







Introduced 2 Years Ago



2 Million Downloads

225,000

Applications in the App Store

5 Billion

Applications Downloaded

100 Million

iOS Devices By End of June





Fast incremental uploads

Simulate any OS version

LLVM compiler for iPhone


App archiving

Validation

Streamlined store submission

Easy ad hoc app sharing



 **iPhone Provisioning Portal login**

Username:

Password:



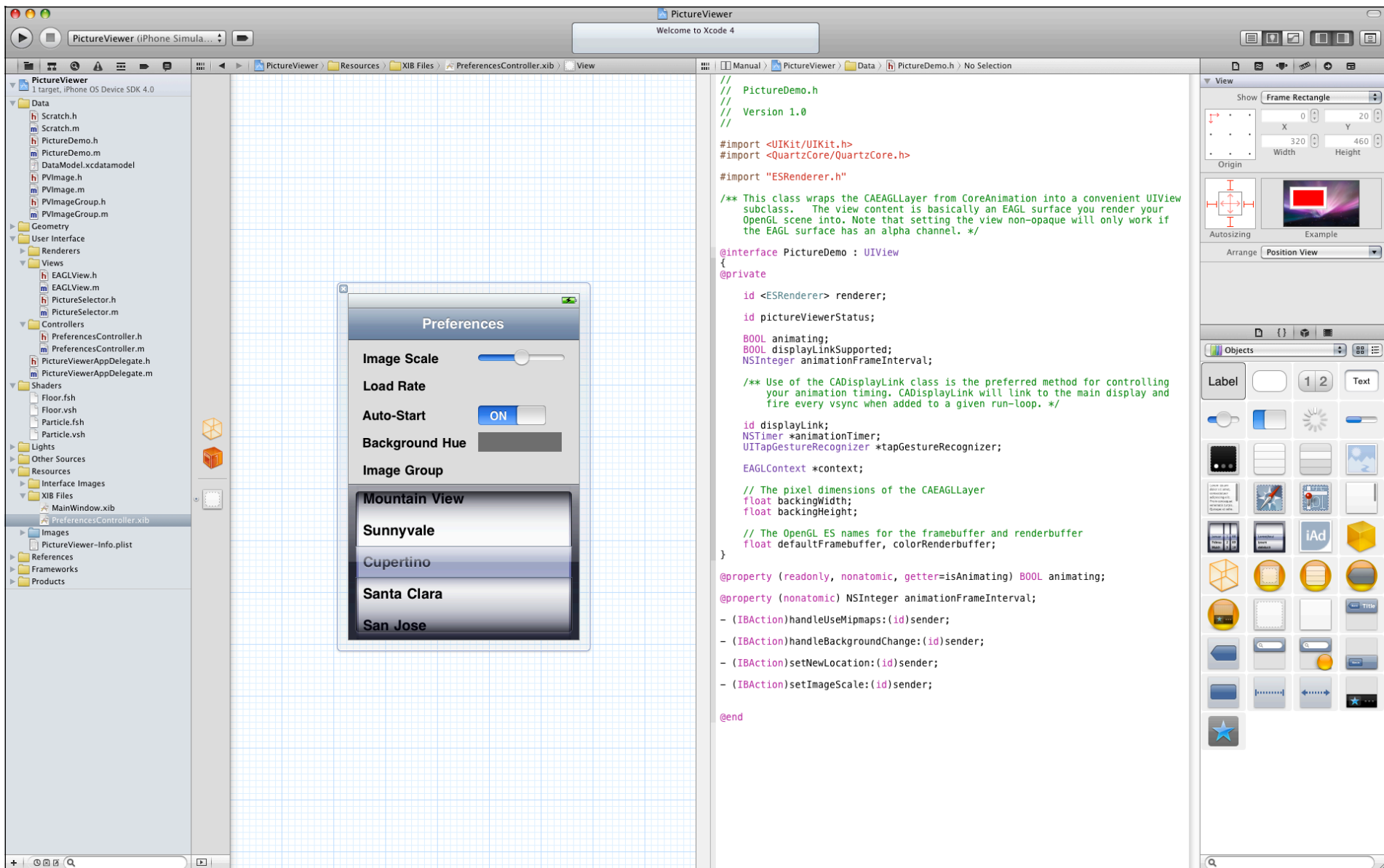
Instant Setup

Demo

Todd Fernandez



Xcode 4



```

//
//  PictureDemo.h
//
//  Version 1.0
//

#import <UIKit/UIKit.h>
#import <QuartzCore/QuartzCore.h>

#import "ESRenderer.h"

/** This class wraps the CAEAGLLayer from CoreAnimation into a convenient UIView subclass. The view content is basically an EAGL surface you render your OpenGL scene into. Note that setting the view non-opaque will only work if the EAGL surface has an alpha channel. */

@interface PictureDemo : UIView
{
@private
    id <ESRenderer> renderer;
    id pictureViewerStatus;

    BOOL animating;
    BOOL displayLinkSupported;
    NSInteger animationFrameInterval;

    /** Use of the CADisplayLink class is the preferred method for controlling your animation timing. CADisplayLink will link to the main display and fire every vsync when added to a given run-loop. */

    id displayLink;
    NSTimer *animationTimer;
    UITapGestureRecognizer *tapGestureRecognizer;

    EAGLContext *context;

    /** The pixel dimensions of the CAEAGLLayer
    float backingWidth;
    float backingHeight;

    /** The OpenGL ES names for the framebuffer and renderbuffer
    float defaultFramebuffer, colorRenderbuffer;
}

@property (readonly, nonatomic, getter=isAnimating) BOOL animating;
@property (nonatomic) NSInteger animationFrameInterval;
- (IBAction)handleUseMipmaps:(id)sender;
- (IBAction)handleBackgroundChange:(id)sender;
- (IBAction)setNewLocation:(id)sender;
- (IBAction)setImageScale:(id)sender;

@end

```



```

Find: Layer
Replace:
Display Results in Find Smart Group
Contains
Ignore case
Options...

EAGLView.m
+ (Class)layerClass
return [CAEAGLLayer class];
// Get the layer
CAEAGLLayer *eaglLayer = (CAEAGLLayer *)self.layer;
CAEAGLLayer *eaglLayer = (CAEAGLLayer *)self.layer;
CAEAGLLayer *eaglLayer = (CAEAGLLayer *)self.layer;
CAEAGLLayer *eaglLayer = (CAEAGLLayer *)self.layer;
eaglLayer.opaque = TRUE;
eaglLayer.drawableProperties = [NSDictionary dictionaryWithObjectsAndKeys:
[renderer resizeFromLayer:(CAEAGLLayer *)self.layer],
[renderer resizeFromLayer:(CAEAGLLayer *)self.layer],
[renderer resizeFromLayer:(CAEAGLLayer *)self.layer],
];

EAGLView.m:32 -initWithCoder:
{
return [CAEAGLLayer class];

//The EAGL view is stored in the nib file. When it's unarchived it's sent -initWithCoder:
- (id)initWithCoder:(NSCoder *)coder
{
if ((self = [super initWithCoder:coder]))
{
// Get the layer
CAEAGLLayer *eaglLayer = (CAEAGLLayer *)self.layer;

eaglLayer.opaque = TRUE;
eaglLayer.drawableProperties = [NSDictionary dictionaryWithObjectsAndKeys:
[NSNumber numberWithInt:FALSE],
[NSNumber numberWithInt:FALSE],
[NSNumber numberWithInt:FALSE],
[NSNumber numberWithInt:FALSE],
[NSNumber numberWithInt:FALSE],
];

renderer = [[ES2Renderer
if (!renderer)
{
renderer = [[ES1Renderer
Found "Layer" - 12 occurrences

```

```

EAGLView.m - PictureViewer
Breakpoints Build and Run Tasks Info

```

Target "PictureViewer" Info

General Build Rules Properties Comments

Configuration: Active (Debug) Search in Build Settings

Show: All Settings

Setting	Value
Architectures	Optimized (armv6 armv7)
Additional SDKs	
Base SDK	iPhone Device 3.2
Build Active Architecture Only	<input type="checkbox"/>
Valid Architectures	armv6 armv7
Build Locations	
Build Products Path	build
Intermediate Build Files Path	build

