Development Guide

BlackBerry Dynamics SDK for Windows 10



Published: 2017-09-01 SWD-20171204142659563

Contents

About this guide	
BlackBerry Dynamics background	6
BlackBerry Dynamics API reference	
Easy Activation	6
Securing cut-copy-paste on devices (Data Leakage Prevention, or DLP)	6
Shared Services Framework	7
Support for client certificates	7
Support for "No Password" security policy	7
BlackBerry Dynamics features currently not or partially supported by BlackBerry Dynamics SDK for Windows 10	8
Older BlackBerry Dynamics SDK for Windows 8.1 no longer supported	8
Locales	9
Requirements	10
BlackBerry Dynamics software versions	10
Software requirements	10
Supported programming languages: C#	10
BlackBerry Dynamics entitlement ID and version.	10
Distinction from and use with native language identifiers	12
Required build-time declarations: URL type	13
Steps to get started with the BlackBerry Dynamics SDK	14
Getting started step-by-step	14
Install the BlackBerry Dynamics SDK for Windows 10	14
Video tutorials for BlackBerry Dynamics SDK for UWP	15
Sample apps	16
Testing and troubleshooting	18
Logging and diagnostics	
Log message categories	
Readying your app for deployment: server setup	19
Details of support for client certificates	20

BlackBerry Dynamics SDK support for personal certificates (PKCS12 or PKI certs)	20
Certificate requirements and troubleshooting	21
Client certificate sharing among BlackBerry Dynamics-based applications	22
Kerberos PKINIT: User authentication with PKI certificates	22

About this guide

1

This guide is an introduction and overview to the BlackBerry Dynamics software development kit (SDK) for Windows 10 operating system, with a focus on installing the SDK, and introducing the sample applications that are packaged with the SDK.

The guide is for software developers who already have an understanding of developing software for the Windows platform. It is not a basic tutorial.

BlackBerry Dynamics background

2

The following sections provide some background information that can help you understand the features of the BlackBerry Dynamics SDK.

The way that these features are implemented in your environment will depend on how your administrator has configured your organization's servers, your network, and other infrastructure.

BlackBerry Dynamics API reference

The BlackBerry Dynamics SDK API reference describes the available interfaces, classes, methods, and much more.

You can access the API reference at https://developer.blackberry.com/files/BlackBerry-Dynamics-for-MS-Windows-UWP/html/ N GD.htm.

Easy Activation

The Easy Activation feature simplifies the provisioning process by allowing a BlackBerry Dynamics app to hand off activation to an app that is already installed on the device and can act as the activation delegate. The user has to retrieve and manually enter an access key only the first time they install a BlackBerry Dynamics app.

For more information about enabling Easy Activation, see the Easy Activation Feature Overview.

Securing cut-copy-paste on devices (Data Leakage Prevention, or DLP)

You can use the BlackBerry Dynamics SDK to protect certain data copied and pasted between apps on your users' devices.

Server administrators must enable the **Data Leakage Prevention** policies in the management console.

To enable sharing among a group of apps, the apps must be provisioned from the same BlackBerry Control service for each user.

If the **Data Leakage Prevention** settings are enabled in your environment, you can work around them when you need to debug your app. For more information, see https://community.blackberry.com/view-doc.jspa? fileName=screencapturecontrol.html&docType=android.

Shared Services Framework

BlackBerry Dynamics-enabled apps can communicate with each other using the Shared Services Framework. There are two kinds of shared services:

- Server-side services
- Client-side services

The BlackBerry Dynamics SDK contains sample apps that show how these services work.

For a conceptual background, see BlackBerry Dynamics Services Framework.

Support for client certificates

BlackBerry Dynamics supports many popular uses of client-side Public Key Infrastructure (PKI) certificates to secure apps and communications:

- General requirements for working with PKI certs
- Description of client certificate sharing among BlackBerry Dynamics apps on a device
- Kerberos PKINIT: client certificates in the Kerberos authentication model. (This is not Kerberos Constrained Delegation, or KCD).

Support for "No Password" security policy

The BlackBerry Dynamics Runtime now supports the security policy for not requiring the end user to set an app password.

