

BlackBerry MDS Studio

Version 4.1

Feature and Technical Overview

BlackBerry MDS Studio Version 4.1 Feature and Technical Overview

Last modified: 22 March 2006

Part number: SWD_X_MDS(EN)-004.003

At the time of publication, this documentation is based on BlackBerry MDS Studio Version 4.1.

Send us your comments on **product documentation**: <https://www.blackberry.com/DocsFeedback>.

©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, BlackBerry, "Always On, Always Connected" and the "envelope in motion" symbol are registered with the U.S. Patent and Trademark Office and may be pending or registered in other countries.

Sun, Java, J2ME, and JavaScript are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All other brands, product names, company names, trademarks and service marks are the properties of their respective owners.

The BlackBerry device 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; D445,428; D433,460; D416,256. Other patents are registered or pending in various countries around the world. Visit www.rim.com/patents.shtml for a list of RIM (as hereinafter defined) 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. In order to protect RIM proprietary and confidential information and/or trade secrets, this document may describe some aspects of RIM technology in generalized terms. 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.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>) and/or licensed pursuant to Apache License, Version 2.0 (<http://www.apache.org/licenses/>). For more information, see the NOTICE.txt file included with the software.

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Research In Motion Limited
295 Phillip Street
Waterloo, ON N2L 3W8
Canada

Research In Motion UK Limited
Centrum House, 36 Station Road
Egham, Surrey TW20 9LF
United Kingdom

Published in Canada

Contents

1 BlackBerry MDS Studio.....	7
Overview	7
Architecture	8
Extensibility	9
2 BlackBerry MDS Applications	11
BlackBerry MDS Application fundamentals.....	11
BlackBerry MDS Application design approaches.....	15
BlackBerry MDS Application development cycle.....	16

BlackBerry MDS Studio

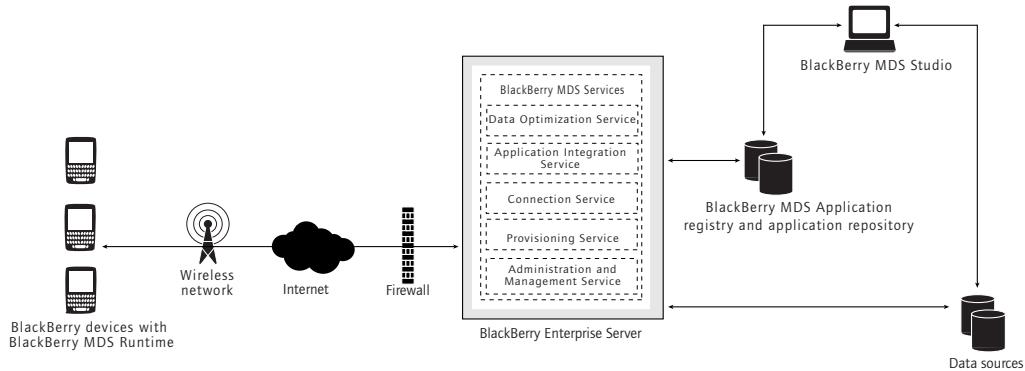
Overview
Architecture
Extensibility

Overview

The BlackBerry MDS Studio™ is a rapid application development tool that is designed to integrate enterprise web services with BlackBerry® devices. The BlackBerry MDS Studio enables you to create client applications with the simplicity and manageability of a browser-based solution. Wizards and editors are designed to enable you to assemble applications rapidly by connecting a series of visual components. A BlackBerry MDS Application is designed to provide the performance, application responsiveness, and user experience of applications created using a native programming language (for example, Java™ or Java Platform, Micro Edition) but without the need for coding.

Architecture

The BlackBerry MDS Studio is designed to mobilize enterprise applications and data sources using the BlackBerry wireless platform.



