Parallels Remote Application Server

GPU Accelerated Application Publishing

Parallels International GmbH
Vordergasse 59
8200 Schaffhausen
Switzerland
Tel: + 41 52 672 20 30
www.parallels.com

Copyright © 1999-2019 Parallels International GmbH. All rights reserved.

This product is protected by United States and international copyright laws. The product's underlying technology, patents, and trademarks are listed at http://www.parallels.com/about/legal/.

Microsoft, Windows, Windows Server, Windows Vista are registered trademarks of Microsoft Corporation.
Apple, Mac, the Mac logo, OS X, macOS, iPad, iPhone, iPod touch are trademarks of Apple Inc., registered in the US and other countries.
Linux is a registered trademark of Linus Torvalds.
All other marks and names mentioned herein may be trademarks of their respective owners.

Contents

Introduction **4**

GPU Acceleration Technologies **5**

RemoteFX vGPU 5

How to Setup RemoteFX vGPU 6

Discrete Device Assignment (DDA) 13

How to Setup Discrete Device Assignment (DDA) 14

RemoteFX vGPU vs. DDA Capabilities 20

Publishing GPU Accelerated Applications with Parallels RAS **22**

Desktop-Based VDI with RemoteFX vGPU 22

Server-Based VDI with RemoteFX vGPU 32

RD Session Hosts with DDA 37

Conclusion **40**

References **41**

Index **42**

CHAPTER 1

Introduction

The number of requests where graphic intensive applications, such as Photoshop, AutoCAD, Solidworks or even heavier applications like Maya and CATIA, that are being made in the Applications and Desktop virtualization projects based on Parallels Remote Application Server (RAS) is on the increase. This comes in hand in hand with the significant Remote Desktop Services (RDS) improvements released in Windows Server 2016. As such, in this document we will be looking at various options available today with Windows Server 2016 to add GPU acceleration to published resources via Parallels RAS.

The two options that can enable GPU acceleration in Windows Server 2016 for virtual desktops are:

- **RemoteFX vGPU**, a graphics virtualization technology that provides a virtual GPU adapter for redirecting API calls from a guest virtual machine (VM) to the physical GPU on the host.
- **Discrete Device Assignment (DDA)**, a GPU passthrough able to pass the physical GPU from a Hyper-V host to the guest VM directly.

This guide talks about each GPU acceleration technology and explains how to use them when publishing resources in Parallels RAS.

CHAPTER 2

GPU Acceleration Technologies

In This Chapter

RemoteFX vGPU 5
Discrete Device Assignment (DDA) 13
RemoteFX vGPU vs. DDA Capabilities 20

RemoteFX vGPU

RemoteFX vGPU is a graphics virtualization technology that allows the processing power of a GPU to be split across various guest operating systems. This was introduced in Windows Server 2008R2 SP1 to provide GPU acceleration for VDI environments. Further improvements were included in Windows 2012 and even more with the release of Windows Server 2016, which include:

- Support for Windows Server 2016 as a guest. This means that Windows Server OS can be used to build VDI (session based VDI or Server based VDI as a Personal Session Desktop). Particularly useful for Service providers is the ability to build VDI solutions using Windows Server 2016 inside tenant VMs and license it through SPLA.
- Improvements in application compatibility and stability now includes support for DirectX 11.1, OpenGL 4.4 and OpenCL 1.1, all of which are required by modern graphics and 3D applications.
- RemoteFX GPU video RAM limit has been extended from 256 MB to 1024 MB. Dedicated Video Memory now can be set directly without playing with monitor number and resolution. Depending on the amount of system memory assigned to the VM, this can provide up to a total of 2 GB of video RAM (1 GB dedicated and 1 GB shared).

Detailed information on what's new in RDS 2016 is available at https://docs.microsoft.com/en-us/windows-server/remote/remote-desktop-services/rds-whats-new

Considering that multiple guest operating systems can use the same physical GPU via RemoteFX vGPUs, this is generally recommended if scaling is a requirement. This type of graphics virtualization is ideal to use where applications do not have as strict GPU requirements when compared to DDA. However, in the application and desktop virtualization it is critical to mention that RemoteFX vGPU and RDSH on Windows 2016 are not supported. This is because only one session can be hosted for each Windows Server 2016 running as a guest VM with RemoteFX vGPU. Instead, RemoteFX vGPU can be utilized through VM based VDIs supporting Windows 7 SP1, Windows 8.1 and Windows 10 or Windows Server 2012 R2 and Windows Server 2016 as guest VMs.

How to Setup RemoteFX vGPU

Setting up guest VMs to use RemoteFX vGPU is a straightforward procedure, which is described in this section.

For this example the below environment was used:

Physical Host

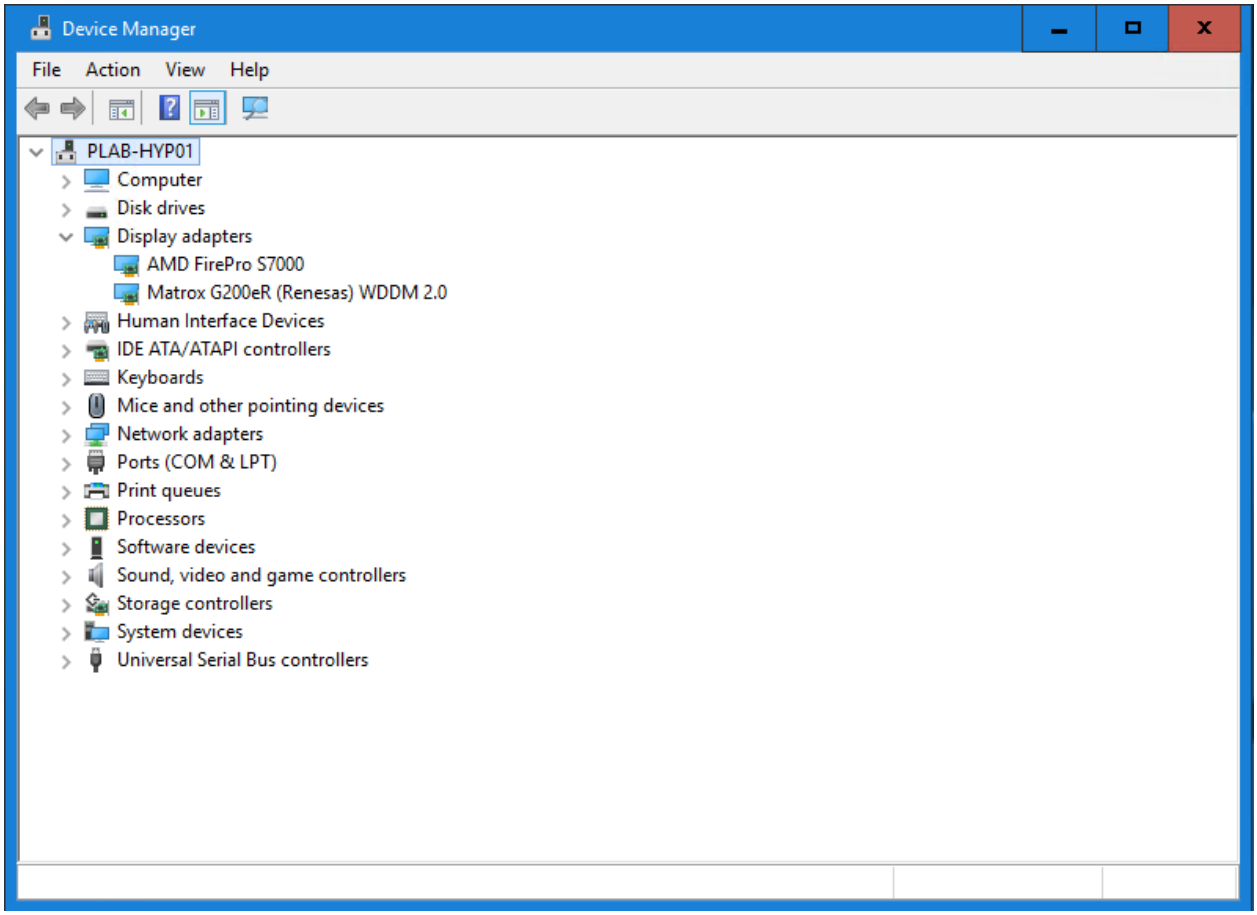
Hostname	PLAB-HYP1
Server System Model	Dell PowerEdge R730
Operating System	Windows Server 2016
Role	Hypervisor - Microsoft Hyper-V
Firmware version	2.21.21.21
GPU	AMD FirePro S7000

Guest VM

Hostname	PLAB-RAS3
Operating System	Windows Server 2016
Role	RDSH

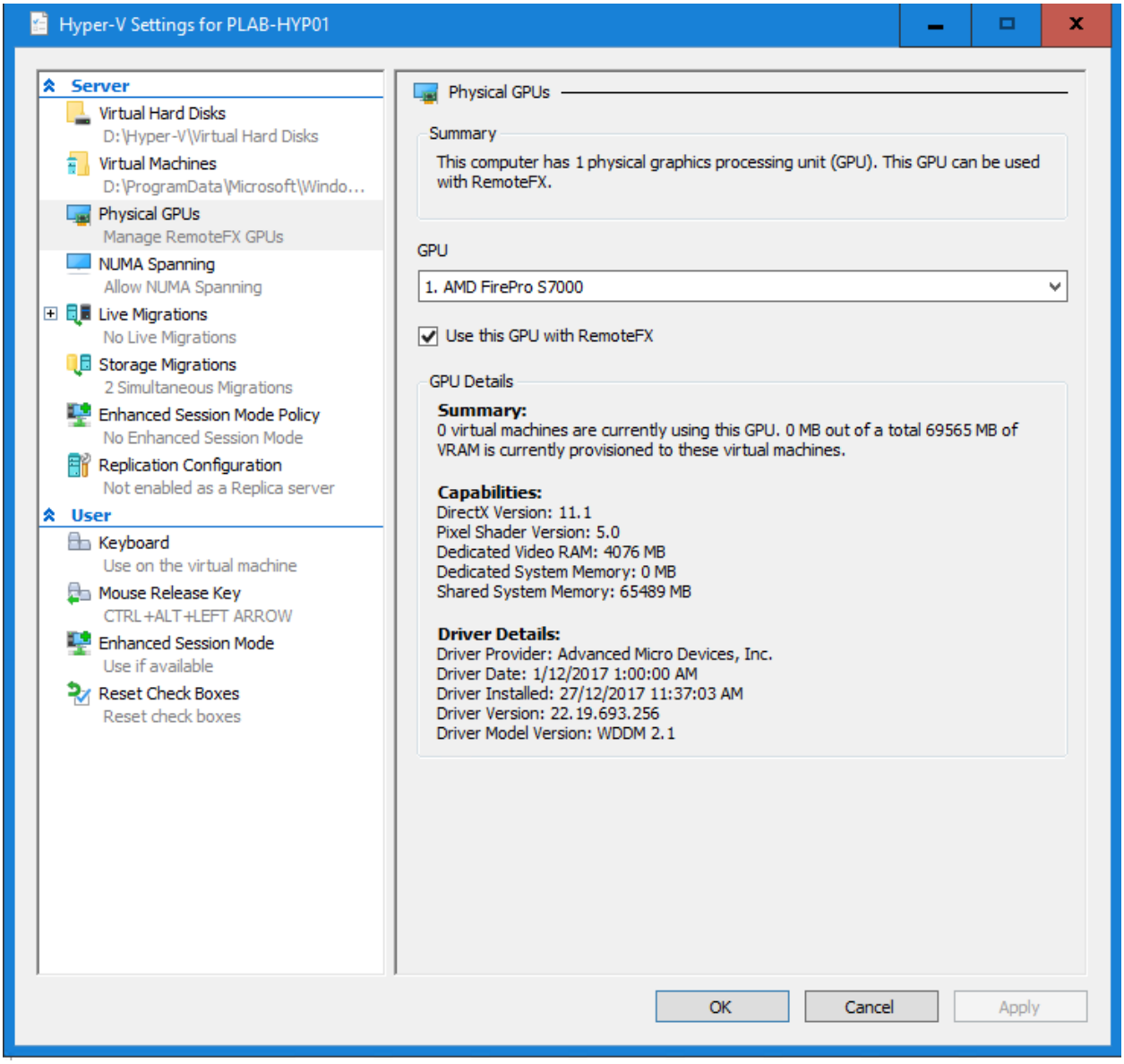
Setup procedure

1 It is important to make sure that the correct vendor drivers are installed on the Hyper-V host. This can be confirmed by looking at hosts' device manager, in this case on PLAB-HYP1.

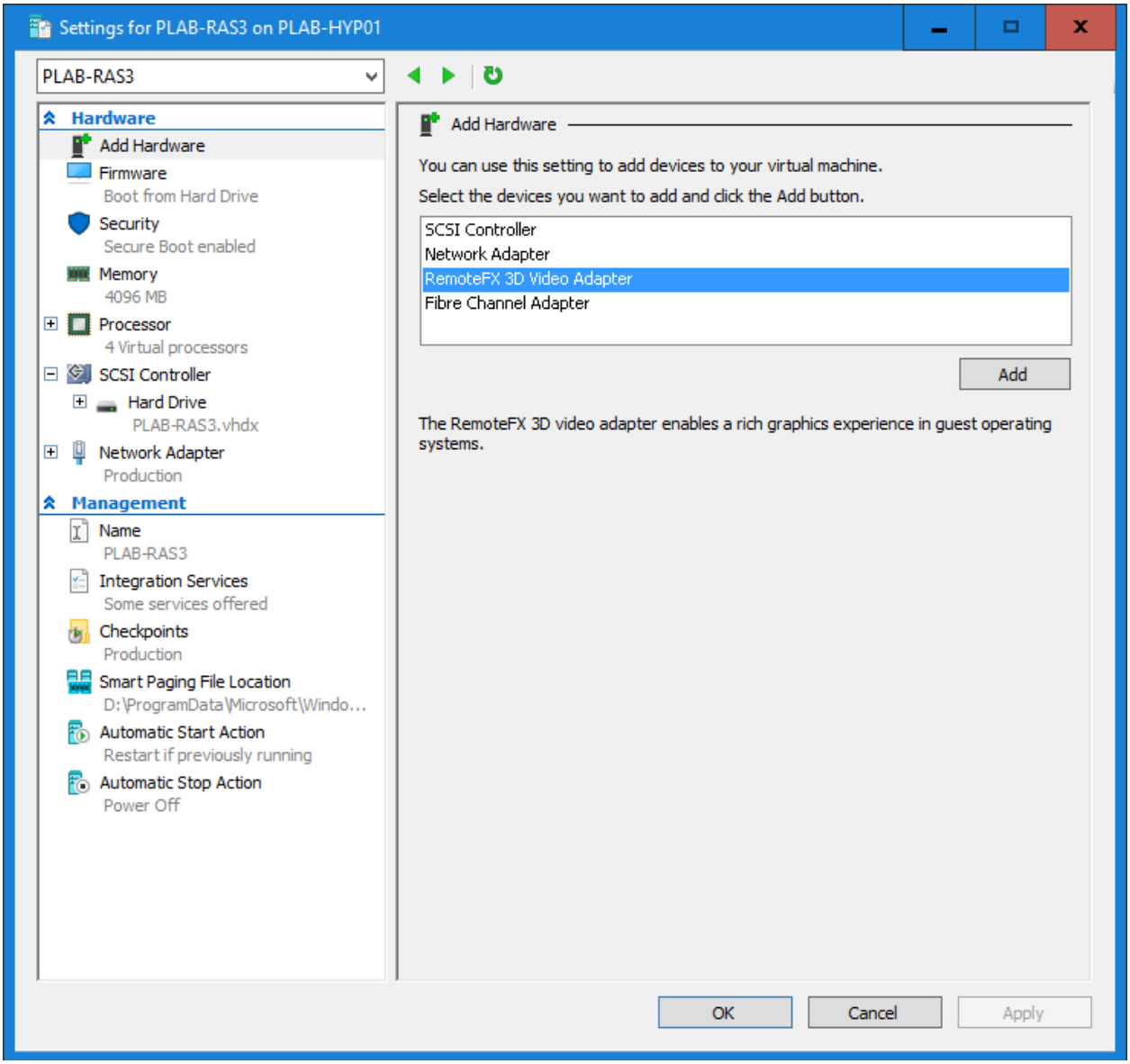


Note: For the list of supported Remote-FX compatible GPUs please see https://cloudblogs.microsoft.com/enterprisemobility/2013/11/05/gpu-requirements-for-remotefx-on-windows-server-2012-r2/

