Using PKI Services Web Application from Internet Explorer on Microsoft Windows Vista Systems

z/OS Cryptographic Services PKI Services versions 7, 8, and 9 only.

Users of Microsoft Windows Vista version operating systems experience problems when accessing the z/OS Cryptographic Services PKI Services Web application through the Internet Explorer browser. Other browsers supported for the Windows Vista operating system, such as the Mozilla Firefox browser, do not experience these problems.

The z/OS Cryptographic Services PKI Services Web application grants the end-user the ability to install an issued certificate directly from the browser. z/OS PKI Services versions 7, 8, and 9 accomplish this using the ActiveX enrollment control interfaces, referred to as `XEnroll.dll`, which is supported for Microsoft Windows XP and earlier versions of the Microsoft Windows operating system.

Beginning with the Vista versions, Internet Explorer for Microsoft Windows is replacing the ActiveX enrollment control `XEnroll.dll` with a new web enrollment facility, `CertEnroll.dll`. ActiveX enrollment control remains available on Microsoft Windows Vista, but the Internet Explorer browser is modified to use `CertEnroll.dll` in place of the ActiveX enrollment control. `CertEnroll.dll` does provide functional compatibility with the `XEnroll.dll` predecessor, but `CertEnroll.dll` does not provide scripting compatibility with `XEnroll.dll`.

The z/OS PKI Services Web application generates the web pages from a template script that employs `XEnroll.dll` scripting language to install certificates. `CertEnroll.dll` scripting language is not currently incorporated into the template. This causes failures for Internet Explorer users that attempt to request, renew, or install certificates through the z/OS PKI Services Web application when attempting these tasks from a computer running the Microsoft Windows Vista version operating systems.

z/OS Cryptographic Services PKI Services Release 10 incorporates support for both ActiveX `XEnroll.dll` and Microsoft Windows Vista `CertEnroll.dll`. For z/OS PKI Services Releases 7, 8, and 9, a local modification can be made to allow these versions to function properly for Microsoft Windows Vista Internet Explorer clients.

This article discusses a local repair that can be applied to the z/OS PKI Services and Microsoft Windows Vista client systems to permit proper functiong of the z/OS PKI Services Web application through the Internet Explorer browser.

What The Local Repair Modifies

The local repair involves modifications to both z/OS Cryptographic Services PKI Services template file on the PKI server system, as well as administrative tasks on the Microsoft Windows Vista client systems to install CAPICOM and to install the z/OS PKI Services CA certificate.

The template script to generate the z/OS PKI Services Web application pages is provided as part of z/OS PKI Services. A sample version of this file is typically located on the PKI server system in the following location:

```
/usr/lpp/pkiserv/samples/pkiserv.tmpl
```

Typically, the system administrator copies this sample file to a run-time location and tailors the file's content to suit the organization's purpose. Typically, the run-time location of this file is:

```
/etc/pkiserv/pkiserv.tmpl
```

Further information concerning this file can be found in the following references:

- **z/OS Cryptographic Services PKI Services Guide and Reference** (SA22-7693-00)
- **Implementing PKI Services on z/OS** from IBM RedBooks (SG24-6968-00)

Additional information about z/OS PKI Services is also available on the World Wide Web at this address:

http://www-03.ibm.com/servers/eserver/zseries/zos/pki/

Local Repair Instructions

A local repair is available for Microsoft Windows Vista systems that are unable to apply the full repair. The instructions for these modifications follow.

Please read all instructions in their entirety before attempting these modifications.

Four items are described in this procedure:

- Task 1: Upgrading the `pkiserv.tmpl` File
- Task 2: Installing CAPICOM on the Microsoft Windows Vista System
- Task 3: Configuring Internet Explorer to Trust PKI Services
- Task 4: Installing z/OS PKI Services CA Certificate

Task 1: Upgrading the pkiserv.tmpl File

System administrators can opt for one of the following approaches to upgrade the z/OS PKI Services Web application template file:

A. Replace the present `pkiserv.tmpl` file with the replacement version included in the official repair, or
B. Modify the present `pkiserv.tmpl` file as instructed in the following procedure.

The first option may be preferrable when few or no modifications have been made to the `pkiserv.tmpl` file. The second option may be preferrable when the organization has made extensive or complicated modifications to this template file.

A. Replacing the Template File

Before replacing the template script file, please select and download the appropriate template for the level of z/OS Cryptographic Services PKI Services that is operational on the PKI server system.

- For z/OS Cryptographic Services PKI Services Release 7, use this link: ftp://ftp.software.ibm.com/eserver/zseries/zos/racf/pkiserv/v1r7/pkiserv.tmpl.
- For z/OS Cryptographic Services PKI Services Release 8, use this link: ftp://ftp.software.ibm.com/eserver/zseries/zos/racf/pkiserv/v1r8/pkiserv.tmpl.
- For z/OS Cryptographic Services PKI Services Release 9, use this link: ftp://ftp.software.ibm.com/eserver/zseries/zos/racf/pkiserv/v1r9/pkiserv.tmpl.

To replace the `pkiserv.tmpl` file, perform the following procedure as the system superuser (the `root` user):

- Rename your current `pkiserv.tmpl` as a backup.
- Download the revised version of the `pkiserv.tmpl` from the above link and save it in the same directory where your `pkiserv.tmpl` file resides.
- Assign `-rw-r--r--` permissions to the new `pkiserv.tmpl` file.
- Edit the new template file to incorporate any modifications that were made to the previous template file.
- Proceed to **Task 2: Installing CAPICOM on the Microsoft Windows Vista System** below.

B. **Upgrading the Current Template File**

To upgrade the current `pkiserv.tmpl` file, perform the following procedure as the system superuser (the `root` user):

- Make a copy of the current `pkiserv.tmpl` file as a backup.

- Edit `pkiserv.tmpl` , modifying the copy to make use of `CertEnroll.dll` interfaces. These modifications are described in the substeps listed below. Modifications that are applicable only to specific versions of z/OS Cryptographic Services PKI Services are labelled to indicate to which versions the step applies.

  a. A new object definition must be inserted into multiple locations within this file. This definition permits the script to make use of the `CertEnroll.dll` interfaces. The object definition must be inserted at four locations.

     - **Location 1 (z/OS PKI Services Release 7 Only)**

Locate the `PKISERV Renew or Revoke a Browser Certificate` page title in the script. The section to be modified should appear as follows:

```
#-- Create a certmgr object for use in the renew process @02A
<TITLE> PKISERV Renew or Revoke a Browser Certificate </TITLE>
<OBJECT
  classid="clsid:127698e4-e730-4e5c-a2b1-21490a70c8a1"
  CODEBASE="xenroll.cab#Version=5,131,3659,0"
  id="certmgr"
>
</OBJECT>
```

Immediately following this code segment, insert the new object definition given in the following example that is highlighted. The resulting code segment should appear as follows:

```
<TITLE> PKISERV Renew or Revoke a Browser Certificate </TITLE>
#-- Create a certmgr object for use in the renew process @02A
<OBJECT
  classid="clsid:127698e4-e730-4e5c-a2b1-21490a70c8a1"
  CODEBASE="xenroll.cab#Version=5,131,3659,0"
  id="certmgr"
>
</OBJECT>
<OBJECT
  classid="clsid:884e2049-217d-11da-b2a4-000e7bbb2b09"
  id="g_objWCF"
>
</OBJECT>
```

- **Location 2**

Next, locate the `Customers Renew or Revoke a Browser Certificate` page title in the script. The section to be modified should appear as follows:

```
<TITLE> Customers Renew or Revoke a Browser Certificate </TITLE>
#-- Create a certmgr object for use in the renew process @02A
<OBJECT
  classid="clsid:127698e4-e730-4e5c-a2b1-21490a70c8a1"
  CODEBASE="xenroll.cab#Version=5,131,3659,0"
  id="certmgr"
>
</OBJECT>
```

Immediately following this code segment, insert the new object definition given in the following example that is highlighted. The resulting code segment should appear as follows:

```
<TITLE> Customers Renew or Revoke a Browser Certificate </TITLE>
#-- Create a certmgr object for use in the renew process @02A
<OBJECT
```

```
    classid="clsid:127698e4-e730-4e5c-a2b1-21490a70c8a1"
    CODEBASE="xenroll.cab#Version=5,131,3659,0"
    id="certmgr"
  >
  </OBJECT>
  <OBJECT
    classid="clsid:884e2049-217d-11da-b2a4-000e7bbb2b09"
    id="g_objWCF"
  >
  </OBJECT>
```

- **Location 3**

  Next, locate the `-AdditionalHeadIE` INSERT in the `Sample INSERTS` section of the script. The section to be modified should appear as follows:

```
  # ============================================================
  #
  # Sample INSERTS
  #
  # ============================================================
  #  @D3C
  <INSERT NAME=-AdditionalHeadIE>
  <OBJECT
    classid="clsid:127698e4-e730-4e5c-a2b1-21490a70c8a1"
    CODEBASE="xenroll.cab#Version=5,131,3659,0"
    id="certmgr"
  >
  </OBJECT>
```

  Immediately following this code segment, insert the new object definition given in the following example that is highlighted. The resulting code segment should appear as follows:

```
  # ============================================================
  #
  # Sample INSERTS
  #
  # ============================================================
  #  @D3C
  <INSERT NAME=-AdditionalHeadIE>
  <OBJECT
    classid="clsid:127698e4-e730-4e5c-a2b1-21490a70c8a1"
    CODEBASE="xenroll.cab#Version=5,131,3659,0"
    id="certmgr"
  >
  </OBJECT>
  <OBJECT
    classid="clsid:884e2049-217d-11da-b2a4-000e7bbb2b09"
    id="g_objWCF"
  >
  </OBJECT>
  </INSERT>
```

- **Location 4**

  Lastly, locate the `returnbrowsercertIE` INSERT within the script. The section to be modified should appear as follows:

```
<INSERT NAME=returnbrowsercertIE>
<HTML>
<HEAD>
<TITLE>MSIE Certificate Install</TITLE>
<OBJECT
  classid="clsid:127698e4-e730-4e5c-a2b1-21490a70c8a1"
  CODEBASE="xenroll.cab#Version=5,131,3659,0"
  id="certmgr"
>
</OBJECT>
```

  Immediately following this code segment, insert the new object definition given in the following example that is highlighted. The resulting code segment should appear as follows:

```
<INSERT NAME=returnbrowsercertIE>
<HTML>
<HEAD>
<TITLE>MSIE Certificate Install</TITLE>
<OBJECT
  classid="clsid:127698e4-e730-4e5c-a2b1-21490a70c8a1"
  CODEBASE="xenroll.cab#Version=5,131,3659,0"
  id="certmgr"
>
</OBJECT>
<OBJECT
  classid="clsid:884e2049-217d-11da-b2a4-000e7bbb2b09"
  id="g_objWCF"
>
</OBJECT>
```

b. Template code modifications are required at several locations within the template file. Those modifications that are required for specific versions of z/OS Cryptographic Services PKI Services are labeled.

