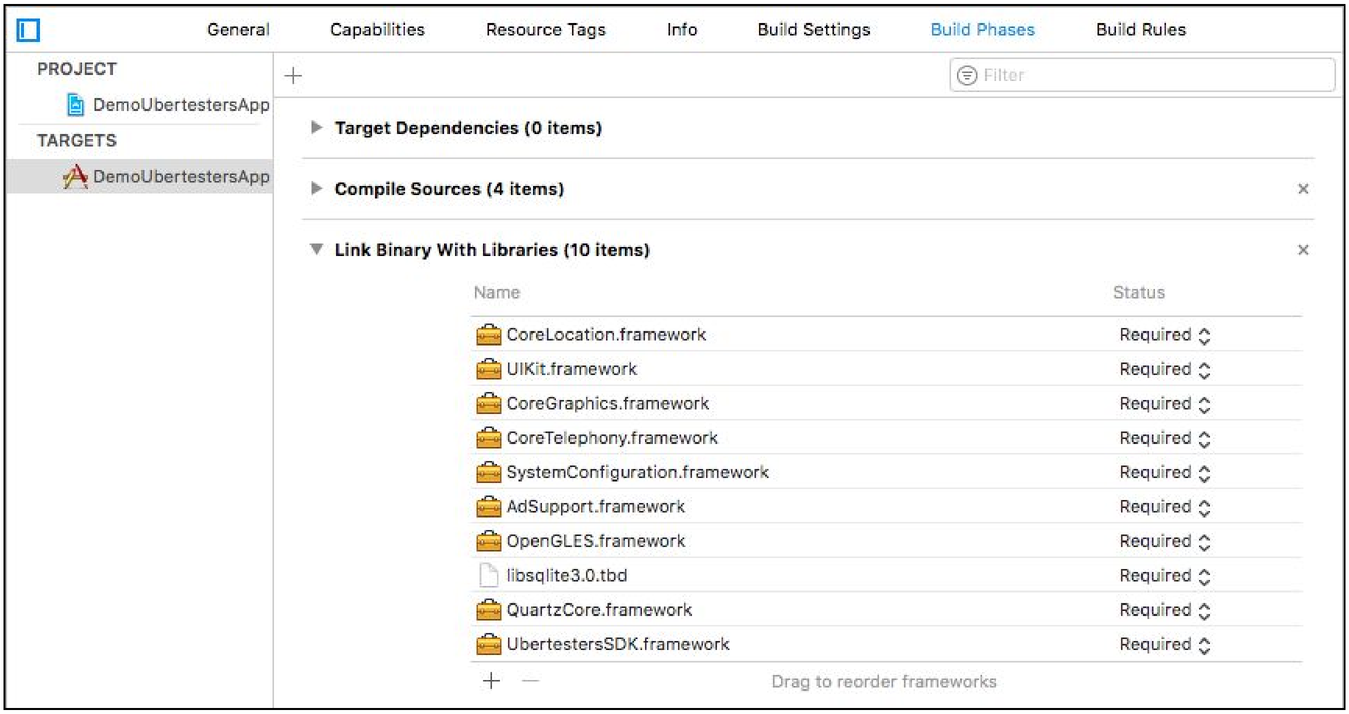
Integration guide for iOS Ubertesters SDK (Swift)

Step 1. Download the latest version of Ubertesters SDK for iOS

1. Download [ubertesters.sdk.ios.zip]() and unzip it.
2. Add UbertestersSDK.framework to your project.

Step 2. Add frameworks required by Ubertesters framework

1. Go to **Build Phases -> Link Binary With Libraries**.
2. Add every of next frameworks (if not added):
    - UIKit.framework
    - CoreGraphics.framework
    - CoreTelephony.framework
    - SystemConfiguration.framework
    - CoreLocation.framework
    - AdSupport.framework
    - OpenGLES.framework
    - QuartzCore.framework
    - libsqlite3.0.tbd



▶ **Target Dependencies (0 items)**

▶ **Compile Sources (4 items)**

▼ **Link Binary With Libraries (10 items)**

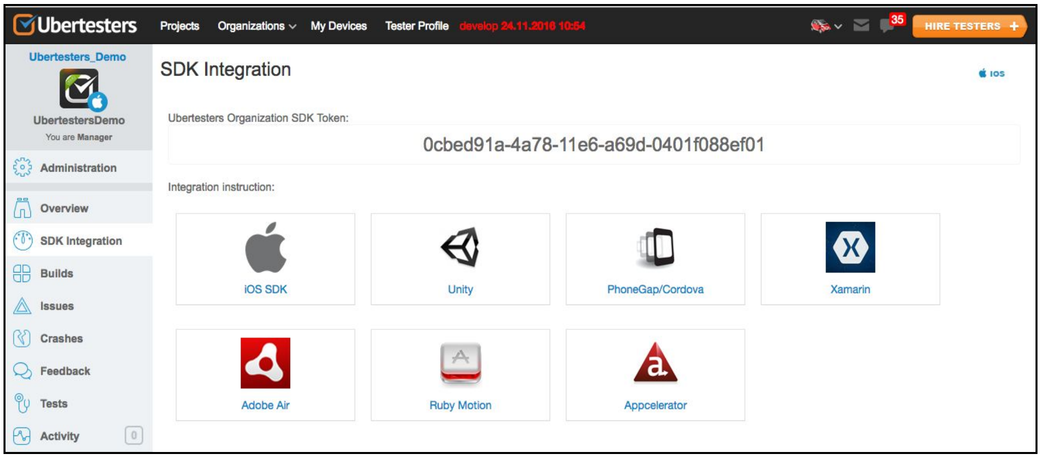
Name	Status
CoreLocation.framework	Required
UIKit.framework	Required
CoreGraphics.framework	Required
CoreTelephony.framework	Required
SystemConfiguration.framework	Required
AdSupport.framework	Required
OpenGLES.framework	Required
libsqlite3.0.tbd	Required
QuartzCore.framework	Required
UbertestersSDK.framework	Required

