Introduction to FDF or Forms Data Format

Written by Apitron Documentation Team

Introduction

All of you probably have seen or filled PDF forms while submitting tax reports, registration data and so on. PDF offers more than enough to support forms based workflow and to make data footprint smaller Adobe has introduced FDF – the Forms Data Format.

Its purpose is isolate the data from the representation, meaning that FDF file doesn't contain drawings, images and fonts which are typically used in "normal" PDF documents to display pages content, because it also doesn't contain any pages as well (it's not completely true because FDF annotations can contain graphical data, it will be covered later). It its simplest form a FDF document contains just a set of fields and their values.

FDF can be used for submitting forms data to the server, receiving the response and filling the interactive forms. It can also be used for exporting PDF forms data to standalone files which can be stored in database or transmitted and later imported back to the original PDF form. Other features include adding custom annotations to the PDF document the FDF is applied to, or composing the PDF document from multiple sources using page templates.

So to summarize what you can do with FDF:

1) Export and import PDF forms data, you can have several interactive PDF forms templates sharing the same set of fields and use single FDF to fill all of them.

2) Add annotations, e.g. watermarks etc. to the target PDF documents

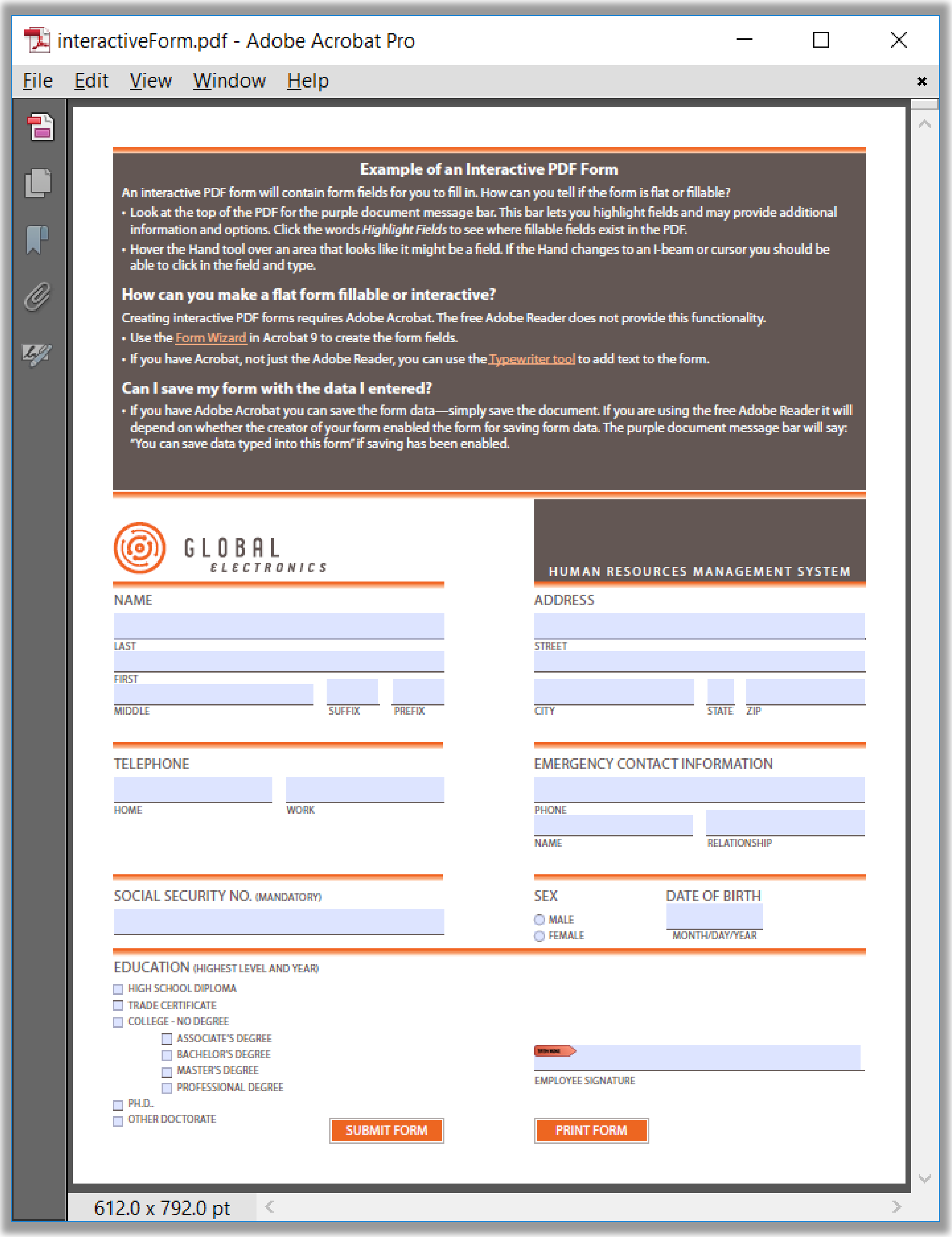
3) Import pages to PDF files from many source PDF documents, e.g. import document cover from one document and other parts from other documents. So you can even let your designers work in parallel on different templates and write code that combines them at the same time.

