

BlackBerry Java Development Environment

Versión: 4.7.0

Guía de conceptos básicos

Contenido

1	Comprensión de BlackBerry y programación de dispositivos BlackBerry.....	5
	Principios de diseño para dispositivos BlackBerry.....	5
	Ciclos de lanzamientos y versiones.....	6
	BlackBerry Java Development Environment.....	6
	API de Java ME y Java para BlackBerry.....	7
	Compatibilidad con las API estándar de Java.....	7
	Compatibilidad con las extensiones de API de Java.....	8
	Soluciones BlackBerry.....	9
	BlackBerry Enterprise Solution.....	9
	BlackBerry Internet Service.....	9
	BlackBerry MDS.....	10
2	Diseño de BlackBerry Java Application.....	11
	Aplicaciones independientes.....	11
	Aplicaciones con sincronización de escritorio.....	11
	Aplicaciones con acceso inalámbrico, sincronización inalámbrica o alerta inalámbrica.....	11
	Aplicaciones MIDlet.....	11
	Aplicaciones CLDC.....	12
	Control de API y firma de código.....	13
	Modelos de objetos.....	13
	Subprocesamiento múltiple.....	13
	Práctica recomendada: uso de subprocesamiento múltiple.....	14
	Prácticas recomendadas para escribir una aplicación BlackBerry Java Application eficaz.....	14
	Práctica recomendada: escritura de código eficaz.....	14
	Práctica recomendada: uso apropiado de los objetos.....	19
	Práctica recomendada: reducción del tamaño del código compilado.....	19
	Compatibilidad de varios idiomas.....	21
	Práctica recomendada: almacenamiento de cadenas de texto en archivos de recursos.....	21
	Compatibilidad multimedia.....	21
	Compatibilidad de audio.....	21
	Compatibilidad de la cámara.....	22
	Compatibilidad de vídeo.....	22
3	Interfaz de usuario y diseño de navegación.....	25

Escritura y navegación del usuario de dispositivos BlackBerry.....	25
Rueda de desplazamiento frente a bola de desplazamiento.....	25
Creación de una interfaz de usuario que sea coherente con las interfaces de usuario estándar de BlackBerry.....	26
4 Gestión de memoria.....	28
Gestión de memoria.....	28
Memoria del dispositivo BlackBerry.....	28
Recursos claves para reservar.....	28
Práctica recomendada: reducción del uso de la memoria.....	29
Gestión de una disponibilidad de memoria baja.....	29
Identificación de una disponibilidad baja de memoria en un dispositivo BlackBerry.....	29
Conservación de recursos.....	30
Práctica recomendada: uso de una selección de estructura de datos eficaz.....	30
Práctica recomendada: consolidación de objetos en grupos de objetos.....	30
Recogida de datos desechables en un dispositivo BlackBerry.....	31
Recogida de datos desechables de la RAM en un dispositivo BlackBerry.....	31
Recogida de datos desechables en un dispositivo BlackBerry.....	31
Recogida de datos desechables inactiva en un dispositivo BlackBerry.....	31
5 Administración de datos.....	33
Compatibilidad con las API para almacenar datos en la memoria constante.....	33
API del almacén constante.....	33
API del sistema de administración de registros MIDP.....	33
API de conexiones de archivo.....	33
Almacenamiento en medios extraíbles.....	34
Acceso a los datos en la tarjeta multimedia microSD.....	34
Copia de seguridad y sincronización de los datos.....	34
6 Transporte inalámbrico de datos.....	36
Gateways inalámbricas.....	36
Uso de BlackBerry Enterprise Server como gateway de red.....	36
Uso de la gateway a Internet del proveedor de servicios inalámbricos.....	36
Opciones de transporte de datos alternativas.....	37
Uso del correo electrónico para transportar datos.....	37
Uso de SMS para transportar datos.....	37
Uso de la mensajería PIN para transportar datos.....	37

7	Integración de aplicaciones BlackBerry.....	38
	Adición de elementos de menú personalizados.....	38
	Integración con aplicaciones de BlackBerry Device Software.....	38
	Acceso a los datos del correo electrónico y del organizador.....	38
	Uso de BlackBerry Messenger con BlackBerry Application.....	38
	Uso de oyentes para responder a los cambios de la aplicación.....	39
8	Consideraciones de seguridad.....	40
	Cifrado de datos y BlackBerry Application.....	40
	Cifrado de datos durante el transporte.....	40
	Cifrado de datos en el dispositivo BlackBerry.....	40
	Acceso a la memoria.....	40
	Autenticación.....	41
	Autenticación de dispositivos BlackBerry y política de TI.....	41
	Autenticación de aplicaciones.....	41
	Autenticación del servidor.....	41
	API controladas y firma de códigos.....	41
	API de BlackBerry con acceso controlado.....	42
	Compatibilidad con la política de TI.....	43
	Control de la aplicación.....	43
	Cifrado de archivos en tarjetas microSD.....	43
	Cifrado de datos en una tarjeta multimedia microSD.....	43
	Uso de la tarjeta multimedia microSD con más de un dispositivo BlackBerry.....	44
	Políticas de TI y tarjetas multimedia microSD.....	44
9	Probar BlackBerry Java Application.....	45
	Cifrado de una aplicación BlackBerry Java Application.....	45
	Verificación previa de BlackBerry Java Application.....	45
	Prueba de aplicaciones en BlackBerry Smartphone Simulator.....	45
	Prueba de aplicaciones en un dispositivo BlackBerry.....	46
	Simuladores de BlackBerry.....	46
	BlackBerry Smartphone Simulator.....	46
	BlackBerry MDS Simulator.....	47
	BlackBerry Email Simulator.....	47

10 Disponibilidad de las aplicaciones	48
Distribución de aplicaciones a través de una conexión de ordenador.....	48
Distribuir una aplicación desde un ordenador.....	48
Distribuir una aplicación desde una página Web.....	48
Distribuir una aplicación para la prueba.....	48
Distribución de aplicaciones a través de la red inalámbrica.....	48
Extracción inalámbrica (iniciada por el usuario).....	49
Inserción inalámbrica (iniciada por el servidor).....	49
11 Glosario	50
12 Aviso legal	53

Comprensión de BlackBerry y programación de dispositivos BlackBerry

1

Los dispositivos BlackBerry® proporcionan un entorno inalámbrico Java® ME que es compatible con las aplicaciones cliente/servidor. Los desarrolladores de aplicaciones pueden crear BlackBerry® Java Application, con interfaces de usuario sofisticadas, para la entrada y búsqueda de datos compatibles con un subprocesamiento múltiple, internacionalización, comunicación de red y el almacenamiento local de datos. Las aplicaciones pueden comunicarse con las redes mediante conexiones estándar TCP y HTTP, a pesar de la red inalámbrica subyacente.

Los desarrolladores de aplicaciones también pueden crear BlackBerry Java Application integrada por completo con aplicaciones principales del dispositivo BlackBerry, como la lista de mensajes, las aplicaciones de organizador, el teléfono y el explorador, para una experiencia de usuario perfecta.

Principios de diseño para dispositivos BlackBerry

Las aplicaciones diseñadas para dispositivos BlackBerry® deben ofrecer un equilibrio entre la mejor experiencia de usuario posible y una duración larga de la batería. Cuando diseñe su aplicación de dispositivo BlackBerry, considere las diferencias entre dispositivos móviles y ordenadores. Los dispositivos móviles:

- tienen un tamaño de pantalla más pequeño que puede mostrar un número limitado de caracteres
- tienen velocidades de procesador más lentas
- utilizan redes inalámbricas que tienen un período de latencia más largo que las LAN estándar
- tienen menos memoria disponible
- tiene una batería que dura menos
- muestran una pantalla al mismo tiempo

Los usuarios de dispositivos móviles no utilizan del mismo modo las aplicaciones en su dispositivo móvil y las aplicaciones en un ordenador. En un dispositivo móvil, los usuarios esperan encontrar la información rápidamente. Por ejemplo, un sistema de gestión de relaciones con clientes (CRM) puede ofrecer una cantidad masiva de información, pero los usuarios sólo requieren una pequeña cantidad de esa información a la vez. La interfaz de usuario del dispositivo BlackBerry se ha diseñado para que los usuarios puedan realizar las tareas fácilmente y acceder a la información rápidamente.

Cuando diseñe aplicaciones para dispositivos BlackBerry, intente ser tan coherente como sea posible con otras aplicaciones del dispositivo BlackBerry. Considere las siguientes directrices:

- Utilice o amplíe los componentes existentes de la interfaz de usuario de forma que la aplicación pueda heredar el comportamiento predeterminado del componente.
- Siga el modelo estándar de navegación tanto como le sea posible para que los usuarios puedan hacer un uso completo del teclado y la bola de desplazamiento.

- Realice todas las acciones disponibles desde el menú. Compruebe que las acciones disponibles en el menú son relevantes para el contexto actual de los usuarios.

Al diseñar su aplicación, debe considerar también las siguientes directrices:

- Céntrese en las tareas inmediatas de los usuarios. Simplifique la selección y presentación de datos para mostrar sólo la información que los usuarios necesitan en cualquier otro momento.
- Muestre la información de manera que haga un uso efectivo de la pequeña pantalla.

Antes de diseñar su aplicación, considere la utilización de las aplicaciones principales en el dispositivo BlackBerry o BlackBerry Smartphone Simulator para conocer más detalles acerca del modelo de navegación y las prácticas recomendadas para diseñar la interfaz de usuario de la aplicación.

Ciclos de lanzamientos y versiones

Todos los dispositivos BlackBerry® incluyen una versión específica de BlackBerry® Device Software y de BlackBerry® Java® Virtual Machine. Para determinar la versión de BlackBerry Device Software de un dispositivo BlackBerry, en las Opciones del dispositivo, haga clic en **Acerca de**. Puede actualizar BlackBerry Device Software. Por ejemplo, puede actualizar un dispositivo BlackBerry con la versión de BlackBerry Device Software 4.0 a BlackBerry Device Software versión 4.1.

Con cada uno de los lanzamientos principales de BlackBerry Device Software y BlackBerry JVM, Research In Motion incluye las correspondientes API de Java y la versión de BlackBerry® Java® Development Environment. La versión de BlackBerry Device Software determina la versión de BlackBerry® Integrated Development Environment que puede utilizar para desarrollar aplicaciones. Por ejemplo, RIM lanzó BlackBerry Device Software versión 4.0 y BlackBerry JDE versión 4.0 al mismo tiempo. BlackBerry JDE versión 4.0 incluye compatibilidad con las API que se introdujeron en BlackBerry Device Software versión 4.0 y BlackBerry JVM versión 4.0. Las aplicaciones creadas mediante BlackBerry JDE versión 4.0 sólo funcionan en dispositivos BlackBerry que ejecutan BlackBerry Device Software versión 4.0 o posterior.

Puede utilizar los siguientes criterios para decidir qué versión de BlackBerry JDE utilizar para desarrollar una aplicación:

- Si la aplicación no necesita utilizar características específicas de hardware del dispositivo BlackBerry ni extensiones de las API recientemente lanzadas, utilice BlackBerry JDE versión 4.0 para desarrollar la aplicación.
- Si la aplicación está diseñada para ejecutarse sólo en el smartphone BlackBerry® Pearl™ 8100, use BlackBerry JDE versión 4.2 o posterior.

BlackBerry Java Development Environment

BlackBerry® Java® Development Environment es un entorno completamente integrado de desarrollo y simulación para crear BlackBerry® Java Application para dispositivos BlackBerry. Gracias a BlackBerry JDE, los desarrolladores pueden crear aplicaciones con el lenguaje de programación Java® ME y las API extendidas de Java para BlackBerry.

BlackBerry Java Development Environment incluye las siguientes herramientas de desarrollo:

- BlackBerry® Integrated Development Environment
- BlackBerry Smartphone Simulator
- Java ME y API de BlackBerry
- aplicaciones de ejemplo

BlackBerry IDE incluye una serie completa de herramientas de edición y depuración que se han optimizado para el desarrollo de BlackBerry Java Application. BlackBerry Smartphone Simulator ofrece un entorno tipo Windows® completo y está diseñado para simular interfaces de usuario y la interacción del usuario, conexiones de red, servicios de correo electrónico y sincronización inalámbrica de datos.

El paquete de componentes de BlackBerry Java Development Environment incluye las siguientes herramientas de desarrollo para el desarrollo en IDE de terceros como NetBeans™ o Eclipse®:

- RAPC: puede utilizar este compilador del símbolo del sistema para compilar archivos .java y .jar en archivos .cod que puede ejecutar en BlackBerry Smartphone Simulator o en un dispositivo BlackBerry.
- JavaLoader: puede utilizar esta herramienta para agregar o actualizar una aplicación en un dispositivo BlackBerry para probar y para ver información sobre los archivos .cod de la aplicación.
- BlackBerry® Signature Tool: puede utilizar esta herramienta para enviar solicitudes de firma de código a BlackBerry® Signing Authority Tool.
- Herramienta de verificación previa: puede utilizar esta herramienta para comprobar parcialmente sus clases antes de cargar su aplicación en un dispositivo BlackBerry.
- JDWP: puede utilizar esta herramienta para depurar aplicaciones utilizando entornos de desarrollo integrados de terceros.

API de Java ME y Java para BlackBerry

Java® ME es una plataforma estándar del sector que define los conjuntos comunes de API de Java para distintos tipos de dispositivos inalámbricos e incorporados. Una aplicación Java ME en un dispositivo BlackBerry® se ejecuta en la máquina virtual BlackBerry® Java® Virtual Machine, que proporciona todos los servicios de tiempo de ejecución a las aplicaciones y realiza funciones tales como asignaciones normales de memoria, comprobaciones de seguridad y recogida de datos desechables.

