# AMYUNI Technologies

# Amyuni PDF Creator for .NET

*For PDF and XPS*

## Quick Start Guide

Updated 17 June 2021

# Contents

## Legal Information

## Acknowledgments

This software uses the deflate algorithm developed by Jean-loup Gailly (jloup@gzip.org) and Mark Adler (madler@alumni.caltech.edu). This software is also based in part on the work of the Independent JPEG Group and on parts of the FreeType library.

The Software includes the Open Source Tesseract library in binary format. The Tesseract library source-code can be obtained from http://code.google.com/p/tesseract-ocr/ and is licensed under the Apache 2.0 license. The Tesseract source-code is not included with the Amyuni distribution.

# Copyright Notice and License Agreement

AMYUNI CONSULTANTS – AMYUNI TECHNOLOGIES
DEVELOPER/APPLICATION LICENSE AGREEMENT
FOR OCR MODULE PRODUCTS

**NOTICE TO USER:**
THIS IS A CONTRACT. BY INSTALLING THIS SOFTWARE YOU ACCEPT ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT.

This AMYUNI ("Amyuni") Developer License Agreement accompanies all Amyuni OCR Module product and related explanatory materials ("Software"). The term "Software" also shall include any upgrades, modified versions or updates of the Software licensed to you by Amyuni.

Please read this Agreement carefully. You will be asked to **accept** this agreement and continue to install or, if you do not wish to accept this Agreement, to **decline** this agreement, in which case you will not be able to use the Software.

Upon your acceptance of this Agreement, Amyuni grants to you a perpetual but nonexclusive license to use the Software, provided that you agree to the following:

**1. Use of the Software**. You may include the Software as part of your own applications to install it on your client's machine at the same time as your own applications. The applications can be either in-house or for external distribution. The Software can be used on either workstations or servers without limitations on the number of users. One Application license is required for every application that includes the Software. **The activation code that is provided to you by Amyuni should be kept confidential and not be revealed to end-users, even in the case where the developer's products are sub-licensed to other developers.**

**2. Copyright and Trademark Rights**. The Software is owned by Amyuni, and its structure, organization and code are the valuable trade secrets of Amyuni. The Software also is protected by Canadian Copyright Law and International Treaty provisions. This Agreement does not grant you any intellectual property rights in the Software.
The Software includes the Open Source Tesseract library in binary format. The Tesseract library source-code can be obtained from http://code.google.com/p/tesseract-ocr/ and is licensed under the Apache 2.0 license. The Tesseract source-code is not included with the Amyuni distribution.

**3. Restrictions.** You agree not to modify, adapt, translate, reverse engineer, decompile, disassemble or otherwise attempt to discover the source code of the Software. You may not sell the product as a standalone application. You will make all necessary steps to prevent users of your applications to use the Software other than from inside your applications. These steps are described in this manual.

**4. No Warranty**. The Software is being delivered to you AS IS and Amyuni makes no warranty as to its use or performance. AMYUNI AND ITS SUPPLIERS DO NOT AND CANNOT WARRANT THE PERFORMANCE OR RESULTS YOU MAY OBTAIN BY USING THE SOFTWARE OR DOCUMENTATION. AMYUNI AND ITS SUPPLIERS MAKE NO WARRANTIES, EXPRESS OR IMPLIED, AS TO MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMYUNI OR ITS SUPPLIERS BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INCIDENTAL OR SPECIAL DAMAGES, INCLUDING ANY LOST PROFITS OR LOST SAVINGS, EVEN IF AN AMYUNI REPRESENTATIVE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY THIRD PARTY. Some states or jurisdictions do not allow the exclusion or limitation of incidental, consequential or special damages, or the exclusion of implied

warranties or limitations on how long an implied warranty may last, so the above limitations may not apply to you.

BY INSTALLING, USING OR DISTRIBUTING THIS PRODUCT, YOU INDICATE ACCEPTANCE OF THE FOREGOING AGREEMENT.