StgLightRig.cpp: PictureViewer - Build Results

Device - 3.2 | Deb... Overview Breakpoints Build Build and Run Tasks Search

All Results Latest Results By Step By Issue Issues Only

Build PictureViewer
Project PictureViewer | Configuration Debug

- Check dependencies
- warning: building for deployment target '3.2' should omit the armv6 architecture.
- CompileXIB MainWindow.xib
- The 'window' outlet of 'Picture Viewer App Delegate' is connected to 'Window' but 'window' is no longer de...
- The 'myOutlet' outlet of 'Picture Viewer App Delegate' is connected to 'Rounded Rect Button' but 'my...'
- The 'glView' outlet of 'Picture Viewer App Delegate' is connected to 'View' but 'glView' is no longer de...
- Precompile PictureViewer_Prefix.pch
- Can't exec '/Developer/Platforms/iPhoneOS.platform/Developer/usr/bin/clang' (No such file or directory)

```

StgLightRig.cpp:1 <<No selected symbol>
#include "StgLightRig.h"
#include <algorithm>
#include <iterator>

StgLightRig::StgLightRig(std::string const &name, RigType type)
{
_name(name)
}

/* Name */
std::string const &StgLightRig::name() const {
return _name;
}

void StgLightRig::setName(std::string const &name) {
_name = name;
}

/* Type */
StgLightRig::RigType StgLightRig::type() const {
return _type;
}

```

Build failed (16 errors, 4 warnings) Failed 4 16 08:14:40 UTC 2008

iSpend - Debugger

10.5 | Development Overview Build and Go Tasks Restart Continue Step Over Step Into

Thread-1	PC	Instruction
0	0x9235349c	<<0000> mov
1	0x923534a1	<<0005> ool
2	0x923534a6	<<000a> _sysenter_trap
3	0x923534ab	<<000f> ret
4	0x923534b0	<<0014> nop

Variable Value Summary

Globals

GDB: Program loaded.

Copyright 2004 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are welcome to change it and/or distribute copies of it under certain conditions. Type "show copying" to see the conditions.
There is absolutely no warranty for GDB. Type "show warranty" for details.
This GDB was configured as "i386-apple-darwin". Program loaded.
sharedlibrary apply-load-rules all
Attaching to program: "/Developer/Examples/AppKit/iSpend/build/Development/iSpend.app/Contents/MacOS/iSpend", process 16645.
(gdb)

Found "NSRect" - 7 occurrences



Single Window

```
PictureViewer (iPhone Simula...
Build PictureViewer: Succeeded | 3:44 PM
No Issues
PictureViewer > Data > PictureDemo.h > @interface PictureDemo

//
// PictureDemo.h
//
// Version 1.0
//

#import <UIKit/UIKit.h>
#import <QuartzCore/QuartzCore.h>
#import "ESRenderer.h"

/** This class wraps the CAEAGLLayer from CoreAnimation into a convenient UIView subclass. The view content is basically an EAGL surface you render your OpenGL scene into. Note that setting the view non-opaque will only work if the EAGL surface has an alpha channel. */

@interface PictureDemo : UIView
{
@private
    id <ESRenderer> renderer;
    id pictureViewerStatus;

    BOOL animating;
    BOOL displayLinkSupported;
    NSInteger animationFrameInterval;

    /** Use of the CADisplayLink class is the preferred method for controlling your animation timing. CADisplayLink will link to the main display and fire every vsync when added to a given run-loop. */

    id displayLink;
    NSTimer *animationTimer;
    UITapGestureRecognizer *tapGestureRecognizer;

    EAGLContext *context;

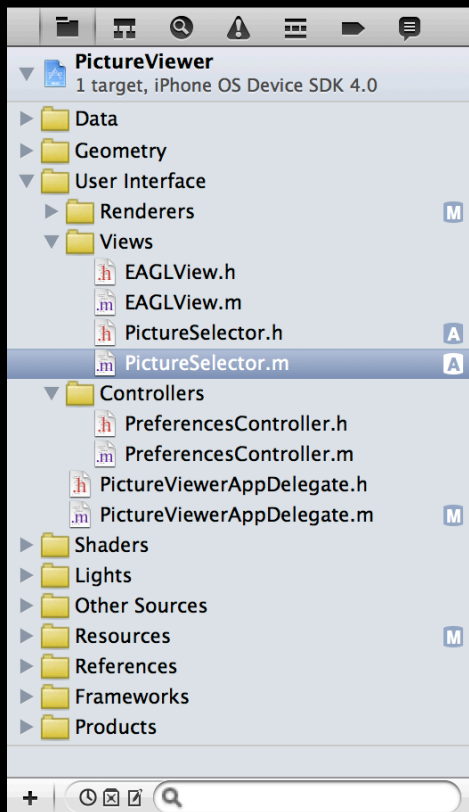
    // The pixel dimensions of the CAEAGLLayer
    float backingWidth;
    float backingHeight;

    // The OpenGL ES names for the framebuffer and renderbuffer
    float defaultFramebuffer, colorRenderbuffer;
}

@property (readonly, nonatomic, getter=isAnimating) BOOL animating;
@property (nonatomic) NSInteger animationFrameInterval;

- (IBAction)handleUseMipmaps:(id)sender;
- (IBAction)handleBackgroundChange:(id)sender;
- (IBAction)setNewLocation:(id)sender;
- (IBAction)setImageScale:(id)sender;

@end
```



Project

Symbol

Search

Issue

Debug

Breakpoint

Log

```
Build PictureViewer: Succeeded | 3:44 PM
No Issues

PictureViewer (iPhone Simula...
PictureViewer > Data > PictureDemo.h > @interface PictureDemo

//
// PictureDemo.h
//
// Version 1.0
//

#import <UIKit/UIKit.h>
#import <QuartzCore/QuartzCore.h>
#import "ESRenderer.h"

/** This class wraps the CAEAGLLayer from CoreAnimation into a convenient UIView subclass. The view content is basically an EAGL surface you render your OpenGL scene into. Note that setting the view non-opaque will only work if the EAGL surface has an alpha channel. */

@interface PictureDemo : UIView
{
@private
    id <ESRenderer> renderer;

    id pictureViewerStatus;

    BOOL animating;
    BOOL displayLinkSupported;
    NSInteger animationFrameInterval;

    /** Use of the CADisplayLink class is the preferred method for controlling your animation timing. CADisplayLink will link to the main display and fire every vsync when added to a given run-loop. */

    id displayLink;
    NSTimer *animationTimer;
    UITapGestureRecognizer *tapGestureRecognizer;

    EAGLContext *context;

    // The pixel dimensions of the CAEAGLLayer
    float backingWidth;
    float backingHeight;

    // The OpenGL ES names for the framebuffer and renderbuffer
    float defaultFramebuffer, colorRenderbuffer;
}

@property (readonly, nonatomic, getter=isAnimating) BOOL animating;

@property (nonatomic) NSInteger animationFrameInterval;

- (IBAction)handleUseMipmaps:(id)sender;
- (IBAction)handleBackgroundChange:(id)sender;
- (IBAction)setNewLocation:(id)sender;
- (IBAction)setImageScale:(id)sender;

@end
```

```
//
//  PictureDemo.h
//
//  Version 1.0
//

#import <UIKit/UIKit.h>
#import <QuartzCore/QuartzCore.h>
#import "ESRenderer.h"

/** This class wraps the CAEAGLLayer from CoreAnimation into a convenient UIView subclass. The view content is basically an EAGL surface you render your OpenGL scene into. Note that setting the view non-opaque will only work if the EAGL surface has an alpha channel. */

@interface PictureDemo : UIView
{
@private
    id <ESRenderer> renderer;
    id pictureViewerStatus;

    BOOL animating;
    BOOL displayLinkSupported;
    NSInteger animationFrameInterval;

    /** Use of the CADisplayLink class is the preferred method for controlling your animation timing. CADisplayLink will link to the main display and fire every vsync when added to a given run-loop. */

    id displayLink;
    NSTimer *animationTimer;
    UITapGestureRecognizer *tapGestureRecognizer;

    EAGLContext *context;

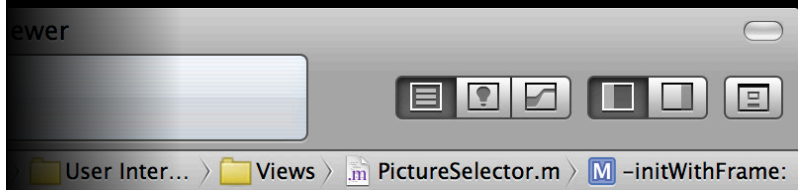
    // The pixel dimensions of the CAEAGLLayer
    float backingWidth;
    float backingHeight;

    // The OpenGL ES names for the framebuffer and renderbuffer
    float defaultFramebuffer, colorRenderbuffer;
}

@property (readonly, nonatomic, getter=isAnimating) BOOL animating;
@property (nonatomic) NSInteger animationFrameInterval;

- (IBAction)handleUseMipmaps:(id)sender;
- (IBAction)handleBackgroundChange:(id)sender;
- (IBAction)setNewLocation:(id)sender;
- (IBAction)setImageScale:(id)sender;

@end
```



