Environment Configuration and Running Samples on Tizen

Table of Contents

Introduction

This document will walk you through:

- building a Titanium Mobile SDK for Tizen,
- configuring Tizen development environment,
- configuring Titanium development environment,
- running a sample Titanium application on Tizen.

It will also explain all the necessary configurations required to complete the steps.

When starting from scratch, please execute the steps in the given order.

Building Titanium Mobile SDK for Tizen

In order to develop Titanium applications for Tizen, the Titanium Mobile SDK for Tizen must be installed on the developer's machine. If the Titanium Mobile SDK is already available, skip this section.

Prerequisites:

- node.js v0.8+ required

Steps:

1. Get the source code from the titanium_mobile_tizen repo into a folder. It will be referred to as <titanium_mobile_tizen>.
2. In the command prompt, navigate to this folder, and execute the command **npm install** (to get all the required nodejs modules).
3. Download the latest (3.0.0) SDK for Win32 from http://builds.appcelerator.com.s3.amazonaws.com/index.html into another folder. It will be referred to as <build_dir>.
4. Do not unzip it.
5. Create a temporary folder <build_dir>/temp.
6. Go back to <titanium_mobile_tizen> and execute **node build.js <build_dir>/mobilesdk-3.0.0.v<NUMBER>-win32.zip <build_dir>/temp**
   In this command, the first parameter is the path to the Mobile Web sdk, and the second is the path to the temporary folder used for unzipping and other operations. This folder should not contain any files.
7. After successful build, the file **<build_dir>/mobilesdk-3.0.0.v<NUMBER>-win32-tizen.zip** will appear. This is the newly built SDK.

Installing Tizen development Environment

Tizen development environment is required for any Tizen development, even if the development itself will be performed using Titanium tools.

Download Tizen SDK 2.0

Download Java Development Kit (JDK) version > 6 from Java Development Kit . (On Windows, make sure Java binaries are in the PATH environment variable).

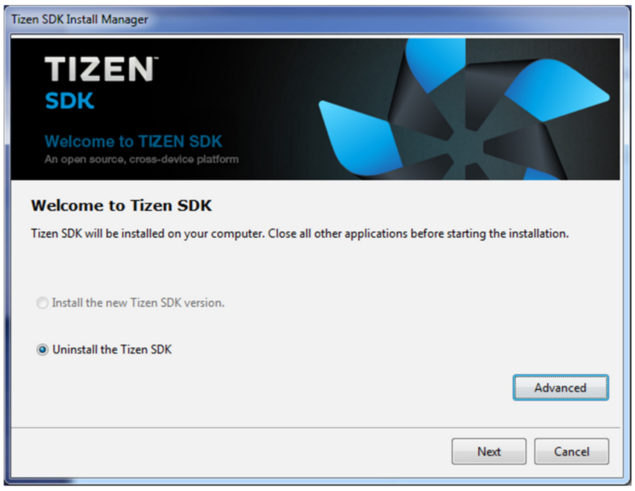
Tizen SDK is the development kit published by the vendor of Tizen, and is unrelated to Titanium. Please do not confuse it with the **Titanium Mobile SDK for Tizen**, which is a Titanium Mobile SDK extension.

Download the Install Manager and the installation image from: https://developer.tizen.org/sdk (be sure to select the installer for that belongs to your operating system and its bitness).

All the following steps assume that installation files for Windows x64 are used; Linux installation and configuration process is similar.

