



tibar

ZBar integration module for Titanium Mobile

 Search projects

[Project Home](#) [Downloads](#) [Wiki](#) [Issues](#) [Source](#)

[Summary](#) [Updates](#) [People](#)

Project Information

★ Star project
[Activity](#) High
[Project feeds](#)

Code license
[GNU Lesser GPL](#)

Content license
[Creative Commons 3.0 BY-SA](#)

Labels
[Barcode](#), [Titanium](#), [ZBar](#),
[Module](#)

👤 **Members**
[lukasz.r...@gmail.com](#),
[las...@googlemail.com](#)

Featured

📄 **Downloads**
[TiBarReaderApp-0.4.1.zip](#)
[TiBarReaderApp-0.4.zip](#)
[tibar-iphone-0.1.zip](#)
[tibar-iphone-0.4.1.zip](#)
[tibar-iphone-0.4.2.zip](#)
[tibar-iphone-0.4.zip](#)
[Show all »](#)

ZBar integration module for Titanium Mobile

DISCLAIMER

This is a beta version, not meant for production. We are not professional Objective-C developers so there may will be better ways to approach this. If you want to contribute please email us. Special thanks goes to ZBar project and Appcelerator team.

Tutorial - How to build a barcode reader enabled iPhone App within five minutes.

PREREQUISITE

- Titanium Mobile SDK 1.4.2
- iOS SDK 4.1
- iPhone 3GS or higher (other devices: wip)

Step 1

Module installation:

- download [TiBar module](#)
- copy into /Library/Application\ Support/Titanium folder

Step 2

Create and run your App:

- create new Titanium Mobile project (Titanium SDK version: 1.4.2)
- edit tiapp.xml file, add following code within tag <ti:app>, use proper version number

```
<modules>
  <module version="0.x.x">tibar</module>
</modules>
```

- edit app.js file, replace all content with following code

```
var win = Titanium.UI.createWindow({
    title:'TiBar Test App',
    backgroundColor:'#fff'
});

var TiBar = require('tibar');
var label = Titanium.UI.createLabel({
    text:'TiBar App',
    textAlign:'center',
    width:'auto'
});

var button = Ti.UI.createButton({
    title: "Scan barcode",
    height:50,
```

```

        width:250,
        bottom:20
    });

    button.addEventListener('click', function(){
        TiBar.scan({
            // simple configuration for iPhone simulator
            configure: {
                classType: "ZBarReaderController",
                sourceType: "Album",
                cameraMode: "Default",
                symbol:{
                    "QR-Code":true,
                }
            },
            success:function(data){
                Ti.API.info('TiBar success callback!');
                if(data && data.barcode){
                    Ti.UI.createAlertDialog({
                        title: "Scan result",
                        message: "Barcode: " + data.barcode + " Symbology:" + data.symbology
                    }).show();
                }
            },
            cancel:function(){
                Ti.API.info('TiBar cancel callback!');
            },
            error:function(){
                Ti.API.info('TiBar error callback!');
            }
        });
    });

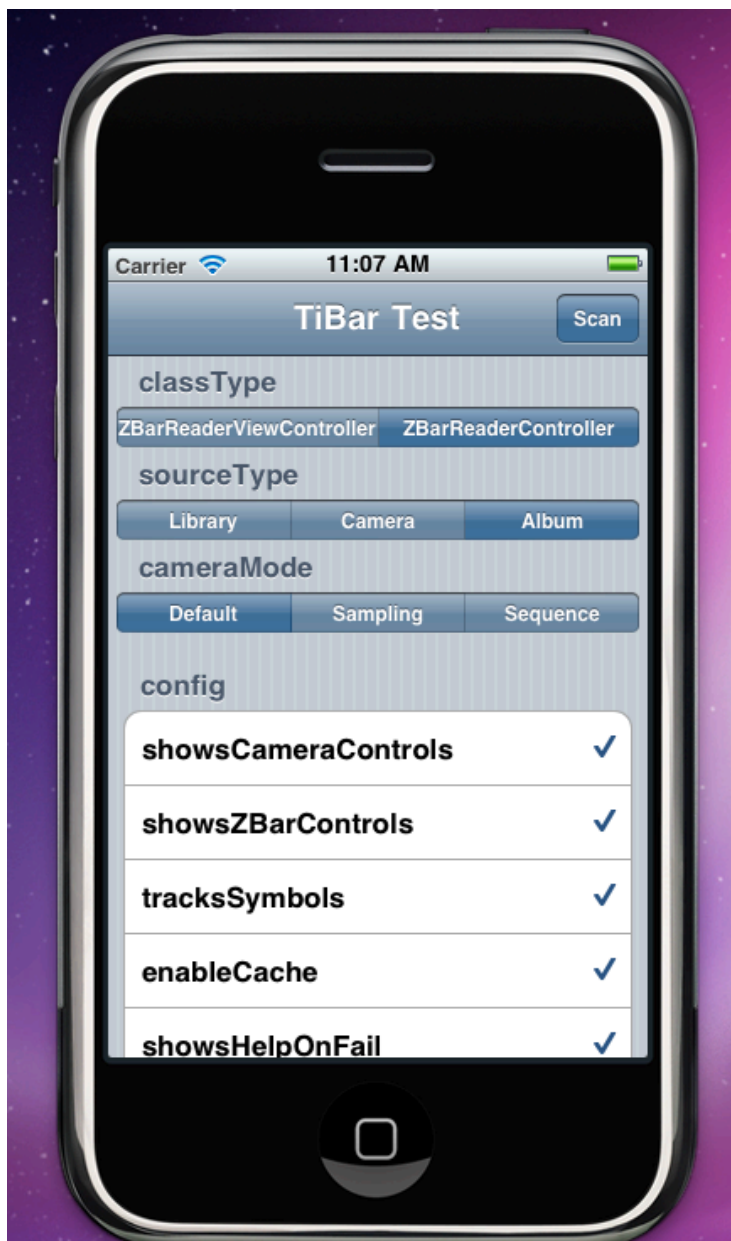
    win.add(label);
    win.add(button);


    win.open();