La MIDP estándar de Java ME gestiona las necesidades de la API y BlackBerry JVM de un dispositivo inalámbrico restringido con una interfaz de usuario. El dispositivo BlackBerry es compatible con la MIDP estándar de Java ME tal como se define en JSR 118. La MIDP estándar de Java ME proporciona un conjunto fundamental de API de Java que son compatibles con cualquier dispositivo BlackBerry, independientemente de su sistema operativo subyacente. Los desarrolladores a menudo pueden construir una aplicación Java utilizando la API MIDP estándar y ejecutando esa aplicación en muchos tipos diferentes de dispositivos.

Compatibilidad con las API estándar de Java

El dispositivo BlackBerry® y BlackBerry® Java® Development Environment son compatibles con el estándar MIDP de Java® ME, que proporciona un conjunto principal de API de Java que puede utilizar para desarrollar aplicaciones de dispositivos inalámbricos. El dispositivo BlackBerry y BlackBerry® Java® Development Environment también son compatibles con los siguientes JSR:

- JSR 30: Connected Limited Device Configuration versión 1.0
(compatible en dispositivos con BlackBerry® Device Software versión 4.0 o anterior)
- JSR 37: Mobile Information Device Profile versión 1.0
(compatible en dispositivos con BlackBerry Device Software versión 4.0 o anterior)
- JSR 75: Paquetes opcionales portátiles para la plataforma J2ME (PDAP), compatibles únicamente con las API de PIM y las API de conexión de archivos para Java ME (compatible en dispositivos con BlackBerry Device Software versión 4.2 o posterior)
- JSR 82: API de Java para Bluetooth®
- JSR 118: Mobile Information Device Profile versión 2.0
- JSR 120: Wireless Messaging API (WMA) versión 1.1
- JSR 135: Mobile Media APIs (MM API) versión 1.1
- JSR 139: Connected Limited Device Configuration versión 1.1
- JSR 172: J2ME Web Services
- JSR 177: Security and Trust Services API para J2ME (SATSA)
- JSR 179: API de ubicación para Java ME
- JSR 185: Tecnología Java para la industria inalámbrica (JTWI)
- JSR 205: Wireless Messaging API 2.0
- JSR 211: Content Handler API
- JSR 226: API de gráficos vectoriales 2D escalables para Java ME
- JSR 238: Mobile Internationalization API

Compatibilidad con las extensiones de API de Java

Los dispositivos BlackBerry® son compatibles con las siguientes API de Java® que no forman parte de las definiciones estándar de JSR y que pueden proporcionar características y funcionalidades mayores además de las disponibles en las bibliotecas estándar de API de MIDP.

API	Descripción
API de interfaz de usuario	Puede utilizar estas API para crear pantallas, elementos del menú y todos los componentes de la interfaz de usuario.
API de almacenamiento de datos constante	Puede utilizar estas API para almacenar los datos personalizados de forma local en la aplicación.
API de redes y E/S	Puede utilizar estas API para establecer conexiones de red y leer o escribir datos en una aplicación de servidor.
Detectores de eventos	Puede utilizar Oyentes de eventos para responder al usuario del dispositivo BlackBerry o eventos iniciados por el sistema en un dispositivo BlackBerry.

API	Descripción
API de integración de aplicaciones	Puede utilizar estas API para integrarse con el correo electrónico existente de BlackBerry, el teléfono, el calendario, los contactos, el explorador, la cámara, el reproductor multimedia y las aplicaciones de la lista de tareas.
Utilidades adicionales	Puede utilizar estas API adicionales para el cifrado y compresión de datos, análisis de XML, conectividad Bluetooth®, servicios basados en ubicación, etcétera.

Soluciones BlackBerry

Los usuarios de dispositivos BlackBerry® pueden utilizar tanto BlackBerry® Enterprise Server como BlackBerry® Internet Service, o pueden utilizar ambos en el mismo dispositivo. La comprensión de las diferencias entre BlackBerry Enterprise Server y BlackBerry Internet Service y saber para qué tipo de usuarios tiene previsto ofrecer compatibilidad, es importante, puesto que puede repercutir en los modos de transporte que utiliza y en cómo puede administrar la sincronización de datos.

BlackBerry Enterprise Solution

BlackBerry® Enterprise Server forma parte de BlackBerry® Enterprise Solution. BlackBerry Enterprise Server existe tras el firewall de la empresa y proporciona una gateway inalámbrica a los usuarios del dispositivo BlackBerry en una empresa para obtener acceso al correo electrónico de la empresa y los datos del organizador. BlackBerry Enterprise Server también ofrece las siguientes características clave:

- cifrado de datos y compresión
- administración de dispositivos BlackBerry y utilidades de control
- abastecimiento simplificado de la aplicación
- gateway autenticada para el acceso a la intranet desde BlackBerry® Java Application

BlackBerry Internet Service

Los usuarios de dispositivos BlackBerry® que no están asociados con BlackBerry® Enterprise Server pueden utilizar BlackBerry® Internet Service. BlackBerry Internet Service es un servicio de correo electrónico y de Internet para dispositivos BlackBerry diseñado para proporcionar a los usuarios con entrega automática de mensajes de correo electrónico, acceso inalámbrico a archivos adjuntos de mensajes de correo electrónico y acceso a contenidos de Internet.

BlackBerry Internet Service incorpora compatibilidad con la conectividad directa HTTP y TCP/IP a Internet desde una aplicación de terceros BlackBerry® Java Application.

BlackBerry MDS

Para permitir que BlackBerry® Java Application tenga acceso a los recursos ubicados detrás del firewall de la empresa, BlackBerry® Enterprise Server incluye BlackBerry® Mobile Data System. BlackBerry MDS proporciona proxies HTTP y TCP/IP para BlackBerry Java Application, que permiten al dispositivo BlackBerry comunicarse con la aplicación y servidores Web tras el firewall de la empresa sin software adicional VPN. Las aplicaciones que envían datos mediante BlackBerry Enterprise Server como gateway pueden capitalizar la conectividad simplificada de la empresa, el cifrado de datos y la compresión, y la independencia de la red inalámbrica que ofrece BlackBerry® Enterprise Solution. BlackBerry MDS también ofrece una interfaz abierta, permitiendo a las aplicaciones de servidor detrás del firewall de la empresa insertar contenido en aplicaciones del dispositivo BlackBerry.

Diseño de BlackBerry Java Application

2

Aplicaciones independientes

Puede utilizar las API de BlackBerry® para crear aplicaciones independientes, por ejemplo, juegos y guías de referencia estáticas que se pueden ejecutar como aplicaciones sin conexión. Puede agregar los datos de recursos necesarios a una aplicación antes de compilarlos. Los usuarios de los dispositivos BlackBerry puede instalar la aplicación en la red inalámbrica o con BlackBerry® Desktop Software. Una vez que la aplicación se encuentra instalada en el dispositivo BlackBerry, no es necesario conectarse a la red inalámbrica ni al ordenador.

Aplicaciones con sincronización de escritorio

Puede utilizar las API de BlackBerry® para crear aplicaciones con capacidades de sincronización de escritorio, como guía de referencia y aplicaciones de organizador. El usuario conecta el dispositivo BlackBerry a un ordenador para administrar y sincronizar los datos ubicados en el ordenador.

Research In Motion® no ofrece conductos de HotSync® o cualquier otro módulo de sincronización de base de datos directo. Debe crear un código de sincronización y el usuario del dispositivo BlackBerry debe iniciar el proceso de sincronización de datos manualmente. Una vez que la aplicación se instala en el dispositivo BlackBerry, el usuario del dispositivo BlackBerry debe sincronizar la información manualmente conectando su dispositivo BlackBerry al ordenador con una conexión de serie, una conexión USB, o una conexión Bluetooth®.

Aplicaciones con acceso inalámbrico, sincronización inalámbrica o alerta inalámbrica

Puede utilizar las API de BlackBerry® para crear aplicaciones que inserten contenido de forma proactiva a través de la red inalámbrica en dispositivos BlackBerry en entornos que utilizan BlackBerry® Enterprise Server. Una aplicación BlackBerry® Java Application para dispositivos BlackBerry utiliza una conexión inalámbrica a Internet o la intranet de la empresa para proporcionar a los usuarios de dispositivos BlackBerry acceso a datos remotos y aplicaciones. BlackBerry® Java® Development Environment proporciona API que puede utilizar en aplicaciones para establecer conexiones de red a servidores en Internet o en la intranet de la empresa.

Aplicaciones MIDlet

El modelo de aplicaciones MIDlet forma parte de la especificación MIDP. La clase principal de una aplicación MIDlet siempre extiende la clase MIDlet y debe utilizar métodos para `startApp()`, `pauseApp()` y `destroyApp()`.

Ventajas	Inconvenientes
<ul style="list-style-type: none"> Las aplicaciones se pueden trasladar a otros dispositivos que sean también compatibles con la especificación MIDP. 	<ul style="list-style-type: none"> Las aplicaciones pueden utilizar sólo las API de interfaz de usuario que existen en la biblioteca <code>javax.microedition.lcdui</code>. El modelo asume que todos los procesos de la aplicación terminan cuando se cierra la aplicación. Las aplicaciones no se pueden iniciar automáticamente en segundo plano cuando el dispositivo se enciende.

Aplicaciones CLDC

CLDC es una especificación de un marco para Java® ME. Una aplicación de CLDC puede prolongar la clase `Application` y comenzar con un método estándar `main()`.

La mayor parte de las aplicaciones de ejemplo que incluye BlackBerry® Java® Development Environment utilizan las API de CLDC, las API de BlackBerry y las API de MIDP. Todo el software principal de BlackBerry® Device Software (incluida la lista de mensajes, la lista de contactos, el calendario y el explorador) utilizan las API de CLDC y las API de BlackBerry.

Ventajas	Inconvenientes
<ul style="list-style-type: none"> Las aplicaciones se pueden trasladar a otras plataformas. 	<ul style="list-style-type: none"> Las aplicaciones no pueden ejecutar subprocesos activos en segundo plano una vez se han cerrado. Las aplicaciones no pueden utilizar las API de IPC para intercambiar información con otras aplicaciones. Las aplicaciones no pueden tener acceso a eventos de pantalla táctil en un dispositivo BlackBerry. Las aplicaciones no pueden utilizar los datos del acelerómetro en un dispositivo BlackBerry. Las aplicaciones no pueden integrar BlackBerry® Browser ni BlackBerry® Maps en la interfaz de usuario. Las aplicaciones no pueden cambiar el icono que se muestra en la pantalla de inicio. Las aplicaciones no pueden obtener acceso a los eventos de la bola de desplazamiento ni de la rueda de desplazamiento.

Control de API y firma de código

Al desarrollar BlackBerry® Java Application para dispositivos BlackBerry, puede utilizar sólo las API públicas de Java publicadas y documentadas en los documentos Javadoc™ en BlackBerry® Java® Development Environment. La máquina virtual BlackBerry® Java® Virtual Machine del dispositivo BlackBerry está diseñada para proteger los datos subyacentes y el sistema operativo, así que las aplicaciones no pueden llamar a API sin documentar o no compatibles ni tienen acceso a los datos que no estén expuestos de manera explícita a través de las API. Si intenta utilizar las API de Java que no son públicas, su aplicación recibirá un mensaje de error en tiempo de ejecución.

Las API públicas están abiertas o firmadas. Las API firmadas exponen los métodos para acceder a datos de usuario del dispositivo BlackBerry u otra información en el dispositivo BlackBerry que se considere vulnerable. Puede utilizar API firmadas, pero debe solicitar y recibir un conjunto de claves de firma de código de Research In Motion. Entonces deberá firmar digitalmente su aplicación antes de instalarla en un dispositivo BlackBerry. La firma del código no certifica ni aprueba una aplicación; permite a RIM identificar al autor de una aplicación que utilice API con distinciones en caso de que la aplicación sea malintencionada.

Para solicitar un conjunto de claves de firma de código, visite <http://na.blackberry.com/eng/developers/javaappdev/codekeys.jsp>. Recibirá su conjunto de claves de firma de código en aproximadamente 10 días.

Modelos de objetos

Si utiliza el modelo de aplicaciones MIDlet o CLDC, debe utilizar un enfoque orientado a objetos cuando diseñe su aplicación para el dispositivo BlackBerry®. En un enfoque orientado a objetos, los desarrolladores utilizan objetos para contener el código que sea común a un proceso o función específicos. Por ejemplo, un desarrollador puede utilizar objetos independientes para controlar la actividad de red, el almacenamiento de datos, el procesamiento de datos y la manipulación, y la interacción de la interfaz de usuario. Al diseñar su aplicación, comience con un buen modelo de objetos.

Subprocesamiento múltiple

El sistema operativo de BlackBerry® es un sistema operativo de subprocesos múltiples, que implica que numerosas aplicaciones y procesos se pueden ejecutar de forma activa en el dispositivo BlackBerry al mismo tiempo. Por ejemplo, las aplicaciones pueden utilizar subprocesos en segundo plano para gestionar comunicaciones de red o tareas que requieren una gran cantidad de uso de procesador con el fin de que no afecten al subproceso principal. Si una aplicación crea subprocesos en segundo plano y un usuario del dispositivo BlackBerry cierra la aplicación, los subprocesos en segundo plano pueden permanecer activos.

Práctica recomendada: uso de subprocesamiento múltiple

Haga un uso efectivo de las capacidades del subprocesamiento múltiple del sistema operativo de BlackBerry®. En particular, cree siempre un nuevo subproceso para las conexiones de red u otras operaciones más lentas (más de una décima de segundo). Utilice subprocesos en segundo plano para los oyentes u otros procesos que se ejecuten en segundo plano cuando la aplicación se inicia.

Prácticas recomendadas para escribir una aplicación BlackBerry Java Application eficaz

Práctica recomendada: escritura de código eficaz

Para permitir que BlackBerry® Java® Application utilice los recursos de forma eficaz, considere las siguientes pautas:

- Utilice variables locales.
- Utilice un método de escritura abreviada para evaluar condiciones con el valor Boolean.
- Marque las clases con el valor final.
- Utilice `int` en lugar de `long`.
- Evite la recogida de datos desechables.
- Utilice variables estáticas para `Strings`.
- Evite el constructor `String(String)`.
- Utilice un `String` constante final o no final, dependiendo de los requisitos de la aplicación.
- Escriba repeticiones eficaces.
- Optimice las subexpresiones.
- Optimice las operaciones de división.
- Evite `java.util.Enumeration`.
- Realice asignaciones con `instanceof`.
- Evalúe condiciones con `instanceof`.
- Evite utilizar `StringBuffer.append(StringBuffer)`.
- Evite devolver `null`.
- Evite pasar `null` a los métodos.
- Sea cuidadoso a la hora de pasar `null` a un constructor.
- Utilice `long` para identificadores únicos.
- Salga de las aplicaciones correctamente.
- Imprima el seguimiento de la pila.

