Intel® HTML5 Development Environment

Tutorial – Creating a Google Android* Plugin

v1.13 : 05.10.2013

Legal Information

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request. Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order. Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: http://www.intel.com/design/literature.htm

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.

Microsoft, Windows, and the Windows logo are trademarks, or registered trademarks of Microsoft Corporation in the United States and/or other countries.

Copyright © 2011-2013, Intel Corporation. All rights reserved.

Purpose

This document will show a step by step process of how to build the default plug-in for the Google Android* platform. Based on that experience, and the more detailed information included in the Intel® HTML5 Development Environment Plug-ins documentation, you should be able to create your very own Android applications that include plug-ins.

You can access the detailed Intel HTML5 Development Environment Plug-ins document at:

http://www.html5dev-software.intel.com/documentation/index.php?DOC=PLUGIN_DOCUMENT

Download the Google Android* SDK

The most recent Google Android SDK can be downloaded for your Microsoft Windows*, Apple Mac* OS X, or Linux* computer from this address:

http://developer.android.com/sdk

Extract the downloaded SDK file to your hard drive. Make sure you have write permissions to that directory. Write down the location so you can reference it later. The Android SDK may ask you to install the Java* SE Development Kit. Please do so during the Android SDK install.

The ADT bundle downloaded from Google should include everything you need to create and test an Android plug-in. For the purposes of this tutorial, it is assumed that you are just going to install the default Eclipse* IDE provided with the bundle.

Download the Sample Android Plug-In Code

Download the sample Android plug-in code from here:

http://www.html5dev-software.intel.com/documentation/index.php?DOC=PLUGIN_SAMPLE_ANDROID

This package includes all the code you'll need to create your sample plug-in. It includes:

- The project to load into the Intel® XDK
- The Java to import into Eclipse
- A completed HTML5 plug-in for test purposes

Save all the files to your hard drive, such as in a folder named *AndroidPlugin*. Just remember where you put them so that you can get to them later. We will be using the name *AndroidPlugin* going forward for the purposes of this document.

Create an Application Project in the Intel® XDK

Next, copy the HTML5 files to a brand new project in the Intel XDK:

- Open the Intel XDK
- Create a new project
- Give the project a name
- Select the "Create your own from scratch" development path
- Click *Next*
- Once the project is created in the XDK, open the project folder
- Delete everything out of the project folder, and replace it with the contents of the *AndroidPlugin* folder you just downloaded from Intel.

If you have some questions about how to install or use the Intel® XDK, take a look at Intel's tutorial on getting started with the XDK here:

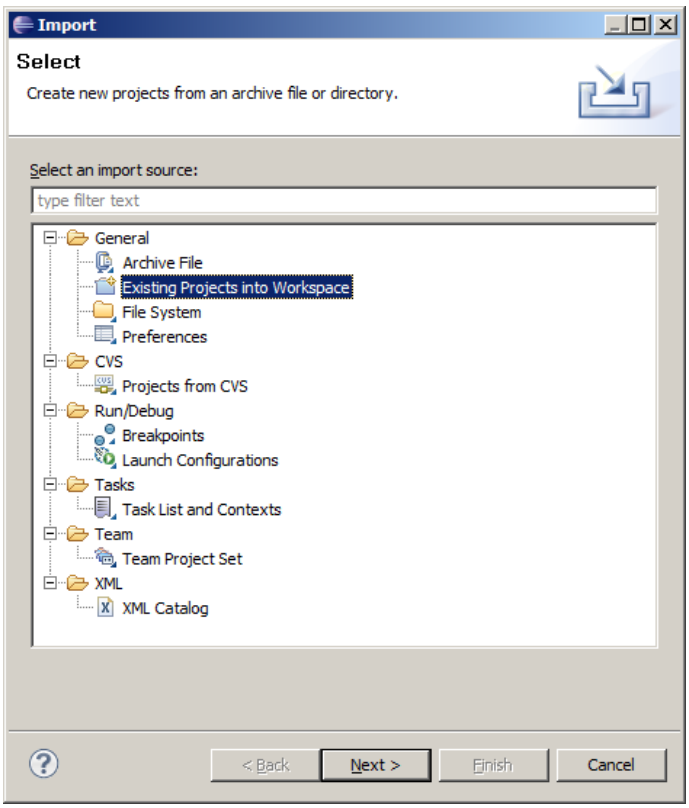
http://www.html5dev-software.intel.com/documentation/index.php?DOC=TUTORIAL_GETTING_STARTED

