CONTENT LOCALIZATION iOS OF BT 2.0 & 3.0 APPLICATIONS

STEP 1.

CREATE TWO SEPARATE APPLICATIONS IN YOUR BUZZTOUCH OR BT-SERVER CONTROL PANEL.

My Fun App – Original
My Fun App Spanish

Create your content like normal of course varying the language for each app that you want localized.

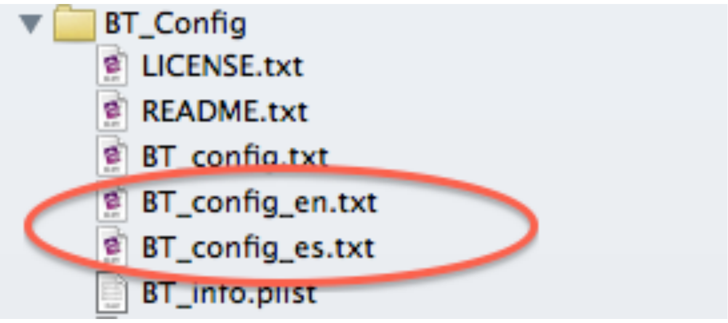
Copy the config data for both applications.

Open a text editor and copy the appropriate language into separate BT_config filenames save as the following.

BT_config_en.txt – My Fun App
BT_config_es.txt – My Fun App Spanish

Note: en & es represent each local language code for iO. For example _fr _de (French, German, etc). For correct codes check this document

Import these two files in your BT iOS project



```
▼ BT_Config
    LICENSE.txt
    README.txt
    BT_config.txt
    BT_config_en.txt
    BT_config_es.txt
    BT_info.plist
```

STEP 2.

Now we will add arrays to your app:

//For Buzztouch 2.0

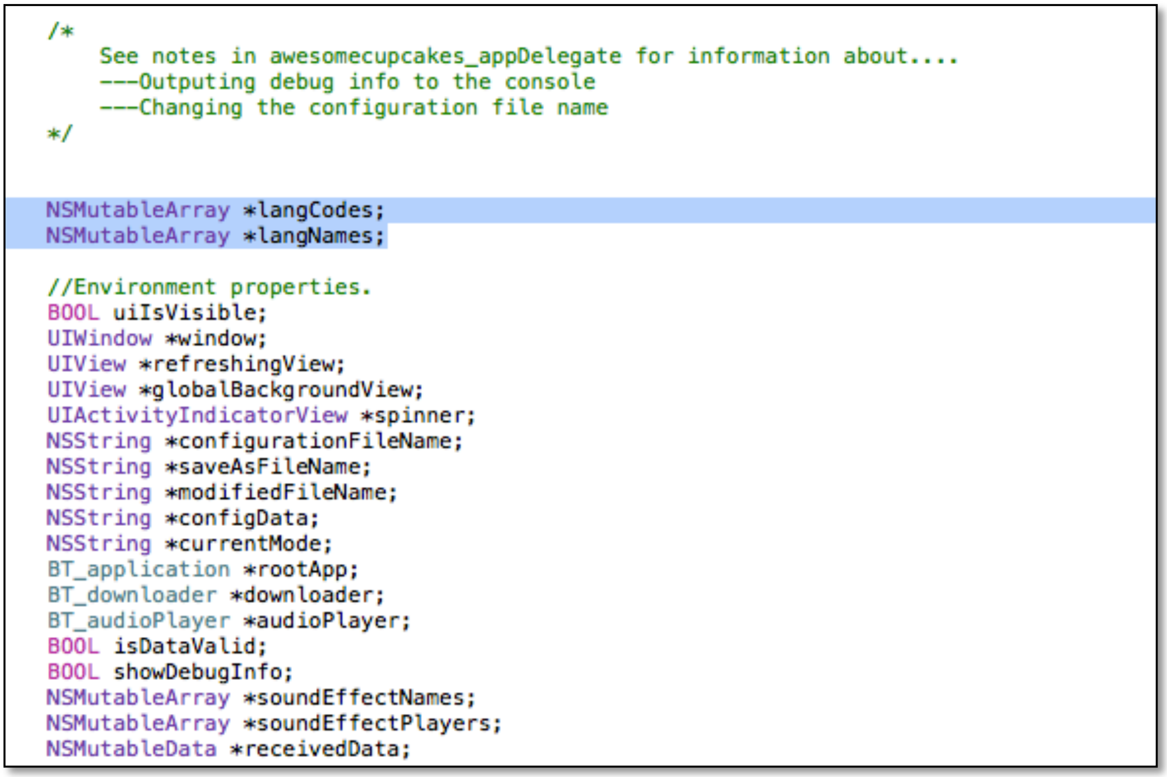
Add to: yourappsdelegate_appDelegate.h

//For Buzztouch 3.0

Added to: BT_Layout/BT_loadConfigDataViewController.h

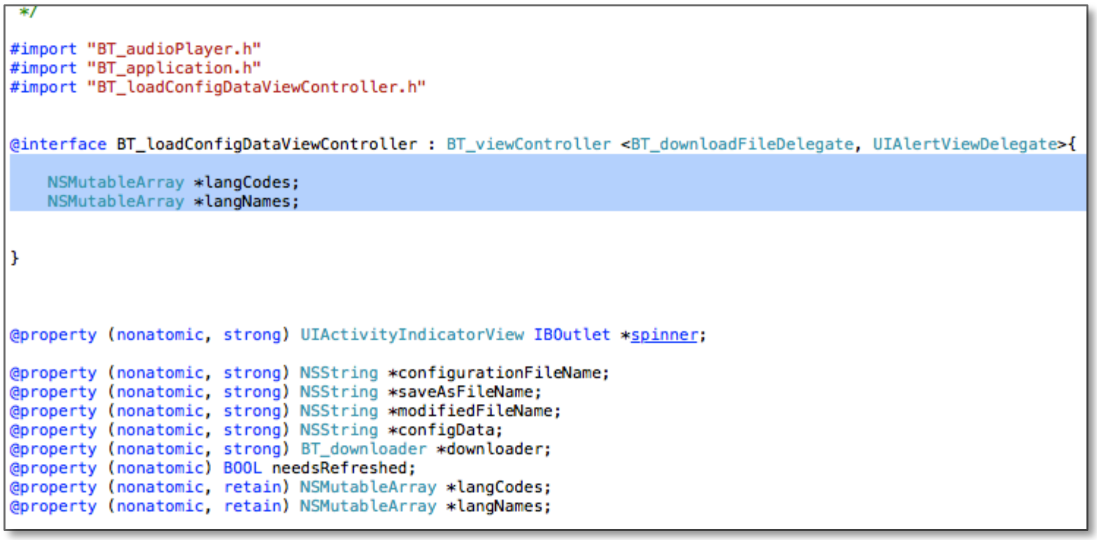
```
NSMutableArray *langCodes;
NSMutableArray *langNames;
```

Buzztouch 2.0 Screenshot



```
/*
    See notes in awesomecupcakes_appDelegate for information about....
    ---Outputing debug info to the console
    ---Changing the configuration file name
*/


NSMutableArray *langCodes;
NSMutableArray *langNames;

//Environment properties.
BOOL uiIsVisible;
UIWindow *window;
UIView *refreshingView;
UIView *globalBackgroundView;
UIActivityIndicatorView *spinner;
NSString *configurationFileName;
NSString *saveAsFileName;
NSString *modifiedFileName;
NSString *configData;
NSString *currentMode;
BT_application *rootApp;
BT_downloader *downloader;
BT_audioPlayer *audioPlayer;
BOOL isDataValid;
BOOL showDebugInfo;
NSMutableArray *soundEffectNames;
NSMutableArray *soundEffectPlayers;
NSMutableData *receivedData;
```

