

# eVehicle Registration SDK Korisničko Uputstvo



Ministarstvo Unutrašnjih  
Poslova Republike Srbije

Jul 2011.

## Sadržaj

1. Opšti opis proizvoda .....	2
2. C/C++ API .....	3
2.1. Strukture.....	3
2.1.1. SD_DOCUMENT_DATA .....	3
2.1.2. SD_VEHICLE_DATA .....	4
2.1.3. SD_PERSONAL_DATA .....	5
2.1.4. SD_REGISTRATION_DATA.....	6
2.2. Funkcije.....	7
2.2.1. sdStartup .....	7
2.2.2. sdCleanup .....	7
2.2.3. GetReaderName .....	8
2.2.4. SelectReader .....	8
2.2.5. sdProcessNewCard .....	9
2.2.6. sdReadDocumentData.....	10
2.2.7. sdReadVehicleData.....	10
2.2.8. sdReadPersonalData.....	11
2.2.9. sdReadRegistration.....	11
3. Kodovi grešaka.....	14

## 1. Opšti opis proizvoda

Biblioteke *eVehicleRegistrationAPI* i *eVehicleRegistrationCOM* se sastoje od skupa funkcija visokog nivoa koji omogućava ciljanoj aplikaciji da u celosti iščita podatke iz kartice Saobraćajne dozvole.

Biblioteka *eVehicleRegistrationAPI* pruža API (Application Programming Interface) u čistom C formatu radi omogućavanja direktne integracije sa C/C++ programskim jezicima.

Ova biblioteka se sastoji od 32-bitnog Windows DLL-a, import biblioteke i heder fajla; poslednja dva fajla moraju biti uključena u C/C++ projekat.

Biblioteka ostvaruje komunikaciju sa smart karticama korišćenjem PC/SC sloja.

Biblioteka *eVehicleRegistrationCOM* sadrži servise po Microsoft COM modelu radi omogućavanja integracije sa većinom programskih jezika, uključujući tzv. „managed“ (upravljane) programske jezike kao što su C# i Visual Basic.

Ova biblioteka je u formi Unicode 32-bitnog DLL-a. Mora biti uključena u klijentski projekat radi dobijanja detalja o interfejsu i rešavanja dinamičkih linkova.

Biblioteke *eVehicleRegistrationAPI* i *eVehicleRegistrationCOM* obezbeđuju middleware potreban za pristup i čitanje podataka iz kartice Saobraćajne dozvole.

## 2. C/C++ API

C/C++ API je implementiran u *eVehicleRegistrationAPI.dll* biblioteci. Da bi mogla da ga koristi, aplikacija treba da uključi „*eVehicleRegistrationAPI.h*” heder fajl, kao i „*eVehicleRegistrationAPI.lib*” fajl, u projekat. Fajl „*eVehicleRegistrationAPI.dll*” mora takođe da bude dostupan i smešten u folder u kome se nalazi kreirana aplikacija, ili u folder koji je naveden u „PATH” promenljivoj, ili u *%windows%\System32* folder.

### 2.1. Struktura

Biblioteka koristi različite C strukture, ili „Plain Old Data strukture”, za razmenu podataka sa aplikacijom koja je poziva. Naredni odeljci opisuju svaku od njih.

Obratiti pažnju na to da je u svim strukturama svako pojedinačno polje definisano pomoću dve promenljive. Prva je niz 8-bitnih karaktera fiksne dužine, dok je druga tipa `long`, dužine 32 bita.

Ove strukture ne moraju biti inicijalizovane pre upotrebe; po vraćanju funkcija ove biblioteke, svi članovi zasnovani na tipu `char` sadrže string popunjen nulama, dok parametar dužine sadrži broj karaktera smeštenih u `char` nizu.

