

Using PowerCLI to synchronize vCenter Notes with Active Directory description fields

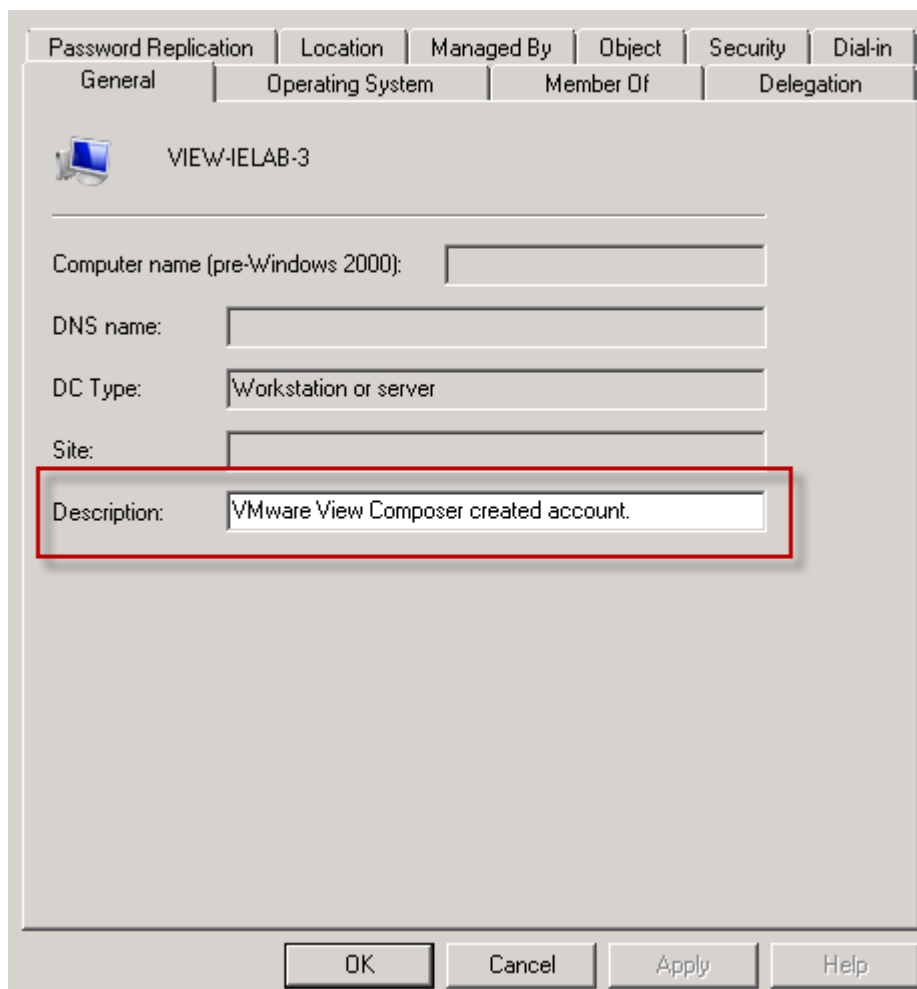
Recently one of our awesome community members posed a question for a PowerCLI script to pull some information from Active Directory to update VMware vCenter guest attributes.

This is the fun of our scripting community because we can save a lot of time by asking the question among members and often they have saved some serious time by putting a script together already.

It is really great when I get to work among really great folks like the one and only [@ExchangeGoddess](#) who is a contributing author to the Petri IT Knowledgebase (<http://www.petri.co.il/author/phoummala-schmitt>) so this was particularly cool to be able to help out.

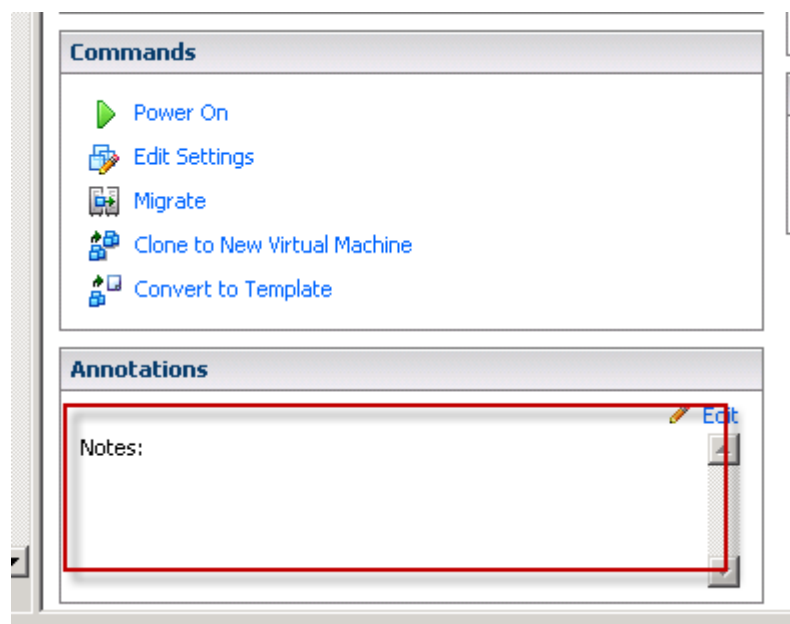
What are we trying to do?

