

BlackBerry Java Development Environment

Version: 4.6.0

Basishandbuch

Inhaltsverzeichnis

1 Grundlagen der BlackBerry-Geräte und der Programmierung für BlackBerry-Geräte.....	5
Entwicklungsgrundlagen für BlackBerry-Geräte.....	5
Versionszyklen und Versionen.....	6
BlackBerry Java Development Environment.....	6
Java ME und Java-APIs für BlackBerry.....	7
Unterstützung für Java-Standard-APIs.....	8
Unterstützung für Java-API-Erweiterungen.....	8
BlackBerry-Lösungen.....	9
BlackBerry Enterprise Solution.....	9
BlackBerry Internet Service.....	10
BlackBerry MDS.....	10
2 BlackBerry Java Application-Entwicklung.....	11
Standalone-Anwendungen.....	11
Anwendungen mit Desktop-Synchronisierung.....	11
Anwendungen mit drahtlosem Zugriff, drahtloser Synchronisierung oder drahtlosen Alarmen.....	11
MIDlet-Anwendungen.....	11
CLDC-Anwendungen.....	12
API-Kontrolle und Codesignaturen.....	13
Objektmodellierung.....	13
Multithreading.....	14
Bewährtes Verfahren: Einsatz von Multithreading.....	14
Bewährte Verfahren für das Schreiben einer effizienten BlackBerry Java Application.....	14
Bewährtes Verfahren: Schreiben von effizientem Code.....	14
Bewährtes Verfahren: Umsichtige Verwendung von Objekten.....	19
Bewährtes Verfahren: Verringern der Größe von kompiliertem Code.....	20
Unterstützung mehrerer Sprachen.....	21
Bewährtes Verfahren: Speichern von Textzeichenfolgen in Ressourcendateien.....	22
Multimedia-Support.....	22
Audiounterstützung.....	22
Bildunterstützung.....	22
Videounterstützung.....	22
3 Benutzeroberflächen- und der Navigationsdesign.....	23

BlackBerry-Gerät: Eingabe und Navigation für Benutzer.....	23
Trackwheel im Vergleich zum Trackball.....	23
Erstellen einer Benutzeroberfläche, die mit normalen BlackBerry-Benutzeroberflächen kompatibel ist.....	24
4 Speicherverwaltung.....	25
Verwalten des Speichers.....	25
BlackBerry-Gerätespeicher.....	25
Zu reservierende wichtige Ressourcen.....	25
Bewährtes Verfahren: Minimieren der Speicherbelegung.....	26
Verwalten von geringem verfügbarem Speicher.....	26
Erkennen von wenig verfügbarem Speicher auf einem BlackBerry-Gerät.....	26
Schonen der Ressourcen.....	27
Bewährtes Verfahren: Verwendung einer effizienten Datenstrukturauswahl.....	27
Bewährtes Verfahren: Konsolidieren von Objekten zu Objektgruppen.....	27
Garbage Collection auf einem BlackBerry-Gerät.....	28
Garbage Collection für den Arbeitsspeicher auf einem BlackBerry.....	28
Vollständige Garbage Collection auf einem BlackBerry-Gerät.....	28
Garbage Collection bei Leerlauf auf einem BlackBerry-Gerät.....	29
5 Datenverwaltung.....	30
Unterstützung für APIs zum Speichern von Daten in permanentem Speicher.....	30
APIs für die dauerhafte Speicherung.....	30
MIDP Record Management System-APIs.....	30
Dateiverbindungs-APIs.....	30
Speicher auf Wechselmedien.....	31
Zugriff auf Daten auf der microSD-Medienkarte.....	31
Sichern und Synchronisieren von Daten.....	31
6 Drahtloser Datentransport.....	33
Drahtlose Gateways.....	33
Verwenden des BlackBerry Enterprise Server als Netzwerk-Gateway.....	33
Verwenden des Internet-Gateways des Mobilfunkansbieters.....	33
Alternative Optionen für den Datentransport.....	34
Datentransport per E-Mail.....	34
Datentransport per SMS.....	34
Datentransport per PIN-Nachrichtenübermittlung.....	34

7	Integrieren von BlackBerry-Anwendungen.....	35
	Hinzufügen benutzerdefinierter Menüelemente.....	35
	Integrieren in Anwendungen aus der BlackBerry.....	35
	Zugriff auf E-Mail- und Terminplannerdaten.....	35
	Verwenden des BlackBerry Messenger mit einer BlackBerry Application.....	35
	Verwenden von Listenern für Reaktionen auf Anwendungsänderungen.....	36
8	Sicherheitsbedingungen.....	37
	Datenverschlüsselung und die BlackBerry Application.....	37
	Datenverschlüsselung beim Transport.....	37
	Datenverschlüsselung auf dem BlackBerry-Gerät.....	37
	Zugriff auf Speicher.....	37
	Authentifizierung.....	38
	Authentifizierungs- und IT-Richtlinie für BlackBerry-Geräte.....	38
	Anwendungsauthentifizierung.....	38
	Serverseitige Authentifizierung.....	38
	Kontrollierte APIs und Codesignaturen.....	38
	BlackBerry-APIs mit kontrolliertem Zugriff.....	39
	Unterstützung für IT-Richtlinien.....	40
	Anwendungssteuerung.....	40
	Dateiverschlüsselung bei microSD-Karten.....	41
	Verschlüsseln der Daten auf einer microSD-Medienkarte.....	41
	Verwenden der microSD-Medienkarte auf mehreren BlackBerry-Geräten.....	41
	IT-Richtlinien und die microSD-Medienkarte.....	41
9	Testen einer BlackBerry Java Application.....	42
	Verschleiern einer BlackBerry Java Application.....	42
	Vorabüberprüfen einer BlackBerry Java Application.....	42
	Testen von Anwendungen im BlackBerry Smartphone Simulator.....	42
	Testen von Anwendungen auf einem BlackBerry-Gerät.....	43
	BlackBerry-Simulatoren.....	43
	BlackBerry Smartphone Simulator.....	43
	BlackBerry MDS Simulator.....	44
	BlackBerry Email Simulator.....	44

10	Verfügbarmachen von Anwendungen	45
	Anwendungsverteilung über eine Computerverbindung	45
	Verteilen einer Anwendung von einem Computer aus	45
	Verteilen einer Anwendung von einer Webseite aus	45
	Verteilen einer Anwendung zu Testzwecken	45
	Anwendungsverteilung über das drahtlose Netzwerk	45
	Drahtloses Pull-Verfahren (vom Benutzer eingeleitet)	46
	Drahtloses Push-Verfahren (vom Server eingeleitet)	46
11	Glossar	47
12	Rechtliche Hinweise	50

Grundlagen der BlackBerry-Geräte und der Programmierung für BlackBerry-Geräte

1

BlackBerry®-Geräte umfassen eine drahtlose Java® ME-Umgebung, die Client-Server-Anwendungen unterstützt. Anwendungsentwickler können eine BlackBerry® Java Application erstellen, die ausgereifte Benutzeroberflächen für Dateneingabe und Datensuche bietet und dabei Multithreading, Internationalisierung, Netzwerkkommunikation und die lokale Datenspeicherung unterstützt. Die Anwendungen können mit Netzwerken über normale TCP- und HTTPS-Verbindungen kommunizieren, unabhängig vom zugrunde liegenden drahtlosen Netzwerk.

Die Anwendungsentwickler können außerdem eine BlackBerry Java Application erstellen, die eng in zentralen BlackBerry-Geräteanwendungen integriert ist (z. B. die Nachrichtliste, Terminplaneranwendungen, Telefon, Browser) und so den Benutzern eine im Wesentlichen nahtlose Arbeitsumgebung bietet.

Entwicklungsgrundlagen für BlackBerry-Geräte

Anwendungen, die für BlackBerry®-Geräte entworfen werden, sollen ein ausgewogenes Verhältnis zwischen dem bestmöglichen Funktionsumfang für die Benutzer und einer langen Akkumutzungsdauer bieten. Beim Entwickeln Ihrer BlackBerry-Geräteanwendung beachten Sie die Unterschiede zwischen mobilen Geräten und Computern. Mobile Geräte

- haben eine kleinere Bildschirmgröße, auf der nur eine begrenzte Anzahl von Zeichen dargestellt werden kann
- haben niedrigere Prozessorgeschwindigkeiten
- nutzen drahtlose Netzwerke mit einer längeren Latenzzeit als normale LANs
- enthalten weniger verfügbaren Speicher
- bieten eine kürzere Akkumutzungsdauer
- können nicht mehrere Bildschirme gleichzeitig darstellen

Benutzer von mobilen Geräten verwenden Anwendungen auf ihrem mobilen Gerät auf andere Weise als Anwendungen auf einem Computer. Auf mobilen Geräten erwarten die Benutzer, dass sie die gesuchten Informationen schnell finden. Ein CRM-System kann beispielsweise eine immense Datenmenge umfassen, aber die Benutzer brauchen nur einen Bruchteil dieser Informationen auf einmal. Die Benutzeroberfläche des BlackBerry-Geräts ist so gestaltet, dass die Benutzer ihre Aufgaben mühelos erledigen und rasch auf Daten zugreifen können.

Wenn Sie Anwendungen für BlackBerry-Geräte entwickeln, versuchen Sie, diese Anwendung so konsistent wie möglich mit anderen BlackBerry-Geräteanwendungen zu halten. Befolgen Sie die folgenden Richtlinien:

- Verwenden oder erweitern Sie nach Möglichkeit vorhandene Benutzeroberflächen-Komponenten, damit Ihre Anwendung das Standardverhalten der Komponente erben kann.
- Halten Sie sich so nah wie möglich an das Standard-Navigationsmodell, sodass die Benutzer die Tastatur und den Trackball uneingeschränkt nutzen können.

- Stellen Sie alle Aktionen in Menüs zur Verfügung. Vergewissern Sie sich, dass die im Menü verfügbaren Aktionen tatsächlich für den aktuellen Kontext der Benutzer von Nutzen sind.

Beim Entwickeln Ihrer Anwendung sollten Sie auch die folgenden Richtlinien beachten:

- Konzentrieren Sie sich stets auf die unmittelbare Aufgabe der Benutzer. Vereinfachen Sie die Datenauswahl und die Datendarstellung so weit, dass nur die tatsächlich nötigen Daten angezeigt werden, die die Benutzer in genau diesem Moment brauchen.
- Legen Sie die Darstellung der Daten so an, dass der kleine Bildschirm optimal ausgenutzt wird.

Bevor Sie mit der Entwicklung Ihrer Anwendung beginnen, sollten Sie ggf. die Hauptanwendungen auf dem BlackBerry-Gerät nutzen oder auch den BlackBerry Smartphone Simulator heranziehen, um mehr über das Navigationsmodell und über bewährte Verfahren zur Gestaltung der Benutzeroberfläche für Ihre Anwendung zu erfahren.

Versionszyklen und Versionen

Alle BlackBerry®-Geräte enthalten eine bestimmte Version der BlackBerry® Device Software und der BlackBerry® Java® Virtual Machine. Um die Version der BlackBerry Device Software für ein BlackBerry-Gerät zu bestimmen, klicken Sie in den Geräteoptionen auf **Info**. Sie können die BlackBerry Device Software aktualisieren. Beispielsweise ist es möglich, ein BlackBerry-Gerät mit BlackBerry Device Software Version 4.0 auf BlackBerry Device Software Version 4.1 zu aktualisieren.

Jede Hauptversion der BlackBerry Device Software und der BlackBerry JVM wird von Research In Motion zusammen mit den zugehörigen Java-APIs und der passenden Version der BlackBerry® Java® Development Environment ausgeliefert. Die Version der BlackBerry Device Software bestimmt die Version der BlackBerry® Integrated Development Environment, mit der Sie Anwendungen entwickeln können. RIM hat beispielsweise BlackBerry Device Software Version 4.0 und BlackBerry JDE Version 4.0 gleichzeitig veröffentlicht. BlackBerry JDE Version 4.0 bietet Unterstützung für die APIs, die mit BlackBerry Device Software Version 4.0 und BlackBerry JVM Version 4.0 eingeführt wurden. Anwendungen, die Sie mit BlackBerry JDE Version 4.0 erstellen, funktionieren nur auf BlackBerry-Geräten, auf denen BlackBerry Device Software Version 4.0 oder höher ausgeführt wird.

Entscheiden Sie anhand der folgenden Kriterien, welche Version der BlackBerry JDE Sie beim Entwickeln einer Anwendung verwenden:

- Wenn die Anwendung keine besonderen Hardware-Funktionen der BlackBerry-Geräte und auch keine erst vor kurzem veröffentlichte API-Erweiterungen verwenden muss, entwickeln Sie die Anwendung mit der BlackBerry JDE Version 4.0.
- Soll die Anwendung ausschließlich auf dem BlackBerry® Pearl™ 8100-Smartphone ausgeführt werden, verwenden Sie BlackBerry JDE Version 4.2.

BlackBerry Java Development Environment

BlackBerry® Java® Development Environment ist eine vollständig integrierte Entwicklungs- und Simulationsumgebung zum Erstellen einer BlackBerry® Java Application für BlackBerry-Geräte. Mit BlackBerry JDE können Entwickler mithilfe der Programmiersprache Java® ME und der erweiterten Java-APIs Anwendungen für BlackBerry erstellen.

BlackBerry Java Development Environment enthält die folgenden Entwicklungstools:

- BlackBerry® Integrated Development Environment
- BlackBerry Smartphone Simulator
- Java ME-APIs und BlackBerry-APIs
- Beispielanwendungen

Die BlackBerry IDE umfasst eine ganze Reihe von Bearbeitungs- und fehlerbehebungstools, die für die Entwicklung einer BlackBerry Java Application optimiert sind. Der BlackBerry Smartphone Simulator bietet eine vollständige Windows®-Umgebung und dient zur Simulation der Benutzeroberflächen und des Benutzereingriffs, der Netzwerkverbindungen, der E-Mail-Dienste und der drahtlosen Datensynchronisierung.

Das BlackBerry Java Development Environment Component Package enthält die folgenden Entwicklungstools für die Entwicklung in Drittanbieter-IDEs wie NetBeans™ oder Eclipse™:

- RAPC: Mit diesem Eingabeaufforderungs-Compiler können Sie JAVA- und JAR-Dateien in COD-Dateien kompilieren, die Sie dann im BlackBerry Smartphone Simulator oder auf einem BlackBerry-Gerät ausführen können.
- JavaLoader: Mit diesem Tool können Sie eine Anwendung auf einem BlackBerry-Gerät zu Testzwecken hinzufügen oder zu aktualisieren sowie Informationen zu den COD-Dateien der Anwendung abrufen.
- BlackBerry® Signature Tool: Mit diesem Tool können Sie Codesignaturen-Anforderungen an das BlackBerry® Signing Authority Tool senden.
- Tool zur Vorabüberprüfung: Mit diesem Tool können Sie Ihre Klassen teilweise überprüfen, bevor Sie Ihre Anwendung auf ein BlackBerry-Gerät laden.
- JDWP: Mit diesem Tool können Sie die Fehlerbehebung von Anwendungen mithilfe der integrierten Entwicklungsumgebung von Drittanbietern ausführen.

Java ME und Java-APIs für BlackBerry

Java® ME ist eine Branchen-Standardplattform, mit der häufig genutzte Gruppen von Java-APIs für verschiedene Arten von drahtlosen und eingebetteten Geräten definiert werden. Eine Java ME-Anwendung auf einem BlackBerry®-Gerät wird in der BlackBerry® Java® Virtual Machine ausgeführt, die sämtliche Runtime-Dienste für die Anwendungen bereitstellt und Funktionen wie wie typische Speicherzuordnungen, Sicherheitsprüfungen und Garbage Collection übernimmt.

Der Java ME-MIDP-Standard ist die Antwort auf den Bedarf der APIs und der BlackBerry JVM nach einem eingeschränkten drahtlosen Gerät mit Benutzeroberfläche. Das BlackBerry-Gerät unterstützt den Java ME-MIDP-Standard gemäß der Definition in JSR 118. Der Java ME-MIDP-Standard liefert eine Kerngruppe von Java-APIs, die von allen BlackBerry-Geräten unterstützt werden, unabhängig von dem jeweils zugrunde liegenden Betriebssystem. Entwickler können häufig eine Java-Anwendung mit den MIDP-Standard-APIs erstellen und diese Anwendung dann auf vielen verschiedenen Arten von Geräten ausführen.

