

DidiSoft OpenSSL Library for .NET / 1.4

Table of contents

Introduction	3
Installation folder	3
Library contents	3
Library DLL files	3
.NET Framework 4.5, 4.6	3
.NET Framework 4.0	4
.NET Framework 2.0, 3.0, 3.5	4
.NET Core	4
Referencing	4
Setup	6
Deployment	6
Switch from Trial to Production version	6
Register the help in MS Visual Studio	7
NuGet	8
Upgrade	9
General rules	9
From version 1.3	9
Examples	10
Online	10
Example Applications	11
Appendix	12
Common Exceptions	12
Exporting keys from a GnuPG keystore	12
Licensing	13
Support	14

Introduction

This manual describes **DidiSoft OpenPGP Library for .NET**.
The reader is assumed to have a background in Microsoft .NET Framework.

About the Library

DidiSoft OpenPGP Library for .NET is a **pure 100% managed library** with no external dependencies.

The library provides high level functions for OpenPGP encryption, decryption, signing and verification of signed data, clear text signing, one pass signing and encryption, key generation, key signing, key revocation, etc.

Internally the library uses the open source BouncyCastle.NET library.

Installation folder

The **default installation folder** for the library is :

C:\Program Files\OpenSSL Library for .NET 1.1

Library contents

After setup the [library installation folder](#) contains the following files and subfolders:

\Bin	Library DLL files
\Examples	Example projects (in C# and VB.NET)
\Help	Library API Help files
\License	License agreement
ReleaseNotes.txt	Release notes files
Documentation.pdf	this file in PDF format
Documentation.doc	this file in doc format (Wordpad, MS Word compatible)

Library DLL files

OpenPGP Library for .NET DLL Files

The library consists of two DLL files, located at the **\Bin** folder in the [library installation folder](#).

They must be [referenced](#) in your project and [distributed](#) with your software in order the library to work.

You will find also additional DLL files in subfolders, that target different variants of the .NET framework.
Please check below for more details.

.NET Framework 4.5, 4.6

The library consists of two main .NET DLL files, located within the **\Bin** folder in the [library installation folder](#).

Targeting **.NET Framework 4.5.x and above**

- 1) \Bin\NET45\DidiSoft.OpenSsl.dll
- 2) \Bin\NET45\DidiSoft.BouncyCastle.dll

For **standard managed applications** that will run on **.NET Framework** the above files must be [referenced](#) in your project and [distributed](#) with your software in order the library to work.

.NET Framework 4.0

The library consists of two main .NET DLL files, located within the **\Bin** folder in the [library installation folder](#).

Targeting **.NET Framework 4.0**

- 1) \Bin\NET40\DidiSoft.OpenSsl.dll
- 2) \Bin\NET40\DidiSoft.BouncyCastle.dll

For **standard managed applications** that will run on **.NET Framework** (including Client profile) the above files must be [referenced](#) in your project and [distributed](#) with your software in order the library to work.

.NET Framework 2.0, 3.0, 3.5

The library consists of two main .NET DLL files, located within the **\Bin** folder in the [library installation folder](#).

Targeting **.NET Framework 2.0, 3.0, 3.5 and Mono**

- 1) \Bin\NET20\DidiSoft.OpenSsl.dll
- 2) \Bin\NET20\DidiSoft.BouncyCastle.dll

For **standard managed applications** that will run on **.NET Framework** (including Client profile) the above files must be [referenced](#) in your project and [distributed](#) with your software in order the library to work.

.NET Core

The library consists of two main .NET DLL files, located within the **\Bin** folder in the [library installation folder](#).

Targeting **.NET Core 1.1**

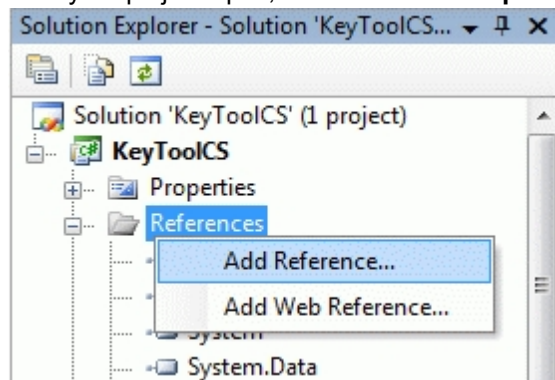
- 1) \Bin\NET45\DidiSoft.OpenSsl.dll
- 2) \Bin\NET45\DidiSoft.BouncyCastle.dll

the above files must be [referenced](#) in your project and [distributed](#) with your software in order the library to work.