Many admins who like to update our Active Directory computer objects with a description field so that we know exactly what our servers or workstations are used for. If you don't already do this, I highly recommend it.



What we want to do is get this information into our vCenter guests though because that is where a significant amount of our administration happens. Once the object is created we don't go back to

Active Directory Users and Computers (ADUC) as much. This is where we can use our Annotations field in vCenter. As you can see, it starts off empty:



How will we do it?

Luckily the content that we are looking for is easily accessible from the properties of the guest objects in both AD and vCenter. We want to go through our vCenter to find VMs, then query Active Directory for the object. From there we query the AD object for a Description field and then feed that content back to the VM guest into the Notes area as the Description field there.

Let's get to the script!

SPECIAL THANK YOU: Thanks to Stevo (see comments section) for adding to the script to fix up issues where I wasn't clearing the variable which could cause some problems.

Our script is pretty simple, but has a few prerequisites needed in order to run. For our environment, we need:

- AD CmdLets - These are installed with the Remote Server Administration Tools (RSAT)
- PowerCLI - This is available from <http://communities.vmware.com/community/vmtn/automationtools/powercli>
- Privileges to read the properties from AD and update within vCenter

Once we meet those needs, create a .PS1 file using the following code:

```
# Import the Active Directory module for Get-ADComputer CmdLet
Import-Module ActiveDirectory
# Connect to vCenter
Connect-VIServer YOURVCENTERSERVERHERE
# Pull your VMs into an array
$Guests = Get-VM
# Loop through the array to read the AD Description field to set the Annotation in
vCenter
ForEach ($Guest in $Guests) {
```

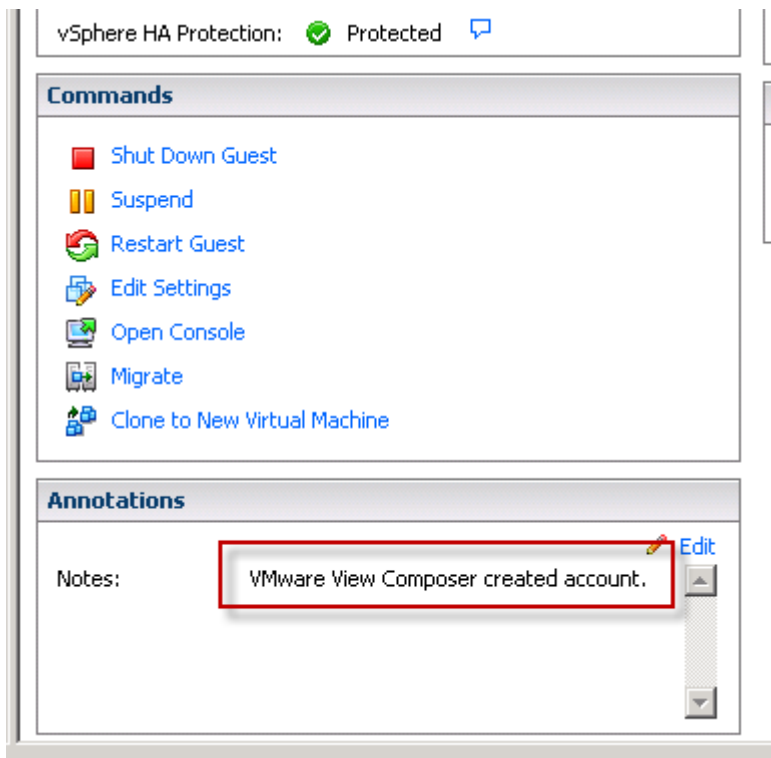
```

# Clear the variable just in case there is no match. It may keep the previous value.
Clear-Variable Description
Clear-Variable ADComputer
$ADComputer = Get-ADComputer "$Guest" -Properties Description
$Description = $ADComputer.Description
Set-VM -VM $Guest -Notes $Description -Confirm:$false
}

```

A quick note about this script is that it requires that your VM object name is the same as the computer name in Active Directory. Secondly, we are doing no error handling in the script, but if the machine isn't found in AD it just errors out and moves to the next object.

Once we run the script, you can check your vCenter object on the Summary tab and as if by magic, there it is:



So now that we have the script, we should take the next step and configure this process to run regularly. I'm all about automation wherever possible, and this is a perfect candidate for a batch process.

Running PowerCLI as a Scheduled Task

Because PowerCLI requires loading PowerShell with some environment wrapped around it we can't just load the modules using **Import-Module** like we can with the RSAT extensions.

While I could write about how to do this, this is a great way to promote some more awesome content, so head on over to view **Magnus Andersson's** post on doing just what we want to do: <http://vcdx56.com/2013/03/27/schedule-powercli-script-in-windows-task-scheduler/>

So now you will have a nightly load of your Active Directory computer object content into your vCenter. You can adjust the machines that are being queried in the vCenter environment by adding some parameters to your **Get-VM** query. It's just that easy ☐

Thanks to Phoummala for the inspiration, and to Magnus ([@magander3](#)) for the great post on Windows Scheduled Tasks using PowerCLI. I hope that you all find this helpful!