How to setup Secure Telnet with VSE

Server and client authentication

Client Setup with IBM Personal Communications
and Attachmate® EXTRA! X-treme™

Last formatted on: Monday, January 25, 2010

Joerg Schmidbauer
jschmidb@de.ibm.com

Dept. 3229
VSE Development
IBM Lab Böblingen
Schönaicherstr. 220

D-71032 Böblingen
Germany

Disclaimer

This publication is intended to help VSE system programmers setting up infrastructure for their operating environment. The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk. Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment. Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of other companies:

Attachmate® EXTRA! X-treme™ is a trademark of Attachmate Corporation
Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.
Microsoft, Windows, Windows XP, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

Many thanks to **Antonio Zampino** from Zampino Consulting, Switzerland (antonio.zampino@bluewin.ch), for the idea of writing this document and providing much technical input!

Contents

Changes

Oct 18, 2007 – initial version. Pending: client authentication with PCOM 5.7 does not work.
Nov 05, 2007 – added description of how to get client authentication to work with PCOM
Dec 07, 2007 – rework on PCOM key database handling and new section PCOM trace
Apr 15, 2008 – added info about TCP/IP fix for Telnet Listener
June 05, 2008 – added TCP/IP 1.5F zap info
July 07, 2008 – updated info about Telnet Listener
Jan 2010 – added new section 4.2 on TELNETD in native SSL mode

1 Introduction

This paper describes the setup of secure Telnet in various scenarios with VSE acting as server. This involves the creation of RSA key pairs and digital certificates on the server and on the client side. For simplification, we do not purchase certificates from official Certificate Authorities (CAs), but create our own set of so called self signed certificates. Self-signed certificates are not signed by an official CA and therefore work only in a closed test environment.

The following software has been used in the test setup.

- z/VSE 4.1.0 GA version
- TCP/IP for VSE/ESA 1.5E as part of z/VSE 4.1 GA version
- VSE Connector Server as part of z/VSE 4.1.0 (job STARTVCS)
- Microsoft Windows XP Professional, SP2
- Java 1.4.2 from Sun Microsystems
- Keyman/VSE, update from 08/2007
- IBM Personal Communications 5.7 for Windows
- Attachmate® EXTRA! X-treme™ V9 Evaluation for Windows

Note: following fixes are necessary for secure Telnet:

- ZP15F202 (TCP/IP **1.5F.** As shipped TCP/IP 1.5F does not support Secure Telnet connections.)

2 Generating the server key and certificates

The easiest way to generate all necessary keys and certificates for the VSE server side is by using the Keyman/VSE utility which is provided by IBM without warranty for free download from

http://www.ibm.com/servers/eserver/zseries/zvse/downloads/

Keyman/VSE is a Java application, which is typically installed on a Personal Computer. It has the following prerequisites.

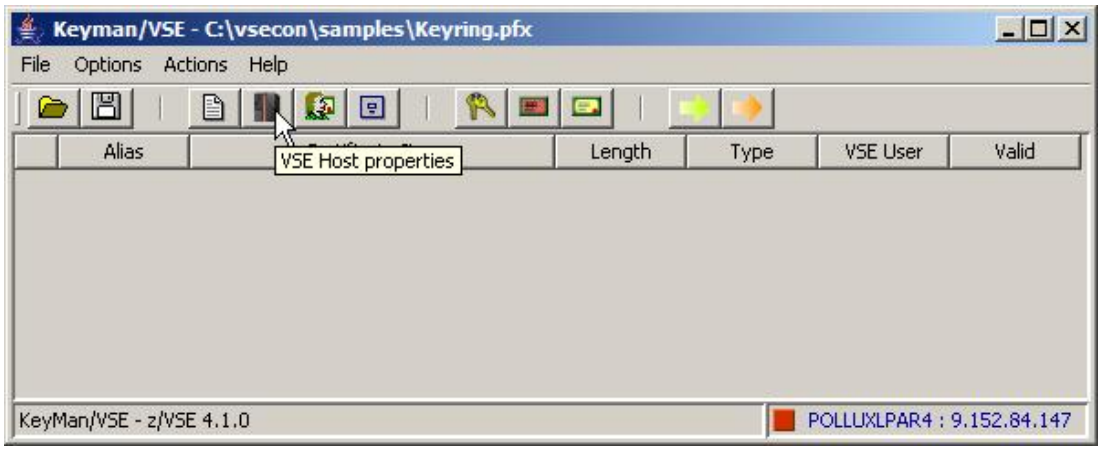
- Java 1.4 or higher on the workstation side
- TCP/IP for VSE/ESA 1.5E on the VSE side
- VSE Connector Server up and running in non-SSL mode on the VSE side

Although Keyman/VSE provides many functions for manually creating keys and certificates, sign certificate requests, and so on, the easiest way for creating the necessary files on VSE is using the Wizard dialog for creating a self-signed keyring. For details about Keyman/VSE functions refer to the HTML-based help of the Keyman tool.

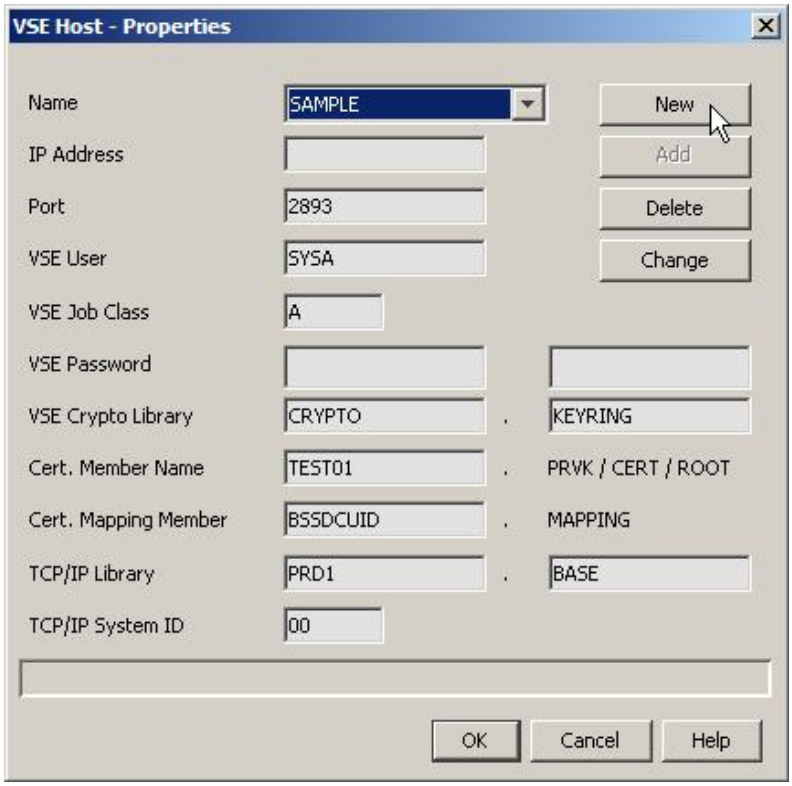
Our first step is to start Keyman/VSE and entering the properties of your VSE system. This information is needed later for sending created keys and certificates to VSE.

2.1 Defining the properties of your VSE system

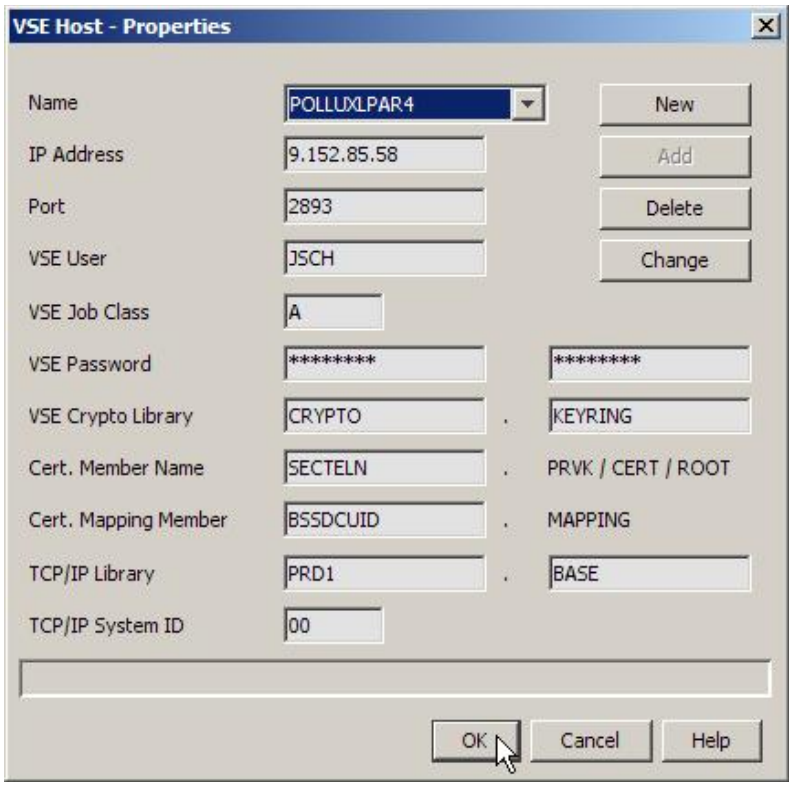
On the main window click on the **VSE host properties** toolbar button.



On the **VSE Host – Properties** dialog box enter the required information for your VSE system. Press the **New** button to create a new VSE host definition.



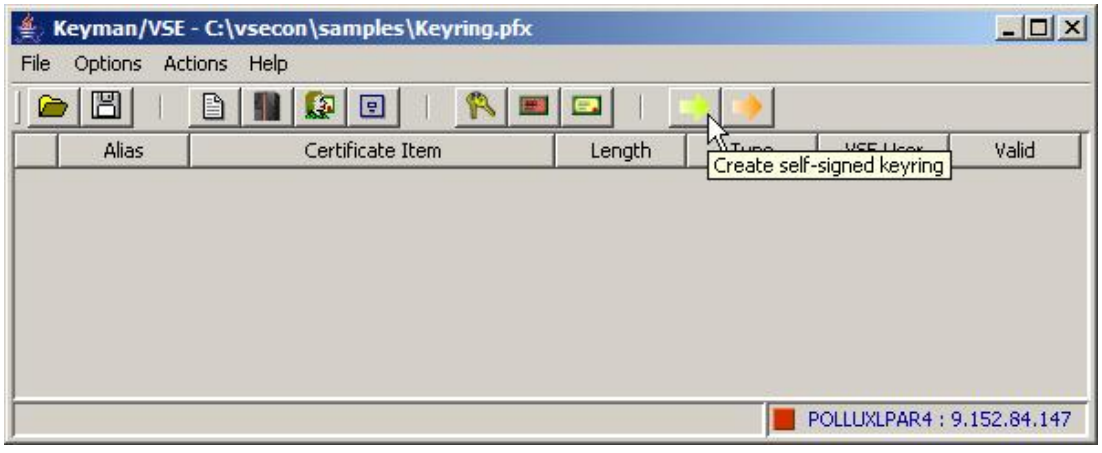
Then enter a unique name for your VSE system, its IP address, the port number of the VSE Connector Server, a VSE user ID together with its password and so on.



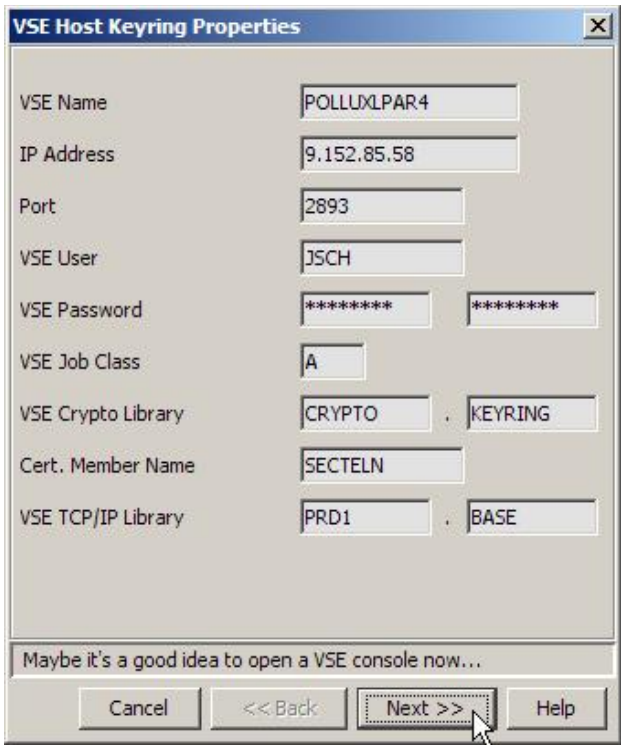
Then press the **Add** button to add the new definition. We are now ready to create the VSE server key and the necessary certificates.

