

# BlackBerry Dynamics Launcher Library

for Android and iOS

Last updated: Tuesday, August 29, 2017

Versions: BlackBerry Launcher Library for Android and iOS 2.6



©2017 BlackBerry Limited. Trademarks, including but not limited to BLACKBERRY, BBM, BES, EMBLEM Design, ATHOC, MOVIRTU and SECUSMART are the trademarks or registered trademarks of BlackBerry Limited, its subsidiaries and/or affiliates, used under license, and the exclusive rights to such trademarks are expressly reserved. All other trademarks are the property of their respective owners. All other trademarks are the property of their respective owners. This documentation is provided "as is" and without condition, endorsement, guarantee, representation or warranty, or liability of any kind by BlackBerry Limited and its affiliated companies, all of which are expressly disclaimed to the maximum extent permitted by applicable law in your jurisdiction.

# Contents

Revision history .....	5
What's new? .....	7
Required: Gradle version 2.14.1 and minSdkVersion 19 settings on Android .....	7
Info: new features in the Launcher Library .....	7
Logging now in parallel with BlackBerry SDK .....	7
About the BlackBerry Launcher Library .....	8
Prerequisite server setup by IT administrator for advanced features .....	8
BlackBerry Dynamics SDK and the BlackBerry Launcher Library correspondence required .....	9
About BlackBerry Dynamics software version numbers .....	9
New in previous releases .....	9
Customization of Launcher icon .....	9
Integrating the Launcher Library for display on Android .....	11
Setup in Gradle and Android Studio for sample application .....	11
Steps to configure the sample application .....	11
Steps to configure an existing Android project .....	11
Use global instance of GDStateListener, not per-activity listener .....	12
Initialization, registration and program setup .....	12
Proguard configuration .....	13
Integrating the Launcher Library on iOS .....	14
Adding to Xcode project .....	14
Initializing .....	14
Starting .....	15
Example of onAuthorized with startServicesWithOptions .....	15

Sample application for Android and iOS .....	17
Moving to production .....	18
Service binding .....	18
Customizing your app's icon: required for iOS .....	18
Customizing the Launcher's logo in BEMS .....	18
Validating the BlackBerry Enterprise Mobility Server SSL certificate .....	18
Prerequisites .....	18
General steps .....	18
BlackBerry Dynamics documentation .....	20

## Revision history

*BlackBerry Dynamics Launcher Library*

Date	Description
2017-08-29	<p><b>Important:</b> On iOS, be sure to add the following line at the end of your initialization code, as highlighted in below. Otherwise, your icons might not be displayed properly. See <a href="#">Initializing</a></p> <pre>[launchPadVC startServicesWithOptions:GTLInternalGDAuthTokenAndPushConnectionManagement];</pre>
2018-08-22	Version number for iOS updated for latest release. No content changes.
2017-07-18	Updated for latest release
2017-04-25	Updated for latest release. See <a href="#">What's new?</a>
2017-03-21	<p>Added new sections:</p> <ul style="list-style-type: none"> <li>• <a href="#">Directly launching the app store via the Launcher Library on Android</a></li> <li>• <a href="#">Directly launching app stores via the Launcher Library on iOS</a></li> </ul>
2017-01-12	Updated code snippets in <a href="#">Initializing</a> for iOS to match the packaged sample application
2017-01-11	<ul style="list-style-type: none"> <li>• Clarified in <a href="#">Prerequisite server setup by IT administrator for advanced features</a> that without BEMS, you are limited to only the most basic features of the BlackBerry Launcher Library</li> <li>• In <a href="#">Moving to production</a>, corrected link to BEMS documentation at <a href="http://help.blackberry.com/en/blackberry-enterprise-mobility-server/current/bems-install-and-config-html/">http://help.blackberry.com/en/blackberry-enterprise-mobility-server/current/bems-install-and-config-html/</a></li> </ul>
2017-01-10	Updated for latest version; no content changes.
2017-01-06	Revised <a href="#">Setup in Gradle and Android Studio for sample application</a>
2016-12-15	Updated for latest release. See details in <a href="#">What's new?</a>
2016-08-24	Version numbers updated for latest release: compatibility with Android N; no content changes.
2016-08-18	Version numbers updated for latest release: compatibility with iOS 10; no content changes.

