OpenSC Installation Guide Windows 7

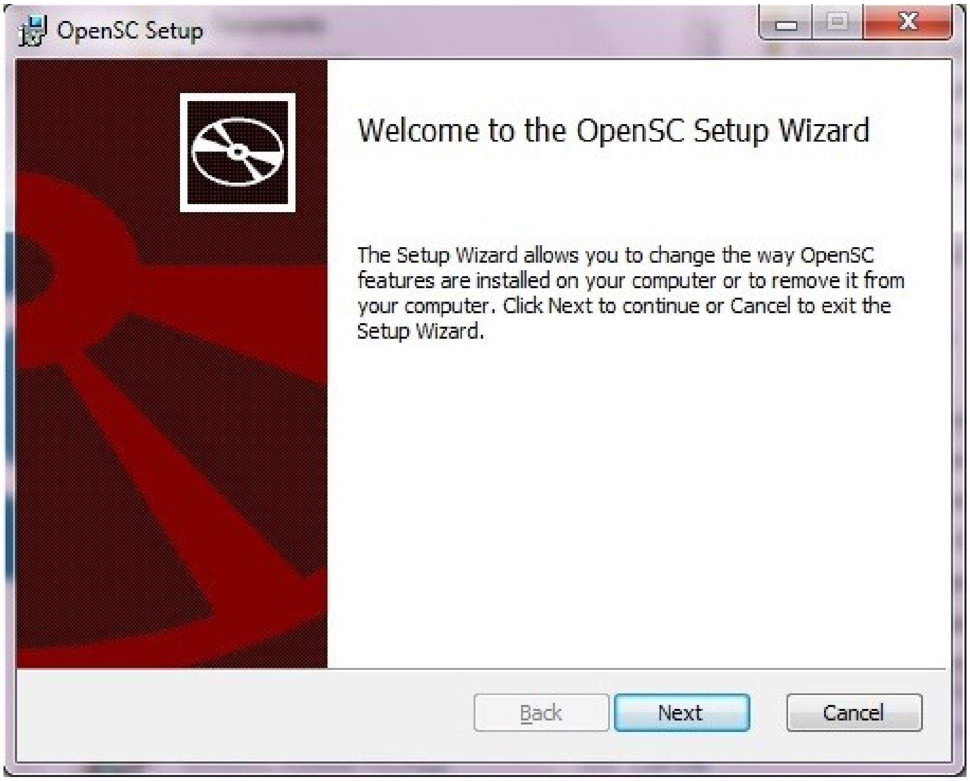
Table of Contents

Setting up OpenSC under Windows 7:

Download the installer from the following website, make sure to choose the 32 bit package for both 32-bit and 64-bit computers:

http://sourceforge.net/projects/opensc/files/OpenSC/opensc-0.12.2

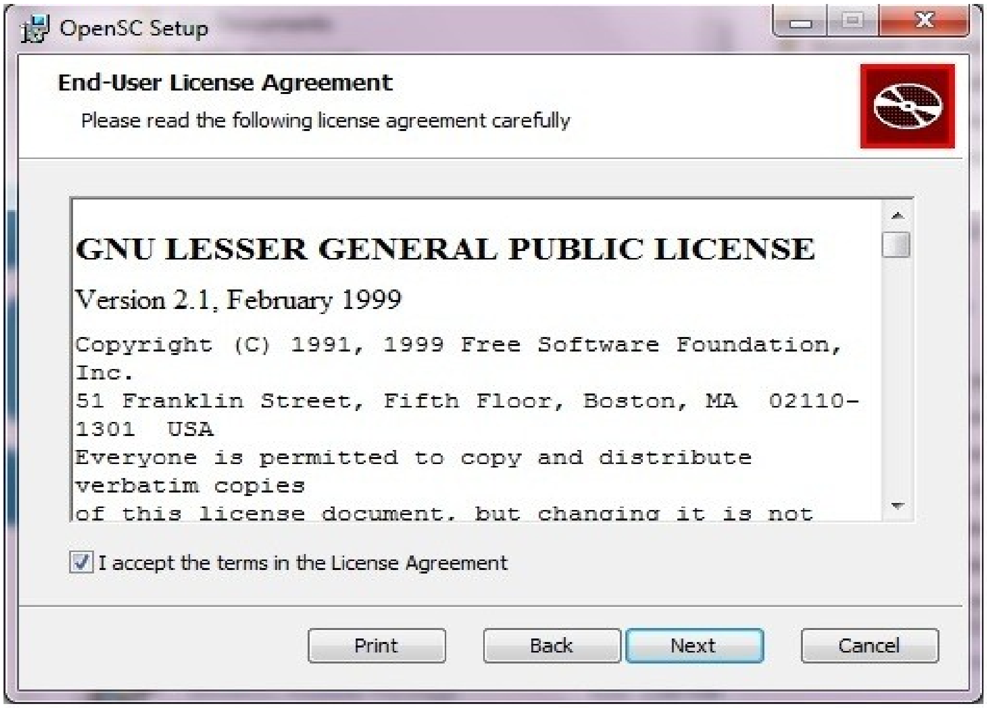
Install the downloaded package:



