
Determining the Best Approach

The remaining chapters of this book cover the capabilities of the BlackBerry application platform and then dig into each application development option in detail. Before any BlackBerry application development project begins, the developer must first make some decisions regarding the following:

- How the application will access server-based data
- Whether the application will manipulate a local copy of the data or access the data on the server
- Whether the application's data will be pushed to the application or pulled down from the server
- Whether the application should be purchased or built
- Which of the platform's supported application technologies will be used to build the application

This chapter covers the options that must be considered when addressing these questions

2.1 Local Versus Server-Based Data

The decision to store data on a device or access it all via a server is a decision that is made based on the needs of the application's users. At the same time, the

developer must consider how this decision will affect the device's performance, battery life, and data plan usage.

The first portable devices (Palm Pilot, Windows CE, the Apple Newton, and others), didn't usually have any wireless network connectivity, unless the user added a Wi-Fi card to the device. For applications targeting these devices, developers usually built some mechanism to synchronize server-based data with the device while it was cradled (connected via cable to a desktop PC). Although this worked, there were issues related to how difficult it was to ensure that the mobile user had the most recent copy of the data. Additionally, it was painful to have two mobile users, with their own copy of the data, modify the same record before synchronizing it with the server. Although some users still own stand-alone mobile devices, the more common device is a network-connected smartphone that has more options available to it for sharing data with others.

An application running on a network-connected smartphone, such as the BlackBerry, can easily connect to a server to interact with data stored there, whether it's for a Customer Relationship Management (CRM) system, Enterprise Resource Planning (ERP) application, or any other type of client-server application. The standard mode for these applications, when interacting with corporate data, is for the data to be stored on a server, allowing multiple clients to access it simultaneously and ensure the users always have access to the most up-to-date data.

As good as the mobile networks are, there are still many situations where it makes sense to have the data local to the device. For mobile sales teams or service technicians that work in areas with limited wireless coverage, they might need to manipulate all or some of the application's data on the device and synchronize updates with the server when convenient.

In many cases, a hybrid approach is an appropriate option. With this approach, a database of relatively stable data, such as part numbers, store locations, zip/postal codes, or area/city codes, is stored on the device and rarely updated. More frequently updated data, such as sales orders, shipment status, invoices and customer contact information, is accessed from the server (over the wireless network) as needed. On the consumer side, a good example is Facebook—older messages never change and can be kept local to the device; friend information can always be retrieved from the network when needed.

With any network-connected application, a developer must consider the nature of the wireless network connection and check for connectivity before attempting any connection and recover gracefully when a connection is not available or terminates in the middle of a transmission. As a best practice, an application must

be able to queue up any network request and deliver and/or retrieve data whenever the connection becomes available. In cases where the mobile user is not waiting for an update, the application should be able to reconnect and deliver the data without user intervention. The application can let the user know data has been queued, but it should not require that the user act to reestablish a connection that has been terminated.

The organization sponsoring the application will have a good understanding of the wireless network coverage options for its territories and decide on the best option for its users based on cost and performance.

2.2 Internal Versus External Access

What is the relationship between the owner of the application data and the mobile user accessing it? If the mobile user is part of the organization that owns the data, it's likely that the BlackBerry device's connection to the application server will be provided through the BlackBerry Mobile Data System (MDS) described in Chapter 4, "The BlackBerry Mobile Data System (MDS)."

If the mobile user is not part of the organization owning the data, it's likely that the data is provided as part of a subscription service or freely available on the Internet. With this option, the application accesses the server through an open firewall port through the organization's Internet connection.

In some cases, an application will use both connections, getting some data from an internal server and other data from an external data source. In this case, the application must keep track of multiple connections simultaneously. The best practice for this scenario is to let the internal application server connect to both internal and external data sources and reduce the load placed on the mobile device.

The options a developer might use for both internal and external network connections are described in detail in Chapter 3, "The Connected BlackBerry."

2.3 Pushing Versus Pulling the Data

Depending on how the application's users interact with the data or how often they need access to the data, it might be easier to just push updates to the application rather than forcing the user to periodically request updates from the server.

If the application's users interact with the server by performing just-in-time queries or lookups of data, the application must accommodate that requirement. On

the other hand, if the application's user needs access to data that only changes periodically or changes on a triggered event (such as an order shipping or a back-ordered part becoming available), the application can minimize the load on the user (forcing him to periodically check for new data) and just push the data to the application behind the scenes when updates are available.

Fortunately, each of the BlackBerry application types (web and Java, enterprise and consumer) support push, so there is no limitation on which application type is used for the application. The options a developer can use to push data to device applications are described in Chapters 5, 6, and 7.

2.4 Building Versus Buying

For a developer, the decision to build or buy is sometimes difficult to make. Depending on the cost of a third-party solution, it might be less expensive to purchase a solution rather than build it.