Security consideration

- Consider the security ramifications in your environment carefully. Enabling "no password" is only one of the design options available. Others include authentication delegation, enabling "no password" on certain devices only, device management, and more. You should consider enabling "no password" for select groups of users whose devices are under tight control via device management profiles or other controls.
- Do not enable "no password" and authentication delegation in the same policy set.
- Enabling "no password" does not permit authentication in the background, because there is no authentication.

Software prerequisites

1. Standalone Good Control 3.0.50.70 or later or BlackBerry UEM 12.7 or later

2. The "No Password" policy must be enabled and applied to the desired users or app group.

User-visible changes and behavior

With a BlackBerry Dynamics app that is protected by a security policy that requires a password, if the administrator changes the security policy to "Do not require a user password":

- The user is shown an informational screen stating that a password is no longer required for the app.
- The user is then in "No Password mode" and is never prompted for password again.

If the user is in "No Password mode" and the administrator changes the security policy to require a password:

- The user is prompted to set a password.
- The user is shown an informational screen stating that a password is now required for the app.

BlackBerry Dynamics features currently not or partially supported by BlackBerry Dynamics SDK for Windows 10

The BlackBerry Dynamics SDK for Windows 10 does not currently support to the following features of BlackBerry Dynamics:

- Application management
- Custom, or third-party, keyboards
- FIPS compliance

Older BlackBerry Dynamics SDK for Windows 8.1 no longer supported

BlackBerry Dynamics SDK for Windows 10is the preferred development kit for all Good-based Windows applications and is the focus of future enhancements. The older BlackBerry Dynamics SDK for Windows 8.1 is primarily for Windows Phone apps. If you are targeting Windows 8.1 exclusively, you can still use the BlackBerry Dynamics SDK for Windows 8.1, but it is better to use the BlackBerry Dynamics SDK for Windows 10.

Locales

The BlackBerry Dynamics SDK supports the following languages. No SDK calls are required to use a particular language; the interface selects the appropriate language based on the language setting the user has configured on their device.

- English (US)
- Dutch
- French
- German
- Italian
- Japanese
- Spanish
- Swedish

BlackBerry Dynamics software versions

BlackBerry Dynamics SDK for Windows 10 3.2.1.2607

Software requirements

Windows development

Item	Requirement
Recommended setup	Windows 10Microsoft Visual Studio 2015
Minimum for building	Windows 8.1USB-connected phone for debugging

Supported programming languages: C#

BlackBerry Dynamics SDK for Windows 10 supports the following programming languages:

C#

BlackBerry Dynamics entitlement ID and version

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 for your apps, as well as for publishing and service provider registration. The BlackBerry Dynamics entitlement ID was formerly known as the app ID or GD App ID.

The entitlement ID is used in the app, in the BlackBerry UEM or standalone Good Control management console for app management, and in some administrative user interfaces on the application developer portal.

Note: The entitlement ID and entitlement version are different from the native application ID and native application version. The native application ID is a unique identifier for the app that is used by the OS and associated platforms (for example, the package name for Android or bundle identifier for iOS). The native application version is the app version number that you must change if you want to distribute a new version of an 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. For more information about when to change the entitlement ID and entitlement version, see the BlackBerry Dynamics API reference.

Requirements for the entitlement ID and entitlement version

Requirement	Description
Required for apps	You must define both the entitlement ID and the entitlement version for all your BlackBerry Dynamics apps, regardless of whether you use the BlackBerry Dynamics Shared Services Framework. Developers and administrators should ensure that the value specified for the GDappVersion key in the app configuration files is the same as the value the administrator specifies in BlackBerry UEM or in standalone Good Control. The entitlement version is independent of any native version identifier. For more information, see Distinction from and use with native language identifiers.
Represent the same app across all platforms	The same entitlement ID must be used to represent the app across all platforms. By default, access to apps varies by the type of app:
	By default, all versions of partner or ISV apps are available to all authorized users in any organization that the app has been published to.
	By default, each version of a BlackBerry Dynamics app requires that the administrator grant access in BlackBerry UEM or in standalone Good Control before users can run the app on users' devices.
Naming scheme	Develop a naming scheme to meet your needs. For example:
	Entitlement ID: com.manufacturingco.gd
	Entitlement version: 1.0.0.0Native application version: 2.0
Entitlement ID format	The general form of an entitlement ID is <company_name>. <app_name>.</app_name></company_name>
	The ID must use reverse domain name form, for example, com.company.example. Use a domain name owned by your organization.
	The ID must not begin with com.blackberry, com.good, com.rim, or net.rim.