Welcome to the OpenSC Setup Wizard

The Setup Wizard allows you to change the way OpenSC features are installed on your computer or to remove it from your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Accept the End User License Agreement:



OpenSC Setup

End-User License Agreement

Please read the following license agreement carefully

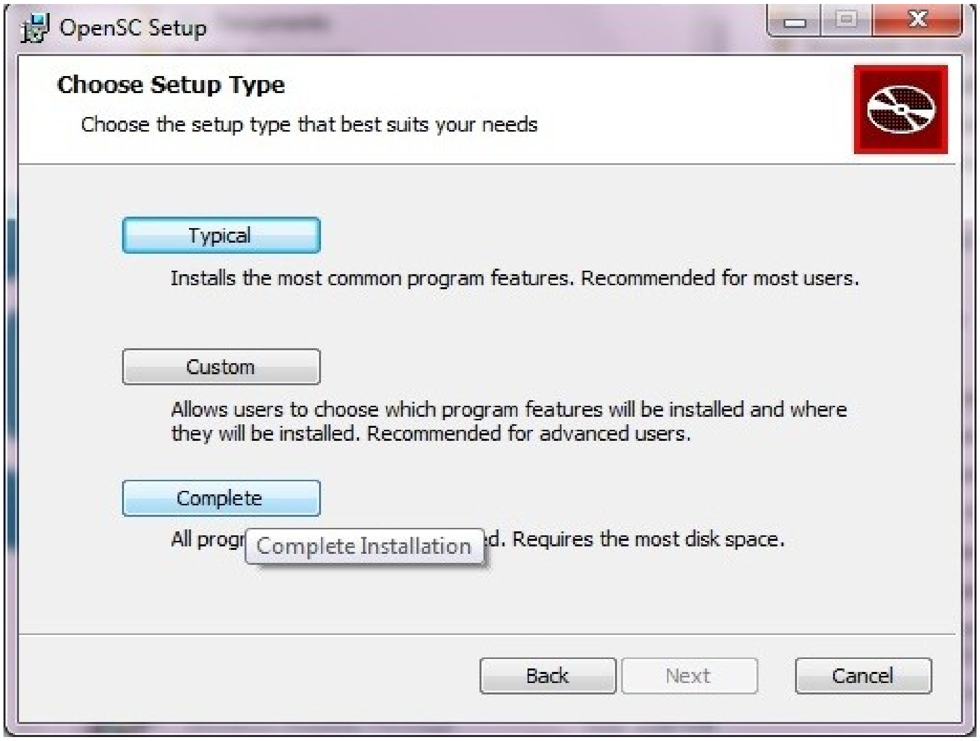
GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

```
Copyright (C) 1991, 1999 Free Software Foundation,
Inc.
51 Franklin Street, Fifth Floor, Boston, MA  02110-
1301  USA
Everyone is permitted to copy and distribute
verbatim copies
of this license document, but changing it is not
```

☑ I accept the terms in the License Agreement

Select “**Complete**” installation:



OpenSC Setup

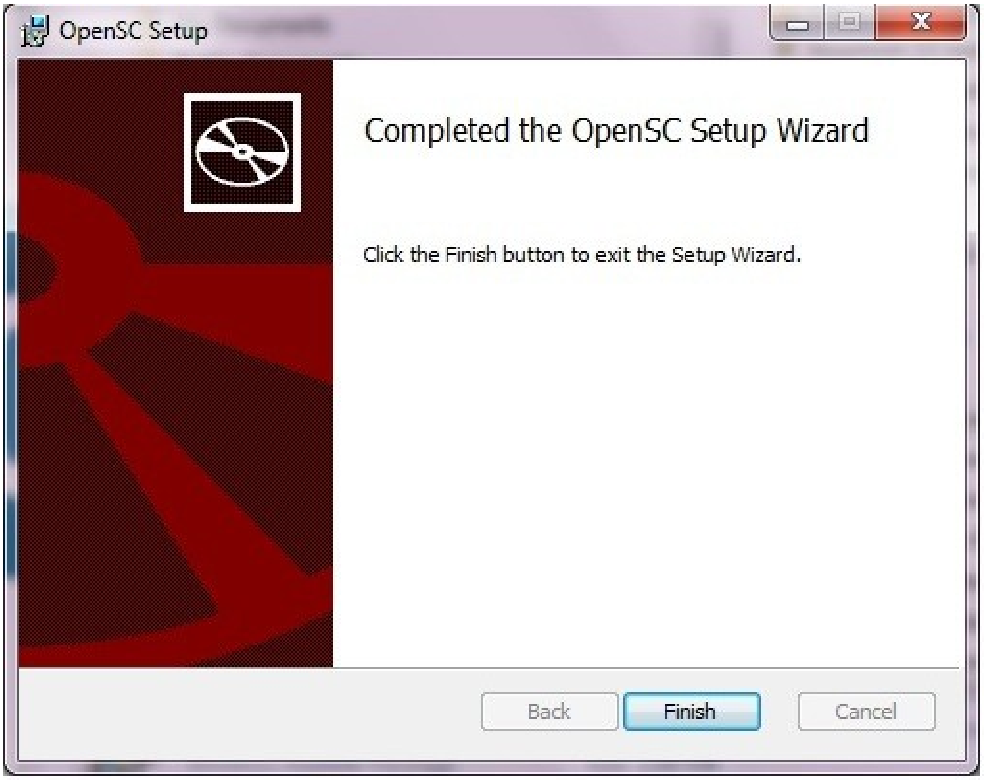
Choose Setup Type

Choose the setup type that best suits your needs

Installs the most common program features. Recommended for most users.

Allows users to choose which program features will be installed and where they will be installed. Recommended for advanced users.

All progr Complete Installation d. Requires the most disk space.

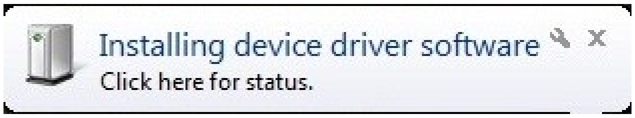


OpenSC Setup

Completed the OpenSC Setup Wizard

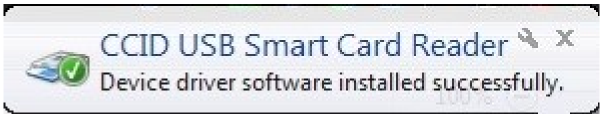
Click the Finish button to exit the Setup Wizard.

Attach the Card Reader to the computer, the drivers are configured automatically:



Installing device driver software

Click here for status.