Unterstützung für Java-Standard-APIs

Das BlackBerry®-Gerät und BlackBerry® Java® Development Environment unterstützen den Java® ME-MIDP-Standard mit einer Kerngruppe von Java-APIs, mit denen Sie Anwendungen für drahtlose Geräte entwickeln können. Das BlackBerry-Gerät und BlackBerry® Java® Development Environment unterstützen außerdem die folgenden JSRs:

- JSR 30: Connected Limited Device Configuration, Version 1.0 (Bibliothek für Gerätekonfigurationen)
(unterstützt auf Geräten mit BlackBerry® Device Software Version 4.0 oder niedriger)
- JSR 37: Mobile Information Device Profile, Version 1.0 (Softwareprofil für mobile Geräte)
(unterstützt auf Geräten mit BlackBerry Device Software Version 4.0 oder niedriger)
- JSR 75: Portable Optional Packages for the J2ME Platform (PDAP; optionale PDA-Pakete); Unterstützung nur für die PIM-APIs und die Dateiverbindungs-API für Java ME (unterstützt auf Geräten mit BlackBerry Device Software Version 4.2 oder niedriger)
- JSR 82: Java-APIs für Bluetooth®
- JSR 118: Mobile Information Device Profile, Version 2.0 (Softwareprofil für mobile Geräte)
- JSR 120: Wireless Messaging API (WMA), Version 1.1 (API für drahtlose Nachrichtenübermittlung)
- JSR 135: Mobile Media APIs (MM-API), Version 1.1 (APIs für mobile Medien)
- JSR 139: Connected Limited Device Configuration, Version 1.1 (Bibliothek für Gerätekonfigurationen)
- JSR 172: J2ME-Webdienste
- JSR 177: Security- und Trust Services-API für J2ME (SATSA)
- JSR 179: Location API for Java ME (Java Standort-API)
- JSR 185: Java Technology for the Wireless Industry (JTWI; Java-Technologie für die drahtlose Industrie)
- JSR 205: Wireless Messaging API 2.0 (API für drahtlose Nachrichtenübermittlung)
- JSR 211: Content Handler API (API für die Verarbeitung von Inhalten)
- JSR 226: Scalable 2D Vector Graphics API für Java ME (API für skalierbare 2D-Vektorgrafiken)
- JSR 238: Mobile Internationalization API (API für die mobile Internationalisierung)

Unterstützung für Java-API-Erweiterungen

BlackBerry®-Geräte unterstützen die nachstehenden Java®-APIs, die nicht zu den JSR-Standarddefinitionen gehören und mehr Funktionen als die MIDP-API-Standardbibliotheken bieten.

API	Beschreibung
Benutzeroberflächen-APIs	Mit diese APIs können Sie Bildschirme, Menüelemente und alle Komponenten der Benutzeroberfläche erstellen.

API	Beschreibung
APIs für die dauerhafte Speicherung von Daten	Mit diesen APIs können Sie benutzerdefinierte Daten lokal innerhalb Ihrer Anwendung speichern.
Netzwerk- und E/A-APIs	Mit diesen APIs können Sie Netzwerkverbindungen einrichten sowie Daten von einer serverseitigen Anwendung lesen bzw. auf diese schreiben.
Ereignis-Listener	Mit den Ereignis-Listenern können Sie auf Ereignisse reagieren, die von den BlackBerry-Gerätebenutzern oder vom System auf einem BlackBerry-Gerät ausgelöst wurden.
Anwendungsintegrations-APIs	Mit diesen APIs können Sie die Integration in die vorhandenen BlackBerry-Anwendungen (E-Mail, Telefon, Kalender, Kontakt, Browser, Kamera, Media Player, Aufgabenliste) realisieren.
Zusätzliche Dienstprogramme	Mit diesen zusätzlichen APIs können Sie die Datenverschlüsselung und -komprimierung, die XML-Analyse, die Bluetooth®-Konnektivität, standortabhängige Dienste und vieles mehr umsetzen.

BlackBerry-Lösungen

BlackBerry®-Gerätebenutzer verwenden den BlackBerry® Enterprise Server oder den BlackBerry® Internet Service – oder auch beide Dienste auf einem Gerät. Die Grundlagen der Unterschiede zwischen dem BlackBerry Enterprise Server und dem BlackBerry Internet Service einerseits und die Besonderheiten der zu unterstützenden Benutzer andererseits sind wichtig, weil sich dies darauf auswirken kann, welche Übertragungsverfahren zum Einsatz kommen und wie die Datensynchronisierung erfolgen soll.

BlackBerry Enterprise Solution

Der BlackBerry® Enterprise Server gehört zur BlackBerry® Enterprise Solution. Der BlackBerry Enterprise Server befindet sich hinter der Unternehmensfirewall und fungiert als drahtloses Gateway für die BlackBerry-Gerätebenutzer in einem Unternehmen, über das diese auf geschäftliche E-Mail- und Terminplanerdaten zugreifen können. Der BlackBerry Enterprise Server bietet außerdem die folgenden wichtigen Funktionen:

- Datenverschlüsselung und -komprimierung
- Verwaltungs- und Überwachungsdienstprogramme für BlackBerry-Geräte
- Vereinfachte Bereitstellung von Anwendungen
- Authentifiziertes Gateway für den Intranetzugriff aus einer BlackBerry® Java Application heraus

BlackBerry Internet Service

BlackBerry®-Gerätebenutzer, die keinem BlackBerry® Enterprise Server zugeordnet sind, können den BlackBerry® Internet Service verwenden. Der BlackBerry Internet Service bietet einen E-Mail- und Internet-Service für BlackBerry-Geräte, mit dem die Benutzer E-Mail-Nachrichten automatisch senden und unterwegs bequem auf E-Mail-Nachrichtenanlagen und Internetinhalte zugreifen können.

Der BlackBerry Internet Service unterstützt direkte HTTP- und TCP/IP-Konnektivität zum Internet über eine BlackBerry® Java Application von Drittanbietern.

BlackBerry MDS

Für den Zugriff einer BlackBerry® Java Application auf Ressourcen hinter der Unternehmensfirewall bietet der BlackBerry® Enterprise Server das BlackBerry® Mobile Data System. Das BlackBerry MDS umfasst HTTP- und TCP/IP-Proxys für eine BlackBerry Java Application, sodass das BlackBerry-Gerät ohne zusätzliche VPN-Software mit Anwendungs- und Webservern hinter der Unternehmensfirewall kommunizieren kann. Anwendungen, die Daten über den BlackBerry Enterprise Server als Gateway senden, profitieren von der vereinfachten Unternehmenskonnektivität, der Datenverschlüsselung und -komprimierung sowie von der Unabhängigkeit von drahtlosen Netzwerken in der BlackBerry® Enterprise Solution. BlackBerry MDS enthält außerdem eine offene Schnittstelle, mit der serverseitige Anwendungen hinter der Unternehmensfirewall in der Lage sind, Inhalte im Push-Verfahren an Anwendungen auf BlackBerry-Geräten zu übermitteln.

BlackBerry Java Application-Entwicklung

2

Standalone-Anwendungen

Mit den BlackBerry®-APIs können Sie eigenständige Anwendungen erstellen, z. B. Spiele oder statische Referenzhandbücher, die sich als Offline-Anwendungen ausführen lassen. Vor dem Kompilieren können Sie die erforderlichen Ressourcendaten zu einer Anwendung hinzufügen. Die Benutzer des BlackBerry-Geräts können die Anwendung über das drahtlose Netzwerk oder mithilfe der BlackBerry® Desktop Software installieren. Sobald eine Anwendung auf dem BlackBerry-Gerät installiert ist, muss sie keine Verbindung mit dem drahtlosen Netzwerk oder dem Computer herstellen.

Anwendungen mit Desktop-Synchronisierung

Mit den BlackBerry®-APIs können Sie Anwendungen mit Funktionen zur Desktop-Synchronisierung erstellen, z. B. Referenzhandbücher und Terminplaneranwendungen. Der Benutzer verbindet das BlackBerry-Gerät mit einem Computer, um so die auf dem Computer befindlichen Daten zu verwalten und zu synchronisieren.

Research In Motion® bietet keine HotSync®-Treiber oder andere Module für die direkte Datenbanksynchronisierung. Sie müssen den Synchronisierungscode erstellen und der BlackBerry-Gerätebenutzer muss den Datensynchronisierungsvorgang manuell starten. Nach der Installation der Anwendung auf dem BlackBerry-Gerät muss der BlackBerry-Gerätebenutzer die Daten manuell synchronisieren. Hierzu wird das BlackBerry-Gerät über eine serielle Schnittstelle, eine USB-Verbindung oder eine Bluetooth®-Verbindung mit dem Computer verbunden.

Anwendungen mit drahtlosem Zugriff, drahtloser Synchronisierung oder drahtlosen Alarmen

Mit den BlackBerry®-APIs können Sie Anwendungen erstellen, die Inhalte proaktiv per Push über das drahtlose Netzwerk an BlackBerry-Geräte in Arbeitsumgebungen übertragen, in denen der BlackBerry® Enterprise Server verwendet wird. Eine BlackBerry® Java Application für BlackBerry-Geräte ermöglicht den BlackBerry-Gerätebenutzern über eine drahtlose Verbindung zum Internet oder zum Unternehmensintranet den Zugriff auf Remote-Daten und -Anwendungen. Die BlackBerry® Java® Development Environment enthält APIs, mit denen Sie in Anwendungen Netzwerkverbindungen zu Servern im Internet oder im Unternehmensintranet einrichten können.

MIDlet-Anwendungen

Das MIDlet-Anwendungsmodell ist Teil der MIDP-Spezifikation. Die Hauptklasse eines MIDlet erweitert immer die MIDlet-Klasse und muss Methoden verwenden für `startApp()`, `pauseApp()` und `destroyApp()`.

Vorteile	Nachteile
<ul style="list-style-type: none"> Die Anwendungen lassen sich auf andere Geräte übertragen, die ebenfalls die MIDP-Spezifikation unterstützen. 	<ul style="list-style-type: none"> Die Anwendungen können nur die Benutzeroberflächen-APIs verwenden, die in der Bibliothek <code>javax.microedition.lcdui</code> vorhanden sind. Das Modell nimmt an, dass alle Anwendungsprozesse beendet werden, sobald die Anwendung geschlossen werden. Anwendungen können beim Einschalten des Geräts nicht automatisch im Hintergrund gestartet werden.

CLDC-Anwendungen

Das CLDC-Anwendungsmodell ist eine Spezifikation eines Rahmens für Java® ME. Eine CLDC-Anwendung erweitert die Klasse `UiApplication` und startet mit einer Standardmethode `main()`.

Die meisten Beispielanwendungen in der BlackBerry® Java® Development Environment nutzen das CLDC-Anwendungsmodell. Alle BlackBerry-Hauptanwendungen (Nachrichtenliste, Kontaktliste, Kalender, Browser) sind als CLDC-Anwendungen angelegt.

Vorteile	Nachteile
<ul style="list-style-type: none"> Die BlackBerry-Benutzeroberflächen-APIs bieten mehr Funktionalität und Flexibilität als die Standardbibliothek <code>javax.microedition.lcc</code> Anwendungen können aktive Hintergrund-Threads ausführen, nachdem sie geschlossen wurden. Anwendungen können beim Einschalten des Geräts automatisch im Hintergrund gestartet werden. 	<ul style="list-style-type: none"> Die Anwendungen lassen sich nicht auf andere Geräte übertragen.

Vorteile	Nachteile
<ul style="list-style-type: none">Anwendungen können Informationen mit anderen Anwendungen über IPC-APIs auszutauschen.Entwickler können wiederverwendbare Bibliotheksmodule erstellen, die von CLDC-Anwendungen importiert werden können.	

API-Kontrolle und Codesignaturen

Wenn Sie eine BlackBerry® Java Application für BlackBerry-Geräte entwickeln, können Sie nur die öffentlichen Java-APIs nutzen, die in den Javadoc™-Dokumenten in der BlackBerry® Java® Development Environment veröffentlicht und dokumentiert sind. Die BlackBerry® Java® Virtual Machine auf dem BlackBerry-Gerät ist so ausgelegt, dass die zugrunde liegenden Daten und das Betriebssystem geschützt werden. Aus diesem Grund können Anwendungen keine undokumentierten oder nicht unterstützten APIs aufrufen oder auf Daten zugreifen, die nicht explizit durch die APIs zur Verfügung gestellt werden. Wenn Sie versuchen, Java-APIs einzusetzen, die nicht öffentlich zugänglich sind, erzeugt Ihre Anwendung eine Fehlermeldung während der Laufzeit.

Öffentliche APIs sind entweder offen oder signiert. Signierte APIs legen die Methoden offen, mit denen die BlackBerry-Gerätebenutzer auf Daten oder andere Informationen auf dem BlackBerry-Gerät zugreifen, die als vertraulich gelten. Sie können signierte APIs verwenden, wobei Sie einen Satz von Codesignierungsschlüsseln von Research In Motion anfordern und empfangen müssen. Anschließend müssen Sie die Anwendung signieren, bevor Sie sie auf einem BlackBerry-Gerät installieren. Die Codesignierung bildet keine Zertifizierung oder Genehmigung einer Anwendung; RIM wird damit lediglich in die Lage versetzt, den Autor einer Anwendung zu identifizieren, die vertrauliche APIs nutzt, falls die Anwendung bösartig ist.

Einen Satz von Codesignierungsschlüsseln erhalten Sie unter www.blackberry.com/developers/downloads/jde/api.shtml. Ihr Satz von Codesignierungsschlüsseln geht Ihnen innerhalb von etwa 10 Tagen zu.

Objektmodellierung

Beim MIDlet- oder CLDC-Anwendungsmodell müssen Sie jeweils einen objektorientierten Ansatz verfolgen, wenn Sie Ihre Anwendung für das BlackBerry®-Gerät entwickeln. Bei einem objektorientierten Ansatz platzieren die Entwickler den Code für einen bestimmten Prozess oder eine Funktion in Objekte. Ein Entwickler kann beispielsweise separate Objekte nutzen, um die Netzwerkaktivität, die Datenspeicherung, die Be- und Verarbeitung der Daten sowie die Interaktion mit der Benutzeroberfläche zu steuern. Wenn Sie Ihre Anwendung entwickeln, sollten Sie mit einem guten Objektmodell beginnen.

Multithreading

Das BlackBerry®-Betriebssystem ist ein Multithreading-fähiges Betriebssystem. Dies bedeutet, dass viele Anwendungen und Prozesse gleichzeitig auf dem BlackBerry-Gerät aktiv ausgeführt werden können. Anwendungen können beispielsweise Hintergrund-Threads für prozessorintensive Aufgaben oder für die Netzwerkkommunikation verwenden, damit sie den Haupt-Thread nicht beeinträchtigen. Wenn eine Anwendung Hintergrund-Threads erstellt und ein BlackBerry-Gerätebenutzer die Anwendung beendet, können die Hintergrund-Threads aktiv bleiben.

Bewährtes Verfahren: Einsatz von Multithreading

Nutzen Sie die Multithreadingfunktionen des BlackBerry®-Betriebssystems effektiv. Erstellen Sie insbesondere immer einen neuen Thread für Netzwerkverbindungen oder andere langwierige Vorgänge (mehr als eine Zehntelsekunde Dauer). Verwenden Sie Hintergrund-Threads für Listener und andere Prozesse, die im Hintergrund ausgeführt werden, wenn die Anwendung gestartet wird.

Bewährte Verfahren für das Schreiben einer effizienten BlackBerry Java Application

Bewährtes Verfahren: Schreiben von effizientem Code

Damit eine BlackBerry® Java® Application die Ressourcen effizient nutzen kann, beachten Sie die folgenden Anweisungen:

- Verwenden Sie lokale Variablen.
- Verwenden Sie Kurzschrift für die Auswertung von Bedingungen vom Typ "Boolean".
- Legen Sie Klassen als "final" fest.
- Verwenden Sie `int` statt `long`.
- Verzichten Sie auf Garbage Collection-Vorgänge
- Verwenden Sie statische Variablen für Objekte vom Typ `String`.
- Vermeiden Sie den Konstruktor `String(String)`.
- Schreiben Sie effiziente Schleifen.
- Optimieren Sie Unterausdrücke.
- Optimieren Sie Divisionsvorgänge.
- Verzichten Sie auf `java.util.Enumeration`.
- Führen Sie Casts mithilfe von `instanceof` durch.
- Bewerten Sie Ausdrücke mithilfe von `instanceof`.
- Vermeiden Sie die Verwendung von `StringBuffer.append(StringBuffer)`.

