# Radeon Developer Panel Documentation

Release 1.3.0

**AMD Developer Tools** 

Jan 22, 2019

### Contents

1	Getting started on Windows	3
2	Getting started on Linux	5
3	How to profile your application3.1Profiling on a local system3.2Capturing using the keyboard on Linux3.3Profiling on a remote system	7 7 11 11
4	How to use the Driver Settings	13
5	Using the Clock settings	17
6	The Connection log	19
7	The Radeon Developer Service7.1Radeon Developer Service for desktop developer system7.2Radeon Developer Service for headless GPU systems	<b>21</b> 21 22
8	Known Issues8.1Cleanup After a RadeonDeveloperServiceCLI Crash8.2Windows Firewall Blocking Incoming Connections8.3Disabling Linux Firewall8.4Setting GPU clock modes on Linux8.5Running the Panel with elevated privileges8.6Radeon Developer Panel connection issues on Linux8.7Missing Timing Data for DirectX 12 Applications8.8Radeon Developer Service Port numbers8.9Problems caused by the presence of non-AMD GPUs and non-AMD CPUs with integrated graphics	23 23 25 25 26 26 26 26 29 29

The Radeon Developer panel is part of a suite of tools that can be used by developers to optimize DirectX12 and Vulkan applications for AMD GCN graphics hardware. The suite is comprised of the following software:

- **Radeon Developer Mode Driver** This is shipped as part of the AMD public driver and supports the developer mode features required for profiling and debugging.
- **Radeon Developer Service (RDS)** A system tray application that unlocks the Developer Mode Driver features and supports communications with high level tools.
- **Radeon Developer Service CLI (Headless RDS)** A console (i.e. non-GUI) application that unlocks the Developer Mode Driver features and supports communication with high level tools.
- **Radeon Developer Panel (RDP)** A GUI application that allows the developer to configure driver settings and generate profiles from DirectX12 and Vulkan applications.
- Radeon GPU Profiler (RGP) A GUI tool used to visualize and analyze the profile data.

This document describes how the Radeon Developer Panel can be used to profile frame-based applications on AMD GCN graphics hardware. The Radeon Developer Panel connects to the Radeon Developer Service in order to collect a profile.

### Getting started on Windows

• To collect complete profiles for DirectX 12, run the script 'AddUserToGroup.bat' in the scripts folder. See *Missing Timing Data for DirectX 12 Applications* for more information

#### Getting started on Linux

- After installing the latest video driver, make sure that the current user is a member of the graphics group. This can be done by typing: "sudo usermod -a -G video \$LOGNAME" Where \$LOGNAME is the user name. Log out and log back in again for this to take effect.
- Install the vulkan shared library by typing "**sudo apt-get install libvulkan1**" or by downloading the Vulkan SDK from the LunarG website.
- Run the scripts/EnableSetClockMode.sh script to enable setting the GPU clock. See Setting GPU clock modes on Linux for more information
- If you wish to capture profiles using the keyboard hotkey, see Capturing using the keyboard on Linux

## Chapter $\mathbf{3}$

#### How to profile your application

#### 3.1 Profiling on a local system

**IMPORTANT:** The application you want to profile must NOT already be running. The panel needs to be configured in advance of starting your application.

1. **Start the RadeonDeveloperPanel.exe** on your local system. The panel will startup up with the Connection tab already highlighted (see below).

Radeon Devel	oper Panel - V1.3.0.2	_	×
	ION		?
Connect	ion status x		
Show connection	n log		
Recent	connections		
Select a rece	It Radeon Developer Service to connect to:		
IP Address	Port		
localhost	N/A		
Remove	Remove all		
New cor	inection		
Make a new o	onnection to a Radeon Developer Service.		
Address ocalhost			
Connect			

The UI has three main elements:

- Connection status to the Radeon Developer Service (currently not connected)
- Recent connections list shows a history of your previous connections
- New connection section that allows you to make either a new local or remote connection
- 1. Connect the Radeon Developer Panel to the Radeon Developer Service using either of the following methods:
  - (a) **Select the localhost entry in the "Recent connections" list, then click the "Connect" button.** This will start the Radeon Developer Service on the local system and establish a connection.
  - (b) **Double-click on any Recent connection**, and RDP will attempt to establish a connection to RDS running on the given host.