Uso de las variables locales

Utilice las variables locales siempre que sea posible. El acceso a las variables locales es más eficaz que el acceso a los miembros de clase.

Uso de la escritura abreviada para evaluar condiciones con el valor Boolean

Para evaluar una condición `Boolean` utilice la escritura abreviada. El código compilado resultante es más corto.

Muestra de código

```
return( boolean_expression );
```

Marcar clases con el valor final

Cuando cree bibliotecas de código, debe marcar las clases con el valor `final` si sabe que los desarrolladores nunca las extenderán. La presencia de la palabra clave `final` permite al compilador generar código más eficiente.

De forma predeterminada, el compilador BlackBerry® Java® Development Environment marca cualquier clase que no extienda en un archivo de aplicación `.cod` como `final`.

Uso de int en lugar de long

En Java®, un valor `Long` es un valor entero de 64 bits. Puesto que los dispositivos BlackBerry® utilizan un procesador de 32 bits, las operaciones pueden ejecutarse dos o cuatro veces más rápido si utiliza un valor `int` en lugar de `Long`.

Cómo evitar la recogida de datos desechables

Evite llamar a `System.gc()` para realizar una operación de recogida de datos desechables ya que puede que tarde mucho en dispositivos BlackBerry® con memoria disponible limitada. Deje que BlackBerry® Java® Virtual Machine recoja datos desechables.

Uso de las variables estáticas para Strings

Al definir campos estáticos (también denominados campos de clase) del tipo `String`, puede aumentar la velocidad de la aplicación utilizando variables estáticas (no `final`) en vez de constantes (`final`). Lo contrario es verdad para los tipos de datos básicos, como `int`.

Por ejemplo, puede crear un objeto `String` de la siguiente manera:

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

Para esta constante estática (denotada por la palabra clave `final`), cada vez que utilice la constante se crea una instancia temporal `String`. El compilador elimina `"x"` y la sustituye con la cadena `"example"` en el código de bytes, para que BlackBerry® Java® Virtual Machine realice una búsqueda de tabla hash cada vez que haga referencia a `"x"`.

Por contra, para una variable estática (sin palabra clave `final`), el `String` se crea una vez. BlackBerry JVM realiza la búsqueda de tabla hash sólo cuando inicializa `"x"`, así que el acceso es más rápido.

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

Puede utilizar constantes públicas (es decir, campos `final`), pero debe marcar las variables como privadas.

Cómo evitar el constructor `String(String)`

En BlackBerry® Java Application, cada cadena citada es una instancia de la clase `java.lang.String`. Cree una `String` sin utilizar el constructor `java.lang.String(String)`.

Muestra de código

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

Escritura de repeticiones eficaces

Si es probable que el contenedor contenga más de un elemento, asigne el tamaño a una variable local.

Si la orden en la que itera a través de los elementos no es importante, puede iterar en dirección opuesta para evitar variables locales adicionales en la pila y para acelerar la comparación.

Muestra de código

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

Optimización de subexpresiones

Si utiliza la misma expresión dos veces, debe utilizar una variable local.

Muestra de código

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

Optimización de operaciones de división

Las operaciones de división pueden ser lentas en los dispositivos BlackBerry® debido a que el procesador no dispone de una instrucción de división de hardware.

Cuando el código divide un número positivo entre dos, debe utilizar, en su lugar, la teclas Mayúscula derecha entre uno (>> 1). Utilice la teclas Mayúsculas derecha (>>) sólo cuando sepa que está trabajando con un valor positivo.

Muestra de código

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

Cómo evitar java.util.Enumeration

Evite utilizar `java.util.Enumeration` a menos que desee ocultar datos (en otras palabras, para devolver una enumeración de los datos en lugar de los mismos datos). Pedir a un vector o una tabla hash un objeto `Enumeration` es lento y crea datos desechables innecesarios. Si otro subproceso puede modificar el vector, sincronice la iteración. Java® SE utiliza un objeto `Iterator` para operaciones similares, pero los objetos `Iterator` no están disponibles en Java® ME.

Muestra de código

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

Realización de asignaciones mediante instanceof

Utilice `instanceof` para valorar si una asignación es o no satisfactoria.

Muestra de código

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

Evaluación de las condiciones mediante instanceof

Para generar un código más pequeño y más rápido, si evalúa una condición mediante `instanceof`, no debe evaluar explícitamente si la variable es null.

Muestra de código

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

Cómo evitar StringBuffer.append (StringBuffer)

Para adjuntar un búfer `String` a otro, BlackBerry® Java Application debe utilizar `net.rim.device.api.util.StringUtilities.append (StringBuffer dst, StringBuffer src[, int offset, int length])`.

Muestra de código

```
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)); }
```

Cómo evitar devolver un valor null

Si escribe un método público que devuelve un objeto, el método debe devolver un valor null sólo bajo las siguientes condiciones:

- La aplicación espera que ocurra un valor null durante el funcionamiento normal de la aplicación.
- El parámetro Javadoc™™ `@return` para el método indica que método que null es un valor de devolución posible.

Si la aplicación no espera un valor de devolución null, el método debe mostrar una excepción apropiada, que fuerce a la persona que llama a tratar específicamente el problema. Puede que la persona que llama del método no necesite comprobar la existencia de un valor de devolución null a menos que la persona que llama emita una excepción del valor null.

Cómo evitar pasar un valor null a los métodos

No pase parámetros null a un método de API a menos que los estados de referencia API indiquen explícitamente que el método es compatible con ellos.

Precaución al pasar un valor null a un constructor

Para evitar ambigüedades al pasar un valor null a un constructor, asigne el valor null al objeto correspondiente.

Si una clase tiene dos o más constructores, pasar un parámetro null podría no identificar de forma única qué constructor utilizar. Como resultado, el compilador comunica un error.

Al asignar valores null a los objetos apropiados, debe indicar de forma precisa qué constructor debe utilizar el compilador. Esta práctica también proporciona compatibilidad si versiones posteriores de la API agregan nuevos constructores.

Muestra de código

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

Uso de long como identificador único

Utilice un identificador `Long` en vez de un identificador `String` para constantes únicas, como GUID, claves de tablas hash e identificadores de estado o contexto.

Para que los identificadores sean únicos a lo largo de la BlackBerry® Java Application, utilice claves que genera la aplicación según un valor hash de `String`. En la entrada `String`, incluya la suficiente información para que el identificador sea único. Por ejemplo, utilice un nombre de paquete completo como `com.rim.samples.docs.helloworld`.

Salida de las aplicaciones correctamente

Antes de invocar `System.exit(int status)`, lleve a cabo una limpieza, elimine los objetos del almacén en tiempo de ejecución que las aplicaciones ya no necesiten.

Impresión del seguimiento de pila

Al depurar una aplicación, para ver el seguimiento de pila, utilice una instancia `Throwable`.

Muestra de código

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

Práctica recomendada: uso apropiado de los objetos

Para permitir que BlackBerry® Java Application utilice eficazmente los recursos de la memoria, considere las siguientes preguntas:

- Dado el tamaño de una aplicación, ¿son necesarios todos los objetos?
- ¿Puede la aplicación almacenar cualquier objeto que represente valores básicos, como `Long`, `Integer` y `Boolean`, como valores básicos y no como objetos?
- ¿Son necesarios todos los objetos constantes?
- ¿Existe cualquier instancia de `Vector` y `HashTable`? ¿Son estas instancias necesarias? En tal caso, cuántos identificadores de `Object` no se utilizan en el `Vector` o bien, `HashTable` porque el tamaño inicial es mayor del necesario?
- ¿Cuántos `Objects` crea su aplicación y desecha más tarde? En otras palabras, ¿cuántos `Objects` crea su aplicación?

Práctica recomendada: reducción del tamaño del código compilado

Para reducir el tamaño del código compilado, considere las siguientes pautas:

- Configure un acceso apropiado.
- Evite la creación de interfaces.
- Utilice clases internas estáticas.
- Evite la inicialización innecesaria de campos.
- Importe las clases individuales.

Configuración de un acceso apropiado

Al crear bibliotecas de código, puede reducir significativamente el tamaño del código compilado utilizando los modificadores de acceso apropiados para campos y métodos. Defina campos como privados siempre que sea posible. Además de ser una excelente práctica de codificación, esto permite al compilador optimizar el archivo `.cod`. Siempre que sea posible, utilice el acceso predeterminado (`package`) en vez de acceso público (es decir, omita las palabras clave `public` y `protected`).

Creación sin interfaces

Al crear bibliotecas de API, evite crear interfaces a menos que tenga previsto realizar múltiples implementaciones de la API. Las interfaces generan código más grande y lento.

Uso de las clases internas estáticas

Al utilizar una clase interna para ocultar una clase dentro de otra, y si la clase interna no hace referencia al objeto de la clase externa, marque la clase interna como estática. Esta acción evita la creación de una referencia a la clase externa.

Si utiliza una clase interna para la identificación de nombres, márkuela como estática.

Muestra de código

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

Cómo evitar la inicialización de campos innecesaria

Siempre que sea posible, permita que los campos se inicialicen automáticamente de la siguiente manera:

- las referencias de objetos se inicializan como null
- int, byte, o long se inicializa como 0
- Boolean se inicializa como false

Debe inicializar explícitamente las variables locales en un método.

Muestra de código

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

Importación de clases individuales

Una aplicación BlackBerry® Java Application que utiliza sólo un pequeño número de clases de un paquete sólo debe importar las clases individuales.

Muestra de código

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

Compatibilidad de varios idiomas

BlackBerry® Integrated Development Environment incluye un mecanismo de recursos para crear recursos de cadena. La API de localización forma parte del paquete `net.rim.device.api.i18n`. Las aplicaciones MIDP no son compatibles con la localización.

BlackBerry Integrated Development Environment almacena recursos para una configuración regional en un objeto `ResourceBundle`. `ResourceBundleFamily` contiene una recopilación de `ResourceBundles`, que agrupa los recursos para una aplicación. La aplicación puede cambiar los idiomas, dependiendo de la configuración regional del usuario del dispositivo BlackBerry, sin requerir nuevos paquetes de recursos.

Puede utilizar BlackBerry Integrated Development Environment para compilar cada paquete de recursos en un archivo `.cod` compilado independientemente. Puede cargar los archivos `.cod` apropiados en los dispositivos BlackBerry con los otros archivos `.cod` de la aplicación.

Los recursos se organizan según una jerarquía basada en la capacidad heredable. Si una cadena no está definida en una configuración regional, se utiliza una cadena de la configuración regional siguiente más cercana.

Práctica recomendada: almacenamiento de cadenas de texto en archivos de recursos

En lugar de utilizar texto en código fuente, diseñe aplicaciones para utilizar los archivos de recursos para la localización (adáptelas a los idiomas o regiones específicas).

Considere las siguientes directrices:

- Almacene las cadenas de texto para cada configuración regional en un solo archivo de recursos.
- En el código fuente, utilice identificadores únicos para utilizar los archivos de recursos apropiados.
- Diseñe la aplicación para recuperar dinámicamente el archivo de recursos apropiado para mostrarlo al usuario del dispositivo BlackBerry® según la configuración regional del usuario del dispositivo BlackBerry.

Compatibilidad multimedia

Compatibilidad de audio

Puede crear una aplicación BlackBerry® Java Application que funcione con los formatos de audio compatibles con el dispositivo BlackBerry. El tipo de formato de audio que es compatible con un dispositivo BlackBerry depende del número de modelo del dispositivo BlackBerry.

Para obtener más información sobre la compatibilidad de audio en un dispositivo BlackBerry, visite www.blackberry.com/developers.

Compatibilidad de la cámara

En un dispositivo BlackBerry® que tenga cámara, cuando un usuario de dispositivo BlackBerry toma una fotografía, el dispositivo BlackBerry almacena la imagen en el sistema de archivos del dispositivo. Una aplicación del dispositivo BlackBerry puede obtener acceso a las imágenes utilizando la API de conexión de archivo para Java ME disponible en la versión 4.2 o posterior de BlackBerry® Java® Development Environment. Una aplicación del dispositivo BlackBerry puede invocar la aplicación Cámara y escuchar eventos cuando las imágenes se agreguen al sistema de archivos.

El dispositivo BlackBerry con BlackBerry® Desktop Software versión 4.7 o posterior, puede permitir a un usuario del dispositivo BlackBerry enviar una imagen a una aplicación del dispositivo BlackBerry. Si un usuario del dispositivo BlackBerry utiliza la aplicación Cámara para tomar una fotografía u obtiene acceso a una imagen a través del explorador de archivos, BlackBerry Device Software agrega un elemento de menú a la aplicación Cámara o muestra un cuadro de diálogo en el explorador de archivos. Si un usuario del dispositivo BlackBerry selecciona el elemento de menú o el cuadro de diálogo, BlackBerry Device Software envía una solicitud a una aplicación del dispositivo BlackBerry que es un gestor de contenidos de la imagen. La aplicación del dispositivo BlackBerry puede aceptar la solicitud y abrir la imagen.

Con el fin de que una aplicación del dispositivo BlackBerry pueda recibir una imagen, la aplicación debe implementar la interfaz `javax.microedition.content.ContentHandler` para registrarse como gestor de contenidos de un tipo de imagen, por ejemplo, .jpg. El registro como gestor de contenidos asocia la aplicación del dispositivo BlackBerry con el tipo de imagen y permite a otras aplicaciones del dispositivo BlackBerry enviar estos archivos a la aplicación del dispositivo BlackBerry. Para obtener más información acerca del uso de la interfaz `ContentHandler`, visite <http://jcp.org/en/jsr/detail?id=211>.

