

# XgxPerf Toolkit

Franline Jose

## ABSTRACT

This application report shows how to use the XgxPerf toolkit to build and benchmark a Qt™ based application. It is assumed that you are familiar with installing and writing applications based on Qt.

Project collateral and source code discussed in this application report can be downloaded from the following URL: <http://www-s.ti.com/sc/techlit/sprabf4.zip>.

## 1 Introduction

[Section 5](#) includes information on getting started on Qt and the links to download XgxPerf.

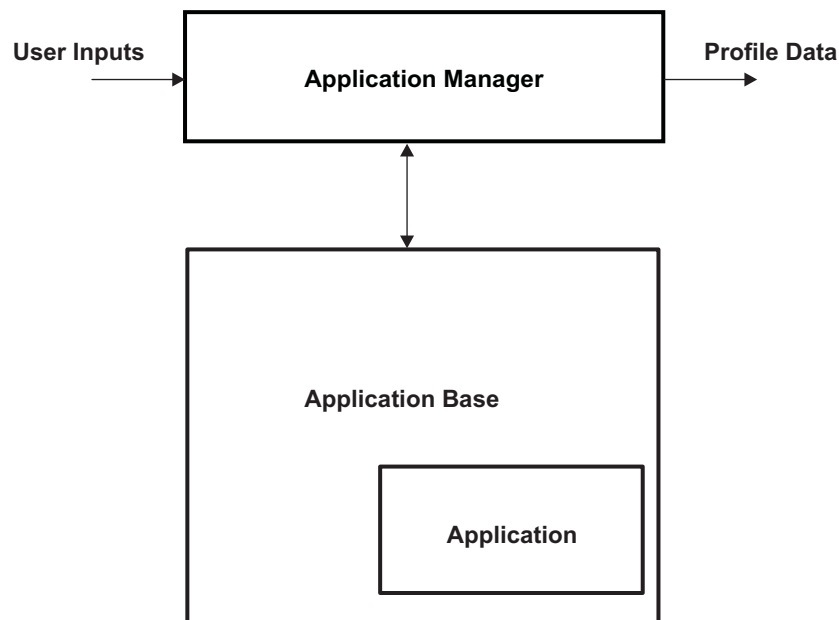
A sample application is also provided with this document.

## 2 XgxPerf

XgxPerf is a light-weight toolkit that you can use to build modular Qt applications that can be profiled. XgxPerf has two main components: the *Application Manager* and the *Application Base*.

The main uses of this toolkit are to profile, benchmark, and provide an example for optimized usage of Qt. It also adds tools like *livemem* used to capture screens at any point in time when the application is being executed. Optimized SVG classes for animation are provided.

XgxPerf is dependant on SVG and Webkit; Qt needs to be built with these options.



**Figure 1. High-Level View of XgxPerf**

## 2.1 Application Manager

Application manager acts as an interface between you and the actual Qt-based application. It has the option to load the actual Qt application and also to accept the relevant parameters via command line.

## 2.2 Application Base

Application base is a class used to profile the application's CPU bandwidth requirements. This class can be used in providing the startup time of the application, as well as the animation time.

applicationmanager\common\applicationbase.h provides the application base class. The base class provides Init, Process, GetAppName, GetAppIcon, GetSwitchingTime and calls.

- Init: initialize the widgets
- Process: timer-based process calls  
The application manager configures the timer based on the fps set.
- GetAppName: returns the application name (deprecated)
- GetAppIcon: returns the icon for application (deprecated)
- GetSwitchingTime: returns the time interval for whichever process is called (deprecated)

## 3 How to Create an Application With XgxPerf

### 3.1 XgxPerf Package

The toolkit comes with an application manager package with the following components:

```
<INSTALL_PATH>\xgxperf\trunk\Packages\OMAP3_Graphics_SDK\xgxperf\xgxperf\applicationmanager
```

```
--common
  --applicationbase.h
--qt_e
  --applicationmanager.cpp
  --applicationmanager.h
  --main.cpp
  --qt_e.pro
```

### 3.2 Creating a New Application

The application is built as a library. The application manager loads the application library into the target and waits for the execution to complete.

