



Platform Specific Parsers contains the following functions :

1. `www_dir` : returns the path for the platform's www directory
2. `config_xml` : returns the path for the platform's config xml file
3. `cordovajs_path` : returns the path from where the platform specific cordova.js is to be copied : `~/cordova/lib/<platform>/<version>/framework/cordova.js`
4. `update_www` : which updates the platform specific www folder (`projectDirectory/platform/<platform name>/www` from the global www folder - `projectDir/www`
5. `update_project` : a config object is passed to this function through which we can extract the app name, app icon, version and package name , and update these in the platform specific code : like update the entries in `activity.info` and `activity.svg` to reflect the new info.

platforms.js
Contains entries of various platforms - name of the platform, host os , url , version , subdirectory

build.js
A combination of prepare and compile command

prepare.js
This file does following :
 1. updates the html content in www folder of the platform - found here : `projectDirectory/platform/<platform name>/www`
 2. updates the app name , package name, icons, version entries through the config xml file : `projectDirectory/config.xml`
 3. Install and Uninstall any files related to different plugins through plugman

compile.js
Executes the script with the help of `superpaw.js` to convert the web stuff to native platform package.

create.js
 *Command - `cordova create dir id name`
 *Creates a new cordova project with the specified directory name, id and name of the project - without any specific platform , platform can be added later with the cordova platform add command

platform.js
 *To add a platform - `cordova platform add <platform-name>`
 *To remove a platform - `cordova platform rm <platform-name>`
 *To list the platform - `cordova platform list`
 *Platform update - update the version of Cordova used for a specific platform; use after updating the CLI
 *check - list platforms which can be updated by 'platform update'
 *For platform add -

1. For the first time if issue the command (that is, no entry corresponding to `~/cordova/lib/<platform>/<version>`) : downloads the project from the project url specified in `platforms.js`
2. Stores the platform code in `~/cordova/lib/<platform>/<version>`, (this is referred whenever next time the command issued)
3. The `add` function runs the executable `create` in `~/cordova/lib/<platform>/<version>/bin/create` file.
4. This executable takes help of `shell.js` to copy the relevant files from the `~/cordova/lib/<platform>/<version>` folder to `projectDirectory/<platform>` folder

- For the platform list command - it checks if the entry in the `platforms.js` or not
- Platform remove command - simply removes the directory from `projectDirectory/platform/<platform>` using the `shell.js` remove command - `shell.rm`
- Platform update copies a newer version from `~/cordova/lib/<platform>/<version>`