

Programmieren für den **BlackBerry**

- > So entwickeln Sie Java-Apps für den BlackBerry
- > Die BlackBerry-API verstehen und effektiv nutzen
- > Administrator-Guide: Apps und Endgeräte mit dem BES verwalten

Klaus Löwenhagen

Programmieren für den BlackBerry

Klaus Löwenhagen

Programmieren für den **BlackBerry**

Mit 34 Abbildungen

Bibliografische Information der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

Alle Angaben in diesem Buch wurden vom Autor mit größter Sorgfalt erarbeitet bzw. zusammengestellt und unter Einschaltung wirksamer Kontrollmaßnahmen reproduziert. Trotzdem sind Fehler nicht ganz auszuschließen. Der Verlag und der Autor sehen sich deshalb gezwungen, darauf hinzuweisen, dass sie weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernehmen können. Für die Mitteilung etwaiger Fehler sind Verlag und Autor jederzeit dankbar. Internetadressen oder Versionsnummern stellen den bei Redaktionsschluss verfügbaren Informationsstand dar. Verlag und Autor übernehmen keinerlei Verantwortung oder Haftung für Veränderungen, die sich aus nicht von ihnen zu vertretenden Umständen ergeben. Evtl. beigefügte oder zum Download angebotene Dateien und Informationen dienen ausschließlich der nicht gewerblichen Nutzung. Eine gewerbliche Nutzung ist nur mit Zustimmung des Lizenzinhabers möglich.

© 2010 Franzis Verlag GmbH, 85586 Poing

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Das Erstellen und Verbreiten von Kopien auf Papier, auf Datenträgern oder im Internet, insbesondere als PDF, ist nur mit ausdrücklicher Genehmigung des Verlags gestattet und wird widrigenfalls strafrechtlich verfolgt.

Die meisten Produktbezeichnungen von Hard- und Software sowie Firmennamen und Firmenlogos, die in diesem Werk genannt werden, sind in der Regel gleichzeitig auch eingetragene Warenzeichen und sollten als solche betrachtet werden. Der Verlag folgt bei den Produktbezeichnungen im Wesentlichen den Schreibweisen der Hersteller.

Herausgeber: Graser

Satz: DTP-Satz A. Kugge, München

art & design: www.ideehoch2.de

Druck: Bercker, 47623 Kevelaer

Printed in Germany

ISBN 978-3-645-60018-7

Danksagung

Ich möchte mich bei den vielen Menschen bedanken, die mir, wenn auch nicht immer bewusst, Ideen für dieses Werk gegeben haben:

Bei Michaela für die endlose Geduld, die sie im Laufe dieses Projekts für mich aufgebracht hat.

Bei meinen Kindern Nele und Merle, die mich auf ihre Weise inspiriert haben.

Einen besonderen Dank möchte ich Raphael Thierjung aussprechen, der mit seinen Tricky Apps einen erheblichen Beitrag geleistet hat.

Nicht zuletzt geht ein ganz großes Dankeschön an die Firma ASPERI, die mir den Messaging Application Server (MAS) zur Verfügung gestellt hat, mit dem sich, wie ich finde, ganz tolle Dinge realisieren lassen.

Zu guter Letzt gilt mein Dank der Firma O3SIS für die unkomplizierte SyncML-Einrichtung.

Inhaltsverzeichnis

1	Einführung.....	15
1.1	Voraussetzungen	16
1.2	Die Grenzen dieses Buches.....	17
1.3	Einige Begriffe vorweg.....	18
1.4	Typografie.....	19
2	Installation und Einrichtung der Entwicklungsumgebung Eclipse	23
2.1	Systemanforderungen.....	23
2.2	Installationsschritte	23
2.3	Erstes Projekt erstellen	25
2.4	Einstellungen in der Entwicklungsumgebung	26
2.5	Checkliste zur Erzeugung eines neuen Projekts	29
2.6	Das erste Codebeispiel schreiben	30
2.7	Beschreibung des Quelltextes.....	31
2.8	Applikations-Icons	34
3	Den BlackBerry programmieren	35
3.1	Das Programm wird im Simulator gestartet.....	35
3.2	Das BlackBerry-System	36
3.3	Bewährte Verfahren für das Schreiben einer effizienten BlackBerry Java Application.....	46
3.4	Unterstützung mehrerer Sprachen.....	54
3.5	Multimedia-Unterstützung: Audiounterstützung.....	54
3.6	Kameraunterstützung.....	55
3.7	Benutzeroberflächen- und Navigationsdesign	57
3.8	Erstellen einer UI, die mit normalen Benutzeroberflächen kompatibel ist.....	59
3.9	Speicherverwaltung	60
3.10	Minimieren der Speicherbelegung.....	61
3.11	Schonen der Ressourcen.....	61
3.12	Garbage Collection auf einem BlackBerry-Gerät	63
3.13	Datenverwaltung.....	64
3.14	Dateiverbindungs-APIs.....	65
3.15	Sichern und Synchronisieren von Daten	66

4	Beispielcodes I	67
4.1	Wissenswertes über Felder.....	67
4.2	User Interface.....	69
4.3	Auswahlfelder.....	70
4.4	Checkbox.....	73
4.5	DayCalc.....	79
4.6	CustomButtons	83
4.7	TabControl	95
5	Beispielcodes II	99
5.1	Screen-Dekor	99
5.2	Nullfelder.....	100
5.3	Dialogboxen.....	100
5.4	Dateiauswahldialog	103
5.5	Splash-Screen.....	111
5.6	DialogClosedListener	115
5.7	Applikations-Switch	117
5.8	Applikationen durch Passwort schützen.....	119
5.9	Backdoor-Programme	121
5.10	Multisample.....	122
5.11	Multiplikationstabelle.....	139
5.12	Primzahlen berechnen	143
5.13	Mail-Management.....	153
6	Drahtloser Datentransport	165
6.1	Drahtlose Gateways	165
6.2	Alternative Optionen für den Datentransport.....	166
6.3	Integrieren von BlackBerry-Anwendungen	167
6.4	Listener für Reaktionen auf Anwendungsänderungen	168
6.5	Sicherheitsbedingungen	169
6.6	Authentifizierung	170
6.7	BlackBerry-APIs mit gesteuertem Zugriff.....	171
6.8	Unterstützung für IT-Richtlinien	172
6.9	Anwendungssteuerung	172
6.10	Dateiverschlüsselung bei MicroSD-Karten	173
6.11	Verwenden der MicroSD-Medienkarte auf mehreren BlackBerry-Geräten	173
6.12	IT-Richtlinien und die MicroSD-Medienkarte.....	174
6.13	Testvorbereitung einer BlackBerry-Applikation.....	174
6.14	Vorabprüfung einer Applikation	174
6.15	Testen von Anwendungen auf dem Gerät	175

6.16	Simulatoren	175
6.17	Verfügbarmachen von Anwendungen	176
7	UI-Guidelines	179
7.1	Richtlinien für Icons	179
7.2	Designhinweise für BlackBerry-Geräte.....	179
8	Die BlackBerry-API	181
8.1	Beschreibung der JDE.....	181
8.2	BlackBerry-API-Package	182
8.3	CLDC-APIs	184
8.4	MIDP-APIs	184
8.5	PDAP-API.....	185
8.6	Java-Code in BlackBerry-Geräten	185
8.7	Restriktionen	186
8.8	Multithreading	186
9	Admin-Guide für den BlackBerry Enterprise Server (BES).....	187
9.1	Verknüpfen von Unternehmensrollen mit BlackBerry- Rollen	188
9.2	Verwalten von Verwaltungsrollen	189
9.3	Hinzufügen von Datenbankbenutzern zu Verwaltungsrollen	190
9.4	Festlegen der Authentifizierung des BlackBerry Managers beim Datenbankserver.....	191
9.5	Anpassen von Verwaltungsrollen	191
9.6	Einrichten der BlackBerry-Umgebung	192
10	Benutzerkonten auf dem BlackBerry Enterprise Server einrichten	205
10.1	Benutzerkonten	205
10.2	Anpassen der Synchronisierung von Terminplanerdaten	207
10.3	Steuern der BlackBerry-Umgebung.....	209
11	Zusätzliche Gerätesoftware und Anwendungen für Benutzer bereitstellen	221
11.1	Hinzufügen von Software zu einem Netzwerklaufwerk.....	222
11.2	Erstellen von Softwarekonfigurationen	225
11.3	Definieren einer Richtlinie zur Anwendungssteuerung.....	226
11.4	Senden von Anwendungen über das drahtlose Netzwerk an BlackBerry-Geräte	228

12	Einrichten von BlackBerry-Geräten.....	229
12.1	Laden von Benutzernachrichten auf BlackBerry-Geräte	229
12.2	Option 1: Geräte mit dem BlackBerry Manager einrichten	231
12.3	Option 2: Geräten über das drahtlose Netzwerk einrichten	231
12.4	Option 3: Geräte mit dem BlackBerry Desktop Manager einrichten	235
12.5	Schützen verloren gegangener oder gestohlener BlackBerry-Geräte	236
12.6	Ausgeben vorhandener BlackBerry-Geräte an neue Benutzer	237
12.7	Bereitstellen von BlackBerry-MDS-Studio-Anwendungen für Benutzer	239
13	Anpassen der BlackBerry-Nachrichtenübermittlung	255
13.1	Verwalten der Nachrichtenumleitung	255
13.2	Verwalten von Umleitungsfiltern	256
14	Verwalten der drahtlosen Nachrichtensynchronisierung	263
14.1	Deaktivieren der drahtlosen Nachrichtensynchronisierung.....	263
14.2	Synchronisieren endgültig gelöschter Nachrichten.....	264
15	Sicherer Nachrichtentransfer	265
15.1	Signaturen und Haftungsausschlüsse in Nachrichten verwenden	267
15.2	Überwachen von Nachrichten, die Benutzer von ihren BlackBerry-Geräten senden.....	271
15.3	Verwalten der Nachrichtenwarteschlange	272
16	Terminplanerdaten drahtlos sichern und wiederherstellen	273
16.1	Löschen der Terminplanerdaten eines Benutzers vom Enterprise Server.....	273
16.2	Deaktivieren der drahtlosen Sicherung	274

17	Adressbuchfelder synchronisieren	275
17.1	Feldzuordnung	275
17.2	Senden von Nachrichten an Benutzer	277
18	Instant Messaging	279
18.1	Konfigurieren der Verbindung zum Instant-Messaging- Server	279
18.2	Steuern einer Instant-Messaging-Sitzung	280
19	Anpassen des Supports für Nachrichtenanhänge	283
19.1	Konfigurieren der Verbindungsherstellung zwischen BlackBerry Enterprise Server und BlackBerry Attachment Service	283
19.2	Herstellen einer Verbindung zwischen Enterprise Server und Attachment Service	283
19.3	Herstellen einer Verbindung zwischen Attachment Service und Enterprise Server	284
19.4	Steuern der Konvertierung von Anlagen durch den Attachment Service	285
19.5	Anpassen der Konvertierung von Anlagen durch den BlackBerry Attachment Service	286
19.6	Konfigurieren der Unterstützung für Anlagendateiformate	287
19.7	Entfernen der Unterstützung für ein Anlagendateiformat	287
19.8	Hinzufügen von Unterstützung für zusätzliche Erweiterungen von Anlagendateiformaten	288
19.9	Steuern der Größe von Anlagendateien zum Minimieren der erforderlichen Konvertierungsressourcen	288
19.10	Festlegen der maximalen Dateigröße einer Anlage	289
19.11	Festlegen der Höchstmaße für Bilder	289
19.12	Steuern der Größe von Anlagendateien zum Minimieren der erforderlichen Übertragungsressourcen	290
19.13	Vermeiden, dass Anlagen geladen werden	291
20	Anpassen des drahtlosen Zugriffs auf Unternehmensanwendungen	293
20.1	Zentrale Push-Server	293
20.2	Festlegen des zentralen Push-Servers	293
20.3	Anpassen der Authentifizierung der BlackBerry-Geräte beim Webserver	294
20.4	Konfigurieren der Authentifizierung des MDS Connection Service bei Servern, die NTLM verwenden	295

20.5	Konfigurieren der Authentifizierung des MDS Connection Service bei Servern, die Kerberos verwenden.....	295
20.6	Konfigurieren der Authentifizierung des MDS Connection Service bei Servern, die LTPA verwenden	295
20.7	Konfigurieren der Authentifizierung des MDS Connection Service beim RSA Authentication Manager.....	296
20.8	Beschränken des Benutzerzugriffs auf Webinhalte.....	296
20.9	Zulassen vertrauenswürdiger Verbindungen zwischen Push-Anwendungen und dem MDS Connection Service.....	302
20.10	Veröffentlichen des MDS-Connection-Service-Zertifikats für vertrauenswürdige Verbindungen zwischen Push-Anwendungen und dem MDS Connection Service	303
20.11	Anpassen der Herstellung vertrauenswürdiger Verbindungen zu Webservern durch Anwendungen.....	304
20.12	Zulassen von Verbindungen von Endgeräten zu vertrauenswürdigen Webservern	306
20.13	Zulassen einer SSL-Verbindung vom MDS Connection Service zu einer Push-Anwendung.....	307
20.14	Verwalten von Push-Anwendungsanfragen.....	311
21	Verwalten von Benutzerkonten	317
21.1	Verwalten von Benutzergruppen	317
21.2	Verwalten von Benutzern	318
22	Verwalten von BlackBerry-Gerätesoftware und drahtlosen Anwendungen.....	321
22.1	Verwalten von Anwendungen auf BlackBerry-Geräten	321
22.2	Verwalten von Softwarekonfigurationen	323
23	Verwalten einer BlackBerry-Domain	325
23.1	Überwachen der BlackBerry-Dienste und -komponenten in einer Domain	325
23.2	Anpassen der Überwachung der Dienste durch den BlackBerry-Controller	326
23.3	Zugriff auf Protokolldateien für BlackBerry-Dienste	331
23.4	Verwalten anderer BlackBerry-Domains	338
23.5	Verwalten von Lizenzschlüsseln.....	338

Anhang	341
A.1 Rollenmatrix.....	341
B.1 Drahtloses Sichern und Wiederherstellen	352
C.1 Persistente Datenspeicherung	352
D.1 Package Class.....	357
E.1 Hash-Kalkulation	366
F.1 BlackBerry und MAS Message Applikation Server	366
G.1 Glossar	375
 Stichwortverzeichnis	 379

1 Einführung

Wie der Titel dieses Buches ankündigt, möchte ich Ihnen eine umfassende Einführung in die Java-Programmierung für den BlackBerry geben. Umfassend heißt, ich möchte mit Ihnen in die Tiefen des BlackBerry-Systems einsteigen und Ihnen alle Elemente und Möglichkeiten der BlackBerry-API vorstellen, damit Sie eigene Programme entwickeln und dieses Buch als Nachschlagereferenz heranziehen können.

Java ist eine Sprache, die mit sehr wenigen Elementen auskommt und eine Kontextunterstützung zur Generierung der Codes mitbringt. Das soll nicht unbedingt heißen, dass die hier vorgestellten Beispiele keine hohe Komplexität besäßen und die Sprache im Handumdrehen zu erlernen wäre. Aus persönlicher Erfahrung kann ich aber sagen, dass man sich recht schnell mit der Sprache zurechtfindet und rasch zu einer ersten Applikation kommt.

Tatsächlich werden Sie aber nicht weit kommen, wenn Sie ausschließlich die einfachen Sprachelemente der API verwenden. Denn neben diesen grundlegenden Elementen gibt es noch weitere Bibliotheken, die Sie tiefer in das Gerät eintauchen lassen.

Der BlackBerry-Hersteller Research In Motion (kurz RIM) bietet die Möglichkeit, eine Entwicklerlizenz zu erwerben, die Ihnen sehr viel mehr Möglichkeiten eröffnet, als nur einfache Java-Codes zu programmieren. Diese Lizenz wird Code-Signing genannt. In den Bibliotheken befinden sich zum Beispiel Funktionen, mit denen Sie Ihr Gerät abschalten oder eine Taste belegen können, mit der das Device gesperrt und nur über einen Code wieder freigegeben werden kann. Diese Möglichkeiten sind nicht im Standard-MDS (Mobile Data System) freigegeben und können erst durch das sogenannte Code-Signing erreicht werden.

Ohne diese Entwicklerfreigabe auf die internen Core-Bibliotheken können Sie zwar einfache Programme schreiben, jedoch ist es unmöglich, mit sogenannten persistenten Objekten zu arbeiten.

Wollte ich auch diese Bibliotheken umfassend darstellen, dann würde das den Rahmen dieses Buches sprengen. Einige Tausend Seiten kämen zustande, wenn alle bis dato veröffentlichten Funktionen dokumentiert würden. Was ich Ihnen in diesem Buch zeigen möchte, ist daher nur ein Ausschnitt aus dem Bibliotheksumfang des BlackBerry.

Wie Sie auch bemerken werden, ist der Funktionsumfang, den die Standardbibliotheken abdecken, nicht gerade groß. In diesem Buch werden wir schnell an diese Grenzen stoßen. Einer dieser Fälle ist das Abschalten des Gerätes aus dem Programm heraus oder der Zugriff auf das Adressbuch.

Ebenso wird der Zugriff auf die Gerätedatenbank ohne Code-Signing nicht unterstützt, allerdings lassen sich Programme, die darauf zugreifen, im Gerätesimulator ohne Probleme aufrufen und testen.

Daher gehe ich an entsprechenden Stellen auch kurz auf diese Punkte ein, damit Sie eine Vorstellung davon erhalten, welche Möglichkeiten Ihnen die Bibliotheken bieten, die Sie im Simulator testen können – auf dem Gerät bekommen Sie bei der Ausführung dieser Programme allerdings eine Fehlermeldung mit dem Hinweis, dass die Software nicht vertrauenswürdig ist und nicht ausgeführt werden kann.

Noch ein wichtiger Hinweis zum MDS

Nach Informationen von Heise Online wird RIM das MDS komplett aufgeben und in Zukunft auf zwei Entwicklungsansätze für Java- und Web-Anwendungen setzen. Die MDS-Runtime und das MDS-Studio sind seit Ende des Jahres 2009 nicht mehr verfügbar; der Support endet Mitte 2010.

Um die Entwicklung von Webanwendungen zu unterstützen, bietet RIM eine aktualisierte Version 1.2 des Plugins für die Microsoft-Entwicklungsumgebung Visual Studio an. Sie benötigt die Visual Studio 2008 Professional Edition auf Windows XP oder Vista in der 32-Bit-Version. Zusätzlich veröffentlicht RIM ein neues Plugin für Eclipse 3.4, das auf den gleichen Windows-Versionen läuft.

Beide Tools enthalten einen Gerätesimulator, sodass man Anwendungen direkt auf dem PC testen kann, ohne sie auf ein BlackBerry-Gerät übertragen zu müssen. Javascript lässt sich mit den neuen Plugins sowohl debuggen als auch profilieren. Darüber hinaus zeigen die Tools den Inhalt von XMLHttpRequest-Übertragungen an.

1.1 Voraussetzungen

Ohne Programmierkenntnisse kommen Sie mit diesem Buch nicht weit. Daher erklärt sich auch der Anspruch des Buches, das sich vom Entwickler an den Entwickler richtet.

Sie sollten sich schon einmal mit der freien Entwicklungsumgebung Eclipse oder Microsofts Visual Studio befasst haben. Mein Augenmerk richtet sich auf die freie Entwicklungsumgebung Eclipse. Anhand detaillierter Schritt-für-Schritt-Anleitungen lernen Sie, wie Sie die Eclipse-Umgebung einrichten und erste Projekte verwirklichen.

Sie sollten auch in der Lage sein, kleine Batch-Programme zu schreiben, denn diese werden wir im Laufe des Buches öfter einsetzen, um die Programme in den Speicher des BlackBerry zu flashen.

Sie sollten also wissen, wie Sie in einer grafischen Oberfläche (GUI) wie Windows oder der Linux-Variante Ubuntu ein Terminalfenster öffnen (bei Windows lautet der Fachausdruck dafür Eingabeaufforderung) und damit umgehen. Dass wir ab und zu auf der Kommandozeile arbeiten werden, mag Sie etwas irritieren, da der gewöhnliche Anwender damit in der Regel nicht mehr vertraut ist und sich auch manche Entwickler an grafische Entwicklungsumgebungen gewöhnt haben.

Dennoch werden Sie nicht auf die Kommandozeile verzichten können, wenn Sie effektiv programmieren und anschließend Ihren Code auf das Gerät übertragen wollen.

Einer der Hauptgründe, warum Sie auf der Kommandozeile arbeiten müssen, ist der, dass bestimmte Loader-Programme eine Vielzahl von Parametern benötigen, um einen Job auszuführen. Für diese Programme gibt es schlichtweg keine grafische Oberfläche.

Sie sollten sich unbedingt angewöhnen, gut dokumentierten Code zu schreiben, diesen im Simulator ausgiebig zu testen, ihn dann mit dem Batch-Programm auf das Gerät zu bringen und den Code dort erneut zu testen.

Und zu guter Letzt möchte ich Ihnen raten, regelmäßig Sicherungen von Ihrem Gerät sowie der Software anzufertigen. Ein sogenanntes Versionsverwaltungssystem ist hier als Mindestvoraussetzung für sicheres Programmieren von mobilen Geräten zu sehen.

Wie ein Versionsverwaltungssystem arbeitet, möchte ich Ihnen kurz erklären. Sie haben zum Beispiel den Quellcode im Repository des Versionsverwaltungssystems »eingescheckt«. Das bedeutet, dieser Code ist fehlerfrei und wurde zuvor getestet. Nun besteht Bedarf an einer Änderung oder Erweiterung.

Jetzt wird der Code aus dem System »ausgescheckt«, und Sie haben eine sogenannte Arbeitsversion auf Ihrem Entwicklungsrechner. Nachdem Sie den Code angepasst und getestet haben, wird die COD-Datei kompiliert.

Diese Datei ist dazu bestimmt mittels einer Batchdatei, auf das mobile Gerät transportiert zu werden. Um den Quellcode wieder sicher aufzubewahren, wird er erneut in das Repository »eingescheckt«. Gleichzeitig erhöht sich der Versionszähler, und Sie haben eine getestete und kompilierte Datei mit einer bestimmten Versionsnummer erzeugt.

Dieser Vorgang kann auch rückgängig gemacht werden, wenn Sie doch lieber die letzte Version oder das letzte Release verwenden wollen. All das erledigen Sie beispielsweise mit der freien Software SubVersion (<http://subversion.tigris.org/>). Eine komplette Beschreibung, wie Versionsverwaltungssysteme funktionieren und wie sie zum Beispiel mit einem Defect Tracking-System zur Fehlerverfolgung gekoppelt werden, würde allerdings einen eigenen Band füllen.

Das ist aber nicht das Ziel dieses Buches. Ich möchte Ihnen hier lediglich einen Denkanstoß für Ihre Arbeit geben.

1.2 Die Grenzen dieses Buches

Was Sie in diesem Buch nicht finden werden, ist die Beschreibung, wie eine komplette Java-Applikation mit Datenbankzugriff entwickelt wird, oder wie Sie komplexe Mail- und Bluetooth-Anwendungen programmieren, wie sie im kommerziellen Bereich angeboten werden. Wir werden uns in diesem Buch um Programme kümmern, die im Gerätesimulator laufen und sich gegebenenfalls auch ohne die Anschaffung einer Entwicklerlizenz kompilieren lassen.

Für alle, die jetzt enttäuscht das Buch beiseite legen wollen: Urteilen Sie nicht zu früh! Es gibt in diesem Werk sehr viele Anregungen und Ideen, auf die Entwickler aufbauen können, um ihre Applikationen aufzuwerten.

Das wichtigste Kriterium ist: Sie sollen die Programmierbeispiele als Anregungen für Ihre eigenen Projekte sehen und dieses Buch als Nachschlagewerk betrachten¹. Ich gehe davon aus, dass Sie professionell programmieren und alle Möglichkeiten einer Programmiersprache (in diesem Fall Java) kennen. Ohne dieses Grundwissen werden Sie schnell an Ihre Grenzen geraten.

1.3 Einige Begriffe vorweg

Entwicklungsumgebung

Die in diesem Buch beschriebene Entwicklungsumgebung ist Eclipse.

Eclipse ist ein quelloffenes² Programmierwerkzeug. Ursprünglich wurde Eclipse als integrierte Entwicklungsumgebung (englisch: Integrated Development Environment, IDE) für die Programmiersprache Java genutzt, aufgrund seiner Erweiterbarkeit wird es aber mittlerweile auch für viele andere Entwicklungsaufgaben eingesetzt.

Für Eclipse gibt es eine Vielzahl von Erweiterungen – sowohl quelloffene als auch von kommerziellen Anbietern entwickelte.

Eclipse selbst basiert auf Java-Technologie, seit der Version 3.0 dient ein sogenanntens OSGI³-Framework namens Equinox als Grundlage.

Exkurs Softwaretest

Unit-Tests /Whitebox-Tests

Unit-Tests oder Modultests⁴ werden vom Entwickler durchgeführt. Sie sind ein wesentlicher Teil einer Software-Teststrategie. Diese Tests dienen zur Verifikation von Modulen einer Software, zum Beispiel von Klassen. Modultests sollen möglichst nicht die Interna einer Methode testen, sondern nur ihre externen Auswirkungen (Rückgabewerte, Ausgaben, Änderungen an Dateien, Datenbanken oder Backend-Systemen).

Werden auch interne Details der Methode geprüft (dies wird als Whitebox-Testing bezeichnet), droht der Test fragil zu werden. Das bedeutet, dass er auch fehlschlagen könnte, obwohl sich die externen Auswirkungen nicht geändert haben. Gute Unit-Tests

¹ Dass Sie in einer freien Entwicklungsumgebung ohne RIM-Entwicklerlizenz gewissen Beschränkungen unterliegen, hat unter anderem mit dem Sicherheitskonzept der Mobilgeräte von RIM zu tun. Denjenigen, die tiefer einsteigen wollen, empfehle ich eine Codelizenz von RIM, mit der sie Signed-Coding realisieren können.

² Die freie Entwicklungsumgebung Eclipse wird im Wikipedia-Eintrag [http://de.wikipedia.org/wiki/Eclipse_\(IDE\)](http://de.wikipedia.org/wiki/Eclipse_(IDE)) genauer beschrieben. Die bei Drucklegung aktuelle Version 3.4 (Projektname *Eclipse Ganymede*) ist hier in diesem Buch verwendet.

³ Das Kürzel OSGI leitet sich von der »Open Services Gateway Initiative« ab. Ziel dieser herstellerübergreifenden Industriallianz war eine servicebasierte Systemplattform. Heute ist nur noch das Kürzel OSGI in Gebrauch, die Langfassung gilt als obsolet.

⁴ Unit-Tests werden auch als Komponententests bezeichnet. Quelle: <http://de.wikipedia.org/wiki/Modultest>

sind zum Beispiel Tests einer Methode, in denen Schleifen und Bedingungen jeweils einmal durchlaufen werden.

Für die Tests auf der Ebene der Endanwender sei das sogenannte Blackbox-Testing empfohlen, bei dem man sich auf das Prüfen der externen Auswirkungen beschränkt.

1.4 Typografie

In diesem Buch werden verschiedene Schrift- und Gestaltungsarten verwendet, um Kontexte, Textteile und Bedeutungen darzustellen:

Quelltext wird in nichtproportionaler Schrift dargestellt und ist grau unterlegt:

Quelltext 1:

```
import net.rim.device.api.system.Application;
```

Die folgenden Zeilen enthalten einen etwas längeren Quelltextabschnitt. Die Zeichenfolgen `»/*«` und `»*/«` markieren Zeilen, die anschließend ausführlicher erklärt werden. Es sind Kommentare, auf die ich im Einzelnen weiter eingehe. Kurzkommentare werden mit einem Doppel-Slash (`//`) eingeleitet.

Quelltext 2:

```
// NOTE: You MUST change the package name to reflect your company
// name package com.acme.blackberry.webicon;

import net.rim.device.api.system.Application;
import net.rim.blackberry.api.browser.BrowserSession;

/**
 * This class allows easy access for launching a browser for
 * devices 4.0 and later
 */
class WebIcon extends Application
{
    public static final int DEFAULT_BROWSER = 0;
    public static final int WAP_BROWSER = 1;
    public static final int BES_BROWSER = 2;
    public static final int WIFI_BROWSER = 3;
    public static final int BIS_BROWSER = 4;
    public static final int UNITE_BROWSER = 5;

    /*
     * Attempts to launch the given browser to the given url.
     * If launching this browser is not successful then the
     * default browser will be launched.
     *
     * @param int browserType
     * @param String url
     */
```

Werden innerhalb des laufenden Textes Quelltextelemente dargestellt, so werden auch diese in nichtproportionaler Schrift dargestellt, allerdings ohne diese Textpassagen grau zu unterlegen.

Ebenfalls in Schreibmaschinenschrift, aber ohne graue Hervorhebung, wird der Inhalt des Konsolenfensters dargestellt. Dabei kann es sich um Benutzereingaben, um Programmausgaben oder um beides handeln:

```
C:\>install.bat
```

Wenn Betriebssystembefehle oder sonstige Ein- oder Ausgaben der Konsole im Fließtext dargestellt werden, werden sie, genau wie Quelltextauszüge, in nichtproportionaler Schrift dargestellt.

Wichtige Anmerkungen, Hinweise oder Tipps sehen folgendermaßen aus:

Dieses ist ein sehr wichtiger Text.

Beziehe ich mich auf Menüpunkte oder Beschriftungen von Dialogelementen, werden diese kursiv gesetzt – wie etwa die Schalter *OK* und *Abbrechen*. Bei einer Hierarchie werden die einzelnen Ebenen durch Schrägstriche getrennt, so etwa bei »Datei/Speichern unter...«.

Tasten, die Sie auf der Tastatur drücken sollen, werden in eckigen Klammern angegeben. So wird die Taste »A« als [A] dargestellt oder die Taste »Strg« als [Strg]. Wenn der numerische Tastenblock gemeint ist, beginnt die Tastenbezeichnung mit einem »N« – die Taste »5« wird auf dem Ziffernblock als [N5] dargestellt und das Pluszeichen als [N+].

Wenn etwas im Fließtext besonders betont werden soll, dann wird es ebenfalls *kursiv* geschrieben.

Datei- und Verzeichnisnamen werden, sofern sie allgemeingültig sind, im Unix-Stil dargestellt, das heißt, dass vorwärtsgerichtete Schrägstriche für die Trennung von Verzeichnisebenen verwendet werden. Bezieht sich eine Verzeichnisangabe nur auf ein bestimmtes System, erfolgt diese Angabe in einer Darstellung, die dem jeweiligen System entspricht. Unabhängig davon werden aber alle Datei- und Verzeichnisnamen in Anführungszeichen gesetzt: »/home/klaus/MyProjects« oder die im Workspace definierten Ablageorte »C:\Dokumente und Einstellungen\klaus\workspace«.

Zitate werden in Anführungszeichen gesetzt, ebenso Texte, die als Querverweise verwendet werden, da diese im Grunde auch Zitate sind. Anführungszeichen werden an einigen Stellen auch nur verwendet, um den Text übersichtlicher zu gestalten – in ähnlicher Weise wie runde Klammern einen Kontext zusammenfassen oder vom restlichen Text abtrennen.

Ich gestatte, alle im Buch abgedruckten Quelltexte zu verändern oder auch unverändert zu verwenden. Sie dürfen auch verändert oder unverändert zur Erstellung kommerzieller Software eingesetzt werden.

Ich gestatte nicht, dass die Quelltexte veräußert werden, weder unter meinem noch unter einem anderen Namen.

Die in diesem Buch benutzte Entwicklungsumgebung ist die zur Zeit der Drucklegung aktuelle von RIM freigegebene Eclipse-Version Ganymede Version 3.4.1, Build-ID M20080911-1700. Weiter werden verwendet das RIM MDS Studio in der Version 2.0.0.75, Build Date: 200810091236, sowie das BlackBerry JDE Plugin for Eclipse in der Version 1.0.0.67.

Im Anhang finden Sie eine Linksammlung mit URL-Angaben für den Download sowie weitere wertvolle Hinweise.

2 Installation und Einrichtung der Entwicklungsumgebung Eclipse

In diesem Kapitel werden wir uns mit der Einrichtung der Entwicklungsumgebung Eclipse befassen. Anhand von Screenshots werden Sie an das erste Programm herangeführt – natürlich das bekannte und allseits beliebte »Hello World«.

2.1 Systemanforderungen

RIM hat eine Java-Umgebung mit einem Eclipse-Plugin zusammengestellt, die eine komplette Entwicklungsumgebung inklusive Gerätesimulator für die geschriebene Software enthält.

Zunächst laden wir uns die Umgebung von der offiziellen Webseite von RIM herunter, die unter folgendem Link erreichbar ist:

<http://na.blackberry.com/eng/developers/javaappdev/javadevenv.jsp>

Da sich dieses Buch vorrangig mit dem Java Development Environment (IDE) für Eclipse beschäftigt, gehe ich im folgenden Abschnitt auf die Systemvoraussetzungen ein, die die Entwicklungsumgebung an Ihr System stellt.

2.1.1 Systemanforderungen unter Windows:

- Windows 2000, Windows XP oder Windows Vista
- Prozessor: Pentium-4-Prozessor (Minimum 3 GHz)
- 1 GB RAM Minimum
- Java Development Kit (JDK) Version Java SE 6.0 oder höher
- Eclipse-Version 3.4 oder höher

2.2 Installationsschritte

Nach dem Download startet man den Installer. Wenn sich Eclipse zuvor noch nicht auf dem Rechner befunden hat, fragt der Installer nach Verzeichnisnamen und schlägt ein

Installationsverzeichnis vor. Der Installer richtet alle nötigen Umgebungen für Eclipse ein, ebenso das IDE-Plugin. Einige Screenshots zeigen den Installationsverlauf:

1. Installation von Eclipse und dem Plugin nach `c:\blackberry\ide\eclipse`

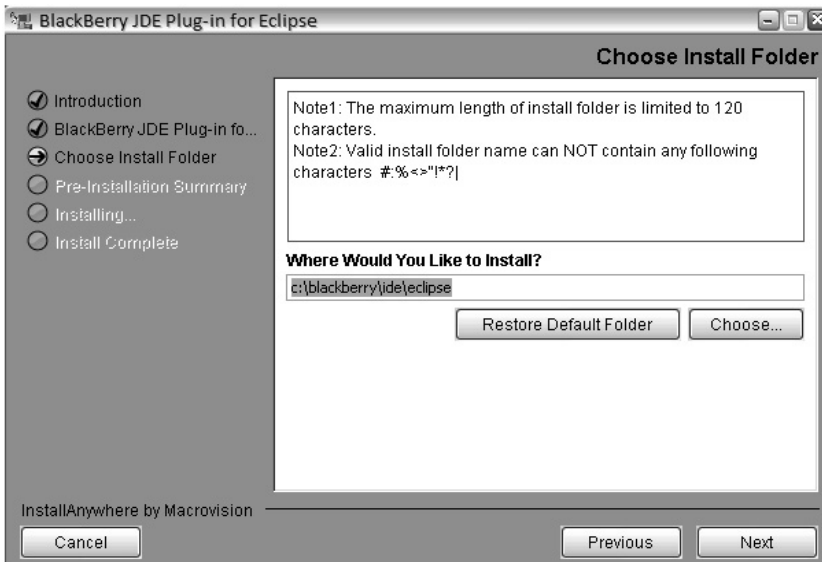


Abbildung 2.1: Installation von Eclipse und Plugin

2. Installation fertigstellen:

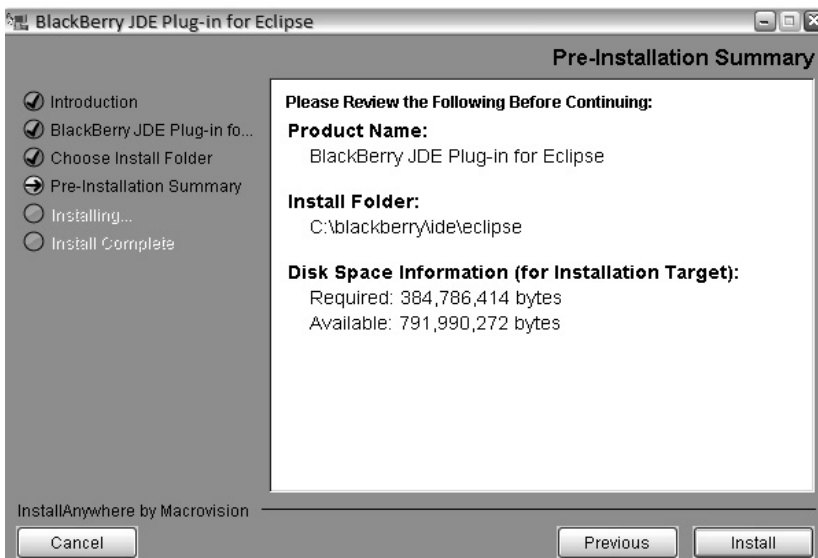


Abbildung 2.2: Eclipse-Installation fertigstellen

Eclipse ist nun startbereit, und die Installation wird abgeschlossen:

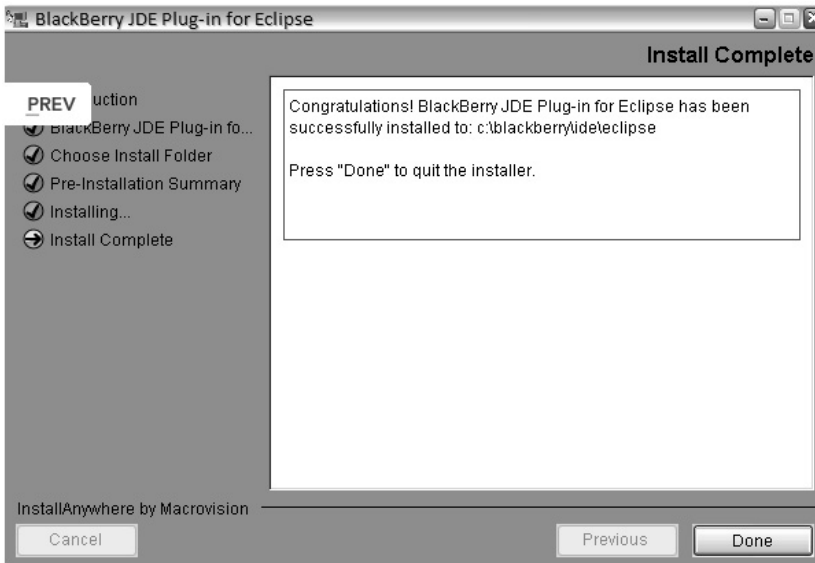


Abbildung 2.3: Abschluss der Installation

2.3 Erstes Projekt erstellen

Nachdem wir Eclipse auf unserem Rechner installiert haben, können wir ein neues Projekt beginnen. Ich zeige Ihnen in einem Screenshot, wie Eclipse nach dem Doppelklick auf das Startsymbol aussieht. Zunächst benötigt Eclipse für die Entwicklungsarbeit einen sogenannten Workspace.

Dieser Workspace ist im Grunde die Verzeichnisstruktur, in der die Projekte unter Eclipse verwaltet werden. Es ist denkbar und praktikabel, sich für ein anderes Projekt einen anderen Workspace einzurichten. Eclipse fragt bei jedem Start den Workspace an. Mit dem Listenfeld können vorherige Workspaces angezeigt werden, und mit der Option »Browse« kann nach einem Workspace gesucht werden.

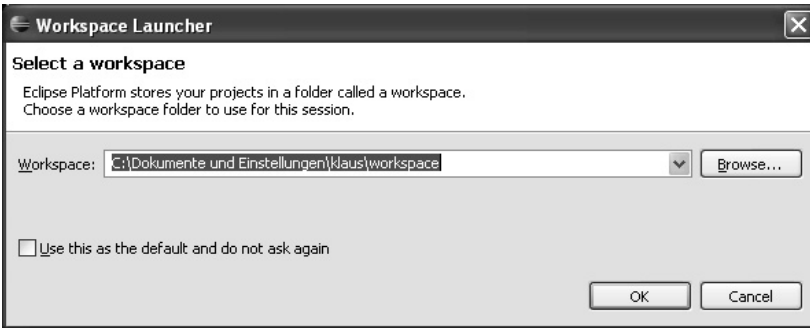


Abbildung 2.4: Ein neuer Workspace wird erzeugt

2.4 Einstellungen in der Entwicklungsumgebung

Nachdem wir nun die Entwicklungsumgebung erfolgreich auf unserem Rechner installiert haben, müssen wir einige Anpassungen vornehmen. Was wir anpassen müssen, sind folgende Inhalte:

1. Auswahl der Option *Configure BlackBerry Workspace* in dem Reiter *Preferences* der Entwicklungsumgebung.

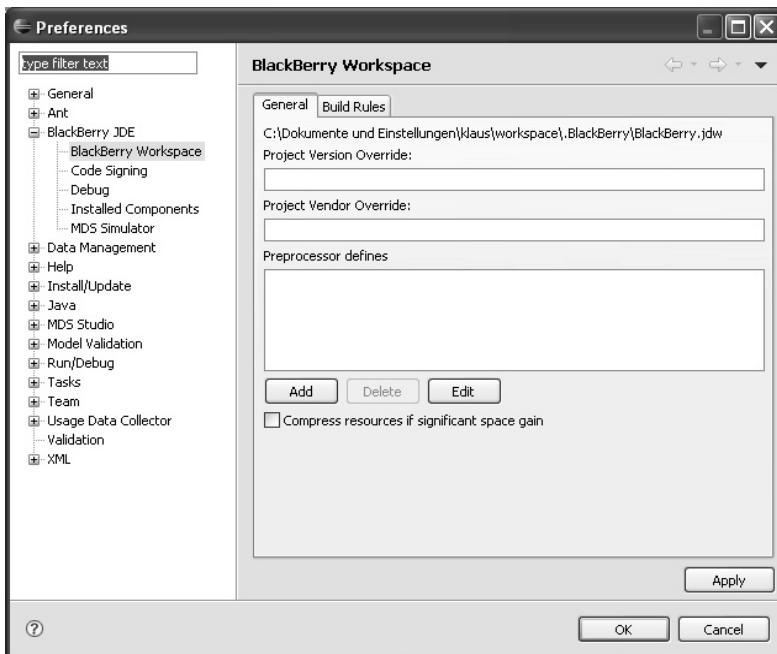


Abbildung 2.5: Einstellungen zum BlackBerry-Workspace

Zu erreichen ist das Menü unter *File/New/Project*. Dann sollten Sie die Option »Blackberry Project« auswählen. Es erscheint der BlackBerry-Projektwizard:

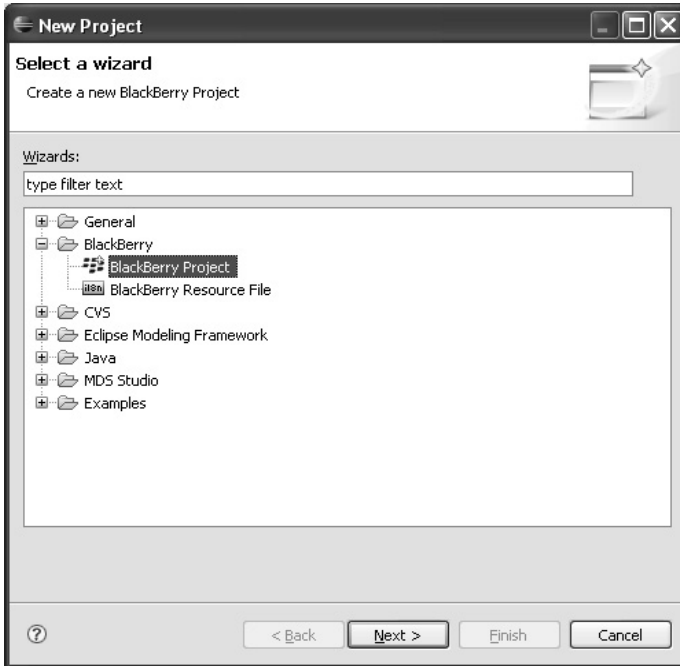


Abbildung 2.6:
Projektwizard

2. Projektnamen vergeben.
Unser Projekt bekommt nun einen Namen.

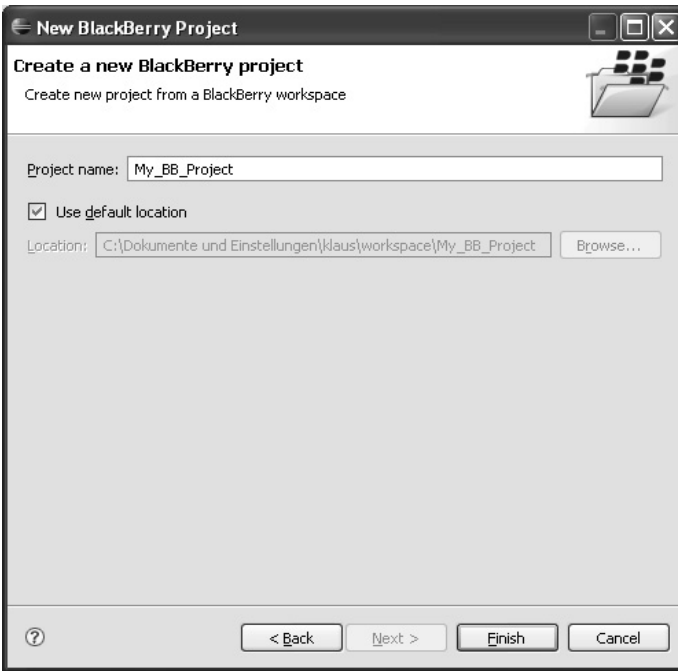


Abbildung 2.7:
Projektamen
vergeben

3. Jetzt haben wir es fast geschafft, müssen aber für unserer Projekt zusätzlich noch eine Klasse definieren. Wir erreichen diesen Dialog unter *File/New/Class*.

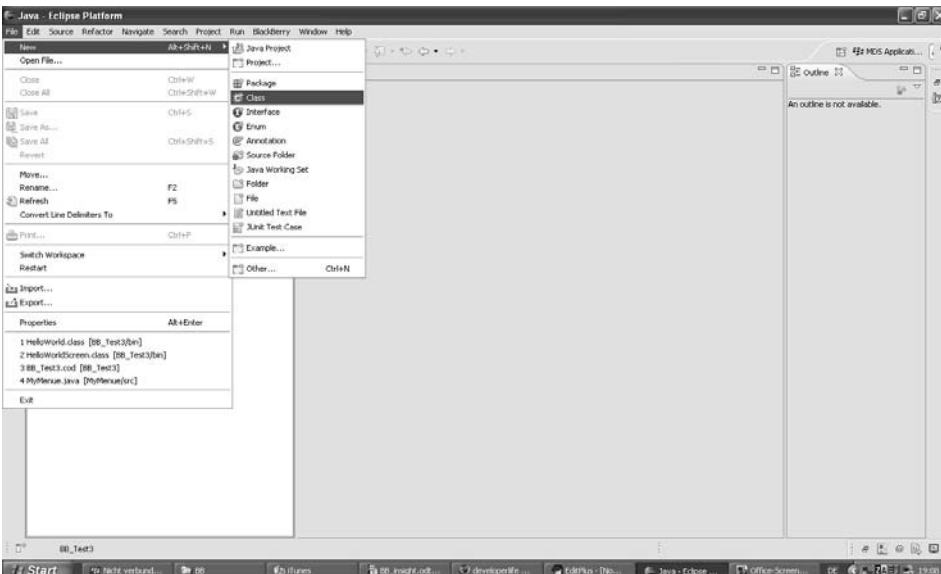


Abbildung 2.8: Dialog *New Class*

Es öffnet sich der folgende Dialog:

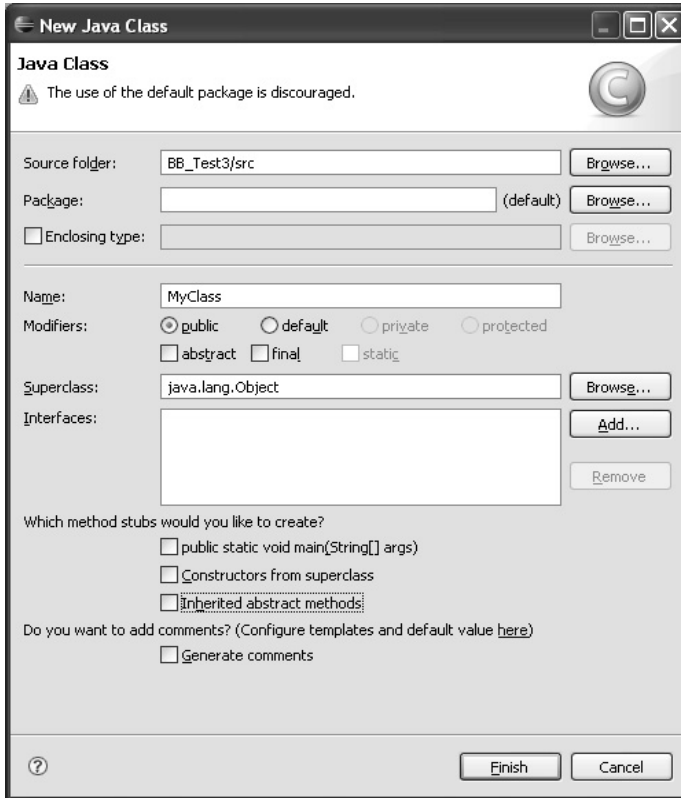


Abbildung 2.9: Neue Java Class benennen

Wir benennen unsere erste Java-Klasse zum Beispiel mit dem Namen `MyClass`. Der Name ist wichtig, da Eclipse daraus einen Konstruktor baut, in den wir alle BlackBerry-Bibliotheken einladen und mit denen wir arbeiten können. Nun sind alle Einstellungen gemacht, und wir können mit dem ersten Code beginnen. Es bietet sich an, hier in einer Checkliste zusammenzufassen, was nötig ist, um ein neues Projekt zu erzeugen.

2.5 Checkliste zur Erzeugung eines neuen Projekts

1. Einen Workspace für das Projekt festlegen
2. *Configure BlackBerry Workspace* aufrufen
3. *File/New/Project* – ein BlackBerry-Projekt auswählen

Der Workspace-Name und der Projektname sind nicht gleich. Zum besseren Verständnis kann man sich einfach vorstellen, dass der Workspace den Mandantenbaum darstellt und das BlackBerry-Projekt innerhalb dieses Mandantenbaums aufgebaut wird.

4. Einen Projektnamen vergeben. Der Name hat, wie gesagt, nichts mit dem Workspace zu tun. Der Workspace liefert die Dokumentenstruktur für das Projekt.
5. Eine neue Java-Klasse erzeugen über *File/New/Class*.
6. Bereit zum Coding.
7. Nach dem Schreiben des Codesegments sollten noch vor dem Kompilieren die *Project-Properties* gesetzt werden.

2.6 Das erste Codebeispiel schreiben

Quelltext 1: Hello World

```
/**
 * HelloWorld.java
 */

import net.rim.device.api.ui.*;
import net.rim.device.api.ui.component.*;
import net.rim.device.api.ui.container.*;
import net.rim.device.api.system.*;

public class HelloWorld extends UiApplication {
    public static void main(String[] args) {
        HelloWorld theApp = new HelloWorld();
        theApp.enterEventDispatcher();
    }
    public HelloWorld() {
        pushScreen (new HelloWorldScreen());
    }
}

final class HelloWorldScreen extends MainScreen {
    public HelloWorldScreen() {
        super();
        LabelField title = new LabelField("HelloWorld Sample",
LabelField.ELLIPSIS
        | LabelField.USE_ALL_WIDTH);
        setTitle(title);
        add(new RichTextField("Hello World!"));
    }
}
```

```
public boolean onClose() {  
    Dialog.alert("Goodbye!");  
    System.exit(0);  
    return true;  
}  
}
```

2.7 Beschreibung des Quelltextes

Die ersten Zeilen sind Kommentare. Diese dienen der besseren Lesbarkeit und Verständlichkeit des Codes.

Quelltext 2: Kommentarzeilen

```
/**  
 * HelloWorld.java  
 */
```

Denn die Kommentare beinhalten in der Regel die Informationen, was dieses Programm macht, wer es geschrieben hat, wann es geschrieben wurde, vielleicht auch noch eine Versionsnummer.

Wenn Sie später diesen Code ändern oder Code von einem Kollegen übernehmen, sind Kommentare hilfreich wie ein Leitfaden, an dem Sie sich orientieren können. Diese programminternen Kommentare werden von den bereits angesprochenen Project-Properties ergänzt. Project-Properties beziehen sich auf die kompilierte Datei und ihre Dateieigenschaften. Auch hier werden *Titel*, *Version*, *Autor*, *Vendor*, *Description* und Ähnliches eingetragen.

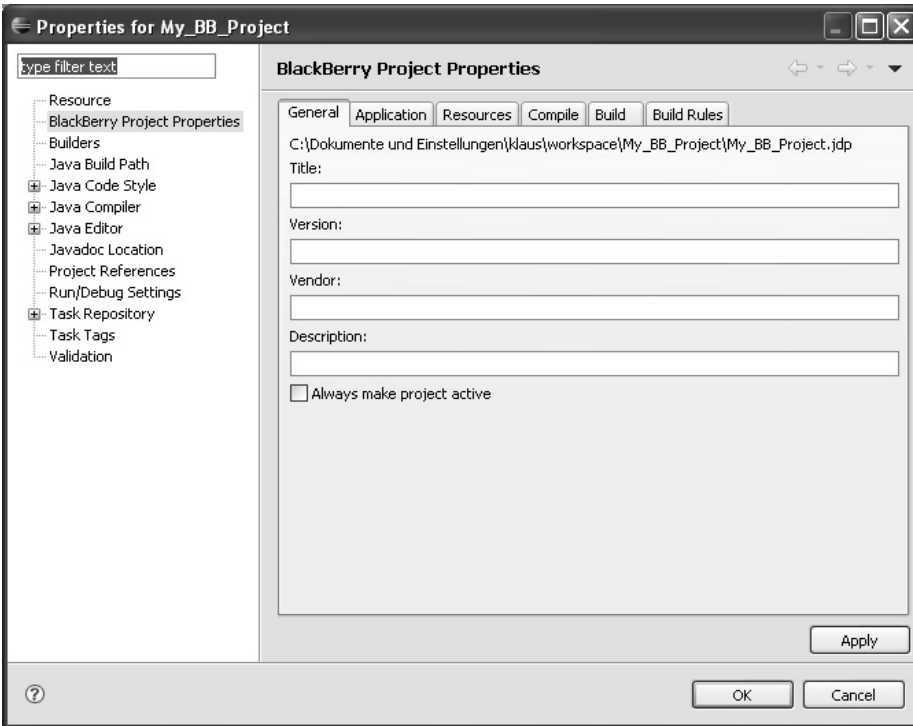


Abbildung 2.10: Project-Properties setzen

Ich habe die Project-Properties wie folgt ausgefüllt, um Ihnen eine Anregung zu geben:

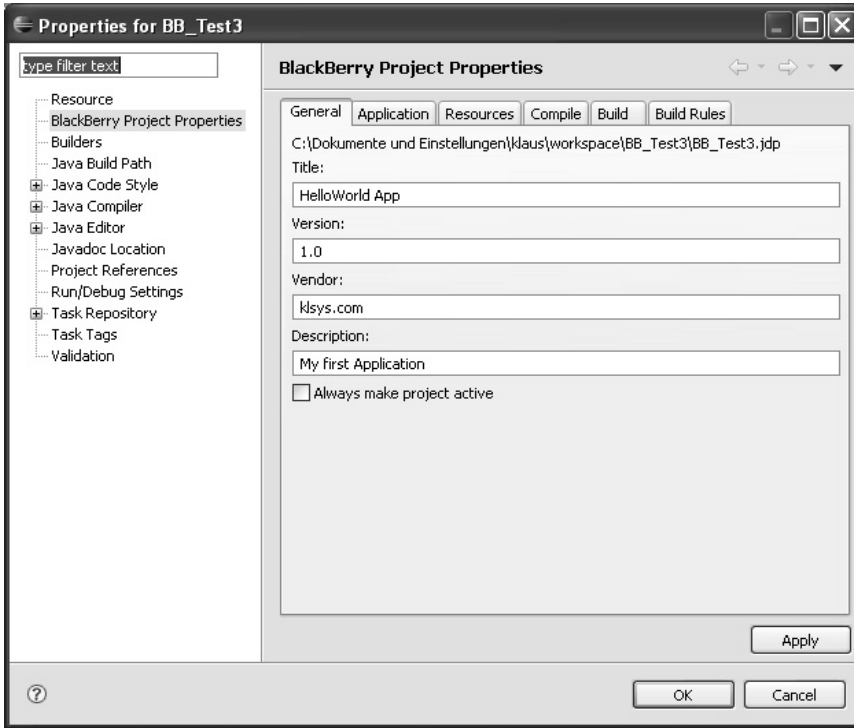


Abbildung 2.11: Ausgefüllte Project-Property-Karte

Da unsere erste Applikation noch kein Icon besitzt, wollen wir das hiermit nachholen. Ein Bild sagt mehr als tausend Worte. Somit sind Icons auch für den BlackBerry durchaus sinnvoll. Jedoch gibt es bei den Icons einiges zu beachten, was die Abmessungen und Formate betrifft.

2.8 Applikations-Icons

Wechseln Sie bitte zu dem Reiter *Resources*. Die Maske sollte wie folgt aufgebaut sein:

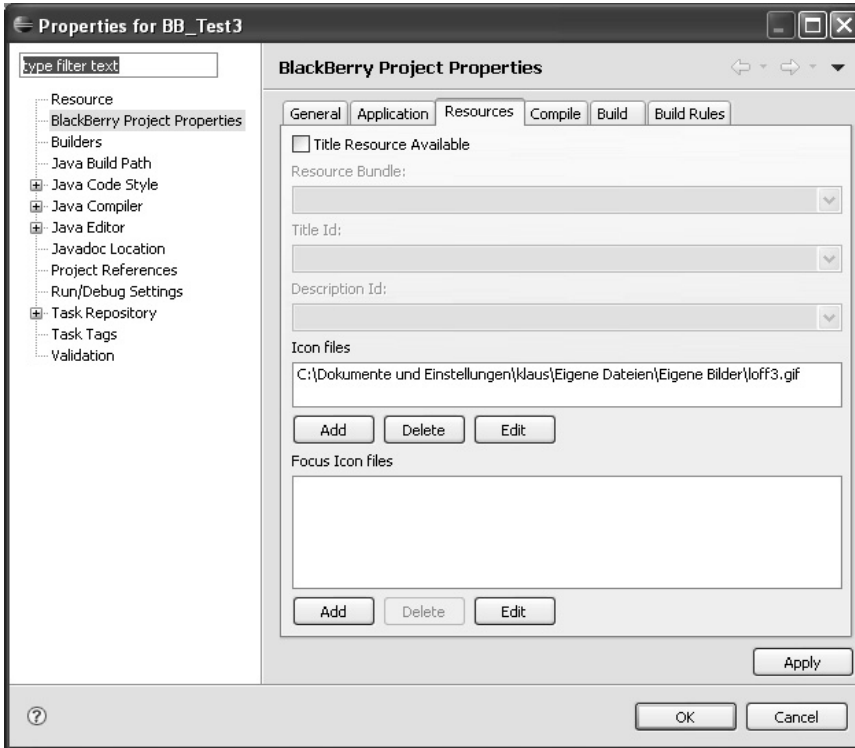


Abbildung 2.12: Angabe der Icondatei

Unter *Icon Files* können Sie das Icon angeben, welches Ihrer Applikation zugeteilt werden soll. Eine ausführliche englische Dokumentation von RIM findet man unter den Smartphones-UI-Guidelines⁵.

⁵ http://docs.blackberry.com/en/developers/deliverables/6625/Dimensions_for_screens_images_and_icons_476251_11.jsp

3 Den BlackBerry programmieren

In diesem Kapitel stelle ich Ihnen das BlackBerry-System vor. Zunächst möchte ich mich aber noch einem wertvollen Tool zuwenden, dem Gerätesimulator. Der Smartphone-Simulator ist eine Anwendung, die Sie auf Ihrem Computer installieren und die eine Abbildung des BlackBerry-Gerätemodells Ihrer Wahl anzeigt.

3.1 Das Programm wird im Simulator gestartet

Nun ist das in Kapitel 2 erstellte Programm startbereit und wird mit *Run/Run* und der Auswahl des BlackBerry-Simulators nach der Bestätigung mit *OK* gestartet. Nach kurzer Ladezeit öffnet sich der Device-Simulator. Der Simulator muss nicht so aussehen wie das aktuelle Modell (etwa die Baureihe Storm 9500), aber im Kern sprechen wir die API der anfangs genannten Modellreihen an. Weitere von RIM freigegebene Device-Simulationsmodelle finden Sie im Anhang F.2 »Links«. Um nun zu der »Hello World«-Applikation zu gelangen, navigiert man durch den virtuellen BlackBerry unter der BB-Taste, *Applications*.



Abbildung 3.1: BlackBerry Device Simulator

Die Navigation erfolgt wie im echten Gerät – das heißt, Sie klicken über den Trackball, die BB-Taste steht für *Escape* und so weiter. Wenn Sie mit dem echten BlackBerry vertraut sind, dürfte es kein Problem für Sie sein, die Anwendung auch im Simulator zu bedienen.



Abbildung 3.2: Device-Simulator mit BlackBerry Bold

3.2 Das BlackBerry-System

BlackBerry-Geräte enthalten eine Java-ME-Ablaufumgebung (ME steht für »Micro Edition«, wird manchmal auch als »Mobile Edition« übersetzt), die Client-Server-Anwendungen unterstützt.

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 Anwendung sollten Sie stets die Unterschiede zwischen mobilen Geräten und PCs beachten (zu Letzteren zählen auch Macs).

Mobile Geräte

- haben eine kleinere Bildschirmgröße, auf der nur eine begrenzte Anzahl von Zeichen dargestellt werden kann,
- haben in der Regel langsamere Prozessoren,
- 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 darüber hinaus auf andere Weise als Anwendungen auf einem Personal Computer. Zum Beispiel erwarten die Benutzer, dass sie die gesuchten Informationen auf mobilen Endgeräten schnell finden.

Ein CRM-System (Customer Relationship Management) kann beispielsweise eine immense Datenmenge umfassen, aber die Benutzer brauchen nur einen Bruchteil dieser Informationen auf einmal. Daher kann eine solche Anwendung durchaus auch für ein mobiles Gerät bereitgestellt werden. Die Benutzeroberfläche des Geräts ist dann so gestaltet, dass die User ihre Aufgaben mühelos erledigen und rasch auf Daten zugreifen können.

Wenn Sie Anwendungen für den BlackBerry entwickeln, sollten Sie versuchen, diese Applikationen so konsistent wie möglich mit anderen BlackBerry-Anwendungen zu halten. Befolgen Sie die folgenden Anweisungen:

- Verwenden oder erweitern Sie nach Möglichkeit vorhandene Benutzeroberflächen-Komponenten, damit Ihre Anwendung das Standardverhalten der Komponente erben kann.
- Halten Sie sich so eng 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 Anwender 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 -darstellung so weit, dass nur die 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 gegebenenfalls die Hauptanwendungen auf dem BlackBerry-Gerät studieren oder auch den BlackBerry Smartphone Simulator heranziehen, um mehr über das Navi-

gationsmodell und über bewährte Verfahren zur Gestaltung der Benutzeroberfläche für Ihre geplante Applikation zu erfahren.

3.2.1 Versionszyklen und Versionen

Alle BlackBerry-Geräte enthalten eine bestimmte Version der BlackBerry Device-Software und der BlackBerry Java Virtual Machine. Um die Version der Device-Software (also des Betriebssystems) eines Gerätes zu bestimmen, klicken Sie in den Geräteoptionen auf *Info*. Sie können die Device-Software auch aktualisieren. Beispielsweise ist es möglich, die Device-Software Version 4.0 eines Gerätes auf die Version 4.1 zu aktualisieren.

Jede Hauptversion der Device-Software und der JVM wird vom Hersteller Research In Motion zusammen mit den zugehörigen Java-APIs und der passenden Version der Entwicklungsumgebung Java Development Environment (JDE) ausgeliefert. Die Version der Entwicklungsumgebung, mit der Sie Anwendungen entwickeln können, richtet sich stets nach der Version der Device-Software, auf die Sie abzielen.

Ein kurzes Beispiel zum besseren Verständnis: RIM hat beispielsweise die Device-Software Version 4.0 und die Version 4.0 der JDE gleichzeitig veröffentlicht. Die JDE-Version 4.0 bietet Unterstützung für die APIs, die mit der Version 4.0 der Device-Software und der JVM-Version 4.0 eingeführt wurden. Anwendungen, die Sie mit der JDE-Version 4.0 erstellen, funktionieren nur auf Geräten, auf denen die Device-Software Version 4.0 oder höher ausgeführt wird.

Entscheiden Sie anhand der folgenden Kriterien, welche Version der JDE Sie für die Entwicklung verwenden:

- Wenn die Anwendung keine besonderen Hardwarefunktionen und auch keine der erst vor Kurzem veröffentlichten API-Erweiterungen verwenden muss, entwickeln Sie die Applikation mit der JDE-Version 4.0.
- Soll die Anwendung ausschließlich auf dem Smartphone-Modell Pearl 8100 ausgeführt werden, verwenden Sie die JDE-Version 4.2.

3.2.2 BlackBerry Java Development Environment

Das Java Development Environment ist eine vollständig integrierte Entwicklungs- und Simulationsumgebung zum Erstellen einer Java-Applikation für BlackBerry-Geräte. Das JDE stellt Entwicklern die Programmiersprache Java ME und die erweiterten Java-APIs für BlackBerry-Anwendungen zur Verfügung.

Die Umgebung enthält die folgenden Entwicklerwerkzeuge und -hilfen:

- die eigentliche Entwicklungsumgebung (Integrated Development Environment, kurz IDE),
- den bereits erwähnten Smartphone-Simulator,

- Java ME-APIs und BlackBerry-APIs,
- Beispielanwendungen.

Die IDE umfasst eine ganze Reihe von Bearbeitungs- und Fehlerbehebungs-Tools, die auf die Entwicklung einer BlackBerry-Applikation ausgelegt sind.

Der Smartphone-Simulator dient zur Simulation der Benutzeroberfläche und der Benutzerinteraktion, der Netzwerkverbindungen, der E-Mail-Dienste und der drahtlosen Datensynchronisierung.

Das Java Development Environment Component Package enthält außerdem die folgenden Entwicklungs-Tools für die Programmierung in Drittanbieter-IDEs wie NetBeans oder Eclipse:

- **RAPC:** Mit diesem Kommandozeilen-Compiler können Sie Java- und JAR-(Java Archive)Dateien in COD-Dateien kompilieren, die Sie dann im Smartphone-Simulator oder auf einem BlackBerry-Gerät ausführen können.
- **JavaLoader:** Mit diesem Tool können Sie eine Anwendung auf einem Gerät zu Testzwecken hinzufügen oder aktualisieren sowie Informationen zu den COD-Dateien der Anwendung abrufen.
- **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 Smartphone laden.
- **JDWP:** Mit diesem Tool können Sie die Fehlerbehebung von Anwendungen mithilfe der integrierten Entwicklungsumgebung von Drittanbietern ausführen.

3.2.3 Java ME und die Java-APIs für den BlackBerry

Java ME ist eine Standardplattform, die häufig genutzte Java-APIs für verschiedene Arten von drahtlosen und eingebetteten Geräten definiert. Eine Java-ME-Anwendung auf einem BlackBerry-Gerät wird in der Virtual Machine ausgeführt. Diese virtuelle Maschine stellt sämtliche Runtime-Dienste für die Anwendungen bereit und übernimmt Funktionen wie typische Speicherzuordnungen, Sicherheitsprüfungen und die Speicherbereinigung (Garbage Collection).

BlackBerry-Geräte unterstützen den Java ME-MIDP-Standard (Mobile Information Device Profile) gemäß der Definition in JSR (Java Specification Request) 118. Der Java ME-MIDP-Standard liefert eine Kerngruppe von Java-APIs, die von einer Reihe mobiler Geräte unterstützt wird, unabhängig vom jeweils zugrunde liegenden Betriebssystem. Entwickler können daher eine Java-Anwendung mit den MIDP-Standard-APIs erstellen und diese Anwendung dann auf vielen verschiedenen Arten von Geräten ausführen.

3.2.4 Unterstützung für Java-Standard-APIs

Neben dem MIDP-Standard werden die folgenden Java-Programmierschnittstellen unterstützt:

- JSR 30: Connected Limited Device Configuration, Version 1.0 (unterstützt auf Geräten mit der Device-Software-Version 4.0 oder niedriger)
- JSR 37: Mobile Information Device Profile, Version 1.0 (unterstützt auf Geräten mit Device-Software-Version 4.0 oder niedriger)
- JSR 75: optionale PDA-Pakete für die Java-ME-Plattform (PDAP); Unterstützung nur für die PIM-APIs (PIM steht hier für Personal Information Manager) und die Dateiverbindungs-API für Java ME (unterstützt auf Geräten mit 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 for Java ME (API für skalierbare 2D-Vektorgrafiken)
- JSR 238: Mobile Internationalization API (API für die mobile Internationalisierung)

3.2.5 Unterstützung für Java-API-Erweiterungen

BlackBerry-Geräte unterstützen außerdem die nachstehenden APIs, die nicht zu den JSR-Standarddefinitionen gehören und einen größeren Funktionsumfang als die MIDP-API-Standardbibliotheken bieten.

Tabelle 3.1: Unterstützung für JAVA-API-Erweiterungen

<i>API</i>	<i>Beschreibung</i>
Benutzeroberflächen-APIs	Mit diesen APIs können Sie Bildschirme, Menüelemente und alle Komponenten der Benutzeroberfläche erstellen.
APIs für die dauerhafte Speicherung von Daten	Mit diesen APIs können Sie benutzerdefinierte Daten lokal innerhalb Ihrer Anwendung speichern.

Tabelle 3.2: Unterstützung für JAVA-API-Erweiterungen

<i>API</i>	<i>Beschreibung</i>
Netzwerk- und E/A-APIs	Mit diesen APIs können Sie Netzwerkverbindungen einrichten sowie Daten von einer serverseitigen Anwendung lesen beziehungsweise auf diese schreiben.
Ereignis-Listener	Mit den Ereignis-Listnern können Sie auf Ereignisse reagieren, die von den Gerätebenutzern oder vom System auf einem BlackBerry-Gerät ausgelöst wurden.
Anwendungsintegrations-APIs	Mit diesen APIs können Sie die Integration in die vorhandenen 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.

3.2.6 BlackBerry-Lösungen

Die Anwender verwenden den BlackBerry Enterprise Server, den BlackBerry Internet Service oder auch beide Dienste auf einem Gerät. Die Unterschiede zwischen dem Enterprise Server und dem 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.

3.2.7 BlackBerry Enterprise Solution (BES)

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

- Datenverschlüsselung und -komprimierung,
- Verwaltungs- und Überwachungsdienstprogramme für BlackBerry-Geräte,
- vereinfachte Bereitstellung von Anwendungen,
- authentifiziertes Gateway für den Intranetzugriff aus einer Java-Applikation heraus.

3.2.8 BlackBerry Internet Service (BIS)

BlackBerry-Benutzer, die keinem Enterprise Server zugeordnet sind, können den BlackBerry Internet Service verwenden.

Der Internet Service bietet einen E-Mail- und Internet-Service, mit dem die Benutzer E-Mail-Nachrichten automatisch senden und unterwegs bequem auf E-Mail-Anhänge und Internetinhalte zugreifen können.

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

3.2.9 Das Mobile Data System (MDS)

Für den Zugriff einer BlackBerry-Applikation auf Ressourcen hinter der Unternehmensfirewall stellt der Enterprise Server das Mobile Data System (MDS) zur Verfügung.

Das MDS umfasst HTTP- und TCP/IP-Proxys für eine Java-Applikation, sodass das BlackBerry-Gerät ohne zusätzliche VPN-Software mit Anwendungs- und Webservern hinter der Firewall kommunizieren kann. Anwendungen, die Daten über den Enterprise Server als Gateway senden, profitieren von der vereinfachten Unternehmenskonnektivität, der Datenverschlüsselung und -komprimierung sowie von der Unabhängigkeit von drahtlosen Netzwerken durch die Enterprise Solution.

Das 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.

3.2.10 Entwicklung von Stand-alone-Anwendungen

Mit den BlackBerry-APIs können Sie eigenständige Anwendungen erstellen, zum Beispiel Spiele oder statische Referenzhandbücher, die sich als Offline-Anwendungen ausführen lassen. Vor dem Kompilieren können Sie die erforderlichen Ressourcen-Daten einer Anwendung hinzufügen.

Die Smartphone-Benutzer können die Anwendung über das drahtlose Netzwerk oder mithilfe der Desktop-Software installieren. Sobald eine Anwendung auf dem Gerät installiert ist, müssen sie für den Betrieb keine Verbindung mit dem drahtlosen Netzwerk oder einem PC oder Server herstellen.

3.2.11 Anwendungen mit Desktop-Synchronisierung

Mit den BlackBerry-APIs können Sie Anwendungen mit Funktionen zur Desktop-Synchronisierung erstellen. Darunter fallen beispielsweise dynamische Referenzhandbücher und Terminplaner-Anwendungen. Der Benutzer verbindet das Gerät mit einem Computer, um so die auf dem Computer befindlichen Daten zu verwalten und zu synchronisieren. Der BlackBerry-Hersteller Research In Motion bietet selbst keine HotSync-Treiber oder andere Module für die direkte Datenbanksynchronisierung an. Sie müssen daher den Synchronisierungscode erstellen, und der Gerätebenutzer muss den Synchronisierungsvorgang manuell starten.

Nach der Installation der Anwendung auf dem Smartphone muss der Benutzer die Daten manuell synchronisieren. Hierzu wird das Gerät über eine serielle Schnittstelle, eine USB-Verbindung oder eine Bluetooth-Verbindung mit dem Computer verbunden.

3.2.12 Anwendungen mit drahtlosem Zugriff, drahtloser Synchronisierung oder drahtlosen Alarmen

Mit den BlackBerry-APIs können Sie Anwendungen erstellen, die Inhalte per Push über das drahtlose Netzwerk an Geräte in Arbeitsumgebungen übertragen, in denen der Enterprise Server verwendet wird.

Die Java-Entwicklungsumgebung enthält APIs, mit denen Sie in Anwendungen Netzwerkverbindungen zu Servern im Internet oder im Intranet eines Unternehmens einrichten können.

3.2.13 MIDlet-Anwendungen

MIDlets sind Smartphone-Anwendungsprogramme, die dem Mobile Information Device Profile (MIDP) folgen. Das MIDlet-Anwendungsmodell ist Teil der MIDP-Spezifikation. Die Hauptklasse eines MIDlet erweitert immer die MIDlet-Klasse und muss Methoden für `startApp()`, `pauseApp()` und `destroyApp()` verwenden.

Vorteile / Nachteile:

- Die MIDlet-Anwendungen lassen sich auf andere Geräte übertragen, die ebenfalls die MIDP-Spezifikation unterstützen.
- MIDlets können nur die Benutzeroberflächen-APIs verwenden, die in der Bibliothek `javax.microedition.lcdui` vorhanden sind.
- Das Modell nimmt an, dass alle Anwendungsprozesse beendet werden, sobald die Anwendungen geschlossen werden.
- MIDlets können beim Einschalten des Geräts nicht automatisch im Hintergrund gestartet werden.

3.2.14 CLDC-Anwendungen

CLDC ist eine Framework-Spezifikation für Java ME. Eine CLDC-Anwendung kann die Application-Klasse erweitern und startet mit einer Standardmethode `main()`.

Die meisten Beispielanwendungen, die in der BlackBerry-Entwicklungsumgebung enthalten sind, verwenden die CLDC-APIs, BlackBerry-APIs und MIDP-APIs. Alle Hauptkomponenten der Device-Software (einschließlich der Nachrichtenliste, der Kontaktliste, des Kalenders und des Browsers) verwenden die CLDC-APIs und die BlackBerry-APIs.

Vorteile / Nachteile:

- CLDC-Anwendungen sind auf andere Plattformen übertragbar.
- CLDC-Anwendungen können keine aktiven Hintergrund-Threads ausführen, nachdem sie geschlossen wurden.
- CLDC-Anwendungen können keine Informationen mit anderen Anwendungen über IPC-APIs austauschen.
- CLDC-Anwendungen können nicht auf Touchscreen-Ereignisse auf einem BlackBerry-Gerät zugreifen.
- CLDC-Anwendungen können keine Daten vom Beschleunigungssensor eines BlackBerry verwenden.
- CLDC-Anwendungen können nicht den BlackBerry-Browser oder BlackBerry-Maps auf der Benutzeroberfläche integrieren.
- CLDC-Anwendungen können das Symbol, das auf der Startseite angezeigt wird, nicht ändern.
- Anwendungen können nicht auf Trackball- und Trackwheel-Ereignisse zugreifen.

3.2.15 API-Steuerung und Codesignaturen

Wenn Sie eine Java-Applikation für BlackBerry-Geräte entwickeln, können Sie nur die öffentlichen Java-APIs nutzen, die in den Javadoc-Dokumenten in der BlackBerry-Entwicklungsumgebung für Java veröffentlicht und dokumentiert sind.

Die Java Virtual Machine auf dem Gerät ist als sogenannte Sandbox ausgelegt, sodass 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 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 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 <http://na.blackberry.com/eng/developers/javaappdev/codekeys.jsp>. Ihr Satz von Codesignierungsschlüsseln geht Ihnen innerhalb von etwa 10 Tagen zu.

3.2.16 Objektmodellierung

Beim MIDlet- oder CLDC-Anwendungsmodell müssen Sie jeweils einen objektorientierten Ansatz verfolgen, wenn Sie Ihre Anwendung für den BlackBerry 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.

3.2.17 Multithreading

Das BlackBerry-Betriebssystem ist ein Multithreading-fähiges Betriebssystem. Dies bedeutet, dass viele Anwendungen und Prozesse gleichzeitig auf dem 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 Gerätebenutzer die Applikation beendet, können die Hintergrund-Threads aktiv bleiben.

3.2.18 Einsatz von Multithreading

Nutzen Sie die Multithreading-Funktionen 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.

3.3 Bewährte Verfahren für das Schreiben einer effizienten BlackBerry Java Application

Beim Schreiben einer effizienten Java-Applikation sind folgende Grundsätze zu beachten:

- Schreiben Sie einen effizienten Code.
- Gehen Sie sparsam mit Objekten um.
- Verringern Sie die Größe des kompilierten Codes.

3.3.1 Schreiben von effizientem Code

Damit eine Java-Applikation für den BlackBerry 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`. Schreiben Sie also den Wert 1 für `true` und den Wert 0 für `false`.
- Legen Sie Klassen als `final` fest.
- Verwenden Sie für Integer-Zahlenwerte den Typ `int` statt `long`.
- Verzichten Sie auf Garbage-Collection-Vorgänge
- Verwenden Sie statische Variablen für Strings.
- Vermeiden Sie den Konstruktor `String (String)`.
- Verwenden Sie in Abhängigkeit von den Anforderungen Ihrer Anwendung einen statischen String, der `final` oder `nonfinal` ist.
- Schreiben Sie effiziente Schleifen.
- Optimieren Sie Unterausdrücke.
- Optimieren Sie Divisionsvorgänge.
- Vermeiden Sie `java.util.Enumeration`.
- Führen Sie Casts mit `instanceof` durch.
- Evaluieren Sie Bedingungen mit `instanceof`.
- Vermeiden Sie die Verwendung von `StringBuffer.append (StringBuffer)`.
- Vermeiden Sie die Rückgabe von `null`.
- Vermeiden Sie die Eingabe von `null` in Methoden.
- Vorsichtsmaßnahmen beim Eingeben von `null` in einen Konstruktor.
- Verwenden Sie `long` für eindeutige Bezeichner.

- Beenden Sie Anwendungen ordnungsgemäß.
- Drucken Sie die Stack-Verfolgung.

Erläuterung:

Lokale Variablen

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

Kurzschrift

Verwenden Sie Kurzschrift für die Auswertung von Bedingungen vom Typ `Boolean`. Der daraus resultierende kompilierte Code ist kürzer.

Codebeispiel:

Quelltext 3:

```
return( boolean_expression );
```

Festlegen von Klassen als »final«

Wenn Sie Codebibliotheken erstellen, kennzeichnen Sie Klassen als `final`, wenn Sie wissen, dass die Entwickler sie nie erweitern werden.

Durch das Stichwort `final` kann der Compiler effizienteren Code generieren.

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

Verwendung von `int` statt `long`

In Java ist `long` ein 64-Bit-Integerwert. Bei BlackBerry-Geräten kommt ein 32-Bit-Prozessor zum Einsatz, weshalb Vorgänge zwei- bis viermal schneller ablaufen, wenn Sie `int` statt `long` verwenden.

Verzichten auf Garbage Collection

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

Verwenden statischer Variablen für Strings

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

Sie können beispielsweise ein String-Objekt wie folgt erstellen:

Quelltext 4:

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

Für diese statische Konstante (gekennzeichnet mit dem Stichwort `final`) wird bei jedem Verwenden der Konstante eine temporäre String-Instanz erstellt. Der Compiler beseitigt "x" und ersetzt es mit dem String "example" im Bytecode, damit die Java Virtual Machine jedes Mal eine Hash-Tabellensuche ausführt, wenn Sie auf "x" verweisen.

Bei einer statischen Variable (ohne Stichwort `final`) wird String dagegen nur einmal erstellt. Die BlackBerry-JVM führt die Hash-Tabellensuche nur dann durch, wenn sie "x" initialisiert, dadurch ist der Zugriff hier schneller.

Quelltext 5:

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

Sie können öffentliche Konstanten verwenden (dies sind die `final`-Felder), Variablen müssen Sie jedoch als privat markieren.

Vermeiden des Konstruktors String(String)

In einer BlackBerry-Java-Applikation ist jeder String in Anführungszeichen eine Instanz der `java.lang.String`-Klasse! Erstellen Sie einen String, ohne den `java.lang.String(String)`-Konstruktor zu verwenden.

Codebeispiel:

Quelltext 6:

```
String str = "abc"; String str = "found " + n + " items";
```

Schreiben effizienter Schleifen

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

Falls die Reihenfolge bei einer 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:

Quelltext 7:

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

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

Codebeispiel:*Quelltext 8:*

```
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 enthält.

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

Codebeispiel:*Quelltext 9:*

```
int = width  $\gg$  1;
```

Verzichten auf `java.util Enumeration`

Verzichten Sie auf die Verwendung von `java.util Enumeration`-Objekten, es sei denn, Sie möchten Daten ausblenden (anders ausgedrückt: eine Auflistung der Daten statt der Daten selbst anzeigen). Das Abfragen eines Vektors oder einer Hash-Tabelle für ein Enumeration-Objekt ist langsam und erzeugt unnötigen Datenmüll.

Wenn ein anderer Thread den Vektor ändern könnte, synchronisieren Sie die Iteration. Die Java Standard Edition verwendet ein Iterator-Objekt für ähnliche Vorgänge, aber Iterator-Objekte sind in Java ME nicht verfügbar.

Codebeispiel:*Quelltext 10:*

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

Durchführen von Casts mit `instanceof`

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

Codebeispiel:*Quelltext 11:*

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

Auswertung von Ausdrücken mit instanceof

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

Codebeispiel:

Quelltext 12:

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

Vermeiden von StringBuffer.append(StringBuffer)

Um einen String-Puffer an einen anderen anzuhängen, sollte Folgendes verwendet werden: `net.rim.device.api.util.StringUtilities.append( StringBuffer dst, StringBuffer src[, int offset, int length ] )`.

Codebeispiel:

Quelltext 13:

```
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 Wert null während des normalen Einsatzes auftritt.
- Der Javadoc Return-Parameter für die Methode ist so festgelegt, dass null ein zulässiger Rückgabewert ist.

