X.509 certificates www.ecologic.co.in

How & Why ?

Using them ..!~

By Training Department

Ecologic Corporation

www.ecologic.co.in

Chandigarh .India

Little BackGround

- A public key certificate, usually just called a digital certificate or certs is a digitally signed document that is commonly used for authentication and secure exchange of information on open networks, such as the Internet, extranets, and intranets. A certificate securely binds a public key to the entity that holds the corresponding private key. Certificates are digitally signed by the issuing certification authority (CA) and can be issued for a user, a computer, or a service. This creates a trust relationship between two unknown entities. The CA is the Grand Pooh-bah of Validation in an organization, which everyone trusts, and in some public key environments, no certificate is considered valid unless it has been attested to by a CA. Example of a popular CA's authority is http://www.verisign.com/ .

Uses of Digital Certificates

- Web user authentication,
- Web server authentication,
- Secure e-mail (Secure/Multipurpose Internet Mail Extensions, or S/MIME),
- Internet Protocol security (IPSec),
- Transport Layer Security (TLS), and code signing.

What X.509 includes:

- An X.509 certificate includes the public key and information about the person or entity to whom the certificate is issued, information about the certificate, plus optional information about the certification authority (CA) issuing the certificat

Where to get a Digital Certificate

- **Class 1 Digital Certificates for individuals are intended for email and can be used for digitally signing documents**

 # Verisign
http://www.verisign.com/products-services/security-services/pki/pki-application/email-digital-id/index.html

 # Thawte
http://www.thawte.com/secure-email/personal-email-certificates/index.html

 # Globalsign
http://www.globalsign.net/digital_certificate/personalsign/index.cfm

 # Comodogroup
http://www.comodogroup.com/products/certificate_services/email_certificate.html

Formats of X.509 Cerficates

- Formats of X.509 Certificates:
- DER Encoded Binary X.509 (.cer)
- Base64 Encoded X.509 (.cer)
- PKCS#7 / Cryptographic Message Syntax Standard (.p7b)
- PKCS#12 / Personal Information Exchange (.pfx)

DER Encoded Binary X.509

- DER (Distinguished Encoding Rules) for ASN.1, as defined in ITU-T Recommendation X.509, is a more restrictive encoding standard than the alternative BER (Basic Encoding Rules) for ASN.1, as defined in ITU-T Recommendation X.209, upon which DER is based. Both BER and DER provide a platform-independent method of encoding objects (such as certificates and messages) for transmission between devices and applications. During certificate encoding, most applications use DER because a portion of the certificate (the Certification Request's Certification Request Info) must be DER-encoded to be signed. This format might be used by certification authorities that are not on Windows 2000 servers, so it is supported for interoperability. DER certificate files use the .cer extension.

Base64 Encoded X.509

- This is an encoding method developed for use with Secure/Multipurpose Internet Mail Extensions (S/MIME) which is a popular, standard method for transferring binary attachments over the Internet. Base64 encodes files into ASCII text format, making corruption less likely as the files are sent through Internet gateways, while S/MIME provides some cryptographic security services for electronic messaging applications, including non-repudiation of origin using digital signatures, privacy and data security using encryption, authentication, and message integrity. The MIME (Multipurpose Internet Mail Extensions) specification (RFC1341 and successors) defines a mechanism for encoding arbitrary binary information for transmission by electronic mail. Because all MIME-compliant clients can decode Base64 files, this format might be used by certification authorities that are not on Windows2000 servers, so it is supported for interoperability. Base64 certificate files use the .cer extension.

Cryptographic Message Syntax Standard (PKCS#7)

- The PKCS#7 format enables the transfer of a certificate and all the certificates in its certification path from one computer to another or from a computer to removable media. PKCS#7 files typically use the .p7b extension and are compatible with the ITU-TX.509 standard. PKCS#7 allows for attributes such as countersignatures to be associated with signatures. Attributes such as signing time can be authenticated along with message content. For information on PKCS #7

Personal Information Exchange (PKCS#12)