BlackBerry wireless platform

BlackBerry MDS Studio

BlackBerry MDS Studio enables you to create BlackBerry MDS Applications using a graphical methodology. In addition, the tool enables you to design the screen, data elements, and application messages visually using a drag-and-drop approach. Wizards and editors further enable you to connect graphical components. The tool also provides you with the ability to connect to enterprise applications, visualize supported operations, and generate a set of messages that you can use to interact with data sources. The BlackBerry MDS Studio is designed to generate the code needed to transmit messages to and from enterprise applications.

The BlackBerry MDS Studio enables the customization of application workflow logic using JavaScript™. When you complete the design of a BlackBerry MDS Application, the tool is designed to generate an Extensible Markup Language (XML) metadata representation of the application through an automatic code generation process.

BlackBerry MDS Runtime

The BlackBerry MDS Runtime™ is the container-based execution environment for BlackBerry MDS Applications. The BlackBerry MDS Runtime contains a control center application that enables users to search for and manage BlackBerry MDS Applications and view application information. The BlackBerry MDS Runtime is designed to provide the underlying services that are used by BlackBerry MDS Applications, including the user interface, data storage, and client-server communication services.

BlackBerry MDS Services

The BlackBerry MDS Services are designed to provide connectivity between BlackBerry MDS Applications on BlackBerry devices and enterprise applications.

Service	Description
Connection Service	The Connection Service enables a system administrator to connect the BlackBerry MDS Services to the Mobile Data Service feature of the BlackBerry Enterprise Server®. The Connection Service is designed to accept and respond to push requests from server-side push applications when the application server is behind the corporate firewall. This service is designed to provide a link to standard servers on the corporate intranet or Internet using standard Internet protocol, such as HTTP or TCP/IP, and encrypt content using the same encryption standard that is used to encrypt BlackBerry messages and other BlackBerry data.
Application Integration Service	The Application Integration Service is designed to support web services and other standard mechanisms for integrating wireless applications with existing enterprise applications and systems. This service is designed to manage the transmission of application data messages between BlackBerry MDS Applications and data sources.
Provisioning Service	The Provisioning Service is designed to control which BlackBerry MDS Applications users can download to their BlackBerry devices, support application discovery from a BlackBerry device, and manage wireless transmission and on BlackBerry devices.
Data Optimization Service	The Data Optimization Service is designed to convert existing server-side content and data to enable wireless transmission and use on BlackBerry devices.
Administrative and Management Service	The Administrative and Management Service is designed to centralize the application lifecycle management, including centralized push installation, upgrade, and removal of applications from BlackBerry devices.

Extensibility

The BlackBerry MDS Studio is designed to integrate with enterprise applications and data sources by using standard technologies such as web services, Web Services Description Language (WSDL), and Simple Object Access Protocol (SOAP).

Using web services

Web services enable incompatible and disparate software systems to interoperate. Web services enable applications to share data and invoke the capabilities of other applications regardless of how the applications were built, the operating system or platform the applications run on, or the devices that access the applications.

Web services provide organizations with a standard mechanism for enabling intra-company communication between heterogeneous internal systems. For example, by using web services a customer relationship management (CRM) system can extract customer order information from a financial or enterprise resource planning (ERP) application. Enterprise system administrators use web services as a long-term solution to help reduce the cost of integrating enterprise applications.

By using the Internet to enable software applications to work together more easily, web services can provide smoother integration within and between businesses, while creating opportunities for these businesses to connect more meaningfully with consumers. Web services enable organizations to decrease the time and cost that is associated with integrating business applications, and web services enable employees to access important data when they are away from their desks.

Each web service has a description, written in WSDL, of the services it provides, which makes the automated discovery and subsequent consumption of web services possible.

Using WSDL

WSDL is an XML format for describing network services as a set of endpoints operating on messages that contain either document-oriented or procedure-oriented information. The operations and messages are described abstractly and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints are combined into abstract endpoints or services. WSDL is extensible, enabling the description of endpoints and their messages, regardless of the message formats or network protocols to communicate.