2.2 Creating a self-signed keyring

Click on **the Create self-signed keyring** toolbar button.



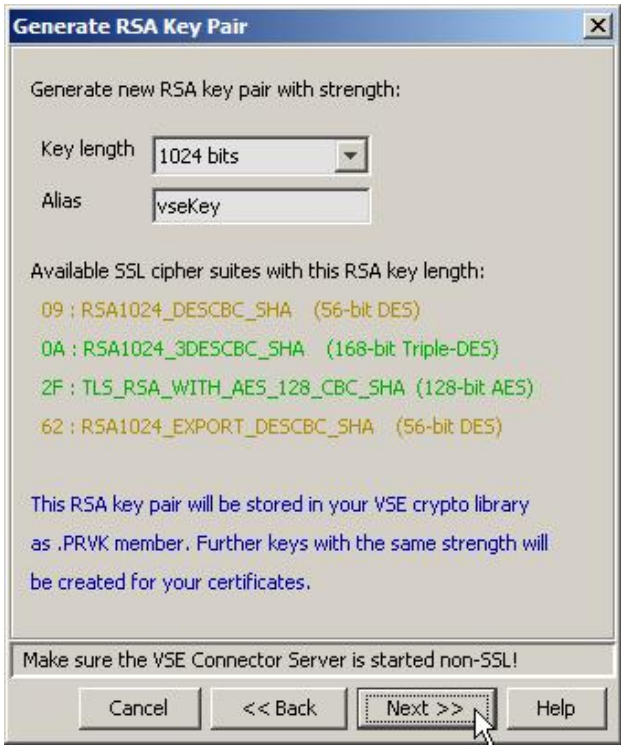
Fill in the required information on the next dialog box



Press **Next**.

On the next dialog specify a password which is used for protecting the local keyring file. You should leave the settings for the encryption of public and private items on **No encryption**. Otherwise there might be problems when reading the file afterwards.

Press **Next**. On the next dialog box specify the key length of your server key and a unique alias string to identify the key. The box shows you a list of available cipher suites with the selected RSA key length. This association has been removed with TCP/IP fix ZP15E250; refer to the notes below Table 1 on page 14.



Press **Next**. On the following dialog box specify the personal information for the VSE ROOT certificate.

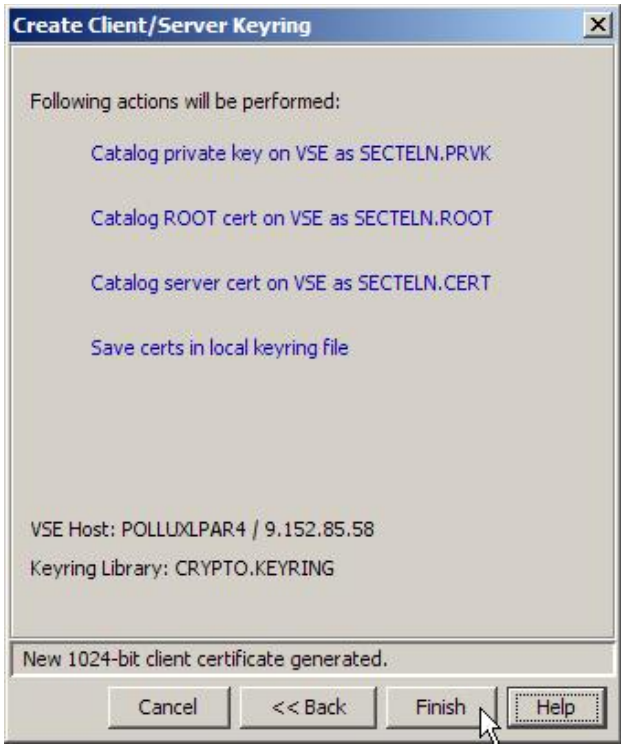
Press **Next**. On the following dialog box specify the personal information for the VSE server certificate.

Note: Attachmate Extra! in some cases requires the Common Name to be identical to the VSE IP address in order to accept the server certificate during the SSL handshake. Refer to section Attachmate Extra! session setup on page 30 for more information and see right hand picture below.

Press **Next**.

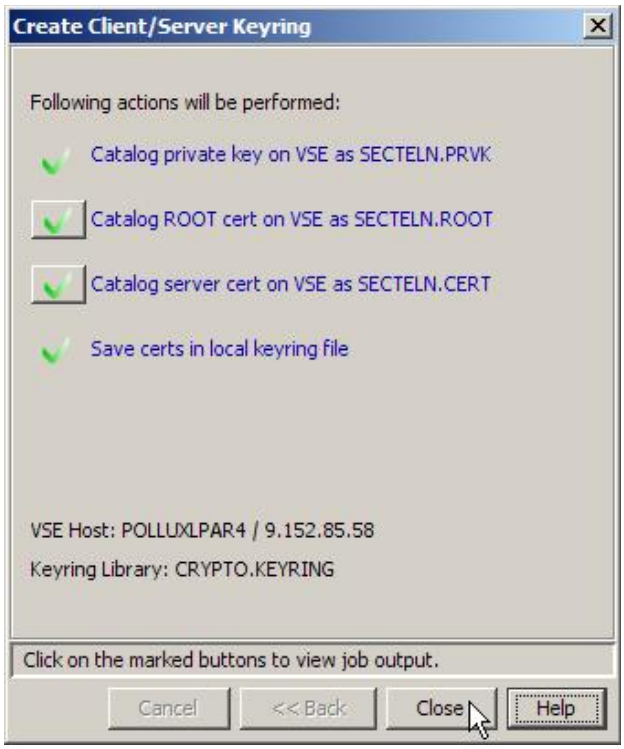
A client certificate is only needed for client authentication (refer to chapter Setting up for client authentication on page 23).

Press **Next**.



Press **Finish**.

This will send all items to VSE and save the certificates in the local keyring file.



Press **Close**.

Now you have three VSE library members cataloged into CRYPTO.KEYRING. The PRVK member contains the RSA key pair, the ROOT member contains the self-signed VSE ROOT certificate, and the CERT member contains the VSE server certificate.

```
LD SECTELN.*

DIRECTORY DISPLAY    SUBLIBRARY=CRYPTO.KEYRING      DATE: 2007-09-21
                                                    TIME: 19:38
-------------------------------------------------------------------
 M E M B E R      CREATION   LAST     BYTES    LIBR CONT SVA  A- R-
NAME     TYPE     DATE      UPDATE   RECORDS   BLKS STOR ELIG MODE
-------------------------------------------------------------------
SECTELN  CERT     07-09-21   -  -      724 B       1 YES   -   -   -
SECTELN  PRVK     07-09-21   -  -     2048 B       3 YES   -   -   -
SECTELN  ROOT     07-09-21   -  -      686 B       1 YES   -   -   -
L113I RETURN CODE OF LISTDIR IS  0
L001A ENTER COMMAND OR END
```

You can also close the Keyman/VSE tool now. As we don't need the server key on the client side, the key was not saved to the local file.

We will need the client keyring file later in order to import the self-signed root certificate into Personal Communications.

3 Setting up a TELNETD

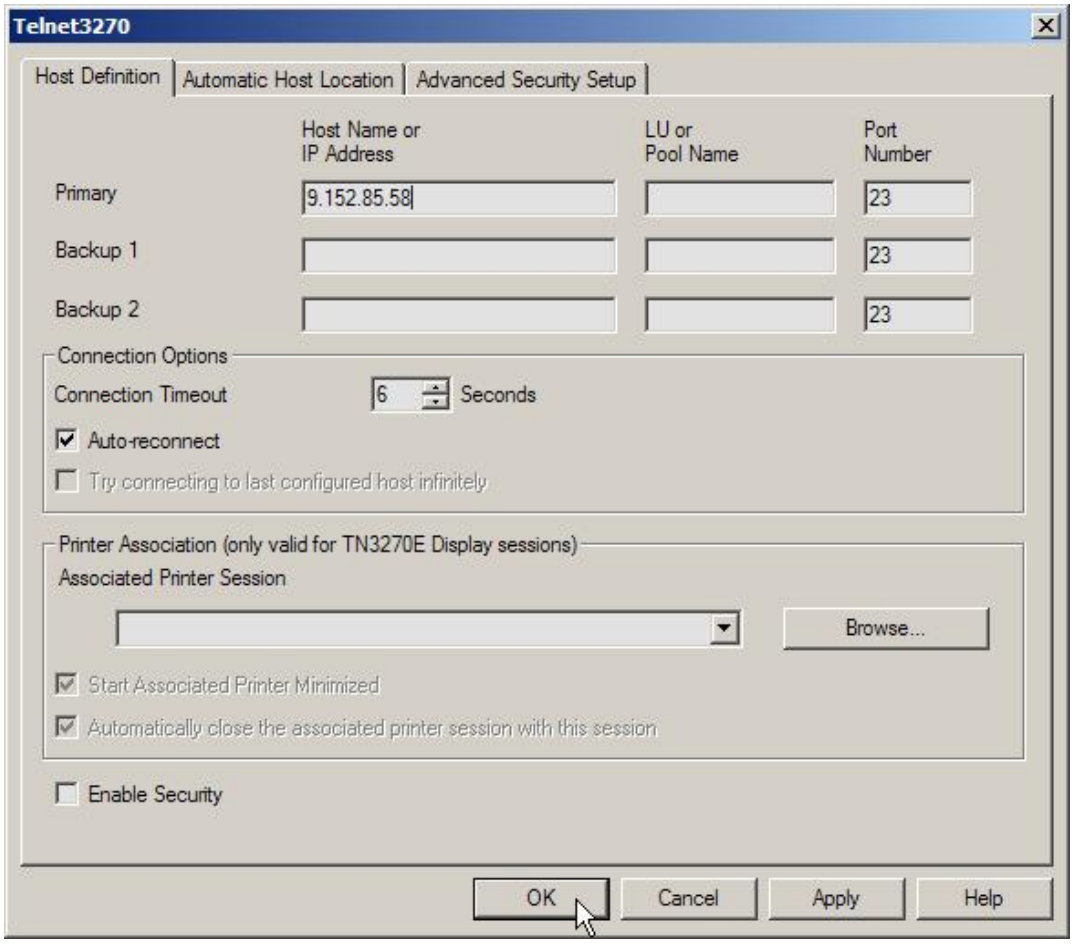
First, let's do the basic setup for unsecure Telnet. We will later add the definitions for secure Telnet. The following command defines a standard Telnet daemon.

```
DEFINE TELNETD,ID=LU,TERMNAME=TELNLU,TARGET=DBDCCICS,PORT=23,COUNT=4, -
    LOGMODE=S3270,LOGMODE3=D4B32783,LOGMODE4=D4B32784, -
    LOGMODE5=D4B32785,POOL=YES
```

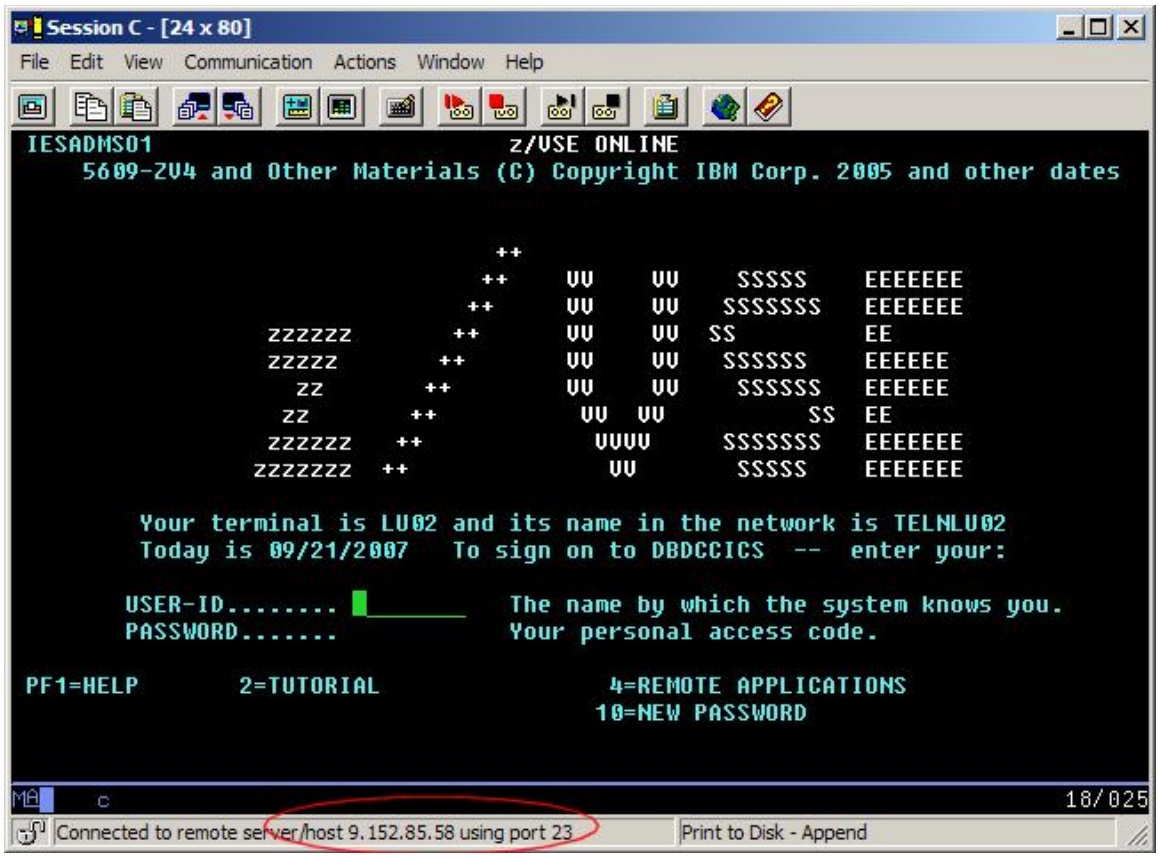
The daemon startup is shown on the VSE console.

```
F7 0097 0030: TEL900I Daemon Startup Telnet Termname: TELNLU04 Port: 23
F7 0097 002F: TEL900I Daemon Startup Telnet Termname: TELNLU03 Port: 23
F7 0097 002E: TEL900I Daemon Startup Telnet Termname: TELNLU02 Port: 23
F7 0097 002D: TEL900I Daemon Startup Telnet Termname: TELNLU01 Port: 23
```

We can now immediately use this daemon with a Telnet 3270 capable client. Following picture is taken from IBM Personal Communications, setting up the session with the VSE IP address.



