

Gnostice PDFOne

Reviewed by Craig Murphy

During the latter quarter of 2004, I reviewed Gnostice's PDF creation tools for use in Win32 Delphi. I was very pleased with their offering, so now I am looking at their new .NET PDF creation product.

Setting the Scene

In my November 2004 review of the Win32 products, I was quoted as saying:

“If you have been looking for a PDF creation and management component suite, you will be hard pushed to find anything to rival Gnostice's eDocEngine and PDFtoolkit bundle”.

Given the calibre of the Gnostice products to date, I see no reason why their .NET products shouldn't be any different. Indeed, I have a suspicion that PDFOne will rival eDocEngine and PDFToolkit...and then I'll be able to say, “the best just got better”. Read on and find out if that's true or not.

What is PDFOne?

In Gnostice's own words:

“Gnostice PDFOne .NET is a 100% .NET component library for developers to implement PDF based software solutions. PDFOne .NET provides a rich set of APIs to create, manipulate and organize PDF documents, process PDF forms and perform other PDF document related tasks from within your .NET applications.”

There's little doubt that PDF files are now a way of life. Virtually every commercial website that offers downloadable documents does so using PDFs. With the profusion of high-quality, low-cost tools such as eDocEngine and PDFOne, even the non-commercial web sites are getting in on the act.

Getting Started

After I downloaded PDFOne (less than 5MB in size), I fired up a Virtual PC with Delphi 2006 installed. PDFOne installation was straightforward, although I was a little puzzled to be offered the choice of .NET 1.1 installation or .NET 2.0 installation. Could I not have both? Anyway, I went for the .NET 1.1 installation on the premise that Delphi 2006 supports .NET 1.1.

Next, I looked at the demonstration application. Previously, in eDocEngine, this was a VCL application. Now it's a .NET application. And since the language of choice in the .NET world appears to be C#, it's not surprising that Gnostice chose to write their demonstration using C#. There are a couple of good reasons for this. Firstly, Delphi 2005 and 2006 will happily load the demonstrations and perform the necessary tweaking to make the project files compatible – indeed, the C# demonstration loaded, compiled and ran inside Delphi 2006 flawlessly. Secondly, the demonstration is cross-platform, or cross-IDE: it will work with Delphi 200x and Visual Studio 200x.

Mind you, it is some demonstration: it comprises over 20,000 lines of code/comments...in a single file! Whilst it's great having everything under one roof, particularly from a user interface perspective (i.e. a lot of functionality in the same place), 20,000 lines is a lot to take in and manage! If I had one whinge, this would be it – I would like to see the demonstration split into a number of smaller, more manageable, demonstrations.

Elsewhere in this issue, you'll find me writing about the Express Editions of Visual Studio, e.g. Visual C#. I am pleased to report that the PDFOne works with Visual C# - I compiled and ran the demonstration application.

Programmatic PDF Creation

I think it's fair to say that we're all familiar with the concept of “printing to a PDF”, i.e. a virtual printer driver that appears to be a real printer but actually creates a PDF file instead of physically printing anything. Virtual printer drivers are all very well if you are able to create something that can then be printed. We might, for example, write an application that publishes its reports to Microsoft Word or Excel – from there we use the virtual printer drivers to create a PDF file. However, what if you want fine-grained control over the creation of the PDF file? What if you want to avoid the potential “file bloat” that the virtual printer drivers have been known to promote?

Well, that's where Gnostice's PDFOne comes in. It offers full programmatic access to what is essentially the underlying Acrobat SDK (for more information, there's a link in the Links section at the end of this review).

Using PDFOne in Delphi 200x

PDFOne is a regular .NET citizen, so it should come as no surprise to learn that its core functionality is served up via a couple of assemblies. I had to browse to the Gnostice installation directory to find these, which isn't a problem (there are a couple of batch files in the installation directory that will add or remove the Gnostice assemblies from the Global Assembly Cache). I'm sure you know what Delphi's Add Reference dialog looks like, but I'm going to include it anyway – Figure 1, adding references to the Gnostice assemblies.

