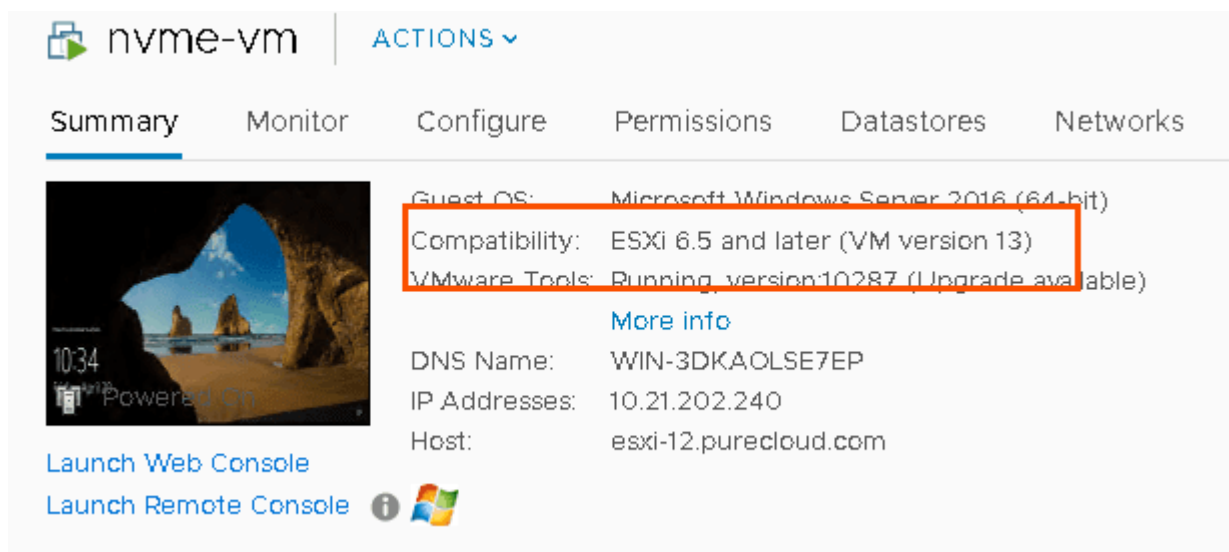


# What's New in Core Storage in vSphere 6.7 Part IV: NVMe Controller In-Guest UNMAP Support



nvme-vm | ACTIONS ▾

Summary | Monitor | Configure | Permissions | Datastores | Networks

Guest OS: Microsoft Windows Server 2016 (64-bit)  
Compatibility: ESXi 6.5 and later (VM version 13)  
VMware Tools: Running, version:10287 (Upgrade available)  
[More info](#)

DNS Name: WIN-3DKAOLSE7EP  
IP Addresses: 10.21.202.240  
Host: esxi-12.purecloud.com

Launch Web Console  
Launch Remote Console ⓘ 🏠

vSphere 6.7 core storage “what’s new” series:

- What’s New in Core Storage in vSphere 6.7 Part I: In-Guest UNMAP and Snapshots
- What’s New in Core Storage in vSphere 6.7 Part II: Sector Size and VMFS-6
- What’s New in Core Storage in vSphere 6.7 Part III: Increased Storage Limits
- What’s New in Core Storage in vSphere 6.7 Part IV: NVMe Controller In-Guest UNMAP Support
- More to come...

Another feature added in vSphere 6.7 is support for a guest being able to issue UNMAP to a virtual disk when presented through the NVMe controller.

The NVMe controller doesn’t use traditional UNMAP—it isn’t SCSI, it uses NVMe of course. If you want to learn more about NVMe, this is my favorite post that explains it:

<https://blogs.cisco.com/datacenter/nvme-for-absolute-beginners>

In the first release of this controller, VMware only supported the basic mandatory admin and I/O command sets. UNMAP is referred to as DEALLOCATE in NVMe (see the SCSI translation reference [here](#) and a deep dive [here](#) and the full spec [here](#)). VMware did not support the dataset management that includes DEALLOCATE.

