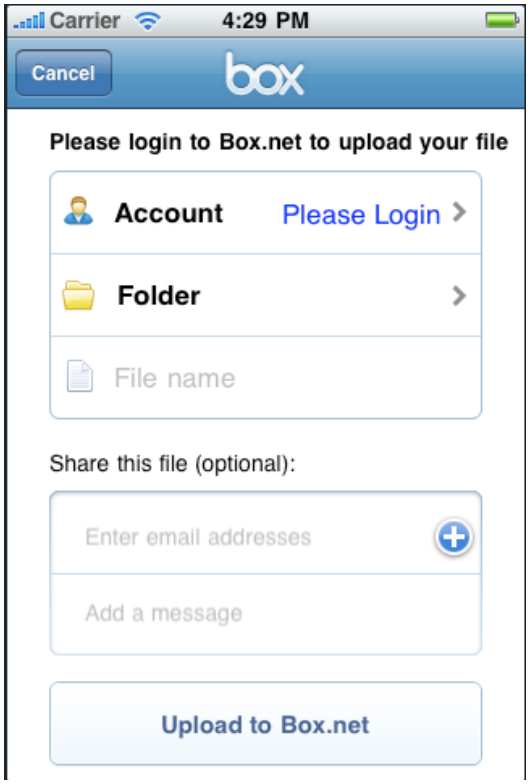
Box Upload Plugin Instructions

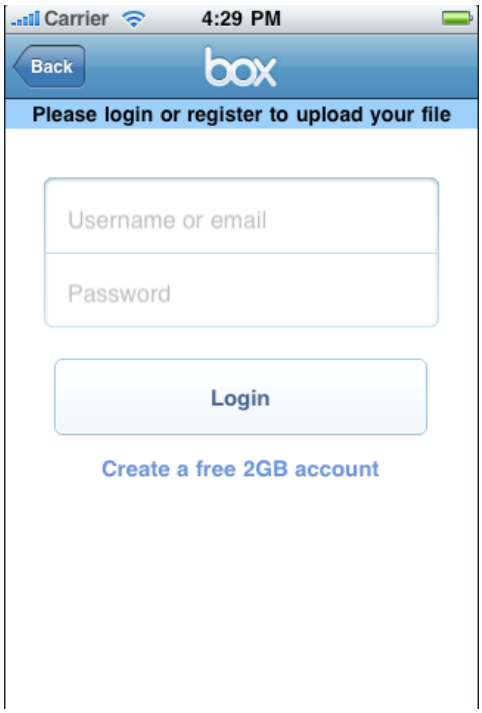
The Box Upload Plugin provides a UI to easily upload files to box.net and it can be easily integrated into existing applications. This document will give an overview of the Upload Plugin functionality as well as a description of how to integrate it into an existing application.

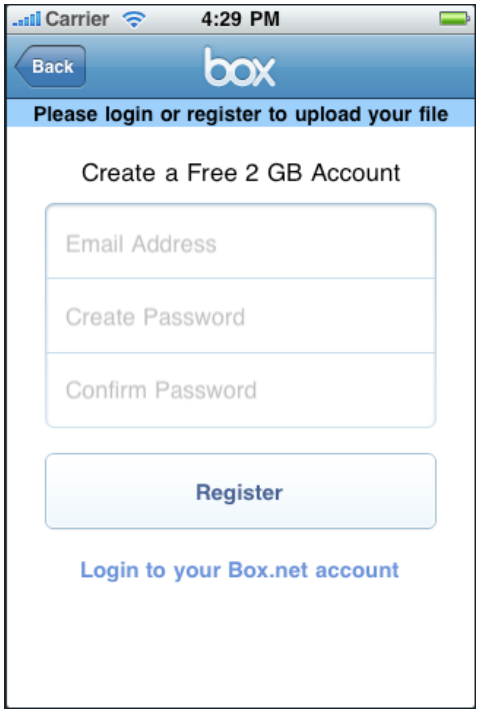
Overview

The Box plugin allows a user to register or login to their box account, select a folder from their “All Files” folder, name the file and share the file. The popup also saves states between application loads so important information such as username and selected folder only needs to be entered once.

