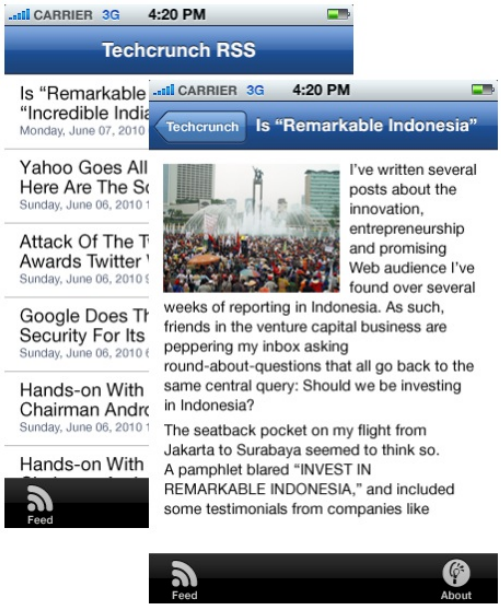
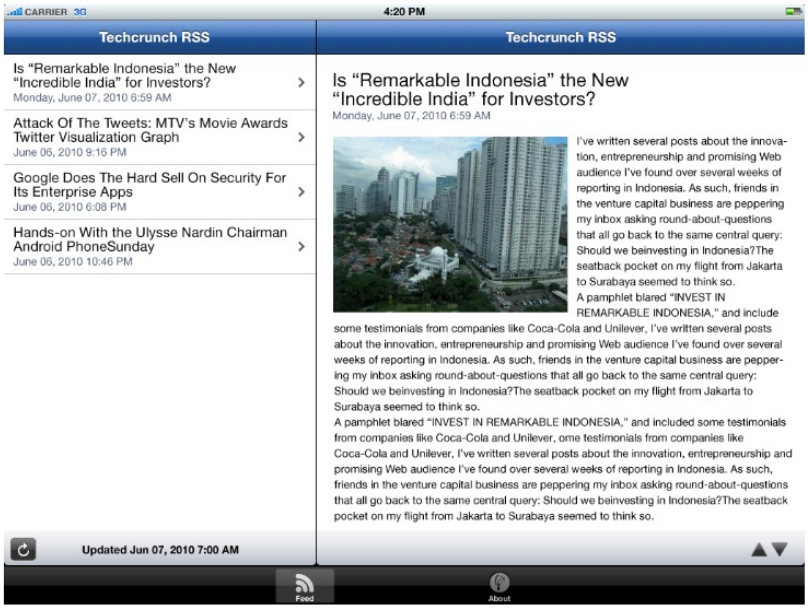
Supporting multiple form-factors using skins

fork and edit tutorial (https://github.ibm.com/MFPSamples/DevCenter/tree/master/tutorials/en/foundation/6.3/advanced-client-side-development/supporting-multiple-form-factors-using-skins.html) | report issue (https://github.ibm.com/MFPSamples/DevCenter/issues/new)