Requirement	Description
	 The ID can contain only lower-case letters, numeric digits, hyphens, and periods.
	 The string must follow the <subdomain> format defined in section 2.3.1 of RFC 1035, as amended by Section 2.1 of RFC RFC 1123.</subdomain>
Entitlement version value	 The value must use one to four segments of digits, separate by periods (x.x.x.x). Each segment can be up to three digits and must not use a leading zero (for example, 01.02 is not valid). A segment can use a single 0.
	The first release of an app should use the entitlement version 1.0.0.0.

Distinction from and use with native language identifiers

The Entitlement ID and Entitlement Version are BlackBerry Dynamics specific metadata and are independent of the identifiers needed by the app platforms themselves. The key point is that the values and the native language identifiers' values can be the same but they do not necessarily have to be. Listed below by platform are the equivalent native identifiers, which are where the values of Entitlement ID and version are stored.

Package.appxmanifest

ForWindows 10, the SDK relies on Package Family Name, which is not explicitly set but is generated by Visual Studio and is displayed in the GUI editor of the package manifest, as shown below.

Unique native identifiers for enterprise apps

If you are developing a private app for use in your enterprise, make sure that the value you choose for the app's native identifiers (Bundle ID and others constructs used on other platforms) is unique, especially with respect to apps that are available through the public app stores.

Duplicate native identifiers can prevent the proper installation or upgrade of your own app.

For all your native identifiers, devise a naming scheme that you can be relatively certain is unique.

Mapping BlackBerry Dynamics entitlement ID to native identifiers

To take advantage of many features, such as Easy Activation, multi-authentication delegation, and the hared services framework, developers need to set up a map in the server between your defined Entitlement ID and the native identifiers on the platforms for which your app is distributed. The native platforms have no knowledge of the Entitlement ID; thus the mapping is needed for the operating systems to take over the actual function of the app.

Native version identifiers: * wildcard allowed for blocking app

The SDK supports use of native version identifiers in keeping with the conventions described by the major vendors. These same conventions apply to the use of the * wildcard in the server to deny apps by native version.

- Platform: Windows 10 / Package / Identity / @Version
- A string in quad notation, "Major.Minor.Build.Revision"
- More information from Microsoft

The * character can be used in native version identifiers, but must always be preceded by a period (.) and must be the last character in the native version string. Examples:

Allowed: 2.3.*

Not allowed: 2.*.3

2.* includes 2.*.*

Required build-time declarations: URL type

Your app must declare a URL type so that it can be discovered by other apps on the same user's device.

This enables BlackBerry Dynamics AppKinetics (shared services) communication, which is required for many BlackBerry Dynamics features. In other words, this scheme is used to discover applications on a device. The URL type and schemes are declared in Package.appxmanifest.

Table 1: URL type discovery schemes

Format	Description	Example
gd-sc3	Core BlackBerry Dynamics communications.	Exactly as shown
<pre>gd-sc3.GD_application_ version_number</pre>	Required only if your app provides a service that must be discoverable.	<pre>com.example.gd.myapp.sc3.1. 0.0.0</pre>
gd-sc3.certificate.sharing	For sharing of SSL certificates among BlackBerry Dynamics apps on the same device	Exactly as shown

Steps to get started with the BlackBerry Dynamics SDK

4

- Download and install the BlackBerry Dynamics SDK. For information, see Install the BlackBerry Dynamics SDK for Windows 10.
- 2. Familiarize yourself with the features of the BlackBerry Dynamics SDK.
 - For general information, see BlackBerry Dynamics background.
 - To view code samples, see Sample apps.
- 3. Understand the Requirements and possible constraints on your programming.
- 4. Become familiar with available classes and methods in the BlackBerry Dynamics API Reference.
- 5. You might want to view Video tutorials for BlackBerry Dynamics SDK for UWP
- 6. Build your app.
- 7. Test your app. For information, see Testing and troubleshooting.
- 8. To help you troubleshoot issues, you can set up logging and diagnostics. For more information, see Logging and diagnostics.
- 9. Deploy your app. For options, see Ready your application for deployment.