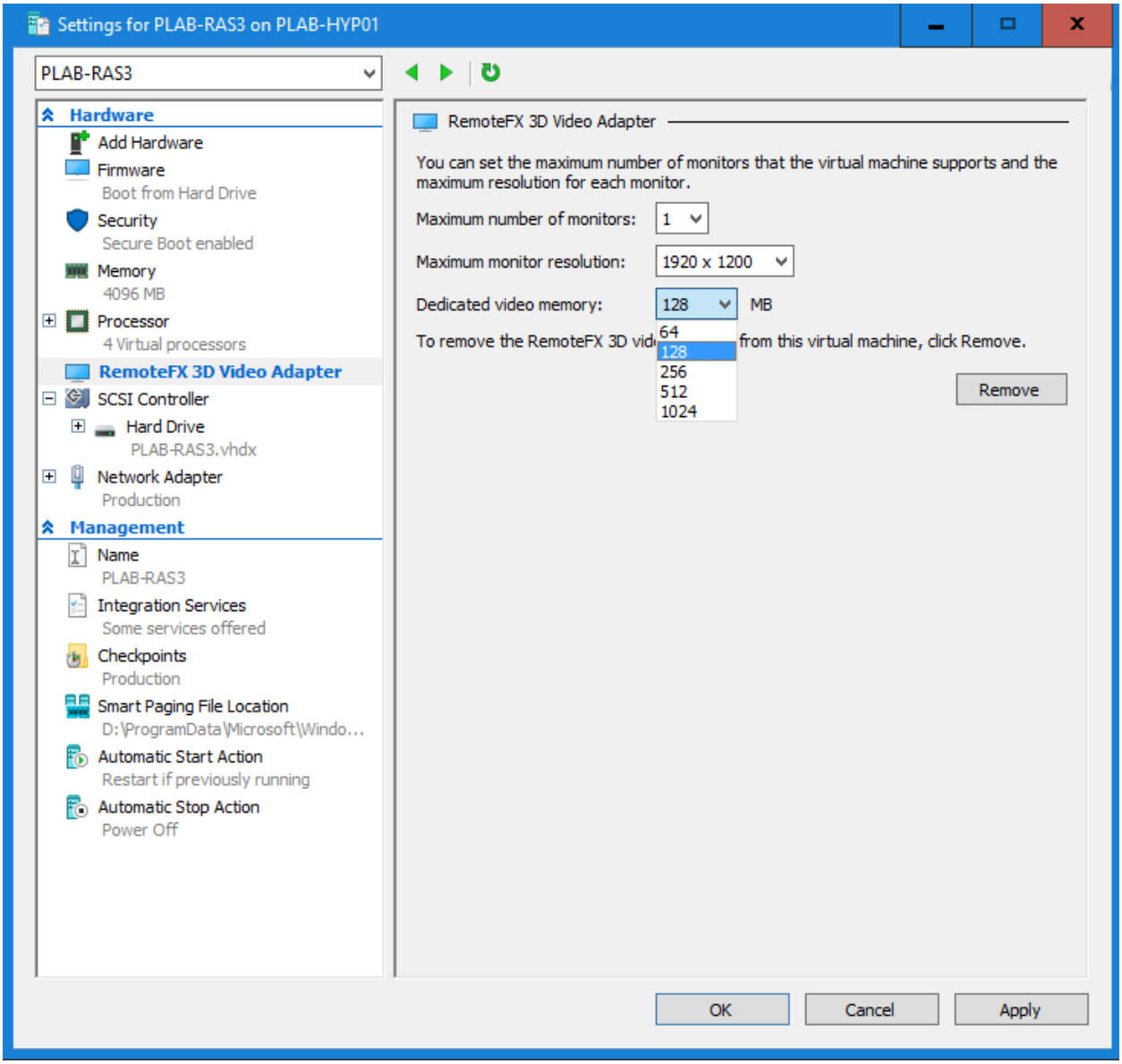
2 To configure RemoteFX vGPU, open the Hyper-V Manager on the host and click on the Hyper-V Settings under the Actions menu on the left-hand side. This will open the Hyper-V Settings windows from where we can set the required GPU to be used with RemoteFX.



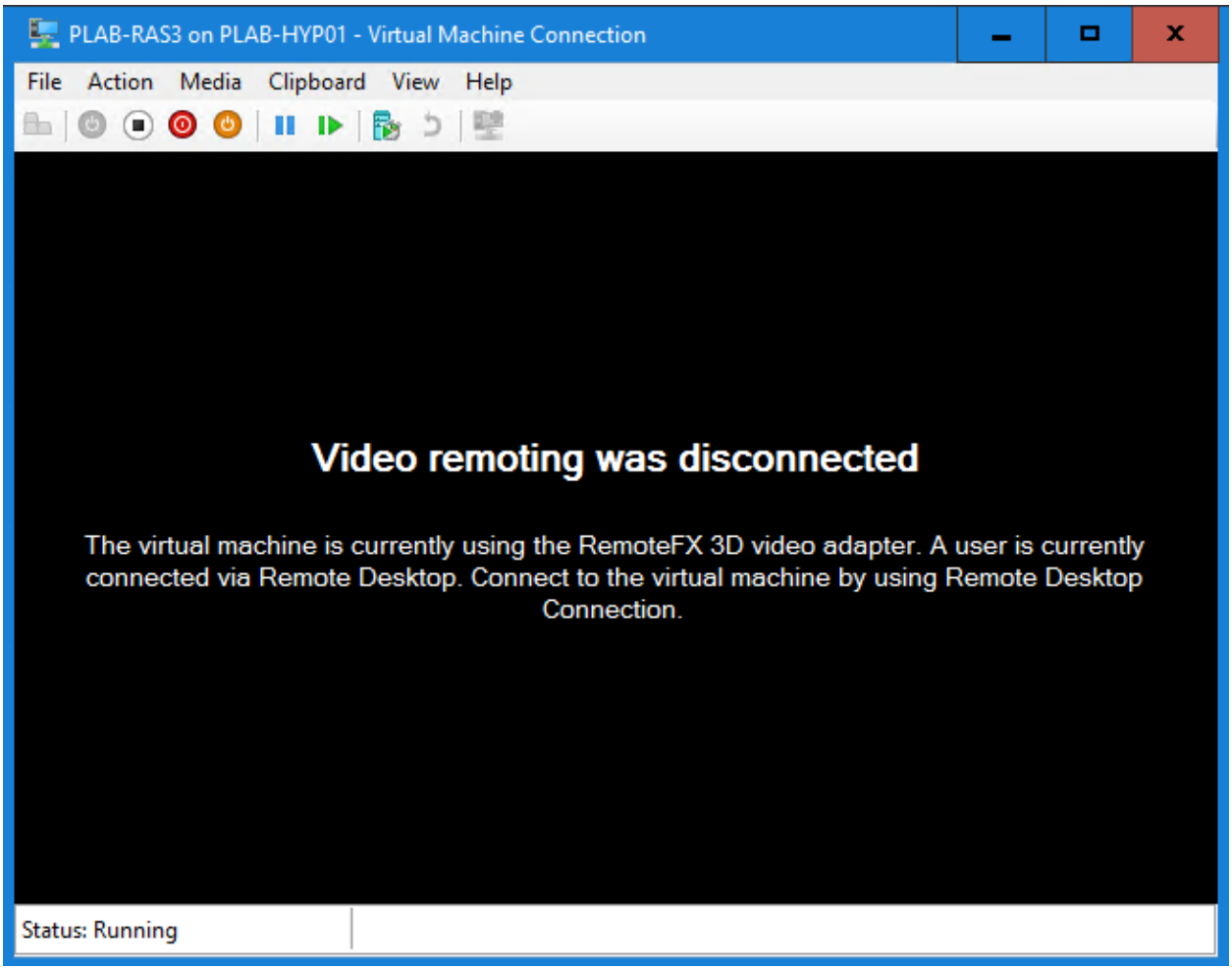
3 Right click and open settings of the intended guest VM to use RemoteFX vGPU and add a RemoteFX 3D Video Adapter.



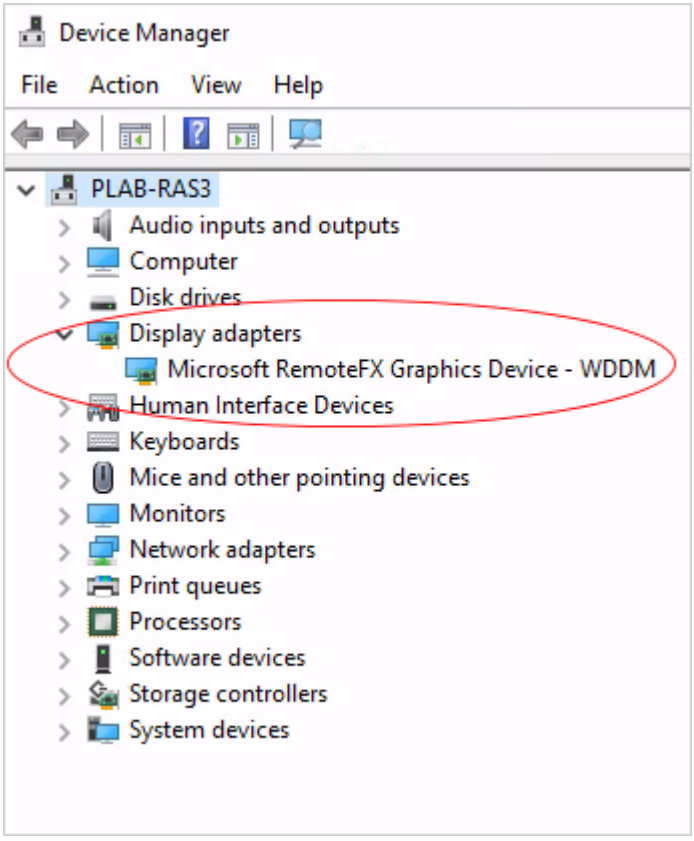
4 Set the maximum number of monitors that this guest VM will support, the resolution for each monitor, and the dedicated video memory.



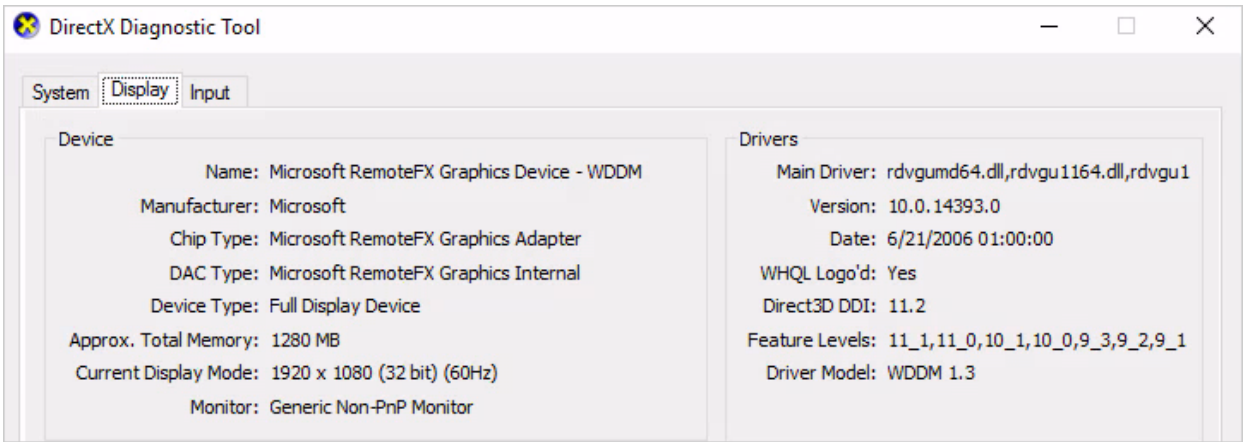
5 After applying the settings, power on and RDP to the VM to confirm that RemoteFX vGPU is working as expected. As a confirmation, the VM console shows that the VM is using the RemoteFX 3D Video Adapter.



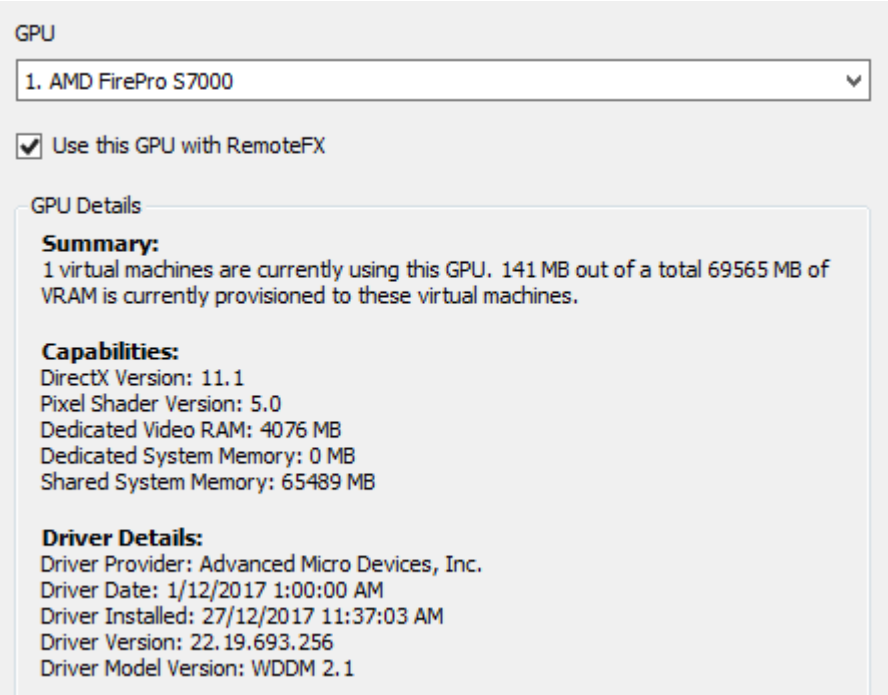
6 Opening the device manager of PLAB-RAS3, which is the guest VM with a RemoteFX 3D video adapter, shows us that Microsoft RemoteFX Graphics Device is being used.



Same can be observed if you run dxdiag from C:\Windows\System32\dxdiag.exe on the guest VM and checking the **Display** properties, as shown on the screenshot below.



By looking at the Hyper-V settings we can easily note how many virtual machines are using this GPU under the GPU details.



Discrete Device Assignment (DDA)

DDA is a hardware pass-through solution which was made available on Microsoft Hyper-V with Windows Server 2016 acting as the host. Unlike RemoteFX vGPU, DDA provides a guest VM with full access to the GPU by enabling PCI-Express devices to be passed directly to the guest VM using native drivers. This enables the full set of capabilities, such as DirectX 12, CUDA, and others, and provides generally better performance compared to RemoteFX vGPU. This, however, comes at a cost on scalability because with DDA, the GPU is directly and exclusively passed to a specific VM and cannot be used by other VMs.

