



Keytool and Certificate Management

A guide to utilizing keytool to assist with Certificates for eMedNY SOAP

TABLE OF CONTENTS

1	Introduction	3
2	Creating a Certificate Signing Request (CSR) and requesting a Certificate.....	4
2.1	Create the “KeyStores” Folder & Open the Command Window	4
2.2	Generate a key pair for your web service client.....	6
2.3	Generate a certificate request.....	7
2.4	Submit the Certificate Signing Request through ePACES.....	7
2.5	Retrieve the Client Certificate from ePACES.....	8
3	Importing the Certificate.....	10
3.1	Importing the Client Certificate into the keystore	10
3.2	Importing the Server Certificate.....	11
3.3	Importing certificates using a MMC	12
3.4	Importing Certificates Using the IE Certificates Wizard.....	19
3.5	Importing the Server Certificate into the Keystore.....	27
3.6	Importing private key and certificates from Java to Windows Key Stores	29
4	Additional Tools and Information.....	34
4.1	keytool web link.....	34
4.2	Requirements for CORE Compliance.....	34
4.3	JSSE Reference Guide.....	34
4.4	WCF – 2 Way SSL using Certificates.....	34

1 Introduction

This document is intended to assist providers with the acquisition and use of security certificates when accessing secured eMedNY Web Services (such as the [File Transfer Service](#) and the [Meds History Service](#).) It details the processes needed to create a private and public key pair, and an associated Certificate Signing Request (CSR) and how to process the resulting signed certificate delivered in response from eMedNY, as well as how to import any server certificates that may also be required (e.g MedsHistory Web Service).

The process to enroll as a user of eMedNY-signed certificates , to submit the Certificate Signing Request created as described in this document, and the retrieval of the user's eMedNY-signed certificate are detailed in a separate document [eMedNY X509 Certificates Guide](#).

Please note that the information provided is to be used as a guide only. It will be of greatest interest to developers using Java or Dotnet technologies on Windows platforms. Developers on Linux-variants should be able to adapt the instructions given herein. Developers using key and certificate stores other than those covered in this document will have to adapt the information here to the fit their own situation.

2 Creating a Certificate Signing Request (CSR) and requesting a Certificate

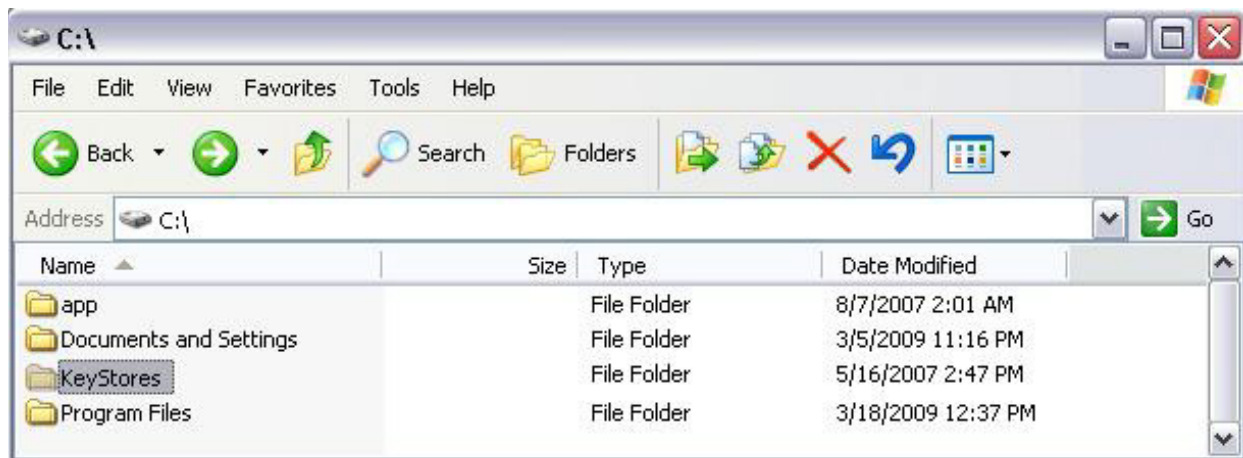
“Keytool” is a key and certificate management utility, which will be used to generate key pairs for your web services client. This utility will need to be installed so that it can be run in any directory. For additional information regarding keytool, see keytool web link in [Section 4.1](#). The following instructions are for Windows-based operating systems. Other operating systems have analogous tools; users should consult the appropriate references or with your IT departments.

eMedNY can provide scripts contained in batch files with these commands. They can be run using instructions contained within these files. Please email emednyproviderservices@gdit.com to request these scripts.

The following instructions are for example purposes only. Words that are in bold are sample text, please use whatever suits the needs of your software in its place.

2.1 Create the “KeyStores” Folder & Open the Command Window

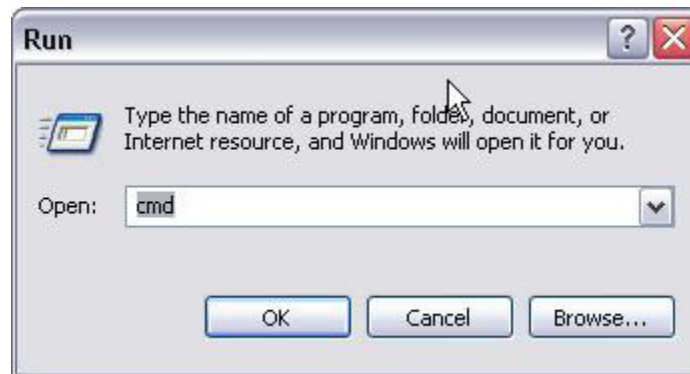
In Windows Explorer you will need to create a new folder for your keytool-related files. In our example, we have called it **KeyStores**:



Next, In Start Menu Click Run:



Type "cmd" and press Enter:



Change directory to the one you created earlier (i.e. **Keystores**):



2.2 Generate a key pair for your web service client

At the command line, type in the following command, substituting your own data for the text in brackets []. A full explanation of this data appears after the command line:

```
keytool -genkeypair -v -alias [Client Alias] -keystore [keystore name].jks -keyalg RSA -sigalg SHA1withRSA -storepass [keystore password] -dname "[Client's DN, see below]" -keypass [client password]
```

Client Alias – An alias set up by the user. It is used to refer to the keystore. A keystore can contain multiple client items, each referenced by its own unique alias.

Keystore Name – The name of the Keystore. A file named [keystore name].jks will be generated.

Keystore Password – A password to access the keystore. We suggest a password should be at least six characters in length and contain at least one number and punctuation mark.

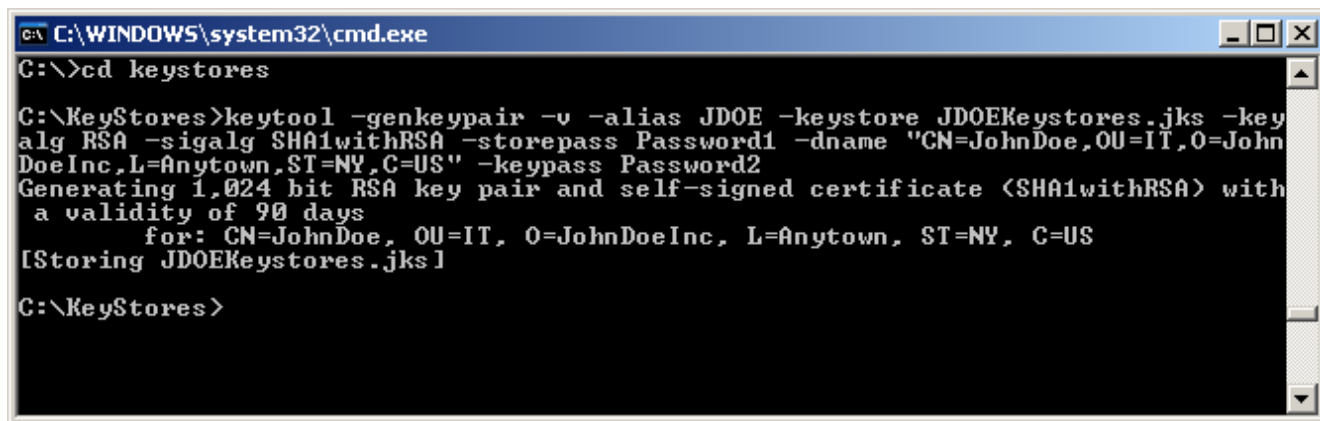
Client's DN – This is information identifying the client. You should populate it with information identifying your organization (using no spaces) as follows: "CN = [Client Name], OU=[Organizational Unit], O=[Organization], L=[City or Locality], S=[State], C=US"

Client Password – The client password to access the private key in this keystore.

This is an example:

```
keytool -genkeypair -v -alias JDOE -keystore JDOEKeystores.jks -keyalg RSA -sigalg SHA1withRSA -storepass Password1 -dname "CN=JohnDoe,OU=IT,O=JohnDoeInc,L=Anytown,ST=NY,C=US" -keypass Password2
```

See below for an example:



```
C:\WINDOWS\system32\cmd.exe
C:\>cd keystores

C:\KeyStores>keytool -genkeypair -v -alias JDOE -keystore JDOEKeystores.jks -keyalg RSA -sigalg SHA1withRSA -storepass Password1 -dname "CN=JohnDoe,OU=IT,O=JohnDoeInc,L=Anytown,ST=NY,C=US" -keypass Password2
Generating 1,024 bit RSA key pair and self-signed certificate (SHA1withRSA) with a validity of 90 days
    for: CN=JohnDoe, OU=IT, O=JohnDoeInc, L=Anytown, ST=NY, C=US
[Storing JDOEKeystores.jks]

C:\KeyStores>
```

2.3 Generate a certificate request

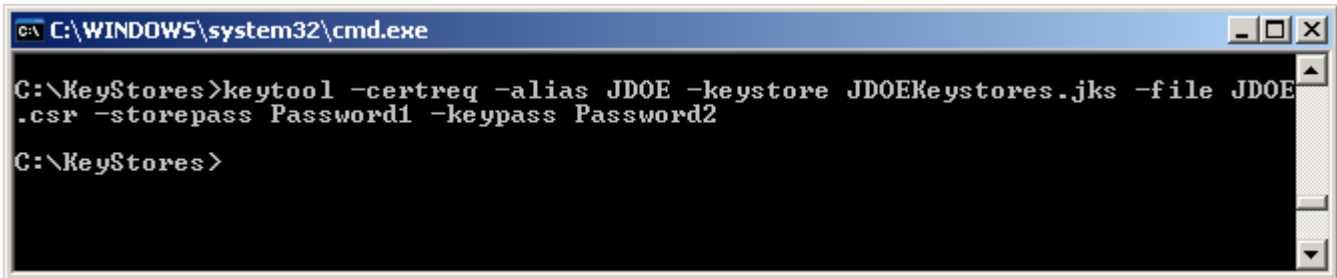
Type the following command to generate a Certificate Signing Request. Replace the text in brackets with the same data you used to generate the Keystore.

```
keytool -certreq -alias [Client Alias] -keystore [Keystore Name].jks -file [Client Alias].csr -storepass [Keystore Password] -keypass [Client Password]
```

See below for an example:

```
keytool -certreq -alias JDOE -keystore JDOEKeystores.jks -file JDOE.csr -storepass Password1 -keypass Password2
```

The image follows that example:



```
C:\WINDOWS\system32\cmd.exe

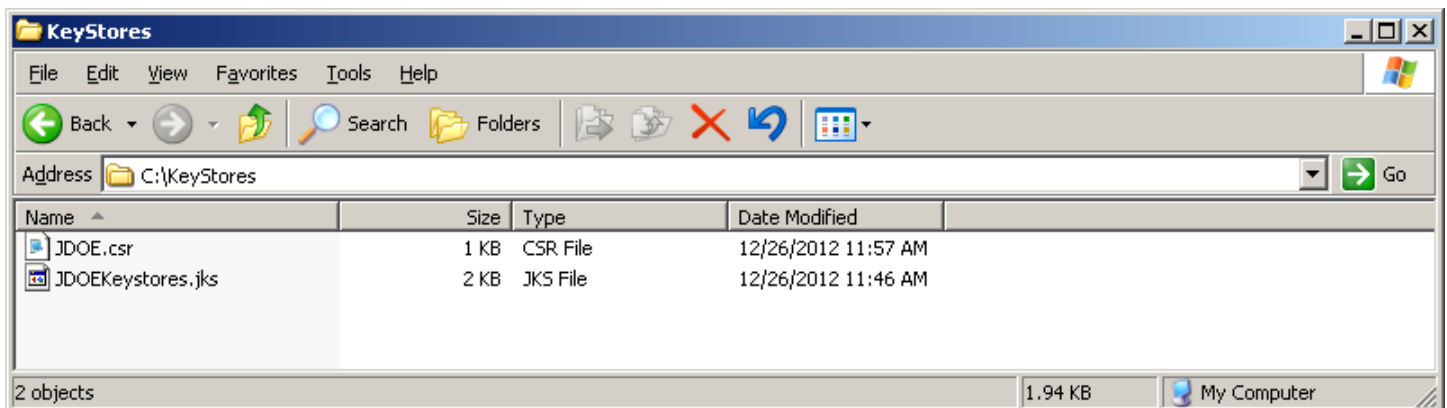
C:\KeyStores>keytool -certreq -alias JDOE -keystore JDOEKeystores.jks -file JDOE.csr -storepass Password1 -keypass Password2

C:\KeyStores>
```

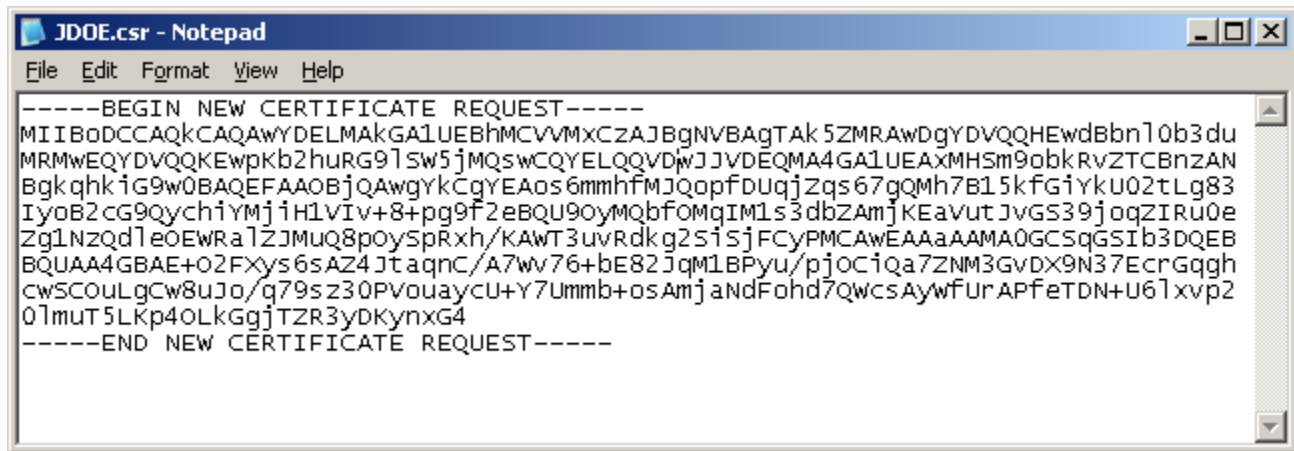
This will result in a Certificate Request file (.csr) being placed into the same directory.

