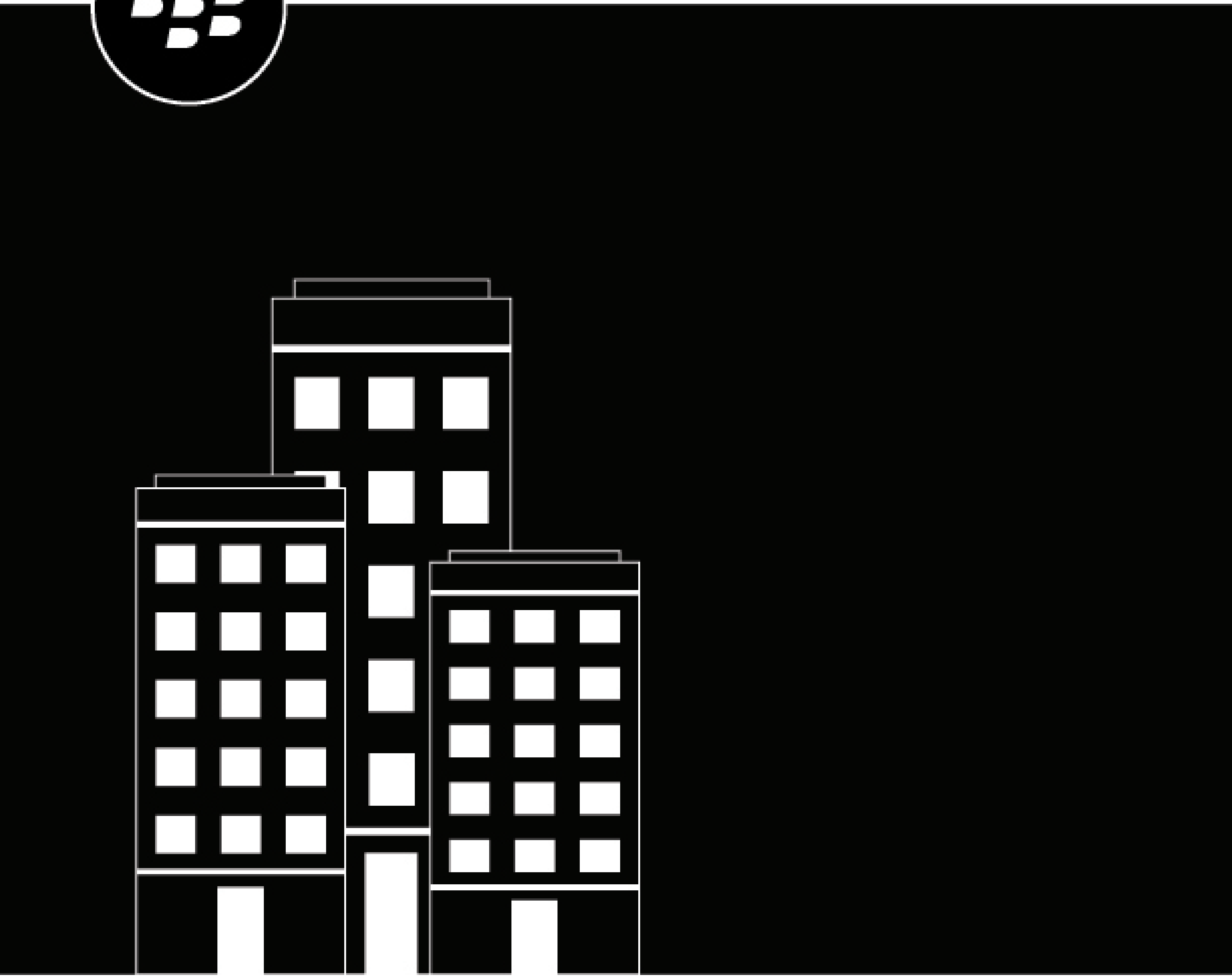


BlackBerry Dynamics

User Certificate Management Protocol and PKI Connector

Reference Guide

Contents

Using a PKI connector to implement custom certificate requirements

BlackBerry UEM can connect to a CA to obtain a certificate and send it to a client BlackBerry Dynamics app for authentication (for example, Entrust or PKI connections), or it can assist the client app in retrieving the certificate directly from a CA (for example, using SCEP to retrieve a PKCS12 file).