When looking at the GPU usage on RDSH running Windows Server 2016 for application and/or desktop publishing, the DDA configuration is the supported way forward. Please note that although some testing has been done on Windows Server 2012 R2, Parallels RAS officially supports and recommends Windows Server 2016 for the DDA solution.

How to Setup Discrete Device Assignment (DDA)

Unlike setting up RemoteFX vGPU, there are various requirements and prerequisites that are needed to be completed to setup a DDA configuration. The necessary steps are described in this section.

For this example the below environment was used:

Physical Host

Hostname	PLAB-HYP1
Server System Model	Dell PowerEdge R730
Operating System	Windows Server 2016
Role	Hypervisor - Microsoft Hyper-V
Firmware version	2.21.21.21
GPU	AMD FirePro S7000

Guest VM

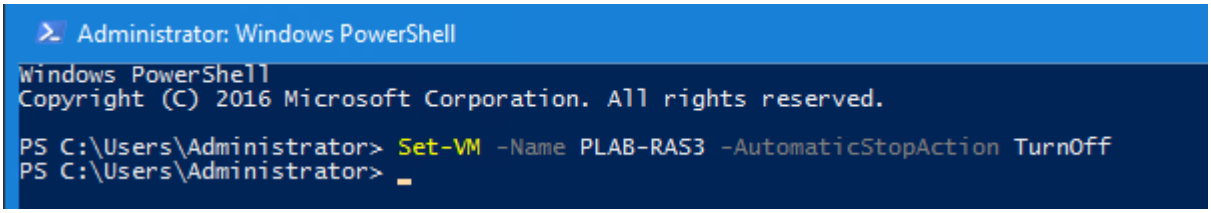
Hostname	PLAB-RAS3
Operating System	Windows Server 2016
Role	RDSH

Note: Make sure that the physical server, BIOS, firmware, and GPUs are DDA-compatible.

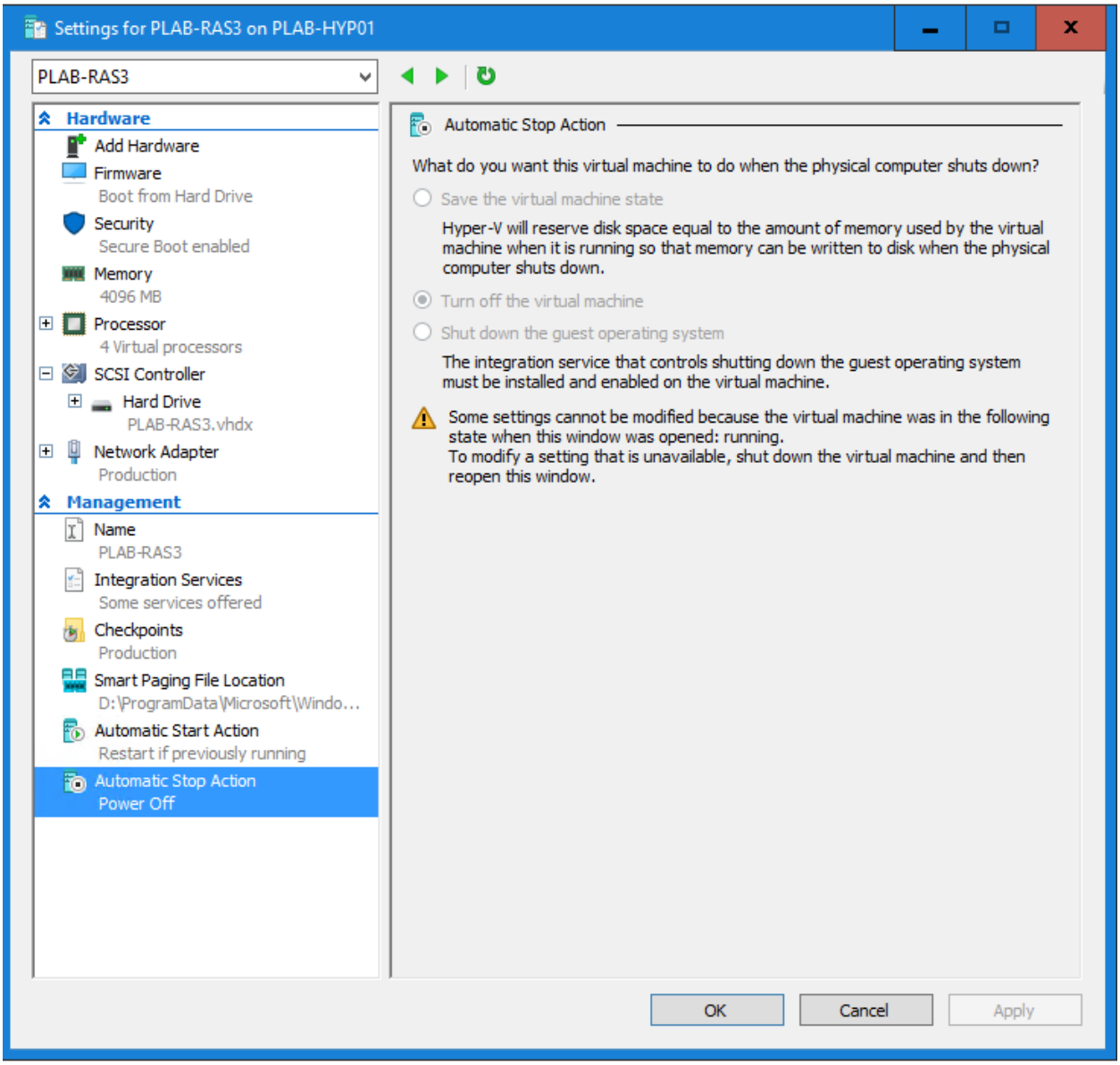
Setup procedure

1 First, we need to configure a VM for DDA (in our case, it's PLAB-RAS3). To do so, we need to configure Automatic Stop Action to TurnOff by opening Powershell as an administrator on the host, PLAB-HYP1.

```
Set-VM -Name VMName -AutomaticStopAction TurnOff
```



This changes the behavior of the VM when the physical computer shuts down.



For graphics devices, some hardware may perform better if the VM is configured in a certain way as shown below:

- Enable Write-Combining on the CPU:

  ```
  Set-VM -GuestControlledCacheTypes $true -VMName VMName
  ```

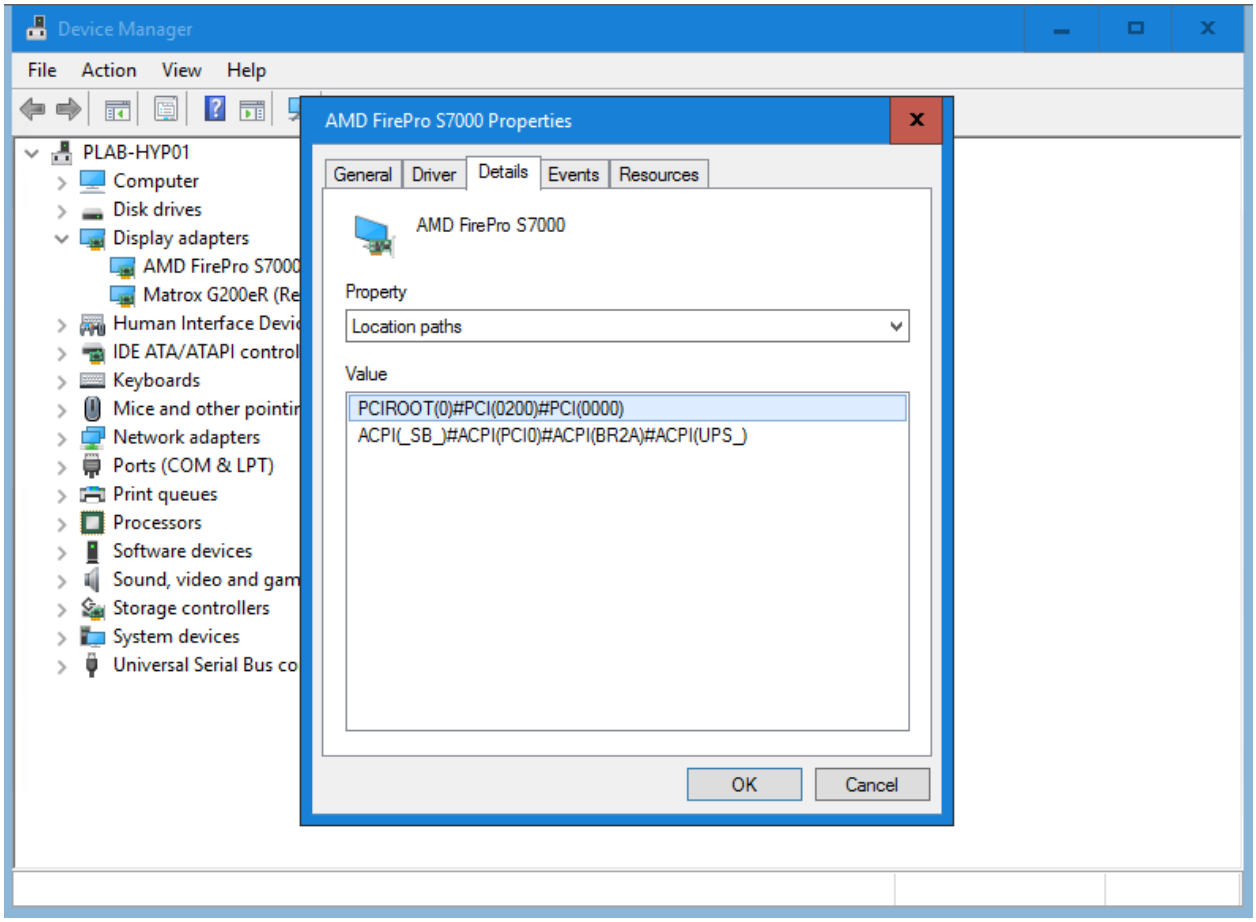
- Configure the 32 bit MMIO space:

  ```
  Set-VM -LowMemoryMappedIoSpace 3Gb -VMName VMName
  ```

- Configure greater than 32 bit MMIO space:

  ```
  Set-VM -HighMemoryMappedIoSpace 33280Mb -VMName VMName
  ```

2 We need to dismount the device from the host partition to be able to assign it to a guest VM. To do so, the PCI location path is required. The PCI location path can be acquired from the Device Manager or via Powershell on the host as shown below:



- Enumerate all PNP Devices on the system

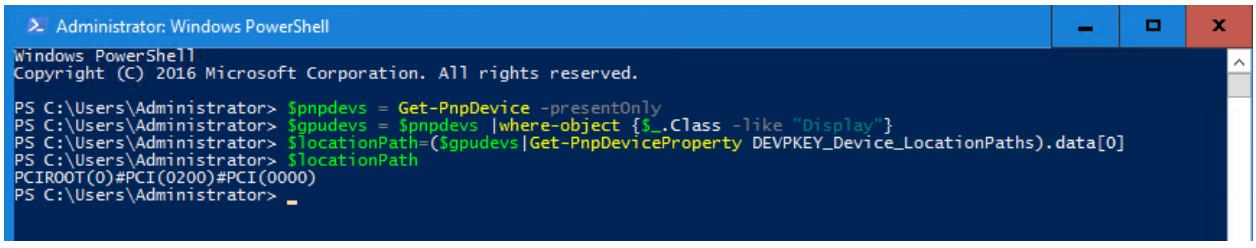
  ```
  $pnpdevs = Get-PnpDevice -presentOnly
  ```

- Select only those devices that are Display devices

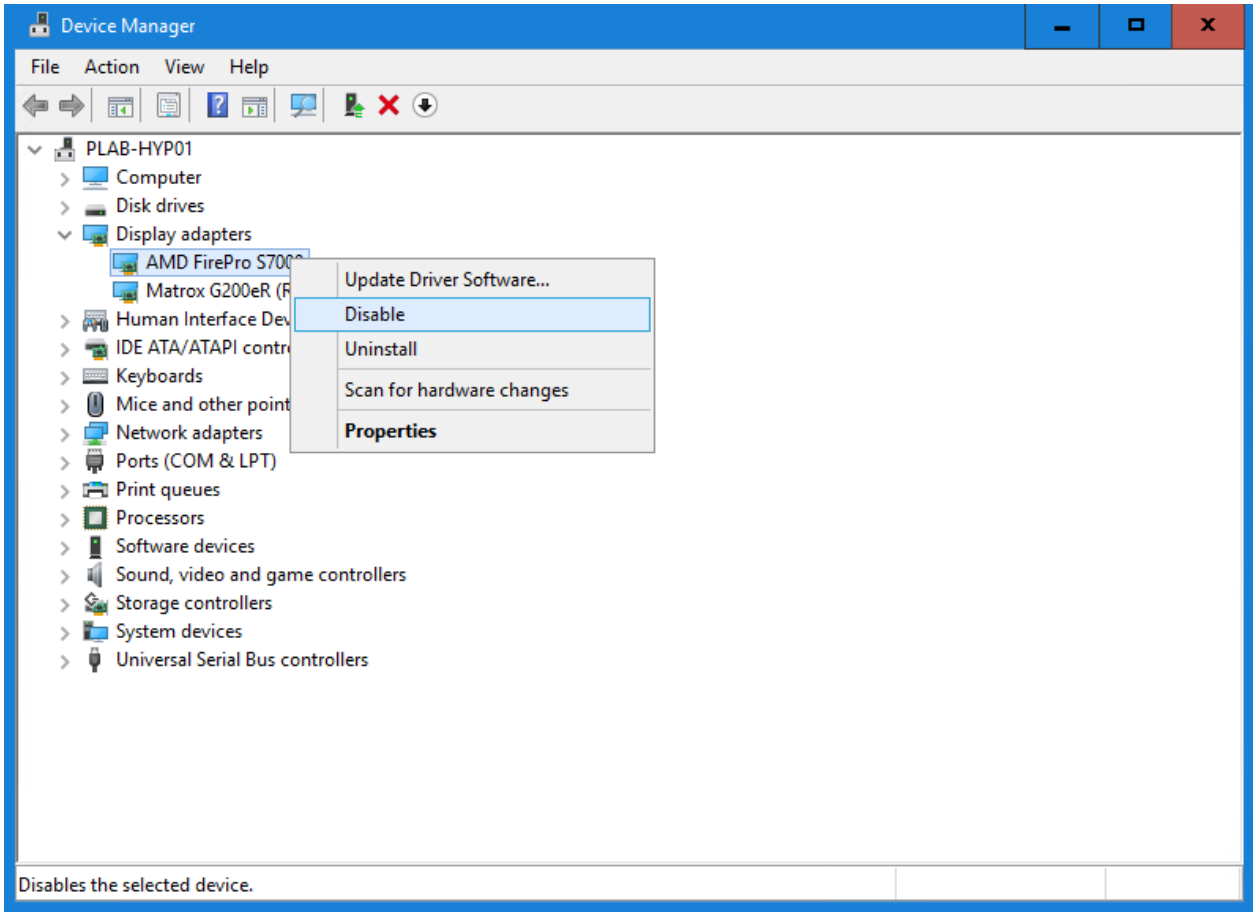
  ```
  $gpudevs = $pnpdevs |where-object {$_.Class -like "Display"}
  ```

- Select the location path of the first device that's available to be dismounted by the host.

  ```
  $locationPath=($gpudevs|Get-PnpDeviceProperty
  DEVPKEY_Device_LocationPaths).data[0]
  ```



3 We now need to disable the GPU graphics device from the host before we dismount it. Again, this can be done from Device Manager or via Powershell on the host as shown below:

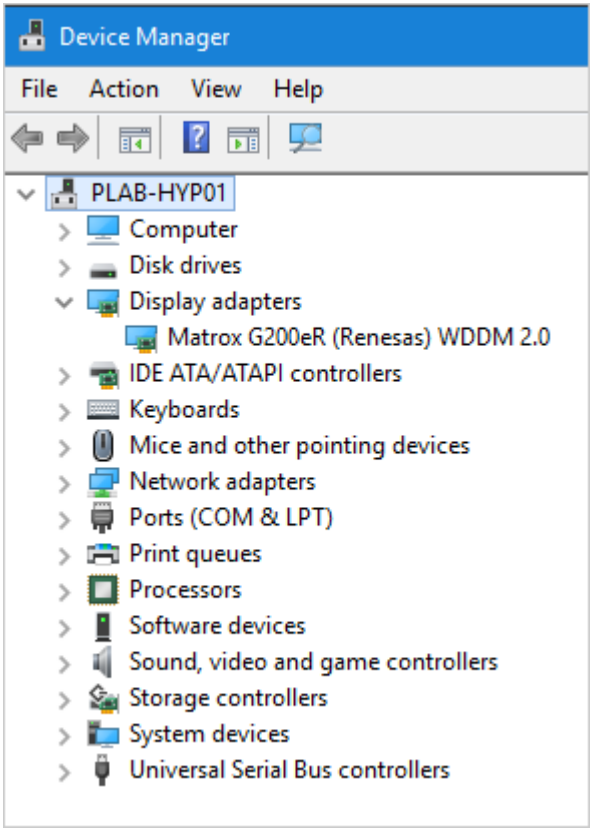


4 Next step is to dismount the video card from the host so that it will no longer be accessible on the parent partition as shown below:

```
PS C:\Users\Administrator> Dismount-VMHostAssignableDevice -force -LocationPath $locationPath
```

Note: Use `-force` if no mitigation driver was installed

5 After dismounting the GPU from the host, note that this is no longer listed under Display adapters from the device manager on the host.