FDF is based on PDF and uses the same file structure, but has some differences described in section **12.7.7 Forms Data Format** of the PDF specification. We prepared a set of code samples which show how to work with FDF using Apitron PDF Kit for .NET, you can find them below. The complete code sample can be downloaded from our github repo.

The code

How to generate FDF from PDF

We’ll be working with the following PDF form:



Example of an Interactive PDF Form

An interactive PDF form will contain form fields for you to fill in. How can you tell if the form is flat or fillable?

- Look at the top of the PDF for the purple document message bar. This bar lets you highlight fields and may provide additional information and options. Click the words *Highlight Fields* to see where fillable fields exist in the PDF.
- Hover the Hand tool over an area that looks like it might be a field. If the Hand changes to an I-beam or cursor you should be able to click in the field and type.

How can you make a flat form fillable or interactive?

Creating interactive PDF forms requires Adobe Acrobat. The free Adobe Reader does not provide this functionality.

- Use the Form Wizard in Acrobat 9 to create the form fields.
- If you have Acrobat, not just the Adobe Reader, you can use the Typewriter tool to add text to the form.

Can I save my form with the data I entered?

- If you have Acrobat you can save the form data—simply save the document. If you are using the free Adobe Reader it will depend on whether the creator of your form enabled the form for saving form data. The purple document message bar will say: "You can save data typed into this form" if saving has been enabled.

GLOBAL ELECTRONICS

HUMAN RESOURCES MANAGEMENT SYSTEM

NAME

LAST

FIRST

MIDDLE SUFFIX PREFIX

ADDRESS

STREET

CITY STATE ZIP

TELEPHONE

HOME WORK

EMERGENCY CONTACT INFORMATION

PHONE

NAME RELATIONSHIP

SOCIAL SECURITY NO. (MANDATORY)

SEX

○ MALE

○ FEMALE

DATE OF BIRTH

MONTH/DAY/YEAR

EDUCATION (HIGHEST LEVEL AND YEAR)

- □ HIGH SCHOOL DIPLOMA
- □ TRADE CERTIFICATE
- □ COLLEGE - NO DEGREE
  - □ ASSOCIATE’S DEGREE
  - □ BACHELOR’S DEGREE
  - □ MASTER’S DEGREE
  - □ PROFESSIONAL DEGREE
- □ PH.D.
- □ OTHER DOCTORATE

EMPLOYEE SIGNATURE

SUBMIT FORM

PRINT FORM

Pic. 1 Interactive PDF form

Here is the code for generating FDF from this PDF form using Apitron PDF Kit

```
private static void GenerateFDF()
{
    using (Stream fileStream = File.Open("../../data/interactiveForm.pdf",
        FileMode.Open, FileAccess.ReadWrite))
    {
        using (FixedDocument target = new FixedDocument(fileStream))
        {
            // dump PDF doc's form data to FDF file
            using (FormsDataDocument fdfDocument = target.AcroForm.ExportFormsData())
            {
                using (FileStream fdfOutputStream = File.Create("interactiveForm.fdf"))
                {
                    fdfDocument.Save(fdfOutputStream);
                }
            }
        }
    }
}
```

You can also use Adobe Acrobat and go to Tools -> Forms -> More Form Options -> Manage Form Data -> Export Data… and save the results as FDF.

