



USB Device Redirection, Configuration, and Usage in VMware Horizon with View

View 5.1 and Later, Through Horizon 6.0.1 with View

WHITE PAPER

Table of Contents

Introduction	4
USB Redirection Overview	4
Definitions of Terms	5
Preliminary Information About USB Devices and View	6
Automatically Blocked Devices	6
USB Devices and the WAN	6
Support of Webcams	6
Support for USB 3	6
Number of USB Devices Which Can Be Plugged into the Client Computer	7
Support for USB Redirection in the Various Horizon Clients	7
Preventing Some USB Devices from Being Redirected to the View Desktop	7
Disabling Redirection for All USB Devices	7
USB Redirection and the RDP and PCoIP Display Protocols	7
USB Redirection and Zero Clients	7
USB Redirection and RDS-Hosted Applications and Desktops	7
USB Redirection and Lack of Direct Network Access to Virtual Desktops	7
USB Basics	9
USB VID and PID	9
Wildcards in USB Device VIDs and PIDs	9
USB Device Families	10
USB Device Filtering	11
USB Interfaces	11
USB Device Splitting	11
Autoconnecting USB Devices to a Virtual Desktop	12
View Agent and Horizon Client Configuration	13
Modifier on the View Agent Policy Setting	13
Effect of Merge and Override Modifiers on Boolean Settings	13
View Agent Configurations	15
Order of Precedence for View Agent Settings	15
Horizon Client Configurations	16
Order of Precedence for Horizon Client Settings	16
Special Details on Horizon Client Settings	16
Details on All USB GPO Configuration Options (Client and Agent)	17
USB Log Analysis and Debugging	21
Enabling Logging	21
Log File Location	21
Using the Horizon Client Logs to Diagnose Configuration Issues	22

Where to Configure USB Redirection	23
Windows Configuration of USB Redirection	23
Linux Configuration of USB Redirection	24
Mac OS X Syntax for Configuring USB Redirection	24
Disabling USB Redirection	26
Practical Worked Examples	27
Simple Filtering Examples	27
Blocking a Single Device from Appearing	27
Blocking All Storage Devices from Appearing in a Desktop Pool (for All Users)	27
Blocking All Devices from Appearing in the Horizon Client Menu	27
Blocking All Devices Except One	28
Blocking All Devices Except for Mass Storage Devices	28
Blocking All Devices Made by One Vendor	28
Chaining Multiple IncludeVidPid Rules to Include Two Devices, but Exclude All Others	28
Using Real-Time Audio-Video and Not Forwarding USB Audio and Video Devices	28
Troubleshooting a Blocked Device	29
Configuring Splitting	31
Configuring Splitting, Example 1	31
Configuring Splitting, Example 2	32
Configuring Splitting: Dictaphone Device-Splitting Examples	33
Summary	35
About the Author	35

Introduction

VMware® Horizon™ with View™ 5.1 and later introduced some complex configuration options for the usage and management of USB devices in a View virtual desktop session. This white paper gives a high-level overview of USB remoting, discusses the configuration options, and provides some practical worked examples to illustrate how these options can be used.

USB Redirection Overview

We are all familiar with using USB devices on laptop or desktop machines. If you are working in a virtual desktop infrastructure (VDI) environment such as View, you may want to use your USB devices in the virtual desktop, too. *USB device redirection* is a function in View that allows USB devices to be connected to the virtual desktop as if they had been physically plugged into it. Typically, the user selects a device from the VMware Horizon Client menu and selects it to be forwarded to the virtual desktop. After a few moments, the device appears in the guest virtual machine, ready for use.

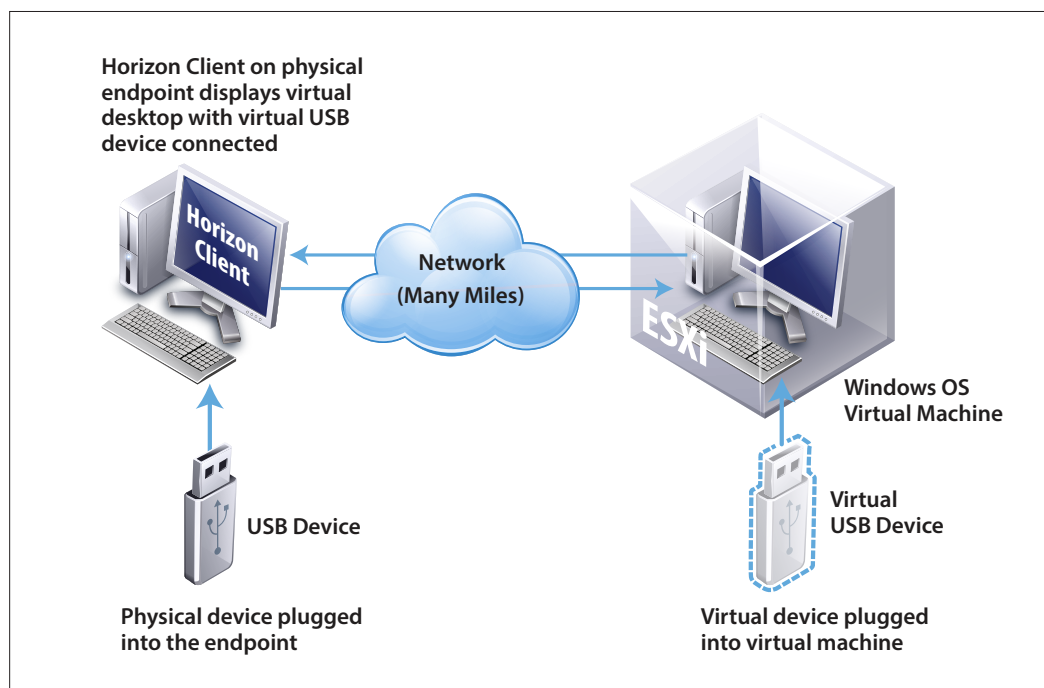


Figure 1: USB Redirection

The USB devices that the administrator has enabled for redirection to the View desktop appear in the Connect USB Device menu at the top of a Windows or Mac Horizon Client (but not Linux). When the user chooses a USB device from this menu, that device becomes redirected to the View desktop.



Figure 2: Available USB Devices As Seen by an End User on a View Desktop

Definitions of Terms

In this paper, various terms are used to describe the components involved in USB redirection. The following are some brief definitions of terms:

- **USB redirection** – Forwarding of the functions of a USB device from the physical endpoint to the View virtual machine.
- **Client computer, or client, or client machine** – Physical endpoint displaying the virtual desktop with which the user interfaces, and where the USB device is physically plugged in.
- **Virtual desktop or guest virtual machine** – The Windows desktop stored in the data center that is displayed remotely on the endpoint. This virtual desktop runs a Windows **guest operating system**, and has the View Agent installed on it.
- **Soft client** – Horizon Client in software format, such as a Horizon Client for Windows or Linux. The soft client is installed on a hardware endpoint, such as a laptop, and displays the virtual desktop on the endpoint.
- **Zero client** – A hardware-based client used to connect to a View desktop. Stateless device containing no operating system. Delivers the client login interface for View.
- **Thin client** – A hardware device similar to a zero client, but with an OS installed. The Horizon Client is installed onto the OS of the thin client. Both devices generally lack local user-accessible storage and simply connect to the virtual desktop in the data center.
- **USB interface** – A function within a USB device, such as mouse or keyboard or audio. Some USB devices have multiple functions and are called composite (USB) devices.
- **Composite (USB) device** – A USB device with multiple functions, or interfaces.
- **HID** – Human interface device. A device with which the user physically interacts, such as mice, keyboards, and joysticks.
- **VID** – The vendor identification, or code, for a USB device, which identifies the vendor that produced the device.
- **PID** – The product identification, or code, which, combined with the VID, uniquely identifies a USB device within a vendor's family of USB products. The VID and PID are used within View USB configuration settings to identify the specific driver needed for the device.
- **USB device filtering** – Restricting some USB devices from being forwarded from the endpoint to the virtual desktop. You specify which devices will be prevented from being forwarded: individual VID-PID device models, device families, such as storage devices, or devices from specific vendors.
- **USB device splitting** – The ability to configure the USB device such that when connected to a View desktop leaves some of the USB interfaces local to the client endpoint, and other interfaces forwarded to the guest. This can result in an improved user experience of the device in a virtual environment.
- **USB Boolean settings** – Simple “on” or “off” settings. For example, whether a specific feature is enabled (true) or disabled (false).

Preliminary Information About USB Devices and View

Before you embark on setting up your devices for USB redirection in View, you need to know a few specifics.

Automatically Blocked Devices

View does not explicitly prevent any devices from working; however, due to various factors such as network latency and bandwidth, some devices do not work well in a VDI environment. By default, some devices are automatically “filtered,” or blocked from being used (for example, mice and keyboards); however, this can be overridden. The section on [USB device filtering](#) below explains this in more detail.

USB Devices and the WAN

Due to the way that USB storage devices work, performance in a virtual desktop can be slow over a WAN. This is due to the fact that before the USB device can appear for use in the Windows guest operating system, the file structure needs to be read from the device. The file structure can be very large depending on how the device has been formatted, and can take significant time to read, so the device may take a long time to appear for use. There are some tricks that can help improve the performance, for example, formatting a USB device as NTFS rather than FAT helps to decrease the initial connection time. The KB article [Redirecting a USB flash drive might take several minutes](#) explains this in more detail.

