PeSIT file transfer with IBM Sterling Secure Proxy
Getting started

Date	Version	Description	Author	Reviewer
10th August 2011	0.9	Initial release	B. Fayn	G. Cazenave
14th November 2011	1.0	Drawing added + minor corrections	B. Fayn	G. Cazenave
21st November 2001	1.1	Minor corrections	B. Fayn	M. Verzeroli
6th February 2012	1.2	Added notes about PEM encryption formats	B. Fayn	

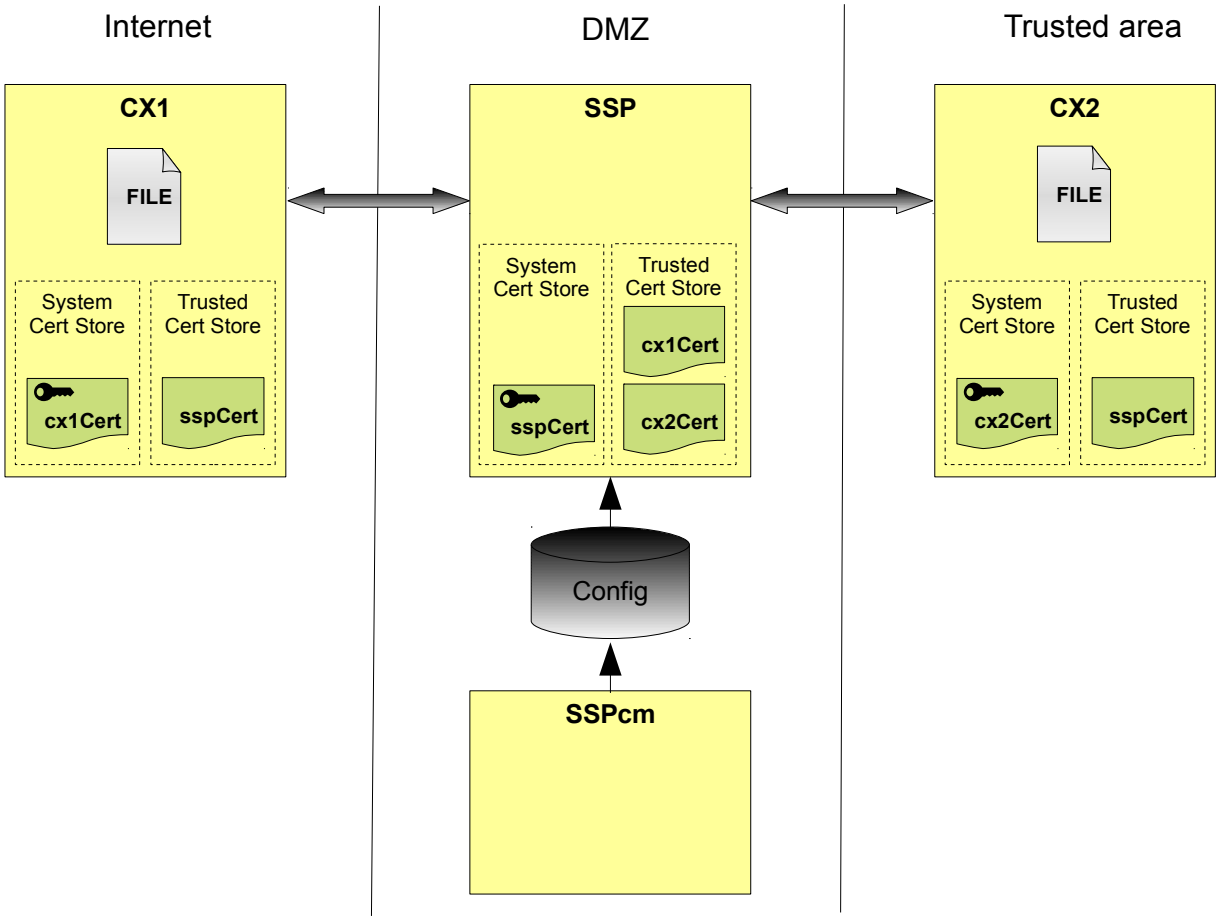
Table of content

Presentation

The purpose of this document is to take people on a fast path to successfully achieve PeSIT file transfer between two instances of IBM Sterling Connect:Express (C:X) isolated to each other by an instance of IBM Sterling Secure Proxy (SSP). This tutorial will cover how to generate SSL certificates, how to install these certificates and how to configure SSL on the C:X instances and SSP.

Architecture

Here is a picture of the architecture used in this tutorial.



This is a very simple architecture with 2 instances of Connect:Express (CX1 and CX2) and 1 instance of Sterling Secure Proxy (SSP) and its configuration manager (named hereafter SSPcm or CM). A production environment will probably use a remote perimeter server, an external authentication server, firewalls, load balancer, and so on. Also, for the secured communication between SSP and CM we will keep using the factory certificates which is not recommended for a production platform.

Requirements

Before going through this tutorial some pieces of hardware and software are required.

Hardware

The following hosts are required.

- A UNIX host named **cx1**. The UNIX flavor used in this tutorial is Ubuntu 10.04.
- A Microsoft Windows host named **cx2**. The Windows version used in this this tutorial is Microsoft Windows XP SP3.
- A UNIX or Microsoft Windows host named **ssp**. For this tutorial we used Microsoft Windows XP SP3.

Software

The following products are required.

- IBM Connect:Express V1.5.00 for UNIX. C:X must be installed on host cx1.
- IBM Connect:Express V3.1.00 for Windows. C:X must be installed on host cx2.
- IBM Sterling Secure Proxy V3.4.0 for UNIX or Windows; SSP must be installed on host ssp.
- An implementation of OpenSSL (V0.9.8 or later) for UNIX or Microsoft Windows. Most of the UNIX flavors come with an OpenSSL package. A few number of implementations can be found on Internet for the Microsoft Windows platform. OpenSSL for Windows is one of them (see the References section).
- The network must be properly configured on each host; ssp must know cx1 and cx2; cx1 and cx2 must know ssp. The traffic must be allowed between ssp and cx1, ssp and cx2, cx1 and ssp, cx2 and ssp. Use the `ping` command to ensure this is the case.

This tutorial will not cover the installation of the above products. Refer to their relevant documentation to know how to install these products (see the References section).

SSL certificates

When a client connects to a SSL server, the server sends back to the client its certificate. The client then checks that the received certificate is trusted with a certificate issued by a Certification Authority (CA). If the certificate is trusted, then the connection can be established and the data can be encrypted using the server public key. Using SSL with SSP and the PeSIT protocol in our architecture requires 3 certificates: 1 for SSP and 1 per instance of Connect:Express

Use of self-signed certificates

For the purpose of this tutorial we will use self-signed certificates. Self-signed certificates are perfectly suitable for test or development environment but not for a production environment where it is strongly recommended to use certificates issued by a trusted Certificate Authority. The advantage of a self-signed certificate is that it is trusted by itself which means that the certificated can also be used as a CA certificate. To succeed the SSL handshake with a server, the self-signed certificate has to be found in the client certificate store.

Creation of the self-signed certificates for SSP and C:X instances

On a host where OpenSSL is installed run the following commands:

```
openssl req -x509 -days 365 -subj "/C=FR/L=Paris/O=IBM/OU=France Labs/CN=SSP Certificate" -newkey
rsa:1024 -keyout sspKey.pem -out sspCert.pem -passout pass:password

openssl pkcs12 -export -in sspCert.pem -inkey sspKey.pem -passin pass:password -certfile
sspCert.pem -name "SSP Certificate" -out sspCert.p12 -passout pass:password

openssl req -x509 -days 365 -subj "/C=FR/L=Paris/O=IBM/OU=France Labs/CN=CX1 Certificate" -newkey
rsa:1024 -keyout cx1Key.pem -out cx1Cert.pem -passout pass:password

openssl pkcs12 -export -in cx1Cert.pem -inkey cx1Key.pem -passin pass:password -certfile
cx1Cert.pem -name "CX1 Certificate" -out cx1Cert.p12 -passout pass:password

openssl req -x509 -days 365 -subj "/C=FR/L=Paris/O=IBM/OU=France Labs/CN=CX2 Certificate" -newkey
rsa:1024 -keyout cx2Key.pem -out cx2Cert.pem -passout pass:password

openssl pkcs12 -export -in cx2Cert.pem -inkey cx2Key.pem -passin pass:password -certfile
cx2Cert.pem -name "CX2 Certificate" -out cx2Cert.p12 -passout pass:password
```

The above commands have created 3 self-signed certificates: **SSP Certificate**, **CX1 Certificate** and **CX2 Certificate** which will be valid during 365 days. For each certificate there are 3 files:

- `(cx1|cx2|ssp)Cert.pem` the certificate which contain the public key;
- `(cx1|cx2|ssp)Key.pem` the encrypted private keys; the password used for the encryption is `password`;
- `(cx1|cx2|ssp)Cert.p12` the certificate and the private key encrypted in a `PKCS12` format file.

Notes:

- PEM files are text files which can be displayed using a simple text editor;
- with the `openssl req` command above, a private key is normally encrypted in Traditional PEM-Encoded format (is starts with `-----BEGIN RSA PRIVATE KEY-----`); if for some reason, the private key is encrypted in PKCS#8 format (it starts with `-----BEGIN ENCRYPTED PRIVATE KEY-----`) then use the `openssl pkcs8` command to change the encryption format.

Import the self-signed certificates in SSP

SSP is using its own certificate stores. The self-signed certificates created above can be imported in the SSP configuration using SSP configuration manager (SSPcm) graphical interface. Before importing the certificates make sure the SSPcm graphical interface is running. If not, run the following Microsoft Windows command:

```
C:\Program Files\Sterling Commerce\SSPcm\bin\startCM.bat
```

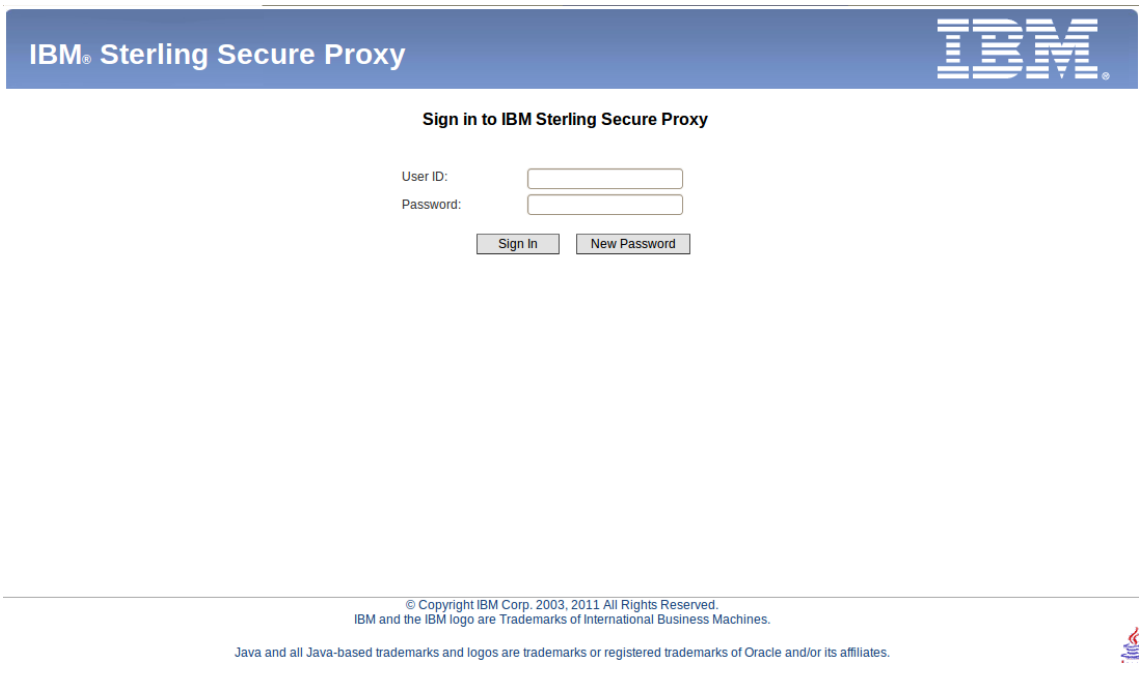
Note:

- Sterling Secure Proxy is available for different platforms (Microsoft Windows and different UNIX flavors). In this tutorial we are using SSP for Microsoft Windows and the command have a `.bat` extension. For UNIX, the commands have the same base name but with a `.sh` extension.

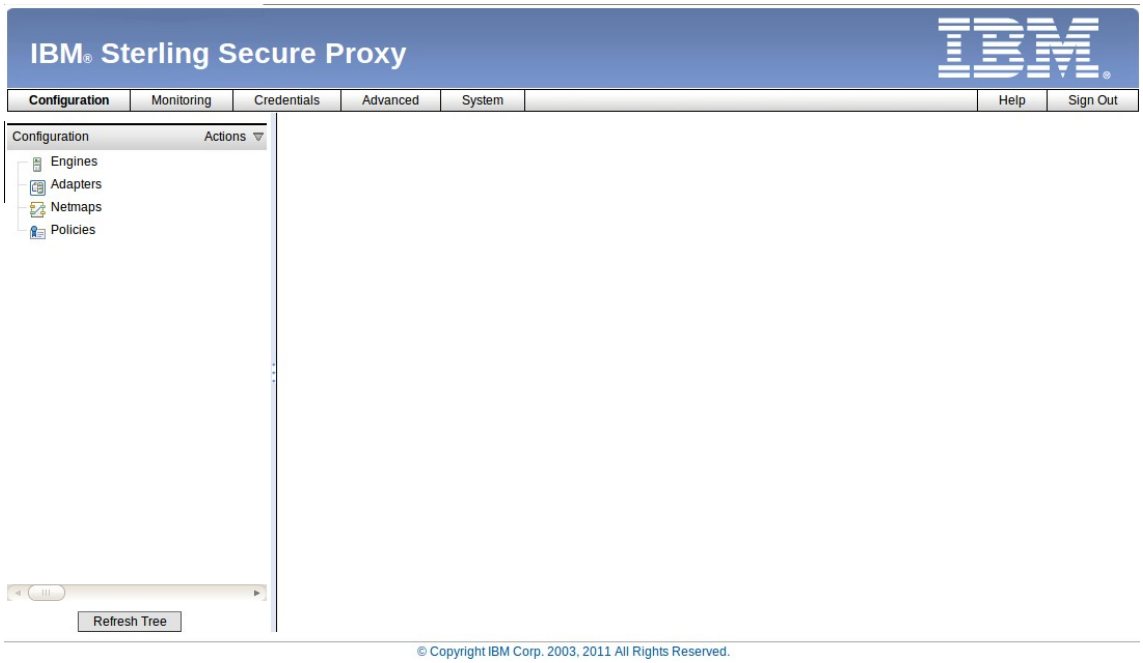
When SSPcm is ready you should see something like:

```
IBM Sterling Secure Proxy Configuration Manager Version 3.4.00, Build 126
(C) Copyright IBM Corp. 2003, 2011 All Rights Reserved.
IBM and the IBM logo are Trademarks of International Business Machines.
IBM Sterling Secure Proxy Configuration Manager Starting...
IBM Sterling Secure Proxy Configuration Manager is ready for Service.

To connect the UI to the Configuration Manager Server, open a web browser to
        https://ssp:8443/SSPDashboard
```

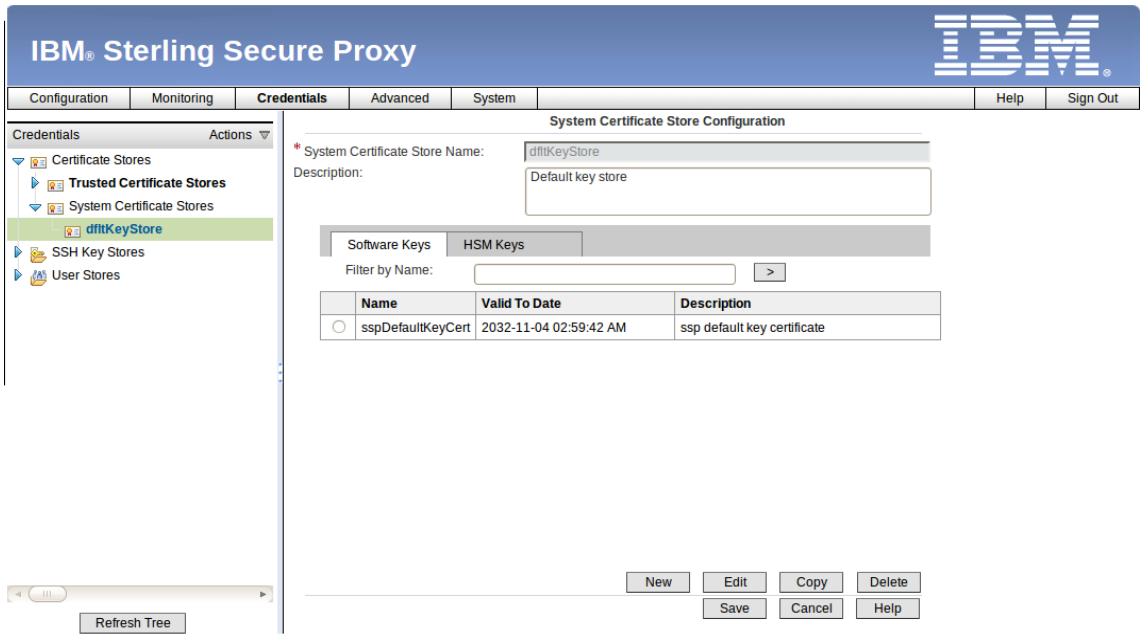
Point your favorite web browser to the displayed URL (https://ssp:8443/SSPDashboard/ in the above snapshot) . Something similar to the following must be shown.



Sign In (default user is `admin` and default password is `password`). On the displayed page select the **Credentials** tab.



From the left, expand **Certificate Stores**, then **System Certificate Stores** and select the default trusted certificate store **dfltKeyStore**.

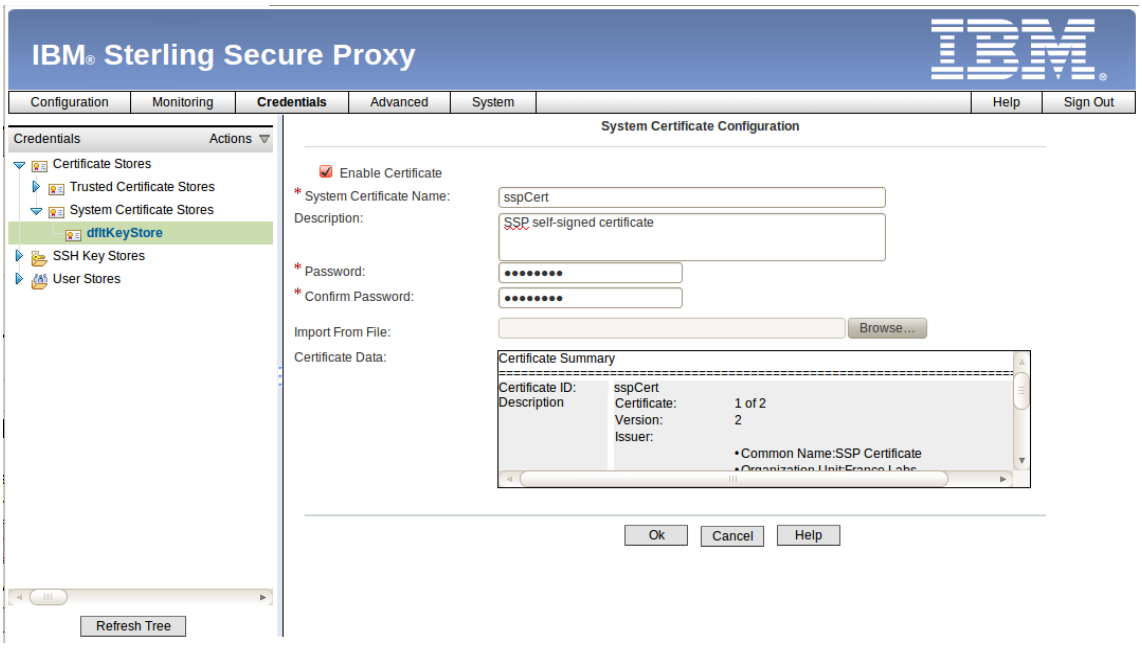


A list of already imported certificates must be displayed.

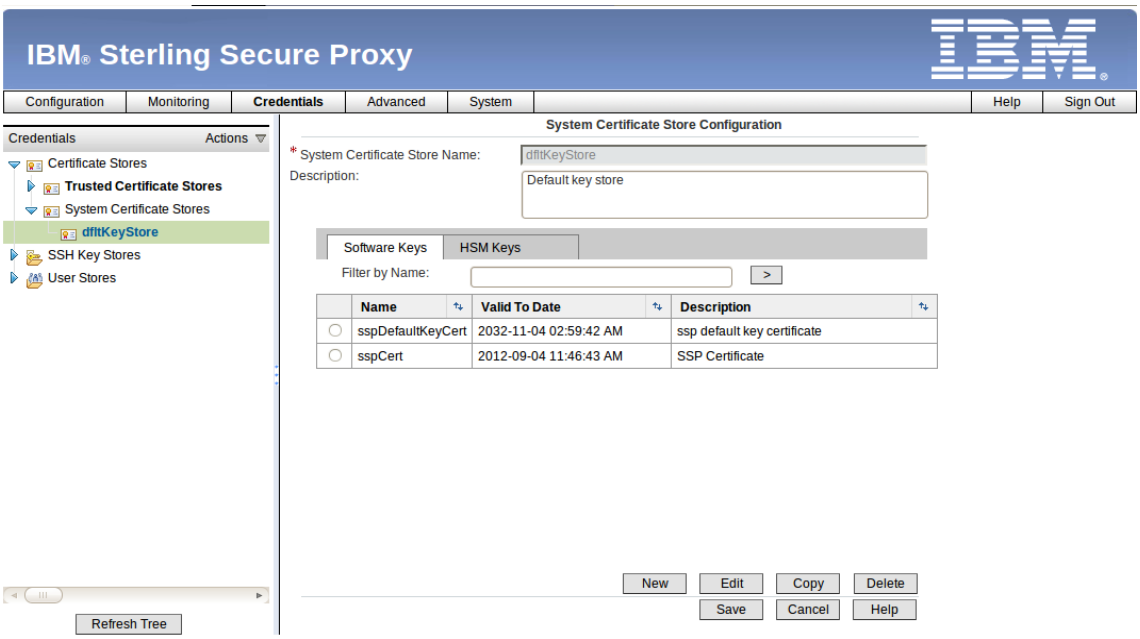
Notes:

- We could keep using the existing `sspDefaultKeyCert` certificate between SSP and the C:X partners but, for the purpose of this tutorial, and for security reason (you should not use for the SSP default certificate in production), we show how to use a new certificate.
- For clarity we keep using the default stores from installation, but it is possible to create other stores.

Click the **New** button and fill the fields **System Certificate Name**, **Description** (optional), **Password** of the encrypted private key and **Import From File**. Use the **Browse...** button to find the `ssp_cert.p12` file which has been created earlier in this paragraph (you will maybe have to copy this file to the host where you browser is running). Once the content of the file has been successfully parsed you should see something similar to the picture below.

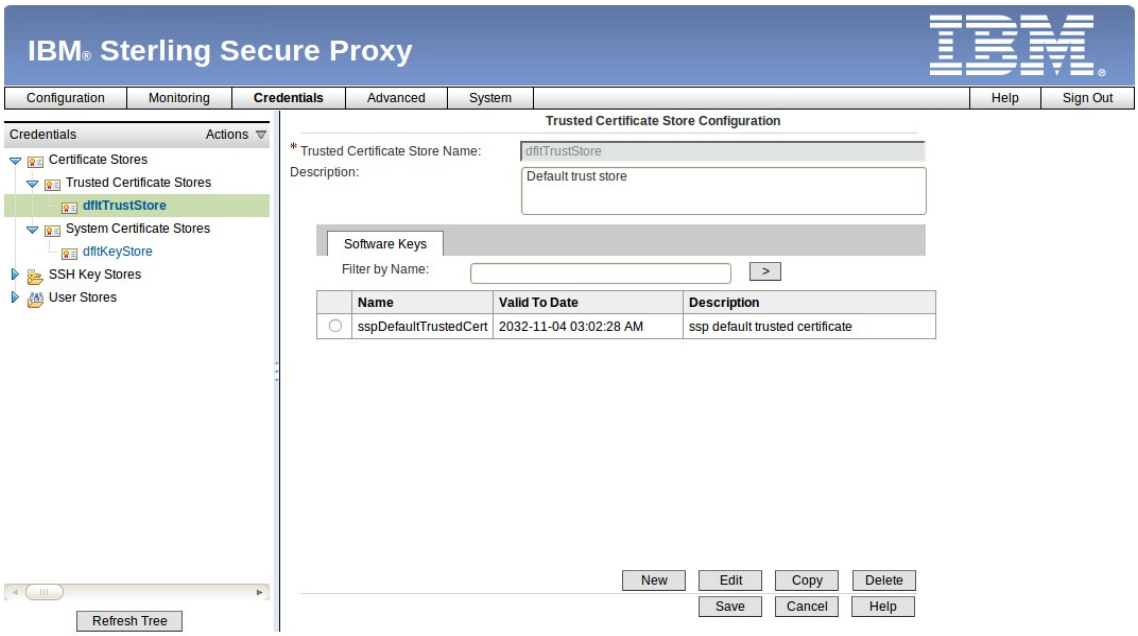


Ensure the **Enable Certificate** check box is ticked and click the **Ok** button. The imported certificate should now appear in the S**ystem Certificate Store Configuration** list.