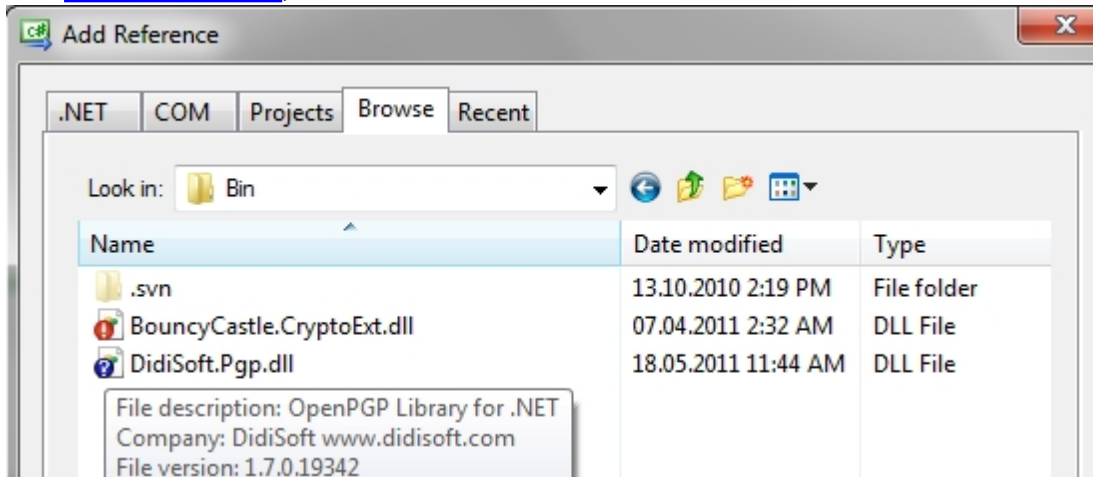
Referencing

To use the class library in **Visual Studio**, you need to add a reference to the [class library DLL](#) files from within your project.

With your project open, in the **Solution Explorer** right click the project and click **Add Reference....**



Click the **Browse** tab and browse to the [class library DLL files](#) (located in the **\Bin** directory where you have [installed the library](#))



Once the reference has been added you can refer to the various classes from within your project by adding an import directive for the library **namespace**:

C#

using DidiSoft.Pgp;

VB.NET

Imports DidiSoft.Pgp

Setup

The library installer will [create a folder](#) containing the [library DLL files](#), documentation and examples.

When we install a **production version** or an **updated version** over an **existing installation** (for example the **trial version**), it will usually **overwrite** the [library DLL files](#) (which is Ok, as we assume that the production version is preferred over the trial and the updated is better than the old one).

If we wish to **keep the old version files**, we have to **manually copy** them somewhere or **choose a different installation folder** during the setup process.

Deployment

The class library's runtime is royalty free which means it may be freely distributed with your application.

The only files that should be distributed with your application are the [class library DLL files](#).

Alternative way to reference the [library DLL files](#) is to place them in the **GAC** (Global Assembly Cache).

To install them in the GAC you can use the command line tool **gacutil** that ships with the .NET Framework SDK.

Below are shown sample executions of gacutil that install the DLL files:

```
gacutil.exe /i DidiSoft.Pgp.dll
gacutil.exe /i BouncyCastle.CryptoExt.dll
```

Switch from Trial to Production version

Please follow these instructions in order to **switch** from the **trial /evaluation/ version** of the library to the **production version** after a successful purchase. The same process should be applied also for **upgrade** to a newer version of the library.

You will receive an order confirmation email with details on how to download the production copy of the library. After installation of the production version there will be **\Bin** folder where the production [library DLL files](#) are located. This is usually the same [location](#) where the trial version files were installed

Library Files in the application folder (Default setup)

If your project is already referencing the trial version DLL files, then please first delete the old references.

Add a [reference](#) to the [library DLL files](#) located in the **\Bin** folder of the [production installation directory](#), **rebuild and redeploy** your application.