UEM uses the BlackBerry Dynamics User Certificate Management protocol to fetch and enroll a certificate when the BlackBerry Dynamics Runtime makes a request for the certificate. The protocol runs over HTTPS and defines JSON-formatted messages. This document details the administrator actions involved in this process and the APIs that UEM uses to execute it. The APIs are supported by UEM version 12.10 or later or BlackBerry UEM Cloud, and are available to BlackBerry Dynamics apps that use the BlackBerry Dynamics SDK version 2.1 or later.

Note: In a BlackBerry UEM Cloud environment, if the PKI connector is behind a firewall, you must have a BlackBerry Connectivity Node installed to allow UEM to communicate with the PKI connector through the BlackBerry Cloud Connector.

If you want to implement specific requirements or procedures when a certificate is retrieved from a CA (for example, if a user's password or smart card authentication is required), you can establish a back-end server that implements this protocol and the associated APIs to accept a request from UEM and interface with your enterprise CA. This server is called a PKI connector. When a BlackBerry Dynamics app makes a certificate request to UEM, UEM calls your PKI connector to interface with your CA and apply any required processes to retrieve and provide the certificate.

Note that UEM may already support your CA solution, so establishing a PKI connector may not be required. For more information about the CA solutions that UEM supports, see Sending CA certificates to devices and apps in the UEM Administration content.

A sample implementation of a PKI connector is described in the PKI connector sample implementation section of this guide. Your organization's developers can use the API documentation in this guide and the sample implementation (a .zip package) to establish a PKI connector that can interact with UEM.

Configuring UEM to retrieve certificates with a PKI connector

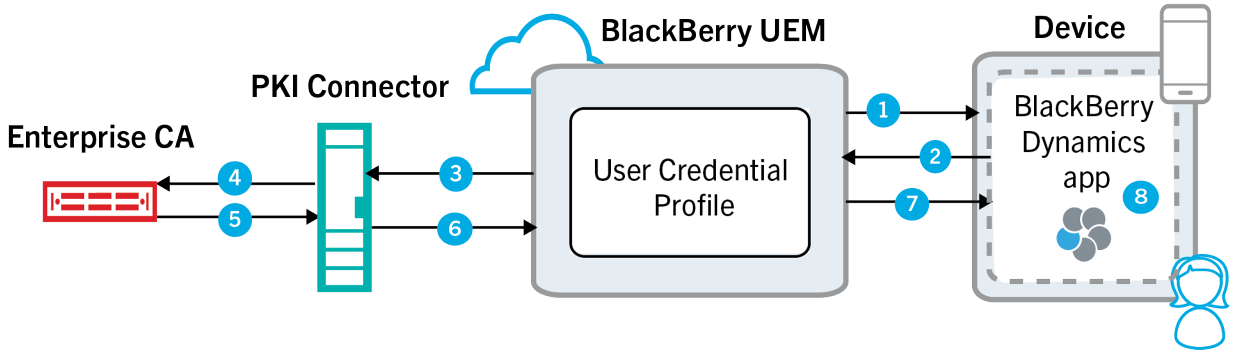
You can create a user credential profile in UEM to enable BlackBerry Dynamics apps on users' devices to obtain certificates from a PKI connector. For more information about connecting UEM to your PKI connector and creating and configuring a user credential profile, see the following sections in the UEM Administration content:

- Connect BlackBerry UEM to a BlackBerry Dynamics PKI connector
- Sending client certificates to devices and apps using user credential profiles
- Create a user credential profile to connect to your BlackBerry Dynamics PKI connector

You can specify the certificate renewal period in the user credential profile. You can also manually force the renewal of certificates in the UEM console. For more information, see Renew certificates that are enrolled through the BlackBerry Dynamics PKI connector.

When you delete a user, device, or certificate from UEM, UEM can notify the PKI connector that a user's public certificate is no longer in use and the PKI connector can revoke the certificates as necessary.