## Revision history

Date	Description
2016-06-29	<ul style="list-style-type: none"> <li>Version numbers updated for latest release.</li> <li>This release of the BlackBerry Launcher Library corresponds with the latest release of the BlackBerry Dynamics SDK for Android and for iOS.</li> <li>The timeout of the retrieval of the profile (viewable contents of the Launcher) has been increased to 24 hours.</li> <li>New <a href="#">Setup in Gradle and Android Studio for sample application</a></li> </ul>
2016-05-26	Updated for latest release of BlackBerry Dynamics SDK, plus changes names or steps in: <ul style="list-style-type: none"> <li><a href="#">Setup in Gradle and Android Studio for sample application</a></li> <li><a href="#">Adding to Xcode project</a></li> </ul>
2016-03-31	For the Android version, added <a href="#">Setup in Gradle and Android Studio for sample application</a> with clarified steps for building.
2016-03-10	Truncated revision history to reduce bulk.
2016-02-29	Added cross-reference to <a href="#">BlackBerry Dynamics Device and Application Management</a> for details about enabling features on Good Control so they appear in the Launcher on user devices.
2016-01-28	Clarified the need for maintaining correspondence between BlackBerry Launcher Library versions and versions of the BlackBerry Dynamics SDK: you need to be concerned with this correspondence only if you use the BlackBerry Launcher Library.
2016-01-19	Version numbers updated for latest release; no content changes.
2015-12-23	Updated for release of BlackBerry Launcher Library for iOS v2.2.0.156. No content changes.
2015-11-02	Clarified exactly which Android support library is required in <a href="#">Integrating the Launcher Library for display on Android</a> .
2015-08-28	First issued

## What's new?

### Required: Gradle version 2.14.1 and minSdkVersion 19 settings on Android

The BlackBerry Dynamics Launcher Library for Android now relies on the following default settings for Gradle.

Minimum supported Gradle version 2.14.1

- **minSdkVersion 19**
- **NSURLSessionConfigurationObject.**

### Info: new features in the Launcher Library

Unless stated below, the following list of new features in this release of the BlackBerry Dynamics Launcher Library applies to both Android and iOS platforms.

No programming is necessary to take advantage of these new features.

- The Launcher Library now supports displaying web apps.
- App icons in Launcher can now be organized by the end-user. This end-user organization is synchronized across all apps the end-user runs.
- The Launcher now includes a search filter to locate specific apps installed on a device.

### Logging now in parallel with BlackBerry SDK

This release of the BlackBerry Launcher Library now honors the log level setting specified for the BlackBerry Dynamics SDK itself. For example, if the SDK has been configured to log at **DEBUG** level, the Launcher Library will also log at **DEBUG** level.

For more information about logging levels in the BlackBerry Dynamics SDK, see the section titled "Log message categories" in [the Development Guide: BlackBerry Dynamics SDK](#) for Android or [Development Guide: BlackBerry Dynamics SDK](#) for iOS.

## About the BlackBerry Launcher Library

The *Launcher* is a feature of BlackBerry-based applications that is a convenient place for users to access different features, such as email, documents, the administrator-defined app store, other Good-based applications, and more.

Now with the BlackBerry Launcher library, you can enable your own Good-based applications to display the Launcher. The library consists of header files and a compiled library for Objective-C for iOS and Java for Android.

This paper describes how to programmatically integrate the Launcher Library in your applications.

**Note:** This paper does not discuss how to enable features so they appear in the BlackBerry Launcher on user devices. This is done in Good Control itself, not in your program. For details about enabling features for your users in the Launcher, including the see administrator-defined app store, see [BlackBerry Dynamics Device and Application Management](#).

## Prerequisite server setup by IT administrator for advanced features

To take advantage of advanced features that do not require programming, the Launcher and the Launcher Library depend on the following server setup by the enterprise IT administrator.

**Note:** You as a developer do not have control over these features directly (which do not require any programming by you) but need the enterprise IT administrator's involvement.

- Good Control for application management is for the administrator to define allowable enterprise-level applications that are displayed by the app store icon in the Launcher. For details on required setup, your administrator should see:
  - *BlackBerry Dynamics Server Installation* at <https://community.good.com/docs/DOC-1043>
  - *BlackBerry Dynamics Device and Application Management* at <https://community.good.com/docs/DOC-1043>
- If the IT enterprise administrator has installed BlackBerry Enterprise Mobility Server (BEMS), your application that integrates the Launcher Library will have the following advanced features *without any extra development work by you*: custom icons, user profile info, presence status, display of BlackBerry Dynamics applications' policy-enabled sections. The Launcher Library discovers other BlackBerry Dynamics apps if the apps are bound to at least one BlackBerry Dynamics service, for example `com.good.gd.launchable` service. For information about how to configure BEMS for using the Launcher, the administrator should see the [BEMS Installation and Configuration Guide for Administrators](#).