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

Vermeiden Sie die Eingabe von null in Methoden:

Geben Sie keine Parameter null in eine API-Methode ein, 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 dem geeigneten Objekt zu.

Wenn eine Klasse mehrere Konstruktoren umfasst, ist bei der Übergabe eines Parameters `null` gegebenenfalls nicht eindeutig ersichtlich, welcher Konstruktor verwendet werden soll. Der Compiler meldet daher einen 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:

Quelltext 14:

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

Verwenden von `long` für eindeutige Bezeichner

Verwenden Sie einen `long`-Bezeichner statt eines `String`-Bezeichners für einmalige Konstanten, etwa GUIDs, Hash-Tabellenschlüssel und Status- oder Kontextbezeichner.⁶

Für Bezeichner, die in der gesamten Applikation einmalig bleiben sollen, verwenden Sie Schlüssel, die von der Anwendung basierend auf einem Hash eines `String` generiert werden. Beziehen Sie in der `String`-Eingabe genug Informationen ein, um den Bezeichner einmalig zu machen. Verwenden Sie zum Beispiel einen vollständigen Paketnamen wie den folgenden:

Quelltext 15:

```
com.rim.samples.docs.helloworld.
```

Ordnungsgemäßes Beenden von Anwendungen

Bevor Sie den folgenden Code:

Quelltext 16:

```
System.exit(int status)
```

aufrufen, nehmen Sie die notwendigen Bereinigungsschritte vor, wie zum Beispiel das Entfernen von Objekten auf dem Laufzeitspeicher, die nicht mehr von Anwendungen benötigt werden.

Drucken der Stack-Verfolgung

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

⁶ Siehe `HashCalculator` (Anhang E)

Codebeispiel:*Quelltext 17:*

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

3.3.2 Umsichtige Verwendung von Objekten

Damit eine BlackBerry-Applikation die Speicherressourcen effizient nutzen kann, beachten Sie die folgenden Anregungen:

- Sind alle Objekte von der Größe der Anwendung her notwendig?
- Kann Ihre Anwendung Objekte, die Primitive darstellen, etwa `long`, `integer` und `boolean`, als Primitive anstatt als Objekte speichern?
- Sind alle permanenten Objekte notwendig?
- Existieren Instanzen von `Vector` und `Hashtable`? Sind diese Instanzen notwendig? Falls ja: Wie viele Objekt-Handles werden nicht in `Vector` oder `Hashtable` verwendet, weil die anfängliche Größe größer ist als die notwendige?
- Wie viele Objekte werden in Ihrer Anwendung erstellt und dann wieder verworfen? Anders gesagt: Wie viele bereichsorientierte Objekte werden in Ihrer Anwendung erstellt?

3.3.3 Verringern der Größe von kompiliertem Code

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

- 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 Codebibliotheken können Sie die Größe des kompilierten Codes mithilfe von geeigneten Zugriffsmodifikationen für Felder und Methoden beträchtlich verringern. Geben Sie Felder nach Möglichkeit als `private` an. Dies ist nicht nur gute Codepraxis, sondern ermöglicht dem Compiler auch, die COD-Datei zu optimieren. Verwenden Sie nach Möglichkeit den Standardzugriff (`package`) statt des öffentlichen Zugriffs (lassen Sie die Stichwörter `public` und `protected` aus).

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 auszubilden, 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:

Quelltext 18:

```
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` sind mit dem numerischen Wert `0` initialisiert.
- Objekte vom Typ `boolean` werden mit `false` initialisiert.
- Lokale Variablen müssen explizit in einer Methode initialisiert werden.

Codebeispiel:

Quelltext 19:

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

Importieren einzelner Klassen

Bei einer Applikation, die nur wenige Klassen aus einem Paket verwendet, sollten die einzelnen Klassen importiert werden.

Codebeispiel:

Quelltext 20:

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

3.4 Unterstützung mehrerer Sprachen

Die integrierte Entwicklungsumgebung für den BlackBerry enthält eine Ressourcenschnittstelle für das Erstellen von String-Ressourcen.

Die Lokalisierungs-API ist Teil des `net.rim.device.api.i18n`-Pakets. Die MIDP-Anwendungen unterstützen keine Lokalisierung.

Die Entwicklungsumgebung speichert Ressourcen für ein Gebietsschema in einem `ResourceBundle`-Objekt. Ein `ResourceBundleFamily`-Objekt 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.

Sie können 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 ein String in einem Gebietsschema nicht definiert ist, wird ein String aus dem nächstgelegenen Gebietsschema verwendet.

3.4.1 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 Anweisungen:

- Speichern Sie die Textzeichenfolgen für jedes Gebietsschema 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 Gerät je nach dem Gebietsschema des Benutzers angezeigt werden soll.

3.5 Multimedia-Unterstützung: Audiounterstützung

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

Weitere Informationen zum Audiosupport auf einem BlackBerry-Gerät finden Sie unter der nachstehend genannten Webadresse⁷.

3.6 Kameraunterstützung

Wenn ein Benutzer ein Bild mit der integrierten Kamera aufnimmt, speichert das Gerät das Bild im internen Dateisystem. Eine Geräteanwendung kann auf die Bilder mithilfe der Dateiverbindungs-API für Java ME zugreifen, die in der Version 4.2 der Entwicklungsumgebung oder höher zur Verfügung steht. Die Geräteanwendung kann die Kameraanwendung aufrufen und Ereignisse abfragen, wenn Bilder dem Dateisystem hinzugefügt werden.

Auf einem Gerät mit der Betriebssystemversion 4.7 oder höher können Sie einem Benutzer erlauben, ein Bild an eine BlackBerry-Anwendung zu senden. Wenn ein Benutzer die Kameraanwendung verwendet, um ein Bild aufzunehmen, oder wenn er über den Datei-Explorer auf ein Bild zugreift, fügt das Betriebssystem der Kameraanwendung ein entsprechendes Menüelement hinzu oder zeigt einen Dialog im Datei-Explorer an.

Wenn ein Benutzer dieses Menüelement oder den Dialog auswählt, sendet das Betriebssystem eine Anforderung an eine Geräteanwendung, die als Content-Handler für dieses Bild fungiert. Die Geräteanwendung kann die Anforderung akzeptieren und das Bild öffnen.

Damit eine Geräteanwendung ein Bild empfangen kann, muss die Anwendung die `javax.microedition.content.ContentHandler`-Schnittstelle implementieren, um sich als Content-Handler bei Bildtypen wie zum Beispiel JPG zu registrieren. Die Registrierung als Content-Handler verknüpft die Geräteanwendung mit dem Bildtyp und ermöglicht anderen BlackBerry-Geräteanwendungen, diese Dateien an die BlackBerry-Geräteanwendung zu senden. Weitere Informationen zur Verwendung der Content-Handler-Schnittstelle finden Sie unter der nachstehenden Webadresse⁸.

Sie können eine Geräteanwendung erstellen, die ein Objekt vom Typ `javax.microedition.media.Player` und Unterklassen der Klasse `javax.microedition.media.Control` verwendet, um ein Video wiederzugeben. Eine BlackBerry-Anwendung ruft eine der `net.rim.device.api.ui.Manager.createPlayer()`-Methoden auf, um ein Player-Objekt und ein Video von unterschiedlichen Quellen, wie beispielsweise einer Datei oder einem Stream, abzurufen.

Ein Player-Objekt ermöglicht einem Gerätebenutzer, die Anwendung als Media-Player zu nutzen und Aktionen wie etwa das Abspielen und Anhalten der Wiedergabe des Videos auszuführen.

⁷ Audiosupport www.blackberry.com/developers

⁸ Informationen zur Content-Handler-Schnittstelle <http://jcp.org/en/jsr/detail?id=211>

Die BlackBerry-Anwendung ruft `Player.realize()` auf, um der Anwendung zu erlauben, Ressourcen abzurufen, die sie zur Wiedergabe eines Videos benötigt, wie zum Beispiel den Speicherort einer Videodatei. Die Anwendung ruft die Methode `Player.prefetch()` auf, um ein Video für die Wiedergabe vorzubereiten. Sie kann die Methode `Player.getControls()` aufrufen, um ein Array an Control-Objekten abzurufen, die die von einem Player unterstützten Vorgänge oder Funktionen darstellen. Beispielsweise ermöglicht es ein `javax.microedition.media.control.VolumeControl`-Objekt einer Anwendung, die Lautstärke zu regeln, mit der ein Video abgespielt wird. Ein `javax.microedition.media.control.VideoControl`-Objekt ermöglicht es einer BlackBerry-Anwendung zu steuern, wie sie ein Video wiedergibt.

Nachdem die Anwendung Informationen zum von ihr unterstützten `javax.microedition.media.Control`-Objekt abrufen kann, kann die Anwendung die Methode `Player.getControl()` aufrufen, indem sie einen String mit dem Namen des Steuerelements als Parameter verwendet, um ein bestimmtes Control-Objekt abzurufen, so etwa:

Quelltext 21:

```
VideoControl myVC = (VideoControl) Player.getControl("<VideoControl>").
```

Die Anwendung kann die Eigenschaften des Control-Objekts durch Aufrufen der Methoden des Objekts festlegen. Die Applikation ruft `Player.start()` auf, um mit der Wiedergabe eines Videos zu beginnen, und `Player.close()`, um die Wiedergabe eines Videos zu beenden.

Sie können zum Beispiel eine Anwendung erstellen, die den Sucher der Kamera zum Aufnehmen eines Videos als `.3gp`-Datei oder zum Streamen des Videos an ein `java.io.ByteArrayOutputStream`-Objekt verwendet. Die Anwendung ruft `Manager.createPlayer("capture://video")` auf, um ein Player-Objekt abzurufen, das die Schnittstellen `VideoControl` und `RecordControl` unterstützt.

Die Anwendung ruft `Player.getControl("RecordControl")` auf und wandelt den Rückgabewert in ein `RecordControl`-Objekt um. Ein `RecordControl`-Objekt ermöglicht es, eine Geräteanwendung zu steuern, wie sie ein Video aufnimmt. Das Aufrufen von `RecordControl.setRecordLocation ("file:///<filename>")` legt den Speicherort der `.3gp`-Datei fest.

Das Aufrufen von `RecordControl.setRecordStream (ByteArrayOutputStream output)` legt den `ByteArrayOutputStream` fest, zu dem das Streaming stattfinden soll.

Die Anwendung ruft `RecordControl.startRecord()`, gefolgt von `Player.start()` auf, um mit der Aufzeichnung eines Videos zu beginnen. Um das Video zu speichern und die Aufzeichnung zu beenden, ruft die Applikation `RecordControl.commit()`, gefolgt von `Player.stop()` auf, um das Video zu speichern und die Aufzeichnung zu beenden.

Sie können eine Geräteanwendung erstellen, die ein Video im BlackBerry-Browser wiedergeben kann. Die Applikation ruft dazu `net.rim.blackberry.api.browser.Browser.getDefaultSession()` auf, um ein `net.rim.blackberry.api.browser`.

BrowserSession-Objekt abzurufen. Das BrowserSession-Objekt ermöglicht der Anwendung die Interaktion mit dem Browser.

Die BlackBerry-Anwendung ruft `BrowserSession.displayPage()` auf, indem sie einen String als Parameter mit dem Format `"file:/// <locationOfVideo>"` verwendet, wobei `<locationOfVideo>` der Speicherort des Videos ist. `BrowserSession.displayPage()` zeigt ein Video im Browser an. Sie können eine Geräteanwendung erstellen, die ein Video wiedergeben kann, für das DRM-Lizenzen (Digital Rights Management)⁹ erforderlich sind.

BlackBerry-Anwendungen verwenden die Klasse `net.rim.device.api.drm.DRMManager`, um DRM-Lizenzen für ein Video zu installieren. Ein Gerät verfügt über ein `DRMManager`-Objekt. Das Aufrufen von `DRMManager.getLicenseChallenge(String contentURL, String[] permissions)` ruft ein Byte-Array ab, das Informationen zu DRM-Lizenzen enthält, die die Anwendung für die Wiedergabe eines Videos erhalten und installieren muss. Das Aufrufen von `DRMManager.installRights(String contentURL, byte[] rightsData, RMRightsInstallCallback callback)` installiert die Lizenzen für ein Video.

Sie können einem Gerätebenutzer erlauben, ein Video an eine Anwendung zu senden. Damit die BlackBerry-Geräteanwendung die Videodatei empfangen und verwenden kann, muss die Anwendung die Schnittstelle `javax.microedition.content.ContentHandler` implementieren, um sich als Content-Handler zu registrieren.

Wenn ein Benutzer über die Kameraanwendung oder den Datei-Explorer auf eine Videodatei zugreift, erscheint ein Menüelement, das es dem Anwender ermöglicht, die Videodatei an Applikationen, die als Content-Handler fungieren, zu senden.

3.7 Benutzeroberflächen- und Navigationsdesign

3.7.1 Eingabe und Navigation für Benutzer

BlackBerry-Geräte enthalten eine Tastatur, ein Trackwheel oder einen Trackball sowie eine Escape-Taste für die Eingabe und die Navigation.

Eine Java-Applikation für BlackBerry-Geräte sollte dem nachstehenden Modell für Eingabe und Navigation so weit wie möglich folgen:

- Das Klicken auf das Trackwheel oder den Trackball oder den Bildschirm öffnet in der Regel ein Menü.
- Durch Drücken der Escape-Taste können Aktionen abgebrochen werden. Weiterhin ist damit die Rückkehr zum vorherigen Bildschirm möglich. Durch Drücken der `[Esc]`-Taste werden Benutzer immer wieder zur Startseite zurückkehren. Durch Halten der `[Esc]`-Taste wird der Browser oder die Medienanwendung geschlossen.

⁹ Spezifiziert nach OMA (OpenMobileAlliance) http://www.openmobilealliance.org/document/PR_German/OpenMobileAlliance_LeadershipPR20080521_German.pdf

Standardmäßig bieten die BlackBerry-Bildschirmobjekte diese Funktionen ohne weitere Anpassung. Sie müssen allerdings die gewünschten Menüelemente und die zusätzliche Benutzeroberflächen- und Navigationslogik hinzufügen.

3.7.2 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 sendet.

Die BlackBerry-Hardware 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 Java-Applikation 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.

3.7.3 Trackball-Bewegung

Mit der Trackball-API filtern Sie die Trackball-Bewegungsdaten, die die Hardware an die Softwareschicht sendet. Die Trackball-API siebt Bewegungen, Rauschen oder unerwünschte Bewegungen aus.

Mit der Trackball-API können Sie auch Einstellungen ändern, zum Beispiel 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 Gerätehardware, solange der Benutzer den Trackball fortlaufend rollt.

Die Trackball-Empfindlichkeit steigt vorübergehend an, wenn der Benutzer den Trackball längere Zeit ununterbrochen rollt.

3.7.4 Trackwheel

Geräte, die vor der Pearl 8100 Series erschienen sind, sind nicht mit einem Trackball ausgestattet. Stattdessen weisen sie an der rechten Geräteseite ein Trackwheel auf. Das Trackwheel ist hier die primäre Steuerung für die Benutzernavigation.

Benutzer haben damit die folgenden Möglichkeiten:

- Das Trackwheel rollen, um den Cursor in vertikaler Richtung zu bewegen.
- Das 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.

3.7.5 Richtlinien

Erkunden Sie Modelle sowohl mit der Trackwheel- als auch mit Trackball-Navigation oder dem Touchscreen, und stellen Sie sicher, dass Ihre Anwendung auf allen Gerätarten gut arbeitet.

3.8 Erstellen einer UI, die mit normalen Benutzeroberflächen kompatibel ist

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

Die Benutzeroberflächen-APIs sind eine Bibliothek von UI-Komponenten, die standardmäßige Layouts und Verhaltensweisen umfassen.

- 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 Gerätebildschirm anzuordnen, zum Beispiel horizontal, vertikal oder von links nach rechts.

Mit den Benutzeroberflächen-APIs können Sie Benutzeroberflächen mit Tabellen, Rastern und anderen speziellen Funktionen erstellen. Die Java-Entwicklungsumgebung nutzt ein standardmäßiges Java-Ereignismodell, um bestimmte Arten von Ereignissen zu empfangen und darauf zu antworten.

Anwendungen können Ereignisse von Gerätebenutzern (etwa wenn der Benutzer auf das Trackwheel oder den Trackball klickt oder etwas über die Tastatur eingibt) sowie Systemereignisse erhalten und darauf reagieren, beispielsweise durch globale Warnungen, Echtzeit-Taktänderungen oder Verbindungen über den USB-Anschluss.

3.9 Speicherverwaltung

Die Java Virtual Machine (JVM) des BlackBerry verwaltet die Speicherauslastung auf dem Gerät. Die JVM ordnet Speicher zu, führt die Garbage Collection durch und tauscht automatisch Daten zwischen SRAM und Flash-Speicher aus.

Die JVM muss außerdem den verfügbaren Speicher zwischen den übrigen Anwendungen auf dem Gerät und Ihrer Java-Applikation 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-Geräte enthalten die folgenden Arten von Speicher:

3.9.1 Arbeitsspeicher-Beschreibung

Das BlackBerry-Betriebssystem und alle Anwendungsmodule sind im Flash-Speicher enthalten. Wenn ein Benutzer das Gerät einschaltet, belegen das Kernbetriebssystem und die Java-Applikationsmodule 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 Java-Applikation im Speicher ablegt.

3.9.2 SRAM

Das SRAM steuert die flüchtigen Datenobjekte und die Runtime-Prozesse.

3.9.3 Erweiterbare MicroSD-Speicherkarte

Auf einer MicroSD-Karte können Mediendateien, Dokumente und Daten aus einer BlackBerry-Applikation gespeichert werden.

3.9.4 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 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 bestimmt die 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 bestimmt die Anzahl von Objekt-Handles im System.

3.10 Minimieren der Speicherbelegung

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

- Verwenden Sie primitive Typen (z. B. `int` oder `boolean`) anstelle von Objekten.
- 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 Datenverarbeitung auf den Server aus. Filtern oder sortieren Sie die Daten beispielsweise, bevor Sie sie an das BlackBerry-Gerät senden.

3.10.1 Verwalten von geringem verfügbarem Speicher

Der Manager für geringen Speicherplatz verwaltet die Speicherressourcen auf dem Gerät, wenn die verfügbaren Ressourcen unter einen bestimmten Schwellenwert fallen.

Der Manager für geringen Speicherplatz versucht, belegten Speicher wieder freizugeben und damit mehr Speicher verfügbar zu machen. Alle Anwendungen, einschließlich Ihrer Java-Applikationen, sollten mit dem Manager für geringen Speicherplatz zusammenarbeiten, um damit so viel Speicher wie möglich freizugeben, wenn das Gerät nur noch eine geringe Speicherreserve besitzt.

3.10.2 Erkennen von wenig verfügbarem Speicher

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

- Die Menge des verfügbaren Flash-Speichers 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 und 800 KB.
- Die Anzahl der verfügbaren permanenten Objekt-Handles sinkt unter 1000.
- Die Anzahl der verfügbaren Objekt-Handles sinkt unter 1000.

3.11 Schonen der Ressourcen

3.11.1 Verwendung einer effizienten Datenstrukturauswahl

Die Datenstrukturauswahl definiert, wie viele Objekt-Handles und wie viel Flash-Speicher eine BlackBerry-Applikation nutzt. Durch eine falsche Datenstrukturauswahl

werden wichtige Ressourcen belegt, ohne den Funktionsumfang der Applikation oder die Arbeitsumgebung für die Benutzer zu verbessern.

Befolgen Sie die folgenden Ratschläge:

- Die Datenstruktur sollte aus der kleinstmöglichen Anzahl von Objekten bestehen, insbesondere, wenn Sie Objekte auf höheren Ebenen verwenden, wie Vector oder Hashtable. 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 Gerät belegt werden. Ein Array mit Primitiven ist ein Objekt und verwendet ein Objekt-Handle.
- String-Objekte sind ebenso effizient wie Byte-Arrays. Ein String-Objekt verwendet nur ein Objekt-Handle und ist gleichwertig, wenn Ihre Anwendung jedes Zeichen als ein Byte speichert. Anders gesagt: Der Wert jedes Zeichens ist kleiner oder gleich dem Dezimalwert 255. Wenn Ihre Anwendung die Zeichen nicht als ein Byte speichern kann, können Sie die Zeichen als String speichern, weil diese Methode dem Speichern eines char-Arrays entspricht.

3.11.2 Konsolidieren von Objekten zu Objektgruppen

Einer der häufigsten Fehler, denen Anwendungsentwickler begegnen, ist der Mangel an permanenten Objekt-Handles. Die Menge des Flash-Speichers 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 verwendet ein permanentes Objekt-Handle und ein Objekt-Handle. Ein vergängliches Objekt verwendet nur ein Objekt-Handle. Ein Datensatz, der zehn String-Felder enthält, die für Elemente wie Name, Telefonnummer und Adresse stehen, verwendet beispielsweise elf permanente Objekt-Handles, eines für das Datensatzobjekt und eines für jeden String.

Wenn in einer BlackBerry-Applikation 3000 Datensätze gespeichert sind, nutzt die Anwendung 33.000 permanente Objekt-Handles, was die Anzahl der verfügbaren permanenten Objekt-Handles auf einem Gerät mit 16 MB Flash-Speicher übersteigt.

Mit der `net.rim.device.api.system.ObjectGroup`-Klasse 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 ein permanentes Objekt-Handle (statt wie bisher elf). Die Objekt-Handles für die String-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 ein Handle zu. Deshalb sollten Sie die Gruppierung von Objekten nur dann aufheben, wenn dies unbedingt nötig ist.

3.12 Garbage Collection auf einem BlackBerry-Gerät

3.12.1 Garbage Collection für den Arbeitsspeicher

Die Java Virtual Machine startet nur dann einen Speicherbereinigungsvorgang für den Arbeitsspeicher, wenn die 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.

3.12.2 Vollständige Garbage Collection

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 gegebenenfalls einen vollständigen Garbage-Collection-Vorgang:
 - Die JVM 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 JVM kann kein neues Objekt zuordnen, weil keine Objekt-Handles verfügbar sind.
 - Das Gerät befindet sich im Leerlauf.

3.12.3 Garbage Collection bei Leerlauf

Der Garbage-Collection-Vorgang findet nicht jedes Mal statt, wenn sich das Gerät im Leerlauf befindet. Dieser Vorgang wird nur dann gestartet, wenn das System eine Speicherbereinigung als förderlich für die optimale Systemleistung und die maximale Akkuleistung erkennt.

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

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

3.13 Datenverwaltung

Das Gerät bietet APIs zum Speichern von Daten in den permanenten Speicher. Die Persistent-Store-APIs und die MIDP-RMS-APIs (RMS steht hier für Record Management System, die APIs bieten Unterstützung für JSR 37 und JSR 118) sind auf allen Java-gestützten Modellen verfügbar.

Geräte, auf denen die Device-Software Version 4.2 oder höher ausgeführt wird, bieten ein traditionelles Dateisystem und unterstützen das Speichern von Inhalten direkt im Dateisystem mithilfe von JSR-75-APIs. Sowohl mit den Persistent-Store-APIs als auch mit den MIDP-RMS-APIs können Sie Daten permanent im Flash-Speicher ablegen.

Die permanent gespeicherten Daten bleiben selbst dann erhalten, wenn Sie den Akku aus dem BlackBerry-Gerät herausnehmen.

3.13.1 APIs für die dauerhafte Speicherung

Die APIs für die dauerhafte Speicherung bieten eine flexible und stabile Schnittstelle für die Datenspeicherung.

Mit den 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.

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

3.13.2 MIDP Record Management System APIs

Die RMS-APIs bieten ein einfaches Datensatzverwaltungssystem, mit dem Sie ein Datenspeicherobjekt erstellen und eine Reihe von Datensätzen innerhalb dieses Objekts beibehalten können.

Jeder Datensatz ist ein Byte-Array, weshalb Sie Ihre Daten zuerst in einem Byte-Array-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 auf diese Weise 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.

3.14 Dateiverbindungs-APIs

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

3.14.1 Zugriff auf Daten auf der MicroSD-Medienkarte

Das `javax.microedition.io.file`-Paket 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 etwa auf Klingeltöne und Kamerabilder für BlackBerry-Geräte zugreifen.

- `ConnectionClosedException`: Diese Ausnahme wird ausgelöst, wenn eine Anwendung eine Methode für eine geschlossene Dateiverbindung aufruft.

- `FileConnection`: Eine Anwendung kann mit dieser API auf Dateien oder Verzeichnisse zugreifen.
- `FileSystemListener`: Eine Anwendung kann mit dieser API den Empfang von Statusbenachrichtigungen aktivieren, wenn die Anwendung einen Dateisystemstamm hinzufügt oder entfernt.
- `FileSystemRegistry`: Eine Anwendung kann diese API als zentrale Registrierung für Dateisystem-Listener verwenden, die auf das Hinzufügen oder Entfernen von Dateisystemen achten.
- `IllegalModeException`: 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.

3.15 Sichern und Synchronisieren von Daten

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

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

Um Daten mit Remote-Datenquellen zu synchronisieren, müssen Sie die Synchronisierungslogik in Ihre Applikation integrieren. Bei den meisten Anwendungen werden Daten an serverseitige Anwendungen mithilfe des normalen HTTP- oder TCP/IP-Protokolls über das drahtlose Netzwerk beziehungsweise 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 Java-Applikation für den BlackBerry kann mithilfe der Desktop-Synchronisierungs-APIs und des Desktop Managers 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 Desktop Manager vom Client lesen kann. Der Gerätebenutzer muss die Synchronisierung manuell starten.

Hierzu muss das Add-In für den Desktop Manager ausgeführt werden, mit dem die Anwendung auf dem Gerät angewiesen wird, die Daten an die Applikation auf dem Computer zu senden. Alternativ können Sie auch Daten über die systemeigenen USB-Protokolle in die Anwendung auf dem Computer schreiben.

4 Beispielcodes I

Ich möchte Ihnen hier in einer Reihe von Beispielcodes Anregungen geben, die Sie in Ihre Projekte einfließen lassen können.

In diesem Abschnitt haben wir es öfter mit Feldern zu tun. Im Folgenden möchte ich Ihnen daher einen kurzen Abriss zu den Feldern geben, mit denen Sie es im Laufe der Programmierung zu tun haben werden.

4.1 Wissenswertes über Felder

Ein Feld ist ein Container, der dazu dient, verschiedene Datentypen im User-Interface zu halten. Ein Beispiel: Ein Date-Feld hat die Aufgabe, ein Datum zu speichern und in unterschiedlichen Formaten anzuzeigen, wie zum Beispiel 2010/01/19. Das BlackBerry User Interface unterstützt den Entwickler dabei, die Funktionen zu nutzen, um ein Feld zu erzeugen.

Das nun folgende Codebeispiel stammt von einem sogenannten Custom Text Field. Dieses Feld kann verschiedene nicht editierbare und separate Stücke von Text enthalten. Jedes Textsegment ist genau an die Puffergröße gebunden.

Quellcode: UI-Felder

```
class MultiElementTextField extends Field {
    // wir müssen das Feld erweitern
    String[] _texts; // ein Array von Textsegmenten
    int[] _spaces; // ein Array, der den verfügbaren Speicherplatz
                  // enthält, der jedem Textsegment zugewiesen ist
    int_buffer; // Ein Integerwert, der den Puffer zwischen jedem
               // Textsegment repräsentiert
    //
    /* Der Konstruktor legt jeden Datenarray und den Puffertwert fest */
    public MultiElementTextField(Object[] data, int buffer, long style)
        throws IllegalArgumentException {
        super(style); // Aufruf von super, dies legt den
                    // Stil des Feldes fest.
        int n = data.length;
        if ((n % 2) != 0) {
            throw new IllegalArgumentException("Invalid number of elements");
        } else {
            _texts = new String[n / 2];
            _spaces = new int[n / 2];
        }
    }
}
```

```

        for (int i = 0; i < n ; i = i + 2) {
            _texts[i / 2] = (String)data[i];
            _spaces[i / 2] = ((Integer)data[i+1]).intValue();
        }
        _buffer = buffer;
    }

    public boolean isFocusable() {
        // Dieses Feld kann den Fokus erhalten, Rückgabewert true
        return true;
    }

    /* Wir benutzen die Methode layout, um die Ausdehnung des Feldes
       festzulegen, und rufen dann setExtent(int, int) für die Gesamtgröße
       des Feldes auf. */
    public void layout(int width, int height) {
        Font font = getFont();
        int _fieldHeight = font.getHeight();
        int _fieldWidth = 0;
        int n = _spaces.length;
        for (int i = 0; i < n; ++i) {
            _fieldWidth += _spaces[i];
        }
        if (i < n - 1) {
            _fieldWidth += _buffer;
        }
        setExtent(_fieldWidth, _fieldHeight);
    }

    /* Die Methode paint zeichnet die Textobjekte auf den Bildschirm, indem
       sie das übergebene Grafikobjekt benutzt. Hier verwenden wir die Werte
       von _buffer und _spaces[], um die Position und Größe jedes
       Textsegments zu definieren. Wir benutzen den Stil ELLIPSIS, um Text
       zu behandeln, der zu lang für den zugewiesenen Platz ist.
       Überzählige Zeichen werden durch ... ersetzt. */
    public void paint(Graphics graphics) {
        int xpos = 0;
        int n = _spaces.length;
        for (int i = 0; i < n; ++i) {
            graphics.drawText(_texts[i], xpos, 0, graphics.ELLIPSIS, _spaces[i]);
            xpos += _spaces[i];
            if (i < n - 1) {
                xpos += _buffer;
            }
        }
    }
}

```

4.2 User Interface

Ich stelle hier ein Programm vor, das das User Interface (UI) »Manager« benutzt. Der Manager reagiert diagonal über die Felder und bestimmt dadurch den natürlichen Fokus während der Eingabe. Hier kommen Techniken zum Einsatz, wie eine UI-Komponente erzeugt wird und wie der UI-Manager auf Ereignisse reagiert.

Als Erstes werden die dafür notwendigen Packages geladen.

Quelltext: Laden der Packages für den UI-Manager

```
import net.rim.device.api.ui.*;
import net.rim.device.api.ui.component.*;
import net.rim.device.api.ui.container.*;
```

Quelltext: UI-Manager

```
import net.rim.device.api.ui.*;
import net.rim.device.api.ui.component.*;
import net.rim.device.api.ui.container.*;

// Verwendung des User Interface Managers
class DiagonalApp extends UiApplication
{
    public static void main(String[] args)
    {
        UI_Manager app = new UI_Manager();
        app.enterEventDispatcher();
    }

    public UI_Manager()
    {
        super();

        MainScreen screen = new MainScreen();
        screen.setTitle("Hier sehen Sie den UI Manager.");

        screen.add(new RichTextField("Feld oberhalb des UI Managers"));

        // hier erhält das erste Feld den Fokus,
        // das zweite Argument wird hierbei auf false gesetzt.
        FeldManager manager = new FeldManager(0, true);
        manager.add(new RichTextField(
            "Erstes Feld in 3 lines",
            RichTextField.USE_TEXT_WIDTH ) );
        manager.add(new RichTextField(
            "Zweites Feld hat den Fokus nicht",
            RichTextField.USE_TEXT_WIDTH | Field.NON_FOCUSABLE ) );
        manager.add(new RichTextField(
            "Letztes feld",
```

```

        RichTextField.USE_TEXT_WIDTH ) );
        screen.add(manager);

        screen.add(new RichTextField("Das Feld neben des UI Managers"));

        pushScreen (screen);
    }
}

```

4.3 Auswahlfelder

Ich möchte hier ein kleines Programm vorstellen, das mit einem Auswahlfeld (Choice-Field) arbeitet. Als Beispiel habe ich mir die Geschwindigkeiten »Mensch«, »Hurricane« und »Licht« ausgesucht, die auf dem Screen angezeigt werden. Klickt der User auf das jeweilige Objekt, dann werden ihm weitere Geschwindigkeitsinformationen geliefert. Es sind drei Wahlfelder, zu denen der User Informationen bekommt.

Zum Beispiel möchte er wissen, wie schnell die Lichtgeschwindigkeit in Meter pro Sekunde ist.

Als Erstes müssen folgende Packages in unser Projekt geladen werden:

Quelltext: Packages für das Programm Auswahlfelder

```

import net.rim.device.api.ui.*;
import net.rim.device.api.ui.component.*;
import net.rim.device.api.ui.container.*;
import net.rim.device.api.system.*;
import net.rim.device.api.i18n.*;

```

Quelltext: Choices

```

/*
 * Auswahl mit MenuItems
 *
 */

import net.rim.device.api.ui.*;
import net.rim.device.api.ui.component.*;
import net.rim.device.api.ui.container.*;
import net.rim.device.api.system.*;
import net.rim.device.api.i18n.*;

public class Geschwindigkeit extends UiApplication
{
    public static void main(String[] args)
    {
        //Wir erzeugen hier eine neue Instanz der Applikation
    }
}

```

```

        //anschließend starten wir die App in einem sog. Event Thread
        //Der Event Thread soll auf Klicks des Users reagieren, um
        //weiterführende
        //Informationen aus dem Hauptelement darzustellen.
        Geschwindigkeit app = new Geschwindigkeit();
        app.enterEventDispatcher();
    }

    public Geschwindigkeit()
    {
        //Hier wird ein neuer Screen angezeigt...
        pushScreen (new GeschwindigkeitInfo());
    }
}

//Wir erzeugen einen neuen Screen, der den Hauptscreen erweitert, das ist ein
//default Standard-Verhalten für alle BlackBerry-Applikationen
final class GeschwindigkeitInfo extends MainScreen
{
    //Hier deklarieren wir Variablen, die wir später verwenden wollen...
    private GeschwindigkeitInfoScreen _GeschwindigkeitInfoScreen;
    private ObjectAuswahlFeld choiceField;
    private int auswahl;

    public GeschwindigkeitInfoScreen()
    {
        //Hier wird der Hauptscreen aktiviert...
        super();

        //Wir geben dem Hauptscreen nun einen Titel...
        LabelField title = new LabelField("Geschwindigkeiten im
        Weltraum", LabelField.ELLIPSIS | LabelField.USE_ALL_WIDTH);
        setTitle(title);

        //Nun erzeugen wir die Textlabel...
        add(new RichTextField("Objekte und deren Geschwindigkeiten"));

        //Hier erzeugen wir nun eine Drop-Down-Liste mit drei
        // Auswahlmöglichkeiten:
        //Objekt, Geschwindigkeit km/h, Geschwindigkeit m/s
        String choices[] = {"Objekt", "km/h", "m/s"};
        choiceField = new ObjectChoiceField("Wählen Sie ein Objekt",
        choices);
        add(choiceField);
    }

    //Beim Verlassen der Applikation soll eine Dialogbox erscheinen
    public boolean onClose()

```



```

        {
            Dialog.alert("Ich hoffe das Programm hat Ihnen gefallen");
            System.exit(0);
            return true;
        }

        //Hier erzeugen wir ein MenuItem.
        //Wenn der User darauf klickt, wird ihm mehr Information geboten,
        //je nachdem, was er gerade gewählt hat...
        private MenuItem _viewItem = new MenuItem("Mehr Informationen", 110,
10) {

            public void run() {

                //Speichert das ausgewählte Element in einer Variable
                auswahl = choiceField.getSelectedIndex();

                //Erzeuge einen neuen Screen mit den Informationen,
                // die der User gewählt hat...
                //
                _infoScreen = new GeschwindigkeitInfoScreen();
                UiApplication.getUiApplication().pushScreen
(_infoScreen);
            }
        };

        //Erzeuge hier ein MenuItem, das dem User die Möglichkeit gibt, die
        //App zu beenden
        private MenuItem _closeItem = new MenuItem("Beenden", 200000, 10) {
            public void run()
            {
                onClose();
            }
        };

        //Überschreibe das makeMenu, um ein neues MenuItem hinzuzufügen
        protected void makeMenu( Menu menu, int instance )
        {
            menu.add(_viewItem);
            menu.add(_closeItem);
        }

        //Hier wird nun eine innere Klasse erzeugt, um den neuen Screen
        //anzuzeigen. Anzeigen des gewählten Hauptmenueeintrags.
        private class InfoScreen extends MainScreen
        {
            public InfoScreen() {
                super();
            }
        }

```

```

//Hier können weitere Felder hinzugefügt werden, die
// zu dem Haupteintrag gehören...
//Zum Beispiel Informationen über km/h und m/s
LabelField lf = new LabelField();
BasicEditField kmField = new BasicEditField("km/h: ",
null, 20, Field.READONLY);
BasicEditField msField = new BasicEditField("m/h: ",
null, 20, Field.READONLY);

add(lf);
add(new SeparatorField());
add(kmField);
add(msField);

//Hier wird nun der Text angezeigt, je nachdem, was
// der User gewählt hat...
//(Werte des selektierten Elements werden in der
// Variablen 'auswahl' gespeichert...)
if (auswahl == 0) {
    lf.setText("Mensch");
    kmField.setText("5 km/h");
    msField.setText("1,39 m/s");

} else if (select == 1) {
    lf.setText("Hurricane");
    kmField.setText("290 km/h");
    msField.setText("80,5 m/s");

} else if (select == 2) {
    lf.setText("Licht");
    kmField.setText("1.079.252.849 km/h");
    msField.setText("299.792.458 m/s");

}

}

}

```

4.4 Checkbox

Eine Checkbox gibt Ihnen schnell einen Überblick, was ausgewählt wurde. Ein kleines Beispiel kennen Sie von Ihrem Handheld. Unter *Verbindungen* sehen Sie, ob das Mobilfunknetz aktiv ist oder alternativ Bluetooth.

In diesem Beispiel zeige ich Ihnen zunächst, wie eine einfache Checkbox programmiert wird. Eine Checkbox hat einen Status. Entweder ist sie »checked« (markiert) oder

»unchecked« (unmarkiert). Eine Prozedur überprüft den Status der Box und gibt ihn wieder aus.

Quelltext: Checkbox Beispiel

```
/*
 * CheckBox
 *
 */

import net.rim.device.api.ui.*;
import net.rim.device.api.ui.component.*;
import net.rim.device.api.ui.container.*;
import net.rim.device.api.system.*;
import java.util.Vector;

class CheckboxListField extends UiApplication implements ListFieldCallback
{
    //Hier werden die Variablen deklariert...
    private static final String[] _elements = {"Erstes Auswahlelement",
        "ZweitesAuswahlelement ", "Drittes Auswahlelement", "Viertes
        Auswahlelement", "Fünftes Auswahlelement", "Sechstes Auswahlelement"};
    private Vector _listData = new Vector();
    private ListField _checkList;
    private MenuItem _toggleItem;

    //Hier eine Klasse, die den Status der Checkbox prüft (checked oder
    //unchecked).
    private class ChecklistData
    {
        private String _stringVal;
        private boolean _checked;

        ChecklistData()
        {
            _stringVal = "";
            _checked = false;
        }

        ChecklistData(String stringVal, boolean checked)
        {
            _stringVal = stringVal;
            _checked = checked;
        }

        //Hier erfolgt das Setzen mit der "Get/set"-Methode.
        private String getStringVal()
        {

```

```

        return _stringValue;
    }

    private boolean isChecked()
    {
        return _checked;
    }

    private void setStringValue(String stringValue)
    {
        _stringValue = stringValue;
    }

    private void setChecked(boolean checked)
    {
        _checked = checked;
    }

    //Überprüfen des checked Status erfolgt in dieser Routine.
    private void toggleChecked()
    {
        _checked = !_checked;
    }
}

public static void main(String args[])
{
    CheckboxListField theApp = new CheckboxListField();
    theApp.enterEventDispatcher();
}

CheckboxListField()
{
    //So, im MenuItem wurde ein neuer Eintrag hinzugefügt, wenn das
    //_checkList Feld den Fokus bekommt...
    //Das folgende MenuItem schaltet den Status zwischen
    // "checked/unchecked" des ausgewählten Elements.
    _toggleItem = new MenuItem ("Options Menu", 200, 10)
    {
        public void run()
        {
            //Lese den Index aus dem gewählten Item.
            int index = _checkList.getSelectedIndex();

            //Lese nun die "ChecklistData" aus...
            ChecklistData data = (ChecklistData)_listData.
elementAt(index);

```

```

        //Schalte den Status um...
        data.toggleChecked();

        //Aktualisiere den Vektor mit den neuen Checklistdaten...
        _listData.setElementAt(data, index);

        //Setze den alten Eintrag der Box ausser Kraft...
        _checkList.invalidate(index);
    }
};

MainScreen myScreen = new MainScreen()
{
    //Überschreibe die Methode "makeMenu", damit wir einen Eintrag
    // für unser Menue hinzufügen können...
    //sobald die checkbox den Fokus bekommt...
    protected void makeMenu(Menu menu, int instance)
    {
        Field focus = UiApplication.getUiApplication().
getActiveScreen().getLeafFieldWithFocus();
        if(focus == _checkList)
        {
            //So, die _checkList Instanz hat nun den Fokus bekommen...
            //Füge nun den Menueeintrag hinzu...
            menu.add(_toggleItem);
        }

        //Erzeuge nun das Standard Menue...
        super.makeMenu(menu, instance);
    }
};

//Nun wird ein Titel dem Screen hinzugefügt...
myScreen.setTitle(new LabelField("Meine Checkboxen"));

_checkList = new ListField()
{
    //Füge nun eine kleine Pause ein, damit der Status geschaltet
    // werden kann...mit der Leertaste...
    protected boolean keyChar(char key, int status, int time)
    {
        boolean retVal = false;

        //Wurde die Leertaste gedrückt??
        if (key == Characters.SPACE)
        {
            //Lese nun den Index der ausgewählten Zeile...

```

```

        int index = getSelectedIndex();

        //Lese nun die Daten für diese Zeile...
        ChecklistData data = (ChecklistData)_listData.
elementAt(index);

        //Schalte nun den Status...
        data.toggleChecked();

        //Aktualisiere den Vektor mit den neuen Daten...
        _listData.setElementAt(data, index);

        //Setze wieder den alten Eintrag außer Kraft...
        invalidate(index);

        //Alles Ok?
        retVal = true;
    }
    return retVal;
}
};

//Setze nun den Rückruf der Listbox...
_checkList.setCallback(this);

int elementLength = _elements.length;

//Fülle nun das Listfield und den Vektor mit Daten.
for(int count = 0; count < elementLength; ++count)
{
    //So, die Boxen werden nun "checked", oder "unchecked".
    if (count % 2 == 0)
    {
        _listData.addElement(new ChecklistData(_elements[count],
true));
    }
    else
    {
        _listData.addElement(new ChecklistData(_elements[count],
false));
    }
    _checkList.insert(count);
}

//Füge nun das ListField zu dem Screen...
myScreen.add(_checkList);

//Führe einen pushScreen aus, um die Anzeige zu aktualisieren...

```

```

        pushScreen (myScreen);

    }

    //Hier ist nun die Callback Routine...

    //Zeichne die Listreihe neu...
    public void drawListRow(ListField list, Graphics graphics, int index,
int y, int w)
    {
        //Lese die ChecklistDaten für die aktuelle Reihe...
        ChecklistData currentRow = (ChecklistData)this.get(list, index);

        StringBuffer rowString = new StringBuffer();

        //Ist sie "checked", dann zeichne eine checked Box,
        //sonst zeichne sie "unchecked"
        if (currentRow.isChecked())
        {
            rowString.append(Characters.BALLOT_BOX_WITH_CHECK);
        }
        else
        {
            rowString.append(Characters.BALLOT_BOX);
        }

        //Füge noch einige Leerzeichen ein...
        rowString.append(Characters.SPACE);
        rowString.append(Characters.SPACE);
        rowString.append(currentRow.getStringVal());

        //Zeichne nun den Text...
        graphics.drawText(rowString.toString(), 0, y, 0, w);
    }

    //Gebe das Objekt zurueck, das den Index hat...
    public Object get(ListField list, int index)
    {
        return _listData.elementAt(index);
    }

    //Gebe den ersten Treffer aus, starte mit dem Suchen bei dem Index...
    //
    public int indexOfList(ListField list, String p, int s)
    {
        //Rückgabe des Elements...
        return _listData.indexOf(p, s);
    }

```

```
//Rückgabe der Breite...  
public int getPreferredWidth(ListField list)  
{  
    return Graphics.getScreenWidth();  
}  
}
```

4.5 DayCalc

In dem folgenden Beispiel möchte ich Ihnen zeigen, wie Sie mit zwei Eingabefeldern rechnen können. Das Programm DayCalc soll Ihnen die Differenz zweier Kalender-Daten anzeigen.



Abbildung 4.1: Das Programm DayCalc im Blackberry-Simulator

Quellcode: DayCalc

```

package com.ems.daycalc;

import net.rim.device.api.ui.UiApplication;

public class DayCalc extends UiApplication {

    /**
     * Einhängepunkt der Anwendung. Dieser Codeblock
     * wird aufgerufen, wenn die App gestartet wird.
     * @param args
     */
    public static void main(String[] args) {
        DayCalc calc = new DayCalc();
        //Dadurch wird der DayCalcScreen im ScreenStack an die
        //oberste Stelle gepusht (wo er letztlich am Display erscheint)
        UiApplication.getUiApplication().pushScreen (new DayCalcScreen());

        //Mit dieser Zeile wird die Anwendung im
        //Blackberry initialisiert
        calc.enterEventDispatcher();
    }
}

```

Quellcode:Screen JavaCode DayCalc

```

/*
 * DayCalc
 *
 */

package com.ems.daycalc;

import net.rim.device.api.ui.Field;
import net.rim.device.api.ui.FieldChangeListener;
import net.rim.device.api.ui.UiApplication;
import net.rim.device.api.ui.component.ButtonField;
import net.rim.device.api.ui.component.DateField;
import net.rim.device.api.ui.component.Dialog;
import net.rim.device.api.ui.component.LabelField;
import net.rim.device.api.ui.container.MainScreen;
import net.rim.device.api.ui.container.VerticalFieldManager;

/**
 * Der {@link DayCalcScreen} dient zum einen als Unterklasse von {@link
 * MainScreen} zur

```

```

* Darstellung der ganzen Komponenten (Buttons etc) auf dem Bildschirm
*
* Gleichzeitig sorgt er als Implementierung von {@link FieldChangeListener}
* für die Bearbeitung von Klicks auf Komponenten (hier z.B. der Button)
* @author Raphael Thierjung
*
*/
public class DayCalcScreen extends MainScreen implements
FieldChangeListener{

    /**
     * Feld zur Einstellung des Startdatums
     */
    private DateField startDate;
    /**
     * Feld zur Einstellung des Enddatums
     */
    private DateField endDate;
    /**
     * Button, auf dessen Klick die Bearbeitung beginnen sollte
     */
    private ButtonField button;

    /**
     * Konstruktor: Initialisiert die Klasse
     */
    public DayCalcScreen(){
        //Durch den super() Aufruf werden die Initialisierungsmaßnahmen
        //der Oberklasse angestoßen
        super();
        //Ist eine Methode der Klasse und initialisiert den Screen
        initializeComponents();
    }

    /**
     * Initialisiert den Screen und belegt ihn mit den Komponenten
     */
    private void initializeComponents() {
        //Setzt den Title der Anwendung. Dieser ist ganz oben abzulesen
        setTitle(new LabelField("Day Calculation Utility"));
        //Der VerticalFieldManager ordnet alle Komponenten Vertikal
        //(übereinander) an
        VerticalFieldManager vfm = new VerticalFieldManager();
        //Fügt ein LabelField (zum Anzeigen einfachem Text) dem Manager hinzu
        vfm.add(new LabelField("Specify your starting date :"));
        //Initialisiert das Feld für das Startdatum
        startDate = new DateField();
    }
}

```

```

//Fügt das Feld dem Manager hinzu
vfm.add(startDate);
//Fügt ein weiteres LabelField dem Manager hinzu
vfm.add(new LabelField("Specify your ending date : "));
//Initialisiert das Feld für das Enddatum
endDate = new DateField();
//und fügt dieses dem Manager hinzu
vfm.add(endDate);
//Initialisiert den Button und gibt ihn den Schriftzug Calculate
button = new ButtonField("Calculate");
//Setzt den Changelistener des Buttons mit einem Verweis auf diese
//Klasse
//Wird auf den Button geklickt, so wird die Methode fieldChanged
//aufgerufen
button.setChangeListener(this);
//Fügt den Button dem Fieldmanager hinzu
vfm.add(button);
//Setzt den Fieldmanager (und damit alle Komponenten, die wir ihn vorher
//hinzugefügt haben) auf den Screen (und damit sichtbar)
this.add(vfm);
}

/**
 * Wird aufgerufen, sobald ein Field, auf den dieser {@link
 * FieldChangeListener} hört, ausgelöst wird
 */
public void fieldChanged(Field field, int arg1) {
    //Prüft, ob der Klick wirklich vom Button kommt
    if(field.equals(button)){
        //holt sich das Startdatum als long-Wert (in ms seit 1.1.1970)
        long begin = startDate.getDate();
        //holt sich das Enddatum als long-Wert (in ms seit 1.1.1970)
        long end = endDate.getDate();
        //Sorgt für die Ausführung einer Runnable innerhalb des
        //UI-Threads. Damit wird verhindert, dass verschiedene Threads
        //gleichzeitig auf dem Bildschirm malen, was zu Komplikationen führen
        //kann
        UiApplication.getUiApplication().invokeLater(new Runnable(){
            //Diese Methode wird im Thread abgearbeitet
            public void run() {
                //Sorgt für die Anzeige eines Dialog-Fenster
                Dialog.inform("The solution of the Calculation is:\n" +
                    "Hier könnte Ihr Ergebnis stehen ;)");
            }
        });
    }
}

```

```

}

/**
 * Verhindert die Anzeige des Save Prompts
 */
public boolean onSavePrompt() {
    return true;
}

/**
 * Sorgt dafür, dass bei Klick auf das Display kein Menü dargestellt wird
 */
protected boolean invokeAction(int action) {
    boolean handled = super.invokeAction(action);
    if (!handled) {
        switch (action) {
            case ACTION_INVOKE: // Trackball click.
            {
                return true;
            }
        }
    }
    return handled;
}
}

```

4.6 CustomButtons

Das folgende Listing besteht aus mehreren kleineren Applikationsteilen, die dazu dienen, einen selbst gestalteten Picture-Button zu realisieren.

Quellcode: Packages, die eingebunden werden

```

import net.rim.device.api.ui.*;
import net.rim.device.api.system.*;

```

Quellcode: PictureBackgroundButton

```

/*
 * Bild auf einem Button
 *
 */

package com.rim.samples.device.custombutton;

import net.rim.device.api.ui.*;
import net.rim.device.api.system.*;

```

```

class PictureBackgroundButtonField extends Field
{
    private String _label;
    private int _labelHeight;
    private int _labelWidth;
    private Font _font;

    private Bitmap _currentPicture;
    private Bitmap _onPicture = Bitmap.getBitmapResource("img/pink_scribble.
bmp");
    private Bitmap _offPicture = Bitmap.getBitmapResource("img/blue_
scribble.bmp");

    /**
     * Konstruktor.
     * @param text - the text to be displayed on the button
     * @param style - combination of field style bits to specify display
        attributes
     */
    PictureBackgroundButtonField(String text, long style)
    {
        super(style);

        _font = getFont();
        _label = text;
        _labelHeight = _font.getHeight();
        _labelWidth = _font.getAdvance(_label);
        _currentPicture = _onPicture;
    }

    //
    // Bei Return: Schreibe den Text auf den Button
    //
    String getText()
    {
        return _label;
    }

    //
    // Feld-Einbindung
    //
    public int getPreferredHeight()
    {
        return _labelHeight + 4;
    }

    //

```

```
// Field einbinden.
//
public int getPreferredWidth()
{
    return _labelWidth + 8;
}

//
// Ändern des Bildes bei onFocus.
//
protected void onFocus(int direction)
{
    _currentPicture = _onPicture;
    invalidate();
}

//
// Ändere das Bild, wenn das Bild nicht den Fokus besitzt
//
protected void onUnfocus()
{
    _currentPicture = _offPicture;
    invalidate();
}

//
// Zeichne das Bild...
//
protected void drawFocus(Graphics graphics, boolean on)
{
    // nop
}

//
// Definiere das Layout
//
protected void layout(int width, int height)
{
    setExtent(Math.min( width, getPreferredWidth()),
        Math.min( height, getPreferredHeight()));
}

//
// Neuzeichnen mit Paint...
//
protected void paint(Graphics graphics)
{
    // Als erstes zeichne den Hintergrund und dann das Bild...
```

```

        graphics.setColor(Color.LIGHTPINK);
        graphics.fillRect(0, 0, getWidth(), getHeight());
        graphics.drawBitmap(0, 0, getWidth(), getHeight(), _currentPicture,
0, 0);

        // Nun zeichne den Text...
        graphics.setColor(Color.BLACK);
        graphics.setFont(_font);
        graphics.drawText(_label, 4, 2,
            (int)( getStyle() & DrawStyle.ELLIPSIS | DrawStyle.HALIGN_MASK ),
            getWidth() - 6 );
    }

    protected boolean navigationClick(int status, int time)
    {
        fieldChangeNotify(1);
        return true;
    }
}

```

Nun folgt der zweite Programmteil. Er zeigt uns die Verwendung der UI-Felder. Zunächst binden wir die Komponenten ein.

Quellcode: Import der Packages

```

import net.rim.device.api.ui.*;
import net.rim.device.api.ui.component.*;
import net.rim.device.api.ui.container.*;

```

Quellcode: CustomButtonsDemo.java

```

/*
 * CustomButton
 *
 */

package com.rim.samples.device.custombutton;

import net.rim.device.api.ui.*;
import net.rim.device.api.ui.component.*;
import net.rim.device.api.ui.container.*;

class CustomButtonsDemo extends UiApplication implements FieldChangeListener
{

    //
    // Einsprungspunkt.
    //
    public static void main(String[] args)

```

```

{
    CustomButtonsDemo theApp = new CustomButtonsDemo();
    theApp.enterEventDispatcher();
}

//
// Hier ist der Standard-Konstruktor.
// Er bringt die Applikation auf den Hauptschirm.
//
CustomButtonsDemo()
{
    CustomButtonField rectangle;
    CustomButtonField triangle;
    CustomButtonField octagon;
    CustomButtonField fixedWidth1;
    CustomButtonField fixedWidth2;
    CustomButtonField fullscreen;
    CustomButtonField colour;
    PictureBackgroundButtonField picture;

    // Der Hauptschirm ist hier...
    MainScreen mainScreen = new MainScreen();

    // Hinzufügen eines Titels des Hauptschirms. Wir verwenden ein
    // einfaches LabelField hier. Die Ellipse option bricht das Label
    // ab und füllt den Text mit "...", falls der Text zu lang ist
    //
    mainScreen.setTitle(new LabelField("Benutzerdefinierter Button" ,
LabelField.ELLIPSIS | LabelField.USE_ALL_WIDTH));

    // Hinzufügen eines vertikalen Feldmanagers mit den userdefinierten
    // Feldern...
    VerticalFieldManager vfm = new VerticalFieldManager();

    // Rechteckiger Button...
    vfm.add(new RichTextField(Field.NON_FOCUSABLE));
    rectangle = new CustomButtonField("Rechteck", CustomButtonField.
RECTANGLE, Field.FOCUSABLE);
    rectangle.setChangeListener(this);
    vfm.add(rectangle);

    // Dreieckiger Button...
    vfm.add(new RichTextField(Field.NON_FOCUSABLE));
    triangle = new CustomButtonField("Dreieck", CustomButtonField.
TRIANGLE, Field.FOCUSABLE);
    triangle.setChangeListener(this);
    vfm.add(triangle);

```



```

        // Achteckiger Button
        vfm.add(new RichTextField(Field.NON_FOCUSABLE));
        octagon = new CustomButtonField("Achteck", CustomButtonField.
OCTAGON, Field.FOCUSABLE);
        octagon.setChangeListener(this);
        vfm.add(octagon);

        // Die nächsten beiden Buttons zeigen, wie die Breite fest definiert
        // wird...
        vfm.add(new RichTextField(Field.NON_FOCUSABLE));
        fixedWidth1 = new CustomButtonField("Feste Breite",
CustomButtonField.FIXED_WIDTH, Field.FOCUSABLE);
        fixedWidth1.setChangeListener(this);
        vfm.add(fixedWidth1);

        vfm.add(new RichTextField(Field.NON_FOCUSABLE));
        fixedWidth2 = new CustomButtonField("Feste Breite: Long!",
CustomButtonField.FIXED_WIDTH, Field.FOCUSABLE);
        fixedWidth2.setChangeListener(this);
        vfm.add(fixedWidth2);

        // Button mit automatischer Breitenanpassung width of the screen.
        vfm.add(new RichTextField(Field.NON_FOCUSABLE));
        fullscreen = new CustomButtonField("Vollbild", CustomButtonField.
FULLSCREEN, Field.FOCUSABLE);
        fullscreen.setChangeListener(this);
        vfm.add(fullscreen);

        // Button mit einem farbigen Hintergrund
        vfm.add(new RichTextField(Field.NON_FOCUSABLE));
        colour = new CustomButtonField("Colour", CustomButtonField.COLOUR_
BACKGROUND, Field.FOCUSABLE);
        colour.setChangeListener(this);
        vfm.add(colour);

        // Button mit einem Bild als Hintergrund. Das Bild ändert sich,
        // wenn der Knopf den Fokus bekommt
        vfm.add(new RichTextField(Field.NON_FOCUSABLE));
        picture = new PictureBackgroundButtonField("Bild", Field.FOCUSABLE);
        picture.setChangeListener(this);
        vfm.add(picture);

        mainScreen.add(vfm);

        // So, die Konstruktion ist fertig. Nun rufen wir pushScreen auf, um
        // den Haupt-Screen zu aktivieren...
        pushScreen (mainScreen);
    }

```

```
//
// hier ein Popup, das uns informiert, welchen Knopf wir gedrückt
// hatten...
public void fieldChanged(Field field, int context)
{

    String text = "Button";

    if (field instanceof CustomButtonField) {
        text = ((CustomButtonField)field).getText();
    }

    else if (field instanceof PictureBackgroundButtonField) {
        text = ((PictureBackgroundButtonField)field).getText();
    }

    Dialog.inform(text + " wurde gedruickt.");

}
}
```

Und hier der letzte Teil dieser Programmreihe. Er zeigt uns die Verwendung von unterschiedlichen Formen, wie ein UI-Feld erzeugt werden kann.

Quellcode: CustomButtonField

```
/*
 * CustomButtonField
 *
 */

package com.rim.samples.device.custombuttons;

import net.rim.device.api.ui.*;
import net.rim.device.api.system.*;

//
// Hier die Klasse, die verschiedene Schaltknöpfe erzeugt...
//
class CustomButtonField extends Field implements DrawStyle
{
    static final int RECTANGLE = 1;
    static final int TRIANGLE = 2;
    static final int OCTAGON = 3;
    static final int FIXED_WIDTH = 4;
    static final int FULLSCREEN = 5;
    static final int COLOUR_BACKGROUND = 4;
```

```
private String _label;
private int _shape;
private Font _font;
private int _labelHeight;
private int _labelWidth;

//
//Erzeuge einen Knopf mit spezifiziertem Label...
//
CustomButtonField(String label)
{
    this(label, RECTANGLE, 0);
}

//
//Erzeuge einen Knopf mit spezifiziertem Label...
//
CustomButtonField(String label, int shape)
{
    this(label, shape, 0);
}

//
//Erzeuge einen Knopf mit spezifiziertem Label...
//
CustomButtonField(String label, long style)
{
    this(label, RECTANGLE, style);
}

//
//Erzeuge einen Knopf mit spezifiziertem Label...
//
CustomButtonField(String label, int shape, long style)
{
    super(style);

    _label = label;
    _shape = shape;
    _font = getFont();
    _labelHeight = _font.getHeight();
    _labelWidth = _font.getAdvance(_label);
}

//
// Bei Return bitte den Text auf den Knopf...
//
```

```
String getText()
{
    return _label;
}

public int getPreferredWidth()
{
    switch(_shape)
    {
        case TRIANGLE:
            if (_labelWidth < _labelHeight)
            {
                return _labelHeight << 2;
            }
            else
            {
                return _labelWidth << 1;
            }

        case OCTAGON:
            if (_labelWidth < _labelHeight)
            {
                return _labelHeight + 4;
            }
            else
            {
                return _labelWidth + 8;
            }

        case FIXED_WIDTH:
            return (_font.getAdvance(" ")) * 35;
            //Setze 35 Leerzeichen dahinter...

        case FULLSCREEN:
            return Display.getWidth();

        default:
            return _labelWidth + 8;
    }
}

public int getPreferredHeight()
{
    switch(_shape)
    {
```

```

        case TRIANGLE:
            if (_labelWidth < _labelHeight)
            {
                return _labelHeight << 1;
            }
            else
            {
                return _labelWidth;
            }

        case OCTAGON:
            return getPreferredWidth();

        default:
            return _labelHeight + 4;
    }
}

protected void drawFocus(Graphics graphics, boolean on)
{
    switch(_shape)
    {
        case TRIANGLE:
            int w = getWidth();
            int h = w >> 1;

            for (int i=h-1; i>=2; --i)
            {
                graphics.invert(i, h - i, w - (i << 1), 1);
            }

            break;

        case OCTAGON:
            int x3 = getWidth();
            int x = 5 * x3 / 17;
            int x2 = x3 - x;
            x3 = x3 - 1;
            x2 = x2 - 1;

            graphics.invert(1, x, getWidth() - 2, x2 - x + 1);

            for (int i=1; i<x; ++i)
            {
                graphics.invert(1+i, x-i, getWidth() - ((i+1)<<1), 1);
                graphics.invert(1+i, x2+i, getWidth() - ((i+1)<<1), 1);
            }
    }
}

```

```

        break;

        default:
            graphics.invert(1, 1, getWidth() - 2, getHeight() - 2);
            break;
    }
}

protected void layout(int width, int height)
{
    // Die zwischengespeicherte Schriftart aktualisieren,
    // falls sie verändert wurde.
    _font = getFont();
    _labelHeight = _font.getHeight();
    _labelWidth = _font.getAdvance(_label);

    // Berechne Breite...
    width = Math.min( width, getPreferredWidth() );

    // Berechne Höhe...
    height = Math.min( height, getPreferredHeight() );

    // Setze nun die Abmessungen...
    setExtent( width, height );
}

protected void paint(Graphics graphics)
{
    int textX, textY, textWidth;
    int w = getWidth();
    int h = getHeight();

    switch(_shape)
    {
        case TRIANGLE:
            h = (w>>1);
            int m = (w>>1)-1;

            graphics.drawLine(0, h-1, m, 0);
            graphics.drawLine(m, 0, w-1, h-1);
            graphics.drawLine(0, h-1, w-1, h-1);

            textWidth = Math.min(_labelWidth,h);
            textX = (w - textWidth) >> 1;
            textY = h >> 1;
            break;

        case OCTAGON:

```

```

        int x = 5*w/17;
        int x2 = w-x-1;
        int x3 = w-1;

        graphics.drawLine(0, x, 0, x2);
        graphics.drawLine(x3, x, x3, x2);
        graphics.drawLine(x, 0, x2, 0);
        graphics.drawLine(x, x3, x2, x3);
        graphics.drawLine(0, x, x, 0);
        graphics.drawLine(0, x2, x, x3);
        graphics.drawLine(x2, 0, x3, x);
        graphics.drawLine(x2, x3, x3, x2);

        textWidth = Math.min(_labelWidth, w - 6);
        textX = (w-textWidth) >> 1;
        textY = (w-_labelHeight) >> 1;
        break;

    case COLOUR_BACKGROUND:
        graphics.setColor(Color.LIGHTBLUE);
        graphics.fillRect(0, 0, w, h);
        graphics.setColor(Color.BLACK);

        textX = 4;
        textY = 2;
        textWidth = w - 6;
        break;

    default:
        graphics.drawRect(0, 0, w, h);

        textX = 4;
        textY = 2;
        textWidth = w - 6;
        break;
    }
    graphics.drawText(_label, textX, textY, (int)(getStyle() &
DrawStyle.ELLIPSIS | DrawStyle.HALIGN_MASK ), textWidth );
}

protected boolean navigationClick(int status, int time)
{
    fieldChangeNotify(0);
    return true;
}
}

```

4.7 TabControl

Im folgenden Programm beschreibe ich, wie sich ein TabControl auf dem Screen realisieren lässt. Tabs sind mit den Reitern von Karteikarten vergleichbar.

Wir benötigen die folgenden Packages:

Quelltext: Realisierung eines TabControls

```
import net.rim.device.api.ui.Field;
import net.rim.device.api.ui.FocusChangeListener;
import net.rim.device.api.ui.UiApplication;
import net.rim.device.api.ui.component.BasicEditField;
import net.rim.device.api.ui.component.LabelField;
import net.rim.device.api.ui.component.SeparatorField;
import net.rim.device.api.ui.container.HorizontalFieldManager;
import net.rim.device.api.ui.container.MainScreen;
import net.rim.device.api.ui.container.VerticalFieldManager;
```

Im folgenden Quellcode zeige ich Ihnen die Realisierung eines TabControls mit drei Reitern, zwischen denen Sie wechseln können.

Quellcode: TabControl

```
//
// TabControl über 3 Seiten
//

package com.rim.samples.tabcontrol;

import net.rim.device.api.ui.Field;
import net.rim.device.api.ui.FocusChangeListener;
import net.rim.device.api.ui.UiApplication;
import net.rim.device.api.ui.component.BasicEditField;
import net.rim.device.api.ui.component.LabelField;
import net.rim.device.api.ui.component.SeparatorField;
import net.rim.device.api.ui.container.HorizontalFieldManager;
import net.rim.device.api.ui.container.MainScreen;
import net.rim.device.api.ui.container.VerticalFieldManager;

public class TabControl extends UiApplication {

    public TabControl() {
        TabControlScreen screen = new TabControlScreen();
        pushScreen (screen);
    }

    //
    // Parameter werden mit args übergeben...
    //

    public static void main(String[] args) {
        TabControl app = new TabControl();
```



```

    app.enterEventDispatcher();
}

private class TabControlScreen extends MainScreen implements
FocusChangeListener {

    private LabelField tab1;

    private LabelField tab2;

    private LabelField tab3;

    private LabelField spacer1;

    private LabelField spacer2;

    private VerticalFieldManager tabArea;

    private LabelField tab1Heading;

    private BasicEditField tab1Field1;

    private BasicEditField tab1Field2;

    private LabelField tab2Heading;

    private BasicEditField tab2Field1;

    private BasicEditField tab2Field2;

    private LabelField tab3Heading;

    private BasicEditField tab3Field1;

    private BasicEditField tab3Field2;

    private VerticalFieldManager tab1Manager;
    private VerticalFieldManager tab2Manager;
    private VerticalFieldManager tab3Manager;

    public TabControlScreen() {
        HorizontalFieldManager hManager = new HorizontalFieldManager();
        tab1 = new LabelField("Seite 1", LabelField.FOCUSABLE | LabelField.
HIGHLIGHT_SELECT);
        tab2 = new LabelField("Seite 2", LabelField.FOCUSABLE | LabelField.
HIGHLIGHT_SELECT);
        tab3 = new LabelField("Seite 3", LabelField.FOCUSABLE | LabelField.
HIGHLIGHT_SELECT);
        spacer1 = new LabelField(" | ", LabelField.NON_FOCUSABLE);
        spacer2 = new LabelField(" | ", LabelField.NON_FOCUSABLE);

        tab1.setFocusListener(this);
        tab2.setFocusListener(this);

```

```

        tab3.setFocusListener(this);
        hManager.add(tab1);
        hManager.add(spacer1);
        hManager.add(tab2);
        hManager.add(spacer2);
        hManager.add(tab3);

        add(hManager);
        add(new SeparatorField());

        tab1Manager = new VerticalFieldManager();
        tab2Manager = new VerticalFieldManager();
        tab3Manager = new VerticalFieldManager();

        tabArea = displayTab1();
        add(tabArea);
    }

    public void focusChanged(Field field, int eventType) {
        if (tabArea != null) {
            if (eventType == FOCUS_GAINED) {
                if (field == tab1) {
                    System.out.println("Wechsel zur Seite 1");
                    delete(tabArea);
                    tabArea = displayTab1();
                    add(tabArea);
                } else if (field == tab2) {
                    System.out.println("Wechsel zur Seite 2");
                    System.out.println("Wechsel zur Seite 1");
                    delete(tabArea);
                    tabArea = displayTab2();
                    add(tabArea);
                } else if (field == tab3) {
                    System.out.println("Wechsel zur Seite 3");
                    System.out.println("Wechsel zur Seite 1");
                    delete(tabArea);
                    tabArea = displayTab3();
                    add(tabArea);
                }
            }
        }
    }

    public VerticalFieldManager displayTab1() {
        if (tab1Heading == null) {
            tab1Heading = new LabelField("Registration");
            tab1Manager.add(tab1Heading);

```

```

    }
    if (tab1Field1 == null) {
        tab1Field1 = new BasicEditField("Username: ", "");
        tab1Manager.add(tab1Field1);
    }
    if (tab1Field2 == null) {
        tab1Field2 = new BasicEditField("Password: ", "");
        tab1Manager.add(tab1Field2);
    }
    return tab1Manager;
}

public VerticalFieldManager displayTab2() {
    if (tab2Heading == null) {
        tab2Heading = new LabelField("Password Zuruecksetzen");
        tab2Manager.add(tab2Heading);
    }
    if (tab2Field1 == null) {
        tab2Field1 = new BasicEditField("Sicherheitsfrage: ", "Wie heisst
Ihr Lieblingstier?");
        tab2Manager.add(tab2Field1);
    }
    if (tab2Field2 == null) {
        tab2Field2 = new BasicEditField("Password: ", "");
        tab2Manager.add(tab2Field2);
    }
    return tab2Manager;
}

public VerticalFieldManager displayTab3() {
    if (tab3Heading == null) {
        tab3Heading = new LabelField("Interressen");
        tab3Manager.add(tab3Heading);
    }
    if (tab3Field1 == null) {
        tab3Field1 = new BasicEditField("Hobbies: ", "");
        tab3Manager.add(tab3Field1);
    }
    if (tab3Field2 == null) {
        tab3Field2 = new BasicEditField("Mitgliedschaften: ", "");
        tab3Manager.add(tab3Field2);
    }
    return tab3Manager;
}

}
}

```

5 Beispielcodes II

Die Beispiellistings in diesem Kapitel zeigen Ihnen in erster Linie, wie Sie Elemente der Benutzeroberfläche ansprechen und in Ihre Blackberry-Applikationen integrieren. Das letzte Beispiel dieses Abschnitts fällt dann ein wenig aus dem Rahmen. Es widmet sich weniger der Gestaltung der UI, sondern einer Kernanwendung des BlackBerry, nämlich dem Mail-Management.

5.1 Screen-Dekor

Die Version 4.6.0 der Blackberry-API beinhaltet ein Set an Dekor-APIs, die es dem Entwickler erlauben, das Aussehen seiner Applikationen einfach zu verändern. Hierzu zählen die Änderungen der Screen-Hintergründe und die Erzeugung von Rahmen um den Screen-Manager herum. Hier ein Ausschnitt, der einen Screen mit grünem Hintergrund und grauer Umrandung erzeugt.

Quelltext: Änderung des Screen-Dekors

```
class DecorTestScreen extends MainScreen {

    private VerticalFieldManager _manager;

    DecorTestScreen() {
        _manager = (VerticalFieldManager)getMainManager();

        Background bg = BackgroundFactory.createSolidBackground(Color.GREEN);
        _manager.setBackground(bg);

        XYEdges edges = new XYEdges(20, 20, 20, 20);
        Border border = BorderFactory.createBevelBorder(edges);
        _manager.setBorder(border);

        _manager.add(new LabelField("test"));
        _manager.add(new LabelField("test 1"));
        _manager.add(new LabelField("test 3"));

    }
}
```

5.2 Nullfelder

Die Methode `MainScreen.setStatus` ermöglicht es dem Entwickler, Informationen in einer Statuszeile im unteren Bereich des Screens anzuzeigen. Wenn der Status gesetzt wurde, kann sie über den Aufruf `NullField` gelöscht werden. Hier ein Codebeispiel, das uns anhand eines `LabelField` zeigt, wie das Feld gelöscht werden kann.

Quelltext: Löschen des Statusbereichs

```
//Instanziere den MainScreen
MainScreen myScreen = new MainScreen();
//Hinzufügen eines Labels in dem Statusbereich
mainScreen.setStatus(new LabelField("Hallo User"));
//Lösche nun den Statusbereich
mainScreen.setStatus(new NullField());
```

5.3 Dialogboxen

Das folgende Beispiel zeigt, wie sich Dialogboxen anpassen lassen. Hierzu benötigen wir die Komponente `net.rim.device.api.ui.component.Dialog`.

Ein angepasster Dialog erlaubt es dem User, spezielle Informationen während der Eingabe abzufragen und im Programmverlauf zu verarbeiten.

Das Beispiel verwendet ein `EditField` (Eingabefeld), welches Benutzereingaben entgegennimmt und diese dann auf dem Haupt-Screen darstellt.

Quellcode: Angepasster Dialog EditField

```
//
//  Angepasster Dialog
//

package com.samples.dialogDataEntry;

import net.rim.device.api.ui.*;
import net.rim.device.api.ui.component.*;
import net.rim.device.api.ui.container.*;
import net.rim.device.api.system.*;

public final class CustomDialog extends Dialog
{
    private EditField entryField;

    public CustomDialog()
    {
        super(Dialog.D_OK, "Angepasster Dialog", 1,
            Bitmap.getPredefinedBitmap(Bitmap.EXCLAMATION), Manager.FOCUSABLE);
```

```
entryField = new EditField
    ("Geben Sie etwas Text ein: ", "", 50, EditField.EDITABLE);
add(entryField);

}

public boolean keyChar(char key, int status, int time)
{
    //Überschreibe die Tastatureingaben

    switch (key)
    {
        case Characters.ENTER:
            //Aktualisiere die Haupt-Klasse mit dem eingegebenen Text...
            DialogDataEntry.setText(entryField.getText());
            this.close();
            break;
            //Überschreibe die Backspace-Taste, um das letzte Zeichen zu
            //entfernen...
        case Characters.BACKSPACE:
            if (entryField.getTextLength() > 0)
            {
                entryField.setText(entryField.getText().substring
                    (0, entryField.getTextLength() - 1));
            }
            break;
            //Überschreibe die Escape-Taste, um das letzte Zeichen zu
            //entfernen...
        case Characters.ESCAPE:
            if (entryField.getTextLength() > 0)
            {
                entryField.setText(entryField.getText().substring
                    (0, entryField.getTextLength() - 1));
            }
            break;
            //Eingabe der Zeichenkette des Users in das EditField...
        default:
            entryField.setText(entryField.getText() + key);
            break;
    }
}

//
//  Hauptklasse
//
package com.samples.dialogDataEntry;

import net.rim.device.api.ui.*;
```

```

import net.rim.device.api.ui.component.*;
import net.rim.device.api.ui.container.*;
import net.rim.device.api.system.*;

class DialogDataEntry extends UiApplication
{
    private MainScreen mainScreen;
    private static RichTextField enteredText;

    public static void main(String[] args)
    {
        DialogDataEntry theApp = new DialogDataEntry();
        theApp.enterEventDispatcher();
    }

    public DialogDataEntry()
    {
        mainScreen = new MainScreen();

        enteredText = new RichTextField("", RichTextField.NON_FOCUSABLE);

        LabelField title = new LabelField("Dialog: Dateneingabe",
            LabelField.ELLIPSIS | LabelField.USE_ALL_WIDTH);

        mainScreen.setTitle(title);

        mainScreen.add(enteredText);

        //Anzeigen auf dem Haupt-Screen...
        pushScreen (mainScreen);

        //Erzeuge eine Instanz und zeige den Dialog an...
        CustomDialog textEntryDialog = new CustomDialog();
        textEntryDialog.show();
    }

    //Erlaube dem Dialog, den Text zu aktualisieren unter Verwendung
    //dieser Methode...
    public static void setText(String newText)
    {
        enteredText.setText(newText);
    }
}

```

5.4 Dateiauswahldialog

Im folgenden Codebeispiel möchte ich Ihnen zeigen, wie Sie einen Dateiauswahldialog realisieren können. Dieser Dialog soll durch die Dateistruktur des Smartphones blättern, Dateiendungen ausfiltern und eine Datei auswählen mit der Erweiterung txt, doc, pdf, xml oder html.

Quellcode: Dateidialog

```
//
// File Dialog
//

package com.samples.fileSelectorPopup;

import net.rim.device.api.ui.UiApplication;
import net.rim.device.api.ui.component.Dialog;
import net.rim.device.api.ui.container.MainScreen;
import net.rim.device.api.ui.MenuItem;
import java.lang.String;

//
// Ein Dateiauswahldialog
//

public final class FileSelectorPopupSample extends UiApplication
{

    public static void main(String[] args)
    {
        FileSelectorPopupSample theApp = new FileSelectorPopupSample();
        theApp.enterEventDispatcher();
    }

    public FileSelectorPopupSample()
    {
        MainScreen mainScreen = new MainScreen();
        mainScreen.setTitle("Dateibrowser");

        MenuItem showAllMenu = new MenuItem("Wählen Sie eine Datei", 40, 40)
        {
            public void run()
            {
                FileSelectorPopupScreen fps = new FileSelectorPopupScreen();
                fps.pickFile();
                String theFile = fps.getFile();
            }
        }
    }
}
```



```

        if (theFile == null)
        {
            Dialog.alert("Es wurde keine Datei ausgewählt.");
        }
        else
        {
            Dialog.alert("Datei wurde ausgewählt: " + theFile);
        }
    }
};

mainScreen.addMenuItem(showAllMenu);

MenuItem showImagesMenu = new MenuItem("Wählen Sie eine Datei", 40,
40)
{
    public void run()
    {
        String imageExtensions[] = {"txt", "doc", "pdf", "xml",
"html"};

        FileSelectorPopupScreen fps = new FileSelectorPopupScreen
(null, imageExtensions);
        fps.pickFile();
        String theFile = fps.getFile();

        if (theFile == null)
        {
            Dialog.alert("Es wurde keine Datei ausgewählt.");
        }
        else
        {
            Dialog.alert("Datei wurde ausgewählt: " + theFile);
        }
    }
};

mainScreen.addMenuItem(showImagesMenu);

pushScreen (mainScreen);
    }
}

```

Quellcode: Dateiauswahl-Screen

```

//
// Dateiauswahl-Screen
//

```

```

package com.samples.fileSelectorPopup;

import net.rim.device.api.ui.container.PopupScreen;
import net.rim.device.api.ui.UiApplication;
import net.rim.device.api.system.Bitmap;
import net.rim.device.api.system.Characters;
import net.rim.device.api.ui.component.*;
import net.rim.device.api.ui.container.DialogFieldManager;
import javax.microedition.io.Connector;
import javax.microedition.io.file.*;
import java.util.*;

//
//  PopupScreen mit einem Dateibrowser
//
public class FileSelectorPopupScreen extends PopupScreen
{
    String _currentPath;          //Variable des aktuellen Pfads...
    String[] _extensions;         //Dateierweiterungsfilter...
    ObjectListField _olf;         //Auflistung der Directories und Felder

    //
    //  Öffne den Screen im Wurzelverzeichnis und zeige alle Dateien und
    //  Verzeichnisse an...
    public FileSelectorPopupScreen()
    {
        this(null, null);
    }

    //
    //  Öffne den Screen beginnend ab dem angegebenen Pfad und filtere die
    //  Ergebnisse anhand der Dateierweiterungen
    //  Hinweis: bei NULL wird ab dem Wurzelverzeichnis begonnen
    //  Hinweis: bei NULL werden alle Filetypen angezeigt
    //
    public FileSelectorPopupScreen(String startPath, String[] extensions)
    {
        super(new DialogFieldManager());
        _extensions = extensions;
        prepScreen(startPath);
    }

    //
    //  Anzeige des Screen und Aufforderung, eine Datei zu wählen...
    //
    public void pickFile()
    {

```

```

        UiApplication.getUiApplication().pushModalScreen(this);
    }

    //
    // Wir sind nun im aktuellen Verzeichnis und der User sucht nach einer
    // Datei... hat der User eine Datei ausgewählt oder wurde keine Datei
    // angegeben?? Rückkehr zum aktuellen Verzeichnis...
    //
    public String getFile()
    {
        return _currentPath;
    }

    //Vorbereitung des DialogFieldManager.
    private void prepScreen(String path)
    {
        DialogFieldManager dfm = (DialogFieldManager)getDelegate();
        dfm.setIcon(new BitmapField(Bitmap.getPredefinedBitmap
(Bitmap.QUESTION)));
        dfm.setMessage(new RichTextField("Waehlen Sie eine Datei"));

        _olf = new ObjectListField();
        dfm.addCustomField(_olf);

        updateList(path);
    }

    //Lese alle Dateien und Verzeichnisse den angegebenen Pfads aus...
    private Vector readFiles(String path)
    {
        Enumeration fileEnum;
        Vector filesVector = new Vector();

        _currentPath = path;

        if (path == null)
        {
            //Lese den Hauptverzeichniseintrag ein...
            fileEnum = FileSystemRegistry.listRoots();

            while (fileEnum.hasMoreElements())
            {
                filesVector.addElement((Object)fileEnum.nextElement());
            }
        }
        else
        {
            //Lese die Dateien und Direktories des aktuellen Pfads ein...

```

```

try
{
    FileConnection fc = (FileConnection)Connector.open("file://" +
path);

    fileEnum = fc.list();
    String currentFile;

    while (fileEnum.hasMoreElements())
    {
        //Verwende den Fileerweiterungs-Filter (*.doc, *.pdf...
        //sofern einer gesetzt wurde)
        if (_extensions == null)
        {

            filesVector.addElement((Object)fileEnum.nextElement());
        }
        else
        {
            currentFile = ((String)fileEnum.nextElement());

            if (currentFile.lastIndexOf('/') == (currentFile.
length() - 1))
            {
                //Füge alle Verzeichnisse hinzu.
                filesVector.addElement((Object)currentFile);
            }
            else
            {
                //Hier handelt es sich um eine Datei. Prüfe, ob diese
                //mit dem Filter uebereinstimmt...
                for (int count = _extensions.length - 1; count >= 0;
--count)
                {
                    if (currentFile.indexOf(_extensions[count]) != -1)
                    {
                        //Bravo... Datei stimmt mit Filter überein.
                        //Füge die Datei ein und suche nicht weiter...
                        filesVector.addElement((Object)currentFile);
                        break;
                    }
                }
            }
        }
    }
}
catch (Exception ex)
{

```

```

        Dialog.alert("Nicht möglich, das Verzeichnis zu öffnen... "
+ ex.toString());
    }

    }
    return filesVector;
}

//Der User wählt einen Eintrag aus dem ObjectListField...
private void doSelection()
{
    //den aktuellen Pfad bestimmen...
    String thePath = buildPath();

    if (thePath == null)
    {
        //Nur Aktualisierung des Screens, wenn ein Verzeichnis gewählt
        //wurde...
        updateList(thePath);
    }
    else if (!thePath.equals("*?*"))
    {
        //Nur Aktualisierung des Screens, wenn ein Verzeichnis ausgewählt
        //wurde.
        //Eine weitere Überprüfung ist erforderlich, um eine
        //NullPointerException zu umgehen...
        updateList(thePath);
    }
    else
    {
        //So, der User hat eine Datei ausgewählt...
        //Schließe nun den Screen.
        this.close();
    }
}

//Aktualisiere die Einträge in dem ObjectListField.
private void updateList(String path)
{
    //Lese alle Dateien und Verzeichnisse in den Pfad ein...
    Vector fileList = readFiles(path);

    //Erzeuge nun ein Array des Vectors.
    Object fileArray[] = vectorToArray(fileList);

    //Aktualisiere das Feld mit den neuen Dateien.
    _olf.set(fileArray);
}

```

```

//Erzeuge nun eine Zeichenkette, die den kompletten Pfad enthält,
//den der User ausgewählt hat.
//Wurde eine Datei ausgewählt, dann beende diesen Screen.
//Rückgabe eines *?*, wenn der User eine Datei ausgewählt hat.
private String buildPath()
{