Install Tizen SDK

Run tizen-sdk-2.0-windows64.exe and click button ***Advanced***

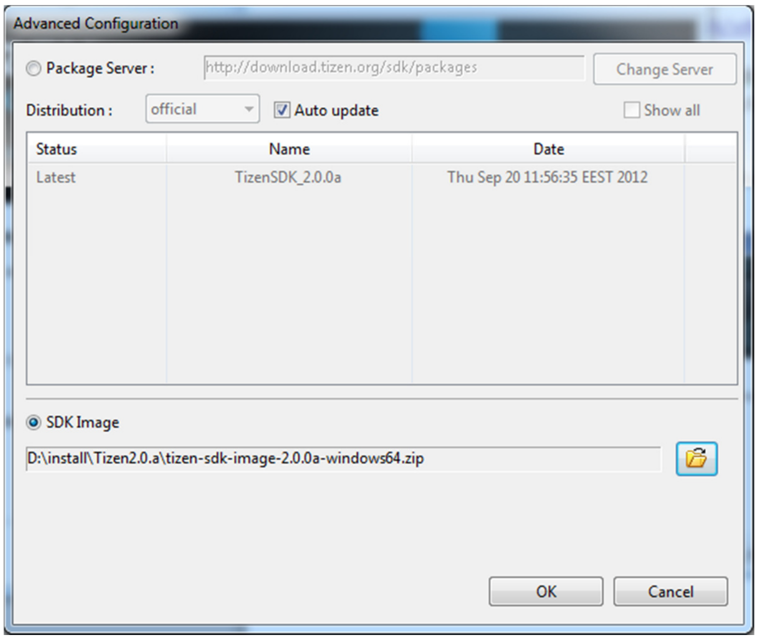

Tizen SDK Install Manager
TIZEN
SDK
Welcome to TIZEN SDK
An open source, cross-device platform

Welcome to Tizen SDK

Tizen SDK will be installed on your computer. Close all other applications before starting the installation.

Install the new Tizen SDK version. [✓ Uninstall the Tizen SDK]

In the next window, select the radio button "SDK Image"



Package Server : http://download.tizen.org/sdk/packages

Distribution : official   [✓ Auto update]

Status	Name	Date
Latest	TizenSDK_2.0.0a	Thu Sep 20 11:56:35 EEST 2012

[✓ SDK Image]

D:\install\Tizen2.0.a\tizen-sdk-image-2.0.0a-windows64.zip

Then select the downloaded tizen-sdk-image-2.0.0a-windows64.zip. Click OK.

SDK installation starts. Use the default settings for all the steps, and go through the installation process as prompted on the screen. As the result, the Tizen SDK installed into ***C:\tizen-sdk*** .

Windows: Install Intel HAXM for Tizen

This step can be skipped on Linux, as hardware acceleration is usually enabled on Linux by default.

Without hardware acceleration, Tizen emulator is very slow even on fastest Windows machines. It is recommended to configure hardware acceleration.

Go to the page https://developer.tizen.org/downloads/sdk/installing-sdk/hardware-accelerated-execution-manager, download and install **IntelHaxmTizen.exe**.

Important: HAXM for Tizen conflicts with HAXM for Android. You might have to Uninstall Android version first.

Reboot your computer.

Create a Tizen Emulator instance

Run the ***Tizen Emulator Manager*** from ***C:\tizen-sdk\tools\emulator\bin\emulator-manager.exe***

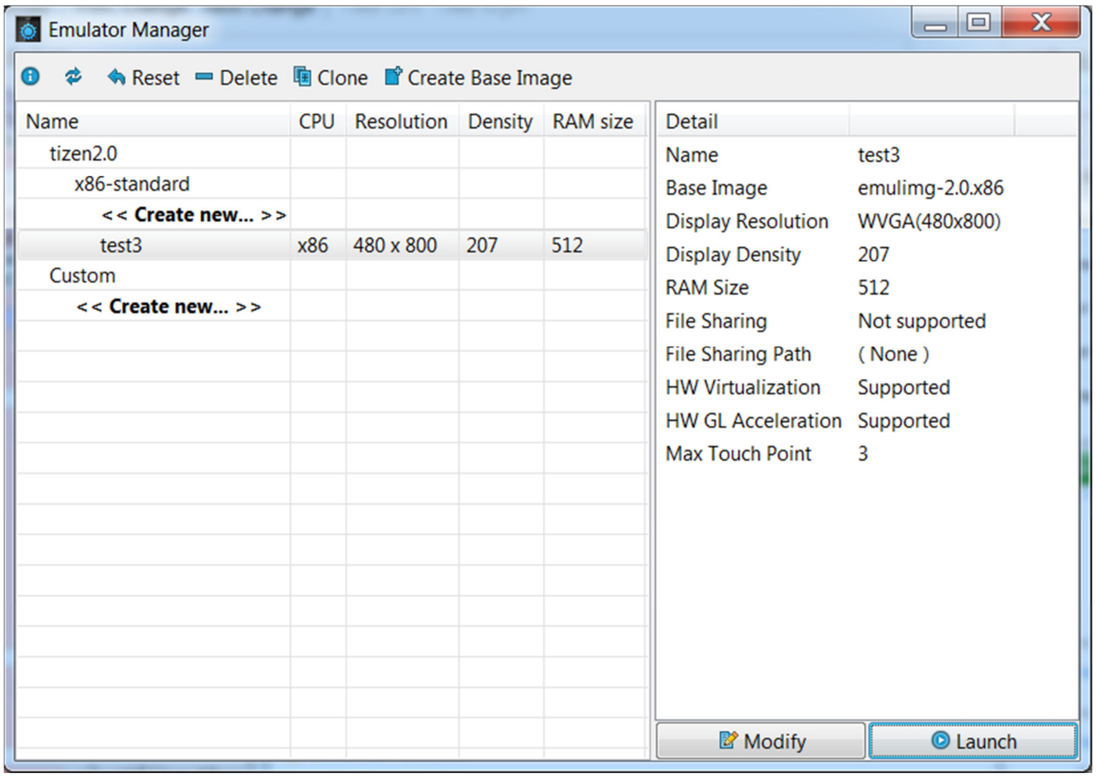
In the tree pane, click “x86-standard -> <<Create new…>> and then click the button Create.