The performance of the redirected USB device varies greatly depending on network latency and reliability. For example, a single USB storage device read-request requires three round-trips between the client and the virtual desktop. A read of a complete file may need multiple USB read operations, and the larger the latency, the longer the round trip takes. An unreliable network link causes retries, and the performance may be further reduced.

For this reason, some devices do not work well over a latent network such as a WAN. Examples of this include USB DVD writers, which require a steady bit rate of data to allow the burn operation to complete correctly, or USB audio and video devices, which require low latency for the data to be useful. Scanners and touch devices such as signature tablets also do not work well over a latent unreliable network.

For testing, you can simulate WAN environments in a virtualized environment with tools such as WANem or hardware network emulators. This simulation can be useful for testing specific device performance in a virtual desktop over latent or unreliable networks in advance of deploying the USB devices to end users.

Support of Webcams

Due to the bandwidth requirements of webcams, which typically consume >60Mbps, webcams are not supported via USB redirection; however, in View 5.2 and later we support these devices using the [Real-Time Audio-Video \(RTAV\)](#) capability.

Support for USB 3

USB 3 devices, as well as USB 2 and USB 1 devices, are now supported with a combination of View Agent 6.0.1 and Horizon Client 3.1. Although USB 3 devices are supported in View, due to the limitations of any network (latency, bandwidth, reliability, and so on), it is not possible to achieve USB 3 “super” speeds with USB redirection to a virtual desktop. The higher the latency, the lower the throughput. If your network has high latency, you have slower performance with lower throughput than if the devices were used locally. It is not expected that there will be any performance enhancements from using a USB 3 device compared to a USB 2 device when connected to a View desktop.

For earlier versions of View Agents or Horizon Clients that do not support USB 3, USB 3 devices often work in USB 2 mode when plugged into a USB 2 port on the client machine. This method should always work when running Windows 8. However, we have found that on other operating systems, depending on the USB chipset on the client motherboard, USB 3 devices may not work properly in USB 2 mode when redirected to the virtual desktop.

Be sure to upgrade to the latest View Agent and Horizon Client versions to take advantage of USB 3 support.

Number of USB Devices Which Can Be Plugged into the Client Computer

In practice, we are not aware of any customer hitting the limits of the number of USB devices that can be plugged into the client computer. Within the guest virtual machine desktop, there is a limit of 32 USB devices. On the client side, which is shared between the client OS, VMware Workstation or VMware Fusion, and View, there is a limit of up to 16 host controllers, each 4 hubs deep with 16 ports per hub. So, yes, there is a limit, but in practice it is unlikely to cause any problems for normal use-cases.

Support for USB Redirection in the Various Horizon Clients

USB redirection in View is supported on Windows, Linux (x86, ARM, and ARM hard-float), and Mac OS X Clients. USB redirection is not currently supported on iOS and Android clients.

Linux does not currently have a UI allowing manual user selection of USB devices to forward; all devices are automatically forwarded by default. All Horizon Clients support splitting and filtering. The examples detailed in the [Practical Worked Examples](#) section should apply to any platform. However, note that the method of configuration and the location of log files vary between platforms.

Preventing Some USB Devices from Being Redirected to the View Desktop

You can prohibit some USB devices from being redirected to the View desktop. The section in this paper on [device filtering](#) explains how this can be done.

Disabling Redirection for All USB Devices

You can prevent every USB device from being redirected to the View desktop. For details, see [Disabling USB Redirection](#).

USB Redirection and the RDP and PCoIP Display Protocols

View USB redirection works independently of the display protocol, and works with both RDP and PCoIP remote display protocols. USB traffic uses TCP port 32111 for both RDP and PCoIP sessions.

USB Redirection and Zero Clients

USB redirection is also supported in Horizon zero clients. Teradici writes the Horizon Client USB code for a zero client.

The USB traffic is redirected using a PCoIP virtual channel (rather than via TCP 32111), and is encapsulated and encrypted inside the standard PCoIP traffic TCP/UDP 4172.

Within the View Agent on the guest, the Teradici USB traffic is then presented to the VMware virtual hub driver as normal. It is possible, therefore, to see some performance and functional differences with USB redirection between a software Horizon Client, such as on Windows or OS X, and a zero client.

If you are using *only* zero clients, then it is not necessary to open TCP port 32111 for USB redirection to work.

USB Redirection and RDS-Hosted Applications and Desktops

Horizon with View does not yet support USB redirection with RDS-hosted applications and desktops, which are multi-session. USB redirection is supported only with single-session View.

USB Redirection and Lack of Direct Network Access to Virtual Desktops

Typically, for deployments that are not able to route network traffic directly between client systems and desktops, often because the desktops are located on internal networks in the “green zone,” the HTTPS secure tunnel or PCoIP Secure Gateway components must be enabled. This allows traffic, including USB traffic, to be routed through the View Connection Server instead of directly from the client to the desktop.

A security server is a special instance of View Connection Server that runs a subset of View Connection Server functions. You can use a security server to provide an additional layer of security between the Internet and your internal network.

A security server resides within a DMZ and acts as a proxy host for connections inside your trusted network.

This design provides an additional layer of security by shielding the View Connection Server instance from the public-facing Internet and by forcing all unprotected session requests through the security server.

A DMZ-based security server deployment requires a few ports to be opened on the firewall to allow clients to connect with security servers inside the DMZ. You must also configure ports for communication between security servers and the View Connection Server instances in the internal network.

See *Firewall Rules for DMZ-Based Security Servers* in the [View Architecture Planning guide](#) for information on specific ports.

Because users can connect directly with any View Connection Server instance from within their internal network, you do not need to implement a security server in a LAN-based deployment.

The HTTPS secure tunnel and PCoIP Secure Gateway can be enabled via the View Administrator UI, as in Figure 2.

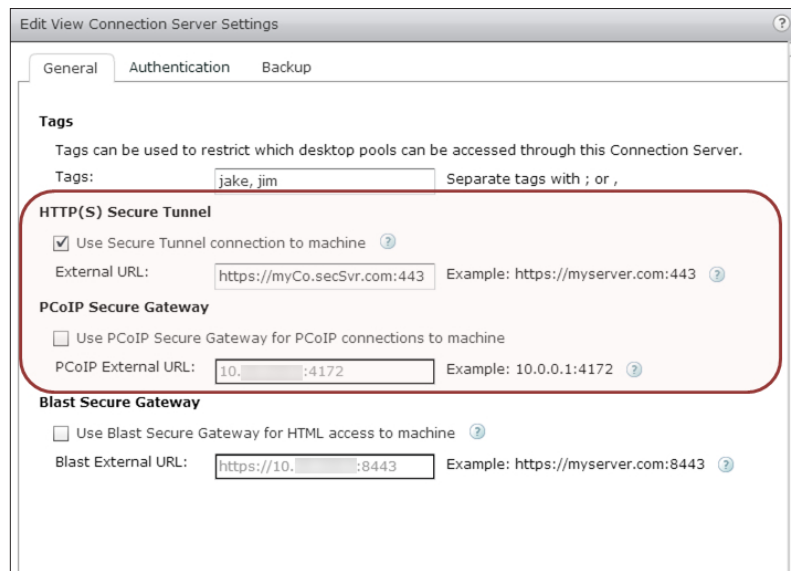


Figure 3: HTTP(S) Secure Tunnel and PCoIP Secure Gateway Configuration

The parameters to configure the secure tunnel and secure gateway are as follows:

- **Use Secure Tunnel connection to machine** – Determines whether the Horizon Client makes a further HTTPS connection to the View Connection Server or security server host when users connect to a View desktop.

If this setting is deselected, the desktop session is established directly between the client system and the View desktop virtual machine, bypassing the View Connection Server or security server host. If the secure tunnel connection is disabled, USB traffic uses TCP port 32111.

If this setting is selected, the connection is tunneled using HTTPS port 443 to the security server, and then the onward connection for USB traffic from the server to the guest desktop uses TCP port 32111.

- **Use PCoIP Secure Gateway for PCoIP connections to machine** – Determines whether the Horizon Client makes a further secure connection to the View Connection Server or security server host when users connect to a View desktop with the PCoIP display protocol. If this setting is deselected, the desktop session is established directly between the client system and the View desktop virtual machine, bypassing the View Connection Server or security server host.

If this setting is selected, and you are using a zero client, then the USB traffic is via the PCoIP Secure Gateway using port 4172.

USB performance is slightly degraded as a result of connections through the secure tunnels.

For more information about the secure tunnels, including firewall considerations, see the [View Architecture Planning guide](#).

USB Basics

USB VID and PID

USB devices are identified primarily by their vendor identification (VID) and product identification (PID). VID and PIDs are unique identifier numbers. A company that wishes to produce USB devices needs to register and pay for a VID. This ID is unique to that supplier. For example, Microsoft has a VID of **0x045E**, and Apple has a VID of **0x05ac**. Depending on how many products the company produces, they may have multiple VIDs within a single company. The product ID is a four-byte identifier that names the specific device. Coupled with the VID, the PID uniquely identifies a driver that the Operating System (OS) must load for a given device. Note that there may be multiple “products” that all use the same VID and PID if they all use the same device driver.

You can see the VID and PID for a device if you look in the device manager. To do this, right-click a device and select **Properties**. Then click the **Details** tab and select **Hardware Ids** from the Property drop-down menu. You can see the VID and PID values reported. In the example below, this Creative Live! Webcam has a VID of **041E** and PID of **4087**.