Process flow: Certificate enrollment using a PKI connector



1. The BlackBerry UEM administrator creates and configures a user credential profile to obtain client certificates for BlackBerry Dynamics apps from the enterprise CA using the organization's PKI connector. The administrator assigns the profile to the user.
2. The user installs and activates a BlackBerry Dynamics app. The BlackBerry Dynamics Runtime sends a request to UEM for a PKI certificate.
3. UEM calls the PKI connector to request the certificate.
4. The PKI connector carries out any custom logic that the organization requires (for example, a user password, smart card authentication, or monitoring of certificate requests) and requests the certificate from the enterprise CA.
5. The CA provides the certificate (key-pair) to the PKI connector.
6. The PKI connector provides the certificate to UEM.
7. UEM provides the certificate to the BlackBerry Dynamics app.
8. The app receives the certificate and uses it for different purposes, for example, to authenticate with the server when prompted, or to sign an email or document.

BlackBerry Dynamics User Certificate Management protocol APIs

BlackBerry UEM uses the following APIs to interact with a PKI connector. This section provides details and samples for each interface. Your PKI connector must implement these APIs to respond to certificate requests from UEM. See the PKI connector sample implementation section for details about a sample PKI connector that BlackBerry provides.

Interface	Description
`getInfo`	UEM uses this API to detect the commands that the PKI connector has implemented and to verify the authentication credentials for the PKI connector that are specified in the UEM management console.
`getUserKeyPair2`	UEM uses this API to fetch an initial key-pair and to renew an existing certificate.This API replaces the deprecated getUserKeyPair API.
`notifyCertificateReceived`	UEM can use this optional API to notify the PKI connector that the certificate has been received and imported by the BlackBerry Dynamics app.
`notifyCertificateRemoved`	UEM can use this optional API to notify the PKI connector that a certificate is no longer in use and can be revoked.

UEM supports HTTPS basic authentication or SSL client authentication with the PKI connector.

getInfo API

UEM uses this API to detect the commands that the PKI connector has implemented and to verify the authentication credentials for the PKI connector that are specified in the UEM management console. If the PKI connector does not implement this command, UEM assumes that the PKI connector supports the getUserKeyPair2 API only.

The HTTP request line is: `GET customerSpecifiedPrefix/pki?operation=getInfo`

`customerSpecifiedPrefix` obtains the path where the PKI connector is hosted. The UEM administrator specifies this path in a URL field in the management console (see Connect BlackBerry UEM to a BlackBerry Dynamics PKI connector in the UEM Administration content).

The API call returns the following JSON response value in the HTTP body:

Element	Type	Required	Details
`operations`	Array of strings	Yes	This element lists the protocol APIs that are implemented by the PKI connector.

Sample request and response:

In this sample, the administrator has specified the following URL for the PKI connector in the management console: https://ra.lifeonthedot.com

Request

```
GET /pki?operation=getInfo HTTP/1.0
Host: ra.lifeonthedot.com
Content-Type: application/json
Content-Length: 0
```

Response

```
HTTP/1.0 200 OK
Host: ra.lifeonthedot.com
Content-Type: application/json
Content-Length: XYZ

{
    “operations” : [“getInfo”, “getUserKeyPair2”]
}
```

getUserKeyPair2 API

UEM uses this API to fetch an initial key-pair and to renew an existing certificate. This API replaces the deprecated getUserKeyPair API.

The HTTP request line is: `POST customerSpecifiedPrefix/pki?operation=getUserKeyPair2`

UEM sends the following values:

Element	Type	Required	Details
`mType`	string	Yes	The value is `initialCert` if UEM is fetching the key-pair for the first time, and `renewCert` if UEM is renewing the certificate.
`user`	string	Yes	The user’s email address or another unique identifier. The certificate subject is created by the issuer.
`authToken`	string	No	An authentication token value provided by the user if it is required by the user credential profile.
`reqId`	string	No	For `initialCert` only, a request ID value to assist the sender in matching the response.
`deviceId`	string	No	For `initialCert` only, the BlackBerry Dynamics device ID. This value can be used to track if two apps on the same device are requesting the certificate at the same time.
`deviceName`	string	No	For `initialCert` only, the device name.