Note that the red dot to left of the "CONNECTION" tab should change to green to indicate that the connection was successful.

You may get a "Failed to connect to RDS" pop up message when running the panel for the very first

time. If the Radeon Developer Service is not running, the panel will try to start the service automatically for local connections and this can fail due to Windows file permissions (the Radeon Developer Service will not be a known application to Windows and the program will not be permitted to start). If you see this message try manually starting the "RadeonDeveloperService.exe" and connect again.

Radeon Developer Page 1	anel - V1.3.0.2				_		$\times$
	SETTING	S CI	LOCKS	PROFILING			
Connection	status	loper Service runn <b>n</b> k with.	Ning at \\.\pipe\	AMD-Developer-Service:0			
New target executable	name					Add to list	
Executable name	Apply settings	Enable profiling					
Remove Active appli	cations						
Actively running app	lications must be r	estarted to enable	applying setting	gs and profile collection.	Ad	dd to targe	ts
Executable name	API DirectX 12						

- 1. Select the executable you want to profile using either of the following methods:
  - (a) Use the "..." button to browse to the executable, or manually type it in the executable name textbox.
  - (b) Select an active process within the Active Applications table, and click the "Add to targets" button. The process will need to be restarted in order to apply settings at application startup, and to enable collection of RGP profiles.
- 2. Click the "Add to list" button to add the new executable to the list of processes that will start in Developer Mode.
- 3. The "Enable profiling" check box should be checked automatically for the application you just added to the list.
- 4. Start your application.

The driver will render an overlay on top of the application's render window if all is working correctly. The overlay will indicate if Profiling is enabled for the application, and will display the Client Id that RDP uses to communicate with the process.



The panel will detect when your application has started, and will switch to the Profiling tab.

Radeon Developer Par	nel - V1.10.0.0				×
CONNECTION	SETTINGS	CLOCKS	PROFILING		?
Profiling targ	jet status				
Process D30 Process ID 182	012 Sample.exe	Capture profile			
API Dire Client ID 707	ectX 12 +	Hotkey: Ctrl+Shift+C			
Recently col	lected profiles				
Profile	Size	Created			
Onen erafia					
	lings				
Directory to store you	ur profiles:				
D:\Development\Profile	5				
Path and filename to	Radeon GPU Profiler				

1. Click the "Capture profile" button or press the Ctrl + Shift + C hotkey to generate an RGP profile. The hotkey can be useful when capturing profiles from applications running full screen or when an app requires focus when rendering. After a few seconds a new profile should appear in the list below.

Note: Certain anti-virus software may block the hotkey feature from monitoring key presses.

1. Right-clicking on a row in the list of recent profiles will open a context menu for the selected file. The context

menu allows you to quickly navigate to the profile location in the filesystem, and rename or delete the file.

Radeon Develop	oer Panel - V1.10.0.0				
CONNECTI	ON SETTI	NGS	CLOCKS	PROFILING	
Profiling	target statu	5			
Process Process ID API Client ID	D3D12 Sample.exe 18292 DirectX 12 7072	He	Capture profile tkey: Ctrl+Shift+C		
Recently	collected pr	ofiles			
Profile		Size	Created		
D3D12 Sample-2	20171205-153811.rgp	2.437 MB	Tue Dec 5 15:29.11 2017		
			Open pr	ofile	
			Show in	Explorer	
			Rename		
			Delete p	rofile	
Open profile					
Profiling	settings				
Directory to sto	re your profiles:				
Diffectory to stor	Desflee				
rath and filenar	me to readeon GPU F	romer:			
/ Charles and a second s	Pelease\PadeooGPUPro	filer exe			

2. To open a profile file in the Radeon GPU Profiler, select the profile in the list and **click the "Open profile" button** or **double-click the selected row**.