The BlackBerry MDS Studio is designed to analyze the description of a web service to determine the following:

- **Data object types:** the entities contained in a web service and the data fields definitions
- **Operations:** the operations that describe what a web service does

Using SOAP

SOAP is a protocol for exchanging information in a decentralized, distributed environment. It is an XML-based protocol that consists of three parts:

- an envelope that defines a framework for describing what is in a message and how to process it
- a set of encoding rules for expressing instances of application-defined data types
- a convention for representing remote procedure calls and responses

In the BlackBerry MDS Studio, application messages are designed to bind to SOAP operations.

BlackBerry MDS Applications

BlackBerry MDS Application fundamentals
BlackBerry MDS Application design approaches
BlackBerry MDS Application development cycle

BlackBerry MDS Application fundamentals

Component-based design

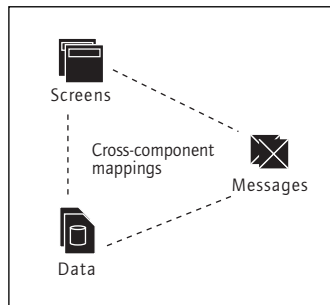
The BlackBerry MDS Studio is designed to assemble applications from a set of data, screen, and message components. The BlackBerry MDS Studio enables you to create instances of these components, customize their properties, and include them in an application design by creating relationships between components within a visual design environment.

Component type	Description
Data	<p>Data components are designed to represent application data structures. Data components are passed between mapped screen and message components and are instantiated from user actions or inbound messages from data sources. Data components store transient and persistent application data on the BlackBerry device. Transient data components are used when data is only needed temporarily by the application; persistent data components are used when data is stored between invocations of the application (including when the BlackBerry device is turned off). You define the structure of a data component from a set of available data types:</p> <ul style="list-style-type: none">• boolean• number• date• strings (including URL, password, phone number, email, and currency)• enumerated types• data component types• built-in standard types
Screen	<p>Screen components are designed to arrange the user interface (UI) elements of a BlackBerry MDS Application. You can set screen components to appear as screens or dialog boxes. You design screens using drag-and-drop tools and editors— with the underlying code designed to be generated for you. You can use menu items, action buttons, and action scripts to design screen transitions. A workflow wizard provides a simple mechanism for developing screen transitions.</p> <p>The screen components provide a set of screen controls, including labels, buttons, edit boxes, text areas, images, choice controls, drop-down lists, radio buttons, menu items, regions, and repetition controls.</p>

Component type	Description
Message	<p>Message components are designed to communicate application data across the wireless network between the client application on the BlackBerry device and the data source. Using the BlackBerry MDS Studio, bindings are created that link specific message components to specific data source operations. An instance of a message component can be either an</p> <ul style="list-style-type: none"> • inbound message: transmits data from the data source to the application on the BlackBerry device • outbound message: transmits data from the application on the BlackBerry device to the data source

Cross-component mappings and workflow

The relationships between data, screen, and message components within a BlackBerry MDS Application are referred to as cross-component mappings. The dynamic behaviors of a BlackBerry MDS Application are determined by these cross-component mappings.



Cross-component mappings

Workflow determines what happens when certain events occur in a BlackBerry MDS Application. Events are related to particular components within the application. The following events trigger an application workflow:

- operation of a screen component, such as the pressing of a button or the selection of a menu item
- arrival of an inbound message
- change in value of a data mapped screen control

Specific events are related to specific components within a BlackBerry MDS Application. The cross-component mappings associated with a component (for the particular type of event) determine the workflow behavior that is triggered when an event occurs. For example, the arrival of an inbound message can result in a change to the value of a data component and result in a screen transition. The resulting change to the data component might also cause the value displayed by a screen control to update. Alternatively, user input to a screen component can result in a change to a data component that, in turn, might result in an outbound message being sent.