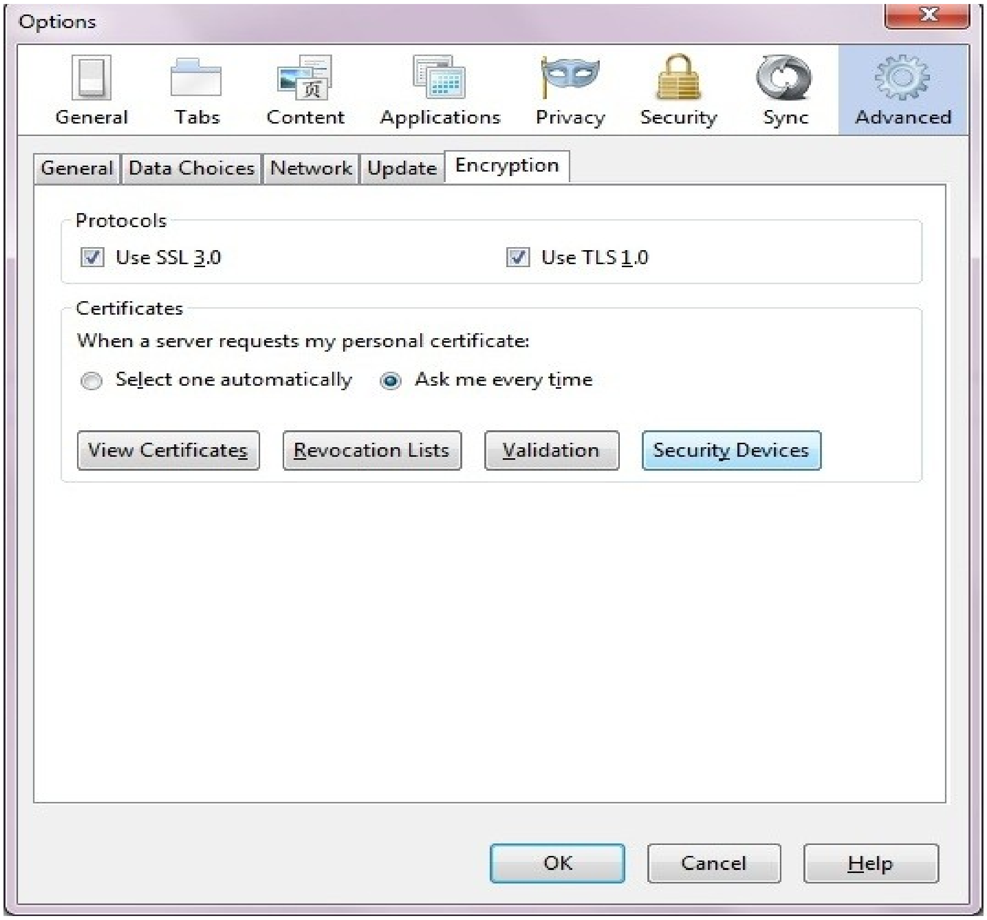


CCID USB Smart Card Reader

Device driver software installed successfully.

Firefox Setup

Select **Options** → **Advanced** → **Encryption** → **Security Devices.**



Protocols

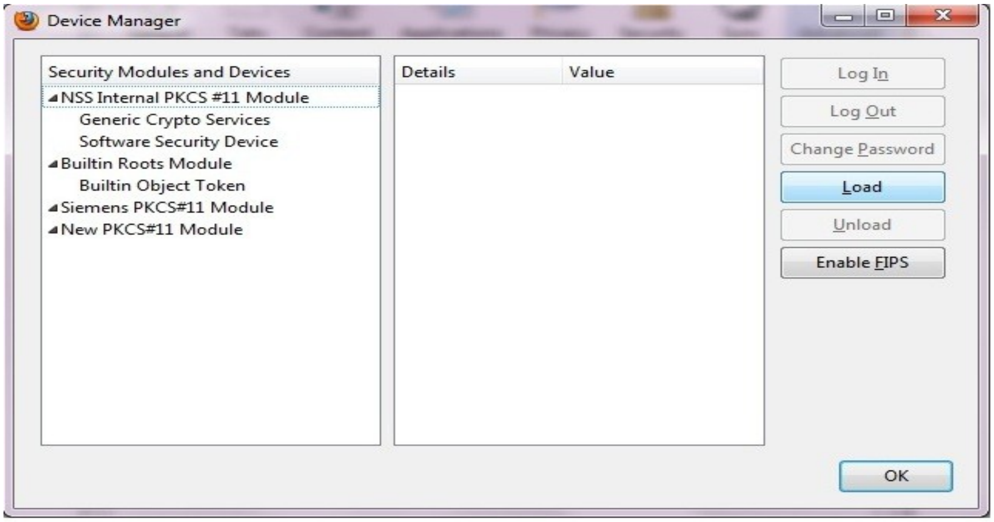
☑ Use SSL 3.0 ☑ Use TLS 1.0

Certificates

When a server requests my personal certificate:

○ Select one automatically ● Ask me every time

Add the module by clicking **"Load".**

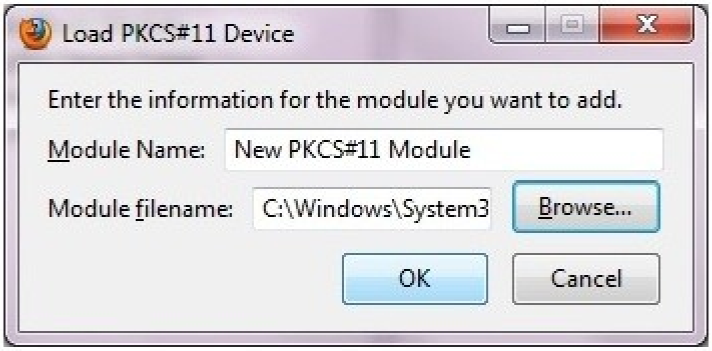


Security Modules and Devices

- NSS Internal PKCS #11 Module
  - Generic Crypto Services
  - Software Security Device
- Builtin Roots Module
  - Builtin Object Token
- Siemens PKCS#11 Module
- New PKCS#11 Module

Click **Browse**.