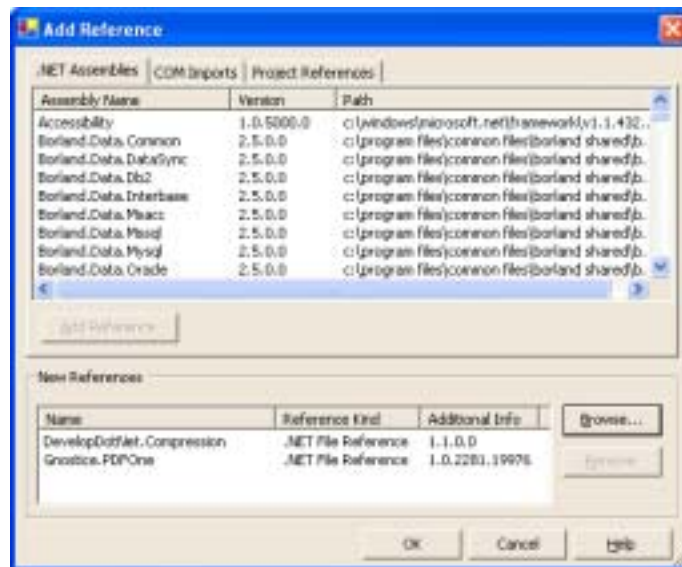


Figure 1: adding the Gnostice assemblies via Add Reference

We also have to add **using Gnostice.PDFOne**; to any unit that will be using PDFOne. Further, the first time our application is executed, PDFOne will ask for a license key to be provided. There is a trial license key available from the Gnostice web site. Figure 2 presents the license key dialog. It's worth noting that this is a once and once only requirement.

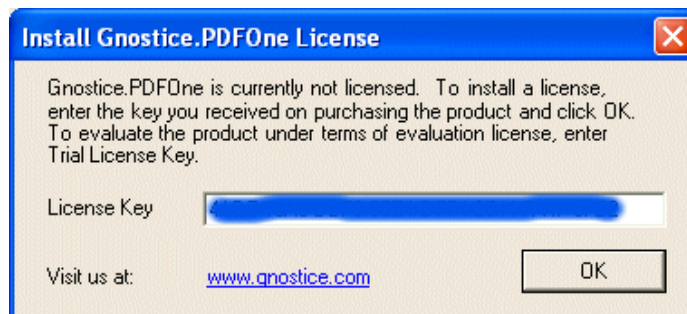


Figure 2: one off license key entry

Listing 1 presents a few lines of code that creates a new PDF document, writes some text, plots some images, draws some rectangles, saves itself and then opens itself up using Adobe Reader. It's very granular, i.e. we can control the fine-grained detail at an element-by-element level. The beauty of this fine-grained detail manifests itself when we look at the size of the PDF file PDFOne creates: only 12K...the two images alone are 9K combined.

```
private void btnCreatePDF_Click(object sender, System.EventArgs e)
{
    string FileName = "ddg_image.pdf";
    PDFDocument d = new PDFDocument();
    d.DocInfo.Author = "Craig Murphy";
    d.DocInfo.Creator = "Gnostice PDFOne.NET ";
    d.DocInfo.Subject = "Programmatic PDF Creation";
    d.DocInfo.Title = "Images";

    d.OpenAfterCreate = true;
    PDFPage p = new PDFPage();
```

```

p.Font          = new PDFFont(new Font("Arial", 18));
p.CursorPosX    = 3.5f;
p.CursorPosY    = 1f;
p.WriteText("Gnostice PDFOne.NET ", new PDFFont(StdType1Font.Helvetica,
        PDFFontStyle.Stroke, 22), 3.3f, 0.6f);

p.WriteText("http://www.richplum.co.uk", 1, 1);
p.Pen.Color     = Color.Red;
p.Pen.Width     = 5;
p.Font         = new PDFFont(new Font("Arial", 14));

p.WriteText("Normal Image", 1, 3);
p.WriteText("Centered Image within a big Rectangle", 1, 5);

RectangleF r = new RectangleF(1,1.5f,4.7f,1.47f);
p.DrawImage(Application.StartupPath + "\\ddg.gif",1,1.5f);
p.DrawRectangle(p.Pen, r, false, true);

r = new RectangleF(1,3.5f,7,1.5f);
p.DrawImage(Application.StartupPath + "\\gnostice.gif", r, false, true);
p.DrawRectangle(p.Pen, r, false, true);

d.AddPage(p);
d.Save(FileName);
d.Close();
}