The Launcher Library does not strictly require the BlackBerry Enterprise Mobility Server. Without it your application still has the default BlackBerry icon and the IT administrator can add other application icons to the Launcher, but you will not have the features of policy-based sections in the Launcher (like docs or RSA SecurID), Presence, or users' profile photos.



## BlackBerry Dynamics SDK and the BlackBerry Launcher Library correspondence required

If you use the BlackBerry Launcher as a front-end to your applications, keep the information here in mind. If you do not use the BlackBerry Launcher library, this note is not relevant to you.

The BlackBerry Launcher Library works on top of the BlackBerry Dynamics SDK.

The BlackBerry Dynamics SDK and the BlackBerry Launcher Library are mutually dependent. Be sure you have installed the latest BlackBerry Dynamics SDK for your platform that corresponds to this release of the BlackBerry Launcher Library and have installed the version of the BlackBerry Launcher Library that corresponds to your installed version of the BlackBerry Dynamics SDK.

Specifically for this release BlackBerry Dynamics SDK version 2.1.xxx corresponds with BlackBerry Launcher Library version 2.3.0.xxx.

**Important:** If you do not ensure that you are using the corresponding versions of the BlackBerry Dynamics SDK and the BlackBerry Launcher Library, your applications will crash.

## About BlackBerry Dynamics software version numbers

The cover of this document shows the base or major version number of the product, but not the full, exact version number (which includes "point releases"), which can change over time while the major version number remains the same. The document, however, is always current with the latest release.

Product	Version
Good Control	4.0.57.102
Good Proxy	4.0.57.88
BlackBerry Dynamics SDK for Android	3.3.0.3073
BlackBerry Dynamics SDK for iOS	3.3.0.3249
BlackBerry Launcher Library for Android	2.6.1.201
BlackBerry Launcher Library for iOS	2.6.1.137

If in doubt about the exact version number of a product, check the BlackBerry Developer Network for the latest release.

## New in previous releases

### Customization of Launcher icon

As a developer, you should be aware that the IT administrator can customize the Launcher to display a corporate logo or other graphic.

## New in previous releases

This customization is not accomplished through programming. It is done via the BlackBerry Enterprise Mobility Server console; see this and other details in [Moving to production](#) .

## Integrating the Launcher Library for display on Android

The Launcher Library can be integrated like any Android library project. Once the library is integrated into the build system (Gradle/Ant) or into your favorite IDE, there are few basic steps you need to follow to integrate launcher functionality into your apps.

### Setup in Gradle and Android Studio for sample application

The sample application for the BlackBerry Launcher Library is based on the RSS Reader sample application delivered with the BlackBerry Dynamics SDK.

- Applications relying on the BlackBerry Launcher Library must use Android support library **com.android.support:support-v13**. This is already configured in the Android sample application's **build.gradle** file, but you must add these dependencies in any new project. See the details in the steps below.
- Be sure that you have updated your Android SDK to API level 23.

### Steps to configure the sample application

The sample application is fully configured except for adding the libraries from the BlackBerry Dynamics SDK for Android.

1. Download and unzip the BlackBerry Launcher Library.
2. Change directory to **GoodLauncherLibrary/SampleLauncherApp/extlibs**
3. From the BlackBerry Dynamics SDK for Android directory, copy the entire subdirectory named **sdk/libs/handheld/gd** to right here (**extlibs**) so that you have **.../SampleLauncherApp/extlibs/gd**
4. To build your project with Gradle, change directories to the top level of **SampleLauncherApp** use either **gradle --info --stacktrace clean assembleDebug** or **gradle --info --stacktrace clean assembleRelease**

### Steps to configure an existing Android project

Prerequisites: These steps assume you have a project already created and a **libs** directory in that project. The **libs** directory must not be at the top level of your project; it must be at the same level as your **build.gradle** file.