Navigate to **C:\Windows\system32\opensc-pkcs11.dll.** Click **OK.**

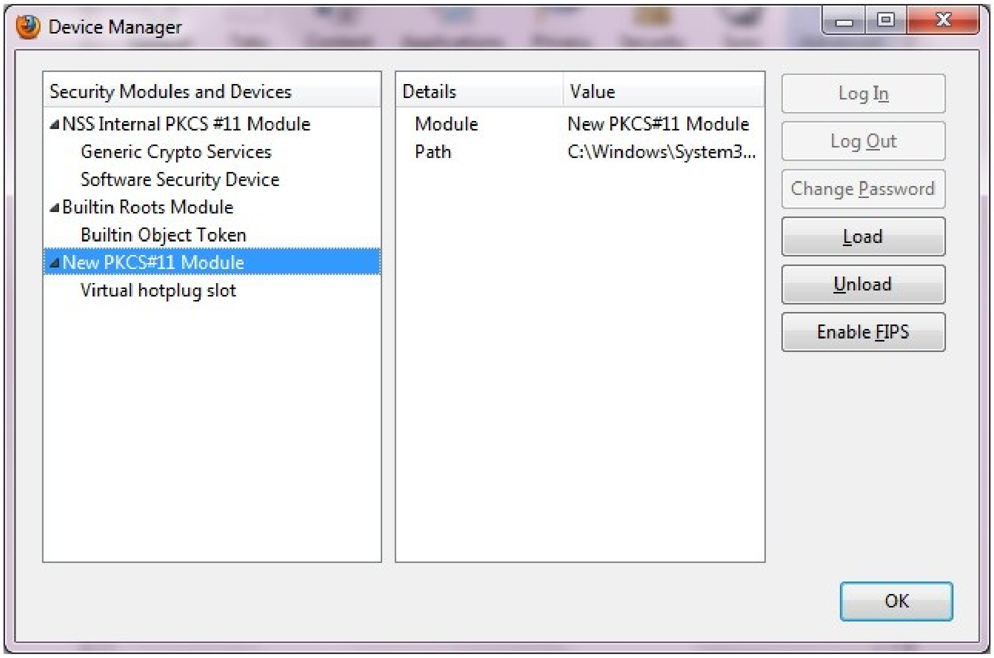


Enter the information for the module you want to add.

Module Name: New PKCS#11 Module

Module filename: C:\Windows\System3

Now the module is loaded.



Security Modules and Devices

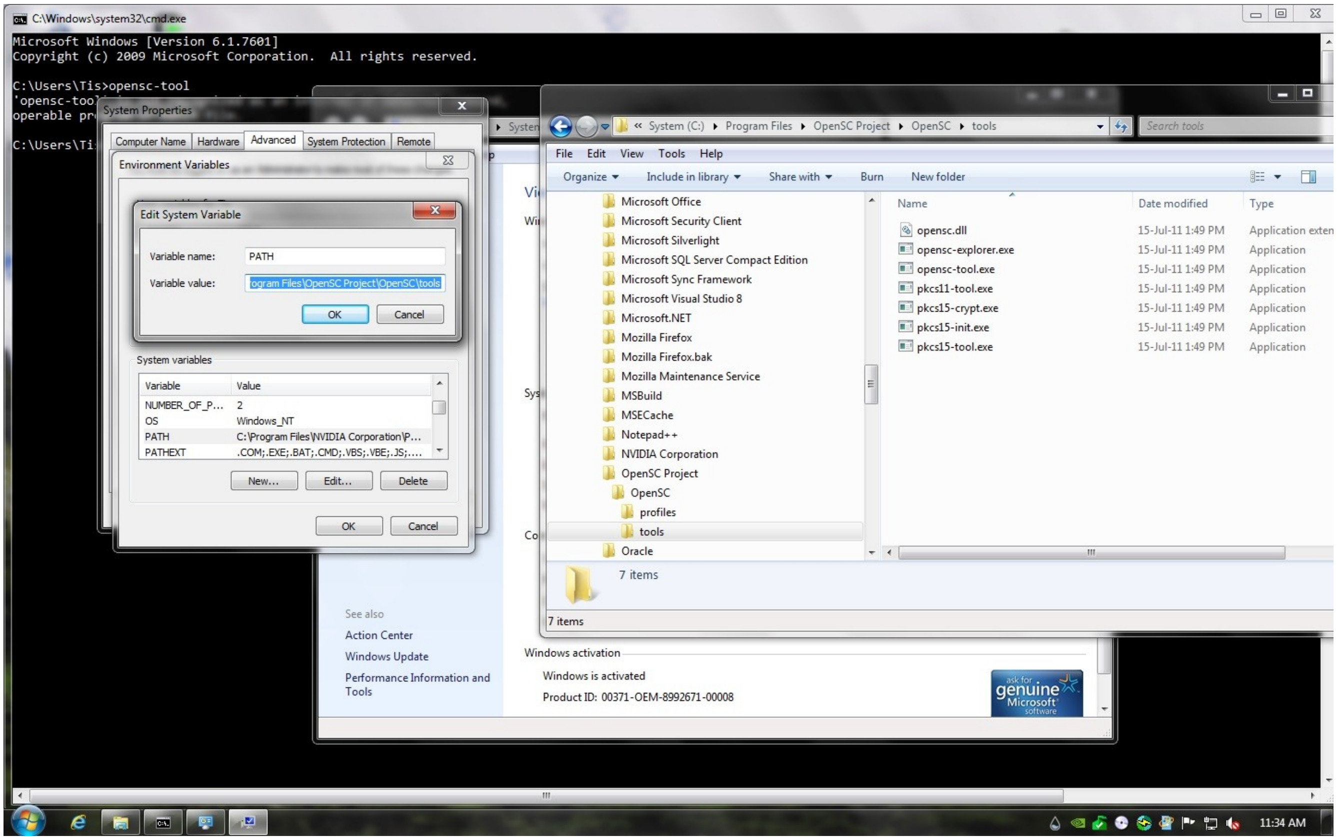
- NSS Internal PKCS #11 Module
  - Generic Crypto Services
  - Software Security Device
- Builtin Roots Module
  - Builtin Object Token
- New PKCS#11 Module
  - Virtual hotplug slot

