“Adventures in Apple Watch”

or

Using Appcelerator Titanium to create an Apple Watch application

*(PS: DON’T forget to add the **TEAM ID** to the tiapp.xml)*

Invalid Swift Support - The SwiftSupport folder is missing

Work at MATCHESFASHION.COM London, UK

Help build the fastest growing luxury fashion eCommerce platform in the world!

javascript html

3

2

I have updated a currently submitted Titanium app and added a watch extension using swift.

Everything works fine if I build and test on sim and build directly to device. I only get an issue when I submit the app to the Apple app store (via XCode Organizer).

The binary submits, passing validation but I get am email from iTunes Connect as follows:

> Dear developer,
>
> We have discovered one or more issues with your recent delivery for "xxxxxxxxxxx". To process your delivery, the following issues must be corrected:
>
> Invalid Swift Support - The SwiftSupport folder is missing. Rebuild your app using the current public (GM) version of Xcode and resubmit it.
>
> Once these issues have been corrected, you can then redeliver the corrected binary.

It seems as though it may be related to a build setting: *Embedded Content Contains Swift Code*.

It looks like this needs to be set to **Yes** if the Titanium project contains embedded Swift.

As of now I am stuck as I cannot submit the app. Is this a Ti problem or is there another step I should follow?

XCode: 7.3, SDK: 5.2.2.GA - Project created and built using only the Ti CLI.

swift itunesconnect appcelerator apple-watch appcelerator-titanium

share edit delete flag

asked Apr 25 '16 at 6:33

Steve

18 • 5

- April 2016
- SDK 5.2.2.GA

- Everything worked in the simulators
- Final project validated and accepted to iTunes Connect

But ½ hour after each successful submission, email regarding invalid Swift Support was returned.

Various attempts were tried using the Stack Overflow suggestions with no positive result.

The time taken to try and resolve this issue was far out weighing any productivity enhancement to the app so the idea was scrapped.

The issue was reported in the Appcelerator JIRA

https://jira.appcelerator.org/browse/TIMOB-23313

Apple Watch App Redux

Objective:

- Use the latest version of the Ti SDK (6.0.1.GA) and see if a watch app can now be built
- Make the app as simple as possible (a ‘noddy’ app)
- Successfully submit and publish to the App Store

Noddy
Slang: *Something easy or mundane*.

Tooling:

- Sublime Text with a custom plugin (erbium - for Ti related functions)
- Latest Appcelerator Titanium installation (using Ti SDK 6.0.1.GA)
- Xcode 8.2.1

Steps:

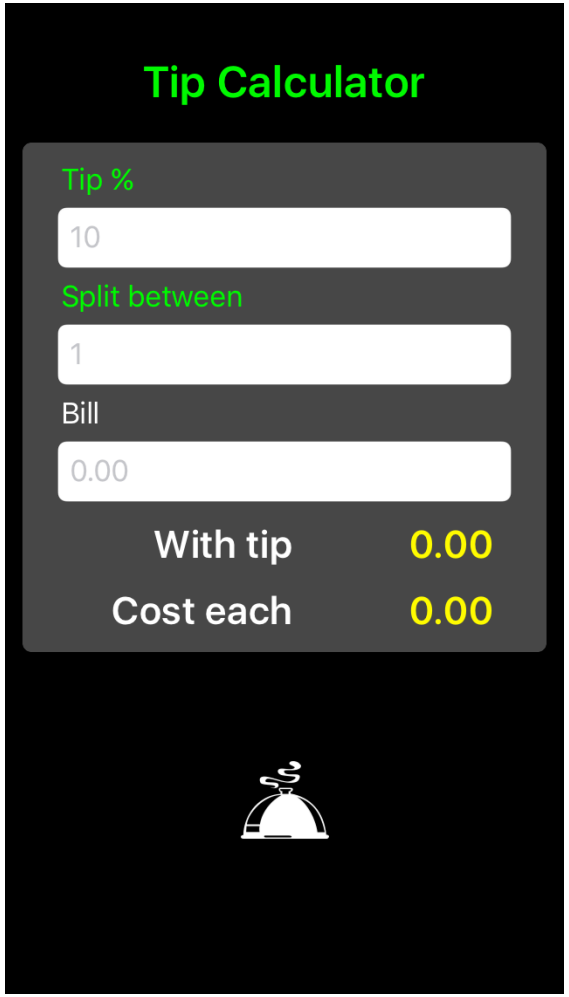
- Create a blank Alloy app
- Develop and test the iPhone app
- Add a watch extension to the project
- Develop and test the watch app