Let’s walk through this.

I have a Windows 2016 Server virtual machine:

The screenshot shows the vSphere interface for a virtual machine named 'nvme-vm'. The 'Summary' tab is active. The 'Guest OS' is 'Microsoft Windows Server 2016 (64-bit)'. The 'Compatibility' is 'ESXi 6.5 and later (VM version 13)', which is highlighted with a red box. The 'VMware Tools' are 'Running, version:10287 (Upgrade available)'. Other details include 'DNS Name: WIN-3DKAOLSE7EP', 'IP Addresses: 10.21.202.240', and 'Host: esxi-12.purecloud.com'. There are buttons for 'Launch Web Console' and 'Launch Remote Console'.

You will notice that my VM is currently hardware version 13, this is not the latest version, which is 14. But let's not upgrade yet.

First, I will add an NVMe controller:

The screenshot shows the 'Edit Settings' dialog for the 'nvme-vm'. The 'Virtual Hardware' tab is selected. A table lists the current hardware settings:

> CPU	4	▼
> Memory	16	GB ▼
> Hard disk 1	200	GB ▼
> SCSI controller 0	VMware Paravirtual	
> Network adapter 1	VM Network ▼	
> CD/DVD drive 1	Client Device ▼	
> USB xHCI controller	USB 3.0	
> Video card	Specify custom settings ▼	

An 'ADD NEW DEVICE' button is in the top right. A dropdown menu is open, showing options: CD/DVD Drive, Host USB Device, Hard Disk, RDM Disk, Existing Hard Disk, Network Adapter, SCSI Controller, USB Controller, SATA Controller, NVMe Controller (highlighted with a blue bar), Shared PCI Device, and PCI Device. An orange arrow points to the 'NVMe Controller' option.

Then I will add a new virtual disk. It must be:

- Thin
- Added to the NVMe adapter

▼ <i>New Hard disk *</i>	40	GB ▼
Maximum Size	15.96 TB	
VM storage policy	Datastore Default ▼	
Location	Store with the virtual machine ▼	
Disk Provisioning	Thin Provision ▼	
Sharing	Unspecified ▼	
Shares	Normal ▼	1000
Limit - IOPs	Unlimited ▼	
Virtual flash read cache	0	MB ▼
Disk Mode	Dependent ▼	
Virtual Device Node	New NVMe Controller ▼	NVME(0:0) New Hard disk ▼

I will now format it with NTFS:

#### New Simple Volume Wizard



##### Format Partition

To store data on this partition, you must format it first.

Choose whether you want to format this volume, and if so, what settings you want to use.

Do not format this volume  
 Format this volume with the following settings:

File system:  ▼  
 Allocation unit size:  ▼  
 Volume label:

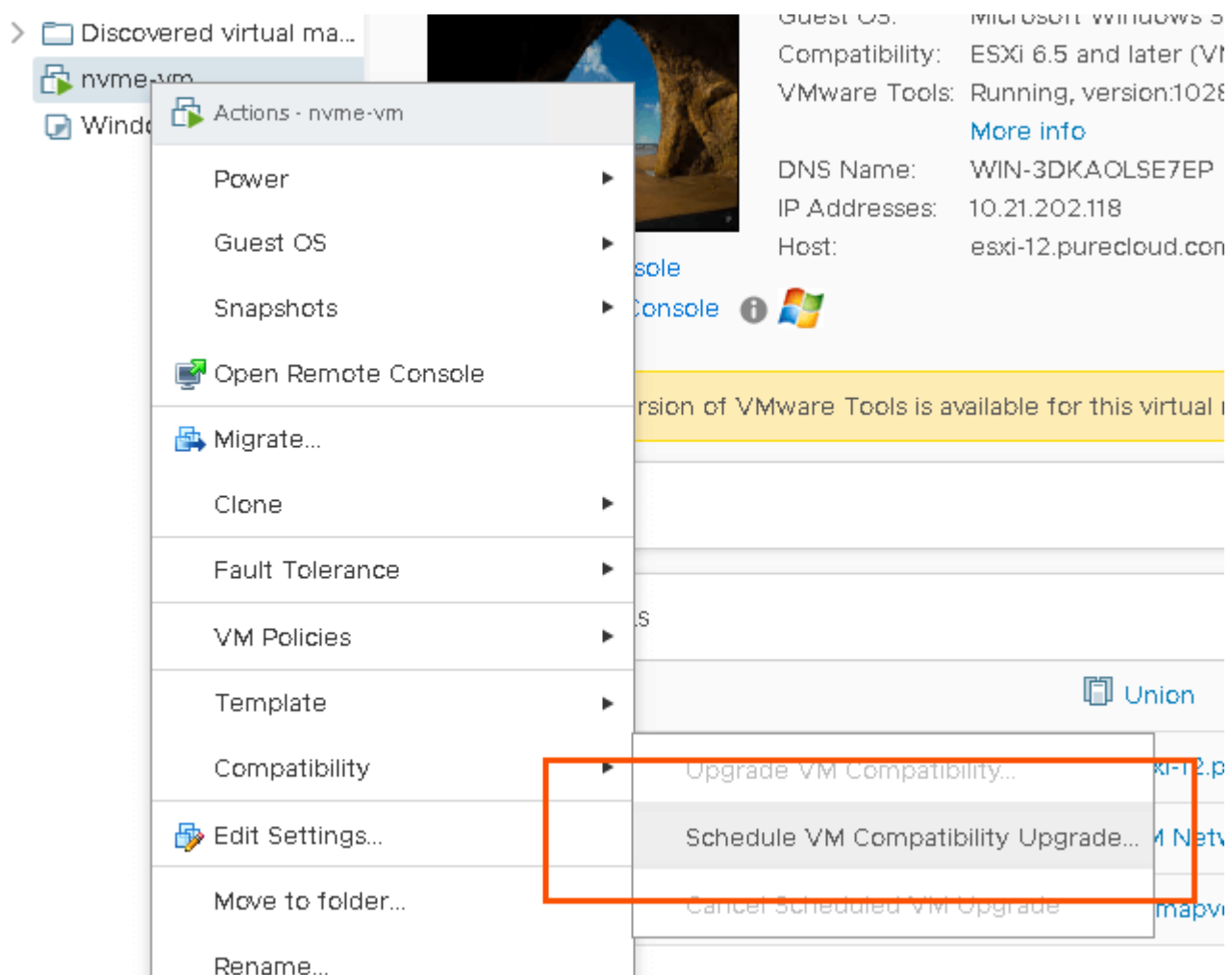
Perform a quick format  
 Enable file and folder compression

I will use the Windows -based tool [SCSI Explorer](#) to look at the device inquiry information. As you can see, in the LBPV page, UNMAP support is listed as 0, which means it is not supported:

```
.....  
.093 Peripheral qualifier 0, peripheral device type 0x00 (0), page  
.093 age length 4  
.093 Logical Block Provisioning (B2h) page  
.093 Threshold exponent 0x00 (0)  
.093 UNMAP command is supported (LBPUS) 0  
.093 WRITE SAME (16) command can be used to unmap LBAs (LBPWS)  
.093 WRITE SAME (10) command can be used to unmap LBAs (LBPWS1)  
.093 Logical block provisioning read zeros (LBPRZ) 0  
.093 Anchored LBAs are supported (ANC_SUP) 0  
.093 Descriptor present (DP) 0  
.093 Provisioning type 0  
.093 "Inquiry (12h)" command completed, adapter status 0 <Success>  
.....
```

Furthermore, if you use the Optimize Drives tool, which allows you to manually run UNMAP/TRIM/DEALLOCATE in Windows, the drive is listed as no optimization available.

Now, let's upgrade to VM hardware version 14.