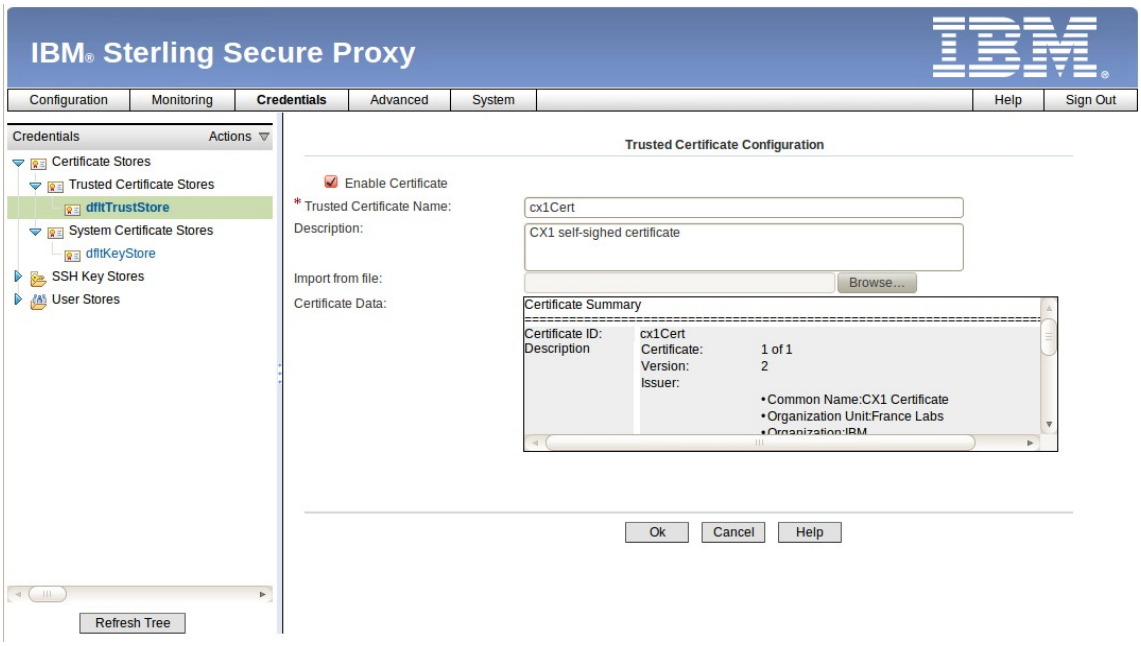


Click the **Save** button to save the configuration. The SSP certificate is now installed in the default system certificates store. It's now the turn of the C:X instances certificates to be installed, but in the default Trusted Certificates store this time.

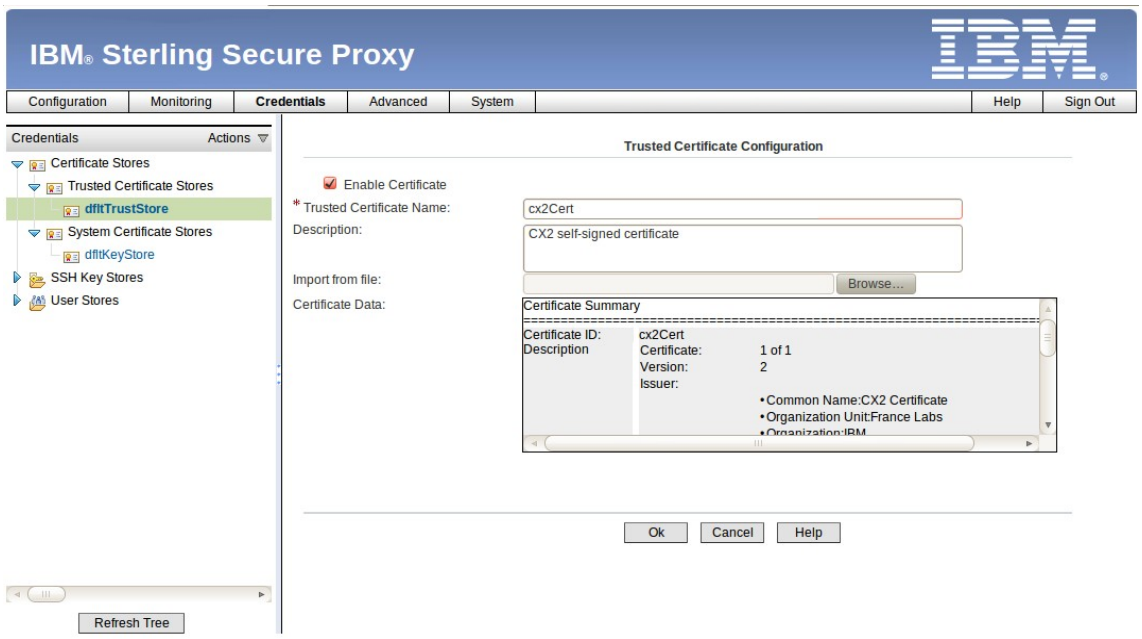
Expand the **Trusted Certificate Stores** and select the default trusted certificate store **dfltTrustStore**.



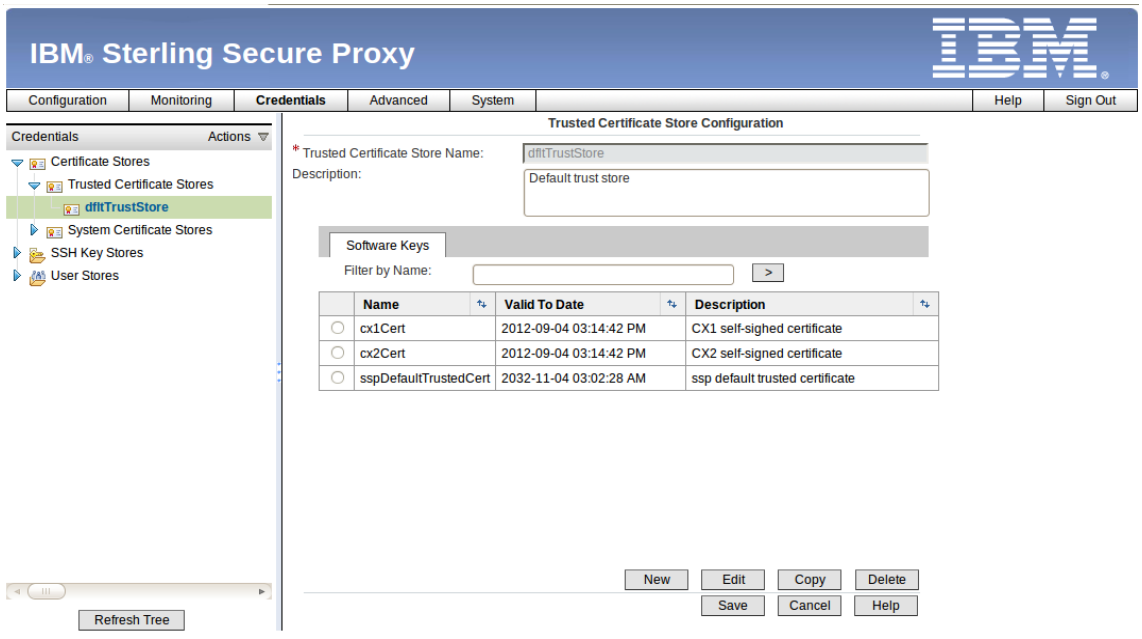
Click the **New** button and fill the fields **Trusted Certificate Name**, **Description** (optional). Use the **Browse...** button to find the `cx1Cert.pem` file which has been created earlier in this paragraph (you will maybe have to copy this file to the host where you browser is running). Once the content of the file has been successfully parsed you should see something similar to the picture below.



Click **Ok** the button. Click the **New** button again and fill the fields **Trusted Certificate Name**, **Description** (optional). Use the **Browse...** button to find the `cx2Cert.pem` file which has been created earlier in this paragraph (you will maybe have to copy this file to the host where you browser is running). Once the content of the file has been successfully parsed you should see something similar to the picture below.



Click **Ok** the button.



Click the **Save** button to save the configuration. The CX1 and CX2 certificates are now installed in the trusted certificates store.

Import the self-signed certificates in Connect:Express

The place where C:X for UNIX and C:X for Microsoft Windows look up certificates is different: C:X for UNIX has its own certificates store while C:X for Microsoft Windows uses the operating system certificates store. For that reason, the way to import certificates differs depending on C:X type.

Import the self-signed certificates in C:X for UNIX

For this tutorial, 2 certificates need to be installed in this C:X instance: the CX1 and the SSP certificate self-signed. To import these self-signed certificates the following must be done.

- Under the user who is running Connect:Express for UNIX copy the files `sspCert.pem` and `cx1Cert.pem` created previously (**Creation of the self-signed certificates for SSP and C:X instances**) in the directory `$TOM_DIR/config/ssl/cert_import/`.
- The same way copy the file cx1`Key.pem` in the directory `$TOM_DIR/config/ssl/priv/`.
- If it is not running already, start the C:X monitor (command `$start_tom`).
- Run the C:X operator interface (command `$sterm`).

- Select option **`5 SSL.`**

- Select option **`2 IMPORT CERTIFICATES`**.

```
C:X/UNIX 150-0 ---------------IMPORT CERTIFICATES------------------------- /cx1
OPTION ===> V

                          I    IMPORT
                          L    LIST
                          U    UPDATE
                          D    DELETE
                          V    VIEW


                               ID ===> ........

  X  EXIT                                                    -F3- END
```

- Select option **`I IMPORT`**, enter `SSPCERT` for the **`ID`** and press the `ENTER` key.

```
C:X/UNIX 150-0 ---------------IMPORT CERTIFICATES------------------------- /cx1
OPTION ===> I

                          I    IMPORT
                          L    LIST
                          U    UPDATE
                          D    DELETE
                          V    VIEW


                               ID ===> SSPCERT

  X  EXIT                                                    -F3- END
```

- Enter `C` for **`TYPE`**, 1 for the **`FORMAT`**, set the **FILE CONTAINING THE CERTIFICATE TO IMPORT** to `sspCert.pem`, press `ENTER` and at the **`DO YOU WANT TO GO ON ?`** prompt press the `ENTER` key.

```
C:X/UNIX 150-0 ---------------IMPORT CERTIFICATES------------------------- /cx1
OPTION ===>

ID      : SSPCERT

TYPE    : C                       (P:PERSONAL,C:CA)

CERTFICATE FILE FORMAT :   1      (1:PEM,2:DER)
FILE CONTAINING THE CERTIFICATE TO IMPORT :
sspCert.pem

KEY FILE FORMAT     :             (1:PEM,2:DER)
FILE CONTAINING THE PRIVATE KEY TO IMPORT (PERSONAL CERTIFICATE) :



PRIVATE KEY PASSWORD (PERSONAL CERTIFICATE) :

TYPE PASSWORD AGAIN


DO YOU WANT TO GO ON ?                    UPD : ............... ........
-ENTER- NEXT FIELD                  -F3- CANCEL                  -F8- COMPLETION
```

- If the certificate has been successfully imported the **`IMPORT CERTIFICATES`** menu must be displayed back.

- Select option **`I IMPORT`**, enter `CX1CERT` for the **`ID`** and press the `ENTER` key.

```
C:X/UNIX 150-0 ---------------IMPORT CERTIFICATES------------------------- /cx1
OPTION ===> I

                                I    IMPORT
                                L    LIST
                                U    UPDATE
                                D    DELETE
                                V    VIEW

                                     ID ===> CX1CERT

 X  EXIT                                                        -F3- END
```

- Set **`TYPE`** to `P`, **`FORMAT`** to 1, **`FILE CONTAINING THE CERTIFICATE TO IMPORT`** to `cx1Cert.pem` , **`KEY FILE FORMAT`** to 1, **`FILE CONTAINING THE PRIVATE KEY TO IMPORT`** to `cx1Key.pem`, the **`PRIVATE KEY PASSWORD`** accordingly, press `ENTER` and at the **`DO YOU WANT TO GO ON?`** prompt press the `ENTER` key.

```
C:X/UNIX 150-0 ---------------IMPORT CERTIFICATES------------------------- /cx1
OPTION ===>

ID      : CX1CERT

TYPE    : P                     (P:PERSONAL,C:CA)

CERTFICATE FILE FORMAT :   1    (1:PEM,2:DER)
FILE CONTAINING THE CERTIFICATE TO IMPORT :
cx1Cert.pem.......................................................................

KEY FILE FORMAT    : 1          (1:PEM,2:DER)
FILE CONTAINING THE PRIVATE KEY TO IMPORT (PERSONAL CERTIFICATE) :
cx1Key.pem........................................................................


PRIVATE KEY PASSWORD (PERSONAL CERTIFICATE) :
********.........................................................................
TYPE PASSWORD AGAIN
********.........................................................................

DO YOU WANT TO GO ON ?                  UPD : ............... ........
-ENTER- NEXT FIELD                -F3- CANCEL                  -F8- COMPLETION
```

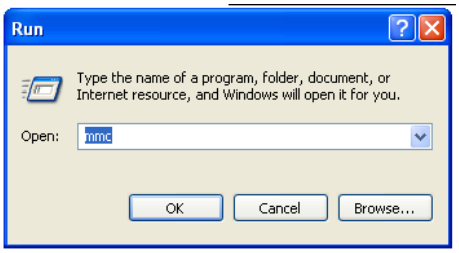
- If the certificate has been successfully imported the **`IMPORT CERTIFICATES`** menu must be displayed back. Press **`L LIST`** to display all the imported certificates.

```
C:X/UNIX 150-0 ---------------IMPORT CERTIFICATES------------------------- /cx1
OPTION ===>
    ID        TYPE NOT BEFORE           NOT AFTER            SUBJET DN

    CX1CERT   P    2011/08/31 13:45:27  2012/08/30 13:45:27  CN=cx1Cert, O=IBM,
    SSPCERT   C    2011/08/29 14:45:17  2012/08/28 14:45:17  CN=sspCert, O=IBM,

    <- -F10-  -F3- END  -F7- PREVIOUS SCREEN  -F8- NEXT SCREEN    -F11- ->
```

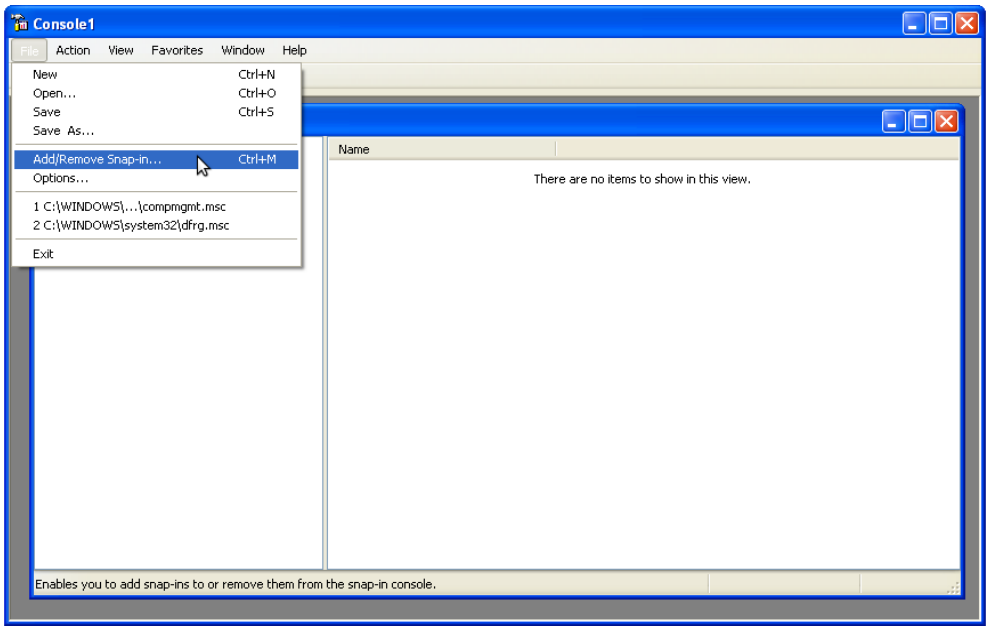
Import the self-signed certificates in C:X for Microsoft Windows

For this tutorial, 2 certificates need to be installed in this C:X instance: the CX2 and the SSP self-signed certificates. The `cx2Cert.p12` and `sspCert.pem` file which has been created previously in the paragraph **Creation of the self-signed certificates for SSP and C:X instances** must be accessible from the C:X host. To import the certificates the following must be done.

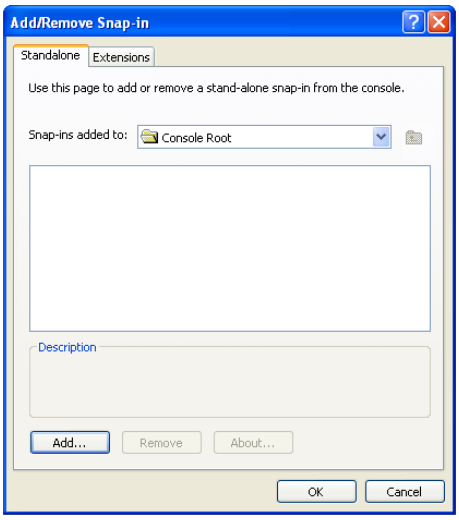
- Importing a certificate in the operating system certificates stores requires the use of the Microsoft Management Console application. Click the **Start** button, select **Run...** and enter `mmc`.



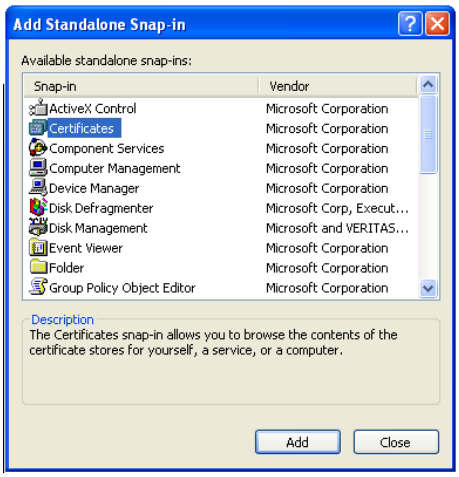
- Select **File** > **Add/Remove Snap-in...**



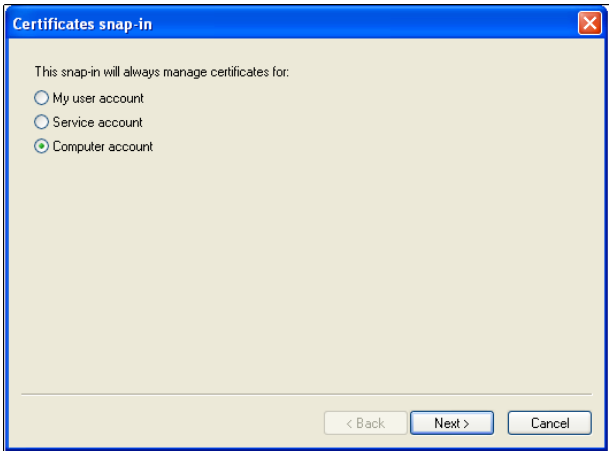
- Select **Add...** button.



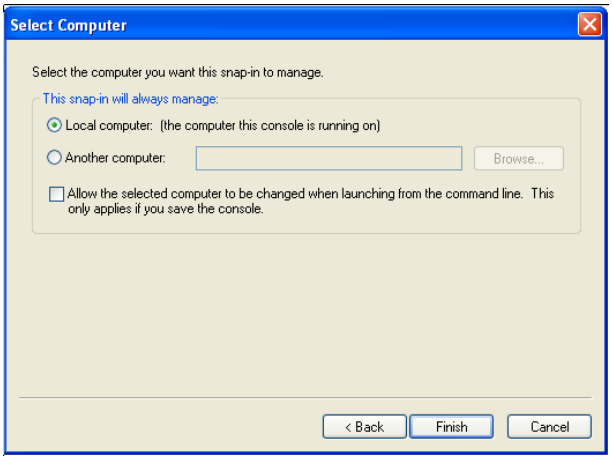
- Select **Certificates** and click the **Add** button.



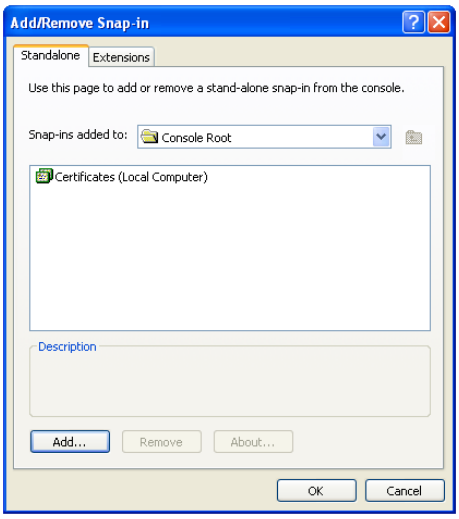
- Select **Computer account**.



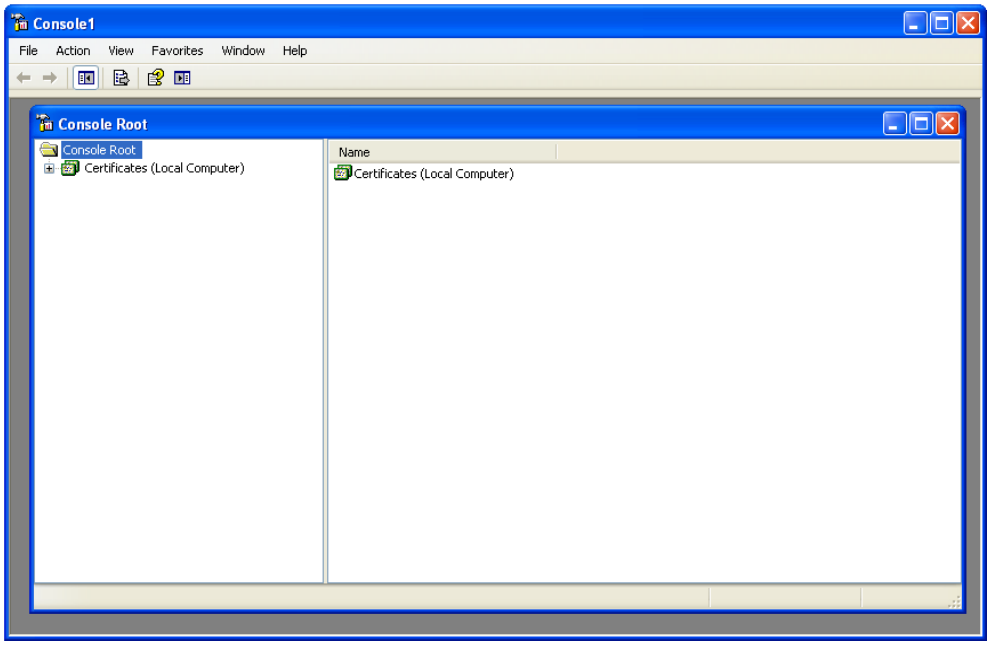
- Select **Local computer** and click the **Finish** button.



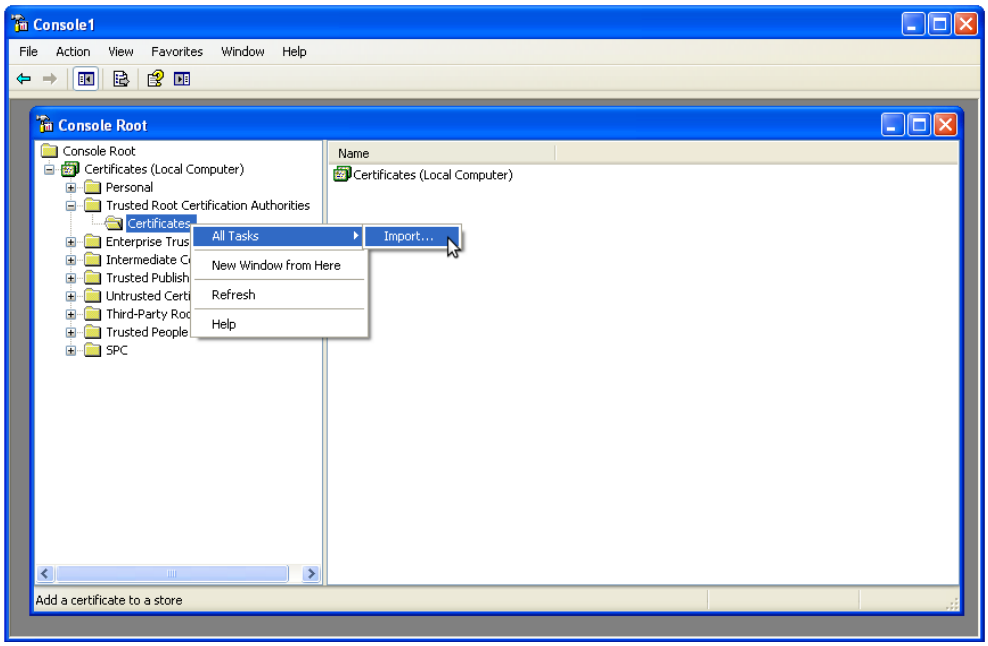
- Click the **Close** button.



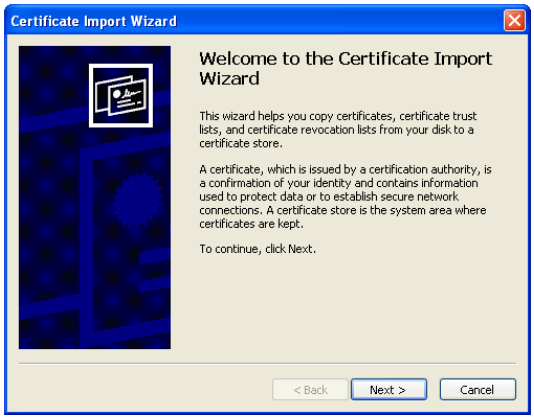
- Click the **OK** button.