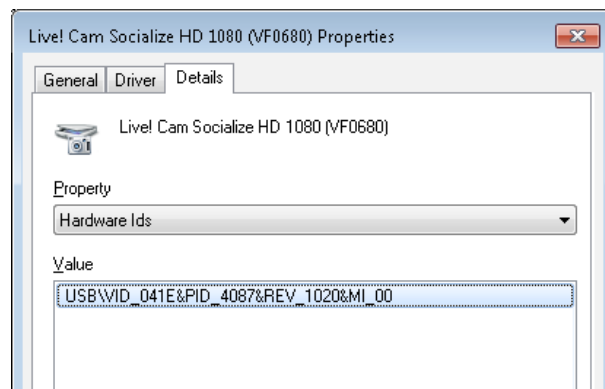


Figure 4: Example of USB Hardware VID and PID in Device Properties Window

Wildcards in USB Device VIDs and PIDs

In USB configurations, you can use the ‘*’ wildcard to indicate unknown characters in the VID and PID specifications.

The standard VID-PID combination in a configuration looks like this:

```
vid-xxxx_pid-yyyy
```

with the number of characters for the VID and PID variable, which is not necessarily four digits long.

To use a wildcard to specify USB devices from any vendor (here, the device type is **5593**):

```
vid-*_pid-5593
```

To use a wildcard to specify all USB devices from one vendor (here, the vendor is **FA11**):

```
vid-FA11_pid-*
```

You can use multiple ‘*’s to indicate the exact number of unknown characters:

```
vid-0781_pid-55**
```

In this example, PIDs have four characters, all starting with ‘55’.

USB Device Families

A USB device can reveal the sort of device it is, for example, a storage device, a webcam video device, and so on. View allows configuration based on this device family level, for example the ability to block all storage devices, but allow everything else. Note that a device does not have to reveal a device family.

The following table shows the device families supported in View 5.3:

DEVICE FAMILY NAME	DESCRIPTION
audio	Any audio-input or audio-output device.
audio-in	Audio-input device such as a microphone.
audio-out	Audio-output device such as a loudspeaker or headphones.
bluetooth	Bluetooth-connected device.
comm	Communications device such as a modem or wired networking adapter.
hid	Human interface device, excluding keyboards and pointing devices. *
hid-bootable	Human interface device that is available at boot time, excluding keyboards and pointing devices.
imaging	Imaging device such as a scanner.
keyboard	Keyboard device. *
mouse	Pointing device such as a mouse. *
other	Device family specified as "other" by the device vendor.
pda	Personal digital assistant.
physical	Force-feedback device such as a force-feedback joystick.
printer	Printing device.
security	Security device such as a fingerprint reader.
smart-card	Smart-card device.
storage	Mass storage device such as a flash drive or external hard disk drive.
unknown	Device family not disclosed.

DEVICE FAMILY NAME	DESCRIPTION
vendor	Device with vendor-specific functions.
video	Video-input device.
wireless	Wireless networking adapter.
wusb	Wireless USB device.
* Although keyboards and mice are HIDs, they are treated differently for USB redirection. Therefore, they are considered different device families within View.	

Table 1: Supported USB Device Families in View

USB Device Filtering

USB device filtering allows specific devices, device families (such as storage devices), or vendor product models to be restricted from being forwarded to the virtual desktop. These rules can be applied locally at the client, or at the virtual desktop in the data center. Administrative group policies (GPOs) can be applied, too, allowing company-wide configurations to be applied across all or some desktops.

USB device filtering is often used by companies to disable the use of mass storage devices on virtual desktops, or perhaps to block a specific device that a user never wants to be forwarded, such as a USB-to-Ethernet adapter which the user is using to connect to the desktop in the first place. Mice and keyboards are by default excluded from redirection.

Complex filter rules can be constructed, for example, to “disallow all products from a specific vendor except for a specific device model which is permitted.” When used in conjunction with USB device splitting, the configuration options can be very powerful. For examples with USB device filtering, see [Simple Filtering Examples](#).

USB Interfaces

Some USB devices perform a single function, and this is referred to as a single device interface. For example, a basic keyboard would most likely be a human interface device (HID), which is a single interface. However, more expensive keyboards can have multiple additional features, or interfaces, included. For example, a keyboard may have the normal buttons, a mouse trackball, a fingerprint scanner, speakers, and so on. This enhanced device would still have only a single physical USB connector, but internally it would have multiple USB “interfaces.” This device would have HID, mouse, security, audio interfaces, and so on.

A *composite USB device* has multiple USB interfaces.

USB Device Splitting

It is sometimes useful to “split” a USB device, whereby some of the device interfaces for a composite USB device are left local to the client machine, and other interfaces are forwarded to the guest virtual machine.

In View 5.1 and later, it is possible to split a composite device such that some parts of the device, such as the mouse, are left local to the client machine, and other parts are forwarded to the virtual desktop. This can result in a much more effective user experience. For example, in the case where a device contains a mouse-type device, if you forward the mouse to the guest, the mouse can only operate inside the Horizon Client window that holds the guest virtual machine. However, if the mouse is left local to the client machine, then it can be used on the client machine as well as inside the guest virtual machine. For details on USB device splitting, see [Configuring Splitting](#).

Autoconnecting USB Devices to a Virtual Desktop

Configuration options allow USB devices to be forwarded automatically to the virtual desktop after they are physically connected to the client device. There are two configuration options that control autoconnection, and these can be set on the Horizon Client via GPO/configuration options:

- **connectUSBOnStartup**: Connect all USB devices to the desktop on launch
- **connectUSBOnInsert**: Connect USB devices to the desktop when they are plugged in

Any device filtering or splitting configurations override these automatic settings.

Alternatively, on Windows and Mac (but not Linux) clients, the Connect USB Device menu on the Horizon Client allows manual selection of which devices are forwarded.

View Agent and Horizon Client Configuration

You can define USB redirection policy settings for both View Agent and the Horizon Client. This section examines View Agent and Horizon Client configuration settings, and explains the interaction of these two configuration locations.

Some configuration options are applied and actionable only at the client machine; for example, split settings, which direct View to split a device's functions between the client and host machines. These client-only settings do not have to be configured on every client machine, however. You can configure them on the guest desktop pool via the GPO template to avoid having to individually configure every client device. These settings are under the **Client Downloadable only** Settings in the GPO template. When the client machine first connects to a View session, these client configuration items are downloaded to the client machine and apply only to the client, not to the agent machine. In addition, there are some configuration items that are set on both the client and the agent machines, for example, Include/Exclude rules. View combines the View Agent settings and the downloaded client settings with any settings that have been configured locally on the client machine and determines which devices can be redirected from the host machine to the virtual desktop. To determine which setting to use when there are duplicate settings, View consults override and merge options that you configure for each setting.

Modifier on the View Agent Policy Setting

On the View Agent, you set a modifier to the policy setting to determine if the View Agent settings override any settings on the Horizon Client, or whether they should be combined with the Horizon Client settings.

The behavior of View Agent USB redirection policy can be controlled via the View Agent modifier, either **m** or **o**:

VIEW AGENT MODIFIER	RESULT OF USING THIS MODIFIER WITH A CONFIGURATION
m (merge)	Horizon Client applies the View Agent policy setting in addition to the Horizon Client policy setting.
o (override)	Horizon Client uses the View Agent policy setting instead of the Horizon Client policy setting.

Table 2: View Agent Merge and Override Configuration Options

For example, if the View Agent is configured like this:

```
IncludeVidPid: o:vid-0911_pid-149a
```

any include VID and PID rules on the client are overridden by this View Agent setting and *only* device **vid-0911_pid-149a** has an Include rule applied.

Note: If you configure the View Agent without the **o** or the **m**, then the configuration rule is invalid and is ignored. This note applies to non-Boolean settings.

Effect of Merge and Override Modifiers on Boolean Settings

The modifiers affect Boolean settings (**Allow...** settings) in a slightly different way:

VIEW AGENT MODIFIER	RESULT OF USING THIS MODIFIER WITH A CONFIGURATION
m (merge) (in command line) Default (in the Group Policy editor)	If Horizon Client settings are configured, View Agent settings are ignored. If Horizon Client settings are not configured, View Agent settings take effect.
o (override) (in both the command line and the Group Policy editor)	Horizon Client settings are overridden regardless.

Table 3: Effect of View Agent Modifiers on Boolean Settings

The **m** modifier means that if Horizon Client settings exist, those are used. If Horizon Client settings do not exist, the View Agent settings are used. The **o** modifier means that the View Agent settings override the Horizon Client settings.

In the Group Policy editor, **m** is expressed as **Default**, and **o** is expressed as **Override**. For example, in the following screenshot of the Group Policy editor for the **Allow Audio Input Devices** setting, each View Agent Boolean value (**Allow** or **Disable**) can be set as **Default** or **Override**.

Allow – Default Client Setting means that the View Agent setting of allowing audio input devices is ignored if a Horizon Client setting for this property exists.

Allow – Override Client Setting would mean that the View Agent setting of allowing audio input devices holds, regardless of the Horizon Client setting. Setting this to **Disable – Override Client** would enforce that this audio input device cannot be used regardless of any settings on the client side.

