BlackBerry Dynamics SDK for Cordova

Development Guide

11.0

Contents

What is the BlackBerry Dynamics SDK?

The BlackBerry Dynamics SDK provides a powerful set of tools that you can use to create secure productivity apps for a BlackBerry UEMdomain. The SDK leverages the full capabilities of the secure BlackBerry Dynamics platform, so you can focus on building your apps rather than learning how to secure, deploy, and manage those apps.

The BlackBerry Dynamics SDK is available for all major development platforms. It allows you to leverage many valuable services, including secure communication, securing data in file systems and databases, inter-app data exchange, presence, push, directory lookup, single sign-on authentication, identity and access management, and more.

This guide will provide:

- Information about supported features
- Development requirements and prerequisites
- Instructions for installing, configuring, and using the SDK
- Considerations for key platform features
- Information about the sample apps provided with the SDK
- Testing and troubleshooting guidance
- Guidance for deploying your app

This guide is intended for intermediate and experienced developers with an understanding of how to create apps for the intended platform. It is not a basic tutorial.

BlackBerry Dynamics API reference

The BlackBerry Dynamics SDK API reference describes the available interfaces, classes, methods, and much more. The API reference for each SDK platform is available at https://developers.blackberry.com/us/en/resources/api-reference.html.

Key features of the BlackBerry Dynamics SDK

This section provides more information about key features of the BlackBerry Dynamics SDK. It does not detail the complete feature set. For more information about the full list of supported features and APIs, see the BlackBerry Dynamics SDK API reference for your platform.

For more information about the requirements and prerequisites to support platform-specific features, see Requirements and support for platform-specific features.

The implementation of some of the features discussed in this section will depend on how the UEM administrator has configured your organization's servers, network, and other infrastructure components. Contact the administrator to clarify whether there are components of a feature that are configured or managed using the management server.

Activation

Infrastructure and enterprise activation

After a BlackBerry Dynamics app is installed on a user's device, the user must activate the app in order to use it. The activation process registers the app with the management server and gives the app access to the full capabilities of the BlackBerry Dynamics platform. The activation process ensures that all end users are fully authorized and permitted to use the app.

Users can activate a BlackBerry Dynamics app manually using an activation password, QR code, or access key provided by the administrator or obtained from UEM Self-Service, by using the UEM Client, through a third-party IDP, such as Active Directory or Okta, or by using the Easy Activation feature described below.

For more information about activating BlackBerry Dynamics apps, see the Activation section in the GDAndroid class reference or GDiOS class reference and Managing BlackBerry Dynamics apps in the UEM Administration content.

Easy Activation

Easy Activation simplifies the process of activating multiple BlackBerry Dynamics apps on a user's device. With Easy Activation, a device user only needs to activate the first BlackBerry Dynamics app on their device; when the user installs additional BlackBerry Dynamics apps, the user can choose to delegate the activation process to the previously activated app. Any BlackBerry Dynamics app can be an activation delegate, but priority is given to the app that is configured as the authentication delegate.

Easy Activation is automatically enabled for all BlackBerry Dynamics apps that are produced by BlackBerry. To enable Easy Activation for your custom BlackBerry Dynamics app, the UEM administrator must specify the app package ID (Android) or bundle ID (iOS) in the BlackBerry Dynamics app settings in the management console. Contact your organization's administrator to provide this information. For instructions for specifying the package ID or bundle ID for an app, see Manage settings for a BlackBerry Dynamics app in the UEM Administration content.

On the application side, Easy Activation is enabled by default by the BlackBerry Dynamics Runtime.

For more information, see the Easy Activation section in the BlackBerry Dynamics Security White Paper.

iOS user enrollment and DEP activation enhancements

The BlackBerry Dynamics SDK for iOS version 8.0 and later and UEM version 12.13 and later feature the following activation enhancements for iOS user enrollment and DEP:

- MDM enrollment and the activation of BlackBerry Dynamics apps doesn't require the UEM Client.
- After a new device is enrolled on UEM, UEM prompts the user to install the BlackBerry Dynamics app that is configured as the authentication delegate (that app must be assigned to the user). When the user opens this app for the first time, it activates automatically. The user can then activate additional BlackBerry Dynamics apps using Easy Activation.

Programmatic activation

Note: This feature is not currently supported for the BlackBerry Dynamics SDK for Cordova.

The programmatic activation feature enables a BlackBerry Dynamics app to activate without any user interaction and without displaying activation prompts or progress screens. This can be useful when targeting your apps to a consumer audience or for developing apps for devices that have limited or no means of user input.

For more information about programmatic activation, see `programmaticActivityInit` in the BlackBerry Dynamics SDK for Android API Reference or `programmaticAuthorize` in the BlackBerry Dynamics SDK for iOS API Reference.

Note the following implementation details:

- Decide whether you want users to specify a password to unlock a BlackBerry Dynamics app after the initial activation. You can use programmatic activation while still requiring users to type a password to unlock the app. This setting is configured in a BlackBerry Dynamics profile in UEM.
- To activate the app, your application server must use the BlackBerry Web Services REST APIs to retrieve the user credentials and to generate an access key. You may need to create a new UEM user account or to lookup an existing user account. See the User resource in the BlackBerry Web Services REST API reference for the available REST APIs that can be used to create a user, lookup a user, and to generate an access key.
- Pass the user credentials to the app and call `programmaticActivityInit` or `programmaticAuthorize` with the retrieved credentials, setting `ShowUserInterface` to false.

- For Android, receive the broadcast event `GD_STATE_ACTIVATION_ACTION` to track activation progress from `NotActivated` to `InProgress` to `Activated`. You can choose to display a progress indicator during this short period.
- For iOS, observe `GDState.BBActivationState` to track activation progress from `NotActivated` to `InProgress` to `Activated`. You can choose to display a progress indicator during this short period.
- Once activation completes, the user is prompted to set a password (unless you've configured the BlackBerry Dynamics profile to not require a password). The app should wait for the `GD_STATE_AUTHORIZED_ACTION` notification.
- You can use `configureUI` (Android/iOS) to customize the UI of the password screen (for example, with a custom logo and colors).

Secure storage

Secure file system

BlackBerry Dynamics apps store data in a secure, encrypted file system. For more information, see BlackBerry Dynamics File I/O Package in the BlackBerry Dynamics SDK for Android API reference, or GDFileManager and GDFileHandle in the BlackBerry Dynamics SDK for iOS API reference.

Core data

BlackBerry Dynamics apps can store Core Data objects in a secure, encrypted store. For more information, see GDPersistentStoreCoordinator in the API reference.

Secure SQL database

BlackBerry Dynamics apps can leverage a secure SQL database that stores and encrypts data on the user's device. The secure SQL database is based on the SQLite library.

For more information, see the BlackBerry Dynamics SQL Database page in the BlackBerry Dynamics SDK API reference (Android/iOS).

Secure communication

The BlackBerry Dynamics platform enables secure data exchange between a BlackBerry Dynamics app on an end user's device and a back-end application server on the Internet or behind the enterprise firewall. Any communication through the enterprise firewall uses the secure BlackBerry Dynamics proxy infrastructure. One app can communicate with multiple application servers.

To learn more about the programming interfaces for secure communication, see GDSocket and GDHttpClient in the BlackBerry Dynamics SDK for Android API reference, or GDSocket, GDURLLoadingSystem, and NSURLSession Support in the BlackBerry Dynamics SDK for iOS API reference.

AppKinetics

AppKinetics, or Inter-Container Communication (ICC), is a method for securely exchanging data and commands between two BlackBerry Dynamics apps on the same device. The exchange uses a consumer-provider model: one app initiates a service request that the other app receives and responds to as a service provider.

For more information about AppKinetics, see the Inter-Container Communication Package in the BlackBerry Dynamics SDK for Android API reference, or GDService and GDServiceClient in the BlackBerry Dynamics SDK for iOS API reference.

Shared Services Framework

BlackBerry Dynamics apps can communicate with each other and application servers using the Shared Services Framework, a collaboration system that is defined by two components: one that provides a service and another that consumes the service.

The provider can be a client-side service, which is a BlackBerry Dynamics app that uses the GDService APIs (Android/iOS), or a server-side service that is provided by an application server or other remote system. The service is consumed by a BlackBerry Dynamics app that communicates with the provider using AppKinetics (a proprietary BlackBerry ICC protocol) for client-side services or a protocol such as HTTPS for server-side services.

The typical steps that are required to consume a service:

1. Service discovery: The BlackBerry Dynamics app (the consumer) queries for service providers using the GDAndroid.getServiceProvidersFor API or the GDiOS.getServiceProvidersFor API. Service discovery is optional but recommended for both types of services because it respects user entitlements and permissions.
2. Provider selection: The consuming app selects the provider. This is handled by the app code.
3. Service request: The consuming app sends a service request to the provider using the GDServiceClient API (Android/iOS) for client-side services or TCP sockets or HTTP over BlackBerry Dynamics secure communication (Android/iOS) for server-side services.
4. Service response: The consuming app receives the provider response using the same interface that was used for the request (the GDServiceClient API (Android/iOS) for client-side services or BlackBerry Dynamics secure communication (Android/iOS) for server-side services).

Client-side services can be used offline and are ideal if the service requires specific user interaction.

Server-side services can be provided by a clustered application server and are ideal if the server software already exists outside of the BlackBerry Dynamics platform.

Client-side and server-side services both require user entitlement in the management console.

If you want your custom BlackBerry Dynamics app to use the Shared Services Framework, the UEM administrator must specify the app package ID (Android) or bundle ID (iOS) in the BlackBerry Dynamics app settings in the management console. Contact your organization's administrator to provide this information. For instructions for specifying the package ID or bundle ID for an app, see Manage settings for a BlackBerry Dynamics app in the *UEM Administration Guide*.