Element	Type	Required	Details
`cmsSigned`	string	No	When a BlackBerry Dynamics app renews an existing certificate, it uses the current private key to authenticate the request to renew. It sends this information in the `cmsSigned` element, and UEM sends this to the PKI connector without making any changes. cmsSigned = base64 encoded (cms-signed-data(CertRequest)) CertRequest includes `reqId, deviceId, deviceName,` and `pkcs10` (base64 encoded CSR).

The API call returns the following response values:

Element	Type	Required	Details
`status`	string	Yes	This can be a value of `success` or `failure`.
`failureInfo`	string	No	This element provides details about a failure.
`payloadType`	string	No	This is a value of `pkcs12`.
`payload`	Base64 encoded object	No	This is a value of `BASE64 Encoded PKCS12`.
`regID`	string	Yes	This is a request ID value.
`password`	string	No	If pkcs12 was password encrypted and `authToken` was not used as the password for encryption, a decryption password may be returned. For certificate renewal, a password is required.

Initial enrollment sample

In this sample, the administrator has specified the following URL for the PKI connector in the management console: https://ra.lifeonthedot.com

Request: The following payload is sent over an SSL connection to https://ra.lifeonthedot.com:

```
POST /pki?operation=getUserKeyPair2 HTTP/1.0
Host: ra.lifeonthedot.com
Content-Type: application/json
Content-Length: XYZ
```

```
{
    "mType": "initialCert",
    "user": "joe.foo@lifeonthedot.com",
    "authToken": "56ht12d0",
    "reqId": "12487",
    “deviceId”: “6e8S8JCLN7Hc5v3cGqvfkfM/C/tAFDS1CFUPJ53ASL”,
    “deviceName”: “Joe’s iPhone6”
}
```

If the server URL was set as https//ra.lifeonthedot.com/foo in the management console, the request will be:

```
POST /foo/pki?operation=getUserKeyPair2 HTTP/1.0
Host: ra.lifeonthedot.com

Content-Type: application/json
Content-Length: XYZ
```

Response:

```
HTTP/1.0 200 OK
Host: ra.lifeonthedot.com
Content-Type: application/json
Content-Length: XYZ

{
    "status":"success",
    "reqId": "12487",
    "payloadType":"pkcs12",
    "password":"clearTextPassword",
    "payload":"BASE64 Encoded PKCS12"
}
```

Renew sample

Request:

```
POST /pki?operation=getUserKeyPair2 HTTP/1.0
Host: ra.lifeonthedot.com
Content-Type: application/json
Content-Length: XYZ

{
    "mType": "renewCert",
    "user": "joe.foo@lifeonthedot.com",
    "cmsSigned": "base64-enoded-CMS-signed-data"
}
```

Response:

```
HTTP/1.0 200 OK
Host: ra.lifeonthedot.com
Content-Type: application/json
Content-Length: XYZ

{
    "status":"success",
    "reqId": "12487",
    "payloadType":"pkcs12",
    "password":"clearTextPassword",
    "payload":"BASE64 Encoded PKCS12"
}
```

notifyCertificateReceived API

UEM can use this optional API to notify a PKI connector that a BlackBerry Dynamics app has successfully received and imported certificates. A PKI connector can use this information for different purposes, for example, to take action if a certificate is not delivered successfully or to identify old certificates and take action to remove them.

The HTTP request line is:

`POST customerSpecifiedPrefix/pki?operation=notifyCertificateReceived`

UEM sends the following values:

Element	Type	Required	Details
`user`	String	Yes	This is the user’s email address.
`receivedCert`	Base64	Yes	This indicates that the certificate was received successfully.
`otherCerts`	Array of Base64 encoded objects	No	This is a list of previously delivered certificates that are also present on the device.
`deviceId`	string	No	This is the BlackBerry Dynamics device ID.
`deviceName`	string	No	This is the device name.

The API call returns the following response values:

Element	Type	Required	Details
`status`	string	Yes	This can be a value of `success` or `failure`.
`failureInfo`	string	No	A value of `retry` causes UEM to send the notification again. Any other value, such as `badRequest` or `unknownUser` will stop the notification process
`removeCerts`	Array of Base64 encoded objects	No	This lists the certificates that UEM will delete from the device.