- **Location 1 (z/OS PKI Services Releases 8 and 9 Only)**

  Locate the `2-Year PKI Windows Logon Certificate` web page source data section within the template. Modifications are required to the user input fields and verification Javascript within the `<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">` subsection. The original contents of this section are as follows:

```
<BODY>
<H1>2-Year PKI Windows Logon Certificate</H1>
<p>
<H2>Choose one of the following:</H2>
<p>
```

```
<ul>
<h3><li>Request a New Certificate</h3>
# This ACTION forces userid/pw authentication and runs the task under
# the client's ID
#<FORM NAME="CertReq" METHOD=POST ACTION=
#                "/[application]/ssl-cgi-bin/auth/careq.rexx" onSubmit=

# This ACTION forces userid/pw authentication but runs the task under
# the surrogate ID
#<FORM NAME="CertReq" METHOD=POST ACTION=
#              "/[application]/ssl-cgi-bin/surrogateauth/careq.rexx" onSubmit=

# This ACTION is for non z/OS clients. The task runs under the
# surrogate ID
 <FORM NAME="CertReq" METHOD=POST ACTION=
               "/[application]/ssl-cgi-bin/careq.rexx" onSubmit=
    "return ValidateEntry(this)">


<INPUT NAME="Template" TYPE="hidden" vALUE="[tmplname]">
<p> Enter values for the following field(s)
#-- User input fields and validation Javascript ---------------
<SCRIPT LANGUAGE="JavaScript">
<!--
function ValidateEntry(frm){
  if (ValidRequestor(frm) &&
   ValidCommonName(frm) &&
   ValidAltEmail(frm) &&
   ValidAltOther_1_3_6_1_4_1_311_20_2_3(frm) &&
   ValidNotifyEmail(frm) &&
   ValidPassPhrase(frm) &&
   ValidSmartcard(frm)) {
# Add your validation Javascript here if needed ---
 return true;
}
else
 return false;
}
//-->
</SCRIPT>
 %%Requestor (optional)%%
 %%CommonName%%
 %%AltEmail (optional)%%
 %%AltOther_1_3_6_1_4_1_311_20_2_3%%
 %%NotifyEmail (optional)%%
 %%PassPhrase%%
 %%Smartcard[browsertype]%%
#-- End user input fields and validation Javascript -----------
```

This code segment must be modified. The required modifications are highlighted in the example below.

```

<BODY>
<H1>2-Year PKI Windows Logon Certificate</H1>
<p>
```

```
<H2>Choose one of the following:</H2>
<p>
<ul>
<h3><li>Request a New Certificate</h3>
# This ACTION forces userid/pw authentication and runs the task under
# the client's ID
#<FORM NAME="CertReq" METHOD=POST ACTION=
#                "/[application]/ssl-cgi-bin/auth/careq.rexx" onSubmit=

# This ACTION forces userid/pw authentication but runs the task under
# the surrogate ID
#<FORM NAME="CertReq" METHOD=POST ACTION=
#              "/[application]/ssl-cgi-bin/surrogateauth/careq.rexx" onSubmit=

# This ACTION is for non z/OS clients. The task runs under the
# surrogate ID
 <FORM NAME="CertReq" METHOD=POST ACTION=
                "/[application]/ssl-cgi-bin/careq.rexx" onSubmit=
    "return ValidateEntry(this)">


<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<p> Enter values for the following field(s)
#-- User input fields and validation Javascript ---------------
<SCRIPT LANGUAGE="JavaScript">
<!--
function ValidateEntry(frm){
  if (ValidRequestor(frm) &&
   ValidCommonName(frm) &&
   ValidAltEmail(frm) &&
   ValidAltOther_1_3_6_1_4_1_311_20_2_3(frm) &&
   ValidNotifyEmail(frm) &&
   ValidPassPhrase(frm) &&
   ValidSmartcard(frm)) {
# Add your validation Javascript here if needed ---
 return true;
}
else
 return false;
}
//-->
</SCRIPT>
 %%Requestor (optional)%%
 %%CommonName%%
 %%AltEmail (optional)%%
 %%AltOther_1_3_6_1_4_1_311_20_2_3%%
 %%NotifyEmail (optional)%%
 %%PassPhrase%%
 %%PublicKey2[browsertype]%%
#-- End user input fields and validation Javascript -----------
```

- **Location 2**

  Locate the `<INSERT NAME=returnbrowsercertIE>` section within the template. Modifications are required to the `INSTALL_OnClick` subroutine in this section. The original contents of this subroutine are:

```
<SCRIPT LANGUAGE="VBScript">
<!--
  Sub INSTALL_OnClick
    Dim pkcs7data, errmsg, rc
    On Error Resume Next
    certmgr.DeleteRequestCert = false
    err.clear
    certmgr.WriteCertToCSP = true
    pkcs7data = "[iecert]"
    certmgr.acceptPKCS7(pkcs7data)
    if err.number <> 0 then
      certmgr.WriteCertToCSP = false
      err.clear
      certmgr.acceptPKCS7(pkcs7data)
    end if
    if err.number <> 0 then
      'Updated errmsg with CAPICOM information @02C
      errmsg = "Your new certificate failed to install. " & _
      "Please ensure that you are using the same browser " & _
      "that you used when making the certificate request. " & _
      "Also ensure that Microsoft CAPICOM is installed. "
      rc = MsgBox (errmsg, 48, "Certificate Installation")
    else
      errmsg = "Your new certificate installed successfully."
      rc = MsgBox (errmsg, 64, "Certificate Installation")
    end if
  End Sub
  // -->
</SCRIPT>
```

This code segment must be modified to attempt certificate enrollment using the `CertEnroll.dll` methods, then to attempt the prior ActiveX enrollment methods should the `CertEnroll.dll` interfaces not be present.

The required modifications are highlighted in the example below. Relocated segments of the code are also highlighted.

```
<SCRIPT LANGUAGE="VBScript">
<!--
 Sub INSTALL_OnClick
    Dim pkcs7data, errmsg, rc
    ' Added for CertEnroll API processing.                          @LDA
    Dim objEnroll
    On Error Resume Next
    pkcs7data = "[iecert]"
    ' CertEnroll.dll API additions follow.                       19@LDA
    Set objEnroll = g_objWCF.CreateObject("X509Enrollment.CX509Enrollment")
    If IsObject(objEnroll) Then
      ' Vista path, use CertEnroll APIs
      err.clear
      objEnroll.initialize(1)  ' ContextUser
      If err.number <> 0 Then
        errmsg = "Error Initializing Enrollment object. " & err.Description
        Call MsgBox(errmsg, 48, "Error Initializing Enrollment object")
```

```
        Exit Sub
      Else
        err.clear
        Call objEnroll.InstallResponse(0, pkcs7data, 1, "")
        If err.number <> 0 Then
          errmsg = "Error Installing Response. " & err.Description
          Call MsgBox(errmsg, 48, "Error Installing Response")
          Exit Sub
        End If
      End If
    Else
      ' Pre-Vista path, use Xenroll APIs
      certmgr.DeleteRequestCert = false
      err.clear
      certmgr.WriteCertToCSP = true
      certmgr.acceptPKCS7(pkcs7data)
      if err.number <> 0 then
        certmgr.WriteCertToCSP = false
        err.clear
        certmgr.acceptPKCS7(pkcs7data)
      end if
    ' Added during CertEnroll API processing modification.          @LDA
    End If
    if err.number <> 0 then
      'Updated errmsg with CAPICOM information @01C
      errmsg = "Your new certificate failed to install. " & _
      "Please ensure that you are using the same browser " & _
      "that you used when making the certificate request. " & _
      "Also ensure that Microsoft CAPICOM is installed. "
      rc = MsgBox (errmsg, 48, "Certificate Installation")
    else
      errmsg = "Your new certificate installed successfully."
      rc = MsgBox (errmsg, 64, "Certificate Installation")
    end if
 End Sub
// -->
</SCRIPT>
```

- **Location 3**

  Locate the `<INSERT NAME=PublicKeyIE>` section within the template. Two modifications are required in this section.

  This first modification is to the `SendReq` subroutine. The original contents of this subroutine are:

```
<INSERT NAME=PublicKeyIE>
<SCRIPT LANGUAGE="VBScript">
<!--
Sub SendReq

 On Error Resume Next
 Dim pkcs10data,DN,i,Message

 DN= ""
```

```
 CommonName= "Unspecified Distinguished Name"
 DN= "CN=" + CommonName + ";"
 certmgr.KeySpec = 1
 KeyUsage = "1.3.6.1.5.5.7.3.2"
 i = document.all.CSP.options.selectedIndex
 certmgr.providerName = document.all.CSP.options(i).text
 certmgr.providerType = document.all.CSP.options(i).value
 If document.CertReq.KeyProt.value = 1 Then
  certmgr.GenKeyFlags = 3
 Else
  certmgr.GenKeyFlags = 1
 End If

 pkcs10data = ""
 pkcs10data = certmgr.CreatePKCS10(DN, KeyUsage)
 document.CertReq.PublicKey.value = pkcs10data

 If Len(pkcs10data) <= 0 Then
  call MsgBox ("PKCS10 Creation Failed", 48, "Certificate request")
 End If

End Sub
// -->
</SCRIPT>
```

This code segment must be modified to attempt certificate enrollment using the `CertEnroll.dll` methods, then to attempt the prior ActiveX enrollment methods should the `CertEnroll.dll` interfaces not be present.