Ensure that following options are set:

- Display Resolution: 480x800
- RAM Size: 512 or 1Gb
- HW Virtualization: Supported
- HW GL Acceleration: Supported.

Beware of the HW Virtualization setting. It should be enabled and show “Supported”. If it is disabled, check if Windows: Install Intel HAXM for Tizen is installed properly. Intel Hardware Accelerated Execution Manager and installation instruction available here: https://developer.tizen.org/downloads/sdk/installing-sdk/hardware-accelerated-execution-manager

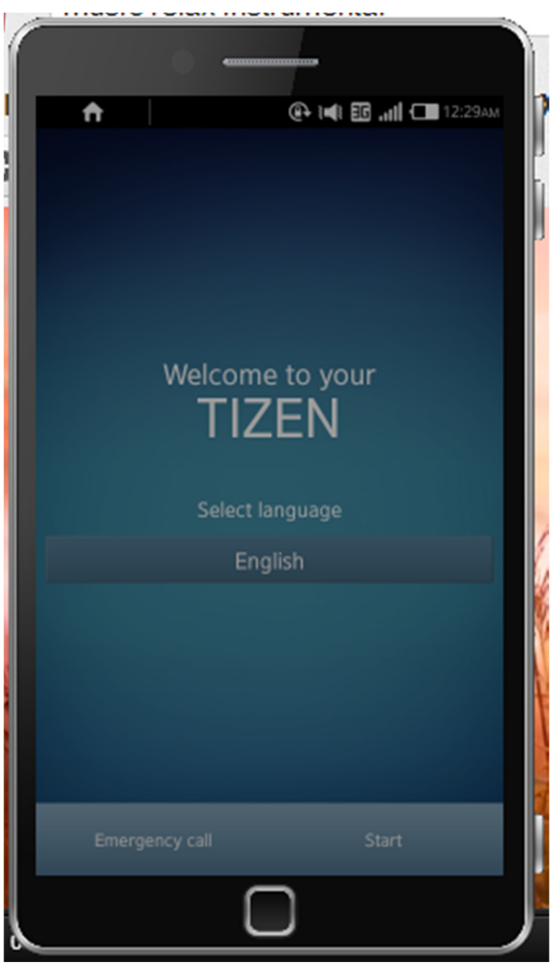
A ready to use emulator settings page looks as on the picture:



Name	CPU	Resolution	Density	RAM size
tizen2.0				
x86-standard				
<< Create new... >>				
test3	x86	480 x 800	207	512
Custom				
<< Create new... >>				

Detail	
Name	test3
Base Image	emulimg-2.0.x86
Display Resolution	WVGA(480x800)
Display Density	207
RAM Size	512
File Sharing	Not supported
File Sharing Path	(None)
HW Virtualization	Supported
HW GL Acceleration	Supported
Max Touch Point	3

Click the Launch button to start it.



For the first start, Tizen shows a welcome screen. Tap Start. On the next screen, click “Next” and click “Finish” on the last screen.

