dropbox\Dropbox\esxui-2976804.vib'
Preparing tm-p1-host1.lab.local for ESXCLI
Installing ESXi Embedded Host Client on tm-p1-host1.lab.local
Action Completed successfully on tm-p1-host1.lab.local
Preparing tm-p1-host2.lab.local for ESXCLI
Installing ESXi Embedded Host Client on tm-p1-host2.lab.local
Action Completed successfully on tm-p1-host2.lab.local
Preparing tm-p1-host4.lab.local for ESXCLI
It appears ESX-UI is already installed on tm-p1-host4.lab.local. Skipping...
Function Complete
You may access your hosts at https://<host ipaddress>/ui
PowerCLI C:\>
PowerCLI C:\>
```

```
function install-HostClient {
<#
.NOTES
=====
```

```
1 function install-HostClient {
2 <#
3 .NOTES
4 =====
5 Created on: 8/13/2015 9:12 AM
6 Created by: Brian Graf
7 Github: http://www.github.com/vtagion
8 Twitter: @vBrianGraf
9 Website: http://www.brianjgraf.com
10 =====
11 .DESCRIPTION
12 This advanced function will allow you to install the ESXi Host Client
13 On all the hosts in a specified cluster.
14 .Example
15 Install-HostClient -Cluster (Get-Cluster Management-CL) -Datastore (Get-Datastore NFS-SAS-300GB-A)
16 -vibfullpath c:\temp\esxui-2976804.vib
17
18 .Example
19 $ds = Get-Datastore Main-shared
20 $Cluster = Main-CL
21 Install-HostClient -Cluster $cluster -Datastore $ds -vibfullpath c:\temp\esxui-2976804.vib
22
23 .Notes
24 You must use shared storage for this to work correctly, otherwise only a single host will be able to install
25 the vib and all others will fail
26 #>
27 [CmdletBinding()]
28 param(
29 [Parameter(Mandatory=$true, ValueFromPipelineByPropertyName=$true, HelpMessage="Must be shared
30 storage across all hosts")]
31 [ValidateScript({Get-Datastore $_})]
32 [VMware.VimAutomation.ViCore.Impl.V1.DatastoreManagement.NasDatastoreImpl]$Datastore,
33
34 [Parameter(Mandatory=$true, ValueFromPipelineByPropertyName=$true, HelpMessage="Please specify a
35 Cluster object")]
36 [ValidateScript({Get-Cluster $_})]
37 [VMware.VimAutomation.ViCore.Impl.V1.Inventory.ComputeResourceImpl]$Cluster,
38
39 [Parameter(Mandatory=$true, ValueFromPipelineByPropertyName=$true, HelpMessage="Specify the full
```

```

40 path of the ESXi Host Client Vib")]
41 [ValidateScript({Get-Item $_})]
42 [String]$vibfullpath
43 )
44 Begin {
45
46 $VIBFile = Get-item $vibfullpath -ErrorAction SilentlyContinue
47
48 # Verify that VIB location is correct
49 if ($VIBFile -eq $null){Throw "oops! looks like $VIBFile doesn't exist in this location."}
50
51 # Save filename to variable
52 $VIBFilename = $vibfile.PSChildname
53
54 # Save datacenter to variable for Datastore path
55 $dc = $Cluster | Get-Datacenter
56
57 #Get-Datastore -Name $Datastore
58
59 # Create Datastore Path string
60 $Datastorepath = "vmstore:\" + $dc + "\" + $Datastore.Name + "\"
61
62 # Verbose info for debugging
63 Write-verbose "DatastorePath = $Datastorepath"
64 Write-verbose "Vibfile = $vibfile"
65 Write-verbose "Vibfullpath = $vibfullpath"
66 Write-verbose "VibFilename = $VIBFilename"
67
68 # check to see if file already exists or not before copying
69 if (!(Test-Path -Path $Datastorepath)) {
70 Copy-DatastoreItem $vibfile $Datastorepath -Force
71 }
72
73 # validate the copy worked. If not, stop script
74 if (!(Test-Path -Path $Datastorepath)) {
75 Throw "Looks like the VIB did not copy to $Datastorepath. Check the filename and datastore path again
76 and rerun this function."
77 }
78
79 # Create VIB path string for ESXCLI
80 $VIBPATH = "/vmfs/volumes/" + $datastore.name + "/" + "$VIBFilename"
81
82 }
83
84 Process {
85
86
87 # $VIBPATH = "/vmfs/volumes/NFS-SAS-300GB-A/esxui-2976804.vib"
88
89 # Get each host in specified cluster that meets criteria
90 Get-VMhost -Location $Cluster | where { $_.PowerState -eq "PoweredOn" -and $_.ConnectionState -eq
91 "Connected" } | foreach {
92
93 Write-host "Preparing $($_.Name) for ESXCLI" -ForegroundColor Yellow
94
95 # Create ESXCLI variable for host for actions

```