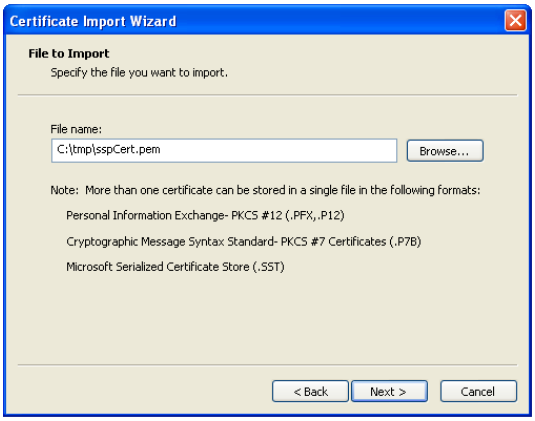
- Expand **Certificates** > **Trusted Root Certification Authorities**. Right click on **Certificates** and select **All Tasks** > **Import...**



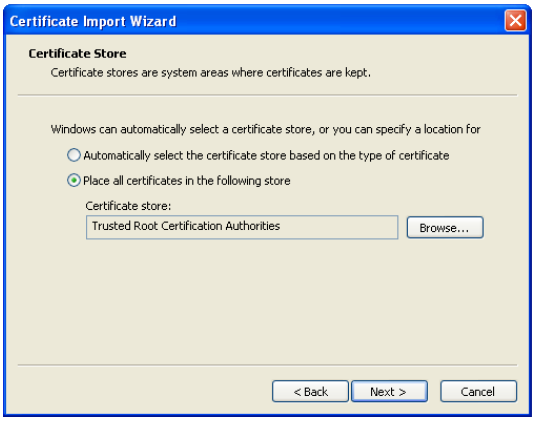
- Click the **Next >** button.



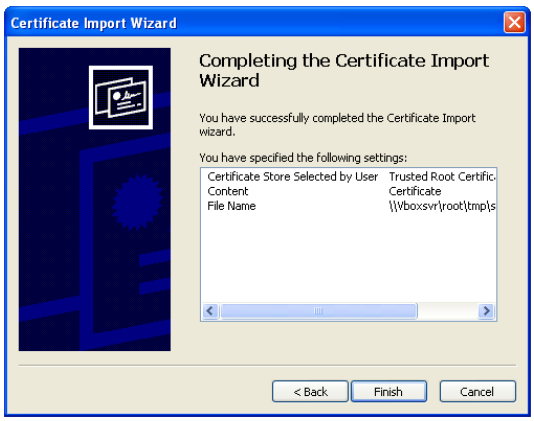
- Set the **File name** field with the full path name of the `sspCert.pem` file (use the **Browse...** button if necessary).



- Click the **Next >** button.

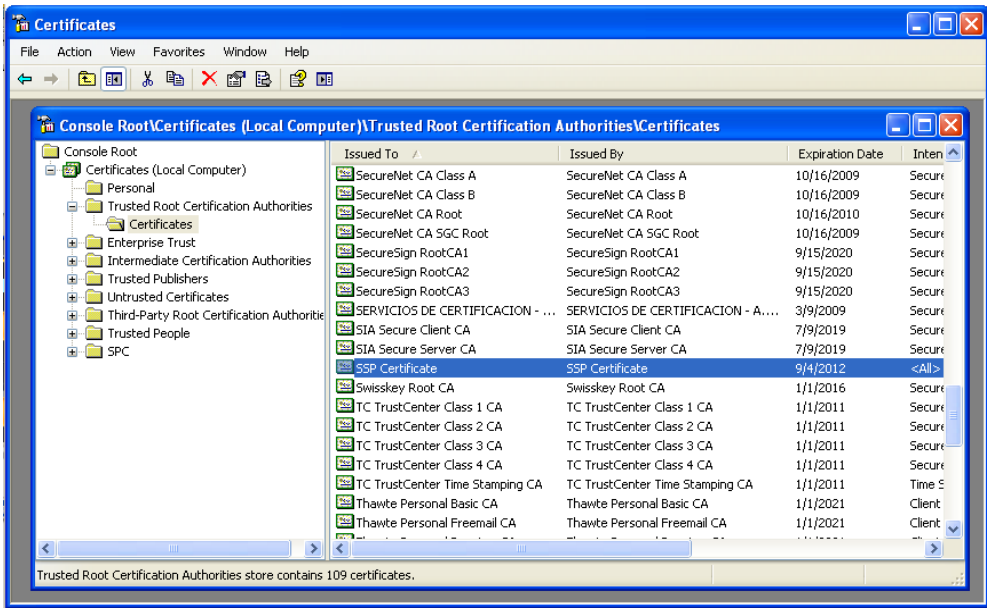


- Check that **Place all certificates in the following store** is selected and points to the **Trusted Root Certification Authorities**. Click the **Next >** button.

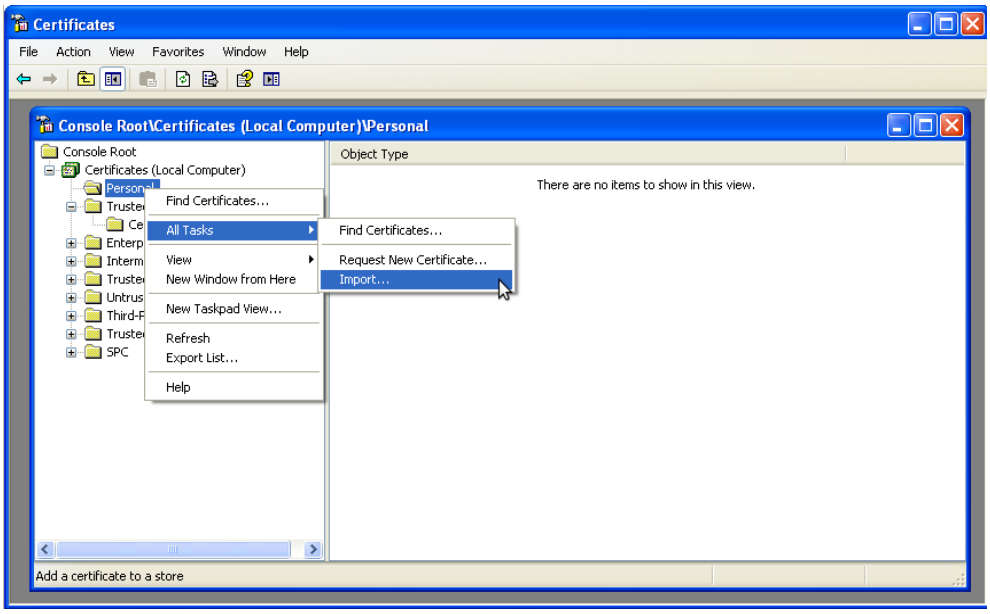


- Click the **Finish** button. A dialog box should report the success of the import.

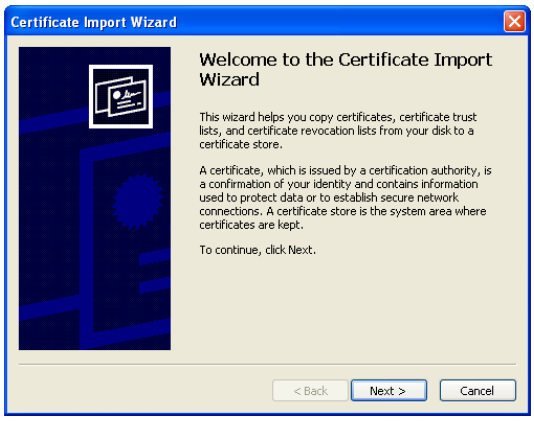
- Click the **OK** button. The **SSP Certificate** should appear in the **Certificates** list of the **Trusted Root Certification Authorities**.



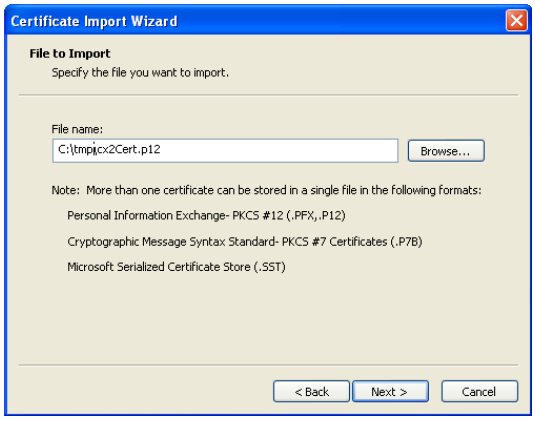
- Expand **Certificates** > **Personal**; right click on **Certificates** and select **All Tasks** > **Import...** (if there are no certificates already installed then right-click **Personal**).



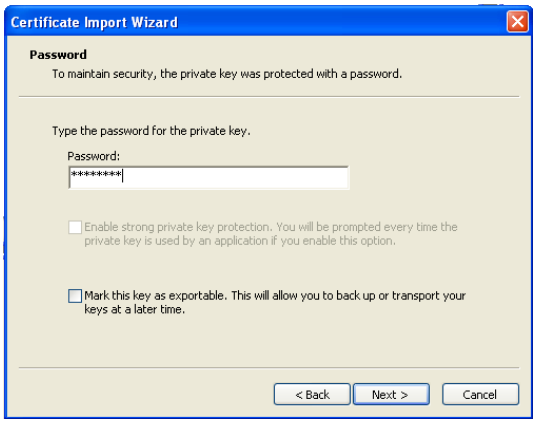
- Click the **Next >** button.



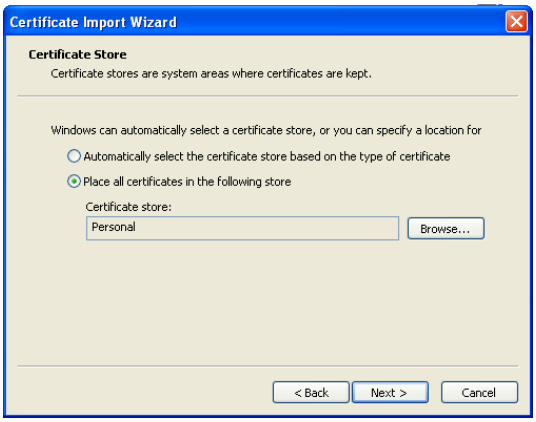
- Set the **File name** field with the full path name of the `cx2Cert.p12` file (use the **Browse...** button if necessary).



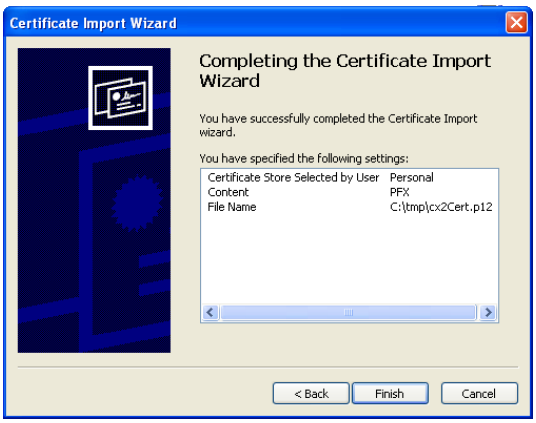
- Click the **Next >** button.



- Set the **Password** with the private key password and click the **Next >** button.

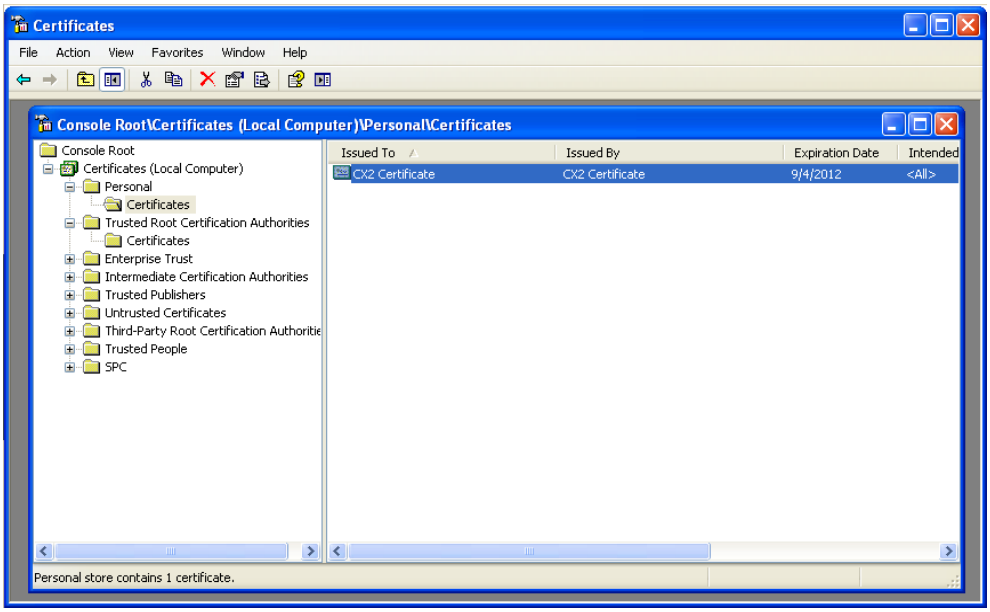


- Check that **Place all certificates in the following store** is selected and points to the **Personal** certificate store. Click the **Next >** button.

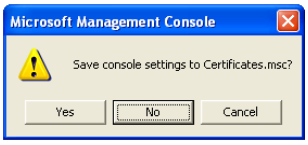


- Click the **Finish** button. A dialog box should report the success of the import.

- Click the **OK** button. The **CX2 Certificate** should appear in the **Certificates** list of **Personal**.



- Close the Microsoft Management Console. Click the **No** button to not save the console settings (the certificates will remain installed).



Configuration

This section explains how to configure the C:X instances and SSP.

Configure Connect:Express for Unix

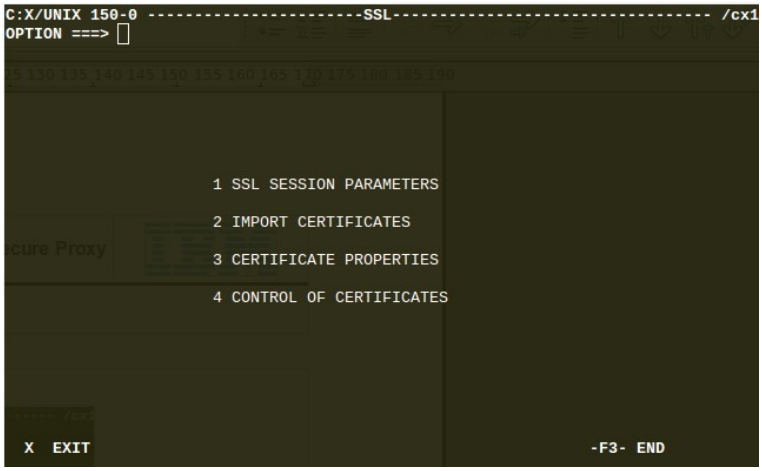
In order to do a secured file transfer to another C:X instance we need to create a few Connect:Express objects. The required certificates should have been already installed as described in the section **Import the self-signed certificates in C:X for UNIX**.

Define the client SSL session

Even if we will not use SSL client authentication in this tutorial, C:X for UNIX requires to define a client session.

- If it is not running already, start the C:X monitor (command `$start_tom`).
- Run the C:X operator interface (command `$sterm`).

- Select option **5 SSL**.



- Select option **1 SSL SESSION PARAMETERS**.

```
C:X/UNIX 150-0 ----------------------SSL---------------------------------- /cx1
OPTION ===>

                    1 SSL SESSION PARAMETERS

                    2 IMPORT CERTIFICATES

                    3 CERTIFICATE PROPERTIES

                    4 CONTROL OF CERTIFICATES

 X  EXIT                                                    -F3- END
```

- Select option **`A ADD`**. Set the **`ID`** to `CX1CLI`. Press the **ENTER** key.

```
C:X/UNIX 150-0 -------------SSL SESSION PARAMETERS----------------------- /cx1
OPTION ===> A

                    A    ADD
                    L    LIST
                    U    UPDATE
                    D    DELETE
                    V    VIEW

                    ID ===> CX1CLI.

 X  EXIT                                                    -F3- END
```

- Set the **`STATE`** to `E`, **`MODE`** to `C`, **`VERIFY OPTIONS`** to `0`, **`CERTIFICATE ID`** to `CX1CERT`, **`TLSV1`**, **`SSLV2`** and **`SSLV3`** to `Y`, leave the **`IP HEADER`** set to `N`. At the **`DO YOU WANT TO GO ON?`** prompt press the `ENTER` key.

```
C:X/UNIX 150-0 -------------SSL SESSION PARAMETERS----------------------- /cx1
OPTION ===>

ID                      : CX1CLI
STATE                   : E                 (E:ENABLED,H:DISABLED)
MODE                    : C                 (C:CLIENT,S:SERVER)
VERIFY OPTIONS          : 0                 (0:NONE,1:PEER
                                            2:PEER + FAIL_IF_NO_PEER_CERT)
CERTIFICATE ID          : CX1CERT.
CIPHER LIST             : ................  (LIST FILE NAME)
SSL PROTOCOL VERSIONS :
TLSV1 : Y SSLV3 : Y SSLV2 : Y

CA LIST (SERVER MODE)   : ........          ('CACERT-ID',#LIST)
DH PARAMETERS (SERVER)  : ................  (FILE NAME)
LOCAL NETWORK ADDRESS (SERVER MODE)  :
NETWORK (T:TCPIP)       : .
TCP/IP : IP ADDRESS     : ...............   TCP PORT : .....
         IP HEADER      : N


DO YOU WANT TO GO ON ?                  UPD : ............... ........
-ENTER- NEXT FIELD                -F3- CANCEL                 -F8- COMPLETION
```

- Press `F3` to go back to the **`MAIN MENU`**.

Define the partner CX2

We need to define the partner to which we will send the file.

- From the **MAIN MENU** select option **1 DIRECTORIES PARTNERS FILES**.
- 

- Select option **1 P-PARTNERS.**
- Select option **A ADD** and set **PARTNER** to CX2SSL. Press the ENTER key.

- Set **PASSWORD** to CX2SSL, **INITIALIZATION STATUS** to E, **PARTNER TYPE** to O, **PROTOCOL NUMBER** to 3, **SESSION TABLE NUMBER** to 1, **MAX NO. CONNECTIONS** to 05/05/10, **TYPE OF CONNECTIONS** to T, **TCPIP HOST** to ssp, **TCPIP PORT** to 16100, **DPCSID ALIAS** to CX1SSL, **SSL PARMID** to CX1CLI, **DPCPSW ALIAS** to CX1SSL, **NUMBER OF RETRIES** to 02, **INTERV.SESS** to 01. At the **DO YOU WANT TO GO ON?** prompt press the ENTER key.

```
C:X/UNIX 150-0 --------------- PARTNERS DIRECTORY ------------------------ /cx1
OPTION ===>

SYMBOLIC NAME ......... : CX2SSL
PASSWORD .............. : CX2SSL          PASSWORD OF PARTNER
INITIALIZATION STATUS . : E               E:ENABLE     H:DISABLE
PARTNER TYPE .......... : O               T/O
PROTOCOL NUMBER ....... : 3               1:ETEBAC 3, 2:FTP, 3:PESIT
SESSION TABLE NUMBER .. : 1               1->9 SESSION TABLES
X25 PORT .............. : .               X25 DEVICE NAME
MAX. NO. CONNECTIONS .. : 05/05/10        01->64  TOT/IN/OUT
TYPE OF CONNECTION .... : T               X, P, T OR M
X25 DIAL NUMBER ....... : ............... 1-15 CHARACTERS
LOCAL DIAL NUMBER ..... : ............... 1-15 CHARACTERS
EXTRA NETWORK FIELD ... : ........        'USER-DATA-FIELD'
FACILITIES ............ : ................
TCPIP HOST ............ : ssp                                PORT . : 16100
TCPIP ADDRESS ......... : ............... DEF FTP FILE .. : ........
DPCSID ALIAS .......... : CX1SSL          SSLPARM ID .... : CX1CLI
DPCPSW ALIAS .......... : CX1SSL          CONTROL OF CERTIFICATES  : ........
NUMBER OF RETRIES ..... : 02              INTERV.SESS ,TRF  : 01, 01 MINUTES

DO YOU WANT TO GO ON ?                    UPD : ............... ........
-ENTER- NEXT FIELD                -F3- CANCEL                  -F8- COMPLETION
```

- Press `F3` to go back to the **`MAIN MENU`**.

Define the file definition

The last object to define is the characteristics a the file to transfer.

- From the **`MAIN MENU`** select option **`1 DIRECTORIES PARTNERS FILES`**.

```
C:X/UNIX 150-0 ------------ DIRECTORIES MANAGEMENT ---------------------- /cx1
OPTION ===>

        1 P-PARTNERS      PARTNERS DIRECTORY MANAGEMENT

        2 F-FILES         FILES DIRECTORY MANAGEMENT

  X  EXIT                                                      -F3- END
```

- Select option **`2 F-FILES`**.
- Select option **`A ADD`** and set **`FILE`** to `TEST`. Press the `ENTER` key.

```
C:X/UNIX 150-0 ----------- FILES DIRECTORY MANAGEMENT -------------------- /cx1
OPTION ===> A

            A    ADD
            L    LIST
            U    UPDATE
            D    DELETE
            V    VIEW


            FILE       ===> TEST...

 X  EXIT                                                    -F3- END
```

- Set **`INITIALIZATION STATUS`** to `E`, **`DIRECTION`** to `*`, **`RECEIVING PARTNER`** to `$$ALL$$`, **`TRANSMITTING PARTNER`** to `$$ALL$$`, **`PRIORIY`** to `2`, **`DEFINITION TYPE`** to `D`, **`PRESENTATION TABLE`** to `1`, **`PARAMETER CARD FILE`** to `N`, **`SPACE TO RESERVER`** to `N`, **`ALLOCATION RULE`** to `0`, **`PHYSICAL NAME`** to `/tmp/TEST_&REQNUMB.txt`, **`RECORD FORMAT`** to `B*`, **`DEFINITION TYPE`** to `D`, **`RECORD LENTGH`** to `512`. Leave the parameters to the default by pressing the `ENTER` key.

```
C:X/UNIX 150-0 --------------- FILES DIRECTORY -------------------------- /cx1
OPTION ===>
SYMBOLIC NAME ......... : TEST

INITIALIZATION STATUS . : E                E:ENABLE     H:DISABLE
DIRECTION ............ : *                T:TRANSMIT R:RECEIVE *:EITHER
RECEIVING PARTNER ..... : $$ALL$$          'NAME',#LISTE, $$ALL$$
TRANSMITTING PARTNER .. : $$ALL$$          'NAME',#LISTE, $$ALL$$
PRIORITY .............. : 2                0:URGENT 1:FAST 2:NORMAL
DEFINITION TYPE ....... : D                D:DYNAMIC F:FIXED
PRESENTATION TABLE .... : 1                1 -> 9 PRESENTATION TABLE
PARAMETER CARDS FILE   : N                Y/N
SPACE TO RESERVE ...... : N                Y/N
ALLOCATION RULE ....... : 0                0:INDIF., 1:PREALL., 2:TO CREATE
PHYSICAL NAME ......... : /tmp/TEST_&REQNUMB.txt
RECORD FORMAT ......... : B*               TF, TV, BF, BU, T*, B*, **
RECORD LENGTH ......... : 00512            1-5 NUMERIC CHARAC.
REMOTE DSN (FTP) ...... :
TYPE/STRUCTURE/MODE FTP : ***              E/A/I/*,F/R/*,B/S/*
STORE UNIQUE (FTP) .... : N                Y/N  FA :    Y/N  NOT :    (0-7)

OPTION : VIEW                              UPD : 11/09/08 10:00 bfn
-ENTER- NEXT FIELD              -F3- CANCEL                  -F8- COMPLETION
```

The physical name of the file is `/tmp/TEST_&REQNUMB.txt`. `&REQNUMB` is a variable which will be replaced by the request number when the transfer request will be executed.

- On the next screen, press the `F8` key. At the **`DO YOU WANT TO GO ON?`** prompt press the `ENTER` key.

```
C:X/UNIX 150-0 --------------- FILES DIRECTORY -------------------------- /cx1
OPTION ===>

SYMBOLIC NAME ......... : TEST              DEFINITION : D   DIRECTION : T


TRANSMISSION :
START EXIT ............ :
START COMMAND ......... :
END EXIT .............. :
END COMMAND ........... :

RECEPTION :
START EXIT ............ :
START COMMAND ......... :
END EXIT .............. :
END COMMAND ........... :


OPTION : VIEW                           UPD : 11/09/08 08:00 bfn
-ENTER- NEXT FIELD                -F3- CANCEL                -F8- COMPLETION
```

At this point the configuration is finished to be able to transfer the file TEST to the CX2 partner.

Define the server SSL session

Because we want CX1 instance to be able to accept inbound connections, it must be configured as a server as well.

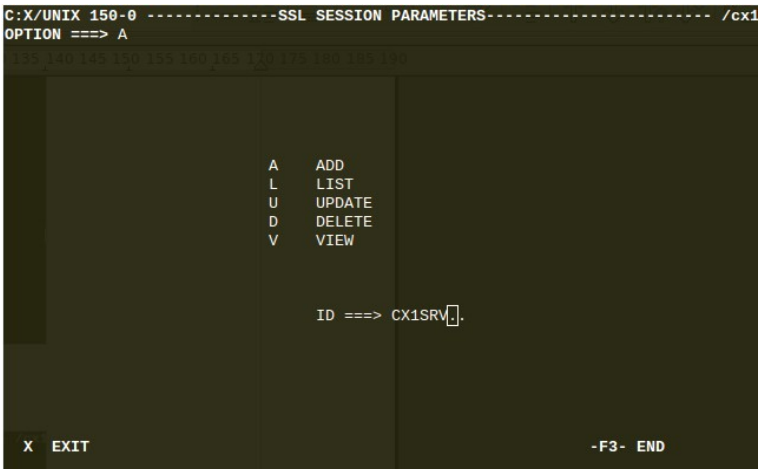
- If it is not running already, start the C:X monitor (command `$start_tom`).
- Run the C:X operator interface (command `$sterm`).

```
C:X/UNIX 150-0 -------------- MAIN MENU (GLOBAL) ------------------------ /cx1
OPTION ===>

                    IBM (R) Sterling Connect:Express (R)
     Licensed Material - Property of IBM. (C) Copyright IBM Corp. 1999, 2011


_ 1  DIRECTORIES     _ 2  MONITOR     _ 3  TABLES          _ 4  REQUEST

     PARTNERS             STATUS           SESSION         _ 5  SSL

     FILES                LOG              PRESENTATION

                          REQUEST DELETION


  X  EXIT                                                  -F3- END
```

- Select option **5 SSL**.

