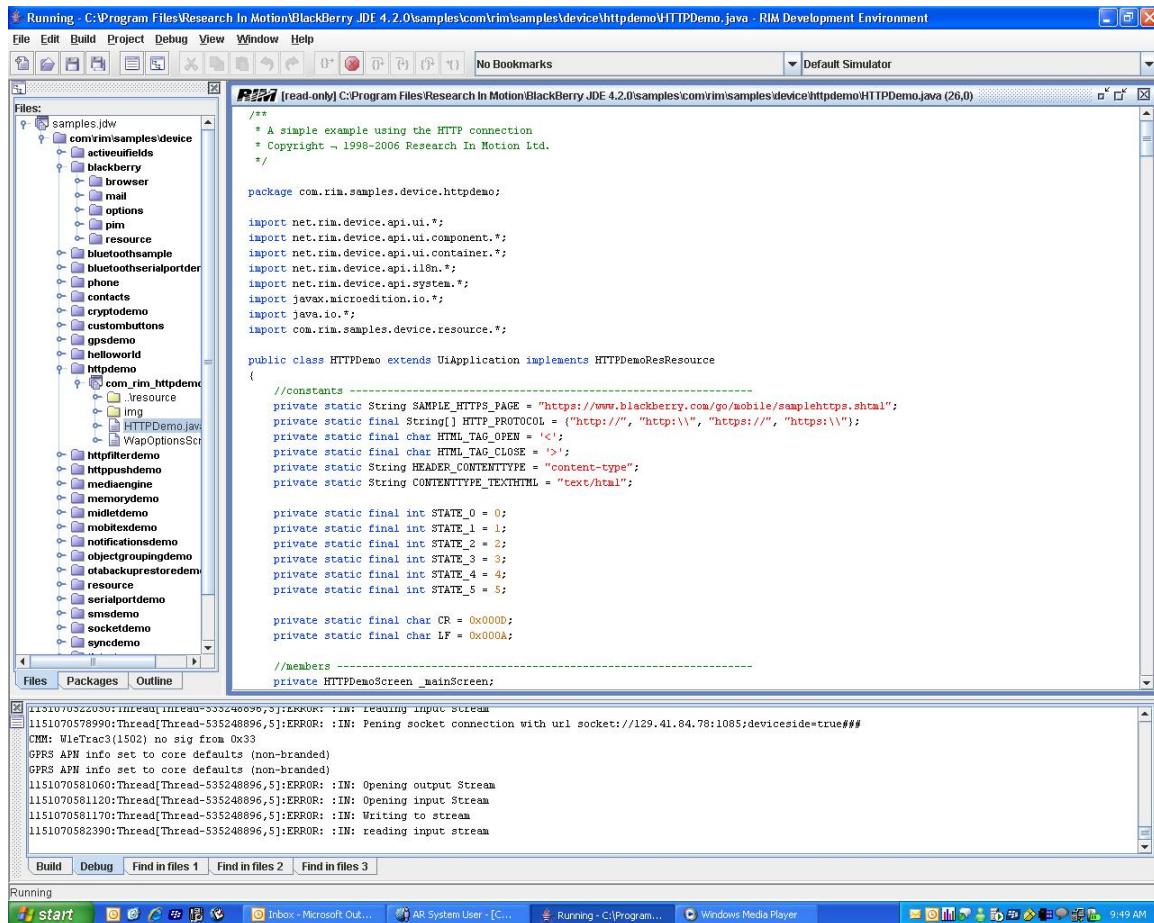


BlackBerry Java Applications for Accessing SAP Applications

BlackBerry Java Development Environment

The *BlackBerry Java Development Environment (JDE)* is a fully integrated development and simulation environment for building custom Java applications for BlackBerry devices. With the BlackBerry JDE, developers can build custom client applications using the Java Micro Edition (Java ME) programming language. Following is a screen shot of the BlackBerry JDE:



BlackBerry Java Development Environment (JDE)

Java Micro Edition (ME) on BlackBerry

Java ME uses the same syntax and programming model as the standard Java language that is used for developing desktop and server-side applications. It is an object-oriented programming language that provides a standard set of libraries and APIs that developers can use to build customized applications. However, there are two main differences that you need to consider when using Java to develop applications for BlackBerry rather than a desktop or server:

- Available Java ME and BlackBerry API libraries
- Designing applications to run on wireless handheld devices