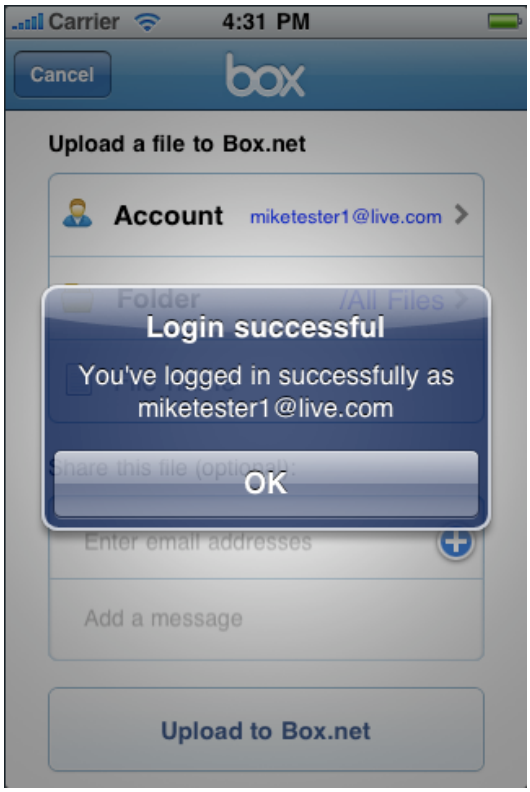
The uploader can be pulled up from any action in an application, for example, after the user has created a voice recording, or a new text document. The first page they are presented with gives them the option to login.



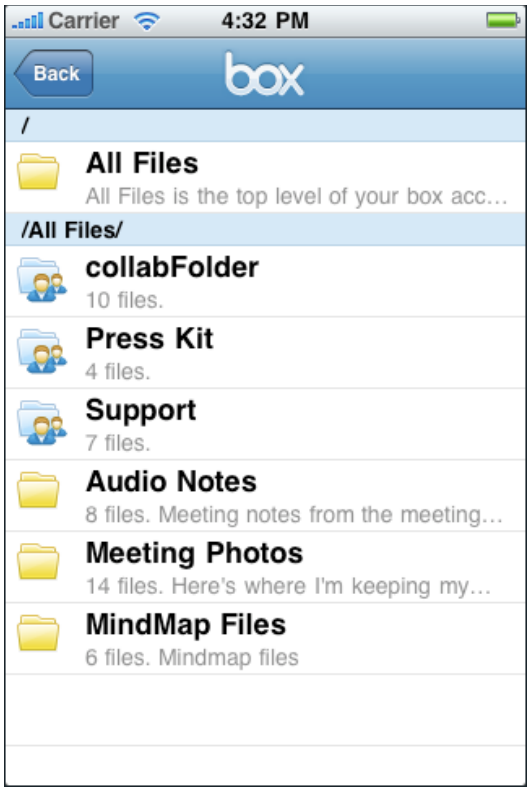




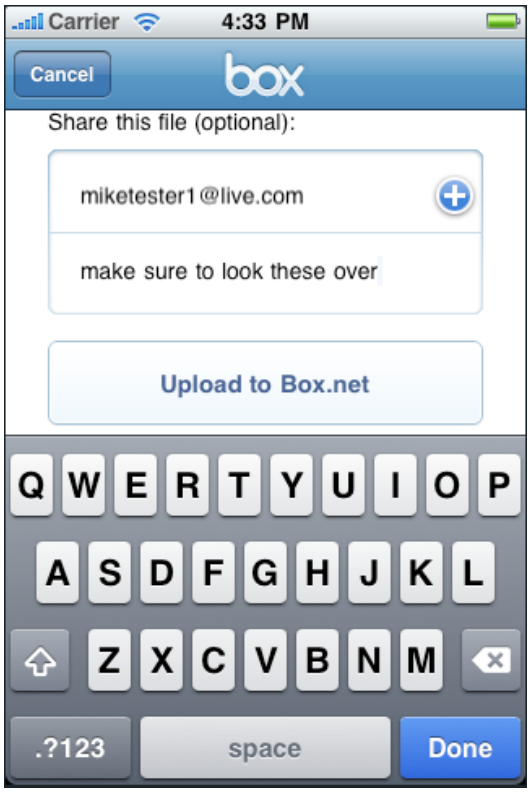
Both login and registration are possible.