Overview

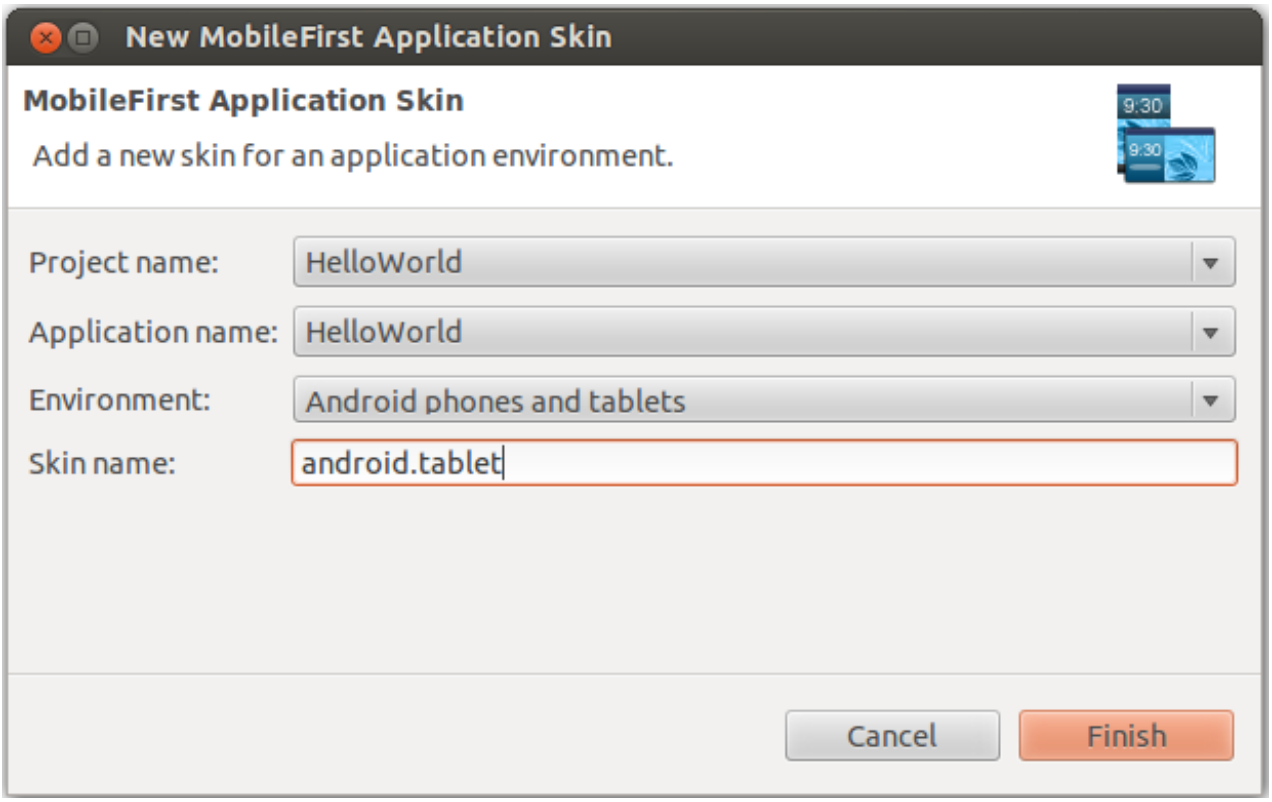
Skins provide support for multiple form factors in a single executable file for devices of the same OS family.
Skins are a sub-variant of an environment.
Skins are packaged together in one application.
The decision on which skin to use is made automatically at run time.

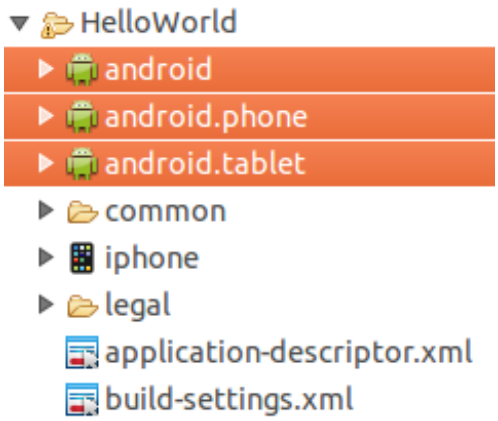




Skin creation

Skins are created by using the IBM MobileFirst Platform Foundation Application Skin wizard.
The skins are placed in directories that are next to the corresponding environment directory.
Before a skin can be added, first add the environment for which you want to create the skin.