#### 3.2 Capturing using the keyboard on Linux

Some applications capture focus or run fullscreen which makes capturing a profile difficult. The RadeonDeveloper-Panel provides a hotkey to allow capturing using the keyboard. Presently, this is Shift-Ctrl-C. On Linux, the hotkey is only available when starting the Panel with elevated privileges (ie sudo RadeoDeveloperPanel). Root privileges are needed in order to read the keyboard device, which by default is found in the path '/dev/input/by-path', and is a file ending with 'event-kbd'. If this path doesn't exist or the keyboard device has a different name, copy the KeyboardDevice.txt file from the docs directory to the root folder where these tools are located and edit this file so it contains the full path and file name of the keyboard device on your system.

#### 3.3 Profiling on a remote system

There are two variations of the Radeon Developer Service: The GUI based RadeonDeveloperService and the Radeon-DeveloperServiceCLI (command line interface). For headless operating systems which do not support a graphical user interface, the RadeonDeveloperServiceCLI executable can be started from a terminal console window.

The following steps are used to connect the RadeoDeveloperPanel to a remote Radeon Developer Service:

- 1. Start the RadeonDeveloperService or RadeonDeveloperServiceCLI executable on the remote system.
  - (a) NOTE: RadeonDeveloperServiceCLI is a command line version of the Radeon Developer Service that has no UI components and is designed to run from the command line. Please note that no system tray icon will appear when the command line version of the service is running.
- 2. Start the RadeonDeveloperPanel executable on your local system. The panel will start up with the Connection tab already highlighted (see below).
- 3. In the **New connection** section, fill in the **Address** text box with the IP address of the remote system running the Radeon Developer Service.
- 4. Click the "Connect new" button. This will establish a connection to the remote system. The red dot to left of the "CONNECTION" tab should change to green to indicate that the connection was successful.

Radeon Develo	per Panel - V1.3.0.2		_	
CONNECT	ION			
Connect	on status			
Connect	on status			
	<b>—×</b> —	-5)		
You are not co	nnected to an instance of the F	Radeon Developer Service.		
Show connection	log			
Recent o	onnections			
Select a recen	Radeon Developer Service to	connect to:		
IP Address	Port			
localhost	N/A			
Demove	Demove all			
Remove	Keniove all			
New con	nection			
Make a new o	nnection to a Radeon Develop	er Service		
Make a new c	intection to a Radeon Develop	er Service.		
Address	Port			
192.168.1.21	27300			
Connect				
Connect				

5. Go to step 3 in "Profiling on a local system" above and continue.

#### How to use the Driver Settings

**NOTE:** Currently, the driver settings are only implemented for DirectX12. Vulkan driver settings will be available soon.

The Radeon developer Panel (RDP) allows the developer to modify driver settings to experiment with features that may affect performance and quality. When you run RDP for the first time the driver settings are empty in the tool and you will need to run your application with the panel once to retrieve the driver settings. This is a one-time setup process.

The important thing to remember is that when you change settings they will only be applied the next time you start the application. Changes to the settings do not effect a currently running application.

