|  |
| --- |
| [Type the company name] |
| SmartCardDriver Reference |
| [Type the document subtitle] |

|  |
| --- |
| [Type the author name]  [Pick the date] |

Table of Contents

[Press ALT+A and F9 to update the TOC and other fields]

SmartCardDriver Reference

Namespaces

[SmartCardDriver](#topic_0000000000000000)

SmartCardDriver Namespace

Κλάσεις

[SmartCardMonitor](#topic_0000000000000001)

SmartCardMonitor Class

Η κλάση IAltSecurity υλοποιεί το interface IAltSecurity

|  |  |
| --- | --- |
| C# |  |
| public class SmartCardMonitor : **IAltSecurity** | |

Απαιτήσεις

**Namespace:**[SmartCardDriver](#topic_0000000000000000)

**Assembly:** SmartCardDriver (in )

Μέθοδοι

[AttachToAllEvents](#topic_000000000000000B), [Authenticate](#topic_000000000000000F), [CardInserted](#topic_000000000000000D), [Connect](#topic_0000000000000008), [Disconnect](#topic_0000000000000009), [DisplayEvent](#topic_000000000000000C), [GetReaderNames](#topic_0000000000000012), [HashData](#topic_0000000000000017), [MonitorException](#topic_0000000000000011), [NoReaderFound](#topic_0000000000000013), [ShowUserInfo](#topic_000000000000000A), [StatusChanged](#topic_0000000000000010), [TrustDevice](#topic_000000000000000E), [UnTrustDevice](#topic_0000000000000016), [isTrusted](#topic_0000000000000014)

Πεδία

[TrainMode](#topic_0000000000000006), [TrustedDevice](#topic_0000000000000007), [\_contextFactory](#topic_0000000000000002), [\_isOk](#topic_0000000000000003), [\_isTrusted](#topic_0000000000000004), [\_monitor](#topic_0000000000000005)

SmartCardMonitor.AttachToAllEvents Method

Monitor all events for smartcards. (CardInserted,CardRemoved,Initialized,StatusChanged,MonitorException)

|  |  |
| --- | --- |
| C# |  |
| private [void](http://msdn.microsoft.com/en-us/library/system.void.aspx) AttachToAllEvents(  **ISCardMonitor** *monitor* ) | |

Παράμετροι

monitor

|  |
| --- |
| The ISCardMonitor monitor to bind to for events |

Πηγαίος κώδικας

|  |
| --- |
| private void AttachToAllEvents(ISCardMonitor monitor)  {  // Point the callback function(s) to the anonymous & static defined methods below.  monitor.CardInserted += (sender, args) => CardInserted(args);  monitor.CardRemoved += (sender, args) => DisplayEvent("CardRemoved", args);  monitor.Initialized += (sender, args) => DisplayEvent("Initialized", args);  monitor.StatusChanged += StatusChanged;  monitor.MonitorException += MonitorException;  } |

Δείτε επίσης

Εφαρμόζεται σε: [SmartCardMonitor](#topic_0000000000000001)

SmartCardMonitor.Authenticate Method

Read the smart card and authenticate it using EMVCO rules

|  |  |
| --- | --- |
| C# |  |
| private **AltsInfo** Authenticate(  [string](http://msdn.microsoft.com/en-us/library/system.string.aspx) *readerName* ) | |

Παράμετροι

readerName

|  |
| --- |
|  |

Επιστρέφει

AltsInfo Authenticated True or false

Πηγαίος κώδικας

|  |
| --- |
| private AltsInfo Authenticate(string readerName)  {  AltsInfo ok = new AltsInfo();  try  {  SmartCard sa = new SmartCard(readerName);  var res = sa.GetAIDs(true);  SmartApplication sapp = sa.Applications[0];  sapp.SelectApplication();  sapp.GetProcessingOptions();  sapp.ReadRecords();      OfflineAuth oa = new OfflineAuth(sapp);    string cardNumber = sapp.GetTagValue(EmvConstants.ResponceType.ReaderRecord, "5A");  string expdate = sapp.GetTagValue(EmvConstants.ResponceType.ReaderRecord, "5F25");  ///Keep the hash data of cardnumber and expiryDate  string token = HashData(cardNumber + expdate, oa.ICC\_KEY\_HASH);    ok.Authenticated = !string.IsNullOrEmpty(token);  ok.AuthDeviceId = token;  }  catch (Exception ex)  {  ok.Authenticated = false;  ok.AuthDeviceId = ex.Message;  }  return ok;  } |

Δείτε επίσης

Εφαρμόζεται σε: [SmartCardMonitor](#topic_0000000000000001)

SmartCardMonitor.CardInserted Method

Card inserted event.

