



mgsSepa.dll

SEPA LIBRARY
MICHAEL GRUBE

Vorwort

Die vorliegende DLL hilft Ihnen mit wenig Aufwand eine Sepa,- Lastschrift oder Überweisung zu erstellen. Dazu binden Sie die DLL in Ihrem Programmcode einfach mit ein. Jede Programmierumgebung, die mit .NET DLL Libraries arbeiten kann, unterstützt die Erstellung.

Die DLL wurde in Visual Studio 2013 in Visual Basic, unter dem Framework 4.5 erstellt. Sollten Sie es in einem niedrigeren Framework benötigen, so senden Sie dazu eine Nachricht.

Die nachfolgenden Informationen beschäftigen sich damit, Ihnen den Einsatz zu zeigen. Dabei werden die Umgebungen Visual Basic .NET und C# zum Einsatz kommen.

Inhaltsverzeichnis

Erstellung der CSV und DAT Datei	- 3 -
Einsatz in Visual Basic .NET.....	- 4 -
Erste Überladung.....	- 5 -
Zweite Überladung.....	- 6 -
Bankleitzahlen Download.....	- 8 -
Datei über Dialog suchen	- 8 -
Support	- 9 -
Quellcode in C#	- 10 -
Abschluss.....	- 11 -

Erstellung der CSV und DAT Datei

Damit die DLL überhaupt eine richtige SEPA-Datei erstellen kann, benötigt sie eine Quelldatei. Diese Datei ist eine einfache Excel-CSV, mit Semikolon oder eine ASC, DAT oder TXT Datei, mit TAB als Trennzeichen sein. Dabei ist die Struktur nicht ganz unwichtig.

Aufbau der CSV Datei. Dabei ist die Beschriftung optional. Ebenso wie die Spaltenbeschriftung lautet ist unwichtig. Sie müssen allerdings die Reihenfolge der Felder beibehalten.

**Nummer; Name; Verwendungszweck; Betrag; IBAN; BIC
1; Max Mustermann; Mein Beitrag für 2016; 15 €; DE74500105171234567890; INGDEFFXXX**

Der Aufbau eine DAT, ASC oder Textdatei kann unterschiedlich sein.

Achten Sie lediglich auf das Trennzeichen. In diesen Formaten wird im Normalfall ein TAB durch die unterschiedlich eingesetzten Programmen verwendet, daher die Unterscheidung.

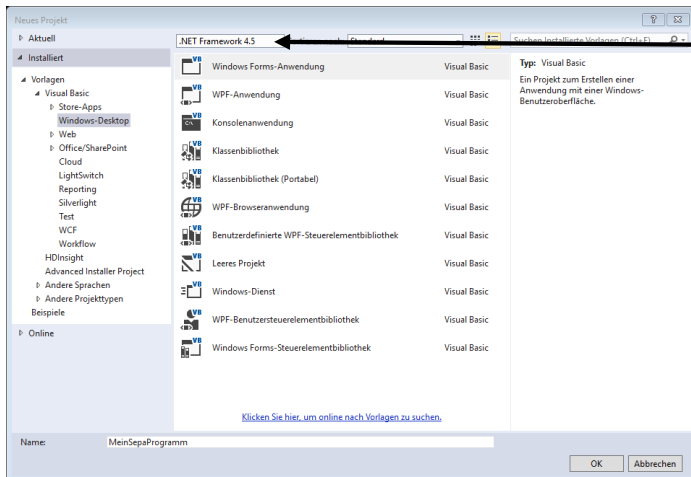
Prinzipiell können Sie aber alle Dateien einlesen, sofern die Reihenfolge und das Trennzeichen bekannt sind.