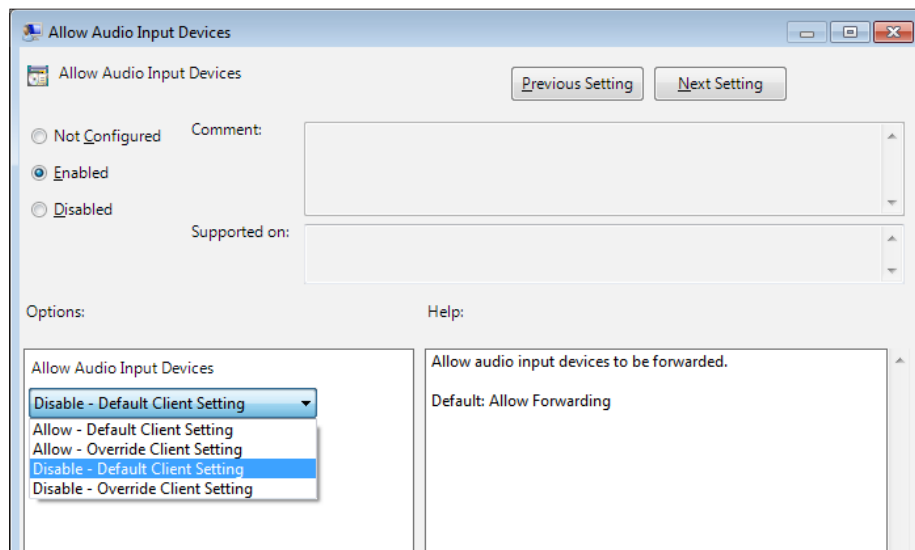


Figure 5: Example of Boolean GPO Modifier Options

An exception to the rule for Boolean settings is **ExcludeAllDevices**.

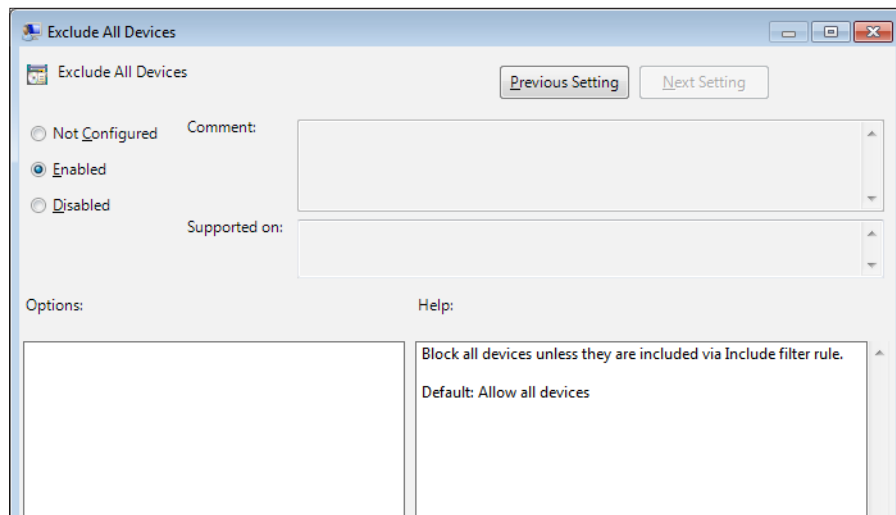


Figure 6: ExcludeAllDevices Boolean Setting

ExcludeAllDevices does not have Default or Override settings. This configuration can be either:

- **Not Configured** (or set to **Disabled**) – All devices are allowed unless explicitly blocked by Exclude rules.
- **Enabled** – All devices are blocked, unless explicitly allowed by Include rules.

View Agent Configurations

This section provides details of the View Agent configurations for USB redirection. The following table lists View Agent split settings, filter settings, and Boolean settings. For full definitions of these settings, see [Details on All USB GPO Configuration Options \(Client and Agent\)](#).

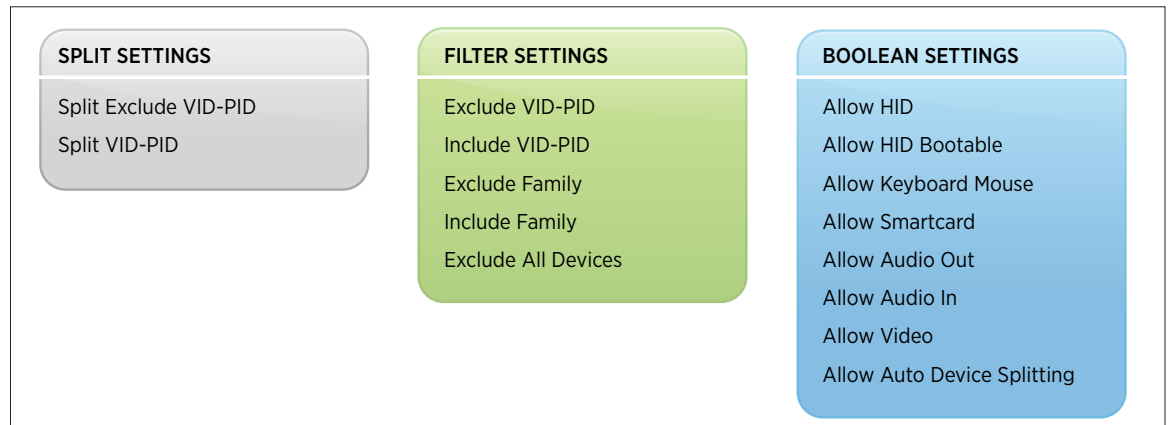


Figure 7: View Agent Configuration Options

Order of Precedence for View Agent Settings

The order in which filter settings are applied on the View Agent is significant, and is worth highlighting here.

View Agent filtering is applied in the following order, where setting 1 is applied first and setting 5 last:

1. Exclude a device by Vendor/Product ID. Wildcards accepted. Default is blank, that is, not set.
2. Include a device by Vendor/Product ID. Wildcards accepted. Default is blank, that is, not set.
3. Exclude a device by USB family. Default is blank, that is, not set.
4. Include a device by USB family. Default is blank, that is, not set.
5. Exclude All Devices. Default is blank, that is, not set.

The other settings are not applied by the View Agent.

Horizon Client Configurations

This section provides details of the Horizon Client configurations for USB redirection. The following table lists Horizon Client split settings, filter settings, and Boolean settings. For full definitions of these settings, see [Details on All USB GPO Configuration Options \(Client and Agent\)](#).

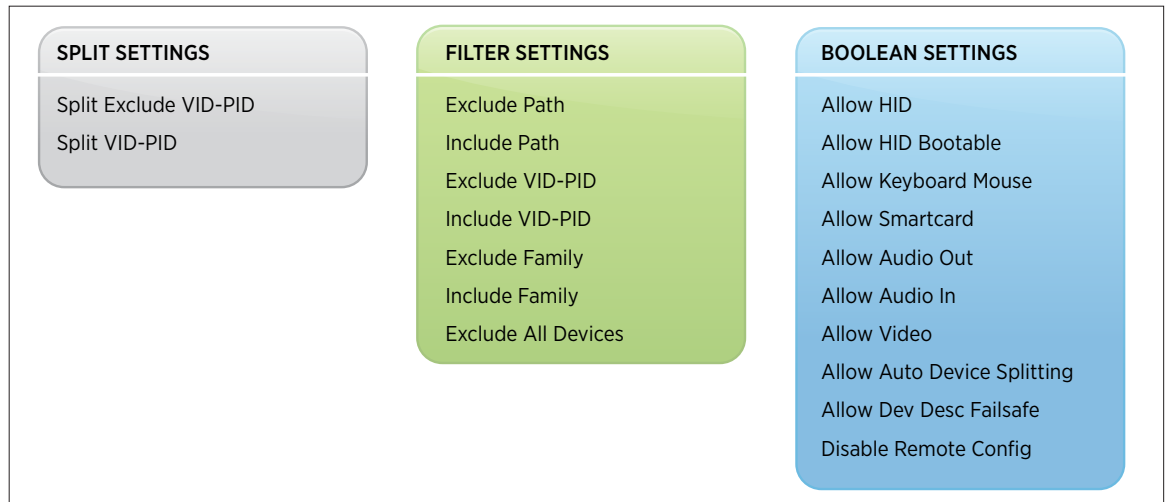


Figure 8: Horizon Client Configuration Options

Order of Precedence for Horizon Client Settings

The order in which filter settings are applied on the Horizon Client is significant, and is worth highlighting here.

Horizon Client filtering is applied in the following order, where setting 1 is applied first and setting 7 last:

1. Exclude a device by hub/port path. Wildcards accepted. Default is blank, that is, not set.
2. Include a device by hub/port path. Wildcards accepted. Default is blank, that is, not set.
3. Exclude a device by Vendor/Product ID. Wildcards accepted. Default is blank, that is, not set.
4. Include a device by Vendor/Product ID. Wildcards accepted. Default is blank, that is, not set.
5. Exclude a device by USB family. Default is blank, that is, not set.
6. Include a device by USB family. Default is blank, that is, not set.
7. Exclude all USB devices. Default is blank, that is, not set.

Special Details on Horizon Client Settings

Following are details on some of the settings that are not so obvious:

AllowAutoDeviceSplitting

(This setting can be configured on the Horizon Client, or configured on the View Agent and pushed down to the Horizon Client.) If this is enabled, View tries to split the functions and interfaces automatically, according to the filter rule settings that are active. For example, a Philips SpeechMike is split so the mouse is terminated locally and the rest of the interfaces are redirected. This happens because the mouse is blocked by default filter rules, but other interfaces are allowed. While this might work, it does not necessarily produce the results that your application needs. For absolute control, configure splitting manually.