```

- launch your app, it will fail :(
- open project in XCode (/[project folder]/build/iphone/*.xcodproj)
- if needed update Base SDK information
 - XCode \ Project \ Edit Project Settings \ Build \ Base SDK - iOS Device 4.1
 - XCode \ Project \ Edit Active Target '[project name]' \ Build \ Base SDK - iOS Simulator 4.1
- close and open XCode project
- add missing frameworks - XCode \ Frameworks \ Add \ Existing Frameworks...
 - AVFoundation.framework
 - CoreMedia.framework
 - CoreVideo.framework
 - QuartzCore.framework
 - libiconv.dylib
- build and run your app - XCode \ Build \ Build and Run
- that's it, close XCode and launch app in Titanium Developer

TiBar Reader App





After you installed TiBar module, download [TiBar Reader App v0.4](#), unzip and import into your Titanium Developer (It is possible that you have to update Base SDK information and add missing frameworks, as described above). This App allows you to test the configuration.

How to test it in simulator?

You can drag an image (or any other file, like a PDF) to the simulator, it will immediately open Safari and display the image. Save the image by Tapping and Holding on it. Now start TiBar Reader App, press Scan button and choose your barcode (NOTE: your barcode image have to be square).

NOTE If you want to change default ZBar help notification, please edit `zbar-help.html` in Resources folder.

Configuration

```
// Configuration
// VC - ZBarReaderViewController - automatic capture
// C - ZBarReaderController - manually capture

var config = {
  classType: "ZBarReaderController", // ZBarReaderViewController, ZBarReaderController
  sourceType: "Album", // Library(C), Camera(VC), Album(C)
  cameraMode: "Default", // Default, Sampling, Sequence
  config:{
    "showsCameraControls":true, // (VC)
    "showsZBarControls":true,
    "tracksSymbols":true, // the tracking rectangle that highlights barcodes
    "enableCache":true,
    "showsHelpOnFail":true,
    "takesPicture":false
  },
  custom:{ // not implemented yet
    "scanCrop":'',
    "CFG_X_DENSITY":'',
    "CFG_Y_DENSITY":'',
    "continuous":''
  },
  symbol:{
    "QR-Code":true,
    "CODE-128":false,
    "CODE-39":false,
    "I25":false,
    "DataBar":false,
    "DataBar-Exp":false,
    "EAN-13":false,
    "EAN-8":false,
    "UPC-A":false,
    "UPC-E":false,
    "ISBN-13":false,
    "ISBN-10":false,
    "PDF417":false
  }
};
```