Import the Plug-In into Eclipse

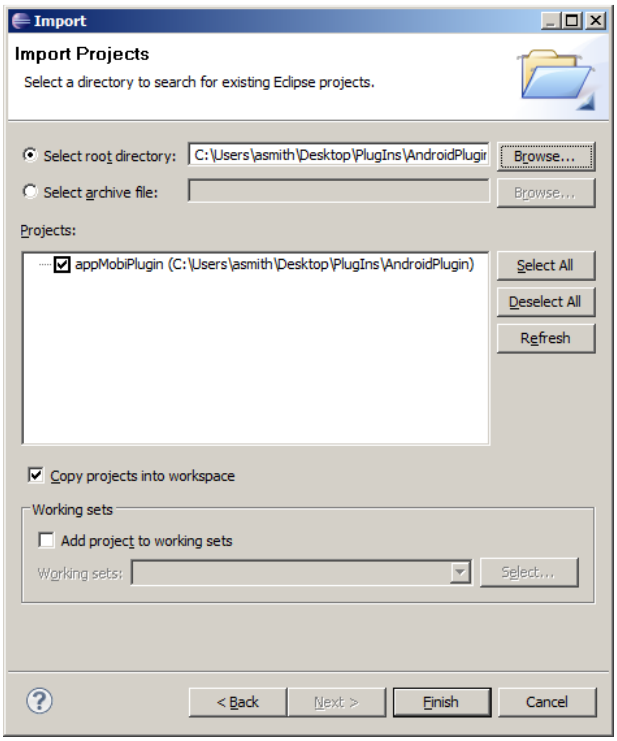
Next, import the contents of the *AndroidPlugin* folder into the Eclipse project:

- Open the Eclipse IDE
- Go to *File -> Import*

- Select *Existing Projects into Workspace*



- Click *Next >*
- Select the *AndroidPlugin* folder that you created after downloading the sample plug-in bundle.