AllowDevDescFailsafe

If enabled, this allows devices that could end up being blocked if the device and configuration descriptor(s) are not disclosed by the devices. Note that you have to set explicit IncludeVidPid rules, too. It is not recommended to use this setting unless instructed by VMware technical support.

DisableRemoteConfig

If set to true, the Horizon Client ignores all View Agent filter settings. The Agent still enforces the View Agent filter settings. It is not recommended to use this setting unless instructed by VMware technical support.

Details on All USB GPO Configuration Options (Client and Agent)

The following table gives details on all USB redirection configuration options, for both Horizon Client and View Agent.

CONFIGURATION OPTION	DESCRIPTION	DETAILS
AllowAutoDeviceSplitting	Allow auto device splitting	Allow the automatic splitting of composite USB devices. The default value is undefined, which equates to false .
SplitExcludeVidPid	Exclude VID-PID device from split	Excludes a composite USB device specified by Vendor and Product IDs from splitting. The format of the setting is vid-xxx1_pid-yyy1[;vid-xxx2_pid-yyy2]... You must specify ID numbers in hexadecimal. You can use the wildcard character (*) in place of individual digits in an ID. For example: vid-0781_pid-55** The default value is undefined.
SplitVidPid	Split VID-PID device (and exclude interface by index number)	Treats the components of a composite USB device specified by Vendor and Product IDs as separate devices. The format of the setting is vid-xxxx_pid-yyy(exintf:zz[;exintf:ww]) You can use the exintf keyword to exclude components from redirection by specifying their interface number. You must specify ID numbers in hexadecimal, and interface numbers in decimal including any leading zero. You can use the wildcard character (*) in place of individual digits in an ID. For example: vid-0781_pid-554c(exintf:01;exintf:02) Note: View does not automatically include the components that you have not explicitly excluded. You must specify a filter policy such as Include VidPid Device to include those components. The default value is undefined.

CONFIGURATION OPTION	DESCRIPTION	DETAILS
<code>AllowAudioIn</code>	Allow audio input devices	Allows audio input devices to be redirected. The default value is undefined, which equates to true .
<code>AllowAudioOut</code>	Allow audio output devices	Allows audio output devices to be redirected. The default value is undefined, which equates to false .
<code>AllowHIDBootable</code>	Allow HID bootable devices	Allows input devices other than keyboards or mice that are available at boot time, also known as HID-bootable devices, to be redirected. The default value is undefined, which equates to true .
<code>AllowDevDescFailsafe</code>	Allow device descriptor failsafe behavior	Allows devices to be redirected even if the Horizon Client fails to get the configuration or device descriptors. To allow a device even if it fails to get the configuration or device descriptors, include it in the Include filters, such as IncludeVidPid or IncludePath . The default value is undefined, which equates to false .
<code>AllowKeyboardMouse</code>	Allow keyboard and mouse devices	Allows keyboards with integrated pointing devices (such as a mouse, trackball, or touch pad) to be redirected. The default value is undefined, which equates to false .
<code>AllowSmartcard</code>	Allow smart cards	Allows smart-card devices to be redirected. The default value is undefined, which equates to false .
<code>AllowVideo</code>	Allow video devices	Allows video devices to be redirected. The default value is undefined, which equates to true .
<code>DisableRemoteConfig</code>	Disable remote configuration	Disables the use of View Agent settings when performing USB device filtering. The default value is undefined, which equates to false .

CONFIGURATION OPTION	DESCRIPTION	DETAILS
ExcludeAllDevices	Exclude all devices unless included via Include setting	<p>Excludes all USB devices from being redirected. If set to true, you can use other policy settings to allow specific devices or families of devices to be redirected. If set to false, you can use other policy settings to prevent specific devices or families of devices from being redirected. If you set the value of ExcludeAllDevices to true on View Agent, and this setting is passed to Horizon Client, the View Agent setting overrides the Horizon Client setting.</p> <p>The default value is undefined, which equates to false.</p>
ExcludeFamily	Exclude device family	<p>Excludes families of devices from being redirected. The format of the setting is <code>family_name_1[:family_name_2]...</code></p> <p>For example: bluetooth;smart-card</p> <p>If you have enabled automatic device splitting, View examines the device family of each interface of a composite USB device to decide which interfaces should be excluded. If you have disabled automatic device splitting, View examines the device family of the whole composite USB device.</p> <p>The default value is undefined.</p> <p>Note: However, mice and keyboards are excluded from redirection by default and do not need to be excluded with this setting.</p>
ExcludeVidPid	Exclude VID-PID device	<p>Excludes devices with specified vendor and product IDs from being redirected. The format of the setting is <code>vid-xxx1_pid-yyy1[:vid-xxx2_pid-yyy2]...</code>. You must specify ID numbers in hexadecimal. You can use the wildcard character (*) in place of individual digits in an ID.</p> <p>For example: vid-0781_pid-****;vid-0561_pid-554c</p> <p>The default value is undefined.</p>

CONFIGURATION OPTION	DESCRIPTION	DETAILS
ExcludePath	Exclude device at hub or port paths	<p>Exclude devices at specified hub or port paths from being redirected. The format of the setting is bus-x[/y].../port-z[/;bus-x2[/y2].../port-z2]...</p> <p>You must specify bus and port numbers in hexadecimal. You cannot use the wildcard character in paths.</p> <p>For example: bus-1/2/3_port-02;bus-1/1/1/4_port-ff</p> <p>The default value is undefined.</p>
IncludeFamily	Include device family	<p>Includes families of devices that can be redirected. The format of the setting is family_name_1[;family_name_2]...</p> <p>For example: storage</p> <p>The default value is undefined.</p>
IncludePath	Include device at hub or port paths	<p>Include devices at specified hub or port paths that can be redirected. The format of the setting is bus-x[/y].../port-z[/;bus-x2[/y2].../port-z2]...</p> <p>You must specify bus and port numbers in hexadecimal. You cannot use the wildcard character in paths.</p> <p>For example: bus-1/2_port-02;bus-1/7/1/4_port-0f</p> <p>The default value is undefined.</p>
IncludeVidPid	Include VID-PID device	<p>Includes devices with specified Vendor and Product IDs that can be redirected. The format of the setting is vid-xxx[_pid-yyy[/;vid-xxx2[_pid-yyy2]...</p> <p>You must specify ID numbers in hexadecimal. You can use the wildcard character (*) in place of individual digits in an ID.</p> <p>For example: vid-0561_pid-554c</p> <p>The default value is undefined.</p>

Table 4: Details on All USB GPO Configuration Options

Note: Configuration setting names, such as **IncludeVidPid**, are case-sensitive, but configuration values, such as **vid-1234_pid-5678**, are case-insensitive. This is particularly important when configuring on Linux or OS X clients.

USB Log Analysis and Debugging

In this section, we take a detailed look at log files. The useful log files for USB can be found on both Horizon Client and the guest virtual machine.

If you are trying to configure splitting or filtering, or generally trying to diagnose why a device is not appearing in the Horizon Client menu, then you should look in the Horizon Client logs. On the Horizon Client, there are logs produced for the USB arbitrator and also the View USB services.

If, however, there is a problem with the device once you have forwarded it to the guest desktop, then the logs from both the Horizon Client and the View Agent are useful.

Enabling Logging

Logging on Windows and Linux clients is enabled by default, however it is not enabled by default on the Mac OS X client.

To enable Mac client logging:

Before launching the client, run the following commands from a terminal window (click on Terminal found in Applications\Utilities):

```
launchctl setenv VMWARE_VIEW_DEBUG_LOGGING "1"
launchctl setenv VMWARE_VIEW_USBD_LOG_OPTIONS "-o log:trace"
launchctl setenv VMWARE_VIEW_USBARBITRATOR_LOG_OPTIONS "--debug 3"
```

If you want to permanently enable these, you can edit the `services.sh` file and append these options. The file `services.sh` can be found here:

```
/Applications/VMware Horizon View Client.app/Contents/Library/services.sh
```

Log File Location

The following table shows the location of the various log files pertinent to USB-device redirection.

LOG TYPE	LOCATION
Windows Client	%PROGRAMDATA%\VMware\VDM\logs\debug-*.txt C:\Windows\Temp\vmware-SYSTEM\vmware-usbarb-*.log
Windows Guest (debug logs on the View Agent side)	%PROGRAMDATA%\VMware\VDM\logs\debug-*.txt
Linux Client	/tmp/vmware-root/vmware-view-usbd-*.log by default Can specify using: view-usbd.log.fileName = "/tmp/usbd.log"
Mac Client	/var/root/Library/Logs/VMware/vmware-view-usbd-<process_id>.log
USB Arbitrator (only on the client side)	/Library/Logs/VMware/vmware-usbarbitrator-<process_id>.log

Table 5: Log File Locations for Horizon Clients and Guests

Using the Horizon Client Logs to Diagnose Configuration Issues

On the Horizon Client, in the debug logs, look for log lines denoted with:

```
<vmware-view-usbd>
```

For example:

```
2013-11-25T14:04:21.625Z DEBUG (0BE4-0810) <vmware-usbd>  
[vmware-view-usbd] Filter Result: [UsbDeviceId: 40000001047faa01]  
Device 'Plantronics C620-M' is allowed
```

In the following sections, typically the logs are abbreviated, with the time, date, and other information at the start of the line removed. This should make it easier to digest the log information being displayed. For example, the above log line would be trimmed to show:

```
[vmware-view-usbd] Filter Result: [UsbDeviceId: 40000001047faa01] Device  
'Plantronics C620-M' is allowed
```

On the guest virtual machine, in the debug logs generated by the View Agent, look for log lines denoted with:

```
[ws_vhub]
```

The [Practical Worked Examples](#) section includes extracts from log files to highlight how the logs can be used to help work through configuration of USB devices.