If TrainMode is off, starts reading a card and initiates the authentication with the card

If TrainMode is on, starts to perfom the pairing with the card

|  |  |
| --- | --- |
| C# |  |
| private [void](http://msdn.microsoft.com/en-us/library/system.void.aspx) CardInserted(  **CardStatusEventArgs** *info* ) | |

Παράμετροι

info

|  |
| --- |
| The info send by the event |

Πηγαίος κώδικας

|  |
| --- |
| private void CardInserted( CardStatusEventArgs info)  {  if (!TrainMode)  {  var res = Authenticate(info.ReaderName);  if (isTrusted(res))  {    Console.Beep(6200, 50);  Console.Beep(6200, 50);  Console.Beep(6200, 50);  Console.Beep(6200, 50);  Console.Beep(6200, 50);  Console.Beep(6200, 50);  \_isOk(res);    }    else  {  Console.Beep(2200, 200);  Console.Beep(1200, 200);  \_isOk(new AltsInfo { Authenticated = false, AuthDeviceId = "Not Trusted" });  }    }    else  {    Console.Out.WriteLine("trainMode On");  TrustDevice(info.ReaderName);  TrainMode = false;  }  } |

Δείτε επίσης

Εφαρμόζεται σε: [SmartCardMonitor](#topic_0000000000000001)

SmartCardMonitor.Connect Method

Start comms with Client

|  |  |
| --- | --- |
| C# |  |
| public [void](http://msdn.microsoft.com/en-us/library/system.void.aspx) Connect(  **Authenticated** *isOk* ) | |

Παράμετροι

isOk

|  |
| --- |
| Callback to use when a card is read |

Εφαρμόζει

[IAltSecurity.Connect](#)

Πηγαίος κώδικας

|  |
| --- |
| public async void Connect(CallBacks.Authenticated isOk)  {  \_isOk = isOk;  // Retrieve the names of all installed readers.  var readerNames = GetReaderNames();    if (NoReaderFound(readerNames))  {  Console.WriteLine("There are currently no readers installed.");  return ;  }    // Create smartcard monitor using a context factory.   // The context will be automatically released after monitor.Dispose()  var monitorFactory = MonitorFactory.Instance;  \_monitor = monitorFactory.Create(SCardScope.System);  AttachToAllEvents(\_monitor); // Remember to detach, if you use this in production!  ShowUserInfo(readerNames);  \_monitor.Start(readerNames);    await Task.Delay(3000);    } |

Δείτε επίσης

Εφαρμόζεται σε: [SmartCardMonitor](#topic_0000000000000001)

SmartCardMonitor.Disconnect Method

Stops all comms

|  |  |
| --- | --- |
| C# |  |
| public [void](http://msdn.microsoft.com/en-us/library/system.void.aspx) Disconnect() | |

Εφαρμόζει

[IAltSecurity.Disconnect](#)

Πηγαίος κώδικας

|  |
| --- |
| public void Disconnect()  {  if (\_monitor == null)  return;  \_monitor.Cancel();  \_monitor.Dispose();  \_monitor = null;  } |

Δείτε επίσης

Εφαρμόζεται σε: [SmartCardMonitor](#topic_0000000000000001)

SmartCardMonitor.DisplayEvent Method

Display info when an event occurs on a reader

|  |  |
| --- | --- |
| C# |  |
| private static [void](http://msdn.microsoft.com/en-us/library/system.void.aspx) DisplayEvent(  [string](http://msdn.microsoft.com/en-us/library/system.string.aspx) *eventName*,  **CardStatusEventArgs** *unknown* ) | |

Παράμετροι

eventName

|  |
| --- |
|  |

unknown

|  |
| --- |
|  |

Πηγαίος κώδικας

|  |
| --- |
| private static void DisplayEvent(string eventName, CardStatusEventArgs unknown)  {  Console.WriteLine(">> {0} Event for reader: {1}", eventName, unknown.ReaderName);  Console.WriteLine("ATR: {0}", BitConverter.ToString(unknown.Atr ?? new byte[0]));  Console.WriteLine("State: {0}\n", unknown.State);  } |

Δείτε επίσης

Εφαρμόζεται σε: [SmartCardMonitor](#topic_0000000000000001)

SmartCardMonitor.GetReaderNames Method

Gets installed smart card reader names

|  |  |
| --- | --- |
| C# |  |
| private [string](http://msdn.microsoft.com/en-us/library/system.string.aspx)[] GetReaderNames() | |