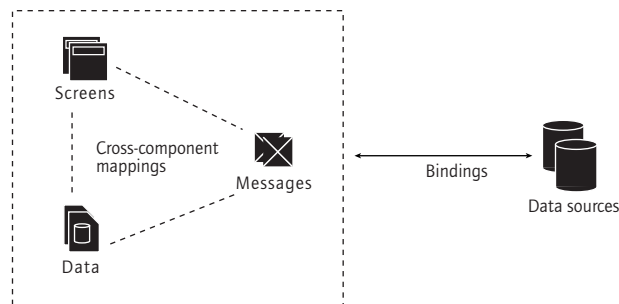
Cross-component mappings are used to define application workflow without the need for explicit coding of behaviors. For applications with more sophisticated needs, you can use JavaScript to augment the basic workflow created through cross-component mappings.

Data sources

In a distributed computing system, applications interact with other networked applications and data sources. In the BlackBerry MDS framework, a data source is classified as a collection of one or more related operations that consume or provide data that are accessible over a network. Service descriptions describe these operations. In the case of web services, the related WSDL describes the interface definition of the web service data source, details related to binding, and the network location.

Data source bindings

Bindings link specific BlackBerry MDS Application messages to specific data source operations. Development of BlackBerry MDS Applications is separated from the data source and implementation platform, providing a loosely-coupled architecture. A BlackBerry MDS Application is designed to bind to more than one application data source over the network, effectively aggregating multiple data sources and server-based applications into a sophisticated composite application for the BlackBerry device.



Data source bindings

Messaging model

BlackBerry MDS Applications are designed to store and process data locally on the BlackBerry device, without the need for a continuously available network connection. The communication model used to send and receive application messages is fully asynchronous. This means that if an outbound message is generated at a time when the BlackBerry device is outside of a wireless coverage area (for example, as a result of user input), the message is placed into an outbound queue until the wireless connection becomes available. When the message is placed into this queue, the application continues functioning, regardless of whether the message has been sent. Messages are transmitted from the queue in the background when the network connection becomes available.

Similarly, inbound messages can be queued by the BlackBerry MDS Services and transmitted to the BlackBerry device only when the wireless connection becomes available. BlackBerry MDS Applications are designed to process inbound messages as asynchronous events. This asynchronous communication model is designed for applications that are inherently tolerant to sudden or unpredictable loss of connection. This feature, combined with the ability to store and process data locally, enables BlackBerry MDS Applications to be designed in such a way that application functionality and enterprise data are available, even when wireless connections are not available.

In BlackBerry MDS Applications, the communication model consists of the following items:

- delivery mode for application messages
- handling messages when a BlackBerry device is outside of a wireless coverage area
- message queue sizes
- categories of notification messages
- size limitations for messages

Delivery modes

Messages that pass between BlackBerry MDS Applications and the BlackBerry MDS Services are categorized into two delivery modes: best-effort and standard. The default delivery mode is standard.

Delivery mode	Description	Delivery guarantee	Processing guarantee	Ordering guarantee
Best-effort	The best-effort delivery mode is a fire-and-forget mode. Messages might not be delivered, and they might be delivered in a different order than they were sent from the application to the BlackBerry MDS Services or from the BlackBerry MDS Services to the application.	No	No	No
Standard	<p>Messages sent from the application are designed to reach the BlackBerry device, even if the BlackBerry device spends time outside of a wireless coverage area or if the BlackBerry MDS Services stop. The sending side transmits the message when a connection becomes available. The application does not confirm receipt of messages, so the application might not process a message even after it reaches the BlackBerry device. Similarly, the BlackBerry MDS Services might not process a message from the application even when the message reaches the BlackBerry MDS Services server.</p> <p>Standard mode messages are designed to be delivered and processed in the order in which they were generated.</p>	Yes	No	Yes

Handling messages when a BlackBerry device is outside of a wireless coverage area

When a BlackBerry device is outside of a wireless coverage area, outbound standard and reliable application messages remain queued on the BlackBerry device until the BlackBerry device returns to a wireless coverage area, at which time the messages are transmitted. Best-effort messages are dropped when a BlackBerry device is outside of a wireless coverage area.