    String newPath = (String)_olf.get(_olf, _olf.getSelectedIndex());

    if (newPath.equals(".."))
    {
        //Gehe nun einen Verzeichniseintrag höher...
        //Entferne das mitgeführte '/';
        newPath = _currentPath.substring(0, _currentPath.length() - 2);

        //Entferne den Rest nach dem letzten '/' (das aktuelle Verzeichnis).
        //Wurde ein '/' nicht gefunden, gehe davon aus, dass der User das
        //Hauptverzeichnis geöffnet hat.
        //Rückgabe von null, um das Hauptverzeichnis anzuzeigen.
        int lastSlash = newPath.lastIndexOf('/');

        if (lastSlash == -1)
        {
            newPath = null;
        }
        else
        {
            newPath = newPath.substring(0, lastSlash + 1);
        }
    }
    else if (newPath.lastIndexOf('/') == (newPath.length() - 1))
    {
        //Endet der Pfad mit einem /, so wurde ein Verzeichnis ausgewählt...
        //Ist das Vorzeichen von _currentPath gleich null, sind wir im
        //Hauptverzeichnis. (es ist dann nicht im Hauptverzeichnis).
        if (_currentPath != null)
        {
            newPath = _currentPath + newPath;
        }
    }
    else
    {
        //Eine Datei wurde ausgewählt...
        _currentPath += newPath;

        //Rückgabe von *?* um die Screen-Aktualisierung zu unterbrechen...
        newPath = "*?*";
    }
}

```

```

    }

    return newPath;
}

//Speichere die Datei und das Verzeichnis, die im Vector aufgeführt
//wurden, und formatiere sie in das Format eines Objekt-Arrays.
private Object[] vectorToArray(Vector filesVector)
{
    int filesCount = filesVector.size();
    int dotIncrementor;
    Object[] files;

    //Wenn wir nicht im Hauptverzeichnis sind, füge ".."
    //an die oberste Stelle des Arrays
    if (_currentPath == null)
    {
        dotIncrementor = 0;
        files = new Object[(filesCount)];
    }
    else
    {
        dotIncrementor = 1;
        files = new Object[(filesCount + dotIncrementor)];

        //Füge .. an oberste Stelle des Arrays und gehe ein Verzeichnis
        //zurück...
        files[0] = (Object)("..");
    }

    for (int count = 0; count < filesCount; ++count)
    {
        files[count + dotIncrementor] = (Object)filesVector.
elementAt(count);
    }

    return files;
}

//Verarbeitung von Trackball-Klicks.
protected boolean navigationClick(int status, int time)
{
    doSelection();
    return true;
}

protected boolean keyChar(char c, int status, int time)
{

```

```

//Schließe diesen Screen, wenn escape ausgewählt (gedrückt) wurde...
if (c == Characters.ESCAPE)
{
    _currentPath = null;
    this.close();
    return true;
}
else if (c == Characters.ENTER)
{
    doSelection();
    return true;
}

return super.keyChar(c, status, time);
}
}

```

5.5 Splash-Screen

Hier können wir Splash-Screens realisieren. Damit diese Funktion auch auf älteren Devices läuft, habe ich die Listings für folgende Versionen der Device-Software bereitgestellt:

- 4.1 und früher
- 4.2 und später

Quelltext: Splash-Screen für die Version 4.1 des Betriebssystems und früher

```

//
// Spash Screen für die Version 4.1 und früher
//

import net.rim.device.api.ui.*;
import net.rim.device.api.ui.component.*;
import net.rim.device.api.ui.container.*;
import net.rim.device.api.i18n.*;
import net.rim.device.api.system.*;
import java.util.*;

public class SplashScreen extends MainScreen {
    private MainScreen next;
    private Timer timer = new Timer();
    private UiApplication application;
    private static final Bitmap _bitmap =
Bitmap.getBitmapResource("mein_image.png");
    public SplashScreen(UiApplication ui, MainScreen next) {
        super(Field.USE_ALL_HEIGHT | Field.FIELD_LEFT);
    }
}

```



```

        this.application = ui;
        this.next = next;
        this.add(new BitmapField(_bitmap));
        SplashScreenListener listener = new SplashScreenListener(this);
        this.addKeyListener(listener);
        this.addTrackwheelListener(listener);
        timer.schedule(new Countdown(), 5000);
        application.pushScreen (this);
    }
    public void dismiss() {
        timer.cancel();
        application.popScreen(this);
        application.pushScreen (next);
    }
    private class Countdown extends TimerTask {
        public void run() {
            DismissThread dThread = new DismissThread();
            application.invokeLater(dThread);
        }
    }
    private class DismissThread implements Runnable {
        public void run() {
            dismiss();
        }
    }
    public static class SplashScreenListener implements KeyListener,
    TrackwheelListener {
        private SplashScreen screen;
        public SplashScreenListener(SplashScreen splash) {
            screen = splash;
        }
        //Ausführen, wenn das Trackwheel gedrückt wurde...
        public boolean trackwheelClick(int status, int time) {
            screen.dismiss();
            return true;
        }
        //Ausführen, wenn das Wheel nicht gedrückt wurde...
        public boolean trackwheelUnclick(int status, int time) {
            return false;
        }
        //Ausführen, wenn das Wheel gerollt wird...
        public boolean trackwheelRoll(int amount, int status, int time) {
            return false;
        }
        public boolean keyChar(char key, int status, int time) {
            //Die ESC-Taste beendet den Splash...
            boolean retval = false;
            switch (key) {

```

```

        case Characters.ESCAPE:
            screen.dismiss();
            retval = true;
            break;
    }
    return retval;
}
// Implementierung des KeyListeners bei keydown...
public boolean keyDown(int keycode, int time) {
    return false;
}
// Implementierung des KeyListeners bei keyRepeat...

public boolean keyRepeat(int keycode, int time) {
    return false;
}
// Implementierung des KeyListeners bei keyStatus...
public boolean keyStatus(int keycode, int time) {
    return false;
}
// Implementierung des KeyListeners bei keyUp...
public boolean keyUp(int keycode, int time) {
    return false;
}
}
}

```

Quellcode: Splash-Screen für die Version 4.2 des Betriebssystems und später

```

//
// SplashScreen
// Version 4.2 oder spätere
//

import net.rim.device.api.ui.*;
import net.rim.device.api.ui.component.*;
import net.rim.device.api.ui.container.*;
import net.rim.device.api.system.*;
import java.util.*;

public class SplashScreen extends MainScreen {
    private MainScreen next;
    private UiApplication application;
    private Timer timer = new Timer();
    private static final Bitmap _bitmap = Bitmap.getBitmapResource("mein_
image.png");

    public SplashScreen(UiApplication ui, MainScreen next) {
        super(Field.USE_ALL_HEIGHT | Field.FIELD_LEFT);
        this.application = ui;
    }
}

```



```

        return retval;
    }
    public boolean keyDown(int keycode, int time) {
        return false;
    }
    public boolean keyRepeat(int keycode, int time) {
        return false;
    }
    public boolean keyStatus(int keycode, int time) {
        return false;
    }
    public boolean keyUp(int keycode, int time) {
        return false;
    }
    public SplashScreenListener(SplashScreen splash) {
        screen = splash;
    }
}

```

5.6 DialogClosedListener

Hier stelle ich einen DialogClosedListener vor. Der Listener reagiert auf eine Methode (dialogClosed) und wird ausgelöst, wenn das spezifizierte Dialogobjekt geschlossen wird.

Quellcode: DialogClosedListener

```

//
// Dialog Listener Hauptapplikation
//

package DialogListener;
import net.rim.device.api.ui.UiApplication;
import net.rim.device.api.ui.Field;
import net.rim.device.api.ui.component.Dialog;
import net.rim.device.api.ui.Screen;
import net.rim.device.api.ui.component.LabelField;
import net.rim.device.api.ui.container.MainScreen;
import net.rim.device.api.system.Bitmap;
public class MainApp extends UiApplication
{
    private MainScreen mainScreen;

    public static void main(String[] args) {
        MainApp theApp = new MainApp();
        theApp.enterEventDispatcher();
    }
}

```

```

    }
    public MainApp() {
        mainScreen = new MainScreen();
        LabelField title = new LabelField("Beispiel des
DialogClosedListener",
LabelField.ELLIPSIS | LabelField.USE_ALL_WIDTH);
        mainScreen.setTitle(title);

        //Anzeige des Hauptscreens
        pushScreen (mainScreen);

        //Erzeuge ein Bitmap und eine MyDialogClosedListener-Instanz
        Bitmap oBitmap = Bitmap.getPredefinedBitmap(Bitmap.INFORMATION);
        MyDialogClosedListener oMyDialogClosedListener = new
        MyDialogClosedListener();
        //Initialisiere das Dialogobjekt
        Dialog dialog = new Dialog(Dialog.D_OK_CANCEL,"DialogClosedListener-
Testprogramm",1,oBitmap,Field.FOCUSABLE | Field.FIELD_HCENTER);

        //Zuweisen des DialogClosedListener zu einem Dialogobjekt
        dialog.setDialogClosedListener(oMyDialogClosedListener);

        //Anzeige des Dialogs
        dialog.show();
    }
}
//
// MyDialogClosedListener
//
package DialogListener;
import net.rim.device.api.ui.component.Dialog;
import net.rim.device.api.ui.component.DialogClosedListener;
public class MyDialogClosedListener implements DialogClosedListener {
    public MyDialogClosedListener() { }

    public void dialogClosed(Dialog dialog, int choice) {

        //Hier fügen Sie Ihren onDialogClosed Code ein...
        System.out.println("Dialog Beendet: " + choice);

    }
}

```

5.7 Applikations-Switch

Um eine UI-Applikation zeitweise in den Hintergrund zu schicken, eignet sich der folgende Code. Einfaches Überschreiben der Methode `protected boolean acceptsForeground()` realisiert den Applikations-Switcher, sodass dieser ein `false` zurückgibt.

Um diese Applikation auszuführen, benötigen Sie den `RuntimeStore` und die Methode `ApplicationManager.launch()`, um Ihre Applikation im Hintergrund laufen zu lassen. Hier der Code:

Quellcode:

```
//
// Applikations-Switcher
//

import net.rim.device.api.ui.UiApplication;
import net.rim.device.api.ui.component.*;
import net.rim.device.api.ui.container.*;
import net.rim.device.api.ui.MenuItem;
import net.rim.device.api.system.RuntimeStore;
import net.rim.device.api.system.ApplicationManager;
import net.rim.device.api.system.ApplicationManagerException;

public class HidingUIApplicationDemo extends UiApplication {
    //"HidingUIApplication"
    //Applikations HashKey
    private static final long ID = 405c06fd;

    private MainScreen _mainScreen;

    private static HidingUIApplicationDemo _app;

    private boolean _acceptsForeground = true;

    public HidingUIApplicationDemo() {
        _mainScreen = new MainScreen();

        LabelField label = new LabelField("Select 'Blende das Menü aus" +
from        " um es vom Applikationsswitcher zu entfernen.");
        _mainScreen.add(label);

        MenuItem hideItem = new MenuItem("Unsichtbar", 1, 1) {
            public void run() {
                setAcceptsForeground(false);
            }
        }
    }
}
```

```

};
_mainScreen.addItem(hideItem);

pushScreen (_mainScreen);
}
//Wir benötigen nun einen Weg, um die versteckte Applikation wieder
//hervorzubringen. Wir wollen nicht ermöglichen, dass die Applikation
//geschlossen oder das Device zurückgesetzt wird...
public static void main(String[] args) {
    RuntimeStore runtimeStore = RuntimeStore.getRuntimeStore();

    //Wenn die Applikation läuft, benötigen wir keine Argumentationen
    //
    if (args == null || args.length == 0) {

        //Prüfen, ob eine Instanz der Applikation bereits läuft und im
        //Store gespeichert ist
        //
        _app = (HidingUIApplicationDemo) runtimeStore.getID();
        if (_app == null || !_app.isAlive()) {
            try {
                //Wenn die Applikation läuft, benötigen wir eine eigene #
                //Instanz mit einem Argument...
                ApplicationManager.getApplicationManager().launch(
                    "Unsichtbare Applikation zeigen?");
            } catch (ApplicationManagerException e) {
                Dialog.alert("ERROR ausgeführte Applikation: " + e);
            }
            //Wenn die Applikation bereits läuft, bringe sie in den
            //Vordergrund
        } else {
            _app.setAcceptsForeground(true);
            _app.requestForeground();
        }
        //Wir haben nun ein Argument, es ist von der ausgeführten
        //Applikation. Also erzeugen wir eine neue Instanz, speichern diese
        //im Arbeitsspeicher und starten die Applikation neu...
        //
    } else {
        _app = new HidingUIApplicationDemo();

        runtimeStore.replace(ID, _app);
        _app.enterEventDispatcher();
    }
    //Diese Methode überschreibt
    protected boolean acceptsForeground() {
        return _acceptsForeground;
    }
}

```

```
//Diese Methode setzt die Applikation in den Vordergrund
//
protected void setAcceptsForeground(boolean bool) {
    _acceptsForeground = bool;
}
}
```

5.8 Applikationen durch Passwort schützen

Diese einfache Applikation erlaubt es dem User, eine Applikation zu starten, wenn das Passwort eingegeben wurde. Das Passwort ist hier zwar hart codiert – besser wäre eine Ablage im Arbeitsspeicher. Aber als kurzer Exkurs, wie eine Applikation mit einem Kennwort geschützt werden kann, ist das schon ganz in Ordnung.

Quellcode: Passwortbeispiel

```
//
// Application Password
//

import net.rim.device.api.i18n.*;
import net.rim.device.api.system.*;
import net.rim.device.api.ui.container.*;
import net.rim.device.api.ui.*;
import net.rim.device.api.ui.component.*;

public class PasswordPopupScreen extends PopupScreen implements
    KeyListener, TrackwheelListener {
    private String _response;
    // Dieses EditField erlaubt es, dass
    // ein Passwort maskiert angezeigt wird,
    // wenn die Eingabe des Users erfolgt...
    private PasswordEditField answer;

    //
    // Das Passwort ist hart kodiert. Eigentlich ist es besser, es
    // im Arbeitsspeicher abzulegen.
    // Aber für dieses Beispiel ist es einfacher, das Passwort hart codiert
    // abzulegen.
    private String password = "geheim";
    public PasswordPopupScreen()
    {
        super(new VerticalFieldManager(),Field.FOCUSABLE);
        LabelField question = new LabelField("Geben Sie bitte das Passwort
ein");
        answer = new PasswordEditField(" ", "");
        add(question);
        add(new SeparatorField());
    }
}
```



```

        add(answer);
    }
    // Diese Funktion wird aufgerufen, wenn das Passwort aufgerufen wird.
    // Es erscheint dann der Passwort-Dialog...
    public void accept() {
        UiApplication.getUiApplication().popScreen(this);
    }
    public void close() {
        UiApplication.getUiApplication().popScreen(this);
    }
    public String getResponse() {
        return _response;
    }

    //
    // Implementation eines Trackwheel Listeners
    //
    public boolean trackwheelClick(int status, int time) {
        _response = answer.getText();
        if (_response.equals(password)) {
            accept();
        }
        else {
            Dialog.alert("Das eingegeben, Passwort ist ungültig!");
        }
        return true;
    }
    // Ausführen, wenn das Trackwheel vorhanden ist...
    public boolean trackwheelUnclick(int status, int time) {
        return false;
    }
    // Ausführen, wenn das Trackwheel gerollt wird...
    public boolean trackwheelRoll(int amount, int status, int time) {
        return true;
    }

    //
    // Implementation eines Key-Listeners
    //
    public boolean keyChar(char key, int status, int time) {
        //Wenn die ESC-Taste gedrückt wird - Applikation bitte verlassen...
        boolean retval = false;
        switch (key) {
            case Characters.ENTER:
                _response = answer.getText();
                if (_response.equals(password)) {
                    accept();
                }
        }
    }

```

```

        // Einen Alarm ausgeben, wenn das Passwort nicht richtig ist...
        else {
            Dialog.alert("Das eingegebene Passwort ist nicht korrekt!");
        }
        retval = true;
        break;
    case Characters.ESCAPE:
        close();
        break;
    default:
        retval = super.keyChar(key,status,time);
    }
    return retval;
}
// Implementation eines KeyListeners: keyDown
public boolean keyDown(int keycode, int time) {
    return false;
}
// Implementation eines KeyListeners: keyRepeat
public boolean keyRepeat(int keycode, int time) {
    return false;
}
// Implementation eines KeyListeners: keyStatus
public boolean keyStatus(int keycode, int time) {
    return false;
}
// Implementation eines KeyListeners: keyUp
public boolean keyUp(int keycode, int time) {
    return false;
}
}

```

5.9 Backdoor-Programme

Auf den BlackBerry-Geräten sind eine Menge sogenannter Backdoor-Sequenzen versteckt, die ausgeführt werden können, sobald sich der User im Home-Screen des Devices befindet. Über das Backdoor **[ALT] + [E] + [S] + [C] + [R]** kann man sich den Systemzustand anzeigen lassen. Ab der JDE-Version 4.2 kann die Backdoor-Funktionalität verwendet werden.

Der folgende kleine Programmabriss zeigt die Implementation des Backdoors, das mit der Tastenkombination ABCD verwendet werden kann.

Quellcode: Backdoor-Sequenz

```
protected boolean openProductionBackdoor(int backdoorCode) {
    switch( backdoorCode ) {
        //
        // Hintertür (BACKDOOR) - konvertiert 4 Zeichen in einen
        // Integerwert mittels Bitshifting und bitweisem OR
        //
        case ( 'A' << 24 ) | ( 'B' << 16 ) | ( 'C' << 8 ) | 'D':
            UiApplication.getUiApplication().invokeLater (new Runnable() {
                public void run()
                {
                    // Quittiert den Empfang der Sequenz...
                    Dialog.inform("Backdoor Sequenz empfangen...");
                }
            });
            return true; // Zurück zum Screen
        }
        return super.openProductionBackdoor(backdoorCode);
    }
}
```

Diese Backdoor-Sequenz funktioniert nur, wenn der implementierte Screen-Code im Vordergrund läuft.

5.10 Multisample

In dieser Programmsammlung werden die folgenden Programmmodule in einer einzigen Applikation zusammengefasst, sodass der User über ein Listenfild entscheiden kann, was er starten möchte. Die Applikation besitzt einen sogenannten `AbstractMainScreen`, der als Template für sämtliche `MainScreens` dieser Anwendung dient. Als Erstes zeige ich Ihnen das Template als Quellcode-Listing.

Quellcode: MainScreen

```
package com.ems.sample.multi.common;

import net.rim.device.api.ui.component.ObjectChoiceField;
import net.rim.device.api.ui.container.MainScreen;

/**
 *
 * Der {@link AbstractMainScreen} dient als Template für sämtliche
 * {@link MainScreen} aus dieser Anwendung.
 *
 * @author Raphael Thierjung
 *
 */
```

```

public abstract class AbstractMainScreen extends MainScreen {

    /**
     * Verhindert die Anzeige des Save-Screens, der den Nutzer
     * auf geänderte Einstellungen hinweist. Dies kann für einige {@link
     * MainScreen} z.B. Option-Screens sinnvoll sein. Für die {@link
     * MainScreen} dieser Beispielanwendung ist der Save-Prompt hingegen
     * überflüssig.
     *
     * @return boolean: true : Gibt an, dass der Save-Prompt bereits
     * "konsumiert" wurde, also keine weitere Handlung erforderlich ist.
     */
    protected boolean onSavePrompt() {
        return true;
    }

    /**
     * Das Überschreiben dieser Methode verhindert, dass bei einem Klick auf
     * den Screen, hinter dem sich kein Element befindet, welches den Klick
     * konsumieren kann, ein Popupmenü erscheint.
     * @return boolean : true: Gibt an, dass der Klick auf den Screen bereits
     * konsumiert wurde.
     */
    protected boolean invokeAction(int action) {
        boolean handled = super.invokeAction(action);
        if (!handled) {
            switch (action) {
                case ACTION_INVOKE: // Trackball click.
                {
                    return true;
                }
            }
        }
        return handled;
    }

    /**
     * Über diese Methode werden die Unterklassen von {@link
     * AbstractMainScreen} identifizierbar sein und
     * korrekt innerhalb von z.B. {@link ObjectChoiceField} angezeigt.
     */
    public abstract String toString();
}

```

Als Nächstes kommen die folgenden Geometrieberechnungsmodule:

- Zylinder
- Quadrat
- Rechteck



Abbildung 5.1: Die Auswahl der zu berechnenden Figuren

Im Modul »Zylinder« lassen sich die Oberfläche, die Fläche des Mantels und das Volumen berechnen. Dieses Berechnungsmodul besteht aus den beiden Codesegmenten:

- `CylinderCalcerFactory`
- und `Cylinder`

Der `FactoryCalculator` übernimmt die Berechnungen für den Mantel und Volumenberechnung eines Zylinders. Hier der Quellcode des Calculators:

Quellcode: `CylinderCalcerFacorty`

```
package com.ems.sample.multi.geo.math.cylinder;

import com.ems.sample.multi.geo.math.AbstractCalcer;
import com.ems.sample.multi.geo.math.ICalcer;

/**
 * Erstellt {@link ICalcer} für die Berechnung von Mantel, Oberfläche und
 * Volumen eines Zylinders.
 */
```

```

* @author Raphael Thierjung
*
*/
public class CylinderCalcerFactory {

    /**
     * Erstellt einen {@link ICalcer} zur Berechnung der Oberfläche von
     * Zylindern.
     *
     * @return den beschriebenen {@link AbstractCalcer}
     */
    public static AbstractCalcer createSurfaceCalcer() {
        return new AbstractCalcer("Oberfläche") {
            protected double doTheCalcJob() {
                double height = ((Double) numbers.elementAt(0)).doubleValue();
                double radius = ((Double) numbers.elementAt(1)).doubleValue() / 2.0;
                return 2 * Math.PI * radius * (height + radius);
            }
        };
    }

    /**
     * Erstellt einen {@link ICalcer} zur Berechnung des Mantels von
     * Zylindern.
     *
     * @return den beschriebenen {@link AbstractCalcer}
     */
    public static AbstractCalcer createNappeCalcer() {
        return new AbstractCalcer("Mantel") {
            protected double doTheCalcJob() {
                double height = ((Double) numbers.elementAt(0)).doubleValue();
                double radius = ((Double) numbers.elementAt(1)).doubleValue() / 2.0;
                return 2 * Math.PI * radius * height;
            }
        };
    }

    /**
     * Erstellt einen {@link ICalcer} zur Berechnung des Volumens von
     * Zylindern.
     *
     * @return den beschriebenen {@link AbstractCalcer}
     */
    public static AbstractCalcer createVolumeCalcer() {
        return new AbstractCalcer("Volumen") {
            protected double doTheCalcJob() {
                double height = ((Double) numbers.elementAt(0)).doubleValue();

```

```

        double radius = ((Double) numbers.elementAt(1)).doubleValue() / 2.0;
        return Math.PI * radius * radius * height;
    }
};
}
}

```

Nun kommen wir auf die mathematischen Funktionen, um einen Zylinder zu berechnen. Da wir Parameter benötigen, um Oberfläche, Mantel und Volumen zu berechnen, werden im folgenden Code die User-Eingaben abgefragt.

Quellcode: Cylinder

```

package com.ems.sample.multi.geo.math.cylinder;

import net.rim.device.api.ui.Manager;
import net.rim.device.api.ui.component.BasicTextField;

import com.ems.sample.multi.geo.math.Geometry;
import com.ems.sample.multi.geo.math.ICalcer;

/**
 * Repräsentiert einen Zylinder und bietet zur Berechnung von
 * dessen Oberfläche, Mantel und Volumen entsprechende Funktionen an.
 *
 * @author Raphael Thierjung
 *
 */
public class Cylinder extends Geometry {

    /**
     * {@link BasicTextField} zur Angabe des Durchmessers.
     */
    private BasicTextField diameterTextField;

    /**
     * {@link BasicTextField} zur Angabe der Höhe.
     */
    private BasicTextField heightTextField;

    /**
     * Fügt die Eingabefelder {@link #diameterTextField} und {@link
     * #heightTextField}
     * dem übergebenen {@link Manager} <code>manager</code> hinzu.
     *
     * @see Geometry#addInputField
     */
    protected void addInputField(Manager manager) {
        manager.add(getDiameterField());
        manager.add(getHeightTextField());
    }
}

```

```

}

/**
 * Befüllt die lokale Variable #calcs mit den entsprechenden Formeln in
 * Form von {@link ICalcer}.
 *
 * @see {@link Geometry#specifyCalcer}
 */
protected void specifyCalcer() {
    calcs = new ICalcer[]{CylinderCalcerFactory.createSurfaceCalcer(),
        CylinderCalcerFactory.createNappeCalcer(), CylinderCalcerFactory.
        createVolumeCalcer()};
}

/**
 * Initialisiert das {@link BasicEditField} {@link #diameterTextField}
 * und gibt dieses zurück.
 *
 * @return das initialisierte {@link BasicEditField}.
 */
public BasicEditField getDiameterField() {
    diameterTextField = new BasicEditField("Durchmesser : ", "0", 40,
        BasicEditField.FILTER_REAL_NUMERIC);
    return diameterTextField;
}

/**
 * Initialisiert das {@link BasicEditField} {@link #heightTextField} und
 * gibt dieses zurück.
 *
 * @return das initialisierte {@link BasicEditField}.
 */
public BasicEditField getHeightTextField() {
    heightTextField = new BasicEditField("Höhe : ", "0", 40,
        BasicEditField.FILTER_REAL_NUMERIC);
    return heightTextField;
}

/**
 * Fügt die Werte aus den {@link BasicEditField} {@link
 * #diameterTextField} und {@link #heightTextField}
 * dem übergebenen {@link ICalcer} zur Berechnung hinzu.
 *
 * @see Geometry#fillCalcer
 */
protected void fillCalcer(ICalcer calcer) {
    calcer.add(getDoubleOfTextField(diameterTextField));
    calcer.add(getDoubleOfTextField(heightTextField));
}

```



```

    }

    /**
     * @see Geometry#toString()
     */
    public String toString() {
        return "Zylinder";
    }
}

```

Nun fehlen uns noch die Module für die Quadratberechnung. In dieser Berechnung möchten wir den Umfang und die Fläche eines Quadrates berechnen lassen. Auch hier haben wir wieder die beiden Module `QuadratCalcerFactory` und `Quadrat`, um diese Aufgabe zu lösen.

Hier nun der Quellcode der Factory:

Quellcode: `QuadratCalcerFactory`

```

package com.ems.sample.multi.geo.math.quadrat;

import com.ems.sample.multi.geo.math.AbstractCalcer;
import com.ems.sample.multi.geo.math.ICalcer;

/**
 * Erstellt {@link ICalcer} für die Berechnung von Umfang und Fläche eines
 * Quadrats.
 *
 * @author Raphael Thierjung
 */
public class QuadratCalcerFactory {

    /**
     * Erstellt einen {@link ICalcer} zur Berechnung des Umfangs von Quadraten.
     *
     * @return den beschriebenen {@link AbstractCalcer}
     */
    public static AbstractCalcer createOutlineCalcer() {
        return new AbstractCalcer("Umfang") {
            protected double doTheCalcJob() {
                return ((Double) numbers.elementAt(0)).doubleValue() * 4;
            }
        };
    }

    /**
     * Erstellt einen {@link ICalcer} zur Berechnung der Fläche von Quadraten.

```

```

*
* @return den beschriebenen {@link AbstractCalcer}
*/
public static AbstractCalcer createAreaCalcer() {
    return new AbstractCalcer("Fläche") {
        protected double doTheCalcJob() {
            double value = ((Double) numbers.elementAt(0)).doubleValue();
            return value * value;
        }
    };
}
}
}

```

Und nun das Java-Programm, um Fläche und Umfang eines Quadrats über User-Eingaben variabel zu halten.

Quellcode: Quadrat

```

package com.ems.sample.multi.geo.math.quadrat;

import net.rim.device.api.ui.Manager;
import net.rim.device.api.ui.component.BasicEditField;

import com.ems.sample.multi.geo.math.Geometry;
import com.ems.sample.multi.geo.math.ICalcer;

/**
 * Repräsentiert ein Quadrat und bietet zur Berechnung von dessen
 * Fläche und Umfang entsprechende Funktionen an.
 *
 * @author Raphael Thierjung
 */
public class Quadrat extends Geometry {

    /**
     * {@link BasicEditField} zur Eingabe der Seitenlänge
     * des Quadrats.
     */
    private BasicEditField sideLengthTextField;

    /**
     * Fügt das {@link BasicEditField} {@link #sideLengthTextField} dem
     * übergebenen {@link Manager} <code>manager</code> hinzu.
     */
    protected void addInputField(Manager manager) {
        manager.add(getSideLengthTextField());
    }
}

```

```

/**
 * Befüllt die lokale Variable #calcs mit den entsprechenden Formeln in
 * Form von {@link ICalcer}.
 *
 * @see {@link Geometry#specifyCalcer}
 */
protected void specifyCalcer() {
    calcs = new ICalcer[]{QuadratCalcerFactory.createOutlineCalcer(),
        QuadratCalcerFactory.createAreaCalcer()};
}

/**
 * Fügt die Werte aus den {@link BasicEditField} {@link
 * #sideLengthTextField} dem übergebenen {@link ICalcer} zur Berechnung
 * hinzu.
 *
 * @see Geometry#fillCalcer
 */
protected void fillCalcer(ICalcer calcer) {
    calcer.add(getDoubleOfTextField(sideLengthTextField));
}

/**
 * Generiert das {@link BasicEditField} {@link #sideLengthTextField} und
 * gibt es zurück.
 *
 * @return das initialisierte {@link BasicEditField}
 */
public BasicEditField getSideLengthTextField() {
    sideLengthTextField = new BasicEditField("Seitenlänge : ", "0", 40,
        BasicEditField.FILTER_REAL_NUMERIC);
    return sideLengthTextField;
}

/**
 * @see ICalcer#toString()
 */
public String toString() {
    return "Quadrat";
}
}

```



Abbildung 5.2: Das Menü für die Berechnung eines Quadrats

Nun kommen wir zum Modul »Rechteck«. Auch hier setzen wir wieder eine Java-Factory und ein Rechteckprogramm ein. Berechnen wollen wir in diesem Modul den Umfang und die Fläche eines Rechtecks.

Quellcode: RectangleCalcerFactory

```
package com.ems.sample.multi.geo.math.rectangle;

import com.ems.sample.multi.geo.math.AbstractCalcer;
import com.ems.sample.multi.geo.math.ICalcer;

/**
 * Erstellt {@link ICalcer} für die Berechnung von Umfang und Fläche eines
 * Rechtecks.
 *
 * @author Raphael Thierjung
 *
 */
public class RectangleCalcerFactory {
    /**
```

```

* Erstellt einen {@link ICalcer} zur Berechnung des Umfangs von
* Rechtecken.
*
* @return den beschriebenen {@link AbstractCalcer}
*/
public static AbstractCalcer createOutlineCalcer(){
    return new AbstractCalcer("Umfang"){
        protected double doTheCalcJob() {
            double width = ((Double)numbers.elementAt(0)).doubleValue();
            double height = ((Double)numbers.elementAt(1)).doubleValue();
            return 2*width+2*height;
        }
    };
}
/**
* Erstellt einen {@link ICalcer} zur Berechnung der Fläche von
* Rechtecken.
*
* @return den beschriebenen {@link AbstractCalcer}
*/
public static AbstractCalcer createAreaCalcer(){
    return new AbstractCalcer("Fläche"){
        protected double doTheCalcJob() {
            double width = ((Double)numbers.elementAt(0)).doubleValue();
            double height = ((Double)numbers.elementAt(1)).doubleValue();
            return width*height;
        }
    };
}
}

```

Nachdem wir die Factory definiert haben, benötigen wir hier wieder User-Input. Das geschieht mit folgendem Code:

Quellcode: Rectangle

```

package com.ems.sample.multi.geo.math.rectangle;

import com.ems.sample.multi.geo.math.Geometry;
import com.ems.sample.multi.geo.math.ICalcer;

import net.rim.device.api.ui.Manager;
import net.rim.device.api.ui.component.BasicTextField;

/**
* Repräsentiert ein Rechteck und bietet zur Berechnung von dessen
* Fläche und Umfang entsprechende Funktionen an.
*

```

```

* @author Raphael Thierjung
*
*/
public class Rectangle extends Geometry {

    /**
     * {@link BasicTextField} zur Angabe der Breite des Rechtecks.
     */
    private BasicTextField widthTextField;

    /**
     * {@link BasicTextField} zur Angabe der Höhe des Rechtecks.
     */
    private BasicTextField heightTextField;

    /**
     * Fügt das {@link BasicTextField} {@link #widthTextField} und {@link
     * #heightTextField} dem übergebenen
     * {@link Manager} <code>manager</code> hinzu.
     */
    protected void addInputField(Manager manager) {
        manager.add(getWidthTextField());
        manager.add(getHeightTextField());
    }

    /**
     * Befüllt die lokale Variable #calcs mit den entsprechenden Formeln in
     * Form von {@link ICalcer}.
     *
     * @see {@link Geometry#specifyCalcer}
     */
    protected void specifyCalcer() {
        calcs = new ICalcer[] { RectangleCalcerFactory.createAreaCalcer(),
            RectangleCalcerFactory.createOutlineCalcer() };
    }

    /**
     * Fügt die Werte aus den {@link BasicTextField} {@link #widthTextField}
     * und {@link #heightTextField}
     * dem übergebenen {@link ICalcer} zur Berechnung hinzu.
     *
     * @see Geometry#fillCalcer
     */
    protected void fillCalcer(ICalcer calcer) {
        calcer.add(getDoubleOfTextField(widthTextField));
        calcer.add(getDoubleOfTextField(heightTextField));
    }

    /**

```

```

    * Initialisiert das {@link BasicEditField} {@link #widthEditField} zur
    * Angabe der Breite eines Rechtecks und gibt diese zurück.
    *
    * @return das initialisierte {@link BasicEditField}
    */
    public BasicEditField getWidthEditField() {
        widthEditField = new BasicEditField("Breite : ", "0", 40,
BasicEditField.FILTER_INTEGER);
        return widthEditField;
    }

    /**
    * Initialisiert das {@link BasicEditField} {@link #heightEditField} zur
    * Angabe der Breite eines Rechtecks und gibt diese zurück.
    *
    * @return das initialisierte {@link BasicEditField}
    */
    public BasicEditField getHeightEditField() {
        heightEditField = new BasicEditField("Höhe : ", "0", 40,
BasicEditField.FILTER_INTEGER);
        return heightEditField;
    }

    /**
    * @see ICalcer#toString()
    */
    public String toString() {
        return "Rechteck";
    }
}

```



Abbildung 5.3: Das Ergebnis der Berechnung des Quadratumfangs

Um diese Module in einem Dialog anzusprechen und die entsprechenden Screens zu laden, benötigen wir ein Programm mit dem Namen `AbstractCalcer.java`, in dem uns Hilfsmittel bereitgestellt werden wie die Ablage der Werte und die Belegung der Klassenvariablen.

Danach benötigen wir eine geometrische Figur mit einer Auswahl an verschiedenen Maßen zur Berechnung – das Programm `Geometrie.java`. Als Hauptmodul kommt nun die Komponente `Icalcer.java` zum Tragen. Der `Icalcer` stellt eine Komponente dar, die Werte über die Methode `{@link #add(double)}` empfängt und speichert, bis die Methode `{@link #calc()}` diese verarbeitet und ein entsprechendes Ergebnis zurückgibt.

Hier der Quellcode:

Quellcode: Icalcer

```
package com.ems.sample.multi.geo.math;

/**
 * Ein {@link Icalcer} stellt eine Komponente dar, die Werte über die
```



```

* Methode {@link #add(double)} empfängt und speichert, bis die Methode
* {@link #calc()} diese verarbeitet und ein entsprechendes
* Ergebnis zurückgibt. Wie und inwieweit die vorherigen eingegebenen Werte
* verarbeitet werden, hängt von der jeweiligen Implementierung ab.
* <p>
* Ein Beispiel wäre ein {@link ICalcer}, der eine einfache Addition
* ausführt, indem bei Aufruf von {@link #calc()} alle vorher per {@link
* #add(double)} hinzugefügten Werte addiert.
* @author Raphael Thierjung
*
*/
public interface ICalcer {
    /**
     * Fügt einen weiteren Wert zum Verrechnen diesem {@link ICalcer} hinzu.
     * @param number : Der hinzuzufügende Wert.
     */
    void add(double number);
    /**
     * Führt die Berechnung aus und gibt den entsprechenden Wert zurück.
     * Im Anschluss werden ALLE Werte dieses {@link ICalcer} gelöscht.
     * @return double : Der jeweils berechnete Wert.
     */
    double calc();
    /**
     * Gibt den Namen dieses {@link ICalcer} zurück.
     * @return String : Der Name dieses {@link ICalcer}
     */
    String toString();
}

```

Um dem Benutzer die Berechnung der verschiedenen Geometriekörper zu ermöglichen, benötigen wir den `GeoMathMainScreen`.

Quellcode: `GeoMathMainScreen`

```

package com.ems.sample.multi.geo;

import net.rim.device.api.ui.Field;
import net.rim.device.api.ui.FieldChangeListener;
import net.rim.device.api.ui.XYEdges;
import net.rim.device.api.ui.component.ObjectChoiceField;
import net.rim.device.api.ui.container.VerticalFieldManager;
import net.rim.device.api.ui.decor.BorderFactory;

import com.ems.sample.multi.common.AbstractMainScreen;
import com.ems.sample.multi.geo.math.Geometry;
import com.ems.sample.multi.geo.math.cylinder.Cylinder;
import com.ems.sample.multi.geo.math.quadrat.Quadrat;
import com.ems.sample.multi.geo.math.rectangle.Rectangle;

```

```

/**
 * Der {@link GeoMathMainScreen} ermöglicht dem Benutzer die Berechnung von
 * Maßen verschiedenster geometrischer Körper.
 * <p>
 * Auf diesem {@link GeoMathMainScreen} wird dazu ein {@link
 * ObjectChoiceField}
 * dargestellt, mit dem der Nutzer den zu berechnenden Körper auswählt.
 * Daraufhin wird auf einem {@link VerticalFieldManager} unterhalb des
 * {@link ObjectChoiceField} ein entsprechendes Formular erstellt, welches
 * Elemente zum Ausfüllen für die Berechnung enthält.
 *
 * @author Raphael Thierjung
 */
public class GeoMathMainScreen extends AbstractMainScreen {
    /**
     * {@link VerticalFieldManager} zur Darstellung der Formulare der
     * verschiedenen berechenbaren geometrischen Formen.
     */
    private VerticalFieldManager functionVFM;
    /**
     * Array von {@link Geometry} Instanzen, zu denen Berechnungen zur
     * Verfügung stehen.
     */
    private Geometry[] geometryFunctions;
    /**
     * {@link ObjectChoiceField} zur Auswahl der verschiedenen zur Verfügung
     * stehenden geometrischen Formen.
     */
    private ObjectChoiceField geoChoiceField;

    /**
     * Standardkonstruktor. Belegt die berechenbaren geometrischen Formen in
     * das lokale Array {@link #geometryFunctions} und befüllt diesen
     * {@link GeoMathMainScreen} mit den entsprechenden GUI-Elementen.
     */
    public GeoMathMainScreen() {
        initFunctionArray();
        init();
    }

    /**
     * Initialisiert den {@link GeoMathMainScreen}, in dem die GUI-Elemente
     * auf diesem platziert werden.
     */
    private void init() {
        this.setTitle("Geometrie");
    }

```

```

    VerticalFieldManager vfm = new VerticalFieldManager
(VVerticalFieldManager.USE_ALL_WIDTH);
    vfm.setBorder(BorderFactory.createRoundedBorder(new XYEdges(5, 5, 5, 5)));
    vfm.add(getGeoChoiceField());

    functionVFM = new VerticalFieldManager();
    functionVFM.setBorder(BorderFactory.createRoundedBorder(new XYEdges(5,
5, 5, 5)));
    add(vfm);
    geometryFunctions[0].createManagerForCalculations(functionVFM);
    add(functionVFM);
}

/**
 * Initialisiert das {@link #geometryFunctions} Array mit den {@link
 * Geometry} Instanzen, die
 * für diese Beispiel-Applikation zur Verfügung stehen.
 */
private void initFunctionArray() {
    geometryFunctions = new Geometry[] { new Quadrat(), new Rectangle(),
new Cylinder() };
}

/**
 * Erstellt das {@link ObjectChoiceField} {@link #geoChoiceField}, in dem
 * die verschiedenen Geotypen ausgewählt werden können, zu denen
 * Berechnungen zur Verfügung stehen.
 *
 * @return : Das initialisierte {@link ObjectChoiceField}
 */
public ObjectChoiceField getGeoChoiceField() {
    geoChoiceField = new ObjectChoiceField ("Geotypen : ",
geometryFunctions, 0, Field.FIELD_HCENTER);
    geoChoiceField.setChangeListener(new FieldChangeListener() {

        public void fieldChanged(Field field, int context) {
            if (context == ObjectChoiceField.CONTEXT_CHANGE_OPTION || context
== 0) {
                functionVFM.deleteAll();
                geometryFunctions[geoChoiceField.getSelectedIndex()].
createManagerForCalculations(functionVFM);
            }

        }

    });
    return geoChoiceField;
}

```

```

}

/**
 * @see AbstractMainScreen#toString()
 */
public String toString() {
    return "Geometrie";
}
}

```

5.11 Multiplikationstabelle

Diese kleine Applikation zeigt Ihnen, wie eine Multiplikationstabelle (auch als Einmal-eins bekannt) realisiert werden kann. Die Applikation kommt mit zwei Java-Programmen, einer `Calc.java`, die zur Berechnung der Multiplikation benötigt wird, und einer `MultiTableScreen.java`, die auf User-Eingaben reagiert.



Abbildung 5.4: Das Grundmenü der Multiplikationstabelle

Quellcode: Calc

```

package com.ems.sample.multi.multtable;

/**
 * Hilfsklasse für die Berechnung der Multiplikation.
 *
 * @author Raphael Thierjung
 *
 */
public class Calc {

    /**
     * Berechnet eine Multiplikation.
     *
     * @param fst : Der erste Faktor
     * @param snd : Der zweite Faktor
     * @return Die Multiplikation der beiden Werte.
     */
    public static int mult(Integer fst, Integer snd) {
        return fst.intValue()*snd.intValue();
    }

}

```

Nun das Hauptprogramm für die Multiplikation `MultiTableScreen.java`, die der User parametrisieren kann.

Quellcode: MultiTableScreen

```

package com.ems.sample.multi.multtable;

import net.rim.device.api.ui.Field;
import net.rim.device.api.ui.FieldChangeListener;
import net.rim.device.api.ui.XYEdges;
import net.rim.device.api.ui.component.ButtonField;
import net.rim.device.api.ui.component.Dialog;
import net.rim.device.api.ui.component.LabelField;
import net.rim.device.api.ui.component.ObjectChoiceField;
import net.rim.device.api.ui.container.HorizontalFieldManager;
import net.rim.device.api.ui.container.VerticalFieldManager;
import net.rim.device.api.ui.decor.BorderFactory;

import com.ems.sample.multi.common.AbstractMainScreen;

/**
 * Der {@link MultiTableScreen} ermöglicht dem User die Berechnung des
 * kleinen 1x1. Dazu hat der {@link MultiTableScreen} zwei {@link
 * ObjectChoiceField} zur Angabe der beiden Faktoren. Mit der Anwahl des

```

```

* entsprechenden {@link ButtonField} wird die entsprechende Multiplikation
* durchgeführt.
*
* @author Raphael Thierjung
*
*/
public class MultTableScreen extends AbstractMainScreen {

    /**
     * {@link ObjectChoiceField} zur Angabe des 1. Faktors.
     */
    private ObjectChoiceField fstNumberChoiceField;
    /**
     * {@link ObjectChoiceField} zur Angabe des 2. Faktors.
     */
    private ObjectChoiceField sndNumberChoiceField;
    /**
     * {@link ButtonField} zum Anstarten der Multiplikation.
     */
    private ButtonField calcButtonField;

    /**
     * Standardkonstruktor. Initialisiert den {@link MultTableScreen}.
     */
    public MultTableScreen() {
        init();
    }

    /**
     * Initialisiert den {@link MultTableScreen} und fügt alle benötigten
     * GUI-Elemente diesem hinzu.
     */
    private void init() {
        this.setTitle("Einmal-Eins");
        VerticalFieldManager vfm = new VerticalFieldManager(
            VerticalFieldManager.FIELD_HCENTER);
        vfm.setBorder(BorderFactory.createRoundedBorder(new XYEdges(5,5,5,5)));
        HorizontalFieldManager hfm = new HorizontalFieldManager(
            HorizontalFieldManager.FIELD_HCENTER|HorizontalFieldManager.FIELD_VCENTER);
        hfm.add(getFstNumberChoiceField());
        hfm.add(new LabelField(" x ", Field.FIELD_HCENTER));
        hfm.add(getSndNumberChoiceField());
        vfm.add(hfm);
        vfm.add(getCalcButtonField());
        add(vfm);
    }
}

```

```

/**
 * Initialisiert das {@link ObjectChoiceField} {@link
 * #fstNumberChoiceField} und gibt dieses zurück.
 *
 * @return das initialisierte {@link ObjectChoiceField}
 */
private ObjectChoiceField getFstNumberChoiceField() {
    fstNumberChoiceField = createNumberChoiceField(10, Field.FIELD_RIGHT);
    return fstNumberChoiceField;
}

/**
 * Initialisiert das {@link ObjectChoiceField} {@link
 * #sndNumberChoiceField} und gibt dieses zurück.
 *
 * @return das initialisierte {@link ObjectChoiceField}
 */
private ObjectChoiceField getSndNumberChoiceField() {
    sndNumberChoiceField = createNumberChoiceField(10, Field.FIELD_LEFT);
    return sndNumberChoiceField;
}

/**
 * Initialisiert das {@link ButtonField} {@link #calcButtonField} und
 * gibt dieses zurück
 *
 * @return das initialisierte {@link ButtonField}.
 */
private Field getCalcButtonField() {
    calcButtonField = new ButtonField("Rechne", Field.FIELD_HCENTER);
    calcButtonField.setChangeListener(new FieldChangeListener() {
        public void fieldChanged(Field field, int context) {
            //Liest das erste ObjectChoiceField aus
            Integer fst = (Integer) fstNumberChoiceField
                .getChoice(fstNumberChoiceField.getSelectedIndex());
            //Liest das zweite ObjectChoiceField aus
            Integer snd = (Integer) sndNumberChoiceField
                .getChoice(sndNumberChoiceField.getSelectedIndex());
            //StringBuffer zum Aufbauen des Ergebnisstrings
            StringBuffer buf = new StringBuffer();
            //Fügt den String zusammen
            buf.append("Ergebnis : ").append(Calc.mult(fst, snd));
            //Zeigt einen Dialog mit dem Ergebnis an
            Dialog.inform(buf.toString());
        }
    });
    return calcButtonField;
}

```

```

/**
 * Erstellt ein {@link ObjectChoiceField} mit Werten von <code>0</code>
 * bis <code>max</code>.
 * @param max : Der Maximalwert, der noch innerhalb des {@link
 * ObjectChoiceField} angezeigt wird.
 * @param style : Styleangabe z.B. @see Field#FIELD_HCENTER
 * @return das initialisierte {@link ObjectChoiceField}
 */
private ObjectChoiceField createNumberChoiceField(int max, long style) {
    ObjectChoiceField field = new ObjectChoiceField("",
        createNumberArray(max), 0, style);
    return field;
}

/**
 * Erstellt ein Array von {@link Integer} von <code>1</code> bis
 * einschließlich <code>max</code>.
 *
 * @param max : Der maximale der einschließlich im Array enthalten sein
 * soll.
 * @return Das erstellte {@link Integer} Array.
 */
private Object[] createNumberArray(int max) {
    Integer[] array = new Integer[max];
    for (int i = 1; i <= max; i++) {
        array[i - 1] = new Integer(i);
    }
    return array;
}

/**
 * @see AbstractMainScreen#toString()
 */
public String toString() {
    return "1x1 Beispiel";
}
}

```

5.12 Primzahlen berechnen

Das hier vorliegende Programm prüft, ob eine eingegebene Zahl eine Primzahl ist. Es besteht wie die Multiplikationslösung aus zwei Java-Programmen, die ich hier vorstellen möchte.



Abbildung 5.5: Der Einstiegsbildschirm des Primzahlen-Programms

Quellcode: PrimeScreen

```
package com.ems.sample.multi.primes;

import net.rim.device.api.ui.Field;
import net.rim.device.api.ui.FieldChangeListener;
import net.rim.device.api.ui.XYEdges;
import net.rim.device.api.ui.component.BasicEditField;
import net.rim.device.api.ui.component.ButtonField;
import net.rim.device.api.ui.component.Dialog;
import net.rim.device.api.ui.component.RichTextField;
import net.rim.device.api.ui.container.VerticalFieldManager;
import net.rim.device.api.ui.decor.BorderFactory;

import com.ems.sample.multi.common.AbstractMainScreen;

/**
 *
 * Auf diesem {@link PrimeScreen} wird dem Nutzer ermöglicht, sowohl
 * Primzahlen zu testen als auch eine Reihe von Primzahlen zu generieren.
```

```
*
* @author Raphael Thierjung
*
*/
public class PrimeScreen extends AbstractMainScreen {

    /**
     * {@link BasicEditField} zur Eingabe der zu testenden
     * Primzahl.
     */
    private BasicEditField primeTestTextField;

    /**
     * {@link RichTextField} zur Anzeige der generierten Primzahlen.
     */
    private RichTextField genPrimeTextField;

    /**
     * Initialisiert diesen {@link PrimeScreen}
     */
    public PrimeScreen() {
        init();
    }

    /**
     * Initialisiert den {@link PrimeScreen}, indem alle benötigten GUI-
     * Elemente diesem hinzugefügt werden.
     */
    private void init() {
        this.setTitle("Primzahl");
        VerticalFieldManager primeTestVfm = new VerticalFieldManager(
            VerticalFieldManager.USE_ALL_WIDTH);
        primeTestVfm.setBorder(BorderFactory.createRoundedBorder(new XYEdges(5,
            5, 5, 5)));
        primeTestVfm.add(getPrimeTestTextField());
        primeTestVfm.add(getPrimeTestButtonField());
        VerticalFieldManager primeGenVfm = new VerticalFieldManager(
            VerticalFieldManager.USE_ALL_WIDTH
            | VerticalFieldManager.USE_ALL_HEIGHT);
        primeGenVfm.setBorder(BorderFactory.createRoundedBorder(new XYEdges(5,
            5, 5, 5)));
        primeGenVfm.add(getGenButtonField());
        primeGenVfm.add(getGenPrimeTextField());
        add(primeTestVfm);
        add(primeGenVfm);
    }

    /**
     * Initialisiert das {@link BasicEditField} {@link #primeTestTextField}
```

```

* zur Eingabe der zu testenden Primzahl.
*
* @return das initialisierte {@link BasicTextField}
*/
public BasicTextField getPrimeTestTextField() {
    primeTestTextField = new BasicTextField("Primzahl zu testen: ", "", 40,
        BasicTextField.FILTER_INTEGER);
    return primeTestTextField;
}

/**
* Generiert das {@link ButtonField} zum Starten des Primzahltests. Dabei
* wird der Wert aus
* {@link #primeTestTextField} in ein Integer transformiert und getestet.
*
* @return das generierte {@link ButtonField}.
*/
public ButtonField getPrimeTestButtonField() {
    ButtonField primeTestButtonField = new ButtonField("Test", FIELD_HCENTER);
    primeTestButtonField.setChangeListener(new FieldChangeListener() {
        public void fieldChanged(Field field, int context) {
            //Variable, in der der ausgelesene Wert abgelegt werden soll
            long number2Test = 23;
            //Gibt an, ob die Eingabe korrekt war
            boolean inputCorrect = true;
            try {
                //Liest den eigentlichen Wert aus dem Eingabefeld aus.
                number2Test = Integer.valueOf(primeTestTextField.getText())
                    .longValue();
            } catch (NumberFormatException e) {
                //Wenn beim Auslesen ein Fehler aufgetreten ist, wird der
                //entsprechende
                //Boolean-Flag gesetzt.
                inputCorrect = false;
            }
            //Wenn die Eingabe nicht korrekt war, wird eine entsprechende
            //Fehlermeldung ausgegeben.
            if(!inputCorrect|number2Test<3){
                Dialog.alert("Bitte einen korrekten Wert eingeben");
                return;
            }
            //Generiert den Ergebnis-String
            StringBuffer resultText = new StringBuffer();
            resultText.append(number2Test).append(" ist ");
            long result = PrimeUtil.testPrime(number2Test);
            if (result == -1)
                resultText.append("eine").append(" Primzahl");
            else

```

```

        resultText.append("keine").append(" Primzahl.\n Grund: ")
            .append(result);
        //und gibt diesen aus.
        Dialog.inform(resultText.toString());
    }
});
return primeTestButtonField;
}

/**
 * Generiert das {@link ButtonField} zum Anstarten der Primzahl-Generierung.
 *
 * @return das initialisierte {@link ButtonField}.
 */
public ButtonField getGenButtonField() {
    ButtonField field = new ButtonField("Generate", FIELD_HCENTER);
    field.setChangeListener(new FieldChangeListener() {
        public void fieldChanged(Field field, int context) {
            genPrimeTextField.setText(PrimeUtil.createPrimes());
        }
    });
    return field;
}

/**
 * Generiert das {@link RichTextField} zum Anzeigen der
 * generierten Primzahlen.
 *
 * @return das initialisierte {@link RichTextField}.
 */
public RichTextField getGenPrimeTextField() {
    genPrimeTextField = new RichTextField();
    return genPrimeTextField;
}

/**
 * @see AbstractMainScreen#toString()
 */
public String toString() {
    return "Primzahl";
}
}

```

Nun die Utility-Klasse zum Testen und Generieren von Primzahlen.

Quellcode: PrimeUtil

```

package com.ems.sample.multi.primes;

import net.rim.device.api.util.Arrays;

/**
 * Utility - Klasse zum Testen und Generieren von Primzahlen.
 *
 * @author Raphael Thierjung
 *
 */
public class PrimeUtil {

    /**
     * Testet eine Zahl auf ihre Primalität <code>n</code> mittels der
     * Probedivision bis einschließlich SQRT(n).
     *
     * @param number2Test : die zu testende Zahl
     * @return -1 : falls die Zahl eine Primzahl ist
     *         m : kleinster Wert durch den n geteilt werden kann.
     */
    public static long testPrime(long number2Test) {
        long end = (long) Math.ceil(Math.sqrt(number2Test));
        for(long i = 2; i <=end; i++){
            if(number2Test%i==0)
                return i;
        }

        return -1;
    }

    /**
     * Erstellt einen Ergebnisstring mit einer Reihe von
     * generierten Primzahlen.
     *
     */
    public static String createPrimes() {
        int n = 2000;
        byte[] noPrime = new byte[n];
        Arrays.fill(noPrime, (byte)0);
        int i = 2;
        while(i*i<n){
            if(noPrime[i]==0){
                for(int j = i*i; j<n;j+=i){
                    noPrime[j]=1;
                }
            }
        }
    }
}

```

```

    }
    i++;
}
StringBuffer buf = new StringBuffer();
for(int k = 2; k<n;k++){
    if(noPrime[k]==0)
        buf.append(k).append(", ");
}
return buf.toString();
}
}

```

Nun wird in den folgenden beiden Programmen `MultiSample.java` und `MultiSampleScreen.java` eine Auswahlliste erstellt, in der der User die Module auswählen kann.

Quellcode: *MultiSample*

```

package com.ems.sample.multi;

import com.ems.sample.multi.geo.GeoMathMainScreen;
import com.ems.sample.multi.multitable.MultTableScreen;
import com.ems.sample.multi.primes.PrimeScreen;

import net.rim.device.api.ui.Screen;
import net.rim.device.api.ui.UiApplication;
/**
 * Dieses Beispiel verdeutlicht den Aufbau einer einfachen Anwendung,
 * sowie die Verwendung der GUI-Elemente in der RIM - Java API.
 * <p>
 * Das Beispiel-Programm ermöglicht die Auswahl von 2 verschiedenen
 * Beispielen.
 *
 * <li> 1 x 1 : Einfaches Modul für das kleine 1 x 1.
 * <li> Primzahlen : Modul zum Primzahl-Test und Primzahl-Generierung.
 * <li> Geomath : Modul zum Berechnen von verschiedenen geometrischen
 * Figuren.
 *
 * <p>
 * Das {@link MultiSample} hat einen Einsprungpunkt für die Anwendung
 * (die {@link #main(String[])} Methode). Ein solcher Einsprung-Punkt
 * ist notwendig, damit die Virtual Machine weiß, womit sie die Anwendung
 * zu starten hat.
 *
 * @author Raphael Thierjung
 */

```

```

public class MultiSample extends UiApplication{

    /**
     * Konstruktor der Hauptanwendung. Dieser sorgt für die Anzeige,
     * des {@link MultiSampleScreen}. Man sollte vermeiden, zuviel
     * Logik in den Konstruktor zu legen, da die eigentliche Anwendung
     * zum Zeitpunkt des Konstruktors noch nicht gestartet ist.
     */
    public MultiSample(){
        pushScreen (new MultiSampleScreen(new Screen[]{new MultTableScreen(),
new PrimeScreen(), new GeoMathMainScreen()}));
    }

    /**
     * Haupteinsprungpunkt für die Anwendung. Wird durch die
     * Virtual Machine angefahren, sobald die Anwendung gestartet wird,
     * unabhängig, ob die Anwendung über das Menü oder automatisiert
     * bei Systemstart ausgeführt wird.
     *
     * @param args : Sammlung von {@link String} Argumente,
     * die dem Anwendungsstart übergeben werden können.
     */
    public static void main(String[] args) {
        MultiSample sample = new MultiSample();
        sample.enterEventDispatcher();
    }

}

```

Nachdem der Benutzer sich über den Eventdispatcher für ein Programm entschieden hat, wird dieses im Hauptprogramm als wählbare Applikation in einem ObjectChoiceField angegeben und aufgerufen.

Quellcode: MultisampleScreen

```

package com.ems.sample.multi;

import net.rim.device.api.ui.Field;
import net.rim.device.api.ui.FieldChangeListener;
import net.rim.device.api.ui.Screen;
import net.rim.device.api.ui.UiApplication;
import net.rim.device.api.ui.XYEdges;
import net.rim.device.api.ui.component.ButtonField;
import net.rim.device.api.ui.component.ObjectChoiceField;
import net.rim.device.api.ui.container.VerticalFieldManager;
import net.rim.device.api.ui.decor.BorderFactory;

import com.ems.sample.multi.common.AbstractMainScreen;

```

```
/**
 *
 * Der {@link MultiSampleScreen} stellt das Hauptmenü der Beispielanwendung
 * dar. Das Hauptmenü besteht aus einem {@link ObjectChoiceField}, mit dem
 * der Nutzer die verschiedenen Beispiele anwählen kann.
 *
 * @author Raphael Thierjung
 */
public class MultiSampleScreen extends AbstractMainScreen {

    /**
     * Ein Array von {@link Screen} Objekten, die über das
     * {@link ObjectChoiceField} {@link #screenChoiceField} angewählt werden
     * können
     */
    private Screen[] sampleScreens;

    /**
     * Das {@link ObjectChoiceField} ermöglicht die Anwahl der {@link Screen}
     * aus dem {@link #sampleScreens} Array.
     */
    private ObjectChoiceField screenChoiceField;

    /**
     * Standardkonstruktor. Sorgt für die Initialisierung dieses
     * {@link MultiSampleScreen} und belegt die lokale Klassenvariable.
     *
     * @param samples:
     * Die {@link Screen} der Beispiel-Module, die durch die
     * Beispielanwendung gestartet werden sollen.
     */
    public MultiSampleScreen(Screen[] samples) {
        this.sampleScreens = samples;
        init();
    }

    /**
     * Initialisiert diesen {@link MultiSampleScreen} und fügt alle GUI-
     * Elemente hinzu.
     */
    private void init() {
        //Setzt den Titel des Screens, welcher oberhalb angezeigt wird.
        this.setTitle("Multiple - Sample");
        //FieldManager ermöglichen die Gruppierung und Organisation von GUI-
        //Elementen.
        VerticalFieldManager vfm = new
        VerticalFieldManager(Field.FIELD_HCENTER);
```



```

//Eine Border sorgt für einen Rahmen um alle Elemente in diesem
//Manager.
vfm.setBorder(BorderFactory.createRoundedBorder(new XYEdges(5, 5, 5, 5)));
//Fügt die beiden Elemente der GUI hinzu.
vfm.add(getSampleChoiceField());
vfm.add(getStartButtonField());
//Fügt dem Screen den Manager hinzu.
add(vfm);
}

/**
 * Erstellt und initialisiert das {@link ObjectChoiceField}, mit dem
 * die Beispiele ausgewählt werden können.
 *
 * @return das beschriebene {@link ObjectChoiceField}.
 */
private ObjectChoiceField getSampleChoiceField() {
    screenChoiceField = new ObjectChoiceField ("Beispiele", sampleScreens);
    return screenChoiceField;
}

/**
 * Erstellt und initialisiert das Start-{@link ButtonField}, mit dem
 * die gewählten Beispiele gestartet werden können.
 *
 * @return den beschriebene {@link ButtonField}.
 */
private ButtonField getStartButtonField() {
    //Erstellung eines zentral platzierten ButtonField.
    ButtonField field = new ButtonField("Start", Field.FIELD_HCENTER);
    //Der FieldChangeListener wird bei Klick auf den Button ausgeführt.
    field.setChangeListener(new FieldChangeListener() {
        public void fieldChanged(Field field, int context) {
            UiApplication.getUiApplication().pushScreen (sampleScreens
[screenChoiceField.getSelectedIndex()]);
        }
    });
    return field;
}

/**
 * Gibt eine Beschreibung dieses {@link MultiSampleScreen} als
 * {@link String} Objekt.
 */
public String toString() {
    return "MainScreen";
}
}

```



Abbildung 5.6: Das Ergebnis des Primzahlentests

5.13 Mail-Management

In den folgenden Java-Listings möchte ich auf das Thema Mail-Management eingehen. Darunter versteht man:

- Notify (Benachrichtigung über eingegangene Mails)
- Receive (Mail-Empfang)
- Send (Mail-Versand)



Abbildung 5.7: Das Grundmenü des E-Mail-Managers

In den folgenden Beispielen möchte ich auf diese Java-Module eingehen und Ihnen den Quellcode an einem Beispiel zeigen.

Fangen wir jedoch mit dem Mail-Listener an. Der Listener oder auch Notifier läuft im Hintergrund und benachrichtigt uns, sobald eine Mail eintrifft.

Quellcode: MailListener

```
package com.ems.sample.mail;

import net.rim.blackberry.api.mail.Folder;
import net.rim.blackberry.api.mail.ServiceConfiguration;
import net.rim.blackberry.api.mail.Session;
import net.rim.blackberry.api.mail.Store;
import net.rim.blackberry.api.mail.event.FolderEvent;
import net.rim.blackberry.api.mail.event.FolderListener;
import net.rim.device.api.servicebook.ServiceBook;
import net.rim.device.api.servicebook.ServiceRecord;
import net.rim.device.api.system.ApplicationDescriptor;
import net.rim.device.api.system.ApplicationManager;
import net.rim.device.api.system.ApplicationManagerException;
```

```

import net.rim.device.api.system.RuntimeStore;

/**
 * Der {@link MailListener} wird von der BlackBerry Mail-API bei Eintreffen
 * einer Nachricht in den {@link Folder} im Mailsystem informiert, in denen
 * dieser registriert wurde.
 * Dieser {@link MailListener} wird in allen verfügbaren {@link Folder}
 * registriert, welche E-Mails verarbeiten, und dabei wird
 * eine Eingangsbox markiert durch den Typ {@link Folder#INBOX}.
 * <p>
 * Der {@link MailListener} ist als Singleton konzipiert, existiert also nur
 * einmalig im gesamten System.
 *
 * @author Raphael Thierjung
 */
public class MailListener implements FolderListener {
    /**
     * Gibt an, ob der {@link MailListener} bereits in der Mail-API
     * registriert ist.
     */
    private boolean registered;
    /**
     * {@link ApplicationDescriptor} zum Starten der Anwendung bei Eintreffen
     * einer Email.
     */
    private ApplicationDescriptor alertDescriptor;
    /**
     * <code>long</code> Wert, welcher den {@link MailListener} eindeutig in
     * dem {@link RuntimeStore} identifiziert.
     */
    protected static long GUID = 0x5AAF4BC5625E42C3L;
    /**
     * Singleton-Instanz-Variable vom {@link MailListener}
     */
    private static MailListener instance;

    /**
     * Standardkonstruktor. Als <code>privat</code> deklariert, damit nur
     * über die Methode {@link #getInstance()} ein {@link MailListener}
     * Objekt erstellt werden kann.
     */
    private MailListener() {
        registered = false;
        //Erstellt einen AlertDescriptor auf Basis des Descriptors der
        //Anwendung, welche diesen Konstruktor aufruft. Da dieser nur über die
        //Anwendung MailSample aufgerufen wird, ist auch dieser Basis dieses
        //ApplicationDescriptors.
    }
}

```

```

        alertDescriptor = new ApplicationDescriptor(ApplicationDescriptor
            .currentApplicationDescriptor(), "Mail Test",
            new String[] { "nongui" });
    }

    /**
     * Gibt die eindeutige Instanz-Variable diese {@link MailListener}
     * Singleton - Instanz zurück.
     * @return {@link MailListener} - Die Singleton Instanz Variable
     */
    public static synchronized MailListener getInstance() {
        //Ist die Instanz-Variable bereits initialisiert worden?
        if (instance == null)
            //Falls nicht, wird versucht, aus dem RuntimeStore eine Instanz über
            //die eindeutigen ID GUID zu erhalten.
            instance = (MailListener) RuntimeStore.getRuntimeStore().get(GUID);
        //Hat der RuntimeStore keine Instanz vorliegen?
        if (instance == null) {
            //Dann wird eine neue Instanz erstellt
            MailListener controller = new MailListener();
            //Und im RuntimeStore für ein späteres Abrufen abgelegt.
            RuntimeStore.getRuntimeStore().put(GUID, controller);
            //Und der lokalen Klassenvariablen zugewiesen.
            instance = controller;
        }
        return instance;
    }

    /**
     * Registriert den {@link MailListener} in allen {@link Folder} der Mail-
     * Systems, welche zum einem
     * vom Typ {@link Folder#INBOX} sind und E-Mails verarbeiten.
     */
    public void registerInMailbox() {
        //Abrufen aller im System registrierten ServiceBook - Einträge
        ServiceRecord[] serviceBookRecords = ServiceBook.getSB().getRecords();

        for(int i = 0; i<serviceBookRecords.length;i++){
            // Abprüfen, ob sich der Eintrag auf ein E-Mail Konto bezieht.
            if (serviceBookRecords[i].getCid().equals("CMIME")) {
                //Erstellen einer ServiceConfiguration zum Abrufen der enthaltenen
                //Mail-Folder
                ServiceConfiguration serviceBookRecord = new ServiceConfiguration
                    (serviceBookRecords[i]);
                Store store = Session.getInstance(serviceBookRecord).getStore();
                Folder[] folders = store.list();
                for(int j = 0; j<folders.length-1;j++){
                    Folder f = folders[j];

```

```

        recurse(f);
    }
}
registered = true;
}

/**
 * Fügt diesen {@link MailListener} dem übergebenen {@link Folder}
 * <code>folder</code> hinzu und läuft dann rekursiv über alle Sub
 * - {@link Folder} des <code>folder</code>.
 *
 * @param folder : Der {@link Folder} in dem dieser {@link MailListener}
 * registriert werden soll.
 */
private void recurse(Folder folder) {
    //Der MailListener wird nur registriert, wenn sich es sich um ein
    //Eingangspostfach handelt.
    if (folder.getType() == Folder.INBOX) {
        folder.addFolderListener(this);
    }
    //Anschließend wird die Methode rekursiv über alle Sub-Folder
    //durchgeführt.
    Folder[] folderArray = folder.list();
    for(int i = 0; i<folderArray.length;i++){
        recurse(folderArray[i]);
    }
}

/**
 * Wird aufgerufen, sobald eine neue Nachricht in den {@link Folder}
 * hinzugefügt wurde, in den dieser {@link MailListener} registriert
 * wurde.
 *
 * <p>
 * Wird eine Nachricht hinzugeführt, dann wird über den lokalen {@link
 * ApplicationDescriptor}
 * {@link #alertDescriptor} die Anwendung gestartet.
 * @see FolderListener#messagesAdded(FolderEvent)
 */
public void messagesAdded(final FolderEvent e) {
    try {
        ApplicationManager.getApplicationManager().runApplication(
            alertDescriptor);
    } catch (ApplicationManagerException e1) {
        e1.printStackTrace();
    }
}
}

```

```

/**
 * Wird aufgerufen, sobald eine Nachricht entfernt wurde. In diesem
 * Beispiel wird dies nicht weiter betrachtet.
 *
 * @see FolderListener#messagesRemoved(FolderEvent)
 */
public void messagesRemoved(FolderEvent e) {
    // Intendet blank
}

/**
 * Gibt an, ob dieser {@link MailListener} bereits im System registriert
 * wurde.
 * @return <code>true</code> {@link MailListener} bereits registriert,
 * <code>false</code> sonst.
 */
public boolean isRegistered() {
    return registered;
}
}

```

Im folgenden Beispiel zeige ich Ihnen den Zugriff auf die Mail-API des Blackberry. E-Mails können über einen Main-Screen abgesetzt werden. Darüber hinaus erscheint bei dem Eintreffen neuer Nachrichten ein Dialogfenster.

Quelltext: MailSample

```

package com.ems.sample.mail;

import net.rim.device.api.ui.UiApplication;
import net.rim.device.api.ui.component.Dialog;

/**
 *
 * Dieses Beispiel zeigt den Zugriff auf die Mail - API des Blackberry.
 * Dabei können wir über einen Main-Screen Emails absetzen. Außerdem wird
 * bei Eintreffen einer neuen Nachricht ein Dialog angezeigt.
 *
 * @author Raphael Thierjung
 *
 */
public class MailSample extends UiApplication{

    /**
     * Konstruktor. Ruft abhängig des <code>boolean normal</code> entweder
     * die Anwendung auf oder zeigt einen Dialog an.
     *

```

```

* @param normal : bei <code>true</code> Der {@link MailScreen} der
* Anwendung wird angezeigt, <code>false</code> zeigt einen Dialog.
*/
public MailSample(boolean normal){
    if(normal)
        pushScreen (new MailScreen());
    else
        invokeLater(new Runnable(){
            public void run() {
                Dialog.inform("Message received");
            }
        });
}

/**
* Haupteinsprungpunkt dieser Anwendung.
*
* @param args : Konstruktorargumente. Enthalten diese das Word "nongui"
* wird anstatt der Anwendung lediglich ein Dialog angezeigt. Dies ist
* der Fall, wenn der {@link MailListener} die
* Anwendung nach Eintreffen einer Nachricht startet.
*/
public static void main(String[] args) {
    if(args.length>0 && args[0].equals("nongui")){
        new MailSample(false).enterEventDispatcher();
    } else {
        new MailSample(true).enterEventDispatcher();
    }
}
}

```

Im folgenden Abschnitt beschreibe ich den dazugehörigen MailScreen. Dieser Screen ist der Haupt-Screen der Applikation. Er ermöglicht es dem User, entweder eine Mail abzusetzen oder den Mail-Listener zu aktivieren.

Quellcode: MailScreen

```

package com.ems.sample.mail;

import net.rim.device.api.ui.Field;
import net.rim.device.api.ui.FieldChangeListener;
import net.rim.device.api.ui.XYEdges;
import net.rim.device.api.ui.component.BasicEditField;
import net.rim.device.api.ui.component.ButtonField;
import net.rim.device.api.ui.component.Dialog;
import net.rim.device.api.ui.component.LabelField;
import net.rim.device.api.ui.container.MainScreen;

```



```

import net.rim.device.api.ui.container.VerticalFieldManager;
import net.rim.device.api.ui.decor.BorderFactory;

/**
 * Der {@link MailScreen} ist der Hauptbildschirm der {@link MailSample}
 * Anwendung. Über den Bildschirm hat der Nutzer die Möglichkeit, per {@link
 * ButtonField} den {@link MailListener} im System zu registrieren oder eine
 * Beispiel-Mail an eine E-Mail-Adresse nach Wahl abzusetzen. @author
 * Raphael Thierjung
 *
 */
public class MailScreen extends MainScreen {
    /**
     * Informations-Text, welcher über dem {@link ButtonField} für die
     * Registrierung des {@link MailListener} angezeigt wird.
     */
    private static final String listenerText = "With a Click on that Button
you can register a Mail-Listener,"
        + " which keeps u informed about incoming and deleted mails.";
    /**
     * Informations-Text, welcher über dem {@link BasicEditField} zur Eingabe
     * einer Ziel-E-Mailadresse für das Absenden einer Beispiel-Mail
     * angezeigt wird.
     */
    private static final String sendingText = "Sends a standard email to an
Email of your choice.";
    /**
     * {@link BasicEditField}, in das der Nutzer die Ziel-Emailadresse für
     * die Beispiemail eingeben kann.
     */
    private BasicEditField textField;

    /**
     * Standardkonstruktor. Initialisiert den Screen und füllt diesen mit
     * GUI - Elementen.
     */
    public MailScreen() {
        init();
    }

    /**
     * Initialisiert den Screen und fügt diesen GUI-Elemente hinzu.
     */
    private void init() {
        setTitle("Mail Sample Application");
        VerticalFieldManager listenerVFM = new VerticalFieldManager(
            VerticalFieldManager.USE_ALL_WIDTH);
        listenerVFM.setBorder(BorderFactory.createRoundedBorder(new XYEdges(5,

```

```

        5, 5, 5)));
    listenerVFM.add(new LabelField(listenerText, Field.USE_ALL_WIDTH
        | Field.USE_ALL_WIDTH));
    listenerVFM.add(getRegisterListenerField());
    VerticalFieldManager mailSendVFM = new VerticalFieldManager(
        VerticalFieldManager.USE_ALL_WIDTH);
    mailSendVFM.setBorder(BorderFactory.createRoundedBorder(new XYEdges(5,
        5, 5, 5)));
    mailSendVFM.add(new LabelField(sendingText, Field.USE_ALL_WIDTH
        | Field.USE_ALL_WIDTH));
    mailSendVFM.add(getEmailTextField());
    mailSendVFM.add(getSendMailButton());
    add(listenerVFM);
    add(mailSendVFM);
}

/**
 * Erstellt das {@link BasicEditField} {@link #textField} und gibt dieses
 * zurück.
 * @return Das initialisierte {@link BasicEditField}
 */
private Field getEmailTextField() {
    textField = new BasicEditField("Emailadress : ", "", 80,
        BasicEditField.FILTER_EMAIL | Field.USE_ALL_WIDTH | Field.FIELD_
HCENTER);
    return textField;
}

/**
 * Erstellt ein {@link ButtonField}, bei dessen Klick eine Beispiemail
 * an den Absender im {@link BasicEditField} {@link #textField}.
 * @return Das initialisierte {@link ButtonField}.
 */
private Field getSendMailButton() {
    ButtonField sendMailButton = new ButtonField("Send Mail",
        Field.FIELD_HCENTER);
    sendMailButton.setChangeListener(new FieldChangeListener() {
        public void fieldChanged(Field field, int context) {
            MailSender.sendMail(textField.getText());
            textField.setText("");
        }
    });
    return sendMailButton;
}

/**
 * Erstellt einen {@link ButtonField}, bei dessen Klick der {@link
 * MailListener} im System registriert wird.

```

```

*
* @return das initialisierte {@link ButtonField}
*
*/
private Field getRegisterListenerField() {
    ButtonField registerListenerButton = new ButtonField(
        "Register Maillistener", Field.FIELD_HCENTER);
    FieldChangeListener changeListener = new FieldChangeListener() {

        public void fieldChanged(Field field, int context) {
            if (MailListener.getInstance().isRegistered())
                Dialog.inform("Listener already registered");
            else
                MailListener.getInstance().registerInMailbox();
        }

    };
    registerListenerButton.setChangeListener(changeListener);
    return registerListenerButton;
}

/**
 * Sorgt dafür, dass bei Klick auf das Display kein Menü dargestellt wird.
 */
protected boolean invokeAction(int action) {
    boolean handled = super.invokeAction(action);
    if (!handled) {
        switch (action) {
            case ACTION_INVOKE: // Trackball click.
            {
                return true;
            }
        }
    }
    return handled;
}

/**
 * Verhindert die Anzeige des Save Prompts
 */
public boolean onSavePrompt() {
    return true;
}
}

```

Der folgende Code ist der Mail-Sender. Dieses Programm ermöglicht es dem User, eine Mail abzusetzen.

Quellcode: MailSender

```

package com.ems.sample.mail;

import net.rim.blackberry.api.mail.Address;
import net.rim.blackberry.api.mail.Folder;
import net.rim.blackberry.api.mail.Message;
import net.rim.blackberry.api.mail.Session;
import net.rim.blackberry.api.mail.Transport;

/**
 * Ermöglicht das Absetzen von Email-Adressen.
 *
 * @author Raphael Thierjung
 *
 */
public class MailSender {

    /**
     * Ermöglicht das Absenden einer Test E-Mail an die übergebene E-Mail-
     * Adresse <code>mailadress</code>.
     *
     * @param mailadress : {@link String} E-Mail Adresse an den die
     * Testnachricht geschickt werden soll.
     */
    public static void sendMail(String mailadress) {
        //Ruft alle Folder im System ab, welche einen Postausgang darstellen.
        Folder[] folders = Session.getDefaultInstance().getStore().list
(Folder.SENT);
        //Erstellen des Arrays aller Empfänger
        Address to[] = new Address[1];
        //Erstellt eine Message und registriert diese im zuerstverfügbaren
        //Postausgang Folder
        Message message = new Message(folders[0]);

        try {
            //Einfügen der Empfängeradresse in den Array. Dabei ist zu beachten,
            // dass keine Leerzeichen vor oder hinter dem String angefügt sind.
            to[0] = new Address(mailadress.trim(), "");

            //Fügt den Empfänger zur vorher erstellten Nachricht hinzu.
            message.addRecipients(Message.RecipientType.TO, to);

            //Setzt den Betreff der Nachricht mit einem Beispielttext.
            message.setSubject("Das ist eine Testnachricht");

            //Setzt den Text der Nachricht mit einem Beispielttext.

```

```

        message.setContent("Ich sende diese Mail mit einem
Beispielprogramm");

        //Setzt die Priorität der Nachricht auf Hoch.
        message.setPriority(Message.Priority.HIGH);

        //Absenden der Nachricht.
        Transport.send(message);