- Vermeiden Sie die Rückgabe von `null`.
- Vermeiden Sie die Eingabe von `null` in Methoden.
- Gehen Sie vorsichtig vor, wenn Sie `null` in einen Konstruktor eingeben.
- Verwenden Sie `long` für eindeutige Bezeichner.
- Beenden Sie Anwendungen ordnungsgemäß.
- Drucken Sie die Stack-Verfolgung.

Verwenden lokaler Variablen

Verwenden Sie so weit wie möglich lokale Variablen. Der Zugriff auf lokale Variablen ist effizienter als der Zugriff auf Klassenmitglieder.

Verwenden von Kurzschrift für die Auswertung Boolescher Bedingungen

Zur Auswertung einer `Boolean` Bedingung verwenden Sie Kurzschrift. Der daraus resultierende kompilierte Code ist kürzer.

Codebeispiel

```
return( boolean_expression );
```

Festlegen von Klassen als "final"

Wenn Sie Code-Bibliotheken erstellen, kennzeichnen Sie Klassen als `final`, wenn Sie wissen, dass die Entwickler sie nie erweitern werden. Mit dem Schlüsselwort `final` kann der Compiler einen noch effizienteren Code generieren.

Standardmäßig kennzeichnet der BlackBerry® Java® Development Environment Compiler sämtliche Klassen, die Sie in einer COD-Datei einer Anwendung nicht erweitern, als `final`.

Verwendung von "int" statt "lang"

In Java® bezeichnet `lang` eine ganze Zahl im 64-Bit-Format. Bei BlackBerry®-Geräten kommt ein 32-Bit-Prozessor zum Einsatz, weshalb Vorgänge zwei- bis viermal schneller ablaufen, wenn Sie `int` statt `lang` verwenden.

Verzichten auf Garbage Collection

Verzichten Sie auf den Aufruf von `System.gc()` zur Ausführung eines Garbage Collection-Vorgangs, weil dieser auf BlackBerry®-Geräten mit begrenztem verfügbarem Speicher unter Umständen zuviel Zeit in Anspruch nimmt. Lassen Sie die Garbage Collection stattdessen über die BlackBerry® Java® Virtual Machine vornehmen.

Verwenden statischer Variablen für Zeichenfolgen

Wenn Sie statische Felder (auch als Klassenfelder bezeichnet) vom Typ `Zeichenfolge` definieren, können Sie die Anwendungsgeschwindigkeit erhöhen, wenn Sie statische Variablen (nicht `final`) statt Konstanten verwenden (`final`). Für primitive Datentypen wie `int` gilt das Gegenteil.

Sie erstellen beispielsweise ein Zeichenfolgeobjekt wie folgt:

```
private static final String x = "Beispiel;"
```

Für diese statische Konstante (gekennzeichnet mit dem Schlüsselwort `final`) wird jedesmal, wenn Sie die Konstante verwenden, eine temporäre Zeichenfolge-Instanz erstellt. Der Compiler beseitigt `"x"` und ersetzt dies durch ein Zeichenfolge-Beispiel `" "` im Bytecode, sodass die BlackBerry® Java® Virtual Machine jedesmal eine Hashtabellensuche durchführt, wenn Sie auf `"x"` verweisen.

Bei einer statischen Variable (ohne Schlüsselwort `final`) wird die Zeichenfolge dagegen nur einmal erstellt. Die BlackBerry JVM führt die Hash-Tabellensuche nur dann durch, wenn `"x"` initialisiert wird, wodurch der Zugriff beschleunigt wird.

```
private static String x = "example";
```

Sie können öffentliche Konstanten verwenden (also Felder `final`), aber die Variablen müssen als privat gekennzeichnet werden.

Vermeiden des Konstruktors `String(String)`

In einer BlackBerry® Java Application ist jede Zeichenfolge in Anführungszeichen eine Instanz der Klasse `java.lang.String`. Erstellen Sie ein Objekt vom Typ `Zeichenfolge` ohne den Konstruktor `java.lang.String(String)`.

Codebeispiel

```
String str = "abc"; String str = "gefundene " + n + " Elemente";
```

Schreiben effizienterer Schleifen

Wenn Ihr Container vermutlich mehrere Elemente enthalten soll, weisen Sie die entsprechende Größe zu einer lokalen Variablen zu.

Falls die Reihenfolge bei der Iteration durch die Elemente nicht wichtig ist, können Sie auch eine Rückwärts-Iteration vornehmen, um so die zusätzliche lokale Variable im Stapel zu vermeiden und den Vergleich zu beschleunigen.

Codebeispiel

```
int size = vector.size(); for( int i = 0; i < size; ++i ) {    ... } for( int  
i = vector.size() - 1; i >= 0; --i ) {    ... }
```

Optimieren von Unterausdrücken

Wenn Sie den gleichen Ausdruck zweimal verwenden, verwenden Sie eine lokale Variable.

Codebeispiel

```
int tmp = i+1; one( tmp ); two( tmp );
```

Optimieren von Divisionsvorgängen

Divisionsvorgänge können auf BlackBerry®-Geräten unter Umständen nur langsam ablaufen, weil der Prozessor keine Divisionsanweisung in der Hardware umfasst.

Wenn in Ihrem Code eine positive Zahl durch 2 geteilt wird, verwenden Sie die Anweisung zum Verschieben um eine Position nach rechts (>> 1). Verwenden Sie die Verschiebung nach rechts (>>) nur dann, wenn Ihnen bekannt ist, dass ein positiver Wert vorliegt.

Codebeispiel

```
int = width >> 1;
```

Verzichten auf java.util.Enumeration

Verzichten Sie auf die Verwendung von Objekten vom Typ `java.util.Enumeration` außer wenn Sie Daten ausblenden möchten (anders gesagt, wenn eine Aufzählung der Daten anstelle der eigentlichen Daten zurückgegeben werden soll). Das Abfragen eines Vektors oder einer Hash-Tabelle für ein Objekt vom Typ `Aufzählung` ist langsam und erzeugt unnötigen Datenmüll. Wenn ein anderer Thread den Vektor ändern könnte, synchronisieren Sie die Iteration. Die Java® SE verwendet ein Objekt vom Typ `Iterator` für ähnliche Vorgänge, aber Objekte vom Typ `Iterator` sind bei Java® ME nicht verfügbar.

Codebeispiel

```
for( int i = v.size() - 1; i >=0; --i ) { o = v.elementAt( i ); ... }  
synchronized( v ) { for( int i = v.size() - 1; i >=0; --i ) { o = v.elementAt  
( i ); ... } }
```

Durchführen von Casts mit instanceof

Verwenden Sie `instanceof`, um zu überprüfen, ob ein Cast funktioniert.

Codebeispiel

```
if( x instanceof String ) { (String)x.whatever(); } else { ... } x
```

Auswertung von Ausdrücken mit "instanceof"

Um den Code zu verkleinern und zu beschleunigen, wenn Sie eine Bedingung mithilfe von `instanceof` auswerten, sollten Sie nicht explizit auswerten, ob die Variable den Wert "Null" aufweist.

Codebeispiel

```
if( e instanceof ExampleClass ) { ... } if( ! ( e instanceof ExampleClass ) )  
{ ... }
```

Vermeiden von `StringBuffer.append` (`StringBuffer`)

Zum Anfügen eines Zeichenfolge-Puffers an einen anderen Puffer sollten Sie die folgende Syntax in der BlackBerry® Java Application verwenden: `net.rim.device.api.util.StringUtilities.append`
(`StringBuffer dst`, `StringBuffer src` [, `int offset`, `int length`]).

Codebeispiel

```
public synchronized StringBuffer append(Object obj) { if (obj instanceof  
StringBuffer) { StringBuffer sb = (StringBuffer)obj;  
net.rim.device.api.util.StringUtilities.append( this, sb, 0, sb ) return  
this; } return append(String.valueOf(obj)); }
```

Vermeiden der Rückgabe von "Null"

Wenn Sie eine öffentliche Methode schreiben, mit der ein Objekt zurückgegeben wird, sollte die Methode nur unter den folgenden Bedingungen den Wert "Null" zurückgeben:

- Ihre Anwendung erwartet, dass ein Null-Wert während des normalen Einsatzes auftritt.
- Im Javadoc™-Parameter `@return` für die Methode ist festgelegt, dass "Null" ein zulässiger Rückgabewert ist.

Wenn Ihre Anwendung keinen "Null"-Rückgabewert erwartet, sollte die Methode eine geeignete Ausnahme auslösen, mit der die explizite Behandlung dieses Problems durch den Aufrufenden erzwungen wird. Der Aufrufende muss ggf. nur dann eine Prüfung auf einen "Null"-Rückgabewert durchführen, wenn der Aufrufende eine "Null"-Ausnahme auslöst.

Vermeiden der Eingabe von "Null" in Methoden

Geben Sie keine Null-Parameter in eine API-Methode, es sei denn, die API-Referenz besagt eindeutig, dass die betreffende Methode dies unterstützt.

Vorsichtsmaßnahmen beim Eingeben von "Null" in einen Konstruktor

Um Unklarheiten beim Eingeben von "Null" in einen Konstruktor zu vermeiden, weisen Sie "Null" per Cast zum geeigneten Objekt zu.

Wenn eine Klasse mehrere Konstruktoren umfasst, ist bei der Übergabe eines Null-Parameters ggf. nicht eindeutig ersichtlich, welcher Konstruktor verwendet werden soll. Der Compiler meldet daher ein Fehler.

Durch Zuweisen von "Null" per Cast zum geeigneten Objekt legen Sie genau fest, welchen Konstruktor der Compiler verwenden soll. Diese Vorgehensweise ermöglicht auch die Aufwärtskompatibilität, wenn spätere Versionen der API neue Konstruktoren enthalten.

Codebeispiel

```
new someObject ((someObject)null );
```

Verwenden von "long" für eindeutige Bezeichner

Verwenden Sie den Bezeichner `long` anstatt eines `Zeichenfolge`-Bezeichners für eindeutige Konstanten, z. B. GUIDs, Hash-Tabellenschlüssel und Status- oder Kontextbezeichner.

Für Bezeichner, die in einer BlackBerry® Java Application eindeutig bleiben müssen, verwenden Sie Schlüssel, die eine Anwendung auf der Grundlage eines Hash für eine `Zeichenfolge` erzeugt. Nehmen Sie in die Eingabe-Zeichenfolge so viele Informationen auf, dass der Bezeichner eindeutig wird. Verwenden Sie zum Beispiel einen vollqualifizierten Paketnamen wie `com.rim.samples.docs.helloworld`.

Ordnungsgemäßes Beenden von Anwendungen

Bevor Sie `System.exit(int status)` aufrufen, führen Sie zunächst ggf. notwendige "Aufräumarbeiten" durch. Entfernen Sie beispielsweise Objekte aus dem Laufzeitspeicher, die von den Anwendungen nicht mehr benötigt werden.

Drucken der Stack-Verfolgung

Wenn Sie Ihre Anwendung debuggen und dabei die Stack-Verfolgung aufrufen möchten, erfassen Sie eine `Throwable`-Instanz.

Codebeispiel

```
catch (Throwable t) { t.printStackTrace(); }
```

Bewährtes Verfahren: Umsichtige Verwendung von Objekten

Damit eine BlackBerry® Java Application die Speicherressourcen effizient nutzen kann, beachten Sie die folgenden Fragen:

- Sind alle Objekte von der Größe der Anwendung her notwendig?
- Kann Ihre Anwendung Objekte, die Primitive darstellen, z. B. `Lang`, `Integer` oder `Boolesche Operatoren` tatsächlich als Primitive speichern statt als Objekte?
- Sind alle permanenten Objekte notwendig?
- Sind Instanzen von `Vektoren` und `Hashtables` vorhanden? Sind diese Instanzen notwendig? Falls ja: Wie viele Objekt-Handles werden nicht in `Vektoren` oder `Hashtables` verwendet, da die anfängliche Größe größer war als die notwendige Größe?
- Wie viele Objekte werden in Ihrer Anwendung erstellt und dann wieder verworfen? Anders gesagt: Wie viele bereichsorientierte Objekte erstellt Ihre Anwendung?

Bewährtes Verfahren: Verringern der Größe von kompiliertem Code

Um die Größe von kompiliertem Code zu verringern, beachten Sie die folgenden Richtlinien:

- Legen Sie den richtigen Zugriff fest.
- Verzichten Sie auf die Erstellung von Schnittstellen.
- Verwenden Sie statische innere Klassen.
- Verzichten Sie auf die Initialisierung unnötiger Felder.
- Importieren Sie einzelne Klassen.

Festlegen des richtigen Zugriffs

Beim Erstellen von Code-Bibliotheken können Sie die Größe des kompilierten Codes mithilfe von geeigneten Zugriffsmodifikationen für Felder und Methoden beträchtlich verringern. Geben Sie Felder nach Möglichkeit als `privat` an. Dies ist nicht nur gute Codepraxis, sondern ermöglicht dem Compiler auch, die `COD`-Datei zu optimieren. Wenn möglich, verwenden Sie den Standardzugriff (`package`) statt den öffentlichen Zugriff (lassen Sie die Schlüsselwörter `public` und `protected` weg).

Vermeiden der Erstellung von Schnittstellen

Wenn Sie API-Bibliotheken anlegen, sollten Sie auf das Erstellen von Schnittstellen verzichten, es sei denn, Sie können mehrere Implementierungen der API absehen. Durch Schnittstellen wird der Code umfangreicher und langsamer.

Verwenden statischer innerer Klassen

Wenn Sie eine innere Klasse verwenden, um damit eine Klasse in einer anderen auszublenden, ohne dass die innere Klasse auf das Objekt der äußeren Klasse verweist, deklarieren Sie die innere Klasse als `statisch`. Damit unterbinden Sie die Erstellung eines Verweises auf die äußere Klasse.

Wenn Sie eine innere Klasse für den Namensumfang verwenden, deklarieren Sie sie als `statisch`.

Codebeispiel

```
class outer { static class inner { ... } }
```

Vermeiden unnötiger Felderinitialisierung

Wenn möglich, sollten Sie die automatische Initialisierung der Felder wie folgt zulassen:

- Objektverweise werden mit "Null" initialisiert. Objekte vom Typ `int`, `byte` oder `long` werden mit "0" initialisiert.
- Objekte vom Typ "Boolean" werden mit "False" initialisiert.

Lokale Variablen müssen explizit in einer Methode initialisiert werden.

Codebeispiel

```
class BetterExample { private int fieldsCount; private Field _fieldWithFocus;  
private boolean _validLayout; private boolean _validLayout; }
```

Importieren einzelner Klassen

Bei einer BlackBerry® Java Application, die nur wenige Klassen aus einem Paket verwendet, sollten die einzelnen Klassen importiert werden.

Codebeispiel

```
import net.rim.blackberry.api.browser.Browser;
```

Unterstützung mehrerer Sprachen

Die BlackBerry® Integrated Development Environment enthält eine Ressourcenmethode für das Erstellen von Zeichenfolgenressourcen. Die Lokalisierungs-API ist Teil des Pakets `net.rim.device.api.i18n`. Die MIDP-Anwendungen unterstützen keine Lokalisierung.

Die BlackBerry Integrated Development Environment speichert Ressourcen für ein Gebietsschema in einem Objekt vom Typ `ResourceBundle`. Ein Objekt vom Typ `ResourceBundleFamily` enthält eine Sammlung von `ResourceBundles`, die die Ressourcen für eine Anwendung gruppieren. Die Anwendung kann zwischen Sprachen wechseln, abhängig vom Gebietsschema des BlackBerry-Gerätebenutzers, ohne dass neue Ressourcenpakete erforderlich wären.

