
RoadByte eID Framework

“Programming the eID and other smart cards... the easy way”



RoadByte Solutions presents the **RoadByte eID Framework** product. For organisations and individuals who want to develop their own eID solutions, but without having to learn the daunting technical details of smart card communications.

Initially the only supported card type is the Belgian electronic identity card (eID).

The RoadByte eID Framework can be downloaded from www.roadbyte.com.

Several other eID products are also available from the web site, but this document will focus on the Framework and will explain how an organisation can use it to create e-ID enabled applications in a matter of days or weeks, not months.

Unique benefits:

- Easy: only general programming experience is sufficient to develop applications that access the eID.
- Fast: read the data on the card in less than 1 second!
- Extensible: currently implemented for the Belgian eID, but can easily be extended later to read eID cards from other countries, or indeed other card types (VISA, SIS, ...)
- Highly responsive
- Support for multiple card readers
- Highly configurable: specify different scenarios for each available card reader
- Available from most major Windows programming environments: .NET, C++, Visual Basic, Delphi, Java, VBA (Office-basic), COM scripting, ...

More time to focus on your own area of expertise

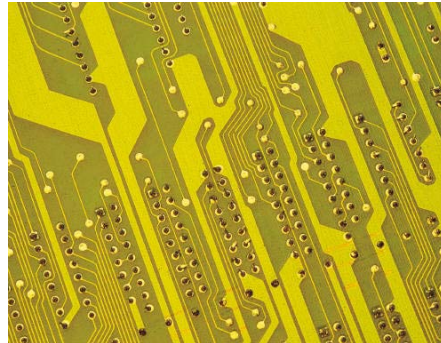


Why reinvent the wheel? By letting the RoadByte Plugin Framework deal with all the technical details of accessing a smart card, you can save months of development time to spend on your own area of expertise.

Smart card fun

Ironic title really, because programming for smart cards comes with quite a steep learning curve and good documentation can be hard to find.

- Limited support in most development environments: resort to Win32 API commands
- Not much documentation available, even on the internet
- Low-level programming is not much fun when you're used to writing windows applications
- Need to implement this for each card type you want to support
- Need to maintain and update code whenever a new version of the card is released



RoadByte eID Framework

Eager to get started on writing eID enabled applications, but frustrated with the limitations of the available alternatives, RoadByte Solutions decided to develop a framework that can be used by professionals to write production standard code.

- Fast
- No dialog boxes
- Easy to install
- Extensible
- Friendly support

Write your own eID code



The eID card has a thousand possible uses, and RoadByte Solutions will be happy to develop a solution that tailors for your specific needs.

But since there is only so much we can do, and since each company will have their own specific requirements, we have decided to make the framework available to everyone.

As a developer, you can now access the eID through our type library from most major Windows programming environments: .NET, C++, Visual Basic, Delphi, Java, VBA (Office-basic), COM scripting, ...

If you want to go even further, you can even write your own plugin for the RoadByte eID Monitor. Your code will then automatically be called as soon as a card is inserted or removed.

Card Type Library

The RoadByte eID framework comes with our own type library, which contains a hierarchy of interfaces that provide easy access to the card details.

- Card
 - Reader
 - CardHolder
 - FirstName
 - FirstNames
 - LastName
 - Name
 - Address
 - Street
 - Number
 - City
 - ...
 - Picture
 - ...
 - ...

Sample source code: .NET C#

We provide a sample C# project that makes building your first eID application a breeze. This is the code that will display the cardholder's name and street as soon as the card is inserted:

```
public override void CardInserted(IRbCard card)
{
    MessageBox.Show("Card inserted: " + card.CardHolder.Name);

    MessageBox.Show("Street: " + card.CardHolder.Address.Street);
}
```

Sample source code: VBA Microsoft Word

The following sample code shows the cardholder name and address from a Word document macro :

```
'Get RoadByte eID Framework
Dim framework As IRbApplication
Set framework = CreateObject("RoadByte.Application")
framework.Open

'Get card reader
Dim cardreader As IRbCardReader
Set cardreader = framework.ActiveCardReader

'Check if card present
If (cardreader Is Nothing) Then
    tbName.Text = "<No card reader found>"
    tbAddress.Text = ""
ElseIf (cardreader.Status <> csPresent) Then
    tbName.Text = "<No card in reader>"
    tbAddress.Text = ""
Else
    Dim cardholder As IRbPerson
    Set cardholder = cardreader.card.Holder
    tbName.Text = cardholder.Name
    tbAddress.Text = cardholder.Address.StreetAndNumber
End If

'Always close framework when done
framework.Close
```

Sample source code: HTML + JavaScript

This code initialises the RoadByte Java Applet in a HTML file:

```
<applet name="roadbyte" code="com.roadbyte.eid.Application" archive="roadbyte.jar">
    Applet not displayed - requires a Java enabled browser
</applet>
```

The following Javascript code loads the attached card readers into a drop-down list:

```
function load_cardreaders()
{
    var cardReaders = roadbyte.getCardReaders();
    cardReaders.Reset();

    var select_readers = document.myform.cardreader;
    select_readers.options.length=0;
    select_readers.options[select_readers.options.length] = new Option("-----");

    while (cardReaders.MoveNext())
    {
        var curReader = cardReaders.getCurrent();
        select_readers.options[select_readers.options.length] = new
            Option(curReader.getName());
    }
}
```

This code shows the data from the card, including the name, address, picture etc:

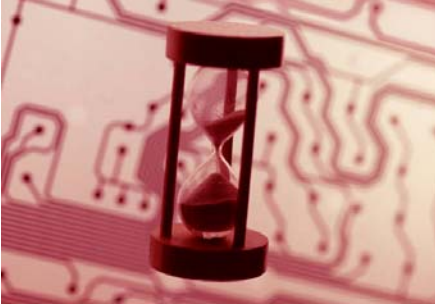
```
var cardDetails = card.getDetails();
document.myform.details_cardnumber.value = cardDetails.getCardNumber();
document.myform.details_chipnumber.value = cardDetails.getChipNumber();
document.myform.details_deliverylocation.value = cardDetails.getDeliveryLocation();
document.myform.details_documenttype.value = cardDetails.getDocumentType();
document.myform.details_validbegin.value = cardDetails.getValidBegin();
document.myform.details_validend.value = cardDetails.getValidEnd();

var cardHolder = card.getHolder();
document.myform.person_name.value = cardHolder.getName();
document.myform.person_firstname.value = cardHolder.getFirstName();
document.myform.person_firstnames.value = cardHolder.getFirstNames();
document.myform.person_lastname.value = cardHolder.getLastName();
document.myform.person_fullname.value = cardHolder.getFullName();
document.myform.person_gender.value = cardHolder.getGender();
document.myform.person_placeofbirth.value = cardHolder.getPlaceOfBirth();
document.myform.person_dateofbirth.value = cardHolder.getDateOfBirth();
document.myform.person_nationality.value = cardHolder.getNationality();
document.myform.person_nationalidnumber.value = cardHolder.getNationalIDNumber();
document.myform.person_noblecondition.value = cardHolder.getNobleCondition();
document.myform.person_isextendedminority.value = cardHolder.getIsExtendedMinority();
document.myform.person_iswhitecane.value = cardHolder.getIsWhiteCane();
document.myform.person_isyellowcane.value = cardHolder.getIsYellowCane();

var address = cardHolder.getAddress();
document.myform.address_streetandnumber.value = address.getStreetAndNumber();
document.myform.address_street.value = address.getStreet();
document.myform.address_number.value = address.getNumber();
document.myform.address_postcode.value = address.getPostCode();
document.myform.address_city.value = address.getCity();

var picture = cardHolder.getPicture();
document.images[1].src = picture.getTempFileName();
```

Ultra-fast access to the Belgian eID card



The RoadByte Plugin Framework uses direct APDU commands to communicate with the smart card and the card reader.

As a result, accessing the card details is extremely fast. Standard text details such as the cardholder's name and address can be retrieved in less than 1 second!

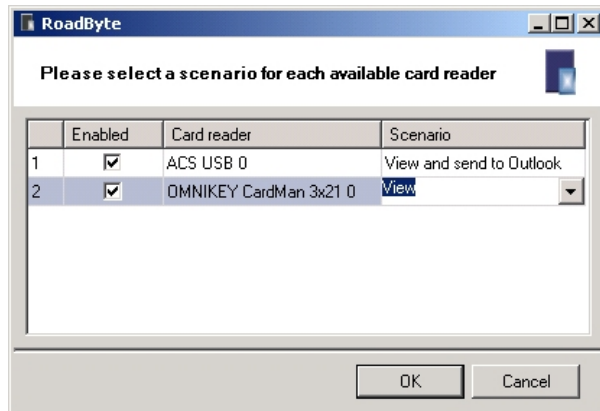
Increased responsiveness

For a better user experience, the framework is designed to respond immediately as soon as a card is inserted. After all, whenever you insert an eID card, you do it for a reason. So why should you have to start up another program first, select your card reader and click a button to read the card?

Multiple card readers

The RoadByte Plugin Framework supports more than just a single card reader. In fact, for each card reader that is connected to your system, you can specify which actions should be performed as soon as a card is inserted into that card reader.

This makes it extremely flexible, especially if you want to define different scenarios for each card reader.



Be prepared.. look beyond the Belgian eID card

Many countries are currently evaluating national electronic identification cards, and not just in Europe. One thing they have in common is that they all use their own standards. This means that what you develop for the Belgian card will not work for a French eID card.

RoadByte Framework is extensible

The RoadByte Plugin Framework can easily be extended to support new card types. In most cases this will not even require a software version update. Downloading a Card Type File (*.ctf) from the RoadByte website is all it will take for your software to be able to read the new French or German eID cards.



This same process applies for different versions of the Belgian eID card. The internal structure of the Belgian eID card will not remain constant. The thing to remember is that we have taken this into account, and you will not have to reinstall a new version of the framework with each new card version. A simple download (possibly even automated) will add the new card file type to the framework's repository.

LICENSE TYPES

The RoadByte Plugin Framework is licensed on a PER MACHINE basis, which means that for each machine on which the Framework is installed, a valid license must exist. Software licenses are provided as files containing unique registration codes and/or serial numbers that are bound to a specific machine.

There are several license types available for the RoadByte eID Framework, each one oriented towards a different target audience. Please refer to our website <http://www.roadbyte.com> for pricing details and to find out which license type is right for you.

Evaluation License



Allows installation and execution of the Framework on a single machine per license, without charge for a trial period of 30 days. During this period the Framework will have the same functionality as the “standard” licence.

Standard License



Allows installation and execution of the Framework on a single machine per license. This licence is limited to 5 different cards per week. This is ideal for home users.

Professional License



Allows installation and execution of the Framework on a single machine per license. This license is limited to 30 different card insertions per hour. This comes down to an average of 1 card every 2 minutes, which should be sufficient for most professional applications.

Enterprise License



Allows installation and execution of the Framework on a single machine per license. This license has no restrictions and is suitable for high-performance applications.