Create a blank Alloy App

This is actually 2 steps using the CLI

1. Create a classic Titanium project - ti create --force --type app --sdk 6.0.1.GA --id uk.spiralarm.tipcalculator --log-level info --name tipCalculator --workspace-dir /Users/accountone/sa --platforms ios --url http://www.spiralarm.uk

2. Add the Alloy files to the project – alloy new /Users/accountone/sa/tipCalculator --force

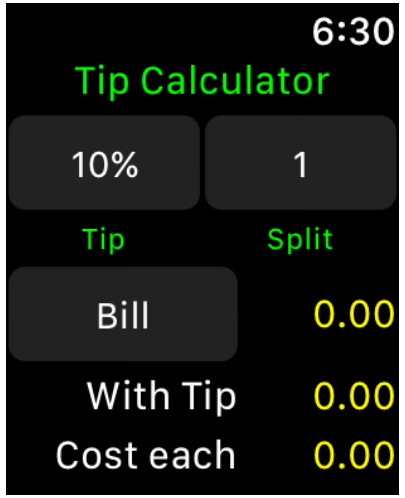


The iPhone app was then created and tested BEFORE adding the watch extension.

This was after the third attempt as I wanted to change the bundle id of the app which caused all kind of issues. The easiest way was to make sure of the bundle ID BEFORE adding the watch extension.

You could probably edit/hack the created Xcode project afterwards – but that is probably a whole new world of pain – especially for me.

Add the Watch Extension



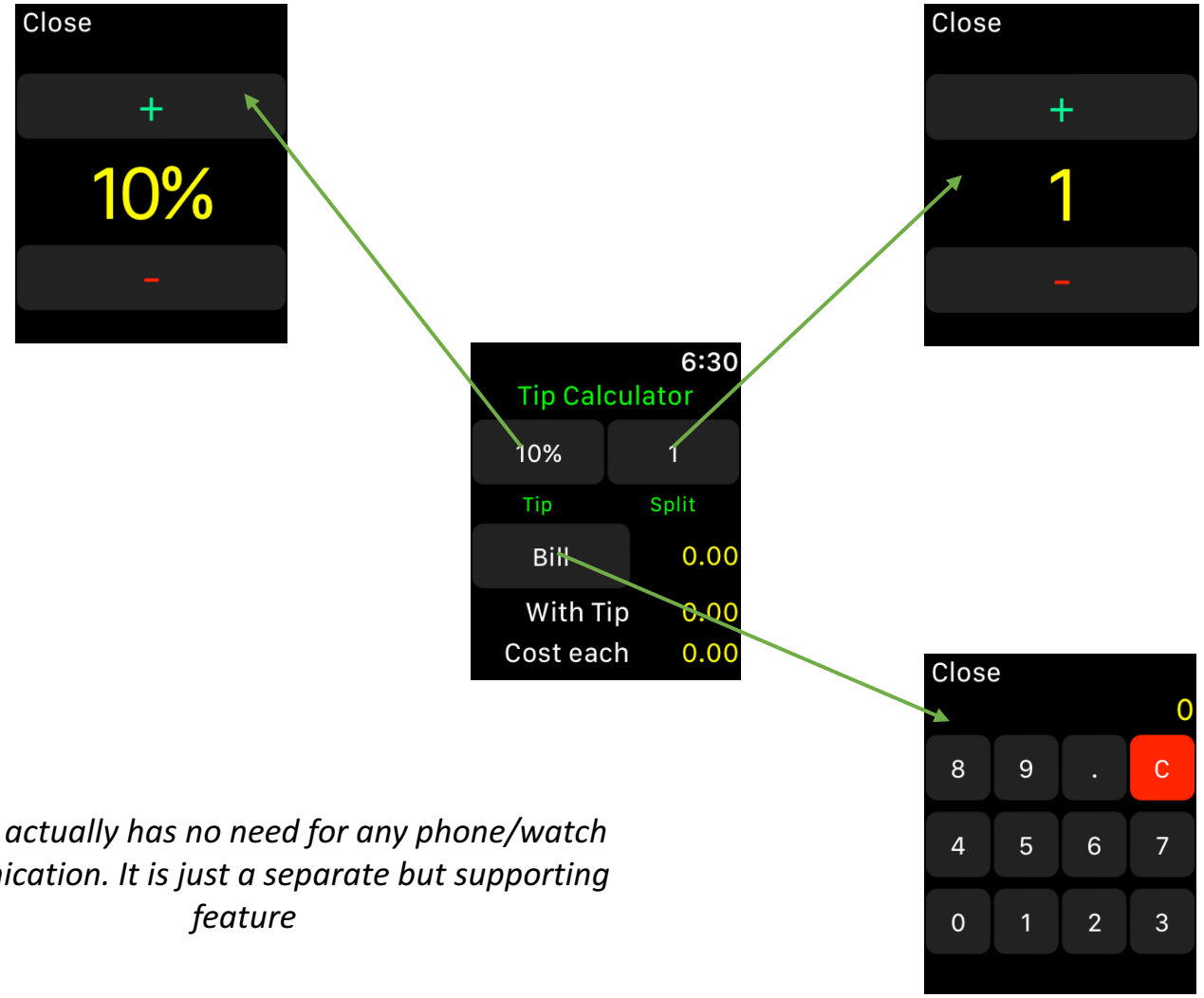
Add the watch app extension to the project- ti create --project-dir /Users/accountone/sa/tipCalculator name tipCalc --type=applewatch --template watchos2-swift

