6.10.2 Using PFX Certificates in Connector/NET

.NET does not provide native support the PEM format. Instead, Windows includes a certificate store that provides platform-dependent certificates in PFX format. For the purposes of this example, use test client certificates from the MySQL server repository (./mysql-test/std\_data). Convert these to PFX format first. This format is also known as PKCS#12.

To complete the steps in this tutorial for PFX certificates, you must have Open SSL installed. This can be downloaded for Microsoft Windows at no charge from [Shining Light Productions](http://www.slproweb.com/products/Win32OpenSSL.html).

**Creating a Certificate File to Use with the .NET Client**

1. From the directory ***server-repository-root***/mysql-test/std\_data, issue the following command.

openssl pkcs12 -export -in client-cert.pem -inkey client-key.pem -certfile cacert.pem -out client.pfx

1. When asked for an export password, enter the password “pass”. The file client.pfx will be generated. This file is used in the remainder of the tutorial.

**Connecting to the Server Using a File-Based Certificate**

1. You will use the PFX file, client.pfx, that you created in the previous step to authenticate the client. The following example demonstrates how to connect using the SslMode, CertificateFile, and CertificatePassword connection string options.
2. using (MySqlConnection connection = new MySqlConnection(
3. "database=test;user=sslclient;" +
4. "CertificateFile=H:\\git\\mysql-trunk\\mysql-test\\std\_data\\client.pfx;" +
5. "CertificatePassword=pass;" +
6. "SslMode=Required "))
8. {
9. connection.Open();

}

The path to the certificate file will need to be changed to reflect your individual installation. When using PFX format certificates, the SslMode connection option validates certificates for all SSL mode values, except None.

**Connecting to the Server Using a Store-Based Certificate**

1. The first step is to import the PFX file, client.pfx, into the Personal Store. Double-click the file in Windows explorer. This launches the Certificate Import Wizard.
2. Follow the steps dictated by the wizard, and when prompted for the password for the PFX file, enter “pass”.
3. Click Finish to close the wizard and import the certificate into the personal store.

**Examining Certificates in the Personal Store**

1. Start the Microsoft Management Console by entering mmc.exe at a command prompt.
2. Select **File**, **Add/Remove snap-in**. Click Add. Select **Certificates** from the list of available snap-ins in the dialog.
3. Click Add button in the dialog, and select the **My user account** radio button. This is used for personal certificates.
4. Click the Finish button.
5. Click OK to close the Add/Remove Snap-in dialog.
6. You will now have **Certificates – Current User** displayed in the left panel of the Microsoft Management Console. Expand the Certificates - Current User tree item and select **Personal**, **Certificates**. The right panel will display a certificate issued to MySQL. This is the certificate that was previously imported. Double-click the certificate to display its details.
7. After you have imported the certificate to the Personal Store, you can use a more succinct connection string to connect to the database, as illustrated by the following code:
8. using (MySqlConnection connection = new MySqlConnection(
9. "database=test;user=sslclient;" +
10. "Certificate Store Location=CurrentUser;" +
11. "SslMode=Required"))
13. {
14. connection.Open();

}

**Certificate Thumbprint Parameter**

If you have a large number of certificates in your store, and many have the same Issuer, this can be a source of confusion and result in the wrong certificate being used. To alleviate this situation, there is an optional Certificate Thumbprint parameter that can additionally be specified as part of the connection string. As mentioned before, you can double-click a certificate in the Microsoft Management Console to display the certificate's details. When the Certificate dialog is displayed click the **Details** tab and scroll down to see the thumbprint. The thumbprint will typically be a number such as ‎47 94 36 00 9a 40 f3 01 7a 14 5c f8 47 9e 76 94 d7 aa de f0. This thumbprint can be used in the connection string, as the following code illustrates:

using (MySqlConnection connection = new MySqlConnection(

"database=test;user=sslclient;" +

"Certificate Store Location=CurrentUser;" +

"Certificate Thumbprint=479436009a40f3017a145cf8479e7694d7aadef0;"+

"SSL Mode=Required"))

{

connection.Open();

}

Spaces in the thumbprint parameter are optional and the value is not case-sensitive.