```

96 $ESXCLI = Get-EsxCli -VMHost $_
97
98 # Check to see if ESX-UI is already installed
99 if (($ESXCLI.software.vib.list() | Select
100 AcceptanceLevel,ID,InstallDate,Name,ReleaseDate,Status,Vendor,Version | Where {$_.Name -match
101 "esx-ui"})) { Write-host "It appears ESX-UI is already installed on $_. Skipping..." -ForegroundColor
102 Yellow} else {
103
104 Write-host "Installing ESXi Embedded Host Client on $($_.Name)" -ForegroundColor Yellow
105
106 # Saving command to variable to use for verification after command is run
107 $action = $ESXCLI.software.vib.install($null,$null,$null,$null,$null,$null,$null,$null,$VIBPATH)
108
109 # Verify VIB installed successfully
110 if ($action.Message -eq "Operation finished successfully.") { Write-host "Action Completed successfully on
111 $($_.Name)" -ForegroundColor Green} else { Write-host $action.Message -ForegroundColor Red}
112 }
113 }
114 }
115 End {
116 Write-host "Function Complete" -ForegroundColor Green
117 Write-Host "You may access your hosts at https://<host ipaddress>/ui" -ForegroundColor Green
118 }
119 }

```

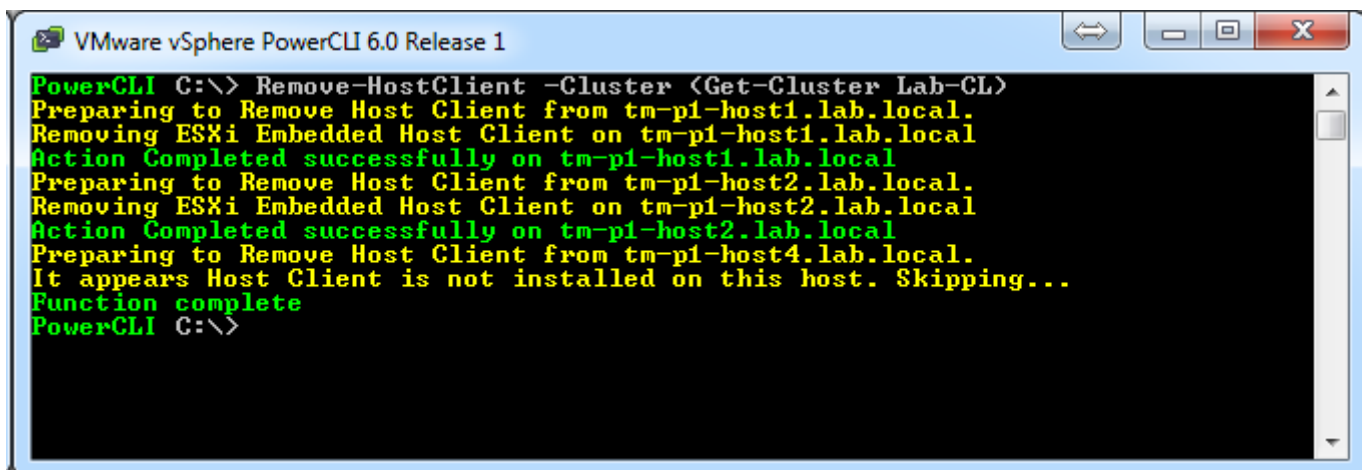
Remove-HostClient

Similar to the ability of the Install-HostClient function. This will allow you to bulk-remove the vib if necessary.

This function only requires a single parameter, which is -Cluster. the command would look like this:



```
1 Remove-HostClient -Cluster (Get-Cluster LAB-CL)
```



You can find this on github [HERE](#).

or Copy and paste from below:



```
1 function Remove-HostClient {
2 <#
3 .NOTES
4 =====
5 Created on: 8/13/2015 9:12 AM
6 Created by: Brian Graf
7 Github: http://www.github.com/vtagion
8 Twitter: @vBrianGraf
9 Website: http://www.brianjgraf.com
10 =====
11 .DESCRIPTION
12 This advanced function will allow you to remove the ESXi Host Client
13 on all the hosts in a specified cluster.
14 .Example
15 Remove-HostClient -Cluster (Get-Cluster Management-CL)
16
17 .Example
18 $Cluster = Main-CL
19 Remove-HostClient -Cluster $cluster
20 #>
21 [CmdletBinding()]
22 param(
23 [ValidateScript({Get-Cluster $_})]
24 [VMware.VimAutomation.ViCore.Impl.V1.Inventory.ComputeResourceImpl]$Cluster
25 )
26 Process {
27
28 # Get all ESX hosts in cluster that meet criteria
29 Get-VMhost -Location $Cluster | where { $_.PowerState -eq "PoweredOn" -and $_.ConnectionState -eq
30 "Connected" } | foreach {
31
32 Write-host "Preparing to remove Host Client from $($_.Name)" -ForegroundColor Yellow
33
34 # Prepare ESXCLI variable
35 $ESXCLI = Get-EsxCli -VMHost $_
36
37 # Check to see if VIB is installed on the host
38 if (($ESXCLI.software.vib.list() | Where { $_.Name -match "esx-ui" })) {
39
40 Write-host "Removing ESXi Embedded Host Client on $($_.Name)" -ForegroundColor Yellow
41
42 # Command saved to variable for future verification
43 $action = $esxcli.software.vib.remove($null,$null,$null,$null,"esx-ui")
44
45 # Verify VIB removed successfully
46 if ($action.Message -eq "Operation finished successfully."){Write-host "Action Completed successfully on
47 $($_.Name)" -ForegroundColor Green} else {Write-host $action.Message -ForegroundColor Red}
48
49 } else { Write-host "It appears Host Client is not installed on this host. Skipping..." -ForegroundColor
50 Yellow }
51 }
```

```
}  
End {Write-host "Function complete" -ForegroundColor Green}  
}
```

As always, read through the script and understand exactly what is going on before running it. This is not supported (nor is the fling).

Cheers!