(i) Java ME and BlackBerry API Libraries

Java ME is an industry standard which defines common sets of Java APIs for different types of mobile and embedded devices. Within Java ME there are several sub-categories, or profiles, defined which

represent different classes of mobile and embedded devices. For example, the Java ME *Mobile Information Device Profile (MIDP)* was defined to address the API needs of constrained mobile devices with a user interface, such as cellular phones or personal digital assistants. Other types of embedded devices, such as TV set-top box and vending machine controllers, are covered by other profiles within Java ME. BlackBerry devices, along with many other wireless handhelds, support the Java ME MIDP standard. The details of the MIDP specification are not critical for this discussion, however, it is important to understand that MIDP provides a common API set that any wireless handheld can support, no matter what its underlying operating system is. Therefore, developers can often build one Java application using the MIDP standard APIs and have that application run on many different types of handhelds.

So BlackBerry devices and the BlackBerry JDE support the Java ME MIDP standard, which provides a good core set of Java APIs for building mobile applications. BlackBerry devices also support a large set of additional Java APIs that are not part of the MIDP specification. These are additional APIs that RIM has exposed to provide developers with more features and capabilities when developing for BlackBerry devices. It is not necessary to use these additional APIs in your applications, but they can often provide greater features and functionality over what is available in the standard MIDP API libraries. Between the MIDP standard APIs and the BlackBerry-specific Java API extensions, the following types of APIs are available when developing your applications:

- **User Interface APIs:** Used to create screens, menu items and all components of the user interface.
- **Persistent Data Storage APIs:** Used to store custom data locally within your application. Note that BlackBerry devices do not provide a relational database model.
- **Networking and I/O APIs:** Used to establish network connections and read/write data to a server-side application
- **Event Listeners:** Used to respond to user- or system-initiated events on the device.
- **Application Integration APIs:** Used to integrate with the existing BlackBerry email, phone, calendar, contacts, browser and task list applications
- **Utilities:** Additional APIs for data encryption and compression, XML parsing, Bluetooth connectivity, etc.

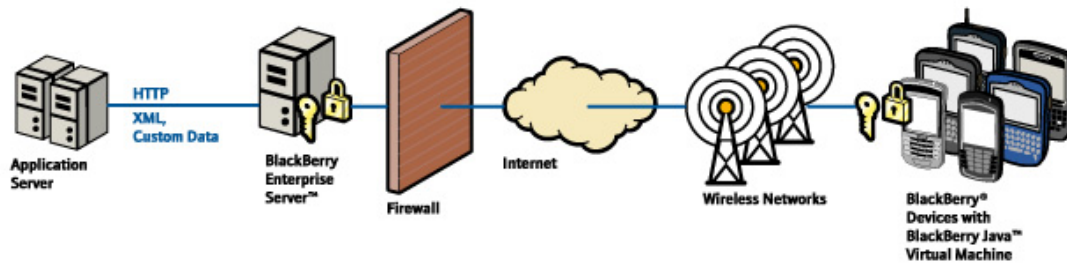
(ii) Designing Applications to Run on Constrained Mobile Devices

When designing Java applications for BlackBerry devices, you must consider the inherent constraints of running applications on a handheld wireless device. Following is a list of items to consider when designing your Java applications for BlackBerry:

- **Processor Speed and RAM:** Because mobile devices do not have the same processing power as the average server or desktop computer, you should strive to optimize the performance of your application and minimize object overhead.
- **Network Connectivity:** Users may roam in and out of network coverage and your applications should be designed to support this. As a result, you can not always rely on a permanent connection to the server and may need to consider how to manage asynchronous transactions.
- **Network Speeds and Bandwidth:** Wireless networks can impose limitations on network speed and latency. Therefore, you should design your applications to send as little data as possible over the wireless network.
- **Multi-Threading:** BlackBerry devices provide a multi-threaded environment, allowing your applications to spawn background threads to manage heavy processing and networking tasks.

Managing Wireless Connectivity and Back-End Integration

Java applications for BlackBerry can leverage the secure connection to the intranet provided by the BlackBerry Enterprise Server and BlackBerry MDS Services. This means that you don't have to worry about how to connect through the corporate firewall from the BlackBerry device or how to encrypt the data. Following is a diagram that illustrates the connectivity model for Java applications for BlackBerry:



Java Applications for BlackBerry: Connectivity Model

When developing your applications, you must consider which protocols you are going to use to connect to the back-end systems and how you are going to manage the exchange of data to ensure reliability and data integrity.