1. To get started with settings configure your connection, connect, and setup your application as shown below.

CONNECTION       SETTINGS       CLOCKS       PROFILING         Connection status	Radeon Developer P	anel - V1.3.0.2				 _	
Connection status         Image: Connection by the state on Developer Service running at \\_bipe\AMD-Developer-Service:0         Image: Connection hop         Securation and         Securation you want to work with.         Image: Connection hop         Securation and         Apply settings         Encourtable name         Encourtable name         Apply settings         Encourtable name         Apply settings and profile collection.         Add to tax         Encourtable name         Apply         DirectX 12	CONNECTION	SETTING	ss c	LOCKS	PROFILING		(
Image: Second bit is a second b	Connection	status					
You are connected to the Radeon Developer Service running at \\.\pipe\AMD-Developer-Service:0   Disconnect Show connection log   Select the application you want to work with.     New target executable name     Apply settings     Enable profiling   D3D12 Sample.exe     Remove     Active applications   Actively running applications must be restarted to enable applying settings and profile collection.     Add to tar	口—		(	5)			
Select the application you want to work with.     New target executable name     Apply settings     Fable profiling     D3D12 Sample.exe     Image: Comparison of the set of	You are connected to Disconnect Sho	to the Radeon Devo	eloper Service runi	ning at \\.\pipe	e\AMD-Developer-Service:0		
New target executable name Apply settings   Executable name Apply settings   D3D12 Sample.exe Image: Comparison of the set o	Select the application	t applicatio	<b>n</b> k with.				
Executable name       Apply settings       Enable profiling         D3D12 Sample.exe       Image: Comparison of the set of th	New target executable	e name	<b>5</b> 11 (11)				Add to list
Remove         Active applications         Actively running applications must be restarted to enable applying settings and profile collection.         Add to ta         Executable name       API         D3D12 Sample.exe       DirectX 12	Executable name	Apply settings	Enable profiling				
Remove         Active applications         Actively running applications must be restarted to enable applying settings and profile collection.       Add to ta         Executable name       API         D3D12 Sample.exe       DirectX 12							
Remove         Active applications         Actively running applications must be restarted to enable applying settings and profile collection.       Add to ta         Executable name       API         D3D12 Sample.exe       DirectX 12							
Active applications       Add to take the second of the seco	Remove						
Actively running applications must be restarted to enable applying settings and profile collection.       Add to tal         Executable name       API         D3D12 Sample.exe       DirectX 12	Active appli	cations					
Executable name     API       D3D12 Sample.exe     DirectX 12	Actively running app	olications must be i	restarted to enable	applying setti	ings and profile collection.	A	Add to targe
D3D12 Sample.exe DirectX 12	Executable name	API					
	D3D12 Sample.exe	DirectX 12					

- 2. **Start your application** and let it run for a short while (few seconds) then terminate the process. This will populate the driver settings in the tool.
- 3. Click on the Settings tab
- 4. Currently, there are two categories of settings (Debug and General), and there are only 4 settings in total. Many more will be made available soon. The General settings are shown below. Click on the small arrow to the right of the setting name to see the possible values and descriptions. The "Default All" button will reset the values back to the original driver settings. Settings can also be exported and imported.

Radeon Developer Panel - V1.0	0.0	-		×
• CONNECTION S	ETTINGS CLOCKS PROFILING			?
Debug		Searc	ch	
General	Disable Color compression optimizations.   ⊥   ⊥   Disables Depth/Stencil compression optimizations   Isables Depth/Stencil compression optimizations   Image: Color compatible MetaData   Ditask of cases where texture compatible meta data will be used   127			
	Default all Impor	t	Expor	rt

- 1. Make the changes you require to the settings and then click on the Connection tab.
- 2. Make sure you have selected the "Apply settings" checkbox on the application you wish to change the settings for.
- 3. Start your application, the settings are applied by the panel as your application starts.
- 4. Profile your application as described in the "How to profile your application" section above.

Using the Clock settings

The Radeon developer Panel (RDP) allows the developer to select from a number of clock modes.



#### Clock stability

Due to desire to operate within reasonable power envelopes, modern GPUs employ techniques which alter the frequencies of their clocks dynamically. This can make development difficult, as there is no single clock frequency which can be assumed.

Select which clock mode you would like your applications to operate in.



Default clock mode used by consumers, clock speed may vary significantly according to power constraints and application load.



Developer-only clock mode. Sets a stable clock frequency that may be lower than the normal operating frequency. Using this mode is advisable to ensure consistency between different runs of your application.

Normal clock mode will run the GPU as it would normally run your application. To ensure that the GPU runs within its designed power and temperature envelopes, it dynamically adjusts the internal clock frequency. This means that profiles taken of the same application may differ significantly, making side-by-side comparisons impossible.

Stable clock mode will run the GPU at a lower, fixed clock rate. Even though the application may run slower than normal, it will be much easier to compare profiles of the same application.

For the Radeon GPU Profiler tool, the clock settings here are not used since the driver forces a profile to take place using peak clocks.