Sample apps that are included with the SDK demonstrate how to use the Shared Services Framework. For more information about how to use the Shared Services Framework, see the following resources:

- GDAndroid.getServiceProvidersFor
- GDiOS.getServiceProvidersFor
- GDService (Android/iOS)
- GDServiceClient (Android/iOS)
- BlackBerry Dynamics Service Definition (Android/iOS)
- Definitions and descriptions of published services

Server-side services can use the Push Channel API (Android/iOS) to send notifications to BlackBerry Dynamics apps. The channel is end-to-end secure at the same level as BlackBerry Dynamics secure communication. As a result, the BlackBerry Dynamics app does not need to poll the application server, which decreases the load on both the app and the application server. Any application server that is a service provider can use the Push Channel.

Data Leakage Prevention

The BlackBerry UEM administrator can use Data Leakage Prevention (DLP) settings in BlackBerry Dynamics profiles (UEM) to configure data protection standards, including enabling or disabling copy and paste between BlackBerry Dynamics apps and non-BlackBerry Dynamics apps, screen captures, dictation, FIPS, and more.

Contact your organization's administrator to configure DLP standards as necessary for your custom BlackBerry Dynamics apps.

Note the following for the different platforms of the SDK:

SDK platform	Notes
BlackBerry Dynamics SDK for Android	The SDK provides the following classes to manage the secure copy, cut, and paste of data: ClipboardManager, GDTextView, GDEditText, GDAutoCompleteTextView, and GDSearchView. The secure copy-cut-paste sample app demonstrates uses of the secure clipboard. Due to current DLP controls, the setOnReceiveContentListener method is not supported in the following components and widgets: • com.good.gd.widget.GDAutoCompleteTextView • com.good.gd.widget.GDEditText • com.good.gd.widget.GDMultiAutoCompleteTextView • com.good.gd.widget.GDSearchView • com.good.gd.widget.GDTextView • com.good.gd.widget.GDWebView • com.blackberry.bbwebview.BBWebView
BlackBerry Dynamics SDK for iOS	The BlackBerry Dynamics Runtime can secure or block text in transit to or from the clipboard, depending on the configuration of the DLP settings. The Runtime secures text by encrypting it when it is cut or copied to the clipboard and decrypting it when it is pasted. These operations are handled automatically by the Runtime, so no development changes are required.
BlackBerry Dynamics SDK for Cordova	You do not need to make any development changes to support DLP for Cordova apps for iOS or Android. Note that for Android Cordova apps, secure cut, copy, and paste is currently blocked.

User authentication

BlackBerry UEM offers the following options to adjust the user experience for accessing BlackBerry Dynamics apps.

Fingerprint and biometric authentication

Various forms of biometric authentication are supported by the BlackBerry Dynamics SDK, including fingerprint authentication and for Android and Touch ID and Face ID for iOS. The BlackBerry UEM administrator can use a BlackBerry Dynamics profile (UEM) to enable biometric authentication. Contact your organization's administrator to enable and configure these features.

For more information, see BlackBerry Dynamics and Fingerprint Authentication.

Authentication delegation

The BlackBerry UEM administrator can configure up to three BlackBerry Dynamics apps on users' devices to act as an authentication delegate (a primary, secondary, and tertiary delegate). When a user opens any BlackBerry Dynamics app, the device will display the login screen of the authentication delegate app. After the user logs in successfully, all of the BlackBerry Dynamics apps on the device are unlocked. The user does not need to enter a password again until the idle timeout is reached.

If you want your custom BlackBerry Dynamics app to be an authentication delegate, the UEM administrator must specify the app package ID (Android) or bundle ID (iOS) in the BlackBerry Dynamics app settings in the management console. Contact your organization's administrator to provide this information. For instructions for specifying the package ID or bundle ID for an app, see Manage settings for a BlackBerry Dynamics app in the *UEM Administration Guide*.

The administrator configures one or more authentication delegate using a BlackBerry Dynamics profile. It is a best practice to configure the most commonly used app as the authentication delegate. Contact your organization's administrator to configure one or more authentication delegates.

Note: If the administrator configures a secondary authentication delegate, the administrator must notify users that if they delete the primary authentication delegate app, the user must unlock the secondary delegate app and set the app password again so that it can be used to authenticate any additional BlackBerry Dynamics apps. The same requirement applies if a tertiary delegate is configured and the primary and secondary delegate apps are deleted.

Do not require a password

Enabled using a BlackBerry Dynamics profile, this setting removes the password login for BlackBerry Dynamics apps. Users cannot choose whether to use a password.

Do not enable authentication delegation and this setting in the same profile or policy set. This feature is supported in UEM 12.7 or later. If the setting is enabled and then disabled at a later date, users are prompted to create a password the next time they log in to a BlackBerry Dynamics app.

You can use the GDAndroid.getInstance().canAuthorizeAutonomously() or [GDiOS sharedInstance].canAuthorizeAutonomously method to check if this feature is enabled. See the GDInteraction sample app (Android) or the SecureStore sample app (iOS) for examples of this method.

Bypass the app unlock screen

Enabled in the UEM Client settings for a specific BlackBerry Dynamics app (UEM), this setting allows an app to completely bypass the password login screen.

For more information and programming guidance, see the Bypass Unlock Developer Guide.

Background Authorize for iOS

Note: This feature is not currently supported for the BlackBerry Dynamics SDK for Cordova.

Background Authorize is a restricted API that allows a recently locked BlackBerry Dynamics app to use the principal BlackBerry Dynamics APIs (such as secure storage and secure communication) when the app is running in the background.

This feature can be useful in scenarios where the app has stopped unexpectedly and is started in the background in response to an APNS message (for example, a new email). If Background Authorize is enabled, the app can download new data and store it in the secure container. When the user brings the app to the foreground they can authorize and immediately access the data (for example, messages).

To access this restricted API, submit a request to the BlackBerry Dynamics Registrar program at BlackBerryDynamicsRegistrar@blackberry.com.

For more information about this feature, see the Background Authorize Developer Guide.

Background Authorize for Android

Note: This feature is not currently supported for the BlackBerry Dynamics SDK for Cordova.

GDAndroid.canAuthorizeAutonomously allows BlackBerry Dynamics apps to background unlock, receive state callback, and use credential-protected storage. The app can use canAuthorizeAutonomously() to check if it is possible to use background unlock, and if possible, authorize with serviceInit().

Web Authentication

The SDK supports ASWebAuthenticationSession. The BlackBerry Dynamics implementation of ASWebAuthenticationSession utilizes BlackBerry Dynamics secure communication and secure storage for cookies. To protect enterprise credentials from being stored in the iOS keychain, the device user will not be able to use the Safari saved passwords feature in the embedded webview.

Initialize an instance of ASWebAuthenticationSession in your app to allow user authentication through a web service, including those operated by a third party. The page will open in a secure, embedded webview in iOS, or the users default browser (if it supports web authentication sessions) on macOS. For more information, see Authenticating a User Through a Web Service.

You can use Single Sign-On (SSO) with ASWebAuthenticationSession applications by enabling keychain group sharing and using the com.good.gd.data group. The prefersEphemeralWebBrowserSession property will be set to YES by the SDK.

Administrative controls

The BlackBerry UEM administrator can use various server settings, policies, and profiles to manage BlackBerry Dynamics apps and ensure that app usage meets the organization's security standards. Consult with your organization's administrator to ensure that your custom apps adhere to the configured settings in the management console.

For more information, see Controlling BlackBerry Dynamics on users devices, Enforcing compliance rules for devices, and Managing BlackBerry Dynamics apps in the *UEM Administration Guide*.

You also have the option to add new management policies and settings that are specific to your custom BlackBerry Dynamics app to the UEM management console so that they can be configured and applied to users by UEM administrators. For more information, see Adding custom policies for your app to the UEM management console.

Advanced security features with CylancePROTECT Mobile

The BlackBerry Dynamics SDK integrates the CylancePROTECT library to support CylancePROTECT Mobile for UEM. CylancePROTECT is a licensed service that offers a suite of features that enhances UEM's ability to detect, prevent, and resolve security threats without disrupting the productivity of your workforce. CylancePROTECT is configured and managed by the UEM administrator. No additional development or integration effort is required if your organization wants to leverage the CylancePROTECT features for custom BlackBerry Dynamics apps.

CylancePROTECT uses a combination of advanced technologies, including:

- The cloud-based CylanceINFINITY service that uses sophisticated AI and machine learning to identify malware and unsafe URL
- The UEM server that provides a complete device management and compliance infrastructure for your organization
- BlackBerry apps that monitor and enforce security standards at the device and user level

The seamless integration of these technologies establishes a secure ecosystem where data is protected and malicious activities are identified at all endpoints and eliminated proactively.

CylancePROTECT includes the following features:

- Malware detection for Android apps (including BlackBerry Dynamics apps) that are uploaded to UEM for internal deployment
- Malware detection on Android devices
- Sideloaded app detection on iOS and Android devices
- Safe browsing with BlackBerry Dynamics apps
- Insecure network detection on iOS and Android devices.
- Insecure Wi-Fi access point detection on Android devices.
- Integrity checking for BlackBerry Dynamics apps on iOS devices using the Apple DeviceCheck framework
- Hardware certificate attestation for BlackBerry Dynamics apps on Android devices

For more information about CylancePROTECT, see the BlackBerry Protect documentation.

Requirements and support for platform-specific features

This section provides the software requirements for using the SDK, as well as prerequisites that are required to support platform-specific features.