#### 2.1.1. SD\_DOCUMENT\_DATA

Ova struktura se koristi za dobijanje informacija o samom dokumentu saobraćajne dozvole.

```
typedef struct groupSD_DOCUMENT_DATA
{
char stateIssuing[50];
long stateIssuingSize;
char competentAuthority[50];
long competentAuthoritySize;
char authorityIssuing[50];
long authorityIssuingSize;
char unambiguousNumber[30];
long unambiguousNumberSize;
char issuingDate[16];
long issuingDateSize;
char expiryDate[16];
long expiryDateSize;
char serialNumber[20];
long serialNumberSize;
} SD_DOCUMENT_DATA;
```

Značenje i lokacija polja u čipu kartice su dati u sledećoj tabeli:

Polje	Značenje	Tag objekta podatka (EC-2003-127 ili ekstenzija)
stateIssuing	Država izdavanja	9F33h
competentAuthority	Ovlašćeni organ	9F35h
authorityIssuing	Organ izdavanja saobraćajne dozvole	9F36h
unambiguousNumber	Jedinstveni broj pod kojim je vozilo upisano u registar	9F38h
issuingDate	Datum izdavanja saobraćajne dozvole	8Eh
expiryDate	Važenje registracije	8Dh
serialNumber	Serijski broj saobraćajne dozvole	C9h

### 2.1.2. SD\_VEHICLE\_DATA

Ova struktura se koristi za dobijanje informacija o vozilu za koje važi saobraćajna dozvola.

```
typedef struct groupSD_VEHICLE_DATA
{
char dateOfFirstRegistration[16];
long dateOfFirstRegistrationSize;
char yearOfProduction[5];
long yearOfProductionSize;
char vehicleMake[100];
long vehicleMakeSize;
char vehicleType[100];
long vehicleTypeSize;
char commercialDescription[100];
long commercialDescriptionSize;
char vehicleIDNumber[100];
long vehicleIDNumberSize;
char registrationNumberOfVehicle[20];
long registrationNumberOfVehicleSize;
char maximumNetPower[20];
long maximumNetPowerSize;
char engineCapacity[20];
long engineCapacitySize;
char typeOfFuel[100];
long typeOfFuelSize;
char powerWeightRatio[20];
long powerWeightRatioSize;
char vehicleMass[20];
long vehicleMassSize;
char maximumPermissibleLadenMass[20];
long maximumPermissibleLadenMassSize;
char typeApprovalNumber[50];
long typeApprovalNumberSize;
char numberOfSeats[20];
long numberOfSeatsSize;
char numberOfStandingPlaces[20];
long numberOfStandingPlacesSize;
char engineIDNumber[100];
long engineIDNumberSize;
char numberOfAxles[20];
long numberOfAxlesSize;
char vehicleCategory[50];
long vehicleCategorySize;
char colourOfVehicle[50];
long colourOfVehicleSize;
char restrictionToChangeOwner[200];
long restrictionToChangeOwnerSize;
char vehicleLoad[20];
long vehicleLoadSize;
} SD_VEHICLE_DATA;
```

Značenje i lokacija polja u čipu kartice su dati u sledećoj tabeli:

Polje	Značenje	Tag objekta podatka (EC-2003-127 ili ekstenzija)
dateOfFirstRegistration	Datum prve registracije	82h
yearOfProduction	Godina proizvodnje	C5h
vehicleMake	Marka	A3h/87h
vehicleType	Tip	A3h/88h
commercialDescription	Komercijalna oznaka (model)	A3h/89h
vehicleIDNumber	Broj šasiје	8Ah
registrationNumberOfVehicle	Registarski broj vozila	81h

Polje	Značenje	Tag objekta podatka (EC-2003-127 ili ekstenzija)
maximumNetPower	Snaga motora u kW	5Ah/91h
engineCapacity	Radna zapremina motora	5Ah/90h
typeOfFuel	Vrsta goriva ili pogona	5Ah/92h
powerWeightRatio	Odnos snaga/masa u kg/kW (samo za motocikle)	93h
vehicleMass	Masa	8Ch
maximumPermissibleLadenMass	Najveća dozvoljena masa	A4h/8Bh
typeApprovalNumber	Homologacijska oznaka	8Fh
numberOfSeats	Broj mesta za sedenje uključujući i mesto vozača	A6h/94h
numberOfStandingPlaces	Broj mesta za stajanje	A6h/95h
engineIDNumber	Broj motora	A5h/9Eh
numberOfAxles	Broj osovine	99h
vehicleCategory	Vrsta vozila	98h
colourOfVehicle	Boja vozila	9F24h
restrictionToChangeOwner	Zabrana otuđenja vozila do	C1h
vehicleLoad	Nosivost vozila	C4h