Details	Value
Module	New PKCS#11 Module
Path	C:\Windows\System3...

Click **OK**. Restart Firefox.

Configuring the System Variable:

Right-click **My Computer**, and then click **Properties**
Click the **Advanced Tab**
Click the **Environment variables**
Click an existing variable, and then click **Edit** to change its value
Select **Path:**



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Tis>opensc-tool
'opensc-too
operable pr

C:\Users\Ti
```

Variable name: PATH
Variable value: rogram Files\OpenSC Project\OpenSC\tools

Variable	Value
NUMBER_OF_P...	2
OS	Windows_NT
PATH	C:\Program Files\NVIDIA Corporation\P...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....

« System (C:) ▸ Program Files ▸ OpenSC Project ▸ OpenSC ▸ tools

Name	Date modified	Type
opensc.dll	15-Jul-11 1:49 PM	Application exten
opensc-explorer.exe	15-Jul-11 1:49 PM	Application
opensc-tool.exe	15-Jul-11 1:49 PM	Application
pkcs11-tool.exe	15-Jul-11 1:49 PM	Application
pkcs15-crypt.exe	15-Jul-11 1:49 PM	Application
pkcs15-init.exe	15-Jul-11 1:49 PM	Application
pkcs15-tool.exe	15-Jul-11 1:49 PM	Application

7 items

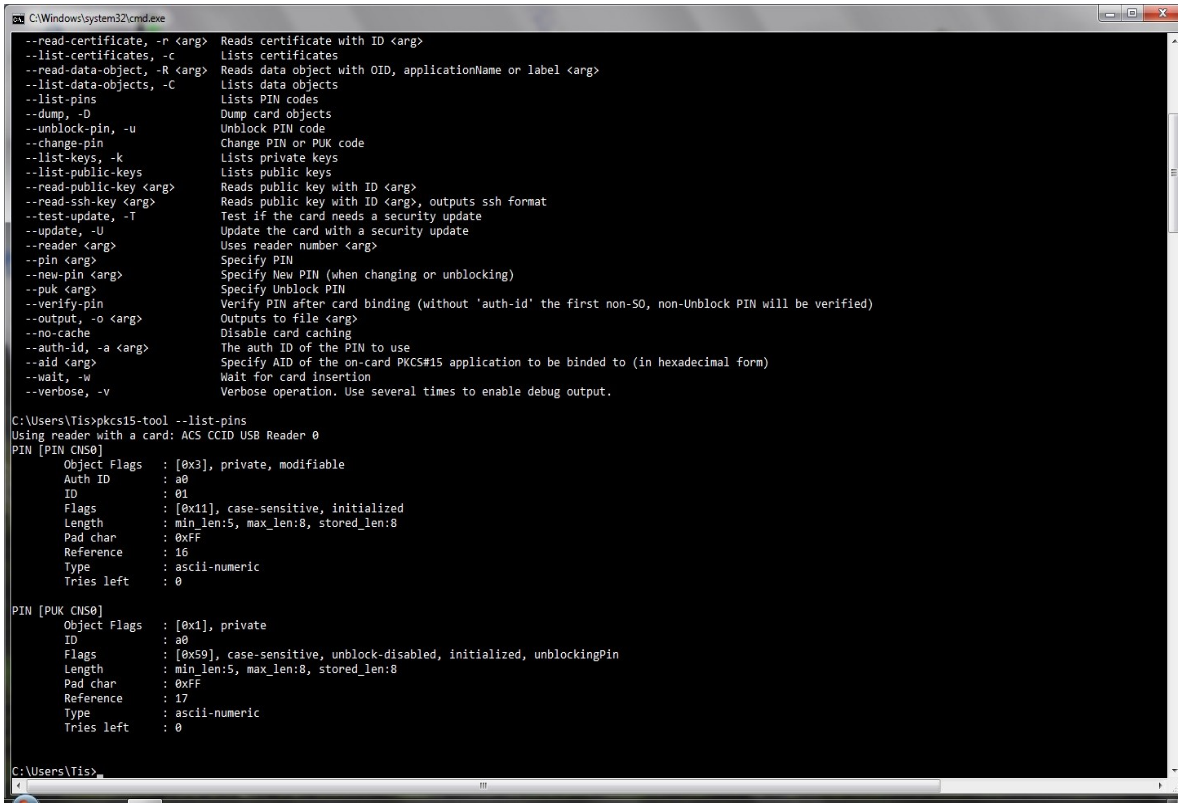
7 items

Windows is activated
Product ID: 00371-OEM-8992671-00008

Add the path to the **tools** folder in the OpenSC installation directory. This is found in **C:\Programs.**

The OpenSC tools can now be accessed from the command prompt.

Run a command prompt, the tools should be configured globally now:



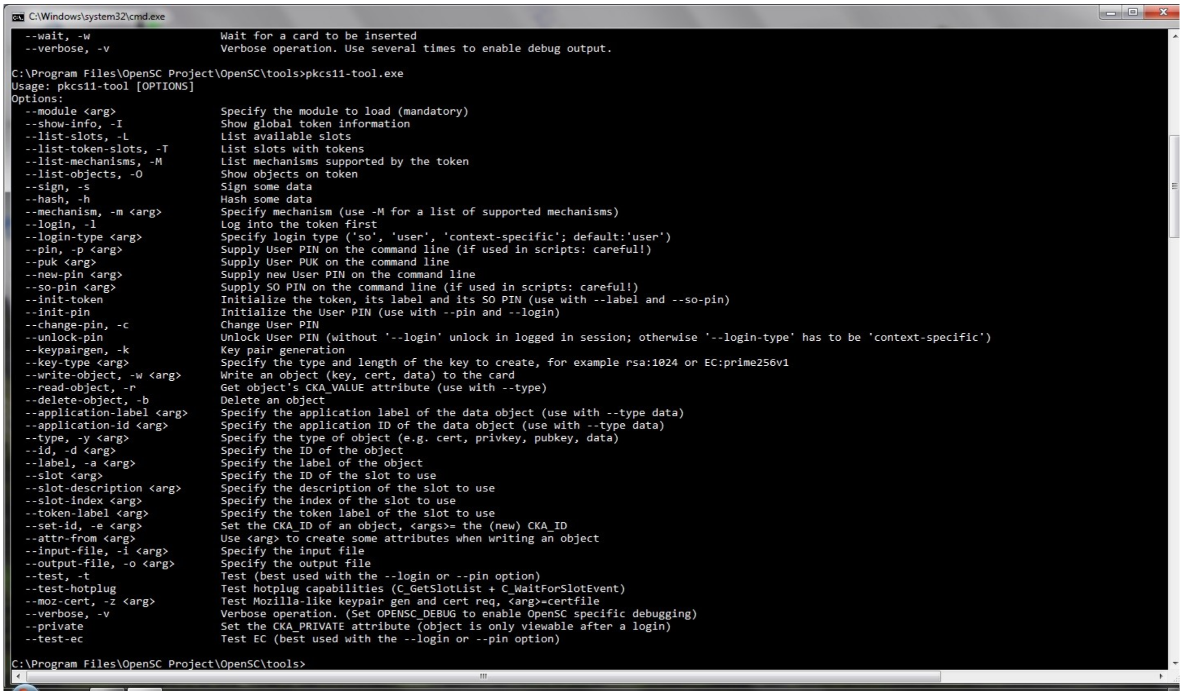
```
  --read-certificate, -r <arg>  Reads certificate with ID <arg>