1. Download and unzip the BlackBerry Launcher Library.
2. Change directory to **GoodLauncherLibrary** and copy the file **launcherlib.aar** to the **libs** directory described in "Prerequisites" above.
3. Make the following declarations in your **build.gradle** file, as shown in these snippets:
  - Declare a dependency on the **launcherlib.aar** library:

```
dependencies {  
    compile(name:'launcherlib', ext:'aar')  
}
```

- Declare the **libs** directory as a repository:

```
repositories{
    flatDir {
        dirs 'libs'
    }
}
```

- Add the Android support library dependencies:

```
dependencies {
    compile 'com.android.support:support-v13:23.2.1'
    compile 'com.android.support:appcompat-v7:23.2.1'
    compile 'com.android.support:design:23.2.1'
}
```

4. From the BlackBerry Dynamics SDK for Android directory, copy the entire subdirectory named **sdk/libs/handheld/gd** to *your\_project\_directory/extlibs/* so that you have *your\_project\_directory/extlibs/gd*
5. To build your project with Gradle use either **gradle --info --stacktrace clean assembleDebug** or **gradle --info --stacktrace clean assembleRelease**

## Use global instance of GDStateListener, not per-activity listener

The Launcher Library and BlackBerry Dynamics SDK for Android does not have a mechanism to determine whether the global state listener is set in order to decide whether it should use its own **GDStateListener** in its activity. Therefore, BlackBerry recommends that you set a global instance of **GDStateListener**.

## Initialization, registration and program setup

**Note:** Applications relying on the BlackBerry Launcher Library must use Android support library **com.android.support:support-v13**.

1. Initialize Launcher Library.

Before any interaction with Launcher, it must be initialized:

```
LauncherButton.initForApplication(Application context, Collection<Class> activities,
    ActivitiesTargetingMethod method);
```

<b>context</b>	A context of the application where Launcher Library is integrated.
<b>activities</b>	A list of the application activities where the the Launcher button should or should not be shown. The third parameter, <b>ActivitiesTargetingMethod</b> , defines whether this is an excluded activities list or included activities list.
<b>method</b>	Can have one of two possible values: <ul style="list-style-type: none"> <li>• <b>LauncherButton.ActivitiesTargetingMethod.Exclusive</b></li> <li>• <b>LauncherButton.ActivitiesTargetingMethod.Inclusive</b></li> </ul>

## 2. Notify Launcher of BlackBerry Dynamics authorization state change

The BlackBerry Dynamics application hosting Launcher Library should let Launcher Library know the authorization state with the following method:

- Set to **true** when BlackBerry Dynamics state `EventListener` notifies **onAuthorized** state,
- Set to **false** when `EventListener` notifies **onLocked** state

## 3. Register `GDSERVICECLIENTLISTENER` if needed

The Launcher Library uses `GDSERVICECLIENT` to start applications listed on the Launcher pad.

The `GDSERVICECLIENT` requires an instance of `GDSERVICECLIENTLISTENER` but only one listener can be set. In case your application already defines `GDSERVICECLIENTLISTENER`, it should be passed to the Launcher Library. Otherwise, the Library's default `GDSERVICECLIENTLISTENER` is set to `GDSERVICECLIENT` and the application's `GDSERVICECLIENTLISTENER` is reset.

```
HostingApp.getInstance().setClientServiceListener(GDSERVICECLIENTLISTENER serviceClientListener);
```

## 4. Handle Launcher Settings button



The Launcher has a settings button

This button is intended to invoke your application settings screen. Your application must implement and then register the `LAUNCHERCOMMANDCALLBACK` interface.

```
HostingApp.getInstance().setOnCommandCallback(LauncherCommandCallback commandCallback)
public interface HostingApp.LauncherCommandCallback {
    public void onSettingsCommand();
}
```

## Proguard configuration

Applications integrating the Launcher Library must have the following directive in the proguard file:

```
-dontwarn com.good.launcher.**
```

Otherwise, the build might fail with too many warnings.

# Integrating the Launcher Library on iOS

The Launcher Library can be integrated into your projects just like any other library.

## Adding to Xcode project

Adding Launcher Library to an Xcode project is no different than adding other libraries, with a few minor additions.

