

Amyuni PDF Creator for .NET

For PDF and XPS

Version 4.0 Professional

Quick Start Guide

Updated March, 2009

Contents

Legal Information.....	3
Acknowledgments	3
Description of the Various Modules	4
acPDFCreatorLib.Net.dll	4
acPDFCreator.Net.dll.....	4
xmllite.dll.....	4
General operation.....	5
New in Version 4.0.....	6
PDF Creator New Features Include:	6
Printing Features Include:	6
Performance Features Include:	7
Graphics Features Include:	7
Adding the PDF Creator .NET Library to Your Application	8
Inserting the PDF Creator Control into a Project.....	10
Setting the Licensing Information.....	11
PDF Creator .NET Library Architecture	13
Links to Support and Documentation:	14
Online Documentation:	14
Frequently Asked Questions:.....	14
Technical Notes:.....	14
User forum:	14
Posting questions to our technical support staff:.....	14

Legal Information

Information in this document is subject to change without notice and does not represent a commitment on the part of AMYUNI. The software described in this document is provided under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of the agreement. It is against the law to copy the software on any medium except as specifically allowed in the license or nondisclosure agreement.

The licensee may make one copy of the software for backup purposes. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems, for any purpose other than the licensee's personal use, without express written permission of AMYUNI.

Copyright 2001-2009, AMYUNI Consultants – AMYUNI Technologies. All rights reserved.

Amyuni and the Amyuni logo are trademarks of Amyuni Technologies Inc.

Microsoft, the Microsoft logo, Microsoft Windows, Microsoft Windows NT and their logos are trademarks of Microsoft Corporation.

All other trademarks are the property of their respective owners.

PDFCreactiveDoc.exe is provided as a sample application for Developers, it can not be distributed with the Developers' application. The source-code for this executable can be requested by contacting support@amyuni.com

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.

Description of the Various Modules

The Amyuni PDF Creator .NET product is composed of two modules that can be integrated and distributed under the developer license agreement.

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.

Version 2.0 of the PDF Creator added the Annotation mode as one of the PDF Creator modes of operation. In Annotation mode, the contents of the original document cannot be modified; the user can only add or modify PDF annotations such as Text, Line, Highlighter or Sticky note annotations.

Note: Microsoft Visual C++ 2005 Redistributable Package (x86) should be installed on systems that don't have version 2005 when deploying their applications.

New in Version 4.0

PDF Creator New Features Include:

- The loading, creation, and display of layered PDF files.
- CJK file support, with the ability to specify the location of custom character maps (CMAPs.)
- Text anti-aliasing for better on-screen document legibility (requires font installation on client PC).
- Internal PDF file structure access through comprehensive class architecture.
- Improved text editing operations such as font size adjustment to fit text into a field.
- Improved text highlighting capabilities.
- The addition of custom attributes to existing PDF Creator objects.
- Area zooming and panning capabilities.
- The ability to add custom metadata to the XMP metadata stream of a PDF.
- Exporting capabilities of a predefined sequence of pages instead of the entire document.
- Triggering of a number of new events such as progress or keyboard events.
- Improved bookmark operation such as drag-and-drop capabilities.
- Improved thumbnail control with a number of new operations and events.
- Ability to move one or multiple pages from one document to another.
- Support for a wider range of PDF files such as the PDF 1.7 format and the compressed XRef table format.
- The addition and extraction of attachments to and from PDF files.
- Exporting capabilities of PDF files into XAML in order to view from a Web page with Silverlight controls.
- Updated PDF/A engine ensures Acrobat 9 compatibility.
- Logging capabilities to ensure detailed detection and warning of errors during the processing of PDF files.
- Resaving capabilities of PDF files using PDF 1.4, 1.5, or PDF/A formats.
- The generation of PDF/A-1a and PDF/A-1b files with embedding of custom XMP metadata.
- Support for Unicode strings for the selection of digital signatures.

Printing Features Include:

- The ability to print a single page to multiple sheets or multiple pages to a single sheet.
- The option to rotate and scale the page during printing.
- Selectable paper tray options depending on the media type assigned to a page.
- A "Print Preview" report state to preview documents before printing.

Performance Features Include:

Improved file processing speed of PDF files that contain large graphics.

Improved file loading capabilities from network drives.

Reduced memory consumption during the processing of large numbers of pages.

Reduced output file size through better page compression and reduced embedded font size.

Improved performance and reliability of Merge operations.

Graphics Features Include:

JPEG2000 compression in addition to the existing image decompression.

Delayed loading of JPEG and PNG images until they are required. This delay improves performance and memory usage during PDF processing.

Improved performance of graphics library for faster decompression of CCITT and JBIG2 images.

Improved support for transparencies and alpha-blending techniques.

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.