Buzztouch 3.0 Screenshot

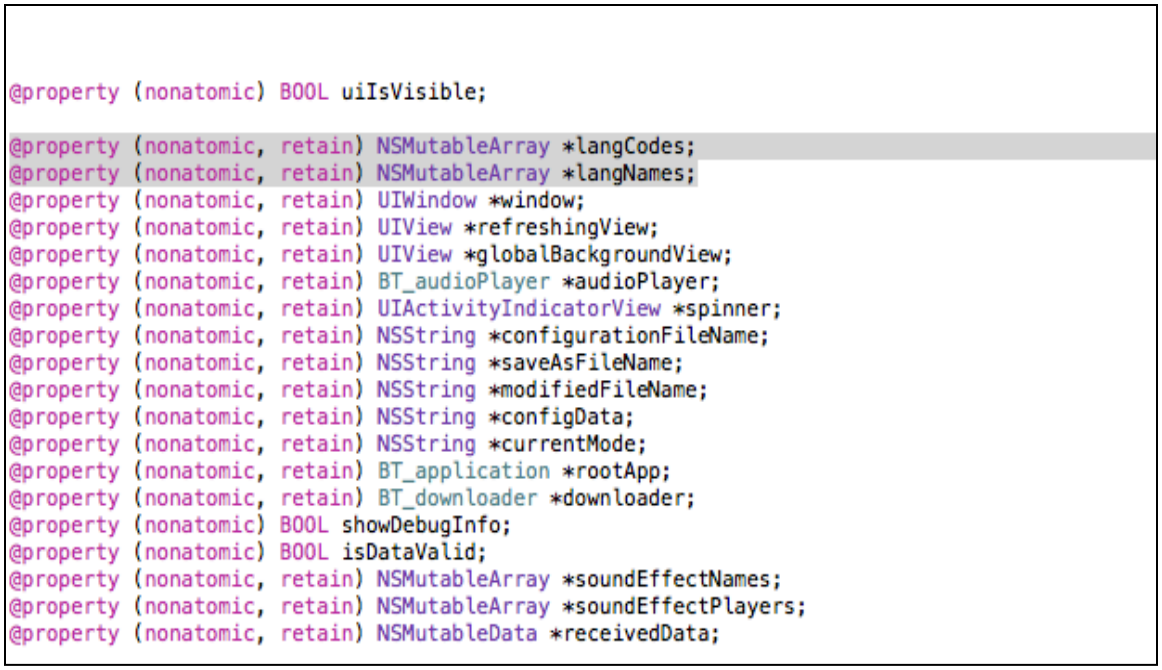


```
 */

#import "BT_audioPlayer.h"
#import "BT_application.h"
#import "BT_loadConfigDataViewController.h"


@interface BT_loadConfigDataViewController : BT_viewController <BT_downloadFileDelegate, UIAlertViewDelegate>{

    NSMutableArray *langCodes;
    NSMutableArray *langNames;


}



@property (nonatomic, strong) UIActivityIndicatorView IBOutlet *spinner;

@property (nonatomic, strong) NSString *configurationFileName;
@property (nonatomic, strong) NSString *saveAsFileName;
@property (nonatomic, strong) NSString *modifiedFileName;
@property (nonatomic, strong) NSString *configData;
@property (nonatomic, strong) BT_downloader *downloader;
@property (nonatomic) BOOL needsRefreshed;
@property (nonatomic, retain) NSMutableArray *langCodes;
@property (nonatomic, retain) NSMutableArray *langNames;
```

Next you will have to add the property to those arrays

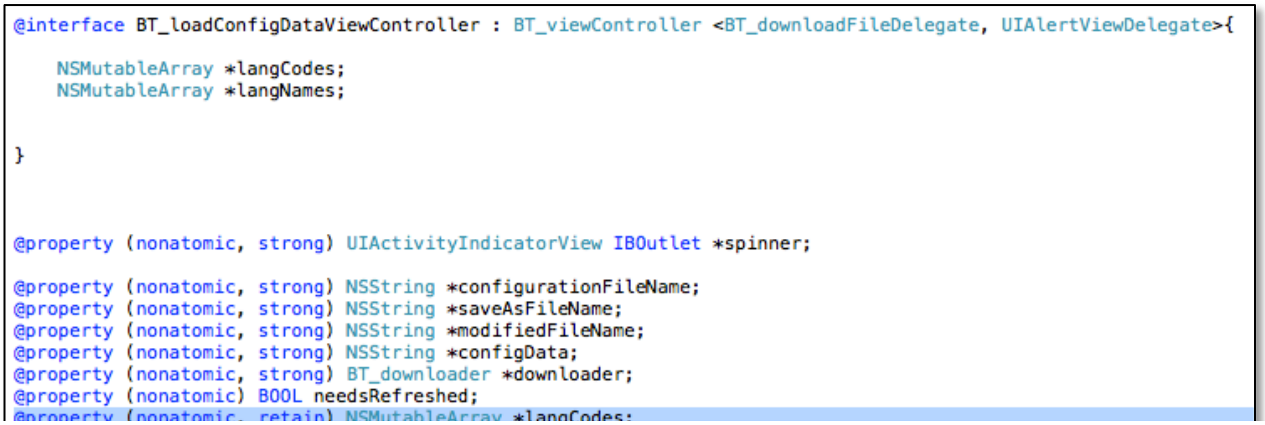
```
@property (nonatomic, retain) NSMutableArray *langCodes;
@property (nonatomic, retain) NSMutableArray *langNames;
```