Nummer	Name	Verwendungszweck	Betrag	IBAN	BIC
1	Max Mustermann	Mein Beitrag für 2016	15 €	DE74500105171234567890	INGDEFFXXX

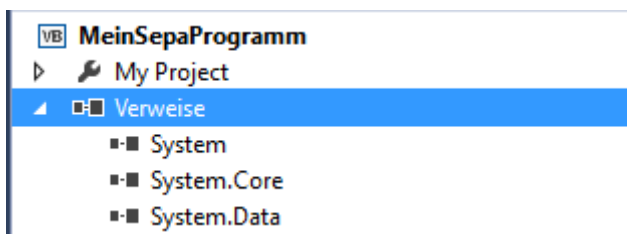
Einsatz in Visual Basic .NET

Damit Sie die DLL in Ihren Anwendungen einsetzen können, benötigen Sie zunächst einen Verweis auf die DLL. Da in der Regel die DLL im gleichen Verzeichnis liegt, wie Ihre Anwendung, benötigen Sie kein Installationsprogramm bzw. Routine zur Registrierung.

Ein Verweis müssen Sie trotzdem vornehmen. Hierzu Erstelle ich ein kleines Programm und dokumentiere dieses.

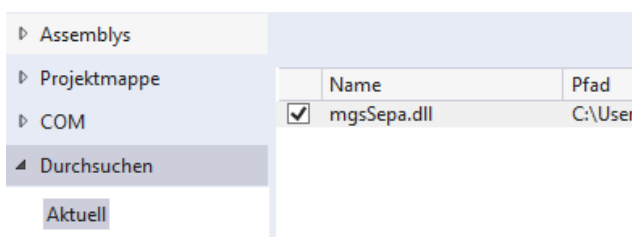


Bei der Erstellung ist das Framework nicht ganz unwichtig. Achten Sie daher darauf.

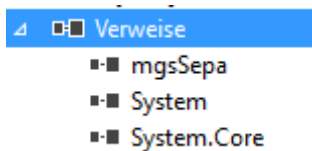


Klicken Sie mit der rechten Maustaste auf Verweise und legen den Pfad auf die mgsSEPA.dll.

Verweis-Manager - MeinSepaProgramm

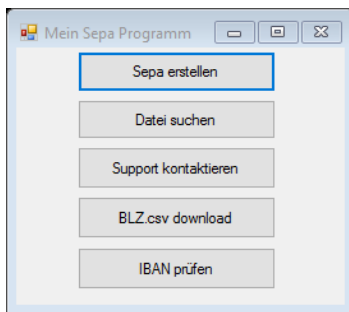


Über den Verweise-Manager navigieren Sie zur DLL



Danach sollte das Projekt wie folgt aussehen.

Nachdem Sie nun den Verweis erstellt haben, könnte der Rest so aussehen.



Hier sehen Sie eine kleine Form und Button. Der Programmcode dahinter ist relativ einfach.

Erste Überladung

Z.B. finden Sie unter „meinSepa.SepaErstellen()“ verschiedene Parameterangaben.

Parameter 1 = „Seriennummer“, (Muss übergeben werden. Bei Demo werden 5 Sätze verarbeitet)

Parameter 2 = „Quelldatei“, (Zu übersetzende Datei)

Parameter 3 = „Satztrennzeichen“, (Trennzeichen der Satzblöcke)

Parameter 4 = „Dialog Anzeigen“, (Wird ein zusätzlicher Dialog zu bearbeiten angezeigt)

Parameter 5 = „Zielordner“, (Fertige Datei speichern)

`Imports mgsSepa`

```
Public Class Form1
```

```
Private Sub btnSepa_Click(sender As Object, e As EventArgs) Handles btnSepa.Click
```

```
Dim meinSepa As New mgsSepa.mgsEinlesen("4711", My.Computer.Info.InstalledUICulture.LCID.ToString)
```

```
Dim bolErstellt As Boolean
```

```
bolErstellt = meinSepa.SepaErstellen("C:\Temp\Daten.csv", ";", True, "C:\Temp")
```

```
If bolErstellt = True Then
```

```
    MessageBox.Show("Datei wurde erfolgreich erstellt", "Hinweis", MessageBoxButtons.OK)
```

```
End If
```

```
End Sub
```

```
Private Sub BLZDownload() ...
```

```
End Class
```

Der obige Quellcode zeigt wie einfach nun die Erstellung einer Sepa-Datei ist. Die DLL bietet aber auch noch weitere Funktionen.