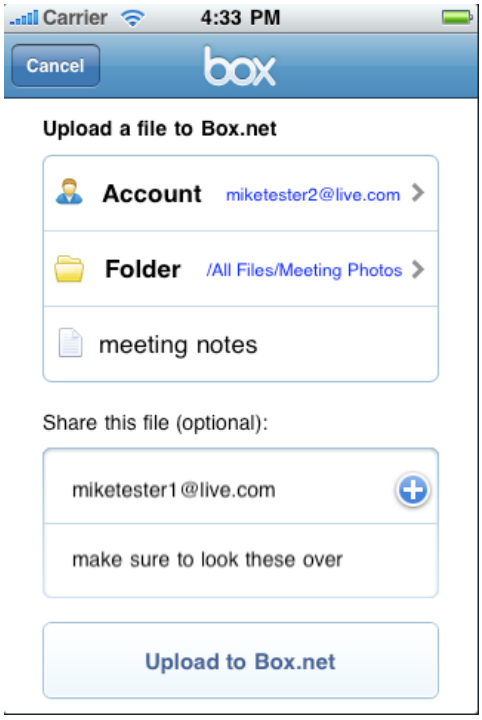


Once the user has logged in, they can choose a folder to upload to

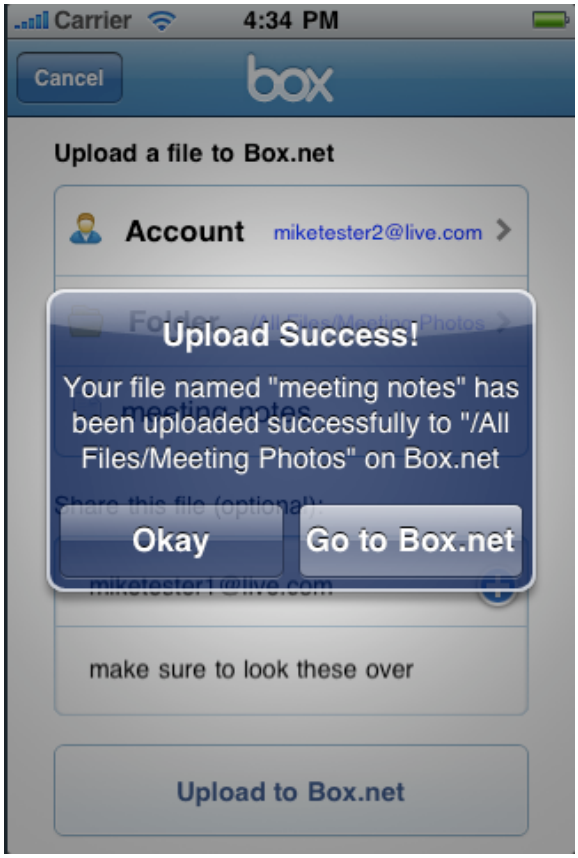


They can then choose to share the file with contacts from their address book enter them manually, and enter a message.