Buzztouch 2.0 Screenshot



```
@property (nonatomic) BOOL uiIsVisible;

@property (nonatomic, retain) NSMutableArray *langCodes;
@property (nonatomic, retain) NSMutableArray *langNames;
@property (nonatomic, retain) UIWindow *window;
@property (nonatomic, retain) UIView *refreshingView;
@property (nonatomic, retain) UIView *globalBackgroundView;
@property (nonatomic, retain) BT_audioPlayer *audioPlayer;
@property (nonatomic, retain) UIActivityIndicatorView *spinner;
@property (nonatomic, retain) NSString *configurationFileName;
@property (nonatomic, retain) NSString *saveAsFileName;
@property (nonatomic, retain) NSString *modifiedFileName;
@property (nonatomic, retain) NSString *configData;
@property (nonatomic, retain) NSString *currentMode;
@property (nonatomic, retain) BT_application *rootApp;
@property (nonatomic, retain) BT_downloader *downloader;
@property (nonatomic) BOOL showDebugInfo;
@property (nonatomic) BOOL isDataValid;
@property (nonatomic, retain) NSMutableArray *soundEffectNames;
@property (nonatomic, retain) NSMutableArray *soundEffectPlayers;
@property (nonatomic, retain) NSMutableData *receivedData;
```

Buzztouch 3.0 Screenshot



```
@interface BT_loadConfigDataViewController : BT_viewController <BT_downloadFileDelegate, UIAlertViewDelegate>{

    NSMutableArray *langCodes;
    NSMutableArray *langNames;


}



@property (nonatomic, strong) UIActivityIndicatorView IBOutlet *spinner;

@property (nonatomic, strong) NSString *configurationFileName;
@property (nonatomic, strong) NSString *saveAsFileName;
@property (nonatomic, strong) NSString *modifiedFileName;
@property (nonatomic, strong) NSString *configData;
@property (nonatomic, strong) BT_downloader *downloader;
@property (nonatomic) BOOL needsRefreshed;
@property (nonatomic, retain) NSMutableArray *langCodes;
```

STEP 3.

Head over to your apps:

//For Buzztouch 2.0

Add to: yourappsdelegate_appDelegate.m

//For Buzztouch 3.0

Add to: BT_Layout/BT_loadConfigDataViewController.m

..and we will synthesize the properties from the header file you just edited.

Right below the last synthesized property where you see the sound effects add this line below it.