1. Move the unzipped Launcher Library files to any desired directory and add each item to the Xcode project file.
2. Under the **Build Phases** of the application target, add **Launcher.framework** to the **Link Binary with Libraries** phase.
3. Verify that the **LauncherBundle.bundle** is added as part of the **Copy Bundle Resources** phase.
4. Ensure that the Main Interface setting for both iPhone and iPad under the target's General settings is cleared; Launcher provides its own root interface.
5. To ensure icons are downloaded, in the app's info.plist file add the following key/value pair:  
**(string: boolean): GDFetchResources : YES**
6. Under the **Other Linker Flags** build setting, add the following: **-ObjC**

## Initializing

The Launcher should be initialized in the `-[id <UIApplicationDelegate>`

`application:didFinishLaunchingWithOptions:]` method. The following is extracted from the sample applications `AppDelegate.m`:

```
//Set the main window
[self setWindow:[GDiOS sharedInstance] getWindow];
```

```
//Set a GDiOS instance and set the delegate
[self setGood:[GDiOS sharedInstance]];
[_good setDelegate:self];
```

```
//Set a flag so that view controllers can check the status
started = NO;
```

```
//Show the Good Authentication UI
[_good authorize];
```

```
//See 'On Authorized' for example app progression in Example of onAuthorized with startServicesWithOptions
return YES;
```

You must also add the following line at the end of your initialization code, as shown in [Example of onAuthorized with startServicesWithOptions](#) :

```
[launchPadVC startServicesWithOptions:GTLInternalGDAuthTokenAndPushConnectionManagement];
```

## Starting

After the GDiOS delegate (usually also the **UIApplicationDelegate**) receives the **onAuthorized:** callback, the Launcher Library needs to start up various services that call into the BlackBerry Dynamics API. Starting launcher requires that startup options be provided. Internally, the library uses the **GDUtility** and **GDPushConnection** classes. These classes are singleton classes with a single delegate. If the host app is using these classes, Launcher Library needs to either retrieve the required information through a delegate callback or through method calls into Launcher.

### Startup options

<b>GTLInternalGDAuthTokenAndPushConnectionManagement</b>	Default. Use this if the host app does not use BlackBerry Dynamics auth tokens ( <b>GDUtility</b> class) or BlackBerry Dynamics push connection. The specified classes are used internally in the Launcher.
<b>GTLHostGDAuthTokenManagement</b>	If the host app uses the <b>GDUtility</b> class internally, use this option. The Launcher makes a delegate callback into the app whenever it requires an auth token (see below).
<b>GTLHostGDPushConnectionManagement</b>	If the host app uses <b>GDPushConnection</b> and specifies a delegate, use this option. Whenever the host app receives the <b>onStatus:</b> call, the value provided should be passed to the Launcher.

## Example of onAuthorized with startServicesWithOptions

The following fragment is extracted from the sample applications **AppDelegate.m**.

**Important:** Be sure to add the following line at the end of your initialization code, as highlighted in below. Otherwise, your icons might not be displayed properly.

```
[launchPadVC startServicesWithOptions:GTLInternalGDAuthTokenAndPushConnectionManagement];
```

AppDelegate.m with **launchPadVC**

```
-(void) onAuthorized:(GDApEvent*)anEvent
{
    /* Handle the Good Libraries authorized event. */

    switch (anEvent.code) {
        case GDErrorNone: {
            if (!started) {
                // launch application UI here
                started = YES;

                //Detect the device
                if ([[UIDevice currentDevice] userInterfaceIdiom] == UIUserInterfaceIdiomPhone)
```

## Integrating the Launcher Library on iOS

```
{
    /*
     * iPhone start - based on single UINavigationController
     */

    //Setup the root view controller
    RootViewController *rootViewController = [[RootViewController alloc]
    initWithNibName:@"RootViewController_iPhone" bundle:nil];

    //Start up the root navigation controller with the root view controller
    _navController = [[UINavigationController alloc] initWithRootViewController:rootViewController];
    GTLauncherViewController *launchPadVC = [[GTLauncherViewController alloc]
    initWithBaseViewController:_navController];
    [self.window setRootViewController:launchPadVC];

    [launchPadVC
    startServicesWithOptions:GTLInternalGDAuthTokenAndPushConne
    ctionManagement];
}
else
{
    .
    .
    .
}
```

GTLHostGDAuthTokenManagement startup option

```
//This method must be implemented if GTLHostGDAuthTokenManagement is a specified startup option
- (void)launcherViewController:(GTLauncherViewController *)controller didRequestGDAuthTokenForServerName:
(NSString *)name completion:
(void (^)(NSString *token))completion { //Implementation specific to the host app
    [[HostAuthTokenManager tokenManager] getAuthTokenForServer:name completion:^(NSString *token) {
        if (completion) {
            completion(token);
        }
    }];
}
```