2.4 Submit the Certificate Signing Request through ePACES

View the **Keystores** directory in Windows explorer.



As you can see above, there are two files in the directory. The .jks is your keystore file. The .csr is your Certificate Signing Request. Open the .csr file in a text editor, such as NotePad. It should look something like this:



```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBODCCAQkCAQAwYDELMAkGA1UEBHMCVVMxCZAJBgNVBAGTAk5ZMRAwdG9yDVQQHEwdBbn10b3du
MRMwEQYDVQQKEwpkb2huRG91SW5jMQswCQYELQQLDwJJVDEQMA4GA1UEAxMHSm9obkRvZTCBnzAN
BgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAos6mmhfMJQopfDUqjZqs67gQMh7B15kfgiYkU02tLg83
IyoB2cG9QychiYmjih1vIv+8+pg9F2eBQU9OymQbfOMqIM1s3dbZAmjKEavutJvGS39jogZIRu0e
Zg1NzQd1eOEWRa1ZJMuQ8poySprxh/KAWT3uvRdkg25i5jFCyPMCAwEAAaAAMA0GCSqGSIb3DQEB
BQUAA4GBAE+O2FXys6sAZ4Jtaqnc/A7ww76+be82JqM1BPYu/pjOCiQa7ZNM3GvDX9N37EcrGqgh
cwSCouLgCw8uJo/q79sz30Pvouaycu+Y7Umb+osAmjandFohd7QwcsAywFurAPFeTDN+U61xvp2
01mut5Lkp40LkGgjTZR3ydkynxG4
-----END NEW CERTIFICATE REQUEST-----
```

Copy this information into the “Certificate Signing Request” Field in ePACES as per the instructions in the [eMedNY X509 Certificates Guide](#).

2.5 Retrieve the Client Certificate from ePACES

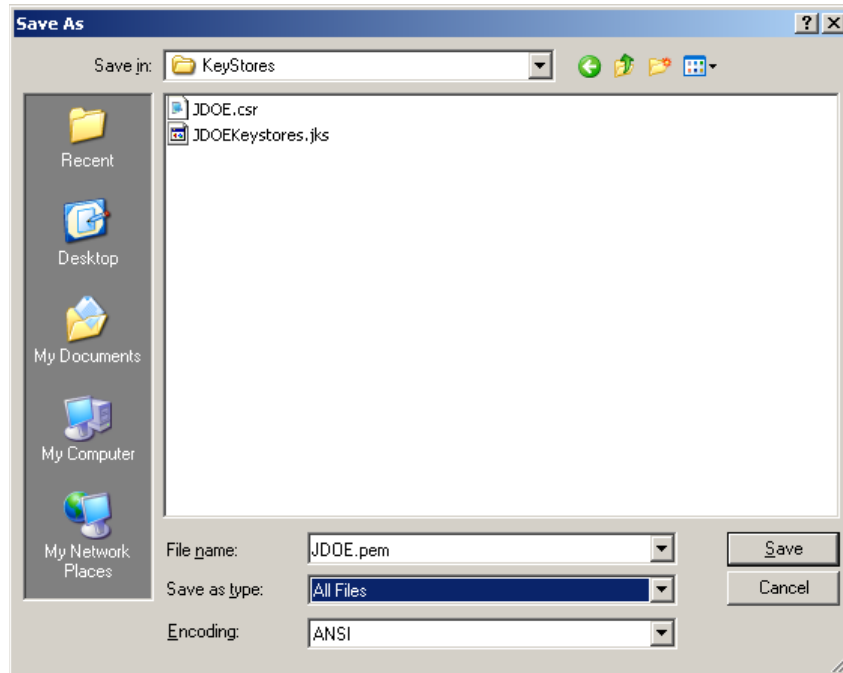
Retrieve the Client Certificate through ePACES as per the instructions in the [eMedNY X509 Certificates Guide](#). Copy the data there into Notepad and save it in the “Keystores” directory under the name [Client Alias].pem

This is a sample of a Client certificate.

```

-----BEGIN CERTIFICATE-----
MIIGJgyJKoZIhvcNAQcCoIIIGFZCCBhMCAQEXADALBgkqhkiG9w0BBwgggX7MIID
jDCCA/vwAwIBAgIBMTANBgkqhkiG9w0BAQUFADA2MQ8wDQYDVQQKEwZ1TWVkdT1kx
IzAhBgNVBAsTGm1EZXYgQ2VydG1mawNhdGUGugXV0aG9yaXR5MB4XDTEwMDkxNTA0
MDAwMFoXDTEyMDMxODAzNTk1OVowXTEyMDMxODAzNTk1OVowXTEyMDMxODAzNTk1OVow
ZW50MmwwcGyDVQQLEWhNPVtMxODAzNTk1OVowXTEyMDMxODAzNTk1OVowXTEyMDMxODAz
MREwDwYDVQDEWhKU01JVEgWMTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoC
ggEBAMtMCX1YvuF1yxns1J1v+U0s1iH01BAYD8MGU/Eus1J6mN6df0QRTAZJzn5e
hwZZutJsr7v+4SraTVxkxqbdEb8zD+Cza7Xp1xk1a9NR+/Tpp4NUT8CPasobsvmt
TBecsunsCyr8rF37uAeTOEB/k8kkJKAD/zu/iz4AI+tyRRY/76sIUQ3c0Trvd1+4
/ertEhxydrw9wxhbykeSyR8vrtxiEb26Nh8mngBwGuMfLdFqmV78zePB4Q00fjs
P120rvzPEI6mQZSRZABXNM0x1inVhycv40NOZg18fe7t0e3Fuc5YSMPkKCQ/N+Js
CX0Chmpx14SO2dTMdJwT8zR14/cCAwEAAaOB/jCB+ZA0BgNVHQ8BAf8EBAMCBPAw
EwYDVR01BAwwCgYIKwYBBQUHAwIwGZMGA1UdHwSB1zCB1DBNoEugSaRHMEUxDZAN
BgNVBAoTBMVNZWR0TEjMCEGA1UECXMabur1diBDZjJ0awZpy2F0ZSBBdXR0b3Jp
dHkxDTALBgNVBAMTBENSTDEwN6A1OD0GMWh0dHA6Ly8yMCA4YmC4xNTcUNDU6ODAA
MS9QS01TZJ2L2Nhy2VydHMvQ1JMM55jcmwwHQYDVRO0BBYEFB1f1ftzukLz5Eho
g1hk4ZsDSQ+WMB8GA1UdIwQYMBaAFHe2ian2duwyEHC6+qudm3kwc2xSMA0GCSqS
SIb3DQEBBQUAA4GBAHFXEDCqd+BTkGos1QXYs2ZFk+qvHzsix08Dktpidy2ok99B
bMNIbvX02afRwzWAdf3LHfysVaaXYgedigDEIVGzppUEjw81AyoHrvy2ouAL7hnk
syHAA4g3/CYz9tDITfAXAA0Yor2BYdoJ9BZ1FM6KSwtyYtUGfMk7L12BQzneMIIC
ZzCCAdCgAwIBAgIBADANBgkqhkiG9w0BAQUFADA2MQ8wDQYDVQQKEwZ1TWVkdT1kx
IzAhBgNVBAsTGm1EZXYgQ2VydG1mawNhdGUGugXV0aG9yaXR5MB4XDTEwMDkxNTA0
MDAwMFoXDTEyMDMxODAzNTk1OVowNjEPMA0GA1UECHMGZU1lZE5ZMSMwIQYDVQQL
EpxtrGV2IEN1cnRpZm1jYXR1IEF1dGhvcml0eTCBnzANBgkqhkiG9w0BAQEFAAOB
jQAwgYkCgYEAass2jjHk0vrvtwK12vp4NhcdeA9jkx4qks0jd48rub+o/hRU5NERz
VJ0z42hx/nIEmZG1P0hm1YewoG6l60M/vBvCBovgs8r/Dw+0ArY09ZFcnrsBbcUJ
vVl7M76ha5+wn076k129Gfz1solwwzceukH1v9j/yDyJHH1vdv3n1scAwEAAaOB
hDCBgTA/Bg1ghkGBhvhCAQ0EMhMWR2VuzXJhdGvkIGJ5IHROzSBTZWN1cm10eSBT
ZXJ2ZXIgzM9yIHovt1MgkFJBQ0YpMA4GA1UddwEB/wQEAWIBBjAPBgNVHRMBAf8E
BTADAQH/MB0GA1UddGQwBBR3tomjdnVMMhIquvq1H2t5FnNsUjANBgkqhkiG9w0B
AQUFAAOBgQBQhDo4u0dZcvtD3J1j9byfJ0savF3JHp++yHLqH1gaHbAhGwi16R8w
5ThNBKZE69WRxuAM1PyHkvwFC401Cb8TwmHttbut5I/Yncp3XLdmerUlKfQ3T8v
Bs3XoQnHHCumpiVIAit139Z/PS5hdowdsI0o9M3K+wY2/duigw2gAENTj
-----END CERTIFICATE-----

```



3 Importing the Certificate

3.1 Importing the Client Certificate into the keystore

Type the following command to import the Client Certificate into your keystore. Replace the text in brackets with the appropriate data – [Cert file name] is the name of the file where you saved the certificate from ePACES. An explanation of the data elements follows the command:

```
keytool -importcert -v -alias [Client Alias] -file [Cert file Name].pem -keystore [Keystore name].jks -storepass [keystore password] -keypass [client password]
```

Client Alias – The same Client Alias as above.

Cert File Name – the Name of the desired certificate file. We recommend the same as the Client Alias for tracking purposes.

Keystore Password – A password to access the keystore. We suggest a password should be at least six characters in length and contain at least one number and punctuation mark. This is the same as used for creating the keystore and generating the CSR.

Client Password – The client password to access the private key in this keystore.

This is an example command:

```
keytool -importcert -v -alias JDOE -file JDOE.pem -keystore JDOEKeystores.jks -storepass Password1 -keypass Password2
```

The following image follows our example:

```

C:\KeyStores>keytool -certreq -alias JDOE -keystore JDOEKeystores.jks -file JDOE
.csr -storepass Password1 -keypass Password2

Top-level certificate in reply:

Owner: OU=rPrd Certificate Authority, O=eMedNY
Issuer: OU=rPrd Certificate Authority, O=eMedNY
Serial number: 0
Valid from: Mon May 11 00:00:00 EDT 2009 until: Tue Feb 09 22:59:59 EST 2021
Certificate fingerprints:
    MD5:   B6:75:E4:87:48:05:7D:19:C8:9C:72:91:51:BA:0B:77
    SHA1:  7B:FB:D2:AA:6F:A0:52:E3:76:92:4B:2C:EF:0C:86:CE:A9:81:5E:E1
    Signature algorithm name: SHA1withRSA
    Version: 3

Extensions:

#1: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
  Key_CertSign
  CrI_Sign
]

#2: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#3: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
  KeyIdentifier [
0000: C1 BA 37 B5 74 45 72 4D   30 37 98 36 0C F4 BE FF   ..7.tErM07.6....
0010: EC 55 1E 74                               .U.t
  ]
]

#4: ObjectId: 2.16.840.1.113730.1.13 Criticality=false

... is not trusted. Install reply anyway? [no]: yes
Certificate reply was installed in keystore
[Storing JDOE.jks]

```

The reason the certificate is indicated as “not trusted” as the certificate returned by eMedNY is a chain consisting of the user’s cert as well as the Certificate Administrator Signer Cert (rPrd Certificate Authority.) It is the latter certificate that is “not trusted.” You can also import only the eMedNY client cert without the CA signer cert if you first import the returned chain into Internet Explorer and then re-export just the eMedNY client cert from the browser. If you need to do this for your software, see the instructions under Importing Certificates using the IE Certificates Wizard.