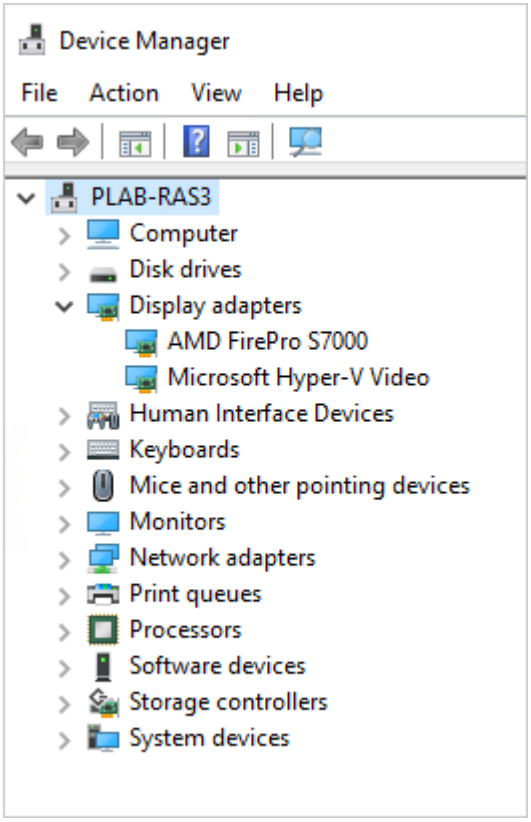


6 Now we can assign the GPU graphics adapter to the guest VM, in this case to PLAB-RAS3.

```
Add-VMAssignableDevice -LocationPath $locationPath -VMName VMName
```

```
PS C:\Users\Administrator> Add-VMAssignableDevice -LocationPath $locationPath -VMName PLAB-RAS3_
```

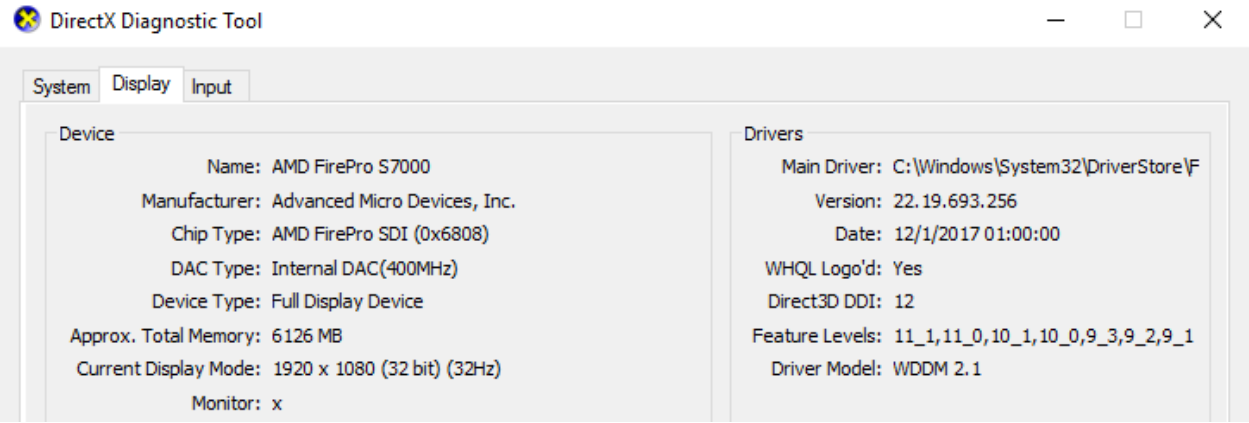
7 Now that GPU is assigned to PLAB-RAS3, we can install the GPU vendor driver on the guest VM and confirm successful installation of the right drivers from Device Manger.



8 Since we are using Windows Server 2016 with the intention of being used as an RDSH, we need to setup a group policy on the same guest VM to use the hardware graphics renderer instead of the Microsoft Basic Render Driver as the default adapter in RDS.

This can be carried out by enabling policy found under Computer Configuration / Administrator Template / Windows Components / Remote Desktop Services / Remote Desktop Session Host / Remote Session Environment to "Use the hardware default graphics adapter for all Remote Desktop Services sessions".

9 To confirm that the physical GPU is used by the guest VM, you can RDP to the guest VM and use C:\Windows\System32\dxdiag.exe on the guest VM and examine Display properties as shown below:



For additional information about the procedure please see Plan for Deploying Devices using Discrete Device Assignment: https://docs.microsoft.com/en-us/windows-server/virtualization/hyper-v/plan/plan-for-deploying-devices-using-discrete-device-assignment

RemoteFX vGPU vs. DDA Capabilities

The following table highlights different capabilities when comparing RemoteFX vGPU and DDA.

Feature	RemoteFX vGPU	Discrete Device Assignment
Device GPU assignment	Para-virtualized (many VMs to one or more GPUs)	1 or more GPU to 1 VM
Scale	Best scale / 1 GPU to many VMs	Low scale / 1 or more GPUs to 1 VM
App compatibility	DX 11.1, OpenGL 4.4, OpenCL 1.1	All GPU capabilities provided by vendor (DX 12, OpenGL, CUDA)
AVC444	Enabled by default (Windows 10 and Windows Server 2016)	Available through Group Policy (Windows 10 and Windows Server 2016)
GPU VRAM	Up to 1 GB dedicated VRAM	Up to VRAM supported by the GPU
Frame rate	Up to 30fps	Up to 60fps
GPU driver in guest	RemoteFX 3D adapter display driver (Microsoft)	GPU vendor driver (Nvidia, AMD, Intel)
Guest OS support	Windows Server 2012 R2 Windows Server 2016 Windows 7 SP1 Windows 8.1 Windows 10	Windows Server 2012 R2 Windows Server 2016 Windows 10 Linux
Hypervisor	Microsoft Hyper-V	Microsoft Hyper-V
Host OS availability	Windows Server 2012 R2 Windows Server 2016 Windows 10	Windows Server 2016

GPU hardware	Enterprise GPUs (such as Nvidia Quadro/GRID or AMD FirePro)	Enterprise GPUs (such as Nvidia Quadro/GRID or AMD FirePro)
Server hardware	No special requirements	Modern server, exposes IOMMU to OS (usually SR-IOV compliant hardware)

Table Reference: https://docs.microsoft.com/en-us/windows-server/remote/remote-desktop-services/rds-graphics-virtualization

CHAPTER 3

Publishing GPU Accelerated Applications with Parallels RAS

Parallels RAS can be used with both GPU acceleration options, RemoteFX vGPU or DDA, to have applications and/or desktops utilize GPU performance. When using RemoteFX vGPU, Parallels RAS can be used to publish desktop VDIs running on Windows 7 SP1, Windows 8.1 and Windows 10. With Parallels RAS v16.x and newer, server-based VDI can also be deployed with RAS VDI Guest Agent running on Windows Server 2012R2 and Windows Server 2016. Both VDI deployment procedures (Desktop or Server OS) are described in the sections that follow this one.

In This Chapter

Desktop-Based VDI with RemoteFX vGPU 22
Server-Based VDI with RemoteFX vGPU 32
RD Session Hosts with DDA..... 37

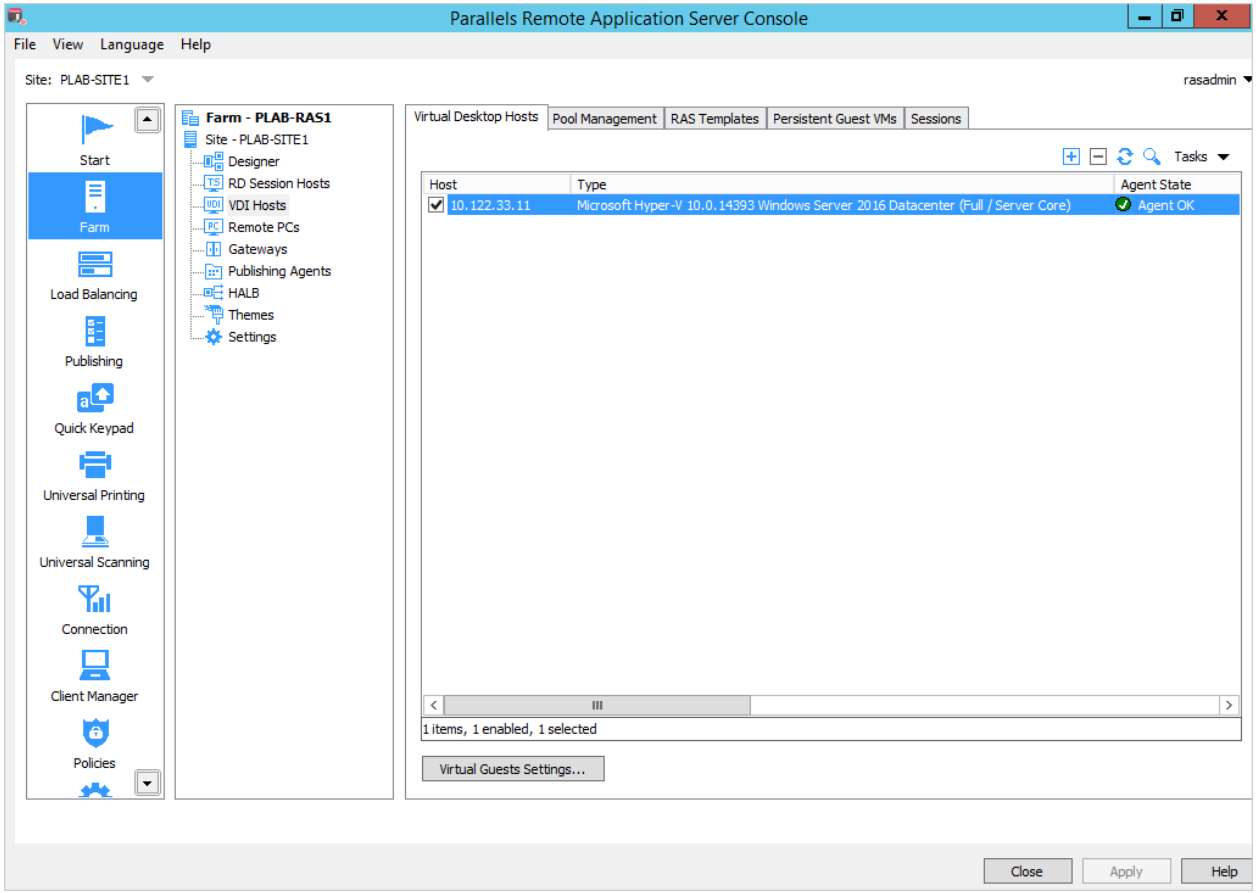
Desktop-Based VDI with RemoteFX vGPU

For the below example we will use a Windows 10 RAS Template with RemoteFX vGPU and linked clones to cater for a small test user base.

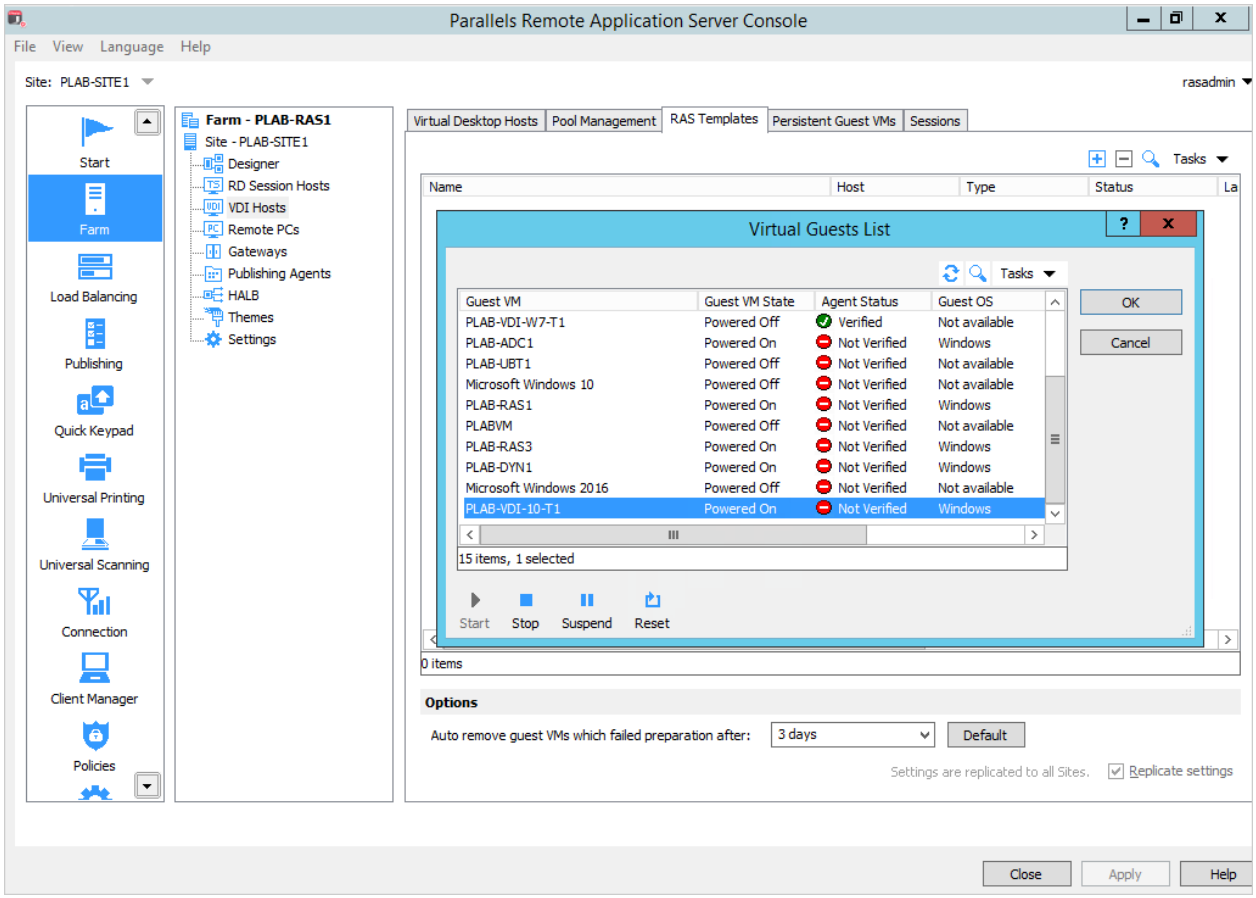
Note: For a complete guide on how to deploy Parallels RAS VDI, please refer to Parallels RAS Administrators Guide.

