

OsiriX Plugin Development



David Geffen
School of Medicine

UCLA Health System



Kyung Sung, PhD

MR Research Lab, UCLA

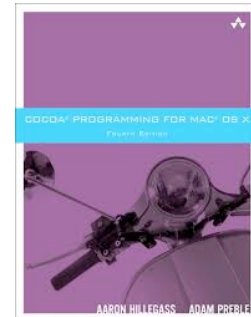
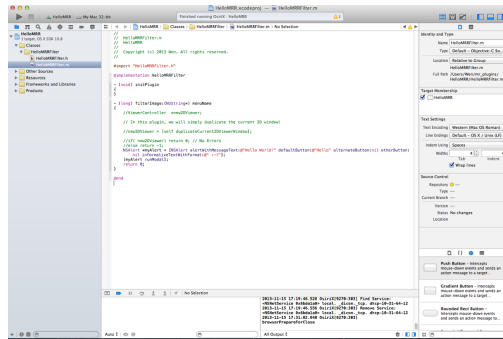
[updated by Wenrui Yang](#)

Overview

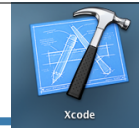
- Getting ready to develop Osirix plugins
 - Install Xcode
 - Get source codes for OsiriX and plugins
- Plugin Examples
 - “Hello MRR” level 1, 2, and 3

Developing Software on Mac OS

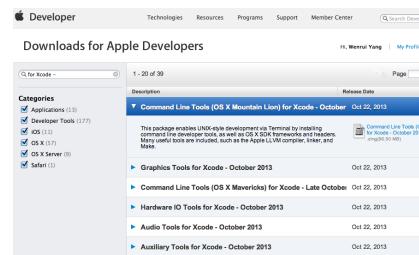
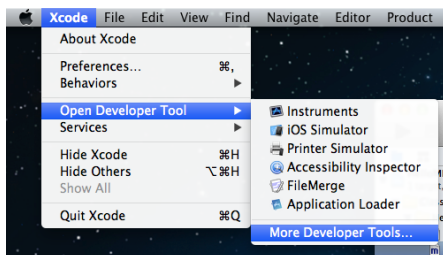
- To develop applications for Mac
 - Learn Cocoa Programming (Objective C language)
 - http://www.amazon.com/Cocoa-Programming-Mac-4th-Edition/dp/0321774086/ref=dp_ob_title_bk
- Get to know Xcode (Integrated development platform)



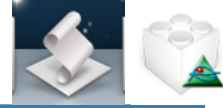
Getting Ready to Develop



- Download Xcode: <https://developer.apple.com/xcode/>
 - Now the command line tool is no longer included in the newest version of Xcode and therefore this tool needs to be installed manually: go to Xcode ->Open Developer Tool -> More Developer Tools;
 - The webpage will be opened: <https://developer.apple.com/downloads/index.action?name=for%20Xcode%20-> (you have to have an apple ID to login to the developer site) and choose Command Line Tools



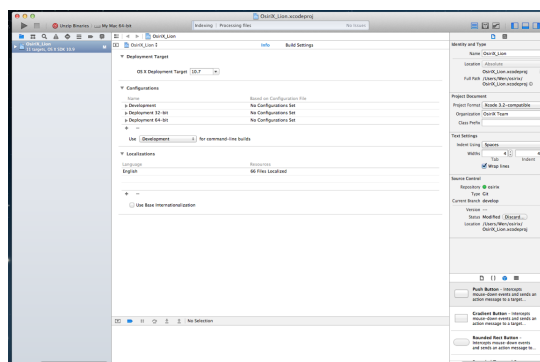
Getting Ready to Develop



- Location of development source code has been moved to **github**.
- Download source code for Osirix and plugins:
 - Open Terminal application (Applications/Utilities/)
 - Type the following:
 - `$ git clone http://github.com/pixmeo/osirix.git`
 - `$ git clone http://github.com/pixmeo/osirixplugins.git`
 - A folder called **osirix** and a folder called **osirixplugins** would be created with the source code after these two commands are done.