- The Personal Information Exchange format (.pfx, also called PKCS#12) enables the transfer of certificates and their corresponding private keys from one computer to another or from a computer to removable media. PKCS#12 (Public Key Cryptography Standard#12) is an industry format that is suitable for transport or backup and restoration of a certificate and its associated private key. This can be between products from the same vendor or different vendors. To use the PKCS#12 format, the cryptographic service provider (CSP) must recognize the certificate and keys as exportable. If a certificate was issued from a Windows2000 certification authority, the private key for that certificate is only exportable if one of the following is true: The certificate is for EFS (encrypting file system) or EFS recovery. The certificate was requested through the Advanced Certificate Request certification authority Web page with the Mark keys as exportable check box selected. Because exporting a private key might expose it to unintended parties, the PKCS#12 format is the only format supported in WindowsXP for exporting a certificate and its associated private key

Main Certificates Properties

Field	Meaning
Version	Which version of X.509
Serial number	This number plus the CA s name uniquely identifies the certificate
Signature algorithm	The algorithm used to sign certificate
Issuer	X.500 name of CA
Validity Period	The starting and ending period
Subject name	The entity whose key being certified
Public Key	The subject Ãs public key and ID of algorithm using it.

Tools to make Certificates

- Permission and assembly management tools
- Certificate Management Tool

Certificate management tools

Application	Description
Makecert	Generate a X.509 certificate for testing purpose only
Certmgr	Assembles certificates into CTL (certificate trust list) and can also be used for revoking lists (CRLs).
Chktrust	Verifies the validity of a file signed with an X.509 certificate
Cert2spc	Creates, for test purposes only, a Software Publisher's Certificate (SPC) from one or more X.509 certificates

How to create Certificates

- DER based X.509 certificate (.cer)
- makecert -sk Ecologic -n "CN=Ecologic Company" Ecologic.cer
- -sk is Subject name
- -n is Specifies the subject's key container location, which contains the private key. If a key container does not exist, it will be created.
- This will create a X.509 certificate (Adhan.cer) in personal folder directory of user currently logged. Now we can retrieve its properties using System.Security.Cryptography.X509Certificates class. The Certificate is also included in source code.

Let us Test this certificate

- **cert2spc Ecologic.cer Ecologicspc.spc**

- This will create (Ecologicspc.spc) file in personal folder directory of user currently logged. However this is for test purposes only. You can obtain a valid SPC from a Certification Authority such as VeriSign or Thawte. The spc file is included in source code

How to : Chktrust : to checks the validity of a file signed with an Authenticode certificat

- **chktrust signedfile**
- The signed file could be any valid application (exe). If application does not have a valid
- signature, the tool displays the Security Warning dialog box. The dialog gives you the option
- to install and run the PE file even though an Authenticode signature could not be found.
- Similarly use Certmgr to manages certificates, certificate trust lists (CTLs), and certificate
- revocation lists (CRLs). The following command displays a default system store called my
- with verbose output (/v).

chktrust signedfile

- The signed file could be any valid application (exe). If application does not have a valid
- signature, the tool displays the Security Warning dialog box. The dialog gives you the option
- to install and run the PE file even though an Authenticode signature could not be found.
- Similarly use Certmgr to manages certificates, certificate trust lists (CTLs), and certificate
- revocation lists (CRLs). The following command displays a default system store called my
- with verbose output (/v).
-
- **certmgr /v /s my**

How to access X.509 properties in VB.NET

```
Imports System.Security.Cryptography.X509Certificates
Imports System.IO

Private Sub Button1_Click (ByVal sender As System.Object,
 ByVal e As System.EventArgs) Handles Button1.Click
 get certificate in Bin directory

 Dim Cert As X509Certificate = X509Certificate.CreateFromCertFile(
  Directory.GetCurrentDirectory & \Ecologic.cer)
 Now retrieve its properties in output window using ToString Mehtod. Since this
 'class it also have many other
 methods that do the same job

 Get the most important values in string format.

 Dim resultsTrue As String = Cert.ToString(True)
 Display the value in output window
 Debug.WriteLine(resultsTrue)

 Now display Serial number in bytes with GetSerialNumber() method
 Dim SerialNumber() As Byte = Cert.GetSerialNumber()

 Dim Sr As Byte
 Debug.Write(Serial Number in Bytes: )
 For Each Sr In SerialNumber
 Debug.Write(Sr)
 Next
End Sub
```

