



# PVRHub

## User Manual

Copyright © Imagination Technologies Limited. All Rights Reserved.

This publication contains proprietary information which is subject to change without notice and is supplied 'as is' without warranty of any kind. Imagination Technologies, the Imagination logo, PowerVR, MIPS, Meta, Enigma and Codescape are trademarks or registered trademarks of Imagination Technologies Limited. All other logos, products, trademarks and registered trademarks are the property of their respective owners.

Filename : PVRHub.User Manual  
Version : PowerVR SDK REL\_3.3@2958700a External Issue  
Issue Date : 17 Jun 2014  
Author : Imagination Technologies Limited

## Contents

<b>1. Introduction</b>	<b>3</b>
1.1. Document Overview	3
1.2. Software Overview	3
<b>2. Installation</b>	<b>4</b>
2.1. Android	4
2.1.1. PVRHubDaemon on Android	4
2.2. Linux	4
2.2.1. Per Terminal	4
2.2.2. Per User	5
<b>3. Usage</b>	<b>6</b>
3.1. Android	6
3.1.1. PVRHubDaemon for Android	6
3.2. Linux	6
3.2.1. Trace a Single Application	7
<b>4. PVRHub Overview</b>	<b>8</b>
4.1.1. Main Screen	8
4.1.2. API Trace Screen	9
4.1.3. Profile Screen	10
4.1.4. Information Screen	11
<b>5. Related Materials</b>	<b>12</b>
<b>6. Contact Details</b>	<b>13</b>
<b>Appendix A. New Features of PVRHubDaemon</b>	<b>14</b>

## List of Figures

Figure 1. PVRHub Main Screen	8
Figure 2. API Trace Screen	9
Figure 3. Profile Screen	10
Figure 4. Information Screen	11

## List of Tables

Table 1. Description of options on the PVRHub main screen	8
Table 2. Description of options on the API Trace screen	10
Table 3. Description of options on the Profile screen	11

# 1. Introduction

## 1.1. Document Overview

The purpose of this document is to serve as a complete user manual for the PVRHub Android Application and the PVRHub Linux scripts. It includes installation instructions, a guide to the functionality of the two applications, a complete listing of all interface options and preferences for each of the applications, as well as instructions on their usage.

## 1.2. Software Overview

PVRHub is an application for Android and Linux that allows for the control and setup of the PVRTrace Recording Libraries. This helps the user avoid manual setup and control. PVRHub also includes the PVRPerfServer functionality to transmit performance data to an instance of PVRTune. There are two main components to PVRHub:

- **Recording Libraries:** these are shim libraries that are installed on a platform and capture all calls to that platform's native graphics libraries. These calls are captured and written into a PVRT file for reading back by the PVRTrace GUI.
- **PVRPerfServer:** this is an application which runs on the target device and reads counters and registers from the hardware and either writes it into a PVRT file, or transmits that data to PVRTune over a network.

*Note: On devices that have not been rooted, users must package the PVRTrace recorder libraries into their APK.*

*Note: For more information about unrooted instructions, see the "PVRTrace Quick Start Guide for Unrooted Android Devices" distributed with the PowerVR Graphics SDK.*

## 2. Installation

### 2.1. Android

To install PVRHub, run `'adb install PVRHub.apk'` on the local machine.

*Note: If the root access is not granted on the target device, PVRHubDaemon must be running before the PVRHub application can be launched.*

#### 2.1.1. PVRHubDaemon on Android

To run PVRHub on a target device where an application cannot gain root access (but the command-line can), it is necessary to install and run PVRHubDaemon. To install PVRHubDaemon:

1. Install PVRHub.apk using the instructions above.
2. Run PVRHub.apk. This unpacks the PVRHubDaemon binary inside the application directory.
3. Close the PVRHub application, ensuring the application is not running in the background.
4. Get a root adb connection (`'adb root'`) or shell (`'adb shell'`). Run as `'su'`.