Mit der BlackBerry Integrated Development Environment können Sie jedes Ressourcenpaket in eine separat kompilierte COD-Datei kompilieren. Sie können die geeigneten COD-Dateien zusammen mit den anderen COD-Dateien für die Anwendung auf BlackBerry-Geräte laden.

Die Ressourcen sind in einer Hierarchie strukturiert, die auf Vererbung beruht. Wenn eine Zeichenfolge in einem Gebietsschema nicht definiert ist, wird eine Zeichenfolge aus dem nächstgelegenen Gebietsschema verwendet.

Bewährtes Verfahren: Speichern von Textzeichenfolgen in Ressourcendateien

Statt Text in den Quellcode einzubinden, lassen Sie die Anwendungen für die Lokalisierung (Anpassung für bestimmte Sprachen und Regionen) auf Ressourcendateien zugreifen.

Befolgen Sie die folgenden Richtlinien:

- Speichern Sie die Textzeichenfolgen für jedes Gebietschema in einer eigenen Ressourcendatei.
- Verwenden Sie in Ihrem Quellcode eindeutige Bezeichner, damit die richtigen Ressourcendateien aufgerufen werden.
- Lassen Sie die Anwendung die richtige Ressourcendatei dynamisch abrufen, die auf dem BlackBerry®-Gerät je nach dem Gebietschema des BlackBerry-Gerätebenutzers angezeigt werden soll.

Multimedia-Support

Audiounterstützung

Sie können eine BlackBerry® Java Application erstellen, die mit den von BlackBerry-Geräten unterstützten Audioformaten arbeitet. Die Art der von einem BlackBerry-Gerät unterstützten Audioformate ist abhängig von der Modellnummer des BlackBerry-Geräts.

Weitere Informationen zum Audiosupport auf einem BlackBerry-Gerät finden Sie unter www.blackberry.com/developers.

Bildunterstützung

Wenn ein BlackBerry®-Gerätebenutzer ein Bild mit einem BlackBerry-Gerät mit integrierter Kamera aufnimmt, speichert das BlackBerry-Gerät das Bild im Dateisystem auf dem BlackBerry-Gerät. Eine BlackBerry® Java Application kann auf die Bilder mithilfe der Dateiverbindungs-API für Java ME zugreifen, die in BlackBerry® Java® Development Environment Version 4.2 oder höher zur Verfügung steht. Die BlackBerry Java Application kann die Kameraanwendung aufrufen und Ereignisse abfragen, wenn Bilder zum Dateisystem hinzugefügt werden.

Videounterstützung

Sie können eine BlackBerry® Java Application erstellen, die Bilder anzeigt und die Multimedia-Inhalte mithilfe von Grafik-API-Klassen verarbeitet, um eine Videodatei auf BlackBerry-Geräten mit integriertem Media Player wiederzugeben.

Benutzeroberflächen- und der Navigationsdesign

3

BlackBerry-Gerät: Eingabe und Navigation für Benutzer

BlackBerry®-Geräte umfassen eine Tastatur, ein Trackwheel oder einen Trackball sowie eine Esc-Taste für die Eingabe und die Navigation. Die Esc-Taste bietet eine einfache Möglichkeit für BlackBerry-Gerätebenutzer, zum vorherigen Bildschirm zurückzuwechseln oder ein Menü oder Dialogfeld auf dem Bildschirm zu schließen.

Eine BlackBerry® Java Application für BlackBerry-Geräte sollte dem nachstehenden Modell für Eingabe und Navigation so weit wie möglich folgen.

- Klicken auf das Trackwheel oder den Trackball öffnet in der Regel ein Menü.
- Durch Drücken auf die Esc-Taste wechseln Sie zum vorherigen Bildschirm oder schließen Sie die Anwendung auf dem Hauptbildschirm.

Standardmäßig bieten die BlackBerry-Bildschirmobjekte diese Funktion ohne weiteres Anpassen. Sie müssen allerdings die gewünschten Menüelemente und die zusätzliche Benutzeroberflächen- und Navigationslogik hinzufügen.

Trackwheel im Vergleich zum Trackball

Trackball-Empfindlichkeit

Die Trackball-Empfindlichkeit bezeichnet den Umfang der Trackball-Bewegung, die erforderlich ist, damit das System die Bewegung als Navigationsereignis erkennt und entsprechend ein Navigationsereignis an die Softwareschicht zu senden. Die BlackBerry®-Gerätehardware misst die physische Trackball-Bewegung mithilfe spezieller Einheiten, die als Ticks bezeichnet werden. Wenn die Anzahl der Ticks entlang einer Achse den Schwellenwert des Systems oder einer BlackBerry® Java Application überschreitet, wird ein Navigationsereignis entlang dieser Achse an die Softwareschicht gesendet, und das System setzt die Tick-Zahl wieder auf Null zurück. Auch nach einer bestimmten Leerlaufzeit wird die Tick-Zahl wieder auf Null zurückgesetzt.

Mit der Trackball-API können Sie die Trackball-Empfindlichkeit festlegen. Eine hohe Trackball-Empfindlichkeit bedeutet einen niedrigeren Tick-Schwellenwert, weshalb schon kleine Trackball-Bewegungen ein Navigationsereignis auslösen. Umgekehrt entspricht eine niedrige Trackball-Empfindlichkeit einem größeren Tick-Schwellenwert, sodass ein Navigationsereignis erst mit größeren Trackball-Bewegungen ausgelöst wird.

Trackball-Bewegung

Mit der Trackball-API filtern Sie die Trackball-Bewegungsdaten, die die Hardware des BlackBerry®-Geräts an die Softwareschicht sendet. Die Trackball-API filtert Bewegungen "Rauschen" oder unerwünschte Bewegungen.

Mit der Trackball-API können Sie auch Einstellungen ändern, z. B. die Beschleunigung beim Bewegen des Trackballs. Eine höhere Einstellung für die Beschleunigung beim Bewegen des Trackballs kann dazu führen, dass die Softwareschicht die Bewegungen als scheinbar schneller erkennt als die BlackBerry-Gerätehardware, solange der Benutzer den Trackball fortlaufend rollt. Die Trackball-Empfindlichkeit steigt vorübergehend an, wenn der Benutzer den Trackball längere Zeit ununterbrochen rollt.

Trackwheel

BlackBerry®-Geräte vor der BlackBerry® Pearl™ 8100 Series sind nicht mit einem Trackball ausgestattet, sondern weisen an der rechten Geräteseite ein Trackwheel auf. Das Trackwheel ist die primäre Steuerung für die Benutzernavigation.

Benutzer haben damit die folgenden Möglichkeiten:

- Trackwheel rollen, um den Cursor in vertikaler Richtung zu bewegen
- Trackwheel rollen und gleichzeitig die Alt-Taste drücken, um den Cursor in horizontaler Richtung zu bewegen
- Mit dem Trackwheel klicken, um Objekte auszuwählen oder das Menü zu öffnen

Richtlinien

- Erkunden Sie Modelle sowohl mit Trackwheel- als auch mit Trackball-Navigation, und stellen Sie sicher, dass Ihre Anwendung auf beiden Gerätearten gut arbeitet.

Erstellen einer Benutzeroberfläche, die mit normalen BlackBerry-Benutzeroberflächen kompatibel ist

Mit MIDP-Standard-APIs und BlackBerry®-Benutzeroberflächen-APIs können Sie Benutzeroberflächen für BlackBerry® Java Application erstellen.

Die BlackBerry-Benutzeroberflächen-APIs sind eine Bibliothek von Benutzeroberflächen-Komponenten, die standardmäßige Layouts und Verhaltensweisen gemäß den Hauptanwendungen der BlackBerry-Geräte bieten.

- Screen-Komponenten liefern ein Standard-Bildschirmlayout, ein Standardmenü und ein Standardverhalten, wenn der BlackBerry-Gerätebenutzer die Esc-Taste drückt oder auf das Trackwheel oder den Trackball klickt.
- Field-Komponenten bieten Standard-Benutzeroberflächen-Elemente für die Datumsauswahl, für Optionen, Kontrollkästchen, Listen, Textfelder, Bezeichnungen und Steuerelemente der Fortschrittsanzeige.
- Layout-Manager geben einer Anwendung die Möglichkeit, Komponenten auf standardmäßige Weise in einem BlackBerry-Gerätebildschirm anzuordnen, z. B. horizontal, vertikal oder von links nach rechts.

Mit den BlackBerry-Benutzeroberflächen-APIs können Sie Benutzeroberflächen mit Tabellen, Rastern und andere speziellen Funktionen erstellen. BlackBerry Java Development Environment nutzt ein standardmäßiges Java-Ereignismodell, um bestimmte Arten von Ereignissen zu empfangen und darauf zu antworten. Anwendungen können Ereignisse von BlackBerry-Gerätebenutzern (z. B. wenn der BlackBerry-Gerätebenutzer auf das Trackwheel oder den Trackball klickt oder etwas über die Tastatur eingibt) sowie Systemereignisse erhalten und darauf reagieren, z. B. globale Warnungen, Echtzeit-Taktänderungen oder Verbindungen über den USB-Anschluss.

Speicherverwaltung

4

Verwalten des Speichers

Die BlackBerry® Java® Virtual Machine verwaltet die Speicherauslastung auf dem BlackBerry-Gerät. Die BlackBerry JVM ordnet Speicher zu, führt die Garbage Collection durch und tauscht automatisch Daten zwischen SRAM und Flash-Speicher aus. Die BlackBerry JVM muss außerdem den verfügbaren Speicher zwischen den Anwendungen auf dem BlackBerry-Gerät und der BlackBerry® Java Application aufteilen. Die Speicherkapazität bezeichnet den insgesamt verfügbaren Speicher; diese Speichermenge ist größer als der tatsächlich verfügbare Speicher, wenn alle Anwendungen und zugehörige Anwendungsdaten auf dem BlackBerry-Gerät vorhanden sind.

BlackBerry-Gerätespeicher

BlackBerry®-Geräte enthalten die folgenden Arten von Speicher:

Arbeitsspeicher	Beschreibung
Blitz	Das BlackBerry-Betriebssystem und alle Anwendungsmodule sind permanent im Flash-Speicher abgelegt. Wenn ein BlackBerry das BlackBerry-Gerät einschaltet, belegen das Kernbetriebssystem und die BlackBerry® Java Application-Module etwa 10 MB bis 15 MB des Flash-Speichers, je nach Version. Der Flash-Speicher kann E-Mail-Nachrichten, Terminplanerdaten und andere persönliche Informationen der BlackBerry-Gerätebenutzer aufnehmen und auch die Daten, die eine BlackBerry Java Application im Speicher ablegt.
SRAM	SRAM steuert die flüchtigen Datenobjekte und die Runtime-Prozesse.
Erweiterbare microSD-Speicherkarte	Auf der microSD-Karte können Mediendateien, Dokumente und permanente Daten aus einer BlackBerry Java Application gespeichert werden.

Zu reservierende wichtige Ressourcen

- Flash-Speicher: Der permanente Speicherplatz, der auf dem BlackBerry®-Gerät verfügbar ist, umfasst eine feste Menge des Flash-Speichers, in der Regel 8 MB bis 64 MB.
- Permanente Objekt-Handles: Die Handles, die jedem permanenten Objekt zugewiesen sind, werden ausschließlich von permanenten Objekten genutzt. Die Menge des Flash-Speichers auf dem BlackBerry-Gerät bestimmt die feste Anzahl von permanenten Objekt-Handles im System.

- Objekt-Handle: Jedem Objekt und jedem Array mit Primitiven ist ein Objekt-Handle zugeordnet. Die Menge des Flash-Speichers auf dem BlackBerry-Gerät bestimmt die feste Anzahl von Objekt-Handles im System.

Bewährtes Verfahren: Minimieren der Speicherbelegung

Um den Runtime-Speicher auf ein Minimum zu beschränken, beachten Sie die folgenden Richtlinien:

- Verwenden Sie primitive Typen (z. B. `int` oder `Boolesche Operatoren`) anstelle von Objekten (z. B. `Zeichenfolge` oder `Integer`).
- Verlassen Sie sich nicht voll und ganz auf den Garbage Collector.
- Verzichten Sie darauf, rasch viele Objekte zu erstellen.
- Setzen Sie die Objektverweise auf "Null", wenn sie nicht mehr verwendet werden.
- Verwenden Sie Objekte so oft wie möglich wieder.
- Lagern Sie arbeitsintensive Verarbeitung auf den Server aus. Filtern oder sortieren Sie die Daten beispielsweise, bevor Sie sie an das BlackBerry®-Gerät senden.

Verwalten von geringem verfügbarem Speicher

Der Manager für geringen Speicherplatz verwaltet die Speicherressourcen auf dem BlackBerry®-Gerät, wenn die verfügbaren Speicherressourcen unter einen bestimmten Schwellenwert fallen. Der Manager für geringen Speicherplatz versucht, belegten Speicher wieder freizugeben und damit mehr Speicher auf dem BlackBerry-Gerät verfügbar zu machen. Alle Anwendungen, einschließlich BlackBerry® Java Application, sollten mit dem Manager für geringen Speicherplatz zusammenarbeiten, um damit so viel Speicher wie möglich freizugeben, wenn das BlackBerry-Gerät nur noch geringe Speicherressourcen besitzt.

Erkennen von wenig verfügbarem Speicher auf einem BlackBerry-Gerät

In den folgenden Situationen versucht der Manager für geringen Speicherplatz, Speicherressourcen freizugeben:

- Die Menge des verfügbaren Flash-Speichers auf dem BlackBerry®-Gerät sinkt unter einen bestimmten Schwellenwert. Der Schwellenwert für den Flash-Speicher ist abhängig von der Menge an freiem Arbeitsspeicher im System. Der Schwellenwert für den Flash-Speicher liegt zwischen 400 KB und 800 KB.
- Die Anzahl der permanenten Objekt-Handles, die auf dem BlackBerry-Gerät verfügbar ist, sinkt unter 1000.
- Die Anzahl der Objekt-Handles, die auf dem BlackBerry-Gerät verfügbar ist, sinkt unter 1000.

Schonen der Ressourcen

Bewährtes Verfahren: Verwendung einer effizienten Datenstrukturauswahl

Die Datenstrukturauswahl definiert, wie viele Objekt-Handles und wie viel Flash-Speicher eine BlackBerry® Java Application nutzt. Durch eine falsche Datenstrukturauswahl werden wichtige Ressourcen belegt, ohne den Funktionsumfang der BlackBerry Java Application oder die Arbeitsumgebung für die BlackBerry-Gerätebenutzer zu verbessern.

Befolgen Sie die folgenden Richtlinien:

- Die Datenstruktur sollte aus der kleinstmöglichen Anzahl von Objekten bestehen, insbesondere wenn Sie Objekte auf höherer Ebene verwenden, wie `Vektoren` oder `Hashtables`. Diese Klassen bieten bedeutende Funktionen, sind jedoch keine effizienten Speichermethoden, und Sie sollten möglichst darauf verzichten, sie im permanenten Speicher zu verwenden.
- Nutzen Sie nach Möglichkeit Primitive anstelle von Objekten, weil Primitive die Anzahl von Objekt-Handles verringern, die auf dem BlackBerry-Gerät belegt werden. Ein Array mit Primitiven ist ein Objekt und verwendet ein Objekt-Handle.
- Zeichenfolge-Objekte sind ebenso effizient wie Byte-Arrays. Ein `Zeichenfolge`-Objekt verwendet nur ein Objekt-Handle und ist gleichwertig, wenn Ihre Anwendung alle Zeichen als ein Byte speichert. Anders gesagt: Der Wert von jedem Zeichen ist kleiner oder gleich dem Dezimalwert 255. Wenn Ihre Anwendung die Zeichen nicht als ein Byte speichern kann, können Sie die Zeichen als `Zeichenfolge` speichern, weil diese Methode dem Speichern eines `char`-Arrays entspricht.

Bewährtes Verfahren: Konsolidieren von Objekten zu Objektgruppen

Einer der häufigsten Fehler, denen Anwendungsentwickler begegnen, ist der Mangel an permanenten Objekt-Handles. Die Menge des Flash-Speichers auf dem BlackBerry®-Gerät bestimmt die feste Anzahl von permanenten Objekt-Handles, die im System verfügbar sind. Abhängig von der Datenstrukturauswahl können gespeicherte Datensätze schnell die Anzahl der permanenten Objekt-Handles ausschöpfen. Ein permanentes Objekt nutzt einen permanenten Objekt-Handle und einen Objekt-Handle, ein vergängliches Objekt dagegen nur ein Objekt-Handle.