1 Add the RemoteFX 3D Video Adapter to a guest VM which will be used as a RAS template (as shown in the previous section). In our case we will be using a guest VM with hostname PLAB-VDI-10-T1.

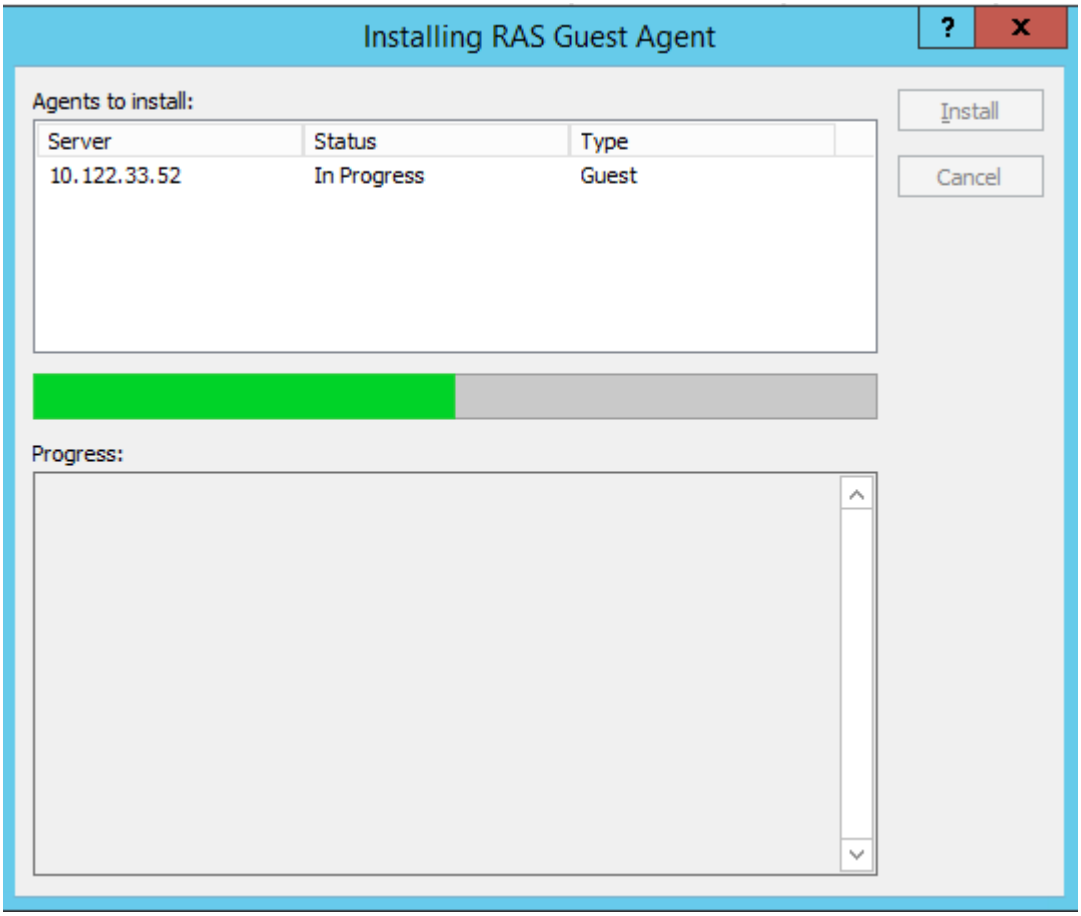
2 Open RAS Console, log in as Parallels RAS Administrator, go to **Farm** category and choose **VDI Hosts**. Add the hypervisor (in our case Microsoft Hyper V 2016) as a RAS VDI Host.



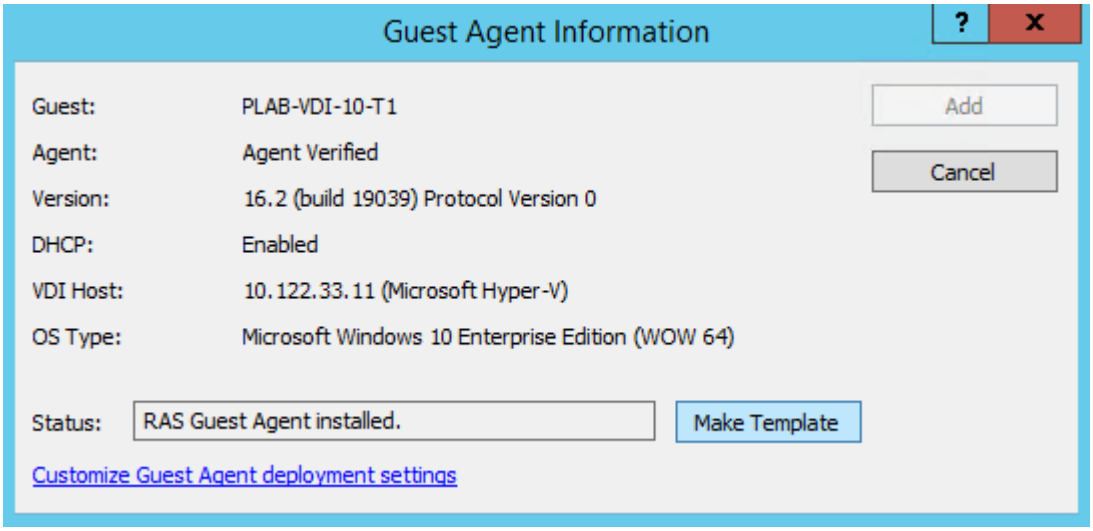
3 Go to **RAS Templates** and choose the guest VM to be used as a RAS Template (PLAB-VDI-10-T1 in this example). The VM must have the RemoteFX 3D Video Adapter already assigned to it (as described in step 1 above.



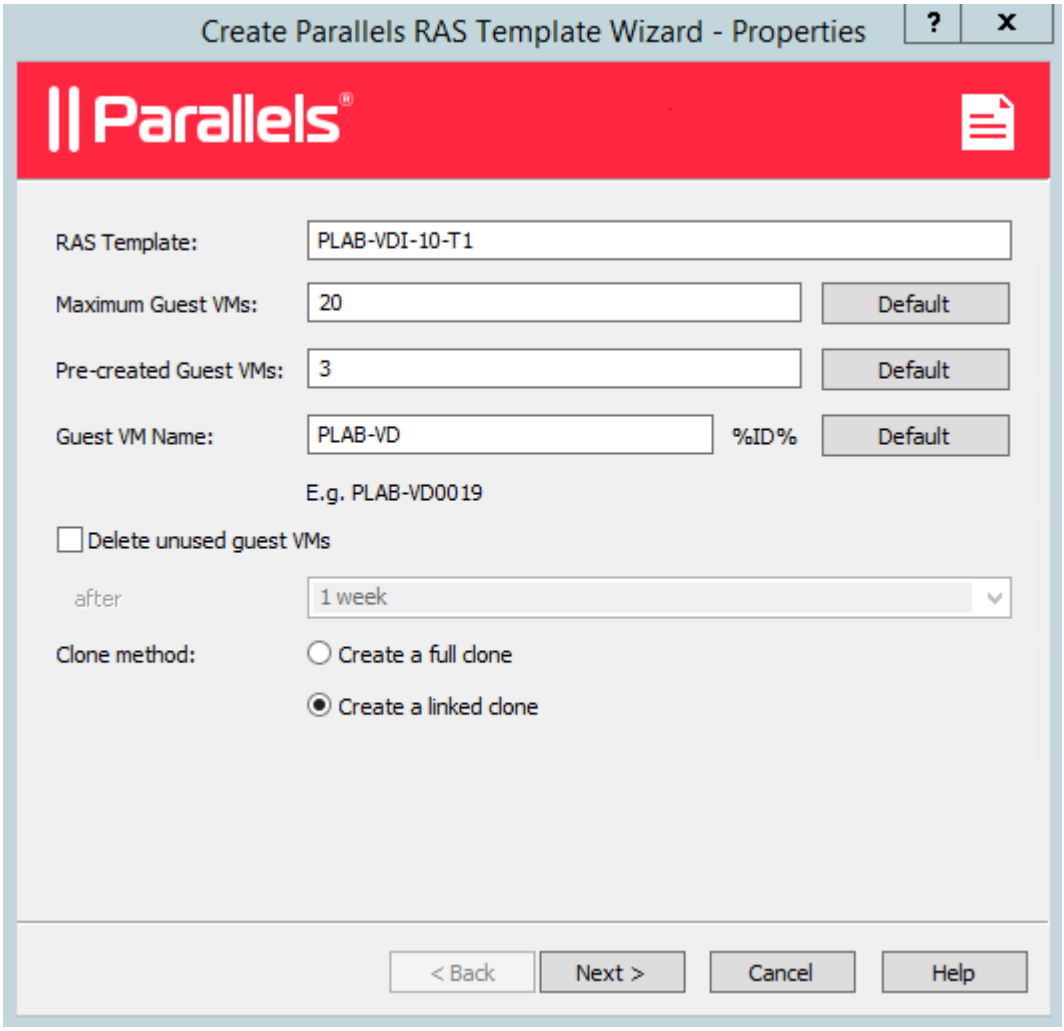
4 The wizard will allow you to deploy the RAS Guest Agent on the guest VM.

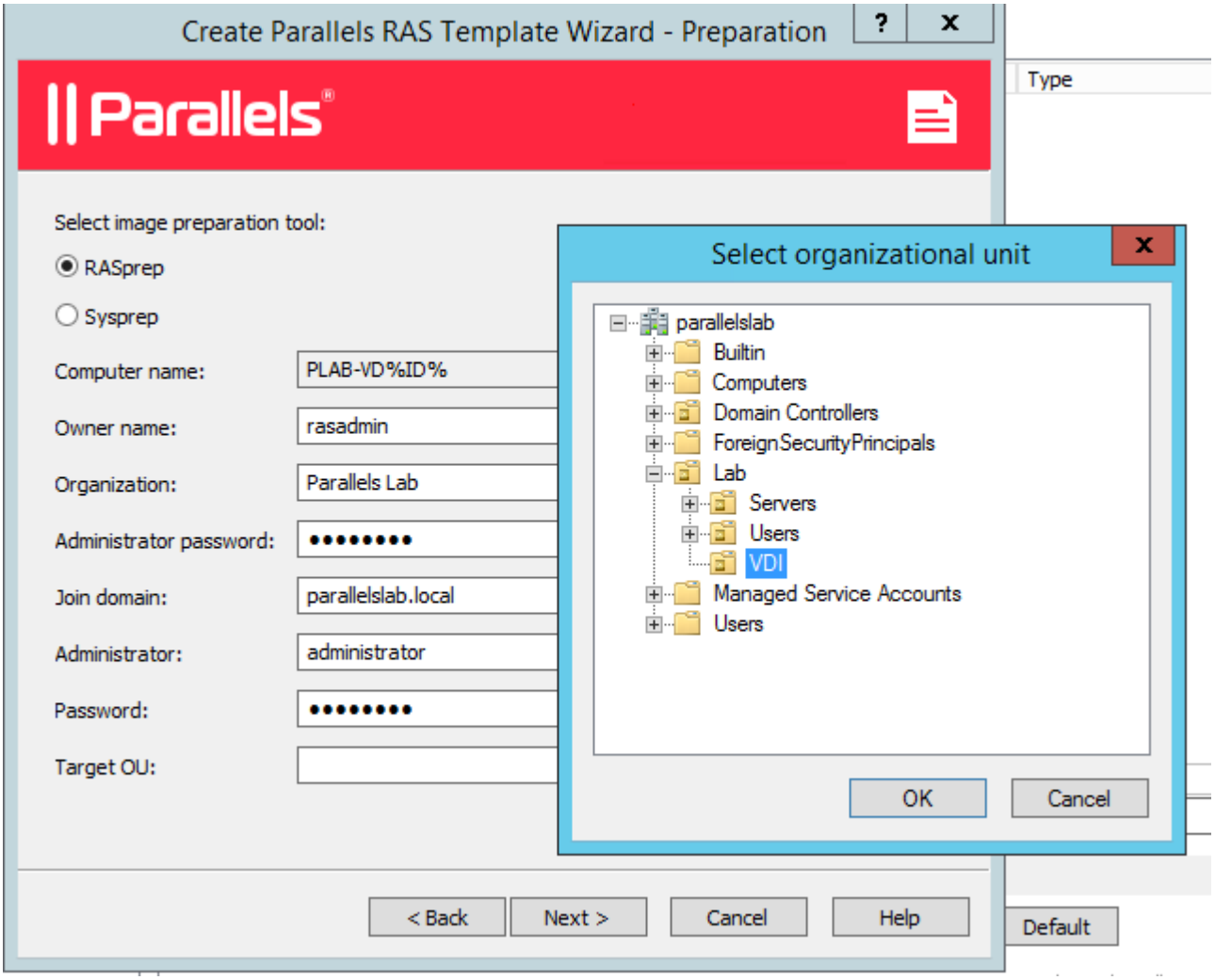


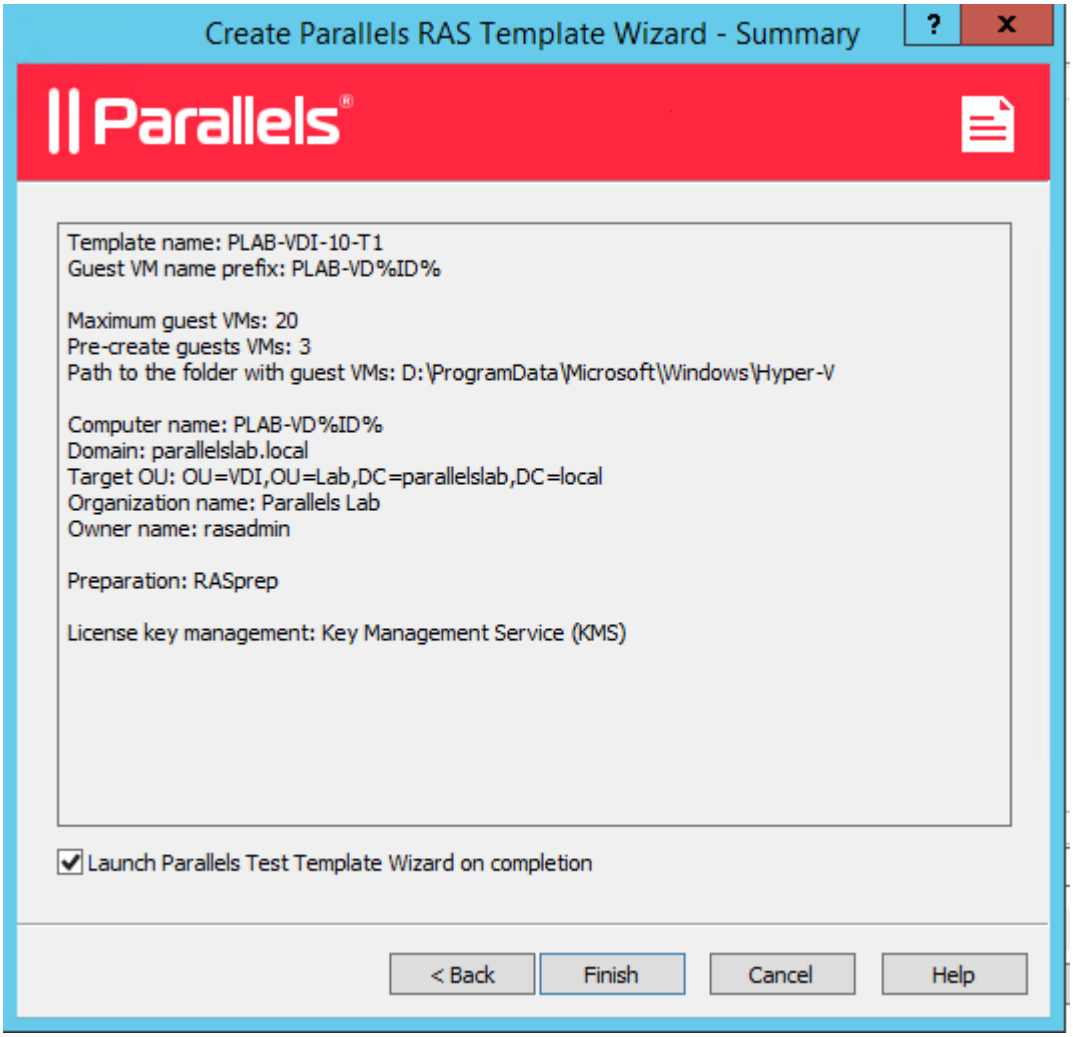
5 Once the RAS Guest Agent is deployed, we can set this VM as our VDI template.