*Note: Step 4 is very important!*

5. Change the PVRHubDaemon permissions using either adb:

```
adb shell chmod 777 /data/data/com.powervr.PVRHub/bin/PVRHubDaemon
```

Or the shell:

```
# chmod 777 /data/data/com.powervr.PVRHub/bin/PVRHubDaemon
```

*Note: To subsequently use all recording and profiling features, PVRHubDaemon must be running before PVRHub is launched.*

*Note: For PVRHubDaemon recovery and system verification features, see Appendix A.*

### 2.2. Linux

With the SDK package successfully installed PVRHub can be installed in one of two ways, per terminal, or per user.

#### 2.2.1. Per Terminal

To install PVRHub per terminal, complete the following instructions:

1. Open a terminal and enter the following:

```
cd <PVRHubFolder>/<Platform>/
```

Where `'<PVRHubFolder>'` is the folder in which PVRHub is installed and `'<Platform>'` is the Linux platform PVRHub is to be ran on (i.e., `Linux_ARMv7`, `Linux_x86_32`, etc.).

2. In the same terminal enter the following:

```
source Scripts/envsetup.sh
```

3. Repeat this process for each terminal in which PVRHub is to be used.

### 2.2.2. Per User

To install PVRHub per user, enter the following into the ``.bashrc`` file of the desired user:

```
if [ -f /Path/to/PVRHub/<Platform>/Scripts/envsetup.sh ]; then
  cd /Path/to/PVRHub/<Platform>/
  source scripts/envsetup.sh
  cd
fi
```

Future interactive shells opened by the user are now able to use PVRHub.

## 3. Usage

### 3.1. Android

To record using PVRHub on Android:

1. Run PVRHub. If PVRHub is being run for the first time, click 'Install'.  
*Note: If the device does not have root access, be sure to install and run PVRHubDaemon before running PVRHub. For more information, see Section 2.1.1.*
2. If using network recording, click on 'Options', click on 'Network' and click on 'Network Wait' if desired.
3. On the 'Options' screen select the desired start and end frame and set the output file.
4. On the 'Setup' page click 'Start Tracing' to trace all applications launched after the button is clicked, then launch the desired applications that need to be traced.
5. If tracing a specific application click 'Trace an Application'. This launches the desired application and closes it once the end frame is reached.
6. Run the PVRTrace GUI and either open the output file, or follow the steps for network recording as previously mentioned.

#### 3.1.1. PVRHubDaemon for Android

The PVRHubDaemon for Android allows the use of the PVRHub GUI on target devices which do not have root access. To run the PVRHubDaemon:

1. Execute PVRHubDaemon from either adb:

```
adb shell ./data/data/com.powervr.PVRHub/bin/PVRHubDaemon
```

Or from the shell:

```
# ./data/data/com.powervr.PVRHub/bin/PVRHubDaemon
```

2. Launch the PVRHub application.

*Note: For PVRHubDaemon recovery and system verification features, see Appendix A.*

### 3.2. Linux

PVRHub consists of a number of scripts that, once installed as described in Section 2.2, can be used to profile or trace an application. These scripts are:

- `pvr_trace`  
Used to trace a single application.
- `pvr_profile`  
Used to profile a single application.
- `start_tracing.sh`  
Used to trace multiple applications.
- `stop_tracing.sh`  
Used to stop tracing after 'start\_tracing.sh' is used.

### 3.2.1. Trace a Single Application

To trace a single application complete the following instructions:

1. Open a terminal.
2. Enter the following:

```
pvr_trace <application>
```

Where '`<application>`' is the path to the application to be traced.

3. Open the output trace file in PVRTrace.

#### Profile a Single Application

To profile a single application complete the following instructions:

- a. Open a terminal.
- b. Enter the following:

```
pvr_profile <application>
```

Where '`<application>`' is the path to the application to be profiled.