3.2 Importing the Server Certificate

For Java based web service clients, doing message level security (e.g. Meds History Service - MHS,) the Server certificate needs to be imported into this keystore or in a separate keystore. We will do it in this keystore.

The Server certificate can be obtained from eMedNY – please email emednyproviderservices@gdit.com if you need this certificate. For this demo, it will be downloaded to:

C:\Keystores\dev_server.pem

And imported using KeyToolUI via Import > Keystore's entry > Trusted Certificate > Regular Certificate

However, this will fail. This is because the eMedNY cert contains a certificate chain (server cert + CA cert) and a jks keystore will not allow one to import a cert as a trusted cert if it contains a chain. We will need the server certificate without a chain (i.e remove the CA cert.) Note we could import the client cert **jsmith.cer** because the keystore contains its private key and we imported into an existing entry and replaced the self-signed cert in it with the **jsmith** cert + CA cert. However as we do not have the server's private key we cannot do the same. (Other alternatives are also possible such as creating a pkcs12 keystore, or obtain the signer certificate and importing it as a trusted cert. However the process given in this document will be applicable to both Java and Dotnet developers on Windows platforms.)

The way around this is to first import the eMedNY Server certificates (servercert + cacert) into the Windows Certificate Store. Then export only the servercert from the Windows Certificate Store and then import the servercert into the keystore as a trusted cert.

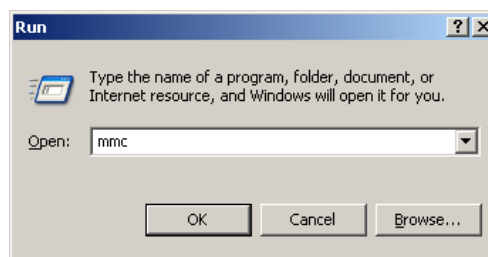
There are two ways of doing this, the first giving a more detailed view into the Windows Certificate Store, while the second is less granular but simpler to use.

- Using a Microsoft Management Console (mmc) for the Windows Certificate Stores
- Using the Internet Explorer Certificates Wizard.

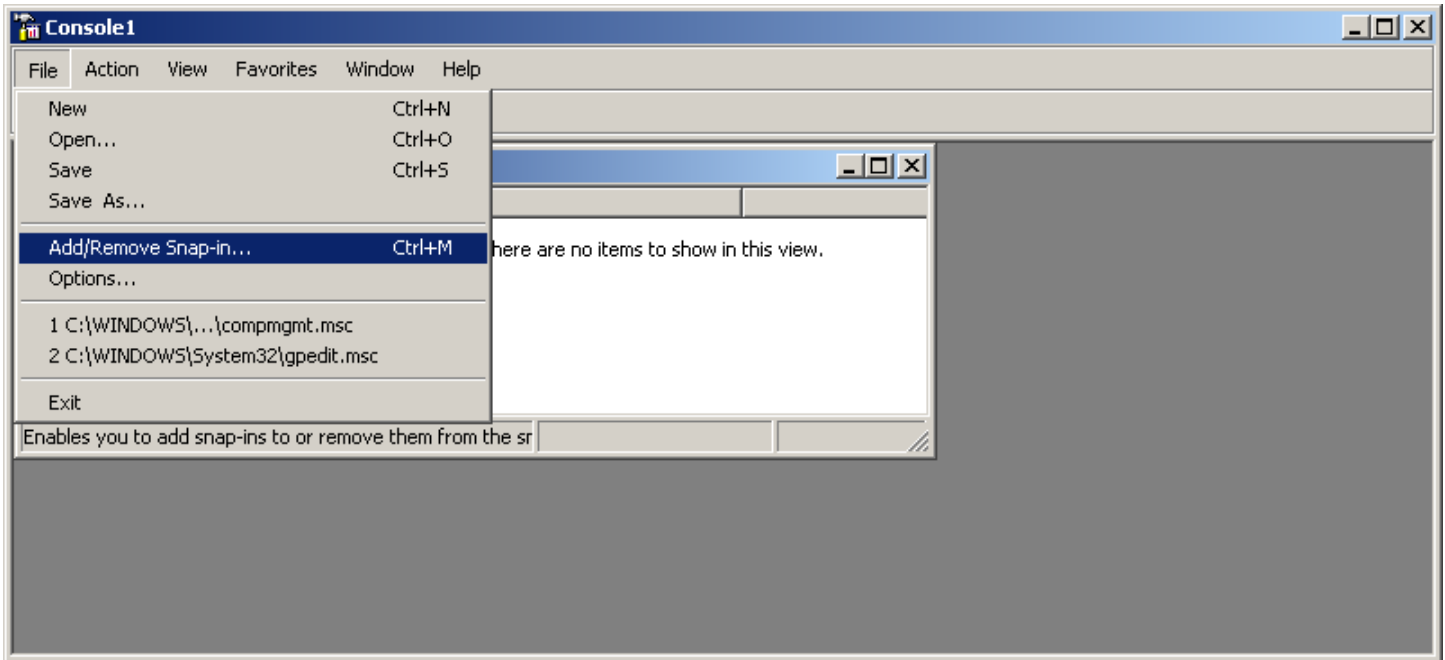
3.3 Importing certificates using a MMC.

First create a Microsoft Management Console (mmc) for the Certificate Store by doing the following:

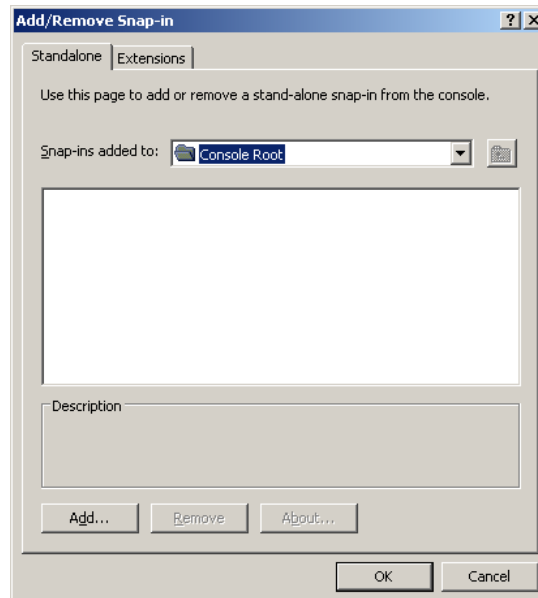
3.3.1 From the Windows task bar, Start > Run > mmc



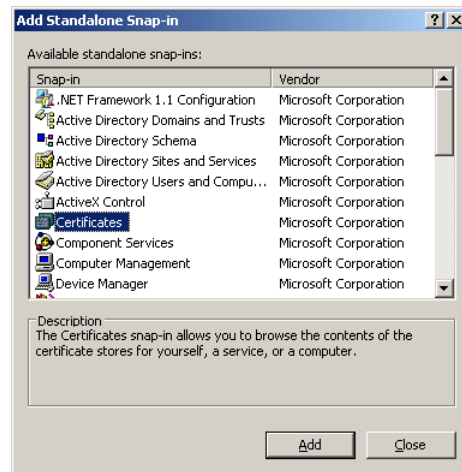
3.3.2 File > Add/Remove Snap-in



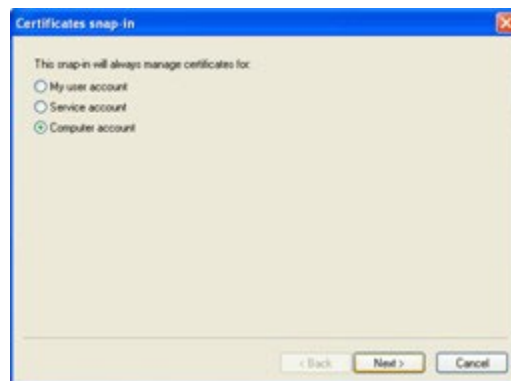
3.3.3 Click "Add..."



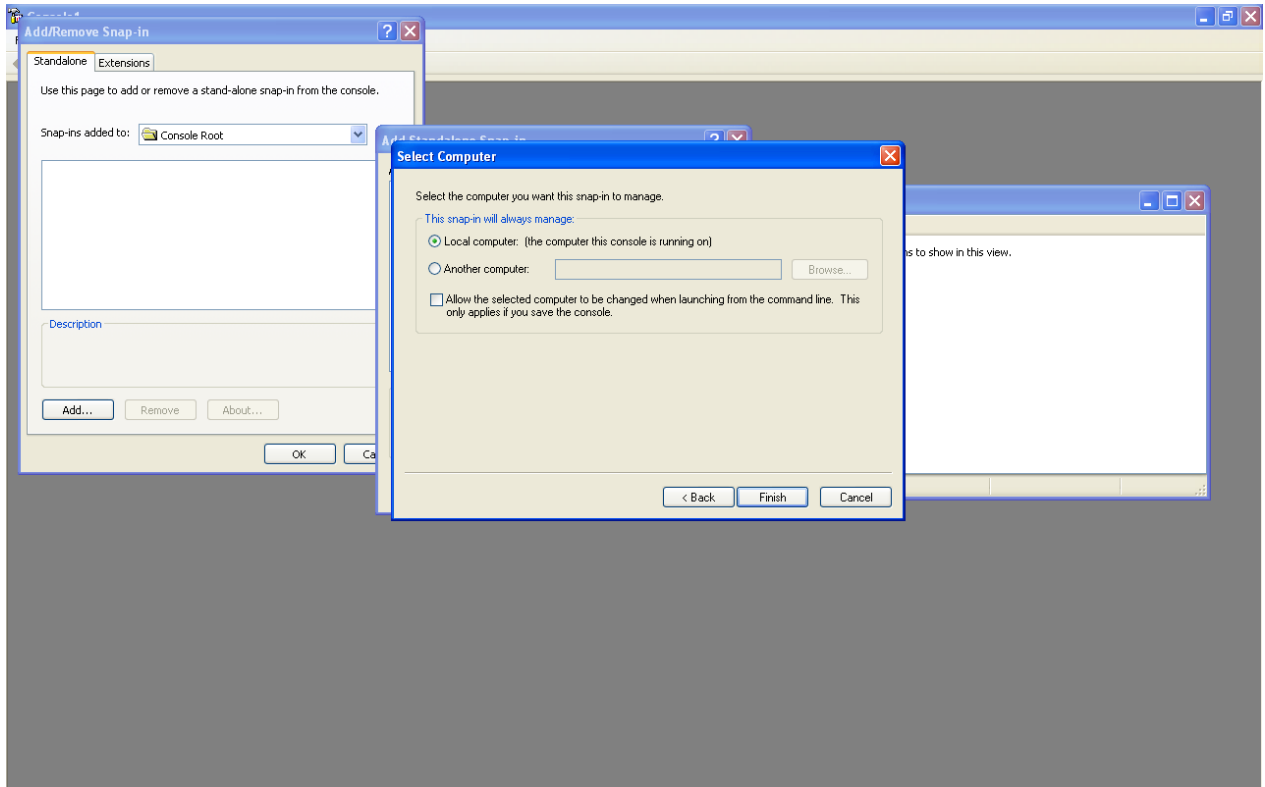
3.3.4 In the "Add Standalone Snap-in", select "Certificates" and Click "Add"



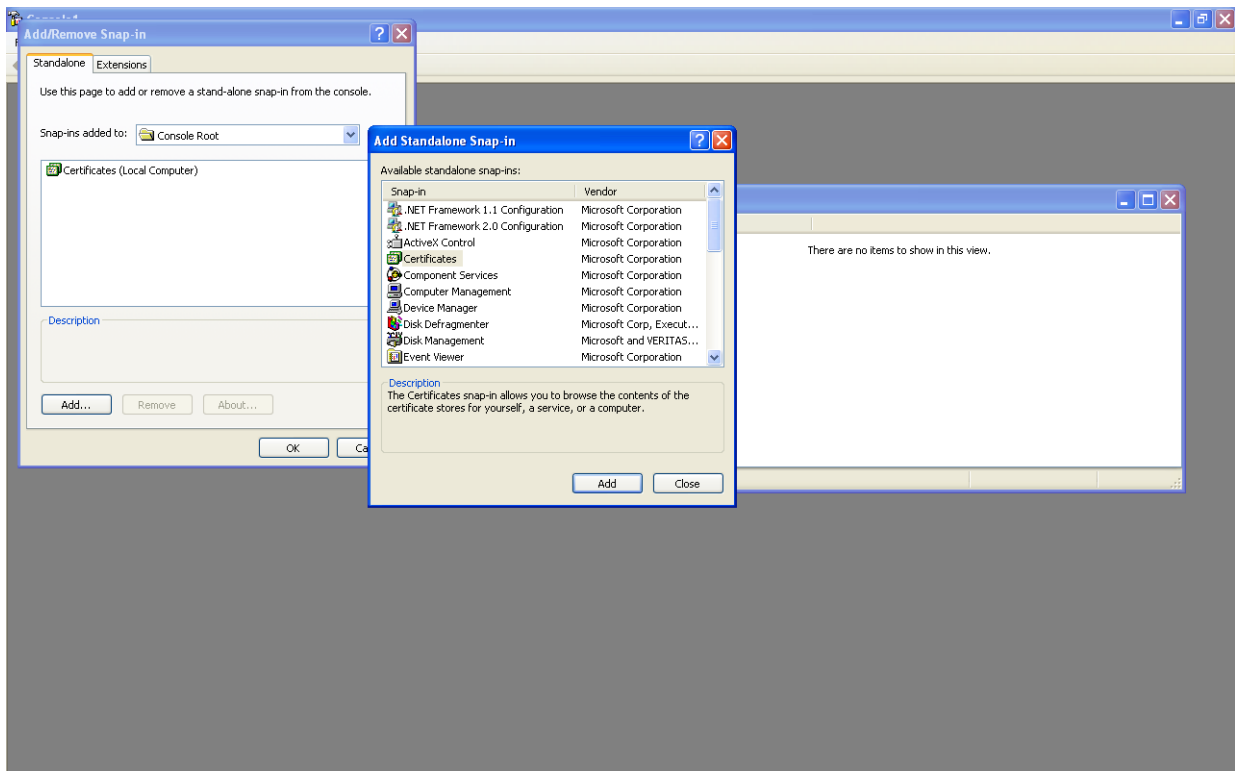
3.3.5 Select “Computer account”, then click “Next”