Run and debug controls

Active launch scheme

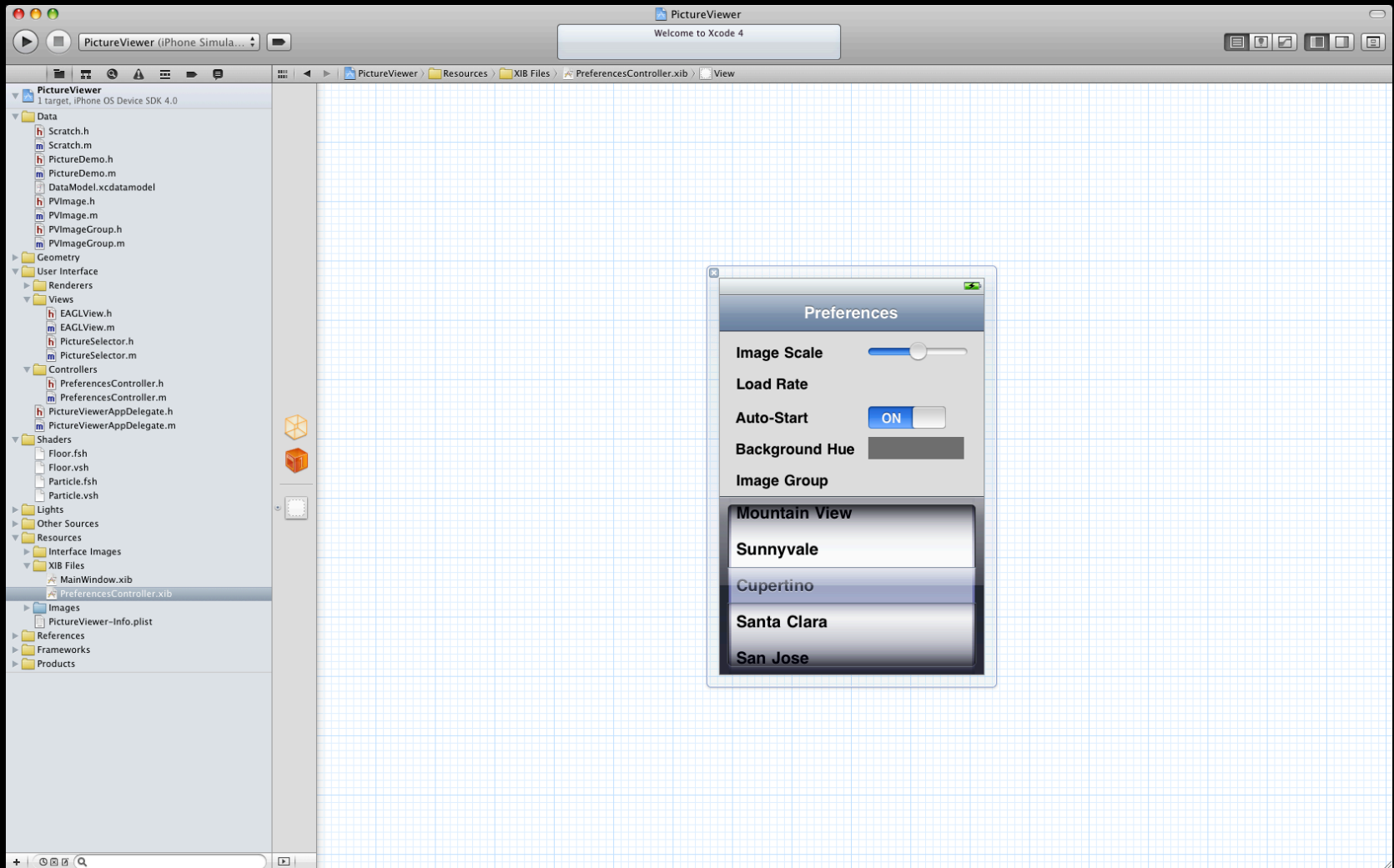
Activity viewer

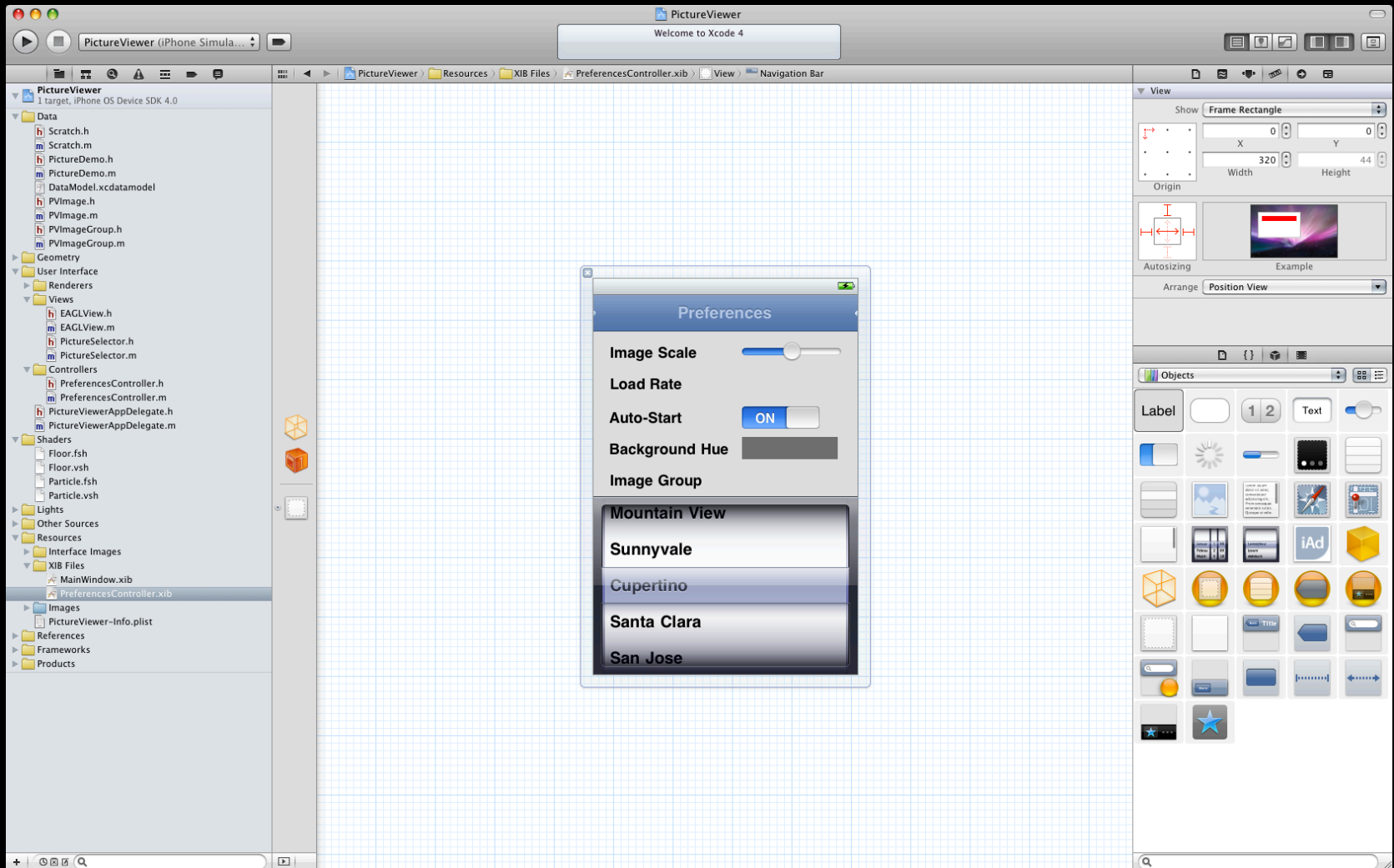
Editor modes

Jump bar



Interface Builder Inside







Fully integrated
Design canvas
Inspectors
Object library
Efficient layout



Assistant

```
PictureViewer
Build PictureViewer: Succeeded | 6:20 PM
No Issues

PictureViewer (iPhone OS Device)
PictureViewer > Data > PictureDemo.m -initWithCoder:

#import "PictureDemo.h"
#import "ES1Renderer.h"
#import "ES2Renderer.h"

@implementation PictureDemo
@synthesize animating;
@dynamic animationFrameInterval;

+ (Class)layerClass
{
    return [CAEAGLLayer class];
}

- (id)initWithCoder:(NSCoder*)coder
{
    if ((self = [super initWithCoder:coder]))
    {
        animating = FALSE;
        displayLinkSupported = FALSE;
        animationFrameInterval = 1;
        displayLink = nil;
        animationTimer = nil;

        // A system version of 3.1 or greater is required to use CADisplayLink.
        NSString *reqSysVer = @"3.1";
        NSString *currSysVer = [[UIDevice currentDevice] systemVersion];
        if ([currSysVer compare:reqSysVer options:NSNumericSearch] != NSOrderedAscending)
            displayLinkSupported = TRUE;

        tapGestureRecognizer = [[UITapGestureRecognizer alloc]
                               initWithTarget:self action:@selector(handleTap)];
        [tapGestureRecognizer setNumberOfTapsRequired:1];
        [self addGestureRecognizer:tapGestureRecognizer];
    }
    return self;
}

- (void)layoutSubviews
{
    [renderer resizeFromLayer:(CAEAGLLayer*)self.layer];
}

- (NSInteger)animationFrameInterval { return animationFrameInterval; }

- (void)setAnimationFrameInterval:(NSInteger)frameInterval
{
    if (frameInterval >= 1)
    {
        animationFrameInterval = frameInterval;
    }
}

- (void)dealloc
{
    [renderer release];
    [super dealloc];
}

@end
```

```

Build PictureViewer: Succeeded | 6:20 PM
No Issues

PictureViewer (iPhone OS Device)
PictureViewer > Data > PictureDemo.m | M -initWithCoder:
PictureViewer > Data > PictureDemo.h | P animating

#import "PictureDemo.h"
#import "ES1Renderer.h"
#import "ES2Renderer.h"

@implementation PictureDemo
@synthesize animating;
@dynamic animationFrameInterval;

+ (Class)layerClass
{
    return [CAEAGLLayer class];
}

- (id)initWithCoder:(NSCoder*)coder
{
    if ((self = [super initWithCoder:coder]))
    {
        animating = FALSE;
        displayLinkSupported = FALSE;
        animationFrameInterval = 1;
        displayLink = nil;
        animationTimer = nil;

        // A system version of 3.1 or greater is required to use CADisplayLink.
        NSString *reqSysVer = @"3.1";
        NSString *currSysVer = [[UIDevice currentDevice] systemVersion];
        if ([currSysVer compare:reqSysVer options:NSNumericSearch] != NSOrderedAscending)
            displayLinkSupported = TRUE;

        tapGestureRecognizer = [[UITapGestureRecognizer alloc]
                               initWithTarget:self action:@selector(handleTap)];
        [tapGestureRecognizer setNumberOfTapsRequired:1];
        [self addGestureRecognizer:tapGestureRecognizer];
    }
    return self;
}

- (void)layoutSubviews
{
    [renderer resizeFromLayer:(CAEAGLLayer*)self.layer];
}

- (NSInteger)animationFrameInterval { return animationFrameInterval; }

- (void)setAnimationFrameInterval:(NSInteger)frameInterval
{
    if (frameInterval >= 1)
    {
        animationFrameInterval = frameInterval;
    }
}

- (void)dealloc
{
    [renderer release];
    [super dealloc];
}
@end

//
// PictureDemo.h
//
// Version 1.0
//

#import <UIKit/UIKit.h>
#import <QuartzCore/QuartzCore.h>

#import "ESRenderer.h"