Where to Configure USB Redirection

You configure USB redirection on either the Horizon Client or the View Agent, or both. To understand the interaction of configurations on the Horizon Client and View Agent, see the section on [Modifier on the View Agent Policy Setting](#).

When you configure the View Agent, you are configuring within the Windows guest operating system of the virtual desktop in the data center. This can be done either directly in the guest virtual machine, or via Active Directory Group Policy, which can push the configuration out to specific users or groups or perhaps even to specific View desktop pools. Using the Group Policy allows for central configuration, rather than configuration on every endpoint.

When you configure the Horizon Client, you are configuring within the operating system on the client machine, which can be Windows, Linux, or Mac OS.

Following are the operating-system-specific instructions for where and how you configure USB redirection.

Windows Configuration of USB Redirection

On Windows, it is easiest to use the Active Directory Group Policy Object (GPO) template for configuration of the Horizon Client or View Agent. Load this into the group policy editor by opening a command window and running `gpedit.msc` or a similar tool. The View Agent and downloadable Horizon Client settings can then be found as follows:

Local Computer Policy > Computer Configuration > Administrative Templates

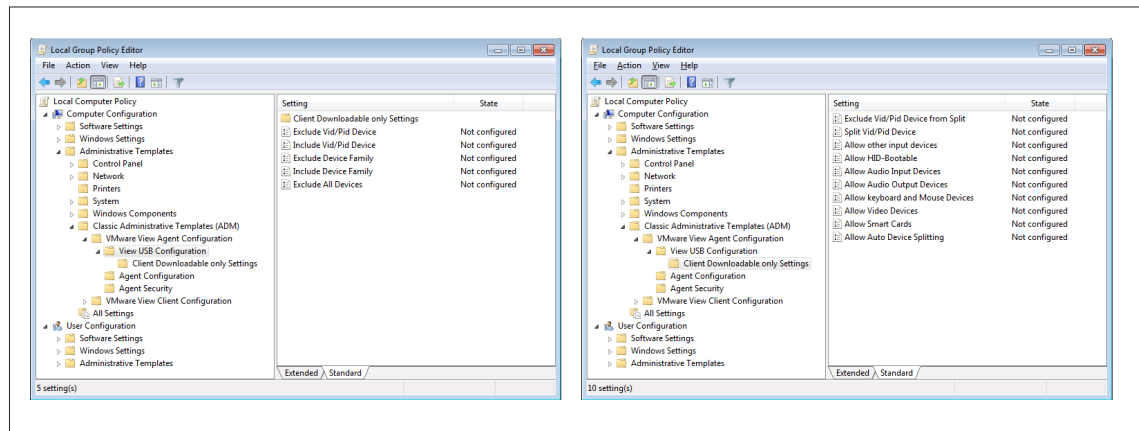


Figure 9: Screenshots of Group Policy Configuration

Note: **Client Downloadable only** settings are downloaded to the client at connection to the View desktop. See [View Agent and Horizon Client Configuration](#) for more information on downloadable client settings.

Linux Configuration of USB Redirection

You can set the USB properties in any one of several configuration files on Linux.

FILE	DESCRIPTION
<code>/etc/vmware/config</code>	The <code>vmware-view-usbd</code> service first examines this file. If USB configuration properties are set in this file, those properties are used.
<code>/usr/lib/vmware/config</code>	If the USB properties are not found in <code>/etc/vmware/config</code> , the <code>/usr/lib/vmware/config</code> file is checked.
<code>~/vmware/config</code>	If USB properties are not found in the other files, the <code>~/vmware/config</code> file is checked.

Table 6: Linux Configuration Files and Relative Search Order

The following syntax is used to set these properties in the Linux configuration files:

```
Viewusb.<property1> = "<value1>"
```

Note: With these properties, you can allow certain types of devices to be redirected or not. Filtering properties are also available so that you can exclude some types of devices and include others. For Linux client versions 1.7 and later, and for Windows clients, properties for splitting composite devices are also available.

Some values require the VID and PID for a USB device. To find the VID and PID, you can search on the Internet for the product name combined with VID and PID. Alternatively, you can look in the `/tmp/vmware-root/vmware-view-usbd-*.log` file after you plug in the USB device to the local system when Horizon Client is running. To set the location of this file, use the `view-usbd.log.fileName` property in the `/etc/vmware/config` file, for example:

```
view-usbd.log.fileName = "/tmp/usbd.log"
```

Mac OS X Syntax for Configuring USB Redirection

On a Mac OS X client, you configure USB capability by opening a shell (`/Applications/Utilities/Terminal.app`) and running a command as root using the following syntax.

- To list the rules:

```
defaults read <domain>
```

For example:

```
defaults read com.vmware.viewusb
```

- To remove a rule:

```
defaults delete <domain> <property>
```

For example:

```
defaults delete com.vmware.viewusb ExcludeVidPid
```

- To set or replace a filter rule:

```
defaults write <domain> <property> <value>
```

For example:

```
defaults write com.vmware.viewusb ExcludeVidPid vid-1234_pid-5678
```


- To set or replace a splitting rule for a composite device:

```
defaults write <domain> <property> <value>
```

Examples:

- `defaults write com.vmware.viewusb AllowAutoDeviceSplitting true`
- `defaults write com.vmware.viewusb SplitExcludeVidPid vid-03f0_pid-2a12`
- `defaults write com.vmware.viewusb SplitVidPid "'vid-0911_pid-149a(exintf:03)'"`

Note: The double and single quotes are required on Mac OS for a configuration with parentheses.

- `defaults write com.vmware.viewusb IncludeVidPid vid-0911_pid-149a`

Disabling USB Redirection

Some highly security-sensitive applications require that USB redirection be disabled to virtual desktops. This can be achieved in one of several ways:

- View pool policy can be used to disable USB redirection for a specific pool. This can be configured from the View Administrator UI:

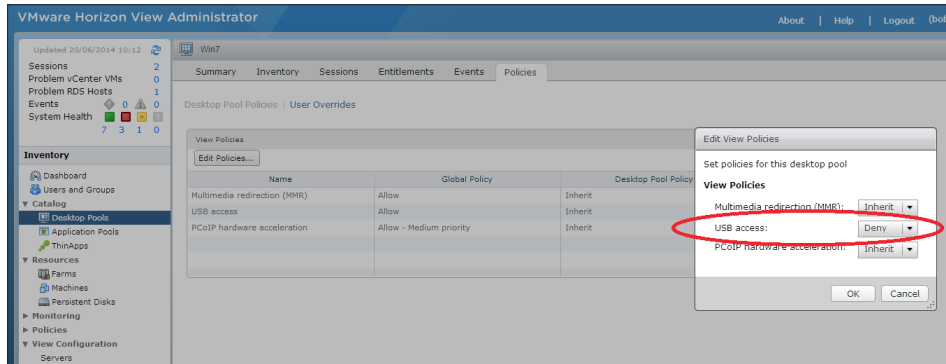


Figure 10: Disabling USB Access from the View Administrator

- User overrides can also be applied to enable or disable USB redirection on a per user basis in a specific pool. This is also done via the same View Administrator UI, with the User Overrides choice.
- The **ExcludeAllDevices** configuration option can be applied on the View Agent or the Horizon Client to prevent any devices from being forwarded.

Note: This can be used in conjunction with an Allow filter rule to permit only a specific device to work and to block all others.

- During installation of the View Agent on the View desktop in the data center, the USB-redirection components of View are installed by default. In a check box, you have the choice of deselecting the installation of these components. Without these components installed, it is absolutely not possible to do USB redirection!

Practical Worked Examples

The following section expands on some of the concepts previously explained with some practical worked examples. It is not possible to cover all combinations in this document, but the aim is to show some real-world configuration examples from the very basic to the more complex. These ideas can be applied to your specific requirements.

The examples that follow include log entries from the Windows client or guest virtual machine. Similar log messages are visible on Linux and Mac clients, and the same theory can be applied. Where possible, the log files displayed are trimmed and edited so that only pertinent information is displayed. Note that this might mean that some intermediate log lines are omitted for brevity.

Simple Filtering Examples

Following are some simple examples of filtering out some USB devices with an Exclude property, sometimes followed by an Include property for exceptions.

Blocking a Single Device from Appearing

You may want a device never to appear in the device menu on the View desktop. A practical reason for this might be that you are using an external USB-to-Ethernet adapter, and it is over this exact network connection that users need to connect to their View desktops. If you forward this USB connection into your View desktops, then the users cannot connect to their virtual desktops!