### 2.1.3. SD\_PERSONAL\_DATA

Ova struktura se koristi za dobijanje informacija o vlasniku i/ili korisniku vozila.

```
typedef struct tagSD_PERSONAL_DATA
{
    char ownersPersonalNo[20];
    long ownersPersonalNoSize;
    char ownersSurnameOrBusinessName[100];
    long ownersSurnameOrBusinessNameSize;
    char ownerName[100];
    long ownerNameSize;
    char ownerAddress[200];
    long ownerAddressSize;
    char usersPersonalNo[20];
    long usersPersonalNoSize;
    char usersSurnameOrBusinessName[100];
    long usersSurnameOrBusinessNameSize;
    char usersName[100];
    long usersNameSize;
    char usersAddress[200];
    long usersAddressSize;
} SD_PERSONAL_DATA;
```

Značenje i lokacija polja u čipu kartice su dati u sledećoj tabeli:

Polje	Značenje	Tag objekta podatka (EC-2003-127 ili ekstenzija)
ownersPersonalNo	JMBG, odnosno matični broj vlasnika vozila	C2h
ownersSurnameOrBusinessName	Prezime vlasnika (firma odnosno naziv za pravna lica)	A1h/A7h/83h
ownerName	Ime vlasnika	A1h/A7h/84h
ownerAddress	Prebivalište (sedište) i adresa vlasnika vozila	A1h/A7h/85h

Polje	Značenje	Tag objekta podatka (EC-2003-127 ili ekstenzija)
usersPersonalNo	JMBG, odnosno matični broj korisnika vozila	C3h
usersSurnameOrBusinessName	Prezime korisnika vozila (firma odnosno naziv za pravna lica)	A1h/A9h/83h
usersName	Ime korisnika vozila	A1h/A9h/84h
usersAddress	Prebivalište (sedište) i adresa korisnika vozila	A1h/A9h/85h

#### 2.1.4. SD\_REGISTRATION\_DATA

Ova struktura se koristi za preuzimanje fajla koji sadrži podatke saobraćajne dozvole sa njegovim digitalnim potpisom i sertifikatom potpisnika dokumenta.

```
typedef struct groupSD_REGISTRATION_DATA
{
    char registrationData[4096];
    long registrationDataSize;
    char signatureData[1024]
    long signatureDataSize;
    char issuingAuthority[4096]
    long issuingAuthoritySize;
} SD_REGISTRATION_DATA;
```

Ova struktura omogućuje verifikaciju digitalnog potpisa fajla koji sadrži podatke saobraćajne dozvole i ispravnost digitalnog sertifikata generisanog od strane punovažnog potpisnika dokumenta. Ove provere su potrebne radi detekcije lažnih ili falsifikovanih kartica saobraćajne dozvole.

Ova struktura se popunjava pomoću *sdReadRegistration* funkcije; po vraćanju, polje *registrationData* sadrži podatke bez dodatnog padding-a (tj. sadrži samo template-e 78h i 71h ili 72h), dok polje *issuingAuthority* sadrži ASN1 sertifikat potpisnika bez dodatnog padding-a.

Polje *signatureData* sadrži pun sadržaj fajla pročitano iz smart kartice. S obzirom da biblioteka ne zna koji je mehanizam korišćen za potpisivanje i zbog toga što ovaj potpis nije enkapsuliran u nekom objektu podataka (kao što je „Signed Data Object“), ona ne može da odredi tačnu dužinu.

## 2.2. Funkcije

### 2.2.1. sdStartup

#### Prototip

```
long sdStartup(long version);
```

#### Namena

Ova funkcija inicijalizuje biblioteku. Interaguje sa PC/SC slojem da bi dobila informaciju o čitačima povezanim u sistem.

Ova funkcija mora biti pozvana jednom i to prva, pre bilo koje druge funkcije ovog SDK.

#### Parametri

`long version` – verzija API-ja, u ovom izdanju ne postoji nikakva kontrola ovog parametra

#### Obrazloženje

Ako je funkcija već bila pozvana, vratiće `ERROR_SERVICE_ALREADY_RUNNING` vrednost.

Ako ni jedan PC/SC čitač nije instaliran, funkcija vraća `SCARD_E_NO_READERS_AVAILABLE` vrednost.

Ako se primi neka PC/SC greška iz PC/SC sloja, onda se ta greška vraća.

Inače, funkcija vraća `S_OK` i sve ostale funkcije mogu da se koriste.