Once an HTTP 200 success response is received, UEM removes this notification task from the queue. For any other error code, UEM tries to send the notification again.

Sample

Request:

```
POST /pki?operation=notifyCertificateReceived HTTP/1.0
Host: ra.lifeonthedot.com
Content-Type: application/json
Content-Length: XYZ
{
"user": "joe.foo@lifeonthedot.com",
"receivedCert": "base64-encode-x509"
}
```

Response:

```
HTTP/1.0 200 OK
Host: ra.lifeonthedot.com
Content-Type: application/json
Content-Length: XYZ
{
"status":"success",
}
```

notifyCertificateRemoved API

UEM can use this optional API to notify the PKI connector that a certificate is no longer in use and can be revoked.

The HTTP request line is:

`POST customerSpecifiedPrefix/pki?operation=notifyCertificateRemoved`

UEM sends the following values:

Element	Type	Required	Details
`user`	String	Yes	This is the user’s email address.
`removedCerts`	Array of Base64 encoded objects	Yes	This is a list of the certificates that are no longer in use.
`reason`	String	No	This element indicates why the certificate is no longer in use with one of the following values: • `userRemoved` • `certRemoved` • `appRemoved` • `duplicate`

Element	Type	Required	Details
`deviceId`	String	No	This is the BlackBerry Dynamics device ID.
`deviceName`	String	No	This is the device name.

The API call returns the following response values:

Element	Type	Required	Details
status	String	Yes	This can be a value of `success` or `failure`.
failureInfo	String	No	A value of retry causes UEM to send the notification again.

Once an HTTP 200 success response is received, UEM removes this notification task from the queue. For any other error code, UEM tries to send the notification again.

Sample

Request:

```
POST /pki?operation=notifyCertificateRemoved HTTP/1.0
Host: ra.lifeonthedot.com
Content-Type: application/json
Content-Length: XYZ
{
"user": "joe.foo@lifeonthedot.com",
"removedCerts": ["base64-encode-x509"],
"reason": "certRemoved"
}
```

Response:

```
HTTP/1.0 200 OK
Host: ra.lifeonthedot.com
Content-Type: application/json
Content-Length: XYZ
{
"status":"success",
}
```

PKI connector errors

The PKI connector can return any of the following failure errors:

Error	Description
`unknownUser`	The user does not exist or is not allowed.
`badRequest`	The format of the request is not valid.
`unknownRequest`	The requested action is not supported.
`authFailure`	The provided authentication credentials are expired or are not valid.
`badAlg`	The algorithm is not supported or is not recognized.
`unknownCert`	The certificate that is used or the reference is not found.
`badMessageCheck`	The signature or integrity check failed.
`badTime`	The time in the signature is not valid.
`unknown`	his is used for other errors that do not fit the categories described above.

PKI connector sample implementation

BlackBerry offers a sample implementation of a PKI connector that you can download and use as a reference for establishing your own PKI connector. The sample implementation demonstrates how UEM can interact with a PKI connector to obtain PKCS12 certificates from an Entrust CA.

Click here to download the sample implementation (gd-ca-adapters.zip). The sample is source code set up as a Maven project.

The sample implementation and the details in this section are intended for enterprise developers who are familiar with the following technologies:

- Java
- SOAP
- eb services
- pplication servers
- Certificate Authorities (CAs)
- SSL/TLS certificates for PKI
- BlackBerry UEM or BlackBerry UEM Cloud and the BlackBerry Dynamics platform

Software requirements for the PKI connector sample implementation

Click here to download the sample implementation (gd-ca-adapters.zip).

Item	Requirement
Java	Java 8 (including the JDK and JRE)
Apache Maven	Apache Maven 2.x.x or 3.x.x
Application server	Apache Tomcat 6, 7, or 8, or your preferred application server
BlackBerry UEM	• BlackBerry UEM version 12.10 or later It is a best practice to deploy the application server on a different computer than the one that hosts BlackBerry UEM • BlackBerry UEM Cloud The BlackBerry Connectivity Node must be installed and configured.
OS	Any of the following: • Microsoft Windows • Linux • macOS
Outbound network connection	Your development computer requires an Internet connection. When you build the PKI connector, Maven will use your Internet connection to obtain other software that is required.