The required modifications are highlighted in the example below. Relocated segments of the code are also highlighted.

```
<INSERT NAME=PublicKeyIE>
<SCRIPT LANGUAGE="VBScript">
<!--
Sub SendReq

 On Error Resume Next
 Dim pkcs10data,DN,i,Message,CommonName
 Dim objEnroll

 DN= ""
 CommonName= "Unspecified Distinguished Name"
 DN= "CN=" + CommonName + ";"
 pkcs10data = ""

 ' CertEnroll APIs for enrollment processing.
 Set objEnroll = g_objWCF.CreateObject("X509Enrollment.CX509Enrollment")
 If IsObject(objEnroll) Then
   Dim objPrivateKey
   Dim objRequest
   Dim provider
   Dim selectedCSP
   Dim objCSPs
```

```
    Set objPrivateKey = g_objWCF.CreateObject("X509Enrollment.CX509PrivateKey")
    If IsObject(objPrivateKey) = FALSE Then
      Message = "Error creating Private Key object: " & vbNewline & _
                Err.Description
      Call MsgBox(Message ,48,"Error creating Private Key object")
      Exit Sub
    End If

    Set objRequest = g_objWCF.CreateObject("X509Enrollment.CX509CertificateRequestPkcs10")
    If IsObject(objRequest) = FALSE Then
      Message = "Error creating Request object: " & vbNewline & _
                Err.Description
      Call MsgBox(Message ,48,"Error creating Certificate Request object")
      Exit Sub
    End If

    ' Setup Private key properties based on the selected provider
    i = document.all.CSP.options.selectedIndex
    provider = LCase(document.all.CSP.options(i).text)
    If InStr(1, provider, "smart", 1) > 0 _
    Or InStr(1, provider, "card", 1) > 0 Then
      ' For Smart Card Providers, retrieve the index of the selected CSP
      ' and set the Private key name, type, and KeySpec
      objPrivateKey.ProviderName = document.all.CSP.options(i).text
      objPrivateKey.ProviderType = document.all.CSP.options(i).value
      objPrivateKey.KeySpec      = 1 ' XCN_AT_KEYEXCHANGE
    Else
      Set selectedCSP = g_objWCF.CreateObject("X509Enrollment.CCspInformation")
      If IsObject(selectedCSP) = FALSE Then
        Message = "Error creating the a CSP Information object: " & vbNewline & _
                  Err.Description
        Call MsgBox(Message ,48,"Error creating CSPInformation object")
        Exit Sub
      End If

      Set objCSPs = g_objWCF.CreateObject("X509Enrollment.CCspInformations")
      If IsObject(objCSPs) = FALSE Then
        Message = "Error creating the CSP Informations object: " & vbNewline & _
                  Err.Description
        Call MsgBox(Message ,48,"Error creating CSPInformations object")
        Exit Sub
      End If

      ' Retrieve the index of the selected CSP and initialize the
      ' CSPInformation object using the provider name
      selectedCSP.InitializeFromName( document.all.CSP.options(i).text )

      ' Add the CSPInformation object to the CSPInformations object
      objCSPs.add( selectedCSP )

      ' Set the PrivateKey objects CspInformations to our object
      objPrivateKey.CspInformations = objCSPs

      ' Set intended usage of private key for KeyExchange purposes
      objPrivateKey.KeySpec = 1 ' XCN_AT_KEYEXCHANGE

      ' Set KeyProtection based on user input
```

```
      If document.CertReq.KeyProt.value = 1 Then
        objPrivateKey.KeyProtection = 2 ' XCN_NCRYPT_UI_FORCE_HIGH_PROTECTION_FLAG
      Else
        objPrivateKey.KeyProtection = 0 ' XCN_NCRYPT_UI_NO_PROTECTION_FLAG @DxC
      End If

      '===================================================================
      ' The ExportPolicy is set to allow the private key to be exported,
      ' other options allow the private key to be exported only once for
      ' archival in a variety of formats, or prevents export of the
      ' private key.
      '   ExportPolicy = 0 = XCN_NCRYPT_ALLOW_EXPORT_NONE
      '   ExportPolicy = 1 = XCN_NCRYPT_ALLOW_EXPORT_FLAG
      '   ExportPolicy = 2 = XCN_NCRYPT_ALLOW_PLAINTEXT_EXPORT_FLAG
      '   ExportPolicy = 4 = XCN_NCRYPT_ALLOW_ARCHIVING_FLAG
      '   ExportPolicy = 8 = XCN_NCRYPT_ALLOW_PLAINTEXT_ARCHIVING_FLAG
      '===================================================================
      objPrivateKey.ExportPolicy = 1 ' XCN_NCRYPT_ALLOW_EXPORT_FLAG
    End If

    Err.clear
    objRequest.InitializeFromPrivateKey 1, objPrivateKey, ""
    If Err.Number <> 0 Then
      Message = "Error initializing request from private key " & vbNewline & _
                Err.Description
      Call MsgBox(Message ,48,"Error initializing Certificate Request object")
      Exit Sub
    End If

    objRequest.Subject = DN

    Err.clear
    objEnroll.InitializeFromRequest( objRequest )
    If Err.Number <> 0 Then
      Message = "Error initializing Enrollment object from request: " & _
                vbNewline & Err.Description
      Call MsgBox(Message ,48,"Error initializing Enrollment object")
      Exit Sub
    End If

    pkcs10data = objEnroll.CreateRequest(1) ' XCN_CRYPT_STRING_BASE64

  Else
    ' XEnroll APIs for enrollment processing
    certmgr.KeySpec = 1
    KeyUsage = "1.3.6.1.5.5.7.3.2"
    i = document.all.CSP.options.selectedIndex
    certmgr.providerName = document.all.CSP.options(i).text
    certmgr.providerType = document.all.CSP.options(i).value
    If document.CertReq.KeyProt.value = 1 Then
     certmgr.GenKeyFlags = 3
    Else
     certmgr.GenKeyFlags = 1
    End If

    pkcs10data = certmgr.CreatePKCS10(DN, KeyUsage)
```

```
 End If

 document.CertReq.PublicKey.value = pkcs10data

 If Len(pkcs10data) <= 0 Then
   call MsgBox ("PKCS10 Creation Failed",48,"Certificate request")
 End If

End Sub
// -->
</SCRIPT>
```

The second modification to the `<INSERT NAME=PublicKeyIE>` section occurs in the `<select name="CSP">` subsection. The original contents of this subsection are:

```
  <select name="CSP">
  <script language="VBScript">
   On Error Resume Next
     Dim i, csp, sv


   certmgr.providerType = 1
   i = 0
   csp = ""
   csp = certmgr.enumProviders(i,0)
   sv = "SELECTED"
   If Len(csp) = 0 Then
    errmsg = "Your PC needs a Windows upgrade before certificates " & _
     "can be requested. Click the 'Tools' option on the browser " & _
     "menu then 'Windows Update' to retrieve the upgrade. "
    Call MsgBox(errmsg,8,"Security Warning")
   End If
   While Len(csp) <> 0
    document.write("<OPTION VALUE=1 " & sv & ">" & csp & "</OPTION>")
    i = i + 1
    csp = ""
    csp = certmgr.enumProviders(i,0)
    sv = ""
   Wend
  </script>
  </select>
```

This subsection must be modified to employ both the `CertEnroll.dll` interfaces as well as retaining the ActiveX enrollment interfaces for prior versions of Windows and other browsers.

The required modifications are highlighted in the example below.

```
  <select name="CSP">
  <script language="VBScript">
   On Error Resume Next
     Dim i, csp, sv
```

```
      ' Modifications for CertEnroll API enrollment process.
      Dim objCSPs
      Dim oOption
      Dim errmsg

      Set objCSPs = g_objWCF.CreateObject("X509Enrollment.CCspInformations")
      If IsObject(objCSPs) Then
        ' Vista path, use CertEnroll APIs
        objCSPs.AddAvailableCsps
        For i = 0 to objCSPs.Count-1
          ' Only include Legacy(Crypto API) providers at this time
          If (objCSPs.ItemByIndex(i).LegacyCsp) Then
            Set oOption   = document.createElement("OPTION")
            oOption.text  = objCSPs.ItemByIndex(i).Name
            oOption.value = objCSPs.ItemByIndex(i).Type
            Document.all.CSP.add(oOption)
          End If
        Next
      Else
        ' Pre-Vista path, use Xenroll APIs
        certmgr.providerType = 1
        i = 0
        csp = ""
        csp = certmgr.enumProviders(i,0)
        sv = "SELECTED"
        If Len(csp) = 0 Then
         errmsg = "Your PC needs a Windows upgrade before certificates " & _
          "can be requested. Click the 'Tools' option on the browser " & _
          "menu then 'Windows Update' to retrieve the upgrade. "
         Call MsgBox(errmsg,8,"Security Warning")
        End If
        While Len(csp) <> 0
         document.write("<OPTION VALUE=1 " & sv & ">" & csp & "</OPTION>")
         i = i + 1
         csp = ""
         csp = certmgr.enumProviders(i,0)
         sv = ""
        Wend
      ' Added for CertEnroll.
      End If
  </script>
  </select>
```

- **Location 4 (z/OS PKI Services Releases 7 and 8 Only)**

  Locate the `<INSERT NAME=-RenewKeySetIE>` section within the template. Modifications are required to the `RenewKeySet` function in this section. The original contents of this function are:

```
<INSERT NAME=-RenewKeySetIE>
<SCRIPT LANGUAGE="VBScript">
<!--
Function RenewKeySet()

 On Error Resume Next
```

```
  Dim pkcs10data, SelectedCert, CertStore, Cert
  'Store Options
  Const CAPICOM_MEMORY_STORE = 0
  Const CAPICOM_LOCAL_MACHINE_STORE = 1
  Const CAPICOM_CURRENT_USER_STORE = 2
  Const CAPICOM_MY_STORE = "My"
  Const CAPICOM_STORE_OPEN_READ_ONLY = 0
  RenewKeySet = 0
  document.renform.PublicKey.value = ""
  'Create SelectedCert
  B64cert = "[iecert]"
  Set SelectedCert = CreateObject("CAPICOM.Certificate")
  SelectedCert.Import(B64cert)

  'Create Cert Store
  Set CertStore = CreateObject("CAPICOM.Store")
  CertStore.Open CAPICOM_CURRENT_USER_STORE, CAPICOM_MY_STORE, CAPICOM_STORE_OPEN_READ_ONLY
  'CAPICOM not installed
  If Err.Number > 0 Then
    RenewKeySet = 1
  ElseIf Err.Number < 0 Then
    RenewKeySet = 2
  Else
    'CAPICOM present, keep on processing
    'Find selected certificate in the store
    Set Cert = CreateObject("CAPICOM.Certificate")
    For i = 1 to CertStore.Certificates.Count
      If CertStore.Certificates(i).Thumbprint = SelectedCert.Thumbprint Then
        Set Cert = CertStore.Certificates(i)
        Exit For
      End If
    Next
    KeyUsage = "1.3.6.1.5.5.7.3.2"
    certmgr.UseExistingKeySet = True
    'Set fields to be reused from the selected cert
    certmgr.ContainerName = Cert.PrivateKey.ContainerName
    certmgr.ProviderName = Cert.PrivateKey.ProviderName
    certmgr.ContainerName = Cert.PrivateKey.ContainerName
    certmgr.ProviderType = Cert.PrivateKey.ProviderType
    certmgr.KeySpec = Cert.PrivateKey.KeySpec

    pkcs10data = ""
    pkcs10data = certmgr.CreatePKCS10(Cert.subjectName, KeyUsage)
    document.renform.PublicKey.value = pkcs10data
  End If
End Function
// -->
</SCRIPT>
```

This subsection must be modified to employ both the `CertEnroll.dll` interfaces as well as retaining the ActiveX enrollment interfaces for prior versions of Windows and other browsers.