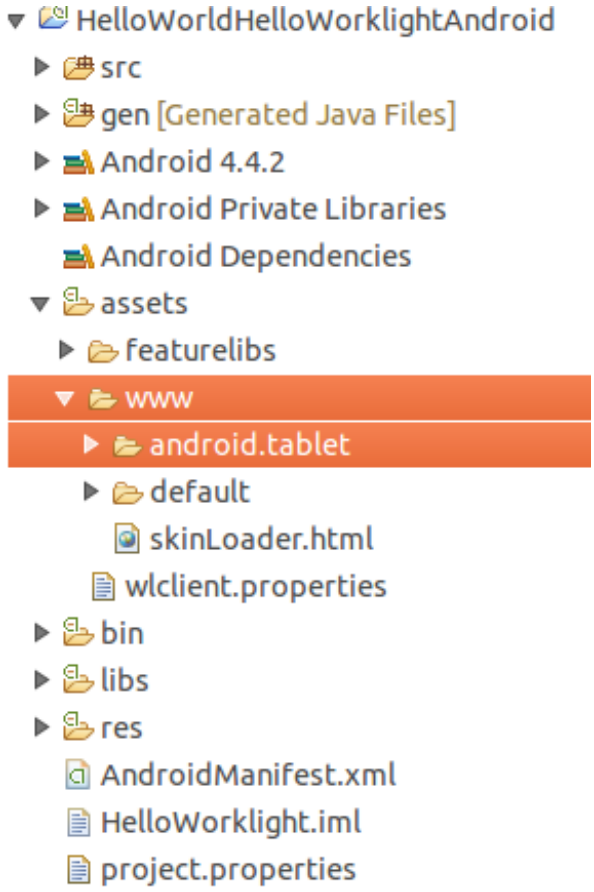




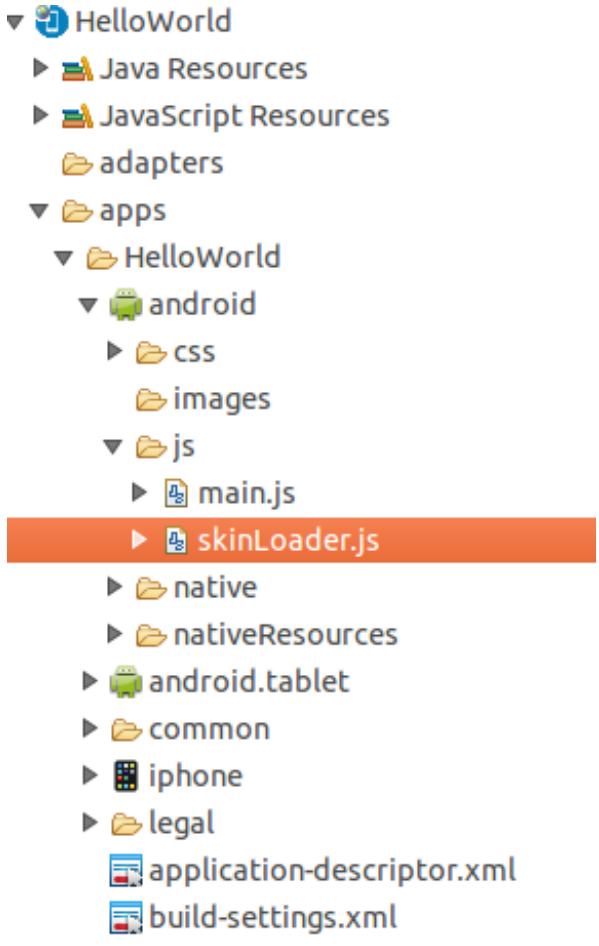
A skin folder contains the following folders: *images*, *CSS*, and *js*.
Create new *.css* and *.js* files in the corresponding folders.
To extend existing code and styling from the default skin, use the same filenames as those of the default skin.
To create new code and styling, use different filenames than those in of the default skin. When you choose this option, you need to copy the HTML file to the skin folder. You must also change the reference to the *.js* and *.css* files in the HEAD element of the HTML file.

Skin packaging

All skins for a specific environment are packaged within the app.



Applying skins at run time



A special, developer-controlled JavaScript file is run at app startup.
It determines which skin to load.
The default *skinLoader.js* contains examples of code.

```
function getSkinName() {
 var skinName = "default";
 if (device.version == "2.2" || device.version == "2.1") {
  skinName = "android.HTML5";<
 }
 return skinName;
}
```

Skin Registration

Skins are automatically registered in *application-descriptor.xml*.
Registration determines the ordering of hierarchy between the common folder, environment, and skin.
If you remove a skin from the project, *application-decsriptor.xml* must be manually modified.

```
   <android version="1.0">
      <skins>
         <skin name="default">
            <folder name="common"/>
            <folder name="android"/>
         </skin>
         <skin name="android.tablet">
            <folder name="common"/>
            <folder name="android"/>
            <folder name="android.tablet"/>
         </skin>
         <skin name="android.phone">
            <folder name="common"/>
            <folder name="android"/>
            <folder name="android.phone"/>
         </skin>
      </skins>
      <worklightSettings include="false"/>
      <security>
         <encryptWebResources enabled="false"/>
         <testWebResourcesChecksum enabled="false" ignoreFileExtensions="png, jpg, jpeg, gif, mp4,
mp3"/>
         <publicSigningKey>Replace this text with the actual public signing key of the certificate used to
sign the APK, available by using the 'Extract public signing key' wizard.</publicSigningKey>
         <packageName>Replace this text with the actual package name of the application, which is the
value of the 'package' attribute in the 'manifest' element in AndroidManifest.xml file.</packageName>
      </security>
   </android>
```