Message queue sizes

BlackBerry MDS Applications have restricted inbound and outbound message queue sizes. These restrictions are imposed by policies sent to the BlackBerry MDS Runtime software on the BlackBerry devices. The system administrator of the BlackBerry MDS Services can push new policies to the BlackBerry MDS Runtime, which might change the limitations of an application's inbound and outbound queue sizes.

Support for push events

The push-based architecture that is used for BlackBerry email is available for custom BlackBerry MDS Applications. Using this push-based architecture, server-side applications can push new data, alerts, and notifications to users proactively.

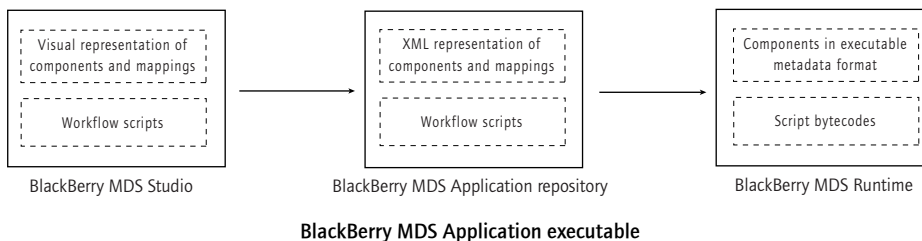
Message size limitations

The maximum size for any inbound message is 32 KB. If the result of an operation performed by a web service requires a message size greater than 32 KB to be sent from the BlackBerry MDS Services to the BlackBerry MDS Application, the BlackBerry MDS Services send an error message to the application.

BlackBerry MDS Application executables

A BlackBerry MDS Application consists of a bundle that includes an XML representation of the components and cross-component mappings that make up the application. The BlackBerry MDS Studio is designed to generate this XML representation from the visual design and optimize the bundle for delivery to the BlackBerry device, including any workflow scripts and resources (static images or animations) that the application requires.

The BlackBerry MDS Services Provisioning Service is designed to provision the application bundle to the BlackBerry device. The BlackBerry MDS Runtime is designed to convert the XML representation of the application into an efficient executable metadata format. The container-based services provided by the BlackBerry MDS Runtime are designed to instantiate the components that make up the application and satisfy the components' runtime behaviors. Any associated JavaScript is converted to the appropriate bytecodes. In addition, the BlackBerry MDS Studio is designed to generate the data source binding information required to bind application messages to the services or operations of the related data sources. The BlackBerry MDS Services is designed to load this information upon application provisioning.



BlackBerry MDS Application design approaches

The BlackBerry MDS Studio provides three application design approaches to build your BlackBerry MDS Applications: quick start, bottom-up, and top-down. The choice of design approach depends on factors such as the nature of the data source, the scope of the application, and development experience.

BlackBerry MDS Application development cycle

Regardless of the application design approach you select, all BlackBerry MDS Applications have the following development cycle:

Development steps	Description
Create application	Use the BlackBerry MDS Studio to create applications. Use the BlackBerry MDS Studio wizards and editors to create data, screen, and message components, and assemble them into a BlackBerry MDS Application.
Publish application	<p>Use the BlackBerry MDS Studio to deposit the applications into the BlackBerry MDS repository. Typically, the BlackBerry MDS Application repository resides with the BlackBerry MDS Services, but you can be install it anywhere, including on your local computer.</p> <p>Use the BlackBerry MDS Studio to publish applications to the BlackBerry MDS registry. The BlackBerry MDS Services installation is designed to configure the BlackBerry MDS Application registry. Each BlackBerry MDS Services instance is coupled with a single BlackBerry MDS Application registry instance. Use the BlackBerry MDS Studio to add, modify, or remove preferences for application registries or repositories.</p> <p>You can add an application to the repository and publish the application in a single step using the BlackBerry MDS Studio publishing wizard.</p>
Implement application	Use a BlackBerry device enabled with the BlackBerry MDS Runtime to search for and download published BlackBerry MDS Applications from the BlackBerry MDS Services.
Run application	Use a BlackBerry device to run BlackBerry MDS Applications and to begin to interact with enterprise applications.



© 2005 Research In Motion Limited
Published in Canada.