Software requirements

Android development

Item	Requirement
Compatibility with previous versions of the SDK	The BlackBerry Dynamics SDK for Android is compatible with these previous releases of the SDK: • 10.0.x • 10.1.x • 10.2.x • 11.0.x • 11.1.x • 11.2.x
Supported Android OS	Android 10.0 or later
Android API Level	Minimum API level: 29
AndroidX libraries	Your apps must use the AndroidX support libraries to compile successfully. The SDK supports the following minimum versions. It is recommended to use the latest stable version of each library. • androidx.appcompat:appcompat:1.0.0 • androidx.cardview:cardview:1.0.0 • androidx.constraintlayout:constraintlayout:1.1.3 • androidx.core:core:1.0.0 • androidx.legacy:legacy-support-v4:1.0.0 • androidx.preference:preference:1.1.0 • androidx.recyclerview:recyclerview:1.0.0
Build requirements	If the app project uses the SDK .jar distribution, load the shared libraries libsbgse.so and libgdndk.so.
Google Play Services	The SDK uses Google Play Services version 17.0.0 to support some of its functionality. If your app uses the following Google Play Services libraries, verify that you are using the following minimum version or later: • com.google.android.gms:play-services-location:17.0.0 • com.google.android.gms:play-services-safetynet:17.0.0

Item	Requirement
Supported CPU architectures	• ARMv7 • ARMv8 • x86 • x86_64
Java compatibility	Applications must be built using Java 8 compatibility. For more information, see https://developer.android.com/studio/write/java8-support.

Suggested versions of platform and tools	• Android Studio 3 or later
	• The following values specified in sdk/libs/handheld/gd/build.gradle. Other versions of tools will work, but the BlackBerry Dynamics library gradle files might need to be updated accordingly.
	  • com.android.tools.build:gradle: 3.6.3 (or follow the instructions from the Android Developers Blog: Preparing your Grade build for package visibility in Android 11 to use a major Gradle version with the latest dot release)
	  • compileSdkVersion 30
	  • buildToolsVersion "30.0.0"

Package visibility restrictions	Package visibility updates were made in SDK version 8.1 and later to address changes in Android 11 to how apps query and interact with other apps that are installed on the same device. These changes impact apps with the target SDK level set to Android 11 or later only (targetSDKLevel=Android 11). If a BlackBerry Dynamics app is not upgraded to SDK version 8.1 and you change the target SDK level to Android 11, the app cannot communicate with other BlackBerry Dynamics apps on the same device, breaking interoperability features (Easy Activation, authentication delegation, and so on). To support the new package visibility queries element, you must either upgrade the Android Gradle plugin to version 3.6.3 or later or follow the instructions from the Android Developers Blog: Preparing your Grade build for package visibility in Android 11 to use a major Gradle version with the latest dot release. Apps that target Android 11 are able to perform interoperability operations with the apps that don't target Android 11. Apps that are not targeted to Android 11 can run on Android 11 and are not affected by the package visibility restrictions.
Character encoding for build files	Build files (for example, settings.json) must use UTF-8 character encoding. Verify that the editor that you plan to use does not add non-UTF-8 characters or headers. In general, Java does not work with UTF-8-BOM (byte order mark).

Item	Requirement
Supported launch modes	Apps built with the BlackBerry Dynamics SDK for Android support the following launch modes in AndroidManifest.xml: • android:launchMode="standard" • android:launchMode="SingleTop" • android:launchMode="singleTask"
BlackBerry Dynamics Launcher Library	The BlackBerry Dynamics Launcher is a user-friendly interface that allows users to easily access and switch between BlackBerry Dynamics apps, configure app settings, and take advantage of other useful features. For more information, see the BlackBerry Dynamics Launcher Framework documentation. The BlackBerry Dynamics SDK and the BlackBerry Dynamics Launcher Library are mutually dependent. See the BlackBerry Dynamics SDK for Android Release Notes for the required version of the BlackBerry Dynamics Launcher Library.
Restricted key prefix	The key prefix "blackberry" is reserved by BlackBerry and should not be used for key values, key attributes, or key elements. For more information and examples, see the Application Policies Definition in the appendix of the API Reference.
Credential manager requirements	Apps that will use the Credential Manager UI require the following updates: In the app level module of your Gradle file (app/build.gradle), in the dependencies section, add `implementation 'androidx.recyclerview:recyclerview:1.0.0'`.

iOS development

Item	Requirement
Compatibility with previous versions of the SDK	The latest release of the BlackBerry Dynamics SDK for iOS is compatible with these previous releases of the SDK: • 10.0.x • 10.1.x • 10.2.x • 11.0.x • 11.2.x • 11.1.x • 11.2.x
Deployment target	iOS 15 or later
Xcode	Xcode 14 or 15

Item	Requirement
Support for Mac Silicon M1 devices	• BlackBerry Dynamics apps using SDK version 11.0 or later are supported for Mac Silicon devices. • BlackBerry Dynamics apps running on Mac Silicon devices are reported as Mac apps in the UEM management console. • UEM administrators can control access to BlackBerry Dynamics apps using the macOS compliance settings available in the management console. • To support Easy Activation and authentication delegation, you will need to provide the macOS bundle ID for the app in the Apps settings in myAccount.
Supported programming languages	• Objective-C • Swift 4, 4.2, 5
Supported Internet Protocols	• IPv4 • IPv6
Info.plist requirements	In the Info.plist file, add the key "Privacy - Camera Usage Description" with the value "Allow camera usage to scan a QR code". This is not required if the app already uses the camera for its own purposes. For ISV apps, it is recommended that you add the following usage descriptions (or your own custom descriptions) to the Info.plist file to inform the user why the app requires location permission: • NSLocationWhenInUseUsageDescription: "Allow BlackBerry to collect location data, including Wi-Fi address and IP address, and usage patterns only when the app is in use. You may change this setting at any time from your device settings." • NSLocationAlwaysUsageDescription: "Always allow BlackBerry to collect location data, including Wi-Fi address and IP address, and usage patterns, even when the app is not in use. You may change this setting at any time from your device settings." • NSLocationAlwaysAndWhenInUseUsageDescription: "Allow BlackBerry to collect location data, including Wi-Fi address and IP address, and usage patterns both when the app is and is not in use. You may change this setting at any time from your device settings."
Native bundle ID	If you develop a BlackBerry Dynamics app for use on both iPhone and iPad devices, use a single native bundle ID for all variations of the app. UEM will only accept a single native bundle ID.

Item	Requirement
Keychain group sharing for multiple apps	Keychain group sharing allows groups of apps to share information that is stored on a device's keychain. Keychain group sharing is required when you are developing multiple inter-related apps. The setting is part of a project's build. To enable keychain group sharing in an Xcode project, open the project file, navigate to the app target Capabilities tab, and turn on Keychain Sharing. You may be asked for your developer password and to choose a development team. The provisioning profiles for each app must come from the same team and must share the same App ID prefix (see row below). For the Keychain Group, specify `com.good.gd.data`. Also, if you intend to use crypto tokens in your app, specify `com.apple.token`. If the settings for keychain group sharing change, it is recommended to do a fresh reinstall of the new version of the app instead of upgrading the old version. This ensures that the new keychain settings take effect.
App ID prefix	An App ID prefix is a unique ID that groups a collection of apps and enables those apps to share keychain and UIPasteboard data. Apps that share keychain data must have a common App ID prefix from Apple. For more information, see Technical Note TN2311: Managing Multiple App ID Prefixes. The Apple App ID prefix is completely independent of the BlackBerry Dynamics entitlement ID.
BlackBerry Dynamics Launcher Library	The BlackBerry Dynamics Launcher is a user-friendly interface that allows users to easily access and switch between BlackBerry Dynamics apps, configure app settings, and take advantage of other useful features. For more information, see the BlackBerry Dynamics Launcher Framework documentation. The BlackBerry Dynamics SDK and the BlackBerry Dynamics Launcher Library are mutually dependent. See the BlackBerry Dynamics SDK for iOS Release Notes for the required version of the BlackBerry Dynamics Launcher Library.
Restricted key prefix	The key prefix "blackberry" is reserved by BlackBerry and should not be used for key values, key attributes, or key elements. For more information and examples, see the Application Policies Definition in the appendix of the API Reference.

Cordova development

Item	Requirement
Development environment	The macOS platform is recommended for developing an app with the BlackBerry Dynamics SDK for Cordova. Development on a Windows computer is supported only for the Android platform.

Item	Requirement

Supported Cordova libraries	• Cordova 10
	• Cordova 11
	Support for Cordova 10 is deprecated and will be removed in a future release.
	Note: cordova-android@10.1.1 is the only supported version for a Cordova 10 project on Android. To upgrade to this version, run the following commands:
	`$ cordova platform remove android`
	`$ cordova platform add android@10.1.1`

Node.js	10.x (LTS) or later
AngularJS	Version 1.x (without Ionic)
Angular	Version 7.x Note the following limitations for Android only: • Synchronous HTTP requests via XMLHttpRequest are not supported. The requests are treated as asynchronous. • File uploads with FormData() are not supported. Use the FileTransfer plug-in for file uploads and downloads. For more information, see Plug-ins available in the BlackBerry Dynamics SDK for Cordova.
Ionic	• Ionic 5 • Ionic 6 **Note:** Support for Ionic 5 is deprecated and will be removed in a future release. The SDK supports Ionic projects with the angular and Ionic 1 types only; projects with a react or vue type are not supported.