#### The Connection log

Click on the "Show connection log" button from the CONNECTION tab to see any logging information that is produced by the the panel activity. Additional information about the connection and any errors encountered by RDP and the RDS are displayed here. Below is an example of typical output from a session that captured a profile. The log can be saved, cleared and copied to the clipboard using the buttons at the bottom.

This log is also saved in a log file located at: "C:\Users\your\_name\AppData\Roaming\RadeonDeveloperDriver\RDPLogFile.txt"

On Linux, this log is located at: "~/.RadeonDeveloperDriver/RDPLogFile.txt"

Radeon Developer Panel	- V1.3.0.2				_	0
CONNECTION	SETTINGS	CLOCKS	PROFILING			
Connection sta	atus					
口——		-5				
You are connected to th	e Radeon Developer Se	rvice running at \\.\pipe	AMD-Developer-Service:	0		
Disconnect Hide con	nection log					
Connection lo	9					
[RDP] Failed to resume driv [RDP] Failed to resume exe	– er on client cution of process 'D3D12Mul	tithreading.exe', ProcessId	= 14000			
[RDP] Wait for driver initiali: [RDP] Attempted to discorr	zation not supported on targ	et client D3D12Multithreadir	ig.exe's protocol version.			
[RDP] Received client conne	ected from unknown client wi	ith id 5292.				
[RDP] Client with Id 6097 ha	as disconnected.					
[RDP] Capture profile butto [RDP] Received dient balter	n has been disabled because d from unknown client with ic	e the application is not profili 1 5292	able.			
[RDP] Processing halted die	nt with id 5292; D3D12Multi	threading.exe: 14000 - AMD	DirectX12 Driver			
[RDP] Updated D3D12Multit	hreading.exe ClientId to 52	92				
[RDP] Connected DriverCor	trolClient to process 'D3D12	Multithreading.exe', Process	sId = 14000			
[RDP] Filtered halted proces	ss with ProcessId = 14000					
[RDP] EnableProfiling() for (	lientId 5292.	littlerooding over DresserId	- 14000			
[RDP] Can enable profiling for	for client ID 5292	iuurreading.exe, processiu	= 14000.			
[RDP] Set profiling flag for F	ProcessId = 14000 to true.					
[RDP] Capture profile butto	n is enabled because the tar	rget application is profilable a	and there is no profile in progre	SS.		
[RDP] Attempting to apply a	app-specific settings for Proc	ess 14000 D3D12Multithrea	ding.exe, currentCLientId 5292	2		
[RDP] Setting 'DxcTcCompa	tibleMetaData' applied succe	essfully	D. Discourse to discut			
[RDP] Resumed execution of [PDP] Driver has been initia	Ized within D3D12Multithread	ding.exe, Processia = 1400	u. Disconnect client.	iencies		
[RDP] Successfully queried	maximum device clocks.	angrexe process. Accompa	ig to realize device dock need	Jeneies.		
[RDP] Successfully set devi	ce clock mode to Normal to q	uery frequency.				
[RDP] Successfully queried	device clock frequencies for	Normal mode.				
[RDP] Successfully set device	ce clock mode to Stable to qu	uery frequency.				
[RDP] Successfully queried ( [RDP] Successfully collected	device clock frequencies for	Stable mode.				
[RDP] Reventing to user's d	ock mode.					
[RDP] Set clock mode to No	rmal.					
[RDP] Driver initialized succ	essfully on target client D3D	12Multithreading.exe.				
[RDP] Client with Id 5292 ha	as disconnected.	- the continuition is not fil				
[RDP] Capture profile butto	n nas been disabled because	e ule application is not profile	able.			

#### The Radeon Developer Service

Two version of the Radeon developer service are provided, one with a configuration UI and system tray icon, and one designed for use with headless GPU system where no UI can be supported.

#### 7.1 Radeon Developer Service for desktop developer system

RadeonDeveloperService(.exe) – Can be used for general use where the system has a monitor and UI (e.g. desktop development machines). The Radeon Developer Service includes a configuration window containing basic service configuration settings and software info. **Double click the Radeon Developer Service system tray icon** to open the configuration window, or right-click on the system tray icon and select 'configure' from the context menu.