// This class wraps the CAEAGLLayer from CoreAnimation into a convenient UIView
// subclass. The view content is basically an EGL surface you render your
// OpenGL scene into. Note that setting the view non-opaque will only work if the
// EGL surface has an alpha channel.
@interface PictureDemo : UIView
{
@private
    id <ESRenderer> renderer;

    id pictureViewerStatus;

    BOOL animating;
    BOOL displayLinkSupported;
    NSInteger animationFrameInterval;

    // Use of the CADisplayLink class is the preferred method for controlling
    // your animation timing. CADisplayLink will link to the main display and
    // fire every vsync when added to a given run-loop.
    // The NSTimer class is used only as fallback when running on a pre 3.1
    // device where CADisplayLink isn't available.
    id displayLink;
    NSTimer *animationTimer;
    UITapGestureRecognizer *tapGestureRecognizer;

    EAGLContext *context;

    // The pixel dimensions of the CAEAGLLayer
    float backingWidth;
    float backingHeight;

    // The OpenGL ES names for the framebuffer and renderbuffer
    float defaultFramebuffer, colorRenderbuffer;
}

@property (readonly, nonatomic, getter=isAnimating) BOOL animating;
@property (nonatomic) NSInteger animationFrameInterval;

// - (IBAction)handleUseMipmaps:(id)sender;

@end

```

Build PictureViewer: Succeeded | 6:20 PM
No Issues

PictureViewer (iPhone OS Device) | PictureViewer > Resources > XIB Files > PreferencesController.xib > View > Navigation Bar

Manual | PictureViewer > Frameworks > UIKit.framework > Headers > UINavigationController.h | No Selection

Preferences

Image Scale

Load Rate

Auto-Start

Background Hue

Image Group

Mountain View

Sunnyvale

Cupertino

Santa Clara

San Jose

```

//
// UINavigationController.h
// UIKit
//
// Copyright 2005-2010 Apple Inc. All rights reserved.
//

#import <Foundation/Foundation.h>
#import <CoreGraphics/CoreGraphics.h>
#import <UIKit/UIView.h>
#import <UIKit/UIInterface.h>
#import <UIKit/UIFont.h>
#import <UIKit/UIKitDefines.h>
#import <UIKit/UIButton.h>

@class UINavigationControllerItem, UIBarButtonItem, UIImage, UIColor;
@protocol UINavigationControllerDelegate;

UIKIT_EXTERN_CLASS @interface UINavigationController : UIView <NSCoding> {
@private
    NSMutableArray *_itemStack;
    CGFloat _rightMargin;
    unsigned _state;
    id _delegate;
    UIView *_titleView;
    UIView *_leftView;
    UIView *_rightView;
    UIView *_prompt;
    UIView *_accessoryView;
    UIColor *_tintColor;
    struct {
        unsigned int animate:1;
        unsigned int animationDisabledCount:10;
        unsigned int transitioningBarStyle:1;
        unsigned int newBarStyle:3;
        unsigned int barStyle:3;
        unsigned int isTranslucent:1;
        unsigned int disableLayout:1;
        unsigned int backPressed:1;
        unsigned int animatePromptChange:1;
        unsigned int pendingHideBackButton:1;
        unsigned int titleAutosizesToFit:1;
        unsigned int usingNewAPI:1;
        unsigned int minibar:1;
        unsigned int forceFullHeightInLandscape:1;
        unsigned int isLocked:1;
        unsigned int shouldUpdatePromptAfterTransition:1;
        unsigned int crossfadeItems:1;
        unsigned int autoAdjustTitle:1;
    } _navbarFlags;
}

@property(n nonatomic, assign) UIBarStyle barStyle;
@property(n nonatomic, assign) id delegate;
@property(n nonatomic, retain) UIColor *tintColor;
@property(n nonatomic, assign, getter=isTranslucent) BOOL translucent __OSX_AVAILABLE_STARTING
(_MAC_NA, __IPHONE_3_0); // Default is NO. Always YES if barStyle is set to
UIBarStyleBlackTranslucent

// Pushing a navigation item displays the item's title in the center of the navigation bar.