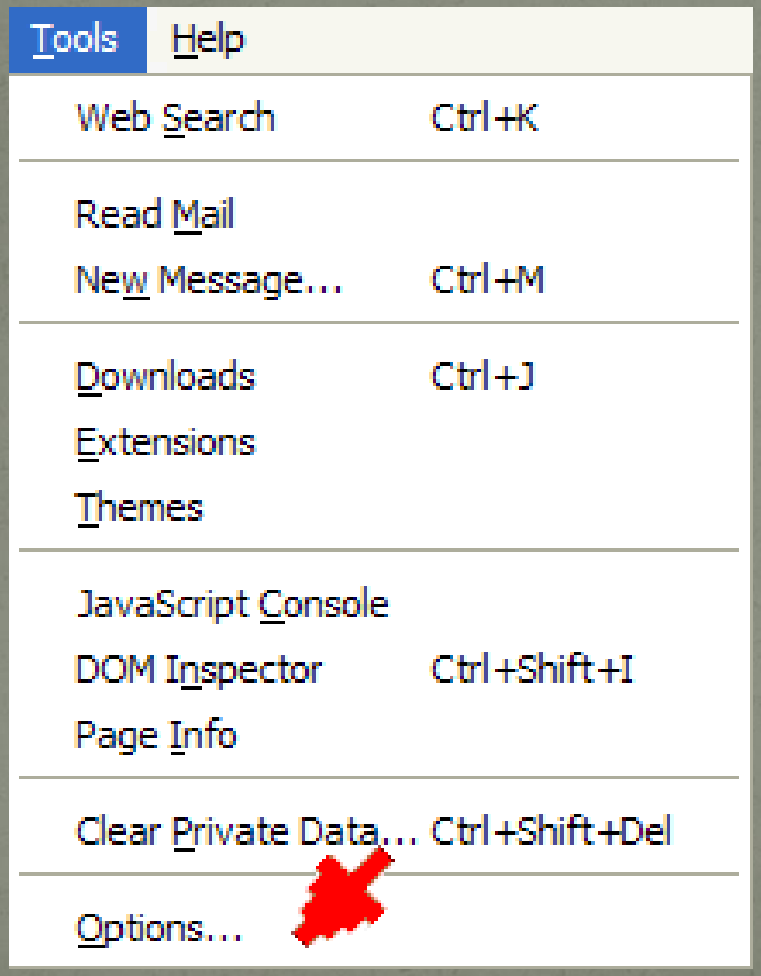
How to create PKCS #12 Certificates a self-signed certificate and the associated private key (.pvk): in .NET.

```
makecert -r -n "CN=TestProject" -b 20/04/2009 -e 01/01/2099 -eku 4.1.2.1.6.6.7.3.7 -sv TestProject.pvk testProject.cer
```