6 Complete the RAS Template Wizard to configure the number of maximum guests and pre-created guests (full or linked clones) along with other settings including hostnames, AD details, Microsoft Licensing details, etc.

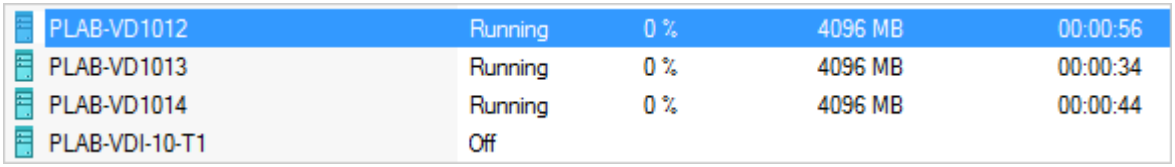



Create Parallels RAS Template Wizard - Preparation
Parallels
Select image preparation tool:
RASprep
Sysprep
Computer name:
PLAB-VD%ID%
Owner name:
rasadmin
Organization:
Parallels Lab
Administrator password:
Join domain:
parallelslab.local
Administrator:
administrator
Password:
Target OU:
< Back
Next >
Cancel
Help
Select organizational unit
parallelslab
Builtin
Computers
Domain Controllers
ForeignSecurityPrincipals
Lab
Servers
Users
VDI
Managed Service Accounts
Users
OK
Cancel
Type
Default

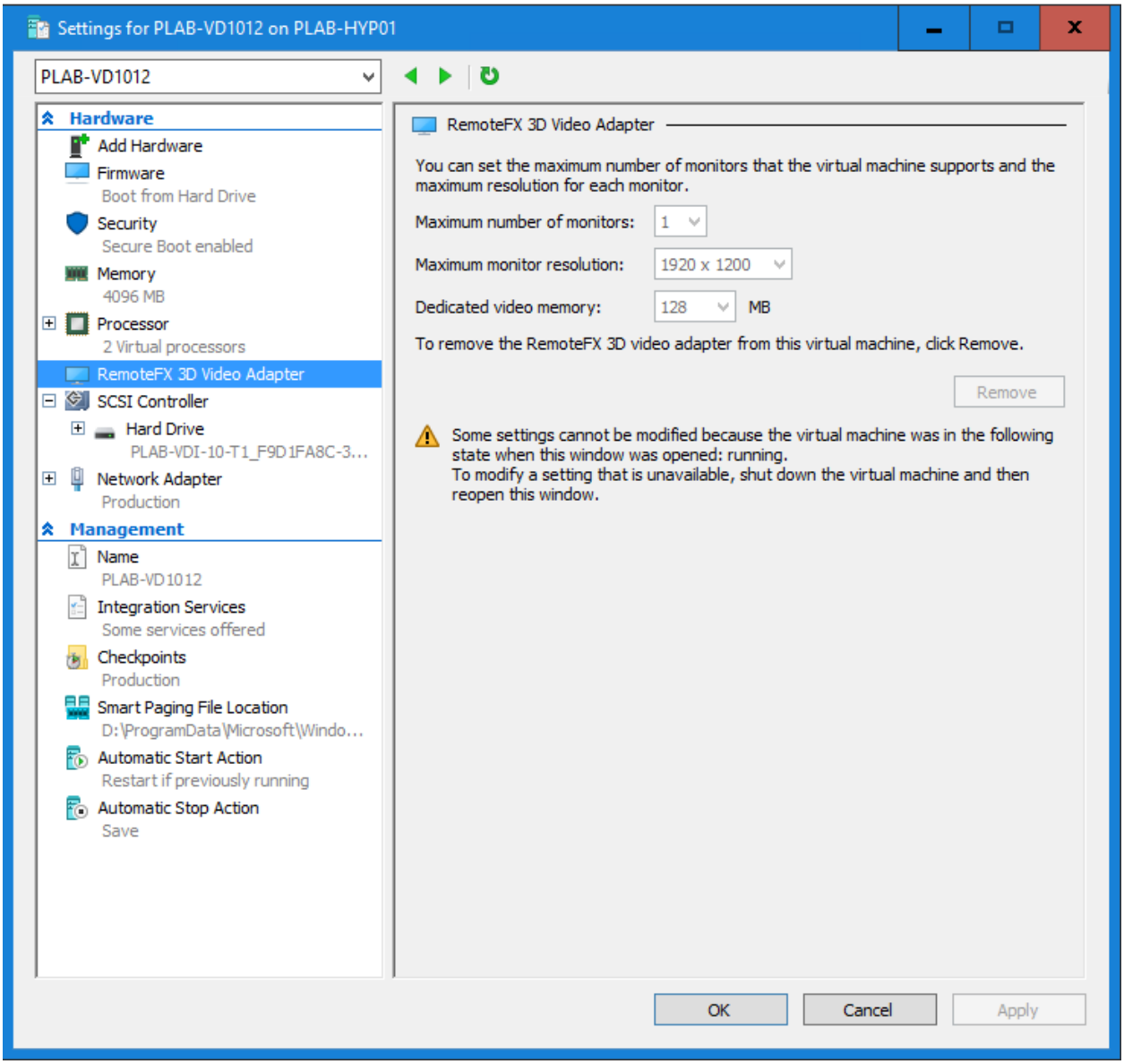


7 It is a good practice to launch the Parallels Test Template Wizard which will create a single clone of template PLAB-VDI-10-T1 and make a series of tests to check the health of the template.

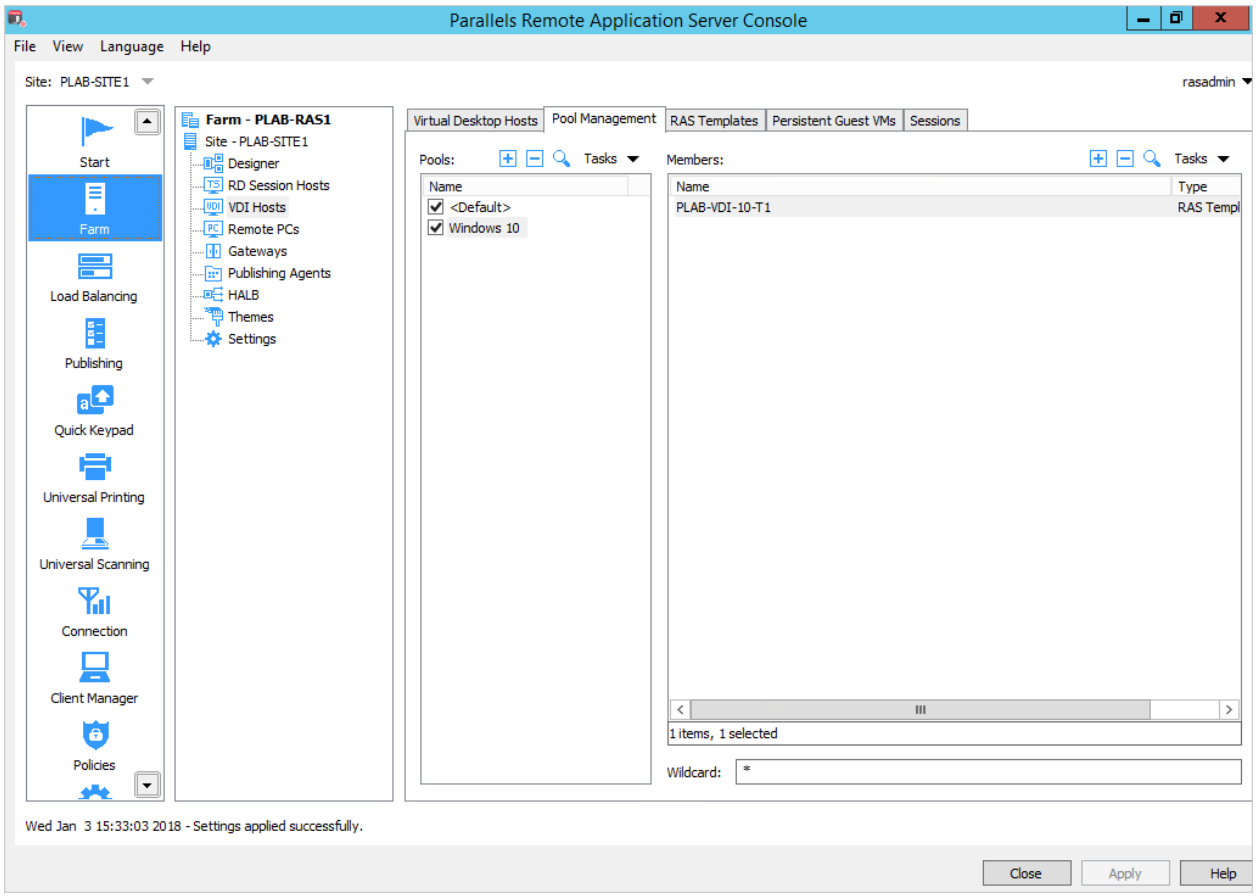
8 Now that the test and cloning have been carried out successfully we can review the cloned VMs. In our case we have cloned three VM instances from PLAB-VDI-10-T1 successfully as seen from the Hyper V manager.

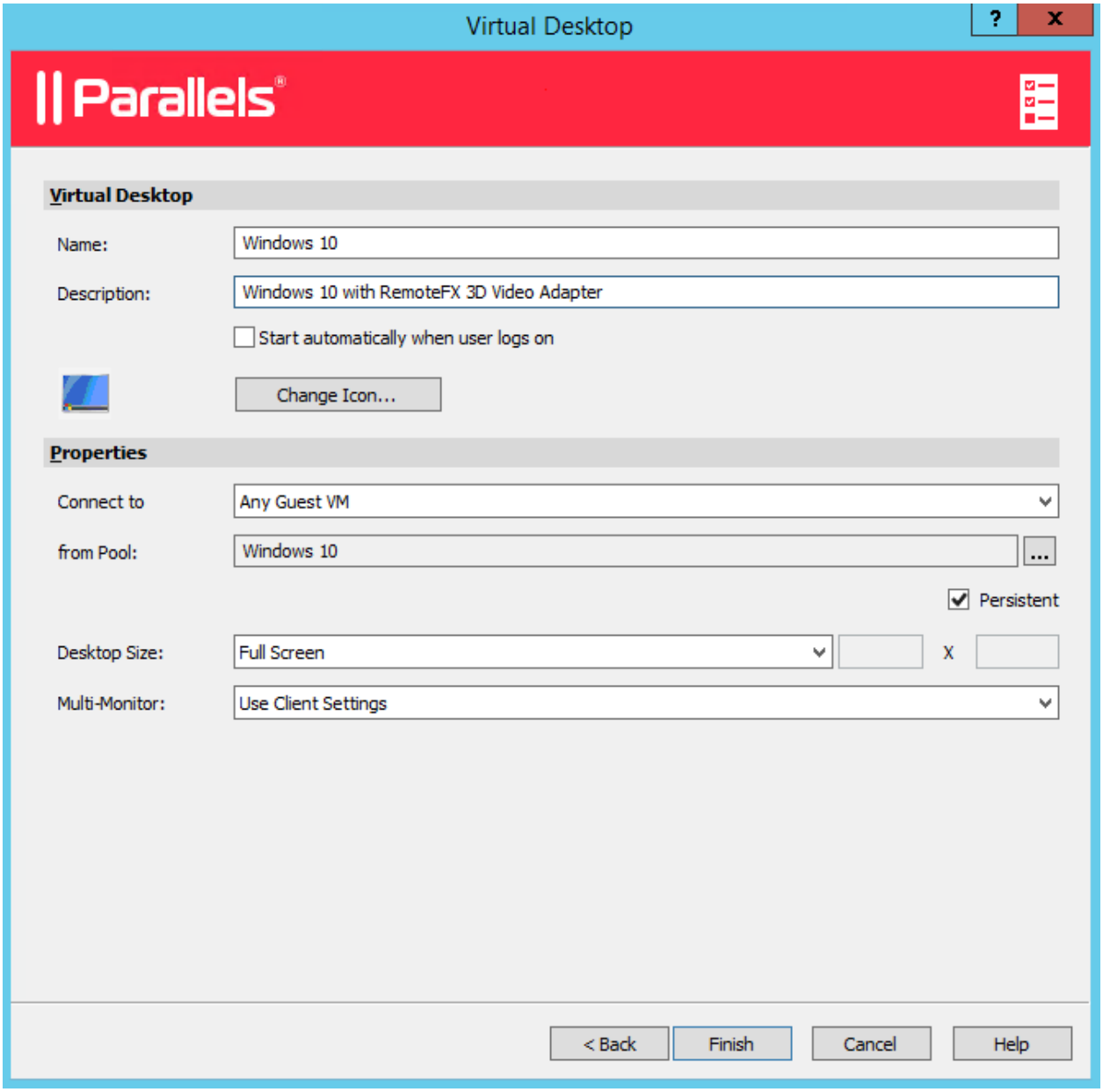


9 Looking at each created VM settings, you can note that all cloned VMs have been automatically configured with RemoteFX 3D video adapter.

