

# **How to export document to PDF/A and its core differences from PDF**

**Written by Apitron Documentation Team**

## Introduction

PDF/A is a format based on PDF and designed for long-term data archiving. The profound description of this standard can be found on Wikipedia or in ISO 19005-1, ISO 19005-2, ISO 19005-3 documents. We won't dive into all specifics here and will limit ourselves to a brief description of its core differences.

The main requirement a PDF/A document should conform to is a self-containment. That is, all the data required to display this document should be embedded, including:

- Fonts, including standard 14 PDF fonts, preferably in a form of font subsets if not all glyphs are used
- Output intent colorspace in a form of ICC profile
- Images, if image colorspace is different from the output intent then conversion is required
- Forms fields appearance

Many features allowed in PDF are prohibited in PDF/A, for example:

- Soft masks and transparency groups
- Some types of annotations
- Links to external docs
- Embedded files

A conformance of a document to PDF/A standard can be validated using many tools, and an "official" one is a Pre-flight, available in Adobe Acrobat Pro. Apitron PDF Kit for .NET supports export to PDF/A on a core level and can be used to create PDF files for archiving and platform-independent viewing. See the code samples provided in [The code](#) section

## The code

Let's create the simple "Hello World!" PDF/A document using Apitron PDF Kit for .NET and check its properties. Functions below demonstrate how to do it using both supported APIs.

### Fixed layout API:

```
/// <summary>
/// Creates PDF/A document using Fixed layout API.
/// </summary>
private void CreatePDFADocumentUsingFixedLayoutAPI()
{
    // create document object and indicate PDF/A support
    FixedDocument doc = new FixedDocument(PdfStandard.PDFA);

    // create text object, set its font and content
    TextObject textObject = new TextObject(StandardFonts.Helvetica,14);
    textObject.AppendText("Hello world! PDF/A document created by Apitron PDF Kit for .NET.");

    // create new page and append text object created above
    Page page = new Page(Boundaries.A4);
    page.Content.Translate(10,820);
    page.Content.AppendText(textObject);
    doc.Pages.Add(page);

    // save PDF/A document
    using (Stream stream = File.Create("pdfa_doc.pdf"))
    {
        doc.Save(stream);
    }
}
```

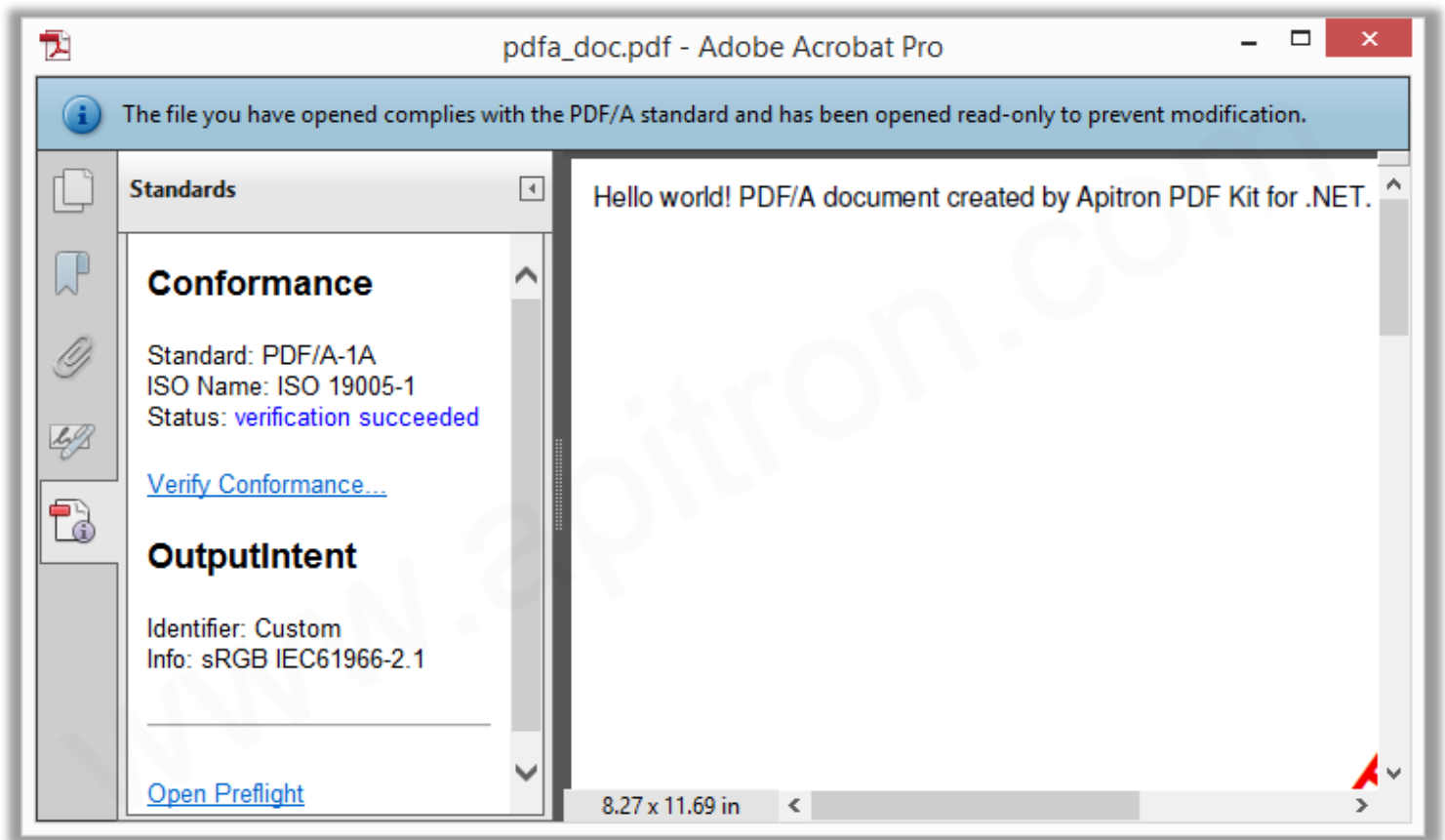
### Flow layout API:

```
/// <summary>
/// Creates PDF/A document using Flow layout API.
/// </summary>
private void CreatePDFADocumentUsingFlowLayoutAPI()
{
    ResourceManager resourceManager = new ResourceManager();
    // create document and set margin
    FlowDocument doc = new FlowDocument(){Margin = new Thickness(10)};

    // create text block and append to document
    TextBlock textBlock = new TextBlock("Hello world! PDF/A document created by Apitron PDF Kit for
.NET.");
    doc.Add(textBlock);

    // save as PDF/A document
    using (Stream stream = File.Create("pdfa_doc.pdf"))
    {
        doc.Write(stream, resourceManager, new PageBoundary(Boundaries.A4), PdfStandard.PDFA);
    }
}
```

Functions above produce identical PDF files looking as one shown below:



Pic. 1 PDF/A document produced by Apitron PDF Kit

On the left panel you can see that produced document successfully passes PDF/A standard conformance validation.

## Conclusion

Apitron PDF Kit for .NET library is available by the following [link](#). It can be used to create cross platform PDF processing applications and is suitable for many PDF generation scenarios. Create Windows, Android and iOS (Xamarin), MONO apps, web or cloud solutions. It's truly cross-platform and we constantly improve it by adding new features and providing updates.

Check out our [free book](#) describing the API and containing ready to use code snippets. Read our [blog](#) or [contact us](#) if you have any questions.