S Radeon Developer Service Configuration X						
Configuration	ı					
Listen port:	27300	G				
About						
Version	1.10.551.0					
Build	551					
Build date	11/22/2017					
Copyright © Al	MD Corporation, All rights r	reserved				

• Listen port – The port that the Radeon Developer Service uses to listen for incoming connections from a remote Radeon Developer Panel. The default port is 27300. Altering the port will disconnect all existing sessions. The circular arrows icon to the right of the Listen port field can be clicked to reset the port to the default value.

• Version info – Software version information for the Radeon Developer Service.

Double click the Radeon Developer Service system tray icon again or right-click on the system tray icon and select 'configure' from the context menu to close the configuration window.

**Please note** that when running both the Radeon Developer Panel and the Radeon Developer Service on the same system the communication between the two uses pipes, not sockets and ports, so setting the port has no effect.

#### 7.2 Radeon Developer Service for headless GPU systems

RadeonDeveloperServiceCLI(.exe) – Command line version for use with headless GPU systems where no UI can be provided. NOTE: This version can also run on a system that has a monitor and UI.

The following command line options are available for RadeonDeveloperServiceCLI:

- 1. port <port number> Overrides the default listener port used by the service (27300 is the default).
- 2. enableUWP Enables UWP support (disabled by default).

**Please note** that the service will need to be explicitly started before starting the Radeon Developer Panel. If the service isn't running, the Radeon Developer Panel will automatically start the UI version of the Radeon Developer Service, which may not be what is required.

#### Known Issues

#### 8.1 Cleanup After a RadeonDeveloperServiceCLI Crash

If the RadeonDeveloperServiceCLI executable crashes on Linux, shared memory may need to be cleaned up by running the RemoveSharedMemory.sh script located in the script folder of the RGP release kit. Run the script with elevated privileges using sudo. If this fails to work, try starting the panel with elevated privileges.

#### 8.2 Windows Firewall Blocking Incoming Connections

1. **Deleting the settings file**. If problems arise with connection or application histories, these can be resolved by deleting the Radeon Developer Panel's settings file at: "C:\Users\your\_name\AppData\Roaming\RadeonDeveloperDriver\RDPSettings.xml"

on Windows. On Linux, the corresponding file is located at:

"~/.RadeonDeveloperDriver/RDPSettings.xml"

- 2. "Connection Failure" error message. This issue is sometimes seen when running the panel for the very first time. The panel tries to start the service automatically for local connections and this can fail. If you see this message try manually starting the "RadeonDeveloperService.exe" and connect again.
- 3. Remote connection attempts timing out. When running the Radeon Developer Service on Windows, the Windows Firewall may attempt to block incoming connection attempts from other machines. The best methods of ensuring that remote connections are established correctly are:
  - (a) Allow the RDS firewall exception to be created within the Windows Firewall when RDS is first started. Within the Windows Security Alert popup, enable the checkboxes that apply for your network configuration, and click "Allow access".

💣 Windows Sec	urity Alert		×				
💮 Windo	ws Firewal	l has blocked some features of this app					
Windows Firewall h and domain netwo	as blocked som	e features of Radeon Developer Service on all public, private					
	<u>N</u> ame:	Radeon Developer Service					
	Publisher:	Advanced Micro Devices, Inc.					
	Pat <u>h</u> :	C:\radeondeveloperservice.exe					
Allow Radeon Deve <u>Domain netv</u>	eloper Service to vorks, such as a	o communicate on these networks: workplace network					
Private netv	vorks, such as m	ny home or work network					
P <u>u</u> blic netwo because the	orks, such as the se networks off	ose in airports and coffee shops (not recommended en have little or no security)					
What are the risks	of allowing an a	pp through a firewall?					
		Allow access Cancel	I				

1. If "Cancel" was previously clicked in the above step during the first run, the exception for RDS can still be enabled by allowing it within the Windows Control Panel firewall settings. Navigate to the "Allow an app or feature" section, and ensure that the checkbox next to the RadeonDeveloperService.exe entry is checked:



🔐 Allowed apps							-		×
$\leftarrow \rightarrow \cdot \uparrow$	Control Panel > System and Security > Wind	lows Firewall	> Allowed	l apps	√ Ū	Sear	rch Con	trol Pa	P
			- 11						^
	Allow apps to communicate through wind	dows Firew	all						
	Io add, change, or remove allowed apps and ports, clip	:k Change sett	ings.	_					
	What are the risks of allowing an app to communicate	?			😯 Cha <u>n</u> ge setti	ings			
	For your security, some settings are managed by	your system ad	Iministrato	or.					
	<u>A</u> llowed apps and features:								
	Name	Domain	Private	Public	Group Policy	^			
	Whateen Store Launcher								
	Withdate State Launcher				100				
	Witness Store Laurcher				100				
	Constant State Cautofue		0	0	100				
	✓ radeondeveloperservice.exe				No				
	The second second		-	-	1000				
	Mill Parriate Desiting				100				
	Herroris Event Log Management	0			100				
	Harrute Event Worksr	0			100				
	Herrote Scheduled Task: Wanagement	0			100				
	Herrote Service Wanagement				100				
	Remark Buttleven				760	~			
				Details	. Remove	e			~
				C	K Car	ncel			

1. Alternatively, disable the Windows Firewall entirely will also allow RDS to be connected to.

**NOTE** The Windows firewall alert in no way indicates that the Radeon Developer tools are trying to communicate to an AMD server over the internet. The Radeon Developer tools do not attempt to connect to a remote AMD server of any description and do not send personal or system information over remote connections. The Radeon Developer Panel needs to communicate with the Radeon Developer Service, which may or may not be on the same machine, and a connection needs to be made between the two (normally via a socket).

#### 8.3 Disabling Linux Firewall

If the remote machine is running Linux and the "**Connection Failure**" error message is displayed, the Linux firewall may need to be disabled. This is done by typing "**sudo ufw disable**" in a terminal. The firewall can be re-enabled after capturing by typing "**sudo ufw enable**".

#### 8.4 Setting GPU clock modes on Linux

Adjusting the **GPU** clock mode on Linux is accomplished by writing to /sys/class/drm/card<n>/device/power\_dpm\_force\_performance\_level, where <n> is the index of the card in question. By default this file is only modifiable by root, so the application being profiled would have to be run as root in order for it to modify the clock mode. It is possible to modify the permissions for the file instead so that it can be written by unprivileged users. The Radeon GPU Profiler package includes the "scripts/EnableSetClockMode.sh" script which will allow setting GPU clock mode in cases where the target application is not, or cannot, run as root. Execute this script before running the Radeon Developer Service and target application, and the GPU clock mode can be updated correctly at runtime. This script needs to be run each time you reboot your machine; the file permissions do not survive system reboots.

#### 8.5 Running the Panel with elevated privileges

As previously mentioned, the panel only needs to be run with elevated privileges if the keyboard shortcut is needed for capturing. On Ubuntu 18.04, a dialog box may pop up indicating that the RadeonDeveloperService is running in headless mode. This is nothing to worry about and will not affect profiling in any way; it just means that the root shell doesn't have access to the user interface so is running without one. The only downside is that there won't be a 'service' icon in the system tray.

#### 8.6 Radeon Developer Panel connection issues on Linux

The Radeon Developer Panel may fail to start the Radeon Developer Service when the Connect button is clicked. If this occurs, manually start the Radeon Developer Service, select localhost from the the Recent connections list and click the Connect button again.

#### 8.7 Missing Timing Data for DirectX 12 Applications

To collect complete profile datasets for DirectX 12 applications, the user account in Windows needs to be associated with the "Performance Log Users" group. If these privileges aren't configured properly, profiles collected under the user's account may not include all timing data for GPU Sync objects.