- c. Open PVRTune and connect to the device running the application to be profiled.

#### Trace Multiple Applications

Any application launched between calling '`start_trace.sh`' and '`stop_trace.sh`' is traced.

To trace multiple applications:

- a. Open a terminal.
- b. Enter the following:

```
source start_tracing.sh
```

- c. Launch an application.
- d. Exit the application.
- e. Open the output file in PVRTrace.
- f. Repeat steps 2-4 of "Trace Multiple Applications" for desired applications.
- g. Enter the following:

```
source stop_tracing.sh
```

## 4. PVRHub Overview

This section provides an overview of PVRHub for Android.

### 4.1.1. Main Screen

Figure 1 depicts the PVRHub main screen. The various options appearing in the interface are described in Table 1.

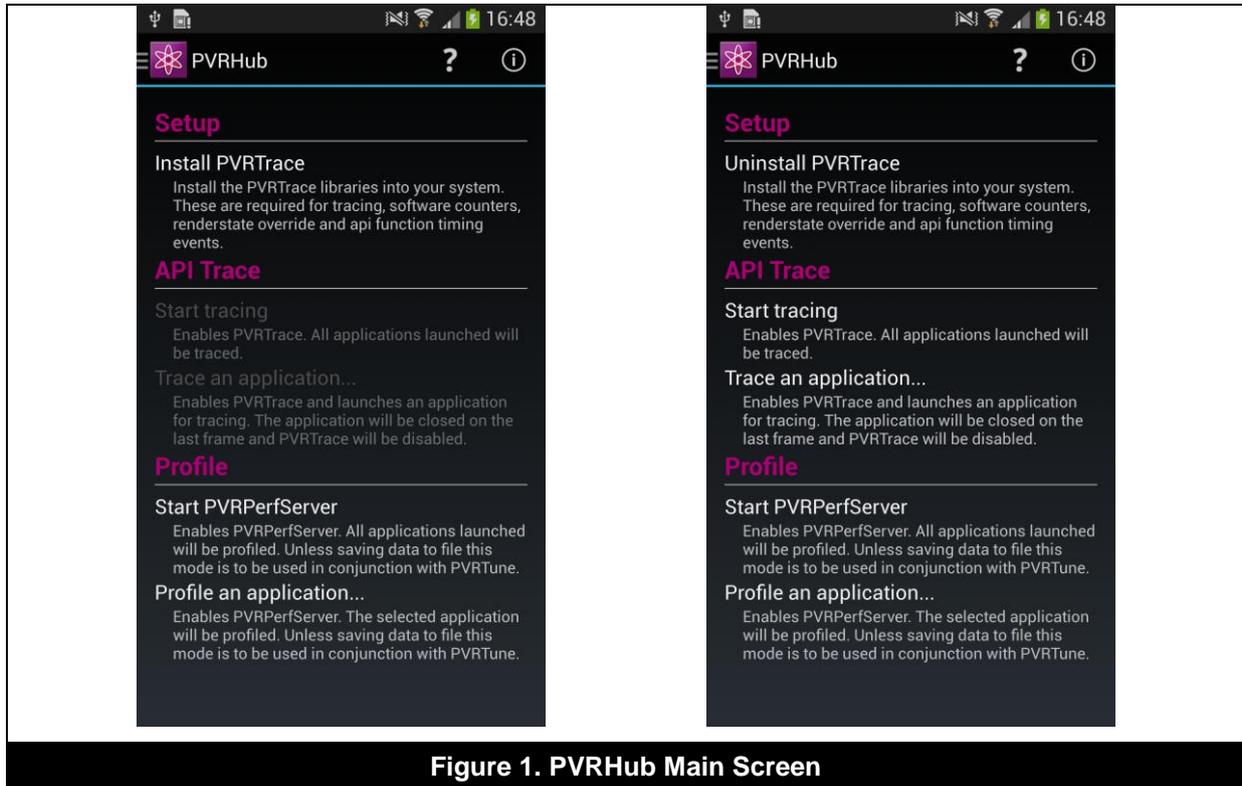


Figure 1. PVRHub Main Screen

Table 1. Description of options on the PVRHub main screen

Option	Description
Setup: Install PVRTrace	Installs the PVRTrace Recording Libraries.
Setup: Uninstall PVRTrace	Removes the PVRTrace Recording Libraries.
API Trace: Start tracing	Sets the PVRTrace Recording Libraries as the currently active graphics libraries. From the point of activation onwards any application launched is traced. <i>Note: When tracing, the PVRHub screen is highlighted with a red border.</i>
API Trace: Stop tracing	Restores the original graphics libraries, disabling recording for any application launched after 'Start tracing' is selected.
API Trace: Trace an application...	Sets the PVRTrace Recording Libraries as the current active graphics libraries for a specific application, launches that application, then exits it once the end frame set in the API Trace screen is hit. Only the selected application is traced.

Option	Description
Profile: Start PVRPerfServer	Starts PVRPerfServer. In addition, if any of 'Enable Software Counters', 'Enable Renderstate Override', or 'API Function Timing Events' are enabled this option sets the PVRTrace Recording Libraries as the currently active graphics libraries and switch them to profiling mode. <i>Note: When profiling, the PVRHub screen is highlighted with a green border.</i>
Profile: Stop PVRPerfServer	Stops PVRPerfServer. In addition, if any of 'Enable Software Counters', 'Enable Renderstate Override', or 'API Function Timing Events' are enabled this option restores the original graphics libraries. This option is primarily for use when profiling whole systems or multiple applications.
Profile: Profile an application	Starts PVRPerfServer (as in 'Start PVRPerfServer'), then launches the desired application. <b>IMPORTANT: Recording is stopped while this option is active.</b>

### 4.1.2. API Trace Screen

Figure 2 depicts the API Trace screen. The various options appearing in the interface are described in Table 2.