Xcode Configuration

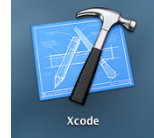
- In the **osirix** folder, there is a project named **Osirix_Lion.xcodeproj**, select the project and run the “Unzip Binaries” target
- Make sure that the OS X build target matches the current version of the operating system of your mac



So Far ...

- You have downloaded:

- Xcode
- Osirix Source Code
- Osirix Plugin Source Code



- Please complete these steps before you ask about development questions.
- Ready for some real development work?



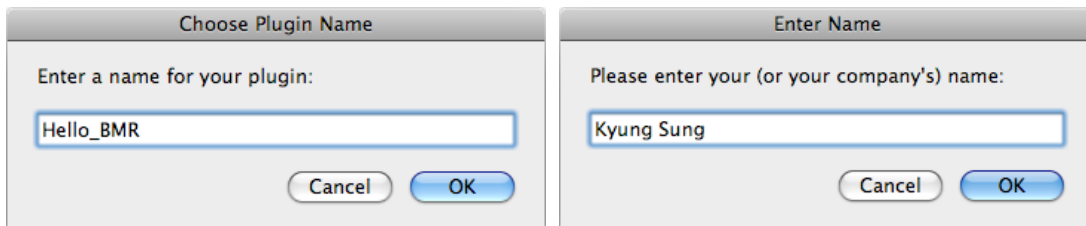
Hello BMR Plugin Example

Hello BMR - Level 1 Objectives

- By the end of the Level 1 of this example, you will be able to:
 - Create a new plugin project
 - Compile (Run) your project
 - Add your plugin to OsiriX

“Hello BMR” - Level 1

- Open “Osirix Plugin Generator.app” (/Applications/Utilities/osirixplugins/_help/)



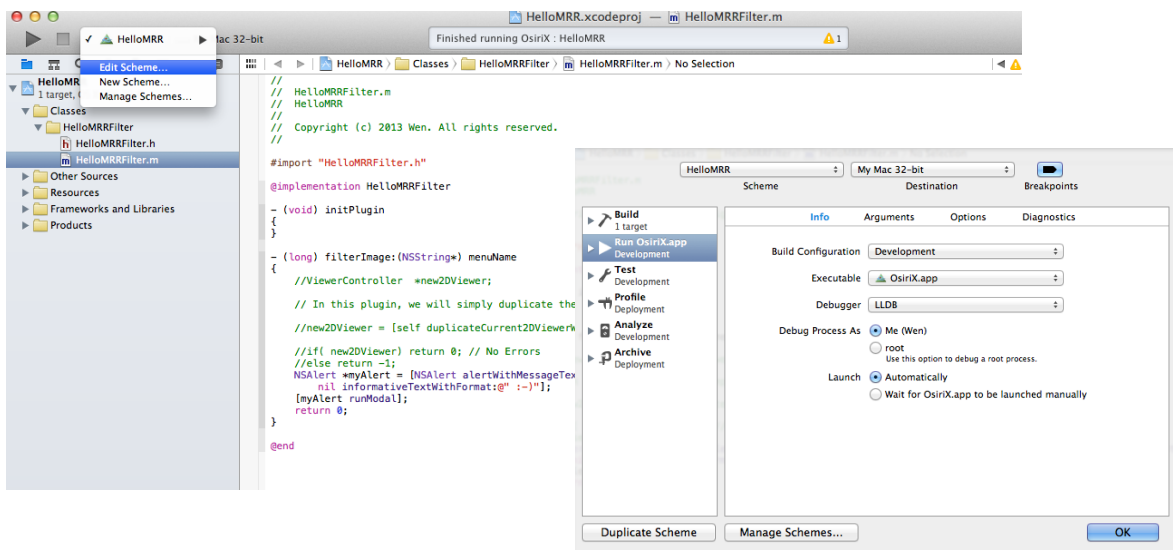
- Once generated, a folder of the project’s name would be created and select and open the Hello_BMR.xcodeproj file to open Xcode. Run the project and a list of files are created for the project

“Hello BMR” - Level 1

- **Hello_BMR.xcodeproj**: your project file folder
- **English.lproj**: contains localized data to create a multi-language plugin
- **Hello_BMRFilter.h** & **Hello_BMRFilter.m**: contain the actual plugin implementations -> your codes go HERE!
- **Info.plist** & **version.plist**: contain information about the plugin name, icon, version number, etc.
- **Osirix Headers**: contain the objects you can access from your plugin, they must ALWAYS be present!