A batch file is provided to add the current user to the group (scripts\AddUserToGroup.bat). The batch file should be run as administrator (Right click on file and select "Run as Administrator"). The script's output is shown below:

c:\Users\pmistry\RGP>AddUserToGroup.bat Copyright (c) 2017 Advanced Micro Devices, Inc. All rights reserved. This script will add the current user to the "Performance Log Users" group Domain: AMD Username: pmistry Computer Name: BDCL10-PMISTRY Running Script as Administrator The command completed successfully. Please reboot your system for these changes to take effect

..exiting

#### c:\Users\pmistry\RGP>

Alternatively, to manually add the active user to the proper group, follow these steps:

- 1. Open the Run dialog by using the Windows Start menu, or through the Windows + R shortcut.
  - (a) Type "lusrmgr.msc" into the Run window, and click OK.

🖅 Run	×
	Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.
<u>O</u> pen:	lusrmgr.msc v
	OK Cancel <u>B</u> rowse

- 2. Within the "Local Users and Groups" configuration window that opens, select the Groups node.
  - (a) Select the Performance Log Users entry. Right-click and select Properties.

🜆 lusrmgr - [Local Users and Grou	ps (Local)\Groups]		
File Action View Help			
🗢 🔿 🖄 📰 🔀 🛛	?		
Local Users and Groups (Local)	Name Access Control Assistance Operators Administrators Backup Operators Cryptographic Operators Distributed COM Users Event Log Readers Cryptographic Operators Event Log Readers Event Log Readers Comparison Guests Hyper-V Administrators Hyper-V Administrators Hyper-V Administrators Full S_IUSRS Network Configuration Operators Performance Log Users Performance Monitor Users Power Users Power Users Remote Desktop Users Remote Management Users Replicator System Managed Accounts Group Users Ssh Users	Description Members of this gro Administrators have Backup Operators of Members are autho Members are allowe Members of this gro Guests have the sam Members of this gro Built-in group used Members in this gro Add to Group All Tasks > Delete Rename Properties Help	oup can remotely query au e complete and unrestricte rized to perform cryptogra ed to launch, activate and u oup can read event logs fro ne access as members of th oup have complete and un by Internet Information Se oup can have some admini oup can access performance ded for backwards comp p are granted the right to p are granted the right to p are managed by the sy rom making accidental o ip can remotely access the

Opens the properties dialog box for the current selection.

1. To add the active user to the group, **click the Add... button**. (If the active user appears within this list, the account is already configured properly.)

Performance Log Users Properties ? ×							
General							
Performance Log Users							
Description: Members of this group may schedule logging of performance counters, enable trace providers, and							
Members:							
		Changes to a user's	aroup members	ain			
Add <u>R</u> emove are not effective until the next time the user logs on.							
	ОК	Cancel App	у Не	lp			

2. Type the active user's account name into the Select Users, Computers, Service Accounts, or Groups dialog, and click OK.

Select Users, Computers, Service Accounts, or Groups	×
<u>S</u> elect this object type: Users, Service Accounts, or Groups	Object Types
From this location:	Locations
Enter the object names to select ( <u>examples</u> ):	Check Names
<u>A</u> dvanced OK	Cancel

3. When the user has been added to the group, restart the machine and log back in. RDS should now be configured

to collect full timing information for DirectX 12 applications.

#### 8.8 Radeon Developer Service Port numbers

Please note that when running both the Radeon Developer Panel and the Radeon Developer Service on the same system the communication between the two uses pipes, not sockets and ports, so setting the port has no effect. In this scenario, it is possible to set the service to listen on a no-default port, leave the panel on the default port, and connection will work fine.

#### 8.9 Problems caused by the presence of non-AMD GPUs and non-AMD CPUs with integrated graphics

The presence of non-AMD GPU's and CPU's on your system can cause the failure to generate a profile or apps to not run at all.

These problems typically occur with Vulkan apps in systems that have:

- 1. A non-AMD CPU with in integrated non-AMD GPU
- 2. A non-AMD discrete GPU

Vulkan applications, by default, use GPU 0 which usually maps to the integrated GPU, or in some cases, the non-AMD discrete GPU. In both cases Vulkan apps will either fail to run, or RGP profiling will not work (no RGP overlay will be present in these cases).

To avoid these issues:

- 1. Disable any non-AMD integrated GPU's in the device manager
- 2. Disable any non-AMD discrete GPU's in the device manager, and/or physically remove from the system.