Assuming that the VID-PID for the device you want to block is `vid-0341_pid-1a11`, set the following rule to exclude the USB-to-Ethernet adapter device from redirection to the View desktop:

```
ExcludeVidPid      vid-0341_pid-1a11
```

Blocking All Storage Devices from Appearing in a Desktop Pool (for All Users)

The following example blocks all USB storage devices from being available to the View desktop for an entire desktop pool, for all users. If you want to apply a configuration for all users of a desktop pool, then you must apply the setting on the View Agent with a suitable GPO to apply to all users or for the specific pool globally:

```
ExcludeDeviceFamily  o:storage
```

Blocking All Devices from Appearing in the Horizon Client Menu

To ensure that no devices can be forwarded to the View desktop, use the Boolean value `ExcludeAllDevices`:

```
ExcludeAllDevices   Enabled
```

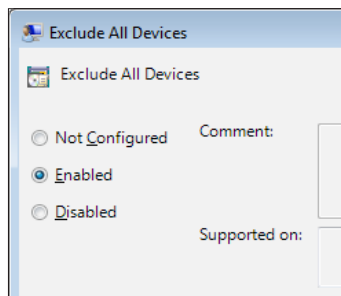


Figure 11: GPO Configuration to Exclude All Devices

Blocking All Devices Except One

If you want to block all USB devices except for one, you need to combine multiple rules. In this example, you want to block all devices except for the “iWidget” device with the VID-PID of `vid-0123_pid-abcd`:

```
ExcludeAllDevices      Enabled
IncludeVidPid          vid-0123_pid-abcd
```

Blocking All Devices Except for Mass Storage Devices

You may want to block all devices except for one family of devices, such as storage devices.

In this case, the only device the corporation wants to be used via USB is a storage device. To block all devices, but enable storage device families:

```
ExcludeAllDevices      Enabled
IncludeFamily          storage
```

Blocking All Devices Made by One Vendor

As an organization, you have decided that any devices made by the vendor “Unreliable-Co-Limited” are undesirable, and often cause blue screens leading to your users’ losing data. You decide that you want to prevent these devices from being used in your VDI environment.

First, identify the vendor ID of Unreliable-Co-Limited. In this example, pretend this is `vid_FA11`. To exclude all devices from this vendor, use a wildcard for the PID.

It is best to set this configuration on the View Agent, so that it is applied across all clients for all users:

```
ExcludeVidPid          o:vid-FA11_Pid-*
```

Chaining Multiple IncludeVidPid Rules to Include Two Devices, but Exclude All Others

If you have several devices you want to allow, but need to block everything else, simply separate each of the Include rules with a semicolon:

```
ExcludeAllDevices      Enabled
IncludeVidPid          vid-0123_pid-abcd; vid-1abc_pid-0001
```

Using Real-Time Audio-Video and Not Forwarding USB Audio and Video Devices

Real-Time Audio-Video performs encoding of the audio and video streams on the client, before transmitting the encoded stream to the guest virtual machine. The result is that the encoded stream uses much less bandwidth and handles high-latency or unreliable networks much better than if the devices were forwarded by USB. However, the devices are still often available for use as USB devices, and if the user forwards a device into the guest virtual machine, then the Real-Time Audio-Video features cannot operate.

To ensure that the devices are not forwarded, and are left local to the client, the following options are available:

- Block Audio and Video device families. This will block all cameras and headphones from being forwarded.
- Block a specific video or audio device, or both.

Option 1: Block Audio and Video families (assuming this is to be done for all users, configuration is set on the View Agent):

```
ExcludeDeviceFamily    o:video;audio
```

Option 2: To block a specific audio or video device, you need the VID and PID from the client logs. In this example, the video device has **Vid-041e_Pid-4087**, and the audio device has **Vid-047f_Pid-aa01**. For example, from the logs:

```
[vmware-view-usbd] Filter Result: [UsbDeviceId: 40000005041e4087]
On enumeration of device: 5/5. Name: Creative VF0680 Live! Cam Socialize
HD 1080 [vmware-view-usbd] DevFltr: Device id: Vid-041e_Pid-4087

[vmware-view-usbd] Filter Result: [UsbDeviceId: 40000007047faa01]
On enumeration of device: 1/5. Name: Plantronics C620-M
[vmware-view-usbd] DevFltr: Device id: Vid-047f_Pid-aa01
```

Set the following configuration to block these devices:

```
ExcludeVidPid      vid-041e_pid-4087;vid-047f_pid-aa01
```

Troubleshooting a Blocked Device

If you connect a Philips SpeechMike device, or other similar dictaphone device, to View, you will often find that the device “does not appear” in the virtual desktop menu for redirection. This worked example shows you why this happens and how to fix it.

Plug in the device and click the Horizon Client menu to enumerate the devices if you do not see an immediate listing. If the Philips SpeechMike device is not available to the Horizon Client, look at the log.

```
[vmware-view-usbd] Filter Result: [UsbDeviceId: 400000060911149a]
On enumeration of device: 1/5. Name: Philips USB Device

[vmware-view-usbd] DevFltr: Device id: Vid-0911_Pid-149a

[vmware-view-usbd] DevFltr: Interface count: 6

[vmware-view-usbd] DevFltr: Interface [0] - Family(s): audio
[vmware-view-usbd] DevFltr: Interface [1] - Family(s): audio,audio-in
[vmware-view-usbd] DevFltr: Interface [2] - Family(s): audio,audio-out
[vmware-view-usbd] DevFltr: Interface [3] - Family(s): mouse
[vmware-view-usbd] DevFltr: Interface [4] - Family(s): hid
[vmware-view-usbd] DevFltr: Interface [5] - Family(s): hid

[vmware-view-usbd] DevFltr: audio-out is blocked using AutoFilter
setting. The setting is ignored as there are multiple audio interfaces
which shouldnt be split [vmware-view-usbd] DevFltr: [Combined] Device
blocked by AutoFilters. Family(s): mouse

[vmware-view-usbd] Filter Result: [UsbDeviceId: 400000060911149a] Device
'Philips USB Device' is blocked
```

In the log extract you can see:

- The device being found is a **Philips USB Device** with **Vid-0911_Pid-149a**
- It has six interfaces (0 to 5) (as detailed above, including mouse, audio, and so on)
- The device is **blocked by AutoFilters**, due to **Family(s): mouse**
- The result of all filtering is that the device is blocked:

```
[vmware-view-usbd] Filter Result: [UsbDeviceId: 400000060911149a]
Device 'Philips USB Device' is blocked
```

The device is blocked because it is a device of family type **mouse**, and mice (and keyboards) are blocked by default.

To allow this device to operate in a View desktop, you can either

- Enable mice to be forwarded
- or
- Explicitly add an **IncludeVidPid** rule for this device

Because you are likely to have a physical mouse that you want to still be able to use in the client, the second option here is preferred. So, add the following configuration:

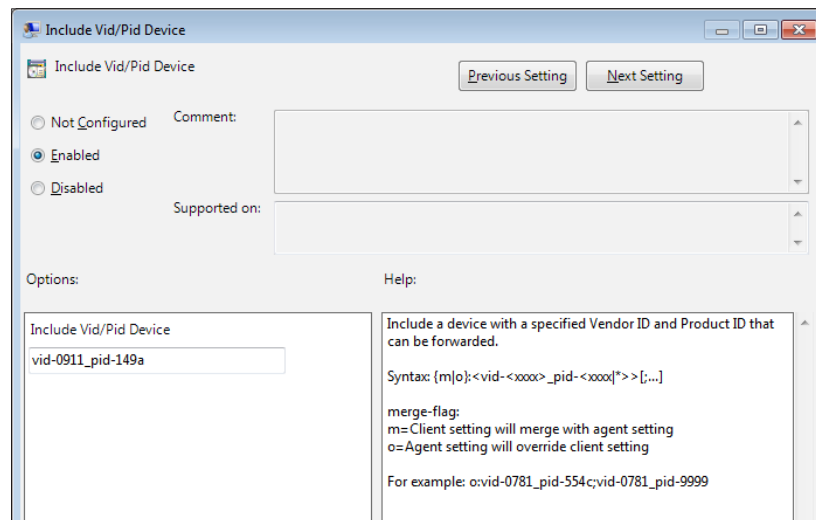


Figure 12: Example of Configuring an IncludeVidPid Rule to Include a Specific Device

```
IncludeVidPid    vid-0911_pid-149a
```

Then, restart the client.

Now, in the logs you can see the following:

```
[vmware-view-usbd] Filter Result: [UsbDeviceId: 400000060911149a]
On enumeration of device: 1/5. Name: Philips USB Device

[vmware-view-usbd] DevFltr: Device id: Vid-0911_Pid-149a
[vmware-view-usbd] DevFltr: Interface count: 6
[vmware-view-usbd] DevFltr: Interface [0] - Family(s): audio
[vmware-view-usbd] DevFltr: Interface [1] - Family(s): audio,audio-in
[vmware-view-usbd] DevFltr: Interface [2] - Family(s): audio,audio-out
[vmware-view-usbd] DevFltr: Interface [3] - Family(s): mouse
[vmware-view-usbd] DevFltr: Interface [4] - Family(s): hid
[vmware-view-usbd] DevFltr: Interface [5] - Family(s): hid
[vmware-view-usbd] DevFltr: [Combined] Device allowed by
'IncludeVidPid'. Id: vid-0911_pid-149a

[vmware-view-usbd] Filter Result: [UsbDeviceId: 400000060911149a] Device
'Philips USB Device' is allowed
```

- In the previous extract, you can clearly see that the `IncludeVidPid` rule has been applied:

```
Device allowed by 'IncludeVidPid'. Id: vid-0911_pid-149a
```

- And the device is now allowed:

```
Filter Result: [UsbDeviceId: 400000060911149a] Device 'Philips  
USB Device' is allowed
```

Following from the previous example, at this stage the device will work if you forward it into the View desktop; however, the mouse on the device will work only within that View desktop. It would not be possible to use the mouse in the client device, too.