The following picture shows a VSE signon screen connected via Telnet.



The next chapter shows how our previously defined Telnet daemon is SSL enabled.

4 VSE host setup for secure Telnet

Setting up SSL for the Telnet protocol is based on the SSL daemon (TLSD) provided by TCP/IP for VSE/ESA. In general there are two modes available:

- SSL in *native* mode. Hereby SSL traffic goes directly to the SSL enabled daemon on VSE. Native mode is supported by HTTPD, FTPD, and TELNETD. In addition to one of these daemons you have to define a TLSD daemon that defines the SSL parameters.

- SSL in *pass-through* mode, where a TLSD serves as a proxy. In this case encrypted traffic goes to the TLSD, which in turn passes the data unencrypted to an HTTPD, FTPD, or TELNETD. The PASSPORT parameter is used to route SSL traffic from an unsecured daemon to the TLSD.

4.1 Setting up pass-through mode

Following TLSD definition is used to secure our previously defined TELNETD. Note that the PORT and PASSPORT parameters are different.

```
DEFINE TLSD,ID=TLSDTELNET,                    Id of this SSL/TLS daemon
   PORT=992,                                  Secure telnet port
   PASSPORT=23,                               Port data is passed to
   CIPHER=2F350A0962,                         Allowed cipher suites
   CERTLIB=CRYPTO,                            Library name
   CERTSUB=KEYRING,                           Sublibrary name
   CERTMEM=SECTELN,                           Member name
   TYPE=1,                                    SSL server authentication
   MINVERS=0300,                              Minimum version required
   DRIVER=SSLD                                Driver phase name
```

Make sure that you specify the same member name (here SECTELN) used when uploading the keyring files to VSE (see section Creating a self-signed keyring on page 6). The daemon startup is shown on the VSE console.

```
F7 0097 0036: TLS900I Daemon Startup Transport Security Layer SSLD
       =81640040
```

Table 1 shows the available cipher suites on VSE. Parameter CIPHER in the TLSD definition lists the hex codes of the ciphers you want to use with this TLSD. When the secure Telnet session is established, the client and server will negotiate one of these cipher suites to be used. The session will fail if there is no cipher suite supported by both sides.

Hex Code	Cipher Suite	Handshaking (*)	Encryption	Min. TCP/IP
01	SSL_RSA_WITH_NULL_MD5	512	None	1.5D
02	SSL_RSA_WITH_NULL_SHA	512	None	1.5D
08	SSL_RSA_EXPORT_WITH_DES40_CBC_SHA	512	40 bits	1.5D
09	SSL_RSA_WITH_DES_CBC_SHA	1024	56 bits	1.5D
0A	SSL_RSA_WITH_3DES_EDE_CBC_SHA	1024	168 bits	1.5D
2F	TLS_RSA_WITH_AES_128_CBC_SHA	1024 / 2048	128 bits	1.5E
35	TLS_RSA_WITH_AES_256_CBC_SHA	1024 / 2048	256 bits	1.5E
62	RSA1024_EXPORT_DESCBC_SHA	1024	56 bits	1.5D

Table 1: available cipher suites on VSE

Notes:

- When using 2048-bit keys you need a Crypto Express2 or PCI-X Cryptographic Coprocessor card.
- AES support was introduced with TCP/IP fix ZP15E214.
- AES-128 is available as hardware function on IBM System z9 processors and is much faster than the software implementations provided by TCP/IP. It is used transparently by TCP/IP when available.
- (*) TCP/IP fix ZP15E250 removes the restriction of allowing some cipher suites only with a specific RSA key length. If you look at the RFC2240 for TLS you will notice that it does not have a RSA key size associated with the specific cipher suites. Any cipher suite can now be used with any of the RSA key sizes.

4.2 Setting up SSL native mode

SSL native mode is used automatically when the DEFINE TLSD contains *the same port number* for the PORT and PASSPORT parameters as shown below.

```
DEFINE  TLSD,ID=TLSDTELNET,                    -
             PORT=992,                         -
             PASSPORT=992,                     -
             CIPHER=2F350A0962,                -
             MINVERS=0300,                     -
             CERTLIB=CRYPTO,                   -
             CERTSUB=KEYRING,                  -
             CERTMEM=SECTELN,                  -
             TYPE=1,                           -
             DRIVER=SSLD

DEFINE TELNETD,ID=TELNVSSL,TCPAPPL=TNSSL0,TARGET=DBDCCICS, -
               COUNT=3,MENU=MENUZ,TYPE=VTAM,POOL=YES,PORT=992, -
               DRIVER=TELNETD
```

With the above definition the TELNETD will natively support SSL, but pick up the necessary SSL configuration information from the DEFINE TLSD keywords.

4.3 Setting up a Telnet Listener daemon

When additional security is required, such as restricting Telnet traffic to specific IP addresses, you may use the following variant of setting up Secure Telnet on VSE.

Note: this scenario requires TCP/IP for VSE/ESA 1.5F with fix ZP15F202. See fix description on

http://www.csi-international.com/csi-support/zaps15f.htm#ZP15F202

In this example, the listener daemon only accepts connections from the below specified IP address, which usually belongs to a dedicated Terminal server. Note that this is only possible with TN3270E.

```
DEFINE TELNETD,TN3270E=L,PORT=992,POOL=YES,DRIVER=TELNETD, -
  ID=TLSL1,IPADDR=3.196.98.105
*
DEFINE TLSD,ID=TLS2,PORT=992,PASSPORT=992,CIPHER=090A62, -
  CERTLIB=CRYPTO,CERTSUB=KEYRING,CERTMEM=SECTEL01,TYPE=1,DRIVER=SSLD, -
  MINVERS=0300
*
DEFINE TELNETD,TERMNAME=TLS32AA,ID=TLS32AA,POOL=YES, -
  TN3270E=E,TYPE=VTAM,MENU=MENU2,LOGMODE=SP3272QN, -
  LOGMODE3=SP3272QN,LOGMODE4=SP3272QN,LOGMODE5=SP3272QN,PORT=992, -
  DRIVER=TELNETD
```

This JCL adds the related VTAM definitions.

```
* $$ JOB JNM=CAT,CLASS=0
// JOB CAT      TEST
// EXEC LIBR,PARM='A S=PRD2.CONFIG'
CATALOG  TLSD.B   REPLACE=YES
TLSD     VBUILD TYPE=APPL
TLS32AA  APPL  AUTH=(ACQ),MODETAB=IESINCLM,EAS=1
TELNLU01 APPL  AUTH=(ACQ),MODETAB=IESINCLM,EAS=1
TELNLU02 APPL  AUTH=(ACQ),MODETAB=IESINCLM,EAS=1
TELNLU03 APPL  AUTH=(ACQ),MODETAB=IESINCLM,EAS=1
```

```
/+
/&
* $$
```

5 Client setup with IBM Personal Communications

Basically, we have to import our self-signed root certificate into the PCOM key database. It is important to import the PFX file into the PCOM key database to not loose the contained private keys.

5.1 Importing the VSE certificates into PCOM

Open the IBM Key Management tool, which is part of the Personal Communications installation. You will find the tool under **Utilities - Certificate Management**.

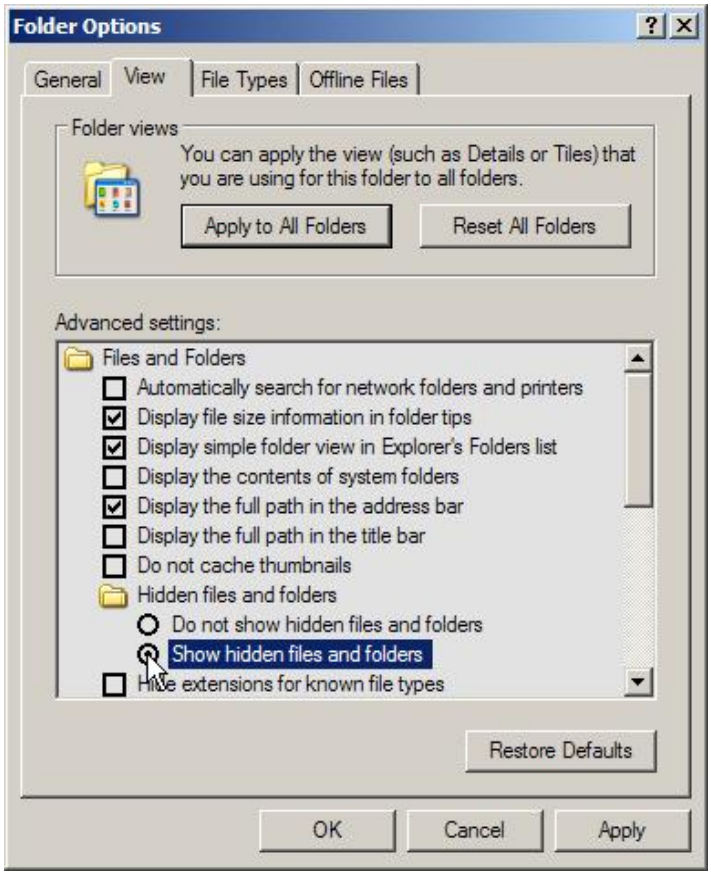
5.1.1 Location of the PCOM key database on Windows