3.3.6 Select “Local computer (the computer this console is running on)”, and click Finish

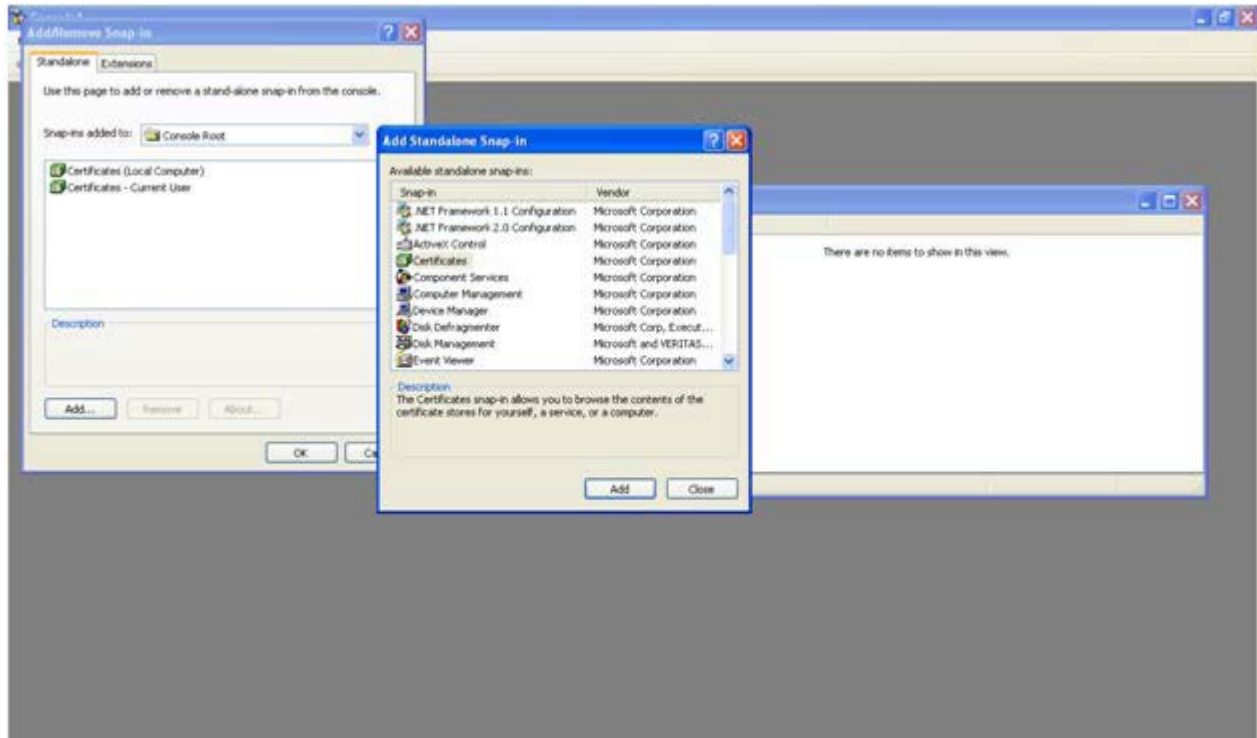


3.3.7 Note that the “Certificates (Local Computer)” has been added to the “Add/Remove Snap-in”

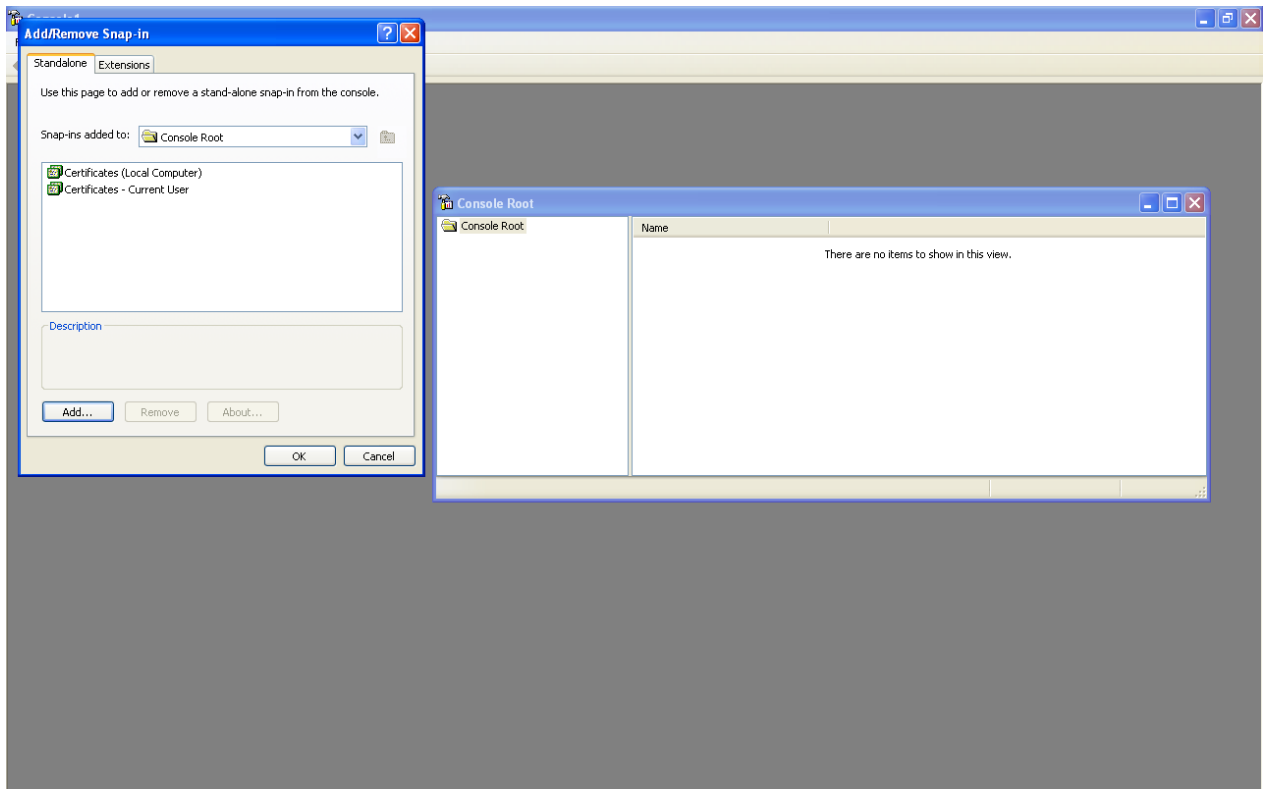


Close the “Add Standalone Snap in”.

3.3.8 In the “Add/Remove Snap-in” click on the “Add..” again and repeat steps 3.3.4 and 3.3.5, but this time for step 3.3.5 select the “My user account”.



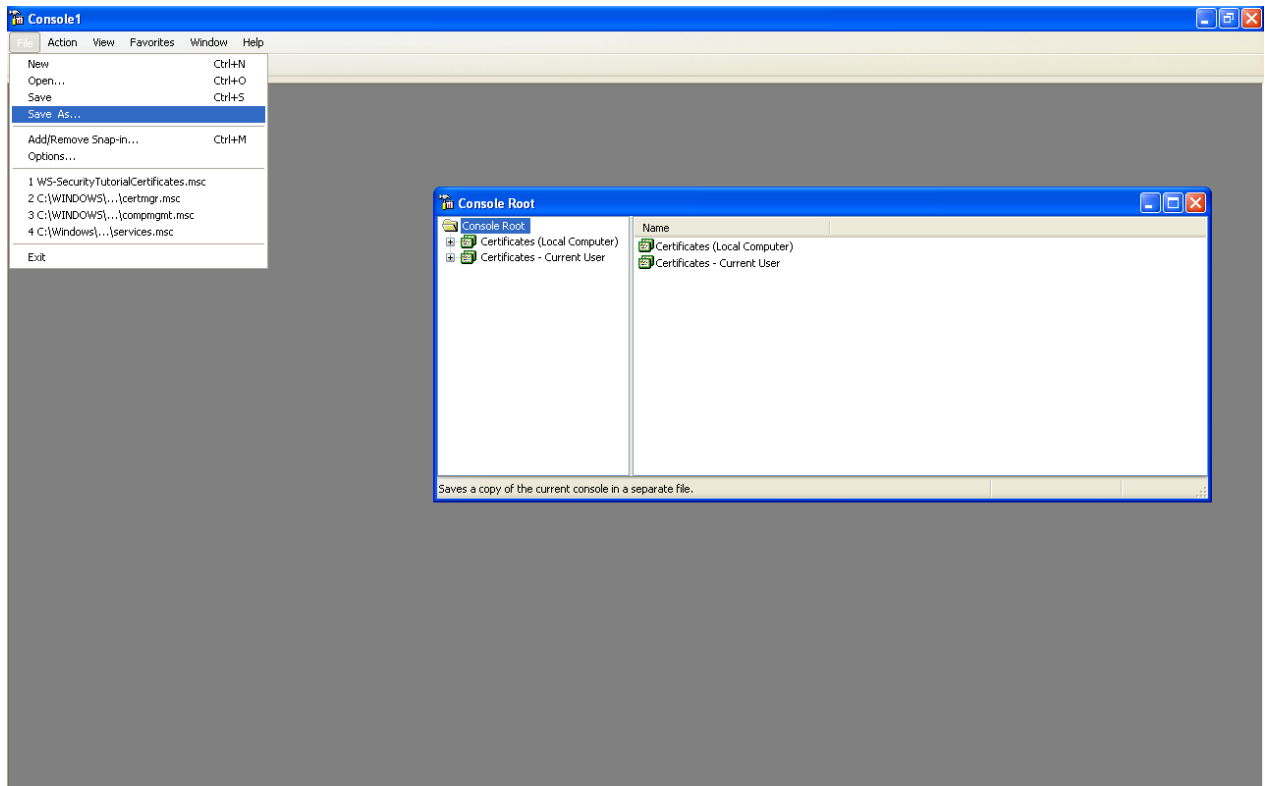
Then Close the “Add Standalone Snap-in”



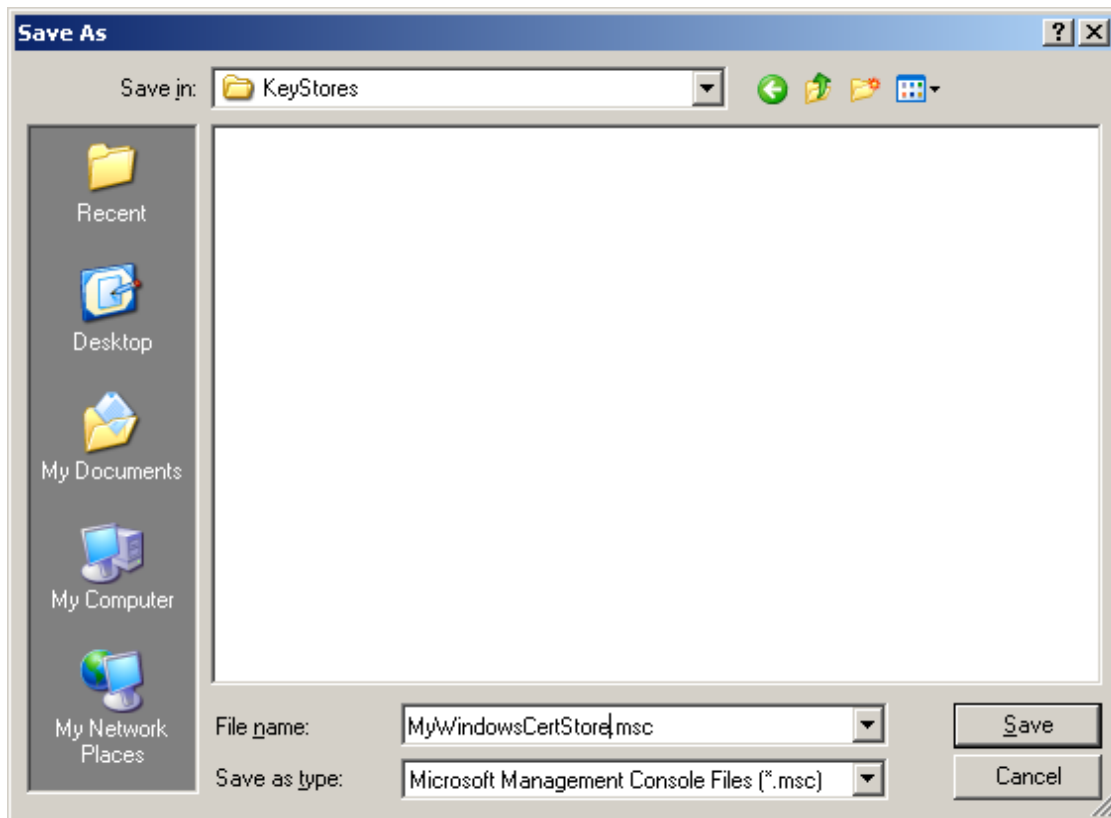
Note that “Certificates – Current User” has also been added to the “Add/Remove Snap-in”