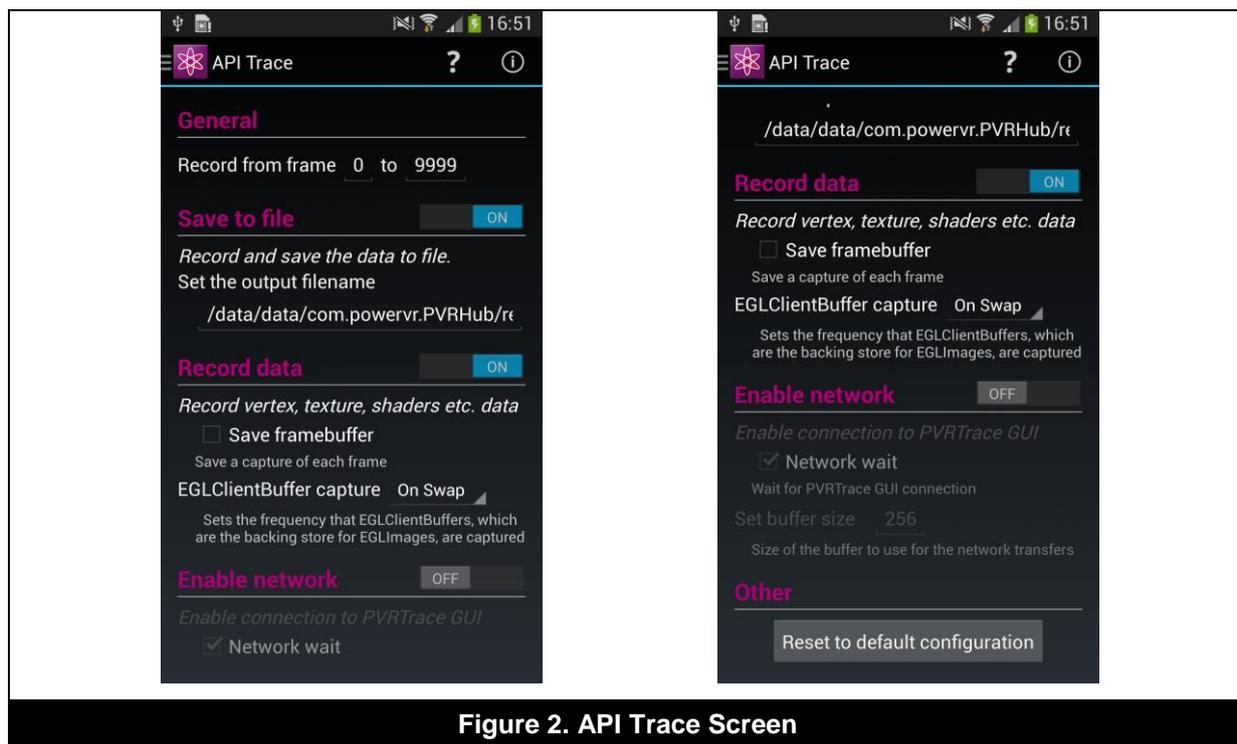


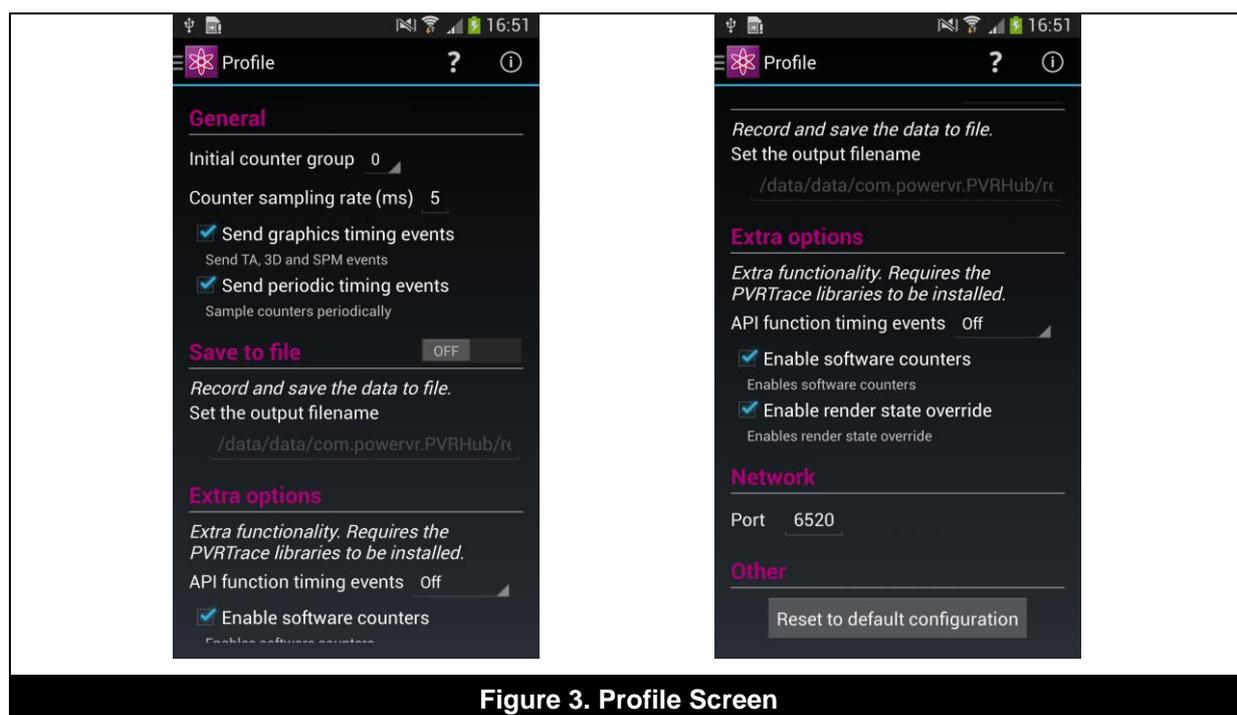
Figure 2. API Trace Screen

**Table 2. Description of options on the API Trace screen**

Option	Description
General	Allows selection of the first and last frames of a traced application to record.
Save to file	This option, when toggled on, saves the trace data to a file on the device.
Save to file: Set the output filename	This states the location and file name of the output trace file.
Record data	With this option toggled on, the PVRTrace Recording Libraries record all buffer contents, all render state changes, every shader, every texture, etc. This option allows the Image Analysis Window of the PVRTrace GUI to be used.
Record data: Save framebuffer	With this option ticked, the PVRTrace Recording Libraries save the contents of the framebuffer at the end of every frame.
Enable network	This option, when toggled on, states whether the PVRTrace Analysis GUI records the trace directly via the network.
Enable network: Network wait	With this option ticked, the PVRTrace Recording Libraries cause the application to wait in the first OpenGL call until the PVRTrace Analysis GUI signals to continue.
Enable network: Set buffer size	Sets the size of the buffer that is filled before sending data to the PVRTrace Analysis GUI when using Network recording. In general the default option is sufficient, but in some cases this option may need to be changed depending on network settings.
Other: Reset to default configuration	Selecting this option resets API trace settings to default.

### 4.1.3. Profile Screen

Figure 2 depicts the API Trace screen. The various options appearing in the interface are described in Table 2.