## Description of Various Modules

### acPDFCreatorLib.Net.dll

This is the managed code library that contains all the classes needed to create and process PDF files. This DLL should be copied to the main application's directory. The namespace for all the classes in this library is Amyuni.PDFCreator.

### acPDFCreator.Net.dll

This DLL contains a visual .NET control that can be placed on a form to view, print or process PDF files. This DLL also contains additional controls such as the status bar. The namespace for all the classes in this library is Amyuni.PDFCreator.PDFCreator.

### xmllite.dll

This module contains the XML parser needed to process XPS documents. This module is needed only if XPS support is required and can be skipped otherwise.

## General Operation

The Amyuni PDF Creator .NET was designed for three main purposes:

- Viewing, editing, annotating and printing PDF documents
- Creating and filling out PDF forms
- Creating reports, forms and general documents directly in PDF, RTF. HTML, Excel, JPEG, and TIFF formats

The library and controls that are part of the PDF Creator .NET can be integrated into most development environments to provide the final user with advanced PDF document management capabilities.

The PDF Creator control can be in one of four modes:

- Design mode.
- Run or compiled mode.
- Annotation mode.
- Print preview mode.

When a blank document is first created, the control is in design mode. The user or developer can add objects to the document, delete objects, modify object properties and do all editing operations allowed by the PDF Creator control.

When an existing PDF document is opened, two things can happen:

1. The PDF document was generated by the PDF Creator and contains all design information generated by the control. In this case, the document is opened in the same state as when it was saved, i.e., either Design, Run, Annotate or PrintPreview modes.

2. The PDF document was generated by another tool such as the Amyuni PDF Converter. In this case, the document is opened immediately in Run mode. The document can only be read or printed and if the user has enough rights on the document, the user can switch the document to annotation or design mode and edit it using the PDF Creator interface.

The documents created by the PDF Creator can contain fields with formulas or data coming from a database. Compiling the document instructs the PDF Creator to compute those formulas and fetch information from the database. Once compiled, the document switches to **Run** mode and only the fields or objects defined as Editable can be modified in Run mode.

> *Note:* Microsoft Visual C++ 2017 Redistributable Package (x86) should be installed on systems that don't have version 2017 when deploying their applications.

## Adding the PDF Creator .NET Library to Your Application

The first step to start using the PDF Creator library with your application is to reference the **acPDFCreatorLib.Net.dll** assembly.

To add the new reference:

1. Open the solution explorer window

2. Right click on references and select Add new reference.
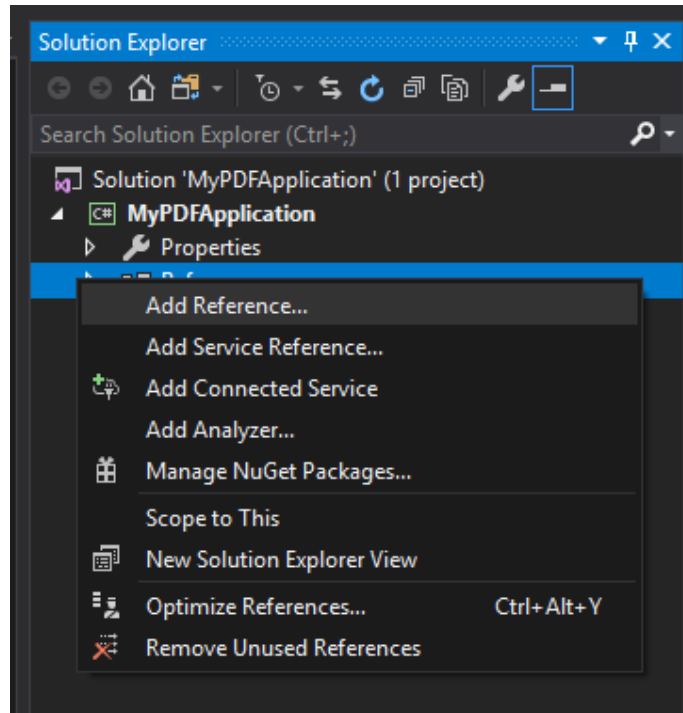


**Figure 1:** Adding a reference in the Solution Explorer Window

This will result in having the DLL copied to your application's executables directory.
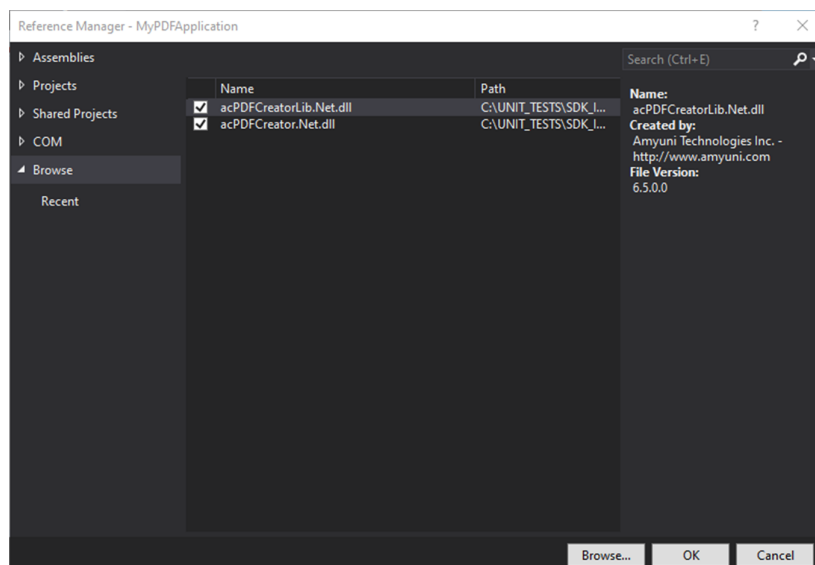
Once the reference to the assembly is added, you can use it in any of your modules by adding:

```
using Amyuni.PDFCreator;
```

The assembly needs to be initialized before any object is referenced. The assembly also needs to be uninitalized before exiting your application to free any internal resources used by the library.

Initialization is done by calling:

```
acPDFCreatorLib.Initialize ();
```

Before terminating the application, the developer should call:

```
acPDFCreatorLib.Terminate ();
```

## Inserting the PDF Creator Control into a Project

This is specific to each development environment, but the procedure is very similar in all cases. The PDF Creator component should be added to the components toolbox as follows:

1. Right-click "Components".

2. Select Add/Remove items.
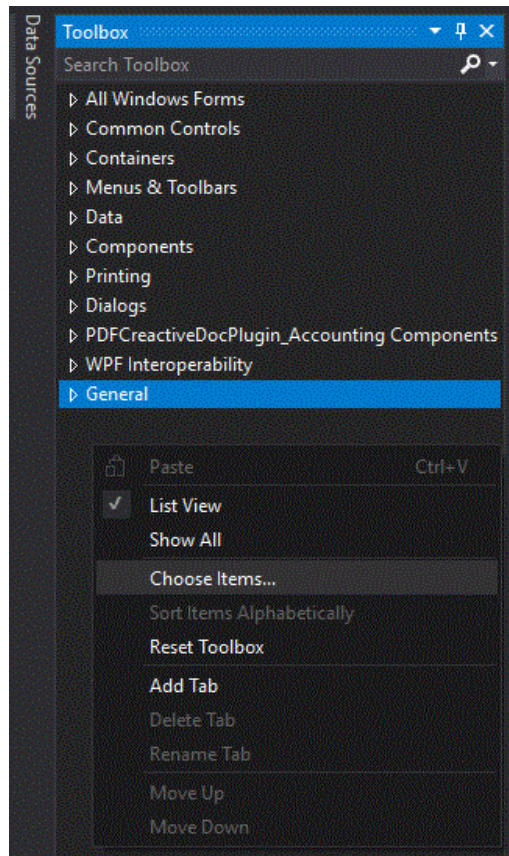
**Figure 3:** Adding PDFCreator to the components toolbox