    } catch (Exception me) {
        System.err.println(me);
    }
}
}

```



Abbildung 5.8: Eingabe der Mailadresse zum Versand

6 Drahtloser Datentransport

In diesem Kapitel geht es um die zentralen Aspekte der drahtlosen Datenübertragung. Wir betrachten, wie Sie den BlackBerry Enterprise Server als Netzwerk-Gateway nutzen, stellen aber auch andere Möglichkeiten des Datentransports vor. Darüber hinaus widmen wir uns Features wie der Datenverschlüsselung, der Authentifizierung sowie der Anwendungsverteilung.

6.1 Drahtlose Gateways

Java-Anwendungen für BlackBerry-Geräte können Verbindungen zu einem drahtlosen Netzwerk über normale HTTP-, HTTPS- und TCP-Socket-Protokolle einrichten. Wenn eine Anwendung eine Verbindung über das drahtlose Netzwerk herstellt, stehen zwei drahtlose Gateways als Proxy für die Verbindung zum Internet oder zum Unternehmensintranet zur Wahl.

Sie können Ihre Anwendung auf das Standard-Gateway abstimmen, das den BlackBerry-Benutzern zur Verfügung steht, oder Sie können Ihren Code so anpassen, dass ein bevorzugtes Gateway verwendet wird. Legen Sie in Ihrer Anwendung fest, dass explizit das bevorzugte Gateway für die Verbindung ausgewählt werden soll und das Ausweichen auf das 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 Netzbetreiber verwenden.

6.1.1 BlackBerry Enterprise Server als Netzwerk-Gateway

Wenn Sie den Enterprise Server als Netzwerk-Gateway verwenden, wird der gesamte Datenverkehr zwischen Ihrer Anwendung und dem Server mithilfe der AES- (Advanced Encryption Standard) oder Triple-DES-Verschlüsselung (Data Encryption Standard) verschlüsselt.

Der 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 MDS-Komponente im Enterprise Server enthält die MDS-Services, die wiederum einen HTTP- und TCP/IP-Proxy-Dienst umfassen, der von der Java-Applikation 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 gegebenenfalls den Enterprise Server als Gateway nutzen. Netzwerkanforderungen werden hinter der Firewall des Unternehmens zum Enterprise Server geleitet, der die Netzwerkanforderung an das Internet durch die Unternehmensfirewall hindurch vornimmt.

Administratoren können eine IT-Richtlinie festlegen, mit der sichergestellt wird, dass der 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 Service¹⁰ als auch mit dem Internet-Gateway Ihres Mobilfunkanbieters verwalten.

6.1.2 Verwenden des Internet-Gateways des Mobilfunkanbieters

Die meisten Mobilfunkanbieter bieten ihr eigenes Internet-Gateway mit direkter TCP/IP-Konnektivität zum Internet an. Manche Mobilfunkanbieter stellen zudem ein WAP-Gateway (Wireless Application Protocol) zur Verfügung, mit dem HTTP-Verbindungen über das WAP-Protokoll erfolgen können.

Eine Java-Applikation für den BlackBerry kann mit beiden Gateways eine Verbindung zum Internet herstellen. Wenn Ihre Anwendung sich an BlackBerry-Gerätebenutzer richtet, die ein bestimmtes Funknetzwerk nutzen, führt die Verwendung des Internet-Gateways des Mobilfunkanbieters meist zu guten Ergebnissen.

Richtet sich Ihre Anwendung dagegen an Gerätebenutzer in verschiedenen Funknetzen, dann 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.

6.2 Alternative Optionen für den Datentransport

Mit den BlackBerry-APIs können Sie, wie gesehen, eine Applikation erstellen, bei der Daten per E-Mail versendet und empfangen werden. E-Mail, SMS und PIN-Nachrichtenübermittlung bieten effektive Möglichkeiten für die Verteilung von Inhalten an BlackBerry-Benutzer, wenn die traditionellen Push-Modelle nicht verfügbar sind.

¹⁰ auch BIS genannt

6.2.1 Datentransport per E-Mail

Eine Java-Applikation kann über die BlackBerry-APIs E-Mail-Nachrichten senden und eingehende E-Mail-Nachrichten abfragen.

Eine Java-Applikation kann auch auf die Details und die Kopfzeilen der E-Mail-Nachrichten zugreifen, die lokal auf dem BlackBerry gespeichert sind, und Listener für Statusänderungen einer E-Mail-Nachricht registrieren.

6.2.2 Datentransport per SMS

Mit den BlackBerry-APIs können Sie Anwendungen erstellen, bei denen Daten per SMS versendet und empfangen werden.

Die APIs ermöglichen einer Anwendung, SMS-Nachrichten zu senden und eingehende SMS-Nachrichten abzufragen.

6.2.3 Datentransport per PIN-Nachrichtenübermittlung

Bei der PIN-Nachrichtenübermittlung wird der Datenkanal statt des Stimmenkanals verwendet, sodass Sie das 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 PIN-Nachrichten erlauben.

Die PIN-Nachrichtenübermittlung bietet eine effektive Möglichkeit, PIN-Anwendungen zu realisieren, die sich ausschließlich an BlackBerry-Benutzer richten.

6.3 Integrieren von BlackBerry-Anwendungen

Eine Java-Applikation für den BlackBerry kann dem Menü für die E-Mail-, Terminplaner- und Telefonanwendungen auf dem Gerät benutzerdefinierte Menüelemente hinzufügen.

Wenn ein Anwender das benutzerdefinierte Menüelement auswählt, startet die Java-Applikation mit einem Verweis auf das Objekt, das der Gerätebenutzer ausgewählt hat.

Beispielsweise kann eine BlackBerry-Applikation der E-Mail-Anwendung ein Menüelement hinzufügen, mit dem der Standort des Absenders angezeigt wird. Wenn ein Gerätebenutzer das Menüelement auswählt, startet die Applikation mit einem Verweis auf das E-Mail-Objekt, das derzeit markiert ist oder das der Benutzer öffnet.

Die Applikation ermittelt den Standort des Absenders anhand der E-Mail-Adresse (entweder durch Nachschlagen in der Kontaktliste oder durch Abrufen von Daten von einem Remote-Server). Sie wird dann in den Vordergrund gerückt und blendet eine Karte ein.

6.3.1 Integrieren in Anwendungen der BlackBerry Device-Software

Eine Geräteanwendung kann eine Anwendung der Device-Software aufrufen (zum Beispiel E-Mail, Terminplaner, Telefon, Browser oder Kamera), um eine Aktion auszuführen oder Daten auf dem Bildschirm anzuzeigen. Die folgenden Beispiele zeigen, wie eine Geräteanwendung eine BlackBerry-Device-Software-Applikation aufrufen kann:

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

6.3.2 Zugriff auf E-Mail- und Terminplanerdaten

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

6.3.3 Verwenden des BlackBerry-Messenger mit einer Applikation

Sie können eine Applikation in die BlackBerry-Messenger-Anwendung integrieren. Dies ist beispielsweise dann von Nutzen, wenn Sie ein Spiel für das Gerät erstellen, bei dem die Spieler abwechselnd einen Spielzug tätigen.

Zum Erstellen einer Java-Applikation, 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 Messenger-Klasse finden Sie in der API-Referenz im Anhang C und D.

6.4 Listener für Reaktionen auf Anwendungsänderungen

Eine Applikation kann Änderungs-Listener für die E-Mail- und Terminplanerdatenspeicher sowie in der Telefonanwendung registrieren.

Die Listener ermöglichen der Applikation eine sofortige Reaktion, wenn der Gerätebenutzer ein lokales Ereignis auslöst. Mit den E-Mail- und Terminplanerdaten-Listnern können Sie eine Applikation benachrichtigen lassen, sobald neue Einträge ankommen oder wenn der Gerätebenutzer Änderungen an den vorhandenen Daten ausführt, etwa Ergänzungen, Löschungen oder Aktualisierungen.

Mit Telefon-Listnern können Sie Anrufaktionen abfragen, zum Beispiel das Einleiten neuer Anrufe oder das Beenden von Anrufen.

6.5 Sicherheitsbedingungen

Wenn Sie den Enterprise Server als Netzwerk-Gateway für Ihre Anwendung verwenden, verschlüsselt der BlackBerry Enterprise Server die Daten an allen Punkten der Verbindung zwischen dem Gerät und dem BlackBerry Enterprise Server hinter der Firewall des Unternehmens mit der AES- oder TripleDES-Verschlüsselung.

Sollen die Daten zwischen dem Enterprise Server und dem Zielsystem weiter verschlüsselt werden, können Sie das HTTPS-Protokoll verwenden und die SSL/TLS-Verschlüsselung heranziehen.

Wenn Ihre Anwendung den Internet Service oder das Internet-Gateway des Mobilfunk-anbieters verwendet, wird der Datenverkehr nicht verschlüsselt. Auf Wunsch Ihrer Benutzer 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.

6.5.1 Datenverschlüsselung auf dem BlackBerry-Gerät

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

6.5.2 Zugriff auf Speicher

Die BlackBerry-Entwicklungsumgebung soll Anwendungen daran hindern, unbeabsichtigt oder vorsätzlich Probleme in anderen Anwendungen oder auf dem Endgerät zu verursachen.

Die Anwendungen können nur in den Gerätespeicher schreiben, den die 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 hierauf erhalten).

Eine Java-Applikation kann nur über bestimmte APIs auf permanenten Speicher oder auf Benutzerdaten zugreifen oder mit anderen Anwendungen kommunizieren. Eine Applikation, 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.

6.6 Authentifizierung

BlackBerry-Benutzer können ein Kennwort für ihre Geräte festlegen. Wenn das Geräte-kennwort aktiv ist, müssen die Nutzer das Kennwort eingeben, um auf die Daten und Anwendungen zugreifen zu können.

Die Verwendung von Geräte-kennwörtern ist ein guter erster Schritt zur Begrenzung des Zugriffs auf Ihre Applikation auf dem BlackBerry.

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

6.6.1 Benutzerauthentifizierung

Bei Anwendungen, in denen Sicherheitsmerkmale von entscheidender Bedeutung sind, sollten Sie gegebenenfalls einen Anmeldebildschirm einrichten, mit dem der Benutzer aufgefordert wird, sich zunächst am Gerät anzumelden, bevor er es verwenden kann.

Die UI-Klassen bieten einfache Kennwortfelder, in denen die Texteingabe in Form von Sternchen wiedergegeben wird. Anmeldebildschirme könnten sich negativ auf den Bedienungskomfort für die Benutzer auswirken. Wenn der Benutzer ein Kennwort zum Schutz des Geräts festlegt, ist für Ihre Anwendung wahrscheinlich kein Anmeldebildschirm mehr erforderlich.

6.6.2 Serverseitige Authentifizierung

Wenn Ihre Anwendung eine Verbindung zu einer Anwendung auf einem Server beziehungsweise zum Internet oder zu einem Intranet herstellt, sollten gegebenenfalls zusätzliche Authentifizierungsfunktionen eingebunden werden, wenn sich die Benutzer 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 Verbindung geöffnet wird. Außerdem können Sie eine erweiterte Authentifizierung mithilfe von Zertifikaten hinzufügen. Bei den meisten Anwendungen ist dies jedoch nicht erforderlich.

6.6.3 Gesteuerte APIs und Codesignaturen

Research In Motion verfolgt die Verwendung vertraulicher APIs aus Gründen der Sicherheit und der Exportsteuerung. In der API-Referenz für die BlackBerry-Entwick-

lungsumgebung bezeichnet RIM eine gesteuerte Klasse oder Methode mit einem Schlosssymbol oder einem Signaturhinweis.

Um gesteuerte 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 einer Signatur von RIM signieren. Auch andere Funktionen, etwa die Möglichkeit zur Ausführung beim Starten der Anwendung, könnten eine Signierung Ihrer Anwendungen notwendig machen.

Der RIM-Registrierungsprozess umfasst den Großteil der gesteuerten 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 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 brauchen dazu nicht Ihren eigentlichen Code an RIM zu senden. Mit dem Signature-Tool können Sie einen SHA-1-Hash Ihrer Codedatei senden, damit das Signatursystem die erforderliche Signatur generieren kann.

6.7 BlackBerry-APIs mit gesteuertem Zugriff

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

Die folgenden Kategorien von RIM-gesteuerten APIs können verwendet werden:

- Laufzeit-APIs
- Application-APIs
- Kryptografie-APIs

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-gesteuerten APIs finden Sie im Anhang D über die Package Class.

6.8 Unterstützung für IT-Richtlinien

Sie können die APIs im Paket `net.rim.device.api.itpolicy` verwenden, um auf die IT-Richtlinieninformationen zum BlackBerry-Gerät zuzugreifen und das Verhalten oder die Funktionen einer Geräteanwendung zu ändern.

Jedes IT-Richtlinienelement besteht aus einem Namen, einer Beschreibung und einem Wert. Der Wert kann ein Wert vom Typ `string`, `integer` oder `boolean` sein.

Die IT-Richtlinieneinstellungen der Geräte werden automatisch synchronisiert und über das drahtlose Netzwerk (ab der Version 3.6 der Device-Software) aktualisiert. Bei früheren Versionen der Device-Software werden die IT-Richtlinieneinstellungen aktualisiert, sobald der Benutzer das Gerät mit dem Computer synchronisiert.

Weitere Informationen zu IT-Richtlinien finden Sie im Kapitel 10.4.1, »Aktivieren der Unternehmensdienst-Richtlinie«.

6.9 Anwendungssteuerung

Die Richtlinienregeln zur Anwendungssteuerung des Enterprise Servers ermöglichen es, die Installation bestimmter Anwendungen auf dem Gerät zuzulassen oder zu unterbinden, und beschränken die Berechtigungen der Anwendungen auf dem Gerät.

Administratoren können beispielsweise mit der Richtlinie zur Anwendungssteuerung sicherstellen, dass ein Spielprogramm nicht auf die Telefonanwendung zugreifen kann.

Der Administrator kann Richtlinien zur Anwendungssteuerung aber nur dann anwenden, wenn das Gerät einem Enterprise Server zugeordnet ist. Wenn der Administrator oder ein Gerätebenutzer der Anwendung den Zugriff auf einen der geschützten Bereiche verweigert, löst die Anwendung eine Controlled Access Exception aus.

6.10 Dateiverschlüsselung bei MicroSD-Karten

Wenn eine Applikation 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-Applikation auf eine kennwortgeschützte Datei zugreifen kann, darf das Gerät nicht gesperrt sein.

Verschlüsselte Dateien besitzen die Dateinamenerweiterung REM und können auf BlackBerry-fremden Plattformen nicht entschlüsselt werden.

Wenn der NVRAM entfernt 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 Gerät und entfernen alle verschlüsselten Mediendateien.

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

6.11 Verwenden der MicroSD-Medienkarte auf mehreren BlackBerry-Geräten

Wenn der Gerätebenutzer die MicroSD-Karte in ein BlackBerry-Gerät einlegt, bei dem kein Gerätekennwort verwendet wird beziehungsweise das ein Kennwort verwendet, das den Hauptschlüssel der Medienkarte nicht erfolgreich entschlüsselt, fordert das BlackBerry-Gerät den Gerätebenutzer dazu auf, das Kennwort für die MicroSD-Karte einzugeben.

Ist das Gerät mit einem Kennwort geschützt, kann der Benutzer bei dieser Aufforderung das Kennwort der MicroSD-Medienkarte durch das BlackBerry-Gerätekennwort ersetzen.

6.12 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.

6.13 Testvorbereitung einer BlackBerry-Applikation

Der Compiler der Entwicklungsumgebung und das JDE-Plugin 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 Entwicklungsumgebung 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 im Java Development Environment keine zusätzliche Verschleierung für Ihre Anwendungen einrichten. Research In Motion nimmt selbst keine zusätzliche Verschleierung eigener Produkte vor.

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

6.14 Vorabprüfung einer Applikation

Sobald Sie Ihre Anwendung fertig entwickelt und kompiliert haben, können Sie sie auf dem BlackBerry-Gerät testen.

Der erste Schritt besteht meist darin, die Entwicklungsumgebung so einzurichten, dass ein Smartphone-Simulator verwendet wird.

Auf dem Smartphone-Simulator wird derselbe Java-Code wie auf den Geräten ausgeführt, weshalb der Smartphone-Simulator eine exakte Testumgebung bietet, in der die Funktionsweise der Anwendungen auf einem Gerät ermittelt werden kann.

6.15 Testen von Anwendungen auf dem Gerät

Sobald Sie Ihre Anwendung im Smartphone-Simulator getestet haben, können Sie sie auf einem BlackBerry-Gerät installieren. Falls Ihre Anwendung signierte APIs verwendet, benötigen Sie gegebenenfalls Codesignierungsschlüssel.

Nach erfolgter Installation der Anwendung auf dem Gerät können Sie die Anwendung öffnen und ihre Funktionalität und Leistung testen.

Zur Fehlerbehebung können Sie Ihr Gerät mit der Entwicklungsumgebung verbinden und mithilfe des Fehlerhebungs-Tools schrittweise durch die Anwendung gehen. Die IDE ist nützlich, wenn Sie versuchen, ein Netzwerk- oder Bluetooth-Problem festzustellen oder auch andere Probleme, die nur schwierig simuliert werden können.

6.16 Simulatoren

Mit dem Smartphone-Simulator simulieren Sie die Verwendung eines BlackBerry, ohne tatsächlich ein Gerät in Händen zu halten. Der Smartphone-Simulator ist eine Anwendung, die Sie auf Ihrem Computer installieren und 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 Gerät (mit Trackwheel oder Trackball und Tastatur), und auch dieselben Anwendungen und Funktionen stehen bereit, zum Beispiel E-Mail-Nachrichten, Telefon und Internet-Browsing.

Der Smartphone-Simulator dient auch als Plattform zum Ausführen von Anwendungen. Die Anwendungen können hierbei Netzwerkverbindungen herstellen, Daten speichern und E-Mail-Nachrichten verarbeiten.

Der Simulator enthält die Anwendungen, die in der Regel auf den meisten Geräten verfügbar sind, und Sie können Ihre eigenen Anwendungen installieren und testen.

Mit dem Simulator können Sie auch verschiedene Konnektivitäts- und Zustandsänderungen simulieren und testen.

Wenn Sie die Tests im BlackBerry Smartphone Simulator durchführen, müssen Sie gegebenenfalls weitere BlackBerry-Dienste simulieren.

Hierzu stehen der BlackBerry MDS Simulator und der BlackBerry Email Simulator bereit.

6.16.1 BlackBerry MDS Simulator

Der MDS Simulator dient zum Simulieren der MDS Connection Service-Komponente auf dem Enterprise Server. Wenn Sie den Smartphone-Simulator gemeinsam mit dem

MDS Simulator einsetzen, können Sie Netz-, Push-HTTP- und Browser-Anwendungen testen, die für den Einsatz mit einem Enterprise Server entwickelt wurden.

6.16.2 BlackBerry E-Mail Simulator

Mit dem E-Mail-Simulator werden Nachrichten zwischen dem Smartphone Simulator und entweder einer Messaging-Anwendung, zum Beispiel Microsoft Outlook oder einem POP3- oder SMTP-Server, gesendet und empfangen. Ein Enterprise Server ist hierzu nicht nötig.

6.17 Verfügbarmachen von Anwendungen

Mit dem Programm zum Laden von Anwendungen im Desktop Manager können Sie Anwendungen auf dem Gerät installieren.

Das Programm zum Laden von Anwendungen bietet den Benutzern eine einfache Möglichkeit, Anwendungen von Computern auf das Gerät herunterzuladen.

6.17.1 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 den Internet Explorer auf dem Computer aufrufen und die Anwendung auf ihrem BlackBerry-Gerät installieren. Wenn Gerätebenutzer die Webseite besuchen, fordert der Application Web Loader sie dazu auf, das Gerät mit dem USB-Anschluss zu verbinden. Anschließend können sie die Anwendung mithilfe eines ActiveX-Steuerelements installieren.

Der Application Web Loader bietet BlackBerry-Benutzern eine einfache Möglichkeit, Anwendungen vom Computer aus zu installieren, ohne den Desktop Manager ausführen zu müssen.

6.17.2 Verteilen einer Anwendung zu Testzwecken

Die BlackBerry-Entwicklungsumgebung umfasst das Befehlszeilen-Tool mit der Bezeichnung *JavaLoader Tool*, das sich im BIN-Ordner im Verzeichnis der BlackBerry-JDE befindet.

Mit dem *JavaLoader Tool* können Sie kompilierte Anwendungsdateien direkt über den USB-Anschluss auf dem Gerät installieren und wieder vom Gerät entfernen. Deskriptor-dateien oder Webseiten werden dabei nicht benötigt.

Das Tool ist von Nutzen, wenn Sie Ihre Anwendung während der Test- und Entwicklungsphase häufig installieren und wieder entfernen; das Tool ist allerdings nicht für die Verwendung durch BlackBerry-Benutzer vorgesehen.

6.17.3 Anwendungsverteilung über das drahtlose Netzwerk

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

6.17.4 Drahtloses Pull-Verfahren (vom Benutzer eingeleitet)

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

6.17.5 Drahtloses Push-Verfahren (vom Server eingeleitet)

In der Enterprise-Server-Umgebung sind die Administratoren in der Lage, Anwendungen zur erforderlichen Installation per Push-Verfahren über das drahtlose Netzwerk an die Gerätebenutzer zu übermitteln.

Der Administrator erstellt eine neue Richtlinie und gibt an, dass die Anwendung für das Gerät erforderlich ist. Die Anwendung wird per Push-Verfahren an die Benutzer übermittelt, ohne dass diese selbst aktiv werden müssen.

In einem Unternehmen ist diese Vorgehensweise von Nutzen, wenn neue Anwendungen an eine große Gruppe von Benutzern gesendet werden sollen.

Mehr zu diesem Thema erfahren Sie im Kapitel 9 »Admin-Guide für den BlackBerry Enterprise Server (BES)«.

7 UI-Guidelines

Verwenden Sie als UI-Elemente für Ihre Applikation einfache Grafiken, die übersichtlich Ihre Applikationen, Aktionen oder Steuerelemente (Controls) darstellen. Verwenden Sie Images, die einen Bezug zu Ihren Applikationen herstellen, wie zum Beispiel E-Mail-Icons.

Beachten Sie, dass Grafiken unterschiedliche Bedeutungen in unterschiedlichen Kulturen haben können. Ein Briefumschlag oder ein Posthorn werden nicht überall gleich verstanden.

7.1 Richtlinien für Icons

Verwenden Sie einfache Perspektiven für Ihre Icons. Richten Sie Ihre Icons mit anderen Applikationen, die im selben Kontext stehen, aus. Planen Sie etwas Raum um jedes Ihrer Icons ein – als Indikator und als Abgrenzung zum Hintergrund.

Wenn zum Beispiel ein Icon mit den Abmessungen 80 x 80 Pixel erscheinen soll, dann entwerfen Sie das Icon-Objekt mit den Abmessungen 68 x 68 Pixel. Begrenzen Sie zudem die Farbpalette.

Benutzen Sie Farbmischungen, die einfach sind. Vermeiden Sie Text und Ziffern in Icons und Grafiken, denn sie erfordern unter Umständen eine Lokalisation.

7.2 Designhinweise für BlackBerry-Geräte

Bevor wir noch tiefer in die Programmierung einsteigen, machen wir einen kurzen, aber wichtigen Exkurs in die Grafik der BlackBerry-Geräte.

Applikationen, die für den BlackBerry entworfen und entwickelt werden, stellen eine Balance zwischen der bestmöglichen User-Erfahrung und einer langen Betriebsdauer des Gerätes dar. Wenn Sie eine Applikation entwickeln, beachten Sie die wesentlichen Unterschiede zwischen Computer und mobilen Geräten (Mobile Devices).

Die grundsätzlichen Eigenschaften von Mobile Devices lassen sich folgendermaßen definieren:

- Sie besitzen ein kleineres Display, und somit begrenzt sich der Zeichensatz in der Darstellung.
- Sie haben in der Regel eine geringere Prozessorleistung.

- Sie benutzen ein drahtloses Netzwerk mit langsameren Übertragungsgeschwindigkeiten.
- Sie haben weniger Arbeitsspeicher.
- Sie haben eine begrenzte Batteriestandzeit.
- Sie zeigen nur einen Screen an. Fenster lassen sich nicht verschieben.
- Benutzer mobiler Geräte verwenden ihre Applikationen in der Regel anders als Applikationen auf einem PC.
- Auf mobilen Geräten erwarten die User, schneller an Informationen zu kommen. Zum Beispiel könnte ein CRM-System (Customer Relationship Management) angeboten werden. Eine große Menge von Informationen steht in diesem System zur Verfügung; jedoch benötigt der Benutzer nur eine kleine Menge dieser Informationen auf einmal.
- Das BlackBerry-Device-UI (User Interface) ist so entworfen, dass der User die Aufgaben leicht aufteilen kann und schnell an die Informationen kommt, die er benötigt.
- Wenn Sie eine Applikation entwickeln, betrachten Sie im Smartphone-Simulator, ob bestimmte Aktionen sinnvoll sind und Anwender schnell an die Informationen gelangen, die Sie zur Verfügung stellen möchten.
- Im Simulator erfahren Sie, wie sich Ihre Applikation später auf dem Gerät verhalten wird und wie Sie durch die Applikation navigieren können.

Betrachten Sie auch die BlackBerry Browser Content Design Guidelines¹¹.

¹¹http://docs.blackberry.com/en/developers/deliverables/6622/BlackBerry_Smartphones-US.pdf

8 Die BlackBerry-API

Die Anwendungsumgebung für den BlackBerry basiert auf zahlreichen bewährten Standards aus der Java-Welt. Dazu zählen CLDC (Connected Limited Device Configuration) und MIDP (Mobile Information Device Profile), die zwei Eckpfeiler der Java Mobile Edition bilden. Darüber hinaus müssen BlackBerry-Entwickler eine Reihe spezifischer Programmierschnittstellen kennen und beachten, um die Eigenschaften der Hardware voll auszunutzen. Dieses Kapitel versucht gibt einen kurzen Überblick.

8.1 Beschreibung der JDE

Die Java-Entwicklungsumgebung für den BlackBerry, kurz JDE, wird mit einem umfangreichen Set von APIs und Tools für Entwicklungsaufgaben ausgeliefert. Applikationen benutzen die Version 4.0 der JDE. Auf BlackBerry-Handhelds gibt es je nach Modell die Softwareversion 3.8 oder 4.0. Zudem verfügt der BlackBerry über eine J2ME- (Java 2 Mobile Environment) Laufzeit-Umgebung, die auf den Spezifikationen CLDC 1.1 und MIDP 2.0 beruht. Die BlackBerry-Softwarekomponenten sind in Abbildung 8.1 als Blockdiagramm dargestellt. Die API-Dokumentation befindet sich im Eclipse-Plugin-Verzeichnis unter *C:\Programme\Eclipse\plugins\net.rim.eide.componentpack4.5.0_4.5.0.16*.

Die APIs beschreiben die Eigenschaften der User-Interfaces für Networking, Lokalisierung (GPS) und andere Einsatzmöglichkeiten.

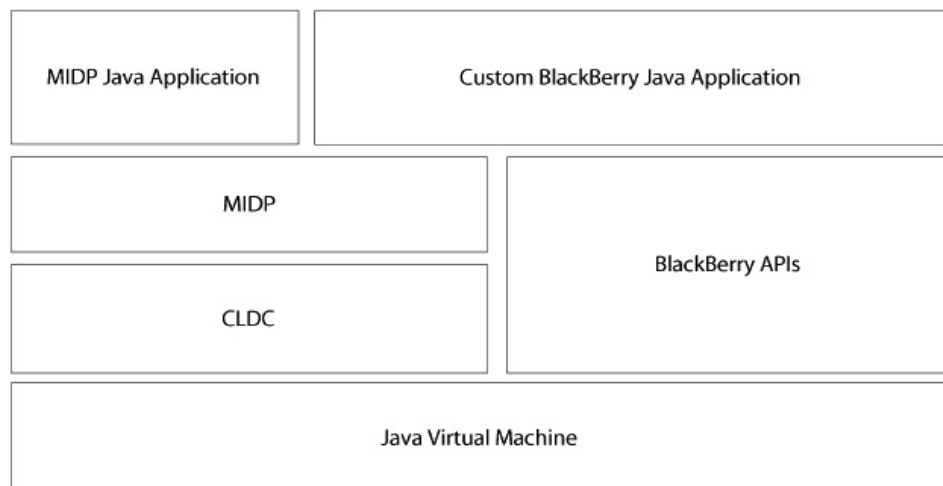


Abbildung 8.1: BlackBerry-Softwarekomponenten

8.2 BlackBerry-API-Package

Ich möchte hier einige der API-Packages vorstellen und erklären.

Tabelle 8.1: API-Packages

<i>BlackBerry-API-Package</i>	<i>Beschreibung</i>
<i>net.rim.blackberry.api.browser</i>	Dieses Package erlaubt den Applikationen, den BlackBerry-Browser aufzurufen.
<i>net.rim.blackberry.api.invoke</i>	Dieses Package erlaubt es Applikationen, Zugriff auf BlackBerry-Apps zu nehmen, wie Aufgaben, Messages, Notizen und Telefon.
<i>net.rim.blackberry.api.mail</i>	Dieses Package ist notwendig, um die Komponenten des internen RIM-E-Mail-Systems in portable Objekte zu konvertieren, damit diese kompatibel zur Mail-API sind. Ebenfalls werden Funktionalitäten mitgegeben, um Mails zu senden oder zu empfangen sowie den Zugriff auf das Mailsystem ermöglichen.
<i>net.rim.blackberry.api.mail.event</i>	Dieses Package definiert Messaging Events und Listener-Schnittstellen, um Mailereignisse zu managen.
<i>net.rim.blackberry.api.menuitem</i>	Dieses Package erlaubt es dem Entwickler, in seinen BlackBerry-Applikationen Menu-Items zu gestalten, so z. B. das Hinzufügen von persönlichen Menu-Items wie Adressbuch, Kalender und Messages.
<i>net.rim.blackberry.api.options</i>	Dieses Package erlaubt dem Entwickler das Hinzufügen von Einträgen in den Handheld-Optionen.
<i>net.rim.blackberry.api.pdap</i>	Dieses Package erlaubt dem Entwickler die Interaktion mit PIM-Applikationen (persönliches Informationsmanagement), einschließlich Adressbuch, Aufgaben und Kalender. Die meisten Funktionen werden durch das MIDP-Paket <code>javax.microedition.pim</code> bereitgestellt.
<i>net.rim.blackberry.api.phone</i>	Dieses Package erlaubt den Zugriff auf erweiterte Eigenschaften der Phone-Applikation.
<i>net.rim.blackberry.api.phone.phonelogs</i>	Dieses Package erlaubt den Zugriff auf die Ruflisten-History.
<i>net.rim.device.api.bluetooth</i>	Dieses Package erlaubt Applikationen die Kommunikation mit drahtlosen Geräten über den seriellen Bluetooth-Port.
<i>net.rim.device.api.browser.field</i>	Dieses Package erlaubt es Applikationen, innerhalb ihres User-Interfaces ein Browserfeld anzuzeigen.
<i>net.rim.device.api.browser.plugin</i>	Dieses Package erlaubt es dem Entwickler, MIME-Support für zusätzliche MIME-Typen einzusetzen.

<i>BlackBerry-API-Package</i>	<i>Beschreibung</i>
<i>net.rim.device.api.collection</i> <i>net.rim.device.api.collection.util</i>	Dieses Package definiert Interfaces und Utility-Klassen, um Datenkollektionen zu handhaben.
<i>net.rim.device.api.compress</i>	Dieses Package erlaubt die Verwendung von Kompressionstools, um Daten in GZIP- und ZLIB-Formate zu komprimieren. Dekompression ist zurzeit noch nicht implementiert. Handhelds können momentan GZIP- und ZLIB- Dateien generieren, jedoch ist der Inhalt nicht gepackt.
<i>net.rim.device.api.i18n</i>	Dieses Package erlaubt die Verwendung von Klassen, um Applikationen zu lokalisieren.
<i>net.rim.device.api.io</i>	Dieses Package erlaubt die Verwendung einer Library-Klasse, um die Datenein- und -ausgabe zu managen.
<i>net.rim.device.api.mime</i>	Dieses Package erlaubt das Arbeiten mit MIME-entschlüsselten Datenströmen.
<i>net.rim.device.api.notification</i>	Dieses Package unterstützt Methoden, um Ereignisse anzustoßen, wie z. B. Alarmnotifikationen.
<i>net.rim.device.api.servicebook</i>	Dieses Package erlaubt Applikationen das Hinzufügen und Löschen sowie den Zugriff auf das Servicebuch.
<i>net.rim.device.api.system</i>	Dieses Package erlaubt den Zugriff auf System-Level-Funktionalität, inklusive Event Listener des Handhelds für Trackwheel, Tastatur und Touchscreen, Bilderzeugung und Programmkontrolle.
<i>net.rim.device.api.ui</i>	Dieses Package erlaubt den Zugriff auf die erweiterten Funktionalitäten auf das BlackBerry User Interface, inklusive Screen- und Feld-Layout-Manager, um Funktionen wie Fokus, Scroll und »Change Listeners« anzusprechen.
<i>net.rim.device.api.ui.component</i>	Dieses Package erlaubt den Zugang auf die Library der Interface-Komponenten, um UI-Applikationen zu erzeugen.
<i>net.rim.device.api.ui.container</i>	Dieses Package erlaubt den Zugriff auf den Interface-Manager, um UI-Applikationen zu erzeugen.
<i>net.rim.device.api.ui.text</i>	Dieses Package erlaubt den Zugriff auf die Klasse, um Text zu filtern. Es geht dabei um Textstrings, die unterschiedliche Daten enthalten, wie z. B. Telefonnummern oder URLs.
<i>net.rim.device.api.util</i>	Dieses Package erlaubt den Zugriff auf die Utility-Methoden und Interfaces, die hilfreich im Kontext des Handheld-Systems stehen, wie Klassen für Arrays, Hash-Tabellen und Zeichenkettenvergleiche.

8.3 CLDC-APIs

CLDC (*Connected Limited Device Configuration*) definiert die kleinstmögliche Konfiguration einer J2ME-Laufzeitumgebung. Zu den größten Einschränkungen in Version 1.0 zählt der Verzicht auf Gleitkommaberechnungen.

Das betrifft sowohl Variablen, die als `float` und `double` sowie als `java.lang.Float` und `java.lang.Double` deklariert sind. Die Weiterentwicklung der CLDC in Version 1.1 führte unter anderem zu einer Wiedereinführung der Gleitkommaunterstützung. Insbesondere Spieleentwickler sind gelegentlich auf Gleitkommaberechnungen angewiesen – wenngleich aus Performance-Erwägungen in vielen Fällen doch eher Festkommazahlen bevorzugt werden.

Tabelle 8.2: CLDC-Package

CLDC-API-Package	Beschreibung
<code>java.io</code>	Dieses Package erlaubt dem Entwickler den Zugriff auf die Systemeingaben und -ausgaben über Datenströme.
<code>java.lang</code>	Dieses Package ist fundamental in der Java-Programmiersprache. Es stellt die grundlegenden Klassen bereit.
<code>java.lang.ref</code>	Dieses Package stellt sogenannte Referenzobjekt-Klassen bereit, die einen limitierten Grad der Interaktion mit der Garbage Collection erlauben.
<code>java.util</code>	Dieses Package beinhaltet die Kollektion der Klassen <code>date</code> und <code>time</code> und diverse <code>utility</code> -Klassen.
<code>javax.microedition.io</code>	Dieses Package beinhaltet die Klassen für generische Verbindungen.

8.4 MIDP-APIs

MIDP (*Mobile Information Device Profile*) ist ein Profil der *Java Micro Edition* (Java ME), das speziell auf die Fähigkeiten kleiner Mobilgeräte wie Mobiltelefone oder PDAs ausgelegt ist. Es umfasst Funktionen zur Ansteuerung und Abfrage von ITU-T-Einhandtastaturen, Miniaturbildschirmen, flüchtigem und nicht-flüchtigem Speicher im Kilobyte-Bereich und Ähnliches.

MIDP-Applikationen heißen MIDlets.

Es existieren bisher das MIDP 1.0 und das MIDP 2.0 beziehungsweise 2.1, das einige Erweiterungen aufweist. MIDP 3.0 ist bereits als JSR 271 in Bearbeitung.

MIDP ist ein Sandbox-Modell, was große Sicherheit gegenüber Computerviren oder Hackerangriffen gewährleistet. Der Benutzung von Netzwerkeinrichtungen geht eine Bestätigung vom Benutzer voraus, der sie explizit für jedes MIDlet bei erforderlichlichem Verbindungsaufbau erlauben muss.

Viren oder vergleichbare Programme sind laut Sun Microsystems ebenfalls nicht möglich, da das Dateisystem außerhalb der Sandbox liegt und darauf nur sehr eingeschränkt zugreifbar ist.

Tabelle 8.3: MIDP-API-Package

<i>MIDP-API-Package</i>	<i>Beschreibung</i>
<i>javax.microedition.lcdui</i>	Dieses Package beinhaltet die MIDP-UI-API. Hiermit lässt sich ein Set an Eigenschaften für die Realisierung eines User-Interfaces für MIDP-Applikationen generieren.
<i>javax.microedition.lcdui.game</i>	Dieses Package beinhaltet Klassen für die Spieleentwicklung für drahtlose Geräte.
<i>javax.microedition.midlet</i>	Dieses Package definiert die MIDP-(Mobile Information Device Profile)Applikationen und die Interaktion zwischen den Applikationen.
<i>javax.microedition.pki</i>	Dieses Package definiert die Zertifikate, die zum Authentifizieren von »secure connections« dienen (Verschlüsselte Verbindungen).
<i>javax.microedition.rms</i>	Dieses Package beinhaltet einen Mechanismus für MID-Lets, um persistente Daten zu speichern und zu lesen.

8.5 PDAP-API

PDAP steht für Personal Digital Assistant Profile.

Tabelle 8.4: MIDP-/ PDAP-API-Package

<i>MIDP-API-Package</i>	<i>Beschreibung</i>
<i>javax.microedition.pim</i>	Dieses Package beinhaltet den Standardmechanismus für PIM-Informationen.

8.6 Java-Code in BlackBerry-Geräten

Der Java-Quellcode wird kompiliert, gepackt und in .cod-Dateien geschrieben. Die .cod-Dateien werden auf das BlackBerry-Gerät geladen und unter der virtuellen Maschine (VM) zur Ausführung gebracht.

Die BlackBerry-Entwicklungsumgebung benutzt dabei nur einen Teil der VM-Architektur, wie in der CLDC-Spezifikation beschrieben. Der Grund hierfür liegt in der begrenzten Speichermenge sowie der limitierten Verarbeitungsgeschwindigkeit. Darüber hinaus erfolgt eine Vorprüfung, bevor der Java-Code in das Mobile geladen wird. Die IDE analysiert den Quellcode automatisch, bevor dieser in ein .cod-File gepackt wird.

Die VM übernimmt den Rest der Prüfung, während die Klassen in das Device geladen werden.

8.7 Restriktionen

Die BlackBerry-VM verwendet die folgenden Restriktionen, die in der CLDC 1.1 spezifiziert wurden:

- keine Objekt-Komplettierung,
- keine `java.lang.Error`-Klassenhierarchie,
- kein »user class loading«,
- keine Reflexion, demnach kein Support der Remote Method Invocation (RMI) oder Jini-Netzwerktechnologie,
- keine sogenannten nativen Methoden,
- keine `Runtime.exec()` zur Ausführung externer Prozesse.

8.8 Multithreading

Die BlackBerry-Java-Umgebung unterstützt eine echte Multithreading-Umgebung der laufenden Applikationen. Das erlaubt es Ihnen, multiple Applikationen simultan, also gleichzeitig zu starten. Ereignisse werden den multiplen Applikationen mitgeteilt. Große Operationen oder Event-basierte Threads laufen dadurch im Hintergrund ab.

9 Admin-Guide für den BlackBerry Enterprise Server (BES)

Dieses Kapitel ist für den Administrator sehr interessant. Hier beschreibe ich, wie eine Integration des BlackBerry in die IT-Infrastruktur mit BES funktioniert. Sie erfahren unter anderem, wie das Rollenmanagement funktioniert und weitere Anpassungen erfolgen.

Die folgende Tabelle soll den Vergleich der beiden Services BIS und BES aufzeigen:

Tabelle: 9.1 Vergleich der Services BIS und BES

<i>Eigenschaft</i>	<i>BlackBerry Internet Service</i>	<i>BlackBerry Enterprise Server</i>
Drahtlose Kalender-synchronisation	Nein	Ja
Computer muss ständig in Betrieb sein	Nein	Nein
Computer benötigt eine permanente Verbindung in das Internet	Nein	Nein
Unterstützte Protokolle	Post Office Protokoll POP Internet Message Access Protokoll (IMAP) Windows Live™ Hotmail Microsoft MSN AOL CompuServe Gmail Yahoo! Mail Microsoft Outlook	IBM Lotus Domino Microsoft Exchange Novell GroupWise
Desktop E-Mail Application Support	IBM Lotus Domino Microsoft Exchange	IBM Lotus Domino Microsoft Exchange Novell GroupWise

<i>Eigenschaft</i>	<i>BlackBerry Internet Service</i>	<i>BlackBerry Enterprise Server</i>
Konfigurierbare Filter	Ja	Ja
E-Mail-Umleitungen	POP IMAP Windows Live Hotmail MSN AOL CompuServe Gmail Yahoo! Mail Microsoft Outlook Web Access	BlackBerry Enterprise Server
Datei-Verzeichnismanagement	Nein	Ja
Backup Utilities	Ja	Ja
Autosignatur	Ja	Ja
Sicherheit	E-Mail-Sendungen zwischen dem BlackBerry-Internetservice und verwendetem Smartphone werden <i>nicht</i> verschlüsselt. Bei einer Übertragung durch das drahtlose Netzwerk werden E-Mail-Nachrichten gemäß der vorhandenen Sicherheitsmodelle des Netzwerkes behandelt. Loggt sich der User in den BlackBerry-Internetservice ein, so werden die Daten über einen Secure Socket Layer (SSL) übertragen.	E-Mail-Sendungen zwischen dem BlackBerry Smartphone und dem Enterprise Server werden durch »Triple Data Encryption« (Triple DES) sowie Advanced Encryption Standard (AES) oder beide Verschlüsselungsarten chiffriert. Wichtig: Versionen des BlackBerry Enterprise Server für Novell Groupwise verwenden nur die Verschlüsselungsmethode AES.

9.1 Verknüpfen von Unternehmensrollen mit BlackBerry-Rollen

Der BlackBerry Enterprise Server verwendet vordefinierte Rollen, die den üblichen Verwaltungsrollen in Unternehmen entsprechen. Diese Rollen steuern, wer bestimmte Aufgaben ausführen und auf vertrauliche Daten Ihres Unternehmens zugreifen kann.

Wenn Sie in Ihrem Unternehmen bereits Windows-Benutzergruppen zu Verwaltungszwecken einsetzen, weisen Sie diese Gruppen den Verwaltungsrollen zu, damit Sie die Rollenzugehörigkeit über die Gruppe verwalten können.

Wenn Sie den BlackBerry Manager starten, prüft dieser die Authentifizierungsinformationen, bestimmt Ihre Verwaltungsrolle und zeigt eine Liste der Aufgaben an, die Sie ausführen können.

9.2 Verwalten von Verwaltungsrollen

Tabelle 9.2: Verwaltungsrollen

<i>Rolle</i>	<i>Beschreibung</i>
Sicherheitsadministrator (<i>rim_db_admin_security</i>)	Diese Administratoren können sämtliche Aufgaben ausführen. Sie sind die einzigen Administratoren, die die Rollenzugehörigkeit verwalten und sensible Sicherheitseigenschaften wie Lizenzen und Codierungsschlüssel ändern können.
Unternehmensadministrator (<i>rim_db_admin_enterprise</i>)	Diese Administratoren können alle Aufgaben im Zusammenhang mit Benutzerkonten, Diensten, dem BlackBerry Enterprise Server und globalen Anwendungsdaten ausführen. Sie können keine Rollenzugehörigkeiten, Lizenzen oder Codierungsschlüssel anzeigen.
Geräteadministrator (<i>rim_db_admin_handheld</i>)	Diese Administratoren können alle Aufgaben im Zusammenhang mit Benutzerkonten und der BlackBerry-Geräteverwaltung ausführen. Dazu zählt das Unterstützen neuer Benutzerkonten, das Implementieren von BlackBerry-Devices (BlackBerry-Geräten), das Verwalten von Softwarekonfigurationen sowie das Verwalten der Installation und des Verhaltens von Drittanbieteranwendungen auf BlackBerry-Geräten.
Senior-Helpdesk-Administrator (<i>rim_db_admin_sr_helpdesk</i>)	Diese Administratoren können sämtliche Aufgaben zur Benutzerkontenverwaltung ausführen. Dazu zählen das Hinzufügen, Verschieben und Löschen von Benutzerkonten, das Aktualisieren und Senden von IT-Richtlinien an BlackBerry-Geräte und das Senden von IT-Verwaltungsbefehlen an BlackBerry-Geräte.
Junior-Helpdesk-Administrator (<i>rim_db_admin_jr_helpdesk</i>)	Diese Administratoren können Aufgaben zur Benutzerkontenverwaltung ausführen. Dazu zählt das Erstellen und Senden von Kennwörtern zur drahtlosen Unternehmensaktivierung sowie das erneute Senden von Servicebüchern oder IT-Richtlinien. Die Administratoren können keine Benutzerkonten hinzufügen, verschieben oder löschen oder bestimmte IT-Verwaltungsbefehle senden.
Schreibgeschützt (<i>rim_db_admin_audit_<Rolle></i>)	Diese Gruppe kann alle zu ihrer Rolle gehörenden Aufgaben und Eigenschaften anzeigen, jedoch weder die Aufgaben ausführen noch die Eigenschaften ändern. Verwenden Sie diesen schreibgeschützten Zugriff für sämtliche Rollen während der Schulung neuer Administratoren.

9.3 Hinzufügen von Datenbankbenutzern zu Verwaltungsrollen

Weisen Sie den Verwaltungsrollen Datenbankbenutzer auf der Grundlage der bestehenden Aufgabenverteilung im Unternehmen zu.

Für das Erstellen von Datenbankbenutzern mit dem BlackBerry Manager benötigen Sie auf dem Datenbankserver Systemadministratorrechte.

Wenn Sie nur den Benutzernamen eingeben, erstellen Sie eine SQL-Anmeldung. Wenn Sie dem Namen einen Domänennamen voranstellen (etwa DOMÄNE\Benutzername), erstellen Sie eine SQL-Anmeldung für einen Windows-Benutzer oder eine Windows-Gruppe. Fügen Sie einen Datenbankbenutzer immer nur einer Verwaltungsrolle hinzu. Die BlackBerry Configuration Database verwendet die restriktivsten Einstellungen, um zu bestimmen, welche Aufgaben der BlackBerry Manager anzeigt. Wenn also ein Datenbankbenutzer gleichzeitig Unternehmensadministrator und Junior-Helpdesk-Administrator ist, werden beispielsweise nur die Aufgaben für den Junior-Helpdesk-Administrator angezeigt.

9.3.1 Hinzufügen eines Datenbankbenutzers zu einer Verwaltungsrolle

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Klicken Sie auf der Registerkarte *Role Administration* (Rollenverwaltung) auf einen Rollennamen.
3. Führen Sie eine der folgenden Aktionen aus:

Tabelle 9.3: Anlegen einer Datenbankbenutzerrolle

Aktion	Vorgehensweise
Hinzufügen eines vorhandenen Datenbankbenutzers zur Verwaltungsrolle	1. Klicken Sie auf <i>List Administrators</i> (Administratoren auflisten). 2. Klicken Sie den Datenbankbenutzer an, der der Rolle hinzugefügt werden soll. 3. Klicken Sie auf <i>OK</i>.
Erstellen eines neuen Datenbankbenutzers und Zuweisen dieses Benutzers zur Verwaltungsrolle	1. Klicken Sie auf <i>Add Administrators</i> (Administratoren hinzufügen). 2. Geben Sie einen neuen Anmeldenamen ein. 3. Geben Sie ein neues Kennwort ein. 4. Bestätigen Sie das neue Kennwort.

4. Klicken Sie auf *OK*.

9.4 Festlegen der Authentifizierung des BlackBerry Managers beim Datenbankserver

Der BlackBerry Manager akzeptiert standardmäßig automatisch die Windows-Authentifizierungsinformationen, mit denen Sie sich bei Ihrem Computer anmelden. Wenn Sie Verwaltungsrollen SQL-Anmeldungen zuweisen, müssen Sie den vom BlackBerry Manager akzeptierten Typ der Authentifizierungsinformationen ändern.

9.4.1 Verwenden von Datenbank-Authentifizierungsinformationen

1. Klicken Sie im BlackBerry Manager im Menü *Tools* (Extras) auf *Options* (Optionen).
2. Klicken Sie auf *Database* (Datenbank).
3. Klicken Sie in der Dropdown-Liste *Authentication* (Authentifizierung) auf *Database Authentication* (Datenbankauthentifizierung).
4. Klicken Sie auf *OK*.
5. Schließen Sie den BlackBerry Manager.
6. Öffnen Sie den BlackBerry Manager erneut.

9.5 Anpassen von Verwaltungsrollen

Im Falle von Veränderungen im Unternehmen müssen Sie einen Datenbankbenutzer unter Umständen aus einer Verwaltungsrolle entfernen oder in eine neue Verwaltungsrolle verschieben.

Wenn Sie einen Datenbankbenutzer in eine neue Verwaltungsrolle verschieben, ändern sich die Datenbankrechte sofort. Die Datenbankbenutzer müssen den BlackBerry Manager neu starten, damit die mit der neuen Verwaltungsrolle verknüpften Aufgaben aktualisiert werden. Wenn die Datenbankanwender den BlackBerry Manager nicht neu starten, hat dies unvorhersehbare Auswirkungen.

9.5.1 Ändern einer Verwaltungsrolle

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Klicken Sie auf der Registerkarte *Role Administration* (Rollenverwaltung) auf die Rolle, der der Datenbankbenutzer zugewiesen wurde.
3. Führen Sie eine der folgenden Aktionen aus:

Tabelle 9.4: Verwalten einer Verwaltungsrolle

Aktion	Vorgehensweise
Verschieben eines Datenbankbenutzers in eine andere Verwaltungsrolle	1. Klicken Sie auf <i>List Administrators</i> (Administratoren auflisten). 2. Klicken Sie auf die Verwaltungsrolle, die jetzt für den Datenbankbenutzer verwendet werden soll. 3. Wählen Sie den Datenbankbenutzer aus. 4. Klicken Sie auf <i>OK</i>. 5. Fordern Sie den Datenbankbenutzer auf, den BlackBerry Manager neu zu starten.
Entfernen eines Datenbankbenutzers aus einer Verwaltungsrolle	1. Klicken Sie auf <i>Remove Administrators</i> (Administratoren entfernen). 2. Klicken Sie in der Dropdown-Liste auf den Datenbankbenutzer. 3. Klicken Sie auf <i>OK</i>.

4. Klicken Sie auf *OK*.

9.6 Einrichten der BlackBerry-Umgebung

- Auswählen eines Verschlüsselungsalgorithmus.
- Erweitern der Gerätedatenverschlüsselung.
- Ersetzen globaler Verschlüsselung von PIN-to-PIN-Nachrichten durch unternehmensspezifische Verschlüsselung.
- Konfigurieren einer BlackBerry-Komponente zur Verwendung eines Proxy-Servers.
- Verknüpfen einer BlackBerry-Komponente mit mehreren BlackBerry Enterprise Servern.
- Konfigurieren der Unterstützung der Adressensuche in einer gehosteten Umgebung.

9.6.1 Auswählen eines Verschlüsselungsalgorithmus

Ab dem Zeitpunkt, zu dem der Benutzer eine Nachricht sendet, bis zu dem Zeitpunkt, zu dem die Nachricht auf dem BlackBerry Enterprise Server eingeht, ist die Nachricht durch die BlackBerry-Standardverschlüsselung verschlüsselt.

Tabelle 9.5: Verschlüsselungstypen

<i>Verschlüsselungstyp</i>	<i>Beschreibung</i>	<i>Anmerkung</i>
Triple DES	Ermöglicht die Verwendung des Triple Data Encryption Standard Algorithmus (Triple DES oder 3DES), um die gesamte Datenkommunikation zwischen dem Enterprise Server und allen BlackBerry-Geräten auf dem BlackBerry Enterprise Server zu ver- und entschlüsseln.	• Standardverschlüsselungsmethode
AES	Ermöglicht die Verwendung des Advanced Encryption Standard Algorithmus (AES), um die gesamte Datenkommunikation zwischen dem Enterprise Server und allen BlackBerry-Geräten auf dem BlackBerry Enterprise Server zu ver- und entschlüsseln.	• Verwendet einen längeren Codierungsschlüssel für höhere Sicherheit und Leistung als Triple DES. • Schützt Benutzerdaten und Codierungsschlüssel vor herkömmlichen Angriffen und Seitenkanalangriffen. • Erfordert die BlackBerry-Desktop-Software Version 4.0 oder höher und die Gerätesoftware 4.0 oder höher.
Triple DES und AES	Ermöglicht die Verwendung des Triple DES- und AES-Algorithmus, um die gesamte Datenkommunikation zwischen dem Enterprise Server und allen BlackBerry-Geräten auf dem Enterprise Server zu ver- und entschlüsseln.	• Ermöglicht die Triple DES-Verschlüsselung auf Geräten, die AES nicht unterstützen (BlackBerry-Gerätesoftware vor Version 4.0) • Bietet AES-Verschlüsselung standardmäßig auf Geräten, die AES unterstützen.

Festlegen eines Verschlüsselungstyps

Wenn Sie den Verschlüsselungsalgorithmus ändern, müssen Sie alle BlackBerry-Geräte in der BlackBerry-Domain erneut aktivieren. Nur so können Gerätebenutzer wieder Nachrichten senden und empfangen.

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen BlackBerry Enterprise Server.
2. Klicken Sie auf der Registerkarte *Server Configuration* (Serverkonfiguration) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *General* (Allgemein).

4. Klicken Sie im Bereich *Security* (Sicherheit) auf *Encryption Algorithm* (Verschlüsselungsalgorithmus).
5. Klicken Sie in der Dropdown-Liste auf einen der folgenden Codierungsschlüssel:
 - *Triple DES*
 - *AES*
 - *Triple DES und AES*
6. Klicken Sie auf OK.

9.6.2 Erweitern der BlackBerry-Gerätedatenverschlüsselung

Eine zusätzliche Technologie für digitale Signaturen und Verschlüsselung gewährleistet die Authentifizierung zwischen Sender und Empfänger sowie die Vertraulichkeit.

Darüber hinaus trägt sie zur Aufrechterhaltung der Datenintegrität und des Datenschutzes ab dem Zeitpunkt bei, zu dem der Verfasser seine Nachricht über das drahtlose Netzwerk versendet, bis zu dem Zeitpunkt, zu dem der Empfänger die Nachricht entschlüsselt und liest.

Schützen von Daten mithilfe des PGP Support Package

Wenn Sie Daten, die vom BlackBerry-Gerät mithilfe des PGP Support Package an den Enterprise Server gesendet werden, signieren und/oder verschlüsseln möchten, müssen Sie in der den Benutzern zugewiesenen IT-Richtlinie die IT-Richtlinienregel für die PGP-Universal-Server-Adresse festlegen. Außerdem müssen alle Benutzer das PGP Support Package auf ihrem Gerät installieren und sich beim PGP Universal Server registrieren.

Wenn der Enterprise Server die IT-Richtlinie an die BlackBerry-Geräte überträgt, auf die Sie die IT-Richtlinie angewendet haben, und die PGP Universal Server-Adressenregel auf den PGP-Universal-Server-URL eingestellt ist, werden die Benutzer, die das PGP Support Package installiert haben, zur Registrierung bei PGP aufgefordert.

Mit dem PGP Support Package 4.2 können Benutzer Nachrichten auf BlackBerry-Geräten, auf denen das PGP Support Package installiert ist, im PGP/MIME-Format empfangen.

Benutzer, die das PGP Support Package und das S/MIME Support Package auf ihrem Gerät installiert und aktiviert haben, können vom PGP Universal Server PGP-Schlüssel mit S/MIME X.509-Zertifikaten herunterladen und entsprechend der PGP-Universal-Server-Richtlinie für E-Mail-Sicherheit verwenden.

Das PGP Support Package unterstützt auch weiterhin Nachrichten im OpenPGP-Format.

Schützen von Daten mithilfe des S/MIME Support Package

Wenn Sie Daten, die vom BlackBerry-Gerät mithilfe des S/MIME Support Package an den Enterprise Server gesendet werden, digital signieren und/oder verschlüsseln möchten,

- müssen Sie auf dem BlackBerry Enterprise Server die S/MIME-Nachrichtenverarbeitung aktivieren,
- müssen die Benutzer das S/MIME Support Package auf ihrem BlackBerry-Gerät installieren und den Zertifikatsynchronisierungsmanager dem Desktop Manager hinzufügen.

Aktivieren der S/MIME-Nachrichtenverarbeitung auf dem BlackBerry Enterprise Server

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *Servers* (Server).
2. Klicken Sie auf der Registerkarte *Server Configuration* (Serverkonfiguration) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Messaging* (Nachrichtenübermittlung).
4. Klicken Sie im Bereich *Secure Messages* (sichere Nachrichten) auf *Enable S/MIME Message Processing* (S/MIME-Nachrichtenverarbeitung aktivieren).
5. Klicken Sie in der Dropdown-Liste auf *True* (Wahr).
6. Klicken Sie auf *OK*.

Festlegen weiterer S/MIME-Verschlüsselungsoptionen

Die ersten drei Schritte erfolgen analog zu oben. Im Bereich *Secure Messages* (sichere Nachrichten) legen Sie allerdings anstelle von *Enable S/MIME Message Processing* die gewünschten Verschlüsselungsoptionen fest. Klicken Sie danach auf *OK*.

9.6.3 Ersetzen globaler Verschlüsselung von PIN-to-PIN-Nachrichten durch unternehmensspezifische Verschlüsselung

Alle BlackBerry-Geräte besitzen standardmäßig einen globalen Peer-to-Peer-Codierungsschlüssel.

Sie können die Anzahl der Geräte begrenzen, die PIN-Nachrichten entschlüsseln können, die Benutzer Ihres Unternehmens von ihrem BlackBerry-Gerät aus gesendet haben, indem Sie einen neuen Peer-to-Peer-Codierungsschlüssel erstellen, der nur den Geräten Ihres Unternehmens bekannt ist.

Geräte mit einem unternehmensspezifischen Peer-to-Peer-Codierungsschlüssel können PIN-Nachrichten an andere BlackBerry-Geräte mit demselben Codierungsschlüssel des Unternehmens senden und Nachrichten von diesen Geräten empfangen.

Erstellen Sie einen neuen Peer-to-Peer-Codierungsschlüssel für das Unternehmen, wenn der aktuelle Schlüssel außerhalb des Unternehmens bekannt geworden ist:

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Klicken Sie auf der Registerkarte *Global* auf *Service Control & Customization* (Dienststeuerung und -anpassung).
3. Klicken Sie auf *Update Peer-to-Peer Encryption Key* (Peer-to-Peer-Codierungsschlüssel aktualisieren).
4. Klicken Sie auf *Set or update the Peer-to-Peer encryption key for all devices within this organization* (Peer-to-Peer-Codierungsschlüssel für alle Geräte innerhalb dieses Unternehmens einrichten oder aktualisieren).
5. Klicken Sie auf *Yes* (Ja).

9.6.4 Konfigurieren einer BlackBerry-Komponente zur Verwendung eines Proxy-Servers

Erstellen Sie Proxy-Zuordnungsregeln für den BlackBerry MDS Connection Service, den BlackBerry Collaboration Service und die BlackBerry MDS Services, um auf URLs im Internet und Intranet zugreifen zu können. Verwenden Sie die Proxy-Methode, die dem Zugriff anderer Anwendungen und Server innerhalb des Unternehmens auf Webinhalte entspricht.

Da Unternehmens-Proxy-Server keinen Datenverkehr zwischen Servern auf derselben Seite der Firewall zulassen, können Sie BlackBerry-Komponenten so konfigurieren, dass sie eine Proxy-Autokonfigurationsdatei (.pac) verwenden oder direkt über einen Proxy-Server auf das Internet zugreifen.

Sie können auch mehrere Proxy-Server konfigurieren, die den Datenverkehr zu bestimmten URLs verwalten, und Sie können URLs konfigurieren, auf die BlackBerry-Komponenten ohne den Umweg über einen Proxy-Server zugreifen können.

Tabelle 9.6: Zugriff auf eine Webseite mit einer PAC-Datei

Aktion	Vorgehensweise
Konfigurieren der PAC-Dateieinstellungen für den BlackBerry MDS Connection Service	1. Klicken Sie auf einen BlackBerry MDS Connection Service. 2. Klicken Sie auf der Registerkarte <i>Connection Services</i> (Verbindungsdienste) auf <i>Edit Properties</i> (Eigenschaften bearbeiten).
Konfigurieren der PAC-Dateieinstellungen für den BlackBerry Collaboration Service	1. Klicken Sie auf einen BlackBerry Collaboration Service. 2. Klicken Sie auf der Registerkarte <i>Collaboration Service</i> (Zusammenarbeitsdienst) auf <i>Edit Properties</i> (Eigenschaften bearbeiten).

Aktion	Vorgehensweise
Konfigurieren der PAC-Dateieinstellungen für die BlackBerry MDS Services	1. Klicken Sie auf einen BlackBerry MDS Services Server. 2. Klicken Sie auf der Registerkarte <i>MDS Services</i> (MDS-Dienste) auf <i>Edit Properties</i> (Eigenschaften bearbeiten).

1. Klicken Sie auf *Proxy* (Proxy).
2. Doppelklicken Sie auf *Proxy Mappings* (Proxy-Zuordnungen).
3. Klicken Sie auf *New* (Neu).
4. Geben Sie im Feld *Universal Resource Locator* (URL) den URL-Ausdruck ein, der von der Proxy-Zuordnungsregel zur Steuerung der Zuordnung verwendet werden soll.
5. Geben Sie im Feld *Description* (Beschreibung) eine Beschreibung für die Proxy-Zuordnungsregel ein.
6. Doppelklicken Sie auf *Proxy-String* (Proxy-Zeichenfolge).
7. Klicken Sie auf *New* (Neu).

Führen Sie in der Dropdown-Liste *Proxy Type* (Proxy-Typ) eine der folgenden Aktionen aus:

Tabelle 9.7: Automatisches Erkennen von PAC-Dateien

Aktion	Vorgehensweise
Automatisches Erkennen von PAC-Dateien	1. Klicken Sie auf <i>AUTO</i> (AUTO). 2. Doppelklicken Sie auf das Feld <i>Proxy String</i> (Proxy-Zeichenfolge) und löschen Sie den Standardwert.
Geben Sie den Speicherort der PAC-Datei an.	1. Klicken Sie auf <i>PAC</i> (PAC). 2. Doppelklicken Sie auf das Feld <i>Proxy String</i> (Proxy-Zeichenfolge), und geben Sie den Namen des Proxy-Servers, die Port-Nummer und den Speicherort der PAC-Datei ein. Beispiel: <i>http://<Proxyserver>:<Port>/<PAC-Dateipfad>/<PAC-Dateiname></i>.

8. Klicken Sie auf *OK*.

9.6.5 Zugreifen auf Webserver über einen Proxy-Server

Wenn BlackBerry-Komponenten über einen Proxy-Server auf Webserver zugreifen, können Sie in den Proxy-Zuordnungsregeln für URLs mehr als eine Proxy-Zeichenfolge festlegen.

BlackBerrys verwenden definierte Proxy-Zeichenfolgen in der Reihenfolge, in der sie für den URL angegeben sind. Wenn es einer BlackBerry-Komponente mit der ersten Proxy-

Zeichenfolge für einen URL nicht gelingt, auf den Webserver zuzugreifen, verwendet sie die nächste Proxy-Zeichenfolge in der Proxy-Zuordnungsregel.

Beispiel: Sie können eine Proxy-Zuordnungsregel erstellen, um für einen bestimmten URL die Verwendung eines nicht standardmäßigen Proxy-Servers zuzulassen.

Für den Fall, dass dieser Server nicht verfügbar ist, können Sie eine sekundäre Proxy-Zeichenfolge definieren, um den URL-Zugriff auf den Webserver über den Standard-Proxy-Server des Unternehmens zu ermöglichen.

1. Führen Sie im BlackBerry Manager im linken Fensterbereich eine der folgenden Aktionen aus:

Tabelle 9.8: Zugriff auf einen Webserver über Proxy

Aktion	Vorgehensweise
Konfigurieren der Proxy-Einstellungen für den BlackBerry MDS Connection Service	1. Klicken Sie auf einen BlackBerry MDS Connection Service. 2. Klicken Sie auf der Registerkarte <i>Connection Service</i> (Verbindungsdienst) auf <i>Edit Properties</i> (Eigenschaften bearbeiten).
Konfigurieren der Proxy-Einstellungen für den BlackBerry Collaboration Service	1. Klicken Sie auf einen BlackBerry Collaboration Service. 2. Klicken Sie auf der Registerkarte <i>Collaboration Service</i> (Zusammenarbeitsdienst) auf <i>Edit Properties</i> (Eigenschaften bearbeiten).
Konfigurieren der Proxy-Einstellungen für BlackBerry MDS Services	1. Klicken Sie auf einen BlackBerry MDS Services Server. 2. Klicken Sie auf der Registerkarte <i>MDS Services</i> (MDS-Dienste) auf <i>Edit Properties</i> (Eigenschaften bearbeiten).

2. Klicken Sie auf *Proxy* (Proxy).
3. Doppelklicken Sie auf *Proxy Mappings* (Proxy-Zuordnungen).
4. Klicken Sie auf *New* (Neu).
5. Geben Sie im Feld *Universal Resource Locator* (URL) den URL-Ausdruck ein, der von der Proxy-Zuordnungsregel zur Steuerung der Zuordnung verwendet werden soll.
6. Geben Sie im Feld *Description* (Beschreibung) eine Beschreibung für die Proxy-Zuordnungsregel ein.
7. Doppelklicken Sie auf *Proxy-String* (Proxy-Zeichenfolge).
8. Klicken Sie auf *New* (Neu).
9. Führen Sie in der Dropdown-Liste *Proxy Type* (Proxy-Typ) eine der folgenden Aktionen aus:

Tabelle 9.9: Proxy Type

Aktion	Vorgehensweise
Konfigurieren eines Proxy-Servers	1. Klicken Sie auf <i>Proxy</i> (Proxy). 2. Doppelklicken Sie auf das Feld <i>Proxy String</i> (Proxy-Zeichenfolge), und geben Sie den Namen des Proxy-Servers und die Port-Nummer ein.
Ausschließen des URL vom Routing über den Proxy-Server	1. Klicken Sie auf <i>DIRECT</i> (Direkt). 2. Doppelklicken Sie auf das Feld <i>Proxy String</i> (Proxy-Zeichenfolge), und löschen Sie den Standardwert.

10. Klicken Sie auf *OK*.

9.6.6 Konfigurieren der Authentifizierung von BlackBerry-Komponenten bei einem Proxy-Server im Namen von BlackBerry-Geräten

1. Führen Sie im BlackBerry Manager im linken Fensterbereich eine der folgenden Aktionen aus:

Tabelle 9.10: Authentifizierung von BlackBerry-Komponenten

Aktion	Vorgehensweise
Konfigurieren der Einstellungen für die Authentifizierung zwischen dem BlackBerry MDS Connection Service und einem Proxy-Server	1. Klicken Sie auf einen BlackBerry MDS Connection Service. 2. Klicken Sie auf der Registerkarte <i>Connection Service</i> (Verbindungsdienst) auf <i>Edit Properties</i> (Eigenschaften bearbeiten).
Konfigurieren der Einstellungen für die Authentifizierung zwischen dem BlackBerry MDS Connection Service und einem Proxy-Server	1. Klicken Sie auf einen BlackBerry MDS Connection Service. 2. Klicken Sie auf der Registerkarte <i>Connection Service</i> (Verbindungsdienst) auf <i>Edit Properties</i> (Eigenschaften bearbeiten).
Konfigurieren der Einstellungen für die Authentifizierung zwischen dem BlackBerry Collaboration Service und einem Proxy-Server	1. Klicken Sie auf einen BlackBerry Collaboration Service. 2. Klicken Sie auf der Registerkarte <i>Collaboration Service</i> (Zusammenarbeitsdienst) auf <i>Edit Properties</i> (Eigenschaften bearbeiten).
Konfigurieren der Einstellungen für die Authentifizierung zwischen den BlackBerry MDS Services und einem Proxy-Server	1. Klicken Sie auf einen BlackBerry MDS Services-Server. 2. Klicken Sie auf der Registerkarte <i>MDS Services</i> (MDS-Dienste) auf <i>Edit Properties</i> (Eigenschaften bearbeiten).

2. Klicken Sie auf *Proxy* (Proxy).
3. Doppelklicken Sie auf *Proxy Mappings* (Proxy-Zuordnungen).
4. Klicken Sie auf einen URL.
5. Klicken Sie auf *Properties* (Eigenschaften).
6. Geben Sie im Feld *User Name* (Benutzername) den Benutzernamen ein, den die BlackBerry-Komponente für die Verbindung zu dem Proxy-Server verwendet, der für den URL definiert wurde.
7. Geben Sie im Feld *Password* (Kennwort) das Kennwort für den Benutzernamen ein.
8. Wiederholen Sie das Kennwort im Feld *Password Confirmation* (Kennwortbestätigung).
9. Klicken Sie auf OK.

9.6.7 Verknüpfen einer BlackBerry-Komponente mit mehreren BlackBerry Enterprise Servern

Weisen Sie einen BlackBerry MDS Connection Service, Collaboration Service und MDS Services Server mehreren Enterprise Servern in der BlackBerry-Domain zu.

Wenn Ihre Domain einen Enterprise Server enthält, werden der MDS Connection Service, Collaboration Service und MDS Services Server automatisch mit dieser Enterprise-Server-Instanz verknüpft.

9.6.8 Zuweisen eines BlackBerry MDS Connection Service zu mehreren BlackBerry Enterprise Servern

Legen Sie den zentralen Push-Server als primären MDS Connection Service fest, mit dem mehrere Enterprise Server Anwendungsdaten an BlackBerry-Geräte senden, und lassen Sie die HTTP-Suche auf BlackBerry-Geräten zu.

Weitere Informationen finden Sie unter »Festlegen des zentralen Push-Servers« in Kapitel 19.2:

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Klicken Sie auf der Registerkarte *Global* auf *Service Control & Customization* (Dienststeuerung und -anpassung).
3. Klicken Sie auf *MDS CS to BES Mapping* (MDS CS-zu-BES-Zuordnung).
4. Klicken Sie im Dialogfeld *BES Mappings* (BES-Zuordnungen) im linken Fensterbereich auf den *BlackBerry MDS Connection Service*.

5. Wählen Sie im rechten Fensterbereich einen oder mehrere BlackBerry Enterprise Server aus.
6. Klicken Sie auf OK.

9.6.9 Einen Collaboration Service mehreren BlackBerry Enterprise Servern zuweisen

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Klicken Sie auf der Registerkarte *Global* auf *Service Control & Customization* (Dienststeuerung und -anpassung).
3. Klicken Sie auf *IM to BES Mapping* (IM-zu-BES-Zuordnung).
4. Klicken Sie im Dialogfeld *IM to BES Mappings* (IM-zu-BES-Zuordnungen) im linken Fensterbereich auf den BlackBerry Collaboration Service.
5. Wählen Sie im rechten Fensterbereich einen oder mehrere Enterprise Server aus.
6. Klicken Sie auf OK.

9.6.10 BlackBerry-MDS-Services mehreren Enterprise Servern zuweisen

Geben Sie die in der MDS-Services-Datenbank veröffentlichten MDS Studio Applications (BlackBerry-MDS-Studio-Anwendungen) für Benutzer auf mehreren Enterprise Servern frei.

MDS Services senden Anwendungen und Daten über den zentralen Push-Server an BlackBerry-Geräte.

Weitere Informationen finden Sie unter »Festlegen des zentralen Push-Servers« in Kapitel 19.2.

Wenn Sie einem Enterprise Server einen MDS Services Server zuweisen und der MDS Connection Service einen Proxy-Server verwendet, weist der BlackBerry Manager eine direkte Verbindung zwischen dem MDS Connection Service und den MDS Services zu.

Die MDS Services können nicht über einen Proxy-Server mit dem MDS Connection Service kommunizieren.

Hinweis: Der BlackBerry Manager fordert Sie beim ersten Zugriff auf den BlackBerry MDS Services Server auf, ein Secure-SocketsLayer (SSL)-Zertifikat zu installieren. Weitere Informationen finden Sie im Kapitel 12.9.2 »Erstellen einer Server-authentifizierung zwischen BlackBerry MDS Services und dem BlackBerry Manager«.

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen BlackBerry Enterprise Server.
2. Klicken Sie auf der Registerkarte *Server Configuration* (Serverkonfiguration) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *MDS Services* (MDS-Dienste).
4. Klicken Sie auf die URL des MDS Services Servers.
5. Klicken Sie in der Dropdown-Liste auf den MDS Services Server.
6. Klicken Sie auf OK.

9.6.11 Konfigurieren der Unterstützung der Adressensuche in einer gehosteten Umgebung

Wenn Sie einen BlackBerry Enterprise Server hosten und mehrere Unternehmen Ihren gehosteten Dienst abonnieren, müssen Sie die Adressensuchfunktion anpassen.

Mit der Adressensuchfunktion können Benutzer auf die Kontaktliste eines Unternehmens, wie beispielsweise die globale Adressenliste (GAL), zugreifen und die Informationen für einen oder mehrere Benutzer auf deren BlackBerry-Geräte herunterladen.

Eine Anpassung der Adressensuchfunktion schränkt den Zugriff von Benutzern auf Kontaktinformationen anderer Unternehmen ein.

Stellen Sie sicher, dass der Name des Unternehmens, dem ein Benutzer angehört, durchgehend richtig und einheitlich in der GAL angegeben ist.

Beispiel: Wenn der Name des Unternehmens teilweise als Akronym und teilweise ausgeschrieben in den Einträgen angegeben ist, kann die Adressensuche zu falschen Ergebnissen führen.

Wenn ein Benutzer nach Kontaktinformationen zu einem anderen Benutzer sucht, dessen Unternehmen nicht angegeben ist, gibt die Adressensuche keine Ergebnisse zurück.

Weisen Sie die Benutzer an, beim Erstellen von Nachrichten die vollständige E-Mail-Adresse einzugeben, um zu vermeiden, dass die Adressensuche fehlschlägt oder falsche Ergebnisse erzielt.

Wenn Sie die Standardfunktion für die Adressensuche nicht verwenden möchten, können Sie den BlackBerry Enterprise Server auch so einrichten, dass er mithilfe des Lightweight Directory Access Protocol (LDAP) nach Benutzeradressen sucht.

Das LDAP bietet Benutzern die Möglichkeit, umfangreichere Adressensuchen in der GAL durchzuführen. Es durchsucht die GAL nicht nach dem Namen des Unternehmens, dem ein Benutzer angehört.

Stattdessen wird ein LDAP-Feld als Filter zum Abrufen von Benutzern mit einem bestimmten Unternehmensnamen verwendet. Sie können dabei ein vorhandenes LDAP-Feld verwenden oder ein benutzerdefiniertes Feld erstellen.

Anpassen der Adressensuchfunktion

1. Geben Sie an der Eingabeaufforderung `regedit` ein.
2. Wechseln Sie im linken Fensterbereich zu *HKEY_LOCAL_MACHINE\Software\Research In Motion\BlackBerry Enterprise Server\Agents*.
3. Stellen Sie sicher, dass der DWORD-Wert *AllowAddressLookup* (Adressensuche zulassen) auf 1 gesetzt ist.
4. Erstellen Sie einen neuen DWORD-Wert namens *HostedServer* (gehosteter Server).
5. Legen Sie als Wert 1 fest.
6. Starten Sie auf dem Computer, auf dem der Enterprise Server installiert ist, im Fenster *Services* (Dienste) den BlackBerry Controller-Dienst erneut.

Suchen von Adressen mit LDAP

1. Geben Sie an der Eingabeaufforderung `regedit` ein.
2. Wechseln Sie im linken Fensterbereich zu *HKEY_LOCAL_MACHINE\Software\Research In Motion\BlackBerry Enterprise Server\Agents*.
3. Erstellen Sie einen neuen DWORD-Wert namens *LDAPSearch* (LDAP-Suche).
4. Legen Sie als Wert 1 fest.
5. Erstellen Sie einen neuen DWORD-Wert namens *LDAPALPSearch* (LDAP-ALP-Suche).
6. Legen Sie als Wert 1 fest.
7. Starten Sie auf dem Computer, auf dem der BlackBerry Enterprise Server installiert ist, im Fenster *Services* (Dienste) den BlackBerry-Controller-Dienst erneut.

Erstellen eines benutzerdefinierten Felds für die LDAP-Adressensuche

1. Geben Sie an der Eingabeaufforderung `regedit` ein.
2. Wechseln Sie im linken Fensterbereich zu *HKEY_LOCAL_MACHINE\Software\Research In Motion\BlackBerry Enterprise Server\Agents*.
3. Erstellen Sie einen Zeichenfolgewart namens *LDAPCompanyField* (LDAP-Unternehmensfeld).
4. Legen Sie als Wert eine Zeichenfolge fest, die das LDAP-Feld repräsentiert, das Sie als das Feld für den Unternehmensnamen verwenden möchten.
5. Starten Sie auf dem Computer, auf dem der Enterprise Server installiert ist, im Fenster *Services* (Dienste) den BlackBerry-Controller-Dienst erneut.

10 Benutzerkonten auf dem Enterprise Server einrichten

In diesem Kapitel sehen Sie, wie Sie Benutzerkonten anlegen und Benutzergruppen verwalten. Darüber hinaus betrachten wir, wie die Synchronisierung der Terminplannerdaten konfiguriert wird.

10.1 Benutzerkonten

Wenn Sie dem Enterprise Server ein Benutzerkonto hinzufügen, braucht sich der Microsoft-Exchange-Posteingang des Benutzers nicht am selben Exchange-Standort oder in derselben Microsoft-Exchange-Routing-Gruppe wie der BlackBerry Enterprise Server zu befinden.

Fügen Sie ein Benutzerkonto immer nur jeweils einem Enterprise Server hinzu.

Wenn Sie ein Benutzerkonto hinzufügen, das sich zuvor auf einem anderen Enterprise Server in einer anderen BlackBerry-Domain befand, oder wenn der Benutzer zuvor den BlackBerry Desktop Redirector verwendete, müssen Sie das BlackBerry-Device (BlackBerry-Gerät) einrichten.

Weitere Informationen dazu finden Sie in Kapitel 12.

10.1.1 Hinzufügen von Benutzerkonten

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen BlackBerry Enterprise Server.
2. Klicken Sie auf der Registerkarte *Server Configuration* (Serverkonfiguration) auf *Common* (Gemeinsam).
3. Klicken Sie auf *Add Users* (Benutzer hinzufügen).
4. Klicken Sie in der Dropdown-Liste *Show Names from the* (Namen anzeigen aus) auf eine Adressengruppe.
5. Klicken Sie in der Benutzerliste auf einen Benutzer.
6. Klicken Sie auf *Select* (Auswählen).
7. Klicken Sie zum Abschluss auf *OK*.

Verwalten von Benutzergruppen

Erstellen Sie Gruppen von Benutzerkonten in der BlackBerry-Domain, um gemeinsame Konfigurationseigenschaften auf die Gruppe anzuwenden oder um Verwaltungsaufgaben für alle Benutzerkonten in der Gruppe zu erledigen.

Benutzerkonten in einer Gruppe können auf verschiedenen Enterprise Servern in der BlackBerry-Domain vorhanden sein. Sobald Sie eine Gruppe erstellt haben, legen Sie die Eigenschaften fest, die Sie auf alle Benutzerkonten in der Gruppe anwenden möchten.

Wenn Sie einer Gruppe Benutzerkonten hinzufügen, werden die Gruppeneigenschaften den Benutzerkonten automatisch zugewiesen.

Sie können Eigenschaften einer vorhandenen Gruppe in eine neue Gruppe kopieren.

10.1.2 Erstellen einer Gruppe

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *User Groups* (Benutzergruppen).
2. Klicken Sie auf die Schaltfläche *Create Group* (Gruppe erstellen).
3. Geben Sie im Feld *Group Name* (Gruppenname) einen Namen ein.
4. Geben Sie im Feld *Description* (Beschreibung) eine Beschreibung ein.
5. Klicken Sie auf OK.
6. Führen Sie eine der folgenden Aktionen aus:

Tabelle 10.1: Erstellen einer Gruppe

Aktion	Vorgehensweise
Hinzufügen von Eigenschaften zu einer Gruppe	1. Klicken Sie auf <i>Edit Group Template</i> (Gruppenvorlage bearbeiten). 2. Legen Sie die gewünschten Eigenschaften fest. Weitere Informationen finden Sie im Kapitel 13 »Anpassen der BlackBerry-Nachrichtenübermittlung«
Kopieren der Eigenschaften einer vorhandenen Gruppe	1. Klicken Sie in der Liste <i>Group Name</i> (Gruppenname) auf die Gruppe, deren Eigenschaften Sie kopieren möchten. 2. Klicken Sie auf <i>Copy Properties to Another Group</i> (Eigenschaften auf eine andere Gruppe übertragen). 3. Klicken Sie auf die Gruppe, in die Sie die Eigenschaften kopieren möchten.

7. Klicken Sie auf OK.

10.1.3 Zuweisen eines Benutzers zu einer Gruppe

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Users* (Benutzer) auf ein Benutzerkonto.
3. Klicken Sie auf *Account* (Konto).
4. Klicken Sie auf *Assign User to Group* (Benutzer einer Gruppe zuweisen).
5. Klicken Sie auf einen Gruppennamen.
6. Klicken Sie auf OK.

10.2 Anpassen der Synchronisierung von Terminplanerdaten

Sie können Terminplanerdaten-Elemente (PIM-Elemente) wie Aufgaben, Notizen und Kontakte so synchronisieren, dass die Einträge auf dem Gerät eines Benutzers mit den Einträgen der E-Mail-Anwendung des Desktops übereinstimmen.

Sie können die Synchronisierungsoptionen global für alle Benutzerkonten in der BlackBerry-Domain oder für ein bestimmtes Benutzerkonto festlegen. Standardmäßig ist die drahtlose Synchronisierung aller Terminplanerdaten-Anwendungen für ein Benutzerkonto aktiviert.

10.2.1 Konfigurieren der Terminplanerdaten-Synchronisierung für alle Benutzerkonten

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Klicken Sie auf der Registerkarte *Global* auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Global PIM Sync* (Globale PIM-Synchronisierung).
4. Führen Sie eine der folgenden Aktionen aus:

Tabelle 10.2: Terminplanerdaten-Synchronisierung

Aktion	Vorgehensweise
Deaktivieren der Synchronisierung des Nachrichtenfilters	1. Klicken Sie im Bereich <i>Message Filters</i> (Nachrichtenfilter) auf <i>Synchronization enabled</i> (Synchronisierung aktiviert). 2. Klicken Sie in der Dropdown-Liste auf <i>False</i> (Falsch).
Deaktivieren der Aufgabensynchronisierung	1. Klicken Sie im Bereich <i>Tasks</i> (Aufgaben) auf <i>Synchronization enabled</i> (Synchronisierung aktiviert). 2. Klicken Sie in der Dropdown-Liste auf <i>False</i> (Falsch).

Aktion	Vorgehensweise
Deaktivieren der Synchronisierung der Nachrichteneinstellungen	1. Klicken Sie im Bereich <i>Message Settings</i> (Nachrichteneinstellungen) auf <i>Synchronization enabled</i> (Synchronisierung aktiviert). 2. Klicken Sie in der Dropdown-Liste auf <i>False</i> (Falsch).
Deaktivieren der Notizensynchronisierung	1. Klicken Sie im Bereich <i>Memos</i> (Notizen) auf <i>Synchronization enabled</i> (Synchronisierung aktiviert). 2. Klicken Sie in der Dropdown-Liste auf <i>False</i> (Falsch).
Deaktivieren der Adressbuchsynchronisierung	1. Klicken Sie im Bereich <i>Address Book</i> (Adressbuch) auf <i>Synchronization enabled</i> (Synchronisierung aktiviert). 2. Klicken Sie in der Dropdown-Liste auf <i>False</i> (Falsch).
Deaktivieren der Terminplanerdaten-Synchronisierung für ein bestimmtes Benutzerkonto	1. Klicken Sie auf <i>Wireless Synchronization Enabled</i> (Drahtlose Synchronisierung aktiviert). 2. Klicken Sie in der Dropdown-Liste auf <i>False</i> (Falsch).

5. Klicken Sie auf OK.

10.2.2 Festlegen der Terminplanerdaten-Synchronisierungsart für alle Benutzerkonten

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Klicken Sie auf der Registerkarte *Global* auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Global PIM Sync* (Globale PIM-Synchronisierung).
4. Gehen Sie zu einer Terminplanerdaten-Anwendung in der Liste.
5. Wählen Sie eine der folgenden Synchronisierungsoptionen:

Server to Device (Server auf Gerät): Synchronisiert nur die Daten vom BlackBerry Enterprise Server zum BlackBerry-Gerät.

Device wins (Gerät hat Vorrang): Die Daten auf dem BlackBerry-Gerät überschreiben die Daten auf dem BlackBerry Enterprise Server.

6. Klicken Sie auf OK.

10.2.3 Definieren der Terminplanerdaten – Datenkonfliktlösung für ein bestimmtes Benutzerkonto

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen BlackBerry Enterprise Server.

2. Doppelklicken Sie auf der Registerkarte *User List* (Benutzerliste) auf ein Benutzerkonto.
3. Klicken Sie auf *PIM Sync* (PIM-Synchronisierung).
4. Wählen Sie für jede Terminplanerdaten-Anwendung in der Liste eine der folgenden Konfliktlösungsoptionen aus:

Server wins (Server hat Vorrang): Die Daten auf dem Enterprise Server überschreiben die Daten auf dem BlackBerry-Gerät.

Device wins (Gerät hat Vorrang): Die Daten auf dem BlackBerry-Gerät überschreiben die Daten auf dem Enterprise Server.
5. Klicken Sie auf OK.

10.3 Steuern der BlackBerry-Umgebung

Themen:

- Festlegen der BlackBerry-Geräte, die eine Verbindung zum Enterprise Server herstellen können
- Steuern des Verhaltens der Geräte und der BlackBerry-Desktopsoftware
- Steuern benutzerdefinierter Anwendungen mit IT-Richtlinienregeln
- Erstellen einer IT-Richtlinie
- Zuweisen einer IT-Richtlinie zu einem Benutzerkonto oder einer Gruppe
- Verwalten von IT-Richtlinien

10.3.1 Festlegen der Geräte, die eine Verbindung zum BlackBerry Enterprise Server herstellen können

Aktivieren Sie die Enterprise Service Policy (Unternehmensdienst-Richtlinie), um zu steuern, welche BlackBerry-Geräte eine Verbindung zum BlackBerry Enterprise Server herstellen können.

Nach dem Aktivieren der Unternehmensdienst-Richtlinie lässt der Enterprise Server weiter Verbindungen von Geräten zu, die dem Enterprise Server bereits hinzugefügt wurden.

Verbindungen von neu hinzugefügten Geräten werden jedoch standardmäßig abgelehnt.

Hinweis: Die Unternehmensdienst-Richtlinie gilt auch für BlackBerry Connect- und BlackBerry Built-In-Geräte.

Definieren Sie in einer sogenannten Genehmigungsliste Gerätekriterien, anhand derer der Enterprise-Server-Zugriff für BlackBerry-Geräte aktiviert oder deaktiviert wird.

BlackBerry-Geräte, die die Kriterien der Genehmigungsliste erfüllen, können auf dem jeweiligen Enterprise Server eine drahtlose Unternehmensaktivierung durchführen.

Sie können folgende Kriterientypen festlegen:

- bestimmte zugelassene Geräte-PINs als Zeichenfolge,
- einen zulässigen Bereich von Geräte-PINs,
- bestimmte zugelassene Hersteller und Gerätemodelle.

Der BlackBerry Manager enthält Listen der zulässigen Hersteller und Modelle auf der Basis der Eigenschaften der Geräte, die dem Enterprise Server bereits hinzugefügt wurden.

In diesen Listen können Sie Elemente deaktivieren, um neue Verbindungen von Geräten eines bestimmten Herstellers oder Modells zu unterbinden.

Sie können einem bestimmten Benutzerkonto gestatten, die Unternehmensdienst-Richtlinie zu überschreiben.

Wenn Sie die Genehmigungsliste dann mit Kriterien definieren, die das BlackBerry-Gerät dieses Benutzers ausschließen, kann das Benutzerkonto immer noch eine Verbindung zum BlackBerry Enterprise Server herstellen.

10.3.2 Aktivieren der Unternehmensdienst-Richtlinie

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Klicken Sie im rechten Fenster auf *Service Control & Customization* (Dienststeuerung und -anpassung).
3. Klicken Sie auf *Enable Enterprise Service Policy* (Unternehmensdienst-Richtlinie aktivieren).
4. Klicken Sie auf *OK*.
5. Klicken Sie auf der Registerkarte *Global* auf *Edit Properties* (Eigenschaften bearbeiten).
6. Klicken Sie auf *Enterprise Service Policy* (Unternehmensdienst-Richtlinie).
7. Legen Sie die gewünschten Eigenschaften fest.
8. Klicken Sie auf *OK*.

10.3.3 Zulassen des Überschreibens der Unternehmensdienst-Richtlinie durch ein Benutzerkonto

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Users* (Benutzer) auf ein Benutzerkonto.
3. Klicken Sie auf *Edit Properties* (Eigenschaften bearbeiten).
4. Klicken Sie auf *ES Policy Override* (UD-Richtlinie überschreiben).
5. Klicken Sie in der Dropdown-Liste auf *True* (Wahr).
6. Klicken Sie auf *OK*.

10.3.4 Steuern des Verhaltens von BlackBerry-Geräten und der BlackBerry-Desktopsoftware

Verwenden Sie eine oder mehrere IT-Richtlinien, um das Verhalten der Geräte und der BlackBerry Desktop Software (BlackBerry-Desktopsoftware) in Ihrem Unternehmen zu steuern.

Eine IT-Richtlinie besteht aus mindestens einer IT-Richtlinienregel. Die Standard-IT-Richtlinie enthält sämtliche Standard-IT-Richtlinienregeln auf dem Enterprise Server. Wenn in einer BlackBerry-Domain ein neues Benutzerkonto auf dem Enterprise Server aktiviert wird, übermittelt der Enterprise Server die Standard-IT-Richtlinie automatisch an das Gerät des Benutzers.

Die Standard-IT-Richtlinienregeln erzwingen nicht das Standardverhalten für die Geräte oder die BlackBerry-Desktopsoftware.

Zum Ändern des Standardverhaltens der Geräte und der Desktopsoftware in Ihrem Unternehmen können Sie eine der folgenden Methoden anwenden:

- Festlegen der Werte der IT-Richtlinienregeln in der Standard-IT-Richtlinie.
- Erstellen oder Importieren einer neuen IT-Richtlinie, Festlegen der Werte der IT-Richtlinienregeln und Zuweisen eines oder mehrerer Benutzerkonten oder -gruppen zur neuen IT-Richtlinie. Um das Verhalten des BlackBerry-Geräts und der BlackBerry-Desktopsoftware über das drahtlose Netzwerk zu aktualisieren, muss der Enterprise Server die IT-Richtlinie erneut an das BlackBerry-Gerät senden.

Standardmäßig ist der Enterprise Server so konfiguriert, dass die IT-Richtlinie erneut an die Geräte der Benutzer gesendet wird, die dieser IT-Richtlinie kurz nach der Aktualisierung der IT-Richtlinie zugewiesen wurden.

Sie können eine IT-Richtlinie auch manuell erneut an das Benutzerkonto eines bestimmten Geräts senden. Sie können den BlackBerry Enterprise Server auch so konfigurieren, dass er IT-Richtlinien in festgelegten Zeitabständen erneut an Geräte sendet, ungeachtet dessen, ob Sie die IT-Richtlinien geändert haben. Wenn das Gerät eine aktu-

alisierte Standard-IT-Richtlinie oder eine neue IT-Richtlinie empfängt, wenden das Gerät und die Desktopsoftware die Konfigurationsänderungen an.

Ändern des Standardverhaltens

Mit einer IT-Richtlinienregel können Sie die Funktionen der Geräte oder der Desktopsoftware auf folgende Weise anpassen und steuern:

- Einstellen einer Regel auf einen Wahr- oder Falsch-Wert.
- Eingeben einer Zeichenfolge, die eine Regel aktiviert und gleichzeitig die Parameter für ihre Nutzung bereitstellt.
- Auswählen eines vordefinierten zulässigen Werts, der einer IT-Regel zugewiesen wird.

Zu einigen IT-Richtlinienregeln gibt es auf dem BlackBerry-Gerät ein entsprechendes, für den Benutzer zugängliches Feld.

- Wenn Sie eine IT-Richtlinienregel auf einen Wahr- oder Falsch-Wert setzen, verhindern Sie, dass der Benutzer einen anderen Wert für ein entsprechendes Feld auf dem Gerät auswählt.
- Wenn Sie eine Zeichenfolge eingeben, die eine IT-Richtlinienregel aktiviert und gleichzeitig die Parameter für die Verwendung der Regel bereitstellt, kann der Benutzer den Wert eines entsprechenden Felds auf dem Gerät nicht ändern.
- Wenn Sie einer IT-Richtlinienregel einen vordefinierten, zulässigen Wert zuweisen, grenzen Sie die Werte ein, die der Benutzer für ein entsprechendes Feld auf dem Gerät festlegen kann. Ein Vorhängeschloss neben einem Feld auf dem Gerät zeigt an, dass die IT-Richtlinie die Einstellung steuert und sie nicht vom Benutzer geändert werden kann.

Sie können einer IT-Richtlinie eine Standard-IT-Richtlinienregel hinzufügen, eine Standard-IT-Richtlinienregel aus ihr entfernen oder den zugewiesenen Wert einer Standard-IT-Richtlinienregel in einer IT-Richtlinie ändern. Sie können einer Standard-IT-Richtlinienregel weder zulässige Werte hinzufügen noch können Sie diese entfernen oder ändern. Auch das Löschen der Standard-IT-Richtlinienregeln ist nicht möglich.

Das Hinzufügen einer neuen IT-Richtlinienregel zu einer IT-Richtlinie, das Entfernen einer neuen IT-Richtlinienregel aus einer IT-Richtlinie und das Ändern des zugewiesenen Wertes einer neuen IT-Richtlinienregel in einer IT-Richtlinie gehen genauso vonstatten wie das Ändern einer Standardregel in einer IT-Richtlinienregel in einer IT-Richtlinie.

Zurücksetzen auf das Standardverhalten

Um das durch eine IT-Richtlinienregel angepasste oder gesteuerte Standardverhalten der Funktion wiederherzustellen, können Sie diese IT-Richtlinienregel auf *Default* (Standard) setzen (sofern diese Einstellung verfügbar ist) oder den zuvor für die IT-Richtlinienregel eingestellten Wert löschen.

Wenn Sie einer neuen IT-Richtlinie Benutzerkonten zugewiesen haben, können Sie die neue IT-Richtlinie löschen, um diese Benutzerkonten für sämtliche Funktionen auf den Geräten und in der BlackBerry-Desktopsoftware auf das Standardverhalten zurückzusetzen.

Der Enterprise Server weist diese Benutzerkonten automatisch erneut der Standard-IT-Richtlinie zu und sendet die Standard-IT-Richtlinie erneut an die Geräte, wodurch die Standardeinstellungen wirksam werden. Sie können die Standard-IT-Richtlinie nicht löschen.

10.3.5 Steuern benutzerdefinierter Anwendungen mit IT-Richtlinienregeln

Erstellen Sie neue IT-Richtlinienregeln, um benutzerdefinierte Anwendungen zu steuern, die von Ihrem Unternehmen für BlackBerry-Umgebungen entwickelt werden.

Nach dem Erstellen einer neuen IT-Richtlinienregel können Sie sie hinzufügen und ihr in jeder beliebigen neuen oder bereits vorhandenen IT-Richtlinie einen Wert zuweisen. Von Ihnen erstellte neue IT-Richtlinienregeln können ausschließlich von Ihren eigenen benutzerdefinierten Anwendungen genutzt werden.

Sie können keine neuen IT-Richtlinienregeln zum Steuern der Standardfunktionen der BlackBerry-Geräte erstellen.

10.3.6 Erstellen einer IT-Richtlinie

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Klicken Sie auf der Registerkarte *Global* auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *IT Policy* (IT-Richtlinie).
4. Doppelklicken Sie im Bereich *IT Policy Administration* (IT-Richtlinienverwaltung) auf *IT Policies* (IT-Richtlinien).
5. Klicken Sie auf *New* (Neu).
6. Doppelklicken Sie auf *IT Policy Name* (IT-Richtliniennamen).
7. Geben Sie für die neue IT-Richtlinie einen Namen ein.
8. Konfigurieren Sie die IT-Richtlinienregeln, indem Sie folgende Aktionen ausführen:
 - Klicken Sie im linken Fensterbereich auf eine Richtliniengruppe.
 - Doppelklicken Sie im rechten Fensterbereich auf eine IT-Richtlinienregel.
 - Legen Sie einen Wert für die IT-Richtlinienregel fest.
9. Klicken Sie auf *OK*.

10.3.7 Erstellen einer IT-Richtlinie auf der Basis einer bestehenden Richtlinie

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Klicken Sie auf der Registerkarte *Global* auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *IT Policy* (IT-Richtlinie).
4. Doppelklicken Sie im Bereich *IT Policy Administration* (IT-Richtlinienverwaltung) auf *IT Policies* (IT-Richtlinien).
5. Klicken Sie auf eine IT-Richtlinie.
6. Klicken Sie auf *New Copy* (Neue Ausgabe).
7. Geben Sie für die neue IT-Richtlinie einen Namen ein.
8. Konfigurieren Sie die IT-Richtlinienregeln, indem Sie folgende Aktionen ausführen:
 - Klicken Sie im linken Fensterbereich auf eine Richtliniengruppe.
 - Doppelklicken Sie im rechten Fensterbereich auf eine IT-Richtlinienregel.
 - Legen Sie einen Wert für die IT-Richtlinienregel fest.
9. Klicken Sie auf *OK*.

10.3.8 Zuweisen einer IT-Richtlinie zu einem Benutzerkonto oder einer Gruppe

Tabelle 10.3: Zuweisen einer IT-Richtlinie zu einem Benutzerkonto

Aktion	Vorgehensweise
Zuweisen einer IT-Richtlinie zu einem Benutzerkonto	1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf <i>BlackBerry Domain</i> (BlackBerry-Domäne). 2. Klicken Sie auf der Registerkarte <i>Global</i> auf <i>Edit Properties</i> (Eigenschaften bearbeiten). 3. Klicken Sie auf <i>IT Policy</i> (IT-Richtlinie). 4. Doppelklicken Sie im Bereich <i>IT Policy Administration</i> (IT-Richtlinienverwaltung) auf <i>IT Policy to User Mapping</i> (IT-Richtlinie mit Benutzer verknüpfen). 5. Klicken Sie im linken Fensterbereich auf ein Benutzerkonto. 6. Wählen Sie im rechten Fensterbereich die gewünschte IT-Richtlinie aus. 7. Klicken Sie auf <i>OK</i>.

Aktion	Vorgehensweise
Zuweisen einer IT-Richtlinie zu einer Gruppe	1. Klicken Sie im linken Fensterbereich des BlackBerry Managers auf <i>User Groups List</i> (Benutzergruppenliste). 2. Klicken Sie in der Liste <i>Group Name</i> (Gruppenname) auf eine Gruppe. 3. Klicken Sie auf <i>Edit Group Template</i> (Gruppenvorlage bearbeiten). 4. Klicken Sie auf <i>IT Policy</i> (IT-Richtlinie). 5. Wählen Sie im rechten Fensterbereich die Option <i>IT Policy Name</i> (IT-Richtliniennamen) aus, um etwaige Benutzerausnahmen für die IT-Richtlinienregeln zu überschreiben. 6. Klicken Sie in der Dropdown-Liste auf eine IT-Richtlinie. 7. Klicken Sie auf <i>Reapply Template</i> (Vorlage erneut anwenden). 8. Klicken Sie auf <i>Yes</i> (Ja). 9. Klicken Sie auf <i>OK</i>.

10.3.9 Erstellen einer IT-Richtlinienregel für eine benutzerdefinierte Anwendung

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Klicken Sie auf der Registerkarte *Global* auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *IT Policy* (IT-Richtlinie).
4. Doppelklicken Sie im Bereich *IT Policy Administration* (IT-Richtlinienverwaltung) auf *IT Policies* (IT-Richtlinien).
5. Klicken Sie auf die gewünschte IT-Richtlinie.
6. Klicken Sie auf *Properties* (Eigenschaften).
7. Klicken Sie in der Liste *Properties* (Eigenschaften) auf *User Defined Items* (Benutzerdefinierte Elemente).
8. Doppelklicken Sie auf *IT Policy Template* (IT-Richtlinienvorlage).
9. Klicken Sie auf *New* (Neu).
10. Führen Sie folgende Aktionen aus:

Tabelle 10.4: Erstellen einer IT-Richtlinienregel für eine benutzerdefinierte Anwendung

Aktion	Vorgehensweise
Festlegen des Namens der IT-Richtlinienregel	Geben Sie einen Namen für die IT-Richtlinienregel ein.
Erklären der Verwendung der IT-Richtlinienregel	Geben Sie für die benutzerdefinierte Regel eine Beschreibung ein.
Festlegen des von der IT-Richtlinienregel verwendeten Wertetyps	Klicken Sie in der Dropdown-Liste auf <i>Boolean</i> (Boolescher Wert), <i>Integer</i> (Ganzzahl), <i>String</i> (Zeichenfolge), <i>Bitmask</i> (Bit-Maske) oder <i>Multiline String</i> (Mehrzeilige Zeichenfolge).
Festlegen, wo die IT-Richtlinienregel durchgesetzt wird	Klicken Sie in der Dropdown-Liste auf <i>Handheld</i> (Endgerät), <i>Desktop</i> (Desktop) oder <i>Both</i> (Beide).
Festlegen des Mindestwertes für die Ganzzahl	Geben Sie den für die IT-Richtlinienregel einer Ganzzahl zulässigen Mindestwert ein.
Festlegen des Höchstwertes für die Ganzzahl	Geben Sie den für die Ganzzahl einer IT-Richtlinienregel zulässigen Höchstwert ein.
Festlegen von Bit-Maskendaten	Geben Sie die für die IT-Richtlinienregel einer Bit-Maske zulässigen Daten ein. Sie können bis zu acht in Verbindung stehende boolesche Werte verwenden. Sie können einem, mehreren oder allen 8-Bit-Werten einen Bit-Optionsnamen zuweisen. Beispiel: Sie können eine IT-Richtlinienregel für eine Bit-Maske mit dem Namen »Zulässige Merkmale« und drei booleschen Bit-Werten erstellen, wobei Bit 0 als »Telefon«, Bit 1 als »Browser« und Bit 2 als »Drittanwendungen« bezeichnet werden kann.

11. Klicken Sie auf *OK*.
12. Geben Sie im Bereich *Policy Item Settings* (Richtlinienelement-Einstellungen) einen Wert für die IT-Richtlinienregel in dieser IT-Richtlinie ein.
13. Klicken Sie auf *OK*.

10.3.10 Ändern einer IT-Richtlinienregel in einer IT-Richtlinie

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Klicken Sie auf der Registerkarte *Global* auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *IT Policy* (IT-Richtlinie).
4. Doppelklicken Sie im Bereich *IT Policy Administration* (IT-Richtlinienverwaltung) auf *IT Policies* (IT-Richtlinien).

5. Klicken Sie in der Richtlinienliste auf eine IT-Richtlinie.
6. Klicken Sie auf *Properties* (Eigenschaften).
7. Konfigurieren Sie die IT-Richtlinienregeln, indem Sie folgende Aktionen ausführen:
 - Klicken Sie im linken Fensterbereich auf eine Richtliniengruppe.
 - Klicken Sie im rechten Fensterbereich auf eine IT-Richtlinienregel.
 - Legen Sie einen Wert für die IT-Richtlinienregel fest.
8. Klicken Sie auf OK.

10.3.11 Ändern oder Löschen von IT-Richtlinienregeln für benutzerdefinierte Anwendungen

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Klicken Sie auf der Registerkarte *Global* auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *IT Policy* (IT-Richtlinie).
4. Doppelklicken Sie im Bereich *IT Policy Administration* (IT-Richtlinienverwaltung) auf *IT Policies* (IT-Richtlinien).
5. Klicken Sie auf *Default* (Standard).
6. Klicken Sie auf *Properties* (Eigenschaften).
7. Klicken Sie in der Liste *Properties* (Eigenschaften) auf *User Defined Items* (Benutzerdefinierte Elemente).
8. Doppelklicken Sie auf *IT Policy Template* (IT-Richtlinienvorlage).
9. Klicken Sie auf eine IT-Richtlinienregel.
10. Führen Sie eine der folgenden Aktionen aus:

Tabelle 10.5: Ändern oder Löschen von IT-Richtlinienregeln

Aktion	Vorgehensweise
Bearbeiten einer benutzerdefinierten IT-Richtlinienregel	1. Klicken Sie auf <i>Properties</i> (Eigenschaften). 2. Ändern Sie die gewünschten Werte.
Löschen einer benutzerdefinierten IT-Richtlinienregel	Klicken Sie auf <i>Remove</i> (Entfernen).

11. Klicken Sie auf OK.

10.3.12 Löschen einer IT-Richtlinie

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Klicken Sie auf der Registerkarte *Global* auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *IT Policy* (IT-Richtlinie).
4. Doppelklicken Sie im Bereich *IT Policy Administration* (IT-Richtlinienverwaltung) auf *IT Policies* (IT-Richtlinien).
5. Klicken Sie zum Löschen auf die benutzerdefinierte IT-Richtlinie.
6. Klicken Sie auf *Remove* (Entfernen).
7. Klicken Sie auf *OK*.

10.3.13 Importieren einer IT-Richtlinie

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Klicken Sie auf der Registerkarte *Global* auf *Service Control & Customization* (Dienststeuerung und -anpassung).
3. Klicken Sie auf *Import IT Policy Definitions* (IT-Richtliniendefinitionen importieren).
4. Klicken Sie auf eine *.xml*-Datei, die Definitionen für IT-Richtlinienregeln enthält.
5. Klicken Sie auf *Open*.
6. Klicken Sie auf *OK*.

10.3.14 Manuelles erneutes Senden einer IT-Richtlinie an ein BlackBerry-Gerät

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Users* (Benutzer) auf ein Benutzerkonto.
3. Klicken Sie auf *IT Admin* (IT-Verwaltung).
4. Klicken Sie auf *Resend IT Policy* (IT-Richtlinie erneut senden).

10.3.15 Automatisches erneutes Senden einer IT-Richtlinie an ein BlackBerry-Gerät

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Server Configuration* (Serverkonfiguration) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Doppelklicken Sie im Bereich *IT Admin* (IT-Verwaltung) auf *Policy Resend Interval* (Intervall für das erneute Senden von Richtlinien).
4. Geben Sie das Intervall (in Stunden) ein, in dem das automatische erneute Senden erfolgen soll.
5. Klicken Sie auf *OK*.

11 Zusätzliche Gerätesoftware und Anwendungen für Benutzer bereitstellen

Diese Kapitel beschäftigt sich mit folgenden Themen:

- Softwarekonfigurationen
- Hinzufügen von Software zu einem Netzwerklaufwerk
- Bereitstellen von Anwendungen für Benutzer
- Erstellen von Softwarekonfigurationen
- Senden von Anwendungen über das drahtlose Netzwerk an BlackBerry-Geräte

Softwarekonfigurationen

Eine Softwarekonfiguration definiert die Anwendungen, die auf bestimmten BlackBerry-Geräten installiert sein sollen, und ermöglicht die Steuerung dieser Anwendungen.

Durch Softwarekonfigurationen können Sie die auf den Geräten in Ihrem Unternehmen installierten nicht standardmäßigen Anwendungen einheitlicher gestalten. Außerdem ist weniger Interaktion mit dem BlackBerry Manager erforderlich, wenn Sie Anwendungen auf BlackBerry-Geräten installieren.

Definieren Sie Softwarekonfigurationen für die folgenden Aufgaben:

- Laden zusätzlicher Gerätesoftware und Anwendungen auf die Geräte mit dem BlackBerry Manager,
- Zuweisen von Richtlinien für die Anwendungssteuerung zu Benutzerkonten, um die auf den Geräten installierten Anwendungen von Drittanbietern zu steuern,
- Senden und Verwalten von Java-Anwendungen und der Nachrichtenübermittlung für Unternehmen auf BlackBerry-Geräten über das drahtlose Netzwerk,
- Überwachen der Versionen der Gerätesoftware und Anwendungen, die auf BlackBerry-Geräten in Ihrem Unternehmen ausgeführt werden.

Wenn auf einem Gerät nicht die aktuellsten Versionen der in der Softwarekonfiguration definierten Gerätesoftware und Anwendungen ausgeführt werden, informiert Sie der BlackBerry Manager, dass auf dem Gerät Anwendungen installiert oder aktualisiert werden müssen.

Bevor Sie eine Softwarekonfiguration erstellen und einem Benutzerkonto zuweisen können, müssen Sie zunächst die entsprechende Gerätesoftware und die Anwendungen auf einem Netzwerklaufwerk installieren und freigeben.

Wenn Sie den Speicherort der Gerätesoftware und der Anwendungen auf dem freigegebenen Netzwerklaufwerk angeben, zeigt die Softwarekonfiguration die Anwendungen an, die auf den Geräten installiert oder verwaltet werden können.

Hinweis: Weitere Informationen finden Sie im Kapitel 12.9 »Bereitstellen von BlackBerry-MDS-Studio-Anwendungen für Benutzer«.

11.1 Hinzufügen von Software zu einem Netzwerklaufwerk

Fügen Sie dem Netzwerklaufwerk Gerätesoftware, Java-Anwendungen oder die Nachrichtenübermittlung für Unternehmen hinzu, damit Sie Anwendungen installieren können, die mit dem BlackBerry Manager verbunden sind, und um Anwendungen mithilfe von Softwarekonfigurationen über das drahtlose Netzwerk an BlackBerry-Geräte zu senden.

Auf dem Netzwerklaufwerk kann sich jeweils nur eine Version jeder Anwendung oder jedes Werkzeugs befinden.

Löschen Sie alte Versionen von Anwendungen oder Werkzeugen im Rahmen Ihrer regelmäßigen Wartungsarbeiten.

11.1.1 Auswählen des Netzwerklaufwerks

Wenn Sie Anwendungen auf einem Netzwerklaufwerk speichern, auf das Benutzer zugreifen können, müssen Sie die Anwendungen nicht mehr manuell an die Benutzer-Computer senden und anschließend installieren, um die Anwendungen auf Geräte zu laden. Weitere Informationen über das Aktualisieren von Gerätesoftware erhalten Sie im Enterprise Server Upgrade Guide (Aktualisierungshandbuch für den Enterprise Server).

Wählen Sie ein zentrales Netzwerklaufwerk aus, um die Software und Werkzeuge zu speichern, mit denen Sie Softwarekonfigurationen erstellen und die Gerätesoftware und Anwendungen auf den Geräten installieren und verwalten. Wählen Sie ein Netzwerklaufwerk, auf das alle Benutzer-Computer in Ihrem Unternehmen zugreifen können, um zukünftige Aktualisierungen der Gerätesoftware zu ermöglichen.

Verwenden Sie nach Möglichkeit ein Netzwerklaufwerk, das sich in der Nähe der Benutzer befindet, um das Datenaufkommen im Unternehmens-LAN zu begrenzen, wenn die Benutzer Anwendungen auf Geräten installieren.

11.1.2 Hinzufügen von Software und Werkzeugen auf dem Netzwerklaufwerk

Warnung: Sie allein sind für die Auswahl, Implementierung und Leistung von Drittanbieteranwendungen verantwortlich, die Sie zusammen mit der Geräte- oder Desktopsoftware verwenden. Research In Motion (RIM) übernimmt keinerlei Verantwortung oder Garantie für die Sicherheit, Kompatibilität, Leistung oder Vertrauenswürdigkeit der Anwendungen von Drittanbietern. Research In Motion übernimmt ferner keinerlei Haftung für Probleme, die Ihnen oder Dritten durch die Verwendung solcher Anwendungen von Drittanbietern entstehen.

Führen Sie eine der folgenden Aktionen aus:

Tabelle 11.1: Hinzufügen von Software und Werkzeugen auf dem Netzwerklaufwerk

Aktion	Vorgehensweise
Installieren der BlackBerry-Geräte-Software	1. Holen Sie die Installationsdatei für die BlackBerry-Geräte-Software von Ihrem Dienstanbieter ein. 2. Kopieren Sie die Installationsdatei für die BlackBerry-Geräte-Software auf das Netzwerklaufwerk. 3. Doppelklicken Sie auf dem Netzwerklaufwerk auf die .exe-Datei. 4. Führen Sie die Installation durch. 5. Stellen Sie sicher, dass die Dateien sich im folgenden Verzeichnis befinden: <i><Laufwerk:>\Programme\Gemeinsame Dateien\Research In Motion\Shared\Loader Files\.</i>
Hinzufügen von Java-Anwendungen	Hinweis: Wenn der Drittanbieter vorschreibt, dass Sie die Anwendung zunächst installieren, bevor Sie die Dateien kopieren können, führen Sie die Installation wie vom Drittanbieter angegeben durch, und kopieren Sie anschließend die benötigten Anwendungs- und Moduldateien in den Anwendungsordner. 1. Erstellen Sie auf dem Netzwerklaufwerk den Netzwerkpfad <i><Laufwerk:>\Programme\GemeinsameDateien\Research In Motion\Shared\Applications\.</i> 2. Kopieren Sie im Ordner <i>Applications</i> (Anwendungen) die .ax-, .cod- und .dll-Dateien in einen Unterordner, um die Struktur der Java-Anwendung beizubehalten.

Aktion	Vorgehensweise
Hinzufügen der Nachrichtenübermittlungsanwendung	1. Erstellen Sie auf dem Netzwerklaufwerk den Netzwerkpfad <code><Laufwerk:>\Programme\GemeinsameDateien\Research In Motion\Shared\Applications\</code>. 2. Führen Sie auf der BlackBerry Enterprise Server-Produkt-CD im Ordner <i>IM</i> eine der folgenden Aktionen aus: • Doppelklicken Sie auf <i>lcs.zip</i>, um BlackBerry Instant Messaging für Microsoft Live Communications Server zu verwenden. • Doppelklicken Sie auf <i>sametime.zip</i>, um BlackBerry Instant Messaging für IBM Sametime zu verwenden. • Doppelklicken Sie auf <i>groupwise.zip</i>, um BlackBerry Instant Messaging für Novell® GroupWise zu verwenden. 3. Extrahieren Sie die <i>.alx</i>- und <i>.cod</i>-Dateien in den in Schritt 1 erstellten Netzwerkpfad.
Hinzufügen von BlackBerry MDS Runtime	1. Erstellen Sie auf dem Netzwerklaufwerk den Netzwerkpfad <code><Laufwerk:>\Programme\GemeinsameDateien\Research In Motion\Shared\Applications\</code>. 2. Erstellen Sie einen Ordner für die Anwendung. 3. Kopieren Sie von der Enterprise-Server-Produkt-CD aus dem Ordner MDS Runtime Environment die Datei <i>MdsRuntime.alx</i> und den entsprechenden Ordner für die Version der Gerätesoftware in den in Schritt 2 erstellten Ordner. Unter www.blackberry.com/developers können Sie die aktuelle Version von BlackBerry MDS Runtime herunterladen.

11.1.3 Bereitstellen von Anwendungen für Benutzer

Bei den meisten Anwendungen müssen Sie vor der Installation auf dem Netzwerklaufwerk einen Software-Index erstellen. Um die Software zu indizieren, erstellen Sie für jede Anwendung die Datei *specification.pkg* und die Indexdatei *PkgDBCACHE.xml*.

Die Indexdateien informieren die Softwarekonfiguration und den BlackBerry Application Loader (das BlackBerry-Programm zum Laden von Anwendungen) über die Anwendungen, die zum Installieren auf den Geräten verfügbar sind.

Nicht alle Dateien erfordern eine Indizierung. Wenn Sie dem Netzwerkverzeichnis die Gerätesoftware Version 4.0 oder höher für Java-basierte Geräte oder die Gerätesoftware Version 2.7 oder höher für C++-basierte Geräte hinzugefügt haben, werden die Indexdateien automatisch erstellt.

11.1.4 Erstellen des Software-Index

1. Geben Sie an der Eingabeaufforderung den Befehl:

```
cd <Laufwerk:>\Programme\Gemeinsame Dateien\Research In Motion\Apploader ein.
```

2. Geben Sie `loader.exe /index` ein. Das Programm zum Laden von Anwendungen erstellt auf dem Netzwerklaufwerk die Software-Indexstruktur und fügt fehlende Indexdateien hinzu.

11.1.5 Neu-Indizieren der Softwareanwendungen

Wenn Sie eine `.alx`-Datei nach dem Erstellen eines Software-Index ändern, müssen Sie die Anwendungen neu indizieren.

1. Geben Sie an der Eingabeaufforderung den Befehl `<Laufwerk:>\Programme\GemeinsameDateien\Research In Motion\Apploder` ein.
2. Geben Sie `loader.exe /reindex` ein. Das Programm zum Laden von Anwendungen aktualisiert auf dem Netzwerklaufwerk die Software-Indexstruktur und fügt fehlende Indexdateien hinzu.

11.1.6 Freigeben des Netzwerklaufwerks

1. Geben Sie `<Laufwerk:>\Programme\Gemeinsame Dateien\Research In Motion` frei.
2. Setzen Sie die Berechtigungsattribute auf Read-Only (Schreibgeschützt).

11.2 Erstellen von Softwarekonfigurationen

Sie müssen für jede BlackBerry-Geräteserie in Ihrem Unternehmen eine Softwarekonfiguration erstellen. Wenn Sie eine Softwarekonfiguration erstellen, können Sie Richtlinien zur Anwendungssteuerung definieren, um die Ressourcen anzugeben, auf die Java-Anwendungen, die Nachrichtenübermittlung für Unternehmen und BlackBerry MDS Runtime auf Geräten jenseits der Firewall des Unternehmens zugreifen können.

Außerdem können Sie mit Richtlinien zur Anwendungssteuerung sicherstellen, dass bestimmte Anwendungen auf den Geräten installiert bleiben oder gelöscht werden. Sie können nur Richtlinien zur Anwendungssteuerung für Geräte definieren, auf denen die Gerätesoftware Version 4.0 oder höher ausgeführt wird. Nachdem Sie eine Softwarekonfiguration erstellt und Richtlinien zur Anwendungssteuerung definiert haben, weisen Sie die Softwarekonfiguration einem Benutzerkonto oder einer Gruppe zu, um die Konfigurationsattribute anzuwenden, die auf Geräten installierten Anwendungen zu überwachen und die auf dem Gerät installierten Anwendungen zu steuern.

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Klicken Sie auf der Registerkarte *Software Configurations* (Softwarekonfigurationen) auf *Add New Configuration* (Neue Konfiguration hinzufügen).
3. Geben Sie im Feld *Configuration Name* (Konfigurationsname) einen Namen ein.

4. Geben Sie im Feld *Configuration Description* (Konfigurationsbeschreibung) eine Beschreibung ein.
5. Definieren Sie den Speicherort der BlackBerry-Gerätesoftware, indem Sie auf *Change* (Ändern) klicken.
6. Geben Sie den Speicherort der BlackBerry-Gerätesoftware ein.
7. Klicken Sie auf OK.
8. Markieren Sie in der Liste *Application Name* (Anwendungsname) das Kontrollkästchen neben der Geräteserie, für die Sie die Gerätesoftware konfigurieren möchten.
9. Erweitern Sie die Struktur für die Gerätesoftware beziehungsweise die Anwendung der entsprechenden Geräteserie (zum Beispiel Software für die Serie 7100).

Führen Sie eine der folgenden Aktionen aus:

Tabelle 11.2: Erstellen von Softwarekonfigurationen

Aktion	Vorgehensweise
Installieren von Anwendungen auf Geräten	Aktivieren Sie das Kontrollkästchen neben der Anwendung.
Kein Installieren von Anwendungen auf BlackBerry-Geräten oder Entfernen von Anwendungen von BlackBerry-Geräten.	Deaktivieren Sie das Kontrollkästchen neben der Anwendung.

10. Klicken Sie auf OK.

11.3 Definieren einer Richtlinie zur Anwendungssteuerung

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Führen Sie in der Registerkarte *Software Configurations* (Softwarekonfigurationen) eine der folgenden Aktionen aus:

Tabelle 11.3: Definieren einer Richtlinie zur Anwendungssteuerung

Aktion	Vorgehensweise
Definieren einer Richtlinie zur Anwendungssteuerung	1. Klicken Sie auf <i>Manage Application Policies</i> (Anwendungsrichtlinien verwalten). 2. Klicken Sie auf <i>New</i> (Neu). 3. Geben Sie einen neuen Richtliniennamen ein. 4. Passen Sie die Regeln der Richtlinie zur Anwendungssteuerung an.

Aktion	Vorgehensweise
Zuweisen einer Richtlinie für die Anwendungssteuerung zu einer Anwendung	1. Klicken Sie in der Liste <i>Configuration Name</i> (Konfigurationsname) auf eine Softwarekonfiguration. 2. Klicken Sie auf <i>Edit Configuration</i> (Konfiguration bearbeiten). 3. Erweitern Sie die Anwendungsstruktur <i>Application Software</i> (Anwendungssoftware). 4. Klicken Sie in der Dropdown-Liste <i>Policy</i> (Richtlinie) auf die Richtlinie zur Anwendungssteuerung, die Sie der Anwendung zuweisen möchten. • Um eine Richtlinie für die Anwendungssteuerung allen Anwendungen zuzuweisen, denen derzeit noch keine solche Richtlinie zugewiesen wurde, klicken Sie auf der Ebene der Anwendungssoftware auf eine Richtlinie für die Anwendungssteuerung. • Um die auf der Ebene der Anwendungssoftware zugewiesene Richtlinie für die Anwendungssteuerung zuzuweisen, klicken Sie auf <i><default></i> (Standard). Der Richtlinienname wird nun durch ein Sternchen gekennzeichnet. • Um die auf dem BlackBerry-Gerät vorkonfigurierten Standardrichtlinien für die Anwendungssteuerung zuzuweisen, klicken Sie auf <i><none></i> (keine).

3. Klicken Sie auf *OK*.

11.3.1 Zuweisen von Softwarekonfigurationen zu Benutzerkonten oder -gruppen

1. Führen Sie im BlackBerry Manager eine der folgenden Aktionen aus:

Tabelle 11.4: Zuweisen von Softwarekonfigurationen zu Benutzerkonten oder -gruppen

Aktion	Vorgehensweise
Zuweisen von Softwarekonfigurationen zu einem Benutzerkonto	1. Klicken Sie im linken Fensterbereich auf einen <i>Enterprise Server</i>. 2. Klicken Sie in der Liste <i>Name</i> auf das Benutzerkonto, dem Sie die Softwarekonfiguration zuweisen möchten. 3. Klicken Sie im unteren Fensterbereich auf <i>Device Management</i> (Geräteverwaltung).
Zuweisen von Softwarekonfigurationen zu einer Gruppe	1. Klicken Sie im linken Fensterbereich auf eine Gruppe. 2. Klicken Sie im rechten Fensterbereich auf <i>Device Management</i> (Geräteverwaltung).

2. Klicken Sie auf *Assign Software Configuration* (Softwarekonfiguration zuweisen).

3. Klicken Sie auf die gewünschte Softwarekonfiguration.
4. Klicken Sie auf OK.

11.4 Senden von Anwendungen über das drahtlose Netzwerk an BlackBerry-Geräte

Sie können Java-Anwendungen, die Nachrichtenübermittlung für Unternehmen und die MDS Runtime über das drahtlose Netzwerk an Geräte mit 16 MB oder mehr Flash-Speicher senden, auf denen die Gerätesoftware-Version 4.0 oder höher ausgeführt wird.

Das drahtlose Herunterladen auf die BlackBerry-Geräte kann bis zu vier Stunden dauern.

11.4.1 Senden von Anwendungen an BlackBerry-Geräte

1. Stellen Sie sicher, dass die IT-Richtlinie Ihres Unternehmens Anwendungen von Drittanbietern auf dem Gerät zulässt. Weitere Informationen finden Sie im Abschnitt *Richtlinien*.
2. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
3. Klicken Sie in der Registerkarte *Software Configurations* (Softwarekonfigurationen) in der Liste *Configuration Name* (Konfigurationsname) auf eine Softwarekonfiguration.
4. Klicken Sie auf *Edit Configuration* (Konfiguration bearbeiten).
5. Erweitern Sie die Anwendung.
6. Klicken Sie auf die gewünschte Anwendung.
7. Klicken Sie in der Dropdown-Liste *Delivery* (Zustellung) auf *Wireless* (Drahtlos).
8. Um sicherzustellen, dass die Anwendung auf dem Gerät installiert bleibt, erstellen Sie eine Richtlinie für die Anwendungssteuerung, und weisen Sie diese anschließend zu. Klicken Sie in der Dropdown-Liste *Disposition* (Bereitstellung) auf *Required* (Erforderlich).
9. Klicken Sie auf OK.

12 Einrichten von BlackBerry-Geräten

Dieses Kapitel befasst sich mit folgenden Themen:

- Laden von Benutzernachrichten auf BlackBerry-Geräte
- Option 1: Einrichten der Geräte mit dem BlackBerry Manager
- Option 2: Einrichten der Geräten über das drahtlose Netzwerk
- Option 3: Einrichten der Geräte mit dem BlackBerry Desktop Manager
- Schützen verloren gegangener oder gestohlener Geräte
- Ausgeben vorhandener BlackBerry-Geräte an neue Benutzer

12.1 Laden von Benutzernachrichten auf BlackBerry-Geräte

Im Rahmen der Einrichtung von BlackBerry-Geräten können Sie den Enterprise Server so konfigurieren, dass bisher von Benutzern gesendete und empfangene Nachrichten auf Geräte geladen werden, auf denen mindestens die Version 4.0 der Gerätesoftware ausgeführt wird.

Der Enterprise Server kann Nachrichten für neue Benutzer und für Benutzer laden, deren PIN (Personal Identification Number, persönliche Identifikationsnummer) sich beim Erhalt eines BlackBerry-Ersatzgeräts ändert.

Standardmäßig lädt der Enterprise Server für einen Benutzer über 5 Tage bis zu 200 Nachrichtenkopfzeilen.

Wenn Sie den Enterprise Server so einrichten, dass der Text und die Kopfzeile von Nachrichten auf ein Gerät geladen werden, kann der Enterprise Server über 14 Tage bis zu 750 Nachrichten für einen Benutzer laden.

Wenn der Enterprise Server einem Gerät Nachrichten hinzufügt, verwendet er dabei die Nachrichtenfilterregeln und die Umleitungseinstellungen für ein bestimmtes Benutzerkonto. Weitere Informationen finden Sie unter »Verwalten der Nachrichtenumleitung« in Kapitel 13.

12.1.1 Ändern der Methode zum Laden von Benutzernachrichten auf ein BlackBerry-Gerät

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Server Configuration* (Serverkonfiguration) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Messaging* (Nachrichtenübermittlung).
4. Führen Sie im Bereich *Message Prepopulation* (Vorabladen von Nachrichten) eine oder mehrere der folgenden Aktionen aus:

Tabelle 12.1: Ändern der Methode zum Laden von Benutzernachrichten

Aktion	Vorgehensweise
Ausschließliches Laden von Nachrichtenkopfzeilen auf das BlackBerry-Gerät	Klicken Sie in der Dropdown-Liste <i>Send Headers Only</i> (Nur Kopfzeilen senden) auf <i>True</i> (Wahr).
Laden von Nachrichtenkopfzeilen und Nachrichtentext auf das BlackBerry-Gerät	Klicken Sie in der Dropdown-Liste <i>Send Headers Only</i> (Nur Kopfzeilen senden) auf <i>False</i> (Falsch).
Festlegen der Anzahl der zurückliegenden Tage, für die Nachrichten geladen werden	Geben Sie im Feld <i>Prepopulation By Message Age</i> (Vorabladen nach Nachrichtenalter) eine Anzahl ein.
Einstellen der Höchstanzahl zu ladender Nachrichten	Geben Sie im Feld <i>Prepopulation By Message Count</i> (Vorabladen nach Nachrichtenanzahl) eine Anzahl ein.

5. Klicken Sie auf OK.

12.1.2 Verhindern des Ladens von Benutzernachrichten auf ein BlackBerry-Gerät

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Server Configuration* (Serverkonfiguration) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Messaging* (Nachrichtenübermittlung).
4. Geben Sie im Bereich *Message Prepopulation* (Vorabladen von Nachrichten) im Feld *Prepopulation By Message Age* (Vorabladen nach Nachrichtenalter) die Menge 0 ein.
5. Geben Sie im Bereich *Message Prepopulation* (Vorabladen von Nachrichten) im Feld *Prepopulation By Message Count* (Vorabladen nach Nachrichtenanzahl) die Menge 0 ein.
6. Klicken Sie auf OK.

12.2 Option 1: Geräte mit dem BlackBerry Manager einrichten

Wenn Sie die Aktivierung und Ersteinrichtung von Geräten steuern möchten, schließen Sie die Geräte an den Computer an, auf dem der BlackBerry Manager installiert ist, und weisen Sie sie Benutzerkonten zu.

Wenn Sie einem Benutzerkonto ein Gerät zuweisen, verknüpfen Sie das Gerät mit dem Nachrichtenkonto des Benutzers und installieren die Servicebücher auf dem Gerät.

12.2.1 Zuweisen eines Geräts zu einem Benutzerkonto

1. Schließen Sie das Gerät an den Computer an, auf dem der BlackBerry Manager installiert ist.
2. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
3. Klicken Sie auf der Registerkarte *Users* (Benutzer) auf das Benutzerkonto, dem das Gerät zugewiesen werden soll.
4. Klicken Sie auf *Device Management* (Geräteverwaltung).
5. Klicken Sie auf *Assign Handheld* (Handheld zuweisen).
6. Klicken Sie auf das Gerät, das dem Benutzerkonto zugewiesen werden soll.
7. Klicken Sie auf *OK*.

12.3 Option 2: Geräten über das drahtlose Netzwerk einrichten

Richten Sie Geräte über das drahtlose Netzwerk ein, damit Benutzer, die ein Gerät neu oder im Austausch erhalten oder erwerben, das Smartphone ohne physische Verbindung mit dem Unternehmensnetzwerk einrichten können.

Durch die drahtlose Unternehmensaktivierung mit dem Vorabladen von Nachrichten und der automatischen drahtlosen Sicherung sind Benutzer, deren Gerät ausgefallen ist, mit einem Ersatzgerät schnell wieder einsatzbereit.

Sie richten die Geräte über das drahtlose Netzwerk ein, indem Sie Kennwörter zur drahtlosen Unternehmensaktivierung an Benutzerkonten senden. Die Benutzer erhalten auf ihren Desktop-E-Mail-Anwendungen eine Nachricht mit dem Kennwort für die drahtlose Unternehmensaktivierung.

Durch die Verwendung von IT-Richtlinien für die Synchronisierung von Terminplannerdaten können Sie festlegen, ob Benutzer das Gerät während der Einrichtung an den

Computer anschließen müssen oder die Einrichtung über das drahtlose Netzwerk durchführen können.

12.3.1 Senden von Terminplanerdaten über den BlackBerry-Router

Die erste Massenladung von Terminplanerdaten vom Enterprise Server während der Einrichtung von BlackBerry-Geräten erfolgt standardmäßig über das drahtlose Netzwerk.

Um Bandbreite zu sparen, können Sie festlegen, dass der Enterprise Server die Massenladung von Terminplanerdaten durch den BlackBerry-Router über das Unternehmens-LAN senden soll. Der Router überträgt die Terminplanerdaten, wenn Benutzer ihr Gerät an den Computer anschließen. Die Benutzer müssen den Device-Manager installiert haben.

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Klicken Sie auf der Registerkarte *Global* auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *IT Policy* (IT-Richtlinie).
4. Doppelklicken Sie im Bereich *IT Policy Administration* (IT-Richtlinienverwaltung) auf *IT Policies* (IT-Richtlinien).
5. Führen Sie eine der folgenden Aktionen aus:

Tabelle 12.2: Senden von Terminplanerdaten über den BlackBerry-Router

Aktion	Vorgehensweise
Deaktivieren der drahtlosen ersten Synchronisierung von Terminplanerdaten unter Verwendung der Standard-IT-Richtlinie	1. Klicken Sie in der Richtlinienliste auf <i>Default</i> (Standard). 2. Klicken Sie auf <i>Properties</i> (Eigenschaften). 3. Klicken Sie auf <i>PIM Sync Policy Group</i> (Richtliniengruppe für PIM-Synchronisierung). 4. Klicken Sie auf die IT-Richtlinienregel <i>Disable Wireless Bulk Loads</i> (Drahtlose Massenladungen deaktivieren). 5. Klicken Sie in der Dropdown-Liste auf <i>True</i> (Wahr).

Aktion	Vorgehensweise
Erstellen einer neuen IT-Richtlinie, Deaktivieren der drahtlosen ersten Synchronisierung von Terminplannerdaten und Senden der IT-Richtlinie an Benutzerkonten	1. Klicken Sie auf <i>New</i> (Neu). 2. Geben Sie einen Richtliniennamen ein. 3. Klicken Sie auf <i>PIM Sync Policy Group</i> (Richtliniengruppe für PIM-Synchronisierung). 4. Klicken Sie auf die IT-Richtlinienregel <i>Disable Wireless Bulk Loads</i> (Drahtlose Massenladungen deaktivieren). 5. Klicken Sie in der Dropdown-Liste auf <i>True</i> (Wahr). 6. Klicken Sie auf <i>OK</i>. 7. Klicken Sie erneut auf <i>OK</i>. 8. Doppelklicken Sie im Bereich <i>IT Policy Administration</i> (IT-Richtlinienverwaltung) auf <i>IT Policies</i> (IT-Richtlinien). 9. Klicken Sie im linken Fensterbereich auf ein Benutzerkonto. 10. Klicken Sie im rechten Fensterbereich auf die neue Richtlinie.

6. Klicken Sie auf *OK*.

7. Fordern Sie die Benutzer auf, ihr Gerät an den Computer anzuschließen. Anschließend wird der BlackBerry Device Manager gestartet.

12.3.2 Kennwörter für die drahtlose Unternehmensaktivierung

Das Kennwort für die drahtlose Unternehmensaktivierung ist für das jeweilige Benutzerkonto spezifisch. Ein Kennwort für die drahtlose Unternehmensaktivierung läuft standardmäßig nach 48 Stunden oder fünf erfolglosen Eingaberversuchen des Benutzers auf dem Gerät ab.

Wenn ein Benutzer ein Kennwort für die drahtlose Unternehmensaktivierung erhalten hat, können Sie für diesen Benutzer erst dann ein neues Kennwort erstellen, wenn das aktive Kennwort abgelaufen ist.

Das Kennwort für die drahtlose Unternehmensaktivierung wird inaktiv, sobald es vom Benutzer erfolgreich auf einem Gerät eingegeben wurde.

12.3.3 Anpassen des Kennworts und der Nachricht für die drahtlose Unternehmensaktivierung

Passen Sie die Standardnachricht für die drahtlose Unternehmensaktivierung, die Benutzer in ihrer Desktop-E-Mail-Anwendung erhalten, so an, dass die Nachricht der Unternehmensrichtlinie für die Nachrichtenübermittlung entspricht oder dass Kontaktangaben für den Support bekannt gegeben werden. Benutzer können sich dann an den Support wenden, wenn während der Aktivierung des Geräts Fehler auftreten.

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Klicken Sie auf der Registerkarte *Global* auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *General* (Allgemein).
4. Führen Sie im Bereich *Administration* (Verwaltung) folgende Aktionen aus:

Tabelle 12.3: Anpassen des Kennworts für die drahtlose Unternehmensaktivierung

Aktion	Vorgehensweise
Anpassen der Nachricht für die drahtlose Unternehmensaktivierung	1. Doppelklicken Sie auf <i>Custom Activation Email Message</i> (Benutzerdefinierte E-Mail-Aktivierungsnachricht). 2. Geben Sie die gewünschten Parameter, den gewünschten Betreff und die gewünschte Nachricht ein.
Festlegen der Länge des Kennworts für die drahtlose Unternehmensaktivierung	1. Doppelklicken Sie auf <i>Auto-generated password length</i> (Automatisch erstellte Kennwortlänge). 2. Geben Sie die Länge des Kennworts für die drahtlose Unternehmensaktivierung ein.
Festlegen des Typs der drahtlosen Unternehmensaktivierung	Klicken Sie im Dropdown-Feld <i>Auto-generated password type</i> (Automatisch erstellter Kennworttyp) auf einen Kennworttyp. Tipp: Klicken Sie für die BlackBerry-Serie 7100 auf den Kennworttyp <i>7100 Friendly</i> (7100-freundlich). Das Kennwort besteht aus Zeichen, für die jeweils nur einmal eine bestimmte Taste zu drücken ist.

5. Klicken Sie auf OK.

12.3.4 Senden eines Kennworts für die drahtlose Unternehmensaktivierung an ein Benutzerkonto

Kennwörter für die drahtlose Unternehmensaktivierung unterstützen keine Zeichen mit Akzenten.

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *User Groups* (Benutzergruppen).
2. Klicken Sie auf der Registerkarte *User Groups List* (Benutzergruppenliste) auf eine Gruppe.
3. Klicken Sie auf *Service Access* (Dienstzugriff).
4. Klicken Sie auf *Generate and send E-mail Activation Password* (Aktivierungskennwort erstellen und per E-Mail senden).
5. Klicken Sie auf OK.

Tabelle 12.4: Senden eines Kennworts für die drahtlose Unternehmensaktivierung

Aktion	Vorgehensweise
Erstellen des Kennworts für die drahtlose Unternehmensaktivierung und Senden dieses Kennworts an den Benutzer in einer Nachricht	1. Klicken Sie auf <i>Generate and send E-mail Activation Password</i> (Aktivierungskennwort erstellen und per E-Mail senden).
Festlegen des Aktivierungskennworts und des Ablaufzeitpunkts dieses Kennworts	1. Klicken Sie auf <i>Set Activation Password</i> (Aktivierungskennwort festlegen). 2. Geben Sie ein Kennwort für die drahtlose Unternehmensaktivierung ein. 3. Bestätigen Sie das Kennwort, indem Sie es erneut eingeben. 4. Klicken Sie in der Dropdown-Liste <i>Password Expires in</i> (Kennwort läuft ab in) auf einen Ablaufzeitpunkt. 5. Klicken Sie auf <i>OK</i> . 6. Teilen Sie dem Benutzer das neue Kennwort mit.

12.4 Option 3: Geräte mit dem BlackBerry Desktop Manager einrichten

Damit die erste Aktivierung und Einrichtung von Geräten durch Benutzer ausgeführt werden kann, schließen Sie die Geräte an die Computer von Benutzern an, auf denen der BlackBerry Desktop Manager installiert ist.

Wenn das Gerät während der Einrichtung an den Computer eines Benutzers angeschlossen ist, sendet der Enterprise Server die Terminplanerdaten durch den BlackBerry-Router über das LAN und nicht über das drahtlose Netzwerk an das Gerät. Wenn die Verbindung zum BlackBerry-Router abbricht, wird die Datenübertragung über das drahtlose Netzwerk fortgesetzt.

1. Stellen Sie sicher, dass der BlackBerry Desktop Manager auf dem Computer des Benutzers installiert ist.
2. Weisen Sie den Benutzer an, den BlackBerry Desktop Manager zu starten und das Gerät an den Computer anzuschließen.

Der Benutzer wird durch eine Nachricht aufgefordert, das Gerät dem E-Mail-Konto zuzuweisen.

In einer zweiten Nachricht wird der Benutzer aufgefordert, einen Codierungsschlüssel für die Aktivierung des Geräts zu erstellen. Nach der vollständigen Aktivierung lädt der Enterprise Server Nachrichten und Terminplanerdaten auf das Gerät.

12.5 Schützen verloren gegangener oder gestohlener BlackBerry-Geräte

Wenn ein Benutzer ein Gerät verliert oder es ihm gestohlen wird, können die enthaltenen Daten geschützt werden. Dazu werden über den BlackBerry Manager Befehle zum Sperren oder Nichtverfügbarmachen abgesetzt.

12.5.1 Schützen eines verloren gegangenen BlackBerry-Geräts

Hinweis: Wenn ein Benutzer das Kennwort eines BlackBerry-Geräts vergisst, auf dem der Inhaltsschutz aktiviert ist, verwenden Sie nicht den Befehl *Set a Password and Lock Handheld* (Kennwort festlegen und Handheld sperren), um das Kennwort entfernt zurückzusetzen.

Wenn Sie das Kennwort des Benutzers entfernt zurücksetzen, fordert das Gerät mit aktiviertem Inhaltsschutz den Benutzer auf, das vergessene Kennwort einzugeben, bevor er das neue Kennwort eingibt.

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Users* (Benutzer) auf ein Benutzerkonto.
3. Klicken Sie auf *IT Admin* (IT-Verwaltung).
4. Klicken Sie auf *Set Password and Lock Handheld* (Kennwort festlegen und Handheld sperren).
5. Geben Sie in die Felder *New Password* (Neues Kennwort) und *New Password Again* (Neues Kennwort wiederholen) ein Kennwort ein, das zwischen 4 und 14 Zeichen lang ist.

Warnung: Verwenden Sie beim Erstellen des Kennworts keine Sonderzeichen, da sie unter Umständen vom BlackBerry-Gerät nicht unterstützt werden.

12.5.2 Schützen eines gestohlenen BlackBerry-Geräts

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Users* (Benutzer) auf ein Benutzerkonto.
3. Klicken Sie auf *IT Admin* (IT-Verwaltung).
4. Klicken Sie auf *Erase Data and Disable Handheld* (Daten löschen und Handheld deaktivieren).
5. Klicken Sie auf *Yes* (Ja).

12.6 Ausgeben vorhandener BlackBerry-Geräte an neue Benutzer

Wenn Sie ein vorhandenes Gerät an einen neuen Benutzer ausgeben möchten, bereiten Sie es für die Neuverteilung vor, indem Sie die Anwendungsdaten des vorherigen Benutzers vom Gerät löschen und Anwendungen zum Handheld hinzufügen beziehungsweise davon entfernen.

Wenn sämtliche Anwendungen und Daten vom Gerät entfernt werden sollen, setzen Sie das Gerät auf die Standardanwendungskonfiguration zurück.

12.6.1 Vorbereiten eines BlackBerry-Geräts auf die Neuverteilung

Führen Sie eine der folgenden Aktionen aus:

Tabelle 12.5: Vorbereiten eines BlackBerry-Geräts auf die Neuverteilung

Aktion	Vorgehensweise
Löschen der Anwendungsdaten des vorherigen Benutzers über ein drahtloses Netzwerk und Sperren des BlackBerry-Geräts	Sperren Sie das BlackBerry-Gerät, und löschen Sie die darauf enthaltenen Daten. Weitere Informationen finden Sie im Kapitel 12.5.2 unter »Schützen eines gestohlenen BlackBerry-Geräts«.
Löschen der Anwendungsdaten des vorherigen Benutzers mit dem BlackBerry Manager	1. Schließen Sie das Gerät an den Computer an, auf dem der BlackBerry Manager installiert ist. 2. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf <i>Local Ports (Device Management)</i> (Lokale Ports – Geräteverwaltung). 3. Klicken Sie in der Liste <i>Connection</i> (Verbindung) auf eine Verbindung. 4. Klicken Sie auf <i>Wipe Device File System</i> (Gerätedateisystem löschen). 5. Klicken Sie auf <i>Yes</i> (Ja). 6. Geben Sie bei Aufforderung das Kennwort des Geräts ein, um die Aufgabe abzuschließen.

Aktion	Vorgehensweise
Installieren oder Entfernen von Anwendungen auf dem BlackBerry-Gerät	1. Schließen Sie das Gerät an den Computer an, auf dem der BlackBerry Manager installiert ist. 2. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf <i>Local Ports (Device Management)</i> (Lokale Ports – Geräteverwaltung). 3. Klicken Sie in der Liste <i>Connection</i> (Verbindung) auf eine Verbindung. 4. Klicken Sie auf <i>Load Device</i> (Interactive) (Gerät laden – Interaktiv). 5. Klicken Sie auf eine Softwarekonfiguration. 6. Klicken Sie auf <i>OK</i>. 7. Führen Sie im Fenster für die Konfiguration der Gerätesoftware eine der folgenden Aktionen aus: • Deaktivieren Sie die Kontrollkästchen neben den zu entfernenden Anwendungen. • Aktivieren Sie die Kontrollkästchen neben den zu installierenden Anwendungen. 8. Führen Sie den Assistenten für das Programm zum Laden von Anwendungen vollständig aus.
Zurücksetzen eines BlackBerry-Geräts auf die werksseitigen Standardeinstellungen	1. Schließen Sie das Gerät an den Computer an, auf dem der BlackBerry Manager installiert ist. 2. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf <i>Local Ports (Device Management)</i> (Lokale Ports – Geräteverwaltung). 3. Klicken Sie in der Liste <i>Connection</i> (Verbindung) auf eine Verbindung. 4. Klicken Sie auf <i>Nuke Device</i> (Gerät auf werksseitige Einstellungen zurücksetzen). 5. Klicken Sie auf <i>Yes</i> (Ja). 6. Klicken Sie auf <i>Load Device</i> (Interactive) (Gerät laden – Interaktiv). 7. Klicken Sie auf eine Softwarekonfiguration. 8. Klicken Sie auf <i>OK</i>. 9. Führen Sie den Assistenten für das Programm zum Laden von Anwendungen vollständig aus.

12.6.2 Neuverteilen des BlackBerry-Geräts an einen Benutzer

Wenn ein Benutzer vor dem Erhalt eines neuverteilten Geräts die bereits empfangenen Nachrichten auf dem Computer sichert, können Sie das Vorabladen von Nachrichten

deaktivieren. Weitere Informationen finden Sie im Kapitel 12.1.2, »Verhindern des Ladens von Benutzernachrichten auf ein BlackBerry-Gerät«.

Wenn ein Benutzer ein Ersatzgerät erhält, richten Sie das Gerät ein, um die neue PIN für die Nachrichtenumleitung zu registrieren.

Weitere Informationen finden Sie unter »Option 2: Geräten über das drahtlose Netzwerk einrichten«.

12.7 Bereitstellen von BlackBerry-MDS-Studio-Anwendungen für Benutzer

Dieser Abschnitt befasst sich mit folgenden Themen:

- Zulassen der Authentifizierung von BlackBerry MDS Services beim BlackBerry Manager
- Konfigurieren der BlackBerry-Anwendungen, die auf BlackBerry-Geräten installiert werden können
- Vorbereiten von BlackBerry-Geräten auf die Installation von BlackBerry-Anwendungen
- Senden von BlackBerry-Anwendungen an BlackBerry-Geräte
- Entfernen von BlackBerry-Anwendungen aus der Datenbank und von BlackBerry-Geräten
- Überwachen von BlackBerry-MDS-Services-Nachrichten
- Festlegen der Verbindung zwischen BlackBerry-MDS-Services und dem BlackBerry MDS Connection Service

12.7.1 Zulassen der Authentifizierung von BlackBerry MDS Services beim BlackBerry Manager

BlackBerry MDS Services speichert ein selbst signiertes Zertifikat in seinem Schlüsselspeicher. Zweck der Installation dieses Zertifikats ist, eine vom Server authentifizierte Kommunikation zwischen BlackBerry MDS Services und dem BlackBerry Manager herzustellen.

Wenn Sie das selbst signierte Zertifikat von BlackBerry MDS Services durch ein signiertes Stammzertifikat einer Zertifizierungsstelle ersetzen, tun Sie dies vor der Authentifizierung beim BlackBerry Manager mithilfe des selbstsignierten Zertifikats.

12.7.2 Erstellen einer Serverauthentifizierung zwischen BlackBerry MDS Services und dem BlackBerry Manager

Der BlackBerry Manager fordert Sie beim ersten Verbindungsaufbau zwischen dem BlackBerry Manager und BlackBerry MDS Services auf, das selbst signierte Zertifikat für MDS Services anzuzeigen und zu installieren.

Das Zertifikat wird als vertrauenswürdiges Stammzertifikat installiert und ermöglicht die sichere Kommunikation zwischen dem BlackBerry Manager und BlackBerry MDS Services. Wenn Sie das selbst signierte Zertifikat von BlackBerry MDS Services durch ein Stammzertifikat einer Zertifizierungsstelle ersetzen, akzeptiert der BlackBerry Manager das Stammzertifikat und authentifiziert sich bei BlackBerry MDS Services.

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen BlackBerry MDS Services Server.
2. Klicken Sie im Dialogfeld zur Zertifikatinstallation auf *View Certificate* (Zertifikat anzeigen).
3. Überprüfen Sie die Zertifikatinformationen.
4. Klicken Sie auf *Install Certificate* (Zertifikat installieren).
5. Beenden Sie den Installationsassistenten, indem Sie die Standardeinstellungen übernehmen.
6. Klicken Sie bei der Anmeldeaufforderung auf *Cancel* (Abbrechen).

12.7.3 Konfigurieren der BlackBerry-Anwendungen, die auf den Geräten installiert werden können

BlackBerry-Anwendungen können von Anwendungsentwicklern mit einem digitalen Zertifikat signiert werden.

Sie verwalten vertrauenswürdige Zertifikate, mit denen Anwendungen von BlackBerry MDS Services authentifiziert werden.

Wenn Anwendungen nicht über vertrauenswürdige Zertifikate verfügen, konfigurieren Sie, ob nicht signierte Anwendungen, die in der Datenbank veröffentlicht wurden, von Benutzern auf BlackBerry-Geräten installiert werden dürfen.

12.7.4 Zulassen der Installation nicht signierter Anwendungen auf Geräten durch Benutzer

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen BlackBerry MDS Services Server.
2. Klicken Sie auf der Registerkarte MDS Services (MDS-Dienste) auf *Edit Properties* (Eigenschaften bearbeiten).

3. Klicken Sie auf *General* (Allgemein).
4. Klicken Sie auf *Allow Unsigned Applications* (Nicht signierte Anwendungen zulassen).
5. Klicken Sie in der Dropdown-Liste auf *True* (Wahr).
6. Klicken Sie auf *OK*.
7. Klicken Sie auf *Common* (Gemeinsam).
8. Klicken Sie auf *Stop Service* (Dienst anhalten).
9. Wenn als Status *Stopped* (Angehalten) angezeigt wird, klicken Sie auf *Start Service* (Dienst starten).

12.7.5 Verwalten eines vertrauenswürdigen Zertifikats

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen MDS Services Server.
2. Führen Sie eine der folgenden Aktionen aus:

Tabelle 12.6: Verwalten eines vertrauenswürdigen Zertifikats

Aktion	Vorgehensweise
Hinzufügen eines Zertifikats zum BlackBerry MDS Services Server	1. Klicken Sie auf <i>Add Certificate</i> (Zertifikat hinzufügen). 2. Geben Sie im Feld <i>Alias</i> (Aliasname) einen Zertifikatsnamen ein. 3. Geben Sie im Feld <i>Certificate file</i> (Zertifikatsdatei) den Zertifikatspfad und den Dateinamen mit der Endung <i>.CER</i> ein. 4. Klicken Sie auf <i>OK</i>.
Entfernen eines Zertifikats vom BlackBerry MDS Services Server	1. Klicken Sie auf der Registerkarte <i>MDS Services</i> (MDS-Dienste) auf <i>Edit Properties</i> (Eigenschaften bearbeiten). 2. Klicken Sie auf <i>Certificate</i> (Zertifikat). 3. Doppelklicken Sie auf <i>BlackBerry MDS Services Certificate Definition</i> (Zertifikatsdefinition von BlackBerry MDS Services). 4. Klicken Sie auf ein Zertifikat. 5. Klicken Sie auf <i>Remove</i> (Entfernen). 6. Klicken Sie auf <i>OK</i>. 7. Klicken Sie erneut auf <i>OK</i>.

12.7.6 Vorbereiten von Geräten auf die Installation von BlackBerry-MDS-Studio-Anwendungen

Vor dem Installieren von MDS-Studio-Anwendungen müssen Benutzer BlackBerry MDS Runtime auf dem Gerät installieren und aktivieren.

BlackBerry MDS Runtime kann sowohl über das Drahtlosnetzwerk als auch mithilfe des Programms zum Laden von Anwendungen installiert werden.

Weitere Informationen finden Sie im Kapitel 11.5, »Senden von Anwendungen über das drahtlose Netzwerk an BlackBerry-Geräte«.

Geräterichtlinien für MDS Services können für folgende Zwecke erstellt und Benutzerkonten beziehungsweise -gruppen zugewiesen werden:

- zum Steuern der Fähigkeit eines Benutzers, BlackBerry-MDS-Studio-Anwendungen auf dem Gerät zu erkennen, zu installieren und zu entfernen,
- zum Steuern des Zugriffs von MDS-Studio-Anwendungen auf andere Daten und Anwendungen auf dem Gerät,
- zum Konfigurieren der lokalen Speicherkapazität für MDS-Studio-Anwendungsnachrichten auf dem Gerät.

12.7.7 Definieren und Verwalten einer MDS-Services-Geräterichtlinie zum Steuern von MDS-Studio-Anwendungen.

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen BlackBerry MDS Services Server.
2. Klicken Sie auf der Registerkarte *MDS Services* (MDS-Dienste) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Device Policies* (Geräterichtlinien).
4. Doppelklicken Sie auf *BlackBerry MDS Services Device Policy Definition* (Geräterichtliniendefinition von BlackBerry MDS Services).
5. Führen Sie eine der folgenden Aktionen aus:

Tabelle 12.7: Definieren und Verwalten einer BlackBerry-MDS-Services-Geräterichtlinie zum Steuern von BlackBerry-MDS-Studio-Anwendungen

Aktion	Vorgehensweise
Erstellen einer Gerä- terichtlinie für Black- Berry MDS Services	1. Klicken Sie auf <i>New</i> (Neu). 2. Doppelklicken Sie auf <i>Policy Name</i> (Richtliniennamen). 3. Geben Sie einen Namen für die Geräterichtlinie von BlackBerry MDS Services ein. 4. Legen Sie die Einstellungen für die Geräterichtlinie von MDS Services fest. Weitere Informationen finden Sie im Policy Reference Guide (Referenzhandbuch für Richtlinien). 5. Klicken Sie auf <i>OK</i>.
Entfernen einer Gerä- terichtlinie für Black- Berry MDS Services	1. Klicken Sie auf den Namen für die Geräterichtlinie von BlackBerry MDS Services. 2. Klicken Sie auf <i>Remove</i> (Entfernen). 3. Klicken Sie auf <i>OK</i>.

12.7.8 Zuweisen einer Geräterichtlinie von MDS Services zu einem Benutzerkonto oder einer Benutzergruppe

Je nach Ihrer Administratorrolle können Sie Benutzerkonten und -gruppen Geräterichtlinien von MDS Services zuweisen.

Damit Sie einer Gruppe eine Geräterichtlinie von MDS Services zuweisen können, muss die Benutzergruppe mindestens ein Benutzerkonto enthalten, und sämtliche in einer Gruppe enthaltenen Benutzerkonten müssen mit demselben MDS Services Server verbunden sein.

1. Führen Sie im BlackBerry Manager im linken Fensterbereich eine der folgenden Aktionen aus:

Tabelle 12.8: Zuweisen einer Geräterichtlinie von BlackBerry-MDS

Aktion	Vorgehensweise
Zuweisen einer Geräte- richtlinie von MDS Ser- vices zu einer Gruppe von Benutzerkonten	1. Klicken Sie auf eine Benutzergruppe. 2. Klicken Sie auf der Registerkarte <i>Users</i> (Benutzer) mit der rechten Maustaste auf eine Spaltenüberschrift. 3. Klicken Sie in der Liste <i>Available columns</i> (Vorhandene Spalten) auf <i>MDS Services Server URL</i> (MDS-Services-Server-URL). 4. Klicken Sie auf <i>Insert</i> (Einfügen). 5. Klicken Sie auf <i>OK</i>. 6. Klicken Sie auf die Spaltenüberschrift <i>MDS Services Server URL</i> (MDS Services-Server-URL), damit Benutzerkonten nach BlackBerry MDS Services Server sortiert werden. 7. Klicken Sie auf Benutzerkonten, die mit demselben BlackBerry MDS Services Server verbunden sind. 8. Klicken Sie auf der Registerkarte <i>Group Configuration</i> (Gruppenkonfiguration) auf <i>MDS Services</i> (MDS-Dienste).
Zuweisen einer Geräte- richtlinie von MDS Ser- vices zu einem Benut- zerkonto	1. Klicken Sie auf einen MDS Services Server. 2. Klicken Sie auf <i>Devices Registered</i> (Registrierte Geräte). 3. Klicken Sie auf der Registerkarte <i>Devices Registered</i> (Registrierte Geräte) auf ein Benutzerkonto. 4. Klicken Sie auf <i>Common</i> (Gemeinsam).

2. Klicken Sie auf *Assign Device Policy* (Geräterichtlinie zuweisen).
3. Klicken Sie in der Dropdown-Liste *Device Policy* (Geräterichtlinie) auf eine Richtlinie von MDS Services.
4. Klicken Sie auf *OK*.

12.7.9 Senden von MDS-Studio-Anwendungen an BlackBerry-Geräte

Je nach Ihrer Administratorrolle können Sie BlackBerry-MDS-Studio-Anwendungen und -Upgrades an Benutzer und Benutzergruppen senden.

12.7.10 Installieren einer MDS-Studio-Anwendung auf einem BlackBerry-Gerät

1. Führen Sie im BlackBerry Manager im linken Fensterbereich eine der folgenden Aktionen aus:

Tabelle 12.9: Installieren einer BlackBerry-MDS-Studio-Anwendung

Aktion	Vorgehensweise
Installieren einer MDS-Studio-Anwendung auf BlackBerry-Geräten für eine Gruppe von Benutzerkonten, die dieselben BlackBerry MDS Services verwenden	1. Klicken Sie auf eine Gruppe. 2. Klicken Sie auf der Registerkarte <i>Users</i> (Benutzer) mit der rechten Maustaste auf eine Spaltenüberschrift. 3. Klicken Sie in der Liste <i>Available columns</i> (Vorhandene Spalten) auf <i>MDS Services Server URL</i> (MDS Services-Server-URL). 4. Klicken Sie auf <i>Insert</i> (Einfügen). 5. Klicken Sie auf <i>OK</i>. 6. Klicken Sie auf die Spaltenüberschrift <i>MDS Services Server URL</i> (MDS Services Server-URL), damit die Benutzer nach BlackBerry MDS Services Server sortiert werden. 7. Klicken Sie auf Benutzerkonten, die mit demselben MDS Services Server verbunden sind. 8. Klicken Sie auf der Registerkarte <i>Group Configuration</i> (Gruppenkonfiguration) auf <i>MDS Services</i> (MDS-Dienste). 9. Klicken Sie auf <i>Install on Device</i> (Auf Gerät installieren). 10. Klicken Sie auf die zu installierende BlackBerry-MDS-Studio-Anwendung.
Installieren einer BlackBerry-MDS-Studio-Anwendung auf einem einzigen BlackBerry-Gerät	1. Klicken Sie auf einen MDS Services Server. 2. Klicken Sie auf <i>Application Registry</i> (Anwendungsregistrierung). 3. Klicken Sie auf eine BlackBerry-MDS-Studio-Anwendung. 4. Klicken Sie auf <i>Device Management</i> (Geräteverwaltung). 5. Klicken Sie auf <i>Install on Device</i> (Auf Gerät installieren). 6. Klicken Sie in der Dropdown-Liste <i>Install application on devices</i> (Anwendung auf Geräten installieren) auf <i>without application installed</i> (Ohne installierte Anwendung). 7. Deaktivieren Sie das Kontrollkästchen <i>Select all</i> (Alle auswählen). 8. Klicken Sie auf die PIN des Geräts, an das die MDS-Studio-Anwendung übermittelt werden soll.

2. Klicken Sie auf *Next* (Weiter).

3. Führen Sie folgende Aktionen aus:

Tabelle 12.10: Installieren einer BlackBerry-MDS-Studio-Anwendung – weitere Aktionen

Aktion	Vorgehensweise
Festlegen der Anzahl der BlackBerry-Geräte, an die die BlackBerry-MDS-Studio-Anwendung gleichzeitig gesendet werden soll	Geben Sie im Feld <i>Group size for pushing</i> (Gruppengröße zum Übermitteln) eine Anzahl ein.
Festlegen des Intervalls (in Minuten), in dem die Anfrage zur Installation der BlackBerry-MDS-Studio-Anwendung an BlackBerry-Geräte gesendet werden soll	Geben Sie im Feld <i>Push interval (minute)</i> (Übermittlungsintervall in Minuten) einen Wert ein.
Festlegen eines bestimmten Zeitpunkts, zu dem die BlackBerry-MDS-Studio-Anwendung an BlackBerry-Geräte gesendet werden soll	1. Aktivieren Sie das Kontrollkästchen <i>Schedule</i> (Zeitplan). 2. Klicken Sie in der Dropdown-Liste <i>Start at</i> (Beginnen um) auf ein Datum. 3. Legen Sie den Startzeitpunkt fest. Hinweis: Wenn Sie keinen Startzeitpunkt festlegen, sendet MDS Services die MDS-Studio-Anwendung sofort.
Konfigurieren der stillen Installation der MDS-Studio-Anwendung auf den angegebenen Geräten	Klicken Sie auf <i>Required</i> (Erforderlich). Hinweis: Wenn Sie die BlackBerry-MDS-Studio-Anwendung nicht still auf dem BlackBerry-Gerät installieren, wird der Benutzer aufgefordert, die BlackBerry-MDS-Studio-Anwendung zu installieren.

4. Klicken Sie auf *Next* (Weiter).

5. Klicken Sie auf *Finish* (Fertigstellen).

12.7.11 Durchführen eines Upgrades einer MDS-Studio-Anwendung auf einem BlackBerry-Gerät

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen BlackBerry MDS Services Server.
2. Klicken Sie auf *Application Registry* (Anwendungsregistrierung).
3. Klicken Sie auf der Registerkarte *Application Registry* (Anwendungsregistrierung) auf die MDS-Studio-Anwendung.
4. Klicken Sie auf *Device Management* (Geräteverwaltung).

5. Führen Sie eine der folgenden Aktionen aus:

Tabelle 12.11: Durchführen eines Upgrades einer BlackBerry-MDS-Studio-Anwendung

Aktion	Vorgehensweise
Durchführen eines Upgrades einer BlackBerry-MDS-Studio-Anwendung auf einem einzigen BlackBerry-Gerät	1. Klicken Sie auf <i>Upgrade on Device</i> (Upgrade auf Gerät durchführen). 2. Klicken Sie in der Dropdown-Liste <i>Upgrade application on devices</i> (Anwendungs-Upgrade auf Gerät durchführen) auf <i>with old version of application</i> (Mit früherer Anwendungsversion). 3. Deaktivieren Sie das Kontrollkästchen <i>Select all</i> (Alle auswählen). 4. Klicken Sie auf die PIN des Geräts, an das das Upgrade der BlackBerry-MDS-Studio-Anwendung übermittelt werden soll.
Durchführen eines Upgrades einer MDS-Studio-Anwendung auf BlackBerry-Geräten und Installieren der Anwendung auf Geräten, auf denen die MDS-Studio-Anwendung derzeit nicht installiert ist	Klicken Sie auf <i>Install on Device</i> (Auf Gerät installieren). Klicken Sie in der Dropdown-Liste <i>Install application on devices</i> (Anwendung auf Geräten installieren) auf <i>with or without application installed</i> (Mit oder ohne installierte Anwendung).

6. Klicken Sie auf *Next* (Weiter).

7. Führen Sie folgende Aktionen aus:

Tabelle 12.12: Durchführen eines Upgrades einer BlackBerry-MDS-Studio-Anwendung – weitere Aktionen

Aktion	Vorgehensweise
Festlegen der Anzahl der Geräte, an die die Upgrade-Anfrage für MDS-Studio-Anwendungen gleichzeitig gesendet werden soll	Geben Sie im Feld <i>Group size for pushing</i> (Gruppengröße zum Übermitteln) eine Menge ein.
Festlegen des Intervalls (in Minuten), in dem die Upgrade-Anfrage für MDS-Studio-Anwendungen an die Geräte gesendet werden soll	Geben Sie im Feld <i>Push interval</i> (minute) (Übermittlungsintervall in Minuten) einen Wert ein.

Aktion	Vorgehensweise
Festlegen eines bestimmten Zeitpunkts, zu dem die Upgrade-Anfrage für BlackBerry-MDS-Studio-Anwendungen an die Geräte gesendet werden soll	1. Aktivieren Sie das Kontrollkästchen <i>Schedule</i> (Zeitplan). 2. Klicken Sie in der Dropdown-Liste <i>Start at</i> (Beginnen um) auf ein Datum. 3. Legen Sie den Startzeitpunkt fest. 4. Klicken Sie in der Dropdown-Liste <i>Expire at</i> (Gültig bis) auf ein Datum. 5. Legen Sie den Ablaufzeitpunkt fest. Hinweis: Wenn Sie keinen Startzeitpunkt festlegen, sendet BlackBerry MDS Services die BlackBerry-MDS-Studio-Anwendung sofort.
Konfigurieren des stillen Upgrades der BlackBerry-MDS-Studio-Anwendung auf den angegebenen BlackBerry-Geräten	Klicken Sie auf <i>Required</i> (Erforderlich). Hinweis: Wenn Sie die BlackBerry-MDS-Studio-Anwendung nicht still auf dem BlackBerry-Gerät aktualisieren, wird der Benutzer aufgefordert, die BlackBerry-MDS-Studio-Anwendung zu installieren.

8. Klicken Sie auf *Next* (Weiter).

9. Klicken Sie auf *Finish* (Fertigstellen).

12.7.12 Entfernen von MDS-Studio-Anwendungen aus der Datenbank und von den Geräten

Entwickler veröffentlichen MDS-Studio-Anwendungen in der Datenbank. Sie können die MDS-Studio-Anwendungen in der Datenbank verwalten. In der Datenbank können mehrere Versionen einer MDS-Studio-Anwendung veröffentlicht werden.

Je nach Ihrer Administratorrolle können Sie MDS-Studio-Anwendungen aus der Datenbank und von Geräten entfernen.

Wenn Sie eine MDS-Studio-Anwendung aus der Datenbank entfernen, funktioniert sie weiterhin auf den Geräten, auf denen die MDS-Studio-Anwendung installiert ist.

Wenn die Benutzer eine zuvor installierte MDS-Studio-Anwendung nicht verwenden sollen, entfernen Sie die MDS-Studio-Anwendung aus der Datenbank und deinstallieren Sie sie dann auf den Geräten.

12.7.13 Entfernen einer MDS-Studio-Anwendung aus der Datenbank

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen BlackBerry MDS Services Server.
2. Klicken Sie auf *Application Registry* (Anwendungsregistrierung).

3. Klicken Sie auf der Registerkarte *Application Registry* (Anwendungsregistrierung) auf die zu entfernende MDS-Studio-Anwendung.
4. Klicken Sie auf *Application Management* (Anwendungsverwaltung).
5. Klicken Sie auf *Delete Application* (Anwendung löschen).
6. Klicken Sie auf *Yes* (Ja).

12.7.14 Entfernen einer BlackBerry-MDS-Studio-Anwendung von einem Gerät

1. Führen Sie im BlackBerry Manager im linken Fensterbereich eine der folgenden Aktionen aus:

Tabelle 12.13: Entfernen einer BlackBerry-MDS-Studio-Anwendung

Aktion	Vorgehensweise
Entfernen einer MDS-Studio-Anwendung von Geräten für eine Gruppe von Benutzerkonten, die dieselben MDS Services verwenden	1. Klicken Sie auf eine Benutzergruppe. 2. Klicken Sie auf der Registerkarte <i>Users</i> (Benutzer) mit der rechten Maustaste auf eine Spaltenüberschrift. 3. Klicken Sie in der Liste <i>Available columns</i> (Vorhandene Spalten) auf <i>MDS Services Server URL</i> (MDS Services-Server-URL). 4. Klicken Sie auf <i>Insert</i> (Einfügen). 5. Klicken Sie auf <i>OK</i>. 6. Klicken Sie auf die Spaltenüberschrift <i>MDS Services Server URL</i> (MDS Services-Server-URL), damit die Benutzer nach MDS Services Server sortiert werden. 7. Klicken Sie auf die Benutzer, die mit demselben MDS Services Server verbunden sind. 8. Klicken Sie auf der Registerkarte <i>Group Configuration</i> (Gruppenkonfiguration) auf <i>MDS Services</i> (MDS-Dienste). 9. Klicken Sie auf <i>Uninstall on Device</i> (Auf Gerät deinstallieren). 10. Klicken Sie auf die zu entfernende MDS-Studio-Anwendung.

Aktion	Vorgehensweise
Entfernen einer BlackBerry-MDS-Studio-Anwendung von einem einzigen Gerät	1. Klicken Sie auf einen MDS Services Server. 2. Klicken Sie auf <i>Applications Installed</i> (Installierte Anwendungen). 3. Klicken Sie auf der Registerkarte <i>Applications Installed</i> (Installierte Anwendungen) auf die vom Gerät zu entfernende MDS-Studio-Anwendung. 4. Klicken Sie auf <i>Device Management</i> (Geräteverwaltung). 5. Klicken Sie auf <i>Uninstall on Device</i> (Auf Gerät deinstallieren). 6. Klicken Sie in der Dropdown-Liste <i>Uninstall application on devices</i> (Anwendung auf Geräten deinstallieren) auf <i>without application installed</i> (Ohne installierte Anwendung). 7. Deaktivieren Sie das Kontrollkästchen <i>Select all</i> (Alle auswählen). 8. Klicken Sie auf die PIN des Geräts, von dem die MDS-Studio-Anwendung entfernt werden soll.

2. Klicken Sie auf *Next* (Weiter).

3. Führen Sie folgende Aktionen aus:

Tabelle 12.14: Entfernen einer BlackBerry MDS-Studio-Anwendung – weitere Aktionen

Aktion	Vorgehensweise
Festlegen der Anzahl der Geräte, an die die Löschanfrage für MDS-Studio-Anwendungen gleichzeitig gesendet werden soll	Geben Sie im Feld <i>Group size for pushing</i> (Gruppengröße zum Übermitteln) eine Anzahl ein.
Festlegen des Intervalls (in Minuten), in dem die Löschanfrage für BlackBerry-MDS Studio-Anwendungen an BlackBerry-Geräte gesendet werden soll	Geben Sie im Feld <i>Push interval (minute)</i> (Übermittlungsintervall in Minuten) einen Wert ein.
Festlegen eines bestimmten Zeitpunkts, zu dem die Löschanfrage für MDS-Studio-Anwendungen an Geräte gesendet werden soll	1. Aktivieren Sie das Kontrollkästchen <i>Schedule</i> (Zeitplan). 2. Klicken Sie in der Dropdown-Liste <i>Start at</i> (Beginnen um) auf ein Datum. 3. Legen Sie den Startzeitpunkt fest. 4. Klicken Sie in der Dropdown-Liste <i>Expire at</i> (Gültig bis) auf ein Datum. 5. Legen Sie den Ablaufzeitpunkt fest. Hinweis: Wenn Sie keinen Startzeitpunkt festlegen, sendet BlackBerry MDS Services die Löschanfrage sofort.

4. Klicken Sie auf *Next* (Weiter).

5. Klicken Sie auf *Finish* (Fertigstellen).

12.7.15 Überwachen von BlackBerry-MDS-Services-Nachrichten

Sie können den Nachrichtenverkehr zwischen MDS Services und Geräten sowie den von MDS-Studio-Anwendungen erzeugten Nachrichtenverkehr überwachen.

Der BlackBerry Manager zeigt überwachte Nachrichten an. Eine übermäßig große Anzahl an Nachrichten von einer bestimmten Anwendung oder Nachrichten einer bestimmten Art lassen unter Umständen auf ein Problem bei einem Gerät, einer MDS-Studio-Anwendung oder den Webdiensten schließen.

Erstellen Sie Filter, um Nachrichten, die zu häufig von Webdienst-Hosts versendet werden, zu blockieren. Wenn Sie einen Filter für einen bestimmten Host erstellen, verarbeitet MDS-Services die Nachrichten von diesem Host nicht und sendet sie auch nicht an BlackBerry-Geräte.

12.7.16 Einrichten der Nachrichtenüberwachung für BlackBerry-MDS Studio-Anwendungen

Wenn Sie MDS-Services neu starten, müssen Sie die Nachrichtenüberwachung erneut einrichten.

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen BlackBerry MDS Services Server.
2. Klicken Sie auf der Registerkarte *MDS Services* (MDS-Dienste) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Message Monitors* (Nachrichtenüberwachung).
4. Doppelklicken Sie auf *BlackBerry MDS Services Monitor Definition* (Überwachungsdefinition von BlackBerry MDS Services).
5. Klicken Sie auf *New* (Neu).
6. Führen Sie eine der folgenden Aktionen aus:

Tabelle 12.15: Einrichten der Nachrichtenüberwachung für BlackBerry-MDS

Aktion	Vorgehensweise
Überwachen der Nachrichten an ein bzw. von einem BlackBerry-Gerät	Geben Sie im Feld <i>PIN</i> die PIN des zu überwachenden BlackBerry-Geräts ein. Hinweis: Wenn Sie mehrere BlackBerry-Geräte überwachen, trennen Sie die PINs durch Kommas.
Überwachen der von einer BlackBerry-MDS-Studio-Anwendung erzeugten Nachrichten	Klicken Sie in der Dropdown-Liste <i>Application</i> (Anwendung) auf den Namen und die Version der BlackBerry-MDS-Studio-Anwendung.

7. Klicken Sie auf *OK*.
8. Klicken Sie erneut auf *OK*.

12.7.17 Anzeigen von Nachrichten der Anwendung

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen MDS Services Server.
2. Klicken Sie auf *Monitor Messages* (Nachrichten überwachen).
3. Führen Sie auf der Registerkarte *Monitor Messages* (Nachrichten überwachen) eine der folgenden Aktionen aus:

Tabelle 12.16: Anzeigen von Nachrichten der BlackBerry-MDS

Aktion	Vorgehensweise
Anzeigen aller Nachrichten, die an bzw. von einem bestimmten BlackBerry-Gerät gesendet werden	Geben Sie im Feld <i>Device</i> (Gerät) die PIN ein.
Anzeigen aller Nachrichten, die an eine bzw. von einer bestimmten BlackBerry-MDS-Studio-Anwendung gesendet werden	Klicken Sie in der Dropdown-Liste <i>Application</i> (Anwendung) auf den Namen der BlackBerry-MDS-Studio-Anwendung.
Filtern angezeigter Nachrichten für ein bestimmtes BlackBerry-Gerät oder eine bestimmte BlackBerry-MDS-Studio-Anwendung in der Nachrichtenliste nach Datum und Zeit	1. Klicken Sie in der Dropdown-Liste <i>Start time</i> (Startzeit) auf ein Datum. 2. Klicken Sie im Zeitfeld auf die Zahlen, und legen Sie die Zeit mit den Pfeiltasten in Stunden, Minuten und Sekunden fest. 3. Klicken Sie auf <i>End time</i> (Endzeit), um ein Datum und eine Uhrzeit festzulegen, wonach keine Nachrichten mehr angezeigt werden.

4. Klicken Sie auf *Search* (Suchen).

12.7.18 Entfernen aller überwachten Nachrichten vom MDS Services Server

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen MDS Services Server.
2. Klicken Sie auf *Monitor Messages* (Nachrichten überwachen).
3. Klicken Sie auf der Registerkarte *Monitor Messages* (Nachrichten überwachen) auf *Purge Messages* (Nachrichten löschen).

12.7.19 Filtern der Kommunikation von einem Webdienst-Host

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen BlackBerry MDS Services Server.

2. Klicken Sie auf der Registerkarte *MDS Services* (MDS-Dienste) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Filters* (Filter).
4. Doppelklicken Sie auf *BlackBerry MDS Services Filter Definition* (Filterdefinition von BlackBerry MDS Services).
5. Führen Sie eine der folgenden Aktionen aus:

Tabelle 12.17: Filtern der Kommunikation von einem Webdienst-Host

Aktion	Vorgehensweise
Blockieren der Kommunikation eines Webdienst-Hosts	1. Klicken Sie auf <i>New</i> (Neu). 2. Geben Sie im Feld <i>Host/Address</i> (Host/Adresse) den vollständigen URL und die Domäne für den Webdienst-Host ein. Beispiel: <i><Host- Name>. <Domäne></i>. 3. Klicken Sie auf <i>OK</i>.
Zulassen der zuvor blockierten Kommunikation von einem Webdienst-Host	1. Klicken Sie auf einen Filter. 2. Klicken Sie auf <i>Remove</i> (Entfernen).

6. Klicken Sie auf *OK*.

12.7.20 Festlegen der Verbindung zwischen MDS Services und dem MDS Connection Service

Wenn Sie MDS Services einen MDS Connection Service hinzufügen, muss für den MDS Connection Service ein vollständiger Domänenname oder eine IP-Adresse angegeben werden.

Für den MDS Connection Service kann nicht der lokale Host oder 127.0.0.1 verwendet werden.

Hinweis: Wenn Sie einen entfernten MDS Connection Service installieren und dieser MDS Connection Service für BlackBerry MDS Services nicht als verfügbar angezeigt wird, können Sie ihn der Liste der für MDS Services verfügbaren MDS Connection Services hinzufügen.

Wenn für entfernte MDS Connection Services ein Proxy-Server verwendet wird, sollten Sie MDS Services vom Enterprise Server entfernen und MDS Services anschließend dem Enterprise Server neu zuweisen.

Auf diese Weise wird der entfernte MDS Connection Service automatisch MDS Services zugeordnet, und die direkte Proxy-Zuordnung zwischen dem MDS Connection Service und MDS Services ist vorhanden. Weitere Informationen finden Sie im Kapitel 9.6.10 »Zuweisen von BlackBerry MDS Services zu mehreren Enterprise Servern« .

- Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen MDS Services Server.
- Klicken Sie auf der Registerkarte *MDS Services* (MDS-Dienste) auf *Edit Properties* (Eigenschaften bearbeiten).
- Klicken Sie auf *Connection Service* (Verbindungsdienst).
- Doppelklicken Sie auf BlackBerry *MDS Connection Service Definition* (Definition des BlackBerry MDS Connection Service).
- Führen Sie eine der folgenden Aktionen aus:

Tabelle 12.18: Festlegen der Verbindung zwischen BlackBerry MDS Services und dem BlackBerry MDS Connection Service

Aktion	Vorgehensweise
Hinzufügen eines neuen MDS Connection Service zur Liste der für MDS Services verfügbaren Verbindungsdienste	1. Klicken Sie auf <i>New</i> (Neu). 2. Doppelklicken Sie auf <i>URL</i> (URL). 3. Geben Sie den vollständigen URL oder den Domänennamen und die Port-Nummer für den Verbindungsdienst ein. 4. Klicken Sie auf <i>OK</i>. 5. Klicken Sie erneut auf <i>OK</i>.
Entfernen eines MDS Connection Service aus der Liste der für BlackBerry MDS Services verfügbaren Verbindungsdienste	1. Klicken Sie auf den URL eines Verbindungsdienstes. 2. Klicken Sie auf <i>Remove</i> (Entfernen). 3. Klicken Sie auf <i>OK</i>.

13 Anpassen der BlackBerry-Nachrichtenübermittlung

Sie können steuern, welche Nachrichten der Enterprise Server wann an die BlackBerry-Geräte umleitet. Durch das Verwalten der Nachrichtenumleitung können Sie die Länge der Nachrichtenwarteschlange und die Belastung des Messaging Agent steuern, der die Umleitungsanfragen verarbeitet.

Die Nachrichtenumleitung wird standardmäßig aktiviert, wenn Benutzerkonten erstmalig dem Enterprise Server hinzugefügt werden.

13.1 Verwalten der Nachrichtenumleitung

Definieren Sie, ob der Enterprise Server Nachrichten, für die keine Filter gelten, zurückhält oder weiterleitet. Sie können den Posteingang und die Unterordner angeben, woraus der Enterprise Server Nachrichten umleitet werden kann.

Sie können die Nachrichtenumleitung an ein Gerät auch vorübergehend aussetzen. Dies ist beispielsweise dann nützlich, wenn sich ein Benutzer zeitweilig in einem Gebiet ohne Mobilfunkabdeckung aufhält und nicht möchte, dass der BlackBerry Enterprise Server während dieser Zeit Nachrichten an das Gerät weiterleitet.

Wenn Sie die Nachrichtenumleitung für ein Benutzerkonto deaktivieren, kann der Benutzer Nachrichten senden, aber nicht empfangen. Der Benutzer kann die Umleitung auf dem Gerät manuell erneut aktivieren.

13.1.1 Ablauf der Nachrichtenumleitung

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf die Registerkarte *Users* (Benutzer).
3. Führen Sie eine der folgenden Aktionen aus:

Tabelle 13.1: Verwalten der Nachrichtenumleitung

Aktion	Vorgehensweise
Weiterleiten von Nachrichten an das BlackBerry-Gerät, wenn keine Filterregeln gelten	1. Doppelklicken Sie auf ein Benutzerkonto. 2. Klicken Sie im linken Fensterbereich auf <i>Filters</i> (Filter). 3. Klicken Sie im Bereich <i>Default Action</i> (Standardaktion) auf <i>Forward messages to BlackBerry device</i> (Nachrichten an BlackBerry-Gerät weiterleiten). 4. Klicken Sie in der Dropdown-Liste auf <i>True</i> (Wahr). 5. Klicken Sie auf <i>OK</i>.
Umleiten von Nachrichten aus Unterordnern des Posteingangs an das BlackBerry-Gerät	1. Klicken Sie auf ein Benutzerkonto. 2. Klicken Sie auf <i>Service Access</i> (Dienstzugriff). 3. Klicken Sie auf <i>Choose Folders for Redirection</i> (Ordner für Umleitung auswählen). 4. Aktivieren Sie das Kontrollkästchen neben den Ordnern, aus denen Nachrichten umgeleitet werden sollen. 5. Klicken Sie auf <i>OK</i>.
Kein Synchronisieren von gesendeten Nachrichten zwischen dem BlackBerry-Gerät und der Desktop-E-Mail-Anwendung	1. Doppelklicken Sie auf ein Benutzerkonto. 2. Klicken Sie auf <i>Redirection</i> (Umleitung). 3. Klicken Sie auf <i>Do Not Save Sent Messages</i> (Gesendete Nachrichten nicht speichern). 4. Klicken Sie in der Dropdown-Liste auf <i>True</i> (Wahr). 5. Klicken Sie auf <i>OK</i>.
Deaktivieren der Nachrichtenumleitung	1. Klicken Sie auf ein Benutzerkonto. 2. Klicken Sie auf <i>Service Access</i> (Dienstzugriff). 3. Klicken Sie auf <i>Disable Redirection</i> (Umleitung deaktivieren).

13.2 Verwalten von Umleitungsfiltern

Umleitungsfilter definieren, welche Nachrichten der Enterprise Server an BlackBerry-Geräte umleitet.

Wenn ein Benutzer eine Nachricht erhält, wendet der Enterprise Server Filter an, um zu bestimmen, wie die Nachricht an das Gerät weitergeleitet werden soll: normal, vorrangig oder gar nicht.

Filter, die auf dem Enterprise Server eingerichtet werden, haben Vorrang vor Filtern, die Benutzer mithilfe des BlackBerry-Desktop-Manager festlegen.

Auf dem Enterprise Server können zwei Arten von Filtern erstellt werden:

- Globale Filter gelten für alle Benutzerkonten auf dem Enterprise Server.
- Benutzerfilter gelten für bestimmte Benutzerkonten.

Benutzer können keine globalen Filter anzeigen. Wenn Sie globale Filter festlegen, informieren Sie die Benutzer darüber, damit sie verstehen, weshalb einige ihrer Filterregeln unter Umständen nicht auf eingehende Nachrichten angewendet werden.

Wenn Sie globale Filter ändern, werden die Änderungen sofort vom Enterprise Server gelesen.

13.2.1 Erstellen eines globalen Filters

Der Enterprise Server wendet Filter der Reihe nach auf Nachrichten an.

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen BlackBerry Enterprise Server.
2. Klicken Sie auf der Registerkarte *Server Configuration* (Serverkonfiguration) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Global Filters* (Globale Filter).
4. Doppelklicken Sie auf *Global Filter Definition* (Definition globaler Filter).
5. Klicken Sie auf *New* (Neu).
6. Doppelklicken Sie im Bereich *New Message Conditions* (Neue Nachrichtenbedingungen) auf *Filter Name* (Filtername).
7. Geben Sie für den neuen Filter einen Namen ein.
8. Legen Sie die Filteroptionen fest.
9. Klicken Sie auf *Action* (Aktion).
10. Führen Sie eine der folgenden Aktionen aus:

Tabelle 13.2: Erstellen eines globalen Filters

Aktion	Vorgehensweise
Zurückhalten von Nachrichten, für die keine Filter gelten	Klicken Sie in der Dropdown-Liste auf <i>Hold</i> (Zurückhalten).

Aktion	Vorgehensweise
Weiterleiten von Nachrichten, für die keine Filter gelten	1. Klicken Sie in der Dropdown-Liste auf <i>Forward</i> (Weiterleiten). 2. Doppelklicken Sie auf <i>Forwarding Options</i> (Weiterleitungsoptionen). 3. Führen Sie eine der folgenden Aktionen aus: • Aktivieren Sie das Kontrollkästchen <i>Header Only</i> (Nur Kopfzeile), damit nur die Nachrichtenkopfzeilen an das Gerät weitergeleitet werden. • Aktivieren Sie das Kontrollkästchen <i>Level1 Notification</i> (Stufe-1-Benachrichtigung), damit Nachrichten der Stufe 1 an das Gerät weitergeleitet werden. • Aktivieren Sie sowohl <i>Header Only</i> (Nur Kopfzeile) als auch <i>Level1 Notification</i> (Stufe-1-Benachrichtigung), damit die Kopfzeilen von Nachrichten der Stufe 1 an das Gerät weitergeleitet werden.

11. Klicken Sie auf *OK*.

13.2.2 Erstellen eines Benutzerfilters

Der Enterprise Server wendet Filter der Reihe nach auf Nachrichten an.

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Doppelklicken Sie auf der Registerkarte *Users* (Benutzer) auf ein Benutzerkonto.
3. Klicken Sie auf *Filters* (Filter).
4. Doppelklicken Sie auf *Filter Rules* (Filterregeln).
5. Klicken Sie auf *New* (Neu).
6. Doppelklicken Sie im Bereich *New Message Conditions* (Neue Nachrichtenbedingungen) auf *Filter Name* (Filtername).
7. Geben Sie für den neuen Filter einen Namen ein.
8. Legen Sie die Filteroptionen fest.
9. Klicken Sie auf *Action* (Aktion).

Führen Sie eine der folgenden Aktionen aus:

Tabelle 13.3: Erstellen eines Benutzerfilters

Aktion	Vorgehensweise
Zurückhalten von Nachrichten, für die keine Filter gelten	Klicken Sie in der Dropdown-Liste auf <i>Hold</i> (Zurückhalten).
Weiterleiten von Nachrichten, für die keine Filter gelten	1. Klicken Sie in der Dropdown-Liste auf <i>Forward</i> (Weiterleiten). 2. Doppelklicken Sie auf <i>Forwarding Options</i> (Weiterleitungsoptionen). 3. Führen Sie eine der folgenden Aktionen aus: • Aktivieren Sie das Kontrollkästchen <i>Header Only</i> (Nur Kopfzeile), damit nur die Nachrichtenkopfzeilen an das Gerät weitergeleitet werden. • Aktivieren Sie das Kontrollkästchen <i>Level1 Notification</i> (Stufe-1-Benachrichtigung), damit Nachrichten der Stufe 1 an das Gerät weitergeleitet werden. • Aktivieren Sie sowohl <i>Header Only</i> (Nur Kopfzeile) als auch <i>Level1 Notification</i> (Stufe-1-Benachrichtigung), damit die Kopfzeilen von Nachrichten der Stufe 1 an das Gerät weitergeleitet werden.

10. Klicken Sie auf *OK*.

13.2.3 Verwalten eines globalen Filters

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Server Configuration* (Serverkonfiguration) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie im linken Fensterbereich auf *Global Filters* (Globale Filter).
4. Doppelklicken Sie auf *Global Filter Definition* (Definition globaler Filter).
5. Klicken Sie in der Liste *Filter Name* (Filtername) auf einen Filter.
6. Führen Sie eine der folgenden Aktionen aus:

Tabelle 13.4: Verwalten eines globalen Filters

Aktion	Vorgehensweise
Aktivieren eines Filters	1. Klicken Sie auf <i>Properties</i> (Eigenschaften). 2. Klicken Sie auf <i>Enabled</i> (Aktiviert). 3. Klicken Sie in der Dropdown-Liste auf <i>True</i> (Wahr).

Aktion	Vorgehensweise
Bearbeiten eines Filters	1. Klicken Sie auf <i>Properties</i> (Eigenschaften). 2. Klicken Sie auf <i>Edit</i> (Bearbeiten). 3. Ändern Sie die gewünschten Einstellungen. 4. Klicken Sie auf <i>OK</i>.
Ändern der Filterreihenfolge	1. Klicken Sie auf <i>Move Up</i> (Nach oben verschieben) oder auf <i>Move Down</i> (Nach unten verschieben), um den Filter in der Liste weiter nach oben oder nach unten zu verschieben. 2. Klicken Sie auf <i>OK</i>. Hinweis: Filter werden vom Enterprise Server in der angezeigten Reihenfolge auf neue Nachrichten angewendet. Sorgen Sie dafür, dass zuerst die am wenigsten restriktiven Filter und zuletzt die restriktivsten Filter angezeigt werden.
Löschen eines Filters	Klicken Sie auf <i>Remove</i> (Entfernen).

7. Klicken Sie auf *OK*.

13.2.4 Verwalten eines Benutzerfilters

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Doppelklicken Sie auf der Registerkarte *Users* (Benutzer) auf ein Benutzerkonto.
3. Klicken Sie im linken Fensterbereich auf *Filters* (Filter).
4. Doppelklicken Sie auf *Filter Rules* (Filterregeln).
5. Klicken Sie in der Liste *Filter Name* (Filtername) auf einen Filter.
6. Führen Sie eine der folgenden Aktionen aus:

Tabelle 13.5: Verwalten eines Benutzerfilters

Aktion	Vorgehensweise
Aktivieren eines Filters	1. Klicken Sie auf <i>Properties</i> (Eigenschaften). 2. Klicken Sie auf <i>Enabled</i> (Aktiviert). 3. Klicken Sie in der Dropdown-Liste auf <i>True</i> (Wahr).
Bearbeiten eines Filters	1. Klicken Sie auf <i>Properties</i> (Eigenschaften). 2. Klicken Sie auf <i>Edit</i> (Bearbeiten). 3. Ändern Sie die gewünschten Einstellungen. 4. Klicken Sie auf <i>OK</i>.

Aktion	Vorgehensweise
Ändern der Filterreihenfolge	1. Klicken Sie auf <i>Move Up</i> (Nach oben verschieben) oder auf <i>Move Down</i> (Nach unten verschieben), um den Filter in der Liste weiter nach oben oder nach unten zu verschieben. 2. Klicken Sie auf <i>OK</i> . Hinweis: Filter werden vom Enterprise Server in der angezeigten Reihenfolge auf neue Nachrichten angewendet. Sorgen Sie dafür, dass zuerst die am wenigsten restriktiven Filter und zuletzt die restriktivsten Filter angezeigt werden.
Löschen eines Filters	Klicken Sie auf <i>Remove</i> (Entfernen).

7. Klicken Sie auf *OK*.

14 Verwalten der drahtlosen Nachrichtensynchronisierung

Mit der drahtlosen Nachrichtensynchronisierung können Benutzer ihre Nachrichten auf dem Enterprise Server verwalten. Dadurch werden Veränderungen des Nachrichtenstatus zwischen dem Gerät und der Desktop-E-Mail-Anwendung automatisch synchronisiert.

Auf dem Enterprise Server ist die drahtlose Nachrichtensynchronisierung standardmäßig aktiviert und wird alle 15 Minuten ausgeführt.

Deaktivieren Sie die drahtlose Nachrichtensynchronisierung, wenn Nachrichten nur dann synchronisiert werden sollen, wenn Benutzer ihr Gerät an den Desktop-Manager anschließen.

Mit der Funktion zum vollständigen Löschen können Benutzer über das drahtlose Netzwerk eine Nachricht von Geräten entfernen, wenn sie eine Nachricht aus ihrer Mailbox durch Drücken der **Umschalttaste** + **Entf** endgültig löschen, eine Nachricht in einen persönlichen Ordner verschieben oder in Outlook archivieren.

Aktivieren Sie die Funktion zum vollständigen Löschen, wenn Benutzer in der Lage sein sollen, Nachrichten auf den Geräten durch das Synchronisieren endgültig gelöschter Nachrichten zwischen der Desktop-E-Mail-Anwendung und BlackBerry-Geräten zu verwalten. Der Enterprise Server entfernt endgültig gelöschte Nachrichten in 15-Minuten-Intervallen von den Geräten.

14.1 Deaktivieren der drahtlosen Nachrichtensynchronisierung

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Server Configuration* (Serverkonfiguration) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Messaging* (Nachrichtenübermittlung).
4. Klicken Sie im Bereich *Messaging Options* (Nachrichtenübermittlungsoptionen) auf *Wireless Message Reconciliation Enabled* (Drahtlose Nachrichtensynchronisierung aktiviert).
5. Klicken Sie in der Dropdown-Liste auf *False* (Falsch).
6. Klicken Sie auf *OK*.

14.2 Synchronisieren endgültig gelöschter Nachrichten

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Messaging* (Nachrichtenübermittlung).
4. Klicken Sie im Bereich *Messaging Options* (Nachrichtenübermittlungsoptionen) auf *Hard Deletes Reconciliation* (Synchronisierung vollständig gelöschter Nachrichten).
5. Klicken Sie in der Dropdown-Liste auf *True* (Wahr).
6. Klicken Sie auf *OK*.
7. Starten Sie auf dem Computer, auf dem der BlackBerry-Dispatcher installiert ist, den Dispatcher unter den Windows-Diensten erneut.

15 Sicherer Nachrichtentransfer

Mithilfe einer Nachrichtenklassifikation können Sie für S/MIME oder PGP aktivierte Benutzer zwingen, E-Mail-Nachrichten von ihrem BlackBerry-Gerät zu signieren und/oder zu verschlüsseln.

Konfigurieren Sie mithilfe der IT-Richtlinienregel für die Nachrichtenklassifikation eine oder mehrere Klassifikationen, die Benutzer auf E-Mail-Nachrichten anwenden können, die sie von ihrem Gerät aus senden.

Auf dem Enterprise Server können Benutzer durch S/MIME oder PGP geschützte Nachrichtenverschlüsselungstypen auf die Klassifikationstypen anwenden, die sie beim Erstellen einer Nachricht ausgewählt haben.

Falls ein Benutzer keine Klassifikation wählt, wendet das BlackBerry-Gerät standardmäßig die erste Klassifikation in der Liste an. Sie können die Reihenfolge ändern, in der die Klassifikationen vom Gerät aufgelistet werden.

Die Verschlüsselungsoptionen auf dem Gerät sind auf die Verschlüsselungstypen begrenzt, die laut den sicheren, auf dem BlackBerry-Gerät installierten und aktivierten Nachrichtenübermittlungspaketen zulässig sind.

Wenn ein Benutzer eine Klassifikation auf eine Nachricht auf dem Gerät anwendet, muss er zusätzlich einen Verschlüsselungstyp auswählen, der von dieser Nachrichtenklassifikation zugelassen wird, oder den Standardverschlüsselungstyp akzeptieren.

Wenn ein Benutzer einen Verschlüsselungstyp anwendet, für den die Signierung und/oder Verschlüsselung einer Nachricht erforderlich ist, und auf dem Gerät kein sicheres Nachrichtenübermittlungspaket installiert und aktiviert ist, kann der Benutzer die Nachricht nicht senden.

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Klicken Sie auf der Registerkarte *Global* (Global) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *IT Policy* (IT-Richtlinie).
4. Doppelklicken Sie im Bereich *IT Policy Administration* (IT-Richtlinienverwaltung) auf *IT Policies* (IT-Richtlinien).
5. Führen Sie eine der folgenden Aktionen aus:

Tabelle 15.1: Erzwingen der sicheren Nachrichtenübermittlung anhand von Klassifikationen

Aktion	Vorgehensweise
Erstellen einer Nachrichtenklassifikation	1. Klicken Sie in der IT-Richtlinienliste auf eine IT-Richtlinie. 2. Klicken Sie auf <i>Properties</i> (Eigenschaften). 3. Klicken Sie auf <i>Security Policy Group</i> (Sicherheitsrichtliniengruppe). 4. Doppelklicken Sie auf die IT-Richtlinienregel <i>Message Classification</i> (Nachrichtenklassifikation). 5. Klicken Sie auf <i>New</i> (Neu). 6. Geben Sie einen Namen für die Anzeige in der Klassifikationsliste auf dem BlackBerry-Gerät ein. 7. Geben Sie in Klammern ein Betreff-Suffix ein, das an den Betreff der Nachricht angehängt wird (z. B. »(N)« für eine Klassifikation namens »Nicht klassifiziert«). 8. Klicken Sie in der Dropdown-Liste auf die Mindestaktion für die Nachrichtenverschlüsselung (z. B. auf <i>Signed</i> (Signiert), damit der Benutzer den Verschlüsselungstyp <i>Sign</i> (Signieren), <i>Encrypt</i> (Verschlüsseln) oder <i>Sign and Encrypt</i> (Signieren und verschlüsseln) für die sicheren Nachrichtenübermittlungspakete auf dem Gerät auswählen kann). 9. Klicken Sie auf <i>Apply</i> (Anwenden). 10. Klicken Sie auf <i>OK</i>.
Erstellen einer Nachrichtenklassifikation unter Verwendung einer bereits vorhandenen Klassifikation	1. Klicken Sie in der IT-Richtlinienliste auf eine IT-Richtlinie. 2. Klicken Sie auf <i>Properties</i> (Eigenschaften). 3. Klicken Sie auf <i>Security Policy Group</i> (Sicherheitsrichtliniengruppe). 4. Doppelklicken Sie auf die IT-Richtlinienregel <i>Message Classification</i> (Nachrichtenklassifikation). 5. Klicken Sie auf einen Anzeigenamen. 6. Klicken Sie auf <i>New Copy</i> (Neue Ausgabe). 7. Geben Sie einen neuen Anzeigenamen ein. 8. Geben Sie ein neues Betreff-Suffix ein. 9. Wählen Sie in der Dropdown-Liste eine Mindestaktion zum Verschlüsseln der Nachricht aus. 10. Klicken Sie auf <i>Apply</i> (Anwenden). 11. Klicken Sie auf <i>OK</i>.

Aktion	Vorgehensweise
Sortieren von Nachrichtenklassifikationen	1. Klicken Sie in der IT-Richtlinienliste auf eine IT-Richtlinie. 2. Klicken Sie auf <i>Properties</i> (Eigenschaften). 3. Klicken Sie auf <i>Security Policy Group</i> (Sicherheitsrichtlinien-gruppe). 4. Doppelklicken Sie auf die IT-Richtlinienregel <i>Message Classification</i> (Nachrichtenklassifikation). 5. Klicken Sie auf einen Anzeigenamen. 6. Führen Sie eine der folgenden Aktionen aus: • Klicken Sie auf <i>Make First</i> (An Anfang), um die ausgewählte Klassifikation an den Anfang der Liste zu verschieben. • Klicken Sie auf <i>Move Up</i> (Nach oben verschieben), um die ausgewählte Klassifikation in der Liste eine Position nach oben zu verschieben. • Klicken Sie auf <i>Move down</i> (Nach unten verschieben), um die ausgewählte Klassifikation in der Liste eine Position nach unten zu verschieben. • Klicken Sie auf <i>Make Last</i> (Ans Ende), um die ausgewählte Klassifikation an das Ende der Liste zu verschieben. 7. Klicken Sie auf <i>Apply</i> (Anwenden).
Entfernen einer Nachrichtenklassifikation	1. Klicken Sie in der IT-Richtlinienliste auf eine IT-Richtlinie. 2. Klicken Sie auf <i>Properties</i> (Eigenschaften). 3. Klicken Sie auf <i>Security Policy Group</i> (Sicherheitsrichtlinien-gruppe). 4. Doppelklicken Sie auf die IT-Richtlinienregel <i>Message Classification</i> (Nachrichtenklassifikation). 5. Klicken Sie auf einen Anzeigenamen. 6. Klicken Sie auf <i>Remove</i> (Entfernen). 7. Klicken Sie auf <i>Apply</i> (Anwenden).

6. Klicken Sie auf *OK*.

15.1 Signaturen und Haftungsausschlüsse in Nachrichten verwenden

Sie können einen Standardhaftungsausschluss oder einen anderen Text so hinzufügen, dass er in sämtlichen Nachrichten, die Benutzer eines Enterprise Servers von BlackBerry-Geräten aus senden, vor dem Nachrichtentext oder nach der Benutzersignatur angezeigt wird. Sie können Haftungsausschlüsse für einen Benutzer oder eine Gruppe

von Benutzern festlegen. Haftungsausschlüsse können nicht von Benutzern geändert werden.

Wenn Sie separat für einen Benutzer und für alle Benutzer auf einem Enterprise Server Haftungsausschlüsse festlegen, können Sie durch das Festlegen von Konfliktregeln steuern, wie der Enterprise Server die Haftungsausschlüsse anwendet.

Sie können auch eine Signatur für ein bestimmtes Benutzerkonto einstellen, die in sämtlichen Nachrichten angezeigt wird, die der Benutzer von dem Gerät aus sendet. Benutzer können ihre Signatur auf den Geräten oder im Desktop Manager ändern.

Wenn innerhalb des Unternehmens Signaturformatrichtlinien durchgesetzt werden sollen, fügen Sie die Signatur dem Unternehmenshaftungsausschluss hinzu.

15.1.1 Erstellen von Signaturen für Benutzerkonten

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Doppelklicken Sie auf der Registerkarte *Users* (Benutzer) auf ein Benutzerkonto.
3. Klicken Sie auf *Redirection* (Umleitung).
4. Geben Sie im Feld *Signature* (Signatur) die Signatur ein, die in Nachrichten angezeigt werden soll, die der Benutzer vom BlackBerry-Gerät aus sendet.
5. Klicken Sie auf *OK*.
6. Klicken Sie erneut auf *OK*.

15.1.2 Erstellen von vorangestellten Haftungsausschlüssen für Benutzerkonten

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Doppelklicken Sie auf der Registerkarte *Users* (Benutzer) auf ein Benutzerkonto.
3. Doppelklicken Sie auf *Prepended Disclaimer Text* (Vorangestellter Haftungsausschlusstext).
4. Geben Sie einen Haftungsausschluss ein.
5. Klicken Sie auf *OK*.
6. Klicken Sie erneut auf *OK*.

15.1.3 Erstellen von nachgestellten Haftungsausschlüssen für Benutzerkonten

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Doppelklicken Sie auf der Registerkarte *Users* (Benutzer) auf ein Benutzerkonto.
3. Doppelklicken Sie auf *Appended Disclaimer Text* (Nachgestellter Haftungsausschlusstext).
4. Geben Sie einen Haftungsausschluss ein.
5. Klicken Sie auf *OK*.
6. Klicken Sie erneut auf *OK*.

15.1.4 Erstellen von vorangestellten Haftungsausschlüssen für alle Benutzerkonten auf einem BlackBerry Enterprise Server

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Server Configuration* (Serverkonfiguration) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Messaging* (Nachrichtenübermittlung).
4. Doppelklicken Sie im Bereich *Messaging Options* (Nachrichtenübermittlungsoptionen) auf *Prepended Disclaimer Text* (Vorangestellter Haftungsausschlusstext).
5. Geben Sie einen Haftungsausschluss ein.
6. Klicken Sie auf *OK*.
7. Klicken Sie erneut auf *OK*.

15.1.5 Erstellen von nachgestellten Haftungsausschlüssen für alle Benutzerkonten auf einem BlackBerry Enterprise Server

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Server Configuration* (Serverkonfiguration) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Messaging* (Nachrichtenübermittlung).
4. Doppelklicken Sie im Bereich *Messaging Options* (Nachrichtenübermittlungsoptionen) auf *Appended Disclaimer Text* (Nachgestellter Haftungsausschlusstext).
5. Geben Sie einen Haftungsausschluss ein.

6. Klicken Sie auf OK.
7. Klicken Sie erneut auf OK.

15.1.6 Erstellen von vorangestellten Haftungsausschlüssen für Benutzergruppen

1. Klicken Sie im linken Fensterbereich des BlackBerry Managers auf *User Groups List* (Benutzergruppenliste).
2. Klicken Sie in der Liste *Group Name* (Gruppenname) auf eine Gruppe.
3. Klicken Sie auf *Edit Group Template* (Gruppenvorlage bearbeiten).
4. Doppelklicken Sie im Bereich *Messaging Options* (Nachrichtenübermittlungsoptionen) auf *Prepended Disclaimer Text* (Vorangestellter Haftungsausschlusstext).
5. Geben Sie einen Haftungsausschluss ein.
6. Aktivieren Sie das Kontrollkästchen links vom Feld, um den Haftungsausschluss auf alle der Gruppe zugewiesenen Benutzerkonten anzuwenden.
7. Klicken Sie auf *Reapply Template* (Vorlage erneut anwenden).
8. Klicken Sie auf *Yes* (Ja).
9. Klicken Sie auf OK.

15.1.7 Erstellen von nachgestellten Haftungsausschlüssen für Benutzergruppen

1. Klicken Sie im linken Fensterbereich des BlackBerry Managers auf *User Groups List* (Benutzergruppenliste).
2. Klicken Sie in der Liste *Group Name* (Gruppenname) auf eine Gruppe.
3. Klicken Sie auf *Edit Group Template* (Gruppenvorlage bearbeiten).
4. Doppelklicken Sie im Bereich *Messaging Options* (Nachrichtenübermittlungsoptionen) auf *Appended Disclaimer Text* (Nachgestellter Haftungsausschlusstext).
5. Geben Sie einen Haftungsausschluss ein.
6. Aktivieren Sie das Kontrollkästchen links vom Feld, um den Haftungsausschluss auf alle der Gruppe zugewiesenen Benutzerkonten anzuwenden.
7. Klicken Sie auf *Reapply Template* (Vorlage erneut anwenden).
8. Klicken Sie auf *Yes*.
9. Klicken Sie auf OK.

15.1.8 Festlegen von Konfliktregeln für vorangestellte Haftungsausschlüsse

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Server Configuration* (Serverkonfiguration) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Messaging* (Nachrichtenübermittlung).
4. Klicken Sie im Bereich *Messaging Options* (Nachrichtenübermittlungsoptionen) auf *Prepended Disclaimer Conflict Rule* (Konfliktregel für vorangestellte Haftungsausschlüsse).
5. Klicken Sie im Dropdown-Menü auf eine Regel.
6. Klicken Sie auf OK.

15.1.9 Festlegen von Konfliktregeln für nachgestellte Haftungsausschlüsse

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Server Configuration* (Serverkonfiguration) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Messaging* (Nachrichtenübermittlung).
4. Klicken Sie im Bereich *Messaging Options* (Nachrichtenübermittlungsoptionen) auf *Appended Disclaimer Conflict Rule* (Konfliktregel für nachgestellte Haftungsausschlüsse).
5. Klicken Sie im Dropdown-Menü auf eine Regel.
6. Klicken Sie auf OK.

15.2 Überwachen von Nachrichten, die Benutzer von ihren BlackBerry-Geräten senden

Wenn Sie gemäß den IT-Richtlinien Ihres Unternehmens verpflichtet sind, Nachrichten zu überwachen, legen Sie eine Adresse für einen nicht sichtbaren Kopienempfänger (BCC-Adresse) fest.

Dadurch wird eine Kopie aller Nachrichten aufbewahrt, die Benutzer von BlackBerry-Geräten senden.

Stellen Sie die BCC-Funktion auf dem Enterprise Server so ein, dass von sämtlichen Nachrichten, die Benutzer von ihren Geräten aus senden, eine Kopie an bestimmte nicht angezeigte Empfänger geht.

Bei Verwendung dieser Funktion werden Adressen in das BCC-Feld der ursprünglichen Nachricht eingetragen.

Dem Absender ist dadurch bewusst, dass von der Nachricht eine Blindkopie erstellt wird.

15.2.1 Senden einer Kopie aller Nachrichten an einen nicht angezeigten Empfänger

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Server Configuration* (Serverkonfiguration) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Messaging* (Nachrichtenübermittlung).
4. Doppelklicken Sie im Bereich *Messaging Options* (Nachrichtenübermittlungsoptionen) auf *Auto BCC Addresses* (Adressen für automatische Blindkopien).
5. Geben Sie die Adressen durch Strichpunkte (;) getrennt ein.
6. Klicken Sie auf OK.

15.3 Verwalten der Nachrichtenwarteschlange

Sie können Nachrichten aus der Nachrichtenwarteschlange entfernen, um Benutzerkonten zu pflegen, die über viele anstehende Nachrichten verfügen.

Wenn Sie anstehende Nachrichten aus der Nachrichtenwarteschlange löschen, vermeiden Sie, dass die Nachrichten vom Enterprise Server an das Gerät des Benutzers gesendet werden. Die Nachrichten werden weiterhin im Posteingang des Benutzers angezeigt.

15.3.1 Löschen anstehender Nachrichten aus der Nachrichtenwarteschlange

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Users* (Benutzer) auf ein Benutzerkonto.
3. Klicken Sie auf *Service Control & Customization* (Dienststeuerung und -anpassung).
4. Klicken Sie auf *Purge Pending Messages* (Anstehende Nachrichten löschen).
5. Klicken Sie auf OK.

Hinweis: Wenn für das Benutzerkonto die drahtlose Kalendersynchronisierung aktiviert ist, werden auch anstehende Kalendernachrichten gelöscht. Diese Nachrichten werden jedoch zu einem späteren Zeitpunkt erneut gesendet. Anstehende IT-Richtlinien oder IT-Verwaltungsbefehle werden nicht gelöscht.

16 Terminplanerdaten drahtlos sichern und wiederherstellen

Die automatische drahtlose Sicherung sichert die Einstellungen und Daten der Benutzerkonten vom Gerät auf dem Enterprise Server automatisch und ohne Benutzerbeteiligung.

Durch das drahtlose Sichern auf dem Enterprise Server können Sie Terminplanerdaten mit neuen BlackBerry-Geräten synchronisieren, ohne dadurch die Leistung des Nachrichtenservers zu beeinträchtigen. Weitere Informationen finden Sie im Anhang B »Drahtloses Sichern und Wiederherstellen«.

Wenn der Enterprise Server die Terminplanerdaten eines Benutzers nicht ordnungsgemäß vom Gerät in die BlackBerry Configuration Database schreibt, sind die bereits gesicherten Terminplanerdaten auf dem Enterprise Server unter Umständen beschädigt.

Löschen Sie die Terminplanerdaten vom Enterprise Server. Durch das Löschen der Terminplanerdaten wird das Gerät des Benutzers zu einer Synchronisierung über das drahtlose Netzwerk mit dem Enterprise Server gezwungen.

16.1 Löschen der Terminplanerdaten eines Benutzers vom Enterprise Server

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Users* (Benutzer) auf ein Benutzerkonto.
3. Klicken Sie auf *Service Control & Customization* (Dienststeuerung und -anpassung).
4. Klicken Sie auf *Clear PIM Sync Backup Data* (Sicherungsdaten der PIM-Synchronisierung löschen).
5. Klicken Sie auf *OK*.

16.2 Deaktivieren der drahtlosen Sicherung

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Doppelklicken Sie auf der Registerkarte *Users* (Benutzer) auf ein Benutzerkonto.
3. Klicken Sie auf *PIM Sync* (PIM-Synchronisierung).
4. Klicken Sie auf *Automatic Wireless Backup Enabled* (Automatische drahtlose Sicherung aktiviert).
5. Klicken Sie in der Dropdown-Liste auf *False* (Falsch).
6. Klicken Sie auf *OK*.

17 Adressbuchfelder synchronisieren

Die reibungslose Kommunikation zwischen E-Mail-Anwendungen auf dem PC und Blackberry-Geräten setzt ein gewisses Maß an Integration der beiden Welten voraus. Deshalb ist es möglich, die Adressbuchfelder der PC-Mailanwendung und der Blackberry-Endgeräte miteinander zu synchronisieren

17.1 Feldzuordnung

Sie können Felder im Adressbuch so zuordnen, dass die Felder eines Kontakts in der Desktop-E-Mail-Anwendung des Benutzers mit den Feldern synchronisiert werden, die Sie auf dem Endgerät des Benutzers festgelegt haben. Sie können zwei Arten von Feldzuordnungen auf dem Enterprise Server erstellen:

- Globale Feldzuordnungen gelten für alle Benutzerkonten in der BlackBerry-Domain.
- Benutzerfeldzuordnungen gelten für bestimmte Benutzerkonten.

Sie können maximal vier benutzerdefinierte Felder eines Kontakteintrags zu BlackBerry-Geräten zuordnen. Wenn Benutzer eine entfernte Adressensuche in der globalen Adressenliste (GAL) anfordern, werden die von Ihnen festgelegten Felder auf den Endgeräten angezeigt.

Standardmäßig können Benutzer zu Kontakteinträgen gehörende Bilder in ihrem Adressbuch synchronisieren. Sie können Bilder zur Desktop-E-Mail-Anwendung oder auf dem Endgerät hinzufügen, löschen und ersetzen.

Der BlackBerry Messaging Agent begrenzt die Dateigröße von Bildern für die Synchronisierung zwischen der Desktop-E-Mail-Anwendung und dem Endgerät auf maximal 32 KB.

Wenn Ihre Nachrichtenumgebung Outlook 2000 enthält, müssen Benutzer das Bild zudem *ContactPicture.jpg* nennen, wenn sie es an den Kontakteintrag anhängen. Andernfalls wird das Bild nicht synchronisiert.

Um die Synchronisierung von Kontaktbildern zu verhindern, deaktivieren Sie die Adressbuchfeldzuordnung für *Pictures* (Bilder).

17.1.1 Zuordnen eines Adressbuchfelds in der Desktop-E-Mail-Anwendung zu einem Adressbuchfeld auf allen Endgeräten

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Klicken Sie auf der Registerkarte *Global* (Global) auf *Service Control & Customization* (Dienststeuerung und -anpassung).
3. Klicken Sie auf *PIM Sync Global Field Mapping* (Feldzuordnung für globale PIM-Synchronisierung).
4. Klicken Sie in der Spalte *Desktop Field* (Desktop-Feld) auf ein Feld.
5. Klicken Sie in der Spalte *Device Field* (Gerätefeld) in der Dropdown-Liste auf das Adressbuchfeld des Endgeräts, das der Desktop-E-Mail-Anwendung zugeordnet werden soll.
6. Klicken Sie auf *OK*.

17.1.2 Zuordnen eines Adressbuchfelds in der Desktop-E-Mail-Anwendung zu einem Adressbuchfeld auf einem bestimmten Endgerät

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Users* (Benutzer) auf ein Benutzerkonto.
3. Klicken Sie auf *Service Control & Customization* (Dienststeuerung und -anpassung).
4. Klicken Sie auf *PIM Sync Field Mapping* (Feldzuordnung für PIM-Synchronisierung).
5. Klicken Sie in der Spalte *Desktop Field* (Desktop-Feld) auf ein Feld.
6. Klicken Sie in der Spalte *Device Field* (Gerätefeld) in der Dropdown-Liste auf das Adressbuchfeld des BlackBerry-Geräts, das der Desktop-E-Mail-Anwendung zugeordnet werden soll.
7. Klicken Sie auf *OK*.

17.1.3 Zuordnen eines benutzerdefinierten Adressbuchfelds zu einem Adressbuchfeld auf allen BlackBerry-Geräten

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Klicken Sie auf der Registerkarte *Global* (Global) auf *Service Control & Customization* (Dienststeuerung und -anpassung).

3. Klicken Sie auf *Edit PIM Sync Field Mapping* (Feldzuordnung für PIM-Synchronisierung bearbeiten).
4. Klicken Sie in der Spalte *Desktop Field* (Desktop-Feld) auf *User Defined String 1* (Benutzerdefinierte Zeichenfolge 1).
5. Klicken Sie in der Spalte *Device Field* (Gerätefeld) in der Dropdown-Liste auf ein benutzerdefiniertes Adressbuchfeld des Endgeräts, das dem Adressbuchfeld des Desktops zugeordnet werden soll.
6. Klicken Sie auf OK.

17.1.4 Zuordnen eines benutzerdefinierten Adressbuchfelds zu einem Adressbuchfeld auf einem bestimmten BlackBerry-Gerät

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Users* (Benutzer) auf ein Benutzerkonto.
3. Klicken Sie im unteren Fensterbereich auf *Service Control & Customization* (Dienststeuerung und -anpassung).
4. Klicken Sie auf *Edit PIM Sync Field Mapping* (Feldzuordnung für PIM-Synchronisierung bearbeiten).
5. Klicken Sie in der Spalte *Desktop Field* (Desktop-Feld) auf *User Defined String 1* (Benutzerdefinierte Zeichenfolge 1).
6. Klicken Sie in der Spalte *Device Field* (Gerätefeld) in der Dropdown-Liste auf ein benutzerdefiniertes Adressbuchfeld des Endgeräts, das dem Adressbuchfeld des Desktops zugeordnet werden soll.
7. Klicken Sie auf OK.

17.2 Senden von Nachrichten an Benutzer

Verwenden Sie den BlackBerry Manager zum Senden von Nachrichten oder PIN-Nachrichten an Benutzer in der BlackBerry-Domain. Da der Nachrichtenserver keine PIN-Nachrichten verarbeitet, ist die Funktion für die Übermittlung von PIN-Nachrichten sinnvoll, um Benutzer über Ausfälle des Nachrichtenservers zu informieren.

Endgeräte filtern keine PIN-Nachrichten, die vom BlackBerry Manager gesendet wurden.

Antworten eines Benutzers auf die Nachricht werden an das Dienstkonto gesendet, das Sie zum Installieren des Enterprise Servers verwendet haben (zum Beispiel BESAdmin).

17.2.1 Senden einer Nachricht an ausgewählte Benutzer

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Users* (Benutzer) auf ein Benutzerkonto.
3. Klicken Sie im unteren Fensterbereich auf *Account* (Konto).
4. Klicken Sie auf *Send Message* (Nachricht senden).
5. Führen Sie den Nachrichtenassistenten vollständig aus.

17.2.2 Senden einer Nachricht an alle Benutzer

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Server Configuration* (Serverkonfiguration) auf *Account* (Konto).
3. Klicken Sie auf *Send Message* (Nachricht senden).
4. Führen Sie den Nachrichtenassistenten vollständig aus.

18 Instant Messaging

Der BlackBerry Collaboration Service stellt auf dem Endgerät eine verschlüsselte Verbindung zwischen dem Instant-Messaging-Server und der Instant-Messaging-Anwendung her.

Um die Bandbreite und den Ressourcenverbrauch in Ihrer Umgebung zu steuern, legen Sie die zulässige Anzahl der geöffneten Instant-Messaging-Sitzungen zwischen dem BlackBerry Collaboration Service und dem Instant-Messaging-Server fest.

Der Collaboration Service unterstützt bis zu 2000 Instant-Messaging-Sitzungsverbindungen mit dem Microsoft Live Communications Server oder dem Lotus Sametime Server.

Die Anzahl der vom BlackBerry Collaboration Service unterstützten Instant-Messaging-Sitzungsverbindungen zum Instant-Messaging-Server von Novell GroupWise ist auf die Anzahl der vorhandenen Windows-Sockets beschränkt.

Sie können festlegen, ob Benutzer, die Sametime verwenden, neben dem Namen eines Sitzungsteilnehmers ein Instant-Messaging-Ortssymbol (von einem BlackBerry-Gerät oder -Computer) anzeigen können. Standardmäßig sehen Sametime-Benutzer es, wenn Sitzungsteilnehmer ihrer Kontaktliste BlackBerry Instant Messaging für Sametime auf BlackBerry-Geräten verwenden.

18.1 Konfigurieren der Verbindung zum Instant-Messaging-Server

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Collaboration Service* (Zusammenarbeitsdienst) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *General* (Allgemein).
4. Führen Sie im Bereich *Connection* (Verbindung) folgende Aktionen aus:

Tabelle 18.1: Konfigurieren der Verbindung zum Instant-Messaging-Server

Aktion	Vorgehensweise
Ändern des Host-Namens des Instant-Messaging-Servers	1. Doppelklicken Sie auf <i>Host</i> (Host). 2. Geben Sie den Host-Namen ein.
Ändern der Port-Nummer des Instant-Messaging-Servers	1. Doppelklicken Sie auf <i>Port</i> (Anschluss). 2. Geben Sie die Port-Nummer ein.
Ändern des Protokolls, mit dem der BlackBerry Collaboration Service eine Verbindung zum Instant-Messaging-Server herstellt (gilt nur für Microsoft Live Communications Server)	Klicken Sie in der Dropdown-Liste <i>Transport Protocol</i> (Transportprotokoll) auf einen der folgenden Protokolltypen: • <i>TLS</i>: Verwenden Sie TLS, wenn gleichzeitig bis zu 2000 Instant-Messaging-Sitzungen unterstützt werden sollen und Sie die Sicherheit zwischen dem BlackBerry Collaboration Service und dem Microsoft Live Communications Server durch Zertifikate erhöhen möchten. • <i>TCP</i>: Verwenden Sie TCP, wenn auf dem Microsoft Live Communications Server keine Zertifikate konfiguriert und nicht gleichzeitig 2000 Instant-Messaging-Sitzungen unterstützt werden sollen.

5. Klicken Sie auf *OK*.

18.2 Steuern einer Instant-Messaging-Sitzung

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Collaboration Service* (Zusammenarbeitsdienst) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *General* (Allgemein).
4. Führen Sie im Bereich *Service* (Dienst) eine oder mehrere der folgenden Aktionen aus:

Tabelle 18.2: Steuern einer Instant-Messaging-Sitzung

Aktion	Vorgehensweise
Festlegen der Anzahl gleichzeitig geöffneter Instant-Messaging-Sitzungen	1. Doppelklicken Sie auf <i>Maximum Simultaneous Sessions</i> (Höchstanzahl für gleichzeitige Sitzungen). 2. Geben Sie eine Zahl ein.

Aktion	Vorgehensweise
Festlegen der zulässigen Leerlaufzeit einer Instant-Messaging-Sitzung (in Sekunden), bevor sie geschlossen wird, damit eine neue Sitzung gestartet werden kann, wenn die maximale Anzahl gleichzeitiger Sitzungen erreicht wurde	1. Doppelklicken Sie auf <i>Idle Timeout</i> (Zeitüberschreitung für Leerlauf). 2. Geben Sie eine Zahl ein.
Festlegen der Inaktivitätszeit einer Instant-Messaging-Sitzung (in Millisekunden), bevor sie geschlossen wird	1. Doppelklicken Sie auf <i>Inactivity timeout</i> (Zeitüberschreitung wegen Inaktivität). 2. Geben Sie eine Zahl ein.
Ausblenden des Ortssymbols auf BlackBerry-Geräten. Standardmäßig sehen Sametime-Benutzer es, wenn Sitzungsteilnehmer ihrer Kontaktliste BlackBerry Instant Messaging für Sametime auf den Endgeräten verwenden (nur für Sametime)	1. Klicken Sie auf <i>Show Mobile Icon</i> (Mobilsymbol anzeigen). 2. Klicken Sie in der Dropdown-Liste auf <i>False</i> (Falsch).
Festlegen des Windows-Domänennamens des Unternehmens, damit Benutzer ihren Benutzernamen nicht eingeben müssen, wenn sie das SIP-Anmeldekonto angeben, um sich bei der Unternehmensanwendung zur Nachrichtenübermittlung auf den Endgeräten anzumelden	1. Doppelklicken Sie auf <i>Default Domain Name</i> (Standarddomänenname). 2. Geben Sie die Windows-Domänenadresse ein.

5. Klicken Sie auf *OK*.

19 Anpassen des Supports für Nachrichtenanhänge

In diesem Kapitel steht der Attachment Service im Mittelpunkt. Dieser Dienst erlaubt es, auf den Endgeräten E-Mail-Anhänge zu öffnen. Unterstützt werden die Microsoft-Office-Dateiformate (Word, Excel, Powerpoint), WordPerfect- und Adobe Acrobat-Dateien sowie HTML- und ASCII und auch im ZIP-Format gepackte Dokumente. Daneben kann auch eine Reihe gängiger Grafikformate, darunter GIF, JPEG, PNG und TIF, geöffnet werden. Die folgenden Abschnitte behandeln, wie der Attachment-Dienst konfiguriert wird.

19.1 Konfigurieren der Verbindungsherstellung zwischen BlackBerry Enterprise Server und BlackBerry Attachment Service

Wenn der BlackBerry Attachment Service auf einem (vom Enterprise Server) entfernten Computer installiert ist, konfigurieren Sie auf den einzelnen Computern bestimmte Verbindungseinstellungen:

- Legen Sie auf dem Enterprise Server die Konfigurationseinstellungen des Connectors so fest, dass der BlackBerry Messaging Agent eine Verbindung zum Attachment Service herstellt, wenn Benutzer auf den Endgeräten Nachrichtenanhänge abrufen möchten.
- Legen Sie auf dem Computer, auf dem der Attachment Service installiert ist, die Einstellungen des Anlagen-Servers so fest, dass eine Verbindung zum Enterprise Server hergestellt wird.

19.2 Herstellen einer Verbindung zwischen Enterprise Server und Attachment Service

1. Klicken Sie in der Taskleiste auf *Start / Programs (Programme) / BlackBerry Enterprise Server / BlackBerry Server Configuration* (BlackBerry-Serverkonfiguration).
2. Klicken Sie auf der Registerkarte *Attachment Server* (Anlagenserver) in der Drop-down-Liste *Configuration Option* (Konfigurationsoption) auf *Connector Configuration* (Connector-Konfiguration).
3. Führen Sie eine der folgenden Aktionen aus:

Tabelle 19.1: Herstellen einer Verbindung zwischen BlackBerry Enterprise Server und BlackBerry Attachment Service

Aktion	Vorgehensweise
Angeben des Namens oder der IP-Adresse des Computers, auf dem der Attachment Service installiert ist	Geben Sie im Feld <i>Server</i> einen Namen oder eine IP-Adresse an. Tipp: Wenn der Attachment Service auf demselben Computer wie der Enterprise Server installiert wurde, ist standardmäßig <i>localhost</i> angegeben.
Angeben der Nummer des TCP/IP-Ports, über den der Anlagen-Connector Anfragen für Anlagendaten an den Attachment Service sendet	Geben Sie im Feld <i>Server Submit Port</i> (Serveranfrage-Port) eine Port-Nummer zwischen 1024 und 65.535 ein.
Angeben der Nummer des TCP/IP-Ports, über den umfangreiche Anlagenkonvertierungsdaten vom Attachment Service angefragt und abgerufen werden	Geben Sie im Feld <i>Server Result Port</i> (Serverergebnis-Port) eine Port-Nummer zwischen 1024 und 65.535 ein.
Festlegen des Intervalls, in dem die Serverergebniszeit angefragt wird, wenn im Attachment Service umfangreiche Anlagen zur Zustellung bereitstehen	Geben Sie im Feld <i>Polling Time(s)</i> (Abrufzeit in Sekunden) einen Wert zwischen 10 und 300 Sekunden an.

4. Klicken Sie auf **OK**.
5. Starten Sie auf dem Computer, auf dem der Enterprise Server installiert ist, in den Windows-Diensten den BlackBerry-Dispatcher neu.

19.3 Herstellen einer Verbindung zwischen Attachment Service und Enterprise Server

1. Klicken Sie in der Taskleiste auf *Start / Programs (Programme) / BlackBerry Enterprise Server / BlackBerry Server Configuration* (BlackBerry-Serverkonfiguration).
2. Klicken Sie auf der Registerkarte *Attachment Server* (Anlagenserver) in der Drop-down-Liste *Configuration Option* (Konfigurationsoption) auf *Attachment Server* (Anlagenserver).
3. Führen Sie eine der folgenden Aktionen aus:

Tabelle 19.2: Herstellen einer Verbindung zwischen BlackBerry Attachment Service und BlackBerry Enterprise Server

Aktion	Vorgehensweise
Angeben der Nummer des TCP/IP-Ports, über den der Attachment Service Dokumentanfragen erhält und Konvertierungsergebnisse zurückgibt	Geben Sie im Feld <i>Submit Port</i> (Anfrage-Port) die Port-Nummer ein, die Sie auf dem Enterprise Server im Feld <i>Server Submit Port</i> (Serveranfrage-Port) angegeben haben.
Angeben der Nummer des TCP/IP-Ports, über den der Attachment Service umfangreiche Anlagenkonvertierungsdaten sendet, wenn diese vom Anlagen-Connector auf dem Server abgerufen wurden	Geben Sie im Feld <i>Result Port</i> (Ergebnis-Port) die Port-Nummer ein, die Sie auf dem Enterprise Server im Feld <i>Server Result Port</i> (Serverergebnis-Port) angegeben haben.
Angeben der TCP/IP-Port-Nummer für Konfigurations- und Administrationszwecke	Geben Sie im Feld <i>Configuration Port</i> eine Port-Nummer zwischen 1024 und 65.535 ein.

4. Klicken Sie auf **OK**.
5. Starten Sie auf dem Computer, auf dem der Attachment Service installiert ist, in den Windows-Diensten den Attachment Service neu.

19.4 Steuern der Konvertierung von Anlagen durch den Attachment Service

Sie können steuern, wie der Attachment Service Anlagen konvertiert, um seine Leistung zu optimieren. Außerdem können Sie die Einstellungen des Anlagenservers konfigurieren, um das Abrufen, Extrahieren und Konvertieren von Anlagendaten zu steuern.

Die Einstellungen des Anlagenservers können nur auf dem Computer geändert werden, auf dem der Attachment Service installiert ist.

Jede Anlagenkonvertierung weist beim Start Speicher zu, beansprucht Speicher bei der Konvertierung und speichert das Anlagen-DOM (Dokumentobjektmodell) lokal auf dem Computer zwischen, auf dem der Attachment Service installiert ist.

Je größer der Zwischenspeicher, desto mehr Speicher wird der jeweils ausgeführten Konvertierung zugewiesen. Die maximale Dateigröße der Anlagen wirkt sich auf den verwendeten Zwischenspeicher aus. Steuern Sie mithilfe der Einstellungen des Anlagenservers die vom Attachment Service verwendete Speichermenge.

Wenn der Enterprise Server eine Anlage empfängt, konvertiert der Attachment Service die Anlage in ein DOM und speichert dieses lokal.

Wenn Benutzer die Anlage auf ihren Endgeräten anzeigen möchten, greift der Attachment Service auf das DOM zu, um die Anfrage zu verarbeiten.

Im Speicher befinden sich nur die zwischengespeicherten Daten, niemals jedoch das Originaldokument.

19.5 Anpassen der Konvertierung von Anlagen durch den BlackBerry Attachment Service

1. Klicken Sie in der Taskleiste auf *Start / Programs* (Programme) / *BlackBerry Enterprise Server / BlackBerry Server Configuration* (BlackBerry-Serverkonfiguration).
2. Klicken Sie auf der Registerkarte *Attachment Server* (Anlagenserver) in der Dropdown-Liste *Configuration Option* (Konfigurationsoption) auf *Attachment Server* (Anlagenserver).
3. Führen Sie eine der folgenden Aktionen aus:

Tabelle 19.3: Anpassen der Konvertierung von Anlagen durch den BlackBerry Attachment Service

Aktion	Vorgehensweise
Verhindern, dass mehrere Anfragen für dieselbe Anlage die erste zwischengespeicherte Kopie des Anlagen-DOM in einer Konvertierung für den Benutzer verwenden	Klicken Sie in der Dropdown-Liste <i>Concurrent Caching</i> (Gleichzeitige Zwischenspeicherung) auf <i>Disabled</i> (Deaktiviert). Hinweis: Der Zwischenspeicher wird 25 Minuten aufrechterhalten (Standardzeit für die Wiederverwertung) oder so lange, bis eine neue Anfrage die Zwischenspeichergrenze für diesen Prozess überschreitet und das zuletzt verwendete Dokument im Zwischenspeicher gelöscht wird.
Festlegen der maximalen Anzahl konvertierter Dokumente, die sich pro Konvertierung (als DOM) im Dokumentzwischenspeicher befinden können	Geben Sie im Feld <i>Document Cache Size</i> (docs) (Dokumentzwischenspeichergröße (Dok.)) einen Wert zwischen 1 und 128 ein.
Festlegen der Anzahl der Konvertierungsanfragen, die der BlackBerry Attachment Service gleichzeitig verarbeiten kann	Geben Sie im Feld <i>Conversion Processes</i> (Konvertierungsprozess) einen Wert zwischen 1 und 64 ein. Hinweis: Orientieren Sie sich bei der Wahl des Wertes am verfügbaren Speicher und den konkurrierenden Diensten auf dem Computer, auf dem der BlackBerry Attachment Service installiert ist.

4. Klicken Sie auf *OK*.
5. Starten Sie auf dem Computer, auf dem der Attachment Service installiert ist, in den Windows-Diensten den Attachment Service neu.

19.6 Konfigurieren der Unterstützung für Anlagendateiformate

Der Attachment Service wandelt Anlagen mit Distillern in unterstützte Dateiformate um, die auf dem Endgerät angezeigt werden können. Alle unterstützten Distiller sind standardmäßig aktiviert.

Deaktivieren Sie einen Distiller, um zu verhindern, dass die Benutzer auf den Endgeräten Anlagen in bestimmten Dateiformaten anzeigen.

Wenn Sie zum Beispiel den *.pdf*-Distiller deaktivieren, können die Benutzer auf dem Endgerät keine PDF-Anlagen mehr anzeigen.

Wenn Sie einen Distiller für ein Anlagendateiformat deaktivieren, löschen Sie in der Formatliste der *Connector*-Konfigurationseinstellungen die Erweiterung für das Dateiformat, sodass die Option zum Öffnen der Anlage nicht auf dem Gerät angezeigt wird

19.7 Entfernen der Unterstützung für ein Anlagendateiformat

1. Klicken Sie in der Taskleiste auf *Start / Programs* (Programme) / *BlackBerry Enterprise Server / BlackBerry Server Configuration* (BlackBerry-Serverkonfiguration).
2. Klicken Sie auf der Registerkarte *Attachment Server* (Anlagenserver) in der Drop-down-Liste *Configuration Option* (Konfigurationsoption) auf *Connector Configuration* (Connector-Konfiguration).
3. Entfernen Sie im Feld *Format Extension* (Formaterweiterung) die Erweiterung für das Dateiformat.
4. Klicken Sie auf der Registerkarte *Attachment Server* (Anlagenserver) in der Drop-down-Liste *Configuration Option* (Konfigurationsoption) auf *Attachment Server* (Anlagenserver).
5. Deaktivieren Sie im Bereich *Distiller Settings* (Distiller-Einstellungen) das Kontrollkästchen neben dem zu löschenden Dateiformat.
6. Klicken Sie auf *OK*.
7. Starten Sie auf dem Computer, auf dem der Enterprise Server installiert ist, in den Windows-Diensten den BlackBerry-Dispatcher neu.
8. Starten Sie auf dem Computer, auf dem der Attachment Service installiert ist, in den Windows-Diensten den BlackBerry-Service neu.

19.8 Hinzufügen von Unterstützung für zusätzliche Erweiterungen von Anlagendateiformaten

Wenn Ihr Nachrichtenserver mit einem Dokumenten-Management-System verbunden ist, das eine Umbenennung der Erweiterungen für das Dateiformat vorschreibt, fügen Sie der Formatliste die entsprechenden Erweiterungen hinzu, um weitere Erweiterungen zu unterstützen.

1. Klicken Sie in der Taskleiste auf *Start / Programs* (Programme) / *BlackBerry Enterprise Server / BlackBerry Server Configuration* (BlackBerry-Serverkonfiguration).
2. Klicken Sie auf der Registerkarte *Attachment Server* (Anlagenserver) in der Drop-down-Liste *Configuration Option* (Konfigurationsoption) auf *Connector Configuration* (Connector-Konfiguration).
3. Geben Sie im Feld *Format Extension* (Formaterweiterung) die Erweiterung für das Dateiformat ein.

Tipp: Damit die Benutzer auf den Endgeräten alle Bildformate anzeigen können, aktivieren Sie das Kontrollkästchen *Image Attachments* (Bildanlagen).

4. Klicken Sie auf OK.
5. Starten Sie auf dem Computer, auf dem der Enterprise Server installiert ist, in den Windows-Diensten den BlackBerry-Dispatcher neu.

19.9 Steuern der Größe von Anlagendateien zum Minimieren der erforderlichen Konvertierungsressourcen

Standardmäßig ist im Attachment Service keine Größenbegrenzung für Anlagendateien eingestellt, die in eine Nachricht eingebettet sind oder aus dem Internet heruntergeladen wurden. Über ein drahtloses Netzwerk an das Endgerät gesendete Datenpakete dürfen nicht größer als 64 KB sein. Die Anzahl der Pakete, die gesendet werden können, ist jedoch unbegrenzt.

Ändern Sie in einer Umgebung mit hohem Datenaufkommen die maximale Dateigröße für die einzelnen Anlagenformate, um die vom Attachment Service während der Anlagenkonvertierung verwendete Speicherplatzmenge zu steuern.

Umgebungen mit hohem Datenaufkommen sind dadurch gekennzeichnet, dass der Attachment Service die folgenden Anfragen erhält:

- Mehrere Benutzer fordern Konvertierungen für große oder komplexe Anlagen an (insbesondere PDF- und ASCII-Textdateien mit mehr als 2 MB).

- Mehrere Benutzer fordern große oder komplexe Dokumente im selben Zeitraum an (0 bis 10 Minuten), während der Attachment Service umfangreiche Konvertierungen durchführt.

19.10 Festlegen der maximalen Dateigröße einer Anlage

1. Klicken Sie in der Taskleiste auf *Start / Programs* (Programme) / *BlackBerry Enterprise Server / BlackBerry Server Configuration* (BlackBerry-Serverkonfiguration).
2. Klicken Sie auf der Registerkarte *Attachment Server* (Anlagenserver) in der Dropdown-Liste *Configuration Option* (Konfigurationsoption) auf *Attachment Server* (Anlagenserver).