Wireless Transport Protocols

Java applications for BlackBerry can use standard HTTP or TCP socket protocols to communicate with back-end applications. Most applications use HTTP as the transport protocol, using GET requests to fetch data and POST requests to submit data. Standard HTTP response codes are used to notify the application whether or not the connection was successful. When using HTTP as the transport protocol, you have to ensure that your back-end application can support these connections. Thankfully most modern programming languages, including Java Enterprise Edition, have built-in support for handling HTTP-based communications. When using HTTP, it is up to the developer to define the formatting for the actual data within the message. Though you can use simple tokenized or delimited text strings for simplicity, it is typically recommended that you use XML to format your data if possible. The BlackBerry Java APIs provide an XML parser and generator to support this.

To Push, or Not to Push?

For most custom wireless applications, the client will initiate the connections to the back-end system. However, there may be situations where you want the back-end application to proactively push data to the client to provide a real-time notification or data update. With BlackBerry, you can do this by leveraging the HTTP PUSH feature of BlackBerry MDS Services. The BlackBerry MDS Services provide a standard push interface that can be used by any back-end application. The back-end application simply establishes an HTTP POST connection to the BlackBerry MDS Services and provides an XML document that describes the target device, the content to be pushed and other configuration parameters describing how and when the data should be delivered. When the data arrives on the device, the application is notified and can take any type of action required (alert the user, save the data locally, etc.).

Connecting Java Applications for BlackBerry to SAP Applications

The BlackBerry JDE can be used to develop a wide variety of wireless applications that connect to SAP applications using standard HTTP protocols. Some common ways in which Java applications for BlackBerry devices may integrate with SAP include:

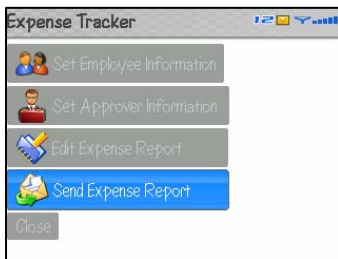
- **Data Retrieval:** Client application for fetching data from SAP using HTTP GET requests.
- **Data Submission:** Form-based client application for submitting data to SAP using HTTP POST connections.
- **Data Synchronization:** Client application allowing offline data storage that synchronizes local updates to the SAP back-end using HTTP POST connections.
- **Data Push and Notifications:** Server-side application that proactively pushes data to a client on the BlackBerry based on an event or trigger.

Using HTTP to Connect to SAP

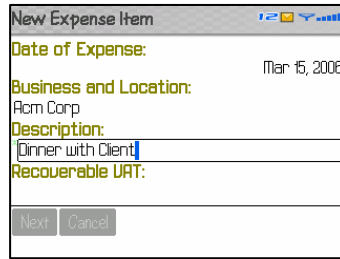
To get your Java application for BlackBerry to talk to SAP, you need to have a back-end application that can support HTTP communications. There may be several ways to do this, depending on the specific environment, but the most common approach seems to be using a Java Servlet to handle the HTTP communications with the BlackBerry application. This Java Servlet would then use JCo on the back-end to interface with the SAP system itself. As mentioned above, it is often best to use XML for formatting the data within these HTTP communications. Both the Java application on BlackBerry and the Java Servlet can be designed to parse and generate XML documents that use the same XML schema.

If you are considering using HTTP PUSH functionality in your applications, you can achieve this in the same way as above. You simply have to write an application that can initiate an HTTP POST to the BlackBerry MDS Services based on an event or trigger. Again, a Java Servlet or Enterprise Java Beans (EJB's) could be used for this. The Servlet or EJB would connect to SAP using JCo to fetch the proper data and would then establish an HTTP POST connection to the BlackBerry MDS Services to push the content out. It is up to the developer to determine what should trigger the push, who should receive the data and what the application should do when the data arrives.

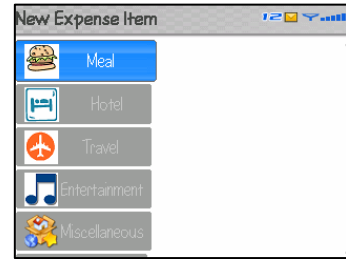
Following are some BlackBerry device screen shots that illustrate a custom Java application for BlackBerry to manage Time and Expense information:



Time & Expense – Options



Time & Expense – Details



Time & Expense – Details

Getting More Information

For more information on developing Java applications for BlackBerry, please refer to the BlackBerry Developer Zone located here:

www.blackberry.com/developers

For more information on managing HTTP connections from a Java application for BlackBerry, please refer to the BlackBerry Java Developer Guide, Volume 1 located on the BlackBerry Developer Zone:

www.blackberry.com/developers

For more information on using Java Servlets and other technologies to handle HTTP connections from an external client application, please refer to your SAP product documentation.