```

Build PictureViewer: Succeeded | 6:20 PM
No Issues

PictureViewer (iPhone OS Device) | DataModel.xcdatamodel | Image | Model Classes | PVIImage.m

ENTITIES
 Image
 ImageFormat
 ImageGroup

FETCH REQUESTS
 HIDPIImages

CONFIGURATIONS
 (Default Configuration)

```

//
// PVIImage.m
//
#import "PVIImage.h"

@implementation PVIImage

@dynamic displayName;
@synthesize fileName;

- (NSString *)displayName {
    NSString * tmpValue;

    // [self willAccessValueForKey:@"displayName"];
    // tmpValue = [self primitiveDisplayName];
    // [self didAccessValueForKey:@"displayName"];

    return tmpValue;
}

- (void)setDisplayName:(NSString *)value {
    // [self willChangeValueForKey:@"displayName"];
    // [self setPrimitiveDisplayName:value];
    // [self didChangeValueForKey:@"displayName"];
}

- (BOOL)validateDisplayName:(id *)valueRef error:(NSError **)outError {
}

@dynamic format;

- (NSString *)format
{
    NSString * tmpValue;

    // [self willAccessValueForKey:@"format"];
    // tmpValue = [self primitiveFormat];
    // [self didAccessValueForKey:@"format"];

    return tmpValue;
}

- (void)setFormat:(NSString *)value
{
    // [self willChangeValueForKey:@"format"];
    // [self setPrimitiveFormat:value];
    // [self didChangeValueForKey:@"format"];
}

- (BOOL)validateFormat:(id *)valueRef error:(NSError **)outError
{
    // Insert custom validation logic here.
    return YES;
}

```

List Add Entity Add Attribute Graph



Counterparts

Sub- and superclasses

Included, included-by

Actions, outlets, bindings

Model classes

Demo

Matthew Firlik



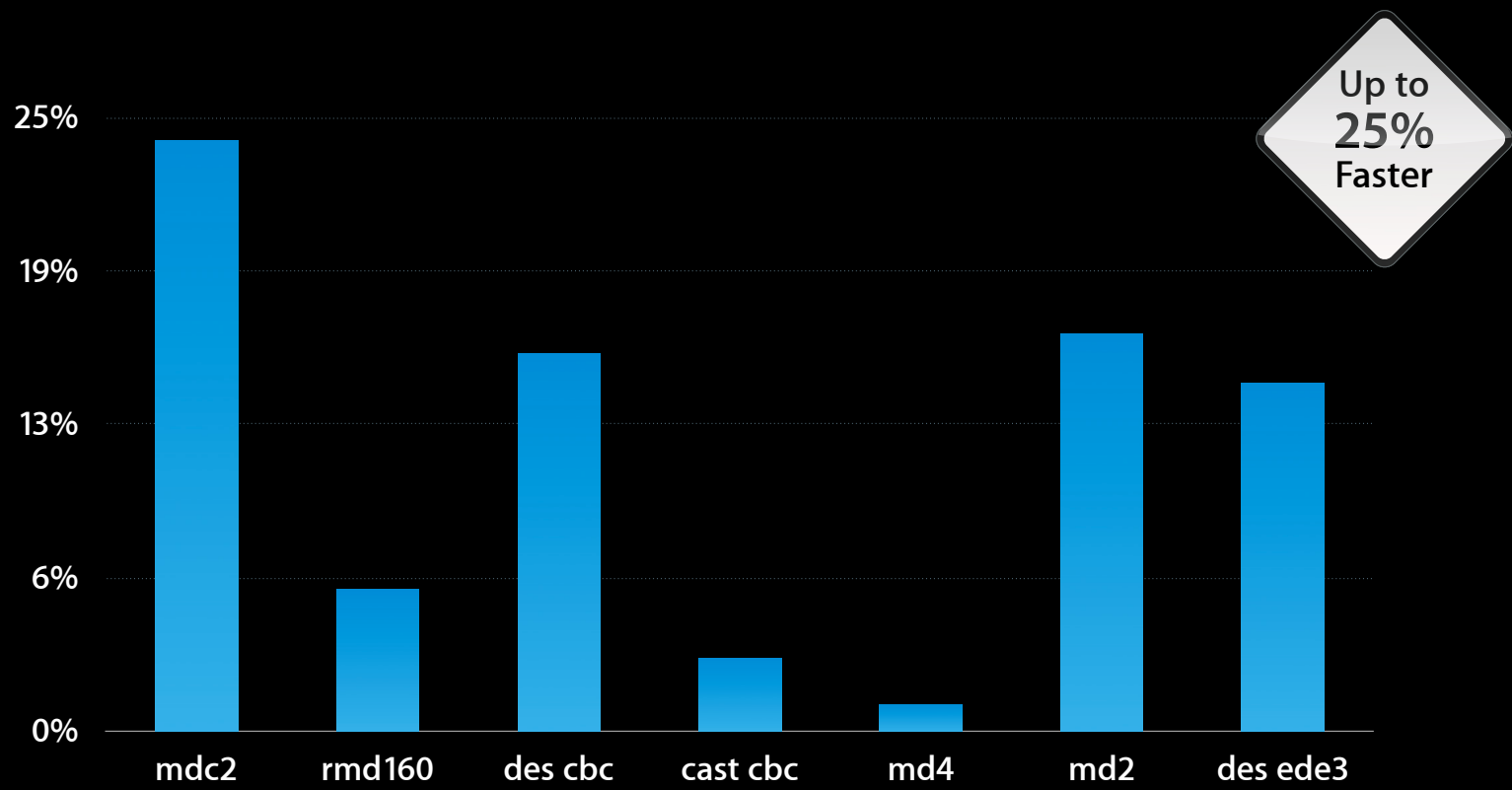
Single window Interface Builder Assistant



