



## Getting Started Guide Create PDF Files in ASP.NET(VB) with PDFBuilderASP

### Introduction

This document is intended to help a new user of the Ciansoft PDFBuilderASP server component to get started writing a simple ASP.NET application to create a PDF file on a web server.

If you are looking for a reference to the various functions provided in PDFBuilderASP, a [comprehensive user manual](#) is available either in the downloaded ZIP file that contains either the full or trial versions of the component, or on our web site.

These instructions assume the trial version is being used. Instructions for the full purchased version are identical except that file names and class names are different, omitting the word 'Trial'.

It is normal practice to develop and test an application on a test server before deployment. This test server can simply be a PC with IIS installed and does not need to have a server operating system. These instructions assume that a standard PC operating system such as Windows 2000, XP or Vista is being used as the test environment.

### Installation of the Component

Before a script can be run that uses this component, the library file PDFBuilderASPTrial.dll must be copied to the PC or server, registered with the operating system, and appropriate permissions must be set so that the library can be accessed and files can be read or written as required.

#### Copying and Registering

Extract the DLL file from the downloaded ZIP file and copy it to the server. The file can be placed in any location, but standard practice is to put DLL files in the System32 directory under the system root (the Windows directory on the C: drive).

To register the file, open a command prompt window. On Windows 2000 or XP this is done from the Run option on the Start menu, entering cmd, or from Start/Programs/Accessories. On Windows Vista, the command prompt must be run as an administrator or else the registration will fail, so from the Start menu go to Accessories, right mouse click on 'Command Prompt' and click 'Run as Administrator'.

At the command prompt, enter the following:

```
regsvr32 C:\Windows\System32\PDFBuilderASPTrial.dll
```

Note that the name of the system root folder might be different, e.g., WINNT, or another location might be used for the file. Modify the path above as appropriate.

If the registration is successful, a message will be displayed to confirm this. Click OK and then close the command prompt window.

#### Setting Permissions

In order to use PDFBuilderASP in a server application, the Internet Guest User must have Read & Execute permission on the DLL file. Right mouse click on the file PDFBuilderASPTrial.dll and select Properties. Click on the Security tab.

In Windows 2000 or XP click the 'Add...' button and select the Internet Guest User account which will be called 'IUSR\_' followed by the machine name. Click 'Add', then 'OK'. Make sure that the 'Read & Execute' checkbox is checked for the Internet Guest User. If it is not, check it and click 'Apply'. Finally click 'OK'.

In Windows Vista click the 'Edit' button and then the 'Add...' button. Type IUSR in the box labelled 'Enter the object names to select' and click 'Check Names'. IUSR will now be underlined confirming that the account was found. Click 'OK'. Make sure that the 'Read & Execute' checkbox is checked for the IUSR account. If it is not, then check it and click 'Apply'. Finally click 'OK' and 'OK' again.

In addition to setting permissions for the DLL file, it is also necessary to set Read permissions for the Internet Guest User on any files that the component will need to read such as images or fonts, and Write permissions on any directories where a PDF file is to be saved using the SaveToFile command.

## Creating a Blank PDF File

To begin using PDFBuilderASP, we will first write a script to create a blank PDF file containing a single page.

On the server root directory, create a new text file using any text editor, e.g., Notepad, and save the file with a file extension of .aspx. The server root is usually the directory C:\inetpub\wwwroot. Note that on Windows Vista, the system may not allow a new file to be saved in this location. In that case, save the file elsewhere and then cut and paste it to the server root.

Write the following lines of text in the file and save it:

```
<%@ Page Language="VB" Debug="True" %>

<%
Dim objPDF = Server.CreateObject("PDFBuilderASPTrial.PDFSvrDoc")
objPDF.DefaultPageSize = 1
objPDF.Units = 3
objPDF.AddPage(0)

Dim OutArray As Array = objPDF.StreamData
Dim ByteArray(OutArray.Length - 1) As Byte
Array.Copy(OutArray, ByteArray, OutArray.Length)
Response.ContentType = "application/pdf"
Response.AddHeader("Content-Disposition", _
    "inline; filename=Test.pdf")
Response.BinaryWrite(ByteArray)
%>
```

The first line is a page header, which is not essential but is good programming practice. This header tells the server which language is being used (VB) and that debugging information should be provided if an error occurs. The script proper starts after this line.

The first line of the script creates an instance of the PDFSvrDoc class in memory. This represents a single PDF document that will be built.

The next two lines set the page size to A4 and the units of measure to millimetres. The next line adds a single page, with the zero indicating that the page is added at the end of the document. For this example with a single page, the location is not important.

The remaining lines create the PDF document as a stream of data and send it to the browser. This sequence of code looks complicated, but you will never need to modify it apart from changing the file name (currently "Test.pdf") if required.

This code can be tested by opening a browser on the same machine that is being used as the server, and using the name 'localhost' to refer to the server root. For example, if the file is called pdf\_test.aspx, then 'http://localhost/pdf\_test.aspx' should be entered in the browser as the URL. Provided that PDF viewing software such as Adobe Reader is installed to allow PDF files to be displayed in the browser, a blank A4 page will be seen.

## Adding Text

Next we will add some text to the blank document. Text is written in text blocks which are then added to pages using the ApplyResource command. Add the following code to the aspx file after the AddPage command and before the code sequence starting with 'Dim OutArray...':

```
Dim Text1 As Integer = objPDF.CreateText  
objPDF.WriteText("This is a line of text.")  
Dim TextIndex1 As Integer = objPDF.ApplyResource(1, Text1)
```

Save the file and call the script from the browser again. This time the page is shown with the line of text in the top left corner. Next we will add some further lines of code to make the text larger and move it to a different position on the page. Add the following line after the CreateText command:

```
ObjPDF.TextSize = 20
```

This increases the size of the text. The next line should be added after the ApplyResource command:

```
ObjPDF.Locate(1, TextIndex1, 20.0, objPDF.PageHeight(1) - 20.0)
```

This positions the text 20mm down from the top of the page and 20mm in from the left side.

## Further Options

These instructions have covered the basics of installing PDFBuilderASP and writing an ASP.NET script to create a simple PDF file and display it in a web browser. There are many further options available to format text and to add images and graphics. For additional information, [consult the user manual](#) and [visit our web site for code examples](#).