The required modifications are highlighted in the example below. Relocated segments of the code are also highlighted.

```
<INSERT NAME=-RenewKeySetIE>
<SCRIPT LANGUAGE="VBScript">
<!--
Function RenewKeySet()

 On Error Resume Next
 Dim pkcs10data, SelectedCert, CertStore, Cert
 Dim Enroll

 'Store Options
 Const CAPICOM_MEMORY_STORE = 0
 Const CAPICOM_LOCAL_MACHINE_STORE = 1
 Const CAPICOM_CURRENT_USER_STORE = 2
 Const CAPICOM_MY_STORE = "My"
 Const CAPICOM_STORE_OPEN_READ_ONLY = 0
 RenewKeySet = 0
 document.renform.PublicKey.value = ""
 'Create SelectedCert
 B64cert = "[iecert]"
 Set SelectedCert = CreateObject("CAPICOM.Certificate")
 SelectedCert.Import(B64cert)

 'Create Cert Store
 Set CertStore = CreateObject("CAPICOM.Store")
 CertStore.Open CAPICOM_CURRENT_USER_STORE, _
                CAPICOM_MY_STORE, _
                CAPICOM_STORE_OPEN_READ_ONLY
 'CAPICOM not installed
 If Err.Number > 0 Then
   WarnMsg = "Microsoft CAPICOM is not installed or " & _
             "disabled.  You may not be able to install " & _
             "the renewed certificate. Do you want to " & _
             "continue anyway? "
   Result = MsgBox(WarnMsg, 52, "Warning")
   'User chooses Yes
   If Result = 6 Then
     RenewKeySet = 1
   'User chooses No
   Else
     RenewKeySet = 2
   End If
 'CAPICOM present, but user chooses not allowing access to it
 ElseIf Err.Number < 0 Then
   RenewKeySet = 2
 Else
   'CAPICOM present, keep on processing
   'Find selected certificate in the store
   Set Cert = CreateObject("CAPICOM.Certificate")
   For i = 1 to CertStore.Certificates.Count
     If CertStore.Certificates(i).Thumbprint = _
       SelectedCert.Thumbprint _
     Then
       Set Cert = CertStore.Certificates(i)
       Exit For
     End If
   Next
   pkcs10data = ""
```

```
    Set Enroll = g_objWCF.CreateObject( _
                              "X509Enrollment.CX509Enrollment")
    If IsObject(Enroll) Then
      ' Vista version uses CertEnroll
      Dim Request
      Dim PrivateKey
      Dim b64cert
      Dim msg

      Set Request = g_objWCF.CreateObject( _
                "X509Enrollment.CX509CertificateRequestPkcs10")
      If IsObject(Request) = FALSE Then
        msg = "Error creating the Request object: " & _
              Err.Description
        Call MsgBox(msg, _
                    48, _
                    "Error creating Certificate Request object")
        Exit Function
      End If

      Set PrivateKey = g_objWCF.CreateObject( _
                              "X509Enrollment.CX509PrivateKey")
      If IsObject(PrivateKey) = FALSE Then
        msg = "Error creating the PrivateKey object: " & _
              Err.Description
        Call MsgBox(msg, _
                    48, _
                    "Error creating Private Key object")
        Exit Function
      End If

      PrivateKey.Existing = True
      PrivateKey.ContainerName = _
                          Cert.PrivateKey.UniqueContainerName
      PrivateKey.ProviderName  = Cert.PrivateKey.ProviderName
      PrivateKey.ProviderType  = Cert.PrivateKey.ProviderType

      Err.clear
      Request.InitializeFromPrivateKey 1, PrivateKey, ""
      If Err.Number <> 0 Then
        msg = "Error initializing request from existing " & _
              "private key " & Err.Description
        Call MsgBox(msg, _
                    48, _
                    "Error initing Certificate Request object")
        Exit Function
      End If

      Err.clear
      Enroll.InitializeFromRequest( Request )
      If Err.Number <> 0 Then
        msg = "Error initializing enroll object from " & _
              "request: " & Err.Description
        Call MsgBox(msg,48,"Error initializing enroll object")
        Exit Function
      End If
```

```
        Err.clear
        ' CreateRequest(1) == XCN_CRYPT_STRING_BASE64
        pkcs10data = Enroll.CreateRequest(1)
        If Err.Number <> 0 Then
          msg = "Error creating the renewal request: " & _
                Err.Description
          Call MsgBox(msg ,48 , "Error creating renewal request")
          Exit Function
        End If

      Else
        ' Non-Vista version uses xenroll
        KeyUsage = "1.3.6.1.5.5.7.3.2"
        certmgr.UseExistingKeySet = True
        'Set fields to be reused from the selected cert
        certmgr.ContainerName = Cert.PrivateKey.ContainerName
        certmgr.ProviderName = Cert.PrivateKey.ProviderName
        certmgr.ProviderType = Cert.PrivateKey.ProviderType
        certmgr.KeySpec = Cert.PrivateKey.KeySpec

        pkcs10data = certmgr.CreatePKCS10(Cert.subjectName, _
                                          KeyUsage)

      End If

      document.renform.PublicKey.value = pkcs10data

    End If
  End Function
  // -->
  </SCRIPT>
```

- **Location 5 (z/OS PKI Services Release 9 Only)**

  Locate the `<INSERT NAME=-RenewKeySetIE>` section within the template. Modifications are required to the `RenewKeyset` function in this section. The original contents of this function are:

```
<INSERT NAME=-RenewKeySetIE>
<SCRIPT LANGUAGE="VBScript">
<!--
Function RenewKeySet()

 On Error Resume Next
 Dim pkcs10data, SelectedCert, CertStore, Cert
 'Store Options
 Const CAPICOM_MEMORY_STORE = 0
 Const CAPICOM_LOCAL_MACHINE_STORE = 1
 Const CAPICOM_CURRENT_USER_STORE = 2
 Const CAPICOM_MY_STORE = "My"
 Const CAPICOM_STORE_OPEN_READ_ONLY = 0
 RenewKeySet = 0
 document.renform.PublicKey.value = ""
 'Create SelectedCert
 B64cert = "[iecert]"
```

```
  Set SelectedCert = CreateObject("CAPICOM.Certificate")
  SelectedCert.Import(B64cert)

  'Create Cert Store
  Set CertStore = CreateObject("CAPICOM.Store")
  CertStore.Open CAPICOM_CURRENT_USER_STORE, CAPICOM_MY_STORE, CAPICOM_STORE_OPEN_READ_ONLY
  'CAPICOM not installed
  If Err.Number > 0 Then
    WarnMsg = "Microsoft CAPICOM is not installed or disabled. " &  _
              "You may not be able to install the renewed " & _
              "certificate. Do you want to continue anyway? "
    Result = MsgBox(WarnMsg, 52, "Warning")
    'User chooses Yes
    If Result = 6 Then
      RenewKeySet = 1
    'User chooses No
    Else
      RenewKeySet = 2
    End If
  'CAPICOM present, but user chooses not allowing access to it
  ElseIf Err.Number < 0 Then
    RenewKeySet = 2
  Else
    'CAPICOM present, keep on processing
    'Find selected certificate in the store
    Set Cert = CreateObject("CAPICOM.Certificate")
    For i = 1 to CertStore.Certificates.Count
      If CertStore.Certificates(i).Thumbprint = SelectedCert.Thumbprint Then
        Set Cert = CertStore.Certificates(i)
        Exit For
      End If
    Next
    KeyUsage = "1.3.6.1.5.5.7.3.2"
    certmgr.UseExistingKeySet = True
    'Set fields to be reused from the selected cert
    certmgr.ContainerName = Cert.PrivateKey.ContainerName
    certmgr.ProviderName = Cert.PrivateKey.ProviderName
    certmgr.ContainerName = Cert.PrivateKey.ContainerName
    certmgr.ProviderType = Cert.PrivateKey.ProviderType
    certmgr.KeySpec = Cert.PrivateKey.KeySpec

    pkcs10data = ""
    pkcs10data = certmgr.CreatePKCS10(Cert.subjectName, KeyUsage)
    document.renform.PublicKey.value = pkcs10data
  End If
End Function
// -->
</SCRIPT>
```

This subsection must be modified to employ both the `CertEnroll.dll` interfaces as well as retaining the ActiveX enrollment interfaces for prior versions of Windows and other browsers.

The required modifications are highlighted in the example below. Relocated segments of the code are also highlighted.

```
  <INSERT NAME=-RenewKeySetIE>
  <SCRIPT LANGUAGE="VBScript">
  <!--
  Function RenewKeySet()

   On Error Resume Next
   Dim pkcs10data, SelectedCert, CertStore, Cert
   ' Added for CertEnroll API process.
   Dim Enroll
   'Store Options
   Const CAPICOM_MEMORY_STORE = 0
   Const CAPICOM_LOCAL_MACHINE_STORE = 1
   Const CAPICOM_CURRENT_USER_STORE = 2
   Const CAPICOM_MY_STORE = "My"
   Const CAPICOM_STORE_OPEN_READ_ONLY = 0
   RenewKeySet = 0
   document.renform.PublicKey.value = ""
   'Create SelectedCert
   B64cert = "[iecert]"
   Set SelectedCert = CreateObject("CAPICOM.Certificate")
   SelectedCert.Import(B64cert)

   'Create Cert Store
   Set CertStore = CreateObject("CAPICOM.Store")
   CertStore.Open CAPICOM_CURRENT_USER_STORE, CAPICOM_MY_STORE,
CAPICOM_STORE_OPEN_READ_ONLY
   'CAPICOM not installed
   If Err.Number > 0 Then
     WarnMsg = "Microsoft CAPICOM is not installed or disabled. " &  _
               "You may not be able to install the renewed " & _
               "certificate. Do you want to continue anyway? "
     Result = MsgBox(WarnMsg, 52, "Warning")
     'User chooses Yes
     If Result = 6 Then
       RenewKeySet = 1
     'User chooses No
     Else
       RenewKeySet = 2
     End If
   'CAPICOM present, but user chooses not allowing access to it
   ElseIf Err.Number < 0 Then
     RenewKeySet = 2
   Else
     'CAPICOM present, keep on processing
     'Find selected certificate in the store
     Set Cert = CreateObject("CAPICOM.Certificate")
     For i = 1 to CertStore.Certificates.Count
       If CertStore.Certificates(i).Thumbprint = SelectedCert.Thumbprint Then
         Set Cert = CertStore.Certificates(i)
         Exit For
       End If
     Next

     ' CertEnroll API processing.
     pkcs10data = ""
     Set Enroll = g_objWCF.CreateObject("X509Enrollment.CX509Enrollment")
     If IsObject(Enroll) Then
```

```
        ' Vista version uses CertEnroll
        Dim Request
        Dim PrivateKey
        Dim b64cert
        Dim msg

        Set Request = g_objWCF.CreateObject("X509Enrollment.CX509CertificateRequestPkcs10")
        If IsObject(Request) = FALSE Then
          msg = "Error creating the Request object: " & Err.Description
          Call MsgBox(msg ,48,"Error creating Certificate Request object")
          Exit Function
        End If

        Set PrivateKey = g_objWCF.CreateObject("X509Enrollment.CX509PrivateKey")
        If IsObject(PrivateKey) = FALSE Then
          msg = "Error creating the PrivateKey object: " & Err.Description
          Call MsgBox(msg ,48,"Error creating Private Key object")
          Exit Function
        End If

        PrivateKey.Existing = True
        PrivateKey.ContainerName = Cert.PrivateKey.UniqueContainerName
        PrivateKey.ProviderName  = Cert.PrivateKey.ProviderName
        PrivateKey.ProviderType  = Cert.PrivateKey.ProviderType

        Err.clear
        Request.InitializeFromPrivateKey 1, PrivateKey, ""
        If Err.Number <> 0 Then
          msg = "Error initializing request from existing private key " & Err.Description
          Call MsgBox(msg ,48,"Error initing Certificate Request object")
          Exit Function
        End If

        Err.clear
        Enroll.InitializeFromRequest( Request )
        If Err.Number <> 0 Then
          msg = "Error initializing enroll object from request: " & Err.Description
          Call MsgBox(msg ,48,"Error initializing enroll object")
          Exit Function
        End If

        Err.clear
        pkcs10data = Enroll.CreateRequest(1) ' XCN_CRYPT_STRING_BASE64
        If Err.Number <> 0 Then
          msg = "Error creating the renewal request: " & Err.Description
          Call MsgBox(msg ,48 , "Error creating renewal request")
          Exit Function
        End If

      Else
        ' Non-Vista version uses xenroll
        KeyUsage = "1.3.6.1.5.5.7.3.2"
        certmgr.UseExistingKeySet = True
        'Set fields to be reused from the selected cert
        certmgr.ContainerName = Cert.PrivateKey.ContainerName
        certmgr.ProviderName = Cert.PrivateKey.ProviderName
        certmgr.ProviderType = Cert.PrivateKey.ProviderType
```

```
        certmgr.KeySpec = Cert.PrivateKey.KeySpec

        pkcs10data = certmgr.CreatePKCS10(Cert.subjectName, KeyUsage)
      ' Added during CertEnroll addition.
      End If
      document.renform.PublicKey.value = pkcs10data
    End If
  End Function
  // -->
  </SCRIPT>
```