Compatibilidad de vídeo

Puede crear una aplicación de dispositivo BlackBerry® que pueda utilizar un objeto `javax.microedition.media.Player` y subclases de la clase `javax.microedition.media.Control` para reproducir un vídeo. Una aplicación del dispositivo BlackBerry invoca uno de los métodos `net.rim.device.api.ui.Manager.createPlayer()` para recuperar un objeto `Player` y recuperar un vídeo de fuentes diferentes como un archivo o una secuencia. Un objeto `Player` permite al usuario del dispositivo BlackBerry utilizar la aplicación como un reproductor multimedia y llevar a cabo acciones como la reproducción y la pausa de la reproducción del vídeo.

La aplicación del dispositivo BlackBerry invoca `Player.realize()` para permitir a la aplicación recuperar los recursos que necesita, como la ubicación de un archivo de vídeo, para reproducir un vídeo. La aplicación del dispositivo BlackBerry invoca `Player.prefetch()` para permitir a la aplicación preparar la reproducción de un vídeo. La aplicación del dispositivo BlackBerry puede invocar `Player.getControls()` para recuperar una matriz de objetos `Control` que represente las operaciones o funciones con las que un `Player` es compatible. Por ejemplo, un objeto `javax.microedition.media.control.VolumeControl` permite a una aplicación del dispositivo BlackBerry controlar el volumen que utiliza para reproducir un vídeo. Un objeto `javax.microedition.media.control.VideoControl` permite a una aplicación del dispositivo BlackBerry controlar cómo se reproduce un vídeo.

Una vez la aplicación de dispositivo BlackBerry recupera información en los objetos `javax.microedition.media.Control` con la que es compatible, la aplicación puede invocar `Player.getControl()`, utilizando como parámetro una cadena con el nombre del control, para recuperar un objeto de `Control` específico. Por ejemplo, `VideoControl myVC = (VideoControl) Player.getControl("<VideoControl>")`. La aplicación del dispositivo BlackBerry puede establecer las propiedades del objeto `Control` invocando los métodos del objeto `Control`. La aplicación del dispositivo BlackBerry invoca `Player.start()` para iniciar la reproducción de un vídeo e invoca `Player.close()` para detener la reproducción de un vídeo.

Puede crear una aplicación de dispositivo BlackBerry que pueda utilizar el visor de la aplicación de cámara para registrar un vídeo en un archivo `.3gp` o transmitir el vídeo a un objeto `java.io.ByteArrayOutputStream`. La aplicación del dispositivo BlackBerry invoca `Manager.createPlayer("capture://video")` para recuperar un objeto `Player` que sea compatible con las interfaces `VideoControl` y `RecordControl`.

La aplicación del dispositivo BlackBerry invoca `Player.getControl("RecordControl")` y asigna el valor de retorno como objeto `RecordControl`. Un objeto `RecordControl` permite a una aplicación del dispositivo BlackBerry controlar cómo graba vídeo. Al invocar `RecordControl.setRecordLocation("file:///<filename>")` se configura la ubicación del archivo `.3gp`. Al invocar `RecordControl.setRecordStream(ByteArrayOutputStream output)` se configura `ByteArrayOutputStream` para realizar la transmisión. La aplicación del dispositivo BlackBerry invoca `RecordControl.startRecord()` seguido de `Player.start()` para iniciar la grabación de un vídeo. La aplicación del dispositivo BlackBerry invoca `RecordControl.commit()` seguido de `Player.stop()` para guardar el vídeo y detener la grabación.

Puede crear una aplicación de dispositivo BlackBerry que pueda reproducir vídeo en BlackBerry® Browser. La aplicación del dispositivo BlackBerry invoca `net.rim.blackberry.api.browser.Browser.getDefaultSession()` para recuperar un objeto `net.rim.blackberry.api.browser.BrowserSession`. El servicio `BrowserSession` permite a la aplicación del dispositivo BlackBerry interactuar con BlackBerry Browser. La aplicación del dispositivo BlackBerry invoca `BrowserSession.displayPage()` utilizando como parámetro un `String` con el formato "file:///<locationOfVideo>" donde <locationOfVideo> es la ubicación del vídeo. `BrowserSession.displayPage()` muestra un vídeo en BlackBerry Browser.

Puede crear una aplicación de dispositivo BlackBerry que pueda reproducir un vídeo que necesite licencias de gestión de derechos digitales. La aplicación del dispositivo BlackBerry utiliza la clase `net.rim.device.api.drm.DRMManager` para instalar licencias de DRM para un vídeo. Un dispositivo BlackBerry tiene un objeto `DRMManager`. Al invocar `DRMManager.getLicenseChallenge(String contentURL, String[] permissions)` se recupera una matriz de bytes que contiene información acerca de las licencias de DRM que debe obtener e instalar la aplicación del dispositivo BlackBerry con el fin de reproducir un vídeo. Al invocar `DRMManager.installRights(String contentURL, byte[] rightsData, DRMRightsInstallCallback callback)` se instalan las licencias de un vídeo.

Puede permitir a un usuario de dispositivo BlackBerry enviar un vídeo a una aplicación de dispositivo BlackBerry. Para que la aplicación de dispositivo BlackBerry reciba y utilice el archivo de vídeo, la aplicación debe implementar la interfaz `javax.microedition.content.ContentHandler` para registrarse como un controlador de contenido. Si un usuario

de dispositivo BlackBerry obtiene acceso a un archivo de vídeo a través de la aplicación Cámara o el explorador de archivos, aparece un elemento de menú que permite al usuario del dispositivo BlackBerry enviar el archivo de vídeo a aplicaciones del dispositivo BlackBerry, gestores de contenido.

Interfaz de usuario y diseño de navegación

3

Escritura y navegación del usuario de dispositivos BlackBerry

Los dispositivos BlackBerry® incluyen un teclado, una rueda o bola de desplazamiento y una tecla Escape para la escritura y navegación. La tecla Escape proporciona una manera sencilla a los usuarios de dispositivos BlackBerry para retroceder a la pantalla anterior o eliminar un menú o cuadro de diálogo de la pantalla.

BlackBerry® Java Application para dispositivos BlackBerry debe seguir el siguiente modelo de escritura y navegación en la medida de lo posible.

- Al hacer clic en la rueda o bola de desplazamiento se invoca normalmente un menú.
- Pulsar la tecla Escape cambia la pantalla a la pantalla anterior o cierra la aplicación en la pantalla principal.

De forma predeterminada, los objetos de pantalla de BlackBerry ofrecen esta funcionalidad sin personalización; sin embargo, debe agregar elementos del menú, así como una interfaz de usuario adicional y lógica de navegación.

Rueda de desplazamiento frente a bola de desplazamiento

Sensibilidad de la bola de desplazamiento

La sensibilidad de la bola de desplazamiento se refiere a cuánto movimiento la bola necesita para que el sistema identifique el movimiento como un evento de navegación y para distribuir un evento de navegación en la capa de software. El hardware del dispositivo BlackBerry® mide el movimiento físico de la bola de desplazamiento mediante unidades denominadas tics. Si el número de tics a lo largo de un eje supera el umbral del sistema o de una aplicación BlackBerry® Java Application, se distribuye un evento de navegación a lo largo de ese eje en la capa del software y el sistema restablece el recuento de tics en cero. Los recuentos de tics también se restablecen en cero una vez se supera cierto tiempo de inactividad.

Puede utilizar la API de la bola de desplazamiento para establecer la sensibilidad de la bola de desplazamiento. Una sensibilidad alta de la bola de desplazamiento equivale a un umbral de tics más pequeño, lo que significa que pequeños movimientos de la bola de desplazamiento activarán eventos de navegación. Por el contrario, una sensibilidad baja de la bola de desplazamiento equivale a un umbral mayor de tics, lo cual significa que son necesarios movimientos más grande de la bola de desplazamiento para generar eventos de navegación.

Desplazamiento de la bola de desplazamiento

Puede utilizar la API de la bola de desplazamiento para filtrar los datos de los movimientos de la bola que el hardware del dispositivo BlackBerry® envía a la capa del software. La API de la bola de desplazamiento puede filtrar "el ruido de movimiento" o los movimientos no deseados.

Puede utilizar también la API de la bola de desplazamiento para cambiar configuraciones como la aceleración del movimiento de la bola de desplazamiento. El aumento de la configuración de la aceleración del movimiento de la bola de desplazamiento puede hacer que la capa de software detecte que los movimientos de la bola se mueven a una velocidad mayor que la velocidad detectada por el hardware del dispositivo BlackBerry, siempre que el usuario desplace continuamente la bola de desplazamiento. La sensibilidad de la bola de desplazamiento aumenta temporalmente a medida que el usuario desplaza la bola sin detenerse.

Rueda de desplazamiento

Los dispositivos BlackBerry® anteriores a BlackBerry® Pearl™ 8100 Series no incluyen una bola de desplazamiento. En su lugar, incluyen una rueda de desplazamiento en el lado derecho del dispositivo. La rueda de desplazamiento constituye el control principal para la navegación del usuario.

Los usuarios pueden

- desplazar la rueda de desplazamiento para mover el cursor verticalmente
- desplazar la rueda de desplazamiento mientras pulsan la tecla Alt para mover el cursor horizontalmente
- hacer clic en la rueda de desplazamiento para seleccionar objetos o abrir el menú

Directrices

- Familiarícese con los modelos de navegación con rueda de desplazamiento y con bola de desplazamiento y compruebe que la aplicación funciona bien con ambos.

Creación de una interfaz de usuario que sea coherente con las interfaces de usuario estándar de BlackBerry

Puede utilizar las API estándar de MIDP y las API de la interfaz de usuario de BlackBerry® para crear interfaces de usuario de BlackBerry® Java Application.

Las API de la interfaz de usuario de BlackBerry son una biblioteca de componentes de interfaz de usuario diseñados para ofrecer diseños y comportamientos predeterminados que son coherentes con las aplicaciones principales del dispositivo BlackBerry.

- Los componentes de pantalla ofrecen un diseño estándar de pantalla, un menú predeterminado y un comportamiento estándar cuando el usuario del dispositivo BlackBerry pulsa la tecla Escape o hace clic en la rueda o bola de desplazamiento.
- Los componentes de Campo proporcionan elementos de interfaz de usuario estándar para la selección de fechas, opciones, casillas de verificación, listas, campos de texto, etiquetas y controles de la barra de progreso.
- Los administradores de diseño ofrecen una aplicación con la capacidad de distribuir componentes en una pantalla de dispositivo BlackBerry de forma estándar, tanto horizontalmente como verticalmente, o en un eje izquierda a derecha.

Puede utilizar las API de la interfaz de usuario de BlackBerry para crear interfaces de usuario que incluyan tablas, cuadrícula y otras características especializadas. BlackBerry Java Development Environment utiliza un modelo de eventos estándar de Java para recibir y responder a tipos específicos de eventos. Las aplicaciones pueden recibir y responder a eventos de usuarios de

dispositivos BlackBerry, como cuando el usuario del dispositivo BlackBerry hace clic en la rueda o bola de desplazamiento, o escribe en el teclado y en los eventos del sistema, por ejemplo, alertas globales, cambios del reloj en tiempo real y conexiones de puerto USB.

Gestión de memoria

4

Gestión de memoria

BlackBerry® Java® Virtual Machine administra el uso de la memoria en el dispositivo BlackBerry. BlackBerry JVM asigna la memoria, realiza la recogida de datos desechables e intercambia automáticamente los datos entre la memoria SRAM y la memoria flash. BlackBerry JVM también debe compartir la memoria disponible entre las aplicaciones del dispositivo BlackBerry y BlackBerry® Java Application. Las capacidades de la memoria representan la cantidad total de memoria disponible, que es más grande que la memoria de trabajo disponible cuando todas las aplicaciones y los datos de la aplicación asociados existen en el dispositivo BlackBerry.

Memoria del dispositivo BlackBerry

Los dispositivos BlackBerry® incluyen los siguientes tipos de la memoria:

Memoria	Descripción
flash	El sistema operativo BlackBerry y todos los módulos de aplicaciones se almacenan en la memoria flash. Cuando el usuario del dispositivo BlackBerry activa el dispositivo, el sistema operativo principal y los módulos de BlackBerry® Java Application utilizan aproximadamente de 10 MB a 15 MB de memoria flash, dependiendo de la versión. La memoria flash puede almacenar los mensajes de correo electrónico del usuario del dispositivo BlackBerry, los datos del organizador y cualquier otra información personal, así como los datos que BlackBerry Java Application almacena en la memoria.
SRAM	SRAM controla los objetos de datos provisionales y los procesos en tiempo de ejecución.
tarjeta de memoria extensible microSD	La tarjeta microSD almacena archivos multimedia, documentos y datos de BlackBerry Java Application.

Recursos claves para reservar

- Memoria flash: el espacio de almacenamiento constante disponible en el dispositivo BlackBerry® es una cantidad fija de memoria flash, normalmente de 8 MB a 64 MB.

- Identificadores de objetos constantes: los identificadores asignados a cada objeto constante son consumidos únicamente por objetos constantes. La cantidad de memoria flash en el dispositivo BlackBerry determina el número fijo de identificadores de objetos constantes en el sistema.
- Identificadores de objeto: cada objeto y matriz de elementos básicos tienen un procesamiento de objetos asociados. La cantidad de memoria flash en el dispositivo BlackBerry determina el número fijo de identificadores de objetos en el sistema.

Práctica recomendada: reducción del uso de la memoria

Para reducir la memoria en tiempo de ejecución, considere las siguientes pautas:

- Utilice los tipos básicos (como `int` o bien, `Boolean`) en vez de objetos (como `String` o bien, `Integer`).
- No dependa enteramente del recolector de datos desechables.
- Evite crear muchos objetos rápidamente.
- Configure las referencias a objetos en `null` cuando termine de utilizarlas.
- Vuelva a utilizar objetos tantas veces como sea posible.
- Traslade el procesamiento más complicado al servidor. Por ejemplo, puede filtrar o clasificar los datos antes de enviarlos al dispositivo BlackBerry®.