3. Klicken Sie im Bereich *Distiller Settings* (Distiller-Einstellungen) in der Spalte *Max. File Size (Kb)* (Max. Dateigröße in KB) auf den Wert neben dem bearbeiteten Distiller.
4. Geben Sie einen Wert in Kilobyte an. In einer Umgebung mit hohem Datenaufkommen empfiehlt RIM die folgenden Dateigrößen:

Tabelle 19.4: Festlegen der maximalen Dateigröße einer Anlage

<i>Dateiformat</i>	<i>Empfohlene Größe</i>
Adobe Acrobat 1.1, 1.2, 1.3 und 1.4	weniger als 2000 KB
Microsoft Excel 97, 2000, 2003 und XP	weniger als 2000 KB
Microsoft PowerPoint 97, 2000, 2003 und XP	weniger als 2000 KB
Microsoft Word 97, 2000, 2003 und XP	weniger als 2000 KB
Corel WordPerfect 6.0, 7.0, 8.0, 9.0 (2000) und 10.0	weniger als 2000 KB
ASCII-Text	weniger als 100 KB
HTML	weniger als 100 KB
ZIP-Archive	weniger als 2000 KB
Bilder	weniger als 2000 KB
Audio	weniger als 2000 KB

5. Klicken Sie auf OK.

19.11 Festlegen der Höchstmaße für Bilder

Sie können die Abmessungen der Bildanlagen festlegen, die Benutzer auf Endgeräten anzeigen können.

Standardmäßig legt der Attachment Service für Bildanlagen eine maximale Breite von 5000 Pixeln und eine Höhe von 4000 Pixeln fest.

Wenn Sie zulassen, dass der Attachment Service größere Bildanlagen konvertiert, empfiehlt es sich, den Attachment Service auf einem entfernten Computer zu installieren.

1. Geben Sie auf dem Computer, auf dem der BlackBerry Attachment Service installiert ist, an der Eingabeaufforderung `regedit` ein.
2. Gehen Sie zum Ordner `HKLM\Software\Research In Motion\BBAttachEngine\Distillers\LoadImageDistiller\`.
3. Doppelklicken Sie in der Liste *Name* auf den Schlüssel *MaxWidth* (Maximale Breite).
4. Geben Sie im Feld *Value data* (Datenwerte) die maximale Breite in Pixeln an.
5. Klicken Sie auf OK.
6. Gehen Sie zum Ordner `HKLM\Software\Research In Motion\BBAttachEngine\Distillers\LoadImageDistiller\`.
7. Doppelklicken Sie in der Liste *Name* auf den Schlüssel *MaxHeight* (Maximale Höhe).
8. Geben Sie im Feld *Value data* (Datenwerte) die maximale Höhe in Pixeln an.
9. Klicken Sie auf OK.
10. Starten Sie auf dem Computer, auf dem der Attachment Service installiert ist, in den Windows-Diensten den Attachment Service neu.

19.12 Steuern der Größe von Anlagendateien zum Minimieren der erforderlichen Übertragungsressourcen

Der BlackBerry Messaging Agent dient zum Empfangen von Anlagen, die lokal auf bestimmten Endgeräten mit der Version 4.2 der Gerätesoftware erstellt wurden, und zum Synchronisieren dieser Anlagen mit dem Nachrichtenserver. Diese Anlagen werden nicht vom Attachment Service konvertiert. Sie werden nur vom Messaging Agent empfangen.

Das Endgerät ermittelt, ob der Enterprise Server das Laden von Anlagen auf der Basis neuer CMIME-Einträge in das Servicebuch unterstützt. Die BlackBerry-Desktopsoftware Version 4.2 ist erforderlich, um sicherzustellen, dass diese neuen Servicebuch-Einträge bei Aktualisierungen des Servicebuchs über eine physische Verbindung zum Computer mit der BlackBerry-Desktopsoftware gültig bleiben.

Standardmäßig begrenzt der Messaging Agent die Dateigröße einer von einem Endgerät hochgeladenen Anlage auf maximal 3072 KB (3 MB). Werden gleichzeitig mehrere Anlagen geladen, begrenzt der BlackBerry Messaging Agent zudem die Gesamtdateigröße dieser Anlagen auf maximal 5120 KB (5 MB).

Zwischen dem Endgerät und dem Nachrichtenserver über das drahtlose Netzwerk übertragene Datenpakete dürfen maximal eine Größe von 64 KB haben. Überschreitet

eine von einem Endgerät hochgeladene Anlage die maximal zulässige Paketgröße, teilt das Gerät die Anlage in mehrere Pakete. Der Messaging Agent speichert alle Pakete bis zum Empfang des letzten Pakets im Cache-Speicher und sendet die Anlage anschließend an den Nachrichtenserver.

Um in einer Umgebung mit hohem Datenaufkommen den vom Messaging Agent beim Laden von Anlagen belegten Speicherplatz und die Anzahl der Transaktionen zu steuern, führen Sie eine der folgenden Aktionen aus:

- Ändern der maximalen Dateigröße,
- Vermeiden, dass Anlagen geladen werden.

19.12.1 Ändern der maximalen Größe von Dateianlagen

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Server Configuration* (Serverkonfiguration) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie im linken Fensterbereich auf *Messaging* (Nachrichtenübermittlung).
4. Führen Sie eine der folgenden Aktionen aus:

Tabelle 19.5: Ändern der maximalen Größe von Dateianlagen

Aktion	Vorgehensweise
Ändern der maximalen Dateigröße einer Anlage zum Hochladen	Geben Sie im Feld <i>Maximum Upload Attachment Size</i> (Maximale Größe hochzuladender Anlagen) einen Wert zwischen 1 und 3072 ein.
Ändern der maximalen Dateigröße für mehrere Anlagen, die gleichzeitig hochgeladen werden.	Geben Sie im Feld <i>Maximum Upload Total Attachment Size</i> (Maximale Gesamtgröße hochzuladender Anlagen) einen Wert zwischen 1 und 5120 ein, der außerdem größer als die maximale Größe einer einzelnen hochzuladenden Anlage ist.

19.13 Vermeiden, dass Anlagen geladen werden

Wenn Sie die Dateigröße von hochzuladenden Anlagen festlegen, können Benutzer nur bestimmte Anlagen (Zertifikate und Adressbucheinträge) hochladen, die kleiner als ein Paket sind.

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Server Configuration* (Serverkonfiguration) auf *Edit Properties* (Eigenschaften bearbeiten).

3. Klicken Sie im linken Fensterbereich auf *Messaging* (Nachrichtenübermittlung).
4. Geben Sie im Feld *Maximum Upload Attachment Size* (Maximale Größe hochzuladender Anlagen) den Wert 0 ein.

20 Anpassen des drahtlosen Zugriffs auf Unternehmensanwendungen

BlackBerry-Endgeräte erlauben auch den Zugriff auf unternehmensweite Anwendungen. Unter anderem ist es möglich, Daten aus der SAP-Welt auf den BlackBerry abzurufen. In diesem Zusammenhang sind die Authentifizierung und die Datensicherheit zentrale Themen: BlackBerrys müssen sich bei den Unternehmensanwendungen anmelden und eine sichere Datenübertragung gewährleisten können. Damit beschäftigt sich dieses Kapitel.

20.1 Zentrale Push-Server

Legen Sie mit dem BlackBerry Manager einen MDS Connection Service in einer BlackBerry-Domain als zentralen Push-Server fest. Der zentrale Push-Server empfängt Push-Anfragen von Anwendungen. Er stellt eine Verbindung zum BlackBerry-Gerät her, über die Daten von Anwendungen gesendet werden.

Pro BlackBerry-Domain kann nur ein MDS Connection Service als zentraler Push-Server festgelegt werden. Wenn Sie einen MDS Connection Service als zentralen Push-Server festlegen, löst dieser den zuvor als zentralen Push-Server verwendeten MDS Connection Service ab. Sie müssen zunächst die vorherige Push-Server-Festlegung eines MDS Connection Service aufheben, bevor Sie sie einem anderen Connection Service zuweisen. Wenn Sie den zentralen Push-Server ändern, müssen Sie Ihre Push-Anwendungsentwickler entsprechend informieren.

Wenn Sie den MDS Connection Service deaktivieren, wird auch der BlackBerry Collaboration Service deaktiviert.

20.2 Festlegen des zentralen Push-Servers

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen BlackBerry MDS Connection Service.
2. Klicken Sie auf der Registerkarte *Connection Service* (Verbindungsdienst) auf die Schaltfläche *Common* (Gemeinsam).
3. Klicken Sie auf *Set as Push Server* (Als Push-Server einrichten).

4. Wenn Sie MDS Services installiert haben, stellen Sie sicher, dass der zentrale Push-Server in der Liste der MDS Connection Services angezeigt wird, die für die BlackBerry MDS Services zur Verfügung stehen. Weitere Informationen finden Sie im Kapitel »Festlegen der Verbindung zwischen BlackBerry MDS Services und dem BlackBerry MDS Connection Service«.

20.3 Anpassen der Authentifizierung der BlackBerry-Geräte beim Webserver

Legen Sie fest, ob sich die Endgeräte direkt bei einem Inhalts-Webserver authentifizieren, oder ob sich der MDS Connection Service im Namen der BlackBerry-Geräte beim Webserver authentifiziert.

Wenn Sie die Endgeräte so konfigurieren, dass sie sich direkt bei Webservern authentifizieren, werden die Benutzer nach der Authentifizierung alle 30 Minuten aufgefordert, die Anmeldeinformationen auf den authentifizierten Geräten einzugeben.

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen BlackBerry MDS Connection Service.
2. Klicken Sie auf der Registerkarte *Connection Service* (Verbindungsdienst) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *HTTP*.
4. Führen Sie eine der folgenden Aktionen aus:

Tabelle 20.1: Konfigurieren der Authentifizierung der BlackBerry-Geräte bei Webservern

Aktion	Vorgehensweise
Konfigurieren der BlackBerry-Geräte für die direkte Authentifizierung bei Webservern	1. Klicken Sie auf <i>Support HTTP Authentication</i> (HTTP-Authentifizierung unterstützen). 2. Klicken Sie in der Dropdown-Liste auf <i>False</i> (Falsch).
Konfigurieren der Authentifizierung des BlackBerry MDS Connection Service bei Webservern mithilfe von HTTP Basic im Namen von BlackBerry-Geräten	1. Klicken Sie auf <i>Support HTTP Authentication</i> (HTTP-Authentifizierung unterstützen). 2. Klicken Sie in der Dropdown-Liste auf <i>True</i> (Wahr).

5. Doppelklicken Sie auf *Authentication Timeout* (Authentifizierungs-Timeout).
6. Geben Sie in Millisekunden an, wie lange Authentifizierungsangaben auf dem Webserver gespeichert werden.
7. Klicken Sie auf *OK*.

20.4 Konfigurieren der Authentifizierung des MDS Connection Service bei Servern, die NTLM verwenden

Konfigurieren Sie unter <Laufwerk:>\Programme\Research In Motion\BlackBerry Enterprise Server\MDS\Servers\ServerInstance\config die Datei *MDSLogin.conf* sowie die Konfigurationsdatei von Java Authentication and Authorization Service (JAAS).

Weitere Informationen finden Sie unter:

<http://java.sun.com/j2se/1.5.0/docs/guide/security/jgss/tutorials/LoginConfigFile.html>

20.5 Konfigurieren der Authentifizierung des MDS Connection Service bei Servern, die Kerberos verwenden

Konfigurieren Sie unter <Laufwerk:>\Programme\Research In Motion\BlackBerry Enterprise Server\MDS\Servers\ServerInstance\config die Konfigurationsdatei für Kerberos 5 (*krb5.conf*).

Weitere Informationen finden Sie unter der Internetadresse <http://web.mit.edu/kerberos/www/krb5-1.3/krb5-1.3.3/doc/krb5-admin.html#krb5.conf>

20.6 Konfigurieren der Authentifizierung des MDS Connection Service bei Servern, die LTPA verwenden

Aktivieren Sie das Speichern von Cookies, damit der MDS Connection Service sich bei Webservern authentifizieren kann, die LTPA (Lightweight Third-Party Authentication, zu deutsch »leichtgewichtige Authentifizierung durch Dritte«) verwenden.

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen MDS Connection Service.
2. Klicken Sie auf der Registerkarte *Connection Service* (Verbindungsdienst) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *HTTP*.
4. Klicken Sie auf *Support HTTP Cookie Storage* (Speichern von HTTP-Cookies unterstützen).
5. Klicken Sie in der Dropdown-Liste auf *True* (Wahr).
6. Klicken Sie auf *OK*.

20.7 Konfigurieren der Authentifizierung des MDS Connection Service beim RSA Authentication Manager

Wenn Sie die RSA-Authentifizierung aktivieren, müssen Benutzer ihre Anmeldeinformationen auf den Endgeräten eingeben, bevor sie auf Intranet- oder Internetinhalte zugreifen können. Wenn die Proxy-Authentifizierung konfiguriert wurde, wird der Benutzer aufgefordert, sich beim Proxy-Server zu authentifizieren.

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen MDS Connection Service.
2. Klicken Sie auf der Registerkarte *Connection Service* (Verbindungsdienst) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf die Schaltfläche *RSA Authentication* (RSA-Authentifizierung).
4. Führen Sie folgende Aktionen aus:

Tabelle 20.2: Konfigurieren der Authentifizierung des BlackBerry MDS Connection Service beim RSA Authentication Manager

Aktion	Vorgehensweise
Aktivieren der RSA-Authentifizierung	1. Klicken Sie auf <i>Enable RSA Authorization Support</i> (Unterstützung der RSA-Autorisierung aktivieren). 2. Klicken Sie in der Dropdown-Liste auf <i>True</i> (Wahr).
Festlegen der Dauer (in Minuten), die ein Endgerät mit dem Unternehmensnetzwerk verbunden sein kann, bevor sich der Benutzer erneut anmelden muss	1. Doppelklicken Sie auf <i>RSA Authentication Timeout</i> (Timeout für RSA-Inaktivität). 2. Geben Sie eine Zahl ein.
Festlegen der Dauer (in Minuten), die ein authentifiziertes Endgerät während der Verbindung mit dem Unternehmensnetzwerk inaktiv sein kann, bevor sich der Benutzer erneut anmelden muss	1. Doppelklicken Sie auf <i>RSA Inactivity Timeout</i> (Timeout für RSA-Inaktivität). 2. Geben Sie eine Zahl ein.

5. Klicken Sie auf *OK*.

20.8 Beschränken des Benutzerzugriffs auf Webinhalte

Erstellen Sie Regeln für die Pull-Zugriffssteuerung, um die Webserver einzuschränken, auf die der MDS Connection Service im Namen eines Benutzers zugreift. Sie weisen Benutzern Pull-Regeln zu, um zu steuern, von welchen Webservern Inhalte angefragt

werden können. Der MDS Connection Service überträgt die Inhalte, die Benutzer von ihren Endgeräten aus anfragen.

20.8.1 Beschränken der Anfragen für Webinhalte von BlackBerry-Geräten aus

Legen Sie fest, ob Regeln für die Zugriffssteuerung auf Anfragen für Webinhalte vom Endgerät aus angewendet werden. Aktivieren Sie die Pull-Berechtigung, um die Webinhalte einzuschränken, die die Benutzer auf BlackBerry-Geräten empfangen können.

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen MDS Connection Service.
2. Klicken Sie auf der Registerkarte *Connection Service* (Verbindungsdienst) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Access Control* (Zugriffssteuerung).
4. Klicken Sie auf *Pull Authorization* (Pull-Berechtigung).
5. Klicken Sie in der Dropdown-Liste auf *True* (Wahr).
6. Klicken Sie auf *OK*.

20.8.2 Erstellen und Zuweisen einer Regel zu Anfragen für bestimmte Webinhalte

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Klicken Sie auf der Registerkarte *Global* auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie im linken Fensterbereich auf *Access Control* (Zugriffssteuerung).
4. Führen Sie folgende Aktionen aus:

Tabelle 20.3: Erstellen und Zuweisen einer Regel zu Anfragen für bestimmte Webinhalte

Aktion	Vorgehensweise
Erstellen einer eindeutigen Pull-Regel	1. Doppelklicken Sie auf <i>Pull Rules</i> (Pull-Regeln). 2. Klicken Sie auf <i>New</i> (Neu). 3. Doppelklicken Sie auf <i>Name</i>. 4. Geben Sie einen Namen für die Regel an. 5. Doppelklicken Sie auf <i>Description</i> (Beschreibung). 6. Geben Sie eine Beschreibung für die Regel an. 7. Klicken Sie auf <i>OK</i>. 8. Klicken Sie erneut auf <i>OK</i>.

Aktion	Vorgehensweise
Erstellen Sie ein URL-Muster.	1. Doppelklicken Sie auf <i>URL Patterns</i> (URL-Muster). 2. Klicken Sie auf <i>New</i> (Neu). 3. Doppelklicken Sie auf <i>URL Pattern</i> (URL-Muster). 4. Geben Sie das URL-Muster des Webserver an, für den die Pull-Regel den Zugriff steuert. 5. Klicken Sie in der Dropdown-Liste <i>Service Name</i> (Dienstname) auf eine der folgenden Optionen: • <i>http</i>: Die Regel gilt, wenn Benutzer von BlackBerry-Geräten aus eine Verbindung zu einer HTTP-Webseite anfordern. • <i>https</i>: Die Regel gilt, wenn Benutzer von Endgeräten aus eine Verbindung zu einer HTTPS-Webseite anfordern und SSL oder TLS im Proxy-Modus aktiviert ist. • <i>ldap</i>: Die Regel gilt, wenn Benutzer von Endgeräten aus auf ein Benutzerprofil oder Zertifikat zugreifen. Der MDS Connection Service ruft das Benutzerprofil oder Zertifikat aus dem LDAP-Verzeichnis ab. • <i>ocsp</i>: Die Regel gilt, wenn Benutzer von BlackBerry-Geräten aus den Sperrstatus eines Zertifikats überprüfen. Der BlackBerry MDS Connection Service ruft den Sperrstatus des Zertifikats vom OCSP-Server ab. • <i>tcp</i>: Die Regel gilt, wenn Benutzer von BlackBerry-Geräten aus eine Verbindung ins Internet oder Unternehmensintranet mithilfe anderer Standard-Internet-Protokolle anfordern. 6. Doppelklicken Sie auf <i>Description</i> (Beschreibung). 7. Geben Sie eine Beschreibung für das URL-Muster ein. 8. Klicken Sie auf <i>OK</i>. 9. Klicken Sie erneut auf <i>OK</i>.
Zuweisen einer Regel zu einem URL-Muster und Definieren des URL-Zugriffs	1. Doppelklicken Sie auf <i>URL Pattern Rules</i> (Regeln für URL-Muster). 2. Klicken Sie im linken Fensterbereich auf die Pull-Regel. 3. Führen Sie im rechten Fensterbereich eine der folgenden Aktionen aus: • Um zu verhindern, dass der Benutzer, dem die Regel zugewiesen ist, auf einen dem URL-Muster entsprechenden URL zugreift, wählen Sie die Option <i>Deny</i> (Verweigern). • Um zuzulassen, dass der Benutzer, dem die Regel zugewiesen ist, auf einen dem URL-Muster entsprechenden URL zugreift, wählen Sie die Option <i>Allow</i> (Zulassen). 4. Klicken Sie auf <i>OK</i>.

20.8.3 Zuweisen von Regeln zu Benutzerkonten oder Gruppen

1. Führen Sie im BlackBerry Manager im linken Fensterbereich eine der folgenden Aktionen aus:

Tabelle 20.4: Zuweisen von Regeln zu Benutzerkonten oder Gruppen

Aktion	Vorgehensweise
Zuweisen einer Pull-Regel zu einem einzelnen Benutzerkonto	1. Klicken Sie auf <i>BlackBerry Domain</i> (BlackBerry-Domäne). 2. Klicken Sie auf der Registerkarte <i>Global</i> auf <i>Edit Properties</i> (Eigenschaften bearbeiten). 3. Klicken Sie im linken Fensterbereich auf <i>Access Control</i> (Zugriffssteuerung). 4. Doppelklicken Sie auf <i>User Rules</i> (Benutzerregeln). 5. Klicken Sie im linken Fensterbereich auf eine Regel. 6. Wählen Sie im rechten Fensterbereich die Option für ein Benutzerkonto. 7. Klicken Sie auf <i>OK</i>.
Zuweisen einer Pull-Regel zu den Benutzern einer Gruppe	1. Klicken Sie auf eine Gruppe. 2. Klicken Sie auf der Registerkarte <i>Group Configuration</i> (Gruppenkonfiguration) auf <i>Edit Group Template</i> (Gruppenvorlage bearbeiten). 3. Klicken Sie auf <i>Access Control</i> (Zugriffssteuerung). 4. Doppelklicken Sie auf <i>Pull Rule Set</i> (Pull-Regel festlegen). 5. Aktivieren Sie das Kontrollkästchen für die Pull-Regel, die Sie der Gruppe zuweisen möchten. 6. Klicken Sie auf <i>OK</i>. 7. Aktivieren Sie das Kontrollkästchen neben <i>Pull Rule Set</i> (Pull-Regel festlegen). 8. Klicken Sie auf <i>Reapply Template</i> (Vorlage erneut anwenden). 9. Klicken Sie auf <i>Yes</i> (Ja).

2. Klicken Sie auf *OK*.

Beschränken des Benutzerzugriffs auf bestimmte Medien

Sie können steuern, welche Medientypen (zum Beispiel Audio und Video) Benutzer auf den Endgeräten empfangen können.

Legen Sie mit Standarddefinitionen für MIME-Medientypen fest, ob der MDS Connection Service die Medien an das BlackBerry-Gerät senden kann. Sie können auch die Dateigröße jedes Medientyps begrenzen. Weitere Informationen zu MIME-Medientypen finden Sie unter der Internetadresse www.iana.org.

Sie können festlegen, dass Benutzer Medien (zum Beispiel Videodaten) in keinem Format empfangen oder dass sie sie nur in bestimmten Formaten abrufen können (zum Beispiel *mp4*). Wenn Sie nur bestimmte Formate sperren möchten, müssen Sie beim Erstellen der Beschränkung die Definition des Medientyps und des Subtyps eingeben (zum Beispiel *Video/mp4*).

20.8.4 Erstellen von Beschränkungen für Medieninhalte

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Klicken Sie auf der Registerkarte *Global* auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Media Content Management* (Medieninhalte verwalten).
4. Doppelklicken Sie auf *Media Content Types* (Medieninhaltstypen).
5. Klicken Sie auf *New* (Neu).
6. Geben Sie im Feld *Media Content Type* (Medieninhaltstyp) den Medientyp und optional einen Subtyp ein.
7. Führen Sie eine der folgenden Aktionen aus:

Tabelle 20.5: Erstellen von Beschränkungen für Medieninhalte

Aktion	Vorgehensweise
Verhindern, dass der MDS Connection Service die Medien an BlackBerry-Geräte sendet	1. Klicken Sie in der Dropdown-Liste <i>Disallow content</i> (Inhalt sperren) auf <i>True</i> (Wahr). 2. Klicken Sie auf <i>OK</i>.
Zulassen, dass der MDS Connection Service die Medien bis zu einer bestimmten Dateigröße an BlackBerry-Geräte sendet	1. Geben Sie unter <i>Maximum KB/Connection</i> (Maximale KB/ Verbindung) die maximale Dateigröße ein. 2. Klicken Sie in der Dropdown-Liste <i>Disallow content</i> (Inhalt sperren) auf <i>False</i> (Falsch). 3. Klicken Sie auf <i>OK</i>.

8. Klicken Sie auf *OK*.

20.8.5 Verwalten von Beschränkungen für Medieninhalte

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Klicken Sie auf der Registerkarte *Global* auf *Edit Properties* (Eigenschaften bearbeiten).

3. Klicken Sie auf *Media Content Management* (Medieninhalte verwalten).
4. Doppelklicken Sie auf *Media Content Types* (Medieninhaltstypen).
5. Klicken Sie auf eine Medientypbeschränkung.
6. Führen Sie eine der folgenden Aktionen aus:

Tabelle 20.6: Verwalten von Beschränkungen für Medieninhalte

Aktion	Vorgehensweise
Ändern einer bestehenden Beschränkung für Medieninhalte	1. Klicken Sie auf <i>Eigenschaften</i>. 2. Ändern Sie die Dateigröße und/oder den Medientyp. 3. Klicken Sie auf <i>OK</i>.
Löschen einer bestehenden Beschränkung für Medieninhalte	Klicken Sie auf <i>Entfernen</i> .

7. Klicken Sie auf *OK*.

20.8.6 Festlegen der Behandlung von Webanfragen der Endgeräte durch den MDS Connection Service

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen MDS Connection Service.
2. Klicken Sie auf der Registerkarte *Connection Service* (Verbindungsdienst) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *HTTP*.
4. Führen Sie folgende Aktionen aus:

Tabelle 20.7: Behandlung von Webanfragen

Aktion	Vorgehensweise
Zwischenspeichern von Cookies im Namen von Endgeräten und Aktivieren des MDS Connection Service, um HTTP-Anfragen von Endgeräten Cookie-Informationen hinzuzufügen Hinweis: Wenn das Endgerät in seinen HTTP-Anfragen JavaScript-Unterstützung benötigt, werden Cookies auf dem BlackBerry-Gerät verarbeitet.	1. Klicken Sie auf <i>Support HTTP Cookie Storage</i> (Speichern von HTTP-Cookies unterstützen). 2. Klicken Sie in der Dropdown-Liste auf <i>True</i> (Wahr).

Aktion	Vorgehensweise
Festlegen der Dauer (in Millisekunden), die die HTTP-Verbindung auf das Senden von Daten durch das Endgerät wartet	1. Doppelklicken Sie auf <i>HTTP Device Connection Timeout</i> (Timeout für HTTP-Geräteverbindung). 2. Geben Sie eine Zahl ein.
Festlegen der Dauer (in Millisekunden), die die HTTP-Verbindung auf das Senden von Daten durch den Webserver wartet	1. Doppelklicken Sie auf <i>HTTP Server Connection Timeout</i> (Timeout für HTTP-Serververbindung). 2. Geben Sie eine Zahl ein.
Festlegen der maximalen Anzahl der vom BlackBerry MDS Connection Service unterstützten HTTP-Umleitungen Hinweis: Eine HTTP-Umleitung tritt auf, wenn der BlackBerry-Browser eine Webseite von einem Webserver anfordert und der Webserver einen Umleitungsstatuscode mit dem neuen URL der Webseite zurückgibt.	1. Doppelklicken Sie auf <i>Maximum Number of Redirects</i> (Maximale Anzahl der Umleitungen). 2. Geben Sie eine Zahl ein.

5. Klicken Sie auf *OK*.

20.9 Zulassen vertrauenswürdiger Verbindungen zwischen Push-Anwendungen und dem MDS Connection Service

Generieren Sie eine Datei namens *webserver.keystore*, die ein Zertifikat für den MDS Connection Service enthält. Für Push-Anwendungen muss dieses Zertifikat eine HTTP-über-SSL-Verbindung zum MDS Connection Service herstellen, wenn Inhalt per Push-Verfahren auf ein Endgerät übertragen wird.

Erstellen Sie mithilfe des Schlüsselwerkzeugs ein selbstsigniertes Zertifikat für den MDS Connection Service, oder importieren Sie ein signiertes Zertifikat von einer vertrauenswürdigen öffentlichen Zertifizierungsstelle. Exportieren Sie das Zertifikat des MDS Connection Service mithilfe des Schlüsselwerkzeugs aus der Datei *webserver.keystore*, und importieren Sie es in die Schlüsselspeicher anderer Anwendungen, beispielsweise von Windows oder Java.

Weitere Informationen über das Schlüsselwerkzeug finden Sie unter <http://java.sun.com/j2se/1.5.0/docs/tooldocs/windows/keytool.html>. Weitere Informationen zu Anforderungen für Apache Tomcat finden Sie unter <http://tomcat.apache.org/tomcat-5.5-doc/ssl-howto.html>.

20.10 Veröffentlichen des MDS-Connection-Service-Zertifikats für vertrauenswürdige Verbindungen zwischen Push-Anwendungen und dem MDS Connection Service

1. Gehen Sie auf dem Computer, auf dem der BlackBerry MDS Connection Service installiert ist, zu `<Laufwerk:>/Programme/Java/<JRE-Version>/bin`, und führen Sie an der Eingabeaufforderung eine der folgenden Aktionen aus:

Tabelle 20.8: Veröffentlichen des MDS-Connection-Service-Zertifikats

Aktion	Vorgehensweise
Generieren eines selbstsignierten Zertifikats und Veröffentlichen in der Datei <i>webserver.keystore</i>	1. Geben Sie <code>keytool -genkey -alias tomcat -keyalg RSA -keystore webserver.keystore</code> ein. 2. Geben Sie die erforderlichen Informationen ein. 3. Bestätigen Sie die Eingabe. Ist sie richtig, geben Sie Yes (Ja) ein.
Veröffentlichen eines öffentlich signierten Zertifikats in der Datei <i>webserver.keystore</i>	1. Geben Sie <code>keytool -import -trustcacerts -alias tomcat -file <trustedserver.cer> -keystore webserver.keystore</code> ein. 2. Geben Sie das Kennwort für den Schlüsselspeicher ein. 3. Bestätigen Sie mit Yes (Ja), um das Zertifikat zum Schlüsselspeicher hinzuzufügen.

2. Kopieren Sie die Datei *webserver.keystore* in das Verzeichnis `<Laufwerk:>/Programme/Research In Motion/BlackBerry Enterprise Server/MDS/webserver`.

20.10.1 Exportieren des MDS-Connection-Service-Zertifikats zur Bereitstellung für andere Anwendungen

1. Gehen Sie auf dem Computer, auf dem der BlackBerry MDS Connection Service installiert ist, zu `<Laufwerk:>/Programme/Java/<JRE-Version>/bin`, und geben Sie an der Eingabeaufforderung Folgendes ein: `keytool -export -alias tomcat -file <Server.cer> -keystore <Laufwerk:>/Programme/Research In Motion/BlackBerry Enterprise Server/MDS/webserver/webserver.keystore -storepass <Kennwort>`
2. Geben Sie das Kennwort für den Schlüsselspeicher ein.

20.10.2 Zulassen des vertrauenswürdigen MDS-Connection-Service-Zertifikats für Java-Anwendungen

1. Gehen Sie auf dem Computer, auf dem der MDS Connection Service installiert ist, zu `<Laufwerk>/Programme/Java/<JRE-Version>/bin`, und geben Sie an der Eingabeaufforderung Folgendes ein: `keytool -import -trustcacerts -alias <Alias> -file <Server.cer> -keystore <Anwendungsschlüsselspeicher>`
2. Geben Sie das Kennwort für den Schlüsselspeicher ein.
3. Bestätigen Sie mit *Yes* (Ja), um das Zertifikat zum Schlüsselspeicher hinzuzufügen.

20.11 Anpassen der Herstellung vertrauenswürdiger Verbindungen zu Webservern durch Anwendungen

Konfigurieren Sie, wie Anwendungen auf BlackBerry-Geräten Zertifikatinformationen für vertrauenswürdige und nicht vertrauenswürdige Webserver abrufen. Der MDS Connection Service unterstützt LDAP (Lightweight Directory Access Protocol), OCSP (Online Certificate Status Protocol), SSL und TLS (Transport Layer Security). Zertifikate authentifizieren Anwendungen beim MDS Connection Service.

Konfigurieren Sie eine Schlüsselspeicherdatei, damit die Endgeräte und die Anwendungen eine Verbindung zu nicht vertrauenswürdigen Servern herstellen können, wenn auf dem Computer, auf dem der MDS Connection Service installiert ist, kein Zertifikat für den Server gespeichert ist. Mithilfe der Schlüsselspeicherdatei kann eine Push-Anwendung eine HTTP-über-SSL-Verbindung zum MDS Connection Service herstellen, wenn Inhalte per Push-Verfahren auf ein Endgerät übertragen werden.

20.11.1 Konfigurieren der Anfrage für Zertifikate vertrauenswürdiger Anwendungen durch den MDS Connection Service bei LDAP-Servern

Definieren Sie einen Benutzernamen und ein Kennwort für die Authentifizierung des MDS Connection Service bei LDAP-Servern im Namen von BlackBerry-Geräten.

Ändern Sie die standardmäßigen LDAP-Port-Parameter nur im Falle eines Port-Konflikts mit einem anderen Dienst auf demselben Computer. Wenn Sie Port- oder Host-Angaben ändern, müssen Sie den MDS Connection Service neu starten, um die Konfigurationsangaben erneut zu laden.

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen MDS Connection Service.
2. Klicken Sie auf der Registerkarte *Connection Service* (Verbindungsdienst) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *LDAP*.

4. Legen Sie die Einstellungen des LDAP-Servers fest.
5. Klicken Sie auf OK.

20.11.2 Konfigurieren der Abfrage eines Zertifikatstatus durch den MDS Connection Service von einem OCSP-Server

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen MDS Connection Service.
2. Klicken Sie auf der Registerkarte *Connection Service* (Verbindungsdienst) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *OCSP*.
4. Führen Sie eine der folgenden Aktionen aus:

Tabelle 20.9: Konfigurieren der Abfrage eines Zertifikatstatus

Aktion	Vorgehensweise
Konfigurieren des OCSP-Handlers für das Akzeptieren der vom BlackBerry-Gerät angegebenen OCSP-Responder	1. Klicken Sie auf <i>Use Device Responders</i> (Geräte-Responder verwenden). 2. Klicken Sie in der Dropdown-Liste auf <i>True</i> (Wahr).
Konfigurieren des OCSP-Handlers für die Verwendung der OCSP-Responder-Erweiterung in einem Zertifikat	1. Wenn ein Zertifikat vorhanden ist, klicken Sie auf <i>Use Certificate Extension Responders</i> (Responder-Erweiterungen verwenden). 2. Klicken Sie in der Dropdown-Liste auf <i>True</i> (Wahr).
Legen Sie den Standard-URL des OCSP-Responders fest.	1. Doppelklicken Sie auf <i>Default Responder URL</i> (Standard-URL des Responders). 2. Geben Sie den URL des OCSP-Responders ein.
Festlegen des URL des Servers, auf dem sich die Zertifikatsperrliste (CRL) befindet	1. Doppelklicken Sie auf <i>Default CRL Server URL</i> (Standard-URL des CRL-Servers). 2. Geben Sie den URL des CRL-Servers ein.
Festlegen des URL des Servers, auf dem sich die PGP-Schlüssel befinden	1. Doppelklicken Sie auf <i>Default PGP Server URL</i> (Standard-URL des PGP-Servers). 2. Geben Sie den URL des PGP-Servers ein.

5. Klicken Sie auf OK.

20.11.3 Zulassen der Herstellung von Verbindungen zu nicht vertrauenswürdigen Webservern durch BlackBerry-Geräte

Ein Webserver ist nicht vertrauenswürdig, wenn auf dem Enterprise Server kein Zertifikat für den Webserver gespeichert ist.

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen MDS Connection Service.
2. Klicken Sie auf der Registerkarte *Connection Service* (Verbindungsdienst) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *TLS/HTTPS*.
4. Führen Sie eine der folgenden Aktionen aus:

Tabelle 20.10: Zulassen der Herstellung von Verbindungen zu nicht vertrauenswürdigen Webservern

Aktion	Vorgehensweise
Zulassen abgehender Anfragen vom BlackBerry-Gerät, die der BlackBerry MDS Connection Service mit HTTPS verschlüsselt	1. Klicken Sie auf <i>Allow Untrusted HTTPS Connections</i> (Nicht vertrauenswürdige HTTPS-Verbindungen zulassen). 2. Wählen Sie in der Dropdown-Liste <i>True</i> (Wahr).
Zulassen abgehender Anfragen vom BlackBerry-Gerät, die der BlackBerry MDS Connection Service mit TLS verschlüsselt	1. Klicken Sie auf <i>Allow Untrusted TLS Connections</i> (Nicht vertrauenswürdige TLS-Verbindungen zulassen). 2. Wählen Sie in der Dropdown-Liste <i>True</i> (Wahr).

20.12 Zulassen von Verbindungen von Endgeräten zu vertrauenswürdigen Webservern

Verwenden Sie das Schlüsselwerkzeug, um dem Schlüsselspeicher des Enterprise Servers ein Zertifikat für einen Webserver hinzuzufügen und Verbindungen zum vertrauenswürdigen Webserver zuzulassen.

1. Kopieren Sie das Zertifikat von einer sicheren Website in eine *.cer*-Datei.
2. Kopieren Sie auf dem Computer, auf dem der BlackBerry MDS Connection Service installiert ist, die *.cer*-Datei in den Ordner *<Laufwerk:>/Programme/Java/<JRE-Version>/lib/security*.
3. Gehen Sie an der Eingabeaufforderung zu *<Laufwerk:>/Programme/Java/<JRE-Version>/bin*.
4. Geben Sie `keytool -import -trustcacerts -alias <Aliasname> -file <Zertifikatdateiname> -keystore cacerts` ein.

5. Geben Sie das Kennwort für den Schlüsselspeicher ein.
6. Bestätigen Sie mit *Yes* (Ja), um das Zertifikat zum Schlüsselspeicher hinzuzufügen.

Weitere Informationen über das Schlüsselwerkzeug finden Sie unter java.sun.com/j2se/1.5.0/docs/tooldocs/windows/keytool.html

20.13 Zulassen einer SSL-Verbindung vom MDS Connection Service zu einer Push-Anwendung

1. Klicken Sie in der Taskleiste auf *Start / Programs* (Programme) / *BlackBerry Enterprise Server / BlackBerry Server Configuration* (BlackBerry-Serverkonfiguration).
2. Konfigurieren Sie auf der Registerkarte *Mobile Data Service* (Mobiler Datendienst) die Schlüsselspeicherinformationen.

Es kann immer nur eine Schlüsselspeicherdatei vorhanden sein. Die Datei muss *webserver.keystore* heißen und sich im Verzeichnis *<Laufwerk:>/Programme/Research in Motion/BlackBerry Enterprise Server/MDS/webserver* befinden.

3. Klicken Sie auf *Create Keystore File* (Schlüsselspeicherdatei erstellen).
4. Klicken Sie bei einer entsprechenden Aufforderung auf *Yes* (Ja), um die vorhandene Schlüsselspeicherdatei zu überschreiben.
5. Klicken Sie auf *OK*.

20.13.1 Beschränken der Ressourcen, auf die Push-Anwendungen zugreifen können

Steuern Sie, welche Push-Anwendungen Inhalte an Endgeräte senden können, ohne dass die Inhalte zuerst von Benutzern angefragt werden müssen.

Mit Regeln für die Push-Zugriffssteuerung können Sie Benutzern und Push-Anwendungen Push-Regeln zuweisen, die steuern, welche Anwendungen Anfragen an Benutzer senden können.

20.13.2 Beschränken des Zugriffs der Push-Anwendung auf Ressourcen eines Enterprise Servers

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen MDS Connection Service.
2. Klicken Sie auf der Registerkarte *Connection Service* (Verbindungsdienst) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Access Control* (Zugriffssteuerung).

4. Führen Sie folgende Aktionen aus:

Tabelle 20.11: Beschränken des Zugriffs der Push-Anwendung auf Ressourcen

Aktion	Vorgehensweise
Beschränken des Zugriffs von Push-Anwendungen auf den BlackBerry MDS Connection Service, um Inhalte an Benutzer zu senden	1. Klicken Sie auf <i>Push Authentication</i> (Push-Authentifizierung). 2. Klicken Sie in der Dropdown-Liste auf <i>True</i> (Wahr).
Beschränken, dass Push-Anwendungen Inhalte an bestimmte BlackBerry-Geräte senden	1. Klicken Sie auf <i>Push Authorization</i> (Push-Berechtigung). 2. Klicken Sie in der Dropdown-Liste auf <i>True</i> (Wahr).
Verschlüsseln von Push-Anfragen mit SSL oder TLS.	1. Klicken Sie auf <i>Push Encryption</i> (Push-Verschlüsselung). 2. Klicken Sie in der Dropdown-Liste auf <i>True</i> (Wahr).

5. Klicken Sie auf OK.

20.13.3 Erstellen und Zuweisen einer Regel zu einer Push-Anwendung

Wenn MDS Services installiert sind, erstellen Sie einen Push-Initiator und ein Kennwort, damit MDS Services mit dem MDS Connection Service kommunizieren. Stellen Sie den Push-Initiator für die MDS Services bereit. Weitere Informationen finden Sie im Abschnitt 20.13.5 »Zuweisen eines Push-Initiators zu BlackBerry MDS Services«.

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Klicken Sie auf der Registerkarte *Global* auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie im linken Fensterbereich auf *Access Control* (Zugriffssteuerung).

4. Führen Sie folgende Aktionen aus:

Tabelle 20.12: Erstellen und Zuweisen einer Regel zu einer Push-Anwendung

Aktion	Vorgehensweise
Erstellen einer eindeutigen Push-Regel	1. Doppelklicken Sie auf <i>Push Rules</i> (Push-Regeln). 2. Klicken Sie auf <i>New</i> (Neu). 3. Doppelklicken Sie auf <i>Name</i>. 4. Geben Sie einen Namen für die Regel an. 5. Doppelklicken Sie auf <i>Description</i> (Beschreibung). 6. Geben Sie eine Beschreibung für die Regel an. 7. Klicken Sie auf <i>OK</i>. 8. Klicken Sie erneut auf <i>OK</i>.
Erstellen eines Push-Initiators für eine Push-Anwendung	1. Doppelklicken Sie auf <i>Push Initiators</i> (Push-Initiatoren). 2. Klicken Sie auf <i>New</i> (Neu). 3. Doppelklicken Sie auf <i>Push Principal Name</i> (Push-Hauptname). 4. Geben Sie den Namen der Anwendung ein, die die Push-Anfragen sendet, die von einer Push-Regel gesteuert werden. 5. Doppelklicken Sie auf <i>Credentials</i> (Anmeldeinformationen). 6. Geben Sie das Kennwort für die Anwendung ein. 7. Doppelklicken Sie auf <i>Description</i> (Beschreibung). 8. Geben Sie eine Beschreibung der Anwendung an. 9. Klicken Sie auf <i>OK</i>. 10. Klicken Sie erneut auf <i>OK</i>.
Zuweisen einer Push-Regel zu einem Push-Initiator	1. Doppelklicken Sie auf <i>Push Initiator Rules</i> (Push-Initiator-Regeln). 2. Klicken Sie im linken Fensterbereich auf eine Regel. 3. Wählen Sie im rechten Fensterbereich die Option für einen Push-Initiator aus. 4. Klicken Sie auf <i>OK</i>.

20.13.4 Zuweisen von Regeln zu Benutzerkonten oder Gruppen

1. Führen Sie im BlackBerry Manager im linken Fensterbereich eine der folgenden Aktionen aus:

Tabelle 20.13: Zuweisen von Regeln zu Benutzerkonten oder Gruppen

Aktion	Vorgehensweise
Zuweisen einer Push-Regel zu einem einzelnen Benutzerkonto	1. Klicken Sie auf <i>BlackBerry Domain</i> (BlackBerry-Domäne). 2. Klicken Sie auf der Registerkarte <i>Global</i> auf <i>Edit Properties</i> (Eigenschaften bearbeiten). 3. Klicken Sie im linken Fensterbereich auf <i>Access Control</i> (Zugriffssteuerung). 4. Doppelklicken Sie auf <i>User Rules</i> (Benutzerregeln). 5. Klicken Sie im linken Fensterbereich auf eine Regel. 6. Wählen Sie im rechten Fensterbereich die Option für ein Benutzerkonto. 7. Klicken Sie auf <i>OK</i>.
Zuweisen einer Push-Regel zu den Benutzern einer Gruppe	1. Klicken Sie auf eine Gruppe. 2. Klicken Sie auf der Registerkarte <i>Group Configuration</i> (Gruppenkonfiguration) auf <i>Edit Group Template</i> (Gruppenvorlage bearbeiten). 3. Klicken Sie auf <i>Access Control</i> (Zugriffssteuerung). 4. Doppelklicken Sie auf <i>Push Rule Set</i> (Push-Regel festlegen). 5. Markieren Sie das Kontrollkästchen für die Push-Regel, die Sie der Gruppe zuweisen möchten. 6. Klicken Sie auf <i>OK</i>. 7. Markieren Sie das Kontrollkästchen neben <i>Push Rule Set</i> (Push-Regel festlegen). 8. Klicken Sie auf <i>Reapply Template</i> (Vorlage erneut anwenden). 9. Klicken Sie auf <i>Yes</i> (Ja).

2. Klicken Sie auf *OK*.

20.13.5 Zuweisen eines Push-Initiators zu BlackBerry MDS Services

Fügen Sie den MDS Connection Service, für den eine Zugriffsrichtlinie für einen MDS Services-Push-Initiator definiert wurde, der Liste der MDS Connection Services hinzu, die für MDS Services verfügbar sind.

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen MDS Services Server.

2. Klicken Sie auf der Registerkarte *MDS Services* (MDS-Dienste) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Connection Service* (Verbindungsdienst).
4. Doppelklicken Sie auf *BlackBerry MDS Connection Service Definition* (Definition des MDS Connection Service).
5. Klicken Sie auf *New* (Neu).
6. Doppelklicken Sie auf *URL*.
7. Geben Sie den vollständigen URL oder den Domännennamen und die Port-Nummer für den MDS Connection Service ein.
8. Geben Sie im Feld *Push Initiator* (Push-Initiator) den Namen des MDS Services-Push-Initiators an.
9. Klicken Sie auf *OK*.
10. Klicken Sie erneut auf *OK*.

20.14 Verwalten von Push-Anwendungsanfragen

Der MDS Connection Service sendet Push-Anwendungsanfragen an Endgeräte. Konfigurieren Sie, wie der Enterprise Server Push-Anwendungsanfragen handhabt.

20.14.1 Zulassen der Übertragung anwendungsabhängiger Push-Anfragen zwischen den Endgeräten und dem MDS Connection Service an Geräte-Ports

Konfigurieren Sie den MDS Connection Service so, dass er anwendungsabhängige Push-Anfragen zwischen BlackBerry-Geräten und dem MDS Connection Service an Geräte-Ports zulässt.

Anwendungen, die mithilfe abhängiger Push-Anfragen den MDS Connection Service darüber informieren, ob eine Push-Anfrage auf dem Endgerät erfolgreich empfangen wurde, haben eindeutige Port-Nummern. Den für eine Anwendung definierten Port-Wert erhalten Sie vom jeweiligen Anwendungsentwickler.

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen MDS Connection Service.
2. Klicken Sie auf der Registerkarte *Connection Service* (Verbindungsdienst) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Push/PAP*.
4. Doppelklicken Sie auf *Device Ports Enabled for Reliable Pushes* (Für abhängige Pushes aktivierte Geräte-Ports).

5. Geben Sie die Port-Nummer des Geräts ein. Trennen Sie mehrere Port-Nummern durch Kommas.
6. Klicken Sie auf *OK*.
7. Klicken Sie auf *Restart Service* (Dienst neu starten).

20.14.2 Speichern von Push-Anwendungsanfragen in der BlackBerry Configuration Database

Wenn Push-Anfragen, die Ergebnisbenachrichtigungen verwenden, an eine Gruppe gesendet werden, bei der sich Benutzer auf mehreren Enterprise Servern innerhalb der BlackBerry-Domain befinden, müssen Sie die Push-Anfragen speichern.

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen MDS Connection Service.
2. Klicken Sie auf der Registerkarte *Connection Service* (Verbindungsdienst) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Push/PAP*.
4. Klicken Sie auf *Store Push Submissions* (Push-Einreichungen speichern).
5. Klicken Sie in der Dropdown-Liste auf *True* (Wahr).
6. Klicken Sie auf *OK*.
7. Klicken Sie auf *Restart Service* (Dienst neu starten).

20.14.3 Löschen von Push-Anfragen der BlackBerry Configuration Database

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
2. Klicken Sie auf der Registerkarte *Global* auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Push Control* (Push-Steuerung).

4. Führen Sie folgende Aktionen aus:

Tabelle 20.14: Löschen von Push-Anfragen der BlackBerry Configuration Database

Aktion	Vorgehensweise
Festlegen der maximalen Anzahl von in der Configuration Database gespeicherten Push-Nachrichten	1. Doppelklicken Sie auf <i>Maximum Stored Push Messages</i> (Maximal gespeicherte Push-Nachrichten). 2. Geben Sie eine Zahl ein.
Festlegen der maximalen Dauer (in Minuten), die eine Push-Nachricht gespeichert wird, bevor sie aus der BlackBerry Configuration Database gelöscht werden kann	1. Doppelklicken Sie auf <i>Maximum Push Message Age</i> (Maximales Alter einer Push-Nachricht). 2. Geben Sie eine Zahl ein.

5. Klicken Sie auf *OK*.

6. Klicken Sie auf *Restart Service* (Dienst neu starten).

20.14.4 Konfigurieren der Anzahl gleichzeitiger Push-Anwendungsanfragen, die der MDS Connection Service verarbeiten kann

Bestimmen Sie, wie viele aktive Push-Verbindungen der MDS Connection Service verarbeiten kann, bevor die Verbindungen in eine Warteschlange gestellt werden oder eine Nachricht über die Nichtverfügbarkeit des Dienstes an das Endgerät gesendet wird.

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen MDS Connection Service.
2. Klicken Sie auf der Registerkarte *Connection Service* (Verbindungsdienst) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Push/PAP*.
4. Führen Sie eine der folgenden Aktionen aus:

Tabelle 20.15: Konfigurieren der Anzahl gleichzeitiger Push-Anwendungsanfragen

Aktion	Vorgehensweise
Festlegen der maximalen Anzahl der Push-Verbindungen, die gleichzeitig verarbeitet werden, bevor Verbindungen in eine Warteschlange gestellt werden	1. Doppelklicken Sie auf <i>Maximum Number of Active Connections</i> (Maximale Anzahl aktiver Verbindungen). 2. Geben Sie eine Zahl ein.
Festlegen der maximalen Anzahl der Push-Verbindungen, die in die Warteschlange gestellt werden, bevor eine Nachricht über die Nichtverfügbarkeit des Dienstes an das BlackBerry-Gerät gesendet wird	1. Doppelklicken Sie auf <i>Maximum Number of Queued Connections</i> (Maximale Anzahl der in die Warteschlange aufgenommenen Verbindungen). 2. Geben Sie eine Zahl ein.

- 5. Klicken Sie auf *OK*.
- 6. Klicken Sie auf *Restart Service* (Dienst neu starten).

20.14.5 Manuelles Leeren der Push-Warteschlange

Die Push-Warteschlange wird täglich automatisch gelöscht. Sie können die Warteschlange aber auch manuell löschen.

- 1. Gehen Sie in Microsoft SQL Server Enterprise Manager zum Verzeichnis *Console Root/Microsoft SQLServers/SQL Server Group/<BlackBerry Configuration Database-Server>/Management/SQL ServerAgent/Jobs*.
- 2. Starten Sie den Prozess *RIMPurgeMDSMsg<Datenbankname>*.

20.14.6 Konfigurieren von Verbindungen zu BlackBerry-Geräten durch den MDS Connection Service

Bestimmen Sie, ob Endgeräte dauerhafte Verbindungen zum MDS Connection Service herstellen können und wie viele dauerhafte Verbindungen maximal zulässig sind. Ändern Sie die standardmäßigen LDAP-Port-Parameter nur im Falle eines Port-Konflikts mit einem anderen Dienst auf demselben Computer. Wenn Sie Port- oder Host-Angaben ändern, müssen Sie den MDS Connection Service neu starten, um die Konfigurationsangaben neu zu laden.

- 1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen MDS Connection Service.
- 2. Klicken Sie auf der Registerkarte *Connection Service* (Verbindungsdienst) auf *Edit Properties* (Eigenschaften bearbeiten).
- 3. Klicken Sie auf *General* (Allgemein).
- 4. Führen Sie eine der folgenden Aktionen aus:

Tabelle 20.16: Konfigurieren der Herstellung von Verbindungen zu BlackBerry-Geräten durch den BlackBerry MDS Connection Service

Aktion	Vorgehensweise
Festlegen der maximalen Datenmenge (in KB), die der MDS Connection Service an das Endgerät senden kann	1. Doppelklicken Sie auf <i>Maximum KB/Connection</i> (Maximale KB/Verbindung). 2. Geben Sie eine Zahl ein.
Festlegen der Dauer (in Millisekunden), nach der das Endgerät eine Bestätigung senden muss, bevor der MDS Connection Service alle anstehenden Inhalte für das Gerät verwirft	1. Doppelklicken Sie auf <i>Flow Control Timeout</i> (Flusskontroll-Timeout). 2. Geben Sie eine Zahl ein.

Aktion	Vorgehensweise
Zulassen der Herstellung dauerhafter TCP-Socket-Verbindungen zum MDS Connection Service durch Java-Anwendungen auf den Endgeräten	1. Doppelklicken Sie auf <i>Use Persistent Socket</i> (Dauerhaftes Socket verwenden). 2. Klicken Sie auf <i>True</i> (Wahr).
Festlegen der maximalen Anzahl der Threads, die der MDS Connection Service gleichzeitig verarbeiten kann, bevor er die Verarbeitung von Anfragen ablehnt	1. Doppelklicken Sie auf <i>Thread Pool Size</i> (Größe des Thread-Pools). 2. Geben Sie eine Zahl ein.
Festlegen der maximalen Anzahl dauerhafter TCP-Verbindungen, die gleichzeitig zwischen den Endgeräten und dem MDS Connection Service geöffnet sein können, bevor dieser die Verarbeitung von Anfragen ablehnt	1. Doppelklicken Sie auf <i>Maximum Simultaneous Persistent Sockets</i> (Maximale Anzahl gleichzeitiger dauerhafter Sockets). 2. Geben Sie eine Zahl ein.
Ändern des Ports, den der Webserver auf Anfragen von Push- Anwendungen überwacht Hinweis: Informieren Sie die Entwickler von Push-Anwendungen, wenn Sie diese Einstellung ändern.	1. Doppelklicken Sie auf <i>Web Server Listen Port</i> (Webserver überwacht Port). 2. Geben Sie die Port-Nummer ein.
Ändern des Ports, auf dem der Webserver HTTPS-Anfragen von Endgeräten empfängt	1. Doppelklicken Sie auf <i>Web Server SSL Listen Port</i> (Webserver überwacht SSL-Port). 2. Geben Sie die Port-Nummer ein.
Festlegen der Häufigkeit, mit der der MDS Connection Service die Configuration Database auf Veränderungen beim MDS Connection Service und an den Verwaltungseinstellungen für den Collaboration Service prüft.	1. Doppelklicken Sie auf <i>Admin Configuration Cycle Timer</i> (Zeitgeber für Administrationskonfigurationszyklus). 2. Geben Sie das Intervall ein.

21 Verwalten von Benutzerkonten

BlackBerry-Anwender sind in einem Unternehmen in der Regel in Benutzergruppen eingeteilt. Den einzelnen Gruppen können unterschiedliche IT-Richtlinien, Softwarekonfigurationen und Benutzerrechte zugewiesen werden. Auch einzelne Benutzerkonten lassen sich zwischen verschiedenen Benutzergruppen hin und her verschieben, zum Beispiel, wenn ein Mitarbeiter in eine andere Abteilung wechselt. Dieses Kapitel bietet einen Überblick über das Management der unterschiedlichen Anwender.

21.1 Verwalten von Benutzergruppen

Sie können Ausnahmen für Eigenschaften einer Gruppe festlegen, indem Sie die Eigenschaften eines einzelnen Benutzerkontos ändern, nachdem es einer Gruppe hinzugefügt wurde. Wenn es in einer Gruppe Ausnahmen bei den Benutzerkonteneigenschaften gibt und Sie die Gruppeneigenschaften ändern und verwenden, überschreiben die aktualisierten Gruppeneigenschaften alle Ausnahmen bei den Eigenschaften, die für einzelne Benutzerkonten festgelegt wurden.

Weitere Informationen zum Ändern von Eigenschaften einzelner Benutzerkonten finden Sie im Kapitel 13 über das Anpassen der BlackBerry-Nachrichtenübermittlung.

Wenn Sie ein Benutzerkonto aus einer Gruppe entfernen, ist das Benutzerkonto weiterhin in der globalen Benutzerliste enthalten, es wird jedoch nicht mehr in den Benutzergruppenlisten angezeigt.

21.1.1 Ändern von Gruppeneigenschaften

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *User Groups* (Benutzergruppen).
2. Klicken Sie auf der Registerkarte *User Groups List* (Benutzergruppenliste) auf eine Gruppe.
3. Klicken Sie auf *Edit Group Template* (Gruppenvorlage bearbeiten).
4. Ändern Sie die gewünschten Einstellungen.
5. Speichern Sie die Änderungen, indem Sie auf *Apply* (Anwenden) klicken.
6. Aktivieren Sie die Kontrollkästchen neben den zu ändernden Eigenschaften.

7. Klicken Sie auf *Reapply Template* (Vorlage erneut anwenden).
8. Klicken Sie auf *Yes* (Ja).
9. Klicken Sie auf *OK*.

21.1.2 Verwalten einer Gruppe

- Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *User Groups* (Benutzergruppen).
- Klicken Sie auf der Registerkarte *User Groups List* (Benutzergruppenliste) auf eine Gruppe.
- Klicken Sie auf *Group Admin* (Gruppenverwaltung).
- Führen Sie eine der folgenden Aktionen aus:

Tabelle 21.1: Verwalten von Benutzergruppen

Aktion	Vorgehensweise
Umbenennen einer Gruppe	1. Klicken Sie auf <i>Modify Group Definition</i> (Gruppendefinition ändern). 2. Geben Sie im Feld <i>Group Name</i> (Gruppenname) einen neuen Namen ein. 3. Klicken Sie auf <i>OK</i>.
Löschen einer Gruppe	1. Klicken Sie auf <i>Delete Group</i> (Gruppe löschen). 2. Klicken Sie auf <i>Yes</i> (Ja).
Verschieben einer Gruppe auf einen anderen BlackBerry Enterprise Server (BES)	1. Klicken Sie auf <i>Move Group to BES</i> (Gruppe auf BES verschieben). 2. Klicken Sie auf den Enterprise Server, der als Zielserver gilt. 3. Klicken Sie auf <i>OK</i>. 4. Klicken Sie auf <i>Yes</i> (Ja).

21.2 Verwalten von Benutzern

Sie können Benutzerkonten innerhalb der BlackBerry-Domain zwischen Benutzergruppen oder von einem Enterprise Server auf einen anderen verschieben. Neue Servicebücher werden über das drahtlose Netzwerk an das Endgerät gesendet.

Wenn Sie auf dem Nachrichtenserver den Anzeigenamen eines Benutzerpostfachs verschieben oder ändern, aktualisiert der Enterprise Server das Benutzerkonto innerhalb von 15 Minuten. Wenn Sie ein verstecktes Postfach verschieben, das in der globalen Adressenliste (GAL) nicht angezeigt wird, müssen Sie das Benutzerkonto auf dem Enterprise Server manuell aktualisieren.

Wenn Sie ein Benutzerkonto vom Enterprise Server entfernen, können die BlackBerry-Informationen des Benutzers in seinem Postfach verbleiben. Auf diese Weise können Sie Benutzerkonten erneut hinzufügen oder es den Benutzern ermöglichen, ihr Endgerät weiterhin als BlackBerry-Desktop-Redirector-Benutzer zu verwenden. Wenn Sie ein Benutzerkonto, dessen BlackBerry-Informationen beibehalten wurden, erneut hinzufügen, kann der Benutzer sein Gerät weiterhin mit der Konfiguration und den Berechtigungen verwenden, die vor dem Entfernen für das Benutzerkonto galten.

21.2.1 Verschieben und Löschen eines Benutzerkontos

- Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen BlackBerry Enterprise Server.
- Klicken Sie auf der Registerkarte *Users* (Benutzer) auf ein Benutzerkonto.
- Klicken Sie auf *Account* (Konto).
- Führen Sie eine der folgenden Aktionen aus:

Tabelle 21.2: Verschieben und Löschen eines Benutzerkontos

Aktion	Vorgehensweise
Verschieben eines Benutzerkontos in eine andere Gruppe	1. Klicken Sie auf <i>Assign To Group</i> (Zu Gruppe zuweisen). 2. Klicken Sie auf eine Gruppe, in die das Benutzerkonto verschoben werden soll. 3. Klicken Sie auf <i>OK</i>.
Entfernen eines Benutzerkontos aus einer Gruppe	1. Klicken Sie auf <i>Remove From Group</i> (Aus Gruppe entfernen). 2. Klicken Sie auf <i>Yes</i> (Ja).
Verschieben eines Benutzerkontos auf einen anderen BlackBerry Enterprise Server	1. Klicken Sie auf <i>Move User</i> (Benutzer verschieben). 2. Klicken Sie auf den Enterprise Server, der als Zielsever gilt. 3. Klicken Sie auf <i>OK</i>.
Entfernen eines Benutzerkontos vom BlackBerry Enterprise Server	1. Klicken Sie auf <i>Delete User</i> (Benutzer löschen). 2. Klicken Sie auf <i>Yes</i> (Ja). 3. Führen Sie eine der folgenden Aktionen aus: • Klicken Sie auf <i>No</i> (Nein), wenn die BlackBerry-Informationen im Postfach eines Benutzers verbleiben sollen. • Klicken Sie auf <i>Yes</i> (Ja), wenn die BlackBerry-Informationen aus dem Postfach eines Benutzers entfernt werden sollen.

21.2.2 Manuelles Aktualisieren eines Benutzerkontos

- Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
- Klicken Sie auf der Registerkarte *Users* (Benutzer) auf ein Benutzerkonto.
- Klicken Sie auf *Account* (Konto).
- Klicken Sie auf *Reload User* (Benutzer neu laden).
- Klicken Sie auf *OK*.

22 Verwalten von BlackBerry-Gerätesoftware und drahtlosen Anwendungen

Software wächst. Neue Funktionen werden hinzugefügt, Fehler werden behoben. Das gilt auch für das BlackBerry-Betriebssystem und die darauf laufenden Anwendungen. Die BlackBerry-Plattform bietet die Möglichkeit, die Softwarewartung zentralisiert vorzunehmen, also vorhandene Software zu aktualisieren und obsolete zu entfernen. Dieses Kapitel bietet einen Überblick über die damit verbundenen Optionen.

22.1 Verwalten von Anwendungen auf BlackBerry-Geräten

Sie können Java-Anwendungen, die Nachrichtenübermittlungsanwendung und MDS Runtime von den Endgeräten aus über ein drahtloses Netzwerk aktualisieren oder entfernen. Es kann aber bis zu vier Stunden dauern, diese Anwendungen von den Geräten aus durch den Enterprise Server zu aktualisieren oder zu entfernen.

Sie können Richtlinien zur Anwendungssteuerung aktualisieren, um den Zugriff der auf den Geräten installierten Anwendungen auf die Endgeräte und Ressourcen hinter der Unternehmensfirewall zu ändern. Außerdem können Sie diese Richtlinien entfernen, wenn Sie sie nicht mehr benötigen.

Sie allein sind für die Auswahl, Implementierung und Leistung von Drittanbieteranwendungen verantwortlich, die Sie zusammen mit der Gerätesoftware oder Desktopsoftware verwenden. RIM übernimmt keinerlei Verantwortung oder Garantie für die Sicherheit, Kompatibilität, Leistung oder Vertrauenswürdigkeit von Anwendungen von Drittanbietern. Ferner übernimmt RIM keinerlei Haftung für Probleme, die Ihnen oder Dritten durch die Verwendung solcher Anwendungen von Drittanbietern entstehen.

22.1.1 Aktualisieren einer Anwendung auf einem Endgerät

- Fügen Sie die Anwendung auf dem Netzwerklaufwerk hinzu, oder aktualisieren Sie sie. Weitere Informationen finden Sie in Kapitel 11.1.

- Indizieren Sie die Anwendung erneut. Weitere Informationen finden Sie ebenfalls im Kapitel 11.1.

Hinweis: Anwendungen, denen eine Richtlinie zur Anwendungssteuerung mit einer auf *Required* (Erforderlich) gesetzten Disposition zugewiesen wurde, empfangen die Anwendungsaktualisierung über das drahtlose Netzwerk.

22.1.2 Entfernen einer Anwendung von einem Endgerät

- Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne)
- Klicken Sie auf der Registerkarte *Software Configurations* (Softwarekonfigurationen) auf *Manage Application Policies* (Anwendungsrichtlinien verwalten).
- Doppelklicken Sie auf eine Richtlinie zur Anwendungssteuerung.
- Klicken Sie in der Dropdown-Liste *Disposition* (Disposition) auf *Disallowed* (Verhindern).
- Klicken Sie auf *OK*.

22.1.3 Ändern oder Löschen einer Richtlinie zur Anwendungssteuerung

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne)
2. Klicken Sie auf die Registerkarte *Software Configurations* (Softwarekonfigurationen).
3. Klicken Sie auf *Manage Application Policies* (Anwendungsrichtlinien verwalten).
4. Klicken Sie auf die Anwendungsrichtlinie.
5. Führen Sie eine der folgenden Aktionen aus:

Tabelle 22.1: Ändern oder Löschen einer Richtlinie zur Anwendungssteuerung

Aktion	Vorgehensweise
Ändern einer Richtlinie zur Anwendungssteuerung	1. Klicken Sie auf <i>Properties</i> (Eigenschaften). 2. Ändern der Eigenschaften einer Richtlinie zur Anwendungssteuerung 3. Klicken Sie auf <i>OK</i> .
Löschen einer Richtlinie zur Anwendungssteuerung	Klicken Sie auf <i>Remove</i> (Entfernen).

6. Klicken Sie auf *OK*.

22.2 Verwalten von Softwarekonfigurationen

Softwarekonfigurationen werden auf dem Computer verwaltet, auf dem der BlackBerry Manager installiert ist. Sie können eine Softwarekonfiguration ändern, um Anwendungen, die auf Endgeräten installiert werden sollen, zu aktualisieren oder zu ändern. Außerdem können Sie Benutzern eine andere Softwarekonfiguration zuweisen.

22.2.1 Anpassen einer Softwarekonfiguration

Führen Sie im BlackBerry Manager im linken Fensterbereich eine der folgenden Aktionen aus:

Tabelle 22.2: Verwalten einer Softwarekonfiguration

Aktion	Vorgehensweise
Ändern von Softwarekonfigurationen	1. Klicken Sie auf <i>BlackBerry Domain</i> (BlackBerry-Domäne). 2. Klicken Sie auf der Registerkarte <i>Software Configurations</i> (Softwarekonfigurationen) in der Liste <i>Configuration Name</i> (Konfigurationsname) auf eine Softwarekonfiguration. 3. Klicken Sie auf <i>Edit Configuration</i> (Konfiguration bearbeiten). 4. Führen Sie in der Liste <i>Application Name</i> (Anwendungsname) eine der folgenden Aktionen aus: • Aktivieren Sie das Kontrollkästchen neben den Anwendungen, die auf BlackBerry-Geräten zu installieren sind. • Deaktivieren Sie das Kontrollkästchen neben den Anwendungen, die von BlackBerry-Geräten zu entfernen sind. 5. Klicken Sie auf <i>OK</i>.
Zuweisen anderer Softwarekonfigurationen zu Benutzern	1. Klicken Sie auf einen Enterprise Server. 2. Klicken Sie in der Liste <i>Users</i> (Benutzer) auf einen Benutzer, dem die Softwarekonfiguration zugewiesen werden soll. 3. Klicken Sie auf <i>Device Management</i> (Geräteverwaltung). 4. Klicken Sie auf <i>Assign Software Configuration</i> (Softwarekonfiguration zuweisen). 5. Klicken Sie auf eine Softwarekonfiguration. 6. Klicken Sie auf <i>OK</i>.

Aktion	Vorgehensweise
Entfernen von Softwarekonfigurationen von Benutzern	1. Klicken Sie auf einen Enterprise Server. 2. Klicken Sie in der Liste <i>Users</i> (Benutzer) auf einen Benutzer, dem die Softwarekonfiguration zugewiesen werden soll. 3. Klicken Sie auf <i>Device Management</i> (Geräteverwaltung). 4. Klicken Sie auf <i>Assign Software Configuration</i> (Softwarekonfiguration zuweisen). 5. Klicken Sie auf <i>none</i> (Keine). 6. Klicken Sie auf <i>OK</i>.
Löschen einer Softwarekonfiguration	1. Klicken Sie auf <i>BlackBerry Domain</i> (BlackBerry-Domäne). 2. Klicken Sie auf der Registerkarte <i>Software Configurations</i> (Softwarekonfigurationen) in der Liste <i>Configuration Name</i> (Konfigurationsname) auf eine Softwarekonfiguration. 3. Klicken Sie auf <i>Delete Configuration</i> (Konfiguration löschen). 4. Klicken Sie auf <i>OK</i>.
Erstellen einer neuen Softwarekonfiguration unter Verwendung einer bereits vorhandenen Softwarekonfiguration	1. Klicken Sie auf <i>BlackBerry Domain</i> (BlackBerry-Domäne). 2. Klicken Sie auf der Registerkarte <i>Software Configurations</i> (Softwarekonfigurationen) in der Liste <i>Configuration Name</i> (Konfigurationsname) auf eine Softwarekonfiguration. 3. Klicken Sie auf <i>Copy Configuration</i> (Konfiguration kopieren). 4. Doppelklicken Sie auf die kopierte Softwarekonfiguration. 5. Geben Sie der Softwarekonfiguration im Feld <i>Configuration Name</i> (Konfigurationsname) einen neuen Namen. 6. Nehmen Sie die gewünschten Änderungen an den Eigenschaften der Softwarekonfiguration vor. Weitere Informationen finden Sie im Abschnitt »Erstellen von Softwarekonfigurationen« 7. Klicken Sie auf <i>OK</i>.

23 Verwalten einer BlackBerry-Domain

Von einer BlackBerry-Domain spricht man, wenn mehrere Instanzen des Enterprise Servers ein und dieselbe Konfigurationsdatenbank (Configuration Database) verwenden. Auch eine solche Domain lässt sich verwalten, überwachen und aktualisieren. Dieses Kapitel gibt dazu wertvolle Tipps.

23.1 Überwachen der BlackBerry-Dienste und -komponenten in einer Domain

Bei fehlgeschlagenen Operationen ermittelt der BlackBerry-Controller standardmäßig die entsprechenden Prozesse und startet sie neu, wodurch der Enterprise Server im Fall nicht reagierender Threads oder inaktiver Dienste weiter funktionieren kann.

Der BlackBerry-Controller überwacht die folgenden BlackBerry-Services (BlackBerry-Dienste) und -komponenten:

- BlackBerry-Dispatcher,
- BlackBerry-Router,
- BlackBerry Messaging Agent,
- BlackBerry Attachment Service,
- BlackBerry Collaboration Service,
- Live Communications Server Connector,
- BlackBerry Synchronization Service,
- BlackBerry Policy Service,
- BlackBerry MDS Connection Service,
- BlackBerry MDS Services,
- BlackBerry-Datenbankkonsistenz-Dienst.

Standardmäßig werden die Registrierungsschlüssel, die den BlackBerry-Controller steuern, nicht angezeigt. Um festzulegen, wie der Controller die Dienste überwacht, müssen Sie die Registrierungsschlüssel erstellen, die den BlackBerry-Controller steuern, und die Standardwerte ändern.

Warnung: Starten Sie den BlackBerry-Controller nicht neu. Bei einem Neustart des Controllers werden auch die Messaging Agents neu gestartet, was einige Zeit in Anspruch nehmen kann. Während des Neustarts der Messaging Agents können die Benutzer auf den Endgeräten weder Nachrichten senden noch empfangen.

23.2 Anpassen der Überwachung der Dienste durch den BlackBerry-Controller

- Starten Sie auf dem Computer, auf dem der BlackBerry-Dienst installiert ist, den Registrierungseditor *regedit*.
- Wechseln Sie im linken Fensterbereich zu *HKEY_LOCAL_MACHINE/Software/Research In Motion/BlackBerry Enterprise Server*.
- Klicken Sie auf *Controller*.
- Führen Sie eine der folgenden Aktionen aus:

Tabelle 23.1: Anpassen der Überwachung der BlackBerry-Dienste durch den BlackBerry-Controller

Aktion	Vorgehensweise	Standard
Kein Neustarten der Messaging Agents, falls sie nicht reagieren	1. Erstellen Sie einen neuen DWORD-Wert namens <i>RestartAgentsOnCrash</i> (Agenten bei Systemausfall neu starten). 2. Doppelklicken Sie auf den neuen Wert. 3. Geben Sie in das Feld <i>Value data</i> (Datenwerte) 0 ein.	1
Festlegen der maximalen Anzahl von Neustarts der Messaging Agents pro Tag	1. Erstellen Sie einen neuen DWORD-Wert namens <i>MaxAgentRestartsPerDay</i> (Maximale Anzahl an Neustarts für den Agenten pro Tag). 2. Doppelklicken Sie auf den neuen Wert. 3. Geben Sie im Feld <i>Value data</i> (Datenwerte) eine Zahl ein.	10

Aktion	Vorgehensweise	Standard
Eingeben eines Limits für die Anzahl von verpassten Funktionsprüfungen, die der Controller zulässt, bevor er die Messaging Agents neu startet	1. Erstellen Sie einen neuen DWORD-Wert namens <i>WaitToRestartAgentOnHung</i> (Auf Neustart des Agenten bei Blockierung warten). 2. Doppelklicken Sie auf den neuen Wert. 3. Geben Sie im Feld <i>Value Data</i> (Datenwerte) eine Zahl größer 4 ein, damit dem Controller vor dem Neustart der Messaging Agents ausreichend Zeit zum Überwachen von Thread-Funktionsprüfungen zur Verfügung steht. Funktionsprüfungen finden alle 10 Minuten statt. Wenn die Funktionsprüfung keine Antwort vom zu überwachenden Thread erhält, wird die verpasste Funktionsprüfung in der Protokolldatei des BlackBerry Messaging Agent als Wert für die Wartezeit verzeichnet. Beispiel: <pre>[20148] (05/12 12:21:00):{0xC28} Thread: *** No Response *** Thread Id=0xB00, Handle=0x558, WaitCount=2,</pre>	6
Kein Neustarten der Messaging Agents, wenn der BlackBerry-Controller nicht reagierende Threads entdeckt	1. Erstellen Sie einen neuen DWORD-Wert namens <i>WaitToRestartAgentOnHung</i> (Auf Neustart des Agenten bei Blockierung warten). 2. Doppelklicken Sie auf den neuen Wert. 3. Geben Sie 0 ein.	6
Kein Neustarten der Messaging Agents innerhalb eines bestimmten Zeitraums, wenn der BlackBerry-Controller einen nicht reagierenden Thread entdeckt	1. Erstellen Sie die folgenden DWORD-Werte: • <i>RestartAgentOnHungBlackoutFrom</i> (Neustart nach Blockierung des Agenten von) • <i>RestartAgentOnHungBlackoutTo</i> (Neustart nach Blockierung des Agenten bis) 2. Wählen Sie bei jedem neuen Wert die Option <i>Decimal</i> (Dezimal) aus. 3. Geben Sie unter <i>RestartAgentOnHungBlackoutFrom</i> (Neustart nach Blockierung des Agenten von) die Untergrenze des Zeitraums an. Die Werte reichen von 0 bis 23, wobei 0 für 0:00 Uhr und 23 für 23:00 Uhr steht. 4. Geben Sie unter <i>RestartAgentOnHungBlackoutTo</i> (Neustart nach Blockierung des Agenten bis) die Obergrenze des Zeitraums an. Die Werte reichen von 0 bis 23, wobei 0 für 0:00 Uhr und 23 für 23:00 Uhr steht.	-

Aktion	Vorgehensweise	Standard
	Wenn zum Beispiel der Wert für <i>RestartAgentOnHungBlackoutFrom</i> (Neustart nach Blockierung des Agenten von) auf 8 und der Wert für <i>RestartAgentOnHungBlackoutTo</i> (Neustart nach Blockierung des Agenten bis) auf 17 gesetzt ist, startet der BlackBerry Controller die BlackBerry Messaging Agents zwischen 8:00 Uhr und 17:00 Uhr nicht neu.	
Deaktivieren des Zeitraums, in dem der BlackBerry-Controller die Messaging Agents im Falle eines nicht reagierenden Threads nicht neu starten darf	1. Doppelklicken Sie auf <i>RestartAgentOnHungBlackoutFrom</i> (Neustart nach Blockierung des Agenten von). 2. Geben Sie in das Feld <i>Value data</i> (Datenwerte) 0 ein. 3. Klicken Sie auf <i>OK</i>. 4. Doppelklicken Sie auf <i>RestartAgentOnHungBlackoutTo</i> (Neustart nach Blockierung des Agenten bis). 5. Geben Sie in das Feld <i>Value data</i> (Datenwerte) 0 ein.	-
Neustarten der Messaging Agents ohne Erstellen einer Datei <i>user.dmp</i>, wenn der BlackBerry-Controller nicht reagierende Threads ermittelt	1. Erstellen Sie einen neuen DWORD-Wert namens <i>RestartAgentOnHung</i> (Neustart nach Blockierung des Agenten). 2. Doppelklicken Sie auf den neuen Wert. 3. Geben Sie in das Feld <i>Value data</i> (Datenwerte) 0 ein. Der Wert <i>WaitToRestartAgentOnHung</i> (Auf Neustart nach Blockierung des Agenten warten) überschreibt diesen Wert. Um diese Option zur Datenerfassung zu verwenden, laden Sie die Anwendung »User Mode Process Dump« der Microsoft OEM Support Tools herunter und installieren sie.	1
Festlegen der Höchstzahl an <i>user.dmp</i>-Dateien, die täglich pro Enterprise Server erstellt werden, bevor der BlackBerry-Controller die Messaging Agents neu startet	1. Erstellen Sie einen neuen DWORD-Wert namens <i>MaxUserDumpPerDay</i> (Maximale Anzahl an Userdumps pro Tag). 2. Doppelklicken Sie auf den neuen Wert. 3. Geben Sie im Feld <i>Value data</i> (Datenwerte) eine Zahl ein. Um diese Option zur Datenerfassung zu verwenden, laden Sie die Anwendung »User Mode Process Dump« der Microsoft OEM Support Tools herunter und installieren sie. Weitere Informationen finden Sie unter www.support.microsoft.com.	3

Aktion	Vorgehensweise	Standard
Festlegen der Anzahl an 10-Minuten-Intervallen, in denen die Messaging Agents neu gestartet werden sollen, wenn der BlackBerry-Controller keine Funktionsprüfungen von den Messaging Agents erhält	1. Erstellen Sie einen neuen DWORD-Wert namens <i>MissedHeartbeatThreshold</i> (Schwellenwert für verpasste Heartbeats). 2. Doppelklicken Sie auf den neuen Wert. 3. Geben Sie im Feld <i>Value data</i> (Datenwerte) eine Zahl ein. Funktionsprüfungen finden alle 10 Minuten statt. Beispiel: Wenn der Wert für <i>MissedHeartbeatThreshold</i> (Schwellenwert für verpasste Heartbeats) auf 3 gesetzt ist, startet der BlackBerry-Controller die Messaging Agents erst nach 30 Minuten.	2
Kein Neustarten der Messaging Agents, wenn der BlackBerry-Controller keine Funktionsprüfungen von den Messaging Agents erhält	1. Erstellen Sie einen neuen DWORD-Wert namens <i>MissedHeartbeatThreshold</i> (Schwellenwert für verpasste Heartbeats). 2. Doppelklicken Sie auf den neuen Wert. 3. Geben Sie in das Feld <i>Value data</i> (Datenwerte) 0 ein.	-
Kein Neustarten des BlackBerry-Dispatchers, wenn er nicht mehr reagiert	1. Erstellen Sie einen neuen DWORD-Wert namens <i>RestartDispatcherOnCrash</i> (Dispatcher nach Absturz neu starten). 2. Doppelklicken Sie auf den neuen Wert. 3. Geben Sie in das Feld <i>Value data</i> (Datenwerte) 0 ein.	1
Kein Neustarten des BlackBerry Collaboration Service, wenn er nicht mehr reagiert	1. Erstellen Sie einen neuen DWORD-Wert namens <i>RestartBBIMOnCrash</i> (BBIM nach Absturz neu starten). 2. Doppelklicken Sie auf den neuen Wert. 3. Geben Sie in das Feld <i>Value data</i> (Datenwerte) 0 ein.	1
Kein Neustarten des Live Communication Servers, wenn er nicht mehr reagiert	1. Erstellen Sie einen neuen DWORD-Wert namens <i>RestartLCSOnCrash</i> (LCS nach Absturz neu starten). 2. Doppelklicken Sie auf den neuen Wert. 3. Geben Sie in das Feld <i>Value data</i> (Datenwerte) 0 ein.	1
Kein Neustarten des BlackBerry-Routers, wenn er nicht mehr reagiert	1. Erstellen Sie einen neuen DWORD-Wert namens <i>RestartRouterOnCrash</i> (Router nach Absturz neu starten). 2. Doppelklicken Sie auf den neuen Wert. 3. Geben Sie in das Feld <i>Value data</i> (Datenwerte) 0 ein.	1

Aktion	Vorgehensweise	Standard
Kein Neustarten des BlackBerry Policy Service, wenn er nicht mehr reagiert	1. Erstellen Sie einen neuen DWORD-Wert namens <i>RestartPolicyServerOnCrash</i> (Richtlinienserver nach Absturz neu starten). 2. Doppelklicken Sie auf den neuen Wert. 3. Geben Sie in das Feld <i>Value data</i> (Datenwerte) 0 ein.	1
Kein Neustarten des BlackBerry Synchronization Service, wenn er nicht mehr reagiert	1. Erstellen Sie einen neuen DWORD-Wert namens <i>RestartSyncServerOnCrash</i> (Synchronisierungsserver nach Absturz neu starten). 2. Doppelklicken Sie auf den neuen Wert. 3. Geben Sie in das Feld <i>Value data</i> (Datenwerte) 0 ein.	1
Kein Neustarten des MDS Connection Service, wenn er nicht mehr reagiert	1. Erstellen Sie einen neuen DWORD-Wert namens <i>RestartMDSOnCrash</i> (MDS nach Absturz neu starten). 2. Doppelklicken Sie auf den neuen Wert. 3. Geben Sie in das Feld <i>Value data</i> (Datenwerte) 0 ein.	1
Kein Neustarten des BlackBerry Attachment Service, wenn er nicht mehr reagiert	1. Erstellen Sie einen neuen DWORD-Wert namens <i>RestartAttachmentServerOnCrash</i> (Anlagenserver nach Absturz neu starten). 2. Doppelklicken Sie auf den neuen Wert. 3. Geben Sie in das Feld <i>Value data</i> (Datenwerte) 0 ein.	1
Kein Neustarten von MDS-Services, falls es nicht reagiert	1. Erstellen Sie einen neuen DWORD-Wert namens <i>RestartMDSServicesOnCrash</i> (MDS-Services nach Absturz neu starten). 2. Doppelklicken Sie auf den neuen Wert. 3. Geben Sie in das Feld <i>Value data</i> (Datenwerte) 0 ein.	1
Kein Neustarten des BlackBerry-Datenbankkonsistenz-Dienstes, wenn er nicht mehr reagiert	1. Erstellen Sie einen neuen DWORD-Wert namens <i>RestartDBConsistencyOnCrash</i> (DB-Konsistenz nach Absturz neu starten). 2. Doppelklicken Sie auf den neuen Wert. 3. Geben Sie in das Feld <i>Value data</i> (Datenwerte) 0 ein.	1

- Klicken Sie auf OK.

23.3 Zugriff auf Protokolldateien für BlackBerry-Dienste

Mit Protokolldateien können Sie die täglichen Aktivitäten der BlackBerry-Dienste überwachen und bei der Problembehebung von BlackBerry-Diensten Fehler oder Informationen ermitteln. Jeder Dienst erstellt eine eigene Protokolldatei. Standardmäßig werden die Protokolldateien in das Verzeichnis *C:\Programme\Research In Motion\BlackBerry Enterprise Server\Logs* geschrieben und vom Enterprise Server in täglichen Ordnern organisiert. Sie können den Speicherort der Protokolldateien ändern.

Standardmäßig erstellen die BlackBerry-Dienste Protokolldateien im Format *<Servername_Kennung_Instance_JJJJMMTT_Protokollnummer.txt>* (zum Beispiel: *BBServer01_MAGT_01_20051020_0001.txt*). Für die von den BlackBerry-Diensten in die Protokolldatei geschriebenen Ereignisse wird eine fünfstellige Zahl verwendet (zum Beispiel 30126). Die erste Ziffer stellt die Protokollebene dar.

Mit Protokollen können Sie überwachen, wann und wie oft Benutzer PIN- und SMS-Nachrichten versenden und von den Endgeräten aus Anrufe tätigen. Standardmäßig ist auf dem Enterprise Server die Anrufprotokollierung aktiviert und die Protokollierung von PIN- und SMS-Nachrichten deaktiviert.

23.3.1 Anpassen der Erstellung von Protokolldateien durch die BlackBerry-Dienste

1. Klicken Sie in der Taskleiste auf *Start / Programs* (Programme) / *BlackBerry Enterprise Server / BlackBerry Server Configuration* (BlackBerry-Serverkonfiguration).
2. Führen Sie auf der Registerkarte *Logging* (Protokollierung) eine der folgenden Aktionen aus:

Tabelle 23.2: Anpassen der Erstellung von Protokolldateien

Aktion	Vorgehensweise
Festlegen des Stammverzeichnisses, in das die BlackBerry-Dienste die Protokolldateien schreiben	1. Klicken Sie auf <i>Browse</i> (Durchsuchen). 2. Gehen Sie zu einem Speicherort auf einem lokalen Datenträger.
Festlegen eines Präfix für alle Protokolldateien	Geben Sie im Feld <i>Log file prefix</i> (Präfix der Protokolldatei) ein Präfix ein.
Speichern aller Protokolldateien im Stammverzeichnis	Deaktivieren Sie das Kontrollkästchen <i>Create daily log folder</i> (Täglichen Protokollordner erstellen).

3. Klicken Sie im Bereich mit den Protokolleinstellungen des BlackBerry-Dienstes auf einen BlackBerry-Dienst.
4. Führen Sie eine der folgenden Aktionen aus:

Aktion	Vorgehensweise
Ändern der aus vier Buchstaben bestehenden Kennung im Namen der Protokolldatei des BlackBerry-Dienstes	1. Klicken Sie auf <i>Debug log identifier</i> (Kennung des Debugging-Protokolls). 2. Geben Sie in der Spalte <i>Setting</i> (Einstellung) eine neue Kennung ein, um dem Dienst die Protokolldatei zuzuordnen, in die er schreibt.
Kein Erstellen täglich neuer Protokolldateien	1. Klicken Sie auf <i>Debug daily log file</i> (Tägliche Debugging-Protokolldatei) 2. Klicken Sie in der Dropdown-Liste der Spalte <i>Setting</i> (Einstellung) auf <i>No</i> (Nein). Dies bedeutet, dass der Name der Protokolldatei nicht das Datum enthält.
Festlegen der Protokollebene	1. Klicken Sie auf die Einstellung <i>Debug log level</i> (Debugging-Protokollebenen). 2. Klicken Sie in der Dropdown-Liste der Spalte <i>Setting</i> (Einstellung) auf eine der folgenden Protokollebenen: • 1: <i>Error</i> (Fehler) • 2: <i>Warning</i> (Warnung) • 3: <i>Information</i> (Informationen). Hiermit können Sie die täglichen Aktivitäten des BlackBerry-Dienstes überwachen. • 4: <i>Debug</i> (Debugging). Hierdurch erhalten Sie zusätzliche Informationen zur Fehlerbehebung des Dienstes. • 5: <i>Verbose</i> (Ausführlich). In diesem Fall werden alle Ereignisse im Zusammenhang mit dem Dienst oder der Komponente protokolliert.
Festlegen der maximalen Größe der Datei	1. Klicken Sie auf <i>Debug log size</i> (Größe des Debugging-Protokolls). 2. Geben Sie in der Spalte <i>Setting</i> (Einstellung) die maximale Größe der Datei in MB an. Bei einem Wert von 0 ist die Größe unbeschränkt. Wenn Sie die Option <i>Debug log auto-roll</i> (Automatische Übertragung des Debugging-Protokolls) aktivieren, wird eine neue Datei erstellt, sobald die Dateigröße den maximalen Wert erreicht. Wenn diese Option deaktiviert ist, wird die vorhandene Datei überschrieben.
Erstellen einer neuen Protokolldatei, wenn der BlackBerry-Dienst neu gestartet wird oder die Größe der Protokolldatei den Maximalwert erreicht	1. Klicken Sie auf <i>Debug log auto-roll</i> (Automatische Übertragung des Debugging-Protokolls). 2. Klicken Sie in der Dropdown-Liste der Spalte <i>Setting</i> (Einstellung) auf <i>Yes</i> (Ja).

Aktion	Vorgehensweise
Festlegen des Zeitraums, nach dem Protokolldateien gelöscht werden	1. Klicken Sie auf <i>Debug log maximum daily file age</i> (Maximales Alter der täglichen Debugging-Protokolldatei). 2. Geben Sie in der Spalte <i>Setting</i> (Einstellung) die Anzahl der Tage an, nach denen Protokolldateien gelöscht werden. Bei einem Wert von 0 ist die Zeitdauer unbeschränkt.
Wiederherstellen der standardmäßigen Protokolleinstellungen für alle aufgeführten BlackBerry-Dienste	Klicken Sie auf die Schaltfläche <i>Reset All</i> (Alle zurücksetzen).

5. Klicken Sie auf *OK*.
6. Starten Sie den BlackBerry-Dienst auf dem Computer, auf dem er installiert wurde, erneut in den Windows-Diensten.

23.3.2 Anpassen der Erstellung einer Protokolldatei durch den BlackBerry MDS Connection Service

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Connection Services* (Verbindungsdienste) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Logs* (Protokolle).
4. Führen Sie eine der folgenden Aktionen aus:

Tabelle 23.3: Anpassen der Erstellung einer Protokolldatei durch den BlackBerry MDS Connection Service

Aktion	Vorgehensweise
Überwachen der Aktivität auf der SRP-(Server Relay Protocol-)Netzwerkebene	1. Klicken Sie auf <i>SRP logging enabled</i> (SRP-Protokollierung aktiviert). 2. Klicken Sie auf <i>True</i> (Wahr).
Überwachen der Aktivität auf der IPPP-Netzwerkebene	1. Klicken Sie auf <i>IPPP logging enabled</i> (IPPP-Protokollierung aktiviert). 2. Klicken Sie auf <i>True</i> (Wahr).
Überwachen der Aktivität auf der UDP-Netzwerkebene	1. Klicken Sie auf <i>UDP logging enabled</i> (UDP-Protokollierung aktiviert). 2. Klicken Sie auf <i>True</i> (Wahr).

Aktion	Vorgehensweise
Überwachen der Aktivität auf der GME- (Gateway Message Envelope-)Netzwerkebene	1. Klicken Sie auf <i>GME logging enabled</i> (GME-Protokollierung aktiviert). 2. Klicken Sie auf <i>True</i> (Wahr).
Überwachen von HTTP-Headern für vom Webserver gesendete Antworten, wenn die Benutzer mit dem Endgerät Inhalte aus dem Internet und dem Intranet abrufen	1. Klicken Sie auf <i>HTTP logging enabled</i> (HTTP-Protokollierung aktiviert). 2. Klicken Sie auf <i>True</i> (Wahr).
Überwachen von HTTP-Headern und Texten der vom Webserver gesendeten Antworten, wenn die Benutzer mit dem Endgerät Inhalte aus dem Internet und dem Intranet abrufen	1. Klicken Sie auf <i>Verbose HTTP logging enabled</i> (Ausführliche HTTP-Protokollierung aktiviert). 2. Klicken Sie auf <i>True</i> (Wahr).
Überwachen verschlüsselter Daten, die mit TLS zwischen dem BlackBerry-Gerät und dem Ausgangs-Webserver gesendet werden	1. Klicken Sie auf <i>TLS logging enabled</i> (TLS-Protokollierung aktiviert). 2. Klicken Sie auf <i>True</i> (Wahr).
Überwachen des vom BlackBerry-Gerät auf dem OCSP-Server abgerufenen Zertifikatssperrstatus	1. Klicken Sie auf <i>OCSP logging enabled</i> (OCSP-Protokollierung aktiviert). 2. Klicken Sie auf <i>True</i> (Wahr).
Überwachen von Anfragen vom BlackBerry-Gerät für den Zugriff auf ein Benutzerprofil oder ein Zertifikat aus dem LDAP-Verzeichnis	1. Klicken Sie auf <i>LDAP logging enabled</i> (LDAP-Protokollierung aktiviert). 2. Klicken Sie auf <i>True</i> (Wahr).
Überwachen der vom BlackBerry-Gerät auf dem CRL-Server abgerufenen Zertifikatssperrlisten	1. Klicken Sie auf <i>CRL logging enabled</i> (CRL-Protokollierung aktiviert). 2. Klicken Sie auf <i>True</i> (Wahr).
Überwachen des vom BlackBerry-Gerät auf dem PGP-Server abgerufenen PGP-Schlüsselstatus und der Sperrdaten	1. Klicken Sie auf <i>PGP logging enabled</i> (PGP-Protokollierung aktiviert). 2. Klicken Sie auf <i>True</i> (Wahr).

5. Doppelklicken Sie auf *Logs* (Protokolle).
6. Klicken Sie auf *Destination* (Ziel).
7. Führen Sie eine der folgenden Aktionen aus:

Tabelle 23.4: Optionen für die Erstellung einer Protokolldatei durch den BlackBerry MDS Connection Service

Aktion	Vorgehensweise
Festlegen der Protokollebene	1. Klicken Sie im Bereich <i>File</i> (Datei) auf <i>Log Level</i> (Protokollebene). 2. Klicken Sie auf eine der folgenden Protokollebenen: • Ereignis • Fehler • Warnhinweis • <i>Informational</i> (Informationen): ermöglicht das Überwachen des normalen MDS-Datenverkehrs. • <i>Debug</i> (Debugging): erleichtert die Fehlerbehebung des BlackBerry MDS Connection Service.
Festlegen des Verzeichnisses, in das der BlackBerry MDS Connection Service die Protokolldatei schreibt	1. Doppelklicken Sie im Bereich <i>File</i> (Datei) auf <i>Location</i> (Speicherort). 2. Geben Sie den Speicherort ein.
Festlegen des Intervalls, in dem der MDS Connection Service Informationen in die Protokolldatei schreibt	1. Doppelklicken Sie im Bereich <i>File</i> (Datei) auf <i>Log Timer Interval</i> (Intervall des Protokollzeitgebers). 2. Geben Sie das Intervall (in Millisekunden) an.
Festlegen der Protokollierungsebene für die UDP-Protokolldatei	1. Klicken Sie im Bereich <i>UDP</i> auf <i>Log Level</i> (Protokollebene). 2. Klicken Sie auf die Protokollebene.
Festlegen des Ports, an den der BlackBerry MDS Connection Service UDP-Protokollnachrichten sendet. Der SNMP-Agent für BlackBerry Enterprise Server empfängt diese Nachrichten über denselben Port.	1. Doppelklicken Sie im Bereich <i>UDP</i> auf <i>Location</i> (Speicherort). 2. Geben Sie den Port, über den eine Verbindung zum SNMP-Agent hergestellt werden soll, im folgenden Format an: <i><Host-Name:Port></i>.
Festlegen der Protokollierungsebene für die TCP-Protokolldatei.	1. Klicken Sie im Bereich <i>TCP</i> (Datei) auf <i>Log Level</i> (Protokollebene). 2. Klicken Sie auf die Protokollebene.
Festlegen des Speicherorts, zu dem der MDS Connection Service eine Verbindung herstellt, um die TCP-Protokollbericht zu senden	1. Doppelklicken Sie im Bereich <i>TPC</i> auf <i>Location</i> (Speicherort). 2. Geben Sie den Speicherort, zu dem der MDS Connection Service eine Verbindung zum Senden der Protokollnachricht herstellen soll, im folgenden Format an: <i><Host-Name:Port></i>.
Festlegen der Protokollierungsebene für das Ereignisprotokoll	1. Klicken Sie im Bereich <i>EventLog</i> (Ereignisprotokoll) auf <i>Log Level</i> (Protokollebene). 2. Klicken Sie auf die Protokollebene.

8. Klicken Sie auf OK.

23.3.3 Optionen für die Erstellung einer Protokolldatei durch den Collaboration Service

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *BlackBerry Collaboration Services* (BlackBerry Collaboration Services) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Logs* (Protokolle).
4. Führen Sie eine der folgenden Aktionen aus:

Tabelle 23.5: Anpassen der Erstellung einer Protokolldatei durch den BlackBerry Collaboration Service

Aktion	Vorgehensweise
Kein Überwachen der Aktivität auf der BlackBerry-Instant-Messaging-Netzwerkebene	1. Klicken Sie auf <i>BBIM logging enabled</i> (BBIM-Protokollierung aktiviert). 2. Klicken Sie in der Dropdown-Liste auf <i>False</i> (Falsch).
Kein Überwachen der Aktivität auf der SRP-Netzwerkebene	1. Klicken Sie auf <i>SRP logging enabled</i> (SRP-Protokollierung aktiviert). 2. Klicken Sie in der Dropdown-Liste auf <i>False</i> (Falsch).
Überwachen der Aktivität auf der GME-Netzwerkebene	1. Klicken Sie auf <i>GME logging enabled</i> (GME-Protokollierung aktiviert). 2. Klicken Sie in der Dropdown-Liste auf <i>True</i> (Wahr).

5. Klicken Sie auf *OK*.

23.3.4 Überwachen von PIN-Nachrichten, SMS-Nachrichten und Anrufen in einer BlackBerry-Domain

1. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf einen Enterprise Server.
2. Klicken Sie auf der Registerkarte *Server Configuration* (Serverkonfiguration) auf *Edit Properties* (Eigenschaften bearbeiten).
3. Klicken Sie auf *Sync Server* (Synchronisierungsserver).
4. Doppelklicken Sie auf *Audit Root Directory* (Prüf-Stammverzeichnis).
5. Geben Sie den absoluten Pfad zum Speicherort an, in dem Sie die Protokolldateien speichern möchten.
6. Klicken Sie auf *OK*.

7. Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain* (BlackBerry-Domäne).
8. Klicken Sie auf der Registerkarte *Global* (Global) auf *Edit Properties* (Eigenschaften bearbeiten).
9. Klicken Sie auf *IT Policy* (IT-Richtlinie).
10. Doppelklicken Sie im Bereich *IT Policy Administration* (IT-Richtlinienverwaltung) auf *IT Policies* (IT-Richtlinien).
11. Klicken Sie in der Liste mit Richtlinien auf eine Richtlinie.
12. Klicken Sie auf *Properties* (Eigenschaften).
13. Klicken Sie auf *PIM Sync Policy Group* (Richtliniengruppe für PIM-Synchronisierung).
14. Führen Sie eine der folgenden Aktionen aus:

Tabelle 23.6: Überwachen von PIN-Nachrichten, SMS-Nachrichten und Anrufen in einer BlackBerry-Domain

Aktion	Vorgehensweise
Überwachen der SMS-Nachrichten, die Benutzer von BlackBerry-Geräten senden	1. Klicken Sie auf <i>Disable SMS Messages Wireless Sync</i> (Drahtlose Synchronisierung von SMS-Nachrichten deaktivieren). 2. Klicken Sie in der Dropdown-Liste auf <i>False</i> (Falsch).
Überwachen der PIN-Nachrichten, die Benutzer von BlackBerry-Geräten senden	1. Klicken Sie auf <i>Disable PIN Messages Wireless Sync</i> (Drahtlose Synchronisierung von PIN-Nachrichten deaktivieren). 2. Klicken Sie in der Dropdown-Liste auf <i>False</i> (Falsch).
Kein Überwachen der von Benutzern auf BlackBerry-Geräten durchgeführten Anrufe	1. Klicken Sie auf <i>Disable Phone Call Log Wireless Sync</i> (Drahtlose Synchronisierung des Anrufprotokolls deaktivieren). 2. Klicken Sie in der Dropdown-Liste auf <i>True</i> (Wahr).

- Klicken Sie auf *OK*.
- Starten Sie den BlackBerry Synchronization Service auf dem Computer, auf dem er installiert wurde, erneut in den Windows-Diensten. BlackBerry Enterprise Server erstellt die Protokolldateien in den folgenden Formaten:
 - *PINLog_<JJJJMMTT>.csv*
 - *SMSLog_<JJJJMMTT>.csv*
 - *PhoneCallLog_<JJJJMMTT>.csv*

23.4 Verwalten anderer BlackBerry-Domains

Verwalten Sie eine andere BlackBerry-Domain, indem Sie eine Verbindung zwischen dem BlackBerry Manager und einer anderen Configuration Database herstellen.

23.4.1 Verbinden des BlackBerry Managers mit einer anderen BlackBerry-Domain

1. Klicken Sie im BlackBerry Manager im Menü *Tools* (Extras) auf *Options* (Optionen).
2. Klicken Sie auf *Database* (Datenbank).
3. Führen Sie folgende Aktionen aus:

Tabelle 23.7: Verbinden des BlackBerry Managers mit einer anderen BlackBerry-Domain

Aktion	Vorgehensweise
Festlegen des Datenbankservers, zu dem eine Verbindung hergestellt werden soll	1. Doppelklicken Sie auf <i>Database Server Name</i> (Name des Datenbankservers). 2. Geben Sie den Namen des Datenbankservers ein, auf dem sich die BlackBerry Configuration Database befindet.
Festlegen der Configuration Database, zu der eine Verbindung hergestellt werden soll	1. Doppelklicken Sie auf <i>Database Name</i> (Name der Datenbank).
Aktivieren der ausführlichen Protokollierung aller Anrufe in der Configuration Database	› Klicken Sie in der Dropdown-Liste <i>Log Database Calls</i> (Datenbankanrufe protokollieren) auf <i>True</i> (Wahr).

4. Klicken Sie auf *OK*.
5. Schließen Sie den BlackBerry Manager.
6. Öffnen Sie den BlackBerry Manager.

23.5 Verwalten von Lizenzschlüsseln

Der Lizenzschlüssel für den Client-Zugriff steuert, wie viele Benutzerkonten gleichzeitig auf einem Enterprise Server vorhanden sein können. Wenn die Anzahl zulässiger Benutzerkonten überschritten wird, informiert Sie der Lizenz-Manager, dass Sie weitere Lizenzen für den Client-Zugriff benötigen.

Warnung: Wenn Sie eine Probeversion des Lizenzschlüssels für den Client-Zugriff verwenden und dieser Schlüssel abläuft, wird der BlackBerry-Dispatcher automatisch deaktiviert, wodurch die gesamte Synchronisierung zwischen dem Enterprise Server und den Endgeräten angehalten wird. Sie müssen einen neuen Lizenzschlüssel für den Client-Zugriff erwerben, bevor Sie einen Neustart durchführen können. Wenn Sie eine Probeversion des Lizenzschlüssels verwenden, können Sie diesen Schlüssel nicht mehr verwenden, sobald Sie einen Dauerlizenzschlüssel für den Client-Zugriff erworben haben.

Um Lizenzschlüssel für den Client-Zugriff einfacher auf Computer in verschiedenen BlackBerry-Domains zu übertragen oder Probleme mit Lizenzschlüsseln für den Client-Zugriff zu beheben, können Sie die Lizenzschlüssel aus dem BlackBerry Manager in eine Textdatei kopieren.

23.5.1 Hinzufügen oder Löschen von Lizenzschlüsseln

- Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain*.
- Klicken Sie auf der Registerkarte *Global* (Global) auf *Account* (Konto).
- Klicken Sie auf *License Management* (Lizenzverwaltung).
- Führen Sie eine der folgenden Aktionen aus:

Tabelle 23.8: Hinzufügen oder Löschen von Lizenzschlüsseln

Aktion	Vorgehensweise
Hinzufügen eines Lizenzschlüssels für den Client-Zugriff	1. Geben Sie die Informationen für den neuen Lizenzschlüssel an. 2. Klicken Sie auf <i>Add License</i> (Lizenz hinzufügen). 3. Klicken Sie auf <i>Close</i> (Schließen).
Löschen eines Lizenzschlüssels für den Client-Zugriff	1. Klicken Sie mit der rechten Maustaste auf den zu löschenden Lizenzschlüssel. Klicken Sie auf <i>Remove License Key</i> (Lizenzschlüssel löschen). 2. Klicken Sie auf <i>Close</i> (Schließen).

23.5.2 Kopieren eines Lizenzschlüssels in eine Textdatei

- Klicken Sie im BlackBerry Manager im linken Fensterbereich auf *BlackBerry Domain*.
- Klicken Sie auf der Registerkarte *Global* (Global) auf *Account* (Konto).
- Klicken Sie auf *License Management* (Lizenzverwaltung).

- Klicken Sie mit der rechten Maustaste auf einen Lizenzschlüssel. Klicken Sie auf *Copy Key* (Schlüssel kopieren).
- Öffnen Sie einen Texteditor.
- Fügen Sie den Lizenzschlüssel in die Datei ein.
- Speichern Sie die Datei.

Anhang

A.1 Rollenmatrix

A.1.1 Domänenaufgaben

Tabelle A.1: Domänenaufgaben

<i>Symbol/ Register- karte</i>	<i>Aufgaben-/ Eigen- schaftenseite</i>	<i>Eigen- schaften</i>	<i>Sicher- heits- admi- nistrator</i>	<i>Unterneh- mens- administ- rator</i>	<i>Geräte- admi- nistra- tor</i>	<i>Senior Helpdesk- Admi- nistrator</i>	<i>Junior Helpdesk- Admi- nistrator</i>
BlackBerry- Domain			Bearbeiten	Bearbeiten	Anzeigen	Anzeigen	Anzeigen
	Find User		Bearbeiten	Bearbeiten	Anzeigen	Anzeigen	Anzeigen
	Enable Enterprise Service Policy		Bearbeiten	Bearbeiten	-	-	-
	Find Handheld		Bearbeiten	Bearbeiten	Anzeigen	Anzeigen	-
	License Management		Bearbeiten	Bearbeiten	-	-	-
	Edit PIM Sync Global Field Mapping		Bearbeiten	Bearbeiten	-	-	-
	Global Properties		Bearbeiten	Bearbeiten	-	-	-
		General	Bearbeiten	Bearbeiten	-	-	-
		IT Policy	Bearbeiten	Bearbeiten	-	-	-
		Access Control	Bearbeiten	Bearbeiten	-	-	-
		Push Control	Bearbeiten	Bearbeiten	-	-	-
		Global PIM Sync	Bearbeiten	Bearbeiten	-	-	-
		WLAN Confi- guration	Bearbeiten	Bearbeiten	-	-	-

<i>Symbol/ Register- karte</i>	<i>Aufgaben-/ Eigen- schaftenseite</i>	<i>Eigen- schaften</i>	<i>Sicher- heits- admi- nistrator</i>	<i>Unterneh- mens- administ- rator</i>	<i>Geräte- admi- nistra- tor</i>	<i>Senior Helpdesk- Admi- nistrator</i>	<i>Junior Helpdesk- Admi- nistrator</i>
		Media Content Mana- gement	Bearbeiten	Bearbeiten	-	-	-
		Enter- prise Service Policy	Bearbeiten	Bearbeiten	-	-	-
	Send Message		Bearbeiten	Bearbeiten	-	-	-
	Update Peer-to- Peer Encryption Key		Bearbeiten	Bearbeiten	-	-	-
	Reset PIM Sync Global Field Mapping		Bearbeiten	Bearbeiten	-	-	-
	Import IT Policy Definitions		Bearbeiten	Bearbeiten	-	-	-
	MDS CS to BES Mapping		Bearbeiten	Bearbeiten	-	-	-
	IM to BES Mappings		Bearbeiten	Bearbeiten	-	-	-
Role Administ- ration			Bearbeiten	-	-	-	-
	Add Admi- nistrators		Bearbeiten	-	-	-	-
	List Administ- rators		Bearbeiten	-	-	-	-
	Remove Ad- ministrators		Bearbeiten	-	-	-	-

A.1.2 BlackBerry Enterprise Server – Aufgaben

Tabelle A.2: BlackBerry Enterprise Server – Aufgaben

<i>Symbol/ Register- karte</i>	<i>Aufgaben-/ Eigenschaf- tenseite</i>	<i>Eigen- schaften</i>	<i>Sicher- heitsadmi- nistrator</i>	<i>Unterneh- mensad- ministrator</i>	<i>Geräte- admi- nistra- tor</i>	<i>Senior Helpdesk- Admi- nistrator</i>	<i>Junior Helpdesk- Admi- nistrator</i>
Servers			Bearbeiten	Bearbeiten	Anzeigen	Anzeigen	Anzeigen
	Add Users		Bearbeiten	Bearbeiten	-	Bearbeiten	-
	Clear Statistics				-	-	-
	Disable BlackBerry MDS Connection Service				-	-	-
	Server Pro- perties				-	-	-
		BES Alert			-	-	-
		Global Filters			-	-	-
		General			-	-	-
		IT Admin			-	-	-
		Messa- ging			-	-	-
		Sync Server	Bearbeiten	Bearbeiten	-	-	-
		MDS Services	Bearbeiten	Bearbeiten	-	-	-
	Remove BlackBerry Enterprise Server		Bearbeiten	Bearbeiten	-	-	-
	Restart BlackBerry Enterprise Server		Bearbeiten	Bearbeiten	-	-	-
	Stop BlackBerry Enterprise Server		Bearbeiten	Bearbeiten	-	-	-
	Send Message		Bearbeiten	Bearbeiten	-	Bearbeiten	-

<i>Symbol/ Register- karte</i>	<i>Aufgaben-/ Eigenschaf- tenseite</i>	<i>Eigen- schaften</i>	<i>Sicher- heitsadmi- nistrator</i>	<i>Unterneh- mensad- ministrator</i>	<i>Geräte- admi- nistra- tor</i>	<i>Senior Helpdesk- Admi- nistrator</i>	<i>Junior Helpdesk- Admi- nistrator</i>
Connection Services			Bearbeiten	Bearbeiten	-	-	-
	Clear Statistics		Bearbeiten	Bearbeiten	-	-	-
	BlackBerry MDS Connection Service Properties		Bearbeiten	Bearbeiten	-	-	-
		General	Bearbeiten	Bearbeiten	-	-	-
		http	Bearbeiten	Bearbeiten	-	-	-
		LDAP	Bearbeiten	Bearbeiten	-	-	-
		Access Control	Bearbeiten	Bearbeiten	-	-	-
		Logs	Bearbeiten	Bearbeiten	-	-	-
		OCSF	Bearbeiten	Bearbeiten	-	-	-
		Push/PAP	Bearbeiten	Bearbeiten	-	-	-
		Proxy	Bearbeiten	Bearbeiten	-	-	-
		RSA Authentication	Bearbeiten	Bearbeiten	-	-	-
		Stats	Bearbeiten	Bearbeiten	-	-	-
		TLS/HTTPS	Bearbeiten	Bearbeiten	-	-	-
	Restart Service		Bearbeiten	Bearbeiten	-	-	-
	Set as Push Server		Bearbeiten	Bearbeiten	-	-	-
	Unset as Push Server		Bearbeiten	Bearbeiten	-	-	-
	Start Service		Bearbeiten	Bearbeiten	-	-	-
	Stop Service		Bearbeiten	Bearbeiten	-	-	-
	Restart Service		Bearbeiten	Bearbeiten	-	-	-
	BlackBerry Collaboration Service Properties		Bearbeiten	Bearbeiten	-	-	-

Symbol/ Register- karte	Aufgaben-/ Eigenschaft- tenseite	Eigen- schaften	Sicher- heitsadmi- nistrator	Unterneh- mensad- ministrator	Geräte- admi- nistra- tor	Senior Helpdesk- Admi- nistrator	Junior Helpdesk- Admi- nistrator
		General	Bearbeiten	Bearbeiten	-	-	-
		Proxy	Bearbeiten	Bearbeiten	-	-	-
		Logs	Bearbeiten	Bearbeiten	-	-	-
«Name des MDS Ser- vices- Servers»			Bearbeiten	Bearbeiten	Anzeigen	Anzeigen	Anzeigen
	«MDS Services server name» Properties		Bearbeiten	Bearbeiten	-	-	-
		General	Bearbeiten	Bearbeiten	-	-	-
		Filters	Bearbeiten	Bearbeiten	-	-	-
		Device Policies	Bearbeiten	Bearbeiten	-	-	-
		Certifi- cate	Bearbeiten	Bearbeiten	-	-	-
		Connec- tion Service	Bearbeiten	Bearbeiten	-	-	-
		Message Monitors	Bearbeiten	Bearbeiten	-	-	-
		Security	Bearbeiten	-	-	-	-
		Proxy	Bearbeiten	Bearbeiten	-	-	-
		JDBC Drivers	Bearbeiten	Bearbeiten	-	-	-
	Start Service		Bearbeiten	Bearbeiten	-	-	-
	Stop Service		Bearbeiten	Bearbeiten	-	-	-
	Add Certificate		Bearbeiten	Bearbeiten	-	-	-
Applicati- ons In- stalled			Bearbeiten	Bearbeiten	Bear- beiten	Bearbeiten	Anzeigen
	Remove Application from List		Bearbeiten	Bearbeiten	Bear- beiten	-	-

Symbol/ Register- karte	Aufgaben-/ Eigenschaft- tenseite	Eigen- schaften	Sicher- heitsadmi- nistrator	Unterneh- mensad- ministrator	Geräte- admi- nistra- tor	Senior Helpdesk- Admi- nistrator	Junior Helpdesk- Admi- nistrator
	Quarantine Application		Bearbeiten	Bearbeiten	Bear- beiten	-	-
	Reinstate Application		Bearbeiten	Bearbeiten	Bear- beiten	-	-

A.1.3 Gruppenaufgaben

Tabelle A.3: Gruppenaufgaben

Symbol/ Register- karte	Aufgaben-/ Eigen- schaftenseite	Eigen- schaften	Sicher- heits- administ- rator	Unterneh- mensadmi- nistrator	Geräte- admi- nistra- tor	Senior Helpdesk- Administ- rator	Junior Helpdesk- Administ- rator
User Groups			Bearbeiten	Bearbeiten	Anzeigen	Anzeigen	Anzeigen
User Groups List			Bearbeiten	Bearbeiten	Anzeigen	Anzeigen	Anzeigen
	Edit Group Template		Bearbeiten	Bearbeiten	Anzeigen	Anzeigen	Anzeigen
		Redirec- tion	Bearbeiten	Bearbeiten	Anzeigen	Anzeigen	Anzeigen
		Filters	Bearbeiten	Bearbeiten	Anzeigen	Anzeigen	Anzeigen
		Security	Bearbeiten	Bearbeiten	Anzeigen	Anzeigen	Anzeigen
		IT Policy	Bearbeiten	Bearbeiten	Anzeigen	Anzeigen	Anzeigen
		PIM Sync	Bearbeiten	Bearbeiten	Anzeigen	Anzeigen	Anzeigen
		Access Control	Bearbeiten	Bearbeiten	Anzeigen	Anzeigen	Anzeigen
	Create Group		Bearbeiten	Bearbeiten	-	-	-
	Modify Group Definition		Bearbeiten	Bearbeiten	-	-	-
	Delete Group		Bearbeiten	Bearbeiten	-	-	-
	Copy Properties to Another Group		Bearbeiten	Bearbeiten	-	-	-

<i>Symbol/ Register- karte</i>	<i>Aufgaben-/ Eigen- schaftenseite</i>	<i>Eigen- schaften</i>	<i>Sicher- heits- administ- rator</i>	<i>Unterneh- mensadmi- nistrator</i>	<i>Geräte- admi- nistra- tor</i>	<i>Senior Helpdesk- Administ- rator</i>	<i>Junior Helpdesk- Administ- rator</i>
	Update Group Membership		Bearbeiten	Bearbeiten	Anzeigen	Anzeigen	-
	Move Group to BlackBerry Enterprise Server		Bearbeiten	Bearbeiten	-	-	-
	Send Message		Bearbeiten	Bearbeiten	-	-	-
	Set Activation Password		Bearbeiten	Bearbeiten	-	-	-
	Generate and Email Activation Password		Bearbeiten	Bearbeiten	-	-	-
	Resend IT Policy		Bearbeiten	Bearbeiten	-	-	-
	Assign IT Policy		Bearbeiten	Bearbeiten	-	-	-
	Resend Peer-to-Peer Key		Bearbeiten	Bearbeiten	-	-	-
	Resend Service Book		Bearbeiten	Bearbeiten	-	-	-
	Reset PIM Sync Field Mapping		Bearbeiten	Bearbeiten	-	-	-
	Clear PIM Sync Backup Data		Bearbeiten	Bearbeiten	-	-	-
	Purge Pending Messages		Bearbeiten	Bearbeiten	-	-	-
	Clear Statistics		Bearbeiten	Bearbeiten	-	-	-
	Export Stats To File		Bearbeiten	Bearbeiten	-	-	-
	Assign Device Policy		Bearbeiten	Bearbeiten	-	-	-
	Install on Device		Bearbeiten	Bearbeiten	-	-	-
	Uninstall on Device		Bearbeiten	Bearbeiten	-	-	-
	Assign Software Configuration		Bearbeiten	Bearbeiten	Bearbei- ten	-	-

Symbol/ Register- karte	Aufgaben-/ Eigen- schaftenseite	Eigen- schaften	Sicher- heitsad- ministr- ator	Unterneh- mensadmi- nistrator	Geräte- admi- nistrato- r	Senior Helpdesk- Administ- rator	Junior Helpdesk- Administ- rator
	Update Configuration Check Status		Bearbeiten	Bearbeiten	Bearbei- ten	-	-
	Export Asset Summary Data		Bearbeiten	Bearbeiten	Bearbei- ten	-	-
Software Configura- tions			Bearbeiten	Bearbeiten	Bearbei- ten	-	-
	Add New Configuration		Bearbeiten	Bearbeiten	Bearbei- ten	-	-
	Edit Configuration		Bearbeiten	Bearbeiten	Bearbei- ten	-	-
	Copy Configuration		Bearbeiten	Bearbeiten	Bearbei- ten	-	-
	Delete Configuration		Bearbeiten	Bearbeiten	Bearbei- ten	-	-
	Manage Appli- cation Policies		Bearbeiten	Bearbeiten	Bearbei- ten	-	-

A.1.4 Benutzeraufgaben

Tabelle A.4: Benutzeraufgaben

Symbol/ Register- karte	Aufgaben-/ Eigenschaf- tenseite	Eigen- schaf- ten	Sicher- heitsad- minist- rator	Unterneh- mensad- ministrato- r	Gerä- tead- minist- rator	Senior Helpdesk- Administ- rator	Junior Helpdesk- Administ- rator
Users			Bearbeiten	Bearbeiten	Bearbei- ten	Bearbeiten	Bearbeiten
	Set Activation Password		Bearbeiten	Bearbeiten	-	Bearbeiten	Bearbeiten
	Reload User		Bearbeiten	Bearbeiten	-	Bearbeiten	Bearbeiten
	Clear In-Cradle Flag		Bearbeiten	Bearbeiten	-	Bearbeiten	Bearbeiten

<i>Symbol/ Register- karte</i>	<i>Aufgaben-/ Eigenschaft- enseite</i>	<i>Eigen- schaf- ten</i>	<i>Sicher- heitsad- minist- rator</i>	<i>Unterneh- mensad- ministrato- r</i>	<i>Gerä- tead- minist- rator</i>	<i>Senior Helpdesk- Administ- rator</i>	<i>Junior Helpdesk- Administ- rator</i>
	Choose Folders for Redirection Hinweis: Um die Ordnerumleitung für einen Benutzer einzurichten, müssen Sie über die erforderlichen Berechtigungen für den Nachrichtenserver auf der Mailbox des Benutzers verfügen.		Bearbeiten	Bearbeiten	--	-	-
	Add Users		Bearbeiten	Bearbeiten	-	Bearbeiten	-
	Assign To Group		Bearbeiten	Bearbeiten	-	Bearbeiten	-
	Clear Statistics		Bearbeiten	Bearbeiten	-	Bearbeiten	-
	Delete User		Bearbeiten	Bearbeiten	-	Bearbeiten	-
	Export Stats To File		Bearbeiten	Bearbeiten	-	Bearbeiten	Bearbeiten
	Find User		Bearbeiten	Bearbeiten	-	Bearbeiten	Bearbeiten
	Generate and Email Activation Password		Bearbeiten	Bearbeiten	-	Bearbeiten	Bearbeiten
	Assign IT Policy		Bearbeiten	Bearbeiten	-	Bearbeiten	-
	Resend IT Policy		Bearbeiten	Bearbeiten	-	Bearbeiten	Bearbeiten
	Erase Data and Disable Handheld		Bearbeiten	Bearbeiten	-	Bearbeiten	-

<i>Symbol/ Register- karte</i>	<i>Aufgaben-/ Eigenschaf- tenseite</i>	<i>Eigen- schaf- ten</i>	<i>Sicher- heitsad- minist- rator</i>	<i>Unterneh- mensad- ministrato- r</i>	<i>Gerä- tead- minist- rator</i>	<i>Senior Helpdesk- Administ- rator</i>	<i>Junior Helpdesk- Administ- rator</i>
	Disable Con- nection and Collaboration Services		Bearbeiten	Bearbeiten	-	Bearbeiten	-
	Move User		Bearbeiten	Bearbeiten	-	Bearbeiten	-
	Resend Peer- to-Peer Key		Bearbeiten	Bearbeiten	-	Bearbeiten	Bearbeiten
	Clear PIM Sync Backup Data		Bearbeiten	Bearbeiten	-	Bearbeiten	-
	Edit PIM Sync Field Mapping		Bearbeiten	Bearbeiten	-	Bearbeiten	-
	Reset PIM Sync Field Mapping		Bearbeiten	Bearbeiten	-	Bearbeiten	-
	User Properties		Bearbeiten	Bearbeiten	Bearbei- ten	Bearbeiten	Bearbeiten
		Filters	Bearbeiten	Bearbeiten		Bearbeiten	Bearbeiten
		IT Policy	Bearbeiten	Bearbeiten	-	Bearbeiten	-
		PIM Sync	Bearbeiten	Bearbeiten	-	Bearbeiten	-
		Redirec- tion	Bearbeiten	Bearbeiten	-	Bearbeiten	Bearbeiten
		Security	Bearbeiten	Bearbeiten	-	Bearbeiten	Bearbeiten
		WLAN Configu- ration	Bearbeiten	Bearbeiten	-	Bearbeiten	-
		Advan- ced	Bearbeiten	Bearbeiten	-	-	-

A.1.5 BlackBerry-Geräteverwaltungsaufgaben

Tabelle A.5: BlackBerry-Geräteverwaltungsaufgaben

<i>Symbol/ Register- karte</i>	<i>Aufgaben- /Eigenschaften- seite</i>	<i>Sicherheits- administrator</i>	<i>Unterneh- mensad- ministrator</i>	<i>Gerä- tead- minist- rator</i>	<i>Senior Helpdesk- Administ- rator</i>	<i>Junior Helpdesk- Administ- rator</i>
Local Ports (Device Manage- ment)		Bearbeiten	Bearbeiten	Bearbei- ten	-	-
	Handheld Properties	Bearbeiten	Bearbeiten	Bearbei- ten	-	-
	Load Handheld	Bearbeiten	Bearbeiten	Bearbei- ten	-	-
	Load Handheld (Interactive)	Bearbeiten	Bearbeiten	Bearbei- ten	-	-
	Nuke Handheld	Bearbeiten	Bearbeiten	Bearbei- ten	-	-
	Configure Port	Bearbeiten	Bearbeiten	Bearbei- ten	-	-
	Retrieve Summary Properties	Bearbeiten	Bearbeiten	Bearbei- ten	-	-

A.1.6 Menü »Tools« (Extras)

Tabelle A.6: Menü Tools (Extras)

<i>Symbol/ Register- karte</i>	<i>Aufgaben-/ Eigenschaf- tenseite</i>	<i>Eigen- schaf- ten</i>	<i>Sicher- heitsad- minist- rator</i>	<i>Unterneh- mensad- ministrator</i>	<i>Gerä- tead- minist- rator</i>	<i>Senior Helpdesk- Administ- rator</i>	<i>Junior Helpdesk- Administ- rator</i>
Tools			Bearbeiten	Bearbeiten	Bearbei- ten	Bearbeiten	Bearbeiten
	Options		Bearbeiten	Bearbeiten	Bearbei- ten	Bearbeiten	Bearbeiten
		Data- base	Bearbeiten	Bearbeiten	Bearbei- ten	Bearbeiten	Bearbeiten
		General	Bearbeiten	Bearbeiten	Bearbei- ten	Bearbeiten	Bearbeiten
		Serial Ports	Bearbeiten	Bearbeiten	Bearbei- ten		

B.1 Drahtloses Sichern und Wiederherstellen

Die folgende Liste umfasst BlackBerry-Gerätedaten, die vom BlackBerry Enterprise Server nicht über das drahtlose Netzwerk gesichert werden

Tabelle B.1: Daten, die nicht über das drahtlose Netzwerk gesichert werden

<i>Daten</i>	<i>Beschreibung</i>
Nachrichten	Nachrichten, die auf dem BlackBerry-Gerät vor dem angegebenen Vorabladeterminum empfangen wurden, nicht als gespeichert markiert wurden, sich in Ordnern befinden, die nicht umgeleitet werden oder denen Nachrichtenfilter zugewiesen wurden, um das Umleiten an das BlackBerry-Gerät zu verhindern.
Inhaltsspeicher	Gespeicherte Bilder und Klingeltöne.
Servicebücher	Sämtliche Servicebücher.
Gruppenadressen	Gruppenadressen, die Benutzer auf dem BlackBerry-Gerät erstellen, werden lokal gespeichert und nicht synchronisiert.
RMS-Datenbanken	Java-Anwendungen von Drittanbietern, die in Java ME entwickelt wurden.
Java-Anwendungen	Java-Anwendungen, die von Drittanbietern in der BlackBerry-Java-Entwicklungsumgebung erstellt wurden und die über das drahtlose Netzwerk an Endgeräte gesendet werden.
Enterprise Messenger	Unternehmensanwendung zur Nachrichtenübermittlung, die Sie über das drahtlose Netzwerk an BlackBerry-Geräte senden.
BlackBerry-MDS-Studio-Anwendungen	BlackBerry-MDS-Studio-Anwendungen, die Sie über das drahtlose Netzwerk an BlackBerry-Geräte senden.

C.1 Persistente Datenspeicherung

C.1.1 Zweck der persistenten Datenspeicherung

Die persistente Datenspeicherung hat vorwiegend Bedeutung beim Senden und Empfangen von E-Mails, Browser Requests, Push- und Pull-Datenservices sowie der Synchronisierung des Kalenders und weiteren installierten Applikationen auf dem Device.

Persistente Daten werden im Flash-Speicher des Gerätes geschrieben und können von anderen Applikationen wieder ausgelesen werden.

Die Datenübertragungsgeschwindigkeit in den Flash-Speicher ist langsamer als im herkömmlichen RAM (Random Access Memory) eines PC. Zudem sollte man auch beach-

ten, dass die Übertragungsgeschwindigkeit zwischen einem drahtlosen Netzwerk langsamer ist als in einem verkabelten Netzwerk.

Außerdem sollte man bedenken das Schreib- und Lesezugriffe auf den Flashspeicher die CPU-Belastung steigern und somit die Akkulaufzeit erheblich vermindern.

Wie schreibe ich Daten in den Flash-Speicher? Es gibt bei einem auf Java basierendem BlackBerry oder auch einem anderen Smartphone zwei Möglichkeiten, diese Aufgaben zu bewältigen:

- Nutzen der BlackBerry-Persistent-API
- sogenannte MIDP-Datensatz-Speicherung

Die Nutzung der BlackBerry-Persistent-API erfordert das sogenannte Code-Signing. Durch einen Antrag bei RIM kann man sich eine Signatur für circa 20 Dollar kaufen. RIM generiert eine Code-Signatur. Damit kann auf die API zugegriffen werden.

Create/Open

Hier nun einige Beispiele, wie man auf den persistenten Speicher und die persistenten Objekte zugreifen kann. Beginnen wir mit Create/Open: Der Aufruf `PersistentStore.getPersistentObject(long_key)` bekommt eine Referenz zu einem existierenden `PersistentObject` zurück oder kann ein neues Objekt erzeugen – falls dieses Objekt noch nicht existiert.

Durch die Benutzung eines statischen Konstruktors wird sichergestellt, dass nur ein `PersistentObject` erzeugt wird, wenn ein Objekt dieser Klasse erzeugt wurde. Jedes Mal startet ein Prozess, der den statischen Block aufruft.

Beispiel:¹²

Quelltext 22: Persister Speicher