- **Location 6 (z/OS PKI Services Release 8 and 9 Only)**

  Locate the `<INSERT NAME=SmartcardNS>` section within the template. The name of this section must be changed. The original contents of this section are:

```
  <INSERT NAME=SmartcardNS>
  <SCRIPT LANGUAGE="JavaScript">
  <!--
  function ValidSmartcard(frm){
   alert("Cannot use a Mozilla based browser for this certificate type");
   return false;
  }
  //-->
  </SCRIPT>
  </INSERT>
```

  The name of this section must be changed. The required modifications are highlighted in the example below.

```
  # Changed name of insert from SmartCardNS to PublicKey2NS to match
  # CGI scripts
  <INSERT NAME=PublicKey2NS>
  <SCRIPT LANGUAGE="JavaScript">
  <!--
  function ValidSmartcard(frm){
   alert("Cannot use a Mozilla based browser for this certificate type");
   return false;
  }
  //-->
  </SCRIPT>
  </INSERT>
```

- **Location 7 (z/OS PKI Services Release 8 and 9 Only)**

  Locate the `<INSERT NAME=SmartcardIE>` section within the template. Two modifications are required in this section.

  The first modification required in this section is to the `SendReq` subroutine. The original contents of this subroutine are:

```
```

```
<INSERT NAME=SmartcardIE>
<SCRIPT LANGUAGE="VBScript">
<!--
Sub SendReq

 On Error Resume Next
 Dim pkcs10data,DN,i,Message

 DN= ""
 CommonName= "Unspecified Distinguished Name"
 DN= "CN=" + CommonName + ";"
 certmgr.KeySpec = 1
 KeyUsage = "1.3.6.1.5.5.7.3.2"
 i = document.all.CSP.options.selectedIndex
 certmgr.providerName = document.all.CSP.options(i).text
 certmgr.providerType = getProviderType(certmgr.providerName)
 certmgr.GenKeyFlags = 0

 pkcs10data = ""
 pkcs10data = certmgr.CreatePKCS10(DN, KeyUsage)
 document.CertReq.PublicKey.value = pkcs10data

 If Len(pkcs10data) <= 0 Then
  call MsgBox ("PKCS10 Creation Failed",48,"Certificate request")
 End If

End Sub
// -->
</SCRIPT>
```

This subsection must be modified to employ both the `CertEnroll.dll` interfaces as well as retaining the ActiveX enrollment interfaces for prior versions of Windows and other browsers. *Please notice that the name of this INSERT section is being changed by this modification.*

The required modifications are highlighted in the example below. Relocated segments of the code are also highlighted.

```

# Changed name of insert from SmartCardIE to PublicKey2IE to match
# CGI scripts
<INSERT NAME=PublicKey2IE>
<SCRIPT LANGUAGE="VBScript">
<!--
Sub SendReq

On Error Resume Next
' Modified for CertEnroll API processing
Dim pkcs10data,DN,i,Message,CommonName
Dim objEnroll

DN= ""
CommonName= "Unspecified Distinguished Name"
DN= "CN=" + CommonName + ";"

' Additions for CertEnroll API processing.
pkcs10data = ""
```

```
  Set objEnroll = g_objWCF.CreateObject("X509Enrollment.CX509Enrollment")
  If IsObject(objEnroll) Then
    ' This is the Vista path which uses CertEnroll APIs
    Dim objPrivateKey
    Dim objRequest

    Set objPrivateKey = g_objWCF.CreateObject("X509Enrollment.CX509PrivateKey")
    If IsObject(objPrivateKey) = FALSE Then
      Message = "Error creating X509PrivateKey object: " & Err.Description
      Call MsgBox(Message ,48,"Error creating Private Key object")
      Exit Sub
    End If

    Set objRequest = g_objWCF.CreateObject("X509Enrollment.CX509CertificateRequestPkcs10")
    If IsObject(objRequest) = FALSE Then
      Message = "Error creating PKCS10 Request object: " & Err.Description
      Call MsgBox(Message ,48,"Error creating Certificate Request object")
      Exit Sub
    End If

    i = document.all.CSP.options.selectedIndex
    objPrivateKey.ProviderName = document.all.CSP.options(i).text
    objPrivateKey.ProviderType = document.all.CSP.options(i).value
    objPrivateKey.KeySpec      = 1 ' XCN_AT_KEYEXCHANGE

    Err.clear
    Call objRequest.InitializeFromPrivateKey(1, objPrivateKey, "")
    If Err.Number <> 0 Then
      Message = "Error initializing the Request from private key: " & _
                vbNewline & Err.Description
      Call MsgBox(Message ,48,"Error initializing request")
      Exit Sub
    End If

    objRequest.Subject = DN

    Err.clear
    objEnroll.InitializeFromRequest( objRequest )
    If Err.Number <> 0 Then
      Message = "Error initializing Enrollment object from request: " & _
                vbNewline & Err.Description
      Call MsgBox(Message ,48,"Error initializing Enrollment object")
      Exit Sub
    End If

    pkcs10data = objEnroll.CreateRequest(1) ' XCN_CRYPT_STRING_BASE64

  Else
    ' This is the non-Vista path which uses Xenroll APIs

    certmgr.KeySpec = 1
    KeyUsage = "1.3.6.1.5.5.7.3.2"
    i = document.all.CSP.options.selectedIndex
    certmgr.providerName = document.all.CSP.options(i).text
    certmgr.providerType = getProviderType(certmgr.providerName)
    certmgr.GenKeyFlags = 0
```

```
    pkcs10data = certmgr.CreatePKCS10(DN, KeyUsage)

    ' Added during CertEnroll API processing addition.
  End If

  document.CertReq.PublicKey.value = pkcs10data

  If Len(pkcs10data) <= 0 Then
    call MsgBox ("PKCS10 Creation Failed",48,"Certificate request")
  End If

  End Sub
  // -->
  </SCRIPT>
```

The second modification to this section is to the `<select name="CSP">` subsection. The original contents of this subsection are:

```
  <select name="CSP">
  <script language="VBScript">
   On Error Resume Next
     Dim i, csp, sv


    certmgr.providerType = 1
    i = 0
    csp = ""
    csp = certmgr.enumProviders(i,0)
    sv = "SELECTED"
    If Len(csp) = 0 Then
     errmsg = "Your PC needs a Windows upgrade before certificates " & _
      "can be requested. Click the 'Tools' option on the browser " & _
      "menu then 'Windows Update' to retrieve the upgrade. "
     Call MsgBox(errmsg,8,"Security Warning")
    End If
    While Len(csp) <> 0
     '=========================================================
     '
     ' Edit this If statement to add or remove smartcard
     ' providers as desired
     '
     '=========================================================
     If Mid(csp,1,7) = "Datakey" _
     Or Mid(csp,1,7) = "Gemplus" _
     Or Mid(csp,1,16) = "Infineon SICRYPT" _
     Or Mid(csp,1,12) = "Schlumberger" Then
      document.write("<OPTION VALUE=" & """ & csp & """ & sv & ">" & csp & "</OPTION>")
     End If
     i = i + 1
     csp = ""
     csp = certmgr.enumProviders(i,0)
     sv = ""
    Wend
  </script>
  </select>
```

This subsection must be modified to employ both the `CertEnroll.dll` interfaces as well as retaining the ActiveX enrollment interfaces for prior versions of Windows and other browsers.

The required modifications are highlighted in the example below.

```
<select name="CSP">
<script language="VBScript">
 On Error Resume Next
   Dim i, csp, sv

   ' CertEnroll API processing additions.
   Dim objCSPs
   Dim oOption
   Dim provider
   Dim msg

   Set objCSPs = g_objWCF.CreateObject("X509Enrollment.CCspInformations")
   If IsObject(objCSPs) Then
     ' This is the Vista version which uses CertEnroll APIs
     objCSPs.AddAvailableCsps
     For i = 0 to objCSPs.Count-1
       If (objCSPs.ItemByIndex(i).LegacyCsp) Then
         provider = LCase(objCSPs.ItemByIndex(i).Name)
         If InStr(1, provider, "smart", 1) > 1 _
         Or InStr(1, provider, "card", 1) > 1 Then
           Set oOption   = document.createElement("OPTION")
           oOption.text  = objCSPs.ItemByIndex(i).Name
           oOption.value = objCSPs.ItemByIndex(i).Type
           Document.all.CSP.add(oOption)
         End If
       End If
     Next
   Else
     ' This is the non-Vista version which uses Xenroll APIs
     certmgr.providerType = 1
     i = 0
     csp = ""
     csp = certmgr.enumProviders(i,0)
     sv = "SELECTED"
     If Len(csp) = 0 Then
      errmsg = "Your PC needs a Windows upgrade before certificates " & _
       "can be requested. Click the 'Tools' option on the browser " & _
       "menu then 'Windows Update' to retrieve the upgrade. "
      Call MsgBox(errmsg,8,"Security Warning")
     End If
     While Len(csp) <> 0
      '=======================================================
      '
      ' Edit this If statement to add or remove smartcard
      ' providers as desired
      '
      '=======================================================
      If Mid(csp,1,7) = "Datakey" _
```

```
      Or Mid(csp,1,7) = "Gemplus" _
      Or Mid(csp,1,16) = "Infineon SICRYPT" _
      Or Mid(csp,1,12) = "Schlumberger" Then
       document.write("<OPTION VALUE=" & """ & csp & """ & sv & ">" & csp & "</OPTION>")
      End If
      i = i + 1
      csp = ""
      csp = certmgr.enumProviders(i,0)
      sv = ""
     Wend
   ' Added for CertEnroll API processing additions.
   End If

</script>
</select>
```

- Proceed to **Task 2: Installing CAPICOM on the Microsoft Windows Vista System**