On Windows, the PCOM key database is located in folder

C:\Documents and Settings\All Users\Application Data\IBM\Personal Communications

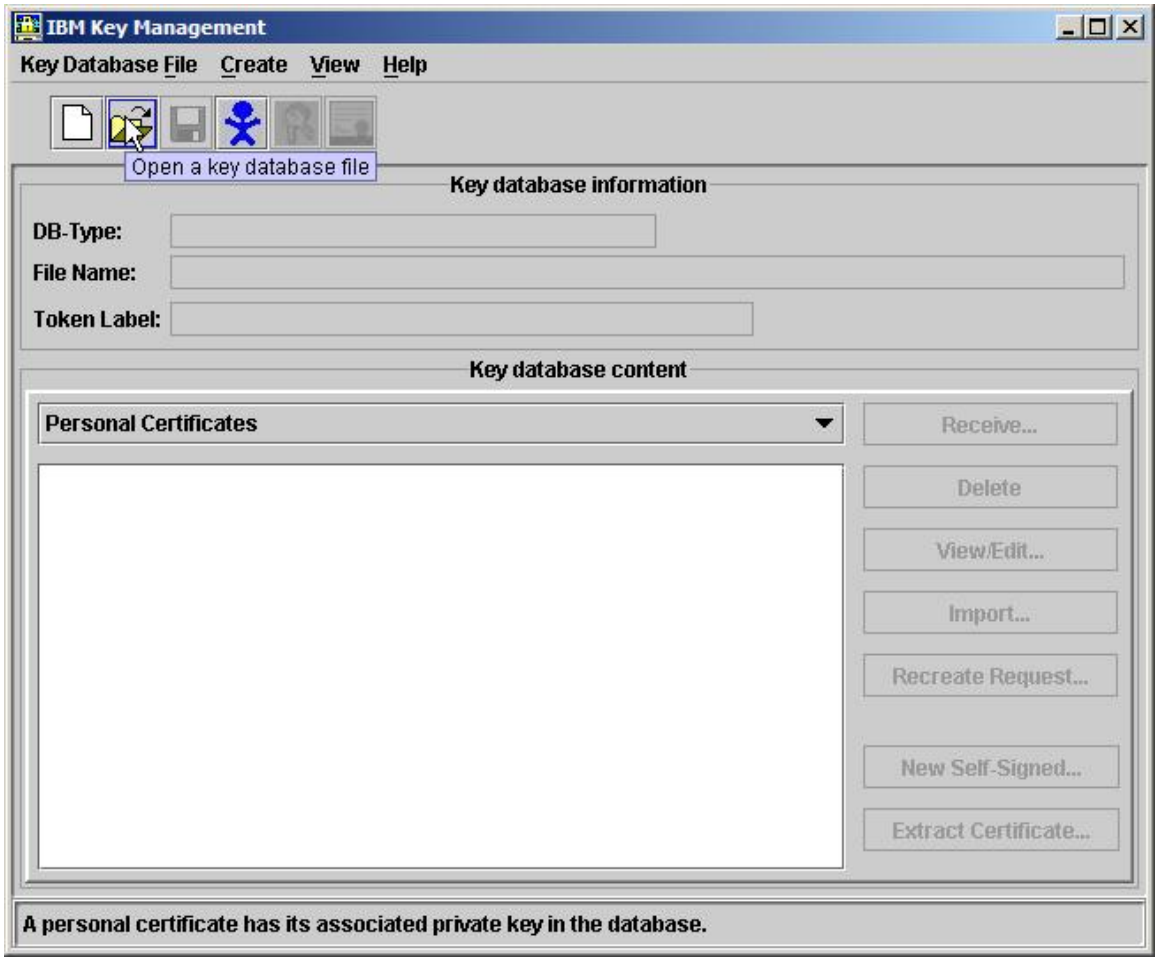
This folder is not visible via the Windows Explorer when your current folder options specify to not show hidden files and folders. You can change this in the Windows Explorer via menu Tools – Folder options.

In the below dialog box click on **Show hidden files and folders.**

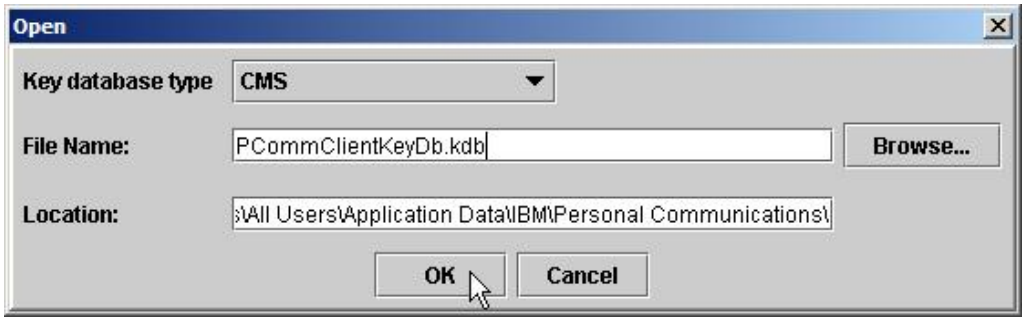


5.1.2 Opening the PCOM key database

Now open the PCOM key database.



If the fields in the below box are empty, browse to the KDB location, see Location of the PCOM key database on Windows on page 16.



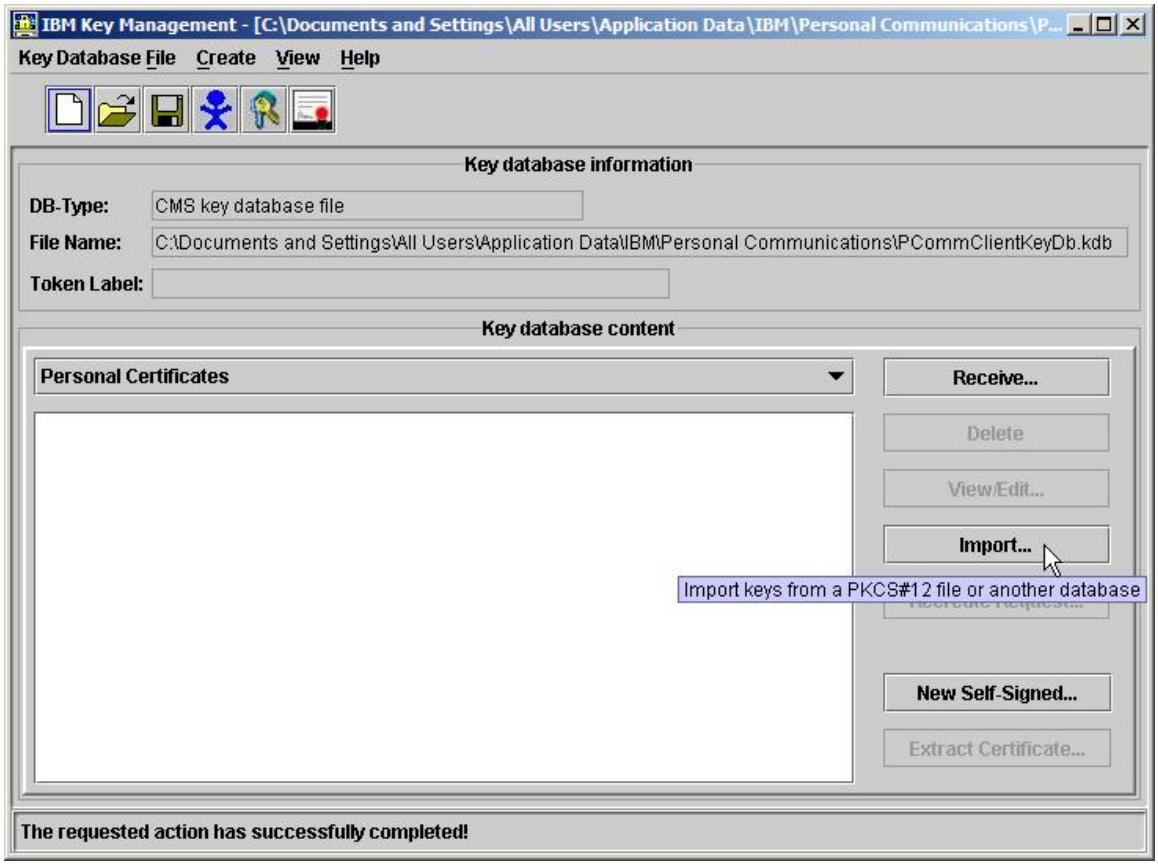
Press **OK**.

You are now prompted for the key database password. The default password for the PCOM key database is *pcomm*.

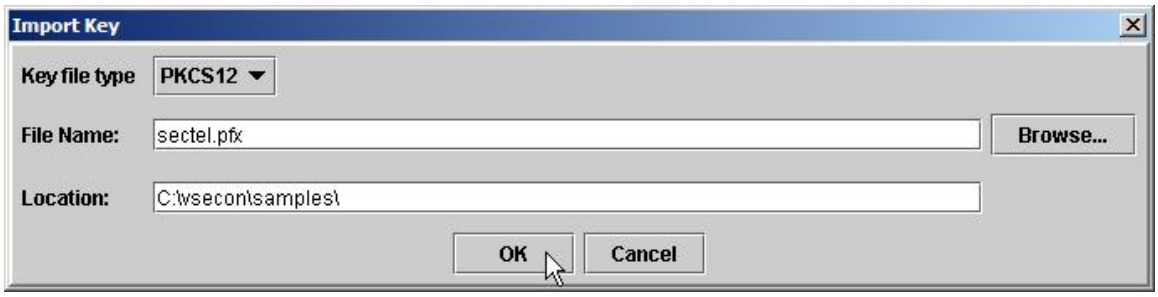
Enter the password and press **OK**.

5.1.3 Importing the PFX file into the PCOM key database

It is important to import the certificate with its private key, i.e. to import the certificate as a **Personal certificate**. In the PCOM key management GUI select **Personal Certificates** and click on **Import**. This will also import our VSE client certificate, which is needed later for client authentication.



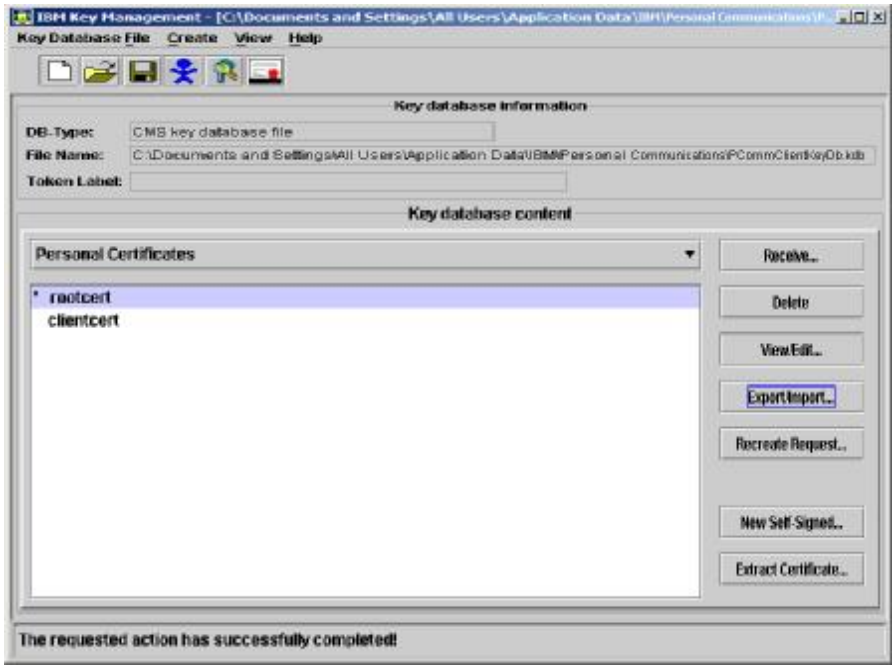
On the next dialog box select Key file type **PKCS12** and enter the name and location of the VSE keyring file.



Press **OK**.

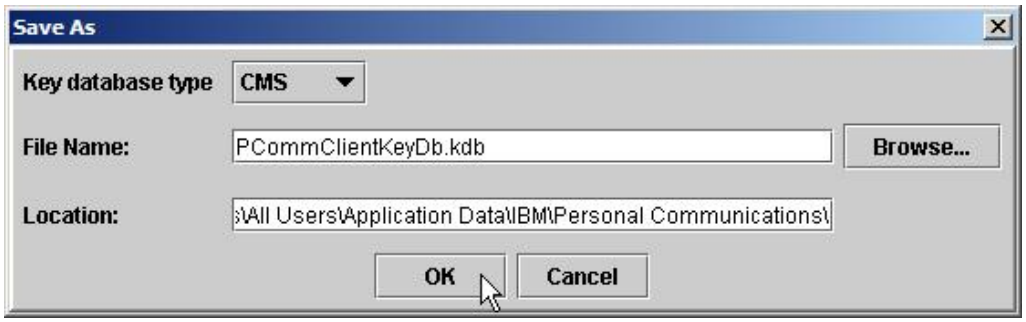
Enter the password of the keyring file and press **OK**.

Note: exporting the VSE certificates in base64 text form from Keyman/VSE and using the **Add** function in the PCOM key management tool, would result in loosing the private keys and the certificates would be imported as signer certificates, rather than as personal certificates. In this case client authentication would not work.



Now save the key database file and close the PCOM certificate management tool. I would recommend changing the default password before saving the KDB. You can do this via menu **Key Database File** – **Change password.** After changing the password, use option **Stash password** to save an encrypted copy of the password in a separate file for verifying the password in future.

To save the KDB select menu **Key Database File** – **Save as…** and proceed with the dialogs.

