

# VMware silently adds native USB 3.0 support to ESXi 5.5

by Andreas Peetz at Monday, November 03, 2014



The October 2014 patch of ESXi 5.5 already got a lot of attention, because it introduced additional Transparent Page Sharing (TPS) management capabilities to prepare for TPS being disabled by default in upcoming ESXi releases. And William Lam pointed out that this patch enables support for the Apple Mac Pro 6,1 ... but there is even more goodness in this patch!

When adding the associated Image Profiles to the [VibMatrix I](#) noticed that this bundle includes a *new* package named *xhci-xhci*. The related [KB2087362](#) article only includes the standard disclaimer, but no information about what this really means: *xHCI* stands for [Extensible Host Controller Interface](#), a USB standard that supports USB 3.0 (or SuperSpeed) controllers and devices.

That means with the latest ESXi 5.5 patch you are - for the first time - able to utilize *USB Passthrough* with USB 3.0 devices!

Here are the steps to get this working:

## 1. Install the ESXi 5.5 October 2014 patch

With an Internet connected standalone ESXi host you can use the well known `esxcli` method to patch the host directly from the VMware Online Depot:

Enable SSH access on your host, log in to it (e.g. using `putty`) and run the following commands:

```
1 # open firewall for outgoing http requests:
2 esxcli network firewall ruleset set -e true -r httpClient
3 # Update using the ESXi 5.5 p03 Imageprofile from the
  VMware Online depot
```

```
4  esxcli software profile update -d
   https://hostupdate.vmware.com/software/VUM/PRODUCTION/
   main/vmw-depot-index.xml -p
   ESXi-5.5.0-20141004001-standard
5  # Reboot your host
6  reboot
```

If you do not have Internet access with your ESXi host then you need to [download the Patch Offline Bundle](#) from MyVMware for Offline patching.

Please note: This patch bundle also includes the ESXi 5.5 Update 2 that was released earlier, because [ESXi patches are cumulative](#).

## 2. Load the xhci VMkernel module

After you have booted your newly patched ESXi host you can check with the shell command `lsusb` what USB devices were detected. You will probably not see any USB 3.0 devices then, because the necessary `xhci` driver module is *not loaded automatically* (at least not on my whitebox that uses an add-on Renesas chip based controller).

Go and load the module with the command

```
vmkLoad_mod
xhci
```

in an ESXi shell. After that `lsusb` should also list your USB 3.0 device(s).

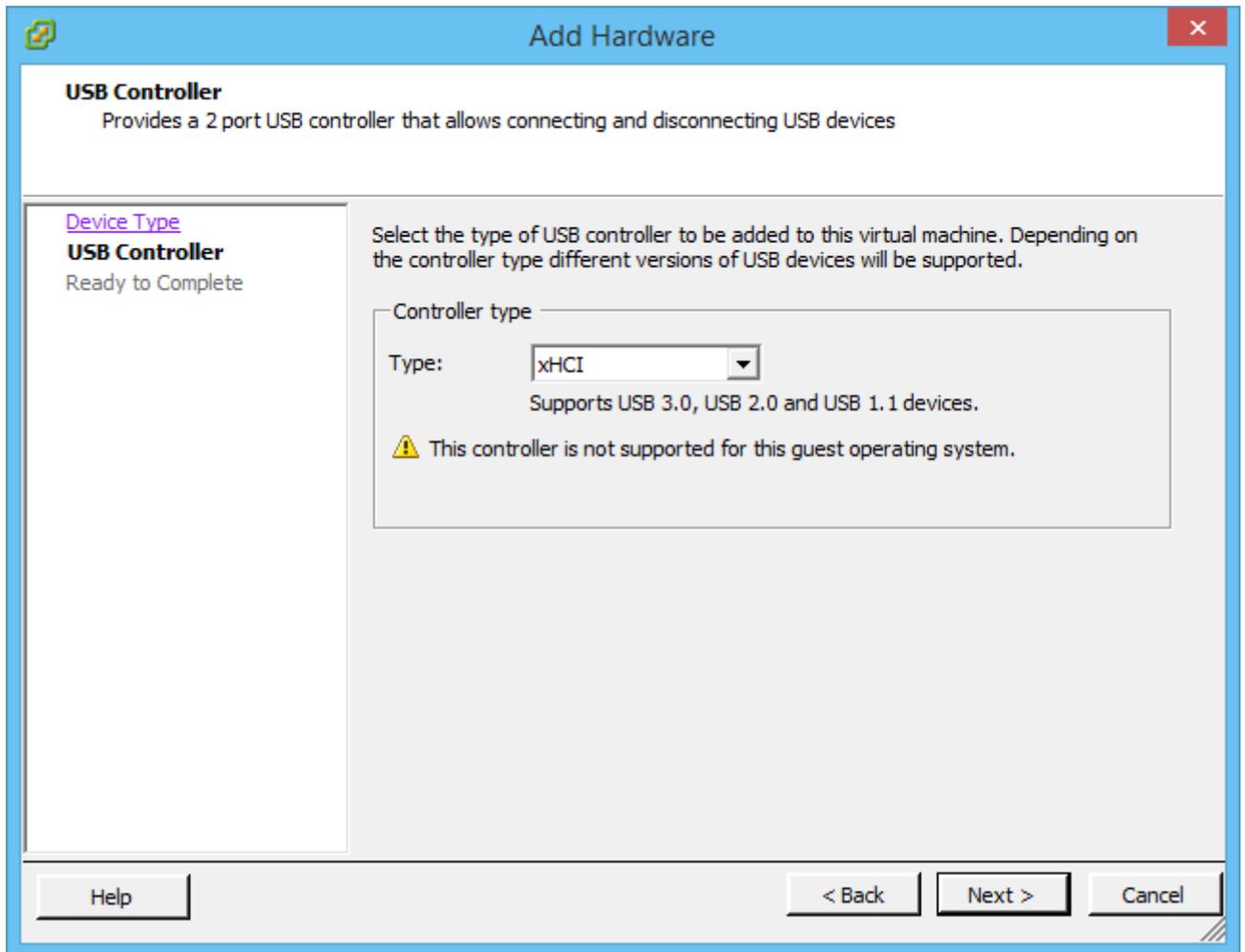
If you want to make use of your USB 3.0 devices on a regular basis then I recommend adding this command to the file `/etc/rc.local.d/local.sh`. Then it will always be automatically executed at boot time.