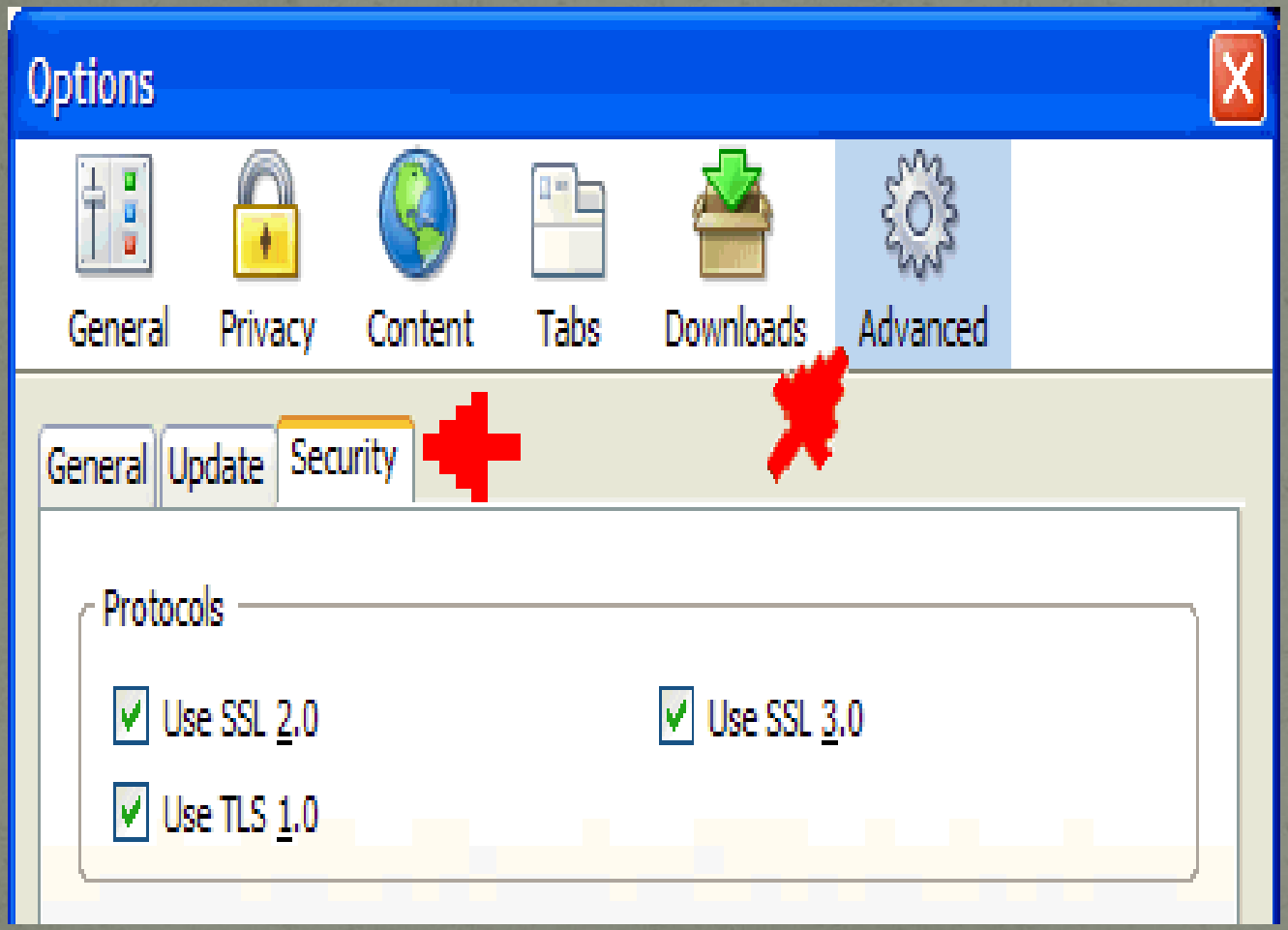
```
cert2spc Test Project.cer TestProject.spc
```

```
pvkimprt -pfx CodeProject.spc CodeProject.pvk
```

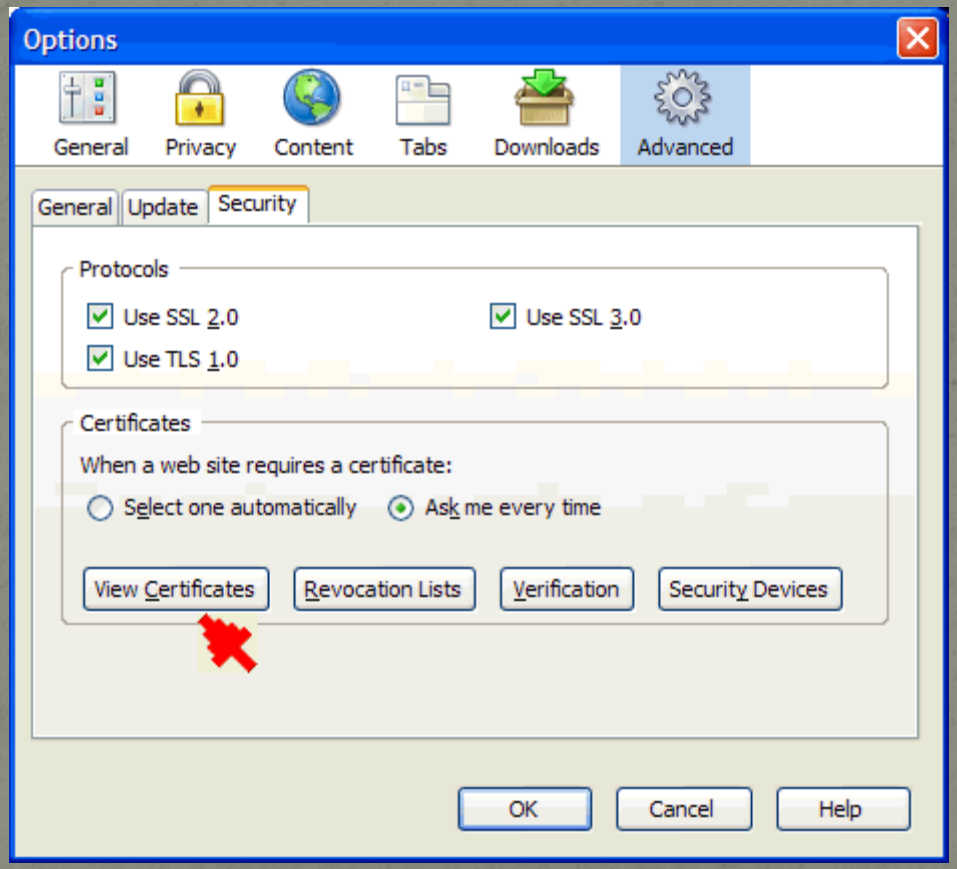
How to Import an Internet Explorer PFX File to Firefox