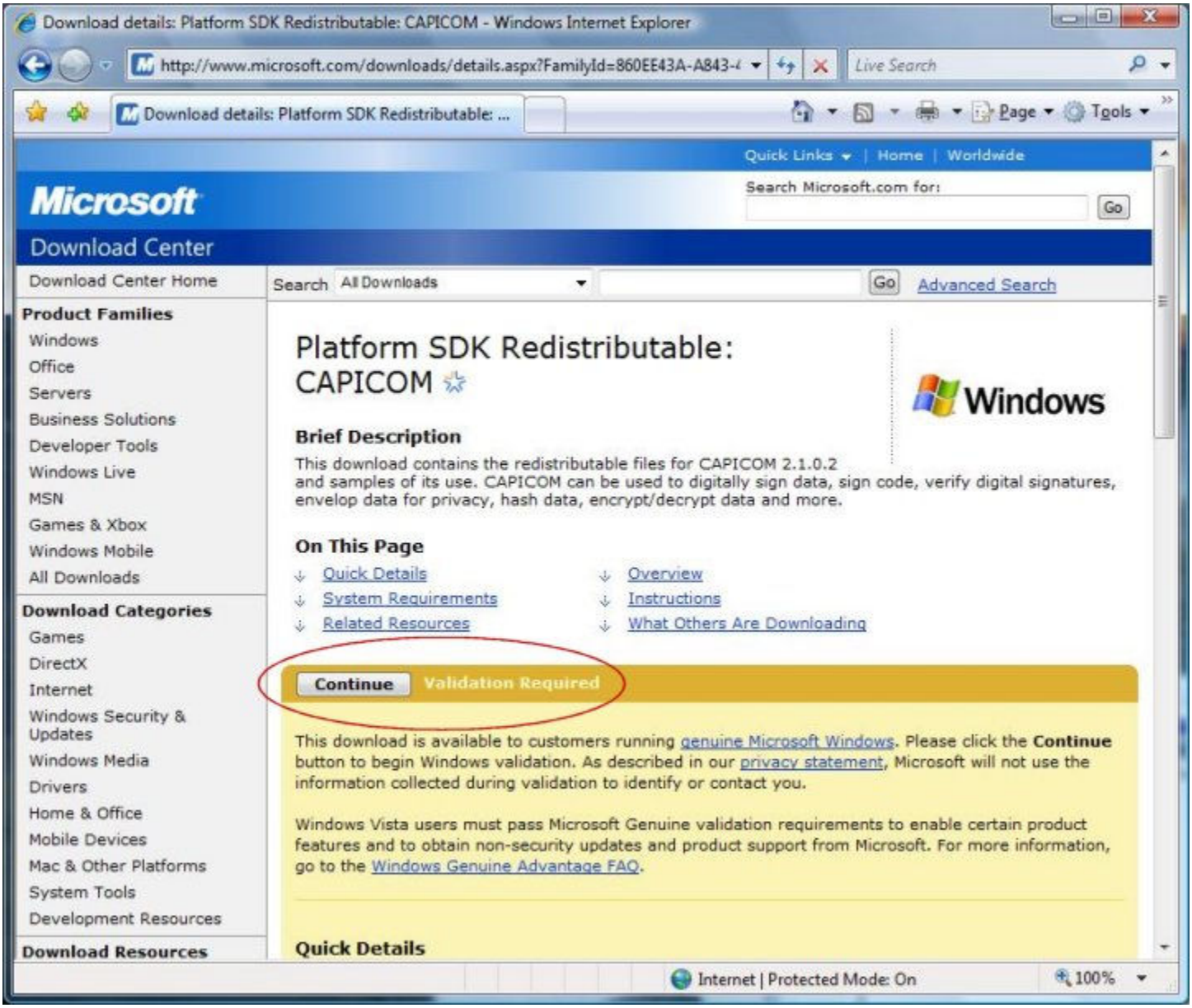
Task 2: Installing CAPICOM on the Microsoft Windows Vista System

CAPICOM is an ActiveX control created by Microsoft that provides access to cryptographic functions. It can be used to digitally sign data, verify digital signature and digital certificates, add certificates to the certificate stores, and to encrypt or decrypt data. CAPICOM is required to support z/OS PKI Services certificate renewal through Internet Explorer browsers.

Microsoft provides two methods for installing CAPICOM:

A. CAPICOM is available as part of the Microsoft Platform Software Development Kit (SDK). Installing CAPICOM in this manner causes more software that required by z/OS PKI Services to be installed on the Microsoft Windows Vista system. However, if the Microsoft Windows Vista client intends to make use of the Microsoft Platform SDK, this is a potential option for installing CAPCIOM.
B. CAPICOM is available as a security patch from Microsoft. Installing CAPCIOM in this manner causes the least amount of software installation for z/OS PKI Services.

Installing CAPICOM through either method requires verification of the Microsoft Windows Vista operating system through Windows Genuine Advantage. When attempting to download the desired version of CAPICOM, the Microsoft website may present you with a page similar to the following, requesting to verify the operating system.



Click on the button labeled `Continue`, and allow the browser to activate any necessary ActiveX controls to perform the verification.

To install CAPICOM, the user must run as the system administrator on the Microsoft Windows Vista system where CAPICOM is to be installed.

A. **Microsoft Platform SDK**

Microsoft provides two methods for installing CAPICOM from the Microsoft Platform SDK. Either method is permitted by z/OS Cryptographic Services PKI Services.

1. **Installing the full Microsoft Platform SDK for Vista.** While this approach causes more software than required by z/OS PKI Services to be installed, it may be a viable option for systems that have installed or plan to install the Microsoft Platform SDK. This method installs and registers CAPICOM on the Microsoft Windows Vista system.

   To locate the Microsoft Platform SDK for Vista, follow the link lsited below:

   http://www.microsoft.com/downloads/details.aspx?familyid=C2B1E300-F358-4523-B479-F53D234CDCCF&displaylang=en

Consult the material on the Web page for instructions on properly downloading and installing the Microsoft Platform SDK.

2. **Installing the CAPICOM subset of the Microsoft Platform SDK.** This approach installs CAPICOM as well as software development samples on the Microsoft Windows Vista system. When this option is selected, the Microsoft Windows Vista system administrator must also register the CAPICOM installation on the Windows VIsta system.

   To locate the download package and the instructions for installing and registering the package, follow the link listed below:

   http://www.microsoft.com/downloads/details.aspx?FamilyID=860EE43A-A843-462F-ABB5-FF88EA5896F6&displaylang=en

B. **CAPICOM Security Patch**

This method installs the minimal set of software necessary for z/OS Cryptographic Services PKI Services.

To access the download for the CAPICOM security patch, follow the link listed below:

http://www.microsoft.com/downloads/details.aspx?FamilyId=CA930018-4A66-4DA6-A6C5-206DF13AF316&displaylang=en

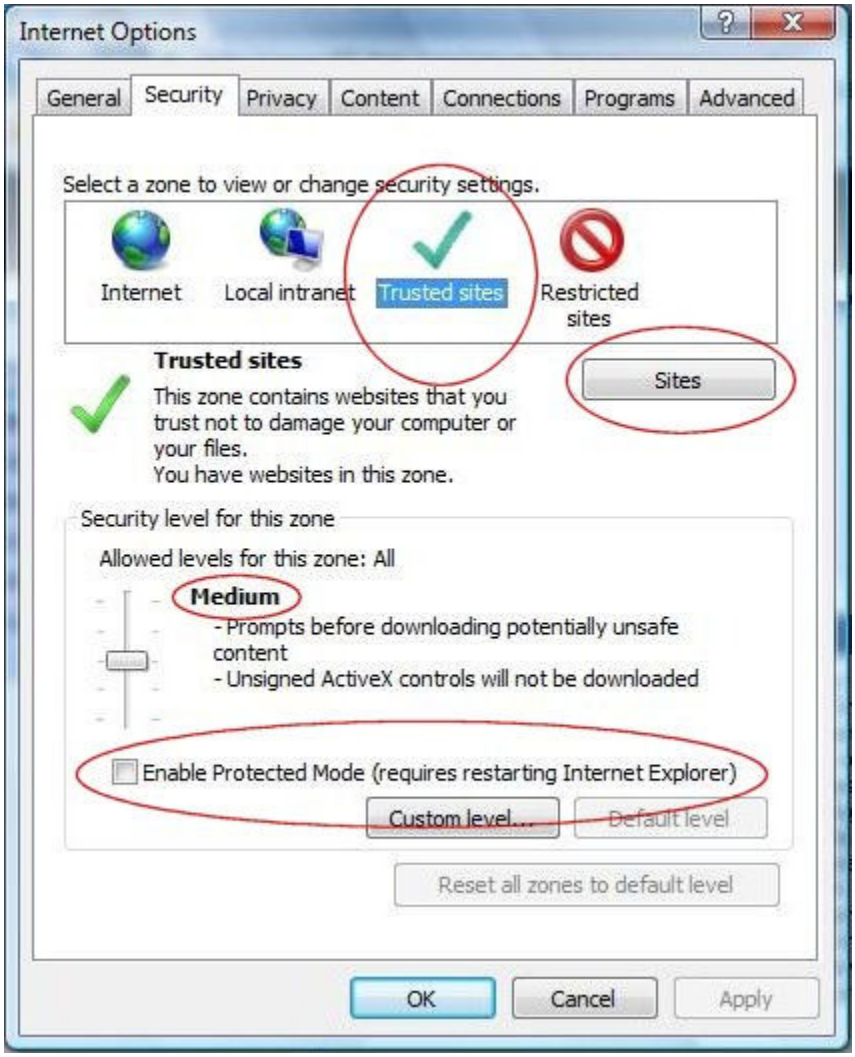
Once the install package has been downloaded, run the install package on the Microsoft Windows Vista system.

After installing CAPICOM on the Microsoft Windows Vista system, proceed to **Task 3: Configuring Internet Explorer to Trust PKI Services.**

Task 3: Configuring Internet Explorer to Trust PKI Services

Any systems using Microsoft Windows Vista version operating systems to request or administer certificates using the z/OS Cryptographic Service PKI Services Web application through Internet Explorer must adjust their Internet Explorer configuration. The configuration adjustment described in this item adds the PKI Services system to the list of trusted sites recognized by Internet Explorer.

- From the Microsoft Windows Vista system, launch the Internet Explorer browser.
- Click on `Tools->Internet Options` to reveal the Internet Options panel.
- Select the `Security` tab to reveal the Security panel.
- Select `Trusted sites` in the window labeled `Select a zone to view of change security settings` by clicking once on the `Trusted sites` icon. This icon is usually presented as a green check mark.