“Hello BMR” - Level 1

- To build the project as a plugin, we need to edit the scheme: when select the HelloBMR project, you can choose “edit scheme” under which you can choose “Osirix” as the Executable



“Hello BMR” - Level 1

```
//  
// Hello_BMRFilter.m  
// Hello_BMR  
//  
// Copyright (c) 2010 Kyung. All rights reserved.  
//  
  
#import "Hello_BMRFilter.h"  
  
@implementation Hello_BMRFilter  
  
- (void) initPlugin  
{  
}  
  
- (long) filterImage:(NSString*) menuName  
{  
    ViewController *new2DViewer;  
  
    // In this plugin, we will simply duplicate the current 2D window!  
  
    new2DViewer = [self duplicateCurrent2DViewerWindow];  
  
    if( new2DViewer) return 0; // No Errors  
    else return -1;  
}  
  
@end
```

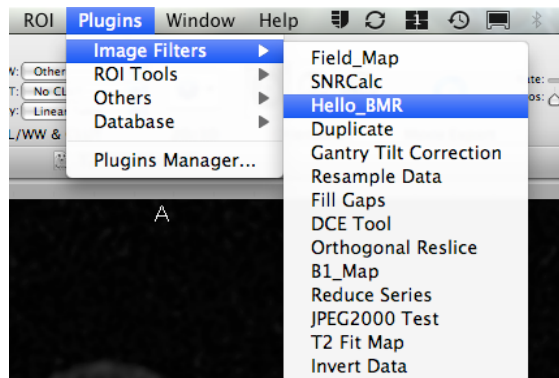
When OsiriX is launched, it will look in the 'plugin' folders for files with the .osirixplugin extensions

The main function of a plugin. OsiriX will always use this function to call your plugin.

You can call virtually any function of OsiriX.

“Hello BMR” - Level 1

- Once the project is built, a file of the type .osirixplugin will be created under the same directory in build->development
- Copy it to /Library/Application Support/OsiriX/Plugins/ for it to show in the plugin menu in OsiriX
- Then you can test the plugin in OsiriX



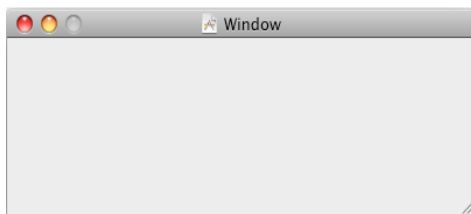
info.plist

- You can change the type of your plugin in the menu by changing the “pluginType” field in info.plist file
 - imageFilter: Image Filters
 - roiTool: ROIs tools
 - other: Others
 - fushionFilter: Fusion
 - Database: database

Key	Value
▼ Information Property List	(10 items)
Localization native development re	English
Executable file	Hello_BMR
Icon file	
InfoDictionary version	6.0
Bundle OS Type code	BNDL
Bundle creator OS Type code	????
Bundle version	1.0
Principal class	Hello_BMRFilter
▶ MenuTitles	(1 item)
pluginType	imageFilter