## 3. Configure USB passthrough

The method to use a host-connected USB device with a VM is called *USB passthrough*, and the steps to configure it are described [in the vSphere docs](#). Here is a short summary:

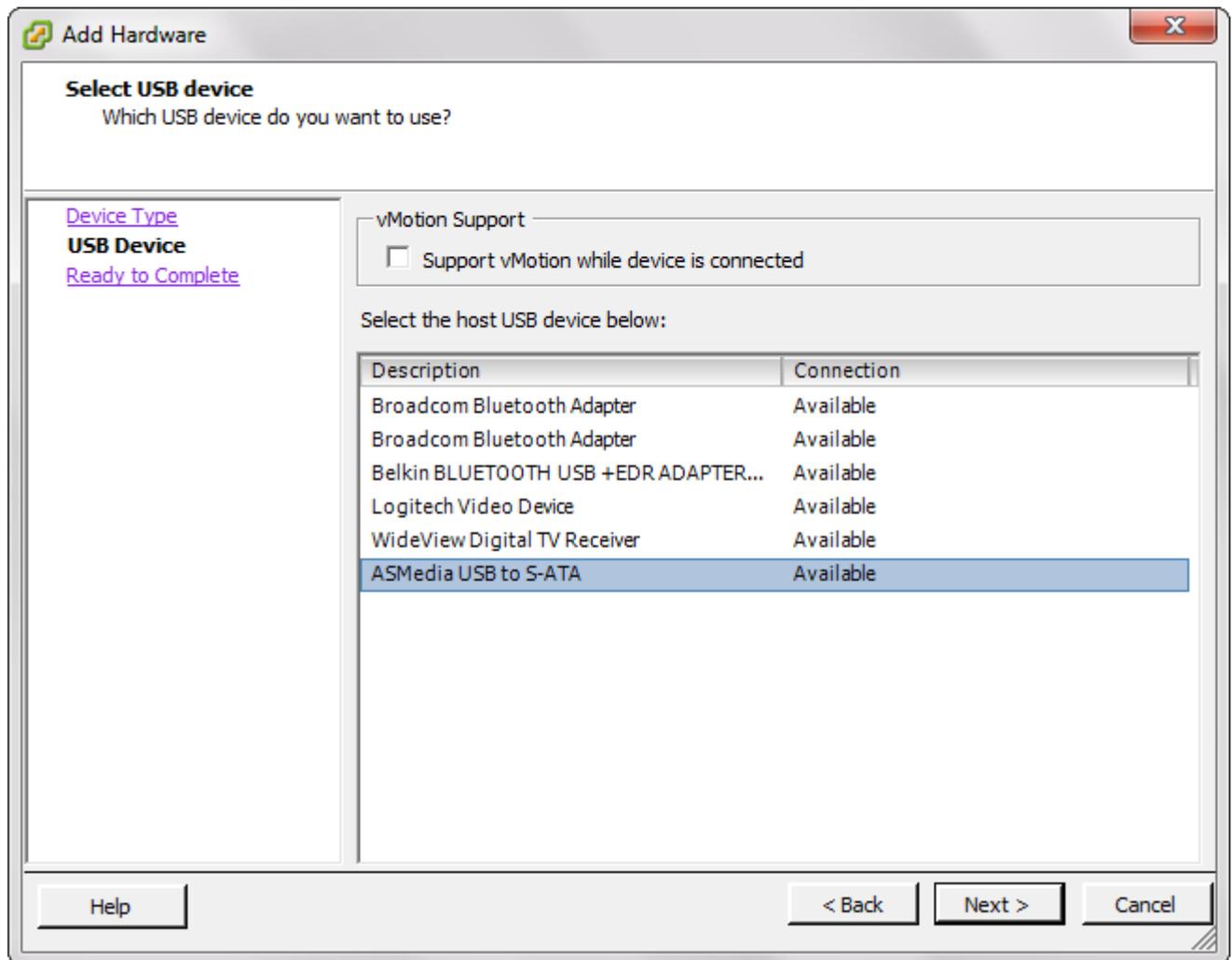
In the hardware properties of the VM add a USB controller. If you want to use a USB 3.0 device at full speed then you need to add a

