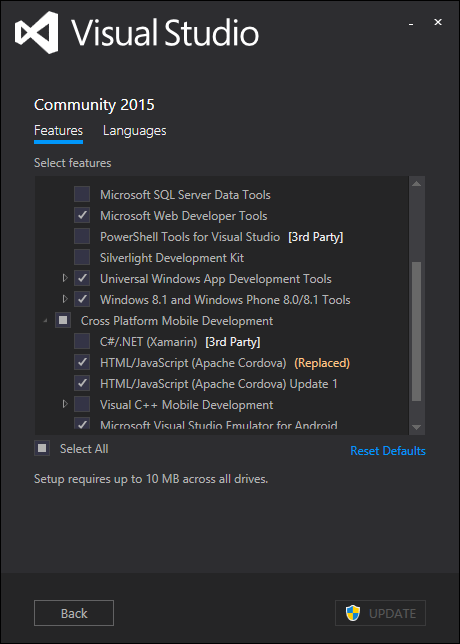
Steps:

Software to Use

Visual Studio 2015

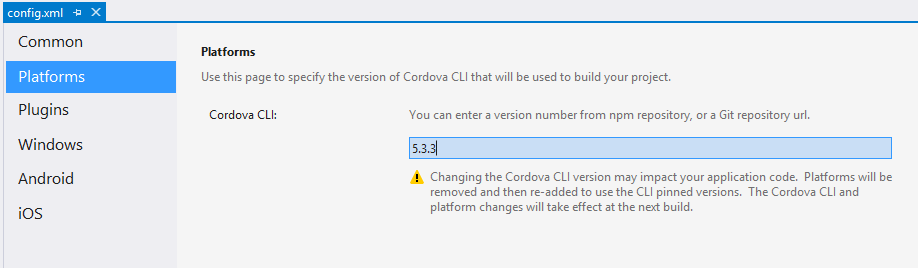
Windows and Web Development > Microsoft Web Developer Tools  
 HTML/JavaScript (Apache Cordova)  


Visual Studio > Extension and Updates

Web Essentials 2015  
Open Command Line  
Add New File

Update Apache Cordova

Use the Cordova CLI using config.xml. Enter the version number into the text field. When the file is saved, the files are downloaded. You can get the latest version from this URL:  
<https://github.com/apache/cordova-cli/releases>



Npm install –g cordova

Update folder Res images. Res images contains icons and splash screens for resolution specific devices.

Update the config.xml   
 The images are missing from the default res > icon > windows folder.

<icon src="res/icons/windows/Square44x44Logo.scale-100.png" width="44" height="44" />  
<icon src="res/icons/windows/Square71x71Logo.scale-100.png" width="71" height="71" />

Update Typescript to latest version

Download typescript for Visual Studio 2015

<http://www.microsoft.com/en-us/download/details.aspx?id=48593>  
 Click the details button to reveal a list of all version. Download the appropriate version

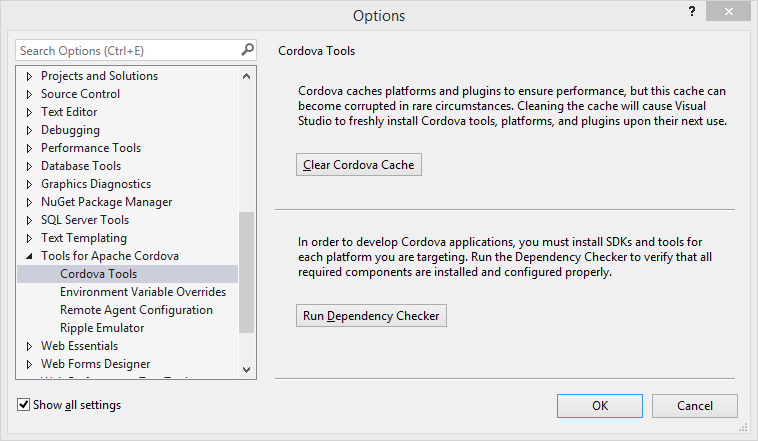
Manual Process for CLI. Does not apply to Visual Studio 2015.

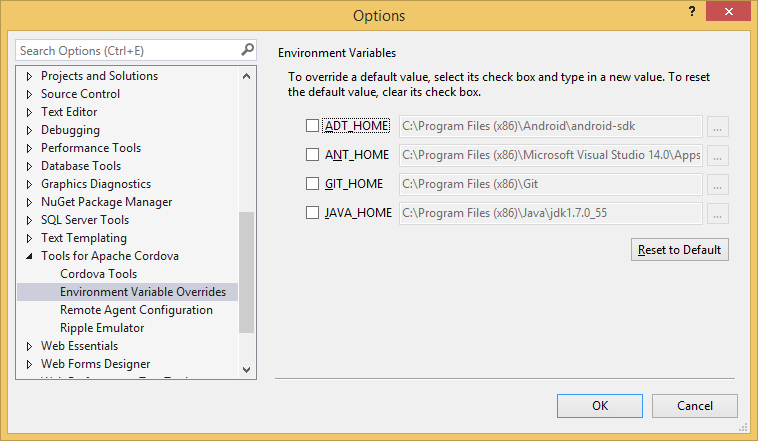
C:\Program Files (x86)\Microsoft SDKs\TypeScript\1.5\tsc

npm install -g typescript  
npm install -g typescript@next  
tsc –version  
where tsc