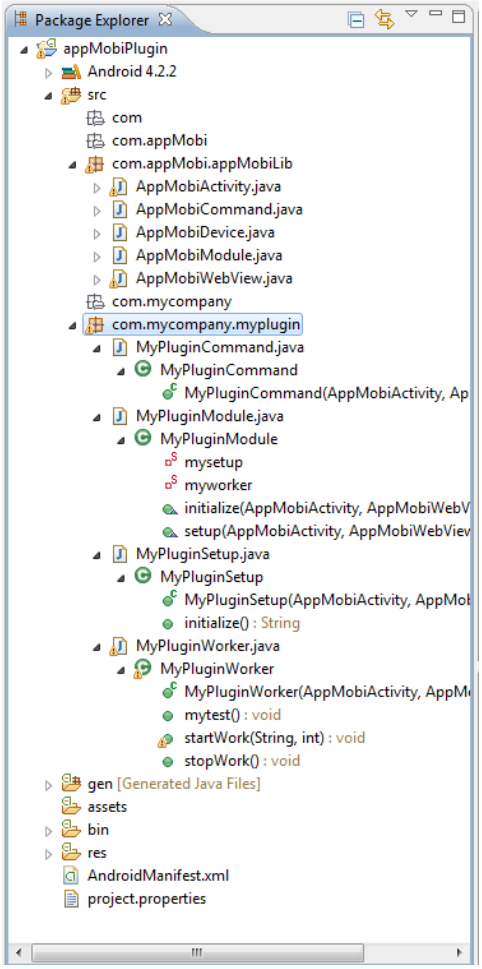
- Click *Finish* to complete the import

Export the Plug-In

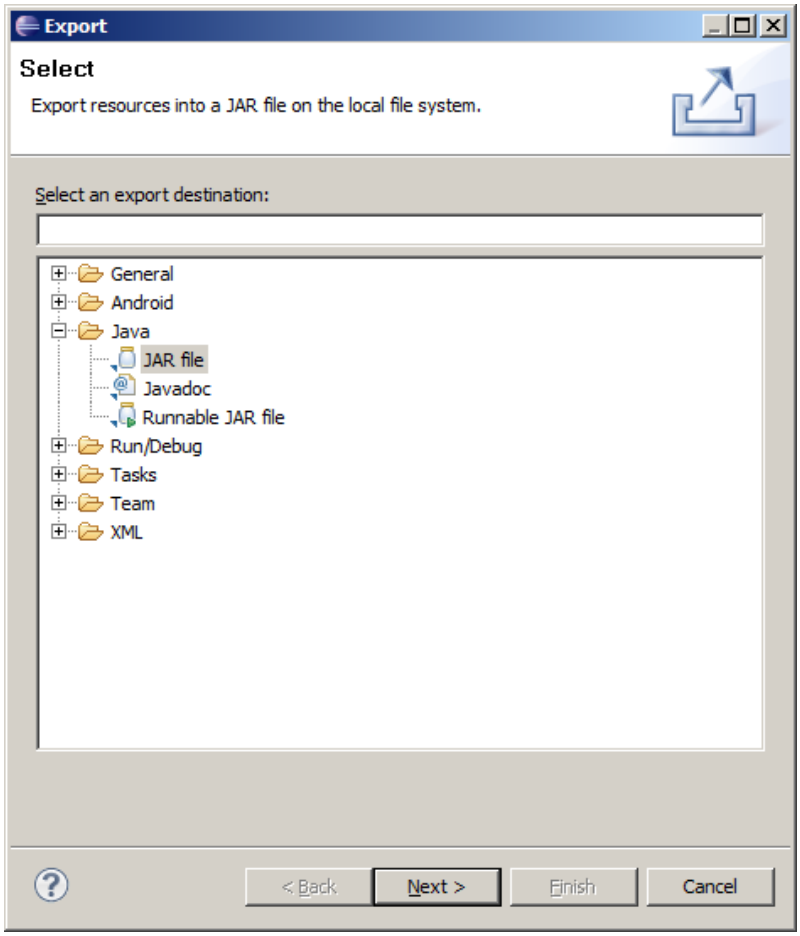
Next you'll export the plug-in code as a *.jar* file.

NOTE: appMobi class files should NOT be included, so make sure you on export your plugin directory, not the "src" directory.

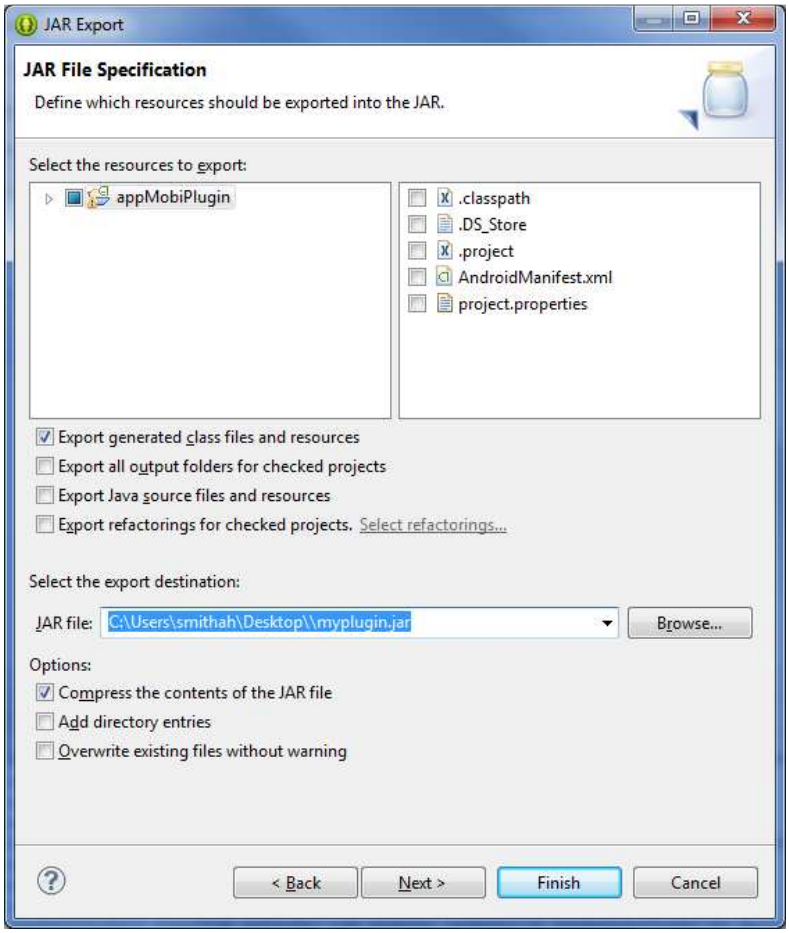
- From the *Package Explorer* select the plugin source directory named *com.mycompany.myplugin*