Configuring Splitting

In this section are examples of how to split the functions of USB devices.

Configuring Splitting, Example 1

Following from the previous example, to improve the user experience, it is desirable to configure splitting such that the mouse device is left local to the client.

Looking again at the log extract, you can see that the device has six device interfaces, and interface [3] is of type `mouse`:

```
[vmware-view-usbd] DevFltr: Interface count: 6  
[vmware-view-usbd] DevFltr: Interface [0] - Family(s): audio  
[vmware-view-usbd] DevFltr: Interface [1] - Family(s): audio,audio-in  
[vmware-view-usbd] DevFltr: Interface [2] - Family(s): audio,audio-out  
[vmware-view-usbd] DevFltr: Interface [3] - Family(s): mouse **** ←  
[vmware-view-usbd] DevFltr: Interface [4] - Family(s): hid  
[vmware-view-usbd] DevFltr: Interface [5] - Family(s): hid
```

In this case, to leave this interface local to the client, you can configure the device with the following two lines, in this order:

```
IncludeVidPid      vid-0911_pid-149a      (this is as before, and is still needed)  
Split VidPid      vid-0911_pid-149a(exintf:03)
```

Now, if you re-launch the client and look in the logs you see:

```
[vmware-view-usbd] DevFltr: [Combined] Blocked by SplitSetting:  
SplitVidPid(exintf) setting. Interface num: 3  
...  
[vmware-view-usbd] Filter Result: Device can be partially forwarded.  
Allowed interface(s): 0,1,2,4,5
```

The Philips USB device should now work optimally in a View desktop, with the mouse interface excluded from redirection!

Configuring Splitting, Example 2

Connecting a composite-device keyboard, which contained a keyboard, extra control buttons, such as volume controls, and a fingerprint scanner, to a Horizon Client resulted in the device not being available in the View desktop. Analyzing the logs showed the following:

```
[vmware-view-usbd] Filter Result: [UsbDeviceId: 4000000705ba0008] On
enumeration of device: 1/5. Name: DigitalPersona U.are.U® 4000B
Fingerprint Keyboard

[vmware-view-usbd] DevFltr: Device Filter got device:
[vmware-view-usbd] DevFltr: Device id: Vid-05ba_Pid-0008
[vmware-view-usbd] DevFltr: Interface count: 3
[vmware-view-usbd] DevFltr: Interface [0] - Family(s): keyboard
[vmware-view-usbd] DevFltr: Interface [1] - Family(s): hid
[vmware-view-usbd] DevFltr: Interface [2] - Family(s): vendor
[vmware-view-usbd] DevFltr: [Combined:Phase] Starting 1(a)
[vmware-view-usbd] DevFltr: [Combined:Phase] Finished 1(a)
[vmware-view-usbd] DevFltr: [Combined:Phase] AutoDeviceSplitting
blocked.Skipping 1(b)
[vmware-view-usbd] DevFltr: [Combined:Phase] Starting 2
[vmware-view-usbd] DevFltr: [Combined] Device blocked by AutoFilters.
Family(s): keyboard
[vmware-view-usbd] DevFltr: [Combined:Phase] Finished 2
[vmware-view-usbd] DevFltr: [Remote:Phase] Starting 3
[vmware-view-usbd] DevFltr: [Remote:Phase] Finished 3

[vmware-view-usbd] Filter Result: [UsbDeviceId: 4000000705ba0008] Device
'DigitalPersona U.are.U® 4000B Fingerprint Keyboard' is blocked
```

Similar to the previous examples, this device is blocked due to its being a keyboard device:

```
Device blocked by AutoFilters. Family(s): keyboard
```

However, because it is a keyboard device, it is necessary for the keyboard to remain local to the client so that it can be used on the client device as well as in the View desktop. For example, the user needs to be able to key in the password to log in to the client!

However, the other interfaces, specifically in this case the HID and the **vendor** interface (which happens to be a fingerprint scanner), need to be available in the View desktop. In this case, splitting again must be configured.

To leave the keyboard interface [0] local to the client, and to allow the entire device to be visible in the View desktop, configure as follows:

```
IncludeVidPid      vid-05ba_pid-0008
Split VidPid       vid-05ba_pid-0008(exintf:00)
```


After the Horizon Client is restarted, the logs show that the device is now allowed to be forwarded:

```
[vmware-view-usbd] DevFltr: Device id: Vid-05ba_Pid-0008
[vmware-view-usbd] DevFltr: Interface count: 3
[vmware-view-usbd] DevFltr: Interface [0] - Family(s): keyboard
[vmware-view-usbd] DevFltr: Interface [1] - Family(s): hid
[vmware-view-usbd] DevFltr: Interface [2] - Family(s): vendor
[vmware-view-usbd] DevFltr: [Combined] Blocked by SplitSetting: SplitVidPid(exintf) setting. Interface num: 0
[vmware-view-usbd] DevFltr: [Combined] Device allowed by 'IncludeVidPid'. Id: vid-05ba_pid-0008
[vmware-view-usbd] Filter Result: [UsbDeviceId: 4000000705ba0008] Device 'DigitalPersona U.are.U® 4000B Fingerprint Keyboard' can be partially forwarded. Allowed interface(s): 1,2
```

Configuring Splitting: Dictaphone Device-Splitting Examples

Nuance has a software solution that can be used with View that enables SpeechMike-type dictaphone devices in some scenarios to be better used with the Nuance software than by using USB redirection. Specifically, Nuance has a plug-in that encodes the audio on the Horizon Client, and transmits that via a virtual channel directly into the dictation software in the guest virtual machine. This avoids any audio distortion to transported USB data caused by dropped or corrupted network packets and reduces the number of audio encode and decode operations resulting in higher audio quality.

However, for this solution to work, the HID interfaces of the dictaphone devices which control the capabilities such as record and play still need to be remoted to the guest virtual machine. The following is a list of the specific devices, along with the necessary View Agent configurations that permit the devices to be used and splits them such that *all* interfaces are left local to the client, except for the HID:

DEVICE	INCLUDE AND SPLIT CONFIGURATION
Dictaphone PowerMic II and III	Include VidPid Device o:vid-0554_pid-1001 Split VidPid Device o:vid-0554_pid-1001(exintf:00;exintf:01;exintf:02)
Philips SpeechMike II	Include VidPid Device o:vid-0911_pid-149a Split VidPid Device o:vid-0911_pid-149a(exintf:00;exintf:01;exintf:02;exintf:03)
Philips SpeechMike II with barcode scanner	Include VidPid Device o:vid-0911_pid-14a4 Split VidPid Device o:vid-0911_pid-14a4(exintf:00;exintf:01;exintf:02;exintf:03)
Philips SpeechMike III	Include VidPid Device o:vid-0911_pid-0c1c Split VidPid Device o:vid-0911_pid-0c1c(exintf:00;exintf::01;exintf:02;exintf:03)

DEVICE	INCLUDE AND SPLIT CONFIGURATION
Philips SpeechMike Premium	Include VidPid Device <code>o:vid-0911_pid-0c1c</code> Split VidPid Device <code>o:vid-0911_pid-0c1c(exintf:00;exintf:01;exintf:02;exintf:03)</code>
Grundig Digta SonicMic II	Include VidPid Device <code>o:vid-15d8_pid-0025</code> Split VidPid Device <code>o:vid-15d8_pid-0025(exintf:00;exintf:01;exintf:02;exintf:04;exintf:05)</code>
Olympus DR 2300	Include VidPid Device <code>o:vid-07b4_pid-0256</code> Split VidPid Device <code>o:vid-07b4_pid-0256(exintf:01;exintf:02;exintf:03;exintf:04;exintf:05;exintf:06)</code>

Table 7: Dictaphone Device Configurations

Note: It is possible to configure more than one device in the configuration entry. For example, if you use several devices and need to support such a solution with several variants of devices, you must separate them by use of a semicolon.

As an example, to configure Philips SpeechMike II and III, enter the following:

```

Include VID-PID Device
o:vid-0911_pid-149a;vid-0911_pid-0c1c

Split VID-PID Device
o:vid-0911_pid-149a(exintf:00;exintf:01;exintf:02;exintf:03);vid-0911_
pid-0c1c(exintf:00;exintf:01;exintf:02;exintf:03)
    
```

Summary

This white paper provides an overview of the USB redirection capability within VMware Horizon with View, as well as details about how to configure the product. The discussion also includes troubleshooting with relevant log files when trying to understand why certain devices are being blocked. Finally, some practical examples of configuration have been provided to help act as templates for real-world configuration use-cases.

About the Author

Peter Brown, a Senior R&D Manager in End-User Computing at VMware and based in London, UK, wrote this paper. He recently corrected a misprint and updated USB 3 information for Horizon 6.0.1 with View.

Peter wishes to thank Pete Barber, Tarique Chowdhury, Hans Christenson, Phil Lee, Phil Sturdy, Mike Pryor, Yunxia Cheng, Tony Huynh, Caroline Arakelian, and Tom Vandenbosch for their help providing subject matter expertise, guidance, and comments.