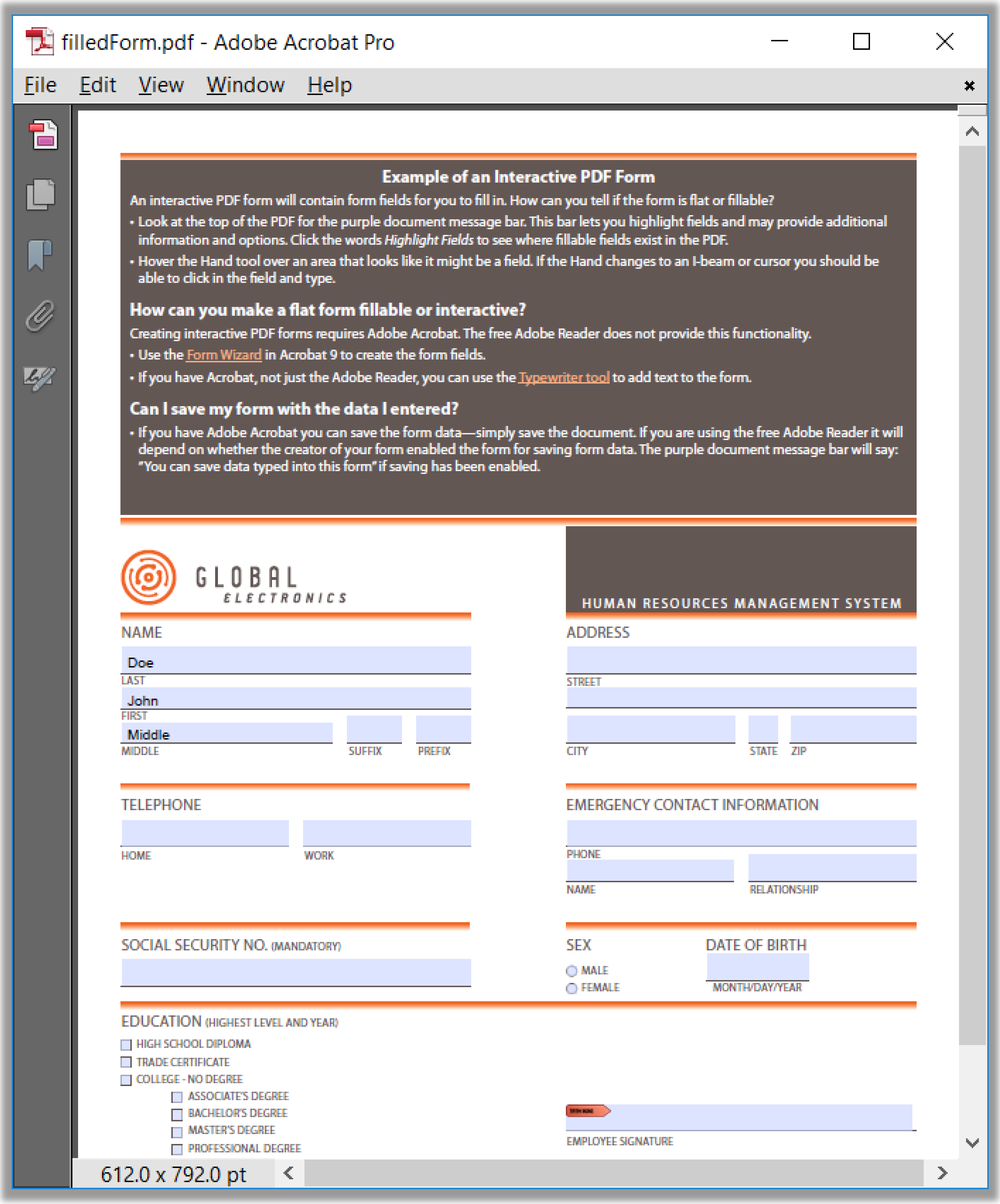
Fill PDF form using FDF

We'll fill the same form using exported FDF file. As the originally the form was empty we will set several values to show how the FDF content can be altered on the fly.

```
private static void FillFormUsingLoadedFDF()
{
    using (Stream inputStream = File.Open("../../data/interactiveForm.pdf",
        FileMode.Open, FileAccess.ReadWrite), outputStream =File.Create("filledForm.pdf"))
    {
        // open the target PDF file
        using (FixedDocument target = new FixedDocument(inputStream))
        {
            // open the prepared FDF doc
            using (Stream fdfStream = File.Open("../../data/formData.fdf", FileMode.Open))
            {
                using (FormsDataDocument fdfDocument = new FormsDataDocument(fdfStream))
                {
                    // fill the FDF and apply it
                    fdfDocument.Fields["Name_First"].SetValue("John");
                    fdfDocument.Fields["Name_Last"].SetValue("Doe");
                    fdfDocument.Fields["Name_Middle"].SetValue("Middle");

                    target.ApplyFormsDataDocument(fdfDocument);
                }
            }

            target.Save(outputStream);
        }
    }
}
```

You can also simply open the FDF file in Adobe Reader to get it applied to the default PDF file or use Tools -> Forms -> More Form Options -> Manage Form Data -> Import Data…

See the resulting document below:



Example of an Interactive PDF Form

An interactive PDF form will contain form fields for you to fill in. How can you tell if the form is flat or fillable?