LLVM Compiler 2

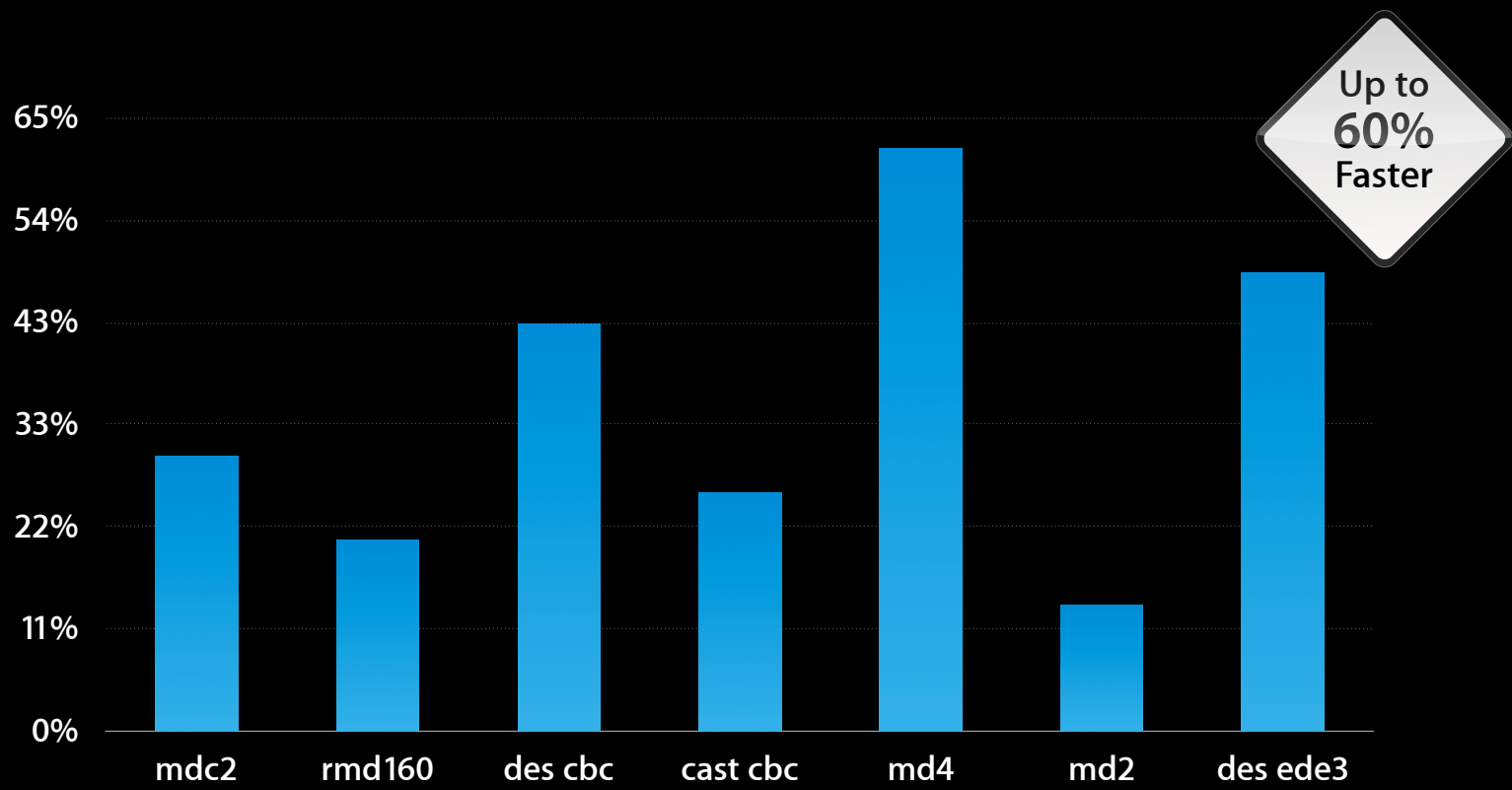


Compile Time



LLVM percent faster than GCC

Runtime Performance on Mac



LLVM percent faster than GCC

Runtime Performance on iPhone

```
struct A { int F; } Str;
```

```
int compute1(int, float, struct A, int, double, double);  
int compute2(int);
```

```
int foo(struct A *Ptr, int X) {
```

```
    compute1(X, 4.0f, X + 42, 92, 3.0f, 2.0f);
```

❗ Incompatible type passing 'int', expected 'struct A'

```
    return X + compute2(X ? ((Str.F + 40) + Str) / 42 + Str.F : Ptr->F);
```

❗ Invalid operands to binary expression

```
}
```

C++



Fix-it


Teh quick brown fox jumped over the lazy dog|


The


Ignore Spelling

Learn Spelling

```
void doSomething(int num) {  
    StgRefPtr<StgLightRig> rig = NULL;  
  
    if (num = rig->lights().size()) {  
    }  
}
```

 **Using the result of an assignment as a condition without parentheses**

 Use '==' to turn this assignment into an equality comparison

 Place parentheses around the assignment to silence this warning

Done

Fix



Misspelled symbols

Accidental assignments

Incorrect parenthesis

Missing semicolons

Wrong enums, structs, classes

```
-(id)makeMeAnObject {
    NSObject *objectID =
        [NSString stringWithString:@"Hi"];
    [objectID autorelease];
    return objectID;
}
```

1. Method returns an Objective-C object with a +0 retain count

2. Object sent -autorelease message

3. Object returned to caller with a +0 (non-owning) retain count

Demo

Mike Ferris



LLVM compiler 2

C++ support

Enhanced code completion

Static analysis

Fix-it



Version Editor

Subversion

Git

Subversion



Checkout, commit, update

Status in navigator

Easy file compare

Branching

Merging

```
PictureViewer (iPhone OS Device)
Welcome to Xcode 4

PictureViewer > User Interface > GL View > ES2Renderer.mm > No Selection

15 #include "Vec3.h"
16 #include "DirectionalForce.h"
17 #include "CollisionPlane.h"
18 #include "RandFieldForce.h"
19 #include "Defrobulator.h"
20 #include <vector>
21
22 // #define ENABLE_MULTI_SAMPLING
23 #define USE_VAO
24 #define ENABLE_MIP_MAPS
25 #define GENERATE_MIP_MAPS
26
27 #define PARTICLE_WIDTH 2
28 #define PARTICLE_HEIGHT 1
29 #define PARTICLE_DEPTH 7
30 #define ZOOM_IN 15.0f
31 #define ZOOM_OUT 45.0f
32 #define PICTURE_OFFSET 3
33 #define TRANSITION_TIME 1.0f
34 #define ZOOM_WIDTH 1.6f
35
36 #define BYTE_OFFSET(a, b) ((char*)(a) - (char*)(b))
37
38 namespace ParticleUniform
39 {
40     enum Enum
41     {
42         ViewProjMtx,
43         WorldMtx,
44         Color,
45         Picture,
46         Frame,
47         LightMap,
48         LightMapPercent,
49         Specularity,
50         GrayPercent,
51         Max
52     };
53
54 GLint s_particleUniforms[ParticleUniform::Max];
55 GLint s_floorUniforms[FloorUniform::Max];
56
57 // attribute index
58 enum {
59     ATTRIB_VERTEX,
60     ATTRIB_TEXCOORD,
61     NUM_ATTRIBUTES
62 };
63
64 struct Viewport
65 {
66     float x;
67     float y;
68     float width;
69 };
70
71 #include "Particle.h"
72 #include "Mtx.h"
73 #include "Vec3.h"
74 #include "DirectionalForce.h"
75 #include "CollisionPlane.h"
76 #include "RandFieldForce.h"
77 #include <vector>
78
79 // #define ENABLE_MULTI_SAMPLING
80 #define USE_VAO
81 #define ENABLE_MIP_MAPS
82 #define GENERATE_MIP_MAPS
83
84 #define PARTICLE_WIDTH 2
85 #define PARTICLE_HEIGHT 1
86 #define ZOOM_IN 15.0f
87 #define ZOOM_OUT 45.0f
88 #define PICTURE_OFFSET 3
89 #define TRANSITION_TIME 1.0f
90 #define ZOOM_WIDTH 1.6f
91 #define ZOOM_HEIGHT 0.8f
92
93 #define BYTE_OFFSET(a, b) ((char*)(a) - (char*)(b))
94
95 namespace ParticleUniform
96 {
97     enum Enum
98     {
99         ViewProjMtx,
100        WorldMtx,
101        Color,
102        Picture,
103        Frame,
104        LightMap,
105        LightMapPercent,
106        GrayPercent,
107        Max
108     };
109
110 namespace FloorUniform
111 {
112     enum Enum
113     {
114         DiffuseMap,
115         LightMap,
116         Max
117     };
118
119 GLint s_particleUniforms[ParticleUniform::Max];
120 GLint s_floorUniforms[FloorUniform::Max];
121
122 // attribute index
123 enum {
124     ATTRIB_VERTEX,
```

```
PictureViewer (iPhone OS Device)
Welcome to Xcode 4

PictureViewer > User Interface > GL View > ES2Renderer.mm > No Selection

15 #include "Vec3.h"
16 #include "DirectionalForce.h"
17 #include "CollisionPlane.h"
18 #include "RandFieldForce.h"
19 #include "Defrobulator.h"
20 #include <vector>
21
22 // #define ENABLE_MULTI_SAMPLING
23 #define USE_VAO
24 #define ENABLE_MIP_MAPS
25 #define GENERATE_MIP_MAPS
26
27 #define PARTICLE_WIDTH 2
28 #define PARTICLE_HEIGHT 1
29 #define PARTICLE_DEPTH 7
30 #define ZOOM_IN 15.0f
31 #define ZOOM_OUT 45.0f
32 #define PICTURE_OFFSET 3
33 #define TRANSITION_TIME 1.0f
34 #define ZOOM_WIDTH 1.6f
35
36 #define BYTE_OFFSET(a, b) ((char*)(a) - (char*)(b))
37
38 namespace ParticleUniform
39 {
40     enum Enum
41     {
42         ViewProjMtx,
43         WorldMtx,
44         Color,
45         Picture,
46         Frame,
47         LightMap,
48         LightMapPercent,
49         Specularity,
50         GrayPercent,
51         Max
52     };
53 };
54
55 GLint s_particleUniforms[ParticleUniform::Max];
56 GLint s_floorUniforms[FloorUniform::Max];
57
58 // attribute index
59 enum {
60     ATTRIB_VERTEX,
61     ATTRIB_TEXCOORD,
62     NUM_ATTRIBUTES
63 };
64
65 struct Viewport
66 {
67     float x;
68     float y;
69     float width;
70 };
71
72 #include "Particle.h"
73 #include "Mtx.h"
74 #include "Vec3.h"
75 #include "DirectionalForce.h"
76 #include "CollisionPlane.h"
77 #include "RandFieldForce.h"
78 #include <vector>
79
80 // #define ENABLE_MULTI_SAMPLING
81 #define USE_VAO
82 #define ENABLE_MIP_MAPS
83 #define GENERATE_MIP_MAPS
84
85 #define PARTICLE_WIDTH 2
86 #define PARTICLE_HEIGHT 1
87 #define ZOOM_IN 15.0f
88 #define ZOOM_OUT 45.0f
89 #define PICTURE_OFFSET 3
90 #define TRANSITION_TIME 1.0f
91 #define ZOOM_WIDTH 1.6f
92 #define ZOOM_HEIGHT 0.8f
93
94 #define BYTE_OFFSET(a, b) ((char*)(a) - (char*)(b))
95
96 namespace ParticleUniform
97 {
98     enum Enum
99     {
100         ViewProjMtx,
101         WorldMtx,
102         Color,
103         Picture,
104         Frame,
105         LightMap,
106         LightMapPercent,
107         GrayPercent,
108         Max
109     };
110 };
111
112 namespace FloorUniform
113 {
114     enum Enum
115     {
116         DiffuseMap,
117         LightMap,
118         Max
119     };
120 };
121
122 GLint s_particleUniforms[ParticleUniform::Max];
123 GLint s_floorUniforms[FloorUniform::Max];
124
125 // attribute index
126 enum {
127     ATTRIB_VERTEX,
```

