

How to deploy to Blackberry Playbook

Android Virtual Device

The current Playbook is a 1024x600 display with an LCD density of 169 – you can use the Android Virtual Device Manager to create such a device. It should Target Android 2.3.3. I would recommend doing all testing and debugging on an Android Virtual Device as that will debug directly to Basic4Android. There is no way to debug in the Simulator so you should be pretty certain it will work before you move to that stage. Blackberry provides tools to debug directly with the Simulator but that is only for coders using Java/Eclipse. I guess you could take your Basic4Android Java code, put it in Eclipse and then debug there but you are then debugging/coding in Java.

Blackberry Playbook Simulator

The Blackberry Playbook Simulator is 1280x768 with a DPI of 160. This is likely to be the size of a future model of Playbook.

The simulator will give a Phone.Manufacturer of “Research in Motion” and a Phone.Model of “BlackBerry Runtime for Android Apps”. The black bezel of a Playbook is touch sensitive and the Simulator is too. To change orientation, click the bezel in the bottom right and swipe up to the top left. Even if you state your app is of fixed orientation, the Simulator will still try to convert it if you switch. The Playbook has no buttons: the equivalent of the “menu” button is shown by a swipe from top bezel down; the “back” button equivalent is a diagonal swipe from the bottom bezel but coders are recommended to minimize use of this. There is also a back button on the bottom right of the bar –see next.

At the bottom of the screen is a bar that is 35 pixels tall. The button on the right toggles whether this bar is fixed (as shown) or auto-hides. If it is not fixed, you can redisplay it by clicking the black border at the top of the Simulator and swipe downward. If you are designing a full screen app, I guess you may need to account for its presence but I don’t know how you detect it. The simulator is 1270x768 so an app designed for the 1024x600 of the Playbook doesn’t reach the bottom of the simulator screen.



The simulator seems to have problems with redrawing and is very flicker – this is a known problem. RIM recommends giving the Simulator 2 processors in VMware but I don’t see a difference.

There are currently no phones capable of using the new Blackberry OS that runs Android. RIM want apps that are designed for tablets, not phones and may reject an app that is just a very large phone app. Your app should therefore be designed for Tablet use at either 1024x600 and to future proof, design for 1280x768.

Actual Playbook

The real playbook will give a Phone.Manufacturer of “RIM” and a Phone.Model of “BlackBerry Runtime for Android Apps”. On a real Playbook, the bottom edge of your app will be covered by the bar and if you flick the switch to remove it, your app is missing a strip – since you can’t detect that

the bar is gone, you can't repaint to cope. The bar at the bottom of a real Playbook is 45 pixels, not 35 as on a Simulator. My full size app says its Activity dimensions are 1024x555.

The swiping movement for changing orientation does not work on a real Playbook – I guess you just turn the device. I am glad to say that if you fix orientation in your app, the device will rotate to that orientation on running. Curiously, if you switch to other apps, your device will stay fixed as per your app's preferences. I have not tested to see what happens if two different Android apps with different forced orientations are running together.

Icons

Your app icon should be 86x86 with a small transparent border. The icon should ideally look like it is lit from above as the Blackberry gives it shade underneath. Compile your app with this 86x86 icon. However when designing the icon, bear in mind you need to submit a 480x480 pixel PNG image to use as a product icon. If your product is an application, it should match the design of your application icon. Scaling 86x86 to 480x480 will look awful.



Development

The main site for Android developers is

<https://bdsc.webapps.blackberry.com/android/>

This has lots of simple advice that you should read in full. It is not like many development sites that make you feel like you have to read a textbook before starting – more like a “quickstart guide”. You should request code signing keys in advance for when you need to sign and upload your finished product for submission. NB take a note of everything you enter here as you will need to use exactly the same information later. You will have an author-name, Keystore password, a CSK password, a registration PIN and a Developer Zone password once you are all through! The author-name should match the company name you specified when you created a developer certificate. The keys will be sent by email – store them somewhere safe along with all the above names/passwords/PINs. Also, read the release notes to be up-to-date with known bugs before you tear your own hair out. See the section on Submission before choosing your company name.

I recommend keeping your app and package name all in lowercase.

Download command line tools and Simulator from

<https://bdsc.webapps.blackberry.com/android/tool/>

Create your own batch files with the following commands as this will make it easy for you to enter the paths to both your APK file and the SDK.