### 2.2.2. sdCleanup

#### Prototip

```
long sdCleanup();
```

#### Namena

Ova funkcija oslobađa sve alocirane resurse.

Koristi se kada usluge biblioteke nisu više potrebne.

#### Obrazloženje

Ako biblioteka nije uspešno inicijalizovana, funkcija vraća `ERROR_SERVICE_NOT_ACTIVE` vrednost.

Inače, biblioteka obavlja čišćenje resursa, i funkcija vraća `S_OK`.



### 2.2.3. GetReaderName

#### Prototip

```
long GetReaderName(long index, char* readerName, long* nameSize);
```

#### Namena

Ova funkcija obezbeđuje ime jednog od PC/SC čitača povezanih u sistem. Aplikacija koja je poziva je može iskoristiti za prikazivanje liste svih povezanih čitača, ili je može koristiti sa drugim funkcijama ove biblioteke.

#### Parametri

`long index` – indeks čitača, prvi čitač ima indeks 0, funkciju treba pozivati više puta sa inkrementiranjem ovog parametra za 1 da bi se dobila imena svih instaliranih čitača

`char* readerName` – adresa niza karaktera koji sadrži ime čitača nakon izvršenja funkcije

`long* nameSize` – pointer na `long` koji na ulasku sadrži veličinu niza `readerName`, a na izlasku iz funkcije stvarnu dužinu imena

#### Obrazloženje

Ako je `readerName` ili `nameSize` null, funkcija vraća `ERROR_INVALID_PARAMETER` vrednost.

Ako funkcija `sdStartup` nije bila pozvana, funkcija vraća `ERROR_SERVICE_NOT_ACTIVE` vrednost.

Ako je parametar `index` manji od 0 ili veći ili jednak stvarnom broju čitača, funkcija vraća `SCARD_E_UNKNOWN_READER` vrednost.

Ime određenog čitača se dobija iz PC/SC sloja; ako funkcija ne uspe da ga dobije, vraća vrednost `SCARD_E_READER_UNAVAILABLE`. Ako se dobije neka druga PC/SC greška iz PC/SC sloja, onda se vraća ta greška.

Ako je vrednost na koju pokazuje `nameSize` premala da bi sadržala ime, parametar `nameSize` se postavlja na potrebnu dužinu i funkcija vraća vrednost `SCARD_E_INSUFFICIENT_BUFFER`.

Inače, ime čitača se kopira u `readerName` bafer sa nultim karakterom na kraju, parametar `nameSize` se postavlja na dužinu iskopiranih karaktera (uključujući nulu) i funkcija vraća `S_OK`.

### 2.2.4. SelectReader

#### Prototip

```
long SelectReader(char* reader);
```

#### Namena

Ova funkcija selektuje određeni čitač. Ovaj odabir čitača je obavezan pre početka čitanja kartica, ali u čitaču ne mora da se nalazi kartica u trenutku pozivanja ove funkcije da bi bila izvršena uspešno.

## Parametri

`char* reader` – niz karaktera koji sadrži ime odabranog čitača

## Obrazloženje

Ako je `reader` null, funkcija vraća `ERROR_INVALID_PARAMETER` vrednost.

Ako funkcija `sdStartup` nije bila pozvana, funkcija vraća `ERROR_SERVICE_NOT_ACTIVE` vrednost.

Ako ne postoji čitač sa zadatim imenom, funkcija vraća `SCARD_E_UNKNOWN_READER` vrednost.

Inače, čitač biva odabran i funkcija vraća `S_OK`.

## 2.2.5. sdProcessNewCard

### Prototip

```
long sdProcessNewCard();
```

### Namena

Ova funkcija uspostavlja konekciju sa smart karticom koja je trenutno umetnuta u odabrani čitač, selektuje aplikaciju saobraćajne dozvole koja se nalazi u čipu kartice i kešira neke podatke struktura fajla koji sadrži podatke saobraćajne dozvole.

Pozivanje ove funkcije je obavezno pre početka čitanja nove kartice. Mora biti pozvana nakon umetanja nove kartice, ili, drugim rečima, pre bilo koje sekvence koja uključuje `sdReadDocumentData`, `sdReadVehicleData` i `sdReadPersonalData`.

### Parametri

Nema ih.

### Obrazloženje

Ako funkcija `sdStartup` nije bila pozvana, funkcija vraća `ERROR_SERVICE_NOT_ACTIVE` vrednost.