Click on the OK button

3.3.8 Save the mmc console

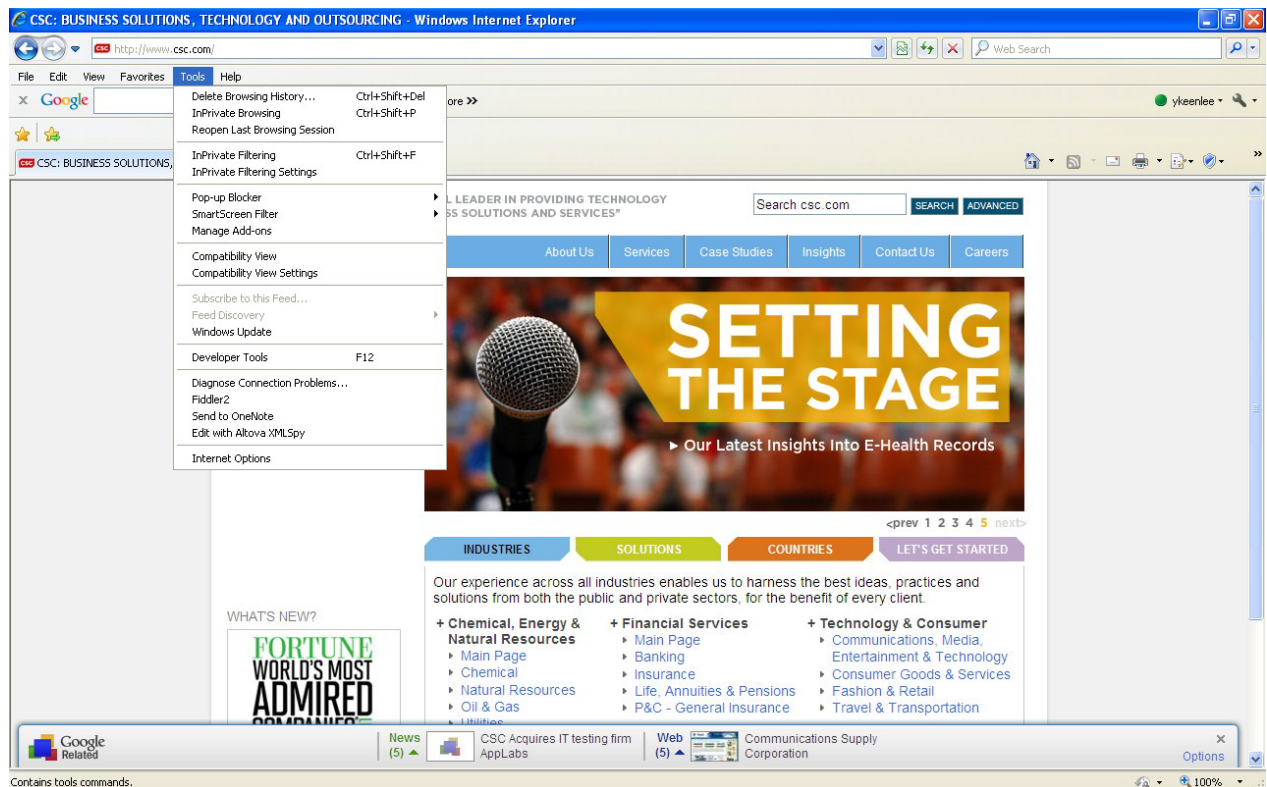


For example, in the **Keystores** folder as **“MyWindowsCertStore”**

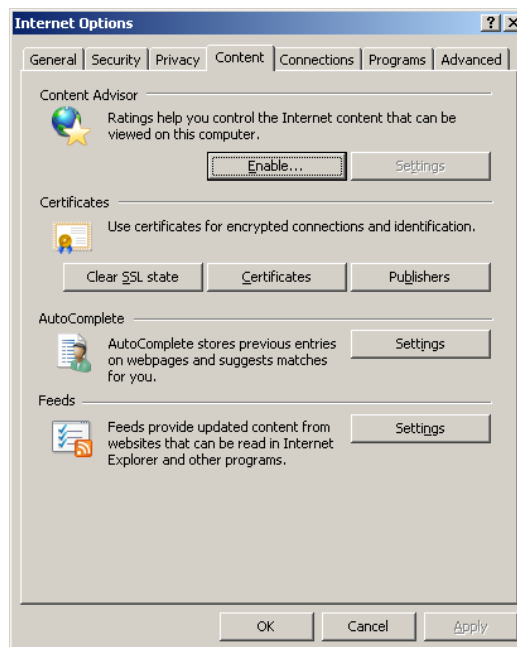


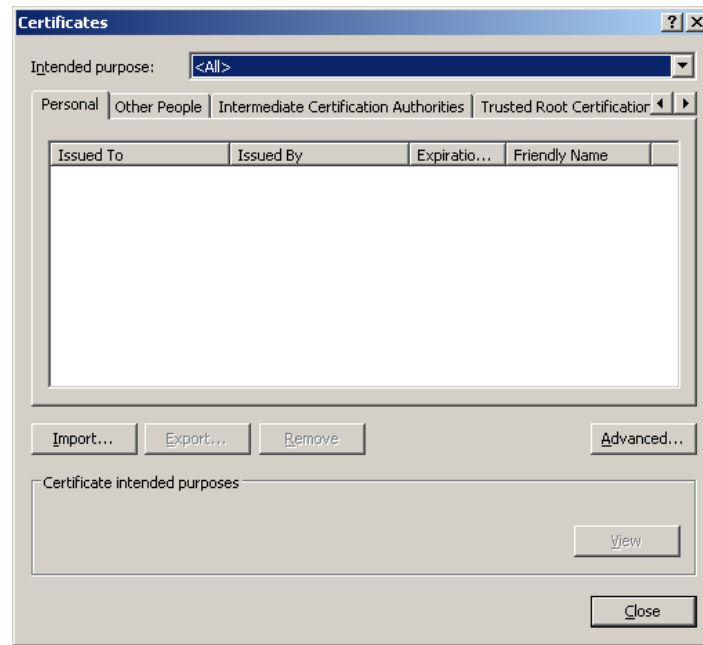
3.4 Importing Certificates Using the IE Certificates Wizard

Open Internet Explorer go to the Tools > Internet Options menu



Go to the Contents tab and click on the Certificates button



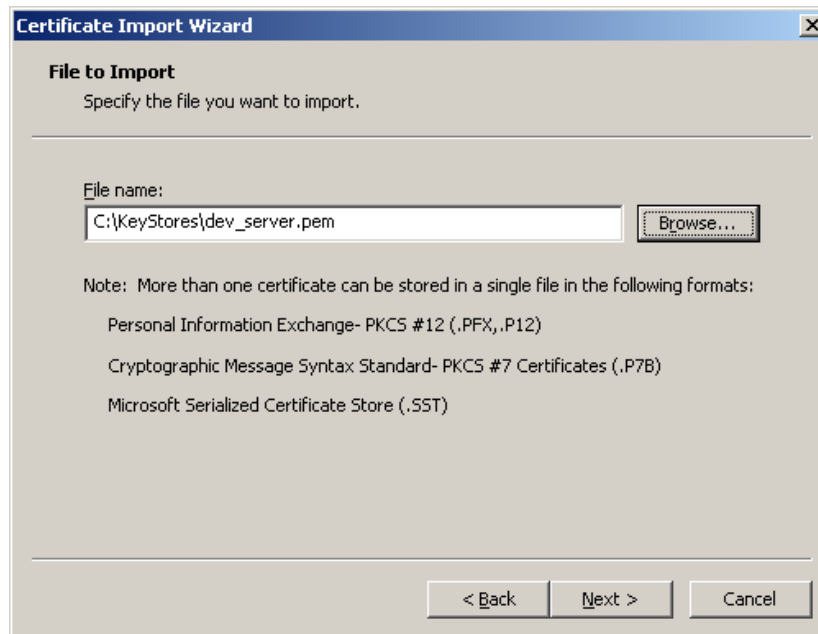


Click on Import...



Click the Next button

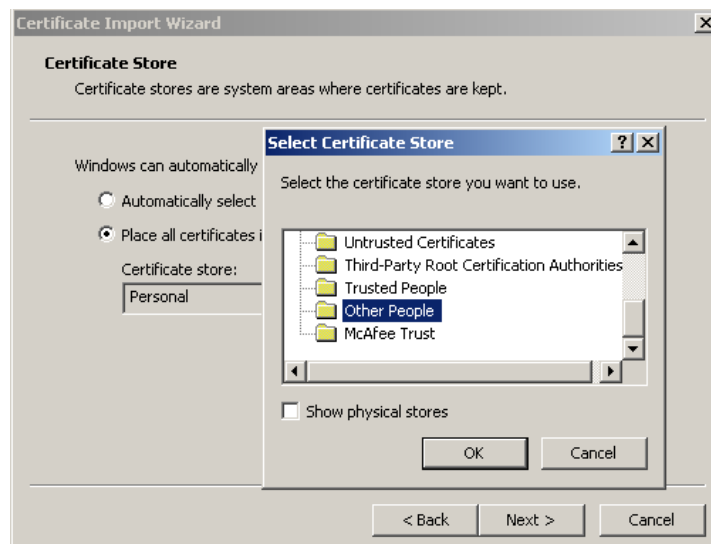
Navigate to the **dev_server.pem** file which you downloaded from eMedNY

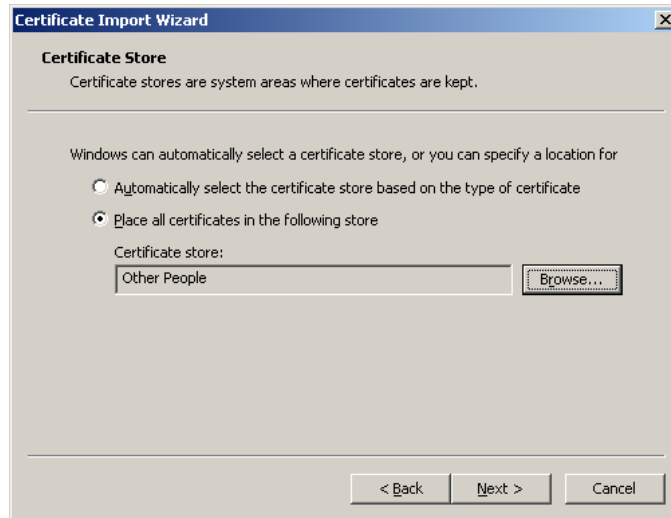


Click Next

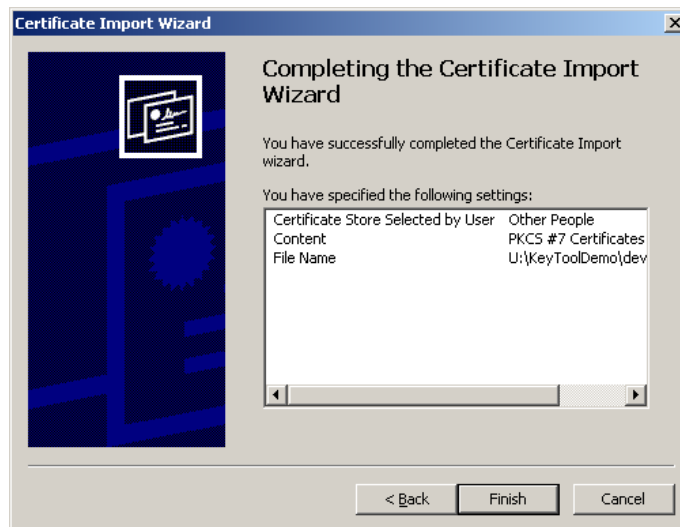
Notice it says Personal store. Click on Browse, select the "Other People" store. (Note: you can also put it in the "Personal" store, but as it is a server certificate, it might be more appropriate to put it in the "Other People" store and leave the "Personal" store for your own certificates. Also note that these are put into what is known as the "Certificates – Current User" stores. There is also a "Certificates (Local Computer)" stores which can also be used.)

And click OK



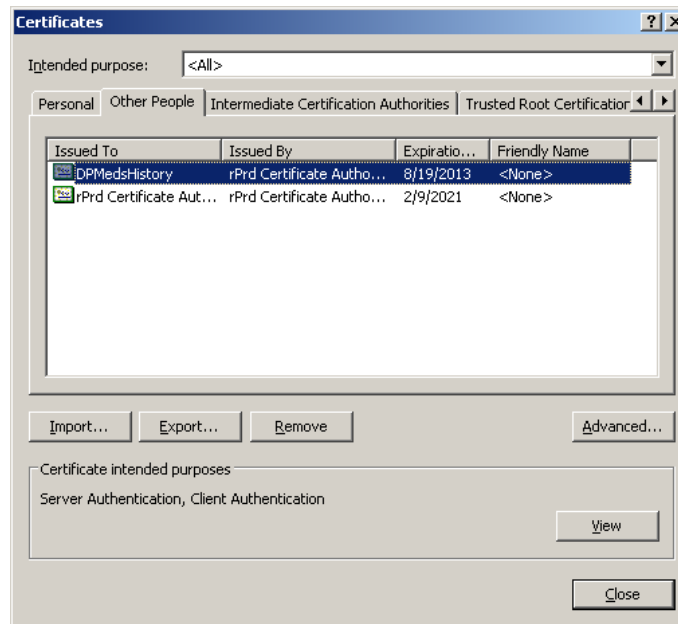


Click Next



Click Finish. Note the import was successful and click OK.