```

Listing 1 – Creating a PDF programmatically using PDFOne.NET

Personally, I like having this fine-grained control over my “printed” output. It does mean a little more upstream work, but it pays dividends downstream when the user of the output asks for precise or specific tweaks to be made. Granted larger tweaks are a little more involved, but if you are prepared to invest a little time upstream building a solid object model that sits ahead of PDFOne, larger changes shouldn’t be too onerous.

If we were to run Listing 3 through Delphi 200x or Visual Studio (and the Express Editions), is the output that you might expect.

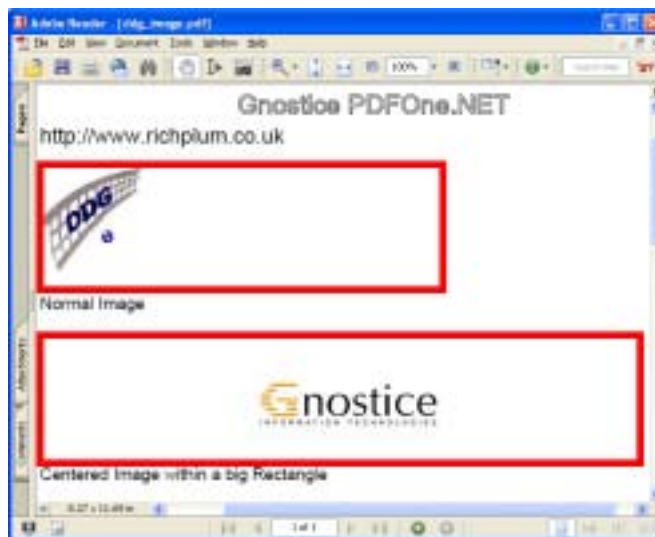


Figure 3: output from Listing 1

Other Features

Figure 4 presents a snapshot of the demonstration application (which is itself fully featured). I’ve only touched on the Standard Drawing features in Listing 1. Given the size of the demonstration application, you can understand why I’m not writing about absolutely everything in the product. Suffice to say, there’s little chance that it won’t be able to meet your PDF creation needs.

Creation
Standard Drawing
Advanced Drawing
Advanced Features
Acroforms
Create Forms
Manage Forms
Manipulation
Manage Bookmark
Manage Annotation
Merge Documents
Insert Pages
Extract Pages

Figure 4: PDFOne covers these features in its demonstration application

Homing in on one specific feature, annotations, Figure 5 presents a tab page from the demonstration application. Annotations are essentially areas within a PDF that have tooltip-style help associated with them.