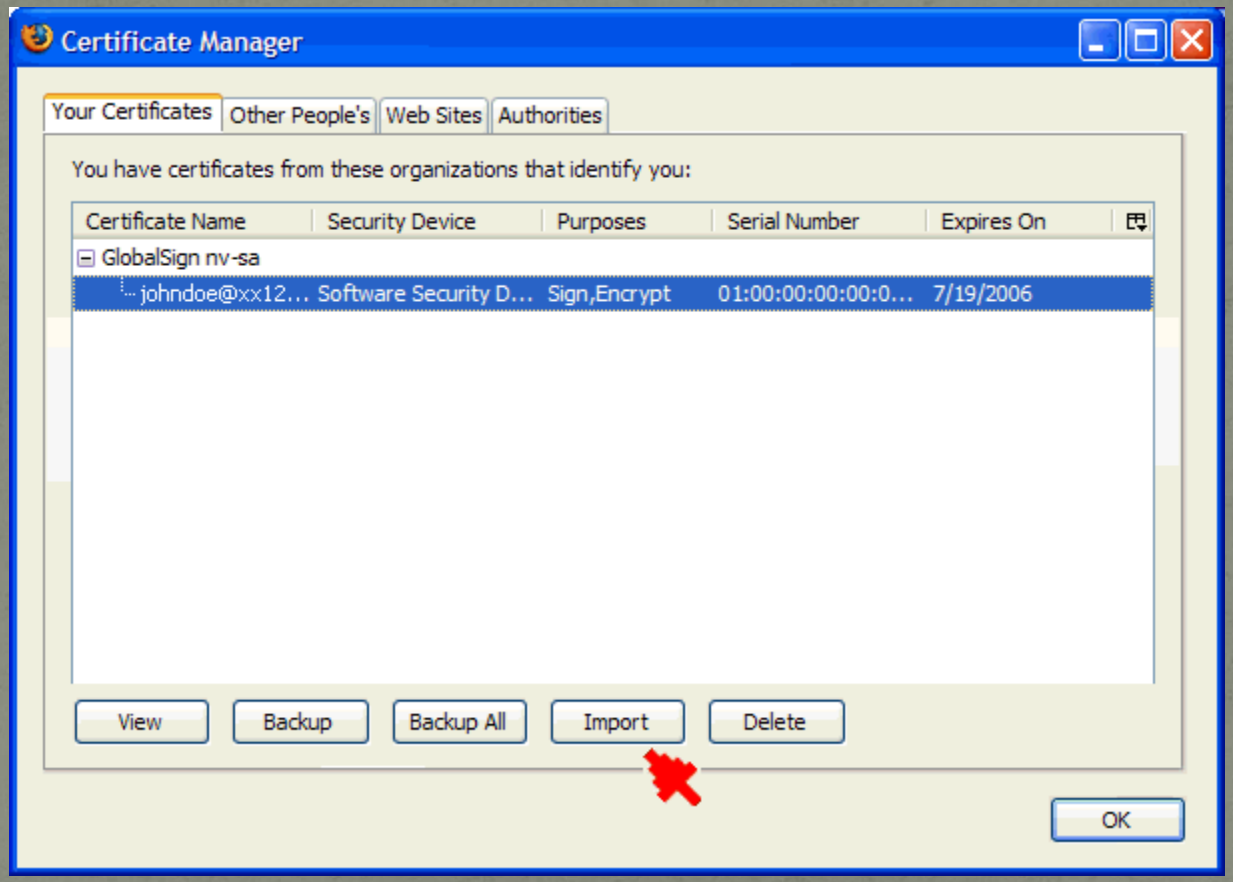
How to Import an Internet Explorer PFX File to Firefox