Other things to confirm are installed:  
 Android SDK Manager

Check if you have all the SDK’s by clicking Tools > Options > Tools for Apache Cordova > Cordova Tools > Run Dependency Checker.

  
 Java JDK is installed and using 64bit version. Ie C:\Program Files\Java\jdk1.8.0\_60  
 Tools > Options > Tools for Apache Cordova



1. Create a blank TypeScript Cordova Project.  
   Add Git version control if necessary.  
   TortoisGit > Settings > Remote > add Origin. Git Commit, Push to add git changes.
2. Remove Project\_Readme.html
3. Update Cordova-Android to latest version. Instructions on how to do it.  
   add to config.xml <engine name="android" spec="4.1.0" />  
   Use these instructions to install globally  
   <http://cordova.apache.org/announcements/2015/07/21/cordova-android-4.1.0.html>   
   If the project is published, before updating the config.xml then you’ll need to use the command prompt to upgrade the cordova-android version. Use this:  
   `cordova platform remove android`  
   Latest Releases: <https://github.com/apache/cordova-android/releases>
4. Update config.xml
   1. Update Android > Minimum SDK Version to 14. This is because we’re using cordova-android 4.1.0.
   2. Update the Common > package name. Package name should be domain name in reverse. Io.github.dbiele
5. Add Gulp(gulpfile.js). Add devdependencies to package.json.
6. Confirm taco.json has been updated with the latest version of cordova set in config.json. Taco is using for IOS building and the remote agent uses the CLI version defined in the taco.json.
7. Confirm tsconfig.json exists in scripts folder
   1. Compile with AMD. Make sure this is set in the tsconfig.json file. Add "module": "amd"

{

"compilerOptions": {

"noImplicitAny": false,

"noEmitOnError": true,

"removeComments": false,

"sourceMap": true,

"target": "es5",

"module": "amd",

"emitDecoratorMetadata": true,

"experimentalDecorators": true,

"outDir": "../www/scripts/"

},

"files": [

"./index.ts",

"./app.ts",

"./typings/tsd.d.ts"

]

}

1. Make sure to show all files in the solution explorer. Show all files.

Add Cordova Plugins

1. Crosswalk  
   Add Cordova CrossWalk plugin. Config.xml > plugins > Crosswalk Webview. Note, if there’s a problem installing the apk to the device, delete the previous installs on the device.
2. Cordova-plugin-whitelist  
   Add the “cordova-plugin-whitelist” to dependencies in config.xml. Whitelist is a core cordova plugin. Or use NPM to download the files and install locally.  
   <https://www.npmjs.com/package/cordova-plugin-whitelist>
3. Other Popular Plugins
   1. Cordova-plugin-console
   2. Cordova-plugin-device
   3. Cordova-plugin-inappbrowser
   4. Cordova-plugin-dialogs
   5. Cordova-plugin-splashscreen
   6. Cordova-plugin-statusbar

Update Security for Cordova

1. Cordova Security: Add external domains to config.xml Domain Access
   1. Make sure access origin = \*
   2. Or remove \* and add the names of the domains. [https://\*.jspm.io](https://*.jspm.io) and [https://\*.angularjs.org](https://*.angularjs.org). This is used for the index.html script tags
2. Update www/index.html
   1. <meta charset="utf-8" />
   2. <meta http-equiv="X-UA-Compatible" content="IE=edge">  
      use the latest engine to render the page and execute JavaScript.
   3. <meta name="viewport" content="width=device-width, initial-scale=1">  
      initial-scale property controls the zoom level when the page is first loaded.  
      The maximum-scale, minimum-scale, and user-scalable properties control how users are allowed to zoom the page in or out.  
      Width can be set to a specific number of pixels like width=600 or to the special value device-width value which is the width of the screen in CSS pixels at a scale of 100%
   4. <meta name="description" content="Visual Studio 2015, Cordova, TypeScript, Angular 2 Starter">
   5. Add security policy. See below.
3. Need to add the following code to index.html to allow for the CDN and JavaScript to work correctly:  
   <meta http-equiv="Content-Security-Policy" content="default-src 'self' data: gap: https://ssl.gstatic.com 'unsafe-eval'; style-src 'self' 'unsafe-inline'; media-src \*">

<meta http-equiv="Content-Security-Policy" content="default-src https: 'self' jspm.io; script-src 'self' 'unsafe-inline' <https://jspm.io>">

Add Keystores for release

1. Update build.json to include keystore for releasing to device. Publishing in release mode requires build.json to include keystore.
   1. Create a keystore and save it locally. Information on how to create keystore  
      <https://github.com/Microsoft/cordova-docs/tree/master/tutorial-package-publish#android>
   2. Update build.json to include keystore information.