## Schedule VM Compatibility Upgrade

nvme-vm



Select a compatibility to upgrade the virtual machine on the next reboot.

Compatible with: ESXi 6.7 and later ⓘ

This virtual machine uses hardware version 14, which provides the best performance and latest features available in ESXi 6.7.

Only upgrade after normal guest OS shutdown

CANCEL

OK

Now we are upgraded:

nvme-vm | ACTIONS ▾

Summary | Monitor | Configure | Permissions | Datastores | Networks

Guest OS: Microsoft Windows Server 2016 (64-bit)  
Compatibility: **ESXi 6.7 and later (VM version 14)**  
VMware Tools: Running, version:10267 (Upgrade available)  
[More info](#)

DNS Name: WIN-3DKAOLSE7EP  
IP Addresses: 10.21.202.118  
Host: esxi-12.purecloud.com

[Launch Web Console](#)  
[Launch Remote Console](#) ⓘ

Note that this is a VM hardware version upgrade—this does not require upgrading the VM tools software.

When back in Windows, we can see UNMAP support is now listed as 1, meaning it is supported.

```
43.135 age length 4
43.135 Logical Block Provisioning (B2h) page
43.135 Threshold exponent 8x88 (8)
43.135 UNMAP command is supported (LBPV) 1
43.135 WRITE SAME (16) command can be used to unmap LBAs (LBPWS) 0
43.135 WRITE SAME (10) command can be used to unmap LBAs (LBPWS10) 0
43.135 Logical block provisioning read zeros (LBPRZ) 0
43.135 Anchored LBAs are supported (ANC_SUP) 0
43.135 Descriptor present (DP) 0
43.135 Provisioning type 2
43.151 "Inquiry (12h)" command completed, adapter status 0 (Success), SCSI at
```

Optimize drives also reports it works:

You can optimize your drives to help your computer run more efficiently, or analyze them to find out if they need to be optimized. Only drives on or connected to your computer are shown.

## Status

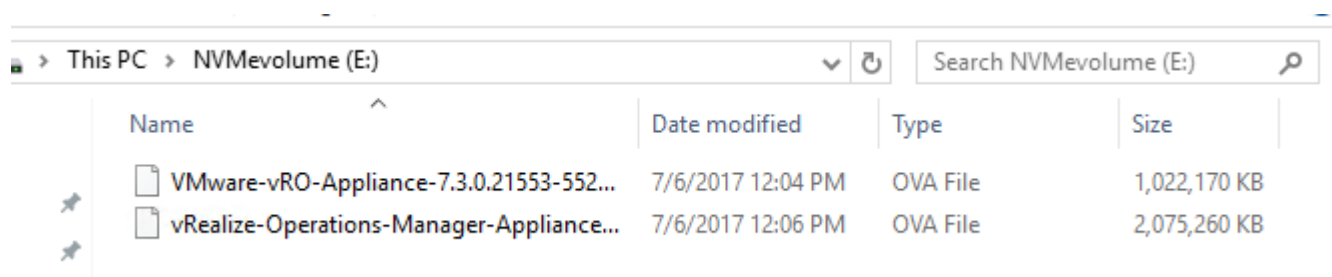
Drive	Media type	Last run	Current status
(C:)	Thin provisioned drive	4/20/2018 10:38 AM	OK (80% space efficiency)
NVMevolume (E:)	Solid state drive	4/20/2018 11:22 AM	OK (0 days since last run)
System Reserved	Thin provisioned drive	Never run	OK

So let's actually test it.