- Select option **1 SSL SESSION PARAMETERS**.

- Select option **A ADD**. Set the **ID** to `CX1SRV`. Press the **ENTER** key.



- Set the **STATE** to `E`, **MODE** to `S`, **VERIFY OPTIONS** to `0`, **CERTIFICATE ID** to `CX1CERT`, **CYPHER LIST** blank, **TLSV1**, **SSLV2** and **SSLV3** to `Y`, **NETWORK** to `T`, **IP ADDRESS** blank, **TCP PORT** to `5100`, leave the **IP HEADER** set to `N`. At the **DO YOU WANT TO GO**

ON? prompt press the ENTER key.

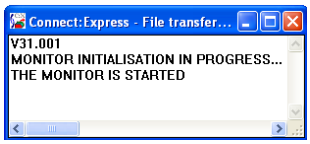
```
C:X/UNIX 150-0 -------------SSL SESSION PARAMETERS----------------------- /cx1
OPTION ===>

ID                          : CX1CLI
STATE                       : E                     (E:ENABLED,H:DISABLED)
MODE                        : C                     (C:CLIENT,S:SERVER)
VERIFY OPTIONS              : 0                     (0:NONE,1:PEER
                                                    2:PEER + FAIL_IF_NO_PEER_CERT)
CERTIFICATE ID              : CX1CERT.
CIPHER LIST                 : ................      (LIST FILE NAME)
SSL PROTOCOL VERSIONS :
TLSV1 : Y SSLV3 : Y SSLV2 : Y

CA LIST (SERVER MODE)       : ........              ('CACERT-ID',#LIST)
DH PARAMETERS (SERVER)      : ................      (FILE NAME)
LOCAL NETWORK ADDRESS (SERVER MODE)  :
NETWORK (T:TCPIP)           : .
TCP/IP : IP ADDRESS         : ...............       TCP PORT : .....
         IP HEADER          : N


DO YOU WANT TO GO ON ?                      UPD : ............... ........
-ENTER- NEXT FIELD                    -F3- CANCEL                  -F8- COMPLETION
```

At this point the configuration is finished. Press F3 to go back to the **MAIN MENU**. For the new server to be enabled, you stop and restart Connect:Express.

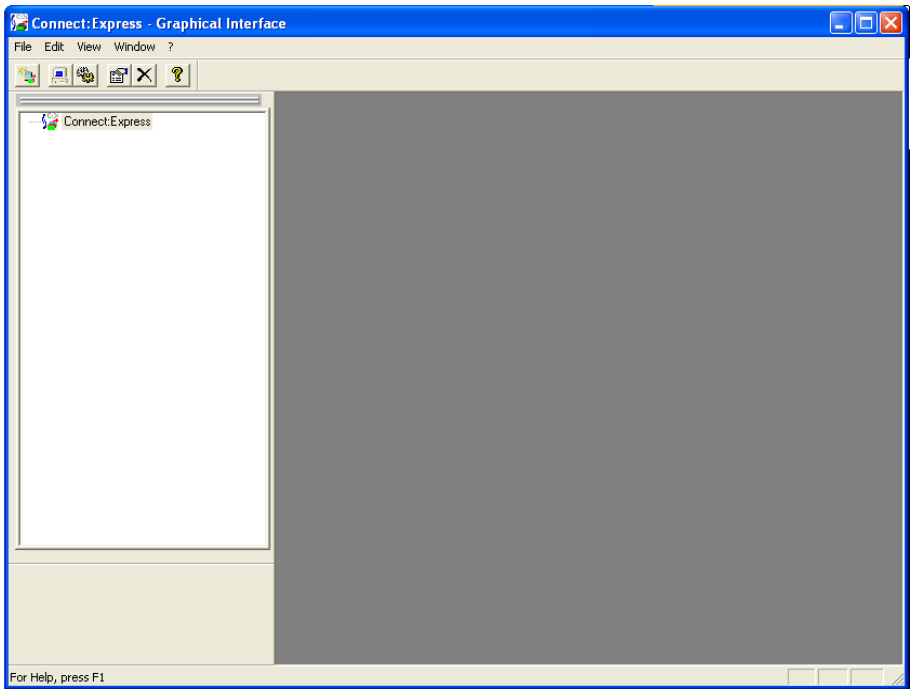
Configure Connect:Express for Windows

To do a secured file transfer to another C:X instance we need to create a few Connect:Express objects. The required certificates should have been already installed as described in the section **Import the self-signed certificates in C:X for Microsoft Windows**. The creation of the configuration objects is done using the CX Graphical Interface and the CX Monitor must be running. Do the following.

- Launch CX Monitor from the **Start** menu: **All Programs** > **CONNECT Express** > **CX2** > **CX Monitor**.



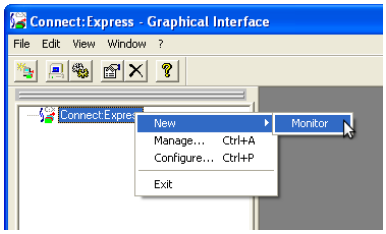
- Launch Graphical Interface from the **Start** menu: **All Programs** > **CONNECT Express** > **CX2** > **CX Graphical Interface**.



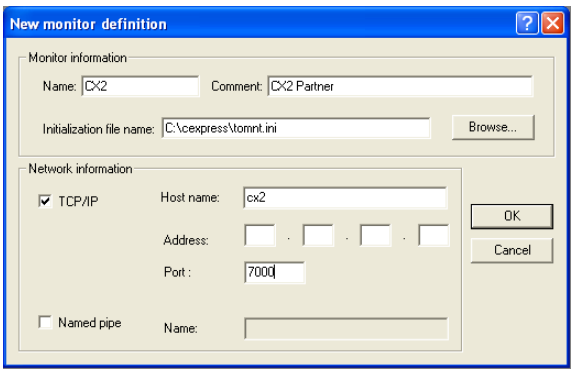
Create the monitor instance CX2

The first object to create is the monitor instance CX2.

- On the left pane, right-click **Connect:Express** > **New** > **Monitor**.



- Set the **Name** to `CX2`, optionally the **Description**, the **Initialization file name** to `tomnt.ini` (use the **Browse...** button to find the file in the C:X installation directory), tick the **TCP/IP** check box, set the **Host name** to `cx2` and the Port to `7000`.



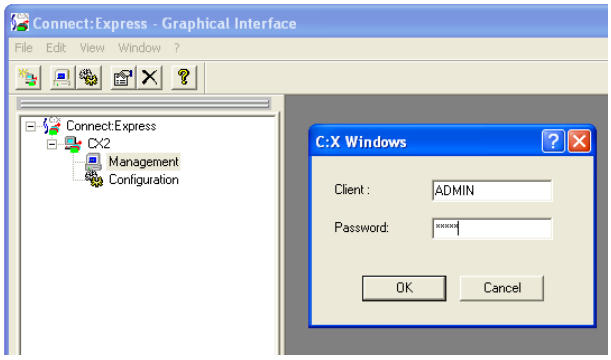
The host name must be the same as the host where the CX Monitor is running. The CX Graphical Interface enables to manage remote C:X instances. The port value is the default from the installation.

- Click **OK**.

Create the client SSL session

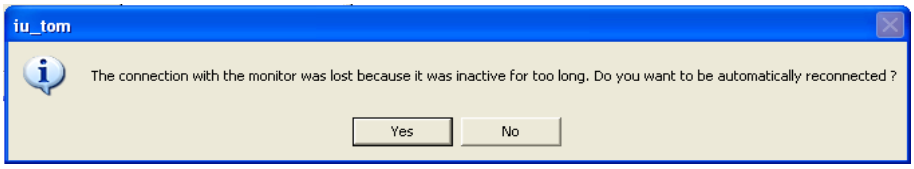
Even if we will not use SSL client authentication in this tutorial, C:X for Windows requires to define a client session.

- Expand the **CX2** node in the left pane and double-click on **Management**. Sign in using `ADMIN` for both the **Client** and the **Password**.



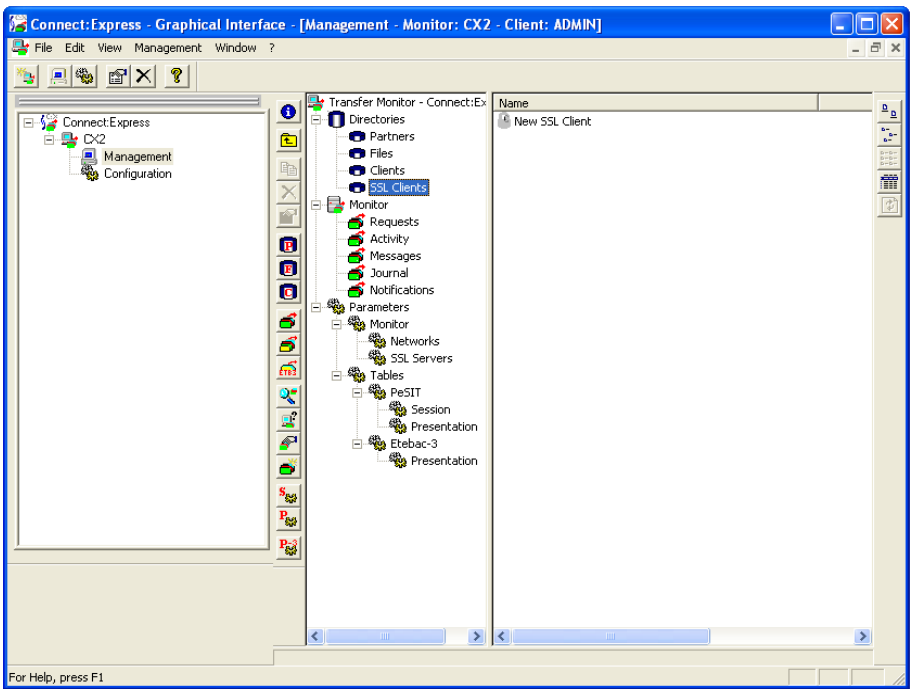
Note:

- The CX Graphical Interface establishes a connection with the CX monitor. This connection has a timeout and in case of inactivity the connection will be stopped. If it happens at some point of a configuration operation, you will see the following message box.

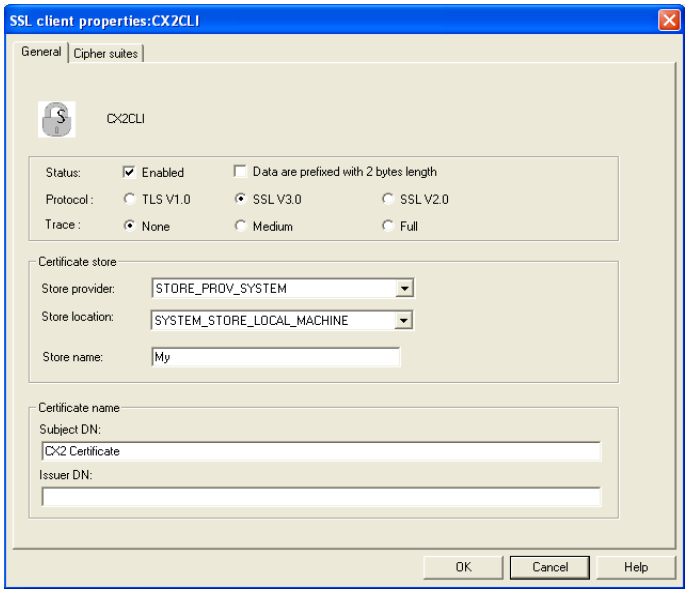


Just click the **OK** button and your work will continue normally.

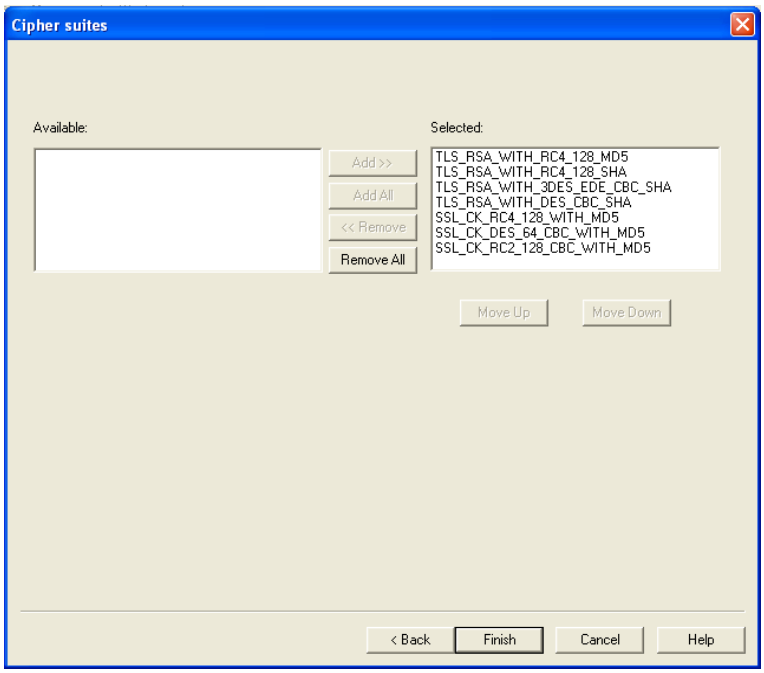
- In the middle pane select **Directories** > **SSL Clients** then double-click on **New SSL Client** in the right pane.



- Set the **Name** to `CX2CLI`, ensure the **Enabled** check box is ticked, select **Protocol** SSL V3.0, **Store provider** `STORE_PROV_SYSTEM`, **Store location** `SYSTEM_STORE_LOCAL_MACHINE`, set **Store provider** to `My`, **Subject DN** to `CX2 Certificate`.



- Click the **Next** button. Click the **Add All** button.

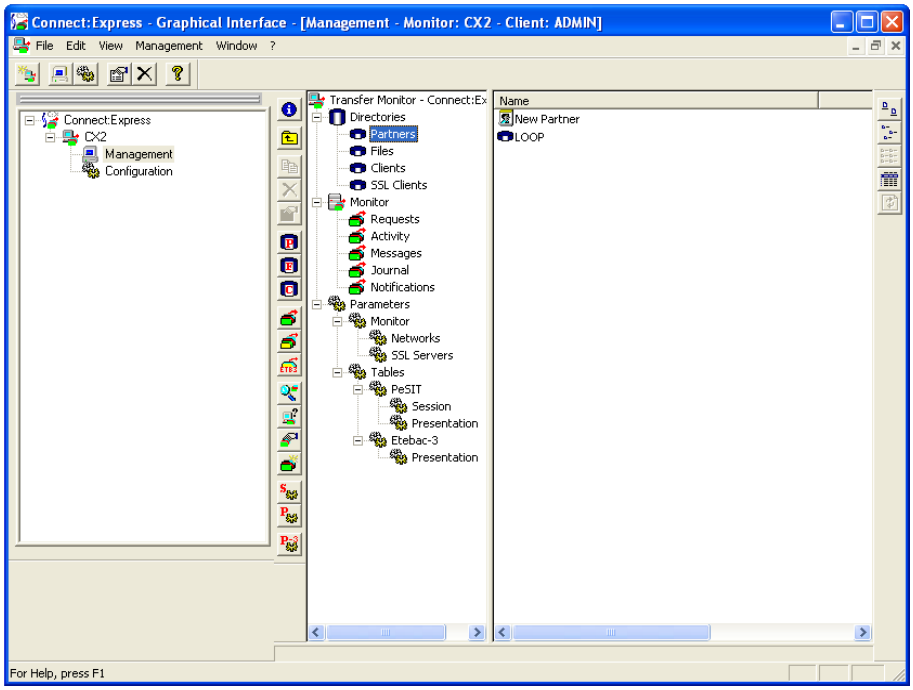


- Click the **Finish** button.

Create the partner CX1

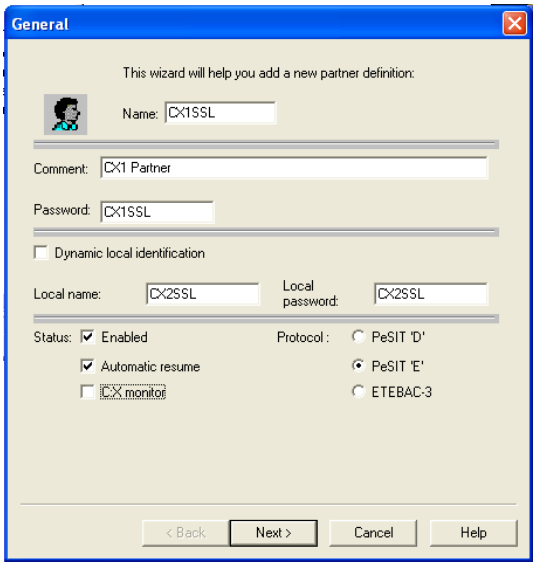
We need to define the partner to which we will send the file.

- In the middle pane select **Directories** > **Partners** then double-click on **New Partner** in the right pane.

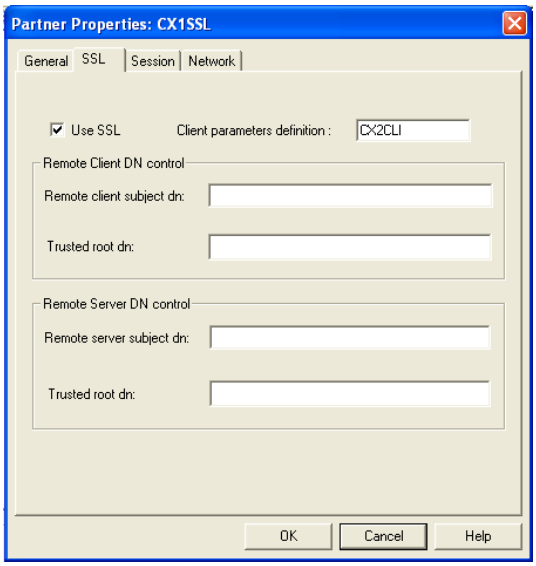


- Set the **Name** to `CX1SSL`, optionally a **Comment**, the **Local name** to `CX2SSL`, **Local password** to `CX2SSL`, leave the

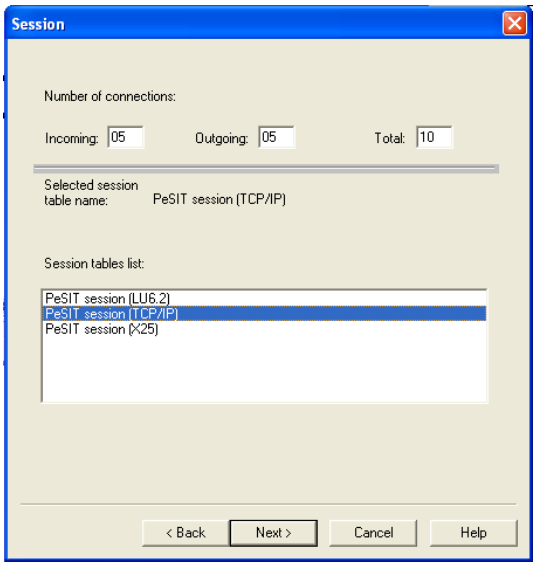
C:X monitor check box non ticked.



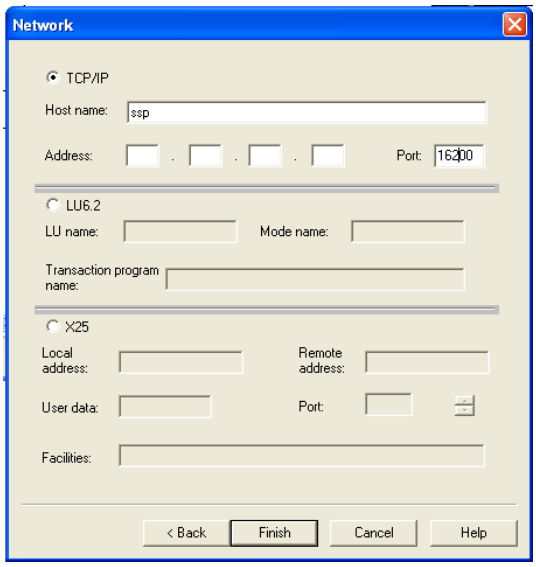
- Click the **Next** > button. Tick the **Use SSL** box, set the **Client parameter definition** to `CX2CLI`.



- Click the **Next** > button. Set **Incoming** to `5`, **Outgoing** to `5`, **Total** to `10`, in the **Session tables list** select `PeSIT session (TCP/IP)`.



- Click the **Next** > button. Set the **Host name** to `ssp`, the **Port** to `16200`.

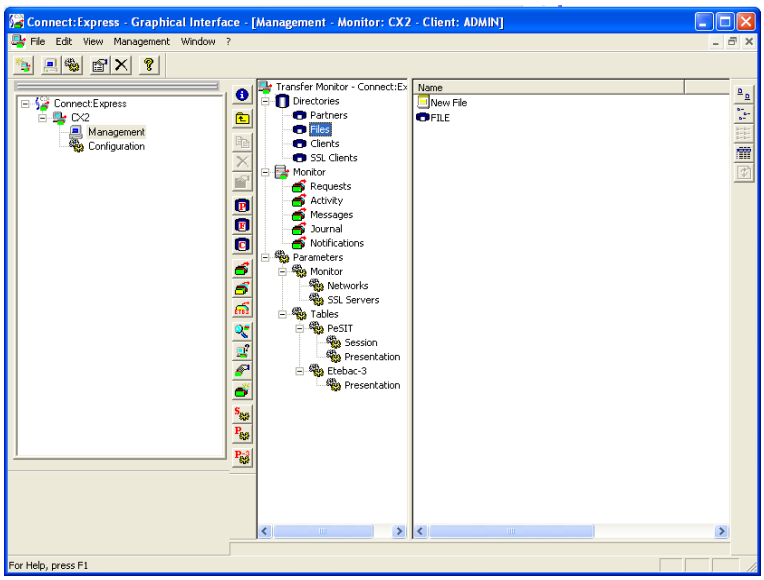


- Click the **Finish** button.

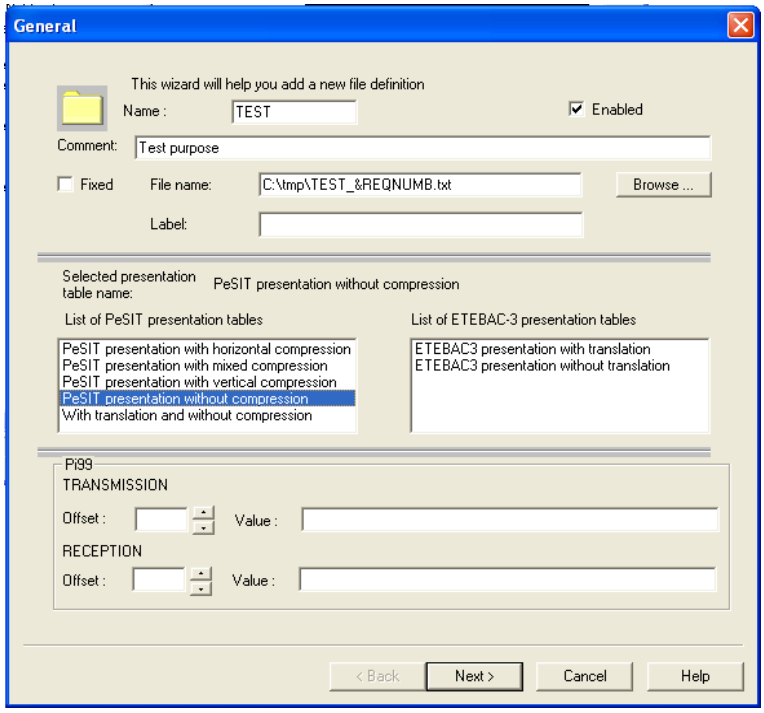
Create the file definition

The last object to define is the characteristics of the file to transfer.

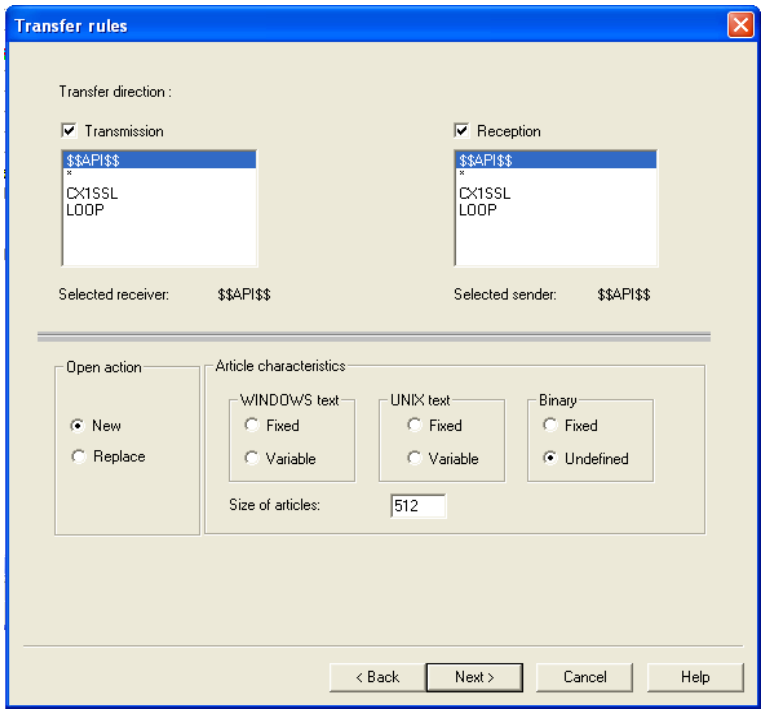
- In the middle pane select **Directories** > **Files** then double-click on **New File** in the right pane.