Gestión de una disponibilidad de memoria baja

El administrador de memoria baja gestiona los recursos de memoria en el dispositivo BlackBerry® cuando los recursos de memoria disponibles caen por debajo de un determinado umbral. El administrador de memoria baja intenta liberar memoria utilizada para proporcionar más memoria disponible en el dispositivo BlackBerry. Todas las aplicaciones, incluidas BlackBerry® Java Application, deben funcionar con el administrador de memoria baja para liberar tanta memoria como sea posible cuando el dispositivo BlackBerry está bajo de recursos de memoria.

Identificación de una disponibilidad baja de memoria en un dispositivo BlackBerry

Las siguientes condiciones pueden provocar que el administrador de memoria baja intente liberar recursos de memoria:

- La cantidad de memoria flash disponible en el dispositivo BlackBerry® cae por debajo de un determinado umbral. El umbral de la memoria flash depende de la cantidad de RAM libre en el sistema. El umbral de la memoria flash varía entre 400 KB y 800 KB.
- El número de identificadores de objetos constantes disponibles en el dispositivo BlackBerry cae por debajo de 1.000.
- El número de identificadores de objetos disponibles en el dispositivo BlackBerry cae por debajo de 1.000.

Conservación de recursos

Práctica recomendada: uso de una selección de estructura de datos eficaz

La selección de la estructura de datos define cuántos identificadores de objetos y cuánta memoria flash consume BlackBerry® Java Application. Una selección incorrecta de la estructura de datos puede consumir recursos clave sin mejorar la funcionalidad de BlackBerry Java Application o la experiencia de usuario de dispositivos BlackBerry.

Considere las siguientes directrices:

- La estructura de datos debe contener el mínimo número posible de objetos, especialmente cuando utiliza objetos de alto nivel como un `Vector` o un `HashTable`. Estas clases ofrecen una funcionalidad considerable pero no constituyen mecanismos eficaces de almacenamiento y debe evitar utilizarlos en el almacén constante si es posible.
- Siempre que sea posible, utilice valores básicos de los objetos, ya que éstos reducen el número de los identificadores de objetos consumidos en el dispositivo BlackBerry. Una matriz de valores básicos es un objeto y consume un identificador de objetos.
- Los objetos `String` son tan eficaces como las matrices de `byte`. Un objeto `String` consume sólo un identificador de objetos y es equivalente si su aplicación almacena todos los caracteres como un `byte`. En otras palabras, el valor de cada carácter es menor que o igual al valor decimal de 255. Si la aplicación no puede almacenar caracteres como un `byte`, puede almacenar los caracteres como un objeto `String` porque equivale a almacenar una matriz `char`.

Práctica recomendada: consolidación de objetos en grupos de objetos

Uno de los errores más comunes con los que se encuentran los desarrolladores de aplicaciones es el agotamiento de los identificadores de objetos constantes. La cantidad de memoria flash en el dispositivo BlackBerry® determina el número fijo de los identificadores de objetos constantes que están disponibles en el sistema. Dependiendo de la selección de la estructura de datos, los registros almacenados pueden agotar rápidamente el número de identificadores de objetos constantes. Un objeto constante consume un identificador de objetos constante y un identificador de objetos. Un objeto provisional consume sólo un identificador de objetos.

Por ejemplo, un registro que contenga diez campos `String`, representando elementos como un nombre, un número de teléfono y una dirección, consume 11 identificadores de objetos constantes, uno para el objeto registrado y otro para cada `String`. Si BlackBerry® Java Application contiene 3.000 registros, la aplicación consume 33.000 identificadores de objetos constantes, lo cual excede el número de identificadores de objetos constantes disponibles en un dispositivo BlackBerry con 16 MB de memoria flash.

Puede utilizar la clase `net.rim.device.api.system.ObjectGroup` para consolidar los identificadores de objetos para un objeto en un grupo. Utilizando el ejemplo del párrafo anterior, si agrupa el registro, éste consume un identificador de objetos constante en vez de 11. Los identificadores de objetos para los campos `String` se consolidan bajo el identificador de objetos de registro.

Cuando consolida los identificadores de objetos en un grupo, el identificador de objetos es de sólo lectura. Debe desagrupar el objeto antes de poder cambiarlo. Una vez realice los cambios, agrupe de nuevo el objeto. Si intenta cambiar un objeto agrupado sin primero desagruparlo, se produce un `ObjectGroupReadOnlyException`.

Desagrupar un objeto tiene un impacto en el rendimiento. El sistema crea una copia del objeto agrupado y asigna identificadores a cada uno de los objetos dentro de ese grupo. Por lo tanto, los objetos sólo deben desagruparse cuando sea estrictamente necesario.

Recogida de datos desechables en un dispositivo BlackBerry

Recogida de datos desechables de la RAM en un dispositivo BlackBerry

BlackBerry® Java® Virtual Machine inicia una operación de recogida de datos desechables de la RAM sólo cuando BlackBerry JVM no puede asignar un objeto debido a falta de espacio en la RAM. La operación de recogida de datos desechables tarda normalmente entre 500 y 600 milisegundos en ejecutarse. La operación de recogida de datos desechables elimina cualquier variable recién asignada a la que la RAM no haga referencia. Para asegurarse de que la falta de una referencia en la RAM es una condición suficiente para eliminar el objeto, la operación de recogida de datos desechables de la RAM sólo se puede realizar cuándo los objetos no se localicen en la memoria flash.

Recogida de datos desechables en un dispositivo BlackBerry

La operación completa de recogida de datos desechables se ejecuta durante 1 segundo como media y debe tardar menos de 2 segundos en terminar. La operación completa de recogida de datos desechables realiza las siguientes acciones:

- Realiza una operación de recogida de datos desechables de la RAM.
- Marca objetos en la memoria flash a los que ya no se hace referencia o que ya no existen.
- Genera identificadores de objetos no constantes en la RAM y la memoria flash.

El sistema puede iniciar una operación completa de recogida de datos desechables en las siguientes situaciones:

- BlackBerry® Java® Virtual Machine no puede asignar un objeto por una falta de espacio disponible en la RAM.
- Un proceso está a punto de exceder el tamaño de la pila actualmente asignado.
- BlackBerry JVM no puede asignar un objeto nuevo porque los identificadores de objetos no están disponibles.
- El dispositivo BlackBerry está inactivo.

Recogida de datos desechables inactiva en un dispositivo BlackBerry

La recogida de datos desechables no se produce cada vez que el dispositivo BlackBerry® pasa a estado inactivo. Esto ocurre solamente cuando el sistema considera que una operación de recogida de datos desechables puede ser beneficiosa para el rendimiento óptimo de sistema y un rendimiento maximizado de la batería.

Para mejorar el rendimiento sin afectar a la experiencia de usuario del dispositivo BlackBerry, el sistema intenta realizar las siguientes operaciones de recogida de datos desechables cuando el dispositivo BlackBerry pasa a estado inactivo:

- Se puede producir una operación completa de recogida de datos desechables cuando el dispositivo BlackBerry pasa a estado inactivo durante un período relativamente pequeño de tiempo.
- Se puede producir una operación exhaustiva de recogida de datos desechables cuando el dispositivo BlackBerry pasa a estado inactivo durante un período significativo de tiempo.

Administración de datos

5

El dispositivo BlackBerry® ofrece API para almacenar datos en la memoria constante del dispositivo BlackBerry. Las API del almacén constante de BlackBerry y las API RMS de MIDP (compatibilidad con JSR 37 y JSR 118) están disponible en todos los dispositivos BlackBerry basados en Java®. Un dispositivo BlackBerry que ejecuta BlackBerry® Device Software versión 4.2 o posterior ofrece un sistema de archivos tradicional y compatibilidad para guardar el contenido directamente en el sistema de archivos mediante las API de JSR 75. Tanto con las API del almacén constante de BlackBerry como las API RMS de MIDP, puede almacenar los datos de forma constante en la memoria flash. Los datos persisten incluso si retira la batería del dispositivo BlackBerry.

Compatibilidad con las API para almacenar datos en la memoria constante

API del almacén constante

Las API del almacén constante de BlackBerry® se han diseñado para proporcionar una interfaz de almacenamiento de datos flexible y sólida. Con las API del almacén constante de BlackBerry, puede guardar objetos completos de Java® en la memoria sin tener que serializar los datos en primer lugar. Al iniciar la aplicación, puede recuperar el objeto de Java de la memoria y procesar la información.

Las API del almacén constante de BlackBerry no ofrecen un modelo de base de datos relacional. Debe crear un modelo de objetos efectivo y administrar las relaciones entre objetos, tal como sea necesario, mediante índices y tablas hash.

API del sistema de administración de registros MIDP

Las API de RMS proporcionan un sistema de administración de registros sencillo que le permite crear un objeto de almacén de datos y mantener una serie de registros en ese objeto. Cada registro es una matriz de bytes, por lo que debe serializar en primer lugar sus datos en un formato de matriz de bytes antes de almacenarlos de forma local. Las API de RMS no ofrecen ningún índice o relación inherente entre los registros. El límite de tamaño para un único almacén de datos de RMS es de un máximo de 64 KB. Una aplicación puede crear múltiples almacenes de datos de RMS para mantener cantidades más grande de datos. Las API de RMS forman parte de la especificación estándar MIDP, así que todos los dispositivos que sean compatibles con MIDP también lo son con las API de RMS.

API de conexiones de archivo

La API de conexión de archivo ofrece un sistema de archivos tradicional y la compatibilidad para guardar los datos directamente en el sistema de archivos en el dispositivo BlackBerry® o en una tarjeta microSD. Puede ver los datos en el sistema de archivos y mover los datos a un ordenador mediante Windows®.

Almacenamiento en medios extraíbles

Acceso a los datos en la tarjeta multimedia microSD

El servicio `javax.microedition.io.file` es compatible con las API de conexión de archivos JSR 75 y se utiliza en aplicaciones para obtener acceso al sistema de archivos para la tarjeta multimedia microSD. También puede implementar la interfaz `FileConnection` para acceder a melodías de llamada e imágenes de cámara del dispositivo BlackBerry®.

Clase o interfaz	Descripción
<code>ConnectionClosedException</code>	Esta excepción se produce cuando una aplicación invoca un método en una conexión de archivos cerrada.
<code>FileConnection</code>	Una aplicación puede utilizar esta API para obtener acceso a los archivos o directorios.
<code>FileSystemListener</code>	Una aplicación puede utilizar esta API para recibir notificaciones de estado cuando la aplicación agrega o elimina una raíz del sistema de archivos.
<code>FileSystemRegistry</code>	Una aplicación puede utilizar esta API como registro central para oyentes del sistema de archivos que reciben la adición o eliminación de los sistemas de archivos.
<code>IllegalModeException</code>	Esta excepción se produce cuando un método requiere un modo de seguridad específico (por ejemplo, LECTURA o ESCRITURA) y la conexión abierta no se encuentra en ese modo.

Copia de seguridad y sincronización de los datos

BlackBerry® Desktop Manager ofrece una herramienta de copia de seguridad y restauración que el usuario del dispositivo BlackBerry puede utilizar para guardar datos del dispositivo en un archivo en el ordenador y para restaurar los datos en el dispositivo BlackBerry.

Si una aplicación utiliza la API de sincronización, BlackBerry Desktop Manager puede realizar copias de seguridad y restaurar la base de datos de la aplicación a la vez que otras bases de datos del dispositivo BlackBerry. Puede utilizar la API de sincronización para crear archivos de datos o para rellenar las bases de datos de la aplicación la primera vez que el dispositivo BlackBerry se conecta al ordenador del usuario del dispositivo BlackBerry.

Para sincronizar los datos en fuentes de datos remotas, debe integrar la lógica de sincronización en BlackBerry Java® Application. La mayoría de las aplicaciones envían los datos a una aplicación de servidor que utiliza protocolos estándar HTTP o TCP/IP a través de la red inalámbrica e Internet o la intranet de la empresa. Puede utilizar las API XML para generar y analizar los datos con formato XML para enviar y recibir a través de la red inalámbrica. Sin embargo, las aplicaciones de cliente y servidor deben leer y escribir los datos adecuadamente y reconocer una transmisión satisfactoria.

BlackBerry Java Application se puede conectar a una aplicación basada en el ordenador para enviar los datos a través de una conexión USB mediante las API de BlackBerry Desktop Synchronization y BlackBerry Desktop Manager. En este caso, debe crear una aplicación para Windows® que pueda leer los datos del cliente a través de una tarea complementaria para BlackBerry Desktop Manager. El usuario del dispositivo BlackBerry debe ejecutar manualmente la sincronización ejecutando el complemento BlackBerry Desktop Manager, que notifica a la aplicación acerca del dispositivo BlackBerry para enviar datos a la aplicación del ordenador. También puede escribir los datos en la aplicación del ordenador utilizando los protocolos USB originales.

Transporte inalámbrico de datos

6

Gateways inalámbricas

Las aplicaciones Java® para dispositivos BlackBerry® pueden utilizar protocolos de socket estándar HTTP, HTTPS y TCP para establecer conexiones a través de la red inalámbrica. Cuando una aplicación establece una conexión a través de la red inalámbrica, puede utilizar una de las dos gateways inalámbricas para delegar la conexión a Internet o a la intranet de la empresa. Puede diseñar la aplicación para que dependa de una gateway predeterminada que está disponible para el usuario del dispositivo BlackBerry, o puede personalizar el código para seleccionar una gateway preferida. Diseñe su aplicación para que seleccione explícitamente la gateway preferida para la conexión y utilice la predeterminada si el método preferido no está disponible. Esto podría reducir el número de los problemas de conexión de red al que se enfrentan sus clientes y permita a su aplicación utilizar un modelo de conectividad eficaz a través de todos los tipos de red y operadores inalámbricos.