Library Files installed in the GAC

If you have **installed** the evaluation [library DLL files](#) in the **Global Assembly Cache (GAC)**, please first **remove** them from there, and afterwards **install** the production [library DLL files](#)

To **check** are there any versions of the DLL files in the GAC you can use the command line tool **gacutil** that ships with .NET Framework SDK

```
gacutil.exe /l DidiSoft.Pgp
gacutil.exe /l BouncyCastle.CryptoExt
```

To **remove** the already installed evaluation version you can type:

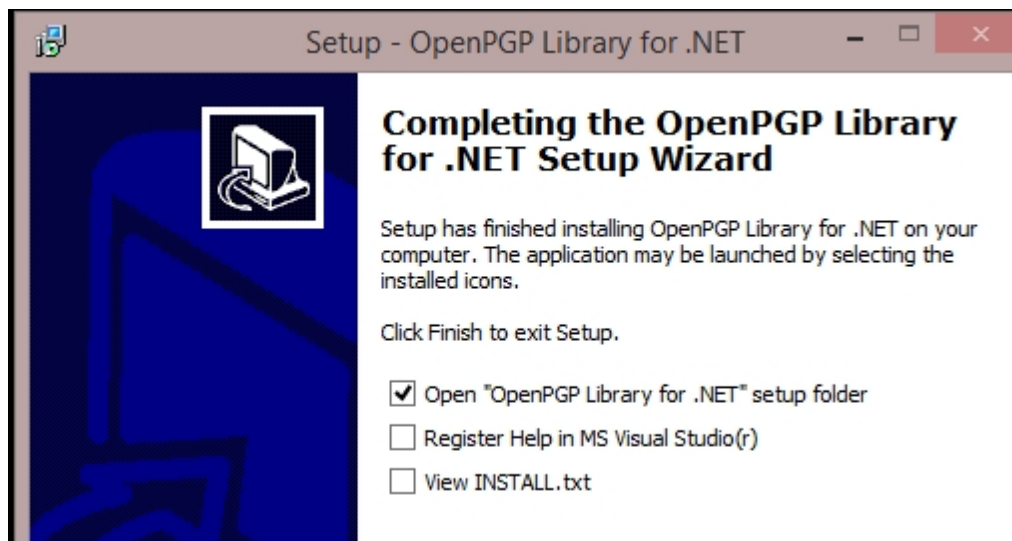
gacutil.exe /u DidiSoft.Pgp
gacutil.exe /u BouncyCastle.CryptoExt

The same commands can be use to [upgrade to a newer production version](#).
If you have any trouble with the replacement of the evaluation copy with the production one, do not hesitate to [contact us](#).

Register the help in MS Visual Studio

In previous versions of the installer the help was automatically registered in the help system of the Microsoft Visual Studio IDE.

As of version 1.7.11.8 this is offered as an optional step at the end of the setup wizard:



If you have skipped this offering, you can manually run it by executing the batch file:
[\[setup folder\]\Help\Register_Help_VS.bat](#)

NuGet

Trial Version

PM> Install-Package DidiSoft.Pgp.Trial

Production Version

Private NuGet Repository: (see below how to register it in Visual Studio)

https://repo.didisoft.com/repository/nuget/

The production NuGet feed is only for existing customers. It is protected with username and password. Below are the details that you have to use in order to access the production feed.

Username: your email address used to access DidiSoft customers' section.

Password: your email address used to access DidiSoft customers' section.

Uninstall the trial package before installing the production

PM> Uninstall-Package DidiSoft.Pgp.Trial

Install the production package

PM> Install-Package DidiSoft.Pgp

Registering the Private NuGet feed in Visual Studio

In Microsoft Visual Studio *select Tools > NuGet Package Manager > Package Manager Settings*

In the tree view select Package Sources - and add new source with details:

Name: *DidiSoft.Pgp*

Source: *https://repo.didisoft.com/repository/nuget/*

Upgrade

General rules

The **upgrade process** consists of replacing the [library References](#) in your project to point to the new [DLL files](#) and **rebuilding** and **redeploying** your application(s).

In most of the cases the upgrade process will be smooth without any code changes as we try to keep **backward compatibility**.

If you encounter compile time errors caused by changes which are not mentioned below, please do not hesitate to [contact our technical support](#) for assistance.