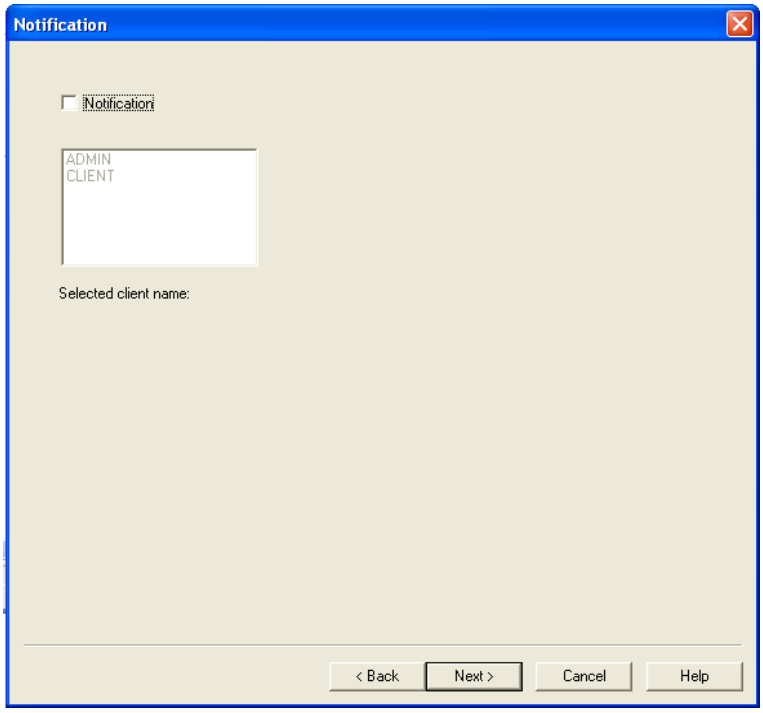
- Set the **NAME** to `TEST`, optionally the **Comment**, the **File name** to `C:\tmp\TEST_&REQNUMB.txt` (the file name will be explained later), in **List of PeSIT presentation table** select `PeSIT presentation without compression`.



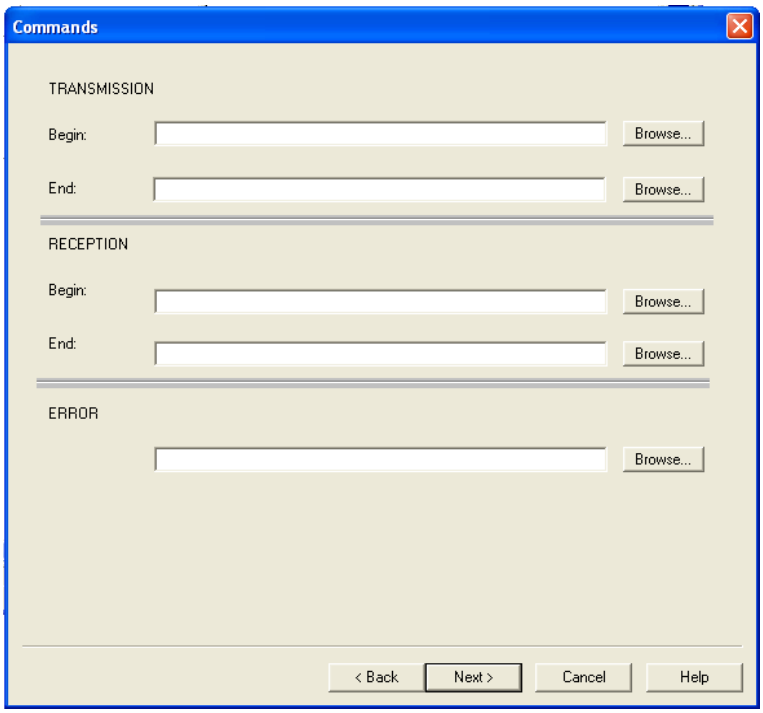
- Click the **Next >** button.
- Tick the **Transmission** check box, select `$$API$$` in the list, tick the **Reception** check box, select `$$API$$` in the list, click the **New** radio button in the **Open action** box, click the **Undefined** radio button in the **Binary** box, set the **Size of articles** to `512`.



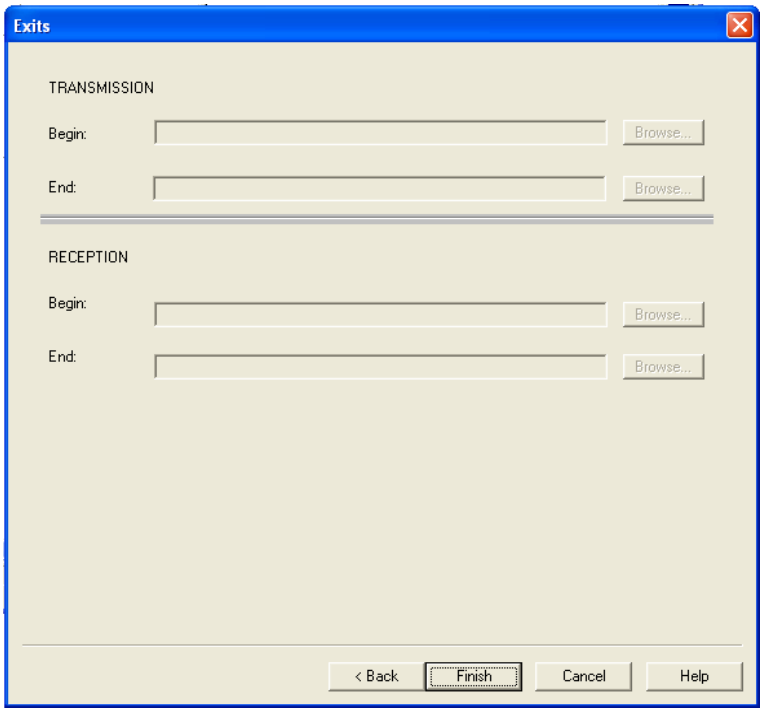
- Click the **Next >** button.
- Leave the **Notification** check box non ticked.



- Click the **Next >** button.
- Leave of the fields blank.



- Click the **Next >** button.

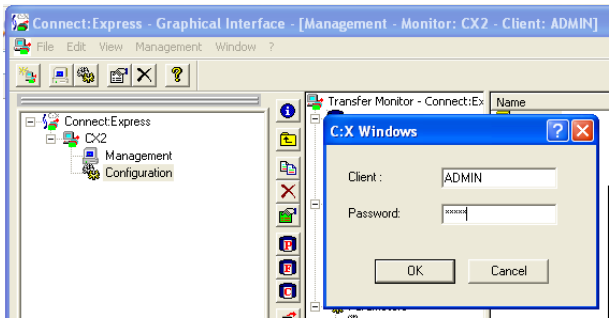


- Click the **Finish** button.

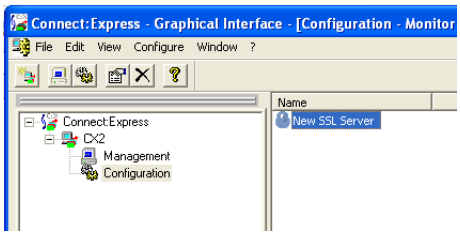
Define the SSL server

Because we want CX2 instance to be able to accept inbound connections, it must be configured as a server as well. This is done with the CX Graphical Interface.

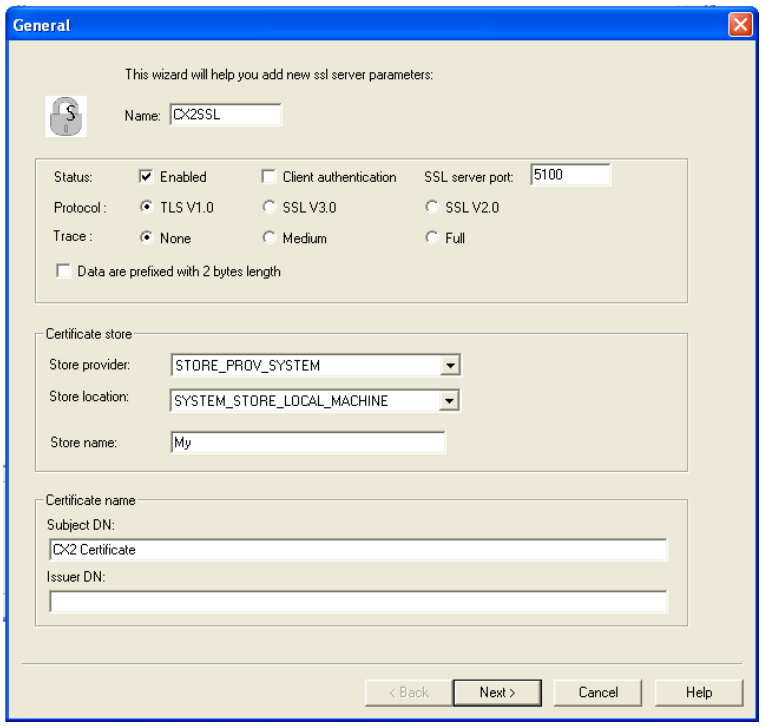
- In the left pane expand the **CX2** node and double-click on **Configuration**. Sign in using `ADMIN` for both the **Client** and the **Password**.



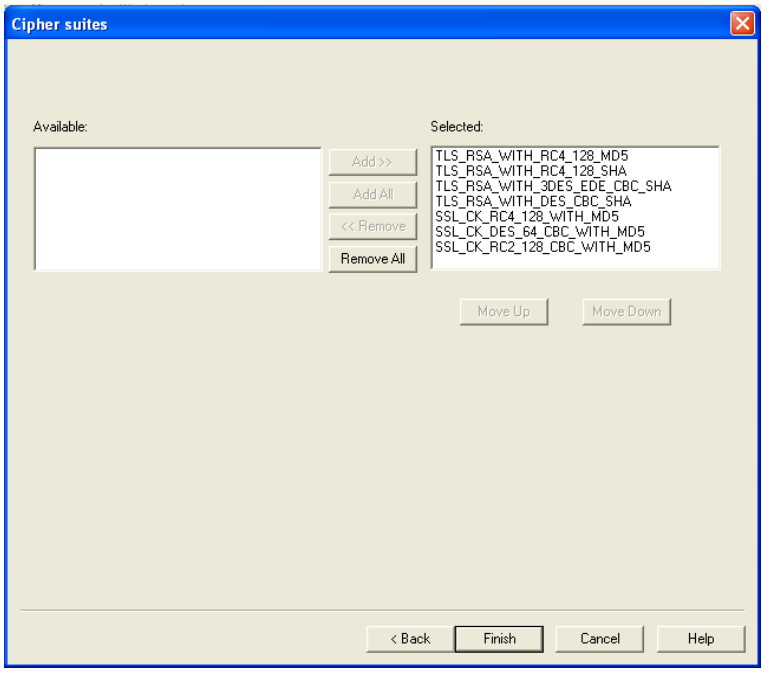
- In the right pane select **SSL Servers** and double-click on **New SSL Server**.



- Set the **Name** to `CX2SSL`, ensure the **Status Enabled** check box is ticked, set **SSL server port** to `5100`, clear the **Client authentication** check box, clear the **Data are prefixed with 2 bytes length**, select **Protocol** TLS V1.0, **Store provider** `STORE_PROV_SYSTEM`, **Store location** `SYSTEM_STORE_LOCAL_MACHINE`, set **Store provider** to `My`, **Subject DN** to `CX2 Certificate`.



- Click the **Next** button. Click the **Add All** button.



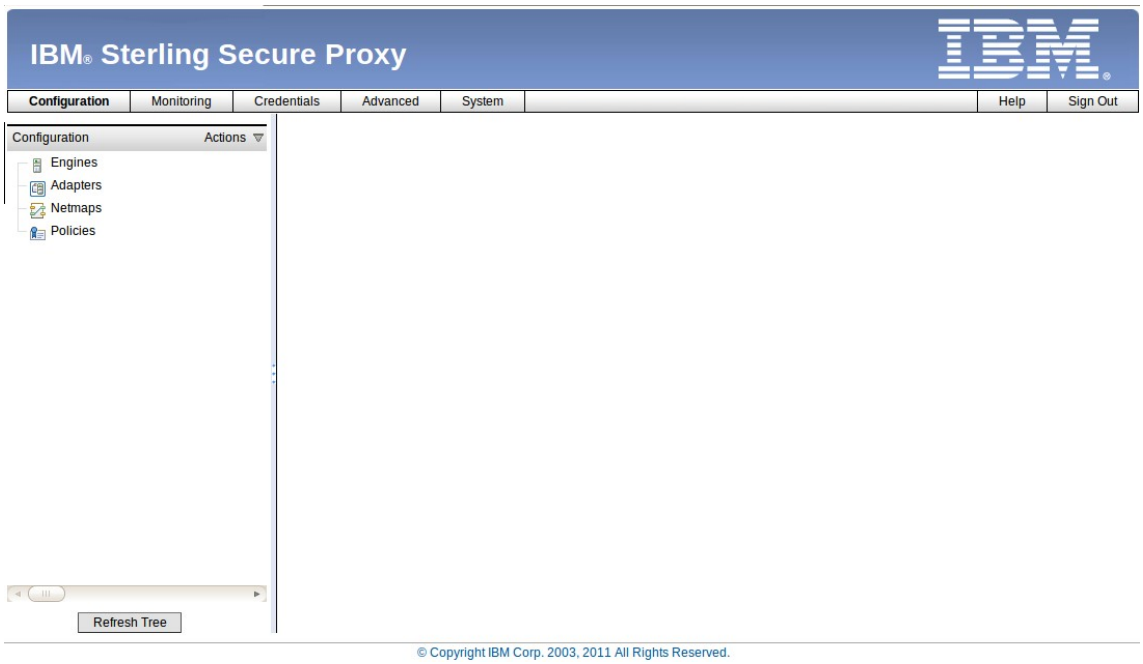
- Click the **Finish** button.

At this point the configuration of the CX2 instance is finished.

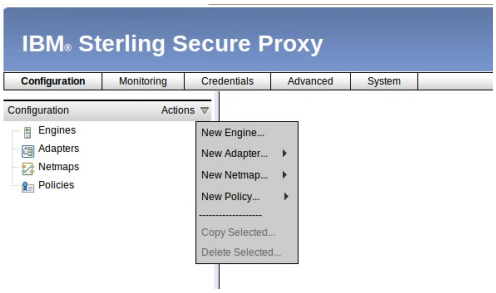
Configure the secured inbound and outbound nodes in SSP

To enable a C:X instance to securely send a file to another C:X instance through SSP some configuration is required at SSP level. In particular, the inbound and outbound connections must be configured. Do the following:

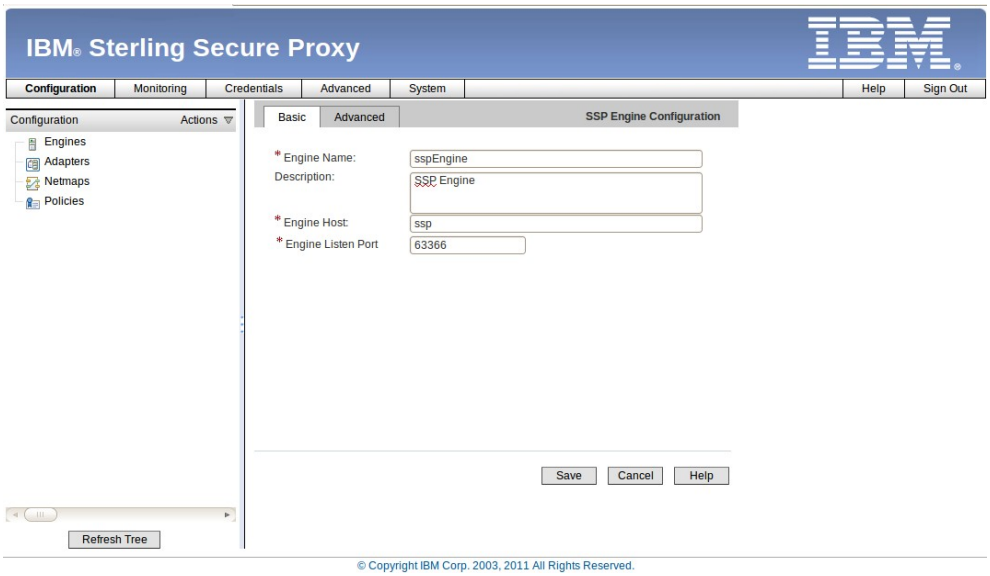
- Connect to the SSPcm graphical interface with administrator privileges (the section **Import the self-signed certificates in SSP** explains how to connect to the SSPcm graphical interface).



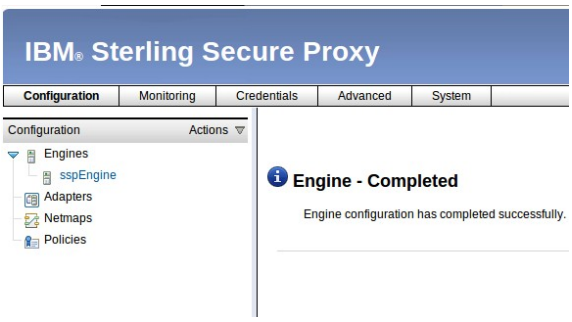
- Open the **Actions** drop down list and select **New Engine...**



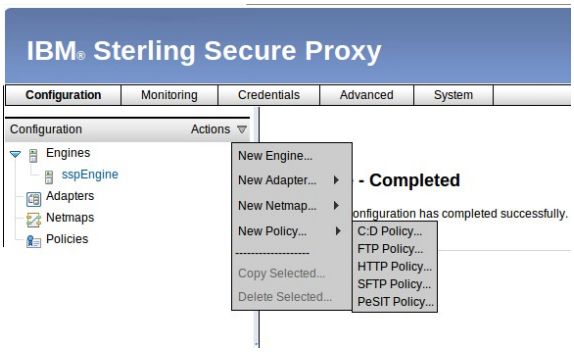
- Set the **Engine Name** with `sspEngine`, optionally the **Description** and the **Engine Host** with `ssp` (host name where SSP will run), leave the default **Engine Listen Port**.



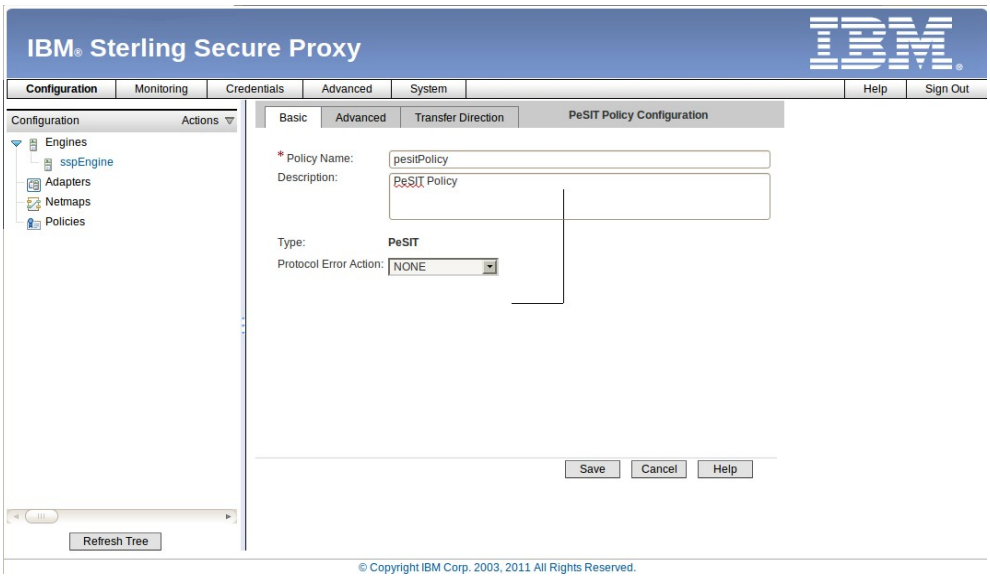
- Click the **Save** button. The newly created `sspEngine` must appear on the left pane as well as a successful message on the right pane.



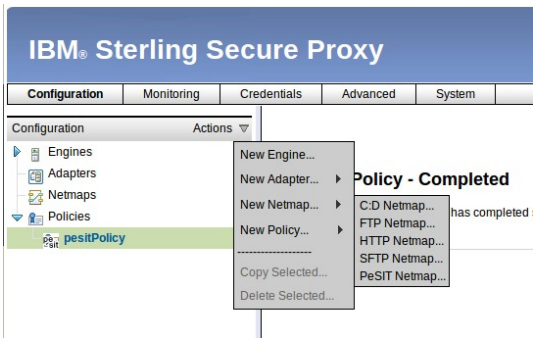
- Open the **Actions** drop down list, expand **New Policy...** and select **PeSIT Policy...**



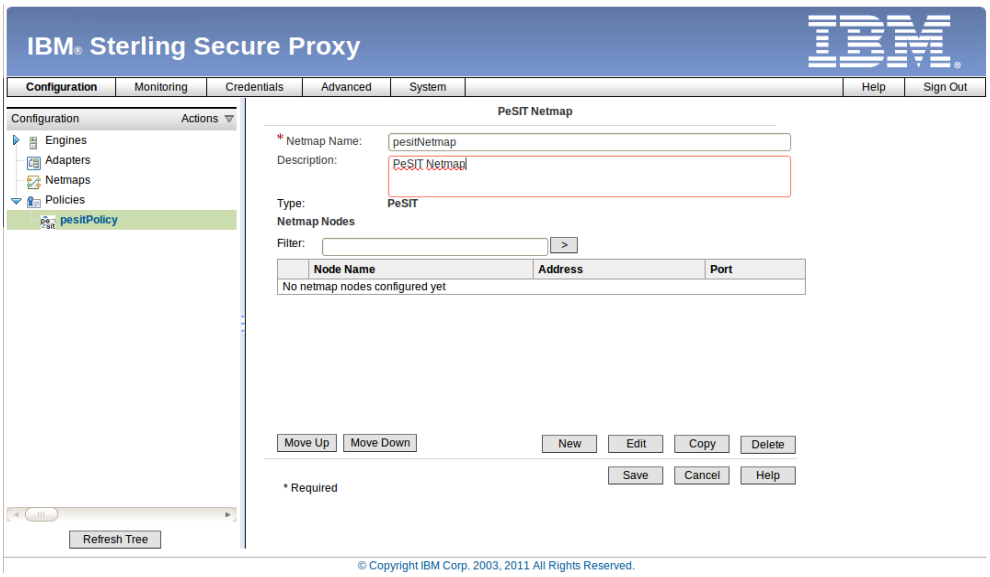
- Set the **Policy Name** to `pesitPolicy`, optionally the **Description**. Click the **Save** button.



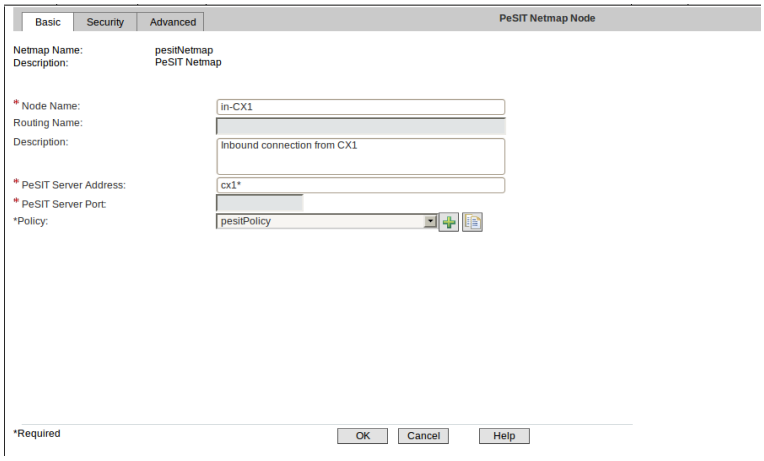
- Open the **Actions** drop down list, expand **New Netmap...** and select **PeSIT Netmap...**



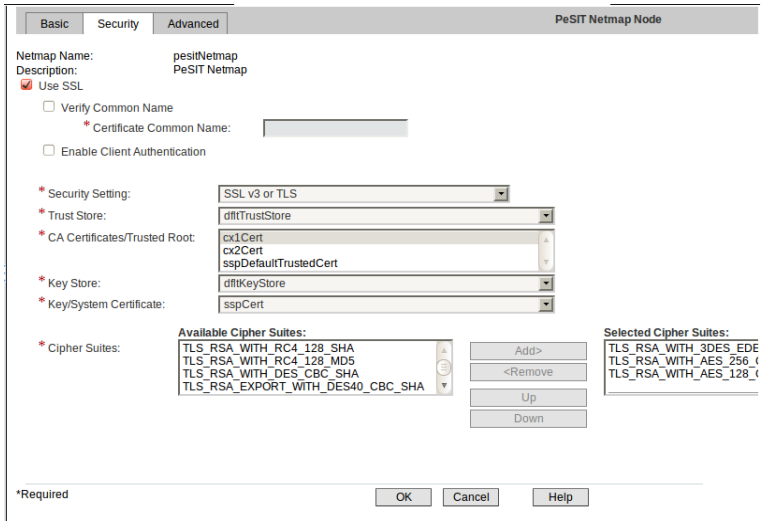
- Set the **Netmap Name** to `pesitNetmap`, optionally the **Description**. Click the **New** button.