Επιστρέφει

Array of reader names installed

Πηγαίος κώδικας

|  |
| --- |
| private string[] GetReaderNames()  {  using (var context = \_contextFactory.Establish(SCardScope.System))  {  return context.GetReaders();  }  } |

Δείτε επίσης

Εφαρμόζεται σε: [SmartCardMonitor](#topic_0000000000000001)

SmartCardMonitor.HashData Method

Creates a hash value to store as the TrustedDevice token

|  |  |
| --- | --- |
| C# |  |
| private static [string](http://msdn.microsoft.com/en-us/library/system.string.aspx) HashData(  [string](http://msdn.microsoft.com/en-us/library/system.string.aspx) *dataToHash*,  [byte](http://msdn.microsoft.com/en-us/library/system.byte.aspx)[] *saltValue* ) | |

Παράμετροι

dataToHash

|  |
| --- |
|  |

saltValue

|  |
| --- |
|  |

Επιστρέφει

Hash value of the cardnumber,expiry date and saltValue (ICC publicKey hash is used)

Πηγαίος κώδικας

|  |
| --- |
| private static string HashData(string dataToHash, byte[] saltValue)  {  var hashAlg = System.Security.Cryptography.HashAlgorithm.Create("SHA256");  List<byte> inputdata = new List<byte>();  inputdata.AddRange(StringTools.HexStringToByteArray(dataToHash));  inputdata.AddRange(saltValue);  string finaldata = StringTools.ByteArrayToHexString(saltValue);  finaldata += StringTools.ByteArrayToHexString(hashAlg.ComputeHash(inputdata.ToArray()));  return finaldata;  } |

Δείτε επίσης

Εφαρμόζεται σε: [SmartCardMonitor](#topic_0000000000000001)

SmartCardMonitor.MonitorException Method

Logs Exceptions to Console

|  |  |
| --- | --- |
| C# |  |
| private static [void](http://msdn.microsoft.com/en-us/library/system.void.aspx) MonitorException(  [object](http://msdn.microsoft.com/en-us/library/system.object.aspx) *sender*,  **PCSCException** *ex* ) | |

Παράμετροι

sender

|  |
| --- |
|  |

ex

|  |
| --- |
|  |

Πηγαίος κώδικας

|  |
| --- |
| private static void MonitorException(object sender, PCSCException ex)  {  Console.WriteLine("Monitor exited due an error:");  Console.WriteLine(SCardHelper.StringifyError(ex.SCardError));  } |

Δείτε επίσης

Εφαρμόζεται σε: [SmartCardMonitor](#topic_0000000000000001)

SmartCardMonitor.NoReaderFound Method

Check if reader is found

|  |  |
| --- | --- |
| C# |  |
| private static [bool](http://msdn.microsoft.com/en-us/library/system.boolean.aspx) NoReaderFound(  [ICollection](http://msdn.microsoft.com/en-us/library/92t2ye13.aspx)<[string](http://msdn.microsoft.com/en-us/library/system.string.aspx)> *readerNames* ) | |

Παράμετροι

readerNames

|  |
| --- |
|  |

Επιστρέφει

True if no readers are found

Πηγαίος κώδικας

|  |
| --- |
| private static bool NoReaderFound(ICollection<string> readerNames)  {  return readerNames == null || readerNames.Count < 1;  } |

Δείτε επίσης

Εφαρμόζεται σε: [SmartCardMonitor](#topic_0000000000000001)

SmartCardMonitor.ShowUserInfo Method

Function to log

|  |  |
| --- | --- |
| C# |  |
| private static [void](http://msdn.microsoft.com/en-us/library/system.void.aspx) ShowUserInfo(  [IEnumerable](http://msdn.microsoft.com/en-us/library/9eekhta0.aspx)<[string](http://msdn.microsoft.com/en-us/library/system.string.aspx)> *readerNames* ) | |

Παράμετροι

readerNames

|  |
| --- |
|  |

Πηγαίος κώδικας

|  |
| --- |
| private static void ShowUserInfo(IEnumerable<string> readerNames)  {  foreach (var reader in readerNames)  {  Console.WriteLine($"Start monitoring for reader {reader}.");  }    Console.WriteLine("Press Ctrl-Q to exit or any key to toggle monitor.");  } |

Δείτε επίσης

Εφαρμόζεται σε: [SmartCardMonitor](#topic_0000000000000001)