* 1. Note: If errors occur, this may be due to previous version of app on device. Delete app on device and rebuild.

Install d.ts files for TypeScript

1. Install TSD and configure TSD to download d.ts files.  
   <https://github.com/dbiele/Cordova_TypeScript_RequireJS_>
   1. npm install tsd –g
2. Open the command prompt at the scripts folder and Install Angular2 d.ts files
   1. tsd install angular2 –s
3. configure TSD to install files in proper location
   1. Move the tsd.json file to the root of the project
   2. Change the “path” and “bundle” in tsd.json to point to new folder scripts/typings
   3. Create TSD task in gulp – see creating tsd gulp task.
   4. Reinstall d.ts files by running gulp task
4. Confirm Angular 2 is the latest version by checking <https://github.com/angular/angular/releases> and <https://github.com/borisyankov/DefinitelyTyped/tree/master/angular2>
5. Add systemJS d.ts files
   1. Tsd install Systemjs -s

Install gulp-tslint

1. Add to package.json  
   DevDependencies > "gulp-tslint"
2. Add the following to gulpfile.js:

gulp.task(**'tslint'**, function () {

// Built-in rules are at

// https://github.com/palantir/tslint#supported-rules

var tslintConfig = {

"rules": {

"semicolon": true,

"requireReturnType": true,

"requireParameterType": true,

"jsdoc-format": true,

"quotemark": [true, "single"],

"variable-name": [true,"allow-leading-underscore"]

}

};

return gulp.src(['scripts/\*\*/\*.ts', '!scripts/typings/\*\*'])

//Custom rules can be added to configuration. rulesDirectory: 'folder/folder'

.pipe(tslint({ configuration: tslintConfig }))

.pipe(tslint.report('verbose', { emitError: true, reportLimit: 0 }));

});

Install SASS

1. Add to package.json devDependencies
   1. Gulp-sass
   2. Gulp-autoprefixer
   3. Es6-promise
2. Add the following to gulpfile.js

require('es6-promise').polyfill();

gulp.task(**'build.css.sass'**, function () {

gulp.src('./scripts/components/materials/components/\*\*/\*.scss')

// Guilp-Sass runs the pre processor on the .scss files using Sass.

// Gulp-AutoPrefixer post processes the .css files using PostCSS.

// CSS and Folder structure is saved to destination folder.

.pipe(sass().on('error', sass.logError))

.pipe(prefixer({

browsers: ['last 2 versions'],

cascade: false

}))

.pipe(gulp.dest('./www/css'));

});

Adding Angular 2 JavaScript Files

1. Add JavaScript files using package.json to download locally, but because this is in alpha, add the following to index.html.
   1. <script src="https://github.jspm.io/jmcriffey/bower-traceur-runtime@0.0.87/traceur-runtime.js"></script>
   2. <script src="https://jspm.io/system@0.16.js"></script>
   3. <script src="https://code.angularjs.org/2.0.0-alpha.33/angular2.dev.js"></script>
   4. <script src="https://code.angularjs.org/2.0.0-alpha.33/router.dev.js"></script>
2. Not using jspm.io and angularjs.org script links
   1. Create the ‘libs’ folder in www/scripts/
   2. Add the following javascript dependency files to www/scripts/libs if not using jspm.io and code.angularjs.org <script> tags
      1. Rx
      2. Rx-lite
      3. Traceur-runtime
      4. Angular2
      5. SystemJS

Can use JSPM for dynamically loading JavaScript development dependency files. Write ES6 modules and load external modules from CDN.

Notes

May require ADB to install the package manually. Visual Studio 2015 can sometimes create conflicts with devices.   
adb install <location of apk file>

List and uninstall APPs on Android device using ADB

1. Open Command Prompt
2. CD to the ADB folder
   1. C:\Program Files (x86)\Android\android-sdk\platform-tools
3. Find the app
   1. adb root
   2. adb shell
   3. pm list packages
4. Uninstall the app
   1. Adb uninstall <name of package example io.cordova….>

Moving non-ts files to www

Adding BrowserSync as a gulp task

var browserSync = require('browser-sync');

gulp.task(**'browser.sync'**, function () {

browserSync.init({

server: {

baseDir:"./www/"

}

});

// add browserSync.reload to the tasks array to make

// all browsers reload after tasks are complete.

gulp.watch("js/\*.js", ['browser.sync.js-watch']);

});

gulp.task(**'browser.sync.js-watch'**, ['js'], browserSync.reload);