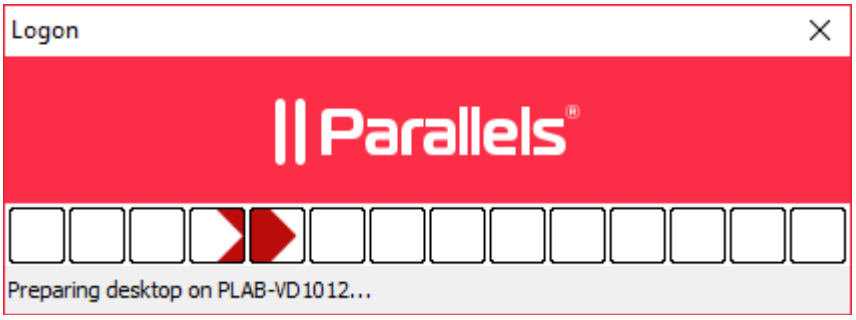


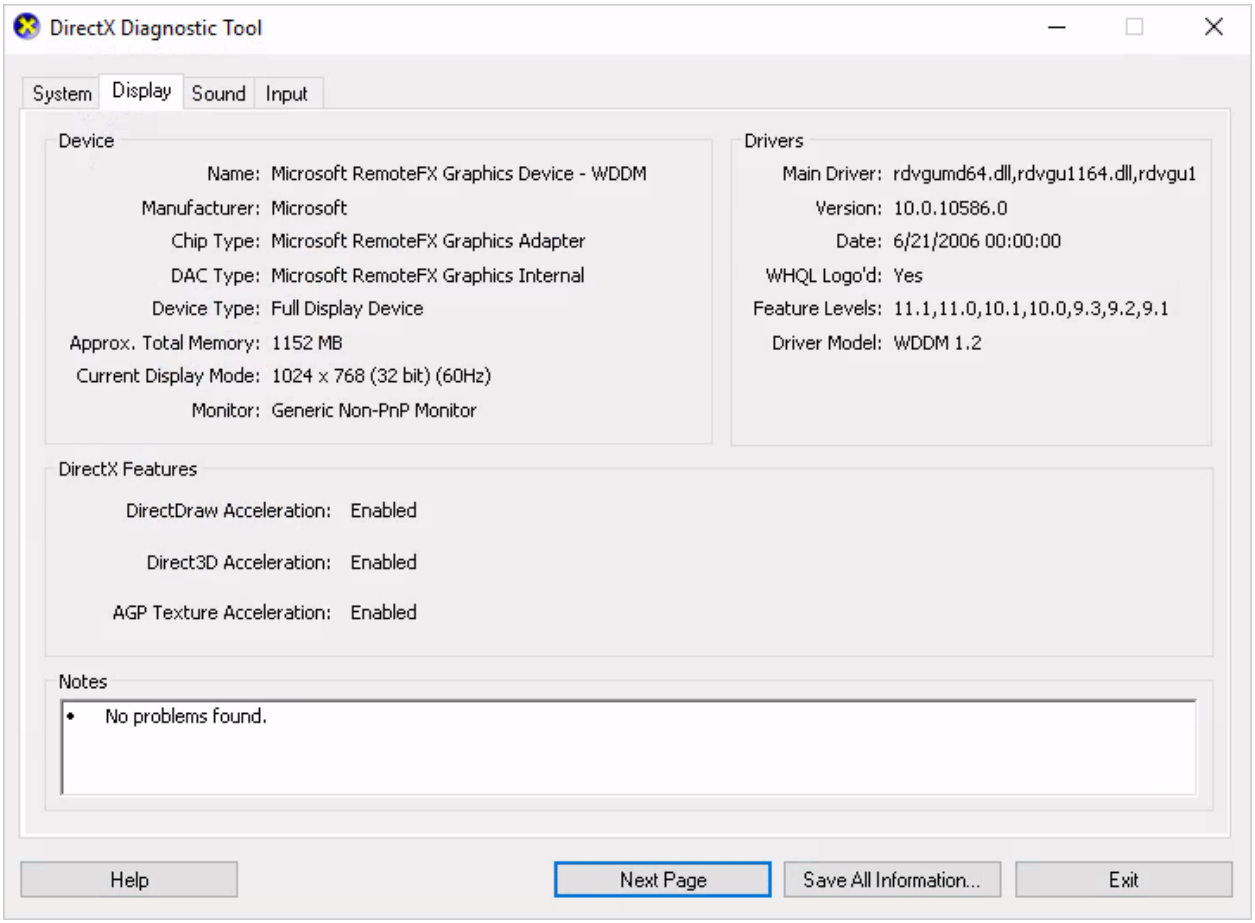
10 Finally, we can go ahead and add the Guest VMs from Template in a pool and publish VDIs from Pool.




Virtual Desktop
Parallels
Virtual Desktop
Name:
Windows 10
Description:
Windows 10 with RemoteFX 3D Video Adapter
Start automatically when user logs on
Change Icon...
Properties
Connect to
Any Guest VM
from Pool:
Windows 10
Persistent
Desktop Size:
Full Screen
X
Multi-Monitor:
Use Client Settings
< Back
Finish
Cancel
Help

11 When the user launches the published desktop, we can confirm which graphics adapter is being used from C:\Windows\System32\dxdiag.exe on the published VM and check out Display properties as shown below.





Server-Based VDI with RemoteFX vGPU

Setting up a server-based VDI is similar to the procedure described in the previous section. The difference is, instead of choosing a Desktop OS for the RAS Template, we will choose a Server OS with RemoteFX 3D Video Adapter.

Before deploying the RAS VDI Guest Agent, we need to allow the Server OS guests to be added as VDI guests by enabling RAS option 'EnableServerGuests'. This can be done via Powershell as follows:

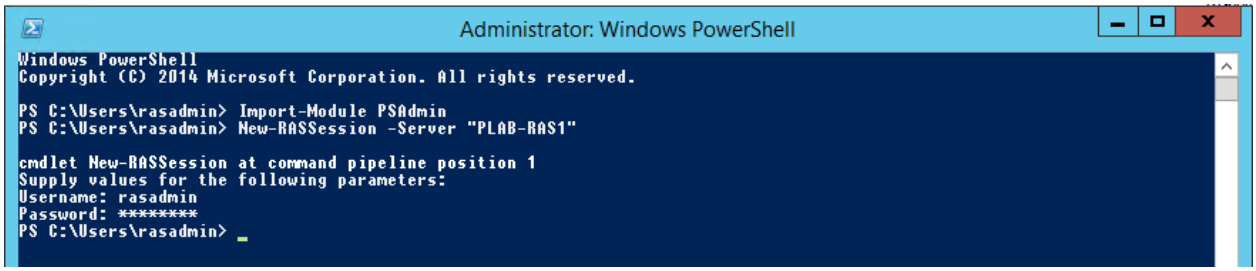
1 Create a session with the RAS setup.

- # Import the Parallels RAS PowerShell module:

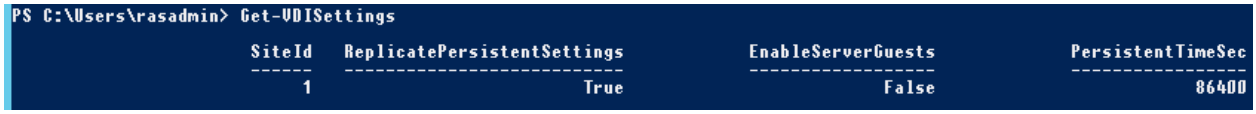
  ```
  Import-Module PSAdmin
  ```

- # Create a new Parallels RAS session.

  ```
  New-RAS Session – Server "RAS Server with Powershell API installed"
  ```



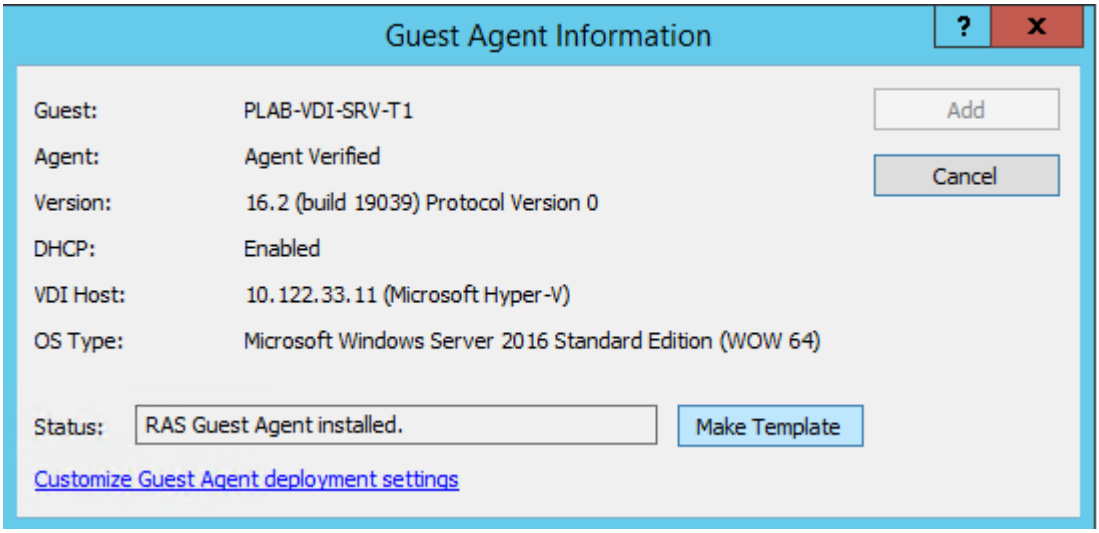
2 Run `Get-VDISettings` to confirm the status of EnableServerGuests, which by default is set to False.



3 We need to change this to true by using `Set-VDISettings [-EnableServerGuests <bool>]`

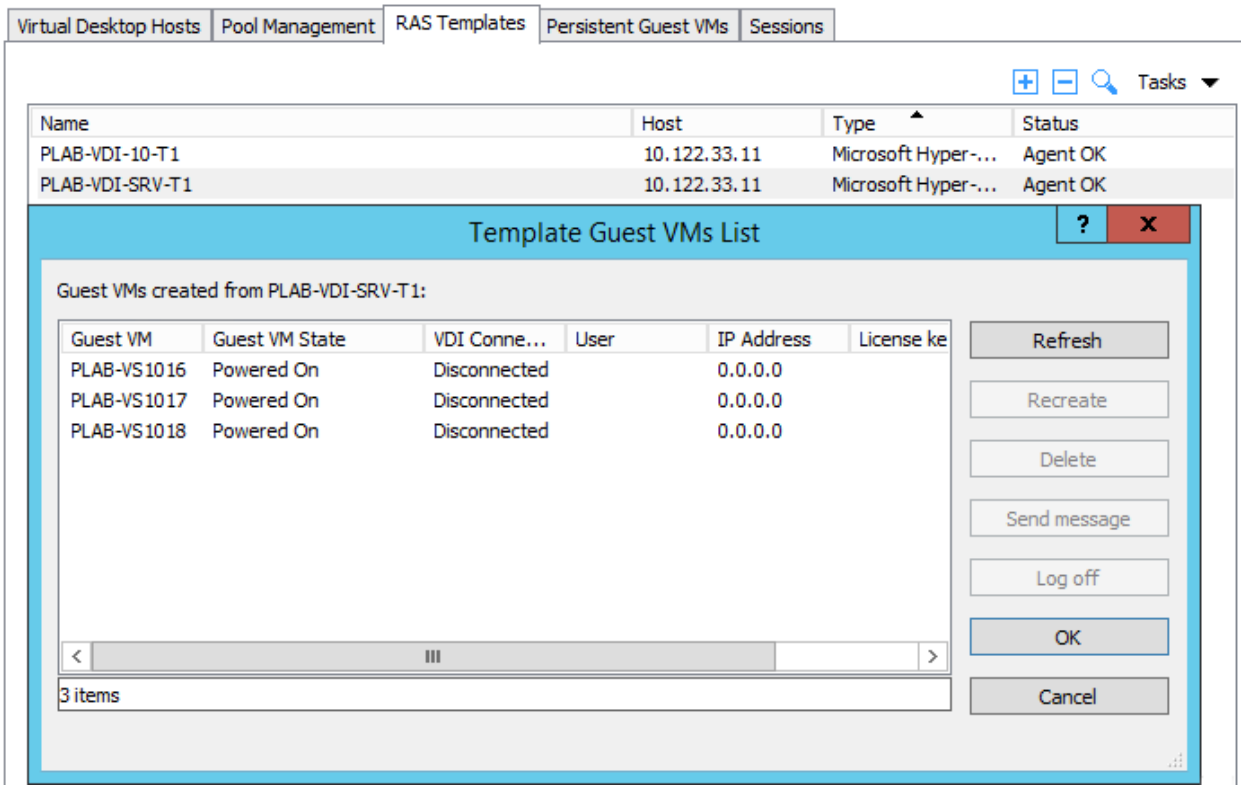
At this point we can continue to use Powershell for deployment or revert to RAS console and carry out the remaining config to setup Server based VDI. For the complete list of RAS Powershell cmdlets, please see http://download.parallels.com/ras/v16/docs/en_US/Parallels-RAS-v16-1-PowerShell-Guide/cmdlets/PSAdmin/index.html

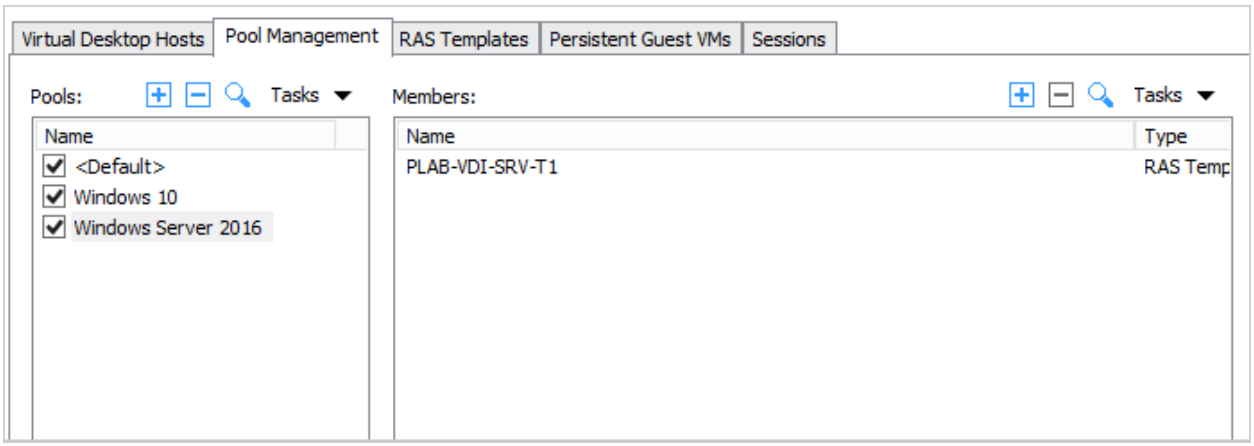
4 Choosing to add the Server OS as a RAS VDI template is now allowed.



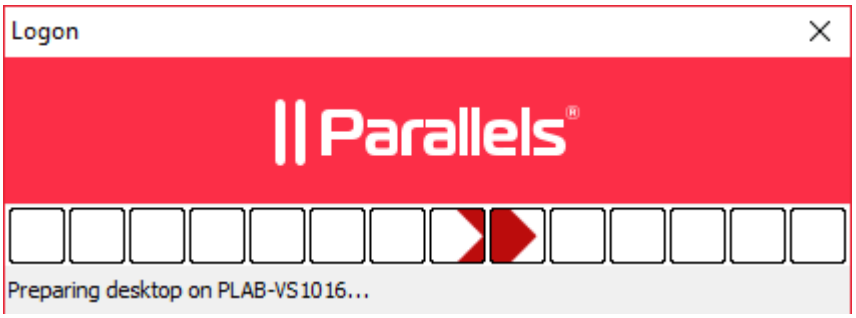
5 The same wizard of desktop based VDI deployment is shown and now we can create full or linked clones from the Server OS template to use Server based VDI with RemoteFX 3D Video Adapter.