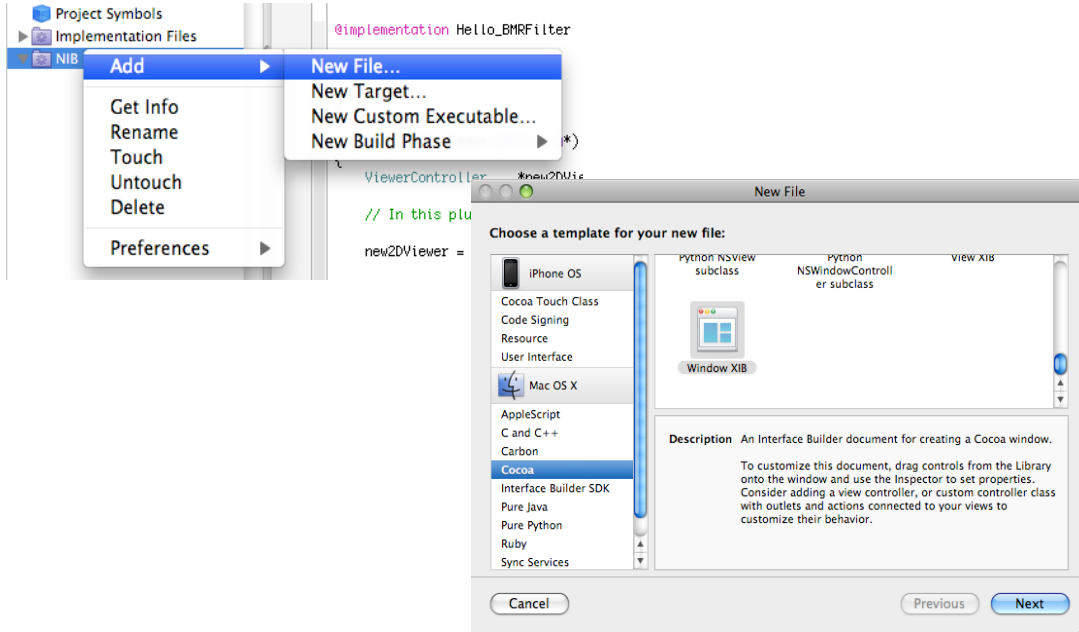
“Hello BMR” - Level 2 Objectives

- By the end of level 2 of this example, you will be able to:
 - Create a new NSWindow
 - Use “IBOutlet” and “IBAction”
 - Learn how to use the “Interface Builder”



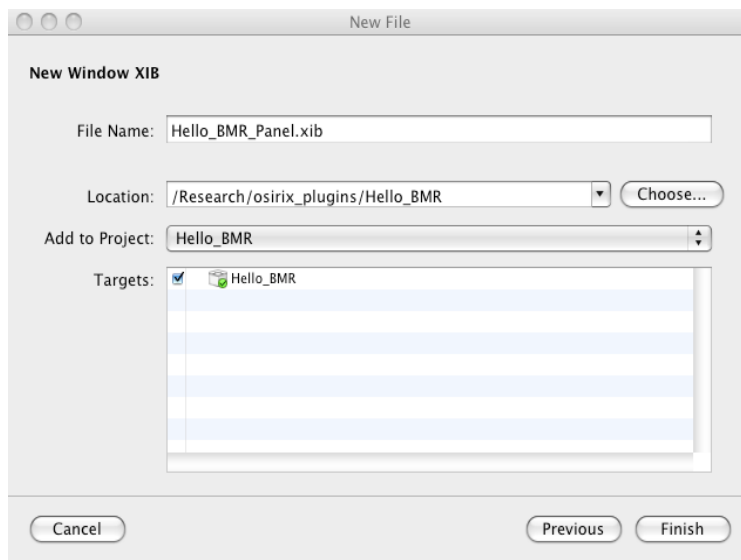
“Hello BMR” – Level 2

- Add a new “XIB” (or “NIB”) file to create the interface



“Hello BMR” – Level 2

- Name your “XIB” file



“Hello BMR” – Level 2

- “IBOutlet” is a hint to Interface Builder when it read the declaration of a class for a .h file.
- “IBAction” acts also as a hint to Interface Builder.
- In Hello_BMRFilter.h

```
@interface Hello_BMRFilter : PluginFilter {
    IBOutlet NSWindow      *myWindow;
}

- (long) filterImage:(NSString*) menuName;
- (IBAction) DuplicateIt: (id)sender;
```