Getting started step-by-step

When you start developing your app, you might want to follow the steps described in the Getting Started with the BlackBerry Dynamics SDK flow. This is a step-by-step walkthrough of the various stages of developing an app for your platform.

Install the BlackBerry Dynamics SDK for Windows 10

The BlackBerry Dynamics SDK for UWP is distributed as NuGet package (.nupkg extension and format).

Before you begin: All software described in Software requirements is ready.

- 1. Download the NuGet file from https://community.blackberry.com/community/gdn/resources and save it to any location on your Microsoft Windows system, which we refer to below as *nupkg_save_location_directoryname*.
- 2. In Visual Studio, either create a new C# Windows Universal project or open an existing project.

- 3. Open the Package Manager Console.
- 4. Set the default project dropdown to the *your_project_name* project.
- 5. Type the following command: Install-Package GD.UWP -Source nupkg_save_location_directoryname
 You can now build your project.

Video tutorials for BlackBerry Dynamics SDK for UWP

Several video tutorials are available for working with the BlackBerry Dynamics SDK for Universal Windows Platform (UWP).

Topic	Link
Create a new app	https://community.good.com/videos/1984
Port an existing app	https://community.good.com/videos/1985
Use app-based services	https://community.good.com/videos/1987

You can download the sample apps from https://community.blackberry.com/community/gdn/resources/samples. After you download the desired samples and open the project solution in Visual Studio, download and install the NuGET package for the BlackBerry Dynamics SDK for Windows 10 (see Install the BlackBerry Dynamics SDK for Windows 10).

Sample app	Description
App configuration	Demonstrates how to collect BlackBerry Dynamics Runtime configuration information, such as the settings for secure cut-copy-paste, current app version, current user ID, and so on.
AppKinetics service and client	Provide an example of Application-Based Services that allows for the editing and exchange of files between the AppKinetics client and server. For more information, see Shared Services Framework.
App-specific policies	Demonstrates how to use BlackBerry Dynamics application-specific policies.
Change password	Demonstrates how to invoke the password change process.
Secure copy-cut-paste	Provides an example of how the TextBox and RichEditBox controls work for the Secure Cut-Copy-Paste policy, which is discussed in Securing cut-copy-paste on devices (Data Leakage Prevention, or DLP).
Services framework	Provides an example of Server-Side and Local Services Discovery based on the Shared Services Framework.
Upload logs	Provides an example of uploading application logs to the management console.
User entitlements	Demonstrates how to generate an authentication token based on challenge and server address.
Save logs	Demonstrates how to securely save log files in the local file system.
Secure HTTP	Provides an example of how to use GDHttpRequestFilter: an HTTP request to specified URIs using the BlackBerry Dynamics infrastructure, with different options for authentication.
Secure sockets	Provides an example of how to use BlackBerry Dynamics sockets to connect to specific URIs.
Secure file system	Demonstrates BlackBerry Dynamics secure file system operations, including file and folder operations, and importing files to the secure container.

Sample app	Description
Secure storage	Demonstrates BlackBerry Dynamics secure database access, including creating a database and making SQL queries.
UI customization	Demonstrates how to customize the appearance of the BlackBerry Dynamics user interface.

Testing and troubleshooting

6

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 might be asked to provide the log file to the BlackBerry Dynamics technical support team, but this is typically only necessary for complex support issues.

Log message categories

Messages in the activity log are assigned to one of four categories:

- Errors: critical failures
- Warnings: failures that arise but from which the BlackBerry Dynamics Runtime has recovered
- Info: normal operational activity
- Detailed: additional diagnostic information used for troubleshooting complex problems

You can configure the logging system to filter some or all messages. By default, only messages that belong to the Errors, Warnings, and Info categories are printed.