Organizations build applications to help solve specific problems that the business faces. If the business problem the application is trying to solve is unique to your business, it is not likely that a solution will be available to purchase. If the business problem is a common problem encountered by many businesses, the likelihood that a third-party solution exists is much higher. Even if the target audience for an application is part of a niche or vertical market, if enough companies have the problem, there might be a solution available.

With third-party solutions, the responsibility for building, testing, and maintaining the application belongs to someone else. The organization purchasing the solution can use the application, knowing that someone else is responsible for it if it breaks. If a new version of the backend application server or mobile platform becomes available, it's the vendor's responsibility to build support for the new version. In the case of more sophisticated applications, because the cost of building the application is distributed across multiple customers, the overall entry cost for the application is low.

If the mobile application is communicating with a proprietary system that your company developed, it's likely that you have to build the solution yourself. If the business problem being solved is unique, but you're talking to an industry standard application platform (such as WebSphere, SAP, Remedy, PeopleSoft, and so on), you can likely leverage integration tools provided by RIM's ISV Alliance Partners to provide the connectivity; you would only be responsible for building the appropriate business logic.

For consumer applications, this is much simpler: It's just a question of the cost for the application (free is best, of course) versus how much is required to build the application yourself.

2.5 Selecting a Development Platform

The most important decision a mobile developer makes is the selection of the development platform/language to use for the application:

- The first thing to consider when selecting a platform is whether the client application already exists for another platform. If the application is already available for the desktop or another mobile platform, the developer will usually (but not always) use a similar technology to build the BlackBerry or mobile version.
- If the application is accessed today through a web browser, it's likely that BlackBerry users will also be able to use the browser. A browser-based application designed for a smaller screen should function just fine on the BlackBerry browser, as long as the application doesn't use ActiveX controls or Java applets. There are limitations on what a BlackBerry browser can and cannot do, which are covered in detail in Chapters 8 and 9.
- If an existing browser-based application runs on Google Gears, it should run on BlackBerry devices that support Gears (BlackBerry Device Software 5.0 and higher).
- If the application exists today as a rich client (non-browser) application, the situation is more difficult. If the existing application is written in Java, there is a chance that the application can be rewritten to run on the BlackBerry. If the existing application is written in Java Micro Edition (JME) and conforms to Mobile Information Device Profile (MIDP) and/or Connected Limited Device Configuration (CLDC) standards, it should run on the BlackBerry with limited modifications. (Chapter 11, "Building BlackBerry Java Applications," covers this in detail.) If the application was written for the Android platform (Android uses Java, but a different flavor of Java), the application must be rewritten for the BlackBerry platform. If the existing application was written in another technology, such as Delphi, any of the .NET languages, Objective-C (Mac OS, iPhone) and others, the application needs to be rewritten for the BlackBerry platform.
- If it's a new application or one created just for a BlackBerry device, select either of the available choices: either Browser or Java. We'll discuss these choices in the following sections.

2.5.1 Browser-Based Applications

Browser-based applications provide the best cross-platform option for mobile devices, and there is no shortage of developers/designers with the skills to build the applications. If the application will consist of static pages or the server-based data for the application can be easily served up through a browser interface, building the application in the browser is a good choice.

Browser-based applications are not the fastest option for mobile devices, so the developer needs to weigh the benefits of easy development against the performance issues inherent in the technology. Although the use of Asynchronous JavaScript and XML (AJAX) can reduce the amount of data delivered to the browser (by not requiring the entire page to update for a new request), in general, the browser is slower than other BlackBerry application technologies, even on high-speed third generation (3G) networks. This is because of the performance costs in setting up and tearing down a high speed connection to the server, and because a web page typically includes both content and layout information, more than just the displayed data is sent across the wireless network.

Build a browser-based application if the users are expecting to use the browser to access the application, if the developer's skills are limited to browser development and/or if the application must run on any mobile device without any additional work. Browser applications are also beneficial in that there is not a client application that must be deployed to mobile devices. With browser applications, the application can be updated at any time without impacting the application's users.

If the requirement is to build a browser application that interacts with local data (calendar, contacts, tasks, notes, and so on) on the BlackBerry device, the application should be created using BlackBerry Widgets or the Google Gears application platform.

Many existing web applications are built on a web services platform where the HTML delivered to the browser is created by applying an Extensible Stylesheet Language Transformation (XSLT) to the output from a web service. It may be possible, instead, to consume the service directly from a rich client application on the BlackBerry. This approach likely provides the mobile user with better performance for the application.

2.5.2 Java Applications

The developer that wants to provide the best possible experience for BlackBerry application users will, in most cases, develop the application using Java. The

applications that BlackBerry owners use most often (Messaging, Calendar, Contacts, Tasks, Browser, Phone, Camera, and more) are all Java applications. Developers that want their applications to look, feel, and operate like the other applications on the BlackBerry will build their applications using Java. A Java application provides the most robust capabilities to the BlackBerry developer but, at the same time, is the most difficult application to build. Java applications are usually hand-crafted; there are not any graphical application builders for BlackBerry Java applications. Because of how robust and capable Java is, it usually takes a lot of code to accomplish many things that are easy to do in other languages.