A ready to use emulator looks as on the picture:

Check if the emulator is reachable

It is important to check if there is a connection between the Tizen developer tools and the emulator, while the emulator is running.

In the console, run **C:\tizen-sdk\tools\ide\bin\web-list.bat**

The output should say that one emulator instance is available.

Note: if the command freezes, please restart the thing called "Samsung Debug Bridge", which is a slightly modified Android tool originally called adb. To do it, execute sdb located in ***C:\tizen-sdk\tools\:***

sdb kill-server
sdb start-server
sdb devices

The first two commands restarts the service. The last one shows a list of available devices/emulators after successful restart.

Installing Titanium Development environment

In the previous steps, we have been using Titanium NodeJS modules only. Now it is time to set up the full Titanium development environment, which will be required for Titanium development.

(You can refer to the "CLI Quick Start" document from Titanium Command-Line Interface Reference here http://docs.appcelerator.com/titanium/3.0/#!/guide/Titanium_Command-Line_Interface_Reference.)

Install Node.js

The CLI requires Node.js 0.8 or later. If you don't have Node installed, install it from:

nodejs.org

Install and Configure the CLI

Skip this section if you already have installed these Titanium NodeJS modules.

Install the titanium CLI.

npm install titanium -g

Log in using your Appcelerator credentials.

titanium login

You are prompted for your Appcelerator network login and password.

Get Titanium Studio and install Titanium Mobile SDK for Tizen

Download or build Titanium Mobile SDK for Tizen.

Install Titanium Studio from www.appcelerator.com.

Install the Titanium Mobile SDK for Tizen with Titanium Studio:

1. Run Titanium Studio
2. Go to the menu Help->Install Specific Titanium SDK
3. Select "Install From URL" and select the SDK zip file via the Browse button.

Configure the CLI

titanium setup

The script prompts you to enter basic information, such as your name, default locale, default SDK version, and default workspace folder.

Important: set the default SDK using this command, because the build script temporally does not allow selecting an SDK and uses default.

Ensure that Java is installed and available

Java runtime is required and should be available in PATH.

To check it execute in console:

java -version

At least java 1.6 is required to build Tizen applications. Install it from official site http://java.com/en/

Building and Running a Sample application

By now everything has been set up for Titanium development on Tizen OS. To verify this, let's build a sample Titanium application on Tizen – Kitchen Sink.

Source control

The source code repository is: https://github.com/appcelerator/titanium_mobile_tizen Check that you can see it in the browser before proceeding with the following steps.

Get sources into your working folder using the **git clone** command:

git clone https://github.com/appcelerator/titanium_mobile_tizen.git

This command will ask you for github credentials.

Prepare the required nodejs modules

Go to into folder titanium_mobile_tizen in any console and execute

npm install

It will download all the required nodejs modules (***see titanium_mobile_tizen/package.json*** for dependencies)

Build, deploy, and debug the sample application

In the file tiapp.xml for the application, change **sdk-version** to the default sdk version, or remove the entry.

In the console, go to in ***<PATH_TO_REPO>\titanium_mobile_tizen\tests\samples\KitchenSink***

In the current folder, execute:

titanium build --platform=tizen

On success build, ***<PATH_TO_REPO>\titanium_mobile_tizen\tests\samples\Tizen\build\tizen***will contain a file with the "wgt" extension. This is the resulting Tizen package containing Titanium Kitchen Sink.

As soon as you succeed in building, you can use the following commands to deploy and debug KitchenSink application on Tizen:

--dev-id	Install widget on this device after successful build	Usage sample: ***titanium build --platform=tizen --dev-id=emulator-26100***
--run-dev-id	Install and run widget on this device after successful build	Usage sample: ***titanium build --platform=tizen --run-dev-id=emulator-26100***
--debug-dev-id	Install widget on this device after build and initiate debugging. There are no Tizen specific debugger. Open result url in browser to debug a widget with Remote Inspector	Usage sample: ***titanium build --platform=tizen --debug-dev-id=emulator-26100*** (this will build and install the widget. Then it will show the debug url in the console, e.g.

		DEBUG URL : http://localhost:51164/inspector.html?page=1 Open this URL in the browser to debug remotely.