  --list-certificates, -c       Lists certificates
  --read-data-object, -R <arg>  Reads data object with OID, applicationName or label <arg>
  --list-data-objects, -C       Lists data objects
  --list-pins                   Lists PIN codes
  --dump, -D                    Dump card objects
  --unblock-pin, -u             Unblock PIN code
  --change-pin                  Change PIN or PUK code
  --list-keys, -k               Lists private keys
  --list-public-keys            Lists public keys
  --read-public-key <arg>       Reads public key with ID <arg>
  --read-ssh-key <arg>          Reads public key with ID <arg>, outputs ssh format
  --test-update, -T             Test if the card needs a security update
  --update, -U                  Update the card with a security update
  --reader <arg>                Uses reader number <arg>
  --pin <arg>                   Specify PIN
  --new-pin <arg>               Specify New PIN (when changing or unblocking)
  --puk <arg>                   Specify Unblock PIN
  --verify-pin                  Verify PIN after card binding (without 'auth-id' the first non-SO, non-Unblock PIN will be verified)
  --output, -o <arg>            Outputs to file <arg>
  --no-cache                    Disable card caching
  --auth-id, -a <arg>           The auth ID of the PIN to use
  --aid <arg>                   Specify AID of the on-card PKCS#15 application to be binded to (in hexadecimal form)
  --wait, -w                    Wait for card insertion
  --verbose, -v                 Verbose operation. Use several times to enable debug output.