The logging locations on the IDE console and device container are configured differently and independently. In principle, you control the console log and the BlackBerry UEM administrator controls the container log.

Readying your app for deployment: server setup

7

You want to test your app on a BlackBerry server before you deploy it into production. You need to become familiar with how to setup and configure such a server. You have options available: either BlackBerry UEM or the older Good Control.

Check with your IT or other department to see what test servers might already be available in your organization.

BlackBerry UEM preferred

BlackBerry UEM is the primary server configuration to test and deploy your app.

If you upgrade from Good Control to BlackBerry UEM, you not only get to use the great feature set that Good Control provides but you also get to take advantage of an enhanced feature set such as:

- Support for more policies for operating systems
- Better app management
- More container types
- Improved administration and provisioning
- Advanced connectivity and networking
- Expanded compliance and integrity checking
- Additional email, content, location, and certificate features
- Access to BlackBerry Web Services APIs

For information on how to use BlackBerry UEM to manage BlackBerry Dynamics apps, see Getting Started with BlackBerry UEM and BlackBerry Dynamics. For more information on the benefits of using BlackBerry UEM, see Benefits of upgrading from Good Control to BlackBerry UEM.

Older Good Control

The older Good Control server is also available, but BlackBerry encourages you to use BlackBerry UEM for your tests and deployment. For information about getting started with Good Control, see Developer Bootstrap: Good Control Essentials.

Details of support for client certificates

8

BlackBerry Dynamics SDK support for personal certificates (PKCS12 or PKI certs)

The BlackBerry Dynamics SDK has been enhanced to support personal certificates for authentication of applications at runtime.

No programming is required by the BlackBerry developer on any of the client BlackBerry Dynamics SDK platforms to take advantage of this feature. All operations are carried about by the BlackBerry Dynamics Runtime. The app must use the BlackBerry Dynamics Secure Communication Networking APIs provided in prior releases, the employee's account must be correctly configured, and the GC must be the 2.0.xx.yy release later.

An enterprise can deploy corporate services requiring two-way SSL/TLS mutual authentication in order to authenticate their employees. Through the enterprise, the employee may be issued or otherwise obtain a password protected Personal Information Exchange file (PKCS12/p12/pfx) containing a SSL/TLS client certificate and private key required by such services for authentication purposes. This file may be installed on various machines and devices, including BlackBerry Dynamics apps, so that access can be granted to these services.

Setup in Good Control

Requirements of the certificates themselves are described in Certificate requirements and troubleshooting.

To deploy Personal Information Exchange files with BlackBerry Dynamics apps, the following steps must be taken to configure the GC and employee's account. These generalized steps are extracted from the Good Control online help topic "PKCS12 Certificate Management for Email and Client Authentication", which has the exact steps.

- After the GC is installed, an administrator may choose to extend the default 24-hour period that an employee's protected Personal Information Exchange file shall be cached by the GC server.
- An administrator must add all BlackBerry Dynamics apps that access services requiring client authentication to the Certificates -> App Usage tab,
- An administrator must enable Use PKCS12 Certificate Management in the employee's security policy,
- An administrator or employee must upload their Personal Information Exchange files to the **Certificates** tab.

Behavior of personal certificates in the app

After the employee activates a BlackBerry Dynamics app enabled for access to server resources requiring client authentication, it receives their Personal Information Exchange files, provided they are still cached on the GC. For each file, the employee is asked to enter their password protecting the file contents, so the identification material can be installed. Once installed, provided the identification is correct, the BlackBerry Dynamics app is granted access to server resources requiring two-way SSL/TLS mutual authentication when connecting.