Custom Java applications provide a rich client experience and have access to most, but not all, of the BlackBerry device's capabilities. RIM reserves some capabilities for its own use, but the list of restricted capabilities is relatively short compared to what else can be done on the device. Chapter 11 describes the Java application development options and capabilities.

2.5.3 MDS Runtime Applications

The MDS Runtime Environment was designed to allow nondevelopers to build rich client applications for the BlackBerry that look and feel like applications written in Java. For these applications, RIM created a special development environment that allows these applications to be built through a point and click, wizard-driven interface. The applications run in a special runtime environment that does not have access to the full suite of Java API's. The applications are much easier to build than Java applications but cannot provide all the functionality that Java applications can.

RIM announced end of life for MDS Runtime and the associated development tools (MDS Studio and the BlackBerry Plug-In for Microsoft Visual Studio versions 1.0 and 1.1) for December 31, 2009. For that reason, MDS Runtime applications and the tools used to build MDS Runtime applications will not be discussed in this book.

2.6 Choosing the Right Tools

When it comes to building mobile applications for BlackBerry, the required tools are readily available. For browser-based applications, you can continue to use the tools currently being used to build the pages. The only difference for the developer is the size and capabilities of the pages being built, the technologies behind them (HTML, XHTML, XML, CSS, and JavaScript, for example) stay

the same. RIM recently announced development tools that allow developers to build and test mobile web applications in both Microsoft Visual Studio and Eclipse.

For building rich client applications for BlackBerry, all the tools you need are freely available from RIM. The tools, all of which only run under Microsoft Windows, are free downloads from the BlackBerry Developer's website (www.blackberry.com/developers), as shown in Figure 2.1. Developers can build and test Browser and Java applications using the Eclipse Integrated Development Environment (IDE) and browser applications using Microsoft Visual Studio. These tools and how they're used are described in subsequent chapters.

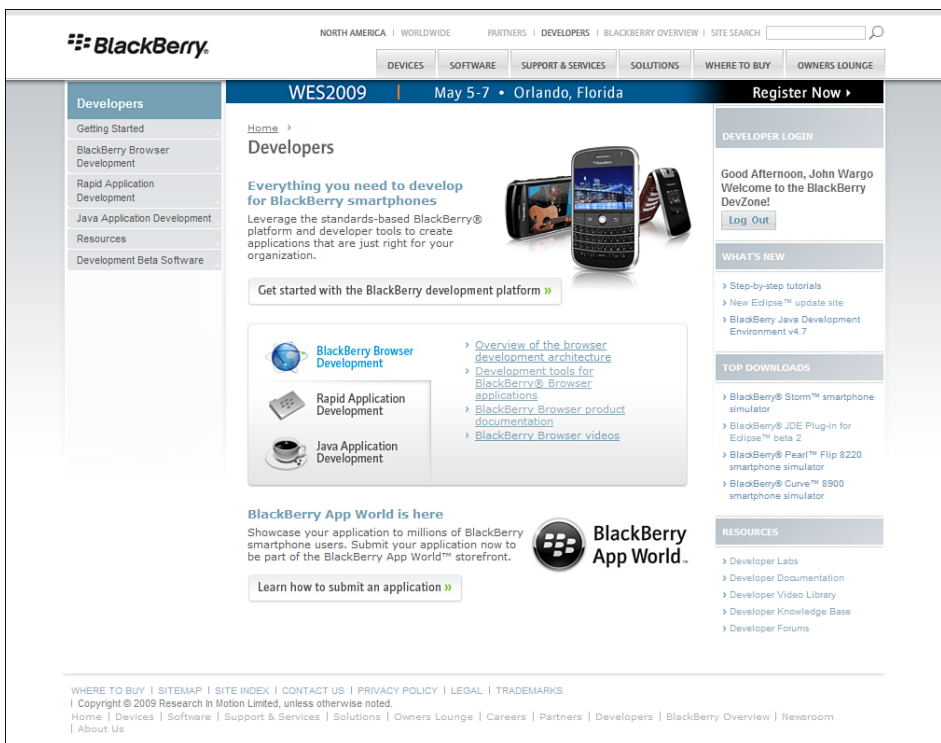


Figure 2.1 BlackBerry Developer's website

Some Java developers are only comfortable with the IDE they have been working in for years. Java developers who do not like Eclipse and do not want to use the RIM JDE can still develop applications for BlackBerry using the Java IDE of

their choice. The BlackBerry JDE Component Package can be integrated with other IDEs to build and test BlackBerry applications.

2.7 Additional Resources

A link to the following resource is available at www.bbdevfundamentals.com.

To highlight some of the choices an organization makes when deciding to build mobile applications, RIM published a guide called *Developing an Application Mobilization Plan for your Business*, located at http://na.blackberry.com/eng/services/Developing_an_Application_Mobilization_Plan_for_your_Business.pdf.

This page intentionally left blank