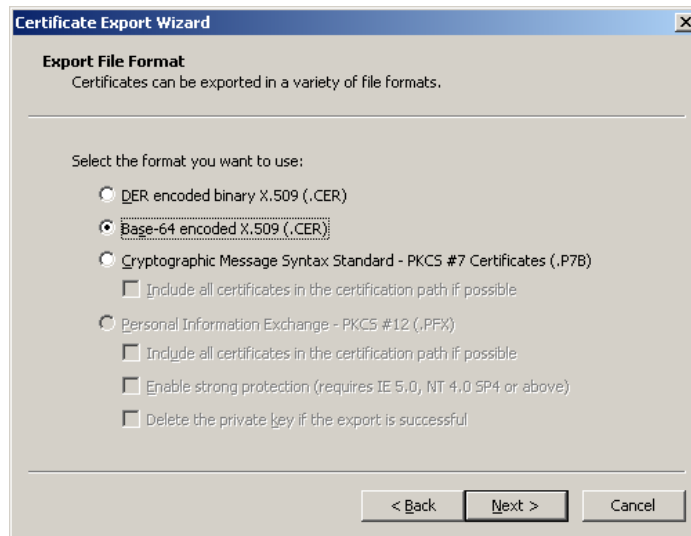
Click on "Other People" tab. The **DPMedsHistory** (which is the name of servercert) issued by "rPrd Certificate Authority" (the eMedNY CA) is now in the Windows Certificate Store. Note also that the CA cert in the chain, **rPrd Certificate Authority** is also imported. Click on the **DPMedsHistory** entry to select it. The Export button is now enabled.



Click on the Export... button, and then Next

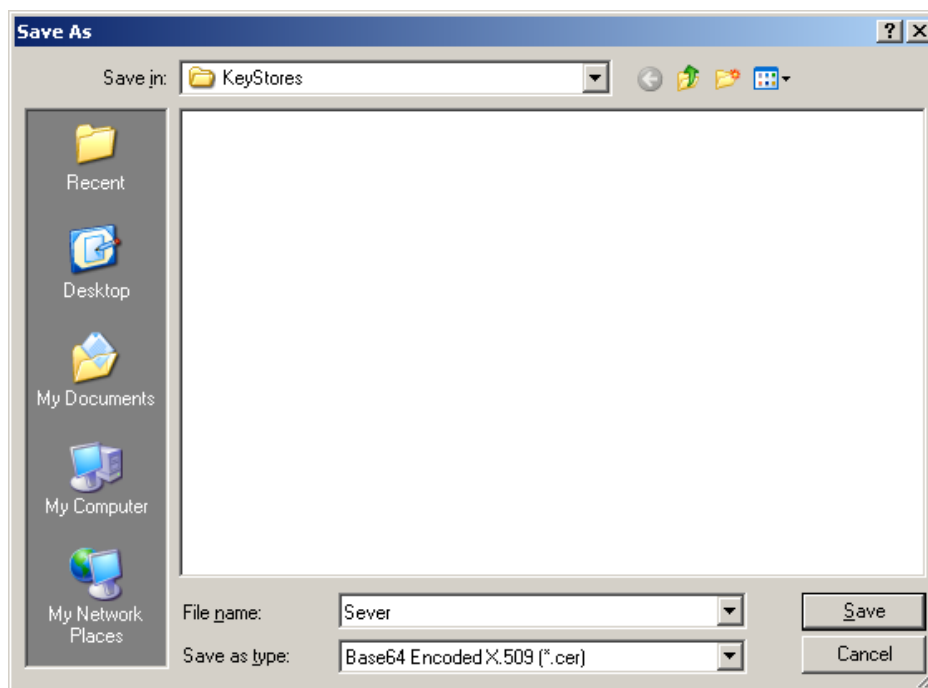


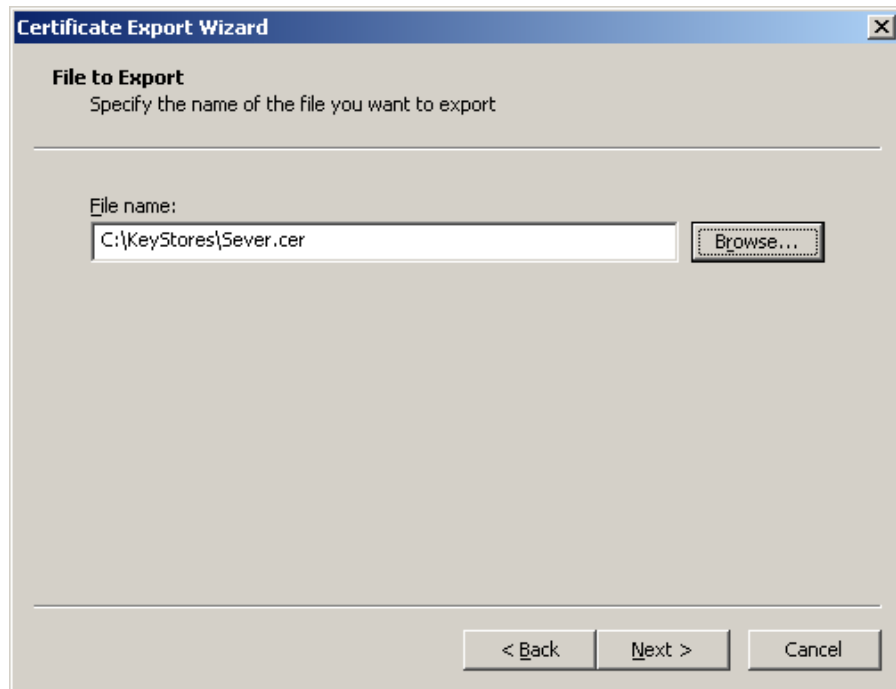
And select the base-64 encoded X.509 radio button,



Click Next

Navigate to the folder you created for your keystores (in our example "KeStores") folder and give the exported file a name (for example "server" here) with extension ".cer" and Save it.



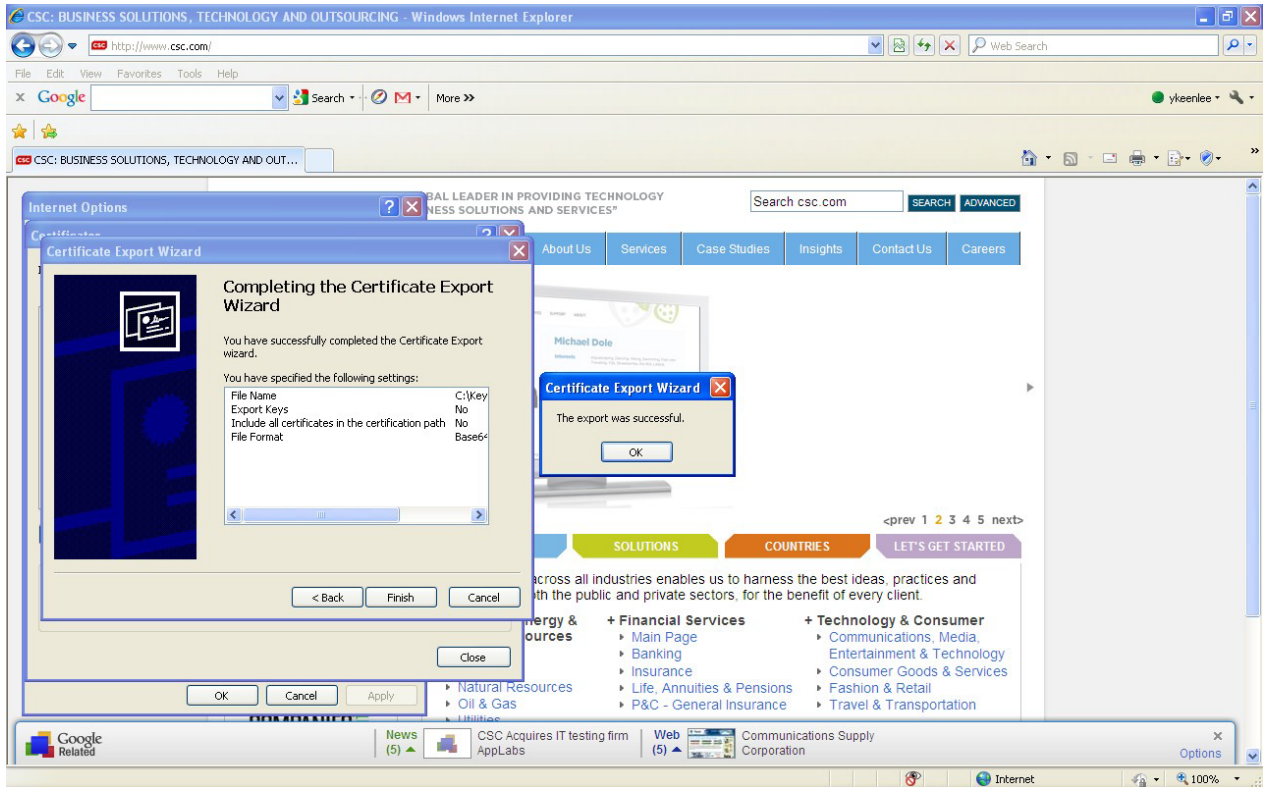


Click Next

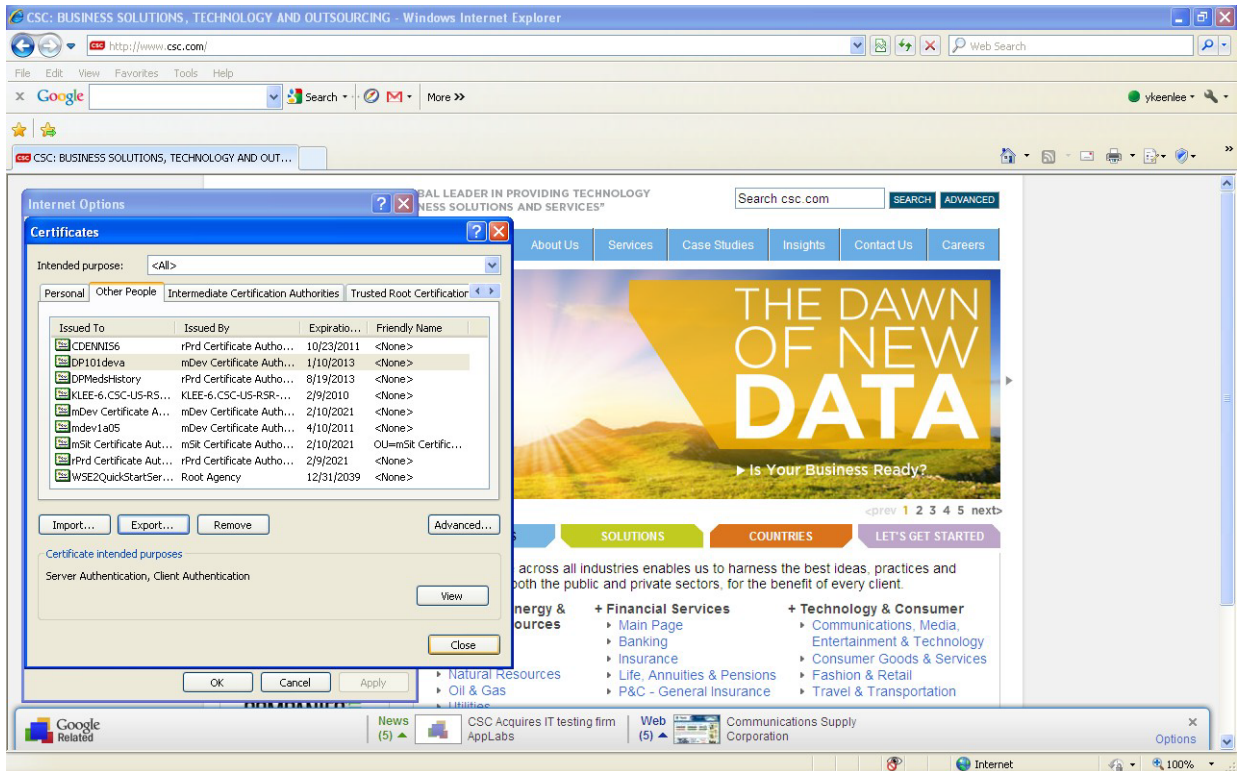


Note that the Keys (i.e private keys if any) are not exported and that all certificates in the certification path are not included (i.e. the CA cert is not included,) which is what we wanted from this digression using the Windows Certificate Store tool.

Click Finish and note that the export was successful



Click OK



Click Close and then OK.

So now we have the “**server.cer**” exported into the **KeyToolDemo** folder

3.5 Importing the Server Certificate into the Keystore

We can now proceed to import the server certificate into the keystore.

Open the command prompt and navigate to the folder you created for your Keystore. In our example, we named it “Keystores.”