Item	Requirement

Compatibility with BlackBerry Dynamics SDK for Android and iOS	This release of the BlackBerry Dynamics SDK for Cordova is compatible with BlackBerry Dynamics SDK for Android and iOS version 11.1 and 11.2.
	
	To downgrade BlackBerry Dynamics SDK for Cordova to use BlackBerry Dynamics SDK for Android 10.1 or 10.2, do the following:
	
	Open plugins/cordova-plugin-bbd-base/scripts/gradle/bbd.gradle and update the dependencies to the following:
	
	`dependencies { implementation 'com.blackberry.blackberrydynamics:android_handheld_platform:10.x+' implementation 'com.blackberry.blackberrydynamics:android_handheld_backup_support:10.x+'}`
	
	To downgrade BlackBerry Dynamics SDK for Cordova to use BlackBerry Dynamics SDK for iOS 10.1 or 10.2, do the following:Open plugins/cordova-plugin-bbd-base/plugin.xml and replace the podspec URL to the following:
	
	• iOS 10.1:
	
	`<pod name="BlackBerryDynamics" podspec="https://software.download.blackberry.com/repository/framework/dynamics/ios/10.1.0.36/BlackBerryDynamics-10.1.0.36.podspec" />`
	
	• iOS 10.2:
	
	`<pod name="BlackBerryDynamics" podspec="https://software.download.blackberry.com/repository/framework/dynamics/ios/10.2.0.83/BlackBerryDynamics-10.2.0.83.podspec" />`
	
	The BlackBerry Dynamics SDK for Cordova requires use of the BlackBerry Dynamics SDK for iOS dynamic framework. The BlackBerry Dynamics SDK for iOS static framework is no longer supported. For more information, see the readme file for the Base plug-in (cordova-plugin-bbd-base).
	
	It is recommended that you always build and test with the most recent release of the BlackBerry Dynamics SDK for Cordova, to take advantage of new fixes and features.

Unsupported BlackBerry Dynamics features	Android: Data Leakage Prevention (DLP)

Using an entitlement ID and version to uniquely identify a BlackBerry Dynamics app

BlackBerry Dynamics apps are uniquely identified by a BlackBerry Dynamics entitlement ID (GDApplicationID) and entitlement version (GDApplicationVersion). The entitlement ID and entitlement version are used to manage end-user entitlement in the management console. The values are also used for app publishing and service provider registration.

These values are specified in the assets/settings.json file for Android or in the Info.plist file for iOS.

If you are using the BlackBerry Dynamics SDK for Cordova, you will instead specify the entitlement ID and version by adding the following lines to the config.xml file:

```
<preference name="GDApplicationID" value="Your_Custom_Application_ID" />
<preference name="GDApplicationVersion" value="Your_Custom_Application_Version" />
```

You must define both the entitlement ID and the entitlement version for all of your BlackBerry Dynamics apps, regardless of whether you use the Shared Services Framework. The same entitlement ID must be used to represent the app across all platforms.

For more information about setting and checking the entitlement ID and version, the proper format to use for these values, and other requirements and considerations:

- See the GDAndroid Class reference in the BlackBerry Dynamics SDK for Android API Reference, especially these sections:
  - Identification
  - Indirect Authorization Initiation
  - void authorize (GDAppEventListener eventListener) throws GDInitializationError
- See the GDiOS Class reference in the BlackBerry Dynamics SDK for iOS API Reference, especially these sections:
  - Identification
  - Authorization
  - - (void) authorize: (id< GDiOSDelegate > _Nonnull) delegate

Relationship between the entitlement ID and version and native identifiers

The entitlement ID (GDApplicationID) and entitlement version (GDApplicationVersion) of a BlackBerry Dynamics app are different from the native identifiers that are required by the app OS platform. The native identifiers for Android are the packageName and packageVersion values in the AndroidManifest.xml file. The native identifiers for iOS are the CFBundleIdentifier and CFBundleVersion in the Info.plist file. The values of the entitlement ID and entitlement version and the platform native identifiers can be the same, but do not have to be.

To take advantage of BlackBerry Dynamics features such as Easy Activation, authentication delegation, certificate sharing, the Shared Services Framework, and more, the UEM administrator must specify the entitlement ID and version and the native identifier (package name or bundle ID) for a custom BlackBerry Dynamics app in the management console. For more information, see Add an internal BlackBerry Dynamics app entitlement and Manage settings for a BlackBerry Dynamics app in the *UEM Administration Guide*.

The native identifiers for your custom BlackBerry Dynamics app should be unique, especially with respect to apps that are available through public app stores. Duplicate native identifiers can prevent the proper installation or upgrade of an app.

You must change the native app version if you want to distribute a new version of the app. You only need to change the entitlement version if the app starts to provide a new shared service or shared service version, or if the app stops providing a shared service or shared service version.

Using the entitlement version for the Shared Services Framework

For BlackBerry Dynamics apps that provide a service that is consumed by other BlackBerry Dynamics apps through the Shared Services Framework, you should include the entitlement version in the app's AndroidManifest.xml file. This allows the SDK routines that work with the services to identify the required version of the service provider.

