

# pxCore –Portable Framebuffer and Windowing Library

Copyright © 2007 John Robinson

## Overview

pxCore is a small lightweight library that provides a portable framebuffer and windowing abstraction for C++. It is designed for folks that want to write their own rasterization routines giving access in a portable way to a framebuffer that can be used easily on a number of operating systems. In addition it gives a simple windowing abstraction that allows you to write applications on a number of operating systems.

This library is intended to hide the complexity of writing rasterization code on multiple platforms and allows you to easily write the following sorts of things in a portable way.

- 2d and 3d rasterizers
- Transition Libraries
- Filter Routines
- Image Processing

In its design a few principles were followed:

- Be Small – A simple windowed application can be built (on Windows) in as little as 8k.
- Be Simple – The framebuffer abstraction supports 32bpp framebuffers and is intended to be minimal yet complete.
- Don't tightly couple the framebuffer and windowing functionality.- Some other framebuffer libraries (PixelToaster for one) don't separate out the framebuffer abstraction from the windowing abstraction. By loosely coupling the two abstractions this library becomes much more valuable; as the framebuffer functionality can be used and integrated with other windowing toolkits easily thereby making YOUR code more reusable.
- Platform native surface construction
- Policy free resizing support – No policy is baked into the window resizing support so that applications completely control their own resizing behavior.
- Portable Keyboard, Mouse and Window events
- Support for portable performance timers
- Basic Animation Support – Support for a basic animation timer event (frames per second) is built into the windowing abstraction making it easy to write applications that animate their contents.

## Error Handling

For all functions that can fail their signature is defined to return an error code typedef called `pxError`. No exceptions are generated or used within the pxCore library. `pxError` is an unsigned long integer and a value of zero indicates that the operation has succeeded.

A non-zero value indicates that the operation has failed. Tests for success should check to see if the result is zero. Tests for failure should check that the result is not zero.

## **Framebuffer**

The pxCore library supports 32bpp framebuffers. The framebuffer support is in the form of a layered abstraction in two parts. The first is a class called pxBuffer. The pxBuffer class is a framebuffer descriptor. It does not allocate or contain the memory for a framebuffer itself but rather describes a region of memory that is to be used as a framebuffer. This descriptor approach allows procedural code utilizing the pxBuffer abstraction to operate on framebuffers that aren't directly managed by pxCore. The pxBuffer class has methods that can be used to describe the layout of the actual framebuffer in memory. These methods include describing the width and height of the framebuffer, the base memory address of the framebuffer and whether the framebuffer's scanlines are laid out upside down or not.

Another class called pxOffscreen (that derives from pxBuffer) is a convenience class for allocating and managing a concrete framebuffer. A portable type definition pxSurfaceNative is the bridge that allows framebuffers to be drawn into a pxWindow in a portable manner. The concrete type that underlies this is platform specific. For example on Windows pxSurfaceNative is a HDC, the concrete definition of this type can be found by examining the corresponding pxOffscreenNative.h header for the appropriate platform. If you wish to blit a pxOffscreen onto a native window surface you should examine this type. Since pxOffscreen is a subclass of pxBuffer, the methods of pxBuffer can be used to extract information about the allocated framebuffer.

The init method of pxOffscreen allocates the requested framebuffer within the pxOffscreen class.

## **Windows**

The pxCore windowing functionality is designed to be simple and lightweight providing a framework that can be used to build basic windowed applications. If the capabilities of the windowing functionality are not suitable to your application the design of the pxCore library allows you to easily integrate with just about any other windowing library and still use the framebuffer abstraction via the pxSurfaceNative portable type.

A "hello world"-type sample is included within the pxCore distribution, called SimpleExample. This example demonstrates the basics of rendering content into a framebuffer and using the windowing functionality to make a window that displays the framebuffer's contents and handles resizing.

The windowing abstraction is provided by the base class pxWindow. In order to create a custom window for your application this class can be subclassed. The event model is based on subclassing pxWindow and overriding the appropriate virtual methods that are used to channel events to the derived class. Virtual methods for receiving keyboard, mouse and windows events are provided. The pxWindow class also supports an event that makes it easy to do simple animated applications. A method called setAnimationFPS

is provided that will cause a method called `onAnimationTimer` to be called the specified number of times per second. *Note: The accuracy of this callback is bound by the underlying platform implementation.*

The `init` method in `pxWindow` actually creates the window. The window is always created hidden. You can use the `setVisibility` to show the window after the `init` method has been successfully called.

## **EventLoop**

Most native windowing platforms utilize an event loop which handles the dequeuing and dispatching of application and windowing events. A portable class called `pxEventLoop` provides an abstraction for creating and “running” such an event loop. `pxEventLoop` has two methods, `run` and `exit`. The `run` method runs the eventloop and does not return until the `exit` method is called. The `exit` method is typically called within a window’s eventhandler.

## **Timers**

`pxCore` also provides a portable set of functions that can be used to measure elapsed time. The functions `pxSeconds`, `pxMilliseconds` and `pxMicroseconds` can be used to measure elapsed time between two different readings with the same given function. A simple example of this is given in this `pxCore` distribution. In addition, a portable function for sleeping a given number of milliseconds is provided.

## **Application Entry Point**

Different platforms may have different application entry points. On Windows it is `WinMain` on Linux and OSX it is `main`. `pxCore` offers a portable alternative called `pxMain` that can be utilized as a portable entry point that is called on all platforms. The entrypoint currently takes no arguments and returns nothing. *Note: Support for command-line arguments may be added at a future time.*

## **License**

`pxCore`: Portable Framebuffer and Windowing Library  
Copyright © 2007 John Robinson

This software is provided ‘as-is’, without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation should be made.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source distribution.

John Robinson

[pxCore@liquidthought.com](mailto:pxCore@liquidthought.com)

<http://www.liquidthought.com>