



## **PVRTrace**

# **Quick Start Guide for Rooted Android Devices**

Public. This publication contains proprietary information which is subject to change without notice and is supplied 'as is' without warranty of any kind. Redistribution of this document is permitted with acknowledgement of the source.

Filename : PVRTrace.Quick Start Guide for Android Rooted  
Version : PowerVR SDK REL\_17.2@4910709a External Issue  
Issue Date : 30 Oct 2017  
Author : Imagination Technologies Limited

## Contents

<b>1. Introduction</b> .....	<b>3</b>
1.1. Document Overview .....	3
1.2. Software Overview.....	3
1.3. Prerequisites.....	3
<b>2. Installation and Tracing</b> .....	<b>4</b>
2.1. Install PVRHub .....	4
2.2. Install PVRTrace Libraries on Device.....	4
2.3. Launch and Trace Application .....	5
2.4. Retrieve the Trace from the Device.....	6
2.5. Analyse the Trace.....	6
<b>3. Contact Details</b> .....	<b>7</b>

## List of Figures

Figure 1. Installing PVRTrace in PVRHub .....	4
Figure 2. Starting PVRTrace Record .....	5
Figure 3. Tracing an application.....	6

# 1. Introduction

## 1.1. Document Overview

The purpose of this document is to provide an overview of the fundamental steps required for running PVRTrace on rooted Android devices.

## 1.2. Software Overview

PVRTrace is a scene recording and analysis utility. It captures all the API calls made by an OpenGL ES application as it is running and records the data for analysis at a later stage. It consists of two main components identified next:

- **Recording Libraries:** These are shim libraries that are installed on the target device and capture all calls to the target device's native graphics libraries. These calls are captured and written into a PVRT file for reading back by the PVRTrace GUI.
- **PVRTrace GUI:** This is the analysis interface of PVRTrace. The GUI presents the contents of pre-recorded PVRT files in a 'human-readable' format. For more information on the PVRTrace GUI see the "PVRTrace User Manual").

## 1.3. Prerequisites

The prerequisites to being able to use PVRTrace on an Android device are as follows:

- The PowerVR Graphics Tools and SDK needs to be installed.
- The user should have a rooted Android device.

## 2. Installation and Tracing

### 2.1. Install PVRHub

PVRHub is the Android application used to install the PVRTrace Recording Libraries on a device. It can be installed using ADB:

```
adb install /<Path_to_PowerVR_SDK>/PVRHub/Android/PVRHub.apk
```

### 2.2. Install PVRTrace Libraries on Device

The installation process for PVRTrace libraries can be done both manually (`adb push`) or by using PVRHub. This document highlights the PVRHub approach only. For more information about the manual approach, consult the “PVRTrace User Manual”.

On starting PVRHub, select `PVRTrace Installer` from the Home screen (Figure 1).