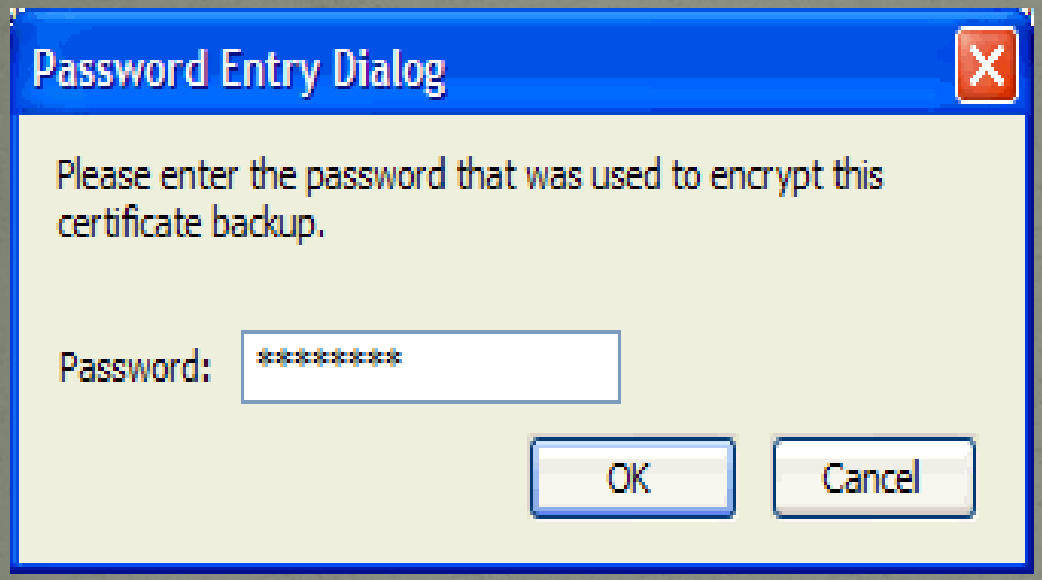
How to Import an Internet Explorer PFX File to Firefox

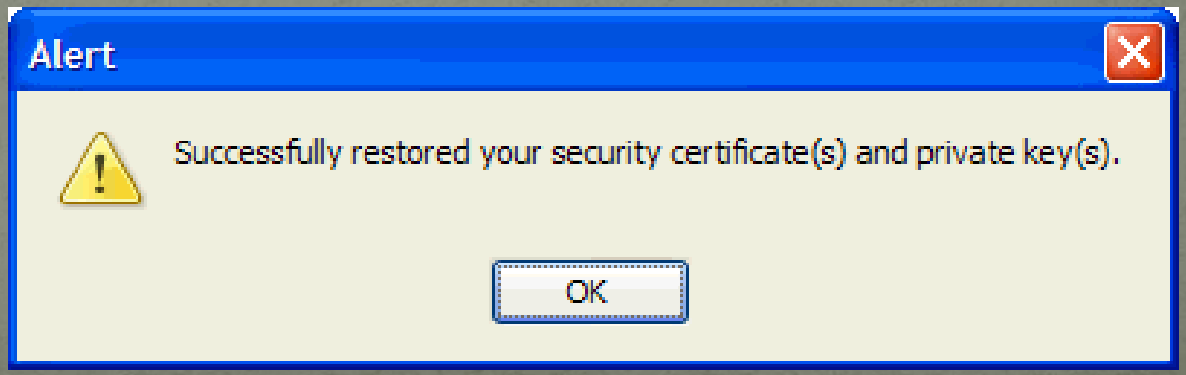


How to Import an Internet Explorer PFX File to Firefox



How to Import an Internet Explorer PFX File to Firefox





X.509 authentication framework with the Web Services

Namespace Involved

```
155         http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
156 wssecurity-secext-1.0.xsd
157          http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
158 wssecurity-utility-1.0.xsd
159
```