Item	Requirement
Entrust account	You must have an Entrust account to authenticate to Entrust services. The sample implementation requires an Entrust username, password, service endpoint, and other details.To set up an Entrust account, visit https://www.entrustdatacard.com/.

Note: The sample implementation relies on several connected systems that might not be feasible to duplicate in an end-to-end test environment, from end-user devices to the Entrust service. It might only be feasible to test the feature in a production environment where all of the components are in place.

Configure Entrust properties

Before you begin:

- Unzip the sample implementation (gd-ca-adapters.zip).
- In the unzipped files, navigate to src/main/resources and make a back-up copy of **entrustMdmWs.properties** in case you want to use the default file at a later date.

1. In the unzipped files, navigate to src/main/resources and open **entrustMdmWs.properties** in your preferred editor.
2. Specify the appropriate value for each of the following properties. You do not need to enclose the values in quotation marks

Property	Optional or Required	Description
entrust.service.url	Required	This is the URL of the Entrust service.
entrust.login	Required	This is the login name for your Entrust account.
entrust.password	Required	This is the Base64 encoded password for your Entrust account.
entrust.digital.id.config.name	Required	This is the Digital Id Configuration provided to you by Entrust.
request.userName	Required	This is the username to use in requests to the Entrust service.
request.userEmail	Required	This is the email address to use in requests to the Entrust service.
entrust.P12LatestKeyOnly=true	Optional	his is a boolean value that must be true to obtain the history of an uploaded PKCS12 certificate.

Property	Optional or Required	Description
entrust.iggroup	Optional	You can use this property to create a Domain Name (DN) in the certificate for an individual Active Directory group. Use this property in combination with entrust.deviceType and entrust.deviceId. If you leave this property blank, the implementation generates a random string to obtain a unique certificate.
entrust.deviceType	Optional	his can be any arbitrary string. If you leave this property blank, the implementation generates a random string to obtain a unique certificate.
entrust.deviceId	Optional	his can be any arbitrary string. If you leave this property blank, the implementation generates a random string to obtain a unique certificate.

3. Save and close the file.

After you finish: Build the PKI connector sample implementation with Maven.

Build the PKI connector sample implementation with Maven

Before you begin:

- Configure Entrust properties.
- Verify that Maven is installed and in your path.
- Verify that you have set the JAVA_HOME and CLASSPATH environment variables

1. Open a command window.
2. Change the directory to the top directory of the unzipped **gd-ca-adapters** package.
3. Execute the following command to start the build: `mvn clean install`
4. Review the output messages for errors and correct those errors if necessary.

A successful build ends with the message`[INFO] BUILD SUCCESS`. The build creates a web archive (.war) file and associated files in the gd-ca-adapters/target sub-directory.

After you finish: Prepare your application server.

Prepare your application server

The PKI connector sample implementation uses an Apache Tomcat application server, but any application server can be used with a PKI connector. It is a best practice to deploy the application server on a different computer than the one that hosts BlackBerry UEM.

Before you begin:

- Build the PKI connector sample implementation with Maven
- Use the Java keytool to create a Keystore to store client certificates and a TrustStore to store certificates from trusted servers (such as Entrust). For instructions, see Generating a Keystore and TrustStore.
- Get a copy of the Entrust server public certificate.

1. Open a command window.
2. Run the following command:

   ```
   cd <unzipped path>/gd-ca-adapters/src/main/java/com/good/adapters/entrust/util
   ```
3. Go to the target/classes folder and run the following command:

   ```
   java com.good.adapters.entrust.util.GenerateSSLCertificateForEntrustCA
    ENTRUST_HOSTNAME <password>
   ```
4. In the list of certificates to trust, select the root certificate.
5. Copy the resulting **jssecacerts** file to the following Java installation directory: $JAVA_HOME/jre/lib/security.
6. Edit the Tomcat **server.xml** file ($TOMCAT_DIR/conf/server.xml) to specify the port that the PKI onnector listens on and the paths to the KeyStore and TrustStore. See the following example (change the port, KeyStore, and TrustStore values as necessary):