Ein Datensatz mit zehn Zeichenfolge-Felder für Elemente wie Name, Telefonnummer und Adresse belegt 11 permanente Objekt-Handles, und zwar einen Handle für das Datensatzobjekt und je einen Handle für jede Zeichenfolge. Wenn in einer BlackBerry® Java Application 3.000 Datensätze gespeichert sind, nutzt die Anwendung 33.000 permanente Objekt-Handles, was die Anzahl der verfügbaren permanenten Objekt-Handles auf einem BlackBerry-Gerät mit 16 MB Flash-Speicher übersteigt.

Mit der Klasse `net.rim.device.api.system.ObjectGroup` können Sie die Objekt-Handles für ein Objekt zu einer Gruppe konsolidieren. Für das obige Beispiel ergibt sich: Wenn Sie den Datensatz gruppieren, belegt der Datensatz nur noch einen permanenten Objekt-Handle (statt wie bisher elf). Die Objekt-Handles für die Zeichenfolge-Felder werden unter dem Objekt-Handle des Datensatzes konsolidiert.

Wenn Sie Objekt-Handles zu einer Gruppe konsolidieren, ist das Objekt-Handle schreibgeschützt. Falls Sie Änderungen am Objekt vornehmen möchten, müssen Sie die Gruppierung zuerst aufheben. Sobald alle Änderungen erledigt sind, gruppieren Sie das Objekt wieder. Wenn Sie versuchen, ein gruppiertes Objekt zu ändern, ohne die Gruppierung zuerst aufzuheben, wird die Ausnahme `ObjectGroupReadOnlyException` ausgelöst.

Das Aufheben der Gruppierung eines Objekts wirkt sich auf die Leistung aus. Das System erstellt eine Kopie des gruppierten Objekts und weist den einzelnen Objekten in der Gruppe jeweils einen Handle zu. Deshalb sollten Sie die Gruppierung von Objekten nur dann aufheben, wenn dies unbedingt nötig ist.

Garbage Collection auf einem BlackBerry-Gerät

Garbage Collection für den Arbeitsspeicher auf einem BlackBerry

Die BlackBerry® Java® Virtual Machine startet nur dann einen Garbage Collection-Vorgang für den Arbeitsspeicher, wenn die BlackBerry JVM aufgrund von mangelndem Arbeitsspeicher kein Objekt mehr zuordnen kann. Der Garbage Collection-Vorgang für den Arbeitsspeicher nimmt in der Regel 500 bis 600 Millisekunden in Anspruch. Beim Garbage Collection-Vorgang werden alle vor kurzem zugeordneten Variablen entfernt, auf die der Arbeitsspeicher nicht mehr verweist. Um sicherzustellen, dass ein fehlender Verweis aus dem Arbeitsspeicher eine ausreichende Bedingung zum Entfernen des Objekts ist, kann ein Garbage Collection-Vorgang für den Arbeitsspeicher nur dann durchgeführt werden, wenn keine Objekte in den Flash-Speicher ausgelagert wurden.

Vollständige Garbage Collection auf einem BlackBerry-Gerät

Ein vollständiger Garbage Collection-Vorgang dauert durchschnittlich 1 Sekunde und sollte insgesamt weniger als 2 Sekunden in Anspruch nehmen. Der vollständige Garbage Collection-Vorgang umfasst die folgenden Aktionen:

- Es wird ein Garbage Collection-Vorgang für den Arbeitsspeicher ausgeführt.
- Alle Objekte im Flash-Speicher, auf die kein Verweis mehr zeigt oder die nicht mehr permanent sind, werden gekennzeichnet.
- Alle nicht permanenten Objekt-Handles im Arbeitsspeicher und im Flash-Speicher werden freigegeben.

In den folgenden Situationen startet das System ggf. einen vollständigen Garbage Collection-Vorgang:

- Die BlackBerry® Java® Virtual Machine kann kein Objekt mehr zuordnen, weil kein Speicherplatz mehr im Arbeitsspeicher verfügbar ist.
- Ein Prozess steht kurz davor, seine derzeit zugeordnete Heap-Größe zu überschreiten.
- Die BlackBerry JVM kann kein neues Objekt zuordnen, weil keine Objekt-Handles verfügbar sind.
- Das BlackBerry-Gerät befindet sich im Leerlauf.

Garbage Collection bei Leerlauf auf einem BlackBerry-Gerät

Der Garbage Collection-Vorgang findet nicht jedesmal statt, wenn das BlackBerry®-Gerät im Leerlauf befindet. Dieser Vorgang wird nur dann gestartet, wenn das System einen Garbage Collection-Vorgang als förderlich für die optimale Systemleistung und die maximale Akkuleistung erkennt.

Um die Leistung zu erhöhen, ohne das Arbeitsumfeld der BlackBerry-Gerätebenutzer zu beeinträchtigen, versucht das System, die folgenden Garbage Collection-Vorgänge auszuführen, wenn sich das BlackBerry-Gerät im Leerlauf befindet:

- Ein vollständiger Garbage Collection-Vorgang kann stattfinden, wenn das BlackBerry-Gerät über relativ kurze Zeit im Leerlauf ist.
- Ein gründlicher Garbage Collection-Vorgang kann stattfinden, wenn das BlackBerry-Gerät über längere Zeit im Leerlauf ist.

Datenverwaltung

5

Das BlackBerry®-Gerät bietet APIs zum Speichern von Daten in den permanenten Speicher auf dem BlackBerry-Gerät. Die BlackBerry Persistent Store-APIs und die MIDP-RMS-APIs (Unterstützung für JSR 37 und JSR 118) sind auf allen Java®-gestützten BlackBerry-Geräten verfügbar. BlackBerry-Geräte, auf denen BlackBerry® Device Software Version 4.2 oder höher ausgeführt wird, bietet ein traditionelles Dateisystem und unterstützt das Speichern von Inhalten direkt im Dateisystem mithilfe von JSR-75-APIs. Sowohl mit den BlackBerry Persistent Store-APIs als auch mit den MIDP-RMS-APIs können Sie Daten permanent im Flash-Speicher ablegen. Die Daten bleiben selbst dann erhalten, wenn Sie den Akku aus dem BlackBerry-Gerät herausnehmen.

Unterstützung für APIs zum Speichern von Daten in permanentem Speicher

APIs für die dauerhafte Speicherung

Die BlackBerry®-APIs für die dauerhafte Speicherung bieten eine flexible und stabile Schnittstelle für die Datenspeicherung. Mit dem BlackBerry-APIs für die dauerhafte Speicherung können Sie ganze Java®-Objekte im Speicher ablegen, ohne die Daten erst serialisieren zu müssen. Beim Starten der Anwendung können Sie das Java-Objekt aus dem Speicher abrufen und die Daten bearbeiten. Für den permanenten Speicher besteht keine Größenbeschränkung; die Größe eines einzelnen Objekts im Speicher ist allerdings auf 64 KB begrenzt.

Die BlackBerry-APIs für die dauerhafte Speicherung bilden kein relationales Datenbankmodell. Sie müssen ein effektives Objektmodell erstellen und die Beziehungen zwischen Objekten je nach Bedarf mithilfe von Indizes und Hash-Tabellen verwalten.

MIDP Record Management System-APIs

Die RMS-APIs bieten ein einfaches Datensatzverwaltungssystem, mit denen Sie ein Datenspeicherobjekt erstellen und eine Reihe von Datensätzen innerhalb dieses Objekts beibehalten können. Jeder Datensatz ist ein Bytearray, weshalb Sie Ihre Daten zuerst in einem Bytearray-Format serialisieren müssen, bevor Sie sie speichern können. Die RMS-APIs bieten keine inhärente Indizierung und keine Beziehungen zwischen Datensätzen. Die Größenbegrenzung für einen einzelnen RMS-Datenspeicher liegt bei maximal 64 KB. Eine Anwendung kann mehrere RMS-Datenspeicher erstellen, um so größere Datenmengen beizubehalten. Die RMS-APIs sind Teil der MIDP-Standardspezifikation. Aus diesem Grund können alle Geräte, die MIDP unterstützen, auch die RMS-APIs unterstützen.

Dateiverbindungs-APIs

Die Dateiverbindungs-APIs stellen ein herkömmliches Dateisystem bereit und unterstützen die Speicherung von Daten direkt im Dateisystem auf dem BlackBerry®-Gerät oder auf einer microSD-Karte. Mit Windows® können Sie Daten im Dateisystem abrufen und auf einen Computer verschieben.

Speicher auf Wechselmedien

Zugriff auf Daten auf der microSD-Medienkarte

Das Paket `javax.microedition.io.file` unterstützt die JSR-75-Dateiverbindungs-APIs und wird in Anwendungen für den Zugriff auf das Dateisystem auf der microSD-Medienkarte verwendet. Sie können auch die `FileConnection`-Schnittstelle implementieren und damit auf Klingeltöne und Kamerabilder für BlackBerry®-Geräte zugreifen.

Klasse oder Schnittstelle	Beschreibung
<code>ConnectionClosedException</code>	Diese Ausnahme wird ausgelöst, wenn eine Anwendung eine Methode für eine geschlossene Dateiverbindung aufruft.
<code>FileConnection</code>	Eine Anwendung kann mit dieser API auf Dateien oder Verzeichnisse zugreifen.
<code>FileSystemListener</code>	Eine Anwendung kann mit dieser API den Empfang von Statusbenachrichtigungen aktivieren, wenn die Anwendung einen Dateisystemstamm hinzufügt oder entfernt.
<code>FileSystemRegistry</code>	Eine Anwendung kann diese API als zentrale Registrierung für Dateisystem-Listener verwenden, die auf das Hinzufügen oder Entfernen von Dateisystemen achten.
<code>IllegalModeException</code>	Diese Ausnahme wird ausgelöst, wenn eine Methode einen bestimmten Sicherheitsmodus (zum Beispiel LESEN oder SCHREIBEN) erfordert und die offene Verbindung sich nicht in diesem Modus befindet.

Sichern und Synchronisieren von Daten

Der BlackBerry® Desktop Manager enthält ein Tool zum Sichern und Wiederherstellen, mit dem ein BlackBerry-Gerätebenutzer die BlackBerry-Gerätedaten in einer Datei auf einem Computer speichern und umgekehrt Daten auf dem BlackBerry-Gerät wiederherstellen kann.

Wenn eine Anwendung die Synchronisierungs-API verwendet, kann der BlackBerry Desktop Manager die Anwendungsdatenbank gleichzeitig mit den anderen BlackBerry-Gerätedatenbanken sichern und wiederherstellen. Mit der Synchronisierungs-API können Sie Datenarchive erstellen oder Anwendungsdatenbanken ausfüllen, sobald das BlackBerry-Gerät eine Verbindung zum Computer des BlackBerry-Gerätebenutzer herstellt.

Um Daten mit Remotedatenquellen zu synchronisieren, müssen Sie die Synchronisierungslogik in Ihre BlackBerry Java® Application integrieren. Bei den meisten Anwendungen werden Daten an serverseitige Anwendungen mithilfe des normalen HTTP- oder TCP/IP-Protokolls über das drahtlose Netzwerk bzw. das Internet oder das Unternehmensintranet gesendet. Mit

XML-APIs können Sie Daten im XML-Format erzeugen und analysieren, die dann über das drahtlose Netzwerk gesendet und empfangen werden. Die clientseitigen und serverseitigen Anwendungen müssen jedoch die Daten fehlerfrei schreiben und lesen und die erfolgreiche Übertragung bestätigen.

Eine BlackBerry Java Application kann mithilfe der BlackBerry-Desktop-Synchronisierungs-APIs und des BlackBerry Desktop Manager auch eine Verbindung zu einer Anwendung auf einem Computer herstellen und die Daten dann über eine USB-Verbindung senden. In diesem Fall müssen Sie eine Anwendung für Windows® erstellen, die die Daten über eine Add-In-Aufgabe für den BlackBerry Desktop Manager vom Client lesen kann. Der BlackBerry-Gerätebenutzer muss die Synchronisierung manuell starten. Hierzu muss das Add-In für den BlackBerry Desktop Manager ausgeführt werden, mit dem die Anwendung auf dem BlackBerry-Gerät angewiesen wird, die Daten an die Anwendung auf dem Computer zu senden. Alternativ können Sie auch Daten über die systemeigenen USB-Protokolle in die Anwendung auf dem Computer schreiben.

Drahtloser Datentransport

6

Drahtlose Gateways

Java[®]-Anwendungen für BlackBerry[®]-Geräte können Verbindungen zum drahtlosen Netzwerk über normale HTTP-, HTTPS- und TCP-Socket-Protokolle einrichten. Wenn eine Anwendung eine Verbindung über das drahtlose Netzwerk herstellt, stehen dabei zwei drahtlose Gateways als Proxy für die Verbindung zum Internet oder zum Unternehmensintranet zur Wahl. Sie können Ihre Anwendung auf den Standard-Gateway abstimmen, der den BlackBerry-Gerätebenutzern zur Verfügung steht, oder Sie können Ihren Code so anpassen, dass ein bevorzugtes Gateway verwendet wird. Legen Sie in Ihrer Anwendung fest, dass explizit der bevorzugte Gateway für die Verbindung ausgewählt werden soll und das Ausweichen auf den Standard-Gateway nur dann zulässig ist, wenn die bevorzugte Methode nicht verfügbar ist. Damit vermindern Sie die Anzahl von Netzwerkverbindungsproblemen, denen Ihre Kunden gegenüberstehen, und Ihre Anwendung kann ein konsistentes Konnektivitätsmodell für alle Netzwerktypen und sämtliche Funkbenutzer verwenden.

Verwenden des BlackBerry Enterprise Server als Netzwerk-Gateway

Wenn Sie den BlackBerry[®] Enterprise Server als Netzwerk-Gateway verwenden, wird der gesamte Datenverkehr zwischen Ihrer Anwendung und dem BlackBerry Enterprise Server mithilfe der AES- oder Triple-DES-Verschlüsselung verschlüsselt. Der BlackBerry Enterprise Server befindet sich hinter der Firewall des Unternehmens und bietet eine inhärente Datenverschlüsselung, sodass die Anwendungen mit den Anwendungs- und Webservern kommunizieren können, die sich im Intranet des Unternehmens befinden. Die BlackBerry[®] Mobile Data System-Komponente in BlackBerry Enterprise Server enthält die BlackBerry MDS Services, die wiederum einen HTTP- und TCP/IP-Proxydienst umfassen, der von der BlackBerry[®] Java[®] Application als sicheres Gateway für die Verwaltung von HTTP- und TCP/IP-Verbindungen zum Intranet herangezogen werden kann.

Wenn Ihre Anwendung eine Verbindung zum Internet herstellt, können Sie ggf. den BlackBerry Enterprise Server als Gateway nutzen. Netzwerkanforderungen werden hinter der Firewall des Unternehmens zum BlackBerry Enterprise Server geleitet, der die Netzwerkanforderung an das Internet durch die Unternehmens-Firewall hindurch vornimmt. Die Administratoren können eine IT-Richtlinie festlegen, mit denen sichergestellt wird, dass der BlackBerry Enterprise Server als Gateway für den gesamten drahtlosen Netzwerkverkehr fungiert, also auch für den Datenverkehr, der für das Internet bestimmt ist.

Wenn Ihre Anwendung eine Verbindung zum Internet herstellt, können Sie die Verbindungen sowohl mit dem BlackBerry[®] Internet Service als auch mit dem Internet-Gateway Ihres Mobilfunkanbieter verwalten.