controller of type *xHCI*. Otherwise the device will only work with USB 2.0 speed:



Please note: This will only work with Guest Operating Systems that include a generic USB 3.0/xHCI driver, and that is any Linux (with kernel version 2.6.38 and later), Microsoft Windows 8/8.1 and Server 2012 (R2), but not Windows 7 (resp. Server 2008 R2) or earlier versions of Windows!

The next step is to add the host-connected USB device to the VM. Add a USB device in its hardware properties and map it to the host device by picking the right one from the presented list:



You can now power on the VM and use the USB device with it!

## Test results

I tested the performance of USB 3.0 passthrough on my whitebox ESXi system with a Windows Server 2012 R2 VM using an external 3TB hard disk and a 64GB thumb drive. I used the free [ATTO Disk Benchmark tool](#) to measure read and write throughput with various block sizes and got the following results:

- CN Memory 3TB External hard disk (using an ASMedia USB to SATA converter): max. 90 MB/s write, max. 140 MB/s read
  - ADATA 64GB flash drive: max. 45 MB/s write, max. 90 MB/s read
- I am aware that these values are far from the theoretically possible USB 3.0 transfer speeds, but I ran the same benchmarks on Windows 8 running *natively* on the same hardware and the results were only slightly better (+5%).

## Summary and conclusion

The latest ESXi 5.5 patch enables you to pass host-connected USB 3.0 devices through to VMs (Linux and Windows 8+, 2012+) where you can achieve great *near-native performance*.

Besides from USB passthrough there is another way to make use of host-connected USB 3.0 devices. Unfortunately you cannot format a USB disk with VMFS to use it as a datastore, but William Lam has described [how to use large USB keys to automate VM deployments using ESXi kickstart](#). This makes it possible to fully automate the provisioning of ESXi and a set of virtual machines as part of the initial deployment. Using a USB 3.0 device for this process will speed it up significantly!

**Update (2014-11-03):**

[William Lam](#) pointed out that there is a better way to enable the autoloading on boot for the xhci module. Use this `esxcli` command once instead of editing the local boot script:



**William Lam** @lamw

Folgen

[@VFrontDe](#) [@tinkererguy](#) no need for hokey local.sh, just use "`esxcli system module set -e true -m xchi`" & can load using ESXCLI too

16:51 - 3 Nov 2014

Antworten

Retweeten

Favorit

You never stop learning ;-) Thanks William! **But** ... there is a typo in William's tweet (it's *xhci*, not *xchi*), and after [Paul Braren](#) and I tested this it turned out that it doesn't really work as expected: You can load the xhci module using

```
esxcli system module load -m  
xhci
```

instead of `vmkload_mod xhci`, but the command

```
esxcli system module set -e true -m  
xhci
```

will *not* lead to the module being automatically loaded at boot time.  
So you still need my workaround with editing `/etc/rc.local.d/local.sh`.