Ako ni jedan čitač nije bio prethodno odabran, funkcija vraća `SCARD_E_UNKNOWN_READER` vrednost.

Ako odabrani čitač ne sadrži smart karticu u sebi, funkcija vraća `SCARD_E_NO_SMARTCARD` vrednost.

Ako selekcija kartične aplikacije bude neuspešna, funkcija vraća `SCARD_E_CARD_UNSUPPORTED` vrednost.

Ako pročitani fajlovi sa podacima saobraćajne dozvole sadrže neispravne podatke, koji ne mogu biti parsirani od strane biblioteke, funkcija vraća `ERROR_INVALID_DATA` vrednost.

Inače, podaci saobraćajne dozvole bivaju keširani, `sdReadXxxxData` može biti pozvana, i funkcija vraća `S_OK`.

## 2.2.6. sdReadDocumentData

### Prototip

```
long sdReadDocumentData (SD_DOCUMENT_DATA* data);
```

### Namena

Ova funkcija popunjava strukturu koja sadrži podatke vezane za sam dokument.

### Parametri

SD\_DOCUMENT\_DATA\* data – adresa *SD\_DOCUMENT\_DATA* strukture

### Obrazloženje

Ako je data null, funkcija vraća E\_POINTER vrednost.

Ako funkcija sdProcessNewCard nije bila pozvana, funkcija vraća ERROR\_INVALID\_ACCESS.

Inače, funkcija popunjava datu strukturu podacima koji su prethodno pročitani iz smart kartice pomoću sdProcessNewCard funkcije i vraća S\_OK.

**Važna napomena:** Obratite pažnju da pozovete funkciju sdProcessNewCard svaki put kada se pokrene nov proces čitanja. U suprotnom, funkcija će vratiti podatke iz kartice saobraćajne dozvole koja je bila prisutna u čitaču kada je sdProcessNewCard poslednji put bila obrađena.

## 2.2.7. sdReadVehicleData

### Prototip

```
long sdReadVehicleData (SD_VEHICLE_DATA* data);
```

### Namena

Ova funkcija popunjava strukturu koja sadrži podatke o vozilu.

### Parametri

SD\_VEHICLE\_DATA\* data – adresa *SD\_VEHICLE\_DATA* strukture

### Obrazloženje

Ako je data null, funkcija vraća E\_POINTER vrednost.

Ako funkcija sdProcessNewCard nije bila pozvana, funkcija vraća ERROR\_INVALID\_ACCESS.

Inače, funkcija popunjava datu strukturu podacima koji su prethodno pročitani iz smart kartice pomoću sdProcessNewCard funkcije i vraća S\_OK.

**Važna napomena:** Obratite pažnju da pozovete funkciju `sdProcessNewCard` svaki put kada se pokrene nov proces čitanja. U suprotnom, funkcija će vratiti podatke iz kartice saobraćajne dozvole koja je bila prisutna u čitaču kada je `sdProcessNewCard` poslednji put bila obrađena.

### 2.2.8. `sdReadPersonalData`

#### Prototip

```
long sdReadPersonalData(SD_PERSONAL_DATA* data);
```

#### Namena

Ova funkcija popunjava strukturu koja sadrži podatke o vlasniku i/ili korisniku vozila.

#### Parametri

`SD_PERSONAL_DATA* data` – adresa `SD_PERSONAL_DATA` strukture

#### Obrazloženje

Ako je `data` null, funkcija vraća `E_POINTER` vrednost.

Ako funkcija `sdProcessNewCard` nije bila pozvana, funkcija vraća `ERROR_INVALID_ACCESS`.

Inače, funkcija popunjava datu strukturu podacima koji su prethodno pročitani iz smart kartice pomoću `sdProcessNewCard` funkcije i vraća `S_OK`.

**Važna napomena:** Obratite pažnju da pozovete funkciju `sdProcessNewCard` svaki put kada se pokrene nov proces čitanja. U suprotnom, funkcija će vratiti podatke iz kartice saobraćajne dozvole koja je bila prisutna u čitaču kada je `sdProcessNewCard` poslednji put bila obrađena.

### 2.2.9. `sdReadRegistration`

#### Prototip

```
long sdReadRegistration(SD_REGISTRATION_DATA* data, long index);
```

#### Namena

Ova funkcija popunjava strukturu koja sadrži podatke saobraćajne dozvole, digitalni potpis i digitalni sertifikat potpisnika dokumenta.