Carrier
4:33 PM
Cancel
box
Share this file (optional):
miketester1@live.com
make sure to look these over
Upload to Box.net
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
.?123
space
Done

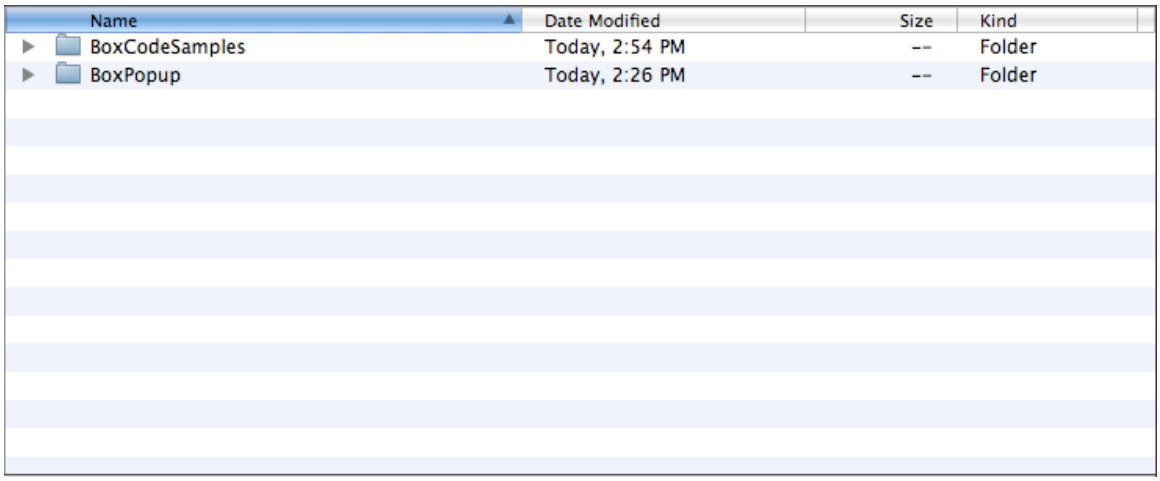

Carrier
4:33 PM
Cancel
box
Upload a file to Box.net
Account
miketester2@live.com
Folder
/All Files/Meeting Photos
meeting notes
Share this file (optional):
miketester1@live.com
make sure to look these over
Upload to Box.net

Once the information has been entered, the user can hit upload and the file gets uploaded to box and shared with anyone who's interested. The user can then choose to return to the application by hitting okay or see the file in a webview in the application on m.box.net.