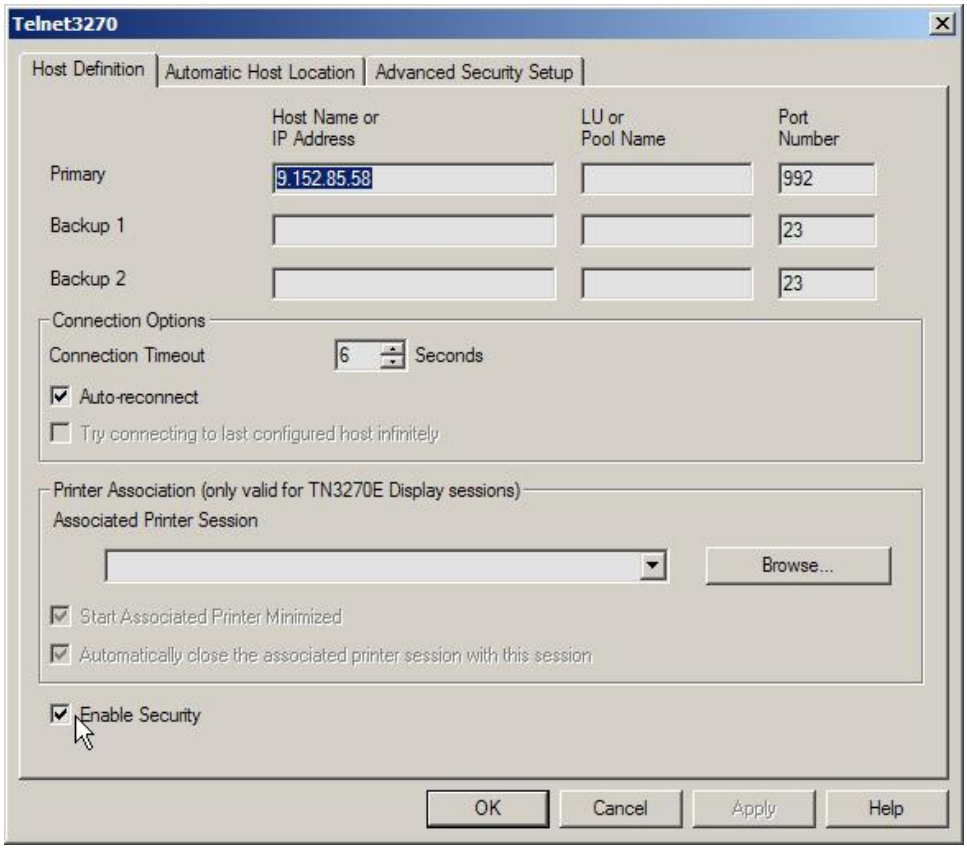


Press **OK**.

Enter the KDB password and press **OK**.

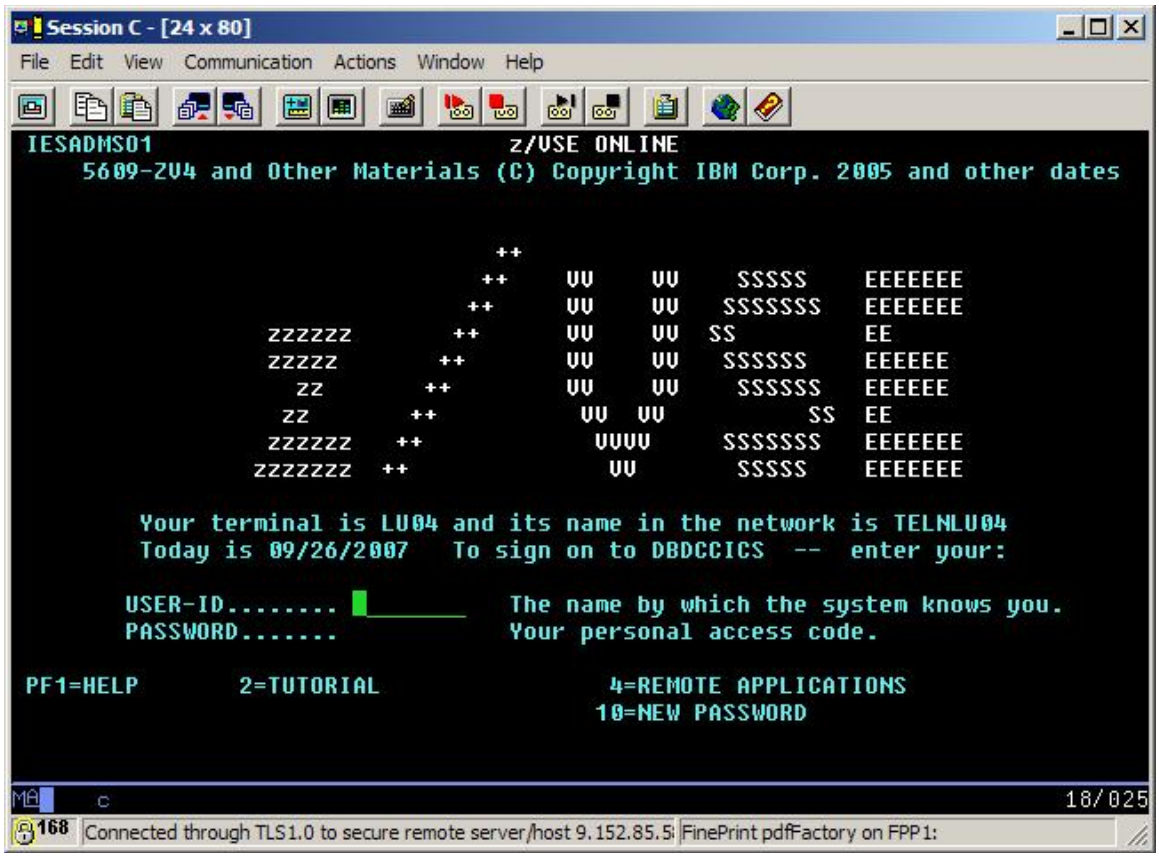
5.2 Starting a secure session

In the PCOM session window we have to change the port number to the secure Telnet port **992** and we have to check the box labeled **Enable Security**.



That's all. When now connecting again to VSE, the connection is secured.

Note: If the Enable Security checkbox is grayed out, then SSL support is not installed. Check Personal Communications Installation information to see if an error occurred during product installation.



A closed lock icon is displayed in the bottom left-hand corner of the session window to indicate that the session is encrypted.

The number **168** indicates the key length of the symmetric key used in this session. Here, it tells us that we are connected with Triple-DES (3 times 56 bits DES). Typical values are 0 (no encryption), 40 (DES with 40-bit key), 56 (DES with 56-bit key), 128 (AES with 128-bit key), 168 (Triple-DES). For a list of supported ciphers refer to Table 1: available cipher suites on VSE on page 14.

5.3 Setting up for client authentication

SSL client authentication provides more security than server authentication, because both communication partners provide a certificate in order to establish trust. To setup client authentication for secure Telnet we have to do four things:

1. Change the TLSD definition on the VSE side to enable client authentication. This is done via the TYPE parameter of the TLSD definition.

2. Make sure our Client Certificate is contained in the Personal Communications key database; see section Importing the VSE certificates into PCOM on page 16. During the SSL handshake the server will request the client's certificate.

3. Restart the TLSD in client authentication mode (type=2).

4. Change the PCOM session definition for client authentication.

5.3.1 Change TLSD for client authentication

In the TLSD definition we have to change the TYPE to 2 (client authentication).

```
DEFINE TLSD,ID=TLSDTELNET,                    Id of this SSL/TLS daemon
   PORT=992,                                  Secure telnet port
   PASSPORT=23,                               Port data is passed to
   CIPHER=2F350A096208,                       Allowed cipher suites
   CERTLIB=CRYPTO,                            Library name
   CERTSUB=KEYRING,                           Sublibrary name
   CERTMEM=SECTELN,                           Member name
   TYPE=2,                                    SSL client authentication
   MINVERS=0300,                              Minimum version required
   DRIVER=SSLD                                Driver phase name
```

5.3.2 Restart TLSD for client authentication

We now have to restart the TLSD with changed TYPE parameter.

```
101 delete tlsd,id=TLSDTELNET
F7-0101 IPN300I Enter TCP/IP Command
F7 0097 00E1: TLS903I Daemon Shutdown TLS Id:TLSDTELN
101 DEFINE TLSD,ID=TLSDTELNET,PORT=992,PASSPORT=23,CIPHER=2F350A096208, -
F7-0101 IPN300I Continue TCP/IP Command
101 CERTLIB=CRYPTO,CERTSUB=KEYRING,CERTMEM=SECTELN,TYPE=2, -
F7-0101 IPN300I Continue TCP/IP Command
101 MINVERS=0300,DRIVER=SSLD
F7-0101 IPN300I Enter TCP/IP Command
F7 0097 0212: TLS900I Daemon Startup Transport Security Layer SSLD
        =81640040
```

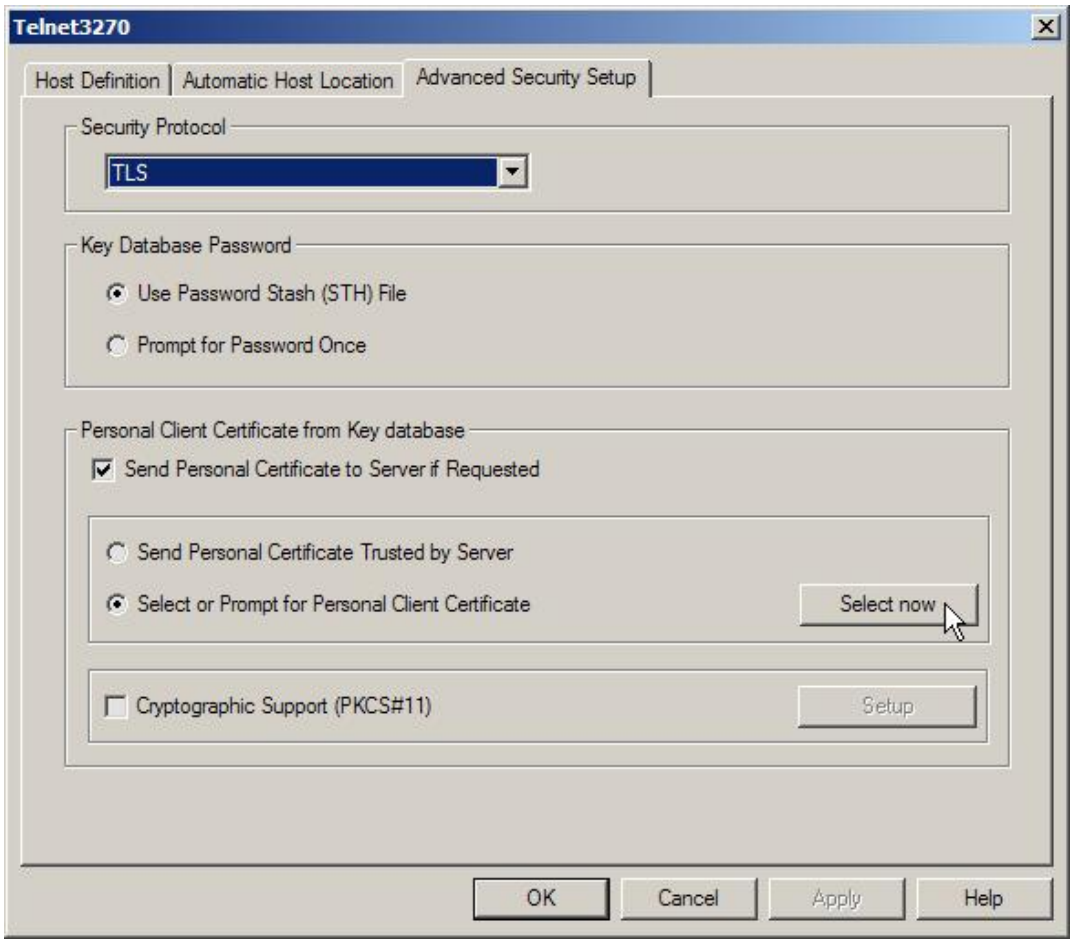
You may verify that the TLSD is now started for client authentication:

```
101 q tlsds
F7 0097 IPN253I << TCP/IP TLS Daemons >>
F7 0097 IPN617I  ID: TLSDTELNET Cipher: 2F350A096208
F7 0097 IPN618I    Port: 992  Passport: 23 Type: Server_Auth
F7 0097 IPN619I    Driver: SSLD  Minimum version: 0300
```

Don’t be confused: type **Server_Auth** is displayed for client authentication, while just **Server** would be displayed for server authentication.

5.3.3 Setting up the PCOM session for client authentication

Finally, we have to enable client authentication in the PCOM session. On the Advanced Security Setup page select Send Personal Certificate to Server if Requested and click on Select or Prompt for Personal Client Certificate. Then click on button Select now.



On the next dialog box enter the PCOM key database password and select the VSE client certificate. You may have to enter the password, leave the box with OK, and enter the box again to see the certificates in the list box. The dialog remembers the password for further use.