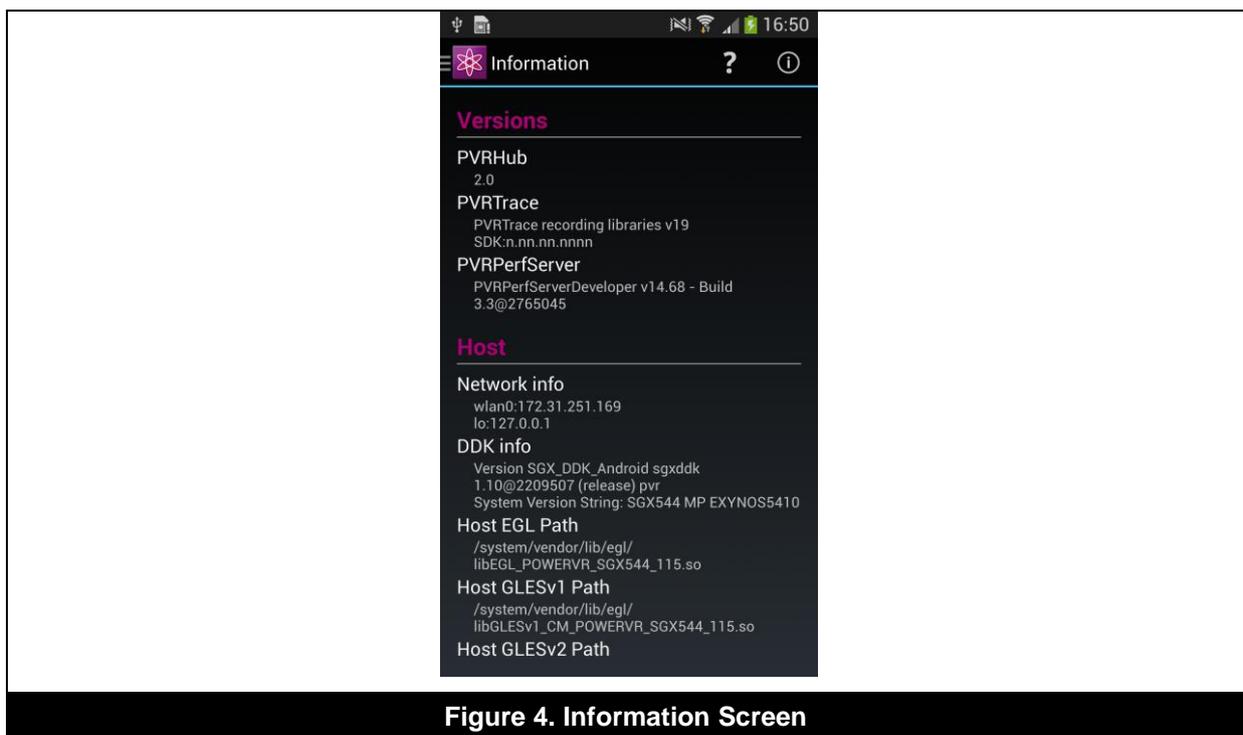
**Figure 3. Profile Screen**

**Table 3. Description of options on the Profile screen**

Option	Description
General: Initial counter group	The initial active counter group when PVRPerfServer starts.
General: Counter sampling rate (ms)	The numbers of milliseconds (ms) to wait between updating counter values.
General: Send graphics timing events	Enables or disables the sending of timing data (i.e., TA/3D etc.).
General: Send periodic timing events	Enables or disables the sending of counter updates.
Save to file	This option, when toggled on, saves the profiling data to a file.
Save to file: Set the output filename	This states the location and file name of the output PVRTune file.
Extra options: API function timing events	Enables or disables the sending of timing data describing the API calls made to the driver. <i>WARNING: Setting this to 'Verbose' may affect performance.</i>
Extra options: Enable software counters	Enables or disables the sending of software counters and additional counter set gained from the API through the use of the PVRTrace Recording Libraries.
Extra options: Enable render state override	Enables or disables the use of RenderState override in PVRTune.
Network: Port	The port that PVRPerfServer binds to for connecting to PVRTune.
Other: Reset to default configuration	Selecting this option resets profile settings to default.

#### 4.1.4. Information Screen

Figure 4 depicts the Information screen, which typically displays versioning and host details.



**Figure 4. Information Screen**

## 5. Related Materials

### Software

- PVRHub
- PVRTrace
- PVRTune

### Documentation

- PVRTune User Manual
- PVRTrace User Manual
- PowerVR Performance Recommendations