- Look at the top of the PDF for the purple document message bar. This bar lets you highlight fields and may provide additional information and options. Click the words *Highlight Fields* to see where fillable fields exist in the PDF.
- Hover the Hand tool over an area that looks like it might be a field. If the Hand changes to an I-beam or cursor you should be able to click in the field and type.

How can you make a flat form fillable or interactive?

Creating interactive PDF forms requires Adobe Acrobat. The free Adobe Reader does not provide this functionality.

- Use the Form Wizard in Acrobat 9 to create the form fields.
- If you have Acrobat, not just the Adobe Reader, you can use the Typewriter tool to add text to the form.

Can I save my form with the data I entered?

- If you have Adobe Acrobat you can save the form data—simply save the document. If you are using the free Adobe Reader it will depend on whether the creator of your form enabled the form for saving form data. The purple document message bar will say: "You can save data typed into this form" if saving has been enabled.

GLOBAL ELECTRONICS

HUMAN RESOURCES MANAGEMENT SYSTEM

NAME

Doe
LAST

John
FIRST

Middle
MIDDLE SUFFIX PREFIX

ADDRESS

STREET

CITY STATE ZIP

TELEPHONE

HOME WORK

EMERGENCY CONTACT INFORMATION

PHONE

NAME RELATIONSHIP

SOCIAL SECURITY NO. (MANDATORY)

SEX

○ MALE
○ FEMALE

DATE OF BIRTH

MONTH/DAY/YEAR

EDUCATION (HIGHEST LEVEL AND YEAR)

- □ HIGH SCHOOL DIPLOMA
- □ TRADE CERTIFICATE
- □ COLLEGE - NO DEGREE
  - □ ASSOCIATE'S DEGREE
  - □ BACHELOR'S DEGREE
  - □ MASTER'S DEGREE
  - □ PROFESSIONAL DEGREE