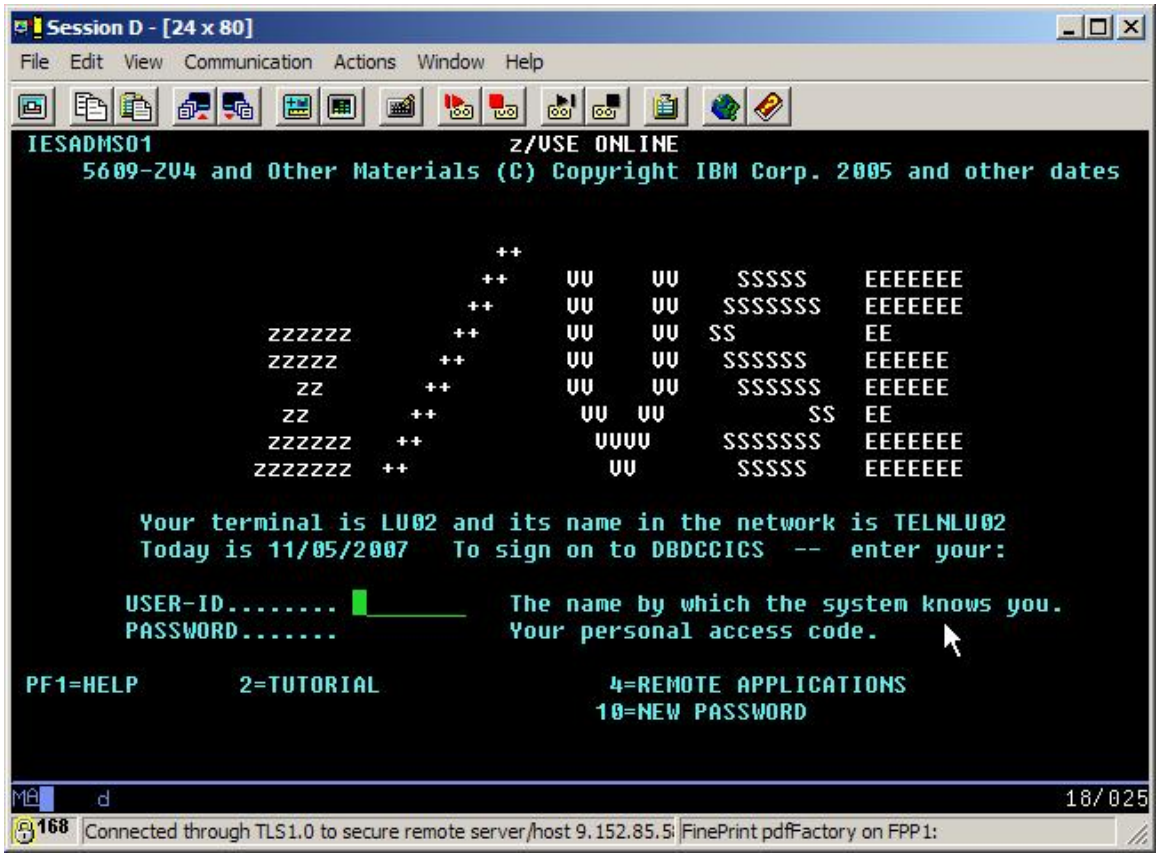
Then press **OK**.

Note: if your certificates don't show up in the list box, they are either

- not correctly imported into the PCOM key database with their private keys, or
- you entered a wrong password.

5.3.4 Connect using client authentication

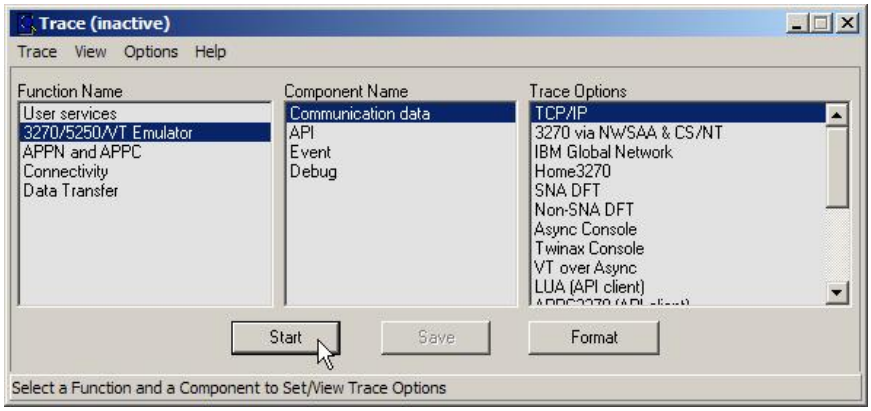
After leaving the session configuration boxes with OK, we are now ready for connect to the VSE TLSD with client authentication.



5.4 Taking a PCOM trace

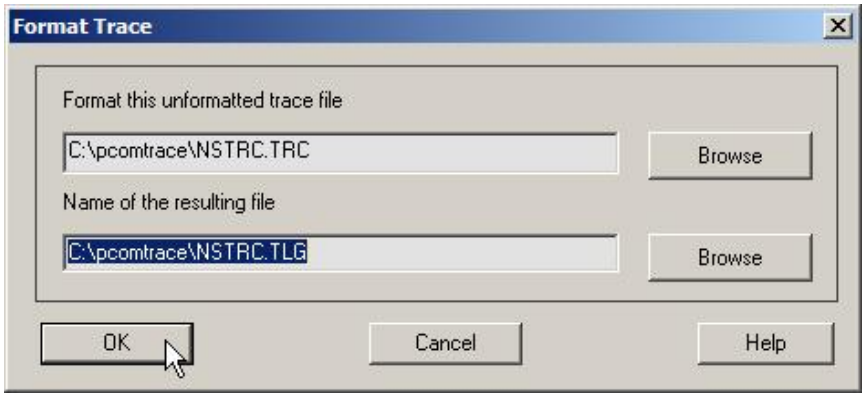
If you get any problems connecting to VSE, you might want to take a trace. PCOM provides a trace function that is activated via the session window. Click on **Actions – Launch – Trace Facility**.

On the trace box, specify a TCP/IP trace.

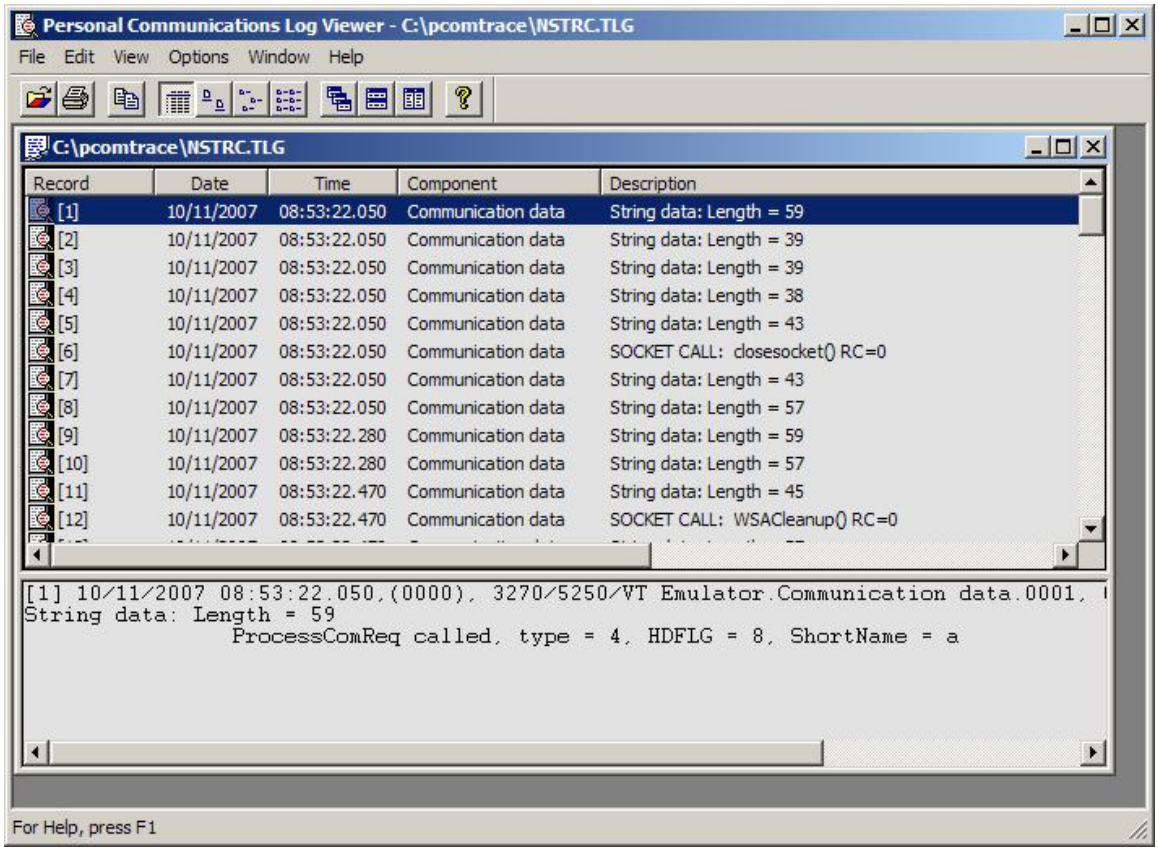


Click on **Start** and try to connect again.

Then stop and format the trace. I would recommend to place the trace output into an “easy to find” folder.



You can now use the PCOM trace viewer to view the formatted trace.



6 Client setup with Attachmate Extra! X-treme

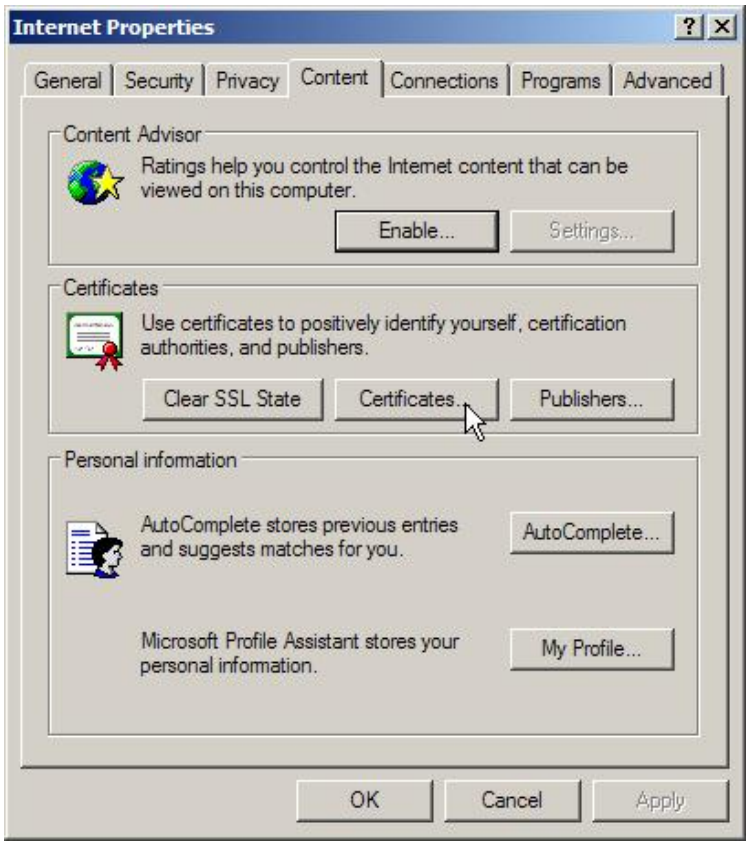
This chapter describes the secure Telnet setup with Attachmate Extra! X-treme V9 Evaluation. The evaluation version of the emulator was downloaded from

http://www.attachmate.com/en-US/Evals/Evaluate.htm

The main difference of Attachmate Extra! To IBM Personal Communications is that Attachmate uses the certificate store of Windows instead of providing an own certificate management tool. We will see later that there are also some differences in the session setup.