From version 1.3

Replace each occurrence of

```
DidiSoft.OpenSsl.X509.Certificate.ToX509Certificate2()
```

with

```
DidiSoft.OpenSsl.X509.Certificate.ToX509Certificate()
```

Examples

All the examples below are available online at our web site:

<http://www.didisoft.com/examples/net-pgp-example/>

and as [Visual Studio solutions](#) located in the **Examples** folder within the library ZIP package.

Online

Introduction

[Introduction to OpenPGP](#)

[Setup](#)

[Getting started with the Library](#)

[Exception handling guidelines](#)

Main functions

[Encrypting](#)

[Decrypting](#)

[Signing](#)

[Verifying](#)

[Signing and Encrypting in one pass](#)

[Decrypting and Verifying in one pass](#)

[Clear text signing](#)

[Detached signing](#)

Keys and KeyStore

[KeyStore basics](#)

[Properties of a key](#)

[Generating RSA keys](#)

[Generating DH/DSS keys](#)

[Generating ECC keys \(New!\)](#)

[Exporting from a KeyStore](#)

[Importing into a KeyStore](#)

[Changing key's password](#)

[Deleting key from a KeyStore](#)

Analyzing OpenPGP archives

[Analyzing OpenPGP archives](#)

OpenPGP Mail

[Sending OpenPGP Email messages](#)

Preferences

[Set preferred cypher](#) (symmetric key algorithm)

[Set preferred compression](#)

[Set preferred hashing](#)

[Content type](#)

[ASCII armored header](#)

Key revocation

[Key revocation basics](#)
[Direct key revocation](#)
[Using a revocation certificate](#)
[Using a designated revoker](#)

Logging

[Logging internal verbose information](#)

WinRT development

[WinRT development guide](#)

Miscellaneous

[Creating SFX executables](#)
[Encrypting with a key and an additional password](#)
[Using X.509 certificates as OpenPGP keys](#)
[JPEG photo images in OpenPGP keys](#)

Haven't found an example or detail you were looking for?
Just [drop us a line](#) and we will send you a code snippet and a thorough explanation.

Example Applications

The class library ships with an example applications located in the **Examples** folder in the distribution package. You will need to rebuild the applications prior to running them. Below is a list with a short description of these applications:

PGPLibExampleCS – console examples in C#

PGPLibExampleVB – console examples in VB.NET

FileToolCS – WinForms C# application demonstrating the basic functions for OpenPGP archives

FileToolVB – WinForms VB.NET application demonstrating the basic functions for OpenPGP archives

KeyToolCS – WinForms C# application demonstrating operations with OpenPGP keys and the library KeyStore object

KeyToolVB – WinForms VB.NET application demonstrating operations with OpenPGP keys and the library KeyStore object

Appendix

Common Exceptions

As of version 1.7.5 the **DidiSoft.Pgp.Exceptions** namespace contains a full set of sub classes of **DidiSoft.Pgp.PGPEXception** that indicate the cause of the error.

Exporting keys from a GnuPG keystore

List key's in GnuPG keystore:

```
gpg --list-keys
```

Generate OpenPGP key pair

```
gpg --gen-key
```

Export Public key

```
gpg --export my_key -o my_public_key.gpg
```

Export Private key

```
gpg --export-secret-key my_key -o my_secret_key.gpg
```

Licensing

The End User **License Agreement** for DidiSoft OpenSSL Library for .NET / 1.4 is located in the folder **\License** in the file **EULA.rtf**.

The production version of the library can be distributed **royalty free without limitation** in binary form.

BouncyCastle license

DidiSoft OpenSSL Library for .NET / 1.4 uses the BouncyCastle cryptographic library which comes with a separate license: **The BouncyCastle License** which is contained in the folder **License\BC**. The BouncyCastle License is an adaptation of the **MIT X11 License** and refers only to the BouncyCastle library and it's documentation and gives you the right to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the BouncyCastle library and it's documentation.

Support

Technical support

To receive general information or **technical support**, please contact us at support@didisoft.com.

Sales

For questions related to **sales**, volume licensing, or OEM licensing, please contact us at sales@didisoft.com.

Updates

If you have purchased the library you can access our [Customers' Portal](#) where you can **download new versions**.

Newsletter

To receive product **update news** you can subscribe to our [Newsletter](#)

For further information, visit us at www.didisoft.com

If you have any **ideas, wishes, questions** or **criticism**, don't hesitate to contact us. We will be glad to hear from you.