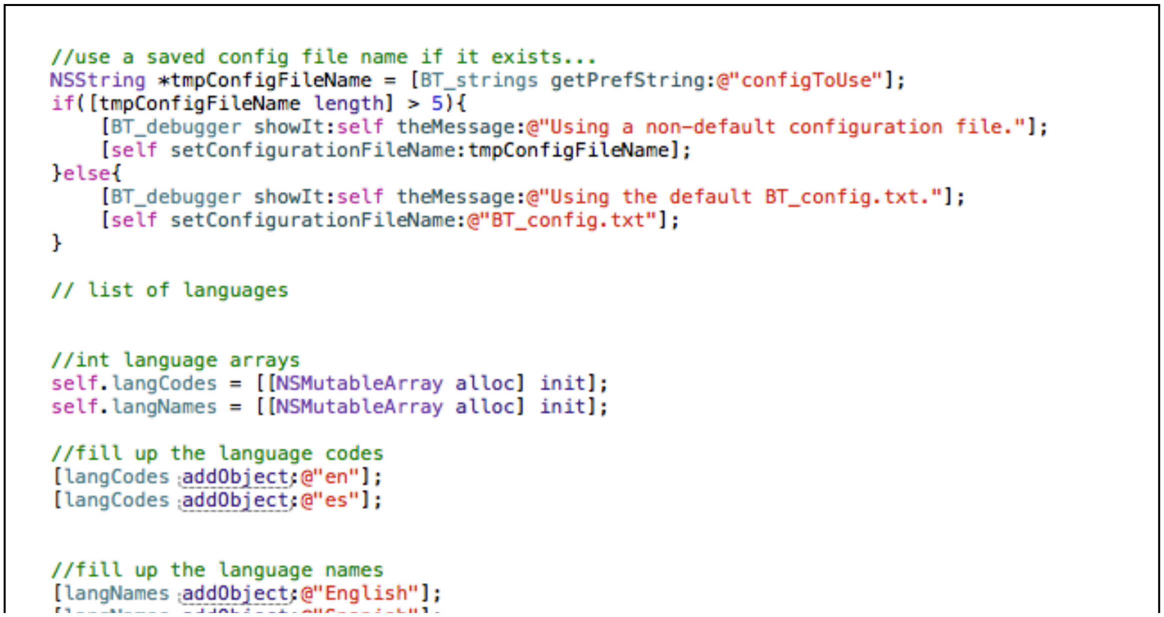
```
@synthesize langCodes, langNames;
```

Next you will build the list of languages by initializing the arrays that we added in the header file.

For BT 2.0
Copy and paste this code around line 89 right below the block of code starting with..

```
//use a saved config file name if it exists...
```

Buzztouch 2.0 Screenshot



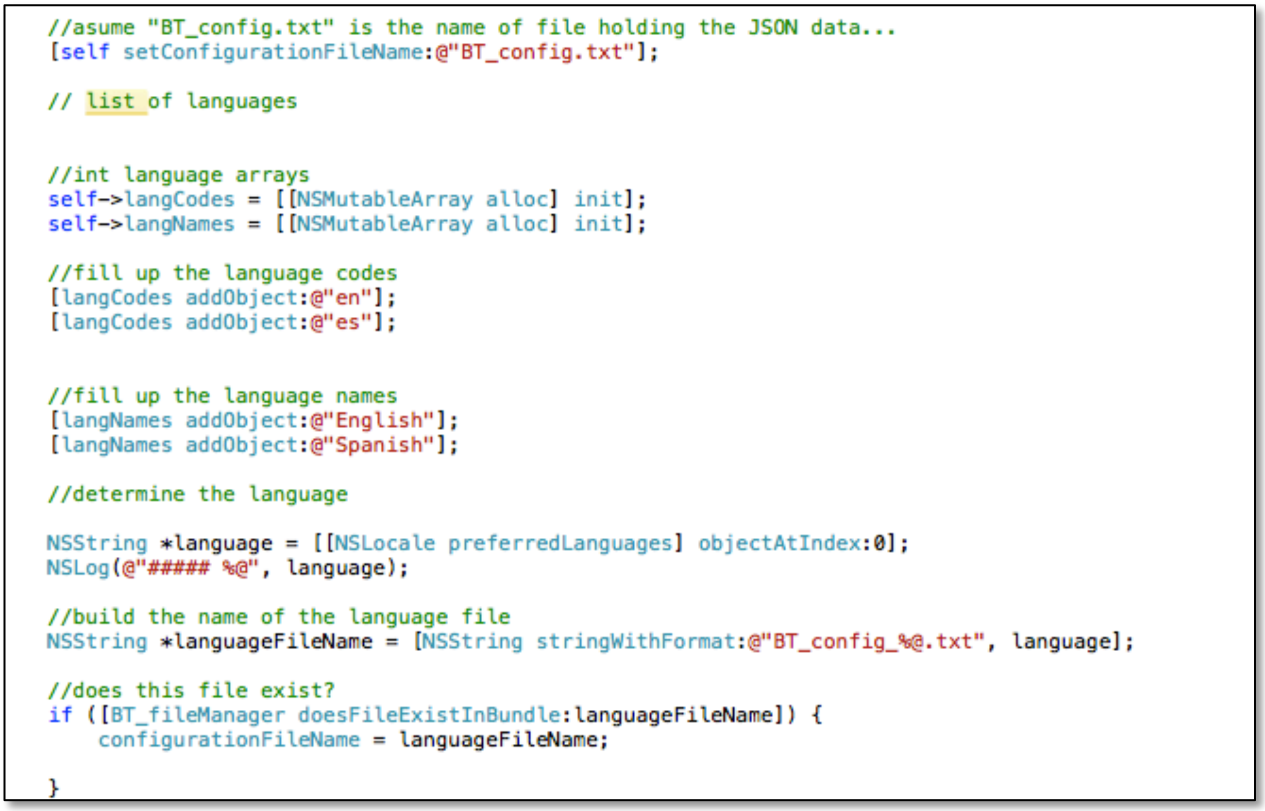
```
//use a saved config file name if it exists...
NSString *tmpConfigFileName = [BT_strings getPrefString:@"configToUse"];
if([tmpConfigFileName length] > 5){
    [BT_debugger showIt:self theMessage:@"Using a non-default configuration file."];
    [self setConfigurationFileName:tmpConfigFileName];
}else{
    [BT_debugger showIt:self theMessage:@"Using the default BT_config.txt."];
    [self setConfigurationFileName:@"BT_config.txt"];
}

// list of languages


//int language arrays
self.langCodes = [[NSMutableArray alloc] init];
self.langNames = [[NSMutableArray alloc] init];

//fill up the language codes
[langCodes addObject:@"en"];
[langCodes addObject:@"es"];


//fill up the language names
[langNames addObject:@"English"];
[langNames addObject:@"Spanish"];
```

