

How to find Local Datastores with ...



[SCampbell](#) 106 posts since

Apr 18, 2005 You can easily create a local datastore with the New-Datastore -Local option.

I am now trying find local datastores.

So far I am at this to return an array of extents for my VMFS partitions

```
$DS=Get-DataStore | Where-Object  
{ $Extents=($_ | Get-View).Info.VMFS.Extents; ...
```

I was going to try to map this to list of LUNs and look at the associated driver. All our shared LUNs will have QLogic or iSCSI S/W drivers

```
$LUNs= (Get-VMHost |  
Get-View).Config.StorageDevice.SCSILun
```

Has anyone found the easy way to Get-Datastore -Local?

Thanks....



[c_shanklin](#) 762 posts since

Dec 3, 2007 1. **Re: How to find Local Datastores with Powershell?** Oct 5, 2008 8:05 PM

You might be able to look at the paths associated with the datastores to check their SAN identifiers, which would indicate network storage. I'm not sure that local datastores even have paths.



[DougBaer](#) 602 posts since

Oct 20, 2004 2. **Re: How to find Local Datastores with Powershell?** Oct 7, 2008 11:48 AM

in response to: [c_shanklin](#) I guess you could walk through your VMhosts and build a list of datastores visible to each host, then weed out the multiple-access ones.

The remaining ones would be local or presented to a single host. Or, you could rely on naming conventions to tell you what is local and what is not.

If you're up for digging into the API a little, there is a boolean called "multipleHostAccess" under the "summary" of the Datastore object.



[SCampbell](#) 106 posts since

Apr 18, 2005 3. **Re: How to find Local Datastores with Powershell?** Oct 7, 2008 12:07 PM

in response to: [DougBaer](#) Hi Doug,

Thanks for this.

I have been digging in the API extensively trying to extend lbert's excellent ESXi provisioning script to add licensing, SSL certificate publication, support for ESX as well as ESXi, etc.

How to find Local Datastores with ...

One thing I want to do is either rename or remove local datastores so they don't accidentally get used during VM provisioning.

Right now, I'm just looking for datastores that end with :storage1 but that's not terribly reliable.

I could wait until I get this added to VC and then enumerate like you suggest which might be interesting.

I will have a look at the multipleHostAccess thingie. That might also just do the trick.

I'll post the finished script... someday.... when its done....

Thanks....



[gstaykov](#) 1 posts since

Jul 3, 2008 4. **Re: How to find Local Datastores with Powershell?** Oct 8, 2008 7:41 AM

You can try using Get-VMHostStorage view

```
$hsv = Get-VMHostStorage -VMHost (Get-VMHost) | Get-View
```

I believe that you can find all needed information in \$hsv.StorageDeviceInfo.ScsiTopology.Adapter
Look for FibreChannelHba, ParallelScsiHba, BlockHba etc.



[SCampbell](#) 106 posts since

Apr 18, 2005 5. **Re: How to find Local Datastores with Powershell?** Oct 10, 2008 8:01 PM

in response to: [gstaykov](#) These functions provide assistance dealing with Datastores. I have included the code in this post and in an attached file.

Comments and improvements are welcomed.

All functions, with some usage information and a sample, is in the attachment

Get-DatastoreHost -Datastore \$DS -IsConnected \$True returns a list of VIM VMHost objects that have access to the store. Use IsConnected if you only want ones that are properly alive.

```
function Get-DatastoreHost
{
Param(
[http:
//VMware.VimAutomation.Client20.DatastoreImpl|http://VMware.VimAutomation.Client20.DatastoreImpl]
$Datastore,
[<b>boolean</b>] $IsConnected=$True
)
($Datastore | Get-View).Host | ForEach-Object
{ get-view -id $_.Key
} | `
```

How to find Local Datastores with ...

```
Where-Object
{ (-not $IsConnected) -or
($_.Runtime.ConnectionState -eq
"connected")
}
```

Get-LocalDatastore -VMHost \$Host returns a list of VIToolkit Datastore objects which are hosted on devices of type ParallelSCSIHBA or BlockHBA. I think these are only owned by a local ESX server. I wanted this code to reliably rename local datastores so they aren't accidentally selected when operations teams provisions VMs

```
function Get-LocalDatastore
{
    Param(
        [http:
//VMware.VimAutomation.Client20.VMHostImpl|http://VMware.VimAutomation.Client20.VMHostImpl]
$VMHost
    )
    $LocalDevices=(Get-VMHostStorage -VMHost $VMHost |
Get-View).StorageDeviceInfo.HostBusAdapter | `
    Where-Object
    { `
        ($_.Key -like
"key-vim.host.ParallelSCSIHBA-*) -or `
        ($_.Key -like
"key-vim.host.BlockHBA-*) `
    } | %
    { [string]($_.Device)
    }
    Get-Datastore -VMHost $VMHost | Where-Object
    {
        (($_ | Get-View).Info.vmfs.extent | %
{ ([string]$_ .DiskName).split(
".",2)[0]
} ) `
        -contains $LocalDevices
    }
}
```

Get-DatastoreDatacenter -Datastore \$DS There are some (e.g., vifs.pl) processes that insist on a Datacenter in some cases. This returns the datacenter VIToolkit object a given Datastore is in. This could also be easily adapted to hosts.

```
function Get-DatastoreDatacenter
{
    Param(
```

How to find Local Datastores with ...

```
$Datastore
)
Get-Datacenter | Where-Object
{
    (($_ | Get-View).Datastore | %
{ (Get-View -Id $_).Summary.Url
}) `
    -contains (($Datastore | Get-View).Summary.Url)
}
}
```

Edited on Oct 10 to replace completely wrong Get-DatastoreDatacenter **Attachments:**

- [datastoreFunctions.ps1](#) (2.5 K)



[doomdevice](#) 98 posts since

Dec 4, 2005 6. **Re: How to find Local Datastores with Powershell?** Jan 4, 2009 2:48 AM

in response to: [SCampbell](#) SCampbell, First of all thanks for the script, it was a great help.

But I found an issue within the local datastore function if there are multiple local devices, because comparing the content of an array with the

content of another array with -contains isn't working in that way.

Furthermore if there are dozens or hundreds of datastores and hosts, I think it's a good idea to filter datastores first occurring on more than one host.

Therefore I created the following function:

```
function IsLocalStorage($ds)
{
    $hstorage = Get-VMHost -Datastore $ds
    $result = $false
    if (!$hstorage.length)
    {
        $LocalDevices=(Get-VMHostStorage -VMHost $hstorage |
Get-View).StorageDeviceInfo.HostBusAdapter | `
        Where-Object { `
            ($_.Key -like "key-vim.host.ParallelSCSIHBA-*") -or `
            ($_.Key -like "key-vim.host.BlockHBA-*") `
        } | %{ [string]( $_.Device) }

        foreach ($ld in $LocalDevices)
        {

            $localstorage = $ds | Where-Object {
                (($_ | Get-View).Info.vmfs.extent | %{
([string]$_ .DiskName).split(":",2)[0]} ) `
                    -contains $ld
            }
        }
    }
}
```

How to find Local Datastores with ...

```
    }  
    if ($localstorage) {$result = $true}  
  }  
  
  return $result  
}
```

Call that function with the datastore object, true means it is local storage, false it isn't.

Comments and improvements are very welcome

Dennis

VI PowerScripter <http://www.powerscripter.net>

Every Click can be a customized function within VI client