To verify your APK will work in a Blackberry, use

```
apk2barVerifier "path\file.apk" "path to Android-SDK"
```

To convert your APK file to a BAR file, use

```
apk2bar "path\file.apk" "path to Android-SDK"
```

NB "path to Android-SDK" has no trailing ". To deploy to Playbook simulator:

Run Simulator, click the body icon with the star on his chest and take a note of the IP address.

NB ignore the website information about uploading debug tokens – this is only needed if you are deploying on a real Playbook. The Simulator is always in debug mode.

```
blackberry-deploy -installApp -device 192.168.xxx.yyy -package  
"path\file.bar" -password playbook
```

Replace 192.168.xxx.yyy with your device's IP address. Ignore the error about user authentication failing. The password can be anything except "".

Run all these batch files from the command-line so you can see their output.

The compile-convert-upload-run cycle can thus all be put into a batch file. Make sure you close your app on the PlayBook before you re-deploy. Older versions of the Simulator didn't seem to always delete the old copy so make sure you have the latest version. Also, the "known issues" page says that the Runtime will start to be flaky/slow after several cycles of this – I have a copy of a virgin simulator directory and use a batch file to refresh the virtual machine if needed. If you have VMware workstation, you could create a snapshot of a virgin machine.

For a real Playbook you need to upload a Debug token first and put the device in Development mode. To create a debug token:

```
blackberry-debugtokenrequest -cskpass <CSK-password> -keystore "path  
\client-certificate-xxxxxxxx.p12" -storepass <keystore-password> -  
deviceId 0xHHHHHHHH "path\DebugToken.bar"
```

Where HHHHHHHH is the device id from your Blackberry settings – NB prefix with 0x. Also, replace your client-certificate p12 file name with your file name.

Your convert/deploy cycle is slightly difference since you need to convert the apk file with the debug token and author name:

```
apk2bar "path\file.apk" "path to Android-SDK" -d  
"path\DebugToken.bar" -a "author name"
```

To deploy, you need to put the device in Development mode, which will prompt for a password – this is the password you will use to deploy the app.

```
blackberry-deploy -installApp -device vvv.xxx.yyy.zzz -package  
"path\file.bar" -password actualpassword
```

NB you can't upload bar files converted with a debug token to the simulator so I recommend you have separate batch files for converting/deploying to the simulator and actual device. Your device IP addresses will also be different.

Submission

RIM is very strict about you removing all references to "Android" from your app. I don't know how in depth they look but my first submission was rejected because of this even though I coded to detect

Blackberry and change text accordingly. They might look for the work “Android” in the BAR file. To be sure, I removed all mention of “Android” from the code and switch text files externally with a batch file to toggle between Android and Blackberry development. I would avoid android in a variable name or else obfuscate your code on compilation.

When you sign your app, you are asked to give a company name associated with your developer certificate. Pick a simple short name possibly without spaces. My certificate is associated with a long company name and the signing tool doesn't like it. It doesn't seem to mind if you enter a different name when you get round to signing from what you entered when you registered – you also need this name when you convert APK files for a real Playbook.

Once you are happy it runs, sign and upload at:

<https://bdsc.webapps.blackberry.com/android/bpaa/>

The on-line signing tools use Java and my version of Internet Explorer crashes when I run it so I use Chrome for this stage (I've not tried Firefox). Make sure you submit a *signed* BAR file as if your first submission is unsigned, the current system won't let you submit a further signed one with the same name (RIM are working to fix this).

I think the online tools are easier for signing/uploading but the command line tools are easier for testing. NB you don't want to sign your app until you have finished testing it as each time you sign it, you need to increment the version number. You don't need to do that if your compile-convert-upload-run cycle is on the Simulator or a “Development mode” Playbook and you use the command-line.

The submission webpage is fairly straightforward. I would recommend taking snapshots of your app from the Android Virtual Device but you will need to do some editing – Paint/Irfanview is all you need. You should remove the little phone icons from the top right by cutting and pasting a black rectangle over them. You will need one “Product Featured Image” of 1920x1186 in PNG format. I have no idea why this resolution so just scale your app from 1024x600 to 1920x1125 and then add a bottom border of 61 pixels. You can submit up to 50 screenshots. Again the image limits are weird – 640x640 is a *maximum* so scale your 1024x600 app to 640x375.

I wouldn't try to take screen shots from the Simulator as you need to try to get it to have a settled image – it tends to flick from the previous image state to the current one. To take them from a real Playbook, press the volume - and + keys simultaneously – the shot will be in your Pictures folder.

Good luck!