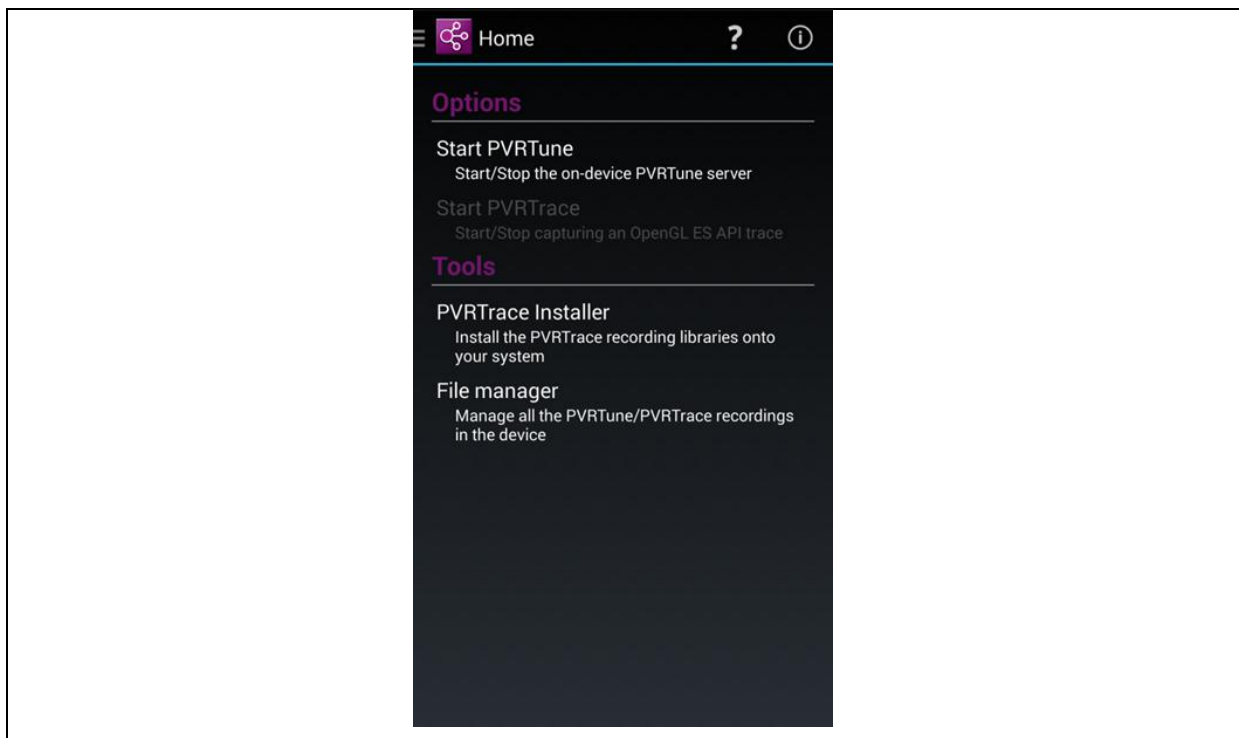


Figure 1. Installing PVRTrace in PVRHub

## 2.3. Launch and Trace Application

To select an application to trace, first choose the option `Start PVRTrace` on the PVRHub interface (Figure 2) and pick the desired application on the next screen.

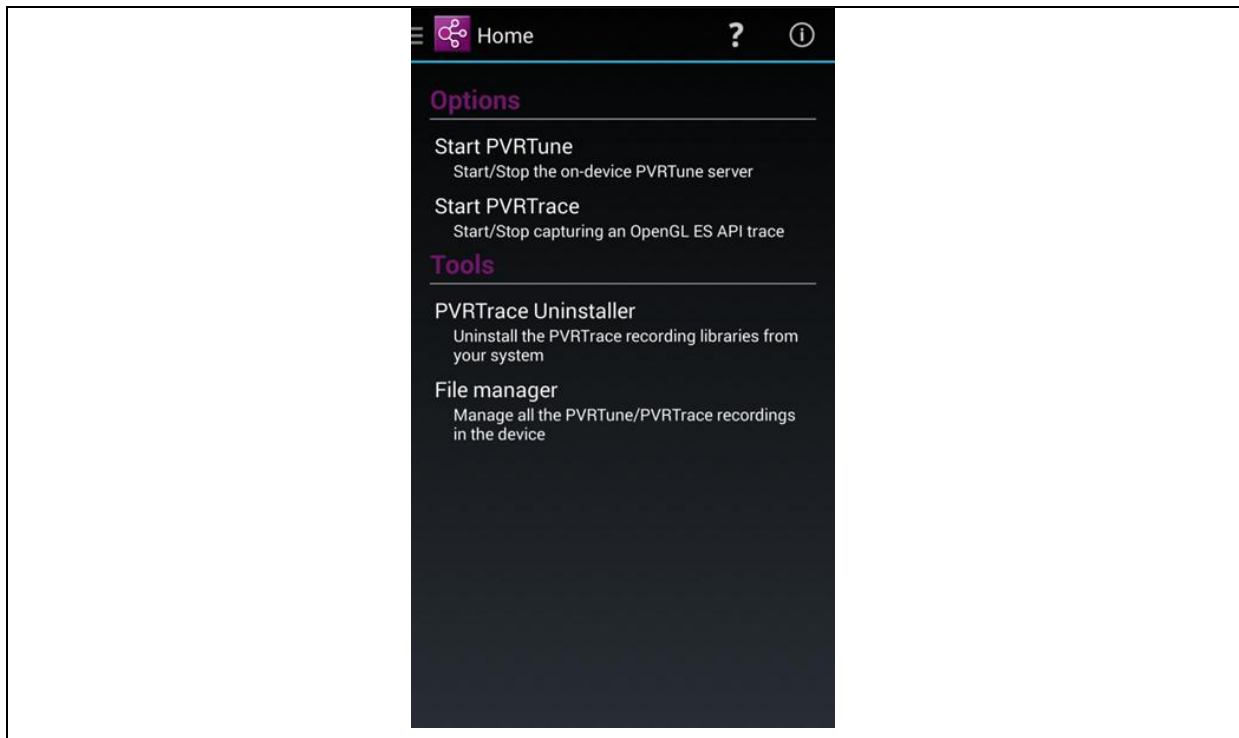
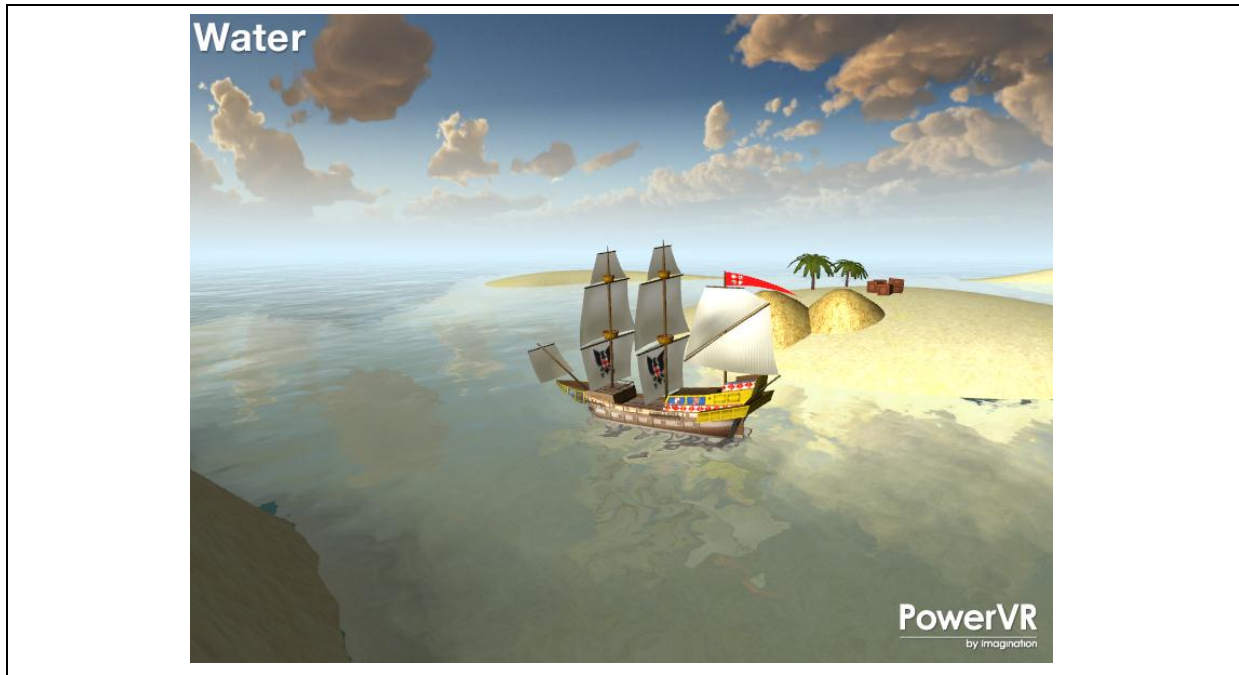