To create a new application:

1. Create a new cpp file say helloworld.cpp, create a new class from applicationbase.

```
class HelloWorld: public ApplicationBase
{
public:
    HelloWorld(QWidget* parent, QRectF inSceneRect, QGraphicsScene *scene, void* data);
    void Init();
    void Process();
    QString GetAppName();
    QIcon GetAppIcon();
    int GetSwitchingTime();
private:
    TestTargetParamsStruct* m_widgetTestInStruct;
    QGraphicsTextItem *pHelloVal;
};
extern "C"
HelloWrlld *
```

```

createNewApp(QWidget* parent,
             QRectF inSceneRect,
             QGraphicsScene *scene,
             void* data
             )
{
    return new HelloWrld(parent, inSceneRect, scene, data);
}

HelloWrld::HelloWrld(QWidget* parent, QRectF inSceneRect,
                    QGraphicsScene *scene, void* data):
ApplicationBase(parent, inSceneRect, scene, data)
{
    //Store the test data locally
    m_widgetTestInStruct = (TestTargetParamsStruct*)data;
}

QString HelloWrld::GetAppName(){return QString("HelloWrld");};
int HelloWrld::GetSwitchingTime()
{
    return (1000/m_widgetTestInStruct->rate_screen_refresh_fps);
};

```

## 2. Add the Init and process functions.

```

void HelloWrld::Init()
{
    pHelloVal = m_scene->addText("Hello World");
    pHelloVal->moveBy(200,200);
}

void HelloWrld::Process()
{
    pHelloVal->moveBy(5,5);
    //Indicate that the processing is complete
    SetProcessCallComplete();
}

```

## 3. Create a .pro file with the following:

```

TEMPLATE = lib
SOURCES = helloworld.cpp
HEADERS += ../applicationmanager/common/applicationbase.h

```

## 4. Build the application as follows:

```

qmake qt_e.pro
    This will create a Makefile based on the Qt configuration.
make

```

The libraries need to be installed in the `/usr/lib` folder. If you want to install the libraries in a different path, use `LD_LIBRARY_PATH` to set the library path.

### 3.3 Building the Application Manager

The application manager can be copied to the target folder where the actual application resides. To build the application manager for a particular target:

1. Go to the `applicationmanager\qt_e` folder
2. `qmake qt_e.pro`
3. `make`

The `qt_e` binary created at this step is the applicationmanager.

An example folder structure will look like:

```
--applicationmanager
--application_name
  --application_name.cpp
  --application_name.pro
```

### 3.4 Executing the Application

To execute the application:

```
#qt_e -qws <params>
```

params = Cookie, numWdgt, fx, transparency, texture, fontsize, fps, prebuiltID, wdgtsize, numFrames, appstring

**example:** `/qt_e -qws 123 3 0 0 0 0 10 99 512 100 "helloworld"`

#### Mandatory params:

- Cookie: number to be appended to the output file – example: 0,1,2
- fps: number of frames per second – example: 1,2
- numFrames: number of frames to run – example: 100,200
- appstring: name of the application library to run

#### Optional params:

- numWdgt – number of widgets in the application
- fx=0,1: enable/disable shadow to widgets when fx=1
- transparency: enable/disable transparency
- texture = 0,1,2: add a background texture 0-no; 1-PNG; 2-SVG
- fontsize – set font size: example: 8,16
- fontsize – set font size: example: 8,16 – set to 99 while executing custom application.
  - 0 – Industrial Automation blocks (Turbines, Boilers, Pipes, Text Info classes)
  - 1 – Text + Pictures (PNG/VG/Colors)
  - 2 – Involves GLWidget, needs powervr (not on HawkBoard)
  - 3 – Webkit Browser
  - 4 – Automotive Tacho
  - 5 – Medical ECG Monitor
  - 5 – Medical ECG Monitor
  - 7 – Sewing Machine UI
- wdgtsize – size of the widget: example:100,256,512

#### livemem

If a screen capture of the application is required, copy `get_screen_dump.sh` file from `\xgxperf\trunk\Packages\OMAP3_Graphics_SDK\xgxperf\server` to the folder where the application is installed.