EMPLOYEE SIGNATURE

Pic. 2 PDF form filled using FDF

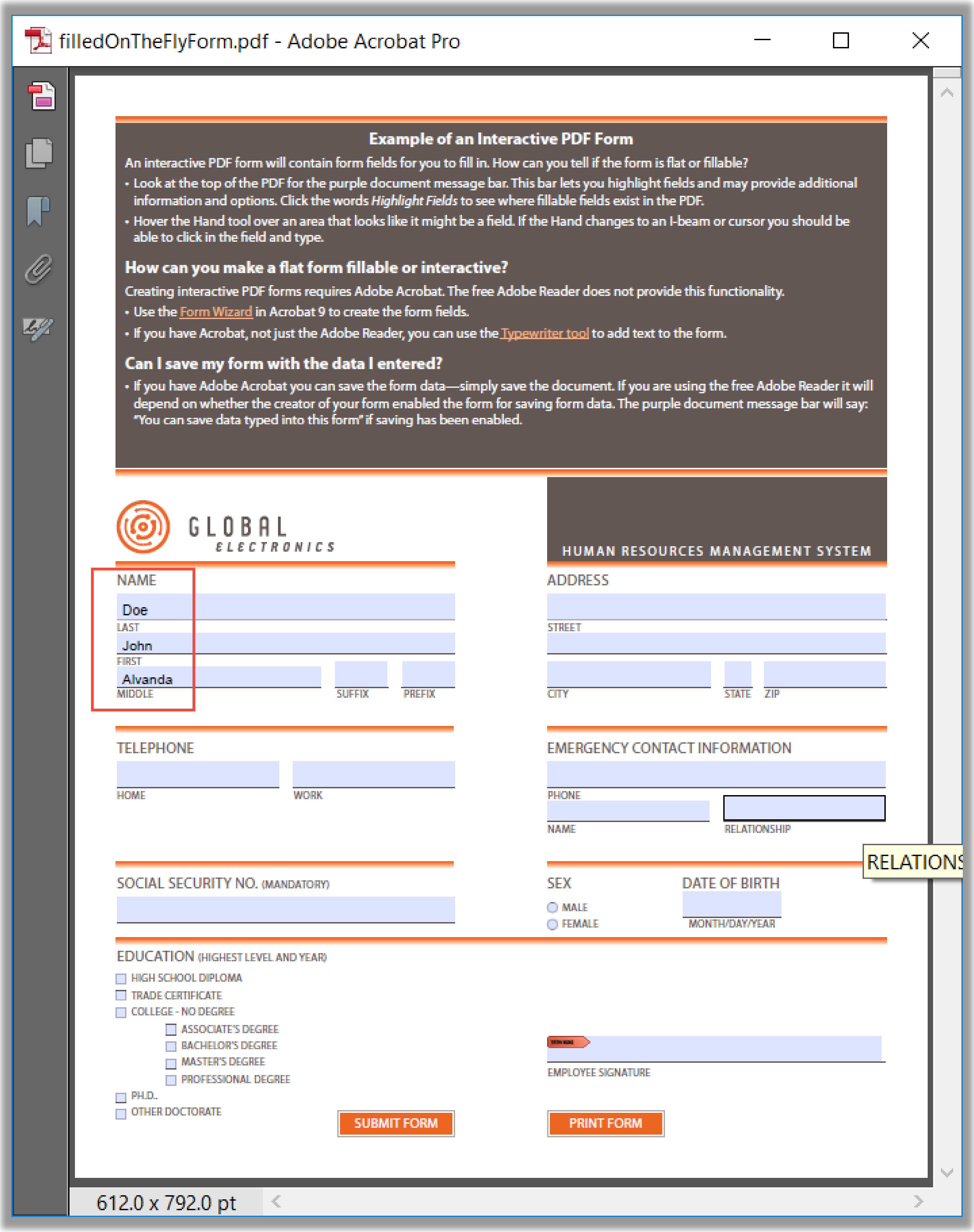
Fill the PDF form using FDF created on the fly