GTLHostGDPushConnectionManagement startup option

```
//GDPushConnectionDelegate callback
- (void)onStatus:(int)status
{
    .
    .
    .
    //The following is required if GTLHostGDPushConnectionManagement is specified as a startup option
    GTLauncherViewController *launcherVC = ...;
    [launcherVC setGDPushConnectionStatus:status];
}
```



## Sample application for Android and iOS

The distribution comes with a sample RSS Reader application that you can examine to see how the BlackBerry Dynamics Launcher Library is implemented.

## Moving to production

### Service binding

When your application is ready, it needs to be bound to a service in Good Control. You can bind it to the Launcher Library service itself.

For details about how to bind an application to a service, see "Binding a Service Version to an Entitlement Version" in the [Good Control Online Help](#).

### Customizing your app's icon: required for iOS

You can set a custom icon for your application in Good Control. See "Editing application details" in the [Good Control Online Help](#).

### Customizing the Launcher's logo in BEMS

If you want a custom logo for the Launcher itself to replace the default "BlackBerry" logo, you can configure one in BEMS.

For more information about setting a customized icon for BlackBerry Dynamics Launcher, see the [BEMS Installation and Configuration content](#).

## Validating the BlackBerry Enterprise Mobility Server SSL certificate

You can have the Launcher Library validate the SSL certificate of any BlackBerry Enterprise Mobility Server server associated with your deployment to secure the communications between the servers.

### Prerequisites

- A copy of the BlackBerry Enterprise Mobility Server server's SSL certificate
- Administrative access to your Good Control server
- The entitlement ID: com.blackberry.feature.validatebemscertificate

### General steps

1. On the BlackBerry Enterprise Mobility Server server, export the SSL certificate. For details, see the [BEMS Installation and Configuration content](#).
2. In your Good Control server, import the BEMS certificate with the menu item **Certificates > Trusted Authorities** tab, **Upload New Certificate** button. See the details in the [Good Control Online Help](#) topic "Certificates > Trusted authorities".

## Moving to production

3. In your Good Control server, entitle your end-users to the entitlement ID `com.blackberry.feature.validatebemscertificate`. You have several optional ways to do this:
  - Easiest way is to entitle the Everyone group
  - You can create a special-purpose group of only those users you want to entitle.
  - You can entitle individual end-users.

All these options are discussed in the [Good Control Online Help](#) topic "Entitling end-users to applications".

# BlackBerry Dynamics documentation

Category	Title	Description
Cross-platform	<ul style="list-style-type: none"> <li><a href="#">Getting Started Guide for Marketplace Partners</a></li> <li><a href="#">Good Control/Good Proxy Platform Overview for Administrators and Developers</a></li> <li><a href="#">Good Cloud Deployment</a></li> </ul>	Overviews of the BlackBerry Dynamics system
	<ul style="list-style-type: none"> <li><a href="#">Good Control Device and Application Management</a></li> <li><a href="#">Good Control Device Management Enrollment: Good Agent for iOS</a></li> <li><a href="#">Good Control Device Management Enrollment: Good Agent for Android</a></li> </ul>	Device and application management on Good Control, including app distribution, with client-side device enrollment details
Security	<a href="#">BlackBerry Dynamics Security White Paper</a>	Description of the security aspects of BlackBerry Dynamics
	<a href="#">BlackBerry Dynamics and Fingerprint Authentication</a>	Discussion of the implementation of BlackBerry security with fingerprint recognition systems: Apple Touch ID and Android Fingerprint
BlackBerry UEM	<a href="#">BlackBerry UEM Administration Guide</a>	Approaches to administering the BlackBerry Unified Endpoint Manager
	<a href="#">Getting Started with BlackBerry UEM and BlackBerry Dynamics</a>	Introductory material to administering the BlackBerry UEM with the BlackBerry Dynamics profile
Good Control	<ul style="list-style-type: none"> <li><a href="#">BlackBerry Secure Enterprise Planning Guide</a></li> <li><a href="#">BlackBerry Secure Servers Compatibility Matrix</a></li> <li><a href="#">BlackBerry Performance Calculator</a></li> </ul>	Guidelines and tools for planning your BlackBerry Secure Enterprise deployment
	<a href="#">Good Control/Good Proxy Server Preinstallation Checklist</a>	Same checklist extracted from the installation guide below
	<a href="#">Good Control/Good Proxy Server</a>	Details on installing Good Control, Good Proxy, and the GC