See the snippet below, or the AppKinectics sample apps, for examples of how to add the entitlement version to the AndroidManifest.xml file for the service-provider app. The GDApplicationVersion value must match the same value that you specified in assets/settings.json.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.good.gd.example.appkinetics
      [...]
     <app
         [...]
         <meta-data android:name="GDApplicationVersion" android:value
 ="your_value_here"/>
     </app>
</manifest>
```

FIPS compliance

It is a best practice to make your BlackBerry Dynamics apps compliant with U.S. Federal Information Processing Standards (FIPS) 140-2.The BlackBerry Dynamics SDK distribution contains FIPS canisters and tools.

The BlackBerry UEM administrator enables FIPS compliance using a BlackBerry Dynamics profile (UEM). If enabled, BlackBerry Dynamics apps must start in FIPS-compliant mode. The SDK determines whether a service is running in FIPS mode when the app communicates with the server to receive policies.

FIPS compliance enforces the following constraints:

- The use of MD4 and MD5 are prohibited. As a result, access to NTLM-protected or NTLM2-protected web pages and files is blocked.
- In secure socket key exchanges with ephemeral keys, with servers that are not configured to use Diffie-Hellman keys of sufficient length, BlackBerry Dynamics retries with static RSA cipher suites.

Note:

- When you enable FIPS compliance, user certificates must use encryption that meets FIPS standards. If a user tries to import a certificate with encryption that is not compliant, the user receives an error message indicating that the certificate is not allowed and cannot be imported.
- For iOS, when you build for testing with the x86 64-bit simulator, FIPS mode is not enforced. As a result, you might see a difference in behavior with the simulator compared to actual operation. BlackBerry recommends that you always test your app on actual iOS hardware and not rely exclusively on the simulation.
- If you use the SDK dynamic framework, FIPS linking is not required.

Declaring a URL type to support BlackBerry Dynamics features

A BlackBerry Dynamics app for iOS devices must declare a URL type so that it can be discovered by other apps on the same device. This enables AppKinetics, which is required for many BlackBerry Dynamics features. The URL type and schemes are declared in the app's Info.plist file.

The URL type must be the same as the app's native bundle ID. Within the URL type declaration, the following URL schemes must be declared. For example, if the native bundle ID of the app is `com.example.gd.myapp` and its entitlement version (GDApplicationVersion) is 1.0.0.0, then the declared URL type is `com.example.gd.myapp` and the schema declarations are as follows:

Format	Description	Example
`com.good.gd.discovery.enterprise`	Always required for enterprise apps (not required for ISV apps)	Exactly as shown
`<bundle_ID>.sc3`	Always required	`com.example.gd.myapp.sc3`
`<bundle_ID>.sc2`	Enables an app to use authentication delegation and is required for all BlackBerry Dynamics apps	`com.example.gd.myapp.sc2`
`<bundle_ID>.sc2.<GDApplicationVersion>`	Required only if your app provides a discoverable service	`com.example.gd.myapp.sc2.1.0.0.0`

Support for WKWebView

The BlackBerry Dynamics SDK for Cordova is compatible with iOS WKWebView. For details about WKWebView support in the BlackBerry Dynamics SDK for iOS, see the BlackBerry Dynamics SDK for iOS Development Guide.

The Base plug-in (`cordova-plugin-bbd-base`) automatically integrates `cordova-plugin-wkwebview-engine` to add support for WKWebView.

`cordova-plugin-wkwebview-engine` changes the underlying web view control to WKWebView instead of UIWebView.

Note the following support details:

- UIWebView support is removed. Use WKWebView instead.
- The WKWebView Engine plugin is compatible with Cordova 6.0.0 and later.
- The XmlHttpRequest interface will be redirected to the BlackBerry Dynamics secure infrastructure.

Steps to get started with the BlackBerry Dynamics SDK

Step	Action
1	Download and install the BlackBerry Dynamics SDK for Cordova and integrate the BlackBerry Dynamics SDK for Android and/or iOS.
2	Install the BlackBerry Dynamics plug-ins. If you want to implement SafetyNet attestation for your BlackBerry Dynamics app, see Implementing SafetyNet attestation for BlackBerry Dynamics apps before you install the Base plug-in.
3	Consult the BlackBerry Dynamics SDK API reference for instructions for implementing the desired features of the BlackBerry Dynamics platform. See the sample apps included in the SDK package for examples of how to implement key features. For additional guidance and code examples, see the Getting started workflow for BlackBerry developers.
4	Test and debug your app. The SDK allows you to test in enterprise simulation mode and supports automated testing.
5	Deploy your app.
6	Optionally, deploy certificates to the BlackBerry Dynamics apps on users' devices.

Install the BlackBerry Dynamics SDK for Cordova

Before you begin:

Visit BlackBerry Developer Downloads to download the BlackBerry Dynamics SDK for Cordova. When you click the link, you are prompted to log in to the Developer site with your BlackBerry Online Account. If you don't already have an account, you can register and create one. Extract the BlackBerry Dynamics SDK for Cordova package to the desired project directory.

Note: If you are extracting the package on Windows, you will need to use a utility tool such as 7zip or WinRAR.

1. If you are developing for Android, install the Android SDK.
2. Integrate the BlackBerry Dynamics SDK for Android and/or the BlackBerry Dynamics SDK for iOS. You can use the Base plug-in to download the BlackBerry Dynamics SDK for Android using Gradle and the BlackBerry Dynamics SDK for iOS using CocoaPods. For additional information, see the BlackBerry Dynamics SDK Development Guide for the desired platform.

   For iOS apps, you must use the BlackBerry Dynamics SDK for iOS dynamic framework. The BlackBerry Dynamics SDK for iOS static framework is no longer supported.
3. If you are developing for Android, install the Android SDK/ADT bundle.

4. If you are developing for iOS, install Xcode.
5. Install Node.js.
   After the installation completes, the `npm` command is available in your terminal shell.

After you finish:

- Install the BlackBerry Dynamics SDK for Cordova plug-ins. If you want to implement SafetyNet attestation for your BlackBerry Dynamics app, see Implementing SafetyNet attestation for BlackBerry Dynamics apps before you install the Base plug-in.
- If you are developing for iOS, in the **config.xml** file, set the following to enable the enterprise discovery scheme that is required for key BlackBerry Dynamics features:
  `<preference name="BBD_Enterprise_Discovery" value="true" />`
- To enable enterprise simulation mode, in the **config.xml** file, set the following:
  `<preference name="GDEnterpriseSimulationMode" value="true" />`
  For more information about enterprise simulation mode, see the Testing and troubleshooting section of the BlackBerry Dynamics SDK Development Guide for the desired platform.

Using plug-ins to enable your apps

The BlackBerry Dynamics SDK for Cordova comes with Cordova-style plug-ins to enable your apps. Each plug-in represents a specific BlackBerry Dynamics feature. For a description of the available plug-ins, see Plug-ins available in the BlackBerry Dynamics SDK for Cordova.

The plug-ins are available for local installation only. You cannot install them from NPM. The SDK includes the full set of plug-ins, so you can install them by using the `cordova plugin add` command and specifying the file system path.

You can use the `cordova-plugin-bbd-all` plug-in to install all the plug-ins with a single command. You can add plug-ins before or after you add a platform.

In Cordova app development, it is not necessary to use an IDE because Cordova provides command-line tools for compiling, building, and running apps. Android Studio and Xcode are also supported.

Plug-ins available in the BlackBerry Dynamics SDK for Cordova

The plug-ins are stored in a plugins folder (BlackBerry_Dynamics_SDK_for_Cordova_<*version*>/plugins) in the SDK package. Each plug-in folder includes a README.md file that provides installation and implementation details about the plug-in.

Plug-in	Description
capacitor-plugin-bbd-base	This plug-in adds all the configurations that are necessary to use BlackBerry Dynamics in an Ionic-Capacitor app. This plug-in is available on BlackBerry GitHub.
cordova-plugin-bbd-all	This plug-in installs all of the BlackBerry Dynamics SDK for Cordova plug-ins.

Plug-in	Description
cordova-plugin-bbd-appkinetics	This plug-in supports the AppKinectics feature (also known as InterContainer Communication) that enables the secure exchange of data and commands between two BlackBerry Dynamics apps on the same device.
cordova-plugin-bbd-application	This plug-in provides access to information that is globally available to any BlackBerry Dynamics app.
cordova-plugin-bbd-base	This is the Base plug-in that provides the necessary configuration that is required for the BlackBerry Dynamics platform. All of the other plug-ins are dependent on this plug-in.
cordova-plugin-bbd-file	This plug-in enables read/write access to files on the device. This plug-in has been removed from the SDK package and is now available on BlackBerry GitHub: cordova-plugin-bbd-file.
cordova-plugin-bbd-file-transfer	This plug-in enables the download and upload of files to or from a remote server. This plug-in replaces the deprecated cordova-plugin-bbd-filetransfer. This plug-in has been removed from the SDK package and is now available on BlackBerry GitHub: cordova-plugin-bbd-file-transfer.
cordova-plugin-bbd-httprequest	This plug-in is used to send HTTP requests over the Internet.
cordova-plugin-bbd-inappbrowser	This plug-in allows you to securely load articles, videos, and other web resources without the user having to leave the app. This plug-in is available on BlackBerry GitHub: cordova-plugin-bbd-inappbrowser.
cordova-plugin-bbd-interappcommunication	This plug-in is used to return information about a service provider app. **Note:** This plugin is deprecated in the BlackBerry Dynamics SDK for Cordova 10.1 release and will be removed in a future release.
cordova-plugin-bbd-launcher	This plug-in is used to show the BlackBerry Dynamics Launcher in a BlackBerry Dynamics app.
cordova-plugin-bbd-mailto	This plug-in supports sending email messages with attachments.
cordova-plugin-bbd-media-capture	This plug-in supports capturing audio, video, and images using the device's camera and stores them in the BlackBerry Dynamics secure container. The plug-in is a fork of cordova-plugin-media-capture and uses a similar JavaScript API. For iOS and Android 10 and earlier, all media file types are supported. For Android 11 and later, only images are supported. This plug-in is available on BlackBerry GitHub: cordova-plugin-bbd-media-capture.

Plug-in	Description
cordova-plugin-bbd-push	This plug-in contains the response returned from the GDPush class.
cordova-plugin-bbd-serversideservices	This plug-in provides the ability to use BlackBerry Dynamics server-based services, returning the necessary information about a service in JSON format.
cordova-plugin-bbd-socket	This plug-in implements the secure socket communication APIs.
cordova-plugin-bbd-specificpolicies	This plug-in is used to read application-specific policies from the UEM management console and return the policies in JSON format.
cordova-plugin-bbd-sqlite-storage	This plug-in is a secure database object that can be used to manipulate data. This plug-in replaces the deprecated cordova-plugin-bbd-sqlite. This plug-in has been removed from the SDK package and is now available on BlackBerry GitHub: cordova-plugin-bbd-sqlite-storage.
cordova-plugin-bbd-storage	This plug-in is an interface for the secure file system and secure storage. The localStorage API is dependant on cordova-plugin-bbd-file.
cordova-plugin-bbd-tokenhelper	This plug-in is used to request a token from the server and to process callback on the response.
cordova-plugin-bbd-websocket	This plug-in implements the secure WebSocket APIs based on the standard WebSocket specification. see https://developer.mozilla.org/en-US/docs/Web/API/WebSocket for more information.

Installing the Base plug-in

The Base plug-in (`cordova-plugin-bbd-base`) is the main and most important plug-in because it adds the configuration that enables the BlackBerry Dynamics SDK in your app. All of the other plug-ins depend on the Base plug-in.

The Base plug-in supports the "mailto:" URL scheme (RFC-2368 and RFC-6068) to allow the app user to compose email messages. To create the email message, the app will use a secure email service provider such as BlackBerry Work, if available, or a native email application, if available and allowed by the enterprise policy.

By default, the Base plug-in does not support email attachments. You can add the MailTo plug-in (`cordova-plugin-bbd-mailto`) to enable email attachments.

Example: Installing the Base plug-in

```
cd BlackBerry_Dynamics_for_Cordova_<version>/plugins
cordova create <CordovaApp> <my.company.package.name> <CordovaAppName>
cd <CordovaApp>
cordova plugin add ../cordova-plugin-bbd-base
```

Usage declaration for Face ID

The BlackBerry Dynamics SDK for iOS version 4.0.0 and later supports Face ID, a facial recognition feature available for some iOS devices. In the BlackBerry Dynamics profile in UEM, the Face ID option is enabled by default. For more details about Face ID, see the BlackBerry Dynamics SDK for iOS API Reference.

BlackBerry Dynamics apps that use Face ID must declare usage of the Face ID capability. The Base plug-in can add a declaration of Face ID usage to an app by inserting the NSFaceIDUsageDescription property into the `Info.plist` file with the value "Enables authentication without a password."

The Base plug-in adds this declaration by default. To add the declaration manually, either globally or for the iOS platform specifically, set the following preference in the root `config.xml` file:

```
<preference name="addFaceIDUsage" value="On" />
```

You can set the value to "Off" to prevent the automatic addition of the usage declaration. As a best practice, you should only prevent the automatic addition of the usage declaration if the app uses Face ID for a purpose other than BlackBerry Dynamics integration.

Localized versions of the usage message are available as `InfoPlist.strings` files in cordova-plugin-bbd-base/src/ios/resources/Localization/. To add and use the localized strings, see About Information Property List Files.

Default webview for BlackBerry Dynamics apps on Android

When you install the Base plug-in, BBWebView becomes the default webview for BlackBerry Dynamics Cordova apps for Android. When an app uses BBWebView, XMLHttpRequest, fetch ajax requests, and HTML form submissions are intercepted and routed through the secure BlackBerry Dynamics infrastructure.

Local files should be loaded using web-like URLs rather than `file://`. For more information, see WebViewAssetLoader.

If you want to revert to the default Cordova webview, remove the following settings from the root config.xml file and rebuild the project:

```
<preference name="webview"
value="com.good.gd.cordova.core.webview.engine.BBDWebViewEngine" />
```

```
<content src="https://appassets.androidplatform.net/assets/www/*.html" />
```

For an Ionic app, you must also remove the following:

```
<preference name="GDWebView-for" value="ionic" />
```

Check the version of installed plug-ins

To check the version of the plug-ins that you've installed, use either of the following commands:

- `cordova plugins list`
- `cordova plugin`

Security of Cordova localStorage

Take note of the following items about the the security of `localStorage` in relation to `GDSecureStorage`. For more information about `GDSecureStorage`, see the BlackBerry Dynamics API reference.

Item	Description
Use localStorage.getLength(), not localStorage.length	To get the length of `GDSecureStorage`, do not use `localStorage.length`. Use the following method instead: `localStorage.getLength()`. This method returns the size of "localStorage" (`GDSecureStorage`).
localStorage operations secured by the SDK	• If you use `localStorage.setItem(key, value)` to write values to `localStorage`, it is stored in `GDSecureStorage` and is protected. • If you use `localStorage.getItem(key)` to read values from `localStorage`, it is read from `GDSecureStorage`. • If you use `localStorage.removeItem(key)` to remove some value from `localStorage`, it is removed from `GDSecureStorage`. • If you use `localStorage.key(index)` to get the key by index from `localStorage`, the key is retrieved from `GDSecureStorage`. • If you use `localStorage.clear()` to clear `localStorage`, `GDSecureStorage` is cleared.
localStorage operations not secured by the SDK	• If you write values to `localStorage` with `localStorage[key] = value` the value is not stored in `GDSecureStorage` and is not protected. It is stored in the native `Storage` object. • If you read a value from `localStorage` with `var value = localStorage[key]` the value is not retrieved from `GDSecureStorage` but from the native `Storage` object. • If you use `localStorage.length` to get the size of `localStorage`, this returns the length of native `Storage` object, not the length of `GDSecureStorage`. Use `localStorage.getLength()` instead.

Using a custom Activity subclass

The BlackBerry Dynamics SDK for Cordova includes an Activity subclass (`com.good.gd.cordova.core.MainActivity`) that is a subclass of BBDCordovaActivity (`com.good.gd.cordova.core.BBDCordovaActivity`). BBDCordovaActivity extends CordovaActivity.

`com.good.gd.cordova.core.MainActivity` is the default main activity in a Cordova app for Android.

If you want to use a custom CordovaActivity subclass, it must extend `com.good.gd.cordova.core.BBDCordovaActivity` to ensure proper integration with the BlackBerry Dynamics SDK for Android. To prevent merging issues, in <*app*>/plugins/cordova-plugin-bbd-base/plugin.xml, in the appropriate edit-config tag, replace `com.good.gd.cordova.core.MainActivity` with your custom `com.good.gd.cordova.core.BBDCordovaActivity` subclass. Run the `cordova prepare` command to apply the changes.

Using a custom Application subclass

The BlackBerry Dynamics SDK for Cordova includes an Application subclass (`com.good.gd.cordova.core.BBDCordovaApp`) that is the main application class in a Cordova app for Android.

If you want to use a custom Application subclass, it must extend `com.good.gd.cordova.core.BBDCordovaApp` to ensure proper integration with the BlackBerry Dynamics SDK for Android. To prevent merging issues, in <*app*>/plugins/cordova-plugin-bbd-base/plugin.xml,

in the appropriate edit-config tag, replace `com.good.gd.cordova.core.BBDCordovaApp` with your custom `com.good.gd.cordova.core.BBDCordovaApp` subclass. Run the `cordova prepare` command to apply the changes.

Implementing SafetyNet attestation for BlackBerry Dynamics apps

BlackBerry UEM version 12.10 and later supports SafetyNet attestation for BlackBerry Dynamics apps. You can use SafetyNet to extend BlackBerry root and exploit detection and to enhance app security and integrity. For more information about SafetyNet attestation, implementation considerations, and instructions for enabling the feature, see the BlackBerry UEM Configuration Guide. This chapter details considerations for developers who want to enable SafetyNet support for their BlackBerry Dynamics apps.

To support SafetyNet, you must add a new library component to the app project, complete SafetyNet registration, and update the BlackBerry Dynamics application policy file.

Adding the GDSafetyNet library to the app project

The BlackBerry Dynamics SDK for Android version 5.0 and later includes a GDSafetyNet library. To support SafetyNet, add this library to the app project dependencies along with the main GDLibrary.

The GDSafetyNet library includes all of the client-side source code that is required to support SafetyNet. No additional app code is required. The GDSafetyNet library requires Google Play Services 11.0 or later to use device SafetyNet APIs. Verify that your BlackBerry Dynamics app is dependent on a single version of Google Play Services only.

Add the following to cordova-plugin-bbd-base/scripts/gradle/bbd.gradle before you add the base plug-in to the app:

```
implementation 'com.google.android.gms:play-services-safetynet:xx.x.x'
implementation 'com.blackberry.blackberrydynamics:android_handheld_gd_safetynet:+'
```

Completing SafetyNet registration

You must obtain an API key from Google and add it to the app's AndroidManifest.xml file. The API key has a quota limit of 10,000 requests each day. If necessary, you can use this online form to request an increase to this quota.

For the following sections of the form, select these answers:

- Method of validating SafetyNet Attestation API responses: Server side - by verifying the signing certificate chain.
- Method of retrying in case of errors: Something else (or unknown). In the "Anything else that you want to tell us" field, type the following: If the SafetyNet Service responds to the device with a transient error, then the device will retry with an exponential back-off. If the SafetyNet Service responds with a valid blob, but the blob is found to contain an error value when it is decoded, then attestation is retried after 15 minutes.

In the AndroidManifest.xml file, add the following to the <application> element:

```
<meta-data android:name="com.blackberry.attestation.ApiKey"
 android:value="YOUR_API_KEY" />
```

Updating the BlackBerry Dynamics application policy file

During a SafetyNet attestation process, BlackBerry UEM uses the app response to verify that it is communicating with the official version of the app. You must provide this information in the application policy file.

Consider the following example from the Greetings Client sample app in the BlackBerry Dynamics SDK:

```
<?xml version="1.0" encoding="utf-8"?>
<apd:AppPolicyDefinition xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:apd="urn:AppPolicySchema1.good.com"
    xsi:schemaLocation="urn:AppPolicySchema1.good.com AppPolicySchema.xsd" >
    <pview>
        <pview>
            <sendto client="None" />
            <desc>SafetyNet Attestation Supported</desc>
            <pe ref="apkCertificateDigestSha256"/>
            <pe ref="apkPackageName" />
            <pe ref="Description" />
        </pview>
    </pview>
    <setting name="apkCertificateDigestSha256">
        <hidden>
            <key>blackberry.appMetadata.android.apkCertificateDigestSha256</key>

 <value>DD:83:CA:47:09:FA:C5:33:75:FE:F4:A1:B5:FB:F4:A8:E8:C2:7A:DF:AF:24:
0D:7B:E3:BA:BD:FB:A9:2B:F9:D6</value>
    </hidden>
    </setting>
    <setting name="apkPackageName">
        <hidden>
            <key>blackberry.appMetadata.android.apkPackageName</key>
            <value>com.good.gd.example.services.greetings.client</value>
        </hidden>
    </setting>
    <setting name="Description" >
        <text>
            <key>snet</key>
            <label>Safety Net</label>
            <value>Safety Net</value>
        </text>
    </setting>
</apd:AppPolicyDefinition>
```

The app is uniquely identified by the combination of the official package name (in the example above, blackberry.appMetadata.android.apkPackageName) and the digest hash of the official signing key (in the example above, blackberry.appMetadata.android.apkCertificateDigestSha256). To determine the digest hash, you can use the following keytool command, specifying the keystore and key name that was used to sign the app:

```
keytool -list -v -keystore <KEYSTORE_NAME> -alias <KEY_NAME>
```

This command will provide a response like the following:

```
Creation date: 4-Sep-2018
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Sample
```