This was the only thing required apart from adding the TEAM ID into the tiapp.xml. The rest of the extension definition in the tiapp.xml was un-touched.

```
<extensions>
          <extension projectPath="extensions/TipCalc/TipCalc.xcodeproj">
                    <target name="TipCalc WatchKit Extension">
                              <provisioning-profiles>
                                        <device/>
                                        <dist-appstore/>
                                        <dist-adhoc/>
                              </provisioning-profiles>
                    </target>
                    <target name="TipCalc WatchKit App">
                              <provisioning-profiles>
                                        <device/>
                                        <dist-appstore/>
                                        <dist-adhoc/>
                              </provisioning-profiles>
                    </target>
          </extension>
</extensions>
```

Watch App Design

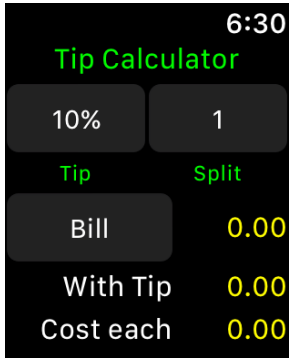
The app is a very simple design (in keeping with the ‘noddy’ principle), consisting of the main screen and 3 modal screens used to adjust some values.



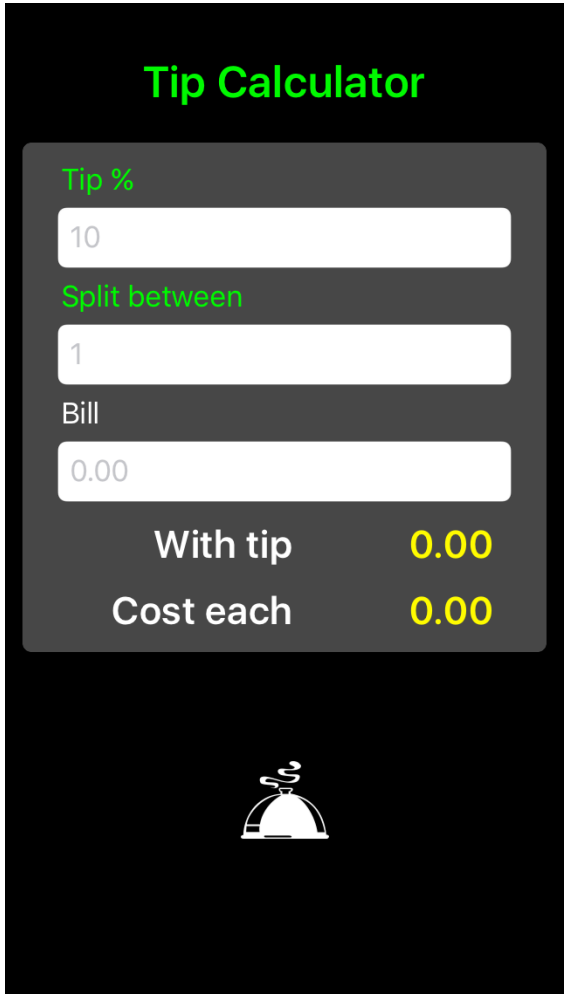
The app actually has no need for any phone/watch communication. It is just a separate but supporting feature