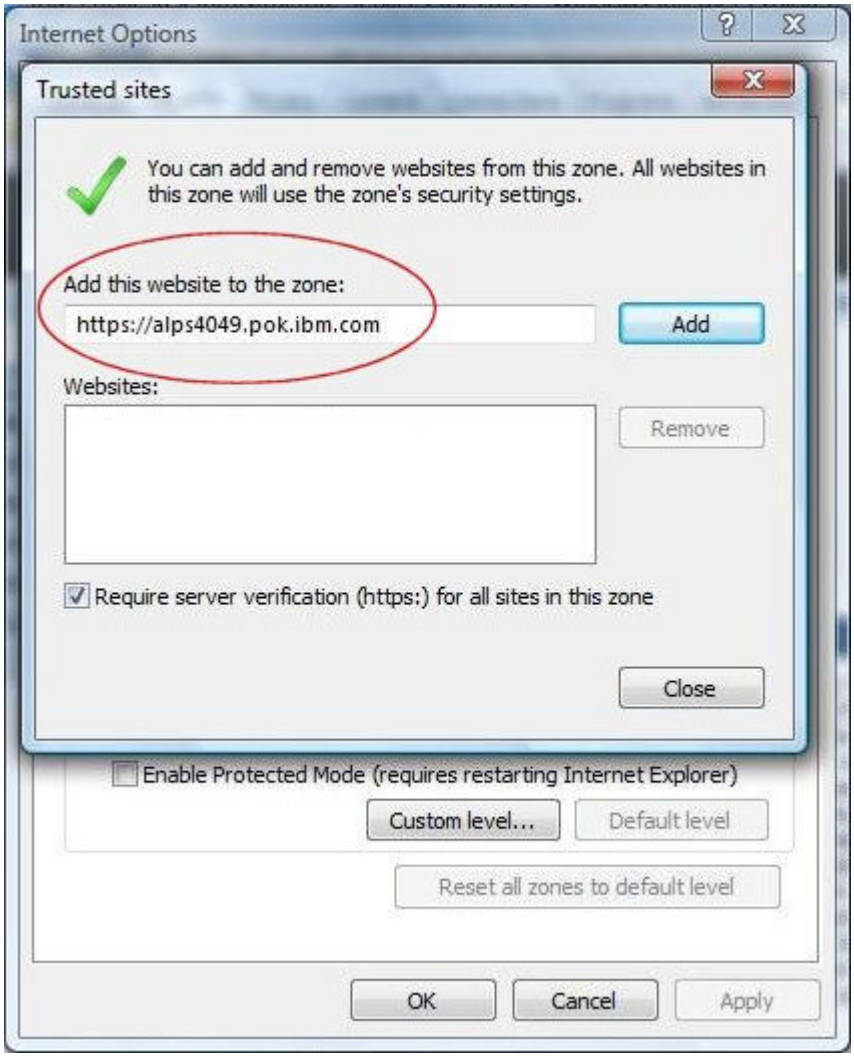


Examine the information on this panel. The z/OS PKI Services Web application functions properly when the slider setting in the `Security level for this zone` area is set to `Medium` and when the `Enable protected mode (requires restarting Internet Explorer)` is **NOT** selected. Adjust these settings if they do not match the z/OS PKI Services recommendations.

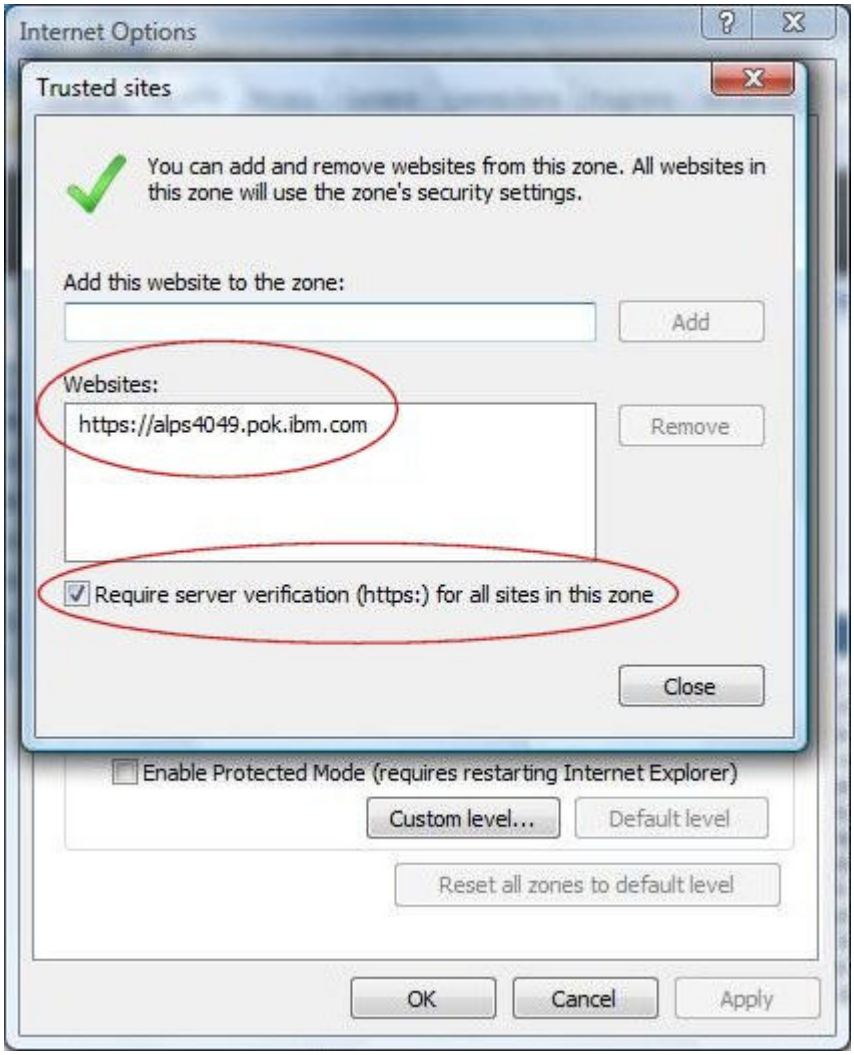
When the `Trusted sites` option is selected, the button `Sites` becomes enabled.

- Click on the `Sites` button that is now presented in the Security tab of the Internet Options panel. This will present the `Trusted sites` panel.
- In the area labeled `Add this website to the zone:`, type the URL for the z/OS Cryptographic Services PKI Services system. Type the URL using **https** as the protocol, not **http**. For example, if the z/OS PKI Services system is `alps4049.pok.ibm.com`, enter the following into the area labeled `Add this website to the zone:`

```
https://alps4049.pok.ibm.com
```



Once the URL has been entered, click on the `Add` button to add this site to the list of trusted sites.



Be sure to leave the box labeled `Require server verification (https:) for all sites in this zone` checked.

- Click on the `Close` button to close the Trusted Sites panel and return to the Security tab of the Internet Options panel.
- Click on the `Apply` button on the Internet Options panel to confirm the configuration changes. After clicking on the Apply button, click on the `OK` button to close the Internet Options panel.
- After the Internet Options panel closes, shut down the Internet Explorer browser to allow these modifications to take effect.
- Proceed to the **Task 4: Installing z/OS PKI Services CA Certificate** instructions below.

Task 4: Installing z/OS PKI Services CA Certificate

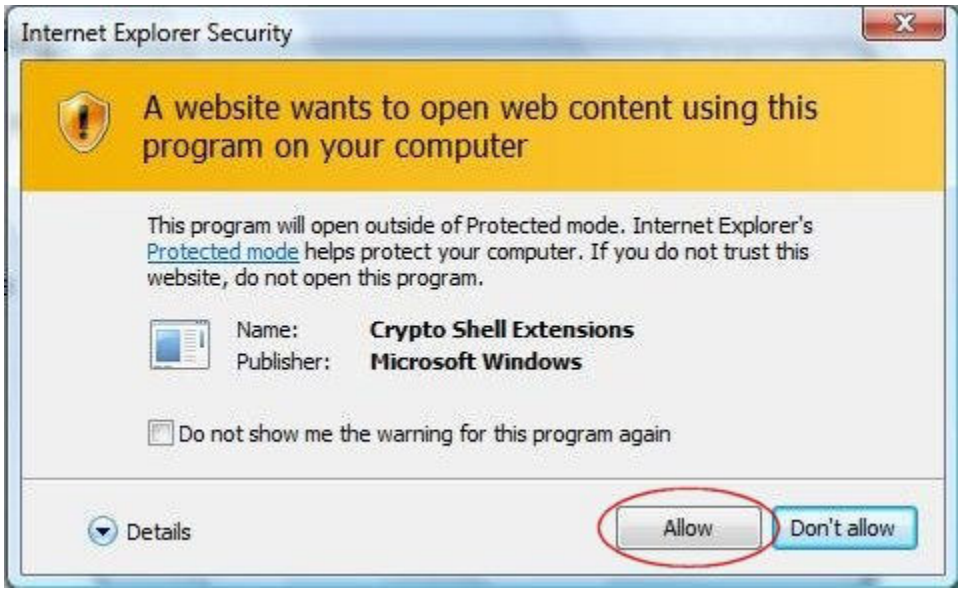
Any systems using Microsoft Windows Vista version operating systems to request or administer certificates using the z/OS Cryptographic Service PKI Services Web application through Internet Explorer must install the z/OS PKI Services certification authority (CA) certificate to enable SSL protected sessions. While the Internet Explorer browser can import the z/OS PKI Services CA certificate, the browser will not correctly install the certificate in the proper location by default. Use the procedure described in this item to properly install the z/OS PKI Services CA certificate.

- On the Microsoft Windows Vista system, start the Internet Explorer browser. In the address field, enter the URL for the z/OS PKI Services Start Page.
- Click on the link labeled `Install the CA Certificate to enable SSL sessions for PKI Services`. Click on this item.

Internet Explorer will present a `File Download - Security Warning` pop-up panel, prompting for a decision to either open or save a file. The pop-up will indicate that this file is from the z/OS PKI Services system. For example, if the z/OS PKI Services system is `alps4049.pok.ibm.com`, the following information should be displayed in this pop-up panel:

```
Type: Security Certificate, number bytes
From: alps4049.pok.ibm.com
```

- Click on the `Save` button in the File Download panel.
- Internet Explorer will display a new `Internet Explorer Security` pop-up panel to warn that a website want to open web content using the browser.