Step 3. Set up Xcode project

- In Xcode right click on info.plist file and select **Open As** -> **Source Code**.
- Add following lines of code.

```
<key>ubertesters_organization_token</key>
    <string>{your_ubertesters_organization_sdk_token}</string>
<key>LSApplicationQueriesSchemes</key>
    <array>
        <string>ubertesters</string>
    </array>
<key>NSPhotoLibraryUsageDescription</key>
    <string>To use photos as issue attachments.</string>
```

- Open http://beta.ubertesters.com/projects and select your project, go to SDK Integration.
- Copy “Ubertesters Organization SDK Token” from SDK Integration page and replace {your_ubertesters_organization_sdk_token} in the code with it.



Ubertesters Organization SDK Token:
0cbed91a-4a78-11e6-a69d-0401f088ef01

.

Step 4. Modify your Application Delegate file

1. Import Ubertesters to your project *-Bridging-Header.h file. Here is an article that describes what the “Bridging Header ” is.

```
#import <UbertestersSDK/Ubertesters.h>
```

2. Initialize Ubertesters SDK in your app delegate file.

```
func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[UIApplicationLaunchOptionsKey: Any]?) -> Bool {
    Ubertesters.shared().initialize()
    return true
}
```

Additional setup

Initialize Ubertesters with different options

It is possible to initialize Ubertesters in 2 ways:

1. Using initializer with the default parameters. When you are using the initializer with the default parameters – UbertestersActivationModeWidget for activation mode and UbertestersLockingModeDisableUbertesters for locking mode:

```
Ubertesters.shared().initialize()
```

2. Using customized initializer

```
Ubertesters.shared().initialize(with:
      UbertestersOptions(UbertestersActivationModeWidget.rawValue |
      UbertestersActivationModeWidget.rawValue))
```

Ubertesters options

The customized initializer is used with the Ubertesters SDK options which determine the behavior of Ubertesters widget (with the help of which you call the bug submission screen) and the behavior of the Ubertesters app when the build/app is not available in Ubertesters system. According to their functions, all the options can be divided in two groups – activation mode options and locking mode options.

You can use the following options for the customized initializer:

Activation mode options

UbertestersActivationModeWidget	The Ubertesters widget (an orange bubble) is shown, and you need to click on it to call a bug submission screen.
UbertestersActivationModeShake	The widget is not shown, you need to shake your phone to call a menu for a bug submission or take a screenshot.
UbertestersActivationModeManual	The Ubertesters widget is not shown, a menu should be called with the help of our API methods. You can specify the conditions under which the Ubertesters menu will be shown.

Locking mode options

UbertestersLockingModeDisableUbertesters	This option will not lock your application if the build is not uploaded on our server
UbertestersLockingModeLockApplication	Ubertesters SDK locks the app completely if the build is not located on our server, it is impossible to navigate through the app

Important

1. Ubertesters SDK is used for beta-testing. Please make sure you’ve removed Ubertesters SDK before publishing your app to App Store.
2. Make sure you are using Apple LLVN compiler 4.2 and greater. Go to BuildSettings of your target and find BuildOptions. There will be row Compiler for C/C++/Objective-C.

Ubertesters SDK API methods

API methods allow the developers to call additional methods and customize our SDK according to their needs.

Remote Logging

To perform remote logging you can use the UTLog method. UT log is a function with a text and a log level. **Remote logging** allows you to send an important system event information to Ubertesters server dedicating to storing and archiving this data. You can check the sent logs in your Ubertesters account in **Activity > Session > Session Tracking**

UTLog methods

UTLogLevelInfo	Sends info logs
UTLogLevelWar	Sends warning logs
UTLogLevelError	Sends error logs

To add UTLog, please use the following line:

```
Ubertesters.shared().utLog("text message", with: UTLogLevelInfo)
```

API methods used with UbertestersActivationModeManual

These methods allow to specify the behavior of Ubertesters bug submission screen in case Manual mode is used.

Method	Action	Example
- (void)makeScreenshot;	Captures screenshot in the app you are testing.	Ubertesters.shared().makeScreenshot()
- (void)showMenu;	Shows Ubertesters menu.	Ubertesters.shared().showMenu()
- (void)hideMenu;	Shows Ubertesters menu.	Ubertesters.shared().hideMenu()

Disable Crash Handler

Method	Action
- (void)disableCrashHandler;	Disables Ubertesters crash handler. Allows using our platform along with third party crash reporters.

```
Ubertesters.shared().disableCrashHandler()
```