- In the **Basic** tab, set the **Node Name** to `in-CX1`, optionally the **Description**, the **PeSIT Server Address** to `cx1*`, select the **Policy** `pesitPolicy`. Here, `cx1` is the host name from which the node `in-CX1` will accept connections. Specifying the wild-card (*) means that SSP will listen connections from `cx1` on all ports. The **PeSIT Server Port** field is then greyed because we implicitly specified all the ports. Click the **Ok** button.

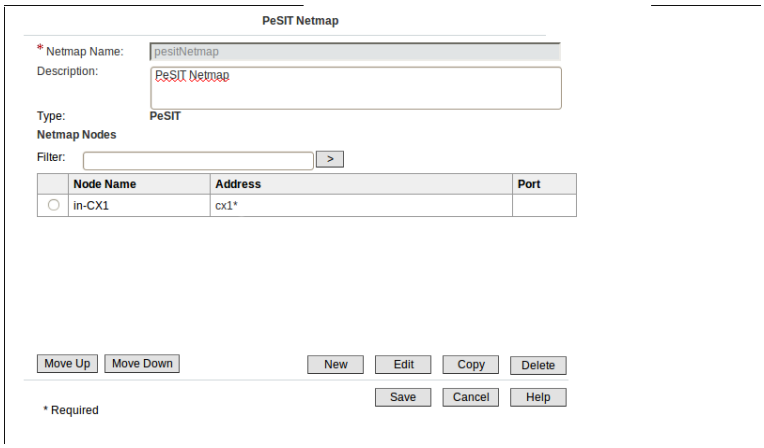


- Select the **Security** tab and, tick the **Use SSL** check box, select the `dfltTrustStore` from the **Trust Store** drop down list, select `cx1Cert` from the **CA Certificates/Trusted Root** drop down list, select `dfltKeyStore` from the **Key Store** drop down list, select `sspCert` for the **Key/System Certificate** drop down list.

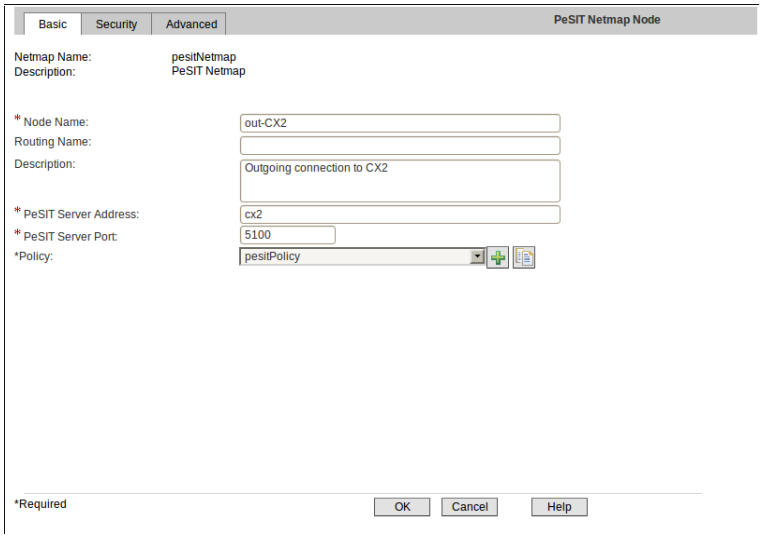


The `cx1Cert` CA certificate will be used to check that the certificate presented by the client `cx1` is trusted. For simplicity we are not using the client authentication but this information needs to be filled. The system certificate sspCert is the certificate the server will send to the cx1 client during the SSL handshake. The cx1 client will successfully check that `sspCert` is trusted if it finds sspCert in its trusted certificates store.

- Click the **Ok** button.

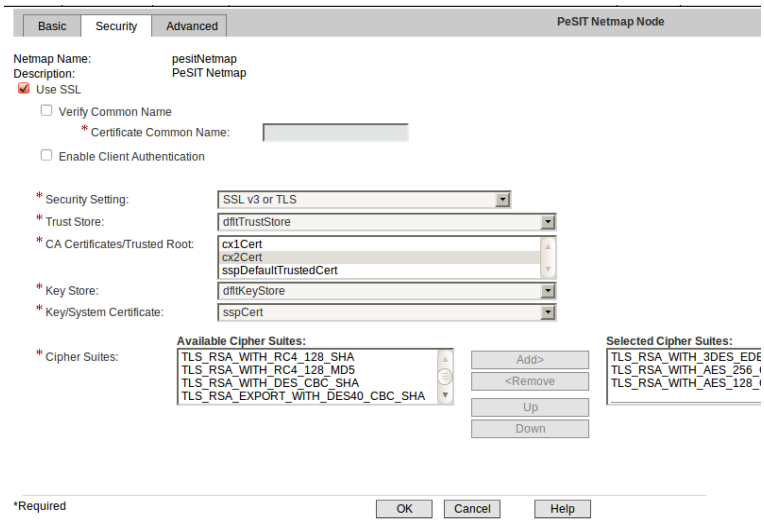


- Click the **New** button and, set the **Node Name** to `out-CX2`, optionally the **Description**, the **PeSIT Server Address** to `cx2`, the **PeSIT Server Port** to `5100` and select the `pesitPolicy` in the **Policy** drop down list.

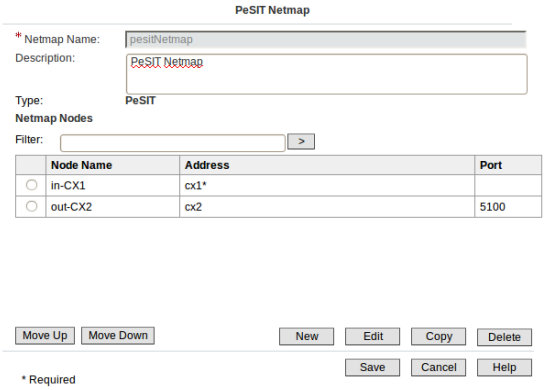


5100 is the value of the port on which the CX2 instance is listening incoming SSL connections.

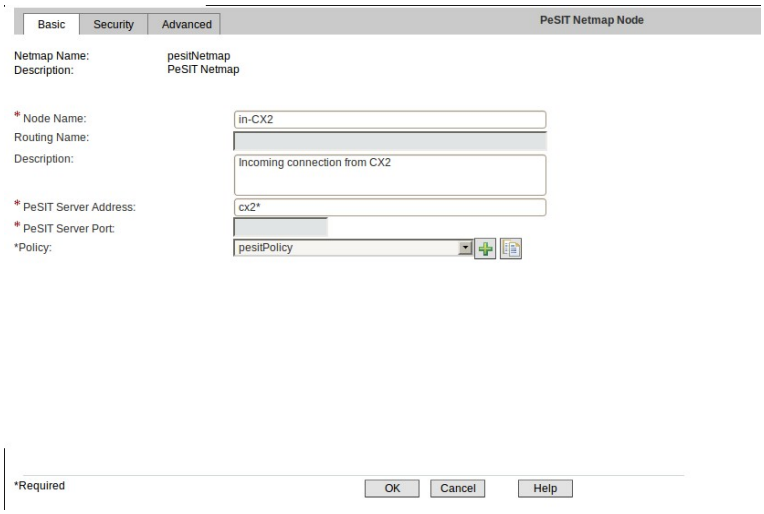
- Select the **Security** tab, tick the **Use SSL** check box, select the `dfltTrustStore` from the **Trust Store** drop down list, select `cx2Cert` from the **CA Certificates/Trusted Root** drop down list, select `dfltKeyStore` from the **Key Store** drop down list, select `sspCert` for the **Key/System Certificate** drop down list.



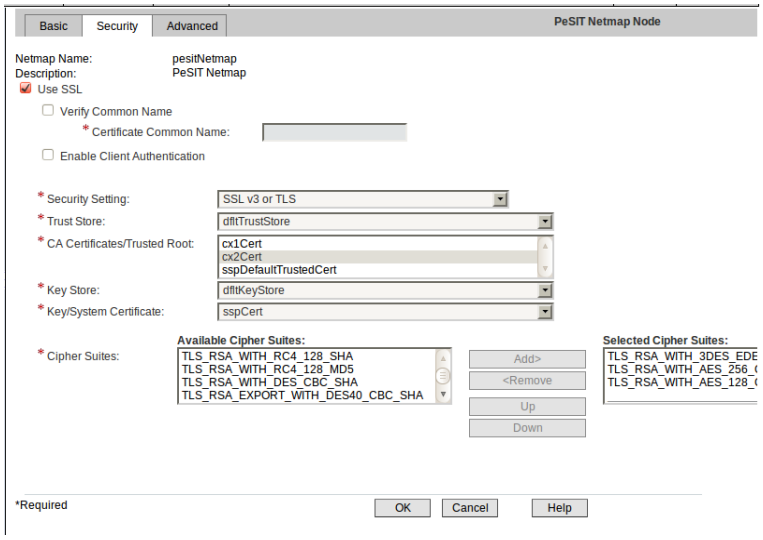
- Click the **Ok** button.



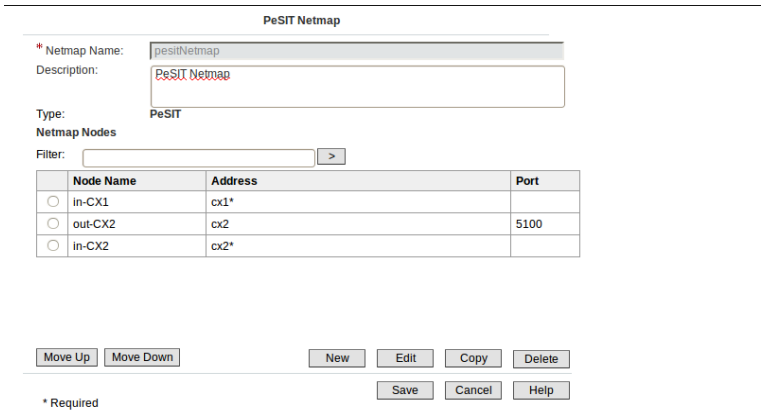
- Click the **New** button and, in the **Basic** tab, set the **Node Name** to `in-CX2`, optionally the **Description**, the **PeSIT Server Address** to `cx2*`, select the **Policy** `pesitPolicy` and click the **Ok** button.



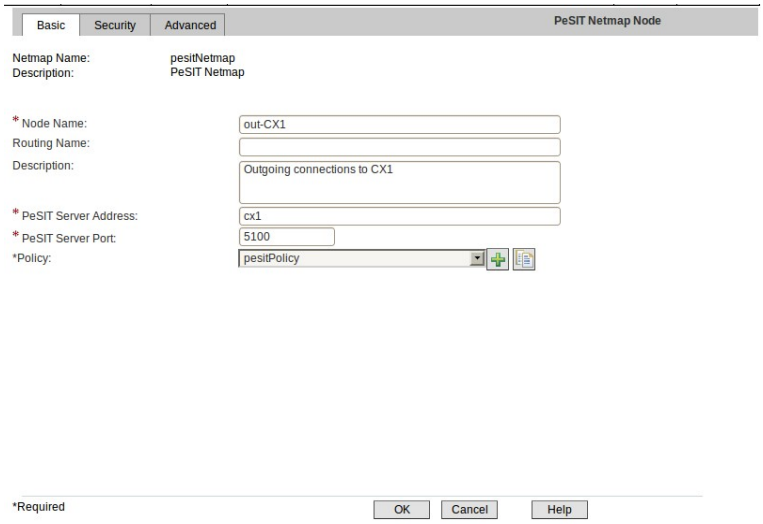
- Select the **Security** tab, tick the **Use SSL** check box, select the `dfltTrustStore` from the **Trust Store** drop down list, select `cx2Cert` from the **CA Certificates/Trusted Root** drop down list, select `dfltKeyStore` from the **Key Store** drop down list, select `sspCert` for the **Key/System Certificate** drop down list.



- Click the **Ok** button.

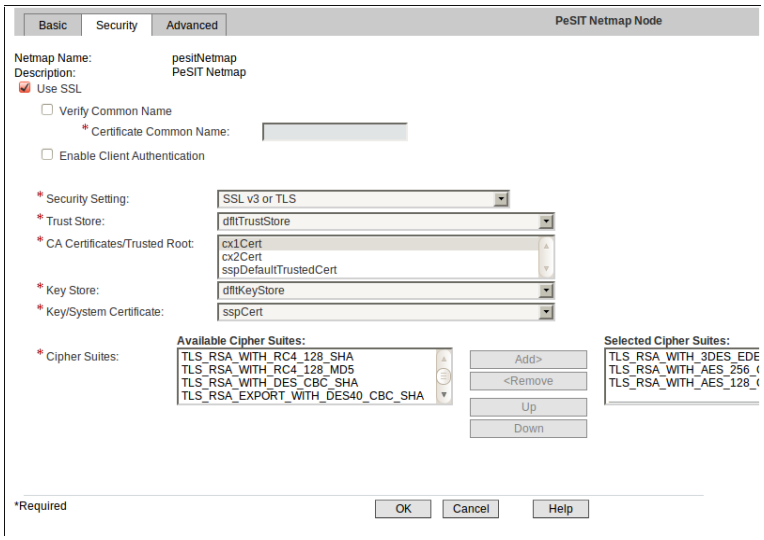


- Click the **New** button and, set the **Node Name** to `out-CX1`, optionally the **Description**, the **PeSIT Server Address** to `cx1`, the **PeSIT Server Port** to `5100` and select the `pesitPolicy` in the **Policy** drop down list.

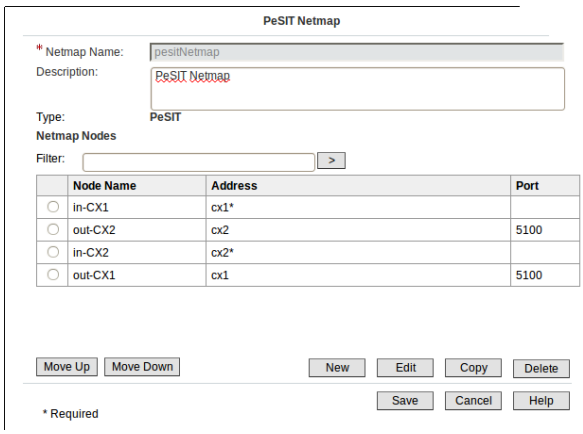


5100 is the value of the port on which the CX1 instance is listening incoming SSL connections.

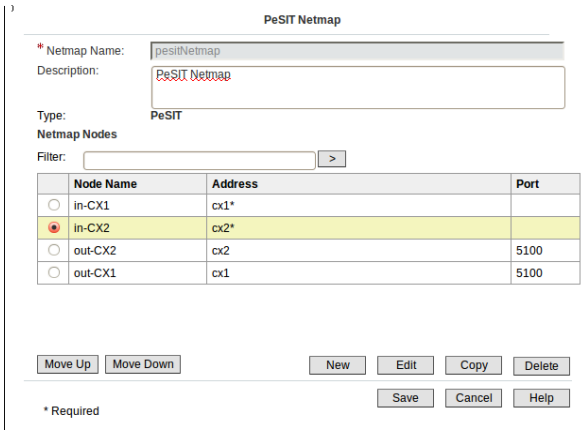
- Select the **Security** tab, tick the **Use SSL** check box, select the `dfltTrustStore` from the **Trust Store** drop down list, select `cx2Cert` from the **CA Certificates/Trusted Root** drop down list, select `dfltKeyStore` from the **Key Store** drop down list, select `sspCert` for the **Key/System Certificate** drop down list.



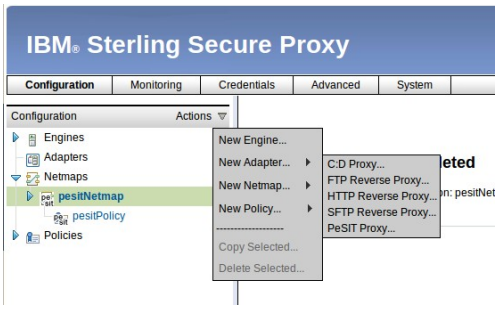
- Click the **Ok** button.



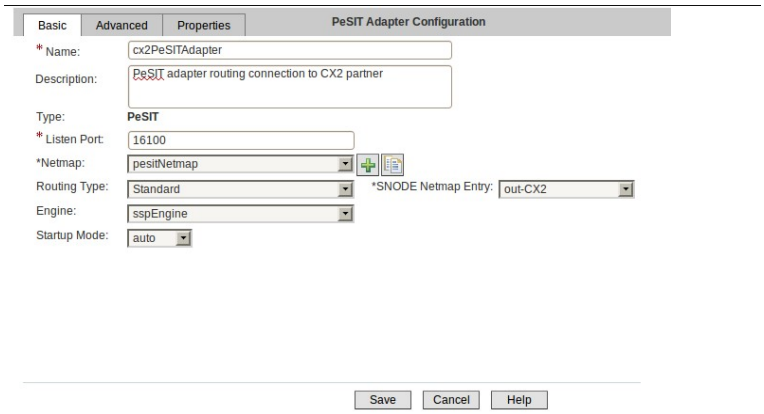
- Because of a known bug in the PeSIT adapter code, the node list must be reorganized. All the incoming nodes must be before the outgoing nodes. Select the node `in-CX2` and click the **Move Up** button.



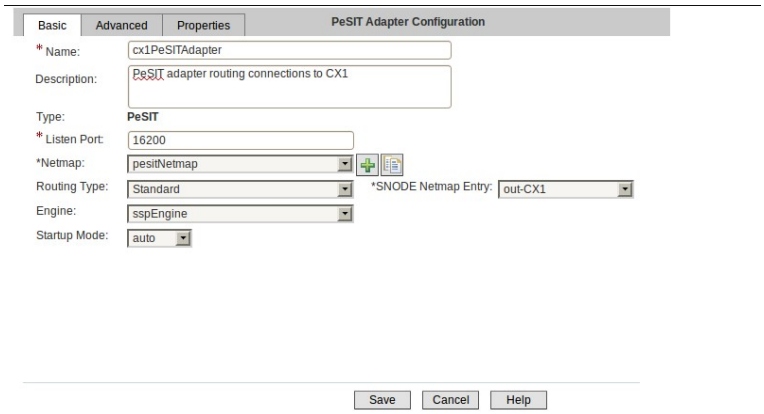
- Click the **Save** button to save changes.
- Open the **Actions** drop down list, expand **New Adapter...** and select **PeSIT Proxy...**



- In the **Basic** tab, set the **Name** to `cx2PeSITAdapter`, optionally a **Description**, the **Listen Port** to `16100`, select `pesitNemap` from the **Netmap** drop down list, select `out-CX2` from the **SNODE Netmap Entry** drop down list, select `sspEngine` from the **Engine** drop down list.



- Click the **Save** button.
- Open the **Actions** drop down list, expand **New Adapter...** and select **PeSIT Proxy...**



Test of file transfers

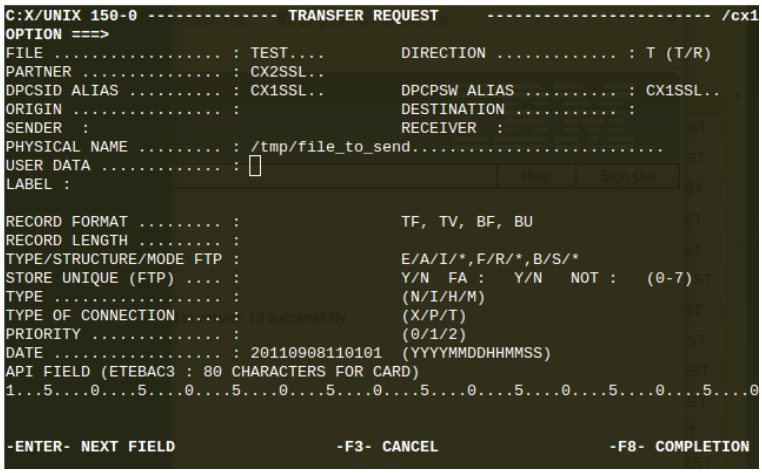
Everything should be now ready to do a file transfer. We will start with transferring a file from CX1 to CX2 then the opposite.

File transfer from CX1

We must choose a file to transfer. For the clarity of this tutorial, we suggest to copy the file `$TOM_DIR/install.txt` to `/tmp/file_to_send` Any other file can be chosen for a transfer.

- Run the C:X operator interface (command `$sterm`).

- Select option **`4 REQUEST`**. Set the **NAME** to `TEST`, **DIRECTION** to `T`, **PARTNER** to `CX2SSL`, **DPCSID ALIAS** and **DPCPSW ALIAS** to `CX1SSL` (not necessary if they have been defined in the partner directory), **DPCPSW ALIAS** to `CX1SSL`, **PHYSICAL NAME** to `/tmp/file_to_send`.



- Press the `F8` key. At the **DO YOU WANT TO GO ON ?** prompt press the `ENTER` key. The request number will be displayed at the bottom of the screen.

```
C:X/UNIX 150-0 -------------- TRANSFER REQUEST ------------------------ /cx1
OPTION ===>
FILE .................. : TEST            DIRECTION ............ : T (T/R)
PARTNER ............... : CX2SSL
DPCSID ALIAS .......... : CX1SSL           DPCPSW ALIAS .......... : CX1SSL
ORIGIN ................ :                  DESTINATION ........... :
SENDER  :                                 RECEIVER  :
PHYSICAL NAME ......... : /tmp/file_to_send
USER DATA ............ :
LABEL :

RECORD FORMAT ......... :                  TF, TV, BF, BU
RECORD LENGTH ......... :
TYPE/STRUCTURE/MODE FTP :                  E/A/I/*,F/R/*,B/S/*
STORE UNIQUE (FTP) .... :                  Y/N  FA :   Y/N   NOT :   (0-7)
TYPE .................. :                  (N/I/H/M)
TYPE OF CONNECTION .... :                  (X/P/T)
PRIORITY .............. :                  (0/1/2)
DATE .................. : 20110908110432  (YYYYMMDDHHMMSS)
API FIELD (ETEBAC3 : 80 CHARACTERS FOR CARD)
1...5....0....5....0....5....0....5....0....5....0....5....0....5....0....5....0

REQUEST ACCEPTED : 06800007 - <CR> TO LEAVE
-ENTER- NEXT FIELD                -F3- CANCEL                    -F8- COMPLETION
```

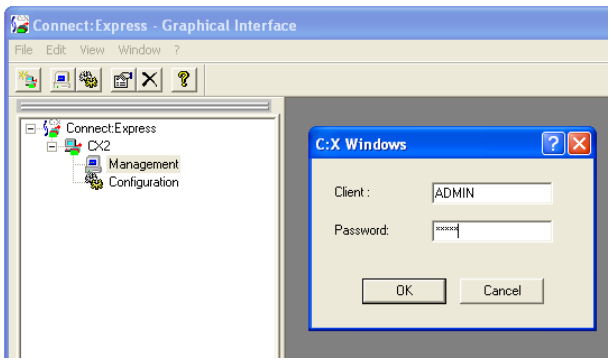
- Press the `ENTER` key then the `F3` key to go back to the **MAIN MENU**.
- Select **2 MONITOR** and **2 INTERROGATION OF LOG**. We should be able to see the success of the file transfer.

```
C:X/UNIX 150-0 -------------- INTERROGATION OF LOG ------------------------ /cx1
OPTION ===>                                          Thu Sep  8 11:16:44 2011
11/09/08 10:38:01 COMMUNICATION OPENED (I) WITH: CX2SSL   REQ: 06800006 PESIT
11/09/08 10:38:01 REQUEST 06800006 TEST     TRANSFER ACCEPTED    STRF 0000032486
11/09/08 10:38:01 REQUEST 06800006 TEST     TRANSFER STARTED     STRF 0000032486
11/09/08 10:38:01 REQUEST 06800006 (R) /tmp/TEST_A6800006.txt
11/09/08 10:38:01 REQUEST 06800006 TEST     TRANSFER ENDED       STRF 0000032486
11/09/08 10:38:01 REQUEST 06800006 RECEIVING     <- CX2SSL    FILE TEST     NUMB
11/09/08 10:38:01 COMMUNICATION CLOSED (I) WITH: CX2SSL   REQ: 06800006 PESIT
11/09/08 11:05:03 REQUEST 06800007 TEST     CX2SSL           SRC=0000 TRC=0000L
11/09/08 11:05:03 REQUEST 06800007 <- bfn      ACCEPTED  (N)
11/09/08 11:05:03 REQUEST 06800007 ABORT -> CX2SSL          SRC=0000 TRC=1250L
11/09/08 11:05:03 REQUEST 06800007 REJECTED <- CX2SSL       SRC=0000 TRC=1250L
11/09/08 11:05:03 COMMUNICATION NOT OBTAINED ->  CX2SSL   REQ: 06800007       RE
11/09/08 11:06:03 REQUEST 06800007 RETRY WITH PARTNER CX2SSL
11/09/08 11:06:04 COMMUNICATION OPENED (O) WITH: CX2SSL   REQ: 06800007 PESIT
11/09/08 11:06:04 REQUEST 06800007 TEST     TRANSFER ACCEPTED    STRF 0000005508
11/09/08 11:06:04 REQUEST 06800007 TEST     TRANSFER STARTED     STRF 0000005508
11/09/08 11:06:04 REQUEST 06800007 (T) /tmp/file_to_send
11/09/08 11:06:04 REQUEST 06800007 TEST     TRANSFER ENDED       STRF 0000005508
11/09/08 11:06:04 REQUEST 06800007 TRANSMITTING  -> CX2SSL    FILE TEST     NUMB
11/09/08 11:06:04 COMMUNICATION CLOSED (O) WITH: CX2SSL   REQ: 06800007 PESIT

    <- -F10-  -F3- END  -F7- PREVIOUS SCREEN  -F8- NEXT SCREEN    -F11- ->
```

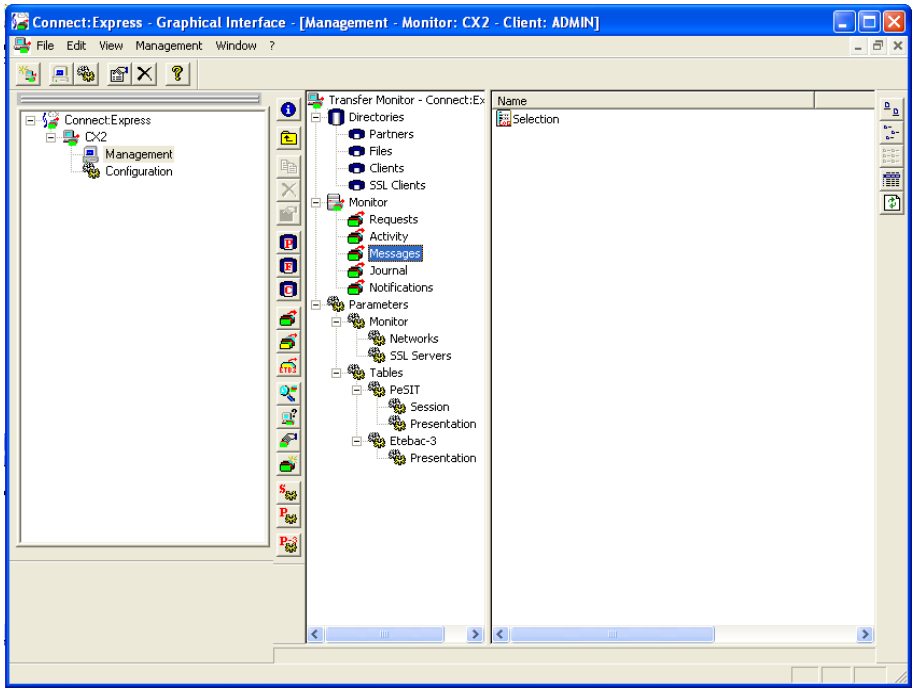
We can see from the log above that the request 06800007 was a transfer of the file `TEST` (`TEST  TRANSFER ACCEPTED`), it has been successful started (`TEST  TRANSFER STARTED`), it was a transmission (`(T)`) of the file `/tmp/file_to_send` and it has ended successfully (`TEST  TRANSFER ENDED`).

