How to flatten PDF form fields

Written by Apitron Documentation Team

Introduction

Many of you became familiar with PDF forms either as developers or as citizens needed to report taxes or request some service from the government. PDF forms are everywhere now and in this article we'll talk about them. A great example of using this PDF feature is our Universal App code sample available by this link.

A PDF form is a collection of objects called "fields" and these fields get stored inside the PDF file when you create or modify them. Some fields can be displayed on PDF page, while others are fully internal; it depends on how the field was created. In order to display the field on page you need to create a "view" for it, in PDF terms it's called a Widget Annotation. See the section *12.5.6.19 Widget Annotations* of the PDF specification for the details.

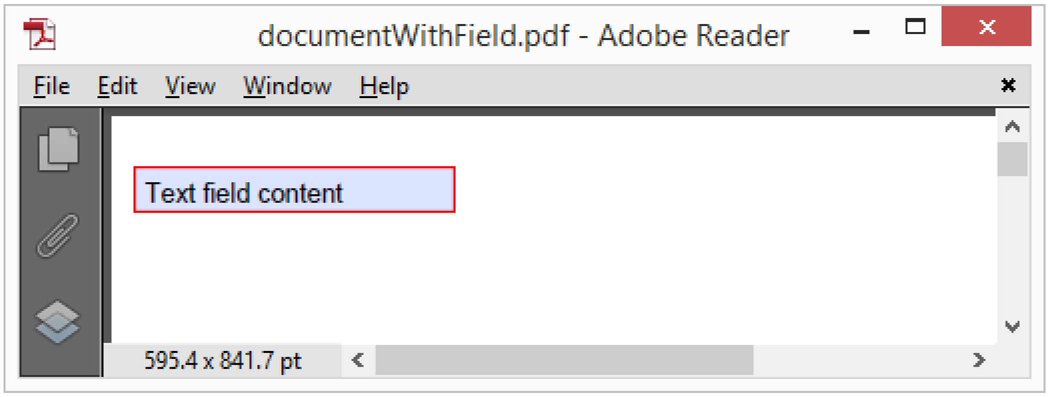
While the field's view is a dynamic object, allowing you to change the value of the field for example (if the read-only flag is not set), sometimes you may need to make it static, turning it to a regular PDF content and unlinking from the parent field. This process is called flattening of the PDF fields and in this article we'll show how to do it using Apitron PDF Kit API.

The code

```
class Program
{
    // creates test PDF document with text field
    private static void CreateTestDocument()
    {
        FixedDocument doc = new FixedDocument();

        // create text field and add to doc's field collection
        TextField textField = new TextField("textField", "Text field content");
        doc.AcroForm.Fields.Add(textField);

        // create new page and add text view to it
        Page page = new Page();

        TextFieldView fieldView = new TextFieldView(textField, new Boundary(10, 800, 150, 820));
        fieldView.BorderColor = RgbColors.Red.Components;

        page.Annotations.Add( fieldView);
        // add page to document
        doc.Pages.Add(page);

        // save document
        using (Stream stream = File.Create("documentWithField.pdf"))
        {
            doc.Save(stream);
        }
    }

    static void Main(string[] args)
    {
        CreateTestDocument();

        using (Stream inputStream = File.Open("documentWithField.pdf",FileMode.Open))
        {
            FixedDocument doc = new FixedDocument(inputStream);

            // we can either flatten all fields or enumerate the collection
            // and flatten specific field by calling its Flatten() method
            doc.AcroForm.FlattenFields();

            // save document
            using (Stream outputStream = File.Create("fieldsFlattening.pdf"))
            {
                doc.Save(outputStream);
            }
        }

        Process.Start("fieldsFlattening.pdf");
    }
}
```

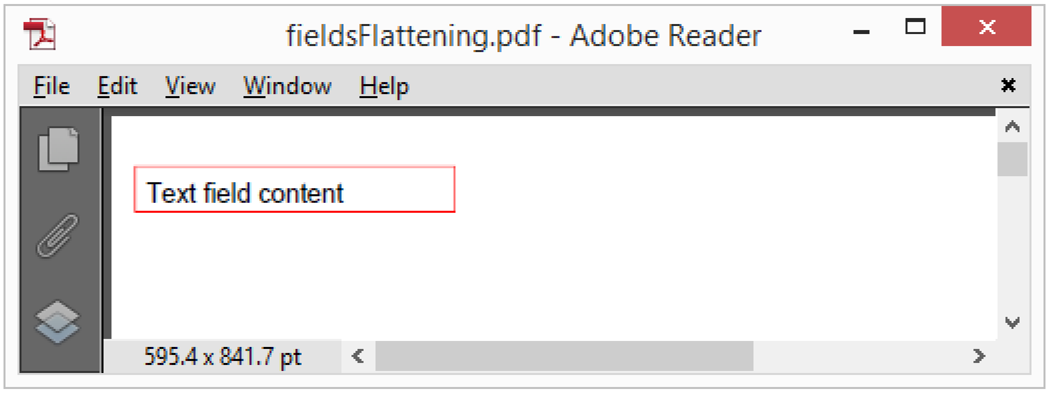
This code creates a test PDF form containing a text field, saves it and later loads and flattens containing fields. The complete sample can be found in our github repo.

Results look as follows:



Text field content

Pic. 1 Test PDF form



Text field content

Pic. 2 Flattened PDF Form

On the first picture you can see the highlighted PDF field that can be edited and saved. On the second picture the flattened PDF form is shown. The view became the part of the page's content and is not linked to the underlying field anymore.

Conclusion

Working with PDF forms is simple and convenient using Apitron PDF Kit for .NET component. Create cross-platform apps targeting desktop, web, and mobile platforms including iOS and Android(via Xamarin) and enjoy the easy to use API and our highly-qualified support. Read more about the component here or just email us if you have any questions about the library and its features.