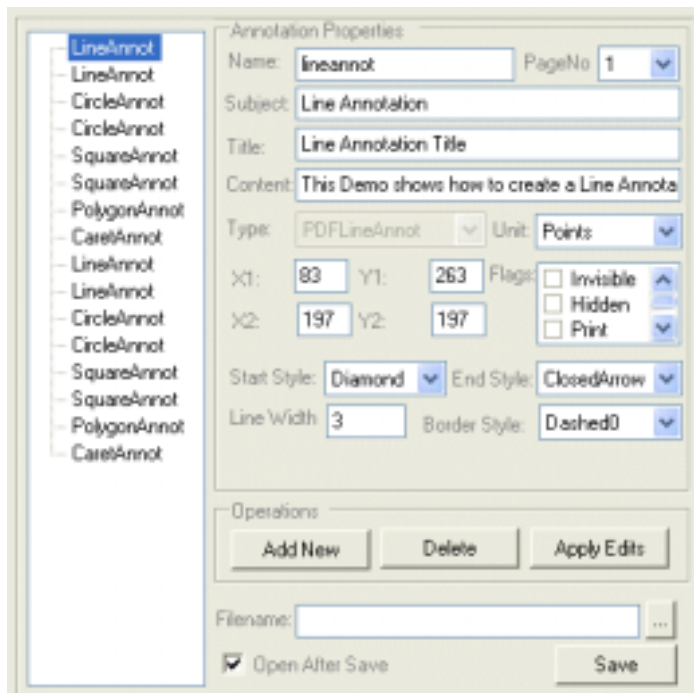


Figure 5: Just one of PDFOne's features – annotations

Figure 6 demonstrates a few things. Firstly, as a side effect, we can see that PDFOne supports a variety of shapes. Secondly, we can see that annotations are customisable: there are two in Figure 6, the red one and the yellow one. The yellow annotation pops up just like a tooltip does. The red annotation is visible at all times and sports a pointer to the precise item being annotated when the mouse moves over the annotation.

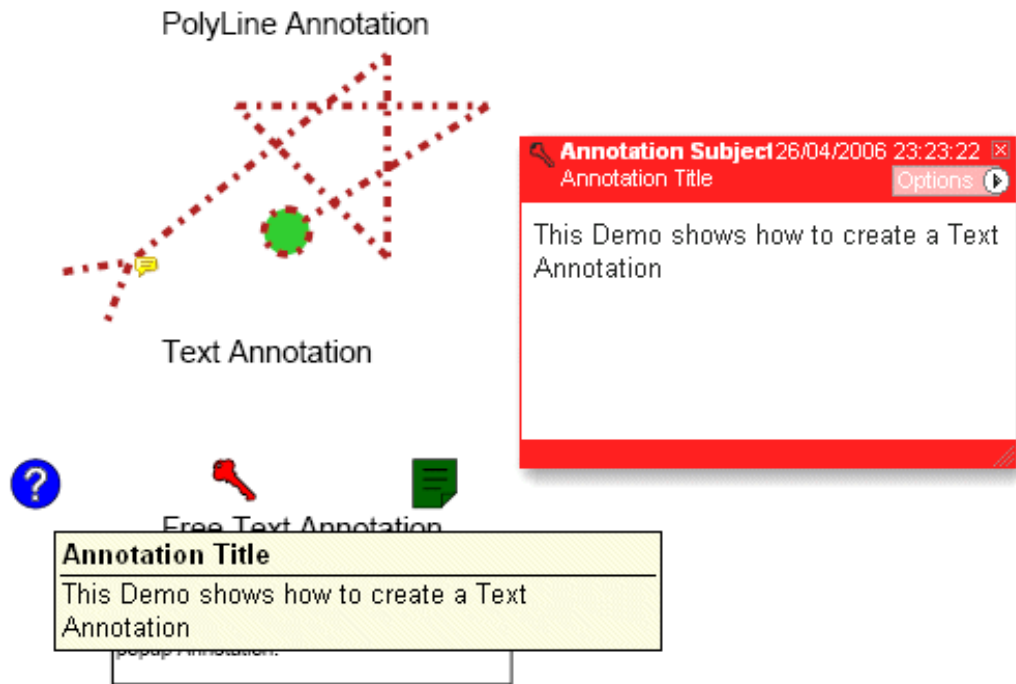


Figure 6: Annotations at work

What About The Free PDF Creators?