SmartCardMonitor.StatusChanged Method

StatusChanged, only logs to console

|  |  |
| --- | --- |
| C# |  |
| private static [void](http://msdn.microsoft.com/en-us/library/system.void.aspx) StatusChanged(  [object](http://msdn.microsoft.com/en-us/library/system.object.aspx) *sender*,  **StatusChangeEventArgs** *args* ) | |

Παράμετροι

sender

|  |
| --- |
|  |

args

|  |
| --- |
|  |

Πηγαίος κώδικας

|  |
| --- |
| private static void StatusChanged(object sender, StatusChangeEventArgs args)  {  Console.WriteLine(">> StatusChanged Event for reader: {0}", args.ReaderName);  Console.WriteLine("ATR: {0}", BitConverter.ToString(args.Atr ?? new byte[0]));  Console.WriteLine("Last state: {0}\nNew state: {1}\n", args.LastState, args.NewState);  } |

Δείτε επίσης

Εφαρμόζεται σε: [SmartCardMonitor](#topic_0000000000000001)

SmartCardMonitor.TrustDevice (String) Method

|  |  |
| --- | --- |
| C# |  |
| private [void](http://msdn.microsoft.com/en-us/library/system.void.aspx) TrustDevice(  [string](http://msdn.microsoft.com/en-us/library/system.string.aspx) *readerName* ) | |

Παράμετροι

readerName

|  |
| --- |
|  |

Εφαρμόζει

[IAltSecurity.TrustDevice](#)

Πηγαίος κώδικας

|  |
| --- |
| private void TrustDevice(string readerName)  {  SmartCard sa = new SmartCard(readerName);  var res = sa.GetAIDs(true);  SmartApplication sapp = sa.Applications[0];  sapp.SelectApplication();  sapp.GetProcessingOptions();  sapp.ReadRecords();  OfflineAuth oa = new OfflineAuth(sapp);    string cardNumber = sapp.GetTagValue(EmvConstants.ResponceType.ReaderRecord, "5A");  string expdate = sapp.GetTagValue(EmvConstants.ResponceType.ReaderRecord, "5F25");  ///Keep the hash data of cardnumber and expiryDate  string token = HashData(cardNumber + expdate, oa.ICC\_KEY\_HASH);    AltsInfo ok = new AltsInfo() { AuthDeviceId = token ,Authenticated = !string.IsNullOrEmpty(token) };  if (ok.Authenticated)  TrustedDevice = ok;  } |

Δείτε επίσης

Εφαρμόζεται σε: [SmartCardMonitor](#topic_0000000000000001)

SmartCardMonitor.TrustDevice (Trusted) Method

Function to train the application to trust a specified card

|  |  |
| --- | --- |
| C# |  |
| public [void](http://msdn.microsoft.com/en-us/library/system.void.aspx) TrustDevice(  **Trusted** *isTrusted* ) | |

Παράμετροι

isTrusted

|  |
| --- |
|  |

Εφαρμόζει

[IAltSecurity.TrustDevice](#)

Πηγαίος κώδικας

|  |
| --- |
| public async void TrustDevice(CallBacks.Trusted isTrusted)  {  \_isTrusted = isTrusted;  await Task.Factory.StartNew(() =>  {  TrainMode = true;    for(int i =0; i<10; i++)  {  Task.Delay(1000).Wait();  Console.Beep(5100, 200);  if (!TrainMode)  {    break;  }  }  \_isTrusted(TrustedDevice);  TrainMode = false;  });    } |

Δείτε επίσης

Εφαρμόζεται σε: [SmartCardMonitor](#topic_0000000000000001)

SmartCardMonitor.UnTrustDevice Method

Function to Untrust device. Nor yet implemented.

|  |  |
| --- | --- |
| C# |  |
| public [void](http://msdn.microsoft.com/en-us/library/system.void.aspx) UnTrustDevice(  **AltsInfo** *devInfo* ) | |

Παράμετροι

devInfo

|  |
| --- |
|  |

Εφαρμόζει

[IAltSecurity.UnTrustDevice](#)

Πηγαίος κώδικας

|  |
| --- |
| public void UnTrustDevice(AltsInfo devInfo)  {  throw new NotImplementedException();  } |

Δείτε επίσης

Εφαρμόζεται σε: [SmartCardMonitor](#topic_0000000000000001)

SmartCardMonitor.isTrusted Method

Check AltsInfo if the card is trusted or not

|  |  |
| --- | --- |
| C# |  |
| public [bool](http://msdn.microsoft.com/en-us/library/system.boolean.aspx) isTrusted(  **AltsInfo** *devToken* ) | |