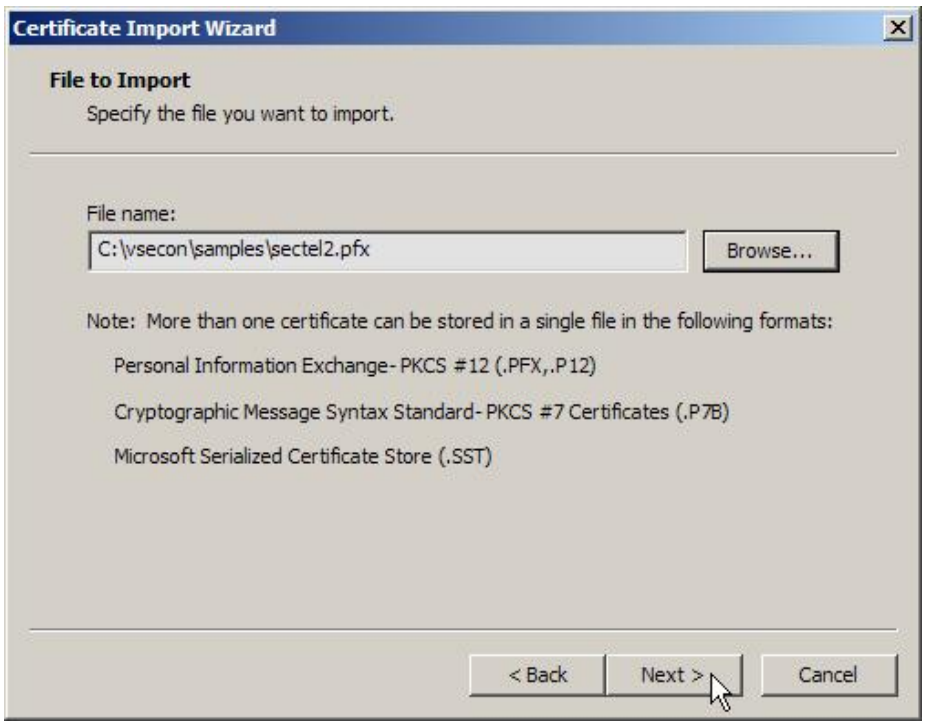
6.1 Import certificates into the Windows certificate store

To import the certificates created in section Creating a self-signed keyring on page 6, open the Windows **Control Panel** and double-click **Internet Options**. On tab **Content** click on **Certificates**.

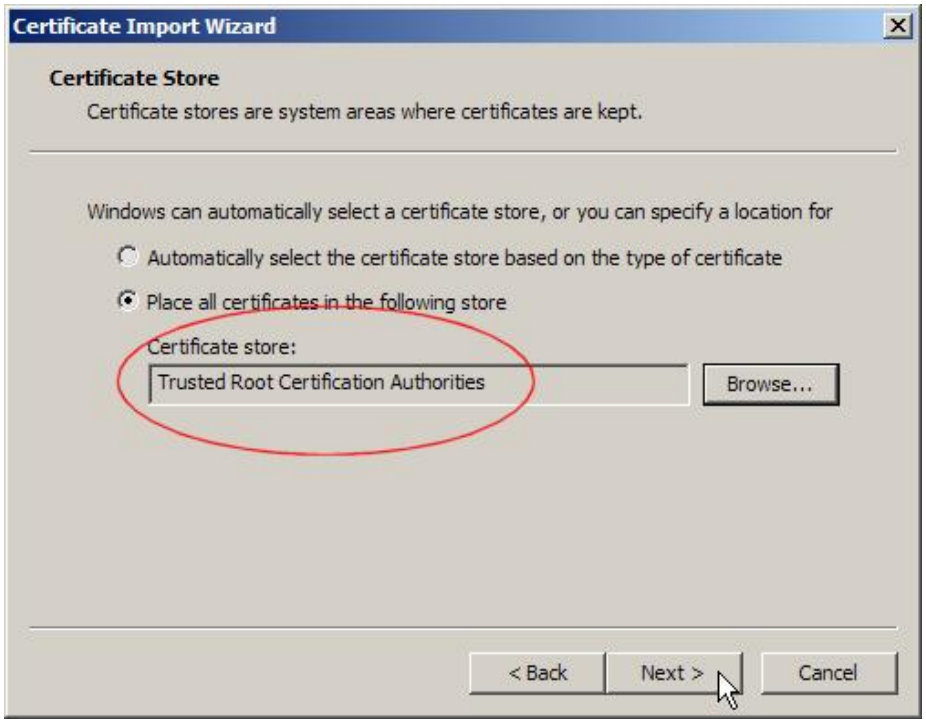


In the next box click on **Import**… and follow the Import Wizard dialogs.

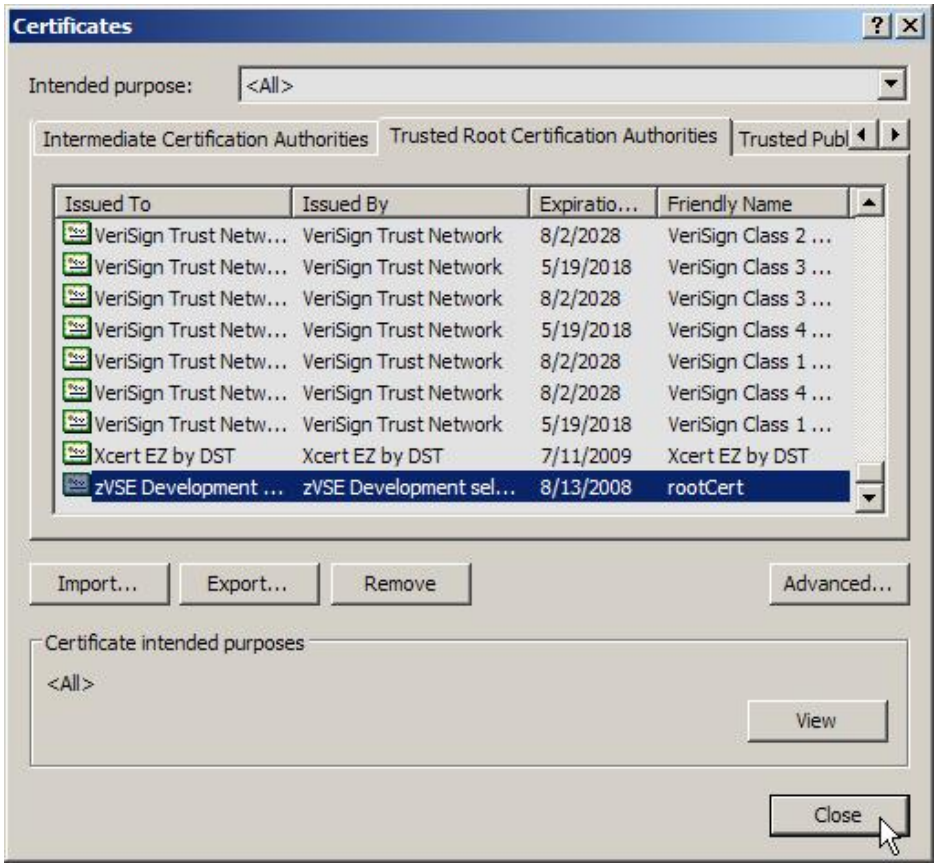
In the below picture, file **sectel2.pfx** is a new keyring file with a server certificate where the Common Name is identical to the VSE IP address, while the original file we created in section Creating a self-signed keyring on page 6 does not fulfill this condition.



In the following box select the lower radio button to place the VSE root certificate into the **Trusted Root Certification Authorities** store. If you would select the upper radio button, Windows would place the certificates into the **Personal** certificate store. This would have the same effect as not importing the root certificate at all (see problem described in section Problem with missing root certificate on page 33).



Press **Next** and finish the dialog.



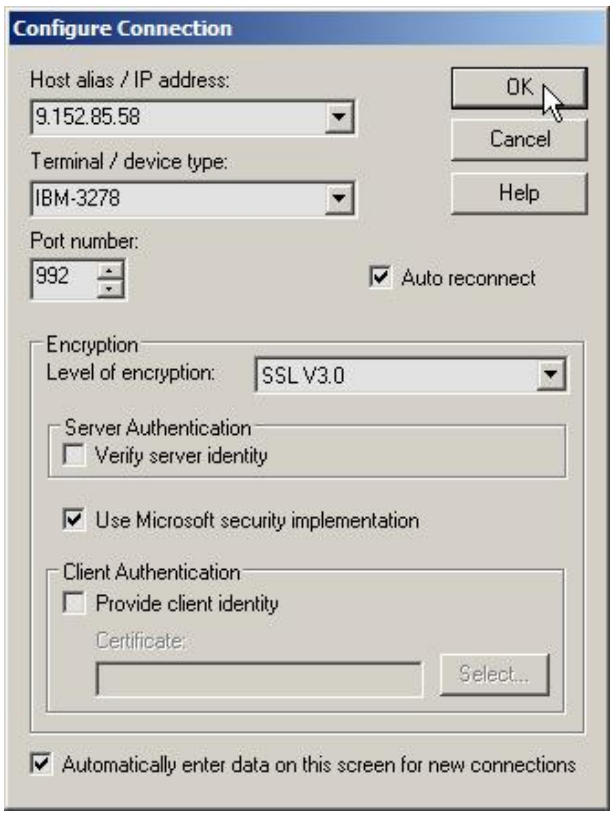
The VSE root certificate is now available in the Windows certificate store. The also contained server certificate has not been imported into this store, because it's not self-signed or signed by a known CA.

6.2 Attachmate Extra! session setup

Start a new Attachmate Extra! session and enter the IP address of your VSE system and the port number of the secure Telnet port. For Terminal / device type select IBM-3278. Checkbox **Use Microsoft security implementation** must be checked in order to get access to the Windows certificate store. Also, **Level of encryption** must be set **to SSL V3.0.**

Note: for unsecure connections, you have to

- Change the port back to your unsecure telnet port (normally 23)
- set the **Level of encryption** to **None** and
- deselect checkbox Use Microsoft security implementation.

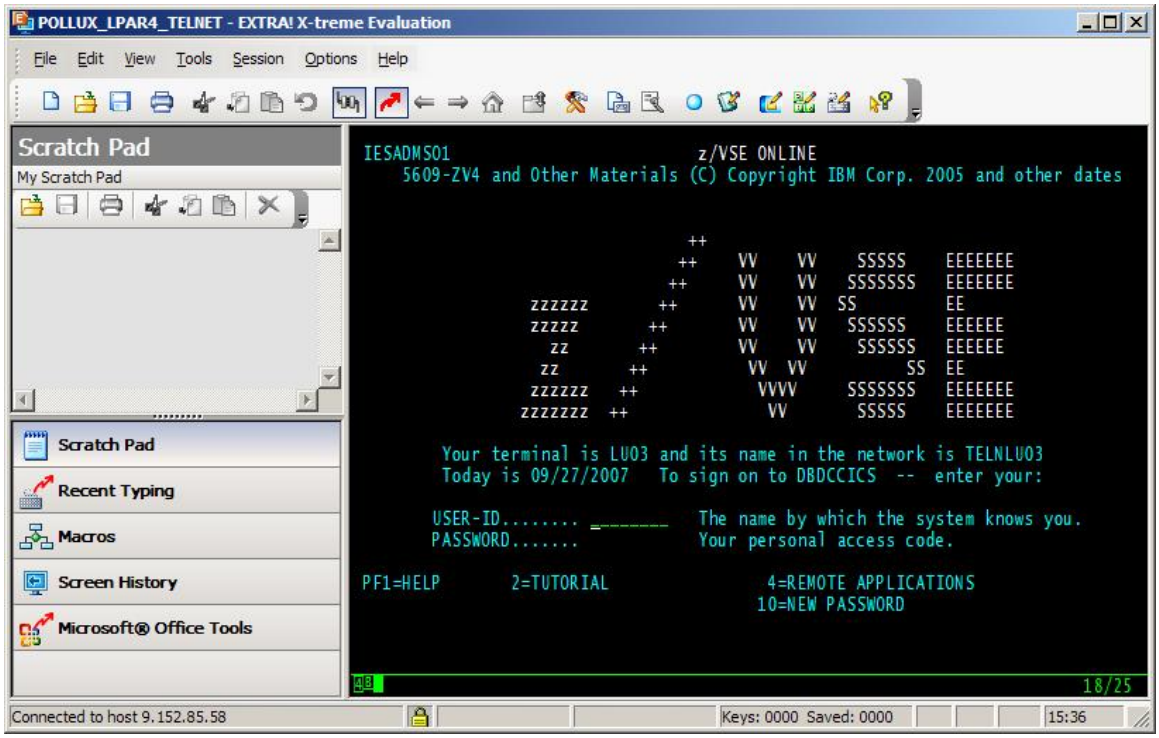


Note: You can define an additional level of security by also marking checkbox **Verify server identity**. In this case the Common Name of your VSE server certificate must be identical to the VSE IP address. This behavior is described in

http://support.attachmate.com/kb/IRE2904.html

It looks like only when **Verify server identity** is marked, the received server certificate is really checked. Otherwise the session is established in any case.

You can now connect to VSE via secure Telnet.



The closed lock icon shows that the session is now encrypted.