```
Public class MyStore implements KeyListener, TrackwheelListener{
    private static PersistentObject store;
    static {
        // key has des com.rim.bbdj.mystore
        store = PersistentStore.getPersistentObject (0xa406067aeb8ca6eBL;
    }
}
```

Beachten Sie hierbei, dass `long_key` einen eindeutigen `long` value darstellt – vergleichbar mit einer ID eines Datenbank-Records. Eine sichere Methode, einen eindeutigen `long` value zu generieren, ist die Erzeugung eines Hashs für unsere Full Qualified Packages Names (voll qualifizierten Package-Namen). Wenn innerhalb der Applikation mehr als ein `PersistentObject` erzeugt und gespeichert werden soll, so kann dieses mit einer sogenannten »descriptive« Tabelle angelegt werden. In dieser Tabelle werden die

¹² Vgl. Jonathan Nobels' Artikel »Give Me A Sign« BlackBerry Entwickler Journal vol.1 no.2 (May 2004)

Identifizier unserer Package-Namen abgelegt, bevor der Hash erzeugt wird. In den folgenden Schritten wird erklärt, wie dieser Hash-Wert mittels der BlackBerry-IDE erzeugt wird:

1. Erzeugen Sie einen String, z. B. *com.rim.bbdj.mystore*.
2. Markieren Sie diesen String.
3. Starten Sie das Programm *HashCalc* und fügen Sie den String entweder in das Feld *Data* oder als »HMAC« im Key-Format »Text String« ein, und lassen Sie sich diesen Key berechnen. Der long value wird erzeugt und lautet als MD4: 836d8563f8371604ddd48847e85f73a4

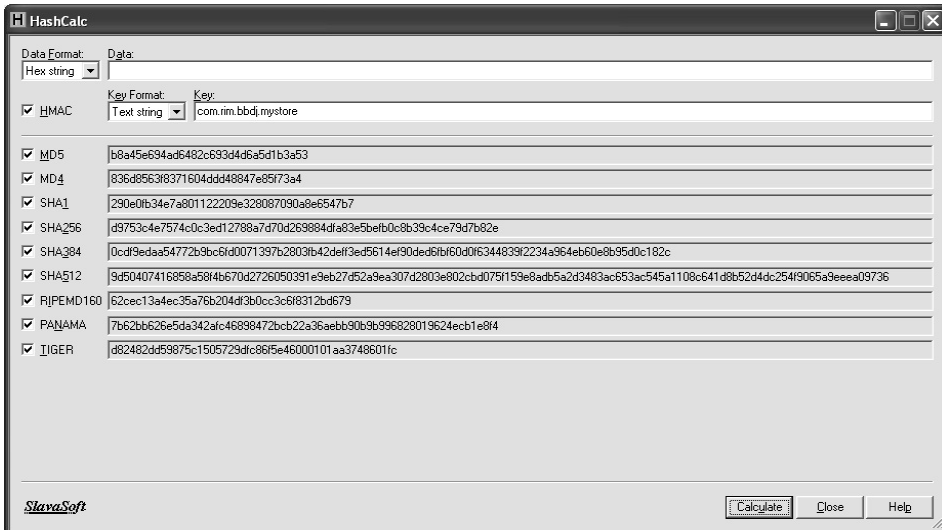


Abbildung C.1: Berechnung eines MD4-Hashkeys

Um den Code verständlich zu halten, empfehle ich die Kommentierung, dass dieser String als long-Key benutzt wird.

C.1.2 Speichern von persistenten Objekten

Um ein `PersistentObject` zu speichern, sind folgende Schritte nötig:

- `PersistentObject.setContents(Object obj)`
- `PersistentObject.commit()`

Hier ein Codebeispiel:

Quelltext 23: Speichern eines synchronen Datenblocks