“Hello BMR” – Level 2

- Hello_BMRFilter.m

```
- (long) filterImage:(NSString*) menuName
{
    NSWindowController *window = [[NSWindowController alloc] initWithWindowNibName:@"Hello_BMR_Panel"
                                                                    owner:self];

    [window showWindow:self]; // This will bring up the window
    return 0; // No Errors
}

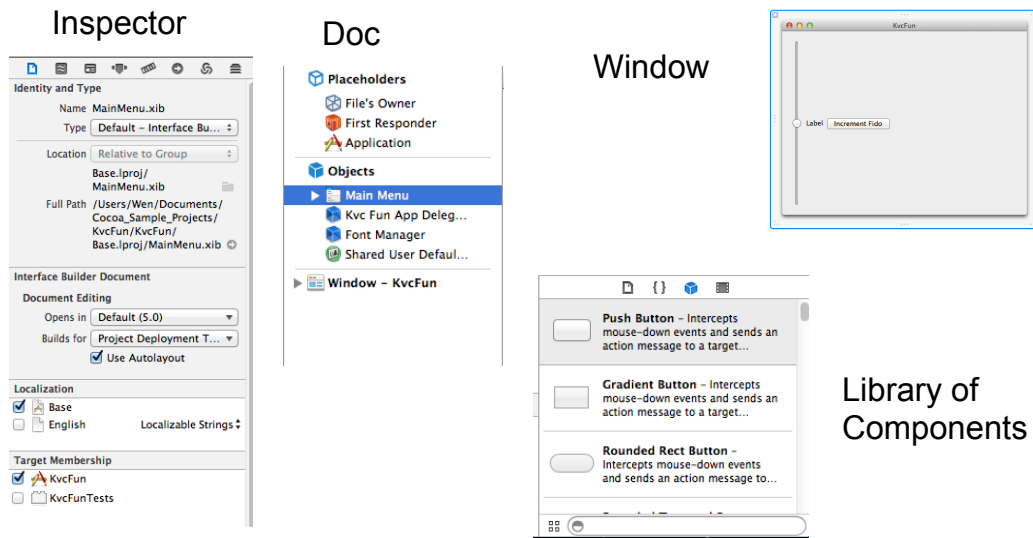
- (IBAction) DuplicateIt: (id)sender;
{
    ViewController      *new2DViewer;

    // In this plugin, we will simply duplicate the current 2D window!

    new2DViewer = [self duplicateCurrent2DViewerWindow];
}
.
```

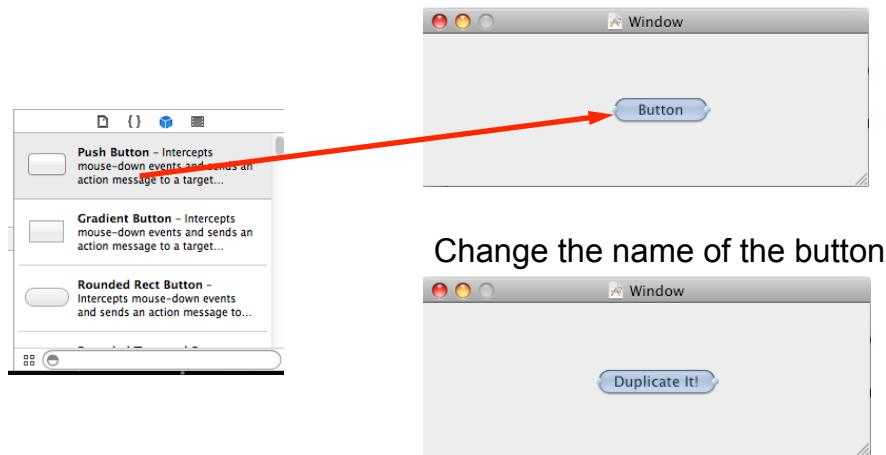
Interface Builder

- Double click “Hello_BMR_Panel.xib” and it will open Interface Builder



Interface Builder

- Create a new “NSButton”

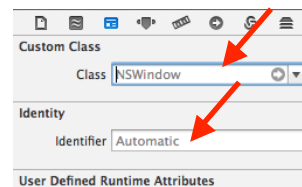
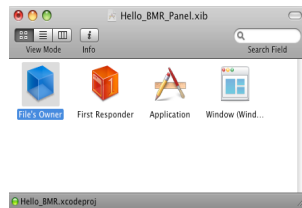


