



APPS ON BLACKBERRY 10: WHAT EVERY DEVELOPER SHOULD KNOW

Get a quick review of what you need to know about creating, deploying, and managing the apps that enterprises rely on today, and will tomorrow, with BlackBerry 10.

 **BlackBerry**

Doing More, With Enterprise Apps and BlackBerry 10

BlackBerry® is focused strategically on the enterprise. With BlackBerry® 10 and BlackBerry® Enterprise Service 10, our goal is to deliver the most useful and secure experience possible. We're invested in bringing enterprises industry-leading, multi-platform management and security solutions, and we're working to provide the best tools, support, management, and developer ecosystem for business applications.

This document is designed to help app developers understand what's involved in both migrating and creating apps for BlackBerry 10. You'll also find out about some of the features that make the BlackBerry 10 experience unique when it comes to developing, deploying, and managing the apps that enterprises will rely on today and tomorrow.

Solving the Challenges of Mobile Applications

BlackBerry 10 solves the biggest challenges typically associated with developing mobile apps.

Out-of-the-box, it provides:

- A connection behind the firewall to any service and data.
- Guaranteed data push, and acknowledgments data has arrived.
- A full application deployment and management solution (BlackBerry World™ and BlackBerry World for Work).
- A secure enterprise container for your application and data (via BlackBerry® Balance™ technology).
- An integrated user experience via BlackBerry Hub.

BlackBerry 10 Software Development Kits (SDKs)

There are four main avenues for BlackBerry app developers.

1. BlackBerry WebWorks

Create your user interface using HTML5 and CSS3, then do your coding or logic in JavaScript. You not only have access to the HTML5 APIs and current JavaScript APIs, but also to BlackBerry® WebWorks™ APIs. These are APIs that BlackBerry has developed to help give you greater access to device features.

BlackBerry WebWorks also allows you to import other JavaScript frameworks that are readily available on the internet. If you need an API that we haven't written into the native BlackBerry 10 experience, you can also create a JavaScript extension – a package you include in your application that's written in native C/C++, that you can reuse in JavaScript. BlackBerry WebWorks is open sourced under the Apache 2.0 license.

With BlackBerry WebWorks, you can also connect to your back-end systems using the MDS Connection Service (MDS CS) – more on this later.

As you're building these applications in BlackBerry WebWorks, the easiest way to test (and there are multiple ways) is using a Chrome extension called Ripple. It allows you to test your web UI and some of the BlackBerry WebWorks-specific APIs right within Chrome. You can also use simulators (available for all three product suites: Java OS, BlackBerry® PlayBook™, and BlackBerry 10). And of course, ultimately, you'll want to test your application on a real device to make sure it functions as you intended.

2. BlackBerry 10 Native Development Kit (NDK)

This next development option boils down to either Cascades or Core Native.

With Cascades, you can build a beautiful user interface (UI) out of the box, so as you'd expect with something like Visual Studio, there are pre-built components. You can use these within your code to make your application look like it's part of the BlackBerry 10 operating system.

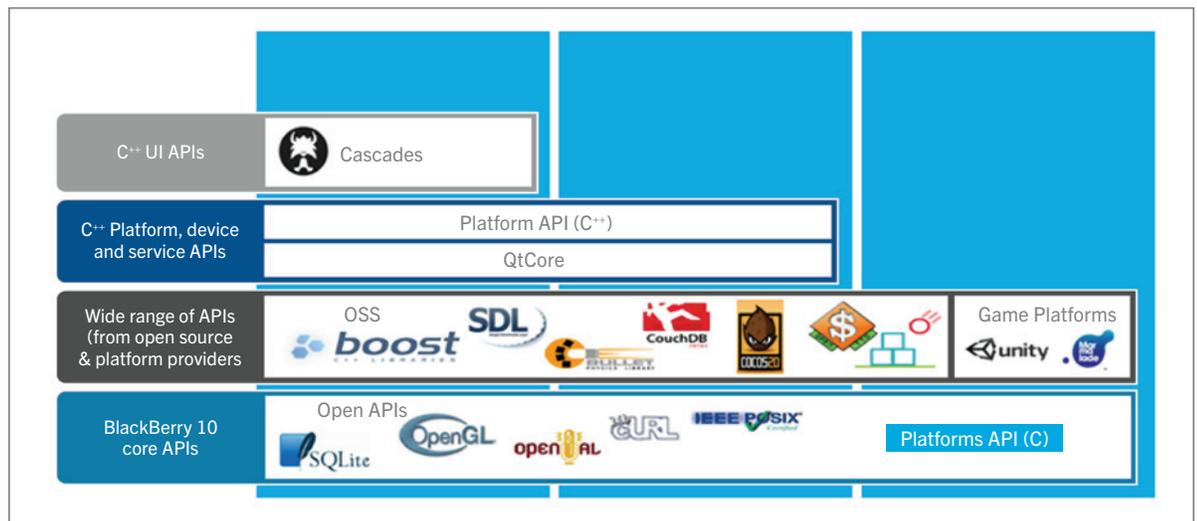
The Core Native side is ideal for porting applications over from existing C/C++ projects (game developers find it very easy to work with, for example).