Type the following command line:

```
keytool -importcert -v -alias [Server Cert Alias] -file [Cert file Name].pem -keystore [Keystore name].jks -storepass [keystore password] -keypass [client password]
```

The following is a sample command:

```
keytool -importcert -v -alias Server -file Server.cer -keystore JDOEKeystores.jks -storepass Password1 -keypass Password2
```

This image follows the sample:

```

C:\WINDOWS\system32\cmd.exe
C:\KeyStores>keytool -importcert -v -alias JDOE -file JDOE.pem -keystore JDOEKey
stores.jks -storepass Password1 -keypass Password2
keytool error: java.lang.Exception: Certificate reply does not contain public ke
y for <JDOE>
java.lang.Exception: Certificate reply does not contain public key for <JDOE>
    at sun.security.tools.KeyTool.validateReply(Unknown Source)
    at sun.security.tools.KeyTool.installReply(Unknown Source)
    at sun.security.tools.KeyTool.doCommands(Unknown Source)
    at sun.security.tools.KeyTool.run(Unknown Source)
    at sun.security.tools.KeyTool.main(Unknown Source)

C:\KeyStores>keytool -importcert -v -alias Server -file Server.cer -keystore JDO
EKeystores.jks -storepass Password1 -keypass Password2
Owner: CN=DPMedHistory, OU=eServers, OU=ePaces, OU=eMedNY-PROD, O=eMedNY
Issuer: OU=rPrd Certificate Authority, O=eMedNY
Serial number: f
Valid from: Tue Feb 16 00:00:00 EST 2010 until: Mon Aug 19 23:59:59 EDT 2013
Certificate fingerprints:
    MD5: 05:52:94:C1:75:3C:A1:68:BB:48:D8:BD:08:F1:AF:E2
    SHA1: C7:B7:ED:92:61:AB:63:31:D2:AC:71:2F:A8:55:D3:60:F4:A9:BD:6D
    Signature algorithm name: SHA1withRSA
    Version: 3

Extensions:

#1: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
    DigitalSignature
    Non_repudiation
    Key_Encipherment
    Data_Encipherment
]

#2: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: B3 AD 22 1E 93 1C FC 9C   F6 17 89 6D FE 51 E7 2D   ..".....m.Q.-
0010: 5B 59 72 8D                               [Yr.
]
]

#3: ObjectId: 2.5.29.31 Criticality=false
CRLDistributionPoints [
    DistributionPoint:
        [CN=CRL1, OU=rPrd Certificate Authority, O=eMedNY]
    , DistributionPoint:
        [URName: http://www.emedny.org/ePaces/cacerts/CRL1.crl]
]

#4: ObjectId: 2.5.29.37 Criticality=false
ExtendedKeyUsages [
    serverAuth
    clientAuth
]

#5: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
0000: C1 BA 37 B5 74 45 72 4D   30 37 98 36 0C F4 BE FF   ..7.tErM07.6....
0010: EC 55 1E 74                               .U.t
]
]

Trust this certificate? [no]: yes
Certificate was added to keystore
[Storing JDOEKeystores.jks]

C:\KeyStores>

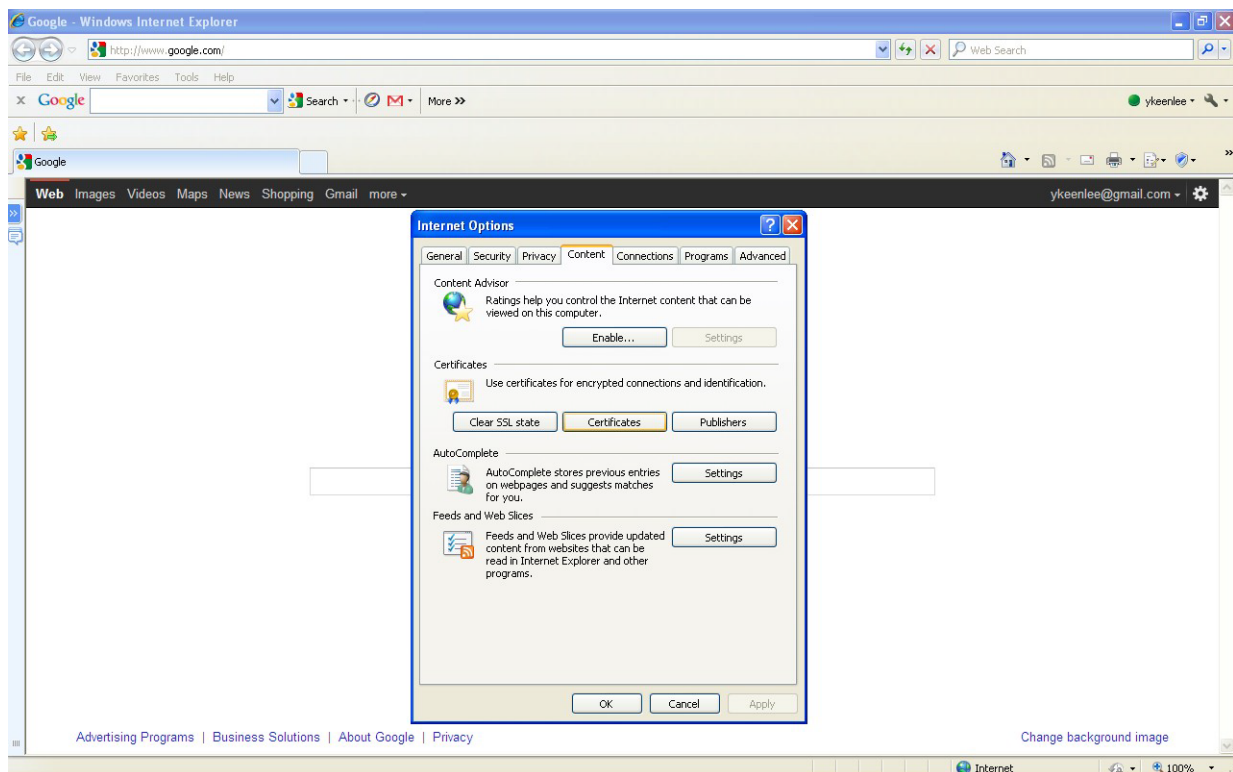
```

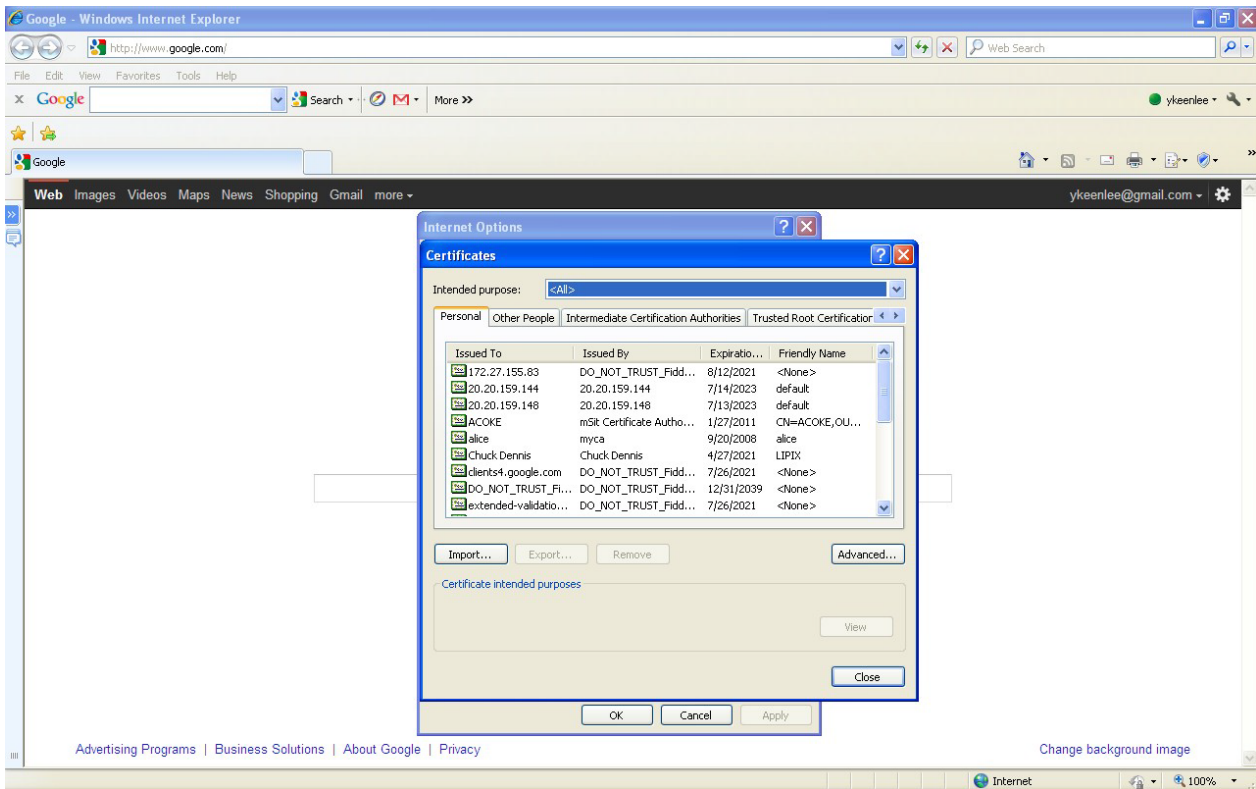
3.6 Importing private key and certificates from Java to Windows Key Stores

If you have a Dotnet application, you will not be able to use the java **JDOEKeystores.jks** so you will have to export the private key and cert of **JDOE** into the Windows Certificate Store. You have already imported the server certificate into that store.

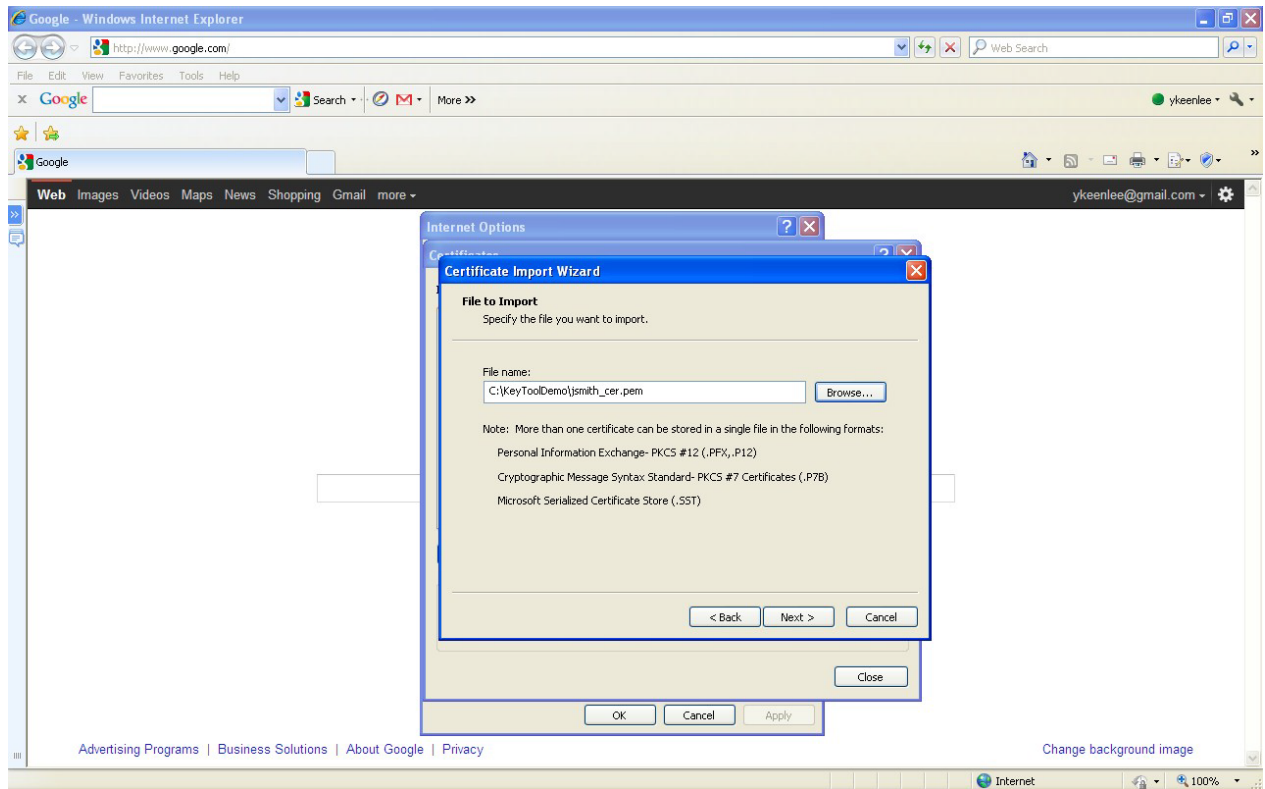
The procedure for doing the **JDOE.pem** is similar.

Open your IE browser, From the menu, go to Tools > Internet options > Content tab and click on Certificates button.