   ```
   <Connector port="8090" protocol="HTTP/1.1"
       connectionTimeout="20000"
       redirectPort="8443" />

   <Connector
       protocol="HTTP/1.1"
       port="8443" maxThreads="200"
       scheme="https" secure="true" SSLEnabled="true"
       keystoreFile="/c:/newcerts/foobar.jks" keystorePass="foobarpwd"
       truststoreFile="/c:/newcerts/cacerts.jks" truststorePass="cacertspassword"
       clientAuth="true" sslProtocol="TLS"/>
   ```

After you finish: Deploy the PKI connector war file to the application server.

Deploy the PKI connector war file to the application server

Before you begin:

- Prepare your application server.
- Rename the .war file that the build process generated (see Build the PKI connector sample implementation with Maven) to fit your organization's naming conventions. If you name the file ROOT.war you can run it by navigating to the root of the application server in a browser. For example, if Tomcat is running the sample implementation in a .war file named ROOT.war, on a host server named myAppServer (port 443) in the BigCompany.com domain, the URL to run the sample implementation is https://myAppServer.BigCompany.com/.

Deploy the .war file to the Tomcat application server. For instructions, see Tomcat Web Application Deployment. You can copy the .war file, either manually or with a deployment manager program, to the webapps directory of your application server.

After you finish: Configure BlackBerry UEM to communicate with the PKI connector.

Configure BlackBerry UEM to communicate with the PKI connector

Before you begin: Deploy the PKI connector war file to the application server.

You must configure BlackBerry UEM or BlackBerry UEM Cloud to communicate with the PKI connector sample implementation. For instructions, see Configuring UEM to retrieve certificates with a PKI connector.

After you finish: Test connectivity to the PKI connector sample implementation.

Test connectivity to the PKI connector sample implementation

In a browser, navigate to ***<Tomcat_host_URL>*****/pki/test**, where *<Tomcat_host_URL>* consists of the host server name you specified in the UEM management console and the domain.

For example, on a host server named myAppServer in the BigCompany.com domain, the URL to run the test is https://myAppServer.BigCompany.com/pki/test.

A successful connection returns the message `It works`.

Adapting the PKI connector sample implementation

Note the following considerations when you modify the PKI connector sample implementation to work with a CA other than Entrust:

Item	Consideration
Start with CertificateServerService.Java	CertificateServerService.java (in src/main/java/com/good/service/) indicates the various touch points that you need to account for when you set up your own PKI connector.
Adhere to the BlackBerry Dynamics User Certificate Management Protocol	You must ensure that any adaption of the sample implementation also implements the standards and processes of the BlackBerry Dynamics User Certificate Management Protocol.
Generating program stubs	The sample implementation relies on the following Entrust files that are included with the software: • AdminServiceV9.wsdl • ServiceV9Common.xsd These definitions are used to generate program stubs with Axis 1.4. Your CA might have similar definitions that you can use for your own adaptation of the PKI connector sample. You may also need to use a more recent version of Axis.
Property handling	You will need to set up a properties file similar to entrustMdmWs.properties for your intended CA.
Format of certificate data	The sample implementation receives a byte stream of the unencoded certificate data from Entrust. The format of the certificate depends on your intended CA. You may need to modify your implementation to process data in a different format

Legal notice

This documentation including all documentation incorporated by reference herein such as documentation provided or made available on the BlackBerry website provided or made accessible "AS IS" and "AS AVAILABLE" and without condition, endorsement, guarantee, representation, or warranty of any kind by BlackBerry Limited and its affiliated companies ("BlackBerry") and BlackBerry assumes no responsibility for any typographical, technical, or other inaccuracies, errors, or omissions in this documentation. In order to protect BlackBerry proprietary and confidential information and/or trade secrets, this documentation may describe some aspects of BlackBerry technology in generalized terms. BlackBerry reserves the right to periodically change information that is contained in this documentation; however, BlackBerry makes no commitment to provide any such changes, updates, enhancements, or other additions to this documentation to you in a timely manner or at all.