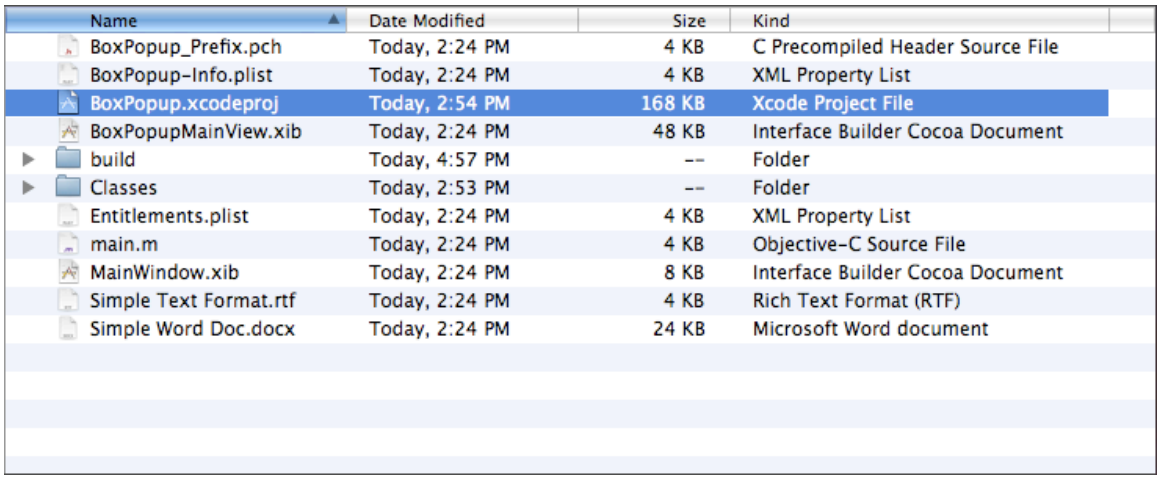


Integration with existing applications

To integrate the box uploader into your application, first download the Objective-C code contained in "Box Popup and Sample Classes.zip." Unzip the folder. "Box Popup and Sample Classes" contains two projects – one called "BoxCodeSamples" the other called "BoxPopup." "BoxCodeSamples" contains the models and parsing code for retrieving data remotely through the OpenBox Api. The BoxPopup project contains the GUI code for the popup and is based on the classes contained in "BoxCodeSamples."

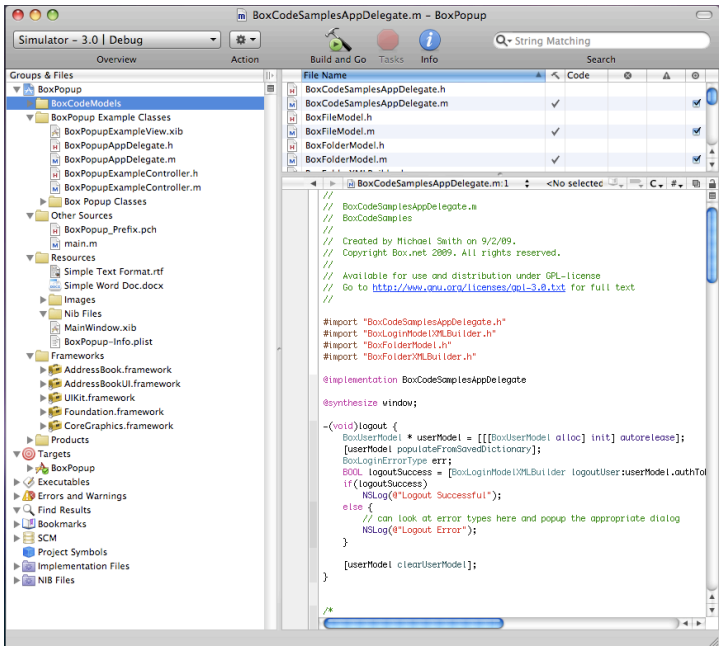


Name	Date Modified	Size	Kind
BoxCodeSamples	Today, 2:54 PM	--	Folder
BoxPopup	Today, 2:26 PM	--	Folder



Name	Date Modified	Size	Kind
BoxPopup_Prefix.pch	Today, 2:24 PM	4 KB	C Precompiled Header Source File
BoxPopup-Info.plist	Today, 2:24 PM	4 KB	XML Property List
BoxPopup.xcodeproj	Today, 2:54 PM	168 KB	Xcode Project File
BoxPopupMainView.xib	Today, 2:24 PM	48 KB	Interface Builder Cocoa Document
build	Today, 4:57 PM	--	Folder
Classes	Today, 2:53 PM	--	Folder
Entitlements.plist	Today, 2:24 PM	4 KB	XML Property List
main.m	Today, 2:24 PM	4 KB	Objective-C Source File
MainWindow.xib	Today, 2:24 PM	8 KB	Interface Builder Cocoa Document
Simple Text Format.rtf	Today, 2:24 PM	4 KB	Rich Text Format (RTF)
Simple Word Doc.docx	Today, 2:24 PM	24 KB	Microsoft Word document

To see how the sample application works open up the xcode project BoxPopup.xcodeproj.



The project contains two main code folders – “BoxModelAndNetworkCode” and “BoxPopupControllerUICode.”

“BoxModelAndNetworkCode” contains all of the models and network code for the application. “BoxPopupControllerUICode” contains all of the view and controller code.