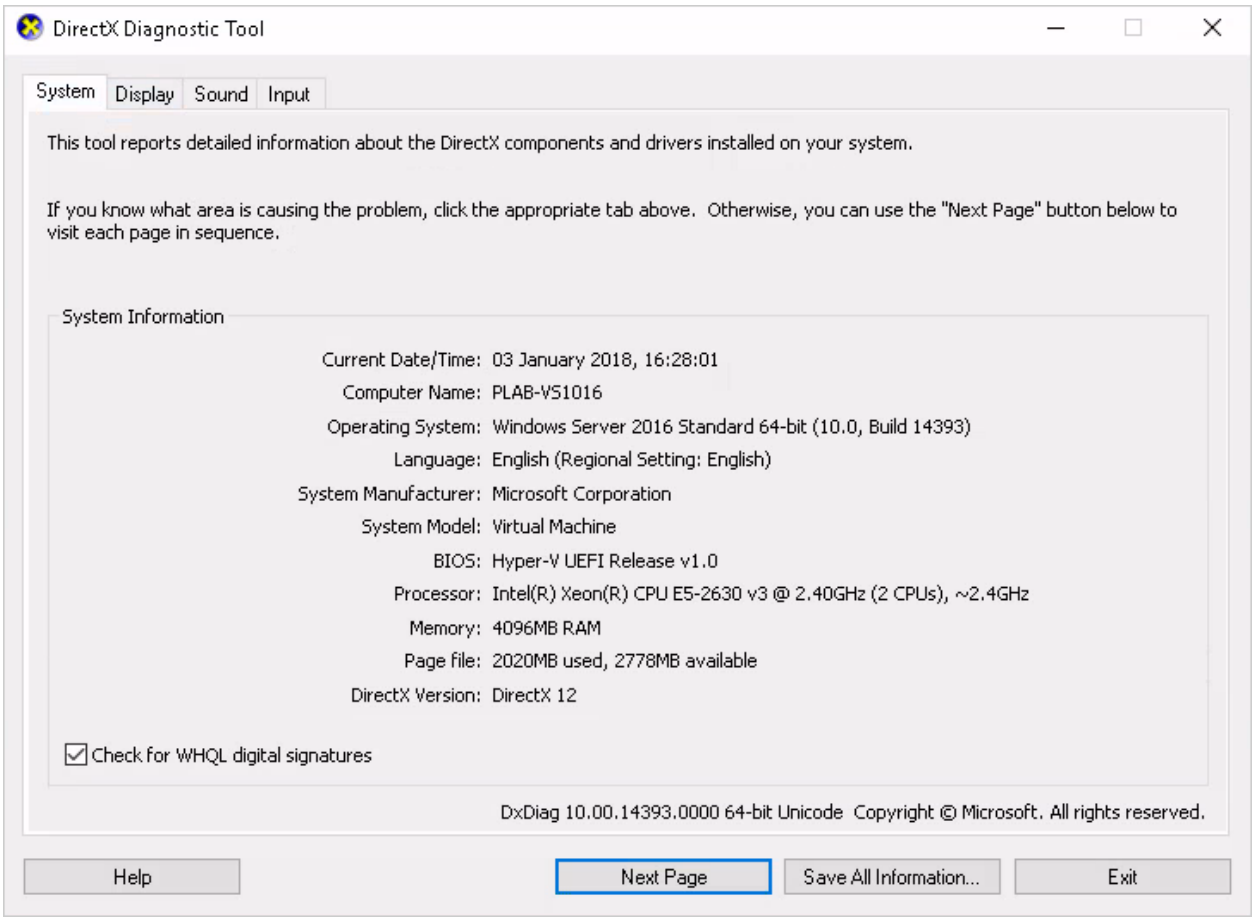
For this example we have pre-created three Server based VDI linked clones and assigned them to a newly created VDI pool named Windows Server 2016. From here, we can move on to publishing and providing access to users.

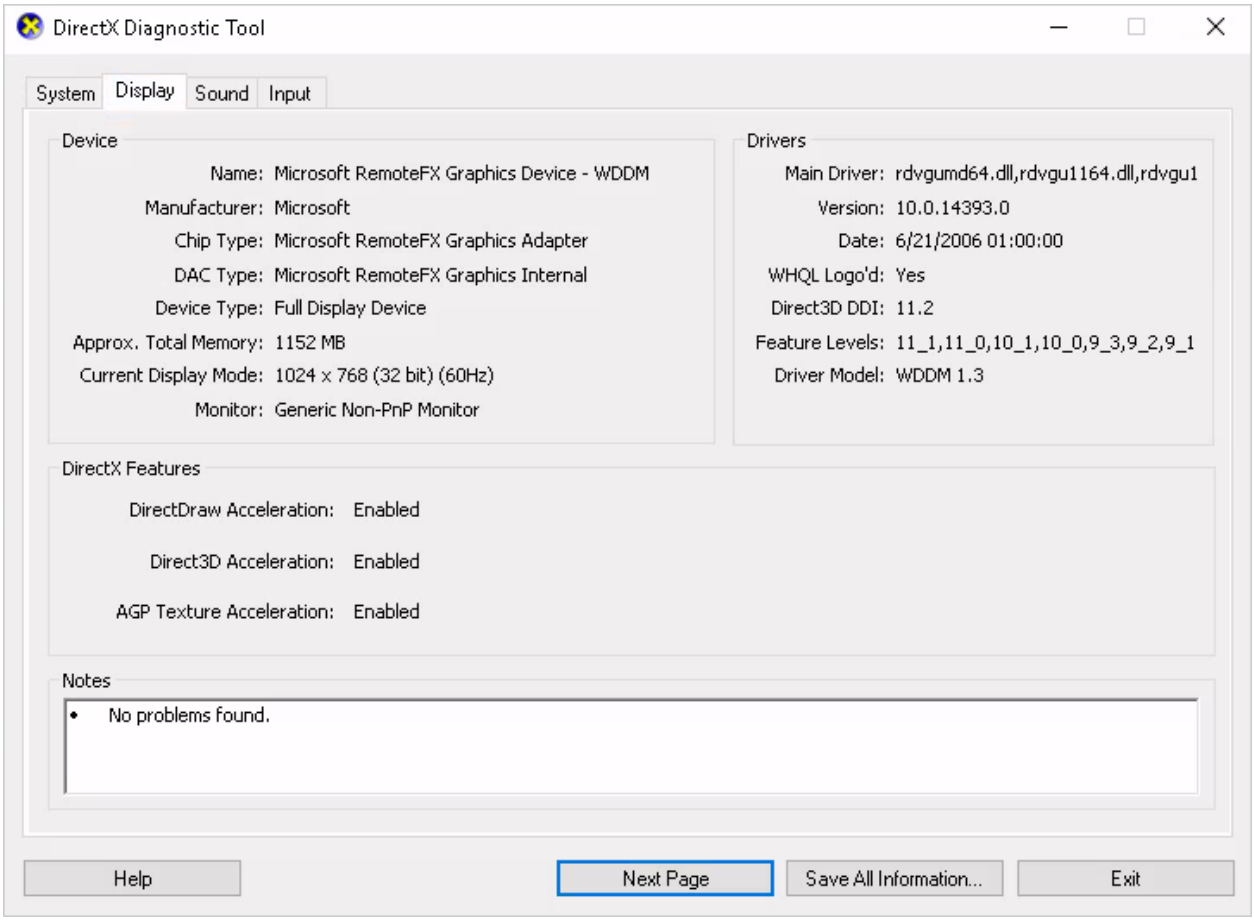




6 As user launches the published desktop, we can confirm which graphics adapter is being used from C:\Windows\System32\dxdiag.exe on the published VM and check out Display properties as shown below.





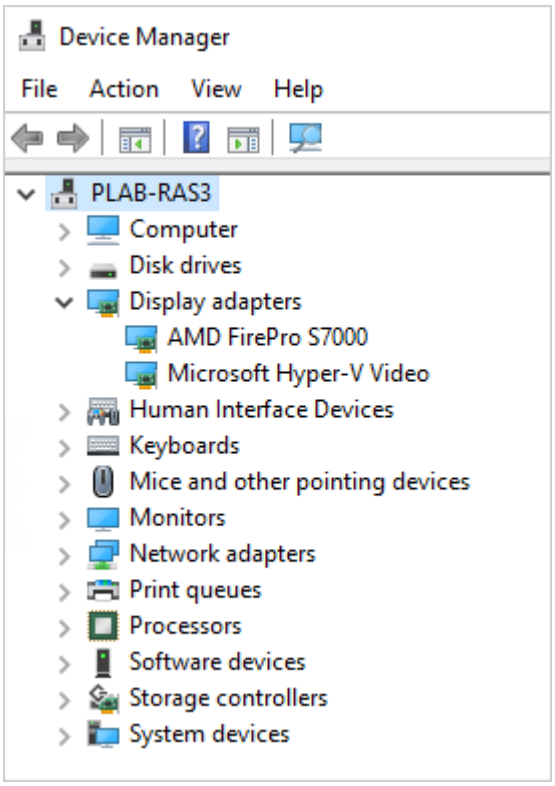


RD Session Hosts with DDA

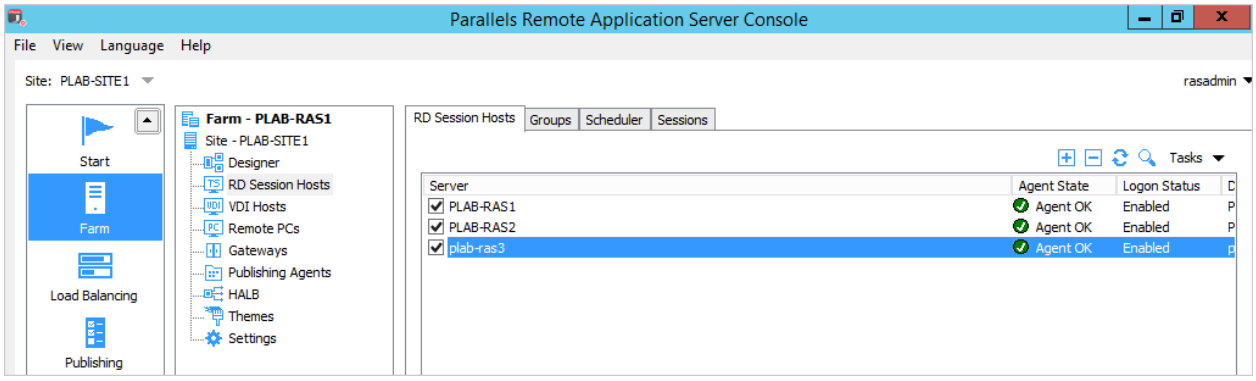
After implementing DDA for Windows Server 2012R2 or Windows Server 2016 used as RDSH (as discussed in the previous section) we need to configure RDSH publishing from Parallels RAS. It's a simple procedure and the standard process of deploying the RAS RD Session Host Agent on that VM. Publishing can then be done either via RAS console or RAS PowerShell.

Note: For the information on how to deploy the RAS RD Session Host Agent and publish applications or session-based desktops from RDSH, please refer to the Parallels RAS Administrators Guide.

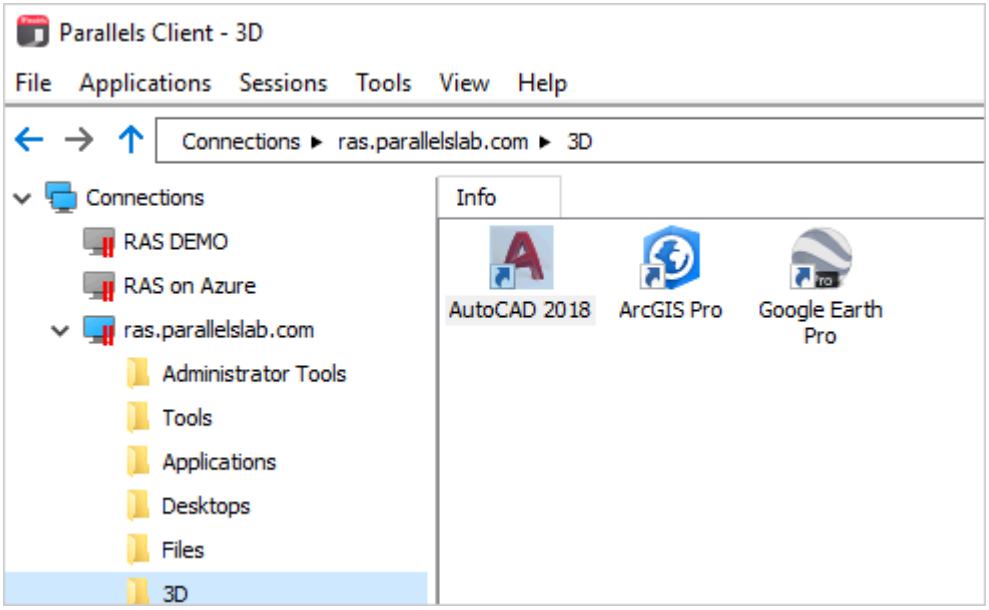
1 Use the Device Manager to confirm that the GPU has been assigned to PLAB-RAS3 and the correct drivers are installed on the RDSH. See **How to Setup Discrete Device Assignment** for instructions (p. 14).



2 Deploy RAS RD Session Host Agent to the intended RDSH with DDA configured.



3 Publish applications and session-based desktops from the same RDSH while utilizing GPU directly.



CHAPTER 4

Conclusion

As discussed in this document, Windows Server 2016 provides two options to utilize the GPU, directly or indirectly, using DDA or improved RemoteFX vGPU respectively via Microsoft Hyper-V 2016. In general, it is recommended to use DDA for the best application compatibility and performance since the VM will have direct access to the GPU and can utilize its full capabilities. However, if applications or desktops do not have such strict GPU requirements, using RemoteFX vGPU might be the ideal setup. Another important point to consider is that with DDA the GPU is solely attached to a particular VM and cannot be used by other VMs, which makes it less scalable. This, however, is the supported way to have RDSH with multiple users utilize physical GPU power.

Parallels RAS is able to support both scenarios as the decision to use RemoteFX vGPU or DDA should be based on application compatibility and workload. If the intention is to utilize RDSH to publish your applications and desktops with GPU acceleration, DDA would be the recommended solution. If using VDI based deployments with non-specific GPU requirements, RemoteFX vGPU could be a more flexible choice.

CHAPTER 5

References

https://docs.microsoft.com/en-us/windows-server/remote/remote-desktop-services/rds-supported-config#support-for-gpu-acceleration-with-hyper-v

https://blogs.technet.microsoft.com/hybridcloudbp/2016/11/15/new-rds-capabilities-in-windows-server-2016-for-service-providers/

https://docs.microsoft.com/en-us/windows-server/remote/remote-desktop-services/rds-graphics-virtualization

https://docs.microsoft.com/en-us/windows-server/virtualization/hyper-v/deploy/deploying-graphics-devices-using-dda

https://cloudblogs.microsoft.com/enterprisemobility/2013/11/05/gpu-requirements-for-remotefx-on-windows-server-2012-r2/

Index

C

Conclusion - 40

D

Desktop-Based VDI with RemoteFX vGPU - 22
Discrete Device Assignment (DDA) - 13

G

GPU Acceleration Technologies - 5

H

How to Setup Discrete Device Assignment (DDA) - 14
How to Setup RemoteFX vGPU - 6

I

Introduction - 4

P

Publishing GPU Accelerated Applications with Parallels RAS - 22

R

RD Session Hosts with DDA - 37
References - 41
RemoteFX vGPU - 5
RemoteFX vGPU vs. DDA Capabilities - 20

S

Server-Based VDI with RemoteFX vGPU - 32