Demo

Max Drukman



Git and Subversion

Branching and merging

Version Editor

Compare files over time

Logs and blame



New Debugger

The screenshot shows the Xcode IDE with the following components:

- Debugger Window:** Shows the thread stack for Thread 1, with a breakpoint at line 136 of `ES2Renderer.mm`.
- Code Editor:** Displays the following Objective-C code:


```

// Setup some of sources
glEnable(GL_CULL_FACE);
glEnable(GL_BLEND);
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
glClearColor(0, 0, 0, 1);

// Create frame texture
NSString *path = [[NSBundle mainBundle] pathForResource:@"Frame" ofType:@"png"];
GLuint texture = [self createTextureFromFile:path];
s_textures.push_back(texture);

path = [[NSBundle mainBundle] pathForResource:@"Floor" ofType:@"png"];
texture = [self createTextureFromFile:path];
s_textures.push_back(texture);

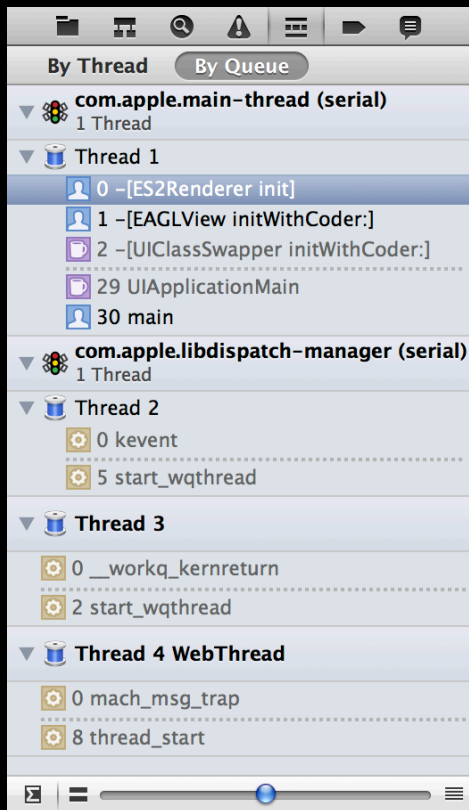
path = [[NSBundle mainBundle] pathForResource:@"Lightmap" ofType:@"png"];
texture = [self createTextureFromFile:path];
s_textures.push_back(texture);

// Load the pictures
NSArray *paths = [[NSBundle mainBundle] pathsForResource:@"png" inDirectory:@"Images"];
for (int i = 0; i < [paths count]; ++i)
{
    GLuint texture;
    texture = [self createTextureFromFile:[paths objectAtIndex:i]];
    s_textures.push_back(texture);
}

GLuint program;
NSString *vtxShdrPath = [[NSBundle mainBundle] pathForResource:@"Particle" ofType:@"vsh"];
NSString *fragShdrPath = [[NSBundle mainBundle] pathForResource:@"Particle" ofType:@"fsh"];
program = [self createProgramFromVtxShdrFile:vtxShdrPath andFragShdrFile:fragShdrPath];
s_programs.push_back(program);

// Get uniform locations
s_particleUniforms[ParticleUniform::ViewProjMtx] = glGetUniformLocation(program, "u_viewProjMtx");
s_particleUniforms[ParticleUniform::WorldMtx] = glGetUniformLocation(program, "u_worldMtx");
s_particleUniforms[ParticleUniform::Color] = glGetUniformLocation(program, "u_color");
s_particleUniforms[ParticleUniform::Picture] = glGetUniformLocation(program, "u_picture");
s_particleUniforms[ParticleUniform::Frame] = glGetUniformLocation(program, "u_frame");
s_particleUniforms[ParticleUniform::LightMap] = glGetUniformLocation(program, "u_lightMap");
s_particleUniforms[ParticleUniform::LightMapPercent] = glGetUniformLocation(program, "u_lightMapPercent");
s_particleUniforms[ParticleUniform::GrayPercent] = glGetUniformLocation(program, "u_grayPercent");

vtxShdrPath = [[NSBundle mainBundle] pathForResource:@"Floor" ofType:@"vsh"];
fragShdrPath = [[NSBundle mainBundle] pathForResource:@"Floor" ofType:@"fsh"];
program = [self createProgramFromVtxShdrFile:vtxShdrPath andFragShdrFile:fragShdrPath];
s_programs.push_back(program);
      
```
- Variable Inspector:** Shows the state of `self` (ES2Renderer) and `NSObject` (NSObject).
- Console:** Displays the standard GDB startup messages, including the Objective-C++ language setting.



Debug navigator

Breakpoints

Threads

Queues

Stack compression

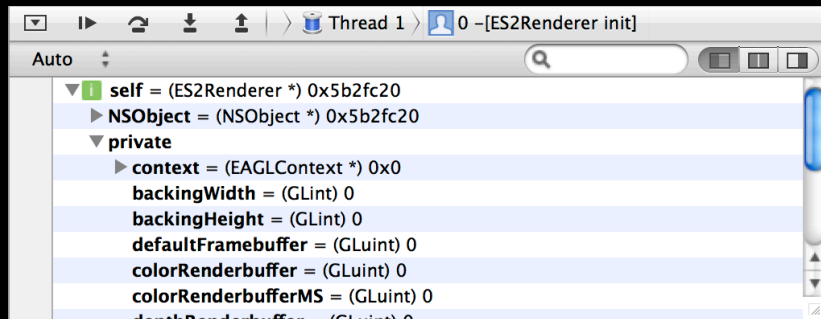
The screenshot displays the Xcode IDE with the following components:

- Debugger:** Shows the execution flow of the `ES2Renderer` thread. The current thread is `Thread 1`, which has stopped at a breakpoint in the `glGetUniformLocation` call for `u_viewProjMtx`.
- Code Editor:** Contains the `ES2Renderer.mm` file. The code includes:
 - OpenGL ES 2.0 initialization: `glEnable(GL_CULL_FACE); glEnable(GL_BLEND); glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA); glClearColor(0, 0, 0, 1);`
 - Texture loading: `NSString *path = [NSBundle mainBundle] pathForResource:@"Frame" ofType:@"png"]; GLuint texture = [self createTextureFromFile:path]; s_textures.push_back(texture);`
 - Shader loading: `GLuint program; NSString *vtxShdrPath = [NSBundle mainBundle] pathForResource:@"Particle" ofType:@"vsh"]; NSString *fragShdrPath = [NSBundle mainBundle] pathForResource:@"Particle" ofType:@"fsh"]; program = [self createProgramFromVtxShdrFile:vtxShdrPath andFragShdrFile:fragShdrPath]; s_programs.push_back(program);`
 - Uniform location retrieval: `s_particleUniforms[ParticleUniform::ViewProjMtx] = glGetUniformLocation(program, "u_viewProjMtx");`
- Debugger Console:** Shows the following output:


```

gdb is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB. Type "show warranty" for details.
This GDB was configured as "x86_64-apple-darwin"./Users/timt/bin:/opt/bin:/opt/local/bin:/opt/
local/sbin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/X11/bin

sharedlibrary apply-load-rules all
Attaching to process 5640.
Pending breakpoint 1 - ""ES2Renderer.mm":136" resolved
Pending breakpoint 2 - ""ES2Renderer.mm":166" resolved
Current language: auto; currently objective-c++
Not safe to look up objc runtime data.
[Switching to process 5640]
(gdb)
      
```