Verwenden des Internet-Gateways des Mobilfunkanbieters

Die meisten Mobilfunkanbieter bieten ihr eigenes Internet-Gateway mit direkter TCP/IP-Konnektivität zum Internet an. Manche Mobilfunkanbieter bieten zudem ein WAP-Gateway (Wireless Application Protocol), mit dem HTTP-Verbindungen über das WAP-Protokoll erfolgen können. Eine BlackBerry[®] Java[®] Application kann mit beiden Gateways eine Verbindung zum Internet herstellen. Wenn Ihre Anwendung sich an BlackBerry-Gerätebenutzer richtet, die ein bestimmtes Funknetzwerk nutzen, führt

meist die Verwendung des Internet-Gateways des Mobilfunkanbieters zu guten Ergebnissen. Richtet sich Ihre Anwendung dagegen an BlackBerry-Gerätebenutzer in verschiedenen Funknetzen, stehen Sie vor einigen Herausforderungen, wenn Sie Ihre Anwendung mit den verschiedenen Internet-Gateways testen und eine konsistente und zuverlässige Arbeitsumgebung bieten sollen. Verwenden Sie daher den BlackBerry® Internet Service, und setzen Sie das Internet-Gateway des Mobilfunkanbieters als Standardverbindungstyp ein, wenn der BlackBerry Internet Service nicht verfügbar ist.

Alternative Optionen für den Datentransport

Datentransport per E-Mail

Mit den BlackBerry®-APIs können Sie eine BlackBerry® Java® Application erstellen, bei der Daten per E-Mail versendet und empfangen werden. E-Mail bietet eine effektive Möglichkeit für die proaktive Verteilung von Inhalten an BlackBerry-Gerätebenutzer, wenn die traditionellen Push-Modelle nicht verfügbar sind. Eine BlackBerry Java Application kann über die BlackBerry-APIs E-Mail-Nachrichten senden und eingehende E-Mail-Nachrichten abfragen. Eine BlackBerry Java Application kann auch auf die Details und die Kopfzeilen der E-Mail-Nachrichten zugreifen, die lokal auf dem BlackBerry-Gerät gespeichert sind, und Listener für Änderungen am Status einer E-Mail-Nachricht registrieren.

Datentransport per SMS

Mit den BlackBerry®-APIs können Sie Anwendungen erstellen, bei denen Daten per SMS versendet und empfangen werden. Die BlackBerry-APIs ermöglichen einer Anwendung, SMS-Nachrichten zu senden und eingehende SMS-Nachrichten abzufragen.

Datentransport per PIN-Nachrichtenübermittlung

Bei der PIN-Nachrichtenübermittlung wird der Datenkanal statt des Stimmenkanals verwendet, sodass Sie das BlackBerry®-Zielgerät über seine eindeutige PIN ansprechen können. Mit der PIN-Nachrichtenübermittlung können Sie nur Daten von einem BlackBerry-Gerät an ein anderes senden. Die BlackBerry-APIs können außerdem einer Anwendung das programmgesteuerte Senden und Empfangen von BlackBerry-PIN-Nachrichten ermöglichen. Die PIN-Nachrichtenübermittlung bietet eine effektive Möglichkeit, PIN-Anwendungen umzusetzen, die sich ausschließlich an BlackBerry-Gerätebenutzer richten.

Integrieren von BlackBerry-Anwendungen

7

Hinzufügen benutzerdefinierter Menüelemente

Eine BlackBerry® Java® Application kann dem Menü für die E-Mail-, Terminplaner- und Telefonanwendungen auf dem BlackBerry-Gerät benutzerdefinierte Menüelemente hinzufügen. Wenn ein BlackBerry-Gerätebenutzer das benutzerdefinierte Menüelement auswählt, startet die BlackBerry Java Application mit einem Verweis auf das Objekt, das der BlackBerry-Gerätebenutzer ausgewählt hat. Beispielsweise kann eine BlackBerry Java Application der E-Mail-Anwendung ein Menüelement hinzufügen, mit dem der Standort des Absenders angezeigt wird. Wenn ein BlackBerry-Gerätebenutzer das Menüelement auswählt, startet die BlackBerry Java Application mit einem Verweis auf das E-Mail-Objekt, das derzeit markiert ist oder das der BlackBerry-Gerätebenutzer öffnet. Die BlackBerry Java Application ermittelt den Standort des Absenders anhand der E-Mail-Adresse (entweder durch Nachschlagen in der Kontaktliste oder durch Abrufen von Daten von einem Remoteserver) und wird dann in den Vordergrund gestellt und blendet eine Karte ein.

Integrieren in Anwendungen aus der BlackBerry

Eine BlackBerry®-Geräteanwendung kann eine Anwendung aus der BlackBerry aufrufen (z. B. E-Mail, Terminplaner, Telefon, Browser oder Kamera), um eine Aktion auszuführen oder Daten auf dem Bildschirm anzuzeigen. Die folgenden Beispiele zeigen, wie eine BlackBerry-Geräteanwendung eine Anwendung aus der BlackBerry aufrufen kann:

- Die BlackBerry-Geräteanwendung ruft den Kalender auf und zeigt ein bestimmtes Datum oder einen bestimmten Kalendereintrag an.
- Die BlackBerry-Geräteanwendung ruft das Adressbuch auf und zeigt einen bestimmten Kontakt an.
- Die BlackBerry-Geräteanwendung ruft den Browser auf und öffnet eine bestimmte Website.
- Die BlackBerry-Geräteanwendung ruft das Telefon auf und wählt eine bestimmte Telefonnummer.

Zugriff auf E-Mail- und Terminplanerdaten

Eine BlackBerry® Java® Application kann über die BlackBerry-APIs auf die Details von E-Mail-Nachrichten, Kontakten, Kalenderereignissen, Aufgaben und Telefonprotokolle zugreifen, die auf dem BlackBerry-Gerät gespeichert sind. Eine BlackBerry Java Application kann die Informationen lesen und aktualisieren sowie neue Einträge erstellen.

Verwenden des BlackBerry Messenger mit einer BlackBerry Application

Sie können eine BlackBerry® Java® Application in die BlackBerry® Messenger-Anwendung integrieren. Dies ist beispielsweise dann von Nutzen, wenn Sie eine Spielanwendung für das BlackBerry-Gerät erstellen, bei der die Spieler abwechselnd einen Spielzug tätigen.

Zum Erstellen einer BlackBerry Java Application, die in die BlackBerry Messenger-Anwendung integriert ist, können Sie die Klassen im Paket `net.rim.blackberry.api.blackberrymessenger` verwenden. Weitere Informationen zur Verwendung der Klasse `BlackBerryMessenger` finden Sie in der BlackBerry-API-Referenz.

Verwenden von Listnern für Reaktionen auf Anwendungsänderungen

Eine BlackBerry® Java® Application kann Änderungs-Listener für die E-Mail- und Terminplanerdatenspeicher sowie in der Telefonanwendung registrieren. Die Listener ermöglichen der BlackBerry Java Application eine sofortige Reaktion, wenn der BlackBerry-Gerätebenutzer ein lokales Ereignis auslöst. Mit den E-Mail- und Terminplanerdaten-Listener können Sie eine BlackBerry Java Application benachrichtigen lassen, sobald neue Einträge ankommen oder wenn der BlackBerry-Gerätebenutzer Änderungen an den vorhandenen Daten ausführt, z. B. Ergänzungen, Löschungen oder Aktualisierungen. Mit Telefon-Listnern können Sie Anruferaktionen abfragen, z. B. das Einleiten neuer Anrufe oder das Beenden von Anrufen.

Sicherheitsbedingungen

8

Datenverschlüsselung und die BlackBerry Application

Datenverschlüsselung beim Transport

Wenn Sie den BlackBerry® Enterprise Server Enterprise Server als Netzwerk-Gateway für Ihre Anwendung verwenden, verschlüsselt der BlackBerry Enterprise Server die Daten an allen Punkten in der Verbindung zwischen dem BlackBerry-Gerät und dem BlackBerry Enterprise Server hinter der Firewall des Unternehmens mit der AES- oder TripleDES-Verschlüsselung. Sollen die Daten zwischen dem BlackBerry Enterprise Server und dem Zielsever weiter verschlüsselt werden, können Sie das HTTPS-Protokoll verwenden und die SSL/TLS-Verschlüsselung heranziehen.

Wenn Ihre Anwendung den BlackBerry® Internet Service Internet Service oder das Internet-Gateway des Mobilfunkanbieter verwendet, wird der Datenverkehr nicht verschlüsselt. Auf Wunsch Ihrer BlackBerry-Gerätebenutzer können Sie die Daten mit HTTPS verschlüsseln, oder Sie können mit den Java®-APIs für die Verschlüsselung Ihre eigene Kryptografie mit symmetrischem oder öffentlichem Schlüssel anwenden.

Datenverschlüsselung auf dem BlackBerry-Gerät

Administratoren können eine IT-Richtlinie festlegen, um sicherzustellen, dass alle Daten der BlackBerry®-Gerätebenutzer, die in den BlackBerry-Geräten gespeichert sind, lokal im Flash-Speicher verschlüsselt werden. Sie können eine BlackBerry Java® Application erstellen, die die Daten mithilfe von APIs registriert, sodass der Verschlüsselungsdienst die Daten mit demselben Sicherheitsschlüssel verschlüsselt und dann erst im Flash-Speicher ablegt.

Zugriff auf Speicher

Die BlackBerry® Java® Development Environment soll Anwendungen daran hindern, unbeabsichtigt oder vorsätzlich Probleme in anderen Anwendungen oder auf dem BlackBerry-Gerät zu verursachen. Die BlackBerry-Anwendungen können nur in den BlackBerry-Gerätespeicher schreiben, den die BlackBerry® Java® Virtual Machine verwendet; der Zugriff auf den virtuellen Arbeitsspeicher oder den permanenten Speicher anderer Anwendungen ist nicht möglich (sofern sie nicht ausdrücklich den entsprechenden Zugriff hierfür erhalten). Eine BlackBerry® Java Application kann nur über bestimmte BlackBerry-APIs auf permanenten Speicher oder auf Benutzerdaten zugreifen oder mit anderen Anwendungen kommunizieren. Eine BlackBerry Java Application, die diese BlackBerry-APIs verwendet, muss durch Research In Motion digital signiert werden. Auf diese Weise ist ein Prüfpfad für Anwendungen gewährleistet, die vertrauliche APIs nutzen.

Authentifizierung

Authentifizierungs- und IT-Richtlinie für BlackBerry-Geräte

BlackBerry®-Gerätebenutzer können ein Kennwort für ihre BlackBerry-Geräte festlegen. Wenn das Gerätekenwort aktiv ist, müssen die BlackBerry-Gerätebenutzer das Kennwort eingeben, um auf die Daten und Anwendungen zugreifen zu können. Die Verwendung von Gerätekenwörtern ist ein guter erster Schritt zur Begrenzung des Zugriffs auf Ihre BlackBerry Java® Application auf dem BlackBerry-Gerät.

Administratoren können mit den IT-Richtlinien im BlackBerry® Enterprise Server sicherstellen, dass die BlackBerry-Geräte im Unternehmen mit einem Kennwort geschützt sind. Mit den IT-Richtlinien können Administratoren außerdem von einem Remotestandort aus ein BlackBerry-Gerät sperren, das Kennwort ändern oder auch alle Daten entfernen.

Anwendungsauthentifizierung

Bei Anwendungen, in denen Sicherheitsmerkmale von entscheidender Bedeutung sind, sollten Sie ggf. einen Anmeldebildschirm einrichten, mit dem der BlackBerry®-Gerätebenutzer aufgefordert wird, sich zunächst beim BlackBerry-Gerät anzumelden, bevor er es verwenden kann. Die UI-Klassen bieten einfache Kennwortfelder, in denen die Texteingabe als Sternchen wiedergegeben werden. Anmeldebildschirme könnten sich negativ auf den Bedienungskomfort der BlackBerry-Gerätebenutzer auswirken, und wenn der BlackBerry-Gerätebenutzer ein Kennwort zum Schutz des BlackBerry-Geräts festlegt, ist für Ihre Anwendung wahrscheinlich kein Anmeldebildschirm mehr erforderlich.

Serverseitige Authentifizierung

Wenn Ihre Anwendung eine Verbindung zu einer Anwendung auf einem Server bzw. zum Internet oder zu einem Intranet herstellt, sollen ggf. zusätzliche Authentifizierungsfunktionen eingebunden werden, wenn die BlackBerry®-Gerätebenutzer sich beim Server anmelden. Die meisten Anwendungen, die eine Authentifizierung der Benutzer erfordern, nutzen die HTTP-Standardauthentifizierung, bei der eine einfache Kombination aus Benutzername und Kennwort zum Einsatz kommt. Sie können die HTTP-Standardauthentifizierung verwenden, indem Sie die richtigen HTTP-Kopfzeilen hinzufügen, während die HTTP-Verbindung geöffnet wird. Außerdem können Sie eine erweiterte Authentifizierung mithilfe von Zertifikaten hinzufügen. Bei den meisten Anwendungen ist dies jedoch nicht erforderlich.

Kontrollierte APIs und Codesignaturen

Research In Motion verfolgt die Verwendung vertraulicher APIs aus Gründen der Sicherheit und der Exportkontrolle. In der BlackBerry®-API-Referenz bezeichnet RIM eine kontrollierte Klasse oder Methode mit einem Schlosssymbol oder einem Signaturhinweis. Um kontrollierte Klassen oder Methoden in Ihren Anwendungen zu verwenden (und bevor Sie die COD-Dateien

der Anwendung auf dem BlackBerry-Gerät installieren können), müssen Sie Ihre Anwendung mit einem Schlüssel oder Signatur von RIM signieren. Auch andere Funktionen, z. B. die Möglichkeit zur Ausführung beim Starten der Anwendung, könnten eine Signierung Ihrer Anwendungen notwendig machen.

Der RIM-Registrierungsprozess umfasst den Großteil der kontrollierten APIs; einige Kryptografieklassen für die Verschlüsselung mit öffentlichen und privaten Schlüsseln enthalten jedoch Technologien von Certicom™. Um diese Klassen zu verwenden, müssen Sie sich direkt bei Certicom registrieren und die entsprechende Genehmigung einholen. Der RIM-Registrierungsprozess erstreckt sich nicht auf die Verwendung von Certicom-Klassen.

Mit dem BlackBerry® Smartphone Simulator können Sie Ihren Code testen und debuggen, bevor Sie die Codesignaturen erhalten. Sie müssen die Anwendung signieren, bevor Sie sie auf BlackBerry-Geräten installieren. Sie müssen nicht Ihren eigentlichen Code an RIM senden. Mit dem BlackBerry® Signature Tool können Sie einen SHA-1-Hash Ihrer Codedatei senden, damit das Signaturstellensystem die erforderliche Signatur generieren kann.

Weitere Informationen zum Registrieren und zum Anfordern von Codesignaturen finden Sie im *BlackBerry Signing Authority Tool Version 1.0 – Password Based Administrator Guide*. Weitere Informationen zum Registrieren und Verwenden von Klassen finden Sie unter www.blackberry.com/developers/index.shtml.

BlackBerry-APIs mit kontrolliertem Zugriff

Sie können Anwendungen im BlackBerry® Smartphone Simulator ausführen, die kontrollierte APIs ohne Codesignaturen verwenden. Sie müssen allerdings Codesignaturen von Research In Motion anfordern, bevor Sie diese Anwendungen auf BlackBerry-Geräten installieren können.

Sie können die folgenden Kategorien von RIM-kontrollierten APIs verwenden:

- Laufzeit-APIs
- BlackBerry® Application-APIs
- BlackBerry-Kryptografie-API

Wenn Sie eines der folgenden BlackBerry-API-Pakete verwenden, benötigen Sie Codesignaturen für Ihre Anwendung, bevor Sie sie auf einem BlackBerry-Gerät installieren können:

- `net.rim.blackberry.api.browser`
- `net.rim.blackberry.api.invoke`
- `net.rim.blackberry.api.mail`
- `net.rim.blackberry.api.mail.event`
- `net.rim.blackberry.api.menuitem`
- `net.rim.blackberry.api.options`
- `net.rim.blackberry.api.pdap`
- `net.rim.blackberry.api.phone`
- `net.rim.blackberry.api.phone.phonelogs`
- `net.rim.device.api.browser.field`

- `net.rim.device.api.browser.plugin`
- `net.rim.device.api.crypto.*`
- `net.rim.device.api.io.http`
- `net.rim.device.api.notification`
- `net.rim.device.api.servicebook`
- `net.rim.device.api.synchronization`
- `net.rim.device.api.system`

Weitere Informationen zu RIM-kontrollierten APIs finden Sie in der BlackBerry-API-Referenz.