Uso de BlackBerry Enterprise Server como gateway de red

Si utiliza BlackBerry® Enterprise Server como gateway de red, todo el tráfico entre la aplicación y BlackBerry Enterprise Server se cifra utilizando el cifrado AES o Triple DES. Puesto que BlackBerry Enterprise Server se encuentra tras el firewall de la empresa y ofrece un cifrado de datos inherente, las aplicaciones se pueden comunicar con servidores de la aplicación y con servidores Web que se ubican en la intranet de la empresa. El componente de BlackBerry® Mobile Data System de BlackBerry Enterprise Server incluye BlackBerry MDS Services, que ofrece un servidor proxy HTTP y TCP/IP proxy para permitir que BlackBerry® Java® Application lo utilice como gateway segura para administrar conexiones HTTP y TCP/IP a la intranet.

Si la aplicación se conecta a Internet, puede que sea capaz de utilizar BlackBerry Enterprise Server como gateway. Las solicitudes de red viajan tras el firewall de la empresa hacia BlackBerry Enterprise Server, que realiza la solicitud de red a Internet a través del firewall de la empresa. Los administradores pueden establecer una política de TI para asegurarse de que BlackBerry Enterprise Server es la gateway para todo el tráfico de red inalámbrica, incluido el tráfico destinado a Internet.

Si la aplicación se conecta a Internet, también puede utilizar o BlackBerry® Internet Service o la gateway de Internet del proveedor del servidor inalámbrico para administrar las conexiones.

Uso de la gateway a Internet del proveedor de servicios inalámbricos

Muchos proveedores de servicios inalámbricos facilitan una gateway a Internet que ofrece conectividad TCP/IP directa a Internet. Algunos proveedores de servicios inalámbricos también proporcionan una gateway WAP que permite que las conexiones HTTP se produzcan mediante el protocolo WAP. BlackBerry® Java® Application puede utilizar cualquiera de estas gateways para conectarse a Internet. Si la aplicación es para usuarios de dispositivos BlackBerry que están en una red inalámbrica específica, la utilización de la gateway a Internet del proveedor de servicios inalámbricos, a menudo, puede ofrecer buenos resultados. Si la aplicación es para usuarios de dispositivos BlackBerry en una variedad de redes inalámbricas, probar la aplicación en las distintas

gateways a Internet y lograr una experiencia constante y segura puede suponer un reto. Puede que le sea útil el uso de BlackBerry® Internet Service y de la gateway a Internet del proveedor de servicios inalámbricos como tipo de conexión predeterminada si BlackBerry Internet Service no está disponible.

Opciones de transporte de datos alternativas

Uso del correo electrónico para transportar datos

Puede utilizar las API de BlackBerry® para crear una aplicación BlackBerry® Java® Application que utilice el correo electrónico como mecanismo de transporte para enviar y recibir datos. El correo electrónico puede ser una manera efectiva de distribuir contenido de forma proactiva a los usuarios de dispositivos BlackBerry si los modelos de inserción tradicionales no están disponibles. BlackBerry Java Application puede utilizar las API de BlackBerry para enviar mensajes de correo electrónico y escuchar mensajes de correo electrónico entrantes. BlackBerry Java Application también puede obtener acceso a los detalles y encabezados de mensajes de correo electrónico que se almacenan de forma local en el dispositivo BlackBerry, así como registrar oyentes para los cambios en el estado de un mensaje de correo electrónico.

Uso de SMS para transportar datos

Puede utilizar las API de BlackBerry® para crear aplicaciones que utilicen SMS como mecanismo de transporte para enviar y recibir datos. Las API de BlackBerry permiten a una aplicación enviar mensajes SMS y escuchar mensajes SMS entrantes.

Uso de la mensajería PIN para transportar datos

La mensajería PIN utiliza el canal de datos en lugar del canal de voz y le permite dirigirse al dispositivo BlackBerry® de destino con su número PIN único. La mensajería PIN sólo se puede utilizar para enviar datos desde un dispositivo BlackBerry a otro. Las API de BlackBerry también pueden permitir a una aplicación enviar y recibir mensajes PIN de BlackBerry de forma programada. La mensajería PIN puede ser una manera efectiva de implementar las aplicaciones PIN dirigidas únicamente a usuarios de dispositivos BlackBerry.

Integración de aplicaciones BlackBerry

7

Adición de elementos de menú personalizados

BlackBerry® Java® Application puede agregar elementos de menú personalizados al menú en el dispositivo BlackBerry para el correo electrónico, el organizador y las aplicaciones de teléfono. Cuando un usuario de dispositivo BlackBerry selecciona un elemento de menú personalizado, BlackBerry Java Application se inicia con una referencia al objeto que selecciona el usuario del dispositivo BlackBerry. Por ejemplo, BlackBerry Java Application puede agregar un elemento de menú denominado Mostrar ubicación del remitente a la aplicación de correo electrónico. Si el usuario del dispositivo BlackBerry selecciona el elemento del menú, BlackBerry Java Application se inicia con una referencia al objeto de correo electrónico que está resaltado actualmente o que abre el usuario. BlackBerry Java Application utiliza la dirección de correo electrónico del remitente para determinar la ubicación del remitente recuperando la dirección de correo electrónico de la lista de contactos, o recuperando los datos de un servidor remoto y, a continuación, pasa a primer plano y muestra un mapa.

Integración con aplicaciones de BlackBerry Device Software

Una aplicación de dispositivo BlackBerry® puede invocar una aplicación de BlackBerry Device Software tal como el correo electrónico, el organizador, el teléfono, el explorador, y las aplicaciones de cámara para realizar una acción o mostrar información. Los siguientes ejemplos demuestran cómo una aplicación de dispositivo BlackBerry puede invocar una aplicación de BlackBerry Device Software :

- La aplicación del dispositivo BlackBerry invoca el calendario para mostrar una fecha o una entrada de calendario específicas.
- La aplicación del dispositivo BlackBerry invoca la agenda para mostrar un contacto específico.
- La aplicación del dispositivo BlackBerry invoca el explorador para abrir una dirección Web específica.
- La aplicación del dispositivo BlackBerry invoca el teléfono para llamar a un número específico.

Acceso a los datos del correo electrónico y del organizador

BlackBerry® Java® Application puede utilizar las API de BlackBerry para obtener acceso a los detalles de los mensajes de correo electrónico, contactos, eventos del calendario, tareas y registros de teléfono que almacena el dispositivo BlackBerry. BlackBerry Java Application puede leer la información, actualizarla y crear entradas nuevas.

Uso de BlackBerry Messenger con BlackBerry Application

Puede integrar BlackBerry® Java® Application con la aplicación BlackBerry® Messenger. Esto podría ser útil si desea crear una aplicación de juegos basada en turnos para el dispositivo BlackBerry.

Para crear una BlackBerry Java Application integrada con la aplicación BlackBerry Messenger, puede utilizar las clases en el paquete `net.rim.blackberry.api.blackberrymessenger`. Para obtener más información acerca del uso de la clase `BlackBerryMessenger`, consulte la Referencia API de BlackBerry.

Uso de oyentes para responder a los cambios de la aplicación

BlackBerry® Java® Application puede registrar oyentes en los almacenes de datos del correo electrónico y del organizador en la aplicación de teléfono. Los oyentes permiten a BlackBerry Java Application emprender una acción inmediata cuando el usuario del dispositivo BlackBerry lleve a cabo un evento local. Puede utilizar los oyentes de datos del correo electrónico y del organizador para notificar a BlackBerry Java Application cuando lleguen entradas nuevas o cuando el usuario del dispositivo BlackBerry realice cambios como adiciones, eliminaciones o actualizaciones en los datos existentes. Puede utilizar oyentes de teléfono para escuchar acciones de llamadas telefónicas, como la iniciación de llamadas nuevas o la finalización de llamadas.

Consideraciones de seguridad

8

Cifrado de datos y BlackBerry Application

Cifrado de datos durante el transporte

Si utiliza BlackBerry® Enterprise Server como gateway de red para la aplicación, BlackBerry Enterprise Server cifra los datos mediante el cifrado AES o TripleDES en todos los puntos de la conexión entre el dispositivo BlackBerry y BlackBerry Enterprise Server tras el firewall de la empresa. Si necesita que los datos se cifren incluso más entre BlackBerry Enterprise Server y el servidor de destino, puede utilizar el protocolo HTTPS y el cifrado SSL/TLS.

Si la aplicación utiliza BlackBerry® Internet Service o la gateway a Internet del proveedor de servicios inalámbricos, el tráfico de datos no se cifra. Si los usuarios de dispositivos BlackBerry lo prefieren, puede utilizar HTTPS para cifrar los datos o puede utilizar las API de Java® para el cifrado y aplicar así su propia clave simétrica o la criptografía de clave pública.

Cifrado de datos en el dispositivo BlackBerry

Los administradores pueden establecer una política de TI para asegurarse de que todos los datos del usuario del dispositivo BlackBerry® almacenados en las aplicaciones del dispositivo BlackBerry se cifran de forma local en la memoria flash. Puede crear una aplicación BlackBerry Java® Application que utilice API para registrar los datos para que el servicio de cifrado cifre los datos con la misma clave de la seguridad antes de almacenarlos en la memoria flash.

Acceso a la memoria

BlackBerry® Java® Development Environment se ha diseñado para evitar que las aplicaciones provoquen problemas de forma accidental o malintencionada en otras aplicaciones o en el dispositivo BlackBerry. Las aplicaciones BlackBerry pueden escribir sólo en la memoria del dispositivo BlackBerry que utiliza BlackBerry® Java® Virtual Machine; no pueden obtener acceso a la memoria virtual ni al almacenamiento constante de otras aplicaciones (a menos que se les otorgue acceso específicamente para hacerlo). BlackBerry® Java Application sólo puede obtener acceso al almacenamiento constante o a los datos del usuario, o comunicarse con otras aplicaciones, a través de API específicas de BlackBerry. Research In Motion debe firmar digitalmente una aplicación BlackBerry Java Application que utilice estas API de BlackBerry, para proporcionar una pista de auditoría de las aplicaciones que utilizan API importantes.

Autenticación

Autenticación de dispositivos BlackBerry y política de TI

Los usuarios de dispositivos BlackBerry® pueden establecer una contraseña para sus dispositivos BlackBerry. Si la contraseña del dispositivo está activa, los usuarios del dispositivo BlackBerry deben ofrecer la contraseña para obtener acceso a los datos y las aplicaciones. El uso de las contraseñas del dispositivo constituye un buen paso para limitar el acceso a BlackBerry Java® Application en el dispositivo BlackBerry.

Los administradores pueden utilizar las políticas de TI ofrecidas en BlackBerry® Enterprise Server para asegurarse de que los dispositivos BlackBerry de la empresa están protegidos con contraseña. Los administradores también pueden utilizar las políticas de TI para bloquear un dispositivo BlackBerry, cambiar la contraseña o eliminar todos los datos.

Autenticación de aplicaciones

Para las aplicaciones en las cuales las características de seguridad son críticas, puede que desee ofrecer una pantalla de inicio de sesión que solicite al usuario del dispositivo BlackBerry® iniciar sesión en la aplicación del dispositivo antes de utilizarlo. Las clases de la interfaz de usuario proporcionan campos de contraseña sencillos que ocultan la entrada de texto con caracteres de asterisco. Las pantallas de inicio de sesión pueden repercutir de forma negativa en la experiencia de usuario del dispositivo BlackBerry y si éste establece una contraseña para proteger el dispositivo BlackBerry, puede que la aplicación no requiera una pantalla de inicio de sesión.

Autenticación del servidor

Si la aplicación se conecta a una aplicación en un servidor o a Internet o una intranet, puede que desee incluir características adicionales de autenticación cuando los usuarios de dispositivos BlackBerry® inicien sesión en el servidor. La mayoría de las aplicaciones que requieren la autenticación del usuario confían en la autenticación básica HTTP, que utiliza una combinación sencilla de nombre de usuario y contraseña. Puede utilizar la autenticación básica HTTP agregando los encabezados HTTP correctos al abrir la conexión HTTP. También puede utilizar una autenticación más avanzada utilizando certificados; sin embargo, la mayoría de las aplicaciones no la requieren.

API controladas y firma de códigos

Research In Motion efectúa el seguimiento de uso de las API con distinciones por razones de seguridad y de control de la exportación. En la referencia API de BlackBerry®, RIM identifica una clase o método controlado con un icono de bloqueo o una nota firmada. Para utilizar las clases o métodos controlados en sus aplicaciones y antes de poder instalar los archivos .cod de la aplicación en el dispositivo BlackBerry, debe firmar su aplicación con una clave, o firma, de RIM. Otra funcionalidad, como la capacidad de ejecutar cuando la aplicación se inicia, puede requerir que firme sus aplicaciones.

Aunque el proceso de registro de RIM cubre el uso de la mayoría de las API controladas, algunas clases de criptografía relacionadas con la criptografía de claves públicas y privadas contienen tecnología de Certicom™. Para utilizar estas clases, debe registrarse y obtener una licencia de Certicom directamente. El proceso de registro de RIM no incluye el uso de las clases de Certicom.

Para probar y depurar el código antes de recibir las firmas de código, puede utilizar BlackBerry® Smartphone Simulator. Debe firmar la aplicación antes de instalarla en dispositivos BlackBerry. No tiene que enviar su código real a RIM. Puede utilizar BlackBerry® Signature Tool para enviar un hash SHA-1 de su archivo de código para que el sistema de autoridad de firmas pueda generar la firma necesaria.

Para obtener más información acerca de cómo registrar y obtener firmas de código, consulte *BlackBerry Signing Authority Tool versión 1.0 - Guía del administrador basada en contraseñas*. Para obtener más información acerca de cómo registrar y utilizar clases, visite www.blackberry.com/developers/index.shtml.

API de BlackBerry con acceso controlado

Puede ejecutar aplicaciones que utilicen una API controlada en BlackBerry® Smartphone Simulator sin firmas de código; sin embargo, debe obtener las firmas de código de Research In Motion antes de instalar estas aplicaciones en los dispositivos BlackBerry.