The example given in this project is contained in “BoxPopupExampleController.h”. This class implements the BoxPopupDelegate protocol which contains 5 functions:

```
@protocol BoxPopupDelegate

@required
-(NSString*)suggestedFileName;
-(BOOL)shouldAppendTimeAndDate;
-(NSData*)data;
-(NSString*)fileExtension;
-(void)popupShouldExit;

@end
```

Check out the file Classes/BoxPopupControllerUICode/PopupControllers/BoxPopupDelegate.h for a full explanation of each of these classes.

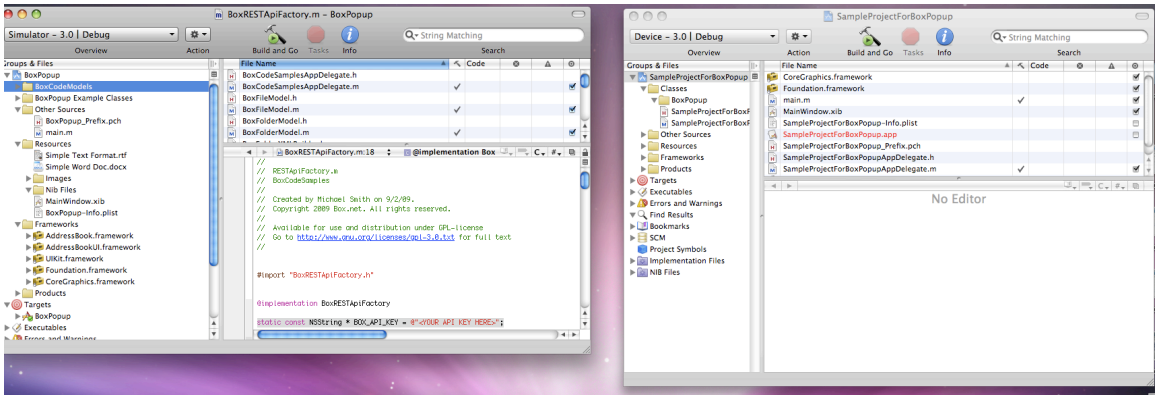
BoxPopupExampleController has a UIView with 5 buttons in it, each demonstrating different possible combinations of return values for the interface. You can build and run the project to see what these different options do.

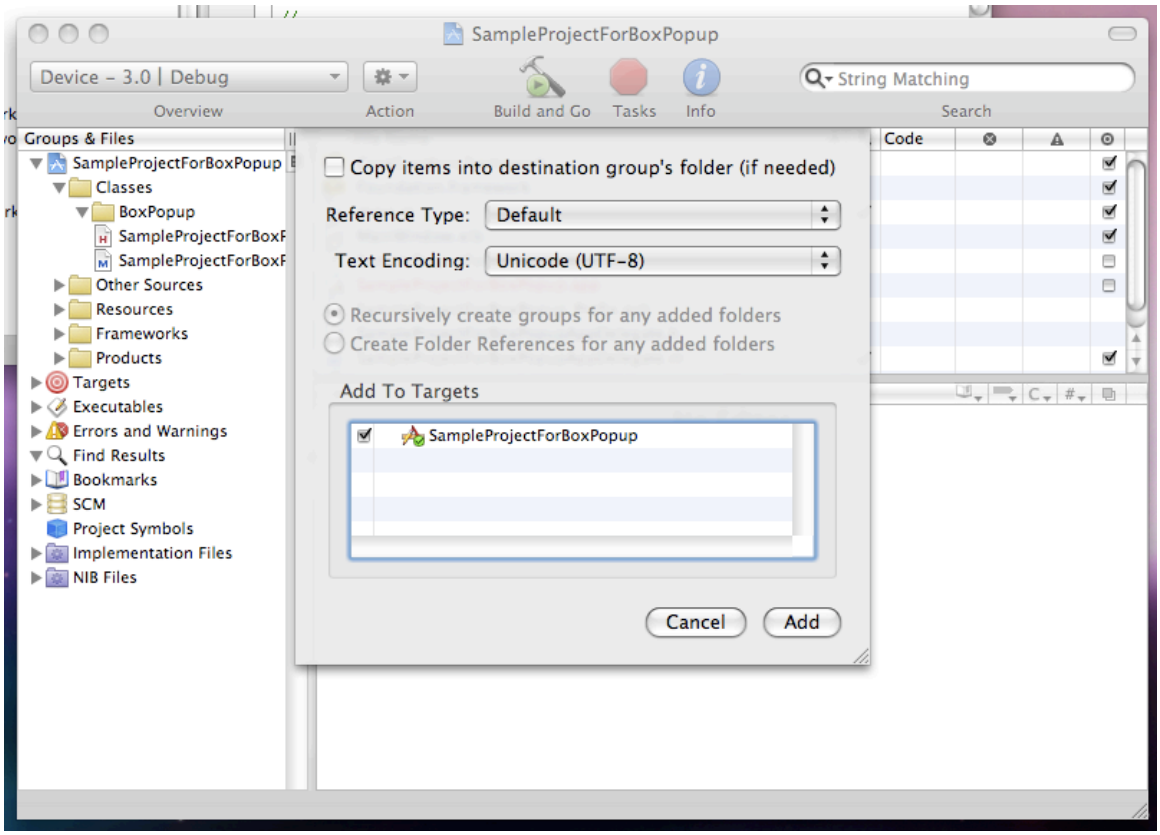
Before you can log in however, you have to make sure that you place your developer key in the file: BoxCodeModels/Models/Model Builders/Utilities/BoxRESTApiFactory.m line 18. Alternatively you can do a search for:

```
static const NSString * BOX_API_KEY = @"<YOUR API KEY HERE>";
```