## 6. 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>

## Appendix A. New Features of PVRHubDaemon

The following are some new features of the PVRHubDaemon:

- **System report generator:** every time the PVRHubDaemon runs, it writes a report in `'/data/local/tmp/PVRHubDaemon_report.txt'` which contains all the relevant information of the Android system. By executing the daemon with `'(-g)'` it will generate a report and terminate (i.e., `'/data/data/com.powervr/PVRHub/bin/PVRHubDaemon -g'`).
- **System recovery:** the PVRHubDaemon can also be used to restore the system in case the trace libraries were wrongly installed. By executing the daemon with `'(-r)'` it will restore the system and terminate (i.e., `'/data/data/com.powervr/PVRHub/bin/PVRHubDaemon -r'`).
- **Backup generation:** the first time that the PVRHubDaemon runs on the system, it backs up the PowerVR DDK inside `'/local/tmp/PowerVR_DDK_backup/'`.
- **Driver name override:** by default, PVRHubDaemon will only look for driver binaries with `'_POWERVR'` in their file name. The optional file `'/data/local/tmp/ddk_suffix.txt'` allows users to override the identifier PVRHubDaemon will use to select a graphics driver. This feature is designed for developers targeting development platforms, where there may be a number of graphics driver builds on the target with different names.

Imagination Technologies, the Imagination Technologies logo, AMA, Codescape, Enigma, IMGworks, I2P, PowerVR, PURE, PURE Digital, MeOS, Meta, MBX, MTX, PDP, SGX, UCC, USSE, VXD and VXE are trademarks or registered trademarks of Imagination Technologies Limited. All other logos, products, trademarks and registered trademarks are the property of their respective owners.