For BT 3.0

Copy and paste this code around line 117 right below the block of code starting with..

```
//asume "BT_config.txt" is the name of file holding the JSON data...
```

Buzztouch 3.0 Screenshot



```
//asume "BT_config.txt" is the name of file holding the JSON data...
[self setConfigurationFileName:@"BT_config.txt"];

// list of languages


//int language arrays
self->langCodes = [[NSMutableArray alloc] init];
self->langNames = [[NSMutableArray alloc] init];

//fill up the language codes
[langCodes addObject:@"en"];
[langCodes addObject:@"es"];


//fill up the language names
[langNames addObject:@"English"];
[langNames addObject:@"Spanish"];

//determine the language

NSString *language = [[NSLocale preferredLanguages] objectAtIndex:0];
NSLog(@"##### %@", language);

//build the name of the language file
NSString *languageFileName = [NSString stringWithFormat:@"BT_config_%@.txt", language];

//does this file exist?
if ([BT_fileManager doesFileExistInBundle:languageFileName]) {
    configurationFileName = languageFileName;

}
```

```
// list of languages


//int language arrays
self.langCodes = [[NSMutableArray alloc] init];
self.langNames = [[NSMutableArray alloc] init];

//fill up the language codes
[langCodes addObject:@"en"];
[langCodes addObject:@"es"];


//fill up the language names
[langNames addObject:@"English"];
[langNames addObject:@"Spanish"];
```

Next we will add code to determine the language of the localized files once loaded in the app.

```
NSLog(@"##### %@", language);
```

With this block of code every time you change your device(s) language on the simulator it will show in the log that you are determining the apps local.

```
2013-04-07 14:46:22.908 m [27089:c07] myfunapp_appDelegate: Using the
default BT_config.txt.
2013-04-07 14:46:22.910 myfunapp[27089:c07] ##### es
```

In the log above I had already changed my simulators language to Spanish and after I rebuilt the application determined that the app should look for the language code “es”.