Ova tri bloka podataka su potrebna za verifikaciju digitalnog potpisa podataka saobraćajne dozvole i za verifikaciju digitalnog sertifikata potpisnika dokumenta. Ove dve verifikacije su neophodne da bi bili sigurni da saobraćajna dozvola nije lažna ili falsifikovana.

#### Parametri

`SD_REGISTRATION_DATA* data` – adresa `SD_REGISTRATION_DATA` strukture

`long index` – indeks, koji se kreće u opsegu od 1 do 3, fajla sa podacima saobraćajne dozvole koji se želi pročitati (četvrti fajl, koji je po domaćoj specifikaciji, nije digitalno potpisan, tako da ga ova funkcija ne koristi)

## Obrazloženje

Ako je `data` `null`, ili je `index` neispravan (nije jednak 1, 2 ili 3), funkcija vraća `ERROR_INVALID_PARAMETER` vrednost.

Ako funkcija `sdStartup` nije bila pozvana, funkcija vraća `ERROR_SERVICE_NOT_ACTIVE`.

Ako nije detektovan ni jedan čitač, ili ako je odabir čitača bio neuspešan, funkcija vraća `SCARD_E_UNKNOWN_READER` vrednost.

Ako je odabrani čitač prazan prilikom ovog poziva, funkcija vraća `SCARD_E_NO_SMARTCARD` vrednost.

Ako je neki čitač odabran i kartica je prisutna u čitaču, ali kartica ne zadovoljava zahteve saobraćajne dozvole, funkcija vraća `SCARD_E_CARD_UNSUPPORTED` vrednost.

Ako je neki pročitani fajl veći od polja koje se koristi za njegovo smeštanje, funkcija vraća `SCARD_E_INSUFFICIENT_BUFFER` vrednost.

Ako neki fajl sa podacima saobraćajne dozvole ne sadrži dva očekivana template-a (78h i 71h ili 72h), funkcija vraća `ERROR_BAD_FORMAT` vrednost.

Ako fajl sertifikata potpisnika dokumenta ne sadrži ASN1 sekvencu, funkcija vraća `ERROR_BAD_FORMAT` vrednost.

Inače, funkcija iščitava tri potrebna fajla iz čipa kartice, uklanja zero-padding iz podataka saobraćajne dozvole i sertifikata potpisnika dokumenta, i vraća `S_OK`. Primetiti da se nikakva obrada ne vrši nad pročitanim digitalnim potpisom, pogotovo se padding ne uklanja, jer fajl digitalnog potpisa sadrži sirov digitalni potpis bez ikakvog ASN.1 ili PKCS identifikatora formata.

Korišćeni indeks određuje fajl koji će biti pročitani. Dozvoljene vrednosti su date u sledećoj tabeli:

Indeks	Fajl sa podacima saobraćajne dozvole		Fajl digitalnog potpisa	Fajl sertifikata potpisnika
	Ime fajla	Identifikator	Identifikator	Identifikator
1	EF_Registration_A	D001h	E001h	C001h
2	EF_Registration_B	D011h	E011h	C011h
3	EF_Registration_C	D021h	E021h	C021h

### 3. Kodovi grešaka

Funkcije Win32 eVehicleRegistrationAPI biblioteke i metode interfejsa iz eVehicleRegistrationCOM biblioteke mogu vratiti greške iz sledećeg skupa:

Naziv greške	Vrednost
ERROR_BAD_FORMAT	11
ERROR_INVALID_ACCESS	12
ERROR_INVALID_DATA	13
ERROR_INVALID_PARAMETER	87
ERROR_SERVICE_ALREADY_RUNNING	1056
ERROR_SERVICE_NOT_ACTIVE	1062
E_POINTER	0x80004003
SCARD_E_INSUFFICIENT_BUFFER	0x80100008
SCARD_E_UNKNOWN_READER	0x80100009
SCARD_E_NO_SMARTCARD	0x8010000C
SCARD_E_INVALID_VALUE	0x80100011
SCARD_E_READER_UNAVAILABLE	0x80100017
SCARD_E_CARD_UNSUPPORTED	0x8010001C
SCARD_E_NO_READERS_AVAILABLE	0x8010002E