Another way to check that the file has been successfully transferred to the CX2 instance is to use the CX Graphical Interface on the cx2 host.

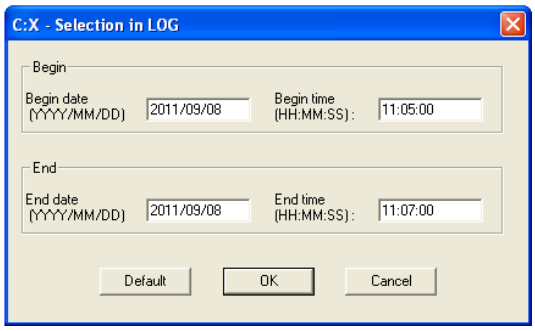
- Expand the **CX2** node in the left pane and double-click on **Management**. Sign in using `ADMIN` for both the **Client** and the **Password**.



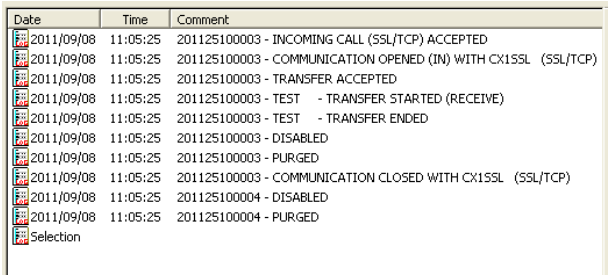
- In the middle pane select **Directories** > **Messages** then double-click on **Selection** in the right pane.



- Optionally use the **Begin time** and **End time** text fields to set the time slot. You can refer to the CX1 log for the time of the transfer (in this example the transfer has been done around 11:06:00). Click the **OK** button in the **Selection in LOG** window.



- We can see from the following log that the request 201125100003 was a reception of the file `TEST` (`TEST TRANSFER STARTED (RECEIVE)`) and it has ended successfully (`TEST – TRANSFER ENDED`).



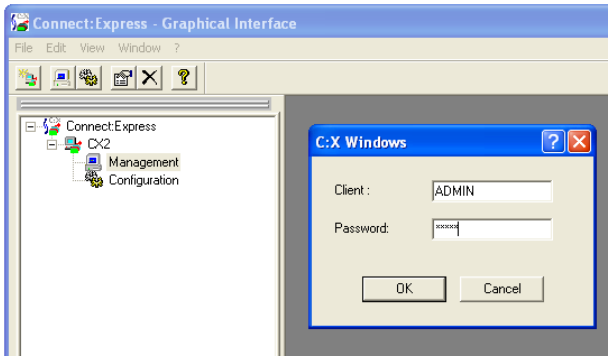
Date	Time	Comment
2011/09/08	11:05:25	201125100003 - INCOMING CALL (SSL/TCP) ACCEPTED
2011/09/08	11:05:25	201125100003 - COMMUNICATION OPENED (IN) WITH CX1SSL (SSL/TCP)
2011/09/08	11:05:25	201125100003 - TRANSFER ACCEPTED
2011/09/08	11:05:25	201125100003 - TEST - TRANSFER STARTED (RECEIVE)
2011/09/08	11:05:25	201125100003 - TEST - TRANSFER ENDED
2011/09/08	11:05:25	201125100003 - DISABLED
2011/09/08	11:05:25	201125100003 - PURGED
2011/09/08	11:05:25	201125100003 - COMMUNICATION CLOSED WITH CX1SSL (SSL/TCP)
2011/09/08	11:05:25	201125100004 - DISABLED
2011/09/08	11:05:25	201125100004 - PURGED
Selection		

Note that the request number (201125100003) is not the same as the request number in CX1 (06800007). This is normal because the 2 C:X instances maintain different request counters.

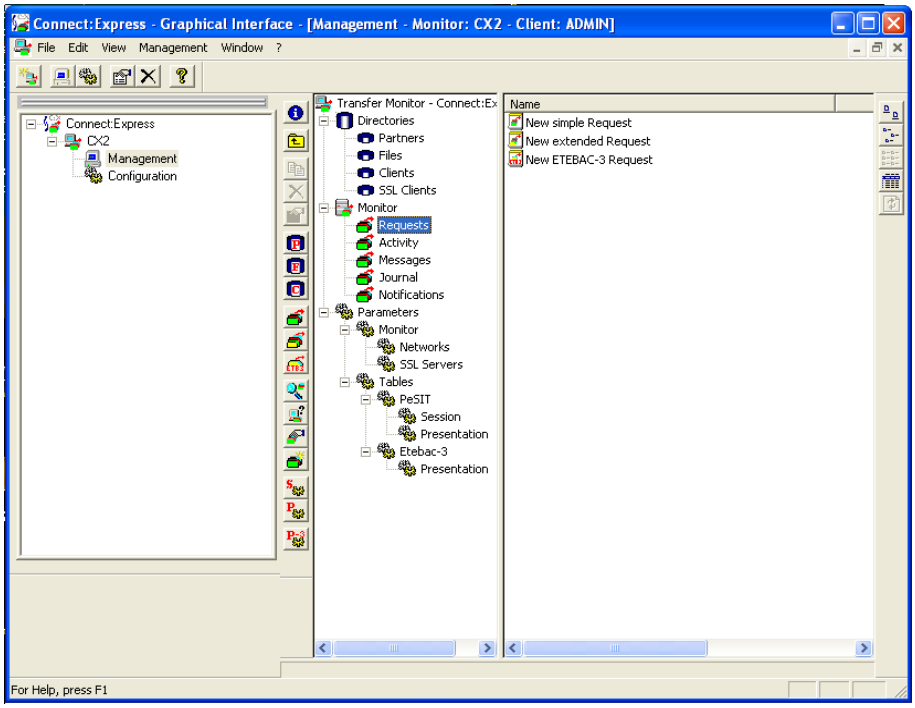
File transfer from CX2

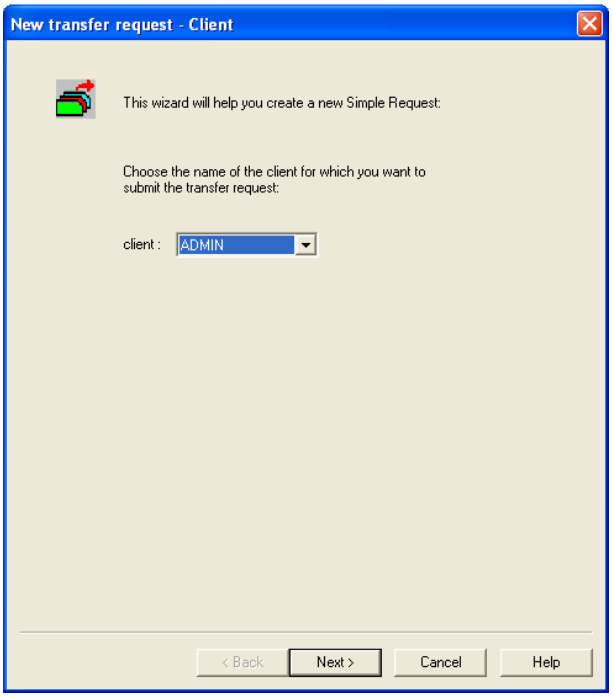
As for the file transfer from CX1 partner we need to choose a file to transfer. For the clarity of this tutorial, we suggest to copy the file `C:\cexpress\tomnt.ini` to `C:\tmp\file_to_send.txt`. Any other file can be chosen for a transfer. Start the CX Graphical Interface to do the transfer.

- Expand the **CX2** node in the left pane and double-click on **Management**. Sign in using `ADMIN` for both the **Client** and the **Password**.

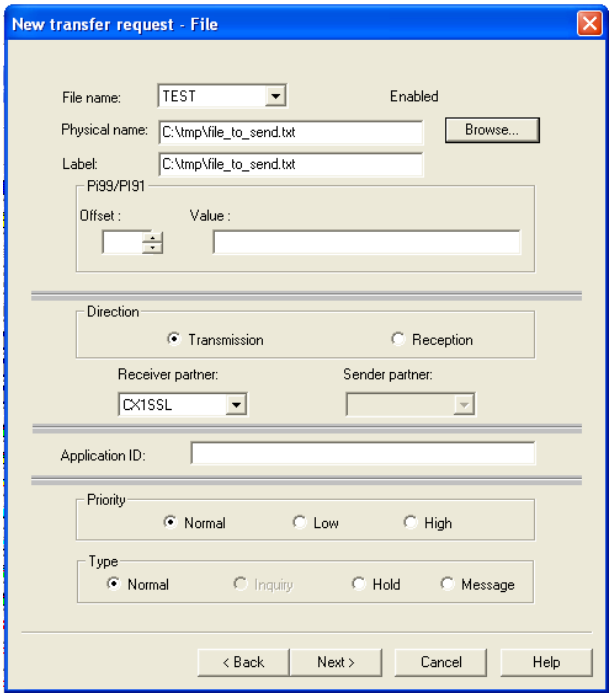


- In the middle pane select **Monitor** > **Requests** then double-click on **New simple Request** in the right pane.

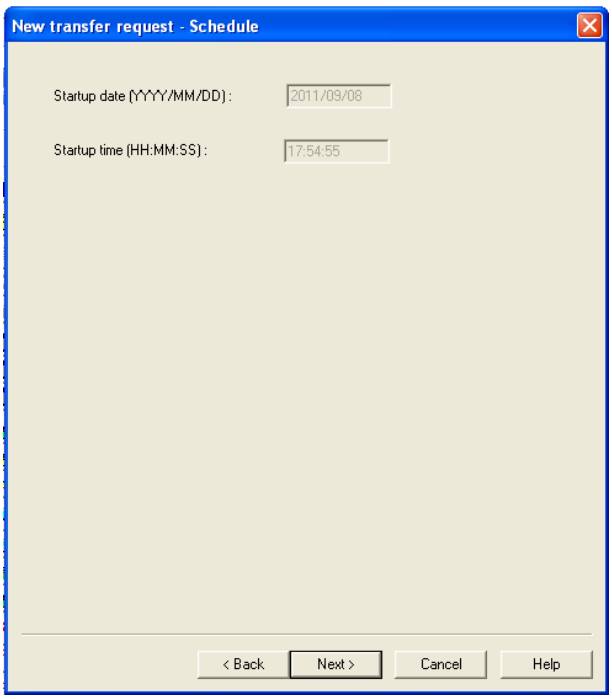




- In the **Client** window, leave `ADMIN` selected in the **client** drop down list and click the **Next >** button.
- In the **File** dialog window, select `TEST` in the **File name** drop down list, set the **Physical name** to `C:\tmp\file_to_send.txt` , leave the value for the **Label**, check that **Transmission** radio button is selected in the **Direction** box, select `CX1SSL` in the **Receiver partner** list.

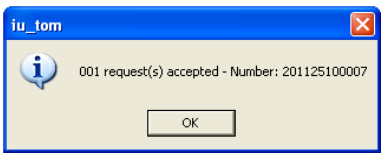


- Click the **Next >** button.
- In the **Schedule** dialog window click the **Next >** button.

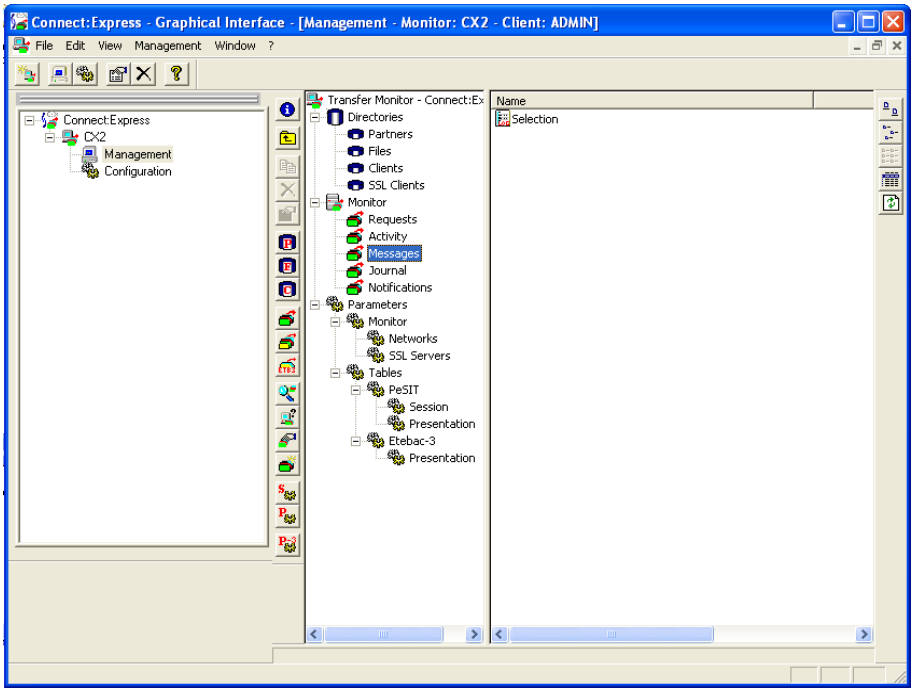


- In the **Notification** dialog window click the **Finish** button.

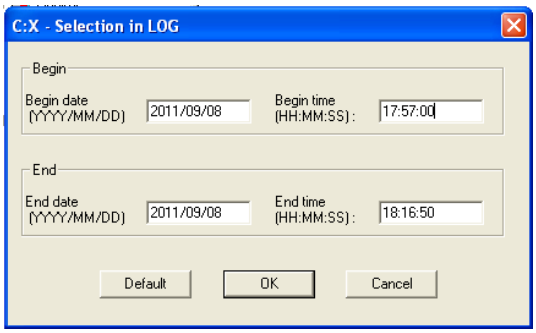
- A message window will be displayed giving the request number.



- Click the **OK** button.
- To check the result of the transfer, select **Directories** > **Messages** in the middle pane then double-click on **Selection** in the right pane.



- Optionally use the **Begin time** and **End time** text fields to set the time slot. You can refer to the CX1 log for the time of the transfer (in this example the transfer has been done around 17:58:00). Click the **OK** button in the **Selection in LOG** window.



- We can see from the following log that the request 201125100007 was a transmission of the file `TEST` (`TEST TRANSFER STARTED (TRANSMIT)`) and it has ended successfully (`TEST – TRANSFER ENDED`).

Date	Time	Comment
2011/09/08	17:58:10	C20112510006 - 201125100007 - ACCEPTED (N)
2011/09/08	17:58:10	201125100007 - SELECTED
2011/09/08	17:58:10	201125100007 - COMMUNICATION OPENED (OUT) WITH CX1SSL (SSL/TCP)
2011/09/08	17:58:10	201125100007 - TRANSFER ACCEPTED
2011/09/08	17:58:10	201125100007 - TEST - TRANSFER STARTED (TRANSMIT)
2011/09/08	17:58:10	201125100007 - TEST - TRANSFER ENDED
2011/09/08	17:58:10	201125100007 - DISABLED
2011/09/08	17:58:11	201125100007 - COMMUNICATION CLOSED WITH CX1SSL (SSL/TCP)
2011/09/08	17:58:11	201125100007 - PURGED
Selection		

Another way to check the result of the transfer is to use the C:X operator interface on host cx1.

- From the **MAIN MENU** select **2 MONITOR** and **2 INTERROGATION OF LOG**. We should be able to see the success of the file transfer.

```
C:X/UNIX 150-0 -------------- INTERROGATION OF LOG ------------------------ /cx1
OPTION ===>                                                Thu Sep  8 18:34:35 2011
11/09/08 11:06:04 REQUEST 06800007 (T) /tmp/file_to_send
11/09/08 11:06:04 REQUEST 06800007 TEST     TRANSFER ENDED       STRF 0000005508
11/09/08 11:06:04 REQUEST 06800007 TRANSMITTING  -> CX2SSL    FILE TEST     NUMB
11/09/08 11:06:04 COMMUNICATION CLOSED (O) WITH: CX2SSL   REQ: 06800007 PESIT
11/09/08 14:06:31 REQUEST 06800008 TEST     CX2SSL             SRC=0000 TRC=0000L
11/09/08 14:06:31 REQUEST 06800008 <- bfn      ACCEPTED  (N)
11/09/08 14:06:32 COMMUNICATION OPENED (O) WITH: CX2SSL   REQ: 06800008 PESIT
11/09/08 14:06:32 REQUEST 06800008 TEST     TRANSFER ACCEPTED    STRF 0000007615
11/09/08 14:06:32 REQUEST 06800008 TEST     TRANSFER STARTED     STRF 0000007615
11/09/08 14:06:32 REQUEST 06800008 (T) /tmp/file_to_send
11/09/08 14:06:33 REQUEST 06800008 TEST     TRANSFER ENDED       STRF 0000007615
11/09/08 14:06:33 REQUEST 06800008 TRANSMITTING  -> CX2SSL    FILE TEST     NUMB
11/09/08 14:06:33 COMMUNICATION CLOSED (O) WITH: CX2SSL   REQ: 06800008 PESIT
11/09/08 17:58:47 COMMUNICATION OPENED (I) WITH: CX2SSL   REQ: 06800009 PESIT
11/09/08 17:58:47 REQUEST 06800009 TEST     TRANSFER ACCEPTED    STRF 0000019010
11/09/08 17:58:47 REQUEST 06800009 TEST     TRANSFER STARTED     STRF 0000019010
11/09/08 17:58:47 REQUEST 06800009 (R) /tmp/TEST_A6800009.txt
11/09/08 17:58:47 REQUEST 06800009 TEST     TRANSFER ENDED       STRF 0000019010
11/09/08 17:58:47 REQUEST 06800009 RECEIVING     <- CX2SSL    FILE TEST     NUMB
11/09/08 17:58:48 COMMUNICATION CLOSED (I) WITH: CX2SSL   REQ: 06800009 PESIT

   <- -F10-  -F3- END  -F7- PREVIOUS SCREEN  -F8- NEXT SCREEN    -F11- ->
```

We can see from the log above that the request 06800009 was a transfer of the file `TEST` (`TEST TRANSFER ACCEPTED`), it has been successfully started (`TEST TRANSFER STARTED`), it was a reception (`(R)`) and the received file has been save under the name `/tmp/TEST_A6800009.txt` (the file name in the definition of the file `TEST` is `/tmp/TEST_&REQNUMB.txt`) and the request has ended successfully (`TEST TRANSFER ENDED`).

Troubleshooting

Connect:Express and Sterling Secure Proxy can provide trace which help to diagnose problems. Enabling trace usually decreases performance and must be left inactive during normal production operation.

Enable trace in C:X for UNIX

Trace can be enabled by setting `STRACE=1` in the `$TOM_DIR/config/sysin` configuration file. This requires however to restart the monitor. There is a dynamic way to change the trace activity. The command:

```
$TOM_DIR/config/ch_conf /STRACE=1
```

will activate the trace, while the command:

```
$TOM_DIR/config/ch_conf /STRACE=0
```

will stop the trace activity.

Trace will be written in different files in the `$TOM_DIR/strf` directory (refer to the **IBM® Sterling Connect:Express® for UNIX - User and Installation Guide** for the details). The more relevant trace files to diagnose SSL problems are the trace files `RTQQQNNNNN.pid` for outbound transfers and `RTQQQNNNNN.pid` for inbound transfers.

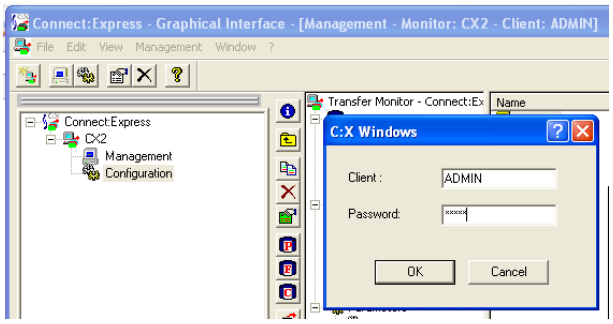
Enable trace in C:X for Microsoft Windows

There are 2 levels of trace in C:X for Microsoft Windows. Both can be activated using the CX Graphical Interface.

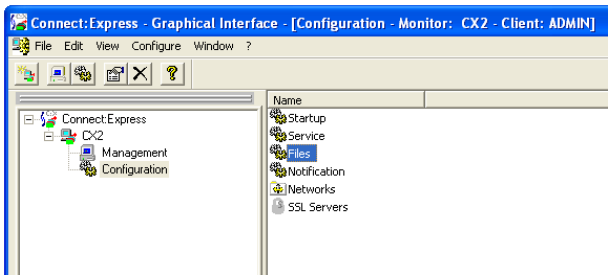
Enable PeSIT protocol trace

A first level of trace to help diagnose PeSIT protocol problems when C:X is used as a server or a client can be enabled.

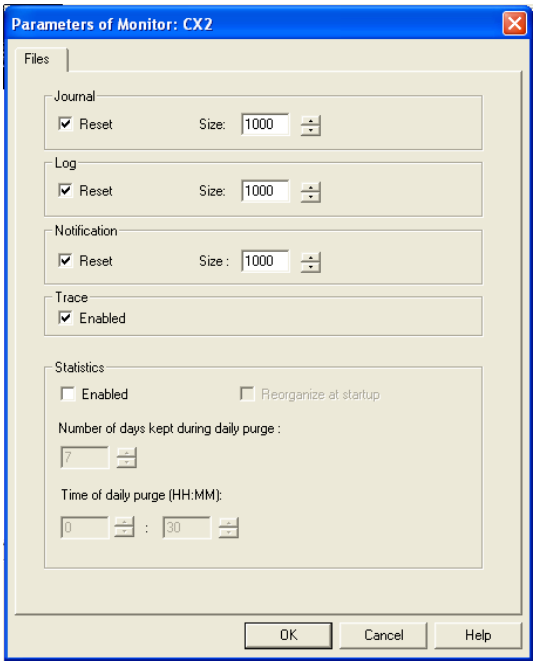
- In the left pane expand the **CX2** node and double-click on **Configuration**. Sign in using `ADMIN` for both the **Client** and the **Password**.



- In the right pane double-click on **Files**.



- Tick the **Enabled** check box in the **Trace** box and click the **OK** button.



- Click the **OK** button.