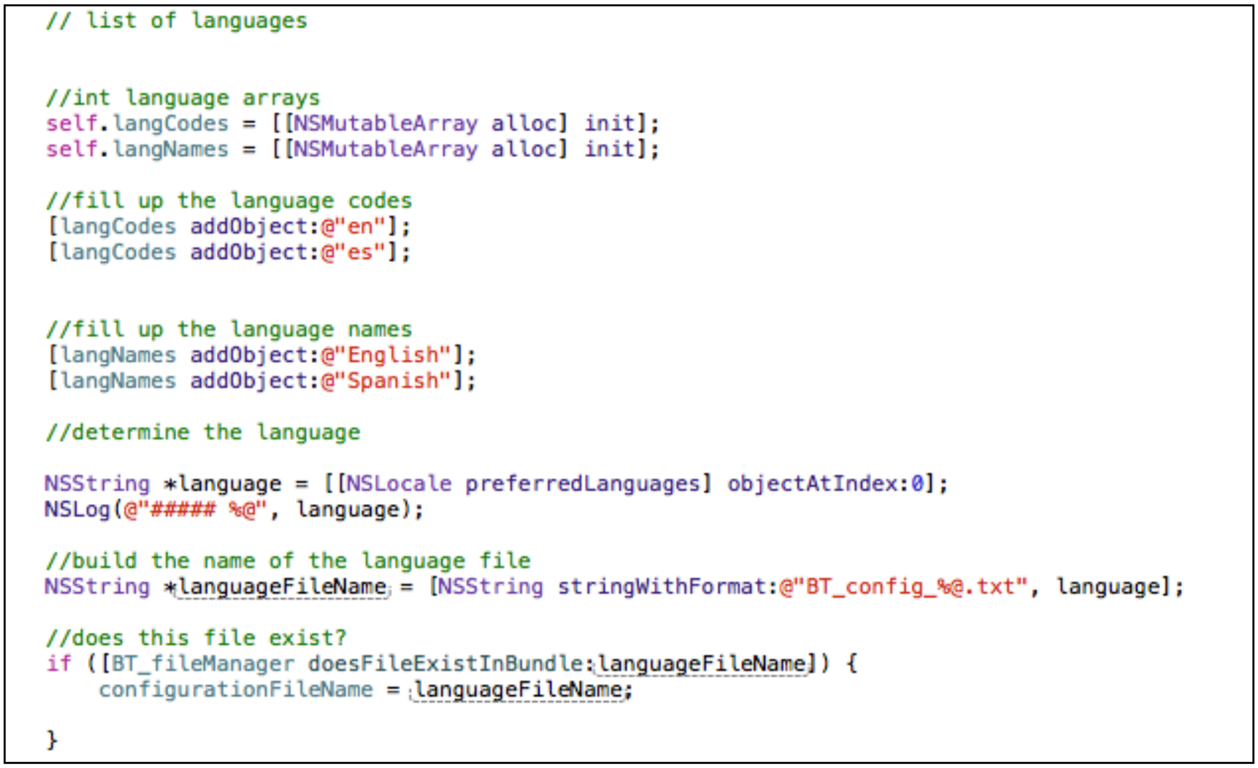
STEP 4.

Next we will build the name of each config file that you’ve added. In this example I’ve added BT_config_en.txt and BT_config_es.txt

So below this is telling xCode to build the language file using the following code.

Added directly below the code you added above.

```
//build the name of the language file
    NSString *languageFileName = [NSString
stringWithFormat:@"BT_config_%@.txt", language];

//does this file exist?
if ([BT_fileManager doesFileExistInBundle:languageFileName]) {
    configurationFileName = languageFileName;

    }
```



```
// list of languages


//int language arrays
self.langCodes = [[NSMutableArray alloc] init];
self.langNames = [[NSMutableArray alloc] init];

//fill up the language codes
[langCodes addObject:@"en"];
[langCodes addObject:@"es"];


//fill up the language names
[langNames addObject:@"English"];
[langNames addObject:@"Spanish"];

//determine the language

NSString *language = [[NSLocale preferredLanguages] objectAtIndex:0];
NSLog(@"##### %@", language);

//build the name of the language file
NSString *languageFileName = [NSString stringWithFormat:@"BT_config_%@.txt", language];

//does this file exist?
if ([BT_fileManager doesFileExistInBundle:languageFileName]) {
    configurationFileName = languageFileName;

}
```

That is it!

STEP 6.

Your files should load appropriately by change your similar or device language by going to

Settings - General – International – Language

Switch back and forth and you'll see it the changes or differently loaded config files.

In this examples case: English and Spanish!!

I have added a cheat sheet to make it easy to copy and paste.

This is based off of webinar done by David – Localizing BT App PART 2 – Minute 8:00 ..

Happy Coding