© 2006 Research In Motion Limited. All Rights Reserved. The BlackBerry and RIM families of related marks, images and symbols are the exclusive properties of Research In Motion Limited. RIM, Research In Motion, "Always On, Always Connected", the "envelope in motion" symbol and BlackBerry are registered with the U.S. Patent and Trademark Office and may be pending or registered in other countries.

Mobitex is a registered trademark of the Swedish Telecommunications Administration. All other brands, product names, company names, trademarks and service marks are the properties of their respective owners.

The BlackBerry handheld and/or associated software are protected by copyright, international treaties and various patents, including one or more of the following U.S. patents: 6,278,442; 6,271,605; 6,219,694; 6,075,470; 6,073,318; D,445,428; D,433,460; D,416,256. Other patents are registered or pending in various countries around the world. Please visit www.rim.com/patents.shtml for a current listing of applicable patents.

This document is provided "as is" and Research In Motion Limited and its affiliated companies ("RIM") assume no responsibility for any typographical, technical or other inaccuracies in this document. RIM reserves the right to periodically change information that is contained in this document; however, RIM makes no commitment to provide any such changes, updates, enhancements or other additions to this document to you in a timely manner or at all. RIM MAKES NO REPRESENTATIONS, WARRANTIES, CONDITIONS OR COVENANTS, EITHER EXPRESS OR IMPLIED (INCLUDING WITHOUT LIMITATION, ANY EXPRESS OR IMPLIED WARRANTIES OR CONDITIONS OF FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, MERCHANTABILITY, DURABILITY, TITLE, OR RELATED TO THE PERFORMANCE OR NON-PERFORMANCE OF ANY SOFTWARE REFERENCED HEREIN OR PERFORMANCE OF ANY SERVICES REFERENCED HEREIN). IN CONNECTION WITH YOUR USE OF THIS DOCUMENTATION, NEITHER RIM NOR ITS RESPECTIVE DIRECTORS, OFFICERS, EMPLOYEES OR CONSULTANTS SHALL BE LIABLE TO YOU FOR ANY DAMAGES WHATSOEVER BE THEY DIRECT, ECONOMIC, COMMERCIAL, SPECIAL, CONSEQUENTIAL, INCIDENTAL, EXEMPLARY OR INDIRECT DAMAGES, EVEN IF RIM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, INCLUDING WITHOUT LIMITATION, LOSS OF BUSINESS REVENUE OR EARNINGS, LOST DATA, DAMAGES CAUSED BY DELAYS, LOST PROFITS, OR A FAILURE TO REALIZE EXPECTED SAVINGS.

This document might contain references to third party sources of information, hardware or software, products or services and/or third party web sites (collectively the "Third-Party Information"). RIM does not control, and is not responsible for, any Third-Party Information, including, without limitation the content, accuracy, copyright compliance, compatibility, performance, trustworthiness, legality, decency, links, or any other aspect of Third-Party Information. The inclusion of Third-Party Information in this document does not imply endorsement by RIM of the Third Party Information or the third party in any way. Installation and use of Third Party Information with RIM's products and services may require one or more patent, trademark or copyright licenses in order to avoid infringement of the intellectual property rights of others. Any dealings with Third Party Information, including, without limitation, compliance with applicable licenses and terms and conditions, are solely between you and the third party. You are solely responsible for determining whether such third party licenses are required and are responsible for acquiring any such licenses relating to Third Party Information. To the extent that such intellectual property licenses may be required, RIM expressly recommends that you do not install or use Third Party Information until all such applicable licenses have been acquired by you or on your behalf. Your use of Third Party Information shall be governed by and subject to you agreeing to the terms of the Third Party Information licenses. Any Third Party Information that is provided with RIM's products and services is provided "as is". RIM makes no representation, warranty or guarantee whatsoever in relation to the Third Party Information and RIM assumes no liability whatsoever in relation to the Third Party Information even if RIM has been advised of the possibility of such damages or can anticipate such damages.

Certain features outlined in this document require a minimum version of BlackBerry Enterprise Server software, BlackBerry Desktop Software, and/or BlackBerry Handheld Software and may require additional application development or third party products and/or services for access to corporate applications.