Zweite Überladung

Z.B. finden Sie unter „meinSepa.SepaErstellen()“ verschiedene Parameterangaben. Überladung der 2 Funktionen mit mehreren Parametern.

```
Imports mgsSepa
```

```
Public Class Form1
```

```
Private Sub btnSepa_Click(sender As Object, e As EventArgs) Handles btnSepa.Click
```

```
Dim meinSepa As New mgsSepa.mgsEinlesen("4711", My.Computer.Info.InstalledUICulture.LCID.ToString)  
Dim bolErstellt As Boolean
```

```
bolErstellt = meinSepa.SepaErstellen("C:\Temp\Daten.csv", ";",  
True, True, "CC", "B2B", "Max Mustermann", "DE12340056789123456789", "INGDEFFXXX",  
"DE00ZZ12345645852", "NOTPROVIDED", "01.02.2016", "pain.008.003.02", True,  
"C:\Temp")
```

```
If bolErstellt = True Then  
    MessageBox.Show("Datei wurde erfolgreich erstellt", "Hinweis", MessageBoxButtons.OK)  
End If
```

```
End Sub
```

```
Private Sub BLZDownload() ...
```

```
End Class
```

Parameter 1 = „Seriennummer“, (Muss übergeben werden. Bei Demo werden 5 Sätze verarbeitet)

Parameter 2 = „Quelldatei“, (Zu übersetzende Datei)

Parameter 3 = „Satztrennzeichen“, (Trennzeichen der Satzblöcke)

Parameter 4 = „Erste Zeile Überschrift“, (Ist erste Zeile Überschrift)

Parameter 5 = „Lastschrift“, (Ist Lastschrift oder Überweisung)

Parameter 6 = „Lastschriftart“, (Lastschrift = CC, Überweisung = TRF)

Parameter 7 = „Einzugsart“, (Basislastschrift = CORE, Eil = CORE1 oder Firmenlastschrift = B2B)

Parameter 8 = „Kontoinhaber“,

Parameter 9 = „IBAN des Kontoinhabers“,

Parameter 10 = „BIC des Kontoinhabers“,

Parameter 11 = „Gläubiger-ID“,

Parameter 12 = „Referenz-ID“,

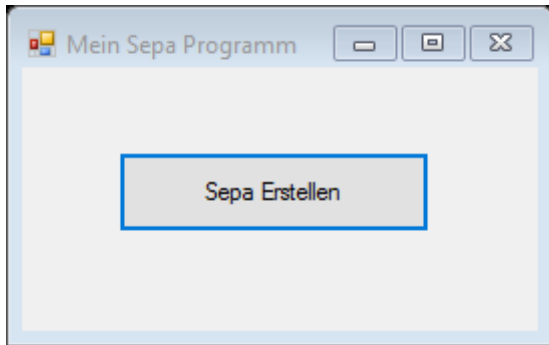
Parameter 13 = „Ausführungsdatum der Bank“,

Parameter 14 = „Pain-Einstellung“,

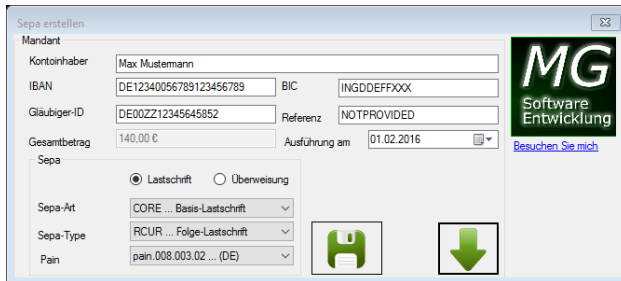
Parameter 15 = „Dialog Anzeigen“, (Wird ein zusätzlicher Dialog zu bearbeiten angezeigt)

Parameter 16 = „Zielordner“, (Fertige Datei speichern)

Beide Funktionen SepaErstellen() überladen sich, so dass die Erste die Zweite aufruft. In diesem Zusammenhang wird automatisch der Dialog mit angezeigt, um nachfolgende Daten wie Kontoinhaber etc. zu ergänzen.