C:\Users\Tis>pkcs15-tool --list-pins
Using reader with a card: ACS CCID USB Reader 0
PIN [PIN CNS0]
        Object Flags   : [0x3], private, modifiable
        Auth ID        : a0
        ID             : 01
        Flags          : [0x11], case-sensitive, initialized
        Length         : min_len:5, max_len:8, stored_len:8
        Pad char       : 0xFF
        Reference      : 16
        Type           : ascii-numeric
        Tries left     : 0

PIN [PUK CNS0]
        Object Flags   : [0x1], private
        ID             : a0
        Flags          : [0x59], case-sensitive, unblock-disabled, initialized, unblockingPin
        Length         : min_len:5, max_len:8, stored_len:8
        Pad char       : 0xFF
        Reference      : 17
        Type           : ascii-numeric
        Tries left     : 0


C:\Users\Tis>_
```

The PIN can also be changed using the OpenSC Tools: *pkcs15-tool --change-pin*



```
  --wait, -w                    Wait for a card to be inserted
  --verbose, -v                 Verbose operation. Use several times to enable debug output.

C:\Program Files\OpenSC Project\OpenSC\tools>pkcs11-tool.exe
Usage: pkcs11-tool [OPTIONS]
Options:
  --module <arg>                Specify the module to load (mandatory)
  --show-info, -I               Show global token information
  --list-slots, -L              List available slots
  --list-token-slots, -T        List slots with tokens
  --list-mechanisms, -M         List mechanisms supported by the token
  --list-objects, -O            Show objects on token
  --sign, -s                    Sign some data
  --hash, -h                    Hash some data
  --mechanism, -m <arg>         Specify mechanism (use -M for a list of supported mechanisms)
  --login, -l                   Log into the token first
  --login-type <arg>            Specify login type ('so', 'user', 'context-specific'; default:'user')
  --pin, -p <arg>               Supply User PIN on the command line (if used in scripts: careful!)
  --puk <arg>                   Supply User PUK on the command line
  --new-pin <arg>               Supply new User PIN on the command line
  --so-pin <arg>                Supply SO PIN on the command line (if used in scripts: careful!)
  --init-token                  Initialize the token, its label and its SO PIN (use with --label and --so-pin)
  --init-pin                    Initialize the User PIN (use with --pin and --login)
  --change-pin, -c              Change User PIN
  --unlock-pin                  Unlock User PIN (without '--login' unlock in logged in session; otherwise '--login-type' has to be 'context-specific')
  --keypairgen, -k              Key pair generation
  --key-type <arg>              Specify the type and length of the key to create, for example rsa:1024 or EC:prime256v1
  --write-object, -w <arg>      Write an object (key, cert, data) to the card
  --read-object, -r             Get object's CKA_VALUE attribute (use with --type)
  --delete-object, -b           Delete an object
  --application-label <arg>     Specify the application label of the data object (use with --type data)
  --application-id <arg>        Specify the application ID of the data object (use with --type data)
  --type, -y <arg>              Specify the type of object (e.g. cert, privkey, pubkey, data)
  --id, -d <arg>                Specify the ID of the object
  --label, -a <arg>             Specify the label of the object
  --slot <arg>                  Specify the ID of the slot to use
  --slot-description <arg>      Specify the description of the slot to use
  --slot-index <arg>            Specify the index of the slot to use
  --token-label <arg>           Specify the token label of the slot to use
  --set-id, -e <arg>            Set the CKA_ID of an object, <args>= the (new) CKA_ID
  --attr-from <arg>             Use <arg> to create some attributes when writing an object
  --input-file, -i <arg>        Specify the input file
  --output-file, -o <arg>       Specify the output file
  --test, -t                    Test (best used with the --login or --pin option)
  --test-hotplug                Test hotplug capabilities (C_GetSlotList + C_WaitForSlotEvent)
  --moz-cert, -z <arg>          Test Mozilla-like keypair gen and cert req, <arg>=certfile
  --verbose, -v                 Verbose operation. (Set OPENSC_DEBUG to enable OpenSC specific debugging)
  --private                     Set the CKA_PRIVATE attribute (object is only viewable after a login)
  --test-ec                     Test EC (best used with the --login or --pin option)

C:\Program Files\OpenSC Project\OpenSC\tools>
```