Puede utilizar las siguientes categorías de API controlada de RIM:

- API de tiempo de ejecución
- API de BlackBerry® Application
- API de cifrado de BlackBerry

Si utiliza cualquiera de los siguientes paquetes de API de BlackBerry, su aplicación necesitará firmas de código antes de poder instalarlos en un dispositivo BlackBerry:

- `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`

Para obtener más información sobre las API controladas de RIM, consulte la referencia API de BlackBerry.

Compatibilidad con la política de TI

Puede utilizar las API en el paquete `net.rim.device.api.itpolicy` para acceder a la información de políticas de TI en el dispositivo BlackBerry® y cambiar el comportamiento o la funcionalidad de una aplicación de dispositivo BlackBerry. Cada elemento de política de TI está formado por un nombre, una descripción y un valor. El valor puede ser uno de estos tres valores: `string`, `integer` o `Boolean`.

La configuración de la política de TI del dispositivo BlackBerry se sincroniza automáticamente y se actualiza a través de la red inalámbrica con la versión 3.6 o posterior de BlackBerry® Device Software. En las versiones anteriores de BlackBerry Device Software, la configuración de la política de TI se actualiza cuando el usuario del dispositivo BlackBerry sincroniza el dispositivo BlackBerry con el ordenador.

Para obtener más información acerca de las políticas de control de TI, consulte la *Guía de referencia de políticas de BlackBerry Enterprise Server*.

Control de la aplicación

Las reglas de la política de control de aplicaciones de BlackBerry® Enterprise Server se han diseñado para permitir o evitar la instalación de aplicaciones específicas en el dispositivo BlackBerry y para limitar los permisos de las aplicaciones en el dispositivo BlackBerry. Por ejemplo, los administradores pueden utilizar la política de control de aplicaciones para asegurarse de que una aplicación de juegos en el dispositivo BlackBerry no pueda obtener acceso a la aplicación de teléfono.

El administrador puede aplicar políticas de control de aplicaciones sólo cuando el dispositivo BlackBerry está asociado a BlackBerry Enterprise Server. Si el administrador o un usuario de dispositivo BlackBerry deniega el acceso de la aplicación a una de las áreas protegidas, la aplicación muestra una excepción `ControlledAccessException`.

Cifrado de archivos en tarjetas microSD

Cifrado de datos en una tarjeta multimedia microSD

Si BlackBerry® Java® Application obtiene acceso a un archivo en la tarjeta de memoria microSD, tiene lugar el descifrado del archivo y el archivo se traslada a la memoria principal para que lo lea una aplicación. Para que BlackBerry Java Application obtenga acceso a un archivo protegido con contraseña, el dispositivo BlackBerry no debe bloquearse. Los archivos cifrados tienen una extensión `.rem` y no se pueden descifrar en plataformas que no sean de BlackBerry.

Si el NVRAM se elimina y la tarjeta multimedia microSD se bloquea con una clave de dispositivo BlackBerry, los datos de la tarjeta multimedia microSD ya no están accesibles. Para eliminar los datos que no sean accesibles, inicie el dispositivo BlackBerry y elimine todos los archivos multimedia cifrados.

El dispositivo BlackBerry utiliza una clave maestra almacenada en la tarjeta multimedia microSD para cifrar los archivos multimedia del dispositivo BlackBerry. La clave maestra evita que el dispositivo BlackBerry tenga que descifrar o volver a cifrar todos los archivos multimedia cuando desactive el cifrado o cambie la contraseña.

Uso de la tarjeta multimedia microSD con más de un dispositivo BlackBerry

Si el usuario del dispositivo BlackBerry® mueve la tarjeta multimedia microSD a un dispositivo BlackBerry que no utiliza una contraseña de dispositivo o que utiliza una contraseña que no descifra correctamente la clave principal de la tarjeta multimedia microSD, el dispositivo BlackBerry solicita al usuario del dispositivo BlackBerry que escriba la contraseña de la tarjeta multimedia microSD. Si el dispositivo BlackBerry tiene una contraseña, el usuario del dispositivo BlackBerry puede utilizar el aviso para cambiar la contraseña de la tarjeta multimedia microSD a una contraseña del dispositivo BlackBerry.

Políticas de TI y tarjetas multimedia microSD

Puede aplicar la política de TI Cifrar los datos escritos en la tarjeta multimedia microSD a cualquier archivo nuevo o modificado que almacene en la tarjeta multimedia microSD. Sólo se cifrarán los archivos que almacene en la tarjeta multimedia microSD después de que un administrador establezca la política de TI. Se cifrará todo el contenido excepto los archivos multimedia.

Probar BlackBerry Java Application

9

Cifrado de una aplicación BlackBerry Java Application

El compilador para BlackBerry® Java® Development Environment y BlackBerry® JDE Plug-in para Eclipse® se configura para reducir el tamaño de la aplicación. El compilador produce un archivo .cod que proporciona servicios de cifrado similares a aquellos que ofrecen los paquetes de cifrado en un esfuerzo por reducir el tamaño de los archivos .cod. Por ejemplo, BlackBerry Java Development Environment elimina la siguiente información de un archivo .cod:

- toda la información de depuración
- los nombres de variables locales
- los números de la línea de origen
- el método privado y los nombres de miembros

Normalmente no es necesario que proporcione cifrado para sus aplicaciones además del cifrado ya existente, de forma predeterminada, lo proporciona BlackBerry JDE. De hecho, Research In Motion no realiza ningún cifrado adicional en sus propios productos.

BlackBerry JDE no incluye compatibilidad con el cifrado a través de herramientas de terceros. Debe incluir un procedimiento de línea de comandos para cifrar los archivos .cod en los dispositivos BlackBerry.

Verificación previa de BlackBerry Java Application

Al verificar previamente las clases, reducirá la cantidad de procesamiento que debe realizar el dispositivo BlackBerry® cuando instala una aplicación. Para verificar parcialmente sus clases antes de instalar su aplicación en un dispositivo BlackBerry, puede utilizar la herramienta de verificación previa, disponible en BlackBerry® Java® Development Environment. Puede utilizar BlackBerry® Smartphone Simulator para verificar previamente los archivos .cod.

Para obtener más información acerca de cómo utilizar BlackBerry Smartphone Simulator, consulte *Manual del usuario de BlackBerry Device Simulator*. Si desea obtener más información acerca del uso de la herramienta de verificación previa, consulte *BlackBerry Java Development Environment Development Guide (Guía de desarrollo de BlackBerry Java Development Environment)*.

Prueba de aplicaciones en BlackBerry Smartphone Simulator

Tras desarrollar y compilar su aplicación, puede probarla en el dispositivo BlackBerry®. El primer paso más común consiste en configurar BlackBerry® Java® Development Environment para que utilice BlackBerry® Smartphone Simulator. BlackBerry Smartphone Simulator ejecuta el mismo código Java que los dispositivos BlackBerry, de modo que BlackBerry Smartphone

Simulator proporciona un entorno adecuado para probar cómo las aplicaciones funcionarán en un dispositivo BlackBerry. BlackBerry JDE incluye las versiones actuales de BlackBerry Smartphone Simulator. Para descargar las versiones adicionales de BlackBerry Smartphone Simulator, visite www.blackberry.com/developers/index.shtml.

Prueba de aplicaciones en un dispositivo BlackBerry

Tras probar su aplicación en BlackBerry® Smartphone Simulator, puede instalar la aplicación en un dispositivo BlackBerry. Si su aplicación utiliza API firmadas, puede que necesite claves de firma de código. Tras instalar la aplicación en el dispositivo BlackBerry, puede abrirla y probar su funcionalidad y rendimiento.

Con el objeto de depurar, puede integrar su dispositivo en BlackBerry® Integrated Development Environment y utilizar la herramienta de depuración para avanzar a través del código de la aplicación. BlackBerry IDE puede ser útil si intenta identificar una red o problema de Bluetooth®, así como otros problemas que son difíciles de simular.

Simuladores de BlackBerry

BlackBerry Smartphone Simulator

BlackBerry® Smartphone Simulator se ha diseñado para emular una experiencia BlackBerry sin utilizar un dispositivo BlackBerry real. BlackBerry Smartphone Simulator es una aplicación que instala en el ordenador y muestra una imagen del modelo del dispositivo BlackBerry de su elección. Esta imagen tiene el aspecto de un dispositivo BlackBerry real. La funcionalidad incluye la misma interacción del usuario de un dispositivo BlackBerry real (incluida la rueda o bola de desplazamiento y el teclado), las mismas aplicaciones y las mismas características, como los mensajes de correo electrónico, el teléfono y la exploración de Internet.

BlackBerry Smartphone Simulator también sirve como plataforma en la que se pueden ejecutar aplicaciones. Esto incluye la capacidad de las aplicaciones para establecer conexiones de red, almacenar datos y gestionar mensajes de correo electrónico. BlackBerry Smartphone Simulator incluye aplicaciones del dispositivo BlackBerry que normalmente están disponibles en los dispositivos BlackBerry por lo que puede instalar y probar sus propias aplicaciones. Puede simular y probar varios cambios de conectividad y estado mediante BlackBerry Smartphone Simulator.

Al utilizar BlackBerry Smartphone Simulator para realizar pruebas, puede que necesite simular servicios adicionales de BlackBerry. BlackBerry® MDS Simulator y BlackBerry® Email Simulator están disponibles para este fin.

Para obtener BlackBerry Smartphone Simulator, visite www.blackberry.com/developers y descargue BlackBerry® Java® Development Environment o el paquete de componentes de BlackBerry Java Development Environment.

BlackBerry MDS Simulator

BlackBerry® MDS Simulator está diseñado para simular el componente BlackBerry MDS Connection Service de BlackBerry® Enterprise Server. Al utilizar BlackBerry Smartphone Simulator con BlackBerry MDS Simulator puede probar redes, insertar HTTP y aplicaciones del explorador diseñadas para su uso con BlackBerry Enterprise Server. Para obtener BlackBerry MDS Simulator, visite www.blackberry.com/developers y descargue BlackBerry® Email and MDS Services Simulator Package.

BlackBerry Email Simulator

BlackBerry® Email Simulator se ha diseñado para enviar y recibir mensajes entre BlackBerry® Smartphone Simulator y cualquier otra aplicación de correo electrónico, como Microsoft® Outlook®, o servidores POP3 y SMTP. No es necesario BlackBerry® Enterprise Server. Para obtener BlackBerry Email Simulator, visite www.blackberry.com/developers y descargue BlackBerry® Email and MDS Services Simulator Package.

Disponibilidad de las aplicaciones

10

Distribución de aplicaciones a través de una conexión de ordenador

Distribuir una aplicación desde un ordenador

Puede utilizar la herramienta Cargador de aplicaciones en BlackBerry® Desktop Manager para instalar aplicaciones en el dispositivo BlackBerry. La herramienta Cargador de aplicaciones ofrece a los usuarios un modo sencillo de descargar las aplicaciones desde sus ordenadores a sus dispositivos BlackBerry.

Distribuir una aplicación desde una página Web

Puede utilizar BlackBerry® Application Web Loader para registrar la aplicación compilada en un sitio Web. Los usuarios pueden utilizar Windows® Internet Explorer® en sus ordenadores para visitar la página Web e instalar la aplicación en sus dispositivos BlackBerry. Cuando los usuarios de dispositivos BlackBerry visitan la página Web, BlackBerry Application Web Loader les solicita que conecten sus dispositivos al puerto USB. A continuación, pueden instalar la aplicación mediante un control ActiveX®. BlackBerry Application Web Loader ofrece a los usuarios de dispositivos BlackBerry una manera sencilla de instalar las aplicaciones desde sus ordenadores sin ejecutar BlackBerry® Desktop Manager.

Distribuir una aplicación para la prueba

BlackBerry® Java® Development Environment incluye una herramienta de línea de comandos denominada JavaLoader que se ubica en la carpeta BIN dentro de la carpeta BlackBerry JDE. Puede utilizar la herramienta JavaLoader para instalar y eliminar rápidamente los archivos de la aplicación compilados en el dispositivo BlackBerry directamente a través del puerto USB. No necesita ningún archivo de descriptor o páginas Web. La herramienta JavaLoader puede ser útil si instala y elimina la aplicación con frecuencia durante la prueba y desarrollo; sin embargo, la herramienta JavaLoader no se ha diseñado para su uso por parte de usuarios de dispositivos BlackBerry.

Distribución de aplicaciones a través de la red inalámbrica

Puede distribuir sus aplicaciones a través de la red inalámbrica para ayudar a proporcionar una mejor experiencia a los usuarios de dispositivos BlackBerry® y para simplificar la distribución de aplicaciones a un grupo grande de personas, puesto que no requiere una aplicación informática. Un usuario del dispositivo BlackBerry puede instalar sus aplicaciones a través de la red inalámbrica.

Extracción inalámbrica (iniciada por el usuario)

Puede registrar aplicaciones compiladas en un sitio Web público o privado. Los usuarios de dispositivos BlackBerry® pueden visitar el sitio Web para descargar aplicaciones a través de la red inalámbrica mediante el explorador de sus dispositivos BlackBerry. El explorador solicita al usuario que instale la aplicación y, a continuación, la aplicación se descarga a través de la red inalámbrica y se instala en el dispositivo BlackBerry.

Inserción inalámbrica (iniciada por el servidor)

En el entorno BlackBerry® Enterprise Server, el administrador puede insertar las aplicaciones para los usuarios de dispositivos BlackBerry a través de la red inalámbrica para la instalación obligatoria. El administrador crea una política nueva y específica que el dispositivo BlackBerry requiere la aplicación. La aplicación se inserta para los usuarios sin que sea necesaria ninguna interacción por parte del usuario. Las empresas pueden considerar práctico este enfoque a la hora de enviar aplicaciones nuevas a un gran número de usuarios de dispositivos BlackBerry.