Sobald Sie auf „Sepa Erstellen“ geklickt haben, die Parameter entsprechend so eingestellt haben, dass der Dialog eingeblendet wird, um die Daten zu überprüfen oder Ergänzungen vorzunehmen, erscheint wie unten dargestellt der Dialog.



In diesem Dialog können Sie noch jede Änderung vornehmen, die Notwendig ist. Allerdings kommt der Gesamtbetrag aus der Datei und kann nicht angepasst werden.

Klicken Sie zum Abschluss auf das Disketten-Symbol und die Sepa-Datei wird gemäß Ihren Vorgaben in das Zielverzeichnis gespeichert, welches Sie dann mit Ihrer Bankensoftware weiterverarbeiten können.

Klicken Sie auf dem Pfeil, erscheint das Fenster ein wenig Größer und Sie können die eingelesene Datei betrachten.

Mandat	Name	Verwendungszweck	Betrag	IBAN	BIC
1	Max Mu...	BEITRAG 2013	70,00 €	DE55123456...	INGDDEFFX...
2	Ines Sc...	BEITRAG 2013	70,00 €	DE55451252...	INGDDEFFX...

Bankleitzahlen Download

Die mgsSEPA.dll verfügt über eine Schnittstelle, die Ihnen die stets aktuellen Bankdaten im gleichen Verzeichnis herunterlädt.

Nachfolgender Programmcode lädt die Datei von meiner Webseite herunter. Diese Datei ändert sich alle 3 Monate, gemäß der Richtlinie der Banken.

```
Imports mgsSepa

Public Class Form1

    Private Sub btnSepa_Click(sender As Object, e As EventArgs) Handles btnSepa.Click ...

    Private Sub BLZDownload()
        Dim mgs As New mgsSepa.mgsEinlesen("4711", My.Computer.Info.InstalledUICulture.LCID.ToString)
        mgs.BLZDownload()
    End Sub

End Class
```

Datei über Dialog suchen

Sie haben auch die Möglichkeit, die Datei mittels „Datei öffnen Dialog“ zu navigieren. Dabei benötigen Sie beim Aufruf das Satztrennzeichen. Standardmäßig ist das Semikolon vorgesehen. Sollten Sie z.B. eine ASC, DAT oder sonstige Datei verwenden, die als Satztrennung ein TAB-Zeichen verwendet, müssen Sie es mit angeben.

```
Private Sub btnDateiEinlesen_Click(sender As Object, e As EventArgs) Handles btnDateiEinlesen.Click
    Dim meinSepa As New mgsSepa.mgsEinlesen("4711", My.Computer.Info.InstalledUICulture.LCID.ToString)
    meinSepa.DateiEinlesen()
End Sub
```

Die andere Variante ist das angeben des Satztrennzeichen.

```
Private Sub btnDateiEinlesen_Click(sender As Object, e As EventArgs) Handles btnDateiEinlesen.Click
    Dim meinSepa As New mgsSepa.mgsEinlesen("4711", My.Computer.Info.InstalledUICulture.LCID.ToString)
    meinSepa.DateiEinlesen(";", vbTab)
End Sub
```

IBAN Gültigkeit prüfen

Sie können jeden verwendeten IBAN prüfen lassen. Dabei beschränkt sich die DLL auf deutsche Kontodaten, da die Berechnung in den EU-Ländern unterschiedlich ist.

```
Private Sub btnBLZ_Click(sender As Object, e As EventArgs) Handles btnBLZ.Click
    Dim IBAN As New mgsIBANCheck
    Dim bolOK As Boolean = IBAN.IBANChecked("DE74123456781234567890")
    If bolOK = False Then
        MessageBox.Show("IBAN ist ungültig", "Hinweis", MessageBoxButtons.OK, MessageBoxIcon.Error)
    End If
End Sub
```