The Add .NET Framework Components dialog box appears as follows:



*Figure 4:* Selecting PDFCreator in the .NET Framework Components

Browse for **acPDFCreatorLib.Net.dll**. The control shows in the list of insertable .NET components as "PDFCreator". Once added to the toolbox, the component appears in the components toolbar and can be inserted on a form:
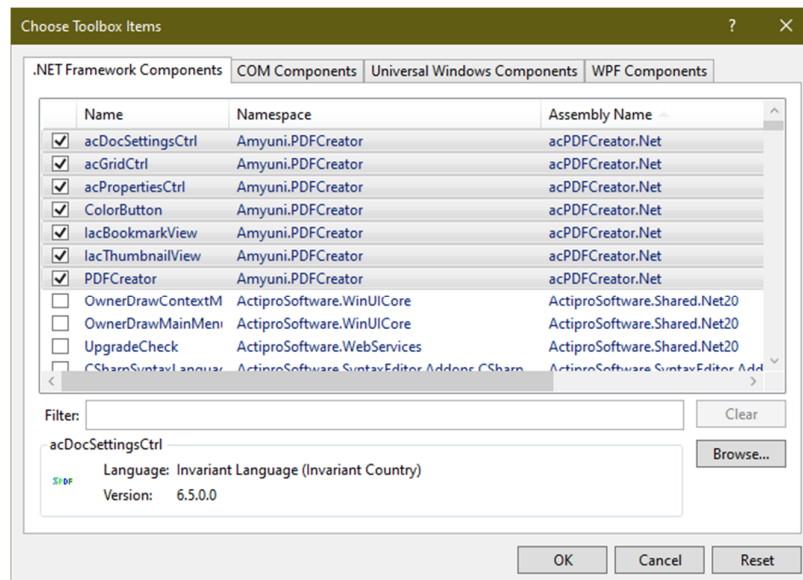
### Setting the Licensing Information

Before any operation can be done on the PDF Creator library or control, the library or the control needs to be activated using the **SetLicenseKey** method as shown below.

A sample code for using the PDF Creator control to open a PDF file for viewing or editing:

```
// All Amyuni classes used here are defined in the Amyuni.PDFCreator
        namespace

// Initialize the Library
    acPDFCreatorLib.Initialize ();

// Set the License key
// acPDFCreatorLib.SetLicensekey (System.String Company,
        System.String LicKey)
    acPDFCreatorLib.SetLicenseKey ("Amyuni Tech.",
        "07EFCD10001…7857EC1193FA");


    Amyuni.PDFCreator.PDFCreator pdfCreator;
    PdfCreator = new Amyuni.PDFCreator.PDFCreator ();

//open a pdf document from file
        System.IO.FileStream file1 = new System.IO.FileStream
        ("Page.pdf", FileMode.Open,
            FileAccess.Read, FileShare.Read);
    IacDocument document = pdfCreator.Document;
    document.Open (file1, "");

//switch to design mode before adding objects to the document
    document.ReportState = IacReportState.acReportStateDesign;

//create a rectangular frame to add to the document
    IacPage page = document.GetPage (1);

    Amyuni.PDFCreator.IacObject frame = page.CreateObject
        (IacObjectType.acObjectTypeText,"Frame 1");



//position the frame
        frame.AttributeByName ("Left").Value = 200;
        frame.AttributeByName ("Top").Value = 600;
        frame.AttributeByName ("Right").Value = 2000;
        frame.AttributeByName ("Bottom").Value = 800;
        frame.AttributeByName ("Text").Value = "Hello";

// save modified document to a new file (file2)
 System.IO.FileStream file2 = new System.IO.FileStream ("test3.pdf",
        FileMode.Create,
            FileAccess.Write, FileShare.Read);
 document.Save (file2, IacFileSaveOption.acFileSaveView);
        file2.Close ();

// terminate library to free resources
```

```
        acPDFCreatorLib.Terminate ();
```