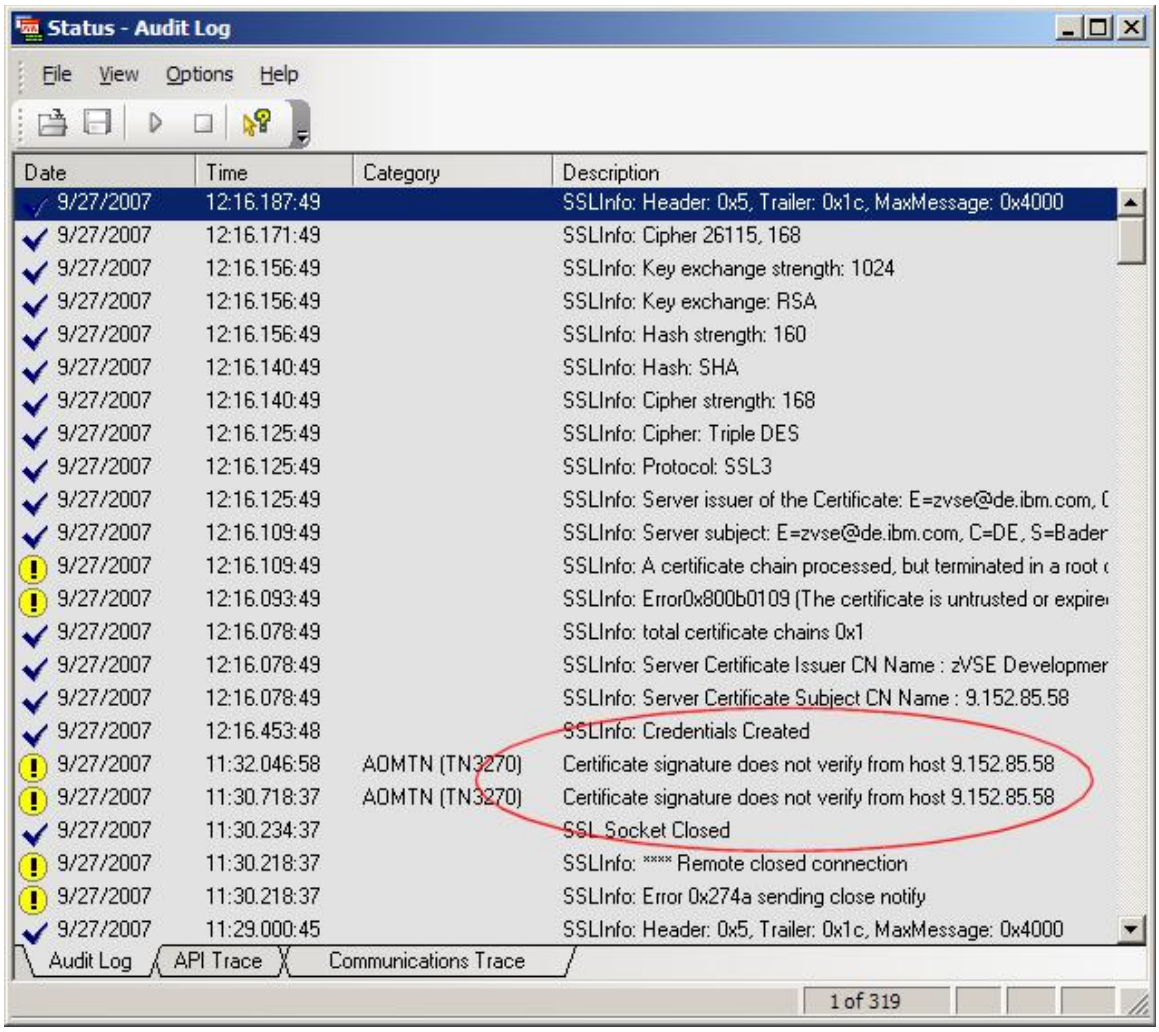
6.3 Viewing the log

To view the Attachmate event log, start application **Status App** in the Attachmate Extra! Program group. The log provides some detailed information about the SSL handshaking process and the used cipher suite for this session. For a list of supported cipher suites refer to Table 1: available cipher suites on VSE on page 14.

Following problems occurred in our test setup.

6.3.1 Problem with option Verify server identity

In the below log, message “Certificate signature does not verify from host 9.152.85.58” indicates that option **Verify server identity** was active, but the Common Name of the received server certificate was not identical to the server’s IP address. After setting up new certificates with having this precondition fulfilled, it worked.



6.3.2 Problem with missing root certificate

Another thing worth mentioning is that the connection is established even when the VSE root certificate is not contained in the Windows certificate store. Following two messages show that it was not possible to verify the received server certificate. However, the connection was established anyway.

```
SSLInfo: A certificate chain processed, but terminated in a root certificate which is not
trusted by the trust provider.
SSLInfo: Error0x800b0109 (The certificate is untrusted or expired!) returned by
CertVerifyCertificateChainPolicy!
```

After importing the keyring file (created in section Creating a self-signed keyring on page 6) into the Windows certificate store, these messages disappeared.

6.4 Setting up for client authentication

To establish client authentication with Attachmate Extra! we have to do three things. We assume that the TLSD is still running in client authentication mode.

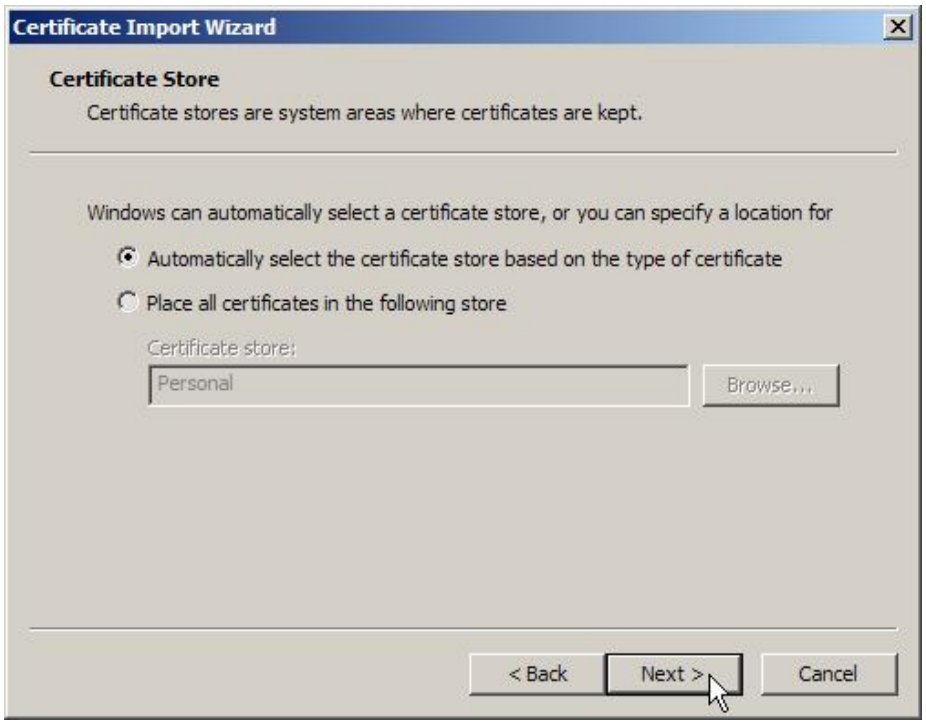
1. Change the TLSD definition on the VSE side to enable client authentication. This is done via the TYPE parameter of the TLSD definition. Refer to section Change TLSD for client authentication on page 24.

2. Import our Client Certificate into the Windows key database.

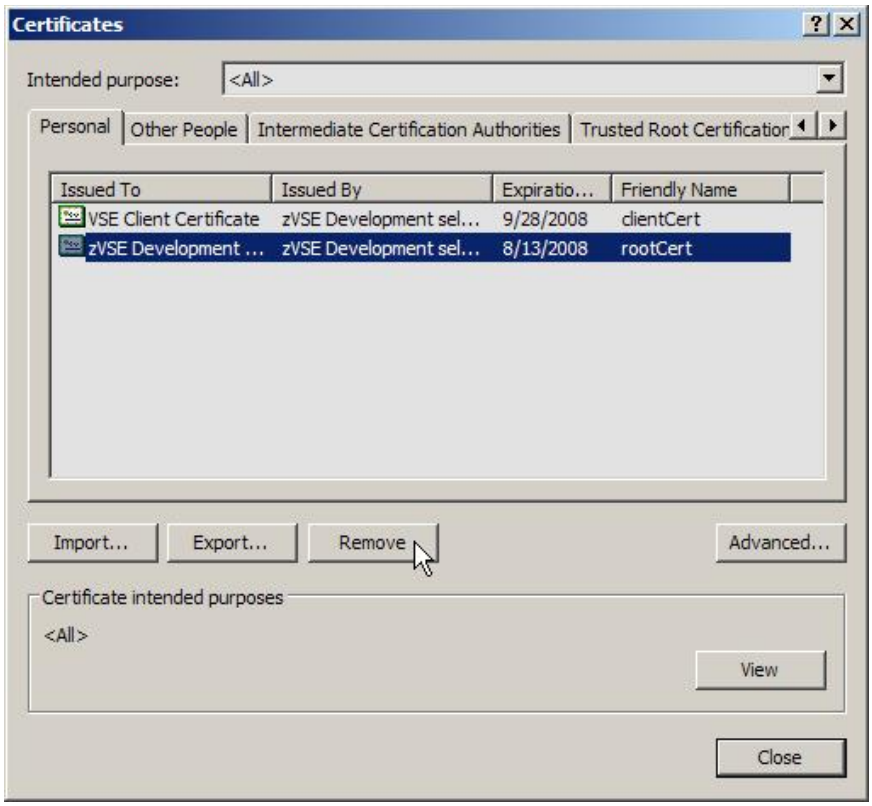
3. Change the Attachmate session definition for client authentication

6.4.1 Import the client certificate into the Windows key database

Open the Windows Control Panel and double-click on **Internet Options**. On tab **Content** click on **Certificates**. In the next box click on **Import**… and follow the Import Wizard dialogs.

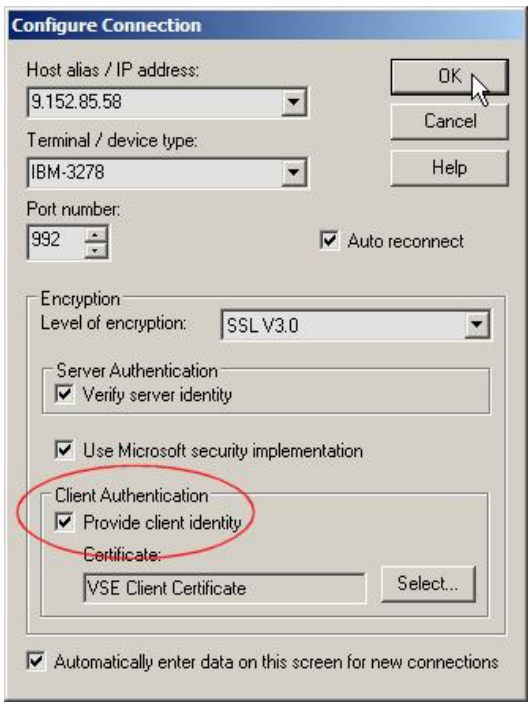


This time select the upper radio button to select the certificate store based on the type of the certificate. The certificates are now stored in section **Personal**. Because we already imported the root certificate, we now have it twice in the store. Just remove it from the **Personal** store, like shown in the below picture.



6.4.2 Change Attachmate session for client authentication

On the Attachmate session setup box select checkbox **Provide client identity** and press button **Select**.



The available certificates are displayed in the drop-down list box. Select the VSE client certificate for use by this session.

That's it. After applying your changes the session will use client authentication.

7 More information

You can find more information on the web pages below.

Personal Communications Administrator's Guide and Reference, SC31-8840
ftp://ftp.software.ibm.com/software/network/pcomm/publications/pcomm_57/pcadmin.pdf

Online admin guide for Personal Communications
http://publib.boulder.ibm.com/infocenter/pcomhelp/v5r9/index.jsp?topic=/com.ibm.pcomm.doc/books/html/admin_guide13.htm

Redbook: Personal Communications Version 4.3 for Windows 95, 98 and NT, SG24-4689
http://www.redbooks.ibm.com/abstracts/sg244689.html?Open

Attachmate® EXTRA! X-treme™ Evaluator's Guide
http://support.attachmate.com/manuals/extra/9.0/060051.1006_Xtreme_evalgd_LR.pdf

TCP/IP for VSE Commands Reference
http://www.csi-international.com/download.htm

Download Keyman/VSE from the VSE Internet homepage
http://www.ibm.com/servers/eserver/zseries/zvse/downloads/

How to setup Secure FTP with VSE, Technical Article, downloadable as PDF from
http://www.ibm.com/servers/eserver/zseries/zvse/documentation/documents.html

How to setup cryptographic hardware for VSE, Technical Article, downloadable as PDF from
http://www.ibm.com/servers/eserver/zseries/zvse/documentation/documents.html