If there is more than one Personal Information Exchange file required per employee, the BlackBerry Dynamics Runtime ensures that the certificate chosen to send to the server meets all of the following criteria:

- 1. Only client certificates suitable for SSL/TLS client authentication are eligible for sending to the server. That is, certificates that have no Key Usage and Extended Key Usage, or Key Usage contains "Digital Signature" or "Key Agreement", or Extended Key Usage contain "TLS Web Client Authentication", and those whose Key Usages and Extended Key Usages do not contradict allowances for SSL/TLS client auth.
- 2. If the server advertises the client certificate authority in the SSL/TLS handshake, only client certificates issued by these authorities will be considered
- Only current client certificates will be considered (that is, certificates that have not expired or are not yet valid)

Usually this is sufficient to identify the correct client certificate, but if there is still more than one certificate meeting all of the above criteria then the first one is used. If the certificate chosen is not the desired one, the administrator or employee can manage this by removing the undesired client certificate from Good Control. The administrator can also increase the chance of success by ensuring the server is configured to advertise the client certificate authority in the SSL/TLS handshake.

Certificate requirements and troubleshooting

Make sure your certificates conform to these requirements:

- Certificates must be in PKCS 12 format: Certificate Authority (CA), public key, and private key, all in the same file.
- The PKCS12 file must end with the extension .p12 or .pfx.
- The PKCS 12 file must be password-protected.

There are many sources of certificates:

- Your own internal certification authority (CA)
- A well-known public CA
- Tools from the Internet, such as OpenSSL's keytool command. For example, the following is sufficient to generate a
 PKCS 12 certificate that is usable with Good Control; substitute your own values for alias the keystore name and the
 keystore password. If in doubt consult information on the Internet about all the possible options on the keytool
 command:

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

Beware of weak ciphers from export

Personal Information Exchange files are encrypted, and therefore must be encrypted with FIPS-strength ciphers if to be used when FIPS is enabled on the employee's security policy.

For their own maximum interoperability with other systems, it is common for third-party applications, for example the macOS keychain, to export identity material (credentials) using weak ciphers.

The administrator or employee can use a tool such as the OpenSSL command line to re-encrypt the file with a FIPS-strength cipher like so, which re-encrypts with the AES-128-CBC cipher:

Client certificate sharing among BlackBerry Dynamics-based applications

The BlackBerry Dynamics SDK supports the "sharing" of a single client certificate among all BlackBerry Dynamics-based applications for an end-user.

If the security policy for authentication via client certificates is enabled in Good Control or UEM and one or more client certificates have been uploaded to the server, those certificates are used for user authentication by all BlackBerry Dynamics-based applications on the user's device.

- No programming is required.
- Client certificates must be enabled in Good Control or UEM and at least one PKCS 12 certificate for a user must be uploaded to server.

Kerberos PKINIT: User authentication with PKI certificates

The BlackBerry Dynamics SDK supports Kerberos PKINIT for user authentication using PKI certificates.

No programming is required to use Kerberos PKINIT.

Important: Kerberos PKINIT is distinct from Kerberos Constrained Delegation (KCD). PKINIT relies on the Key Distribution Center (KCD), which should not be confused with "KCD".

Kerberos PKINIT	Kerberos Constrained Delegation
Kerberos PKINIT authentication is between the BlackBerry Dynamics app and the Windows Key Distribution Center (KDC), which communicate directly, and user authentication is based on certificates issued by Microsoft Active Directory Certificate Services.	Note: For PKINIT, Kerberos Constrained Delegation must not be enabled. If Kerberos Constrained Delegation has been configured, a BlackBerry Dynamics app does not use Kerberos PKINIT to access the defined KCD realms. Instead, when Kerberos Constrained Delegation is used, a trust relation has been previously established between BlackBerry Control and the Key Distribution Center, and BlackBerry Control communicates with the service on behalf of the app. Kerberos
	Constrained Delegation takes precedence over Kerberos

PKINIT, even if the user has a valid certificate.

Key requirements for PKINIT

Organizations that want to use Kerberos for BlackBerry Dynamics apps must make sure the following requirements are met.

Servers

- Kerberos Constrained Delegation must not be enabled.
- Windows Key Distribution Center (KDC) services for KDC server certificates issued by a Certificate Authority (CA) via the Active Directory Certificate Services must come only from the following Windows Server versions. No other server versions are supported.
 - Internet Information Server with Windows Server 2008 R2
 - Internet Information Server with Windows Server 2012 R2
- In BlackBerry Control:
 - The KDC hosts must be in the Allowed Domains of the Connectivity Profile applied to the affected users' policy sets.
 - Valid KDC service certificates must be located either in the BlackBerry Dynamics Certificate Store or the Device Certificate Store.