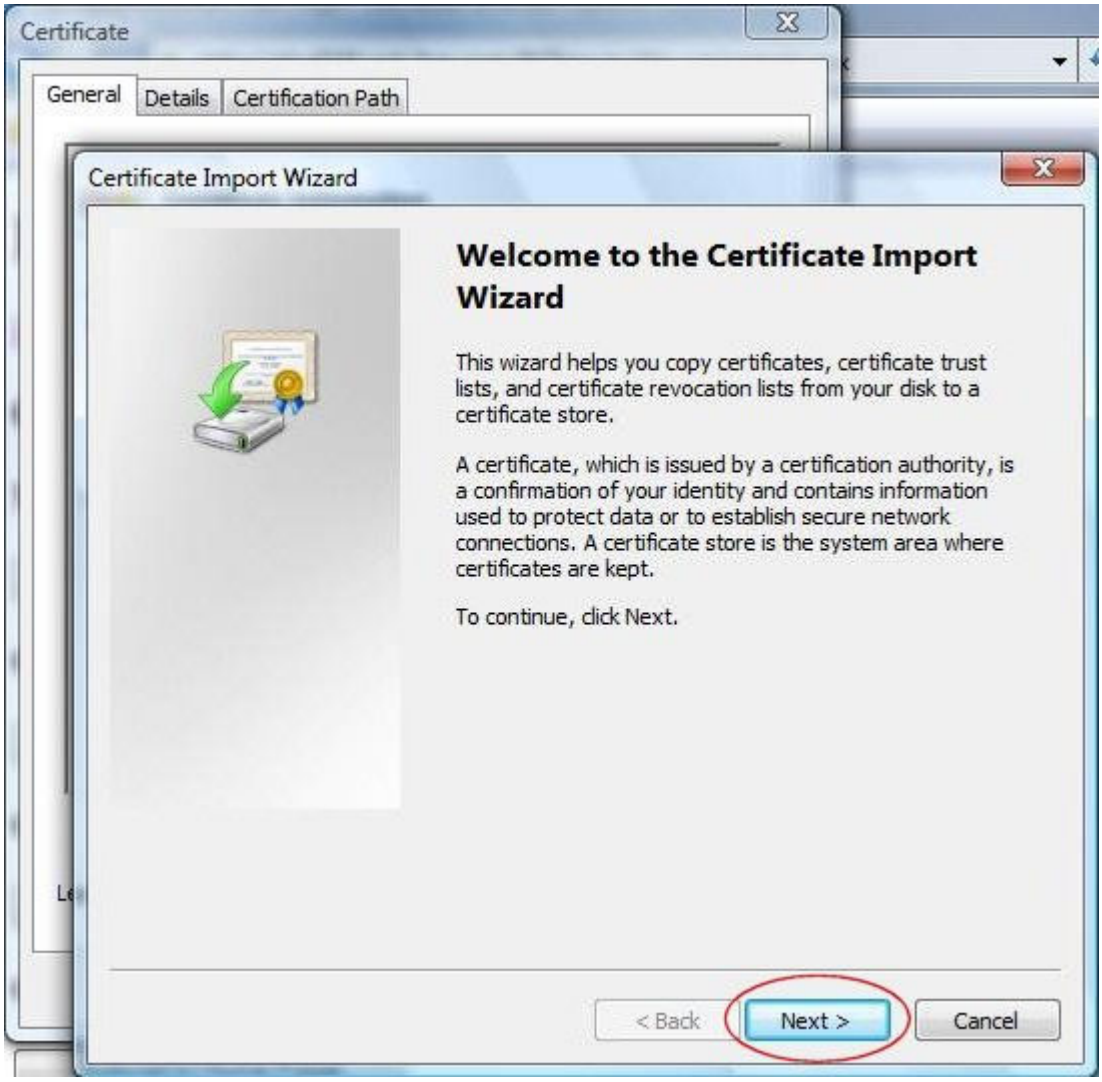


Click on the button labeled `Allow`.

- Internet Explorer will present a new `Certificate` pop-up panel. The panel will indicate that this CA Root certificate is not trusted, and to enable the trust, the certificate must be installed in the Trusted Root Certification Authorities store.

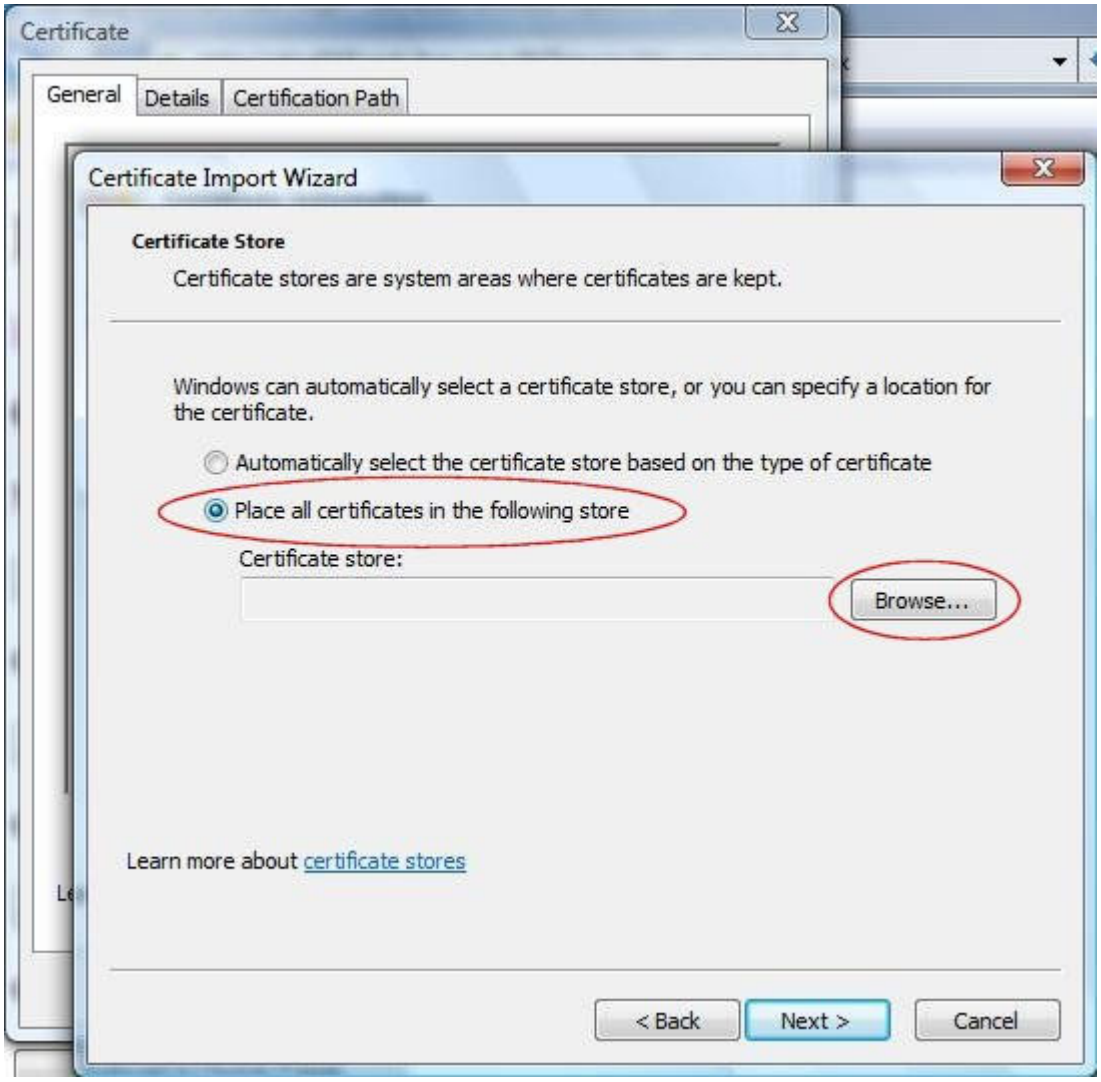
Click on the button labeled `Install Certificate...`.

- Internet Explorer will present a new `Certificate Import Wizard` welcome panel.



Click on the button labeled `Next >`. This displays the `Certificate Store` selections within the Certificate Import Wizard panel.

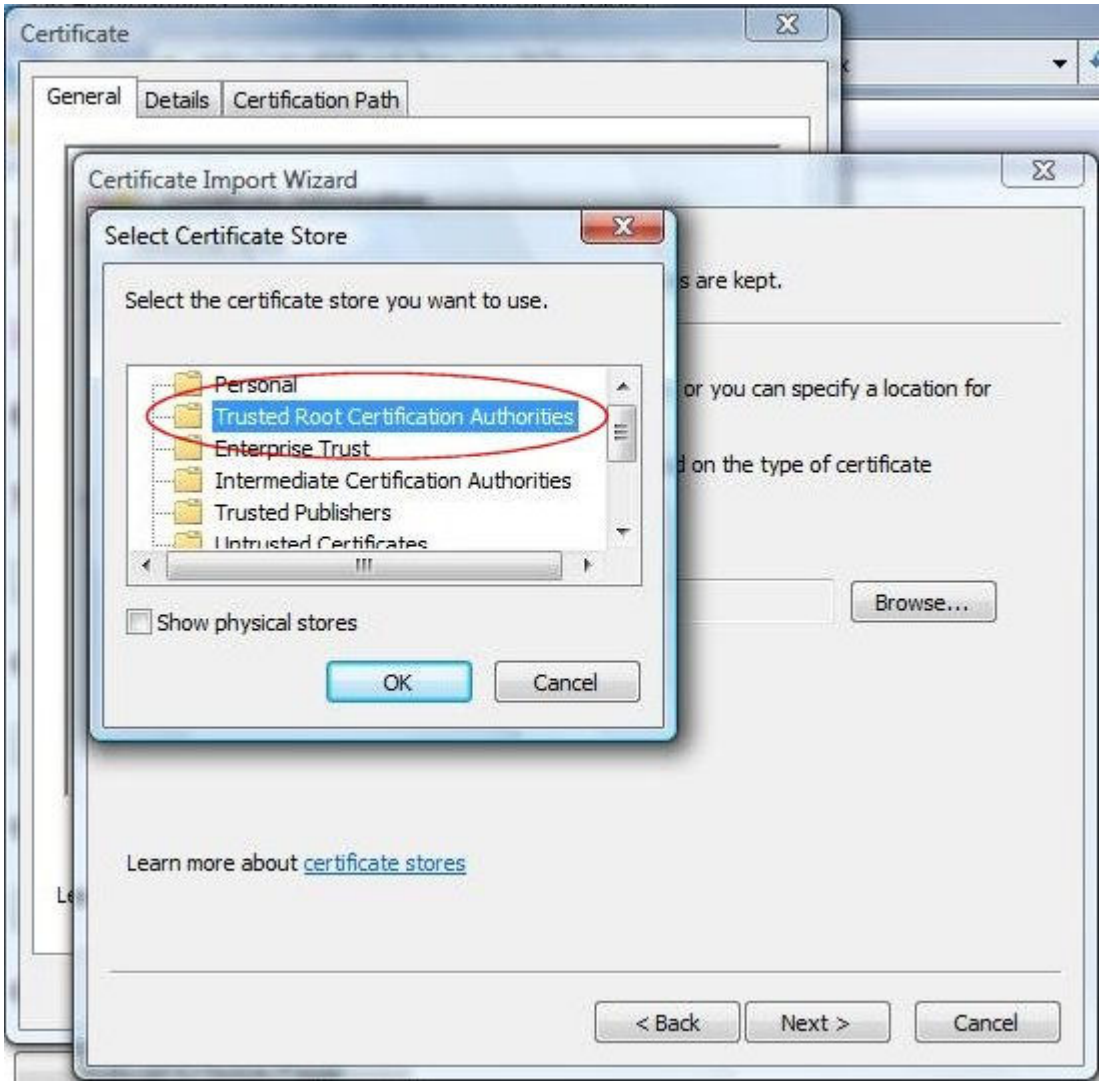
- Click on the radio button labeled `Place all certificates in the following store`.



Click on the button labeled `Browse...`.

Internet Explorer will present a new `Select Certificate Store` pop-up panel. This panel allows for the selection of the proper certificate store.

- Locate the `Trusted Root Certification Authorities` item in the scrolled selection window, and click on this item to highlight and select it.



After selecting the `Trusted Root Certification Authorities` item, click `OK`.

- Click on the button labeled `Next >`. This advances the `Certificate Import Wizard` panel to `Completing the Certificate Import Wizard`.
- Examine the information in the `Completing the Certificate Import Wizard` panel for correctness. In the area labeled `You have specified the following settings:`, the following information should be displayed:

Certificate Store Selected By User	Trusted Root Certification Authorities
Content	Certificate

- Click on the button labeled `Finish`. This will close the `Certificate Import Wizard` panel. Internet Explorer will present a new `Security Warning` pop-up panel, advising that a certificate is about to be installed, and verifying that this is the intended action to take.

Click on the button labeled `Yes`. Internet Explorer will present a confirmation pop-up, indicating that the certificate was successfully imported.

- The z/OS PKI Services CA certificate is now successfully installed. The z/OS PKI Services Web application should now operate successfully through the Internet Explorer browser on this system.