```
private MenuItem saveItem = new MenuItem ("Save", 110, 10) {
    public void run () {
        String username = new String ("Gandalf");
        String password = new String ("Acce5s");
        String [] userinfo = (username, password);

        synchronized (store) {
            store.setContents(userinfo);
            store.commit();
        }
    }
};
```

Abrufen von persistenten Objekten

Um ein PersistentObject auszulesen, wird folgende Methode angewendet:

PersistentObject.getContents()

Quelltext 24: Auslesen eines PersistentObjects

```
private MenuItem getItem = new MenuItem ("GetItem", 110, 11) {
    public void run() {
        synchronized(store) {
            if (store.getContents() == null)
                System.out.println ("Error");
            else
                String[] currentinfo = (String[])store.getContents();
        }
    }
};
```

Löschen eines PersistentObject

Um ein persistentes Objekt zu löschen, wird folgende Methode angewendet:

PersistentStore.destroyPersistentObject (long id)

Quelltext 25: Löschen eines PersistentObjekts

```
PersistentStore.destroyPersistentObject (836d8563f8371604ddd48847e85f73a4);
```

Beispiel für ein PersistentObject

Um ein persistentes Adressbuch zu realisieren, kann folgender Code verwendet werden:

Quelltext 26: Persistentes Adressbuch

```

import java.util.*;
import net.rim.vm.*;

public class AddressBook
{
    static Vector addresses;
    static PersistentObject persist;

    static {
        // Hash of "net.rim.sample.AddressBook".
        long KEY = 836d8563f8371604ddd48847e85f73a4;
        persist = PersistentStore.getPersistentObject( KEY );
        if( persist.getContents() == null ) {
            persist.setContents( new Vector ( ) );
            persist.commit();
        }
        addresses = (Vector)persist.getContents();
    }

    void add( Address a ) {
        addresses.addElement( a );
        persist.commit();
    }
}

```

Erklärung:

Wir binden die Bibliothek `java.util` sowie die `net.rim.vm` in unseren Code über die `import`-Methode ein. Anschließend definieren wir die Klasse `AddressBook` als `public`-Funktion. Nun sprechen wir die Funktion an über

Quelltext 27: Aufruf der Klasse "class AddressBook"

```

static Vector addresses;
    static PersistentObject persist;

```

Der Hash wird als Long-Key definiert.

Quelltext 28: Hash-Key definieren

```

static {
    // Hash of "net.rim.sample.AddressBook".
    long KEY = 836d8563f8371604ddd48847e85f73a4;
    persist = PersistentStore.getPersistentObject( KEY );
    if( persist.getContents() == null ) {
        persist.setContents( new Vector ( ) );
        persist.commit();
    }
}

```

Nun wird an `addresses` der Vektor des `persistentStore`-Objekts übergeben:

Quelltext 29: Übergabe des persistentStore-Objekts

```
addresses = (Vector)persist.getContents();
```

Abschließend greifen wir mit folgender Methode auf den Vektor `static Vector addresses` zu:

Quelltext 30: Zugriff auf den Vektor "Addresses"

```
void add( Address a ) {
    addresses.addElement( a );
    persist.commit();
}
```

Dieser Programmteil ist nur durchführbar aus signierten Clients. Um das Code-Signing durchzuführen, kontaktieren Sie RIM.

Signing generiert Ihre COD-Files als signierte Files, die auf allen BlackBerry-Geräten lauffähig sind.

D.1 Package Class

Die komplette Dokumentation der Package-Class würde den Rahmen dieses Buches sprengen, aber einige JDE-Referenzen möchte ich hier erwähnen.

D.1.1 Die Klasse CommPort

```
java.lang.Object
  javax.comm.CommPort
public abstract class CommPort
```

Ein `CommPort` ist eine abstrakte Klasse, die beschreibt, wie die Kommunikations-Ports durch das darunter liegende System erreichbar gemacht werden.

Die `CommPorts` beinhalten eine High-Level-Methode für das Controlling. `SerialPort` und `ParallelPort` sind Unterklassen des `CommPort` inklusive zusätzlicher Methoden für die Low-Level-Kontrolle der physischen Kommunikations-Ports.

Es gibt keine sogenannten »public constructors« für den `CommPort`. Stattdessen nutzt eine Applikation die statische Methode `CommPortIdentifier.getPortIdentifiers`, um eine Liste der verfügbaren Ports zu generieren.

Die Auswahl des Ports aus der Liste und der Aufruf `CommPortIdentifier.open` werden angewendet, um ein `CommPort`-Objekt zu erzeugen. Abschließend wird geprüft, ob das `CommPort`-Objekt zu einem physikalischen communications device class die `SerialPort` oder `ParallelPort` zugeordnet sind.

Nachdem ein Kommunikations-Port identifiziert und geöffnet wurde, kann er mittels der Low-Level-Klassen wie `SerialPort` und `ParallelPort` konfiguriert werden. Nun kann ein IO-Stream geöffnet werden – entweder lesend oder schreibend.

Wird die Applikation beendet, müssen alle Ports wieder geschlossen werden. Das geschieht mit der `close`-Methode. Wird danach ein weiterer Port-Aufruf durchgeführt, führt dies zu einer `java.lang.IllegalStateException`.

D.1.2 Methodendetails

getName()

```
public java.lang.String getName()
```

Erhält den Namen des Kommunikations-Ports. Dieser Name kann mit etwas in Bezug gebracht werden, was der User identifizieren kann, etwa das Label seiner Hardware.

Rückgabe:

Name des Ports

toString()

```
public java.lang.String toString()
```

Gibt eine Zeichenkette zurück, die den Port repräsentiert.

Rückgabe:

Zeichenkette der verwendeten Ports.

getInputStream()

```
public abstract java.io.InputStream getInputStream()  
                                     Fehlermeldung:  
java.io.IOException
```

Gibt den Eingabestrom zurück. Dies ist der einzige Weg, Daten vom Port zu empfangen. Falls der Port unidirektional ist und keinen Support für empfangene Daten bereitstellt, dann bekommt die Methode `getInputStream()` den Wert `Null` zurück.

Das Leseverhalten des Eingabestroms und dessen Rückgabe durch `getInputStream` sind abhängig von der Kombination der Grenzwerte und Timeout-Werte. Die möglichen Verhalten sind in der nachstehenden Tabelle aufgeführt:

Tabelle: D.1 Kombination und Grenzwerte der Timeouts

Grenzwert		Timeout		Lese Buffer Größe	Leseverhalten
Status	Wert	Status	Wert		
deaktiviert	-	deaktiviert	-	n bytes	Block, solange Daten verfügbar sind
aktiviert	m bytes	deaktiviert	-	n bytes	Block, solange $\min(m,n)$ bytes verfügbar sind
deaktiviert	-	aktiviert	x ms	n bytes	Block für x ms oder solange Daten verfügbar sind
aktiviert	m bytes	aktiviert	x ms	n bytes	Block für x ms oder solange $\min(m,n)$ bytes verfügbar sind

Beachten Sie, dass Rahmenfehler durch Timeouts und Grenzwerte frühzeitig verarbeitet werden, ohne eine Exception zu generieren.

Die Aktivierung des Timeouts oder Grenzwertes durch den Wert NULL ist ein spezieller Fall. Dies führt grundsätzlich der Treiber durch. Er entscheidet, ob Daten angefragt werden.

Rückgabe:

Das InputStream-Objekt ist bereit, den Port auszulesen.

Fehlermeldungen:

`java.io.IOException` – wenn ein I/O-Error auftritt.

getOutputStream()

```
public abstract java.io.OutputStream getOutputStream()
                                     Fehlermeldung
java.io.IOException
```

Gibt einen Output-Stream zurück. Das ist der einzige Weg, um Daten an den Kommunikations-Port zu senden. Ist der Port unidirektional (in eine Richtung), und unterstützt er das Senden von Daten nicht, so liefert `getOutputStream` den Wert NULL zurück.

Rückgabe:

OutputStream-Objekt, welches den Port zum Schreiben benutzen kann.

Fehlermeldungen:

`java.io.IOException` – wenn ein I/O-Error auftritt.

close()

```
public void close()
```


Schließt den Kommunikations-Port. Die Applikation muss dazu die Methode `close` aufrufen, wenn kein Zugriff mehr auf den Port erfolgen soll. Benachrichtigungen auf den Besitzwechsel werden zu allen registrierten Klassen gesendet durch `addPortOwnershipListener`.

enableReceiveThreshold()

```
public abstract void enableReceiveThreshold(int thresh)
                                Fehlermeldung
                                UnsupportedOperationException
```

Aktiviert, bedeutet das »Empfange Grenzwert«. Wird diese Eigenschaft durch den Treiber unterstützt, wird `true` zurückgegeben. Ein `read` des Eingabestroms beginnt unverzüglich.

`enableReceiveThreshold` ist eine beratende Methode, die der Treiber eventuell nicht implementiert hat. Als Standard ist dieser Wert nicht gesetzt.

Eine Applikation kann selbst entscheiden, ob der Treiber durch Aufruf der `enableReceiveThreshold`-Methode dies unterstützt. Wenn die Methode `isReceiveThresholdEnabled` ein `false` zurückliefert, dann wird `ReceiveThreshold` nicht unterstützt. Stellt der Treiber diese Eigenschaft nicht zur Verfügung, kommt nach einem Leseversuch in angemessener Zeitspanne die Fehlermeldung `UnsupportedCommOperationException`.

Lesen Sie hierzu `getInputStream` für das exakte Verhalten nach.

Parameter:

`thresh` – wenn einige Bytes im Puffer sind, unverzüglich vom `read` zurück.

Fehlermeldungen:

`UnsupportedCommOperationException` – wird erzeugt, wenn der Treiber die Grenzwertabfrage nicht unterstützt.

disableReceiveThreshold()

```
public abstract void disableReceiveThreshold()
```

Schaltet `ReceiveThreshold` ab.

isReceiveThresholdEnabled()

```
public abstract boolean isReceiveThresholdEnabled()
```

Prüft, ob `ReceiveThreshold` eingeschaltet ist.

Rückgabe:

Rückgabewert ist `true`, wenn der Treiber diese Eigenschaft unterstützt.

Enthalten ab:

CommAPI 1.1

getReceiveThreshold()

```
public abstract int getReceiveThreshold()
```

Erhält den Integerwert des empfangenen Grenzwertes. Wenn der empfangene Grenzwert abgeschaltet ist oder durch den Treiber nicht unterstützt wird, ist der Rückgabewert bedeutungslos.

Rückgabe:

Anzahl von Bytes des empfangenen Grenzwertes.

enableReceiveTimeout()

```
public abstract void enableReceiveTimeout(int rcvTimeout)
                                     Fehlermeldung
UnsupportedCommOperationException
```

Schaltet `enableReceiveTimeout` ein. Wenn der Treiber diese Eigenschaft unterstützt, wird mit dem Lesen des Eingabe-Streams für diesen Port unverzüglich begonnen.

`enableReceiveTimeout` ist eine beratende Methode, die der Treiber eventuell nicht implementiert hat. Im Standard ist `receive timeout` nicht aktiv.

Eine Applikation kann herausfinden, ob ein Treiber die Eigenschaft unterstützt, indem sie durch den Aufruf von `enableReceiveTimeout` und anschließendem Aufruf von `isReceiveTimeoutEnabled()` prüft, ob diese Methoden unterstützt werden. Wird von der Methode `isReceiveTimeoutEnabled()` ein `false` zurückgegeben, so unterstützt der Treiber diese Methode nicht.

Schlagen Sie unter `getInputStream` (D.2.3) nach für das korrekte Verhalten dieser Methode.

Parameter:

`rcvTimeout` – wenn einige Millisekunden vergangen sind, unmittelbare Rückkehr aus dem `read`, ungeachtet der gelesenen Bytes im Eingabepuffer.

Fehlermeldungen:

`UnsupportedCommOperationException` – erscheint, wenn ein Timeout empfangen wurde und der Treiber diesen nicht unterstützt.

disableReceiveTimeout()

```
public abstract void disableReceiveTimeout()
```

Abschalten von `Receive Timeout`.

isReceiveTimeoutEnabled()

```
public abstract boolean isReceiveTimeoutEnabled()
```

Prüft, ob Receive Timeout aktiviert wurde.

Rückgabe:

Liefert true zurück, falls der Treiber einen Timeout unterstützt.

getReceiveTimeout()

```
public abstract int getReceiveTimeout()
```

Empfängt den Integerwert des empfangenen Timeouts. Ist der empfangene Timeout deaktiviert oder wird er vom Treiber nicht unterstützt, ist der Wert bedeutungslos.

Rückgabe:

Der Wert des empfangenen Timeouts in Millisekunden.

Das serielle Datenformat

An dieser Stelle ist ein kleiner Exkurs in die serielle Datentübertragung sinnvoll, um die nachfolgend genannten Methoden besser verstehen zu können.

Die einzelnen, zu einem Zeichen gehörenden Bits werden der Reihe nach (seriell) gesendet. Jedem zu sendenden Zeichen wird ein Startbit (log. 0) vorangestellt. Danach folgen die Bits des Zeichens selbst (zum Beispiel sieben Bit des ASCII-Codes). Als Abschluss folgen das Paritätsbit (siehe unten) und ein bis zwei Stopbits (log. 1), worauf die Leitung wieder im Ruhezustand ist. Logikpegel, Leitungspegel und die zugehörigen englischen Bezeichnungen sind in der nachfolgenden Tabelle dargestellt.

Tabelle D.2: Signaldefinition bei TTY und RS-232-C

Zustand	Bezeichnung	Pegel (TTY)	Pegel RS-232-C
Log.0	Space, break	Kein Strom	+3V bis +25V
Log.1	Mark, Idle	Strom (20mA)	-3V bis -25V

Jedes Bit liegt für die gleiche, feste Zeitspanne auf der Leitung. Sie wird angegeben in der Einheit Baud (nach Jean Maurice Emile Baudot, einem Pionier der Fernschreibtechnik). Dabei gilt: 1 Baud = 1 Bit pro Sekunde. Damit ist die Geschwindigkeit der Datenübertragung jedoch noch nicht festgelegt, denn zwischen den einzelnen Zeichen kann die Leitung ja beliebig lange im Ruhezustand verharren. Die maximale Datenrate ergibt sich, wenn die einzelnen Zeichen so dicht aufeinander folgen, dass das Startbit eines Zeichens direkt an das letzte Stopbit des vorangehenden Zeichens angrenzt. Bei der Übertragung von 7-Bit-ASCII-Zeichen mit je einem Start-, Paritäts- und Stopbit ergibt sich beispielsweise bei einer Baudrate von 300 Baud eine maximal erreichbare Datenrate von $300/(7+3) = 30$ Zeichen pro Sekunde.

Die Bezeichnung TTY stammt aus der Zeit des Fernschreibers (TTY = Teletype = Fernschreiber). Dabei werden zwei Stationen in Form eines geschlossenen Stromkreises miteinander verbunden, wobei im Ruhezustand ein konstanter Gleichstrom von beispielsweise 20 Milliampere (mA) fließt. Daher wird die TTY-Schnittstelle oft auch als 20-mA-Stromschnittstelle bezeichnet.

Kontrollfluss oder auch Paritätsbit

Wie schon erwähnt, kann zusätzlich zu den Datenbits noch ein Paritätsbit übertragen werden. Dieses Bit ermöglicht eine primitive Fehlerkontrolle. Wird mit gerader Parität gearbeitet, so setzt der Sender das Paritätsbit auf log. 1, falls das zu sendende Zeichen eine ungerade Anzahl von gesetzten (log.1) Bits enthält. Bei einer geraden Anzahl wird das Paritätsbit auf log. 0 gesetzt. Der Empfänger prüft nun nach der gleichen Vorschrift, ob das Paritätsbit zu den Datenbits passt. Falls bei der Übertragung eines der Datenbits verfälscht worden ist, so ist dies also vom Empfänger erkennbar. Nicht erkennbar ist jedoch, welches das verfälschte Bit ist. Ist eine gerade Anzahl von Bits verfälscht, so bleibt dies unerkannt. Die verschiedenen Möglichkeiten, ein Paritätsbit zu senden, zeigt die folgende Tabelle auf:

Tabelle D.3: Optionen für das Paritätsbit

Parität	Paritätsbit-Zustand
gerade / even	0, falls gerade Anzahl von Eins(en) 1, falls ungerade Anzahl
ungerade / odd	0, falls ungerade Anzahl von Eins(en) 1, falls gerade Anzahl
null / zero	stets 0
eins / one	stets 1
keine / none	fehlt

Beim Empfänger ist es meist möglich, die Paritätsprüfung auszuschalten. Dennoch muss der Empfänger in jedem Fall das Vorhandensein des Paritätsbits berücksichtigen, da es ja jedes Zeichen um ein Bit verlängert. Um das Stopbit zum richtigen Zeitpunkt erwarten zu können, muss also der Empfänger stets nach einer der genannten Möglichkeiten passend zum Sender eingestellt sein. Ob das empfangene Paritätsbit geprüft wird, ist eine andere Sache. Eine weitere Prüfungsmöglichkeit bietet das Stopbit. Liegt es nicht auf log. 1, so betrachtet der Empfänger dies als *frame error* (engl. frame = Rahmen, gemeint ist der Rahmen, den das Start- und Stopbit um die Datenbits bilden).

enableReceiveFraming()

```
public abstract void enableReceiveFraming(int framingByte)
                        Fehlermeldung
                        UnsupportedOperationException
```

Aktiviert das empfangene Rahmenwerk (Framing). Wenn das empfangene Rahmenwerk die Bedingung `true` zugewiesen bekommt, beginnt unverzüglich das Auslesen des Eingangsstroms auf diesem Port.

`enableReceiveFraming` ist als Standardwert nicht aktiviert.

Die Applikation selbst kann entscheiden, welche Treiberunterstützung beim Aufruf der Methode `enableReceiveFraming` und der anschließenden Methode `isReceiveFramingEnabled` als erste aufgerufen wird.

Wenn die Methode `isReceiveFramingEnabled` den Wert `false` zurückgibt, wird das empfangene Rahmenwerk nicht durch die Methode unterstützt.

Parameter:

`framingByte` – dieses Byte im Eingangsstrom kennzeichnet das Ende eines empfangenen Rahmens und blockiert das Lesen unmittelbar. Nur die unteren 8 Bits des `framingByte` werden verwendet; so lange, bis die oberen 24 Bits nicht mehr maskiert sind. Ein Wert außerhalb von 0-255 wird in Werte der unteren 8 Bits konvertiert.

Fehlermeldungen:

`UnsupportedCommOperationException` – sie wird ausgegeben, wenn Timeouts empfangen werden.

disableReceiveFraming()

```
public abstract void disableReceiveFraming()
```

Deaktiviert das Empfangen des Rahmens.

isReceiveFramingEnabled()

```
public abstract boolean isReceiveFramingEnabled()
```

Prüft, ob der empfangene Rahmen aktiviert wurde.

Rückgabe:

Gibt den Boole'schen Wert `true` zurück, wenn der Treiber den empfangenen Rahmen unterstützt.

getReceiveFramingByte()

```
public abstract int getReceiveFramingByte()
```

Enthält das aktuelle Byte des empfangenen Rahmens. Der Rückgabewert der Methode `getReceiveFramingByte` ist ein Integerwert, der in den Lower Bits das aktuell verwendete Byte des empfangenen Rahmens ist.

Hinweis: Das Framing steht nicht in Beziehung zum Bit-Level-Framing auf dem Hardware-Level und somit auch nicht im Bezug zu Datenfehlern.

Rückgabe:

Integerwert des aktuell verwendeten Bytes des `ReceiveFraming`.

setInputBufferSize()

```
public abstract void setInputBufferSize(int size)
```

Setzt die Eingabe der Puffergröße.

Parameter:

`size` – Größe des Eingabepuffers.

getInputBufferSize()

```
public abstract int getInputBufferSize()
```

Empfängt die Eingabe der Puffergröße.

Rückgabe:

Die Puffergröße, die aktuell in Gebrauch ist.

setOutputBufferSize()

```
public abstract void setOutputBufferSize(int size)
```

Setzt die Ausgabe der Puffergröße.

Parameter:

`size` – Größe des Ausgabepuffers.

getOutputBufferSize()

```
public abstract int getOutputBufferSize()
```

Empfängt als Ausgabe die Puffergröße. Beachten Sie hierzu, dass diese Methode nur annähernd zur Ermittlung der Puffergröße verwendet werden kann, da sie vom jeweiligen Betriebssystem abhängt.

Returns:

Ausgabe der zurzeit verwendeten Puffergröße.

E.1 Hash-Kalkulation

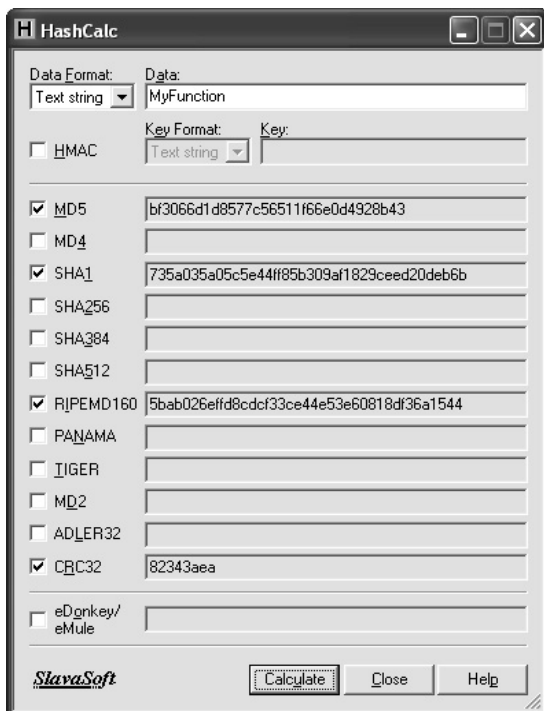


Abbildung E.1: HashCalculator

E.1.1 Komfortable Generierung eines Hashs

Wie wir bereits im Kapitel zur persistenten Datenspeicherung gesehen haben, benötigen wir einen Hash-Wert. Es gibt verschiedene Möglichkeiten, sich einen Hash zu generieren. Ich möchte Ihnen mit dem Hash-Kalkulator ein Tool empfehlen, mit dem sich Hashes bequem generieren lassen. Im Anhang F.2, »Links und weiterführende Informationen« finden Sie den Link auf den Rechner.

F.1 BlackBerry und MAS Message Applikation Server

BlackBerry im Zusammenhang mit einem Messaging Applikation Server

Die Verwendung eines Messaging Servers kann in Bereichen erfolgen, die sensibel sind und überwacht werden müssen und die schnelle Reaktionen auf einen Alarm erfordern. Da das Mail-Management mit dem BlackBerry in Echtzeit geschieht, bietet es sich für sicherheitsrelevante Gebiete an, den MAS (Message Applikation Server) mit einem Handheld-Gerät zu betreiben.

Ich habe mich für den Message Applikation Server von Asperi entschieden.



Abbildung F.1: Message Applikation Server von Asperi

Aufgrund seiner zahlreichen Schnittstellen lassen sich viele Bereiche überwachen und die Alarmer an den BlackBerry weiterleiten. Die Rückseite des MAS lässt auf die vielfältigen Einsatzmöglichkeiten schließen.



Abbildung F.2: Rückseite des Alarmserver

Bis vor Kurzem funktionierte die Nachrichtenübermittlung nur mit Telekommunikationssystemen wie AASTRA von Ericsson, Polycoms Spectralink oder denen von Philips und Cisco über PBX. Die aktuelle Version des Asperi-Alarmserver MAS kann jedoch per Push-Mail Nachrichten versenden und auf E-Mails reagieren.



Abbildung F.3: MAS-Push-Mail

Für die Push-Mail-Funktion muss ein eigener E-Mail-Account auf einem IMAP4-Mail-server eingerichtet sein. Erprobt hat Asperi bis dato MS-Exchange von Microsoft, CommuniGate Pro von CommuniGate Systems und den hMailServer, einen freien E-Mail-Server für Windows. Mit der Push-Mail-Funktion lassen sich nicht nur andere Rechner, sondern auch mobile Geräte erreichen, sofern dort ein E-Mail-Client existiert. MAS kann E-Mails empfangen und daraus Reaktionen generieren, indem es die Betreffzeile auswertet. Geplante Reaktionen kann der Administrator auf dem MAS hinterlegen.

Der kleine Server kann auch Alarme ins Kommunikationsnetz weiterleiten.

Ereignisse an den digitalen Eingängen des MAS, der im Wesentlichen aus einem Industrie-PC besteht, dienen als Quellen für Alarme. Über digitale Ausgänge kann er Reaktionen der angeschlossenen Systeme auslösen.

Zusätzlich gibt es für bestimmte Anwendungen serielle Schnittstellen. Auf dem MAS sind Alarmpläne hinterlegt, die Alarmschwellen und die Art der Reaktionen festlegen. Die Benachrichtigung erfolgt über Push-Mail oder geht wie beim Vorgänger über SMS-Texte an Endgeräte im Festnetz, WLAN, DECT- oder GSM-Netz.

Hier kann zum Beispiel ein BlackBerry-Endgerät die Nachrichten empfangen. Nach Angaben von Asperi erfolgt die Benachrichtigung innerhalb von Bruchteilen einer Sekunde. Erhält der MAS keine Bestätigung der Alarmmeldung in einer vorgegebenen Zeitspanne oder schlägt die geplante Reaktion fehl, sind Eskalationsstufen mit höherer Dringlichkeit und weiteren Verantwortlichen vorgesehen.

Einsatzgebiete sind industrielle Steuerung, Gebäudemanagement, Gesundheitswesen, Energieversorgung, Telekommunikation und Rechenzentren sowie Gemeinden und Behörden. Bei Drucklegung dieses Buches habe ich den MAS mit einer Haustürklingel verbunden und so getestet. Ein erweiterter Testaufbau bestand darin, zusätzlich noch einen Server zu überwachen, dessen Prozessor nicht wärmer als 40 Grad Celsius werden sollte.

In einigen Abständen wurde ein Job gestartet, der das System an seine Grenzen bringen sollte und somit die Temperatur über die eingestellten 40 Grad ansteigen ließ.

Über ein Digitemp-Modul am COM 2 kann man sich einen zuverlässigen Temperaturfühler bauen. Hier erfolgt der Aufbau der Schaltung:

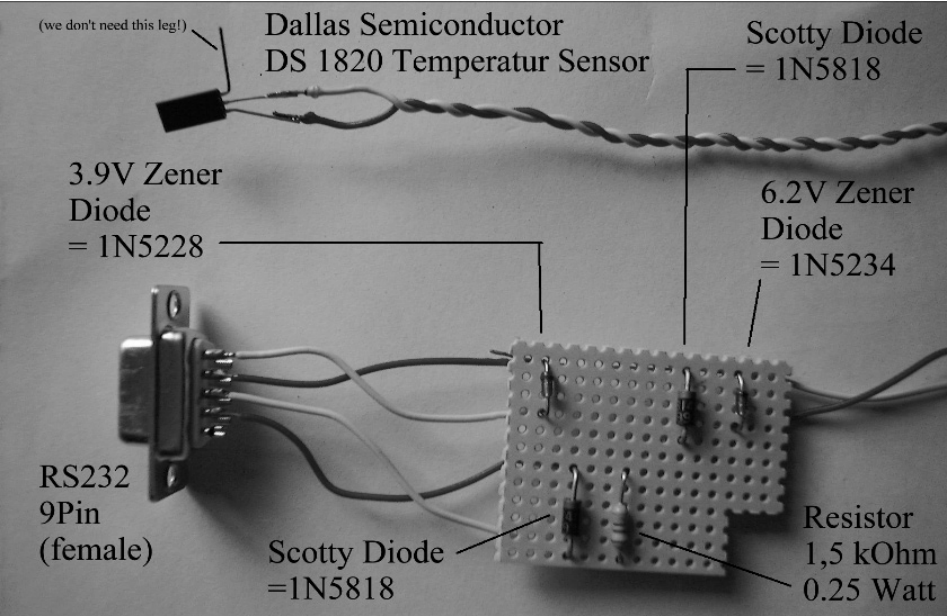


Abbildung F.4: Verdrahtung des DS-1820-Tempersensors

Hier der Schaltplan des Wärmesensors:

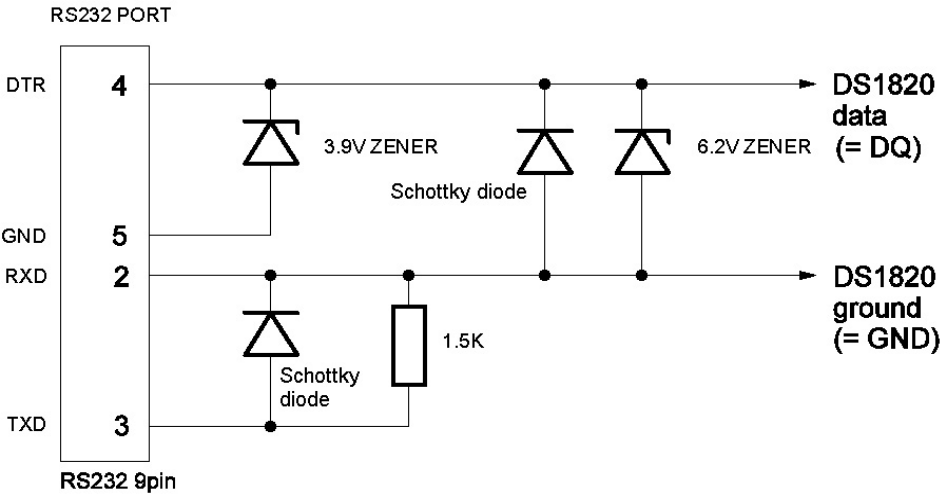


Abbildung F.5: Schaltplan des Wärmesensors

Schalteingänge

MAS mit 16 Eingängen:

Die Schalteingänge IN0 – IN15 befinden sich auf der 36-poligen Steckerleiste an der Rückseite des Gehäuses. Pin 4 ist der gemeinsame GND.

An die Eingänge INx können Sie eine Spannung bis maximal 30 Volt anschließen. Die minimale Schaltschwelle liegt bei 5 VDC.

Schaltausgänge

MAS mit 16 Relaisausgängen:

Die Schaltausgänge Out0 – Out15 befinden sich auf der 36-poligen Steckerleiste an der Rückseite des Gehäuses. Im Gegensatz zur 8-Kanal-Version sind die Ausgänge nicht als Relaisausgänge, sondern als Open-Collector-Ausgänge ausgeführt.

GSM-Modul

Optional kann im MAS ein GSM-Modul installiert werden. Über das GSM-Modul können SMS-Mitteilungen an GSM-Endgeräte gesendet werden. Zum Betrieb des GSM-Moduls muss in den SIM-Kartenleser eine SIM-Karte eines Mobilfunkbetreibers eingesetzt werden.

Über die Systemeinstellungen im Admin-Bereich kann die PIN-Nummer konfiguriert werden. Das GSM-Modul ist nur in Verbindung mit dem MAS mit 4-COM-Schnittstellen verfügbar. Das Modul ist im Gehäuse des MAS integriert und wird intern über die Schnittstelle COM3 angesprochen. COM3 steht deshalb für externe Systeme nicht zur Verfügung.

Die Quittierung über externe GSM-SMS-Benachrichtigung wird unterstützt.

Softwarebeschreibung des MAS

Die Software des Message Applikation Server (MAS) besteht aus zwei Komponenten. Das Control-Programm verwaltet die Ein- und Ausgänge des Systems und wickelt die Protokolle auf den Schnittstellen ab. Der Webserver ist die Schnittstelle zur Programmierung des Systems.

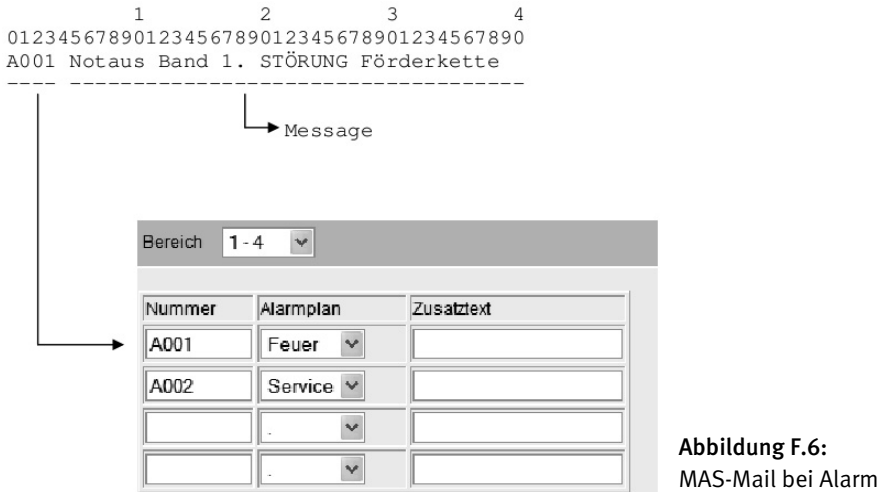
Alle Programme liegen im Flash-Bereich des MAS. Ein Update/Upgrade von Softwarekomponenten kann auf einfache Weise mittels des Browsers erfolgen.

Alarmempfang per E-Mail

Zum Empfang von E-Mails (Auslösen von Alarmen) ist das IMAP4-Protokoll implementiert.

Gegenüber dem POP3-Protokoll hat IMAP4 den Vorteil, dass der Eingang einer E-Mail sofort ohne zeitliche Verzögerung durch Polling registriert wird. Der MAS meldet sich dazu beim IMAP4-Mailserver an und geht in den IDLE-Mode. Sobald eine Mail ein-

trifft, wird dies durch den Mailserver an den MAS gemeldet. Dieser kann dann die eingegangene Mail sofort abholen.



Wie der Betreff der Mail zu interpretieren ist, wird über eine E-Mail-Steuerdatei beschrieben. Der Name der E-Mail-Steuerdatei ist im Parameter *FormatFile* im Bereich *[imap]* der Systemeinstellungen hinterlegt. Der Betreff der Mail muss eine beliebige Kennung sowie den Alarmtext enthalten. Optional kann noch ein Passwort oder Triggerwort enthalten sein. Nur bei Vorhandensein des angegebenen Triggers wird die empfangene Mail weiterverarbeitet.

Beispiel:

```
From: test@maschine.de
To: mas@maschine.de
Subject: A001 Notaus Band 1. STÖRUNG Förderkette

Emailsteuerdatei: recmail.ncs
[LineProtocol]
NrPos=0
NrLen=4
MsgPos=5
MsgLen=120
TrgPos=-1
TrgLen=-1
TrgValue=STÖRUNG,ALARM,
```

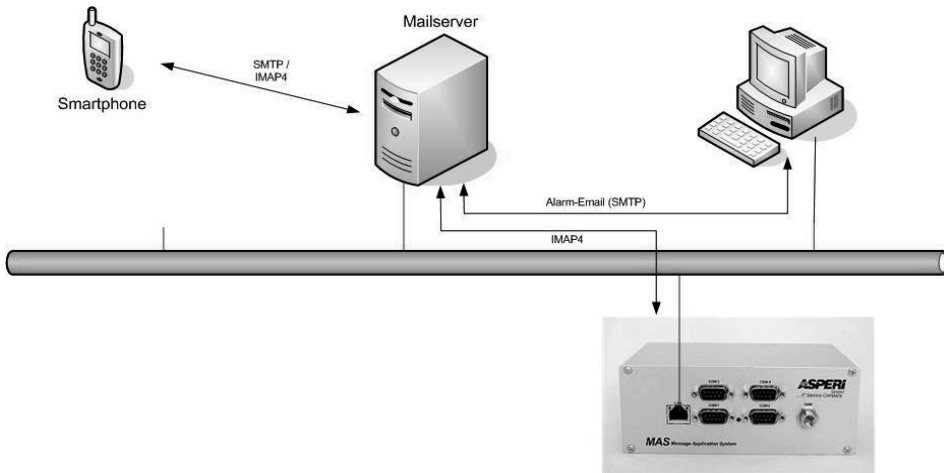


Abbildung F.7: Aufbau der Alarm-Mail

Quittieren eines Alarms per Mail:

Bei einer quittierbaren Meldung fügt der MAS ans Ende der Meldung eine eindeutige Alarm-ID im Format \$ID=xx an, z. B.: A001 Notaus Band 1. STÖRUNG Förderkette \$ID=25

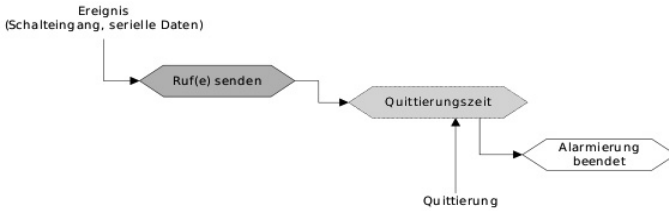
Zum Quittieren dieser Meldung per E-Mail muss die E-Mail einfach zurückgesendet werden.

Wie bei einem normalen Handset wird auf Wunsch (Einstellung *Quittierungsnachricht senden*) eine Bestätigung der Quittierung versendet, zum Beispiel: Q(support@asperi.de) A001 Notaus Band 1...

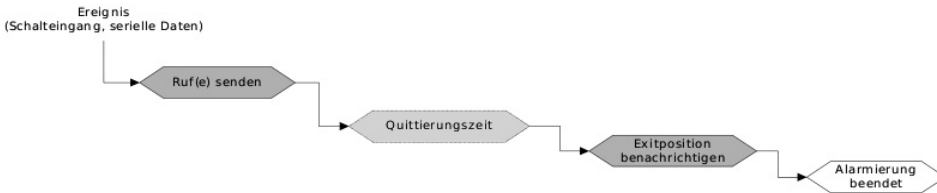
Benachrichtigungsverlauf

Im Folgenden ist der zeitliche Verlauf der Benachrichtigung dargestellt. Nach Eintritt des auslösenden Ereignisses (Schalteingang, Datensatz von serieller Schnittstelle) wird die Benachrichtigung der zugeordneten Empfänger gestartet.

Alarmierungsablauf mit Quittierung innerhalb der Quittierungszeit



Alarmierungsablauf mit ausbleibender Quittierung



Alarmierungsablauf ohne Quittierung

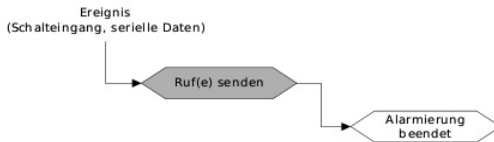


Abbildung F.8: Alarmierungsablauf

Quittierungssteuerung:

Beim Senden von Rufen an zugeordnete Empfänger wird abhängig von der Konfiguration eine Quittierungssteuerung vorgenommen. Ist im Alarmplan die Option *Quittierung* deaktiviert, dann gibt es keine Quittierungszeit, und die Meldungen müssen am Endgerät nicht quittiert werden.

Der MAS steuert bei der Aussendung des Rufes, welches Menü das Endgerät anzeigt. Ist die Option *Quittierung* aktiviert, dann wird nach Aussendung der Rufe die Quittierungszeit gestartet.

Nach Erhalt der Meldung hat der Benutzer am Endgerät folgende Auswahl:

-Annehmen	# Meldung quittieren
-Löschen	# Meldung löschen
-Betrachten	# Meldung lesen

F.1.1 Links und weiterführende Informationen

Linksammlung

Tabelle F.1: Linksammlung / Webadressen

Thema	URL
Eclipse	http://www.eclipse.org/downloads/
Guidelines UI-Interface	http://docs.blackberry.com/en/developers/deliverables/6625/Dimensions_for_screens_images_and_icons_476251_11.jsp
Device Simulators	https://www.blackberry.com/Downloads/entry.do?code=060AD92489947D410D897474079C1477
Informationen über die Datei von Kerberos 5	http://web.mit.edu/kerberos/www/krb5-1.3/krb5-1.3.3/doc/krb5-admin.html#krb5.conf
Informationen über die JAAS-Konfigurationsdatei finden Sie unter	http://java.sun.com/j2se/1.5.0/docs/guide/security/jgss/tutorials/LoginConfigFile.html
Informationen über DOM	http://www.w3.org/DOM/
Hash-Kalkulator	http://www.slavasoft.com
Quellcode-Beispiele	http://www.klsys.com/
ASPERI MAS	http://www.asperi.de/
Digitemp	http://www.digitemp.com/

G.1 Glossar

- AES Advanced Encryption Standard (Erweiterter Verschlüsselungsstandard).
- API Application Programming Interface (Anwendungsprogrammierschnittstelle).
- BlackBerry-MDS BlackBerry Mobile Data System. Der kanadische Blackberry-Hersteller Research in Motion (RIM) setzt in Zukunft auf zwei Software-Entwicklungsansätze: Java- und Web-Anwendungen. Dabei lassen sich aus Elementen der Webentwicklung sogenannte Widgets erzeugen, die HTML, CSS und JavaScript enthalten und als Anwendung zu installieren sind.

Dafür stellt RIM die Entwicklungsumgebung Blackberry-MDS ein. MDS Runtime und MDS Studio sind seit Ende 2009 nicht mehr verfügbar; der Support endet Mitte

2010. Neue Geräte mit OS 4.7.1 und höher werden MDS-Runtime nicht mehr unterstützen.

- CLDC Connected Limited Device Configuration (Bibliothek für Gerätekonfigurationen).
- CRM Customer Relationship Management.
- DOM Document Object Model.
- GUID Globally Unique Identifier (weltweit einzigartiger Identifikator) ist eine einzigartige Identifikationsnummer, die von Microsoft-Anwendungen erstellt wird, um eine bestimmte Einheit, etwa eine Komponente, eine Anwendung, eine Datei, einen Datenbankeintrag oder einen Benutzer, identifizieren oder zurückverfolgen zu können.
- 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.
- MAS Messaging Application Server
- MIDP Mobile Information Device Profile (Softwareprofil für mobile Geräte).
- NVRAM Nichtflüchtiger Arbeitsspeicher (Non-Volatile Random Access Memory).
- PDAP PDA Optional Packages for the J2ME Platform.
- PIM Personal Information Management (Verwaltung persönlicher Daten).
- PIN Personal Identification Number (persönliche Identifikationsnummer).
- POP Post Office Protocol.
- RAPC RIM Application Program Compiler.
- RMS Record Management System (Datensatzverwaltungssystem).
- SHA Secure Hash Algorithm (sicherer Hash-Algorithmus).
- SMS Short Message Service (Kurznachrichtendienst).
- SMTP Simple Mail Transfer Protocol (Simple Mail Transfer-Protokoll).

- SRAM Statischer Arbeitsspeicher.
- SSL Secure Sockets Layer (Netzwerkprotokoll zur sicheren Datenübertragung).
- TCP Transmission Control Protocol (Transmissions-Kontroll-Protokoll).
- TLS Transport Layer Security (Netzwerkprotokoll zur sicheren Datenübertragung).
- Triple DES Triple Data Encryption Standard (Verschlüsselungsstandard).
- VPN Virtual Private Network (virtuelles privates Netzwerk).
- WAP Wireless Application Protocol (Protokoll zur drahtlosen Datenübertragung).
- XML Extensible Markup Language (erweiterbare Auszeichnungssprache).

Stichwortverzeichnis

A

AES 165, 169, 188, 193, 194, 375
 Anlagendateiformate 287
 Anpassungen 26, 187
 API 375
 API-Packages 182
 Arbeitsspeicher 63
 Audiounterstützung 54

B

Benutzer einer Gruppe zuweisen
 207
 Benutzergruppen 188, 206, 234,
 244, 270, 317, 318
 Benutzerkonten 205, 221, 242,
 255, 268, 273, 299, 310, 317
 BES 41, 177, 187
 BIS 19, 42, 166, 187
 BitmapField 106, 112, 114
 BlackBerry-Domain 193, 336

C

choiceField 71, 72
 CLDC 44, 45, 181, 184, 185, 186,
 376
 COD-Datei 17, 47, 52, 171
 Codesignierung 45
 Computerviren 184
 ContentHandler 55, 57
 CRM 37, 180, 376
 CustomButtons 83

D

DateField 80, 81, 82
 Dateigrößen 289

Dateiverbindungs-APIs 65
 Datenobjekte 60
 Datensatzverwaltungssystem 65,
 376
 Datenspeicherung 45, 65, 352,
 366
 Datenstrukturauswahl 61, 62
 Datenverwaltung 64
 Dauerhafte Speicherung 41, 65
 DayCalc 79, 80
 Divisionsvorgänge 46, 49
 DOM 190, 285, 286, 375, 376
 DRM 57
 drop-down list 71

E

E/A 41, 376
 Email 155, 175
 Encryption 165, 308, 375

F

final 46, 89
 Flash 60, 169, 228, 352, 371

G

GAL 202, 275, 318
 Garbage Collection 39, 47, 60, 63,
 64, 184
 Garbage Collector 61
 Genehmigungsliste 210
 Gerätespeicher 169
 getPreferredWidth 79, 85, 91, 92,
 93
 globale Adressenliste 202
 GUID 155, 156, 376

Guidelines 34, 179, 180, 375

Guidelines UI-Interface 375

H

Hackerangriffe 184

HashCalculator 51, 366

Hashtable 52, 62

HMAC 354

HTTP 165, 200, 294, 334, 344,
376

HTTPS 169, 306, 344, 376

HTTP-Standardauthentifizierung
170

I

IDE 38, 185

IMAP 187, 188, 376

IPC 44, 376

J

JavaLoader 39, 176

JDE 38, 181

JDWP 39, 376

JSR 39, 40, 64, 65, 184, 376

JTWI 40, 376

JVM 38, 48, 60, 63, 64, 376

K

Kameraunterstützung 55

Kernbetriebssystem 60

Klassenfelder 47

Kriterientypen 210

L

LDAP 202, 203, 298, 304, 305,
314, 334, 344

Listenfeld 25, 122

long-value 354

M

MAS 366, 376

MDS 42, 165, 196, 239, 325

MenuItem 72, 117, 355

MIDlet 43, 45, 184

MIDP 43, 54, 59, 65, 181, 182,
184, 185, 353, 376

Minimieren 61, 288, 290

Multimedia 54

Multithreading 45, 186

N

Nachrichtensynchronisierung
263

Nachrichtenwarteschlange 255,
272

Navigationsereignis 58

NullField 100

NVRAM 173, 376

O

ObjectChoiceField 71, 122, 136,
150

ObjectGroupReadOnlyException
63

Objektgruppen 62

Objektmodellierung 45

ocsp 298

onClose 31, 71, 72

Optimieren 52, 285

P

pac 196

PAC 196, 197

PDAP 40, 185, 376

Peer-to-Peer-Codierungsschlüssel
195, 196

Permanente Objekt-Handles 60

PGP 194, 265, 305, 334

PIM 182, 185, 232, 276, 277, 337,
341, 346, 347, 376

PIM Sync 233, 337, 341, 346, 347

PIM-Elemente 207

PIN 376

POP 187, 188, 376

Proxy-Autokonfigurationsdatei
196

Push 42

pushscreen 159

Push-Server 200, 201, 293

R

RAPC 39, 376

regedit 203, 290, 326

Ressourcendateien 54

RMS 64, 65, 352, 376

run 72, 75, 82, 103, 104, 112, 114,
117, 122, 159, 355

Runtime 16, 39, 60, 61, 186, 224,
225, 228, 242, 321, 375

S

Schleifen 19, 46, 48

Schlüsselspeicherdatei 304, 307

Schwellenwert 58, 61, 329

SHA 171, 376

SMS 167, 331, 336, 337, 369, 371,
376

SMTP 176, 376

Speicherkapazität 60, 242

Speicherverwaltung 60

SRAM 60, 377

SSL 169, 188, 201, 298, 302, 304,
308, 315, 376, 377

Standard-IT-Richtlinienregeln
211, 212

Stream 55, 357, 359

Synchronisierung 43, 231, 264,
273, 339

T

Tastatur 20, 37, 57, 59, 175, 183

TCP 42, 66, 165, 166, 280, 284,
285, 315, 335, 377

Terminplanerdaten 168, 207, 208,
209, 231, 232, 233, 235, 273

Textzeichenfolgen 54

Thread 45, 49, 71, 82, 315, 327

Throwable-Instanz 51

Tick-Schwellenwert 58

TLS 169, 280, 298, 304, 306, 308,
334, 344, 377

Trackball 36, 37, 44, 57, 58, 59,
83, 123, 162, 175

Trackball-API 58

Trackwheel 44, 57, 59, 112, 120,
175, 183

Triple DES 188, 193, 194, 377
und AES 193, 194

U

UI 183, 185

Unternehmensdienst-Richtlinie
172, 209, 210, 211

Unternehmensfirewall 41, 42,
166, 321

User Interface 67, 69, 180, 183

V

Vector 62

Vendor 31

Virtual Machine 60, 169

VPN 42, 377

W

WAP 19, 166, 377

Workspace 20, 25, 26, 29, 30

X

x ms 359

X.509 194

XML 41, 66, 377

Z

Zentrale Push-Server 293

Programmieren für den BlackBerry

Der BlackBerry ist das Business-Smartphone schlechthin, allen iPhones und Android-Handys zum Trotz. Denn BlackBerry-Geräte können weit mehr als nur Mails versenden und empfangen, passende Apps erweitern den Funktionsumfang erheblich. Smartphone-Experte Klaus Löwenhagen demonstriert in diesem Buch anhand konkreter Praxisbeispiele, wie Sie Java-Applikationen für den BlackBerry programmieren, wie Sie Apps verwalten, Benutzerrichtlinien für die Anwender festlegen und vieles mehr.

► Eine eigene Applikation entwickeln

Anhand konkreter Beispielanwendungen erläutert der Autor, wie Sie Applikationen für BlackBerry-Smartphones entwickeln. Sie lernen unter anderem, wie Sie Elemente der grafischen Benutzeroberfläche in Ihre Apps einbauen, eingehende Mails verarbeiten und komplexere Programme mit mehreren Screens realisieren. Darüber hinaus gibt der Autor Tipps dazu, wie Sie die Systemressourcen des BlackBerry effizient nutzen, damit Ihre Apps schnell und schlank werden.

► Die Smartphone-API

Der Schlüssel für erfolgreiches Entwickeln für den BlackBerry ist das Verstehen der komplexen Programmierschnittstelle. Klaus Löwenhagen stellt die Softwareplattform vor, die auf bewährten Java-Standards basiert, und zeigt, wie Sie die zentralen Klassen und Systemkomponenten richtig einsetzen. Er erläutert auch die Grundzüge des BlackBerry-Betriebssystems und stellt diejenigen Funktionen vor, die Sie brauchen, um Daten auf externen Speicherkarten zu sichern. Sie erfahren außerdem, wie Sie Entwicklungsprojekte mit der Programmierumgebung Eclipse aufsetzen und im Gerätesimulator testen.

► BlackBerry-Administration

Da BlackBerry-Geräte primär im Businessumfeld eingesetzt werden, ist das Zusammenspiel zwischen dem Smartphone und der Infrastruktur im Unternehmen von entscheidender Bedeutung. In einem umfangreichen Admin-Guide erfahren Sie, wie Sie den BlackBerry Enterprise Server (BES) einsetzen und Benutzerkonten einrichten, Anwendergruppen die nötigen Benutzerrechte einräumen und Anwendern die Applikationen zugänglich machen. Der Autor zeigt zudem, wie sich BlackBerry-Geräte gegenüber Unternehmensanwendungen authentifizieren, um einen Datenaustausch zu ermöglichen. Last, but not least demonstriert Klaus Löwenhagen, wie ein Messaging Application Server dringende Alarmmeldungen auf den BlackBerry sendet.

Aus dem Inhalt:

- Einrichten der Eclipse-Entwicklungsumgebung
- Der Simulator-Start
- Die BlackBerry-API
- Dialogfelder
- Gestaltung von Auswahlmenüs
- Passwortgeschützte Apps
- Backdoor-Programme
- Mail-Verwaltung
- Drahtloser Datentransport
- UI-Guidelines
- Admin-Guide für den BlackBerry Enterprise Server (BES)
- Einrichten von Benutzerkonten auf dem BES
- Einrichten von BlackBerry-Geräten
- Push-Server
- Instant Messaging
- Zentrale Verwaltung von Mail-Attachments
- Remote-Zugriff auf Businessanwendungen
- Alarmierung mit dem BlackBerry und einem Messaging Application Server
- Dokumentationstipps

Über den Autor:

Klaus Löwenhagen ist Dipl.-Informatiker und Geschäftsführer des Unternehmens Klsys IT-Solutions. Gleichzeitig ist er als Testmanager beim Mobilnetzbetreiber Telefonica O2 Germany tätig. Zuvor war er als Testmanager bei Siemens Mobile in Kamp-Lintfort für den BlackBox Integration Test verantwortlich. Klaus Löwenhagen lebt in der Nähe von Bielefeld.



Auf www.buch.cd:

- Die kompletten Beispielcodes
- Batch-Datei für die Installation Ihrer Anwendungen auf dem BlackBerry



35,- EUR [D]

ISBN 978-3-645-60018-7

Besuchen Sie unsere Website

www.franzis.de