Interface Builder

- You have created **IBOutlet** and **IBAction**:

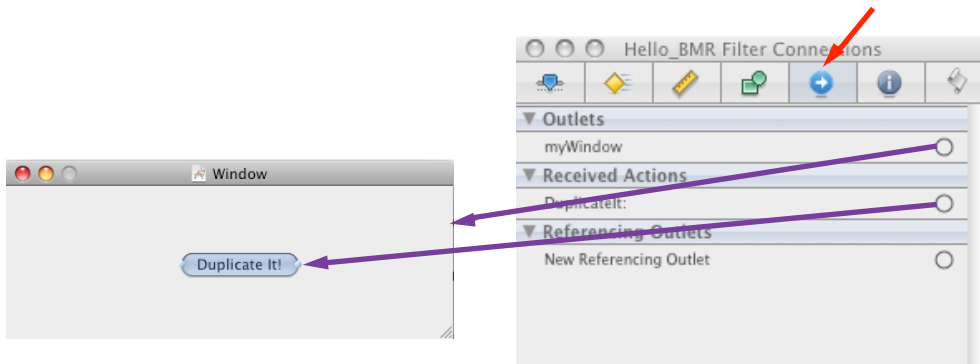
```
@interface Hello_BMRFilter : PluginFilter {  
    IBOutlet NSWindow      *myWindow;  
}  
  
- (long) filterImage:(NSString*) menuName;  
- (IBAction) DuplicateIt: (id)sender;
```

- Click “File’s Owner” in the Doc Window and click “Identity” tab in the Inspector
- Select class identity as “Hello_BMRFilter”



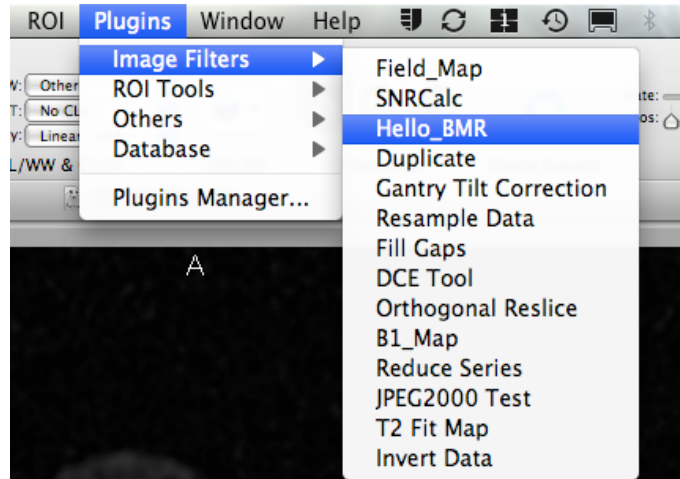
Interface Builder

- Click “Connections” tab
- Click “myWindow” and drag a line to anywhere in the Window
- Click “Duplicate It” and drag a line to the Button



Interface Builder

- Save all and click “Build and Go”
- Run “Hello_BMR” plugin



“Hello BMR” – Level 3 Objectives

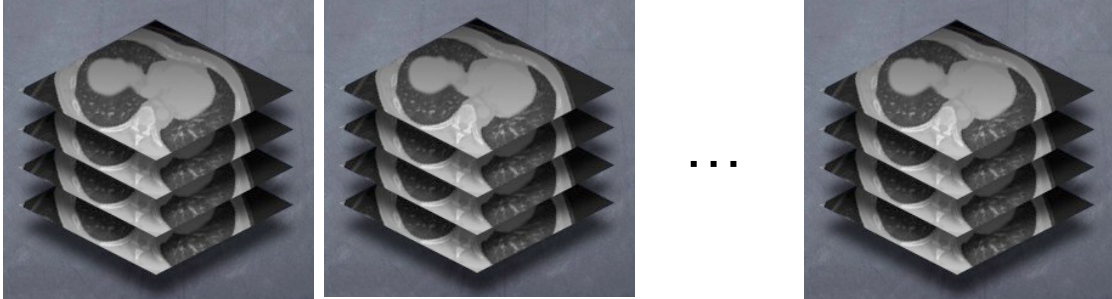
- Through this example (Level 3), you’re able to:
 - Understand some OsiriX objects
 - Create a new plugin using OsiriX objects