It’s easy to create FDF on the fly and apply to one or several PDF forms. Here we’ll apply generated FDF to the two different forms having the same set of fields.

```
private static void CreateFDFOnTheFlyAndFillForm(string pdfFormFile)
{
    using (Stream inputStream = File.Open(string.Format("../../data/{0}.pdf",pdfFormFile),
        FileMode.Open),
        outputStream = File.Create("filledForm.pdf"))
    {
        using (FixedDocument target = new FixedDocument(inputStream))
        {
            // create the forms data document and add
            // a few fields we know the names of along with their values.
            using (FormsDataDocument fdfDocument = new FormsDataDocument())
            {
                // add the first name field
                FdfField firstName = new FdfField("Name_First", FdfFieldType.Text);
                firstName.SetValue("John");

                // add the last name field
                FdfField lastName = new FdfField("Name_Last", FdfFieldType.Text);
                lastName.SetValue("Doe");

                // add the last name field
                FdfField middleName = new FdfField("Name_Middle", FdfFieldType.Text);
                middleName.SetValue("Alvanda");

                fdfDocument.Fields.Add(firstName);
                fdfDocument.Fields.Add(lastName);
                fdfDocument.Fields.Add(middleName);

                // apply the FDF doc to the target and save,
                // now we should have field's values copied to the
                // original PDF doc
                target.ApplyFormsDataDocument(fdfDocument);

                target.Save(outputStream);
            }
        }
    }
}
```

We’ll apply this code to the different versions of the same form, the normal one and compact. They both share the same set of fields and were created to show how the single FDF document can be used in “one data set” -> “many representations scenario”.

The normal filled form:



filledOnTheFlyForm.pdf - Adobe Acrobat Pro

Example of an Interactive PDF Form

An interactive PDF form will contain form fields for you to fill in. How can you tell if the form is flat or fillable?

- Look at the top of the PDF for the purple document message bar. This bar lets you highlight fields and may provide additional information and options. Click the words *Highlight Fields* to see where fillable fields exist in the PDF.
- Hover the Hand tool over an area that looks like it might be a field. If the Hand changes to an I-beam or cursor you should be able to click in the field and type.

How can you make a flat form fillable or interactive?

Creating interactive PDF forms requires Adobe Acrobat. The free Adobe Reader does not provide this functionality.

- Use the Form Wizard in Acrobat 9 to create the form fields.
- If you have Acrobat, not just the Adobe Reader, you can use the Typewriter tool to add text to the form.

Can I save my form with the data I entered?

- If you have Adobe Acrobat you can save the form data—simply save the document. If you are using the free Adobe Reader it will depend on whether the creator of your form enabled the form for saving form data. The purple document message bar will say: "You can save data typed into this form" if saving has been enabled.

GLOBAL ELECTRONICS

HUMAN RESOURCES MANAGEMENT SYSTEM

NAME

Doe
LAST

John
FIRST

Alvanda
MIDDLE

SUFFIX PREFIX

ADDRESS

STREET

CITY STATE ZIP

TELEPHONE

HOME WORK

EMERGENCY CONTACT INFORMATION

PHONE

NAME RELATIONSHIP

SOCIAL SECURITY NO. (MANDATORY)

SEX

○ MALE
○ FEMALE

DATE OF BIRTH

MONTH/DAY/YEAR

EDUCATION (HIGHEST LEVEL AND YEAR)

□ HIGH SCHOOL DIPLOMA
□ TRADE CERTIFICATE
□ COLLEGE - NO DEGREE
  □ ASSOCIATE'S DEGREE
  □ BACHELOR'S DEGREE
  □ MASTER'S DEGREE
  □ PROFESSIONAL DEGREE
□ PH.D.
□ OTHER DOCTORATE