```
Issuer: CN=Sample
Serial number: 27c738c9
Valid from: Tue Sep 04 08:28:10 BST 2018 until: Wed Aug 22 08:28:10 BST 2068
Certificate fingerprints:
     MD5:  4C:30:85:93:5E:96:12:90:CF:A0:77:48:A5:CA:63:8F
     SHA1: 3C:52:A0:2A:76:63:15:C9:20:C1:06:D9:4D:75:7C:14:D6:7C:30:BC
     SHA256:
 DD:83:CA:47:09:FA:C5:33:75:FE:F4:A1:B5:FB:F4:A8:E8:C2:7A:DF:AF:24:0D:7B:E3:
BA:BD:FB:A9:2B:F9:D6
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
```

After you update the application policy file, coordinate with the BlackBerry UEM administrator to upload the app to UEM (see Deploying your BlackBerry Dynamics app) and to upload the application policy file in the management console (see Manage settings for a BlackBerry Dynamics app in the *UEM Administration Guide*). Before the administrator uploads the application policy file, verify that the Android app package ID has been specified or that the app source file has been uploaded; both settings are configured in the app entitlement settings (Android tab) in the management console.

UEM validates the format of the input package name and digest hash. If you update the application policy file and upload the app again, it can take up to 24 hours for the change to synchronize to all UEM instances. When the app is uploaded again, it is removed from the current list of apps that are enabled for attestation and must be added again.

Using the BlackBerry Web Services REST APIs for SafetyNet attestation and status

The BlackBerry Web Services for BlackBerry UEM are a collection of REST APIs that you can use to execute administrative actions in BlackBerry UEM or to retrieve status information about UEM users, groups, devices, and the overall UEM domain. The BlackBerry Web Services version 12.10 and later provide REST APIs that you can use to both initiate and check the status of SafetyNet attestation.

For an introduction to the BlackBerry Web Services REST APIs, see the Getting started section in the REST API reference.

You can use the following APIs to initiate attestation for a specific BlackBerry Dynamics app or for a user's device:

- POST /{tenantGuid}/api/v1/users/{userGuid}/userDevices/{userDeviceGuid}/applications/{appGuid}/commands
- POST /{tenantGuid}/api/v1/users/{userGuid}/userDevices/{userDeviceGuid}/commands

You can use the following APIs to retrieve the attestation status (as well as other status information) for all of a user's devices, for a specific device, or for all of the apps on a specific device:

- GET /{tenantGuid}/api/v1/users/{userGuid}/userDevices
- GET /{tenantGuid}/api/v1/users/{userGuid}/userDevices/{userDeviceGuid}
- GET /{tenantGuid}/api/v1/users/{userGuid}/userDevices/{userDeviceGuid}/applications

The returned status information provides the time that an attestation result was last reported. You can establish a trust window in which an attestation call is not required (for example, four hours). If you do use a REST API for an ATTEST call, you must get the attestation status later (asynchronously), as there is no notification that UEM has completed the attestation process.

You can use the REST APIs for various use cases. For example, if you have a server application that is used by your organization's internal mobile apps, you could have the server application use the REST APIs noted above

to check the app's attestation status (or to initiate an attestation challenge) before releasing data to the app or accepting data from it.

Note the following about the communication channels for the REST APIs:

- The REST APIs that attest the device or get status information for the device communicate with the BlackBerry UEM Client over a direct device channel that doesn't require user authentication.
- The REST APIs that attest a specific BlackBerry Dynamics app remain pending on the UEM server until the app is running on the device. When the app starts and is authenticated, it connects to UEM and the attestation challenge occurs.

Testing the app

After completing the integration tasks, your app is SafetyNet ready and you can proceed with further testing. Verify that the app displays in the list of SafetyNet capable apps in the UEM management console. If the app does not display in the list, it is likely that UEM was not able to parse the application policy file.

When the app appears in the list of SafetyNet capable apps, the administrator can then enable the app for SafetyNet. For instructions, see the BlackBerry UEM Configuration Guide. An app that was signed correctly will activate successfully. An app that was not signed correctly will not activate and a SafetyNet validation failure message will display in the app.

Sample apps

Sample app	Description
ApacheHttp	Demonstrates the use of the `GDHTTPRequest` plug-in APIs.
AppKinetics Client and Server	Demonstrates how to use the file transfer service of the AppKinetics APIs.
Blank	Provides an empty skeleton app that you can use as a starting point.
InAppBrowser	Demonstrates how to use the cordova.InAppBrowser.open API of cordova-plugin-bbd-inappbrowser to securely open articles, videos, and other web resources without the user having to leave the app.
MailTo	Demonstrates how to use the "mailto" service to send email.
OnProgress	Demonstrates the proper usage of the "onprogress" event of `GDFileTransfer` plugin APIs.
Policy	Demonstrates how to use app-specific policies and server-side services.
RssReader	Demonstrates how to use `GDHTTPRequest` and `GDStorage` APIs to fetch RSS feeds.
Secure ICC	Demonstrates Secure Inter-Container Communication (ICC) within an Ionic 6 app. There are two Secure ICC sample apps available: • Secure-ICC sample app for Ionic Cordova with Angular integration • Secure-ICC sample app for Ionic Capacitor with Angular integration They are both an open source apps and are available on the BlackBerry Dynamics for Cordova Sample App GitHub page.
SQLite	Demonstrates how to use secure storage APIs.
UnitTest	Demonstrates how to use all APIs supported in this version of the SDK.
WebSocket	Demonstrates how to use the WebSocket API to communicate with WebSocket servers using the WebSocket protocol. Both ws:// and wss:// schemes are supported. The WebSocket API can send different data types to the server and receives messages of different types from the server.

Note that in SDK version 6.0 and later, the build workflow has changed. As a result, you must specify `npm i` before adding an OS platform. For example:

```
$cd <GD_Cordova_Folder>/SampleApplications/com.blackberry.bbd.example.cdv.blank
$npm i
$cordova platform add android
$cordova platform add ios
$cordova build
```

Testing and troubleshooting

This section provides guidance for testing and troubleshooting issues with your BlackBerry Dynamics apps.

The BlackBerry Dynamics SDK for Android and iOS supports automated testing using an Automated Test Support Library. For instructions for implementing automated testing, see the testing and troubleshooting section in the BlackBerry Dynamics SDK for Android Development Guide or the BlackBerry Dynamics SDK for iOS Development Guide.

Logging and diagnostics

The processing activity of the BlackBerry Dynamics Runtime is logged by the runtime itself. The activity log is written to the BlackBerry Dynamics secure container on the device after deployment. You can configure how your BlackBerry Dynamics apps generate console log information. For more information about console logs controlled by developers and container logs controlled by UEM administrators, see the BlackBerry Dynamics Runtime activity log appendix in the API reference (Android/iOS).

If your app uses SDK version 5.0 or later, and the UEM administrator has turned off "Enable detailed logging for BlackBerry Dynamics apps" in the BlackBerry Dynamics profile (UEM), the app does not generate console log information. This provides additional protection against attacks by malicious users. This change has no impact on how container logs are generated.

The "Enable detailed logging for BlackBerry Dynamics apps" setting is off by default.

For BlackBerry Dynamics apps running SDK version 5.0 or later, console logs are generated only if this setting is turned on or if the app is running in enterprise simulation mode.

For more information about log management and configuring appropriate log settings for BlackBerry Dynamics Cordova apps, see the Logs management section of the README file for the Base plugin (cordova-plugin-bbd-base).

Configure logging for the Xcode console

The messages that are printed to the Xcode console can be configured in the build targets of the app project. Different targets can have different activity logging configurations. You can configure the logging to be detailed or selective. Changes to the console logging configuration take effect the next time the target is built and run, and do not affect the container log.

1. Open the app's **Info.plist** file.
2. Perform one of the following tasks:

Task	Steps
Configure detailed logging	**a.** Add a new row with the following values: • Key: **GDConsoleLogger** • Type: **String** • Value: **GDFilterNone** **b.** In Xcode 9.3 or later, to view the detailed logs, open the console app (Window > Devices and Simulators, or from OSX, Applications > Utilities > Console.app) and enable the following settings in the **Action** menu: • **Include Info Messages** • **Include Debug Messages**
Configure selective logging	**a.** Add a new row with the following values: • Key: **GDConsoleLogger** • Type: **Array** **b.** For each category that you want to omit from the console log, add a row under the logger row as an item in the logger's array. For each item, set the Type to **String** and the value to the desired logging level: **GDFilterDetailed**, **GDFilterInfo**, **GDFilterWarnings**, or **GDFilterErrors**.

Monitoring app log uploads by device users

You can use the GDLogManager class (Android/iOS) to monitor app log file uploads that are initiated by your organization's device users. This class does not manage or display information about log uploads that are initiated by the UEM administrator, or by management console profiles or policies.

GDLogManager provides the following information:

- Upload size
- Amount of data uploaded
- Events that indicate the following states:
  - Upload completed
  - Upload abandoned or canceled
  - Upload suspended
  - Upload resumed after suspension
- Actions for managing a log upload (cancel, suspend, resume)
- Enable detailed logging with detailedLoggingFor (Android/iOS)

When detailed logging is disabled by the management server profiles or policies, calling any API in the GDLogManager class has no effect. It is a best practice to check this setting using the getApplicationConfig API (Android/iOS). If detailed logging is enabled, present the log upload progress UI or any other related UI.

The following sample apps that are included in the BlackBerry Dynamics SDK for iOS demonstrate how to use GDLogManager:

- Objective-C samples: RSSReader, BypassUnlock, GreetingsClient, and GreetingsServer
- All Swift samples

Deploying your BlackBerry Dynamics app

Before you deploy your BlackBerry Dynamics app to your organization's production environment, you should test the app and the deployment process in a BlackBerry UEM environment that is reserved for development testing and evaluation. Coordinate with your organization's administrator to get access to a dedicated test environment.

BlackBerry Dynamics apps are fully supported for BlackBerry UEM. BlackBerry UEM is the recommended enterprise management solution to implement and use going forward, because it provides advanced app and user management features, advanced connectivity and networking options, expanded compliance and integrity checking, and the most recent BlackBerry Web Services REST APIs that your apps can leverage.

See the following resources for more information about distributing and managing your app in a BlackBerry UEM environment:

Task	Resource
Add your app to BlackBerry UEM and distribute it to users	See the following topics in the *BlackBerry UEM Administration Guide*: • Apps • Add an internal BlackBerry Dynamics app entitlement • Managing BlackBerry Dynamics apps
Configure BlackBerry Dynamics profiles that impact app functionality	See the following topics in the *BlackBerry UEM Administration Guide*: • Controlling BlackBerry Dynamics on users devices • BlackBerry Dynamics profile settings • BlackBerry Dynamics connectivity profile settings • Assigning profiles
Collect activity and compliance violation information for BlackBerry Dynamics apps	See the following topic in the *BlackBerry UEM Administration Guide*: • Activity and compliance violation reports for BlackBerry Dynamics apps

Deploying certificates to BlackBerry Dynamics apps

You can use any of the following options to deploy certificates to BlackBerry Dynamics apps. Each method requires configuration in the management console. Coordinate with your organization's administrator to select and configure the desired option.

Option	For more information
Personal Information Exchange files	See Using Personal Information Exchange files in this section.
CA certificate profile	See Sending CA certificates to devices and apps in the *UEM Administration Guide*.
User credential profile	See Sending client certificates to devices and apps using user credential profiles in the *UEM Administration Guide*.
SCEP profile	See Sending client certificates to devices and apps using SCEP in the *UEM Administration Guide*.
Shared certificate profile	See Sending the same client certificate to multiple devices in the *UEM Administration Guide*.

After certificates are distributed to a user's device, those certificates are shared and used by all of the BlackBerry Dynamics apps on the device. No additional programming is required by the app developer to support client certificates.

The management server and BlackBerry Dynamics apps also support the use of Kerberos for service authentication. For more information, see Using Kerberos in this section.

The SDK also provides a Crypto C language programming interface that allows an app to retrieve public key certificates that are stored in the BlackBerry Dynamics credentials store and use those certificates for signing and verification of messages and documents such as PDFs. Note that BlackBerry Infrastructure certificates cannot be retrieved from the store and that the private key will remain inaccessible. For more information, see the Crypto C Programming Interface appendix (Android/iOS) in the API reference.

Using Personal Information Exchange files

An organization can deploy corporate services that require two-way SSL/TLS authentication for users. A user is issued a password-protected Personal Information Exchange file (PKCS12 format, .p12 or .pfx) containing an SSL/TLS client certificate and a private key. This file can be provided to BlackBerry Dynamics apps to grant access to secure corporate services.

The BlackBerry Dynamics SDK supports the use of Personal Information Exchange files to authenticate BlackBerry Dynamics apps and to access secure services. All of the required operations to support client certificates are carried out by the BlackBerry Dynamics Runtime, with no additional programming required to handle the authentication challenge. For more information on how this is handled, refer to *HttpViewController.swift* in the Dynamics-iOS-Swift sample app. The app can use client certificates if:

- The app uses the BlackBerry Dynamics Secure Communication Networking APIs.
- The device user's UEM account is configured to support certificates.
- The certificates satisfy the certificate requirements.

After a user activates a BlackBerry Dynamics app, the app receives the Personal Information Exchange files. For each file, the user is prompted to provide the issued password so that the files and identification material can be installed. When this process is complete, the app can access the server resources that require two-way SSL/TLS authentication.

If more than one Personal Information Exchange file is required per user, the BlackBerry Dynamics Runtime selects the appropriate certificate using the following criteria:

1. Only client certificates that are suitable for SSL/TLS client authentication are eligible to send to the server. Certificates must have no Key Usage or Extended Key Usage, or Key Usage that contains “Digital Signature" or "Key Agreement”, or Extended Key Usage that contains "TLS Web Client Authentication”. Key Usages and Extended Key Usages must not contradict allowances for SSL/TLS client authentication.
2. If the server advertises the client certificate authority in the SSL/TLS handshake, only client certificates that have been issued by that authority are considered.
3. Expired certificates and certificates that are not yet valid cannot be selected.
4. If more than one certificate satisfies the above criteria, the BlackBerry Dynamics Runtime selects the most recently issued certificate.

Configuring support for client certificates

Certificate support is configured in the management console by the administrator. Contact your organization’s administrator to configure certificate support for BlackBerry Dynamics apps.

For more information about configuring certificate support in BlackBerry UEM, see the following:

- Manage settings for a BlackBerry Dynamics app in the *UEM Administration Guide*
- Sending certificates to devices using profiles in the *UEM Administration Guide*
- Connect BlackBerry UEM to a BlackBerry Dynamics PKI Connector in the *UEM Administration Guide*

Certificate requirements

- Client certificates must be in PKCS12 format, with the Certificate Authority (CA), public key, and private key in the same file.
- The PKCS12 file must have a .p12 or .pfx extension
- The PKCS12 file must be password-protected
- The source of the certificate can be your own internal CA, a well-known public CA, or an online tool such as OpenSSL or the Java keytool. You can use the following keytool example to generate a certificate, substituting your own values as required:

```
keytool -genkeypair -alias good123 -keystore good123.pfx -storepass good123 -
validity 365 -keyalg RSA -keysize 2048 -storetype pkcs12
```

- If the organization’s security policy uses FIPS standards, Personal Information Exchange files must be encrypted with FIPS-strength ciphers. If Personal Information Exchange files use a weak cipher, which is common for third-party applications when exporting identity material, you can use a tool like OpenSSL to re-encrypt the files with a FIPS-strength cipher. See the following example:

```
openssl pkcs12 -in weak.p12 -nodes -out decrypted.pem
       <enter password>
       openssl pkcs12 -export -in decrypted.pem -keypbe AES-128-CBC -certpbe
 AES-128-CBC -out strong.p12
      <enter password>
          rm decrypted.pem
```

Using Kerberos

BlackBerry Dynamics apps support both Kerberos PKINIT with PKI certificates and Kerberos Constrained Delegation. Kerberos PKINIT and Kerberos Constrained Delegation are distinct implementations of Kerberos. You can support one or the other for BlackBerry Dynamics apps, but not both.

With Kerberos PKINIT, authentication occurs directly between the BlackBerry Dynamics app and the Windows Key Distribution Center (KDC). User authentication is based on certificates that are issued by Microsoft Active Directory Certificate Services. No additional programming is required by the app developer to use Kerberos PKINIT.

With Kerberos Constrained Delegation, authentication is based on a trust relationship between the management server (BlackBerry UEM and a KDC. The management server communicates with the service on behalf of the app.

For more information about how to configure the desired Kerberos implementation in UEM, including requirements and prerequisites, see Configuring Kerberos for BlackBerry Dynamics apps in the *UEM Administration Guide*.

Legal notice

WERE FORESEEN OR UNFORESEEN, AND EVEN IF BLACKBERRY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW IN YOUR JURISDICTION, BLACKBERRY SHALL HAVE NO OTHER OBLIGATION, DUTY, OR LIABILITY WHATSOEVER IN CONTRACT, TORT, OR OTHERWISE TO YOU INCLUDING ANY LIABILITY FOR NEGLIGENCE OR STRICT LIABILITY.

THE LIMITATIONS, EXCLUSIONS, AND DISCLAIMERS HEREIN SHALL APPLY: (A) IRRESPECTIVE OF THE NATURE OF THE CAUSE OF ACTION, DEMAND, OR ACTION BY YOU INCLUDING BUT NOT LIMITED TO BREACH OF CONTRACT, NEGLIGENCE, TORT, STRICT LIABILITY OR ANY OTHER LEGAL THEORY AND SHALL SURVIVE A FUNDAMENTAL BREACH OR BREACHES OR THE FAILURE OF THE ESSENTIAL PURPOSE OF THIS AGREEMENT OR OF ANY REMEDY CONTAINED HEREIN; AND (B) TO BLACKBERRY AND ITS AFFILIATED COMPANIES, THEIR SUCCESSORS, ASSIGNS, AGENTS, SUPPLIERS (INCLUDING AIRTIME SERVICE PROVIDERS), AUTHORIZED BLACKBERRY DISTRIBUTORS (ALSO INCLUDING AIRTIME SERVICE PROVIDERS) AND THEIR RESPECTIVE DIRECTORS, EMPLOYEES, AND INDEPENDENT CONTRACTORS.

IN ADDITION TO THE LIMITATIONS AND EXCLUSIONS SET OUT ABOVE, IN NO EVENT SHALL ANY DIRECTOR, EMPLOYEE, AGENT, DISTRIBUTOR, SUPPLIER, INDEPENDENT CONTRACTOR OF BLACKBERRY OR ANY AFFILIATES OF BLACKBERRY HAVE ANY LIABILITY ARISING FROM OR RELATED TO THE DOCUMENTATION.

Prior to subscribing for, installing, or using any Third Party Products and Services, it is your responsibility to ensure that your airtime service provider has agreed to support all of their features. Some airtime service providers might not offer Internet browsing functionality with a subscription to the BlackBerry® Internet Service. Check with your service provider for availability, roaming arrangements, service plans and features. Installation or use of Third Party Products and Services with BlackBerry's products and services may require one or more patent, trademark, copyright, or other licenses in order to avoid infringement or violation of third party rights. You are solely responsible for determining whether to use Third Party Products and Services and if any third party licenses are required to do so. If required you are responsible for acquiring them. You should not install or use Third Party Products and Services until all necessary licenses have been acquired. Any Third Party Products and Services that are provided with BlackBerry's products and services are provided as a convenience to you and are provided "AS IS" with no express or implied conditions, endorsements, guarantees, representations, or warranties of any kind by BlackBerry and BlackBerry assumes no liability whatsoever, in relation thereto. Your use of Third Party Products and Services shall be governed by and subject to you agreeing to the terms of separate licenses and other agreements applicable thereto with third parties, except to the extent expressly covered by a license or other agreement with BlackBerry.

The terms of use of any BlackBerry product or service are set out in a separate license or other agreement with BlackBerry applicable thereto. NOTHING IN THIS DOCUMENTATION IS INTENDED TO SUPERSEDE ANY EXPRESS WRITTEN AGREEMENTS OR WARRANTIES PROVIDED BY BLACKBERRY FOR PORTIONS OF ANY BLACKBERRY PRODUCT OR SERVICE OTHER THAN THIS DOCUMENTATION.

BlackBerry Enterprise Software incorporates certain third-party software. The license and copyright information associated with this software is available at http://worldwide.blackberry.com/legal/thirdpartysoftware.jsp.

BlackBerry Limited
2200 University Avenue East
Waterloo, Ontario
Canada N2K 0A7

BlackBerry UK Limited
Ground Floor, The Pearce Building, West Street,
Maidenhead, Berkshire SL6 1RL
United Kingdom

Published in Canada