Watch App Design – Part Two

Although the app design does not call for it, a simple example of communicating between watch and app was put in (at Ket's suggestion).



{tip, bill, split}



Basically, when a calculation is done on the watch the data is also sent to the phone app (manual hand off?)

Watch and phone communication

Various methods were tried to send data in both directions, having various amounts of success with each:

- sendMessage
- userInfo
- applicationContext

applicationContext and userInfo worked, but I could not get the sendMessage to work.

I opted for userInfo as this was stated as being a guaranteed delivery method and the data being delivered would be queued. I also found out that the data would only send correctly if it was all send as text data.

Watch App

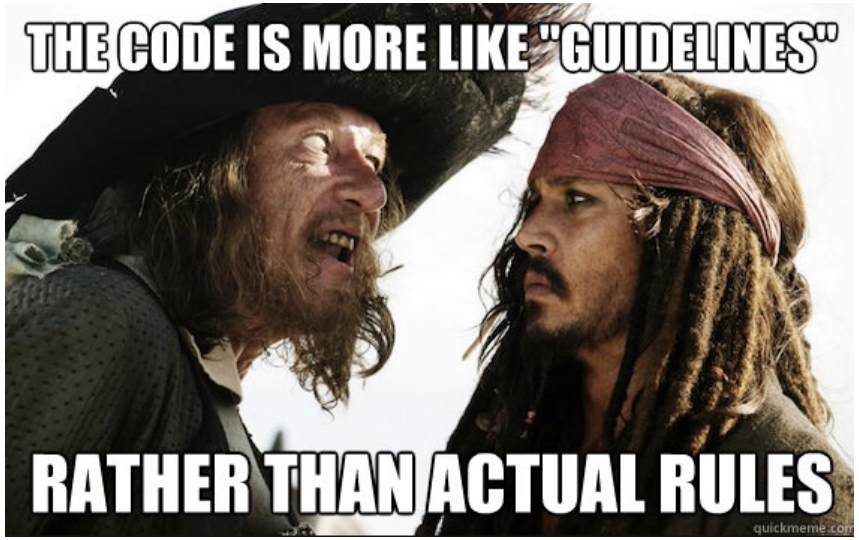
```
let calcInfo = [
        "tip":"\(tipAmount)",
        "split": "\(splitBetween)",

        "bill": "\(billTotal)"

] as [String : Any];
session.transferUserInfo(calcInfo)
```