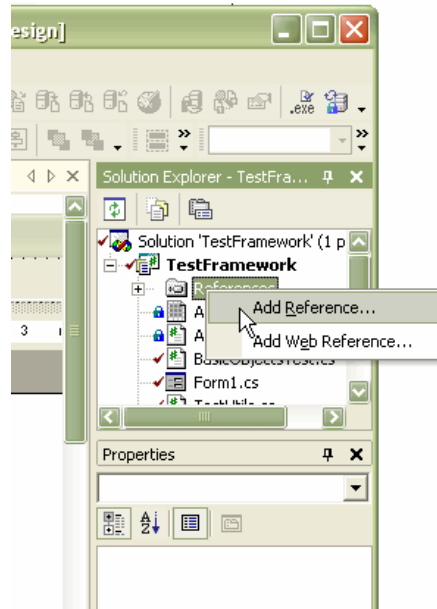


Figure 1: Adding a reference in the Solution Explorer Window

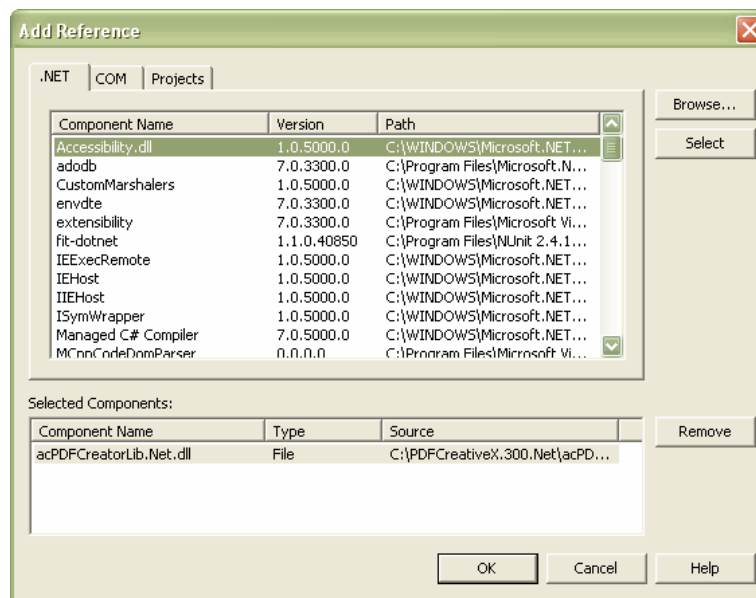


Figure 2: Microsoft Visual Studio Add Reference Dialog

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 uninitialized 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.

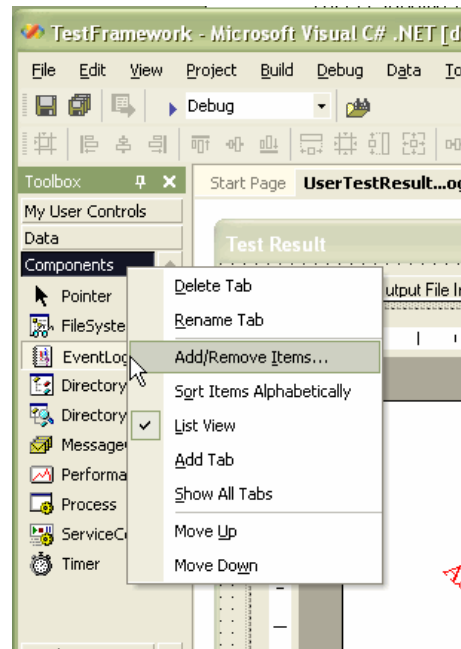


Figure 3: Adding PDFCreator to the components toolbox

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

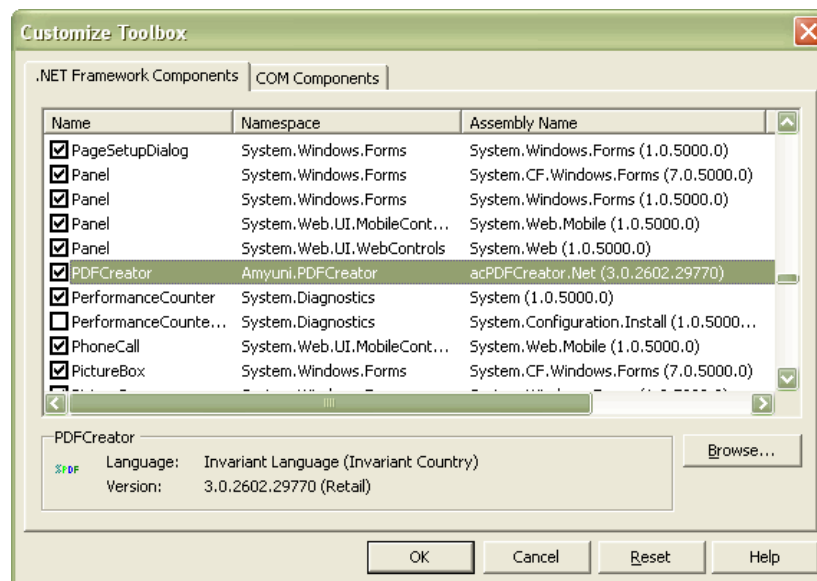


Figure 4: Selecting PDFCreator in the .NET Framework Components

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.", "07EFCDA0100...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 **lacDocument** 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 **lacDocument** implements various interfaces:

lacPrint contains all the methods related to printing a document.

lacUndo contains methods related to undoing and redoing the latest operations.

lacImportExport contains the methods related to importing and exporting documents from/to various formats.

lacDocument contains an array of pages, each represented by the **lacPage** class. Each page contains an array of objects represented by the **lacObject** class. Any object contained in a page, such as text, lines, images, annotations is represented by an **lacObject** object. **lacObject** contains a list of attributes represented by the **lacAttribute** 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"
```

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

lacPageFormat contains the entire page formatting information such as paper size and orientation. Both **lacDocument** and **lacPage** contain a single instance of **lacPageFormat**. The instance contained by **lacDocument** represents the default page format that is used when a new page is created. The instance contained in **lacPage** 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:

http://www.amyuni.com/WebHelp/Developer_Documentation.htm#index.htm

Frequently Asked Questions:

<http://www.amyuni.com/forum/viewforum.php?f=18>

Technical Notes:

<http://www.amyuni.com/en/support/technotes.html>

User forum:

<http://www.amyuni.com/forum/index.php>

Posting questions to our technical support staff:

<http://www.amyuni.com/en/support/techsupport.html>

We also provide some additional tools that can be downloaded free of charge and used with the PDF Converter product. These tools are available at:

<http://www.amyuni.com/en/support/downloads.html>