Category	Title	Description
	<a href="#"><i>Installation</i></a>	database
	<a href="#"><i>Kerberos Constrained Delegation for Good Control</i></a>	Configuration details for integrating the Kerberos authentication system with BlackBerry Dynamics
	<a href="#"><i>Direct Connect for Good Control</i></a>	Configuring BlackBerry Dynamics servers to securely access internal resources from the external Internet
	<a href="#"><i>Good Control Easy Activation Overview</i></a>	A look at the Easy Activation feature
	<a href="#"><i>Good Control/Good Proxy Server Backup and Restore</i></a>	Minimal steps for backing up and restoring the BlackBerry Dynamics system
	<a href="#"><i>Good Control Online Help</i></a> <a href="#"><i>Good Control Server Property and Security Policy Reference</i></a>	Printable copy of the GC console online help
	<a href="#"><i>PKI Cert Creation via Good Control: Reference Implementation</i></a>	A reference implementation in Java for creating end-user PKI certificates via Good Control and a Certificate Authority (CA)
	<a href="#"><i>Good Control Cloud Online Help</i></a>	Printable copy of the Cloud GC console online help
	<a href="#"><i>Technical Brief: BlackBerry Dynamics Application Policies</i></a> with XSD for app-policy XML	Description of formatting application policies for use in Good Control, with examples.
	<a href="#"><i>Development Guide: Good Control Web Services</i></a>	Programmatic interfaces on Good Control <ul style="list-style-type: none"> <li>• Basic control and application management: SOAP over HTTPS. Documentation is in the WSDL files included with GC.</li> <li>• Device management: HTTP API (with JSON) for device management. <a href="#">Zipfile of API reference.</a></li> </ul>
Software Development	<a href="#"><i>Developer Bootstrap: Good Control Essentials</i></a>	Bare minimum of information that a developer of BlackBerry Dynamics applications needs to get started with the Good Control server to test applications.
	<a href="#"><i>BlackBerry Dynamics Shared Services Framework</i></a>	Description of the BlackBerry Dynamics shared services framework for software developers
Android	<ul style="list-style-type: none"> <li>• <a href="#"><i>Development Guide: BlackBerry Dynamics SDK for Android</i></a></li> <li>• <a href="#"><i>API Reference for Android</i></a></li> </ul>	Working with the BlackBerry Dynamics SDK for Android and the essential reference for developers
iOS	<ul style="list-style-type: none"> <li>• <a href="#"><i>Development Guide: BlackBerry Dynamics SDK for iOS</i></a></li> </ul>	Working with the BlackBerry Dynamics SDK for iOS and the essential reference for developers

Category	Title	Description
	<ul style="list-style-type: none"> <li>• <a href="#">API Reference for iOS</a></li> </ul>	
macOS	<ul style="list-style-type: none"> <li>• <a href="#">Development Guide: BlackBerry Dynamics SDK for macOS</a></li> <li>• <a href="#">API Reference for macOS</a></li> </ul>	Working with the BlackBerry Dynamics SDK for macOS and the essential reference for developers
Windows	<ul style="list-style-type: none"> <li>• <a href="#">Development Guide: BlackBerry Dynamics SDK for Universal Windows Platform (UWP)</a></li> <li>• <a href="#">API Reference for UWP</a></li> </ul>	Working with the BlackBerry Dynamics SDK for Universal Windows Platform (UWP) and the essential reference for developers
iOS, Android	<a href="#">BlackBerry Launcher Library</a>	Source code and header files for implementing the popular BlackBerry Launcher interface
Cross-platform	<a href="#">Development Guide: BlackBerry Dynamics SDK for Cordova for iOS and Android</a>	Working with the BlackBerry Dynamics SDK for Cordova plugins
	<a href="#">BlackBerry Dynamics Bindings for Xamarin.Android</a>	Working with the BlackBerry Dynamics SDK and the Xamarin cross-platform integrated development environment
	<a href="#">BlackBerry Dynamics Bindings for Xamarin.iOS</a>	Working with the BlackBerry Dynamics SDK and the Xamarin cross-platform integrated development environment