Client certificates

- The minimum keylength for the certificates must be 2,048 bytes.
- Client certificates must include the User Principal Name (UPN) (for example, user@domain.com) in the Subject Alternative Name (SAN) of object ID (OID) szOID_NT_PRINCIPAL_NAME 1.3.6.1.4.1.311.20.2.3, as specified by Microsoft at https://support.microsoft.com/en-us/kb/287547.
- The domain of the UPN must match the name of the realm of the Windows Key Distribution Center (KDC) service.

- The Extended Key Usage (EKU) property of the certificate must be Microsoft Smart Card logon (1.3.6.1.4.1.311.20.2.2). For information about certificate requirements for smart card logon, see https://technet.microsoft.com/en-us/library/ff404293 (v=ws.10).aspx.
- Certificates must be valid. Validate them against the servers listed above.

Client applications

- In BlackBerry Work, to allow the use of client certificates, you must enable the useEASAuthCert setting.
- Apps must not send any password in the HTTP/HTTPS request.
- Apps must either set the HTTP/HTTPS header www-Authenticate: Negotiate or not set any authorization
 method in the HTTP or HTTPS request, to which the server has responded with 401 wwwAuthenticate:
 Negotiate, as detailed in https://www.ietf.org/rfc/rfc4559.txt.

Key points

The following are key points to note when integrating BlackBerry Dynamics and Kerberos infrastructure:

- The KDC host must be in the Allowed Domains of the Connectivity Profile applied to the affected users' policy sets in BlackBerry Control.
- The KDC host must be listening on TCP port 88 (Kerberos default port).
- BlackBerry Dynamics does not support KDC over UDP.
- BlackBerry Dynamics does not use Domain Name System (DNS) records such as SRV, CNAME, or TXT to locate the correct KDC. That is, the KDC must have an A record (IPv4) or AAAA record (IPv6) in your DNS.
- BlackBerry Dynamics does not use Kerberos configuration files (such as krb5.conf) to locate the correct KDC.
- The KDC can refer the client to another KDC host. BlackBerry Dynamics will follow the referral, as long as the KDC host that is referred to can be reached by BlackBerry Dynamics. This setting is defined in the **Allowed Domains** of the Connectivity Profile that is applied to the affected users' policy sets in BlackBerry Control.
- The KDC can obtain the TGT transparently to BlackBerry Dynamics from another KDC host.

Background on PKINIT, with FAQ

Consider the interactions in this drawing: http://www.ibm.com/developerworks/ibmi/library/i-sso/figure1.jpg

Kerberos PKINIT authentication requires the client (in the drawing, the human John, running a BlackBerry Dynamics-enabled application) to be able to contact:

- When initializing the user session, the user's Key Distribution Center (KDC) Authentication Service (AS) to obtain a Ticket-Granting Ticket (TGT)
- When establishing a connection to a resource (in the drawing, Service "A"), the resource's KDC Ticket-Granting Service (TGS)

In a large organization users and resources might belong to various realms and there may be many KDCs, so how does BlackBerry Dynamics find the right one?

- 1. How does the client locate the user's KDC Authentication Service when initializing the user's session?
 - Password-based authentication

The realm in the user name must contain the host name of the KDC AS. For example:

User: user@MY.REALM.COM

Password: myPassword

• Certificate-based authentication: This is PKINIT.

The realm in the UPN of the user's certificate must contain the host name of the KDC AS. For example:

```
UPN (OID 1.3.6.1.4.1.311.20.2.3): user@MY.REALM.COM
```

2. How does the client locate the resource's KDC Ticket-Granting Service (TGS) when retrieving the resource?

BlackBerry Dynamics attempts to obtain a TGS from the host in the domain of the resources URL. For example:

URL: http://resource.myrealm.com/index.html

The client will connect to KDC TGS running on host myrealm.com on TCP port 88.