Step in, out, over

Step line, instruction, thread

Console

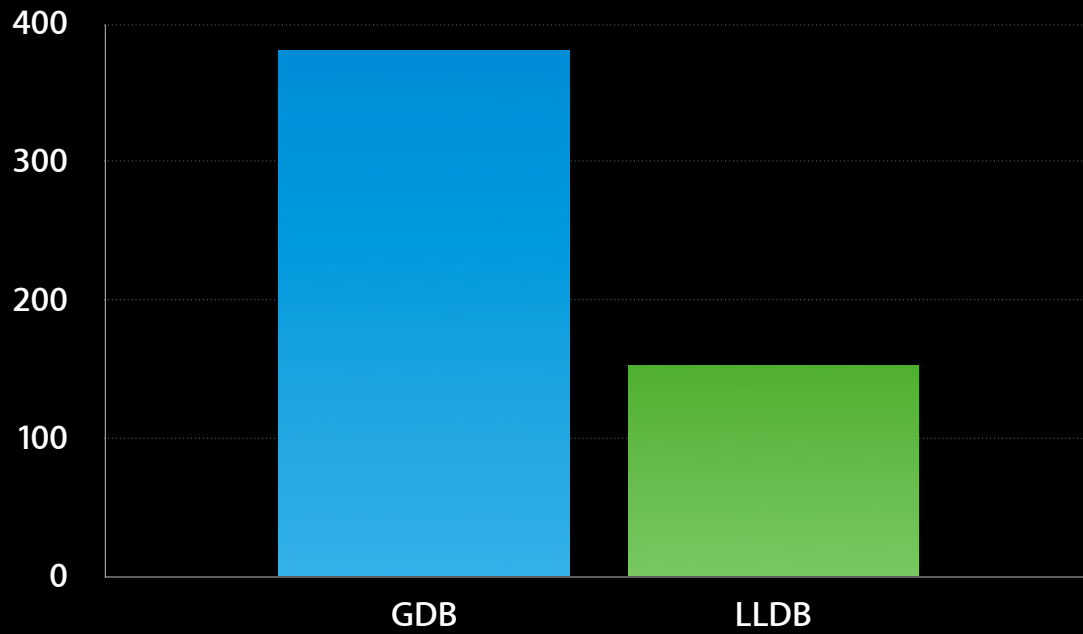
Variables

Automatic filtering

GDB



Symbol Loading Speed



Debugger memory (MB) usage
when stopped at breakpoint

Better Memory Usage

Open Source at LLVM.org

Demo

Dave Payne



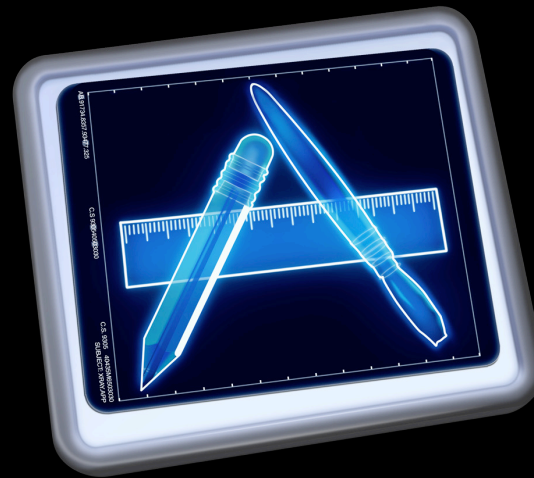
New debugger interface

Auto variables filtering

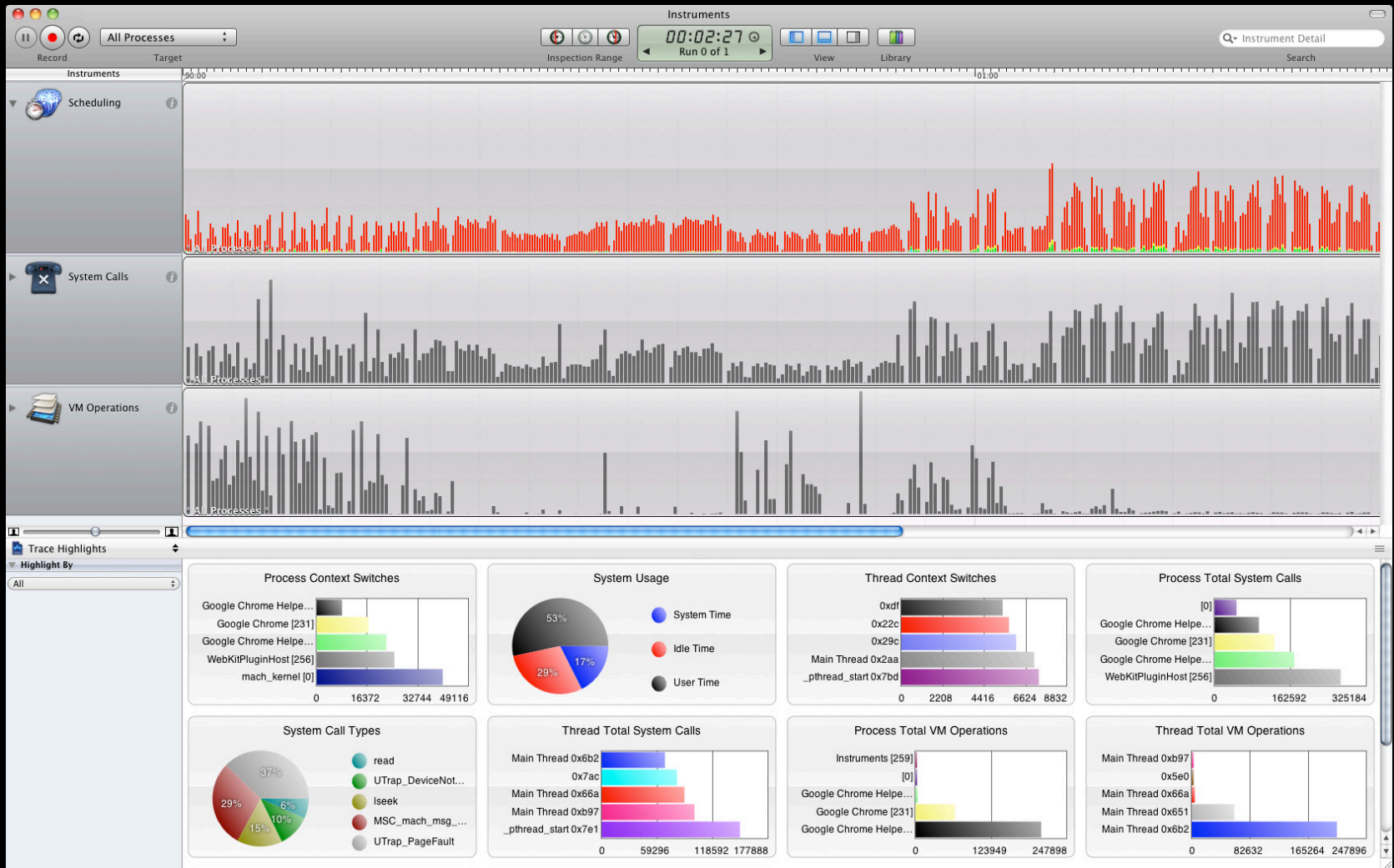
Stack compression

Fast and efficient

LLDB



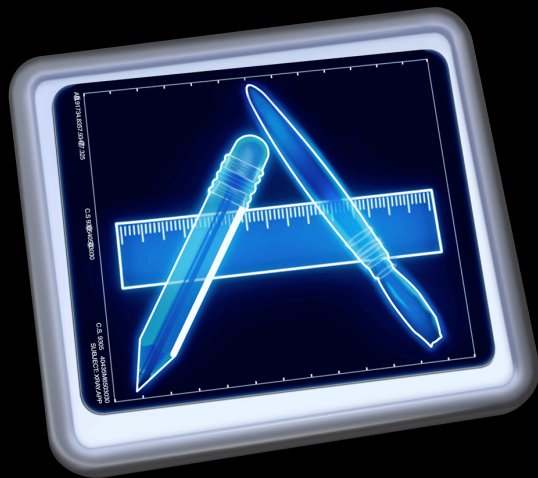
Instruments





Demo

Steve Lewallen



Updated interface
Stack compression
System trace
Heap marking
New OpenGL ES instrument



Xcode 4



Single window

Interface Builder inside

Assistant

LLVM Developer preview
compiler available today

Fix-it

Version Editor

New debugger