Whilst there are some reasonably good free PDF creations out there, they are often rather simplistic and do not offer a large array of features (cue a barrage of e-mail to the contrary!) PDFOne is fully-featured, it supports PDF version 1.3 (Acrobat 4) to version 1.6 (Acrobat 7) and does not require any external software (such as the Adobe PDF SDK) or DLLs. Some free PDF creators do require additional software.

Additionally, PDFOne, like eDocEngine, provides full programmatic access to PDF document creation and management tools. And it's full programmatic access that is documented using a 1.5MB HTML Help file.

Supported Environments

PDFOne will work in the following environments: Microsoft Visual Studio 2003 or 2005 and Borland Developer Studio (C#Builder for .NET) 2005/2006. It requires the .NET framework 1.1 or later. I have tested PDFOne using Delphi 2006 (.NET 1.1), Visual Studio 2005 and Visual C#. Note that PDFOne does not support the Delphi language, an upgrade to allow this is currently being developed. Interestingly, PDFOne is also available for the Java language.

Pricing

Gnostice offer two pricing models: subscription and one-off purchase. The subscription is priced at \$1299 and offers the following benefits:

Immediate Access to PDFOne .NET Professional (worth \$649.00)

Immediate Access to eDocEngine ActiveX/.NET Pro and PDFtoolkit ActiveX/.NET ProPlus (worth \$749.00)

Guaranteed access to Pro and ProPlus editions of the next generation electronic document creation and PDF management tools (.NET)

Guaranteed access to other new Pro and ProPlus editions of .NET products released within 12 months from purchase

For one-off purchases, these are the costs: PDFOne .NET Professional Developer v1.x is \$649; PDFOne .NET Standard Developer v1.x is \$449 and PDFOne .NET Unlimited Internet Server Deployment License is \$999

Conclusions

There's no two ways about it, PDFOne is a very competent product. It brings with it the tried and tested heritage of eDocEngine from the Win32 VCL world. PDFOne is reasonably large product, whilst the learning curve for simple PDFs is not so steep; making use of the advanced features involves a steeper, but easily manageable curve.

With rare exception (Delphi 1.0 being one) there is always a little scepticism surrounding "1.0" products. Personally, I know that PDFOne went through a rigorous beta test lasting at least 6 months and involved users/developers from outside of the Gnostice team. I have used PDFOne in a variety of development environments, with the exception a minor quirk whilst using it with Visual C# inside a virtual machine, I couldn't find fault with it. Incidentally, the quirk didn't raise its head when used outside of a virtual machine.

If you are working in .NET 1.1 or 2.0 and you are looking to create professional PDFs from your application, with fine-grained programmatic control, PDFOne is a worthy contender. The support offered by Gnostice is e-mail based and is well managed: I used the helpdesk a couple of times over the course of this review and found them to be prompt and courteous. Of course, had I read the system requirements as noted above, I would have known that I shouldn't have been trying to convince Delphi 200x to use PDFOne in a vanilla Delphi WinForm as opposed to a C# WinForm.

PDFOne, deal or no deal? That's an easy one: deal – the best PDF creation tool for Win32, eDocEngine, has just got better, welcome to the world of PDF creation in .NET .

Links

Gnostice:

<http://www.gnostice.com>

Feature comparison:

http://www.gnostice.com/pdfonefeatures_n.asp

More information about the Portable Document Format:

<http://www.adobe.com/products/acrobat/adobepdf.html>

Full specifications for the Portable Document Format:

<http://partners.adobe.com/asn/tech/pdf/specifications.jsp>



Craig is an author, developer, speaker, blogger, Certified ScrumMaster and Microsoft Most Valuable Professional (Connected Systems). He specialises in all things XML, particularly web services and XSLT. Craig is evangelical about .NET, C#, Test-Driven Development, Extreme Programming, agile methods and Scrum. He can be reached via e-mail at: ddg@craigmurphy.com, or via his web site: <http://www.craigmurphy.com> (where you can also find the source code and PowerPoint files for all of Craig's articles, reviews and presentations).