My virtual disk is currently 80 MB in size:

```
nvme-vm.vmx~
86016 nvme-vm_1-flat.vmdk
nvme-vm_1.vmdk
```

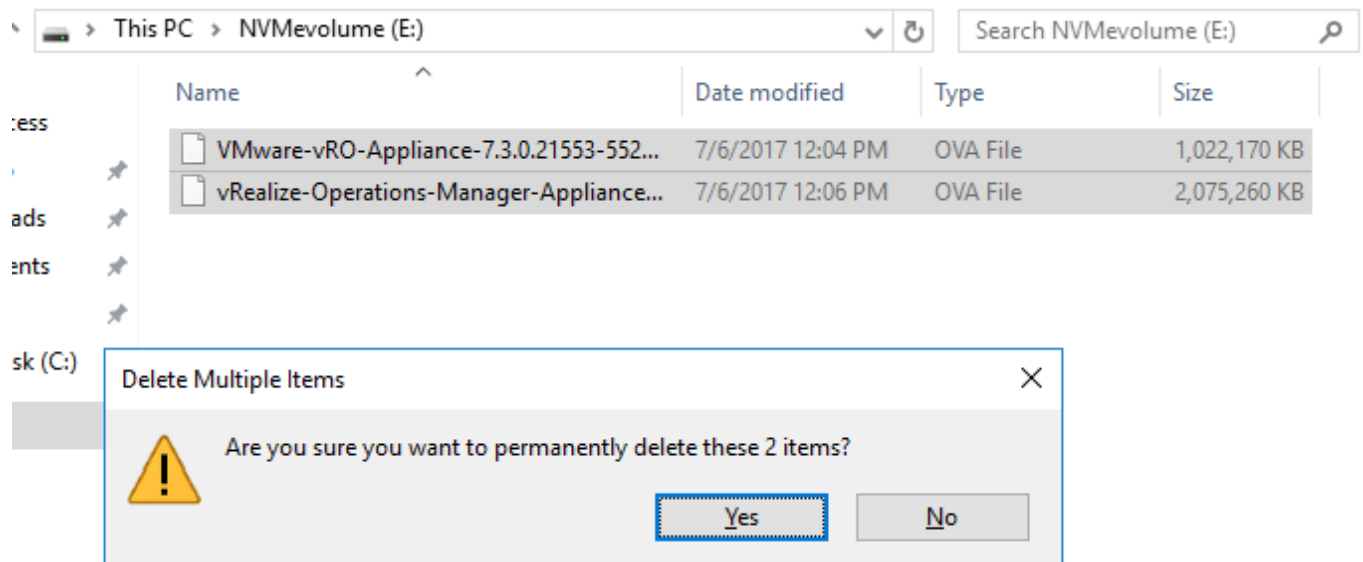
Now I will put some data on it:



And my VMDK is now 3 GB:

```
nvme-vm.vmx~
3172352 nvme-vm_1-flat.vmdk
nvme-vm_1.vmdk
```

Now to delete the files from the volume:



The VMDK shrinks accordingly about 10 seconds later down to about the original size 120 MB:

```
nvme-vm] ls -s
0 nvme-vm_1.vmdk
0 nvme-vm.vmx
0 nvme-vm.vmx~
119808 nvme-vm_1-flat.vmdk
0 nvme-vm_1.vmdk
```

So it works!

# Conclusion

So what is the final requirements?

In my testing Windows 2012 R2 and 2016 both work. I have not tested Linux yet, so I will try it out and report back. But to confirm:

- ESXi 6.7
- VM hardware 14
- Thin virtual disk

So what does this mean? Should you start using this instead of PVSCSI? Maybe. There are potential performance benefits, but VMware does warn it might actually not help in some places too. This [doc](#) says:

*“Because the vNVMe virtual storage adapter has been designed for extreme low latency flash and non-volatile memory based storage, it isn’t best suited for highly parallel I/O workloads and slow disk storage. For workloads that primarily have low outstanding I/O, especially latency-sensitive workloads, vNVMe will typically perform quite well. “*

This seems a bit strange to me, but I need to look into this more. The introduction of reclamation support opens it up to be a contender over PVSCSI, but I am not quite willing to give it the champion title until I know more and do more testing.

In short though, keep it on your radar. VMware introduced a lot more NVMe support into 6.7 and my guess is this is just the start. So I imagine this controller will only continue to become more important moving forward. Stay tuned for more.