Here is a list of available objects in OsiriX:

- `PluginFilter`: your plugin is a sub-class of this object, it contains some useful functions
- `ViewerController`: `NSWindowController` of a '2D Viewer' window
- `DCMView`: `NSOpenGLView` that contains the displayed image
- `DCMPix`: object that contains the pixel data of an image
- `dicomFile`: an object that contains data about the DICOM file
- `ROI`: an object that contains a ROI
- `MyPoint`: an object that describes a 2D point

“Hello BMR” – Level 3

- 4D Viewer



- “pixList” selects one temporal point and returns as an NSArray
 - NSArray *pixList= [viewController pixList];

“Hello BMR” – Level 3

- Select a 2D image from NSArray using “objectAtIndex”



- DCMPix *curPix= [pixList objectAtIndex:i];

“Hello BMR” – Level 3

```
- (IBAction) doCalculation: (id)sender
{
    NSLog(@"doCalculation!!!");

    ViewerController *new2DViewer;
    // In this plugin, we will simply duplicate the current 2D window!
    new2DViewer = [self duplicateCurrent2DViewerWindow];

    int i, x, zSize;
    float *fImage, *fImageNew;
    NSArray *pixList, *pixListNew;
    DCMPix *curPix, *curPixNew;

    pixList = [viewerController pixList];
    pixListNew = [new2DViewer pixList];
    zSize = [pixList count];

    for (i = 0; i < zSize; i++)
    {
        curPix = [pixList objectAtIndex:i];
        curPixNew = [pixListNew objectAtIndex:i];

        // fImage is a pointer on the pixels, ALWAYS represented in float (float*) or in ARGB (unsigned char*)
        fImage = [curPix fImage];
        fImageNew = [curPixNew fImage];
        x = [curPixNew pheight] * [curPixNew pwidth];

        while ( x-- > 0 )
        {
            *fImageNew = ~(*fImage);
            fImage++;
            fImageNew++;
        }
    }

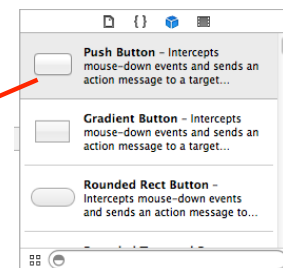
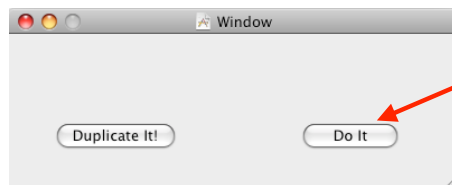
    // We modified the pixels: OsiriX please update the display!
    [new2DViewer needsDisplayUpdate];

    [myWindow close];
}
```

“Hello BMR” – Level 3

- In Hello_BMRFilter.h, add
- (IBAction) doCalculation: (id)sender;

- Create a new Button called “Do It”



- Connect the Button with “doCalculation”
- Save all and “Build and Go”

Summary

- What have you done?
 - Created NSWindow & NSButtons
 - Connected them with IBOutlet & IBAction
 - Used OsiriX objects to get & push pixel-by-pixel values from the Viewer
- Source code for “Hello_BMR” is available..
http://www.stanford.edu/~kyungs/software/Hello_BMR.zip
- If you’d like to learn more, please read this..



Other Resources

- OsiriX development guide:
<http://osirix.svn.sourceforge.net/viewvc/osirix/Documentation/Guides/Development/index.html>
- Keynote presentation about how to start writing a plugin:
<http://www.osirix-viewer.com/OsiriXDevKeynote.html>
- Source Code documentation:
<http://osirix.svn.sourceforge.net/viewvc/osirix/Documentation/OsiriX/html/index.html>
- OsiriX developers group:
<http://tech.groups.yahoo.com/group/osirix-dev/>

Thank you!

Please send comments or questions
to Kyung Sung:

KSung@mednet.ucla.edu