Unterstützung für IT-Richtlinien

Mit den APIs im Paket `net.rim.device.api.i18n` können Sie auf die IT-Richtlinieninformationen auf dem BlackBerry®-Gerät zugreifen und das Verhalten oder die Funktionsweise einer BlackBerry-Geräteanwendung ändern. Jedes IT-Richtlinienelement besteht aus einem Namen, einer Beschreibung und einem Wert. Der Wert kann eine Zeichenfolge, eine ganze Zahl oder ein Boolescher Wert sein.

Die IT-Richtlinieneinstellungen der BlackBerry-Geräte werden automatisch synchronisiert und über das drahtlose Netzwerk mit BlackBerry® Device Software Version 3.6 oder höher aktualisiert. Bei früheren Versionen der BlackBerry Device Software werden die IT-Richtlinieneinstellungen aktualisiert, sobald der BlackBerry-Gerätebenutzer das BlackBerry-Gerät mit dem Computer synchronisiert.

Weitere Informationen zu IT-Richtlinien finden Sie im *Referenzhandbuch für BlackBerry Enterprise Server-Richtlinien*.

Anwendungssteuerung

Die Richtlinienregeln zur Anwendungssteuerung des BlackBerry® Enterprise Server ermöglichen es, die Installation bestimmter Anwendungen auf dem BlackBerry-Gerät zuzulassen oder zu unterbinden und beschränken die Berechtigungen der Anwendungen auf dem BlackBerry-Gerät. Administratoren können beispielsweise mit der Richtlinie zur Anwendungssteuerung sicherstellen, dass eine Spieleanwendung auf dem BlackBerry-Gerät nicht auf die Telefonanwendung zugreifen kann.

Der Administrator kann Richtlinien zur Anwendungssteuerung nur dann anwenden, wenn das BlackBerry-Gerät einem BlackBerry Enterprise Server zugeordnet ist. Wenn der Administrator oder ein BlackBerry-Gerätebenutzer der Anwendung den Zugriff auf einen der geschützten Bereich verweigert, löst die Anwendung eine `ControlledAccessException` aus.

Dateiverschlüsselung bei microSD-Karten

Verschlüsseln der Daten auf einer microSD-Medienkarte

Wenn eine BlackBerry® Java® Application auf eine Datei auf der microSD-Speicherkarte zugreift, erfolgt die Dateientschlüsselung, und die Datei wird zum Lesen durch eine Anwendung in den Arbeitsspeicher verschoben. Damit eine BlackBerry Java Application auf eine kennwortgeschützte Datei zugreifen kann, darf das BlackBerry-Gerät nicht gesperrt sein. Verschlüsselte Dateien besitzen die Dateinamenerweiterung REM und können auf BlackBerry-fremden Plattformen nicht entschlüsselt werden.

Wenn der NVRAM entfernt ist und die microSD-Medienkarte mit einem BlackBerry-Geräteschlüssel gesperrt ist, sind die Daten auf der microSD-Medienkarte nicht mehr zugänglich. Um nicht zugängliche Daten zu entfernen, starten Sie das BlackBerry-Gerät, und entfernen Sie alle verschlüsselten Mediendateien.

Das BlackBerry-Gerät verschlüsselt die Mediendateien auf dem BlackBerry-Gerät mithilfe eines Hauptschlüssels, der auf der microSD-Medienkarte gespeichert ist. Der Hauptschlüssel sorgt dafür, dass das BlackBerry-Gerät nicht alle Mediendateien entschlüsseln oder wieder verschlüsseln muss, wenn Sie die Verschlüsselung deaktivieren oder das Kennwort ändern.

Verwenden der microSD-Medienkarte auf mehreren BlackBerry-Geräten

Wenn der BlackBerry® -Gerätebenutzer die microSD-Medienkarte in ein BlackBerry-Gerät einlegt, bei dem kein Gerätekenwort verwendet wird bzw. bei dem ein Kennwort verwendet wird, das den Hauptschlüssel der microSD-Medienkarte nicht erfolgreich entschlüsselt, fordert das BlackBerry-Gerät den BlackBerry-Gerätebenutzer dazu auf, das Kennwort für die microSD-Medienkarte einzugeben. Ist das BlackBerry-Gerät mit einem Kennwort geschützt, kann der BlackBerry-Gerätebenutzer bei dieser Aufforderung das Kennwort der microSD-Medienkarte durch das BlackBerry-Gerätekenwort zu ersetzen.

IT-Richtlinien und die microSD-Medienkarte

Sie können die IT-Richtlinie über die Verschlüsselung von Daten, die auf der microSD-Medienkarte gespeichert sind, auf beliebige neue oder geänderte Dateien anwenden, die Sie auf der microSD-Medienkarte speichern. Hierbei werden nur die Dateien verschlüsselt, die Sie auf der microSD-Medienkarte speichern, nachdem ein Administrator die IT-Richtlinie festgelegt hat. Der gesamte Inhalt mit Ausnahme von Mediendateien wird verschlüsselt.

Testen einer BlackBerry Java Application

9

Verschleiern einer BlackBerry Java Application

Der Compiler für die BlackBerry® Java® Development Environment und das BlackBerry® JDE Plug-in für Eclipse™ minimieren die Größe einer Anwendung. Der Compiler erstellt eine COD-Datei, die verschleierungsähnliche Services anbietet. Diese Services sorgen ähnlich wie Verschleierungspakete dafür, die Größe der COD-Datei zu verringern. Die BlackBerry Java Development Environment entfernt beispielsweise die folgenden Daten aus einer COD-Datei:

- sämtliche Debug-Informationen
- die Namen lokaler Variablen
- Quellzeilennummern
- die Namen privater Methoden und Mitglieder.

In der Regel müssen Sie neben der standardmäßigen Verschleierung in BlackBerry JDE keine zusätzliche Verschleierung für Ihre Anwendungen einrichten. Research In Motion führt selbst keine zusätzliche Verschleierung eigener Produkte durch.

Die BlackBerry JDE umfasst keine Unterstützung für Verschleierung mit Drittanbietertools. Zur Verschleierung von COD-Dateien für die Verwendung auf BlackBerry-Geräten müssen Sie ein Befehlszeilenverfahren bereitstellen.

Vorabüberprüfen einer BlackBerry Java Application

Wenn Sie Ihre Klassen vorab überprüfen, verringern Sie den Arbeitsaufwand, den das BlackBerry®-Gerät übernehmen muss, wenn Sie Ihre Anwendung installieren. Um Ihre Klassen vor der Installation Ihrer Anwendung auf einem BlackBerry-Gerät teilweise zu überprüfen, verwenden Sie das Tool zur Vorabüberprüfung aus dem Lieferumfang der BlackBerry® Java® Development Environment. Mit dem BlackBerry® Smartphone Simulator können Sie die COD-Dateien vorab überprüfen.

Weitere Informationen zur Verwendung des BlackBerry Smartphone Simulator finden Sie im *Benutzerhandbuch zum BlackBerry Device Simulator*. Weitere Informationen zur Verwendung des Tools zur Vorabüberprüfung finden Sie im *Entwicklungshandbuch für die BlackBerry Java Development Environment*.

Testen von Anwendungen im BlackBerry Smartphone Simulator

Sobald Sie Ihre Anwendung fertig entwickelt und kompiliert haben, können Sie sie auf dem BlackBerry®-Gerät testen. Der erste Schritt besteht meist darin, BlackBerry® Java® Development Environment so einzurichten, dass ein BlackBerry® Smartphone Simulator verwendet wird. Auf dem BlackBerry Smartphone Simulator wird derselbe Java-Code wie auf den BlackBerry-Geräten ausgeführt, weshalb der BlackBerry Smartphone Simulator eine exakte Testumgebung bietet, in der die Funktionsweise der

Anwendungen auf einem BlackBerry-Gerät ermittelt werden kann. Die BlackBerry JDE enthält aktuelle Versionen des BlackBerry Smartphone Simulator. Um zusätzliche Versionen des BlackBerry Smartphone Simulator herunterzuladen, öffnen Sie www.blackberry.com/developers/index.shtml.

Testen von Anwendungen auf einem BlackBerry-Gerät

Sobald Sie Ihre Anwendung im BlackBerry® Smartphone Simulator getestet haben, können Sie Ihre Anwendung auf einem BlackBerry-Gerät installieren. Falls Ihre Anwendung signierte APIs verwendet, benötigen Sie ggf. Codesignierungsschlüssel. Nach erfolgter Installation der Anwendung auf dem BlackBerry-Gerät können Sie die Anwendung öffnen und ihre Funktionalität und Leistung testen.

Zur Fehlerbehebung können Sie Ihr Gerät mit BlackBerry® Integrated Development Environment verbinden und mithilfe des Fehlerbehebungstools schrittweise durch die Anwendung gehen. Die BlackBerry IDE ist von Nutzen, wenn Sie versuchen, ein Netzwerk- oder Bluetooth®-Problem festzustellen oder auch andere Probleme, die nur schwierig simuliert werden können.

BlackBerry-Simulatoren

BlackBerry Smartphone Simulator

Mit dem BlackBerry® Smartphone Simulator simulieren Sie die Verwendung eines BlackBerry-Geräts, ohne tatsächlich ein BlackBerry-Gerät in Händen zu halten. Der BlackBerry Smartphone Simulator ist eine Anwendung, die Sie auf Ihrem Computer installieren, die eine Abbildung des BlackBerry-Gerätemodells Ihrer Wahl anzeigt. Diese Abbildung ist dem Aussehen und der Handhabung des echten BlackBerry-Geräts nachempfunden. Die Funktionsweise ist identisch mit einem echten BlackBerry-Gerät (mit Trackwheel oder Trackball und Tastatur) und auch dieselben Anwendungen und Funktionen stehen bereit, z. B. E-Mail-Nachrichten, Telefon und Internet-Browsing.

Der BlackBerry Smartphone Simulator dient auch als Plattform zum Ausführen von Anwendungen. Die Anwendungen können hierbei Netzwerkverbindungen herstellen, Daten speichern und E-Mail-Nachrichten verarbeiten. Der BlackBerry Smartphone Simulator enthält die BlackBerry-Geräteanwendungen, die in der Regel auf den meisten BlackBerry-Geräten verfügbar sind, und Sie können Ihre eigenen Anwendungen installieren und testen. Mit dem BlackBerry Smartphone Simulator können Sie verschiedene Konnektivitäts- und Zustandsänderungen simulieren und testen.

Wenn Sie die Tests im BlackBerry Smartphone Simulator durchführen, müssen Sie ggf. weitere BlackBerry-Dienste simulieren. Hierzu stehen der BlackBerry® MDS Simulator und der BlackBerry® Email Simulator bereit.

Um den BlackBerry Smartphone Simulator abzurufen, öffnen Sie www.blackberry.com/developers und laden Sie BlackBerry® Java® Development Environment oder BlackBerry Java Development Environment Component Package herunter.

BlackBerry MDS Simulator

Der BlackBerry® MDS Simulator dient zum Simulieren der BlackBerry MDS Connection Service-Komponente auf dem BlackBerry® Enterprise Server. Wenn Sie den BlackBerry Smartphone Simulator gemeinsam mit dem BlackBerry MDS Simulator einsetzen, können Sie Netz-, Push-HTTP- und Browser-Anwendungen testen, die für den Einsatz mit einem BlackBerry Enterprise Server entwickelt wurden. Um den BlackBerry MDS Simulator abzurufen, öffnen Sie www.blackberry.com/developers und laden Sie das BlackBerry® Email and MDS Services Simulator Package herunter.

BlackBerry Email Simulator

Mit dem BlackBerry® Email Simulator werden Nachrichten zwischen dem BlackBerry® Smartphone Simulator und entweder einer Messaging-Anwendung, z. B. Microsoft® Outlook®, oder einem POP3- oder SMTP-Server gesendet und empfangen. Ein BlackBerry® Enterprise Server ist hierzu nicht nötig. Um den BlackBerry Email Simulator abzurufen, öffnen Sie www.blackberry.com/developers und laden Sie das BlackBerry® Email and MDS Services Simulator Package herunter.

Verfügbarmachen von Anwendungen

10

Anwendungsverteilung über eine Computerverbindung

Verteilen einer Anwendung von einem Computer aus

Mit dem Programm zum Laden von Anwendungen im BlackBerry® Desktop Manager können Sie Anwendungen auf dem BlackBerry-Gerät installieren. Das Programm zum Laden von Anwendungen bietet den Benutzern eine einfache Möglichkeit, Anwendungen vom Computern auf das BlackBerry-Gerät herunterzuladen.

Verteilen einer Anwendung von einer Webseite aus

Mit dem BlackBerry® Application Web Loader können Sie Ihre kompilierte Anwendung auf einer Website bereitstellen. Die Benutzer können die Webseite über Windows® Internet Explorer® auf dem Computer aufrufen und die Anwendung auf ihrem BlackBerry-Gerät installieren. Wenn BlackBerry-Gerätebenutzer die Webseite besuchen, fordert der BlackBerry Application Web Loader sie auf, das Gerät mit dem USB-Anschluss zu verbinden. Anschließend können sie die Anwendung mithilfe eines ActiveX®-Steuerelements installieren. Der BlackBerry Application Web Loader bietet BlackBerry-Gerätebenutzern eine einfache Möglichkeit, Anwendungen vom Computer aus zu installieren, ohne den BlackBerry® Desktop Manager ausführen zu müssen.

Verteilen einer Anwendung zu Testzwecken

Die BlackBerry® Java® Development Environment umfasst das Befehlszeilen-Tool mit der Bezeichnung "JavaLoader-Tool", das sich im BIN-Ordner im Verzeichnis von BlackBerry JDE befindet. Mit dem JavaLoader-Tool können Sie kompilierte Anwendungsdateien direkt über den USB-Anschluss schnell auf dem BlackBerry-Gerät installieren und wieder vom Gerät entfernen. Deskriptorendateien oder Webseiten werden dabei nicht benötigt. Das JavaLoader-Tool ist von Nutzen, wenn Sie Ihre Anwendung während der Test- und Entwicklungsphase häufig installieren und wieder entfernen; das JavaLoader-Tool ist allerdings nicht für die Verwendung durch BlackBerry-Gerätebenutzer vorgesehen.

Anwendungsverteilung über das drahtlose Netzwerk

Wenn Sie Ihre Anwendungen über das drahtlose Netzwerk verteilen, bieten Sie damit BlackBerry®-Gerätebenutzern einen höheren Komfort. Gleichzeitig können Sie die Anwendungsverteilung an eine große Personengruppe vereinfachen, weil keine Computeranwendung erforderlich ist. Ein BlackBerry-Gerätebenutzer kann Ihre Anwendungen über das drahtlose Netzwerk installieren.

Drahtloses Pull-Verfahren (vom Benutzer eingeleitet)

Sie können kompilierte Anwendungen auf einer öffentlichen oder privaten Website bereitstellen. BlackBerry®-Gerätebenutzer können die Website aufrufen, um die Anwendungen über das drahtlose Netzwerk mithilfe des Browsers auf ihr BlackBerry-Gerät herunterzuladen. Die Benutzer werden über den Browser aufgefordert, die Anwendung zu installieren. Anschließend wird die Anwendung heruntergeladen und auf dem BlackBerry-Gerät installiert.

Drahtloses Push-Verfahren (vom Server eingeleitet)

In der BlackBerry® Enterprise Server-Umgebung sind die Administratoren in der Lage, Anwendungen zur erforderlichen Installation per Push-Verfahren über das drahtlose Netzwerk an die BlackBerry-Gerätebenutzer zu übermitteln. Der Administrator erstellt eine neue Richtlinie und gibt an, dass die Anwendung für das BlackBerry-Gerät erforderlich ist. Die Anwendung wird per Push-Verfahren an die Benutzer übermittelt, ohne dass diese selbst aktiv werden müssten. In einem Unternehmen ist diese Vorgehensweise von großem Nutzen, wenn neue Anwendungen an eine große Gruppe von BlackBerry-Gerätebenutzern gesendet werden sollen.