The image below shows the overall BlackBerry NDK stack: the BlackBerry 10 Core APIs, the APIs from open source projects, the QtCore, and finally, for the Cascades application developers, QML.

3. Adobe AIR Runtime

With the Adobe AIR Runtime on BlackBerry 10, you can take Adobe AIR applications you've already built and adapt them for BlackBerry 10 really easily. You can also embed Flash content within your Adobe AIR application.

And, you can take the Adobe AIR Runtime and create what's called an AIR native extension. You can write native code, and tie that native code to an AIR API, so you can make use of something that might not exist in the AIR Runtime right now within your application.



4. Android Runtime

The final piece of the puzzle is the Android Player, which is a self-contained virtual machine on BlackBerry® Tablet OS 2.1 and BlackBerry 10.

It differs slightly from regular Java or the BlackBerry Java that you may know now, and you can check out the Android SDK to see some of those differences.

If you've written for Android first, you may find that access to some of the Android components that you may expect is restricted. This Android player is only supported in the personal perimeter of BlackBerry Balance.

Migrating to BlackBerry 10

This section is about taking current applications that you've written for BlackBerry® 6 OS, BlackBerry® 7 OS, and moving them over to the BlackBerry 10 platform.

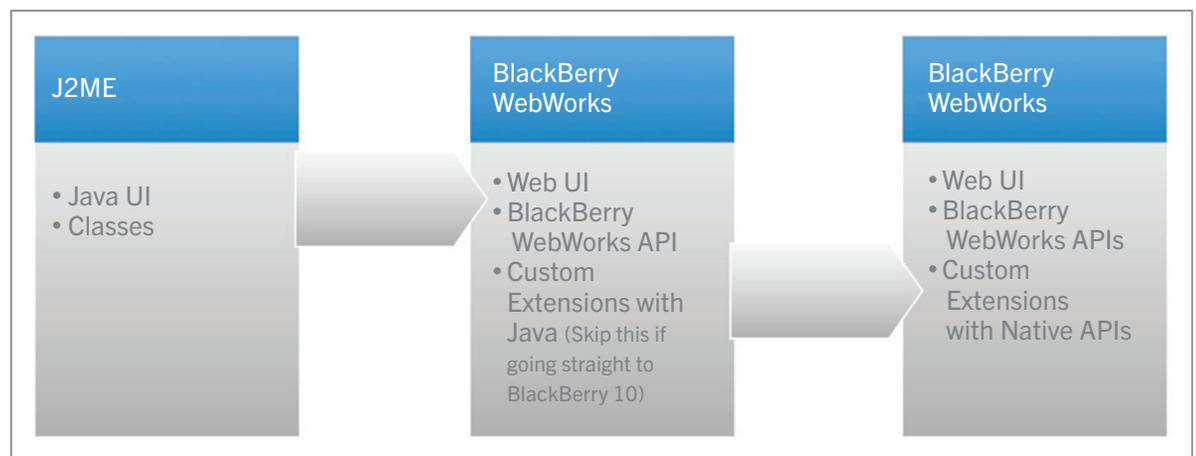
First, let's talk about migrating your Java applications to BlackBerry WebWorks. The two key aspects of J2ME applications are JavaUI and classes (which could be objects or functions).

If you identify which pieces of the puzzle you can migrate over to the WebWorks side, a pattern may emerge. Some of the UI moves very well over to HTML5 and CSS3, and the classes, functions, and objects move well over to JavaScript.