Glosario

11

AES

Advanced Encryption Standard (Estándar de cifrado avanzado)

API

Application Programming Interface (Interfaz de programación de aplicaciones)

CLDC

Connected Limited Device Configuration (Configuración de dispositivos con conexión limitada)

HTTP

Hypertext Transfer Protocol (Protocolo de transferencia de hipertexto)

HTTPS

Hypertext Transfer Protocol over Secure Sockets Layer (Protocolo de transferencia de hipertexto a través de un nivel de socket seguro)

I/O

entrada/salida

IMAP

Internet Message Access Protocol (Protocolo de acceso a mensajes de Internet)

IPC

Comunicación entre procesos

JDWP

Java® Debug Wire Protocol

JSR

Java® Specification Request

JTWI

Java® Technology for the Wireless Industry (Tecnología Java para la industria inalámbrica)

JVM

Java® Virtual Machine

MIDP

Mobile Information Device Profile (Perfil de dispositivo móvil de información)

NVRAM

Memoria de acceso aleatorio no volátil

PDAP

Paquetes opcionales de PDA para la plataforma J2ME

PIM

personal information management (administración de información personal)

PIN

número de identificación personal

POP

Protocolo de oficina de correos

RAPC

Compilador de programas de aplicaciones de RIM

RMS

Sistema de administración de registros (Record Management System)

SHA

Secure Hash Algorithm (Algoritmo de hash seguro)

SMS

Short Message Service (Servicio de mensajes cortos)

SMTP

Protocolo de transferencia simple de correo (Simple Mail Transfer Protocol)

SRAM

Memoria de acceso aleatorio estático

SSL

Secure Sockets Layer (Nivel de socket seguro)

TCP

Transmission Control Protocol (Protocolo de control de transmisión)

TLS

Transport Layer Security (Seguridad de capa de transporte)

Triple DES

Triple Data Encryption Standard (Estándar de cifrado triple de datos)

VPN

virtual private network (red privada virtual)

WAP

Wireless Application Protocol (Protocolo de aplicaciones inalámbricas)

XML

Extensible Markup Language (Lenguaje de marcado extensible)

Aviso legal

12

©2009 Research In Motion Limited. Todos los derechos reservados. BlackBerry®, RIM®, Research In Motion®, SureType®, SurePress™ y las marcas comerciales, nombres y logotipos relacionados son propiedad de Research In Motion Limited y están registrados y/o se utilizan en EE.UU. y en diferentes países del mundo.

Bluetooth es una marca comercial de Bluetooth SIG. Java y Javadoc son marcas comerciales de Sun Microsystems, Inc. Microsoft, Windows Internet Explorer, ActiveX y Windows son marcas comerciales de Microsoft Corporation. HotSync es una marca comercial de PalmSource, Inc. Todas las demás marcas comerciales son propiedad de sus respectivos usuarios.

El smartphone BlackBerry y otros dispositivos y/o el software asociado están protegidos por copyright, tratados internacionales y varias patentes, incluida una o varias de las siguientes de EE.UU. 6.278.442, 6.271.605, 6.219.694, 6.075.470, 6.073.318, D445.428, D433.460, D416.256. Otras patentes están registradas o pendientes de registro en los EE.UU. y en diferentes países del mundo. Visite www.rim.com/patents para obtener una lista de las patentes de RIM (como se las denominará de ahora en adelante).

Esta documentación, incluida cualquier documentación que se incorpore mediante referencia como documento proporcionado o disponible en www.blackberry.com/go/docs, se proporciona o se pone a disposición "TAL CUAL" y "SEGÚN SU DISPONIBILIDAD" sin ninguna condición, responsabilidad o garantía de ningún tipo por Research In Motion Limited y sus empresas afiliadas ("RIM") y RIM no asume ninguna responsabilidad por los errores tipográficos, técnicos o cualquier otra imprecisión, error u omisión contenidos en esta documentación. Con el fin de proteger la información confidencial y propia de RIM, así como los secretos comerciales, la presente documentación describe algunos aspectos de la tecnología de RIM en líneas generales. RIM se reserva el derecho a modificar periódicamente la información que contiene esta documentación, si bien tampoco se compromete en modo alguno a proporcionar cambios, actualizaciones, ampliaciones o cualquier otro tipo de información que se pueda agregar a esta documentación.

Esta documentación puede contener referencias a fuentes de información, hardware o software, productos o servicios, incluidos componentes y contenido como, por ejemplo, el contenido protegido por copyright y/o sitios Web de terceros (conjuntamente, los "Productos y servicios de terceros"). RIM no controla ni es responsable de ningún tipo de Productos y servicios de terceros, incluido, sin restricciones, el contenido, la exactitud, el cumplimiento de copyright, la compatibilidad, el rendimiento, la honradez, la legalidad, la decencia, los vínculos o cualquier otro aspecto de los Productos y servicios de terceros. La inclusión de una referencia a los Productos y servicios de terceros en esta documentación no implica que RIM se haga responsable de dichos Productos y servicios de terceros o de dichos terceros en modo alguno.

EXCEPTO EN LA MEDIDA EN QUE LO PROHÍBA ESPECÍFICAMENTE LA LEY DE SU JURISDICCIÓN, QUEDAN EXCLUIDAS POR LA PRESENTE TODAS LAS CONDICIONES, APROBACIONES O GARANTÍAS DE CUALQUIER TIPO, EXPLÍCITAS O IMPLÍCITAS, INCLUIDA, SIN NINGÚN TIPO DE LIMITACIÓN, CUALQUIER CONDICIÓN, APROBACIÓN, GARANTÍA, DECLARACIÓN DE GARANTÍA DE DURABILIDAD, IDONEIDAD PARA UN FIN O USO DETERMINADO, COMERCIALIZACIÓN, CALIDAD COMERCIAL, ESTADO DE NO INFRACCIÓN, CALIDAD SATISFACTORIA O TITULARIDAD, O QUE SE DERIVE DE UNA LEY O COSTUMBRE O UN CURSO DE LAS NEGOCIACIONES O USO DEL COMERCIO, O RELACIONADO CON LA DOCUMENTACIÓN O SU USO O RENDIMIENTO O NO RENDIMIENTO DE CUALQUIER SOFTWARE, HARDWARE, SERVICIO O CUALQUIER PRODUCTO O SERVICIO DE TERCEROS MENCIONADOS AQUÍ. TAMBIÉN PODRÍA TENER OTROS DERECHOS QUE VARIAN SEGÚN EL ESTADO O PROVINCIA. ES POSIBLE QUE ALGUNAS JURISDICCIONES NO PERMITAN LA EXCLUSIÓN O LA LIMITACIÓN DE GARANTÍAS IMPLÍCITAS Y CONDICIONES. EN LA MEDIDA EN QUE LO PERMITA LA LEY, CUALQUIER GARANTÍA IMPLÍCITA

O CONDICIONES EN RELACIÓN CON LA DOCUMENTACIÓN NO SE PUEDEN EXCLUIR TAL Y COMO SE HA EXPUESTO ANTERIORMENTE, PERO PUEDEN SER LIMITADAS, Y POR LA PRESENTE ESTÁN LIMITADAS A NOVENTA (90) DÍAS DESDE DE LA FECHA QUE ADQUIRIÓ LA DOCUMENTACIÓN O EL ELEMENTO QUE ES SUJETO DE LA RECLAMACIÓN.

EN LA MEDIDA MÁXIMA EN QUE LO PERMITA LA LEY DE SU JURISDICCIÓN, EN NINGÚN CASO RIM ASUMIRÁ RESPONSABILIDAD ALGUNA POR CUALQUIER TIPO DE DAÑOS RELACIONADOS CON ESTA DOCUMENTACIÓN O SU USO, O RENDIMIENTO O NO RENDIMIENTO DE CUALQUIER SOFTWARE, HARDWARE, SERVICIO O PRODUCTOS Y SERVICIOS DE TERCEROS AQUÍ MENCIONADOS INCLUIDOS SIN NINGÚN TIPO DE LIMITACIÓN CUALQUIERA DE LOS SIGUIENTES DAÑOS: DIRECTOS, RESULTANTES, EJEMPLARES, INCIDENTALES, INDIRECTOS, ESPECIALES, PUNITIVOS O AGRAVADOS, DAÑOS POR PÉRDIDA DE BENEFICIOS O INGRESOS, IMPOSIBILIDAD DE CONSEGUIR LOS AHORROS ESPERADOS, INTERRUPCIÓN DE LA ACTIVIDAD COMERCIAL, PÉRDIDA DE INFORMACIÓN COMERCIAL, PÉRDIDA DE LA OPORTUNIDAD DE NEGOCIO O CORRUPCIÓN O PÉRDIDA DE DATOS, IMPOSIBILIDAD DE TRANSMITIR O RECIBIR CUALQUIER DATO, PROBLEMAS ASOCIADOS CON CUALQUIER APLICACIÓN QUE SE UTILICE JUNTO CON PRODUCTOS Y SERVICIOS DE RIM, COSTES DEBIDOS AL TIEMPO DE INACTIVIDAD, PÉRDIDA DE USO DE LOS PRODUCTOS Y SERVICIOS DE RIM O PARTE DE ÉL O DE CUALQUIER SERVICIO DE USO, COSTE DE SERVICIOS SUSTITUTIVOS, COSTES DE COBERTURA, INSTALACIONES O SERVICIOS, COSTE DEL CAPITAL O CUALQUIER OTRA PÉRDIDA MONETARIA SIMILAR, TANTO SI DICHOS DAÑOS SE HAN PREVISTO O NO, Y AUNQUE SE HAYA AVISADO A RIM DE LA POSIBILIDAD DE DICHOS DAÑOS.

EN LA MEDIDA MÁXIMA EN QUE LO PERMITA LA LEY DE SU JURISDICCIÓN, RIM NO TENDRÁ NINGÚN OTRO TIPO DE OBLIGACIÓN O RESPONSABILIDAD CONTRACTUAL, EXTRA CONTRACTUAL O CUALQUIER OTRA, INCLUIDA CUALQUIER RESPONSABILIDAD POR NEGLIGENCIA O RESPONSABILIDAD ESTRICTA.

LAS LIMITACIONES, EXCLUSIONES Y DESCARGOS DE RESPONSABILIDAD SE APLICARÁN: (A) INDEPENDIEMENTE DE LA NATURALEZA DE LA CAUSA DE LA ACCIÓN, DEMANDA O ACCIÓN SUYA, INCLUIDA PERO NO LIMITADA AL INCUMPLIMIENTO DEL CONTRATO, NEGLIGENCIA, AGRAVIO, EXTRA CONTRACTUAL, RESPONSABILIDAD ESTRICTA O CUALQUIER OTRA TEORÍA DEL DERECHO Y DEBERÁN SOBREVIVIR A UNO O MÁS INCUMPLIMIENTOS ESENCIALES O AL INCUMPLIMIENTO DEL PROPÓSITO ESENCIAL DE ESTE CONTRATO O CUALQUIER SOLUCIÓN CONTENIDA AQUÍ; Y (B) A RIM Y A SUS EMPRESAS AFILIADAS, SUS SUCESORES, CESIONARIOS, AGENTES, PROVEEDORES (INCLUIDOS LOS PROVEEDORES DE SERVICIOS DE USO), DISTRIBUIDORES AUTORIZADOS POR RIM (INCLUIDOS TAMBIÉN LOS PROVEEDORES DE SERVICIOS DE USO) Y SUS RESPECTIVOS DIRECTORES, EMPLEADOS Y CONTRATISTAS INDEPENDIENTES.

ADEMÁS DE LAS LIMITACIONES Y EXCLUSIONES MENCIONADAS ANTERIORMENTE, EN NINGÚN CASO NINGÚN DIRECTOR, EMPLEADO, AGENTE, DISTRIBUIDOR, PROVEEDOR, CONTRATISTA INDEPENDIENTE DE RIM O CUALQUIER AFILIADO DE RIM ASUMIRÁ NINGUNA RESPONSABILIDAD DERIVADA DE O RELACIONADA CON LA DOCUMENTACIÓN.

Antes de instalar, usar o suscribirse a cualquiera de los Productos y servicios de terceros, es su responsabilidad asegurarse de que su proveedor de servicios de uso ofrezca compatibilidad con todas sus funciones. Puede que algunos proveedores de servicios de uso no ofrezcan las funciones de exploración de Internet con una suscripción al servicio BlackBerry® Internet Service. Consulte con su proveedor de servicios acerca de la disponibilidad, arreglos de itinerancia, planes de servicio y funciones. La instalación o el uso de los Productos y servicios de terceros con productos y servicios de RIM puede precisar la obtención de una o más patentes, marcas comerciales, derechos de autor u otras licencias para evitar que se vulneren o violen derechos de terceros. Usted es el único responsable de determinar si desea utilizar Productos y servicios de terceros y si se necesita para ello cualquier otra licencia de terceros. En caso de necesitarlas, usted es el único responsable de su adquisición. No instale o utilice Productos y servicios de terceros hasta que se hayan adquirido todas las licencias necesarias. Cualquier tipo de Productos y servicios de terceros que se proporcione con los productos y servicios de RIM se le facilita para su comodidad "TAL CUAL" sin ninguna condición expresa e implícita, aprobación, garantía de cualquier tipo por RIM y RIM no suma ninguna responsabilidad en relación

con ello. El uso de los Productos y servicios de terceros se registrará y estará sujeto a la aceptación de los términos de licencias independientes aplicables en este caso con terceros, excepto en los casos cubiertos expresamente por una licencia u otro acuerdo con RIM.

Algunas funciones mencionadas en esta documentación requieren una versión mínima del software de BlackBerry® Enterprise Server, BlackBerry® Desktop Software y/o BlackBerry® Device Software.

Los términos de uso de cualquier producto o servicio de RIM se presentan en una licencia independiente o en otro acuerdo con RIM que se aplica en este caso. NINGUNA PARTE DE LA PRESENTE DOCUMENTACIÓN ESTÁ PENSADA PARA PREVALECER SOBRE CUALQUIER ACUERDO EXPRESO POR ESCRITO O GARANTÍA PROPORCIONADA POR RIM PARA PARTES DE CUALQUIER PRODUCTO O SERVICIO DE RIM QUE NO SEA ESTA DOCUMENTACIÓN.

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

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

Publicado en Canadá