Glossar

11

AES

Advanced Encryption Standard (Erweiterter Verschlüsselungsstandard)

API

Application Programming Interface (wendungsprogrammierschnittstelle)

CLDC

Connected Limited Device Configuration (Bibliothek für Gerätekonfigurationen)

HTTP

Hypertext Transfer Protocol (Netzwerkprotokoll)

HTTPS

Hypertext Transfer Protocol over Secure Sockets Layer (HTTP über SSL)

E/A

Eingabe/Ausgabe

IMAP

Internet Message Access Protocol

IPC

Interprocess Communication (Kommunikation zwischen Prozessen)

JDWP

Java® Debug Wire Protocol

JSR

Java® Specification Request (Java-Spezifikationsanforderung)

JTWI

Java® Technology for the Wireless Industry (Java-Technologie für die drahtlose Industrie)

JVM

Java® Virtual Machine

MIDP

Mobile Information Device Profile (Softwareprofil für mobile Geräte)

NVRAM

Non-Volatile Random Access Memory (Nichtflüchtiger Arbeitsspeicher)

PDAP

PDA Optional Packages for the J2ME Platform (optionale PDA-Pakete für die J2ME Platform)

PIM

Personal Information Management

PIN

Personal Identification Number (persönliche Identifikationsnummer)

POP

Post Office Protocol

RAPC

RIM Application Program Compiler (Compiler für RIM-Anwendungsprogramm)

RMS

Record Management System

SHA

Secure Hash Algorithm (sicherer Hashalgorithmus)

SMS

Short Message Service

SMTP

Simple Mail Transfer Protocol

SRAM

static random access memory (Statischer Arbeitsspeicher)

SSL

Secure Sockets Layer

TCP

Transmission Control Protocol

TLS

Transport Layer Security (Netzwerkprotokoll zur sicheren Datenübertragung)

Triple DES

Triple Data Encryption Standard (Verschlüsselungsstandard)

VPN

Virtuelles privates Netzwerk

WAP

Wireless Application Protocol (Protokoll zur drahtlosen Datenübertragung)

XML

Extensible Markup Language

Rechtliche Hinweise

12

©2008 Research In Motion Limited. Alle Rechte vorbehalten. BlackBerry®, RIM®, Research In Motion®, SureType® und die zugehörigen Marken, Namen und Logos sind Eigentum von Research In Motion Limited und sind in den USA und weiteren Ländern weltweit als Marken eingetragen und/oder werden dort als Marken verwendet.

Bluetooth ist eine Marke von Bluetooth SIG. Java und Javadoc sind Marken von Sun Microsystems, Inc. Microsoft, Windows Internet Explorer, ActiveX und Windows sind Marken der Microsoft Corporation. HotSync ist eine Marke von Palmsource, Inc. Alle weiteren Marken sind Eigentum ihrer jeweiligen Inhaber.

Das BlackBerry-Smartphone und andere Geräte und/oder die zugehörige Software sind durch Urheberrecht, internationale Verträge und verschiedene Patente geschützt, einschließlich einem oder mehreren der folgenden US-Patente: 6,278,442; 6,271,605; 6,219,694; 6,075,470; 6,073,318; D445,428; D433,460; D416,256. Weitere Patente sind angemeldet oder stehen zur Anmeldung in den USA Anmeldung in verschiedenen Ländern an. Eine aktuelle Liste der RIM-Patente (wie folgt beschrieben) finden Sie unter www.rim.com/patents.

Dieses Dokument und alle Dokumente, die per Verweis in dieses Dokument mit einbezogen werden, z. B. alle unter www.blackberry.com/go/docs erhältlichen Dokumente werden ohne Mängelgewähr und je nach Verfügbarkeit bereitgestellt. Die entsprechenden Dokumente werden ohne ausdrückliche Billigung, Gewährleistung oder Garantie seitens Research In Motion Limited und seinen angegliederten Unternehmen ("RIM") bereitgestellt. RIM übernimmt keine Verantwortung für eventuelle typografische, technische oder anderweitige Ungenauigkeiten sowie für Fehler und Auslassungen in den genannten Dokumenten. Dieses Dokument beschreibt eventuell einige Aspekte der RIM-Technologie in allgemeiner Form, um das Eigentum und die vertraulichen Informationen und/oder Geschäftsgeheimnisse von RIM zu schützen. RIM behält sich das Recht vor, die in diesem Dokument enthaltenen Informationen von Zeit zu Zeit zu ändern. RIM ist jedoch nicht verpflichtet, die Benutzer über diese Änderungen, Aktualisierungen, Verbesserungen oder Zusätze rechtzeitig bzw. überhaupt in Kenntnis zu setzen.

Diese Dokumentation enthält möglicherweise Verweise auf Informationsquellen, Hardware oder Software, Produkte oder Dienste, einschließlich Komponenten und Inhalte wie urheberrechtlich geschützte Inhalte und/oder Websites von Drittanbietern (nachfolgend "Drittprodukte und -dienste" genannt). RIM hat keinen Einfluss auf und übernimmt keine Haftung für Drittprodukte und -dienste, dies gilt u. a. für Inhalt, Genauigkeit, Einhaltung der Urheberrechtsgesetze, Kompatibilität, Leistung, Zuverlässigkeit, Rechtmäßigkeit, Schicklichkeit, Links oder andere Aspekte der Drittprodukte und -dienste. Der Einschluss eines Verweises auf Drittprodukte und -dienste in dieser Dokumentation impliziert in keiner Weise eine besondere Empfehlung der Drittprodukte und -dienste oder des Drittanbieters durch RIM.

SOFERN ES NICHT DURCH DAS IN IHREM RECHTSGEBIET GELTENDE RECHT AUSDRÜCKLICH UNTERSAGT IST, WERDEN HIERMIT SÄMTLICHE AUSDRÜCKLICHEN ODER KONKLUDENTEN BEDINGUNGEN, BILLIGUNGEN, GARANTIEEN, ZUSICHERUNGEN ODER GEWÄHRLEISTUNGEN JEDER ART, EINSCHLIESSLICH, OHNE EINSCHRÄNKUNG, BEDINGUNGEN, BILLIGUNGEN, GARANTIEEN, ZUSICHERUNGEN ODER GEWÄHRLEISTUNGEN HINSICHTLICH DER HALTBARKEIT, EIGNUNG FÜR EINEN BESTIMMTEN ZWECK ODER VERWENDUNGSZWECK, MARKTGÄNGIGKEIT, MARKTGÄNGIGEN QUALITÄT, NICHTVERLETZUNG VON RECHTEN DRITTER, ZUFRIEDENSTELLENDE QUALITÄT ODER DES EIGENTUMSRECHTS ABGELEHNT. DIES GILT AUCH FÜR ZUSICHERUNGEN ODER GEWÄHRLEISTUNGEN, DIE SICH AUS EINEM GESETZ, EINER GEPFLOGENHEIT, USANCEN BZW. HANDELSGEPFLOGENHEITEN ERGEBEN ODER IM ZUSAMMENHANG MIT DER DOKUMENTATION ODER IHRER VERWENDUNG, DER LEISTUNG ODER MANGELNDE LEISTUNG VON SOFTWARE, HARDWARE, DIENSTEN ODER DRITTANBIETER-PRODUKTEN UND -DIENSTEN STEHEN, AUF DIE HIER VERWIESEN WIRD.

MÖGLICHERWEISE HABEN SIE ZUDEM ANDERE LANDESSPEZIFISCHE RECHTE. IN MANCHEN RECHTSGEBIETEN IST DER AUSSCHLUSS ODER DIE EINSCHRÄNKUNG KONKLUDENTER GEWÄHRLEISTUNGEN UND BEDINGUNGEN NICHT ZULÄSSIG. IN DEM GESETZLICH ZULÄSSIGEN UMFANG WERDEN SÄMTLICHE KONKLUDENTEN GEWÄHRLEISTUNGEN ODER BEDINGUNGEN IM ZUSAMMENHANG MIT DER DOKUMENTATION, DIE EINGESCHRÄNKT WERDEN KÖNNEN, SOFERN SIE NICHT WIE OBEN DARGELEGT AUSGESCHLOSSEN WERDEN KÖNNEN, HIERMIT AUF 90 TAGE AB DATUM DES ERWERBS DER DOKUMENTATION ODER DES ARTIKELS, AUF DEN SICH DIE FORDERUNG BEZIEHT, BESCHRÄNKT.

IN DEM DURCH DAS IN IHREM RECHTSGEBIET ANWENDBARE GESETZ MAXIMAL ZULÄSSIGEN AUSMASS HAFTET RIM UNTER KEINEN UMSTÄNDEN FÜR SCHÄDEN JEDLICHER ART, DIE IM ZUSAMMENHANG MIT DIESER DOKUMENTATION ODER IHRER VERWENDUNG, DER LEISTUNG ODER NICHTLEISTUNG JEDLICHER SOFTWARE, HARDWARE, DIENSTE ODER DRITTPRODUKTE UND -DIENSTE, AUF DIE HIER BEZUG GENOMMEN WIRD, STEHEN, EINSCHLIESSLICH, ABER NICHT BESCHRÄNKT AUF DIE FOLGENDEN SCHÄDEN: DIREKTE, VERSCHÄRFTE SCHADENERSATZ NACH SICH ZIEHENDE, BEILÄUFIG ENTSTANDENE, INDIREKTE, KONKRETE, STRAFE EINSCHLIESSENDE SCHÄDEN, FOLGESCHÄDEN ODER SCHÄDEN, FÜR DIE ANSPRUCH AUF KOMPENSATORISCHEN SCHADENERSATZ BESTEHT, SCHÄDEN WEGEN ENTGANGENEN GEWINNEN ODER EINKOMMEN, NICHTREALISIERUNG ERWARTETER EINSPARUNGEN, BETRIEBSUNTERBRECHUNGEN, VERLUSTES GESCHÄFTLICHER DATEN, ENTGANGENER GESCHÄFTSCHANCEN ODER BESCHÄDIGUNG BZW. VERLUSTES VON DATEN, DES UNVERMÖGENS, DATEN ZU ÜBERTRAGEN ODER ZU EMPFANGEN, PROBLEME IM ZUSAMMENHANG MIT ANWENDUNGEN, DIE IN VERBINDUNG MIT RIM-PRODUKTEN UND -DIENSTEN VERWENDET WERDEN, KOSTEN VON AUSFALLZEITEN, NICHTVERWENDBARKEIT VON RIM-PRODUKTEN UND -DIENSTEN ODER TEILEN DAVON BZW. VON AIRTIME-DIENSTEN, KOSTEN VON ERSATZGÜTERN, DECKUNG, EINRICHTUNGEN ODER DIENSTEN, KAPITAL- ODER ANDERE VERMÖGENSSCHÄDEN, UNABHÄNGIG DAVON, OB SCHÄDEN DIESER ART ABZUSEHEN ODER NICHT ABZUSEHEN WÄREN, UND AUCH DANN, WENN RIM AUF DIE MÖGLICHKEIT SOLCHER SCHÄDEN HINGEWIESEN WURDE.

IN DEM DURCH DAS IN IHREM RECHTSGEBIET ANWENDBARE GESETZ MAXIMAL ZULÄSSIGEN AUSMASS ÜBERNIMMT RIM KEINERLEI VERANTWORTUNG, VERPFLICHTUNG ODER HAFTUNG, SEI SIE VERTRAGLICHER, DELIKTRECHTLICHER ODER ANDERWEITIGER NATUR, EINSCHLIESSLICH DER HAFTUNG FÜR FAHRLÄSSIGKEIT UND DER DELIKTSHAFTUNG.

DIE IN DIESEM DOKUMENT GENANNTEN EINSCHRÄNKUNGEN, AUSSCHLÜSSE UND HAFTUNGSAUSSCHLÜSSE GELTEN (A) UNGEACHTET DER VON IHNEN ANGEFÜHRTEN KLAGEGRÜNDE, FORDERUNGEN ODER KLAGEN, EINSCHLIESSLICH, ABER NICHT BESCHRÄNKT AUF VERTRAGSBRUCH, FAHRLÄSSIGKEIT, ZIVILRECHTLICHER DELIKTE, DELIKTSHAFTUNG ODER SONSTIGE RECHTSTHEORIE UND SIND AUCH NACH EINEM WESENTLICHEN VERSTOSS BZW. EINEM FEHLENDEN GRUNDLEGENDEN ZWECK DIESER VEREINBARUNG ODER EINES DARIN ENTHALTENEN RECHTSBEHELFS WIRKSAM; UND GELTEN (B) FÜR RIM UND DIE ZUGEHÖRIGEN UNTERNEHMEN, RECHTSNACHFOLGER, BEVOLLMÄCHTIGTEN, VERTRETER, LIEFERANTEN (EINSCHLIESSLICH AIRTIME-DIENSTANBIETERN), AUTORISIERTE RIM-DISTRIBUTOREN (EBENFALLS EINSCHLIESSLICH AIRTIME-DIENSTANBIETERN) UND DIE JEWEILIGEN FÜHRUNGSKRÄFTE, ANGESTELLTEN UND UNABHÄNGIGEN AUFTRAGNEHMER.

ZUSÄTZLICH ZU DEN OBEN GENANNTEN EINSCHRÄNKUNGEN UND AUSSCHLÜSSEN HAFTEN DIE FÜHRUNGSKRÄFTE, ANGESTELLTEN, VERTRETER, DISTRIBUTOREN, LIEFERANTEN, UNABHÄNGIGEN AUFTRAGNEHMER VON RIM ODER RIM ANGEHÖRENDE UNTERNEHMEN IN KEINER WEISE IM ZUSAMMENHANG MIT DER DOKUMENTATION.

Bevor Sie Drittprodukte bzw. -dienste abonnieren, installieren oder verwenden, müssen Sie sicherstellen, dass Ihr Airtime-Dienstanbieter sich mit der Unterstützung aller zugehörigen Funktionen einverstanden erklärt hat. Einige Airtime-Dienstanbieter bieten möglicherweise keine Internet-Browsing-Funktion in Zusammenhang mit einem Abonnement für den BlackBerry® Internet Service an. Erkundigen Sie sich bei Ihrem Service Provider bezüglich Verfügbarkeit, Roaming-Vereinbarungen, Service-Plänen und Funktionen. Für die Installation und Verwendung von Drittprodukten und -diensten mit den Produkten und Diensten

von RIM sind möglicherweise Patent-, Marken-, Urheberrechts- oder andere Lizenzen erforderlich, damit die Rechte Dritter nicht verletzt werden. Es liegt in Ihrer Verantwortung, zu entscheiden, ob Sie Drittprodukte und -dienste verwenden möchten, und festzustellen, ob hierfür Lizenzen erforderlich sind. Für den Erwerb etwaiger Lizenzen sind Sie verantwortlich. Installieren oder verwenden Sie Drittprodukte und -dienste erst nach dem Erwerb aller erforderlichen Lizenzen. Alle Drittprodukte und -dienste, die Sie mit Produkten und Diensten von RIM erhalten, werden lediglich zu Ihrem Vorteil, ohne Mängelgewähr und ohne ausdrückliche oder konkludente Bedingungen, Billigungen, Garantien, Zusicherungen oder Gewährleistungen jeglicher Art von RIM bereitgestellt, und RIM übernimmt in diesem Zusammenhang keine Haftung. Die Verwendung von Drittprodukten und -diensten unterliegt Ihrer Zustimmung zu den Bedingungen separater Lizenzen und anderer geltender Vereinbarungen mit Dritten, sofern sie nicht ausdrücklich von einer Lizenz oder anderen Vereinbarung mit RIM behandelt wird.

Bestimmte in diesem Dokument angeführte Funktionen erfordern eine Mindestversion von BlackBerry® Enterprise Server, BlackBerry® Desktop Software und/oder BlackBerry® Device Software.

Die Nutzungsbedingungen für RIM-Produkte und -Dienste werden in einer entsprechenden separaten Lizenz oder anderen Vereinbarung mit RIM dargelegt. KEINE DER IN DIESER DOKUMENTATION DARGELEGTEN BESTIMMUNGEN SETZEN IRGENDWELCHE AUSDRÜCKLICHEN SCHRIFTLICHEN VEREINBARUNGEN ODER GEWÄHRLEISTUNGEN VON RIM FÜR TEILE VON RIM-PRODUKTEN ODER -DIENSTEN AUSSER KRAFT.

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

Veröffentlicht in Kanada