Support

Sie haben in der Vollversion die Möglichkeit, den Support per Email zu kontaktieren. Hierzu schreiben Sie einfach folgenden Code.

```
Private Sub btnSupport_Click(sender As Object, e As EventArgs) Handles btnSupport.Click
    Dim meinSepa As New mgsSepa.mgsSupport("4711")
    Dim bolGesendet As Boolean
    bolGesendet = meinSepa.Support("Absender: Email@firma.de",
                                   "Mitteilung: Was möchten Sie mitteilen",
                                   "Telefon: Optional z.B: 030 123456")

    If bolGesendet = False Then
        MessageBox.Show("Ein Fehler ist aufgetreten", "Hinweis", MessageBoxButtons.OK, MessageBoxIcon.Er
    End If
End Sub
```

Sollten Sie mit der Demoversion arbeiten, ist diese Funktion nicht verfügbar.

Quellcode in C#

Auch hier müssen Sie in Ihrem Projekt einen Verweis auf die DLL setzen, bevor Sie damit arbeiten können. Dies funktioniert in C# ebenso wie in Visual Basic.

Aus diesem Grunde zeige ich hier nur den vollständigen Quellcode.

```
using mgsSepa;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnSepaErstellen_Click(object sender, EventArgs e)
        {
            String Sprache = System.Globalization.CultureInfo.CurrentCulture.LCID.ToString();
            bool bolErstellt = false;
            mgsSepa.mgsEinlesen meinSepa = new mgsSepa.mgsEinlesen("4711", Sprache);
            bolErstellt = (bool)meinSepa.SepaErstellen(@"C:\Temp\Daten.csv", ";", true, @"C:\Temp");
            if (bolErstellt == true)
            {
                MessageBox.Show("Datei wurde erfolgreich erstellt", "Hinweis", MessageBoxButtons.OK);
            }
        }

        private void BLZDownload()
        {
            String Sprache = System.Globalization.CultureInfo.CurrentCulture.LCID.ToString();
            mgsSepa.mgsEinlesen meinSepa = new mgsSepa.mgsEinlesen("4711", Sprache);
            meinSepa.BLZDownload();
        }

        private void DateiEinlesen()
        {
            String Sprache = System.Globalization.CultureInfo.CurrentCulture.LCID.ToString();
            mgsSepa.mgsEinlesen meinSepa = new mgsSepa.mgsEinlesen("4711", Sprache);
            meinSepa.DateiEinlesen();
            //Alternative siehe VB-Code
            meinSepa.DateiEinlesen("", System.ConsoleKey.Tab.ToString());
        }

        private void Support()
        {
            mgsSepa.mgsSupport Hilfe = new mgsSepa.mgsSupport("4711");
            bool bolGesendet = false;
            bolGesendet = (bool)Hilfe.Support("info@firma.de",
                                             "Ihre Nachricht",
                                             "Telefonnummer");

            if (bolGesendet == false)
            {
                MessageBox.Show("Fehler beim Versand", "Hinweis", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
}
```

```
private void IBANPruefen()
{
    mgsIBANCheck IBAN = new mgsIBANCheck();
    bool bo1OK = false;
    bo1OK = (bool)IBAN.IBANChecked("DE74123456781234567890");
    if (bo1OK == false)
    {
        MessageBox.Show("IBAN ist ungültig", "Hinweis", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

Abschluss

Software-Hersteller haben die Möglichkeit, sich die DLL individuell anpassen zu lassen. Fragen Sie nach den Möglichkeiten. Ebenso besteht die Möglichkeit, den Quellcode zu erwerben. Auch hierzu bitte eine Kontaktaufnahme.