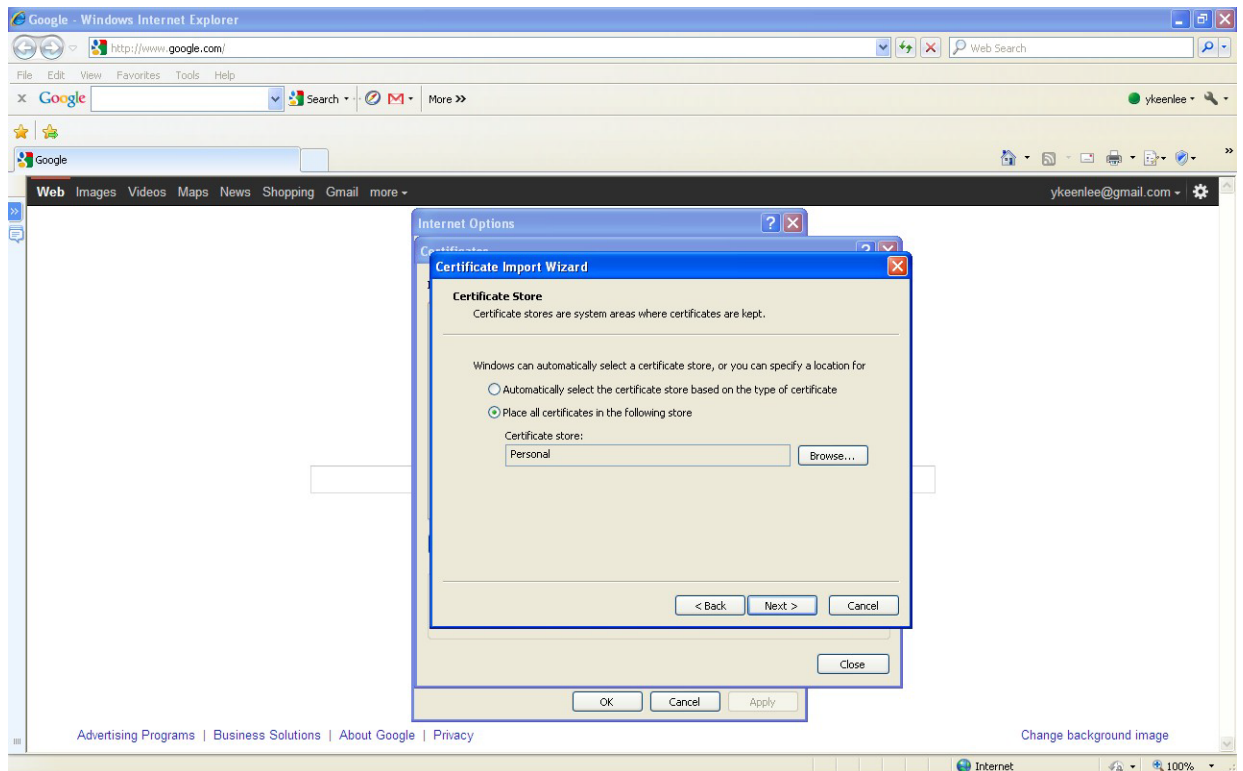




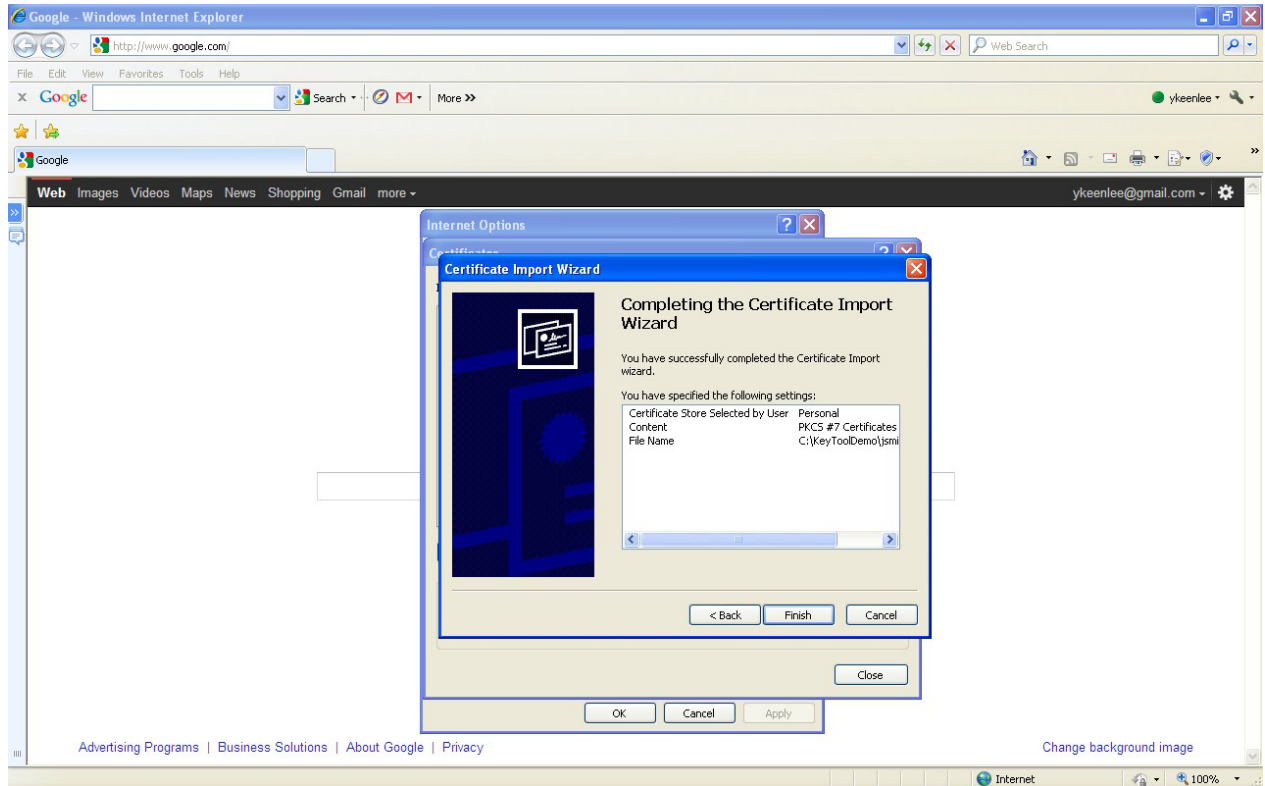
Click Import... , Next and select your **JDOE.pem** file that was returned with your csr request.



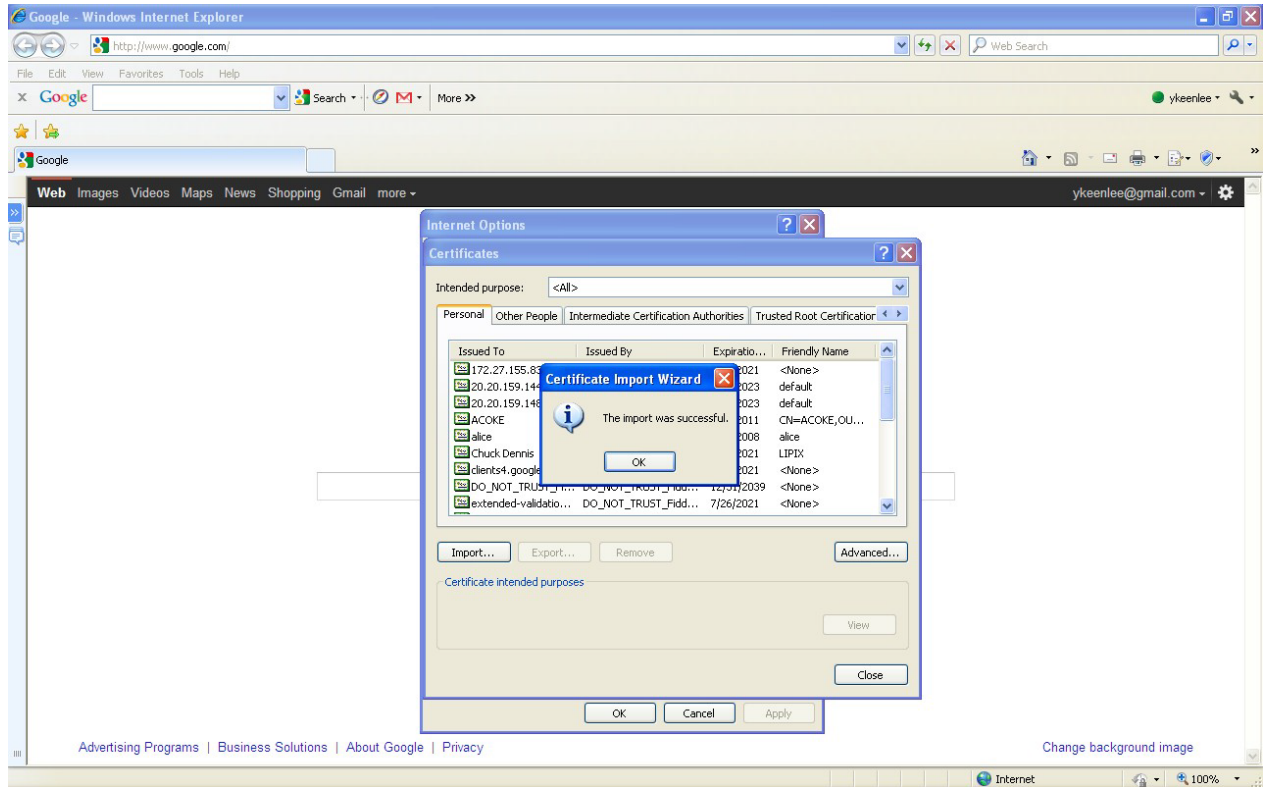
Click Next



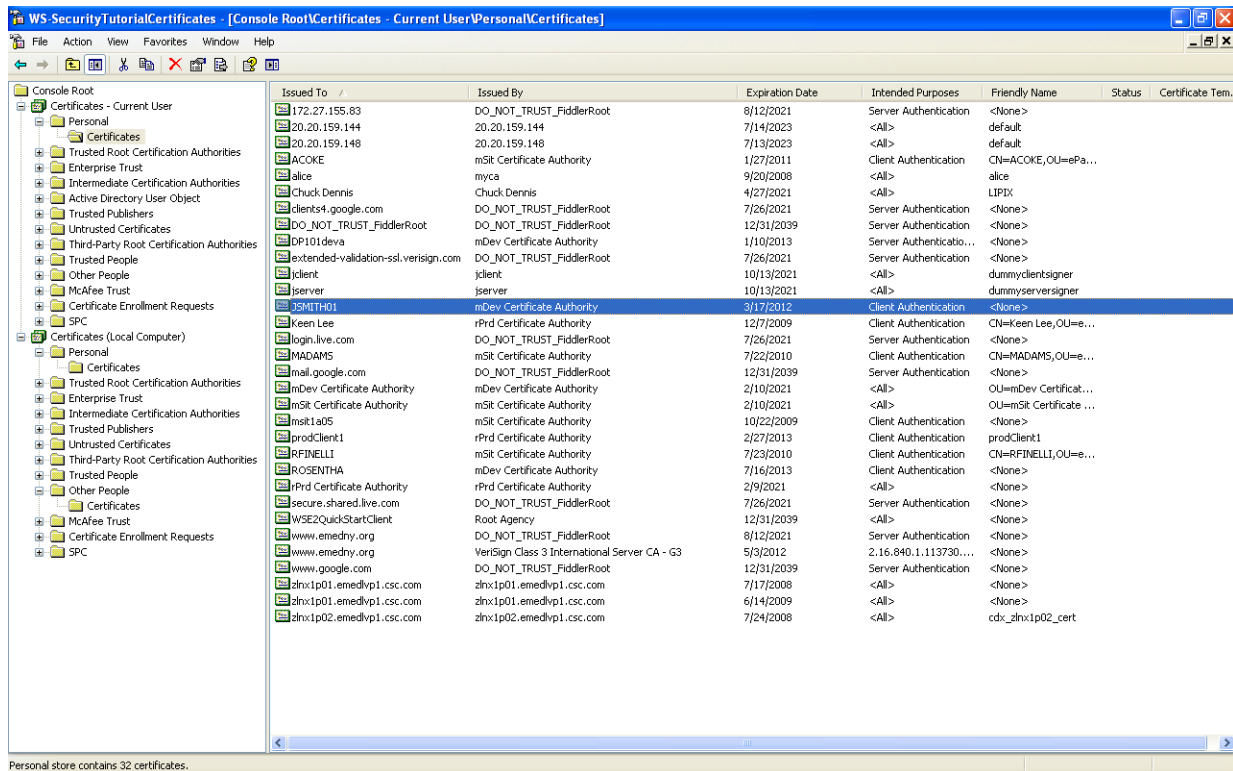
And place the certificate in the "Personal" store. Click Next



And Finish



You will need to create a mmc certificates to view cert for **JDOE** imported into CurrentUser/Personal/Certificates. The Windows Cert Store only shows CurrentComputer stores.



4 Additional Tools and Information

4.1 keytool web link

<http://java.sun.com/j2se/1.3/docs/tooldocs/win32/keytool.html>

4.2 Requirements for CORE Compliance

<http://www.caqh.org/benefits.php>

4.3 JSSE Reference Guide

<http://docs.oracle.com/javase/6/docs/technotes/guides/security/jsse/JSSERefGuide.html#SSLOverview>

4.4 WCF – 2 Way SSL using Certificates

<http://blogs.msdn.com/b/imayak/archive/2008/09/12/wcf-2-way-ssl-security-using-certificates.aspx>

eMedNY neither endorses nor recommends any of the tools linked or referenced in this document. The intent here is strictly informational.



eMedNY is the name of the electronic New York State Medicaid system. The eMedNY system allows New York Medicaid providers to submit claims and receive payments for Medicaid-covered services provided to eligible clients.

eMedNY offers several innovative technical and architectural features, facilitating the adjudication and payment of claims and providing extensive support and convenience for its users.

The information contained within this document was created in concert by eMedNY and DOH. More information about eMedNY can be found at www.emedny.org.