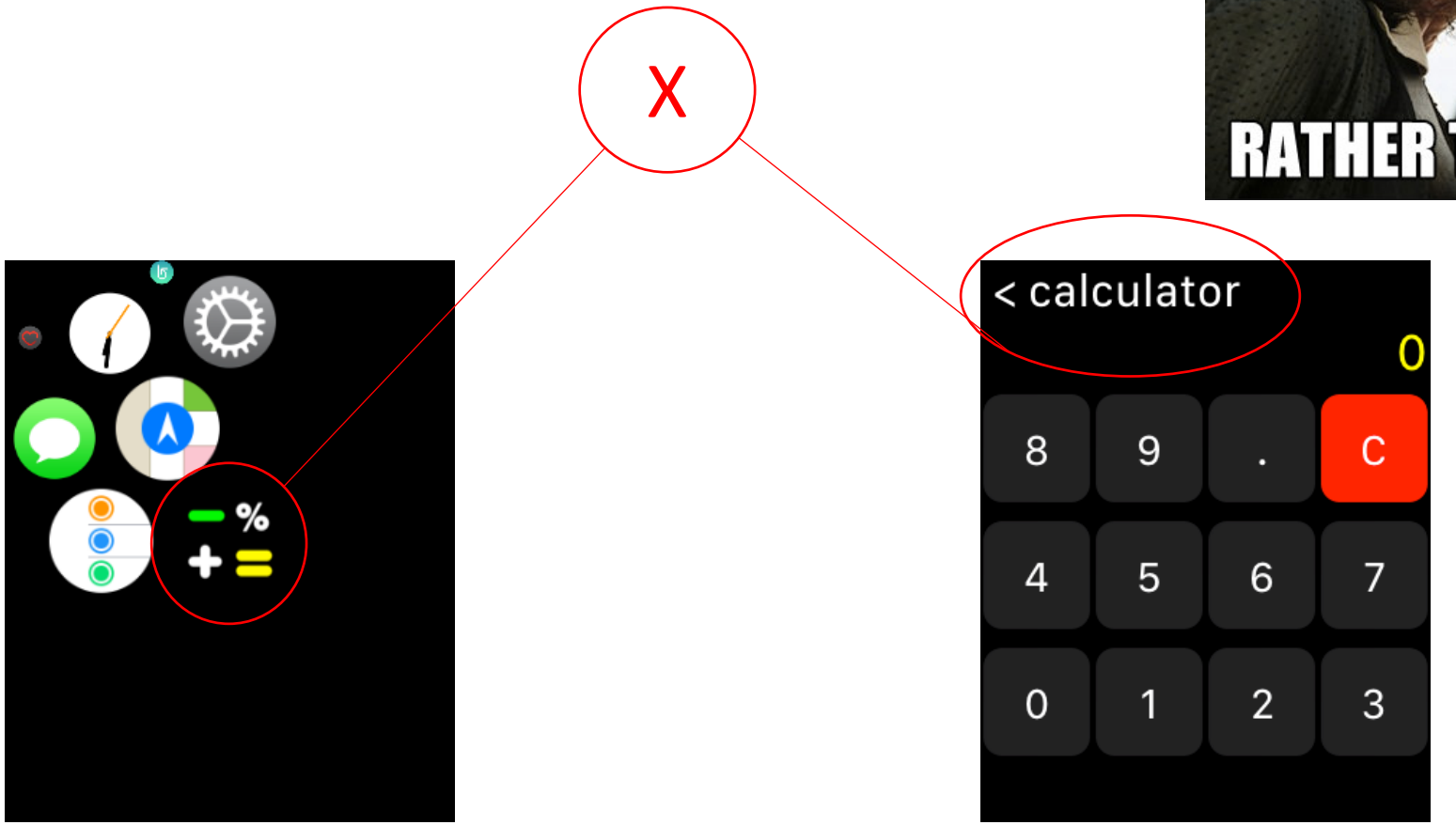
iPhone App

```
Ti.WatchSession.addEventListener('receiveuserinfo', function(e) {
        console.log(JSON.stringify(e));
        if(e.userInfo){
            $.bill.setValue(e.userInfo.bill);
            $.tip.setValue(e.userInfo.tip);
            $.split.setValue(e.userInfo.split);
            calculate();
        }
});
```

Read the Watch HIG

The first submission I did seemed OK, but 24 hours later I had an email from Apple telling me the watch app had been rejected due to not complying with the HIG





Conclusions

- Make sure you set the bundle ID of your main app BEFORE you add the watch app extension
- Remember to add the TEAM ID to tiapp.xml
- No need to edit the tiapp.xml any further
- Follow the Watch HIG

My ‘Noddy’ app is in the app store to download and the source code is available in github:

- Appstore: https://itunes.apple.com/gb/app/watch-tips/id1205407902?mt=8

- Github: https://github.com/magnatronus/Watch-Tips

An Aside:
While developing this app I found another issue that only seems to affect doing an App Store distro build. If the app project directory is on an external drive (i.e. an attached SSD) it errors and will not built.

Associated JIRA ticket : https://jira.appcelerator.org/browse/TIMOB-24405