- Go to *File -> Export*
- Select *Jar File*



- Click *Next >*
- Select an export location



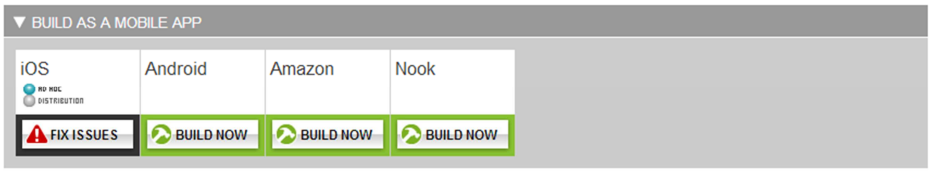
- Only "Export generated class files and resources" should be checked (this should be the default).
- Set the export destination and name the file (e.g. "yourpluginname.jar")
- Click *Finish* and your .jar will be created.

Create the Plug-In File

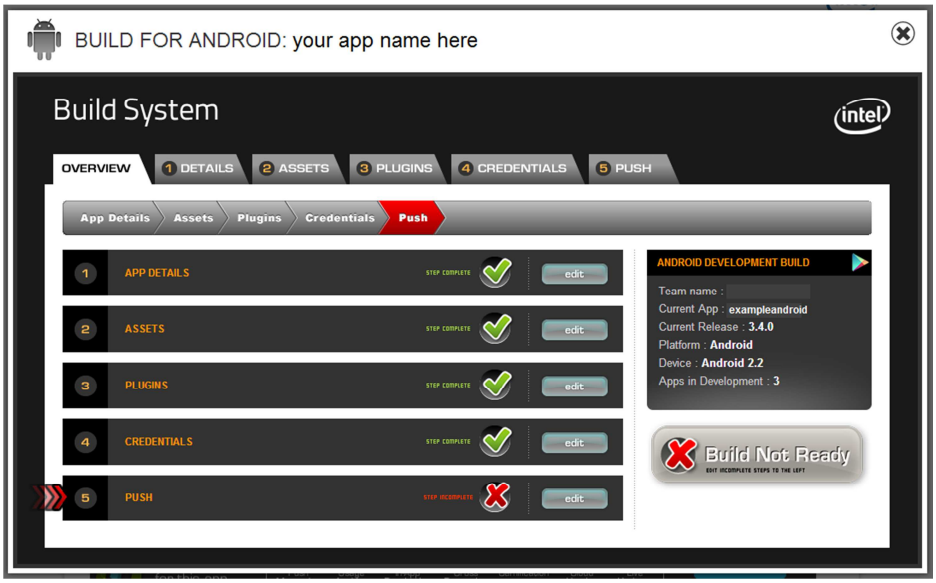
Next, you'll want to put the *.JAR* file that you just exported, as well as the *myplugin.js* file from the application project, into a .zip file. The download includes just such an example file. You'll need to include that bundle during the Intel XDK build process ( ).