## 4 Results and Interpreting

Once the Process call returns, the application manager saves the results in an xml file.

Table 1 shows an example result of the ECG monitor application on the OMAP3530 device.

**Table 1. XgxPerf\_Profile\_Test\_Inputs**

Preconfigured_Test_Name	ecgmonitor
Num_Widgets	3
Effects	0
Transparency	0
Texture_type	0
Font_Size	0
Screen_Refresh_Rate	10
Size_of_widgets	30
Number_of_frames_run	300

Table 2 shows that it took 3 mS to start the application, once it was initiated. Prior to that, it was taking about 35 mS per frame with an average CPU load of 29% on OMAP3530.

**Table 2. XgxPerf\_Profile\_Test\_Output\_Profile\_Parameters**

Screen Init Time (mS)	3
Screen Update Time/Frame (mS)	35
Average CPU Load	29

## 5 References

- Building Qt, [http://processors.wiki.ti.com/index.php/Building\\_Qt](http://processors.wiki.ti.com/index.php/Building_Qt)
- Download XgxPerf:
  - XgxPerf project page: <https://gforge.ti.com/gf/project/gleslayer/>
  - svn checkout: [http://processors.wiki.ti.com/index.php/SVN\\_Access](http://processors.wiki.ti.com/index.php/SVN_Access)
- XgxPerf Support:
  - Linux support forum at <http://e2e.ti.com/forums/>

## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

<b>Products</b>		<b>Applications</b>	
Amplifiers	<a href="http://amplifier.ti.com">amplifier.ti.com</a>	Audio	<a href="http://www.ti.com/audio">www.ti.com/audio</a>
Data Converters	<a href="http://dataconverter.ti.com">dataconverter.ti.com</a>	Automotive	<a href="http://www.ti.com/automotive">www.ti.com/automotive</a>
DLP® Products	<a href="http://www.dlp.com">www.dlp.com</a>	Communications and Telecom	<a href="http://www.ti.com/communications">www.ti.com/communications</a>
DSP	<a href="http://dsp.ti.com">dsp.ti.com</a>	Computers and Peripherals	<a href="http://www.ti.com/computers">www.ti.com/computers</a>
Clocks and Timers	<a href="http://www.ti.com/clocks">www.ti.com/clocks</a>	Consumer Electronics	<a href="http://www.ti.com/consumer-apps">www.ti.com/consumer-apps</a>
Interface	<a href="http://interface.ti.com">interface.ti.com</a>	Energy	<a href="http://www.ti.com/energy">www.ti.com/energy</a>
Logic	<a href="http://logic.ti.com">logic.ti.com</a>	Industrial	<a href="http://www.ti.com/industrial">www.ti.com/industrial</a>
Power Mgmt	<a href="http://power.ti.com">power.ti.com</a>	Medical	<a href="http://www.ti.com/medical">www.ti.com/medical</a>
Microcontrollers	<a href="http://microcontroller.ti.com">microcontroller.ti.com</a>	Security	<a href="http://www.ti.com/security">www.ti.com/security</a>
RFID	<a href="http://www.ti-rfid.com">www.ti-rfid.com</a>	Space, Avionics & Defense	<a href="http://www.ti.com/space-avionics-defense">www.ti.com/space-avionics-defense</a>
RF/IF and ZigBee® Solutions	<a href="http://www.ti.com/lprf">www.ti.com/lprf</a>	Video and Imaging	<a href="http://www.ti.com/video">www.ti.com/video</a>
		Wireless	<a href="http://www.ti.com/wireless-apps">www.ti.com/wireless-apps</a>

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2010, Texas Instruments Incorporated