160 The following namespace prefixes are used in this document:

Prefix	Namespace
S11	http://schemas.xmlsoap.org/soap/envelope/
S12	http://www.w3.org/2003/05/soap-envelope
ds	http://www.w3.org/2000/09/xmldsig#
xenc	http://www.w3.org/2001/04/xmlenc#
wsse	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd
wsu	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd

X.509 authentication framework with the Web Services

Token Types Involved

Token	ValueType URI	Description
Single certificate	#X509v3	An X.509 v3 signature-verification certificate
Certificate Path	#X509PKIPathv1	An ordered list of X.509 certificates packaged in a PKIPath
Set of certificates and CRLs	#PKCS7	A list of X.509 certificates and (optionally) CRLs packaged in a PKCS#7 wrapper

Token References

- In order to ensure a consistent processing model across all the token types supported by WSS: SOAP Message
- Security, the <wsse:SecurityTokenReference> element SHALL be used to specify all references to
- X.509 token types in signature or encryption elements that comply with this profile.
- A <wsse:SecurityTokenReference> element MAY reference an X.509 token type by one of the following
- means:
- Reference to a Subject Key Identifier
- The <wsse:SecurityTokenReference> element contains a <wsse:KeyIdentifier> element that
- specifies the token data by means of a X.509 SubjectKeyIdentifier reference

Reference to an Issuer and Serial Number

- The <ds:X509IssuerSerial> element is used to specify a reference to an X.509 security token by means of
- the certificate issuer name and serial number.
- The <ds:X509IssuerSerial> element is a direct child of the <ds:X509Data> element that is in turn a direct
- child of the <wsse:SecurityTokenReference> element in which the reference is made.

Signature

- Signed data MAY specify the certificate associated with the signature using any of the X.509 security token types and
- references defined in this specification.
- 0 An X.509 certificate specifies a binding between a public key and a set of attributes that includes (at least) a subject
- name, issuer name, serial number and validity interval. Other attributes may specify constraints on the use of the
- certificate or affect the recourse that may be open to a relying party that depends on the certificate. A given public key
- may be specified in more than one X.509 certificate; consequently a given public key may be bound to two or more
- distinct sets of attributes.
- It is therefore necessary to ensure that a signature created under an X.509 certificate token uniquely and irrefutably
- specifies the certificate under which the signature was created.

Key Identifier

- The <wsse:KeyIdentifier> element does not guarantee an immutable and unambiguous reference to the
- certificate referenced. Consequently implementations that use this form of reference within a signature SHOULD
- employ the STR Dereferencing Transform within a reference to the signature key information in order to ensure that
- the referenced certificate is signed, and not just the ambiguous reference. The form of the reference is a bare name
- reference as defined by the XPointer specification [XPointer].
- The following example shows a certificate referenced by means of a KeyIdentifier. The scope of the signature is the
- <ds:SignedInfo> element which includes both the message body (#body) and the signing certificate by means
- of a reference to the <ds:KeyInfo> element which references it (#keyinfo). Since the <ds:KeyInfo>
- element only contains a mutable reference to the certificate rather than the certificate itself, a transformation is
- specified which replaces the reference to the certificate with the certificate. The <ds:KeyInfo> element specifies
- the signing key by means of a <wsse:SecurityTokenReference> element which contains a
- <wsse:KeyIdentifier> element which specifies the X.509 subject key identifier of the signing certific

How to call web service with Certificate

- 'Public Function CallWebService()

```
    Dim certPath As String = "<C:\WSClientCert.cer>"

    ' Create an instance of the Web service proxy.
    Dim mathservice As New WebSvc.math()
    ' TODO: Replace <https://wsserver/securemath/math.asmx> with a valid URL.
    mathservice.Url = "<https://wsserver/securemath/math.asmx>"


    mathservice.ClientCertificates.Add(X509Certificate.CreateFromCertFile(certPath))

    Dim lngResult As Long = 0
    Try
        lngResult = mathservice.Add(Int32.Parse(operand1.Text), Int32.Parse(operand2.Text))
        Dim result As String = lngResult.ToString()
    Catch ex As Exception
        If TypeOf ex Is Exception Then
            Dim we As Net.WebException = TryCast(ex, Exception)
            Dim webResponse As Net.HttpWebResponse = we.Response
            Throw New Exception("Exception calling method. " & ex.Message)
        End If
```