Build the Application

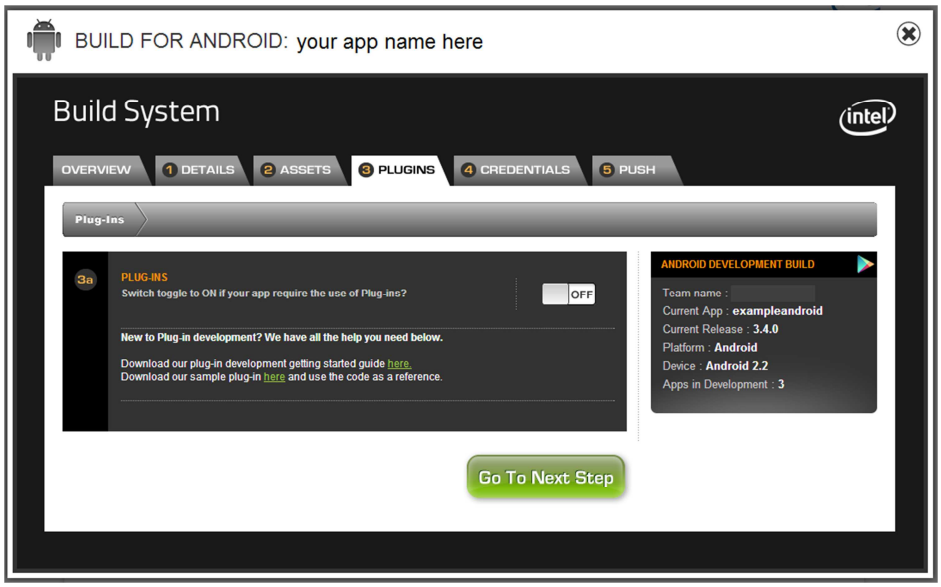
The Intel XDK will take you to the App Dev Center on the Control Center screen with your application selected. Expand the BUILD AS A MOBILE APP section.



Scroll over the Android build button, and choose the Change Build Settings link which will take you to the Build System Overview screen for your application.

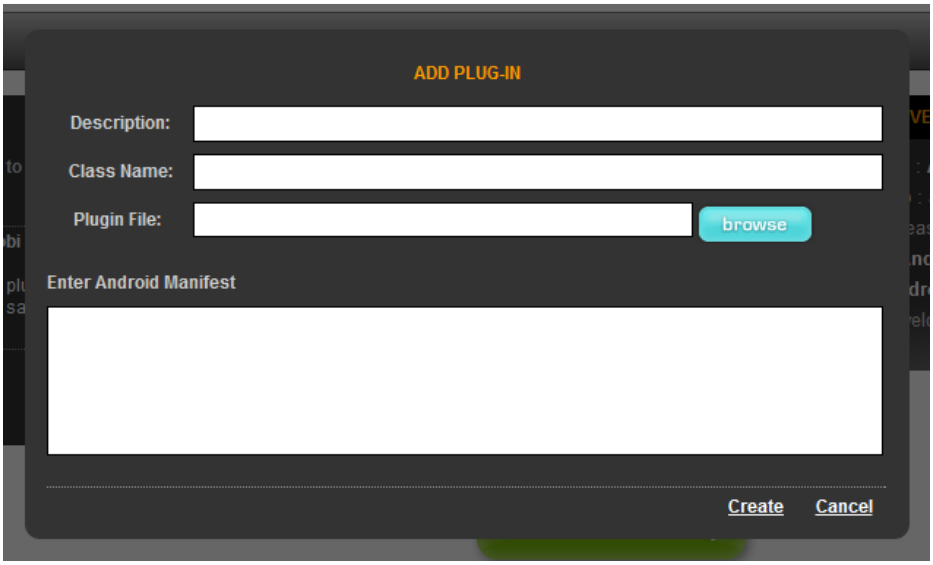


You will build the Android application as you would any other Android application, however on the Plugins tab, set the toggle switch to ON.



Next, click the

button which takes you to a screen where you can add information about your plugin and upload it.



Enter a brief description of your plugin, and the fully qualified Java class name of:

com.mycompany.myplugin.MyPluginModule

Next upload the plugin file you just created. You have now included that plugin for the build of your application.

Once all other build items have been completed, return to the Overview tab and build your app.

For more information on building an app through App Dev Center, see the article "Using the App Dev Center" here:

http://www.html5dev-software.intel.com/documentation/index.php?DOC=ARTICLE_APPDEVCENTER