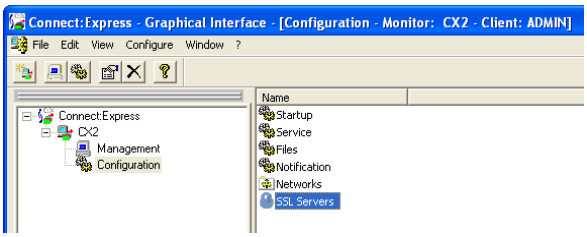
There is no need to restart the monitor. The trace will be written in the file `CYYYYDDDNNNNN.txt` where `YYYY` is the year, `DDD` is the day of the year (001 to 366) and `NNNNN` is the request number. Be aware that restarting the CX monitor during the same day will reset the request counter to 1 and therefore trace files could be overwritten without notice. Below is an example of the trace file `C201125400002.txt`.

```
17:49:57 - ANMINIT - TYPE= 01 LINK= 02
17:49:57 - P1CLI05A (entry) - STATE=0101
17:49:57 - ANMSENDDATA LG= 124
0000  32 32 30 2d 43 3a 45 20:66 6f 72 20 57 69 6e 64  220-C:E for Wind
0010  6f 77 73 20 4e 54 20 33:31 30 30 31 20 31 39 39  ows NT 31001 199
0020  38 2f 30 35 20 30 30 30:30 30 30 30 30 0d 0a 32  8/05 00000000..2
0030  32 30 2d 28 43 29 20 43:6f 70 79 72 69 67 68 74  20-(C) Copyright
0040  20 49 42 4d 20 43 6f 72:70 20 31 39 39 39 2d 32   IBM Corp 1999-2
0050  30 31 30 0d 0a 32 32 30:20 43 32 30 31 31 32 35  010..220 C201125
0060  34 30 30 30 32 32 30 31:31 2f 30 39 2f 31 31 20  40002201 1/09/11
0070  2d 20 31 37 3a 34 39 3a:35 37 0d 0a              - 17:49:57..

17:49:57 - sent LG= 124
17:49:57 - ANMRECEIVEDATA LINK= 2
17:49:57 - ANMRECEIVEDATA DEMANDE= 16000 DEJA=0
RECV= 15
0000  55 53 45 52 20 41 44 4d:49 4e 20 20 20 0d 0a     USER ADMIN   ..

17:49:57 - ANMSENDDATA LG= 10
0000  33 33 31 20 50 41 53 53:0d 0a                    331 PASS..

17:49:57 - sent LG= 10
17:49:57 - ANMRECEIVEDATA LINK= 2
17:49:57 - ANMRECEIVEDATA DEMANDE= 16000 DEJA=0
RECV= 23
0000  50 41 53 53 20 33 38 33:39 33 41 44 30 43 35 44  PASS 38393AD0C5D
0010  31 43 32 43 30 0d 0a                             1C2C0..|

17:49:57 - ANMSENDDATA LG= 6
0000  32 33 30 20 0d 0a                                230 ..

17:49:57 - sent LG= 6
17:49:57 - ANMRECEIVEDATA LINK= 2
17:49:57 - ANMRECEIVEDATA DEMANDE= 16000 DEJA=0
RECV= 27
0000  53 49 54 45 20 31 31 39:32 2e 31 36 38 2e 30 2e  SITE 1192.168.0.
0010  36 32 20 20 20 33 31 30:31 0d 0a                 62   3101..

17:49:57 - P1CLI05A (entry) - STATE=0107
17:49:57 - ANMSENDDATA LG= 39
0000  32 33 30 20 31 31 31 30:30 30 30 30 31 31 31  230 111000000111
0010  31 31 31 31 31 31 31 31:31 31 31 31 31 31 31  111111111111111
0020  31 31 31 31 31 0d 0a                             11111..

17:49:57 - sent LG= 39
17:49:57 - P1CLIX28 (entry) - STATE=0200
17:49:57 - ANMRECEIVEDATA LINK= 2
17:49:57 - ANMRECEIVEDATA DEMANDE= 16000 DEJA=0
RECV= 30
```

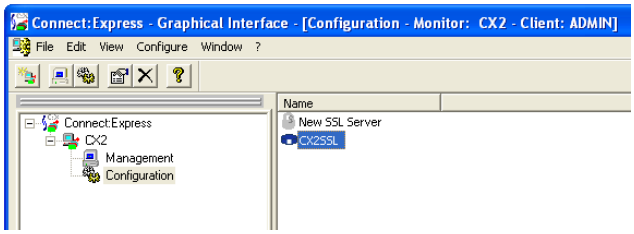
Enable SSL handshake trace for server

The second level of trace which can be activated is the SSL handshake trace. It can be very useful to diagnose problem during a SSL connection between a CX monitor trying to connect to SSP.

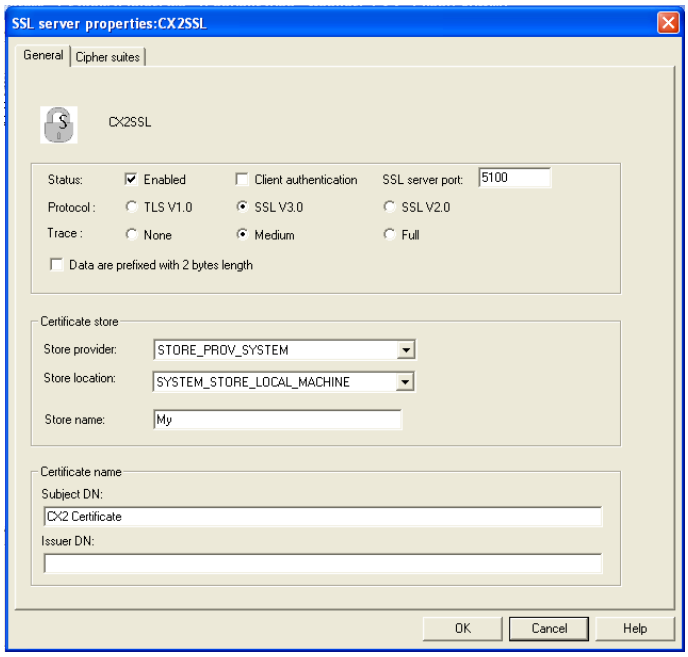
- In the right pane of the **Configuration-Monitor** window, double-click on **SSL Servers**.



- Double-click on **CX2SSL**.



- Select the **Trace** radio button **Medium** or **Full**.



- Restart the CX monitor.

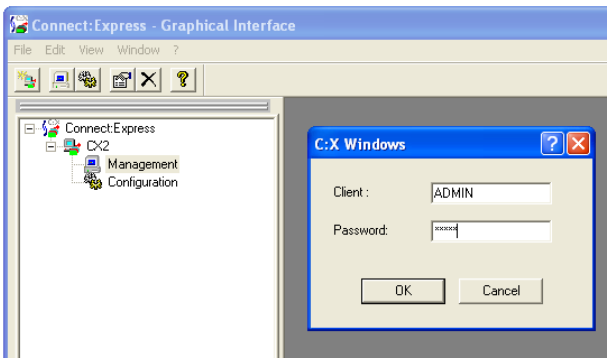
The trace of the SSL handshake when the monitor is used as a server will be written in the file `YYYYDDDNNNNN.txt` where `YYYY` is the year, `DDD` is the day of the year (001 to 366) and `NNNNN` is the request number. Be aware that restarting the CX monitor during the same day will reset the request counter to 1 and therefore trace wiles could be overwritten without notice. Below is an snapshot of the trace file `201125100002.txt`.

```
17:52:54 - ANMINIT - TYPE= 00 LINK= 04
17:52:54 - P1CAL05A - 0101
17:52:54 - CALLOUT - ADDR= HNAME=            ssp PORT=16200
17:52:54 - connect RC= -1
17:52:54 - connect RC= 0 - Adresse:   N° Port: 16200
SSL:HeapHandle = 150000
SSL:Opening Store. Provider = CERT_STORE_PROV_SYSTEM
SSL:Opening Store. Location = CERT_SYSTEM_STORE_LOCAL_MACHINE
SSL:Opening Store. Name = My
Certificate to find: CX2 Certificate
Certificates in: STORE_PROV_SYSTEM : SYSTEM_STORE_LOCAL_MACHINE : My
-------------------------------------------------------------------------------
Certificate subject: C=FR, L=Paris, O=IBM, OU=France Labs, CN=CX2 Certificate

==============================================================================

==============================================================================

CERTIFICATE FOUND (CERT_X500_NAME_STR | CERT_NAME_STR_NO_PLUS_FLAG): C=FR, L=Paris, O=IBM, OU=France Labs, CN=CX2 Certifica

==============================================================================

==============================================================================

Certificate to find: CX2 Certificate
Certificates in: STORE_PROV_SYSTEM : SYSTEM_STORE_LOCAL_MACHINE : Root
-------------------------------------------------------------------------------
Certificate subject: DC=com, DC=microsoft, CN=Microsoft Root Certificate Authority

Certificate subject: OU=Copyright (c) 1997 Microsoft Corp., OU=Microsoft Corporation, CN=Microsoft Root Authority

Certificate subject: C=FR, L=Paris, O=IBM, OU=France Labs, CN=SSP Certificate

Certificate subject: C=US, O=MSFT, CN=Microsoft Authenticode(tm) Root Authority

Certificate subject: O=Microsoft Trust Network, OU=Microsoft Corporation, OU=Microsoft Time Stamping Service Root, OU=Copyr

Certificate subject: O=VeriSign Trust Network, OU="VeriSign, Inc.", OU=VeriSign Time Stamping Service Root, OU="NO LIABILIT

Certificate subject: C=hk, O=C&W HKT SecureNet CA SGC Root

Certificate subject: C=FR, O=Certplus, CN=Class 3TS Primary CA

Certificate subject: C=MX, CN="Autoridad Certificadora del Colegio Nacional de Correduria Publica Mexicana, A.C.", O="Colec

Certificate subject: C=us, S=Utah, L=Salt Lake City, O=Digital Signature Trust Co., OU=United Parcel Service, CN=DST (UPS)

Certificate subject: C=FR, O=Certiposte, CN=Certiposte Classe A Personne
```

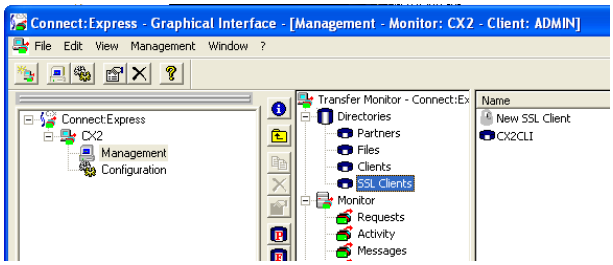
Enable SSL handshake trace for client

To help diagnose SSL problems when SSP tries to connect to a CX monitor, the trace can be enabled at the client level.

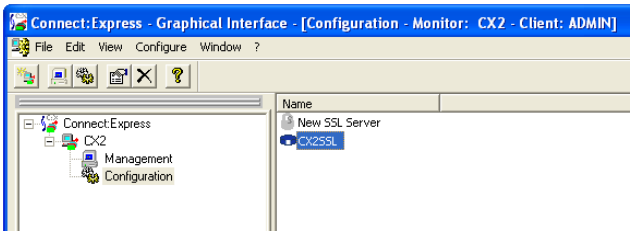
- Expand the **CX2** node in the left pane and double-click on **Management**. Sign in using `ADMIN` for both the **Client** and the **Password**.



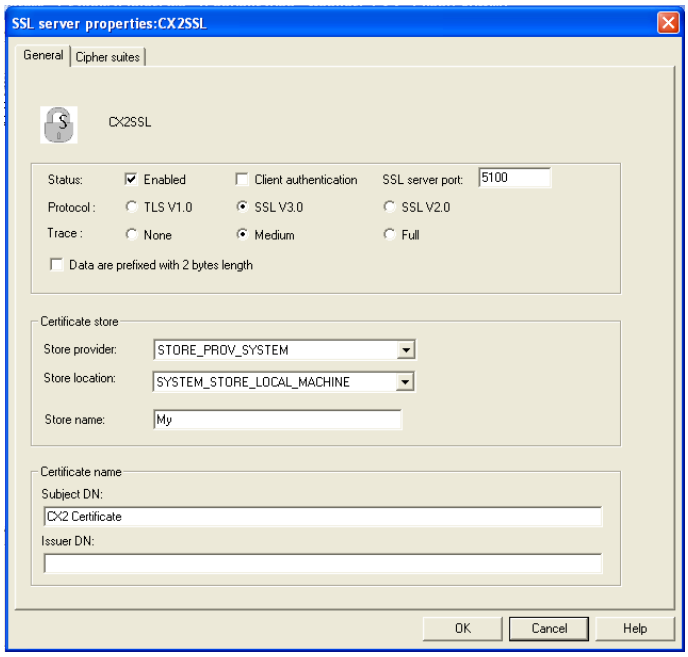
- In the middle pane select **Directories** > **SSL Clients** then double-click on **CX2CLI** in the right pane.



- Double-click on **CX2SSL**.



- Select the **Trace** radio button **Medium** or **Full**.



- Click the **OK** button. There is no need to restart the monitor.

The trace of the SSL handshake when the monitor is used as a client will also be written in the file `YYYYDDDNNNNN.txt`.

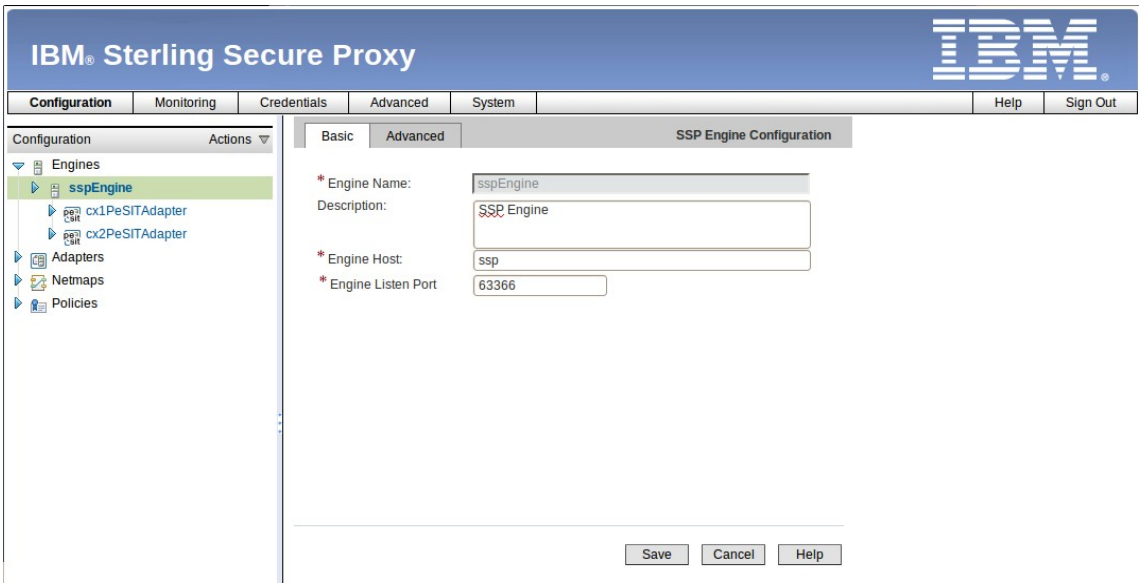
Enable trace in SSP

Sterling Secure Proxy has trace facilities which help to diagnose problems. There are many components in SSP for which the trace can be independently enabled (engines, perimeter server, adapters, etc).

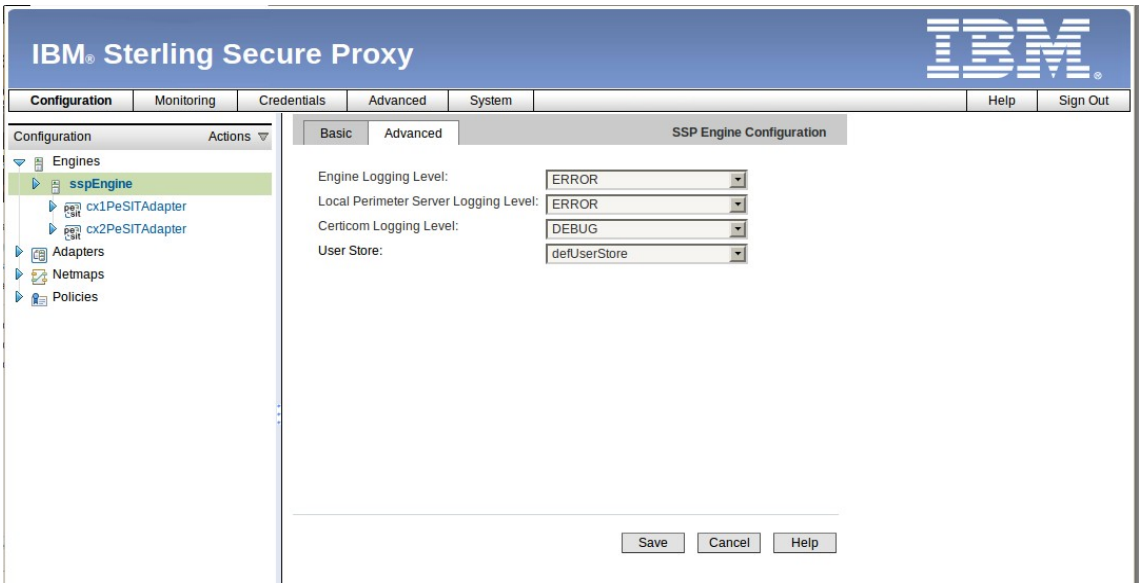
Enabling SSL or TLS trace

Enabling SSL or TLS trace can be very useful to diagnose handshake problems. Enabling trace is done through the SSPcm graphical interface.

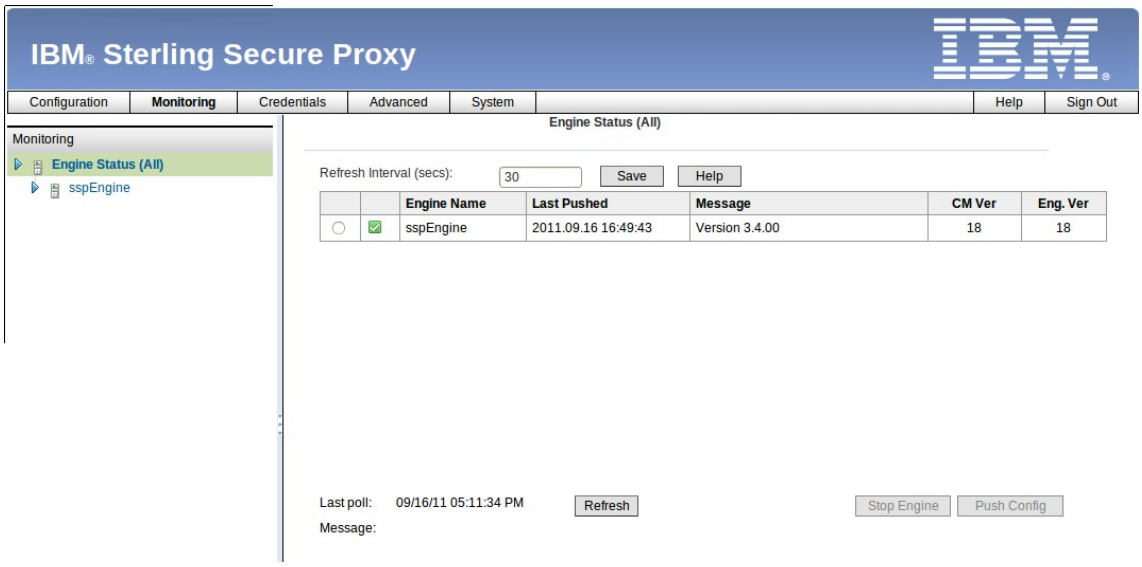
- Connect to the SSPcm graphical interface with administrator privileges (the section **Import the self-signed certificates in SSP** explains how to connect to the SSPcm graphical interface).
- On the left pane expand **Engines** and click **sspEngine**.



- select the **Advanced** tab and select `DEBUG` from the **Certicom Logging Level** drop down list.



- Click the **OK** button to save the new configuration.
- Every time you save the configuration, a version number is incremented. By default, SSP is checking if there is a new configuration in SSPcm every 30 seconds. You can check if the configuration version numbers are the same by selecting the **Monitoring** in the top banner and select **Engine Status** in the left pane.



- Once the trace is enabled you can try a file transfer from the CX1 partner to the CX2 partner. On the SSP host , navigate to the directory where SSP has been installed. Go into the `logs` directory. You should see a file named `certicom.log`. This file contains the trace of the SSL handshake.

```
16 sept. 2011 17:36:32,359 INFO  ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [com.certicom.tls.record.handshake.R{2}]  SSL2.0 RECORD CLIENT HELLO
>      Cipher Suites
>        TLS_RSA_WITH_AES_256_CBC_SHA
>        TLS_RSA_WITH_3DES_EDE_CBC_SHA
>        TLS_RSA_WITH_AES_128_CBC_SHA
>      Challenge
>        F8 62 B4 24 1A 80 15 4E   62 07 B0 42 73 40 96 77
>        3B 7A BC F1 D7 F9 5C 1C   40 16 6E 95 8B 2B A2 D4
>      ProtocolVersion
>        TLS1
>
16 sept. 2011 17:36:32,359 DEBUG ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [com.certicom.tls.interfaceimpl.TLSConnectionImpl{3}]Number of complete handsh
16 sept. 2011 17:36:32,359 DEBUG ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [com.certicom.tls.record.handshake.x{3}]Accepting client hello
16 sept. 2011 17:36:32,359 DEBUG ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [com.certicom.tls.interfaceimpl.i{3}]chooseServerAlias(RSA,null,null)
16 sept. 2011 17:36:32,359 DEBUG ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [com.certicom.tls.interfaceimpl.i{3}]getCertificateChain(RSA0)
16 sept. 2011 17:36:32,359 DEBUG ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [java.lang.Class{3}]isCertificateValid? RSA RSA null null null
16 sept. 2011 17:36:32,359 DEBUG ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [java.lang.Class{3}]valid
16 sept. 2011 17:36:32,359 DEBUG ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [com.certicom.tls.interfaceimpl.i{3}]chooseServerAlias done
16 sept. 2011 17:36:32,359 DEBUG ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [com.certicom.tls.DefaultTLSSessionDB{3}]Session DB cleanupDB
16 sept. 2011 17:36:32,359 DEBUG ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [com.certicom.tls.DefaultTLSSessionDB{3}]Before: [
>    [-1] 1316187110468 bed578cec566a02f28b5880eeb6c966d [1316184620562]
>    [1316187110468] 1316184620562 7a66fc04da9dce2f842982d0f6bc2a0f [1316184440187]
>    [1316184620562] 1316184440187 f94017cca296c849535439a179288527 [1316184080578]
>    [1316184440187] 1316184080578 33a7459d7f6e1435e00b9d089e06d779 [-1]
>  ]
16 sept. 2011 17:36:32,359 DEBUG ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [com.certicom.tls.DefaultTLSSessionDB{3}]Before: {f94017cca296c849535439a17928
16 sept. 2011 17:36:32,359 DEBUG ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [com.certicom.tls.DefaultTLSSessionDB{3}]maxDBSize=4096
16 sept. 2011 17:36:32,359 DEBUG ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [com.certicom.tls.DefaultTLSSessionDB{3}]ttlMillis=3600000
16 sept. 2011 17:36:32,359 DEBUG ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [com.certicom.tls.DefaultTLSSessionDB{3}]Session DB no change
16 sept. 2011 17:36:32,359 DEBUG ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [com.certicom.tls.record.handshake.x{3}]Retreived Session: null
16 sept. 2011 17:36:32,359 INFO  ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [com.certicom.tls.record.a{1}]C<--Local 22
16 sept. 2011 17:36:32,359 INFO  ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [com.certicom.tls.record.handshake.R{1}]  SERVER HELLO
>      SessionID
>        3E 77 9D 20 30 95 31 6E   0E C7 8E 30 8B 10 7D 76
>      Negotiated cipher suite
>        TLS_RSA_WITH_AES_256_CBC_SHA
>      Random Message
>        4E 73 6D 00 60 67 D8 D5   37 A4 20 54 6A DB D0 26
>        4B 21 73 5C 6D 06 95 99   B4 36 32 10 F1 59 0E A8
>      ProtocolVersion
>        SSL3
>      RecordVersion
>        SSL3
>
16 sept. 2011 17:36:32,359 DEBUG ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [com.certicom.tls.interfaceimpl.i{3}]chooseServerAlias(RSA,null,null)
16 sept. 2011 17:36:32,359 DEBUG ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [com.certicom.tls.interfaceimpl.i{3}]getCertificateChain(RSA0)
16 sept. 2011 17:36:32,359 DEBUG ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [java.lang.Class{3}]isCertificateValid? RSA RSA null null null
16 sept. 2011 17:36:32,359 DEBUG ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [java.lang.Class{3}]valid
16 sept. 2011 17:36:32,359 DEBUG ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [com.certicom.tls.interfaceimpl.i{3}]chooseServerAlias done
16 sept. 2011 17:36:32,359 DEBUG ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [com.certicom.tls.interfaceimpl.i{3}]getCertificateChain(RSA0)
16 sept. 2011 17:36:32,359 DEBUG ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [com.certicom.tls.interfaceimpl.i{3}]getPrivateKey(RSA0)
16 sept. 2011 17:36:32,359 INFO  ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [com.certicom.tls.record.a{1}]C<--Local 22
16 sept. 2011 17:36:32,375 DEBUG ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [com.certicom.tls.provider.ProviderManager{3}]
>  null
16 sept. 2011 17:36:32,375 DEBUG ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [com.certicom.tls.provider.ProviderManager{3}]
>  java.lang.reflect.InvocationTargetException
16 sept. 2011 17:36:32,375 INFO  ... sys.CERTICOM - protocol=pesit sessid=1316187392203_6 ... [com.certicom.tls.record.handshake.R{1}]  CERTIFICATE
>      Certificate[0]
>        Certificate:
>          Serial Number: 12284050851697992781
>          Version: 3
>          Type: X.509
>          Subject:
>            C=FR, L=Paris, O=IBM, OU=France Labs, CN=SSP Certificate
>          Issuer:
>            C=FR, L=Paris, O=IBM, OU=France Labs, CN=SSP Certificate
>          Validity:
>            Not Before:Wed Sep 07 16:17:57 CEST 2011
>            Not After :Thu Sep 06 16:17:57 CEST 2012
>          Subject Public Key Info:
>            Public Key Algorithm :RSA
>            RSA Public Key:
```

Some part of the file above have been truncated to make it more readable.

PeSIT trace in SSP

Below is a list of other components for which a trace can be enabled in SSP.

- **PeSIT adapters**: trace for PeSIT adapters is written in files of the form `secureproxy-pesitNetmap.<node_name>.log` in the logs directory. Trace gives interesting informations about the status of inbound and outbound sessions.
- **PeSIT Netmap nodes**: PeSIT trace for inbound nodes (there is no trace for outbound nodes). Trace files are written in the log directory in the form `pesit.<node_name>.<session_id>.trace`. The trace gives the detail of the messages exchanged between the PeSIT node and SSP.

References

- IBM® Sterling Connect:Express® for UNIX - User and Installation Guide: `CXUX15_UserGuide.pdf` (comes with the product).
- IBM® Sterling Connect:Express® for Microsoft Windows - User Guide: `CXWIN31_UserGuide_EN.pdf` (comes with the product).
- IBM Sterling Secure Proxy 3.4 Help: http://help.sterlingcommerce.com/SSP34/index.jsp (requires to create an account)
- OpenSSL for Windows: http://gnuwin32.sourceforge.net/packages/openssl.htm