This software uses the open source ZBar Barcode Reader library, version 1.1, which is available from <http://zbar.sourceforge.net/iphone>

Changelog

0.4.2

- Integrated ZBarSDK 1.1
- Compiled with Titanium Mobile SDK 1.5.1 and iOS 4.2

0.4.1

- Module build with iOS 4.2

0.4

- Module template based on Titanium Mobile SDK 1.4.2
- TiBar use now ZBarSDK 1.0

- Tested with: Titanium Mobile SDK 1.4.2, iOS SDK 4.1, ZBarSDK 1.0

0.3

- Added configuration object
- Added callbacks for cancel and error
- TiBarReaderApp-0.1 includes ZBar help files

0.2

- Data object in success callback returns now symbology

0.1

- Initial preview release
- success callback

QRCode

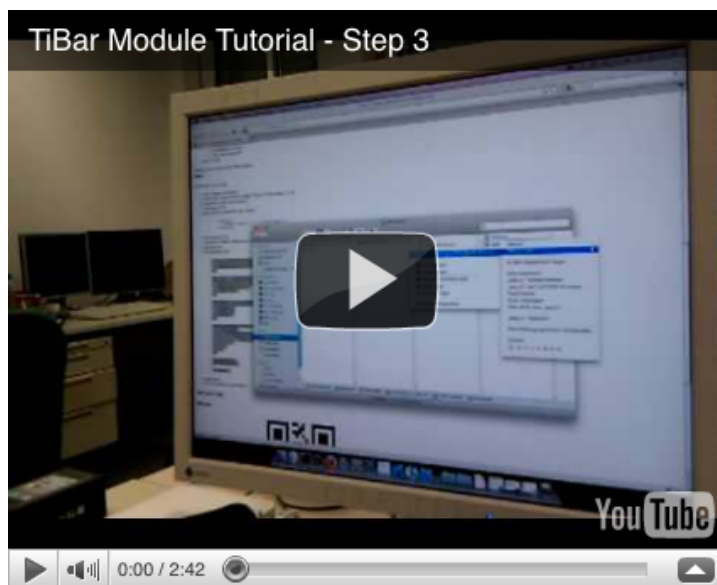
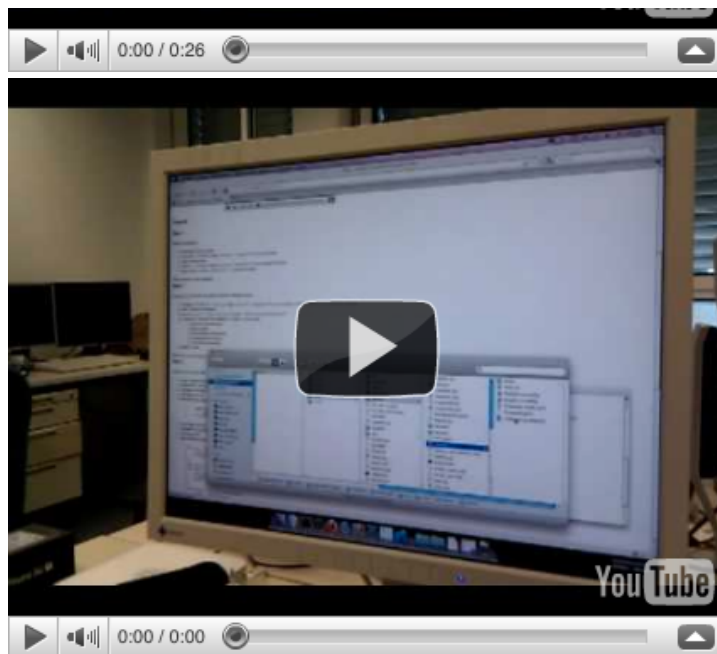


Barcode



Video





To be continued...

The source code and detailed description how we made the integration will follow soon.