Παράμετροι

devToken

|  |
| --- |
|  |

Επιστρέφει

True for trusted smart card

Πηγαίος κώδικας

|  |
| --- |
| public bool isTrusted(AltsInfo devToken) {  if (TrustedDevice == null)  return false;  if (TrustedDevice.Authenticated == false || string.IsNullOrEmpty(TrustedDevice.AuthDeviceId))  return false;      if (devToken == null)  return false;  if (devToken.Authenticated == false || string.IsNullOrEmpty(devToken.AuthDeviceId))  return false;    if (devToken.Authenticated && TrustedDevice.Authenticated)  return devToken.AuthDeviceId == TrustedDevice.AuthDeviceId;    return false;    } |

Δείτε επίσης

Εφαρμόζεται σε: [SmartCardMonitor](#topic_0000000000000001)

TrainMode Field

This flag is true while in training mode

|  |  |
| --- | --- |
| C# |  |
| private [bool](http://msdn.microsoft.com/en-us/library/system.boolean.aspx) TrainMode | |

Πηγαίος κώδικας

|  |
| --- |
| private bool TrainMode; |

Δείτε επίσης

Εφαρμόζεται σε: [SmartCardMonitor](#topic_0000000000000001)

TrustedDevice Field

Trusted device info

|  |  |
| --- | --- |
| C# |  |
| private **AltsInfo** TrustedDevice | |

Πηγαίος κώδικας

|  |
| --- |
| private AltsInfo TrustedDevice; |

Δείτε επίσης

Εφαρμόζεται σε: [SmartCardMonitor](#topic_0000000000000001)

\_contextFactory Field

Smart card system Conext factory. OS bound.

|  |  |
| --- | --- |
| C# |  |
| private readonly **IContextFactory** \_contextFactory | |

Πηγαίος κώδικας

|  |
| --- |
| private readonly IContextFactory \_contextFactory = ContextFactory.Instance; |

Δείτε επίσης

Εφαρμόζεται σε: [SmartCardMonitor](#topic_0000000000000001)

\_isOk Field

Authenticated callback function memory space

|  |  |
| --- | --- |
| C# |  |
| private **Authenticated** \_isOk | |

Πηγαίος κώδικας

|  |
| --- |
| private CallBacks.Authenticated \_isOk; |

Δείτε επίσης

Εφαρμόζεται σε: [SmartCardMonitor](#topic_0000000000000001)

\_isTrusted Field

Trusted callback function memory space

|  |  |
| --- | --- |
| C# |  |
| private **Trusted** \_isTrusted | |

Πηγαίος κώδικας

|  |
| --- |
| private CallBacks.Trusted \_isTrusted; |

Δείτε επίσης

Εφαρμόζεται σε: [SmartCardMonitor](#topic_0000000000000001)

\_monitor Field

|  |  |
| --- | --- |
| C# |  |
| private **ISCardMonitor** \_monitor | |

Πηγαίος κώδικας

|  |
| --- |
| private ISCardMonitor \_monitor; |

Δείτε επίσης

Εφαρμόζεται σε: [SmartCardMonitor](#topic_0000000000000001)

# Δείκτης

[AttachToAllEvents Method](#topic_000000000000000B)

[Authenticate Method](#topic_000000000000000F)

[CardInserted Method](#topic_000000000000000D)

[Connect Method](#topic_0000000000000008)

[Disconnect Method](#topic_0000000000000009)

[DisplayEvent Method](#topic_000000000000000C)

[GetReaderNames Method](#topic_0000000000000012)

[HashData Method](#topic_0000000000000017)

[MonitorException Method](#topic_0000000000000011)

[NoReaderFound Method](#topic_0000000000000013)

[ShowUserInfo Method](#topic_000000000000000A)

[SmartCardDriver Namespace](#topic_0000000000000000)

[SmartCardDriver Reference](#topic_0000000000000018)

[SmartCardMonitor Class](#topic_0000000000000001)

[StatusChanged Method](#topic_0000000000000010)

[TrainMode Field](#topic_0000000000000006)

[TrustDevice (String) Method](#topic_000000000000000E)

[TrustDevice (Trusted) Method](#topic_0000000000000015)

[TrustedDevice Field](#topic_0000000000000007)

[UnTrustDevice Method](#topic_0000000000000016)

[\_contextFactory Field](#topic_0000000000000002)

[\_isOk Field](#topic_0000000000000003)

[\_isTrusted Field](#topic_0000000000000004)

[\_monitor Field](#topic_0000000000000005)

[isTrusted Method](#topic_0000000000000014)