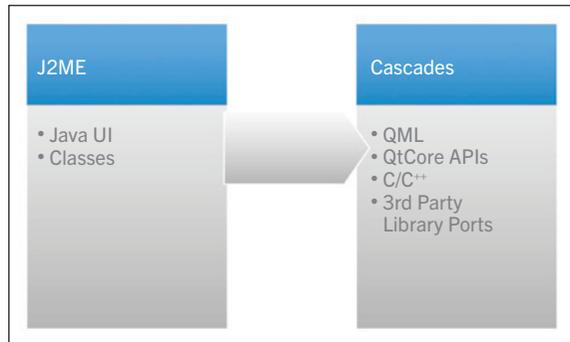
Now you can take some of these classes that you may be already tying into – things like the BlackBerry Calendar or Contacts – and use some of the BlackBerry WebWorks APIs that provide that same functionality, which should be quite a simple migration. If there are classes or APIs you want to access that don't have that functionality inside of BlackBerry WebWorks, you can create a custom JavaScript extension to be able to provide that same access to your application.

Once you've migrated to BlackBerry WebWorks, you can then migrate to BlackBerry 10 easily. You can reuse most of the web UI you've created, the BlackBerry WebWorks APIs, and then take any of the custom JavaScript extensions you created for BlackBerry WebWorks and move those to custom extensions in the Native C/C++.

An important note: this is a migration strategy if you want to start now and you don't have BlackBerry 10 in-house. If you want to go directly from Java to BlackBerry 10, you can skip that middle step and start to port from your Java UI to a BlackBerry 10 UI. And then port your J2ME classes over to either the BlackBerry WebWorks API (where it exists already for those classes), or create your custom extensions.



Migrating to Cascades



The nice thing about migrating to Cascades is that you have that QML or these pre-built components for your UI. First, identify your Java UI classes. You rebuild the components that you want to move into Cascades. Then you take the classes and start to build out the functionality or logic within your application using the C++ and Qt framework.

Again, don't forget you can also port in third-party libraries if you need to.

What About Server Code?

The code that you've written server-side already, will, for the most part, remain the same, as long as you're still providing that data in a standard format – like XML, JSON, or OData.

Push differs slightly. For those of you familiar with BlackBerry Push now: you can still use BlackBerry Push with ports, but the ports become what's called the ApplicationID. The ApplicationID is how your application knows how to receive that data.

Enterprise Architecture and Harnessing the Power of BlackBerry

From a development standpoint, BlackBerry Balance technology means a lot of the heavy work gets done for you. For example, with features like the automatic data-at-rest encryption, you don't have to worry about re-encrypting the data you've pulled down from your server because BlackBerry Balance already encrypts that perimeter within the device.

The data within the application data can't be copied and pasted into personal applications or emails, so you don't have to worry about users leaking information.

This all comes down to saving you, the developer, time – because you don't have to worry about writing that logic into your code.

BlackBerry World for Work

This is a way for the enterprise to provide both mandatory and optional work applications. These apps are only accessible from the 'Work' perimeter. It differentiates BlackBerry World-hosted content from BlackBerry Device Service-hosted content, and once installed, enterprise apps appear in 'Work' perimeter and are available in My World.

BlackBerry Mobile Data System Connection Service

This is one of the most important pieces of the app story, from an enterprise developer's standpoint.

BlackBerry® Mobile Data System Connection Service (MDS CS) provides you with an 'always-on', bi-directional VPN tunnel into your infrastructure that is AES-encrypted, so you don't have to worry about people stealing data out of the air.

It also has managed connectivity to your back-end system, so from an administrator standpoint, you can actually control which internal resources that application can access. When you're developing your application, you can make those connections as if you were behind the firewall, as you would for a laptop or any internal device. Not having to setup a VPN tunnel every time you want to make a connection within your application saves a lot of time.

Full Data Push and Acknowledge

You can push out an alert to the app, and you can also push up to 8kbs of actual content or data, and have it process in the background. If the user is out of coverage, the data will become available when the device reconnects, and if you need it, you'll obtain the acknowledgment from the device back to your server that it's received the data. There are a number of new capabilities associated with BlackBerry Push on BlackBerry 10. For more detail on this and other apps-related topics, see the full, on-demand webcast on enterprise apps (link below).

To learn more about apps and BlackBerry 10:

- Catch the rest of our [BlackBerry 10 webcast series](#) including an on-demand BlackBerry 10 Apps webcast
- Visit the [BlackBerry Developer Site](#)
- Check out our Github site: blackberry.github.com
- And for more on BlackBerry 10, visit: blackberry.com/blackberry10ready

Android is a trademark of Google Inc.

©2013 Research In Motion Limited. All rights reserved. BlackBerry, RIM, Research In Motion and related trademarks, names and logos are the property of Research In Motion Limited and are registered and/or used in the U.S. and countries around the world. All other trademarks are the property of their respective owners.