This documentation might contain references to third-party sources of information, hardware or software, products or services including components and content such as content protected by copyright and/or third-party websites (collectively the "Third Party Products and Services"). BlackBerry does not control, and is not responsible for, any Third Party Products and Services including, without limitation the content, accuracy, copyright compliance, compatibility, performance, trustworthiness, legality, decency, links, or any other aspect of Third Party Products and Services. The inclusion of a reference to Third Party Products and Services in this documentation does not imply endorsement by BlackBerry of the Third Party Products and Services or the third party in any way.

EXCEPT TO THE EXTENT SPECIFICALLY PROHIBITED BY APPLICABLE LAW IN YOUR JURISDICTION, ALL CONDITIONS, ENDORSEMENTS, GUARANTEES, REPRESENTATIONS, OR WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY CONDITIONS, ENDORSEMENTS, GUARANTEES, REPRESENTATIONS OR WARRANTIES OF DURABILITY, FITNESS FOR A PARTICULAR PURPOSE OR USE, MERCHANTABILITY, MERCHANTABLE QUALITY, NON-INFRINGEMENT, SATISFACTORY QUALITY, OR TITLE, OR ARISING FROM A STATUTE OR CUSTOM OR A COURSE OF DEALING OR USAGE OF TRADE, OR RELATED TO THE DOCUMENTATION OR ITS USE, OR PERFORMANCE OR NON-PERFORMANCE OF ANY SOFTWARE, HARDWARE, SERVICE, OR ANY THIRD PARTY PRODUCTS AND SERVICES REFERENCED HEREIN, ARE HEREBY EXCLUDED. YOU MAY ALSO HAVE OTHER RIGHTS THAT VARY BY STATE OR PROVINCE. SOME JURISDICTIONS MAY NOT ALLOW THE EXCLUSION OR LIMITATION OF IMPLIED WARRANTIES AND CONDITIONS. TO THE EXTENT PERMITTED BY LAW, ANY IMPLIED WARRANTIES OR CONDITIONS RELATING TO THE DOCUMENTATION TO THE EXTENT THEY CANNOT BE EXCLUDED AS SET OUT ABOVE, BUT CAN BE LIMITED, ARE HEREBY LIMITED TO NINETY (90) DAYS FROM THE DATE YOU FIRST ACQUIRED THE DOCUMENTATION OR THE ITEM THAT IS THE SUBJECT OF THE CLAIM.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW IN YOUR JURISDICTION, IN NO EVENT SHALL BLACKBERRY BE LIABLE FOR ANY TYPE OF DAMAGES RELATED TO THIS DOCUMENTATION OR ITS USE, OR PERFORMANCE OR NON-PERFORMANCE OF ANY SOFTWARE, HARDWARE, SERVICE, OR ANY THIRD PARTY PRODUCTS AND SERVICES REFERENCED HEREIN INCLUDING WITHOUT LIMITATION ANY OF THE FOLLOWING DAMAGES: DIRECT, CONSEQUENTIAL, EXEMPLARY, INCIDENTAL, INDIRECT, SPECIAL, PUNITIVE, OR AGGRAVATED DAMAGES, DAMAGES FOR LOSS OF PROFITS OR REVENUES, FAILURE TO REALIZE ANY EXPECTED SAVINGS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, LOSS OF BUSINESS OPPORTUNITY, OR CORRUPTION OR LOSS OF DATA, FAILURES TO TRANSMIT OR RECEIVE ANY DATA, PROBLEMS ASSOCIATED WITH ANY APPLICATIONS USED IN CONJUNCTION WITH BLACKBERRY PRODUCTS OR SERVICES, DOWNTIME COSTS, LOSS OF THE USE OF BLACKBERRY PRODUCTS OR SERVICES OR ANY PORTION THEREOF OR OF ANY AIRTIME SERVICES, COST OF SUBSTITUTE GOODS, COSTS OF COVER, FACILITIES OR SERVICES, COST OF CAPITAL, OR OTHER SIMILAR PECUNIARY LOSSES, WHETHER OR NOT SUCH DAMAGES

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