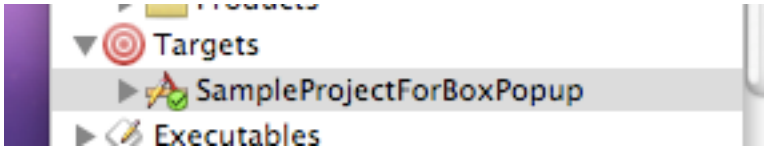
and insert your key in place of `<YOUR API KEY HERE>`.

To integrate the code into your own project, simply drag and drop the BoxModelAndNetworkCode and the BoxPopupControllerUICode logical folders into your project. You can optionally check the "Copy items into destination group's folder (if needed)" check box. More updates will be coming to this popup soon, so it may be easier to leave the check box blank and simply replace it as newer versions of the BoxPopup come out.

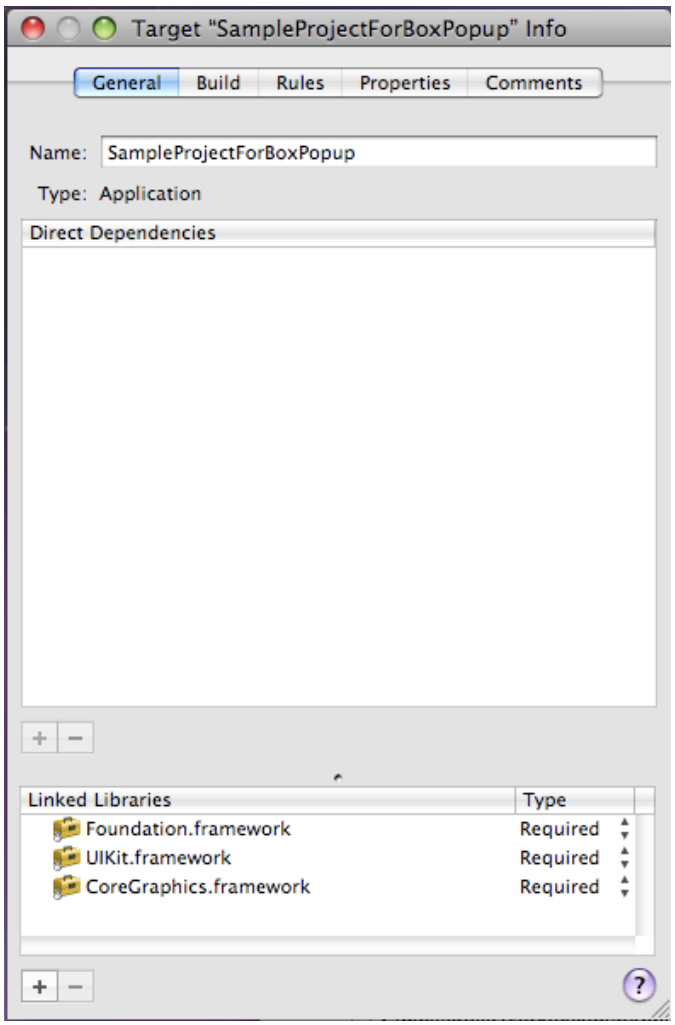




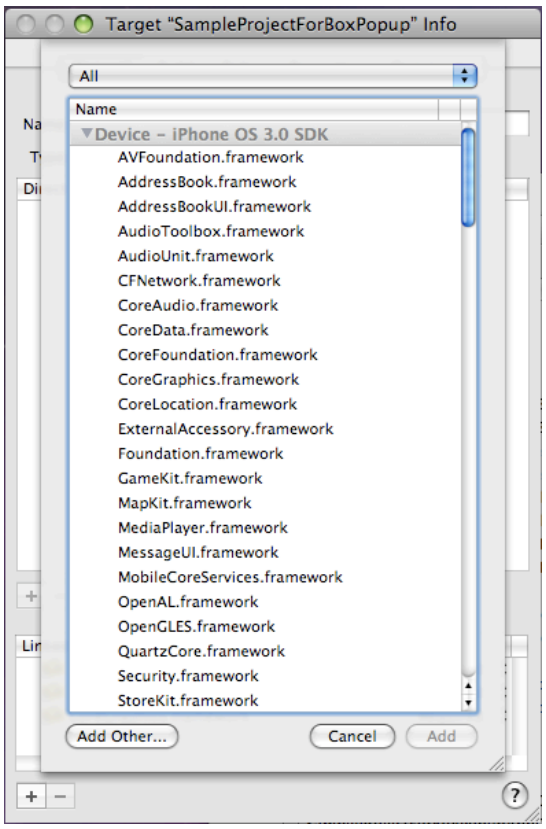
Make sure that your project contains the AddressBook.framework and AddressBookUI.framework frameworks. To add these, to your target, double click on your target:



Go to the “General” tab



Click the plus button at the bottom left of the screen, under “Linked Libraries.”



And select both AddressBook.framework and AddressBookUI.framework.

To popup the Box Uploader, follow the example given in "BoxPopupExampleController.h"

In some action:

```
-(IBAction) popupBoxAction:(id)sender;
```

```
    // Initialize the Popup controller, set the delegate and present it
as a modal view controller, then release.
    _boxPopupController = [[BoxPopupController alloc]
initWithNibName:@"BoxPopupController" bundle:nil];
    _boxPopupController.popupDelegate = self;
    [self presentModalViewController:_boxPopupController animated:YES];
      [_boxPopupController release];
```

Make sure the class implements the BoxPopupDelegate

```
@interface BoxPopupExampleController : UIViewController
<BoxPopupDelegate>
```

and the delegate methods:

```
-(NSData*)data {

      NSData * data;

      <…>

      return data;

}


-(NSString*)fileExtension {

      return @"gif";

}


-(void)popupShouldExit {
    [self dismissModalViewControllerAnimated:YES];
}


-(NSString*)suggestedFileName {

      return @"SampleFile";

}


-(BOOL)shouldAppendTimeAndDate {

      return YES;

}
```

That should allow you to integrate the Box popup into your application and get your users seamlessly uploading content to the web.