

# **How to add watermark to pdf document**

**Written by Apitron Documentation Team**

## What is a watermark

Watermark is usually a semitransparent drawing added on top of the page content which can be created using various ways. This type of marking your documents becomes necessary when you have to indicate a particular purpose the document is designed for or to give some handling instructions. Examples are: “For internal reading only”, “Do not copy”, “Top Secret” etc. It’s also useful for placing banners indicating the product name, the document was created by, or its evaluation state.

We’ll describe several watermarking approaches in this post and provide C# code samples which generate watermarks programmatically.

## Image watermark

This type of watermark is simple and convenient. You create an image containing your message and draw it over the page content.

Pros:

- Easy to create and use, single image XObject can be shared by all pages
- Provides a simple way to use any picture as watermark

Cons:

- May affect resulting file by increasing its size significantly if image used is big enough
- For the image to become transparent it has to include some kind of transparency mask and this fact can be a problem for non-transparency aware readers
- Raster images don't scale well, so this watermark may become pixelated when zoomed
- Becomes a part of page content

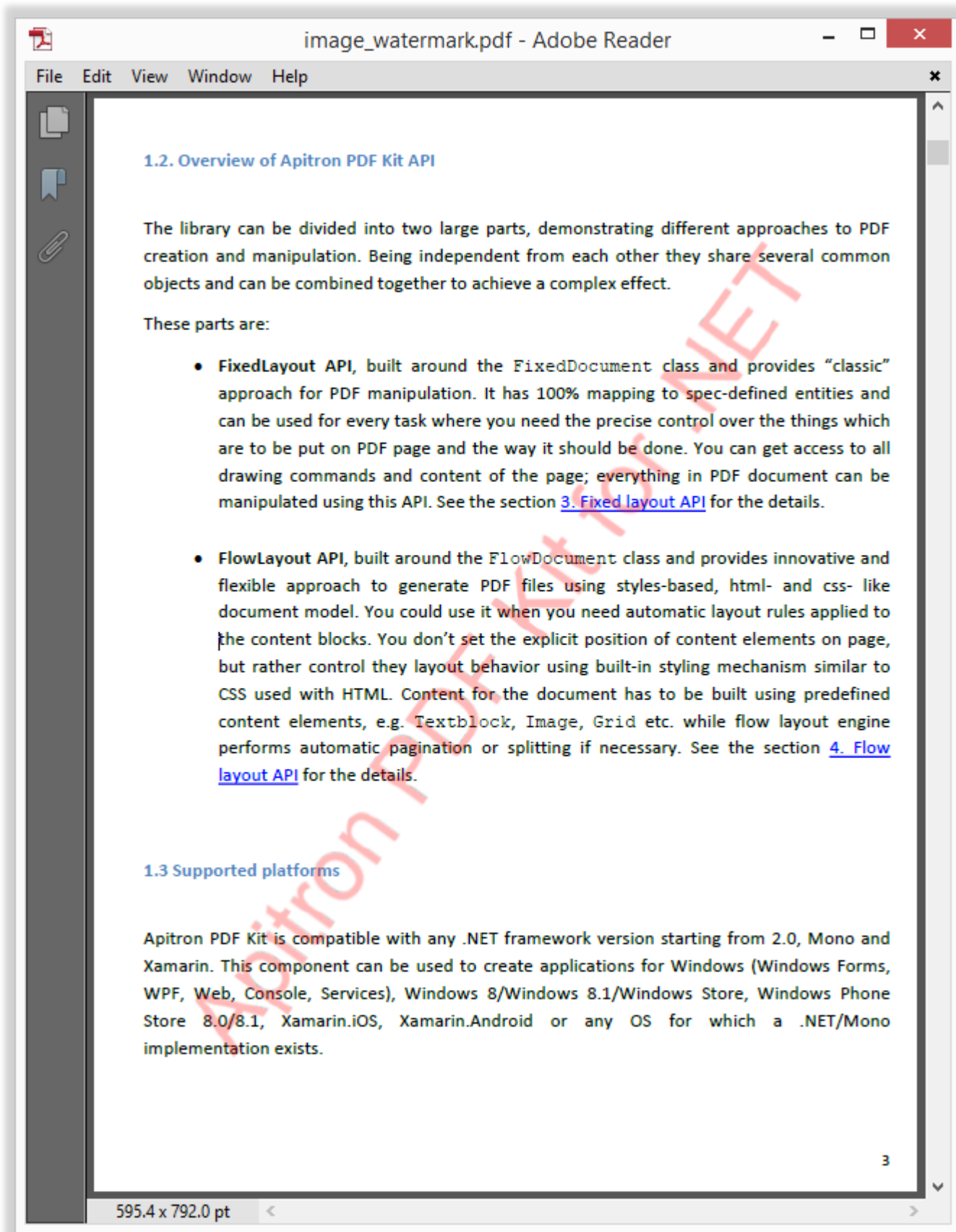
See the C# code snippet below that shows how to add image watermark:

```
/// <summary>
/// Adds image watermark to PDF document.
/// </summary>
public void AddImageWatermark()
{
    // open existing document
    using (Stream file = File.OpenRead("Apitron PDF Kit in Action.pdf"))
    {
        FixedDocument pdfDocument = new FixedDocument(file);
        // register image XObject
        pdfDocument.ResourceManager.RegisterResource(new Image("watermark", "watermark.png", true));

        // add image watermark for each page
        foreach (Page page in pdfDocument.Pages)
        {
            page.Content.AppendImage("watermark", 0, 0, page.Boundary.MediaBox.Width,
                page.Boundary.MediaBox.Height);
        }

        // save watermarked file
        using (Stream stream = File.Create("image_watermark.pdf"))
        {
            pdfDocument.Save(stream);
        }
    }
}
```