Here is the same sample in C# for using the PDF Creator library as opposed to the PDF Creator control:

```
// initialize library and set license key
   acPDFCreatorLib.Initialize ();
   acPDFCreatorLib.SetLicenseKey ("Amyuni Tech.",
       "07EFCDAB0100...009EE2F79");

// Open a PDF document from file
        System.IO.FileStream file1 = new System.IO.FileStream
       ("test.pdf", FileMode.Open,

       FileAccess.Read, FileShare.Read);

        IacDocument document = new IacDocument (null);
        document.Open (file1, "");

// switch to design mode before adding objects to the document
        document.ReportState = IacReportState.acReportStateDesign;

// create a rectangular frame to add to the document
        Amyuni.PDFCreator.IacObject frame =
   document.GetPage (2).CreateObject (IacObjectType.acObjectTypeText,
       "Frame 1");

// position the frame
   frame.AttributeByName ("Left").Value = 200;
        frame.AttributeByName ("Top").Value = 600;
        frame.AttributeByName ("Right").Value = 2000;
        frame.AttributeByName ("Bottom").Value = 800;
        frame.AttributeByName ("Text").Value = "Hello";

// save modified document to a new file
        System.IO.FileStream file2 = new System.IO.FileStream
       ("test3.pdf", FileMode.Create,

       FileAccess.Write, FileShare.Read);
        document.Save (file2, IacFileSaveOption.acFileSaveView);

// terminate library to free resources
   acPDFCreatorLib.Terminate ();
```

## PDF Creator .NET Library Architecture

The main entry point to the library is the **IacDocument** class. This class provides the developer the methods needed to create a blank document, open an existing document, accessing various parts of the document, saving **IacDocument** implements various interfaces:

- **IacPrint** contains all the methods related to printing a document.
- **IacUndo** contains methods related to undoing and redoing the latest operations.
- **IacImportExport** contains the methods related to importing and exporting documents from/to various formats.

**IacDocument** contains an array of pages, each represented by the **IacPage** class. Each page contains an array of objects represented by the **IacObject** class. Any object contained in a page, such as text, lines, images, annotations is represented by an **IacObject** object. **IacObject** contains a list of attributes represented by the **IacAttribute** class. The developer communicates with objects by reading or setting various object attributes. Here's an example of changing the text for a specific object:

```
IacAttribute text = object.AttributeByName ("Text")
text.Value = "Change the text of this object"
```

**IacDocument** and **IacPage** are also derived from **IacObject**. This enables the developer to read and write various attributes that are not directly available through the **IacDocument** or **IacPage** classes. These attributes are detailed in the coming sections and include various things such as the document title or subject.

**IacPageFormat** contains the entire page formatting information such as paper size and orientation. Both **IacDocument** and **IacPage** contain a single instance of **IacPageFormat**. The instance contained by **IacDocument** represents the default page format that is used when a new page is created. The instance contained in **IacPage** represents the page format for each specific page.

## Links to Support and Documentation:

If you have any questions or problems with our products, the following resources are available to you through our web site:

**Online Documentation:**
https://www.amyuni.com/WebHelp/Developer_Documentation.htm#index.htm

**Frequently Asked Questions:**
https://www.amyuni.com/forum/viewforum.php?f=18

**Technical Notes:**
https://www.amyuni.com/en/resources/technicalnotes/

**User forum:**
https://www.amyuni.com/forum/index.php

**Posting questions to our technical support staff:**
https://www.amyuni.com/en/support/getsupport/

We also provide some additional tools that can be downloaded free of charge and used with the PDF Creator product. These tools are available at:
http:s//www.amyuni.com/en/resources/freetools/