Figure 2. Starting PVRTrace Record

The logcat output should display details similar to:

```
D/PVRHub ( 8728): Enabling PVRTrace...
I/PVRTrace(I) ( 9162): Config file found: /data/data/com.powervr.PVRHub/pvrtraceconfig.json
I/PVRTrace(I) ( 9162): Reading config section: *
I/PVRTrace(I) ( 9162): Reading config section: com.powervr.OGLES2Water
I/PVRTrace( 9162): Setting EglLibraryPath default to '/system/lib64/egl/libEGL_emulation.so'.
I/PVRTrace( 9162): Setting EsLibraryPath default to
'/system/lib64/egl/libGLESv1_CM_emulation.so'.
I/PVRTrace( 9162): Setting Es2LibraryPath default to
'/system/lib64/egl/libGLESv2_emulation.so'.
I/PVRTrace(I) ( 9162): Opening trace file
'/data/data/com.powervr.PVRHub/files/rec/com.powervr.OGLES2Water.pvrtrace' for recording.
I/com.powervr.OGLES2Water( 9162): PVRShell: EGL 1.4 initialized
I/PVRTrace(I) ( 9162): Loading host library '/system/lib64/egl/libGLESv2_emulation.so'
I/PVRTrace(I) ( 9162): Host library '/system/lib64/egl/libGLESv2_emulation.so' loaded
I/PVRTrace(I) ( 9162): Recording... Current frame: 1
I/PVRTrace(I) ( 9162): Recording... Current frame: 2
I/PVRTrace(I) ( 9162): Recording... Current frame: 3
I/PVRTrace(I) ( 9162): Recording... Current frame: 4
I/PVRTrace(I) ( 9162): Recording... Current frame: 5
I/PVRTrace(I) ( 9162): Recording... Current frame: 6
I/PVRTrace(I) ( 9162): Recording... Current frame: 7
```

All API calls are now being traced. It is possible to allow the application to run for as many frames as is considered necessary (Figure 3).



**Figure 3. Tracing an application**

## 2.4. Retrieve the Trace from the Device

The trace can be obtained:

```
adb pull /data/data/com.powervr.PVRHub/files/rec/com.powervr.OGLES2Water.pvrtrace
```

It is also possible to change the name and location of the recording from PVRHub. Notice that writing to the SD card requires the permission:

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

## 2.5. Analyse the Trace

To analyse the trace, run PVRTrace GUI. For more information about PVRTrace GUI, consult the “PVRTrace User Manual”.

### 3. Contact Details

For further support, visit our forum:

<http://forum.imgtec.com>

Or file a ticket in our support system:

<https://pvrsupport.imgtec.com>

To learn more about our PowerVR Graphics SDK and Insider programme, please visit:

<http://www.powervrinsider.com>

For general enquiries, please visit our website:

<http://imgtec.com/corporate/contactus.asp>