The image below demonstrates the execution results:



Pic. 1 Image watermark sample

## Form XObject watermark

This type of watermark assumes basic knowledge of PDF drawing system. Using this approach it's easy to create vector-based drawings suitable for watermarking.

Pros:

- Compactness, single watermark form XObject can be shared by all pages
- Scales well if it contains vector drawings only, requires no transparency mask

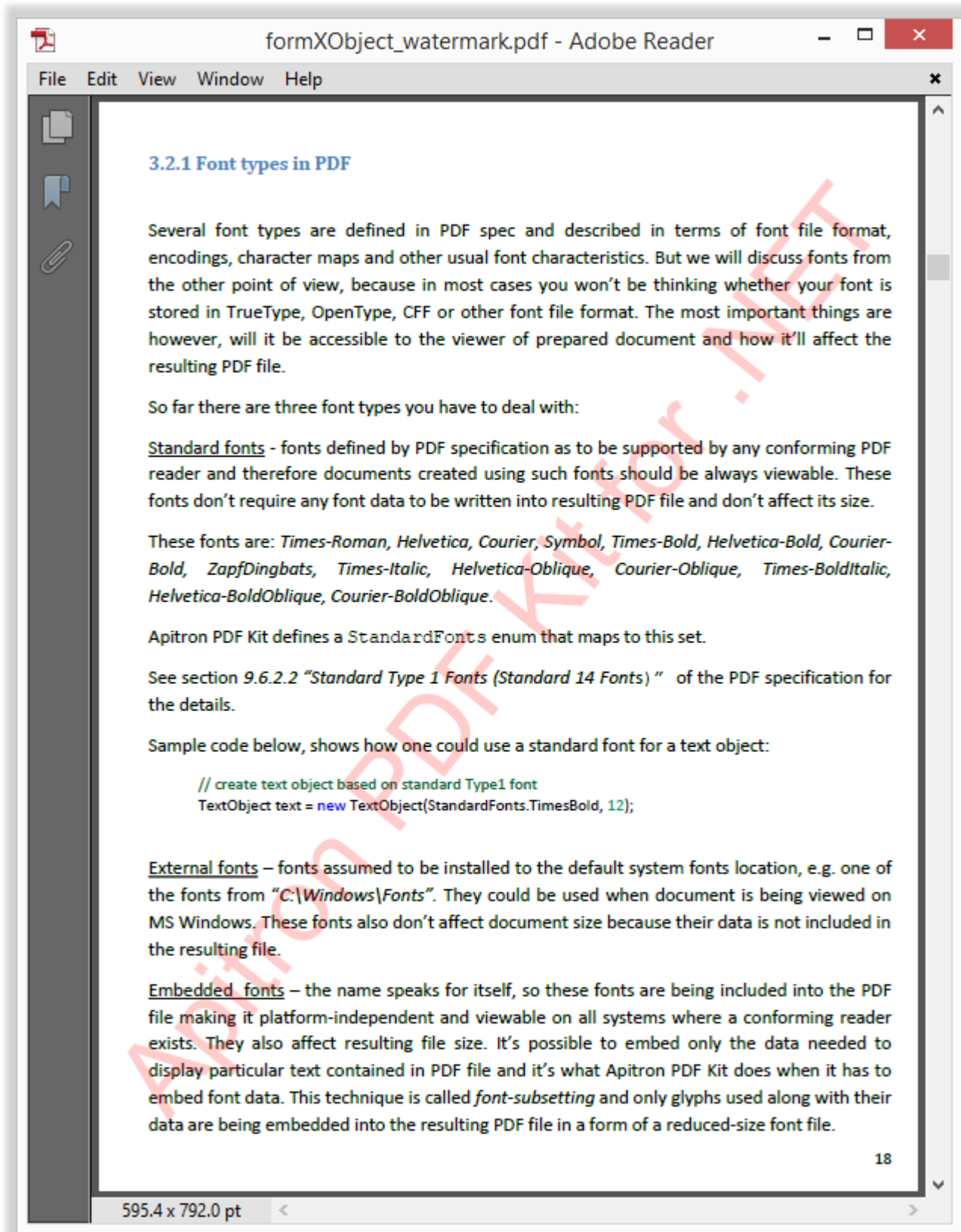
Cons:

- Requires some knowledge of PDF drawing system
- Becomes a part of page content

Let's create a simple text-based watermark using the C# code below:

```
public void AddFormXObjectWatermark()
{
    // open existing document
    using (Stream file = File.OpenRead("Apitron PDF Kit in Action.pdf"))
    {
        FixedDocument pdfDocument = new FixedDocument(file);
        // define watermark transparency using graphics state
        GraphicsState watermarkGS = new GraphicsState("gs0"){CurrentNonStrokingAlpha = 0.2};
        // register graphics state object
        pdfDocument.ResourceManager.RegisterResource(watermarkGS);
        // create watermark form XObject
        FixedContent watermark = new FixedContent("watermark", pdfDocument.Pages[0].Boundary.MediaBox);
        // register form XObject
        pdfDocument.ResourceManager.RegisterResource(watermark);
        // define text and transformation for it
        TextObject watermarkText = new TextObject(StandardFonts.Helvetica,48);
        watermarkText.AppendText("Apitron PDF Kit for .NET");
        watermark.Content.ModifyCurrentTransformationMatrix(1,1.25,-1.25,1,50,50);
        // define current color and transparency
        watermark.Content.SetGraphicsState("gs0");
        watermark.Content.SetDeviceNonStrokingColor(RgbColors.Red.Components);
        // draw watermark text
        watermark.Content.AppendText(watermarkText);
        // add watermark to each page
        foreach (Page page in pdfDocument.Pages)
        {
            page.Content.AppendXObject("watermark");
        }
        // save watermarked file
        using (Stream stream = File.Create("formXObject_watermark.pdf"))
        {
            pdfDocument.Save(stream);
        }
    }
}
```

The result is shown below. You may notice that it looks sharper because of its vector nature:



Pic. 2 Watermark added using form XObject

## Watermark annotation

A watermark annotation can be used to represent graphics that is to be printed at a fixed size and position on a page, regardless of the dimensions of the printed page.

Pros:

- Compactness, designed specifically for watermarks
- Can be easily managed using page annotations dictionary
- Requires no transparency mask

Cons:

- Requires some knowledge of PDF drawing system and annotations

```
// Adds watermark annotation to the document.
public static void AddWatermarkAnnotation()
{
    // open existing document
    using (Stream file = File.OpenRead("Apitron PDF Kit in Action.pdf"))
    {
        FixedDocument pdfDocument = new FixedDocument(file);
        // define watermark transparency using graphics state and register graphics state object
        GraphicsState watermarkGS = new GraphicsState("gs0") { CurrentNonStrokingAlpha = 0.2 };
        pdfDocument.ResourceManager.RegisterResource(watermarkGS);
        // create watermark content
        FixedContent watermark = new FixedContent("watermark", pdfDocument.Pages[0].Boundary.MediaBox);
        // define text and transformation for it
        TextObject watermarkText = new TextObject(StandardFonts.Helvetica, 48);
        watermarkText.AppendText("Apitron PDF Kit for .NET");
        watermark.Content.ModifyCurrentTransformationMatrix(1, 1.25, -1.25, 1, 50, 50);
        // define current color and transparency
        watermark.Content.SetGraphicsState("gs0");
        watermark.Content.SetDeviceNonStrokingColor(RgbColors.Red.Components);
        // draw watermark text
        watermark.Content.AppendText(watermarkText);
        // create watermark annotation object for each pages
        foreach (Page page in pdfDocument.Pages)
        {
            WatermarkAnnotation watermarkAnnotation = new WatermarkAnnotation(page.Boundary.MediaBox);
            watermarkAnnotation.Appearance.Normal = watermark;
            page.Annotations.Add(watermarkAnnotation);
        }
        using (Stream stream = File.Create("watermark_annotation.pdf"))
        {
            pdfDocument.Save(stream);
        }
    }
}
```

The code creating watermark annotation produces the same results as the code that adds form XObject watermark.

## Watermarks removal

It's possible to remove watermarks from PDF file however we don't recommend doing it because it can cause legal problems. Techniques used involve content analysis as well as annotations checks. There is no 100% reliable method, however, to remove all watermarking information using single algorithm, because watermarks might be hidden in PDF metadata or other less evident places.

For example, one may use fully transparent image which would appear only when the document is being printed. Think of watermark as of piece of info hidden inside the PDF file, it can be just anything.

## Conclusion

Adding watermarks is not a tricky task and, as you can see, it can be completed quite easy using Apitron PDF Kit for .NET component. This component is available for many platforms and makes you able to create applications for Windows and Windows Store, Xamarin.iOS and Xamarin.Android, OS X or any other system where a .NET/MONO can run. ASP.NET and Azure environments are supported as well. You may [visit](#) its product page or browse documentation [here](#).