EMPLOYEE SIGNATURE

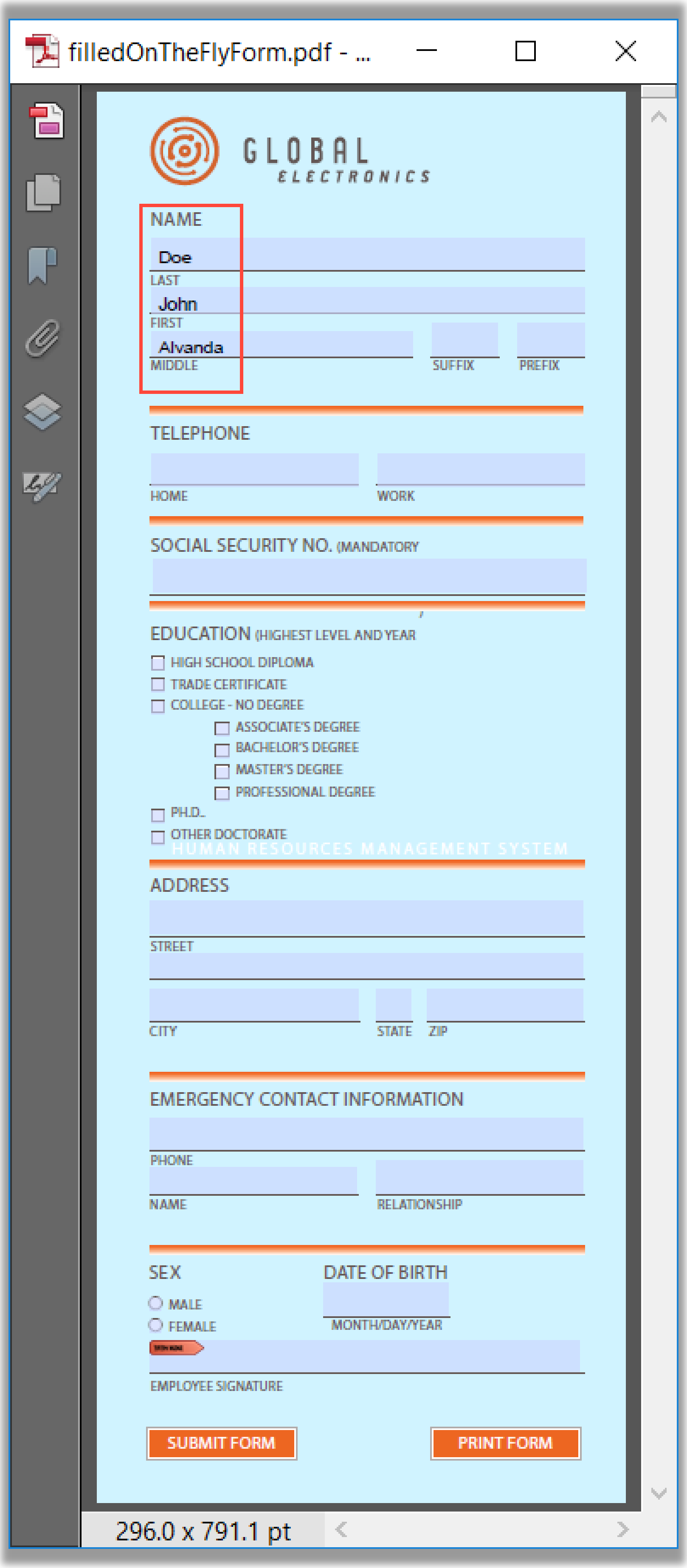
SUBMIT FORM

PRINT FORM

612.0 x 792.0 pt

Pic. 3 The normal version of the form filled using FDF created on the fly

The compact version of the same form:



NAME

Doe
LAST

John
FIRST

Alvanda
MIDDLE

SUFFIX

PREFIX

TELEPHONE

HOME

WORK

SOCIAL SECURITY NO. (MANDATORY

EDUCATION (HIGHEST LEVEL AND YEAR

□ HIGH SCHOOL DIPLOMA
□ TRADE CERTIFICATE
□ COLLEGE - NO DEGREE
□ ASSOCIATE'S DEGREE
□ BACHELOR'S DEGREE
□ MASTER'S DEGREE
□ PROFESSIONAL DEGREE
□ PH.D.
□ OTHER DOCTORATE

HUMAN RESOURCES MANAGEMENT SYSTEM

ADDRESS

STREET

CITY

STATE

ZIP

EMERGENCY CONTACT INFORMATION

PHONE

NAME

RELATIONSHIP

SEX

○ MALE
○ FEMALE

DATE OF BIRTH

MONTH/DAY/YEAR

EMPLOYEE SIGNATURE

Pic. 4 Compact version of the PDF form filled using FDF created on the fly

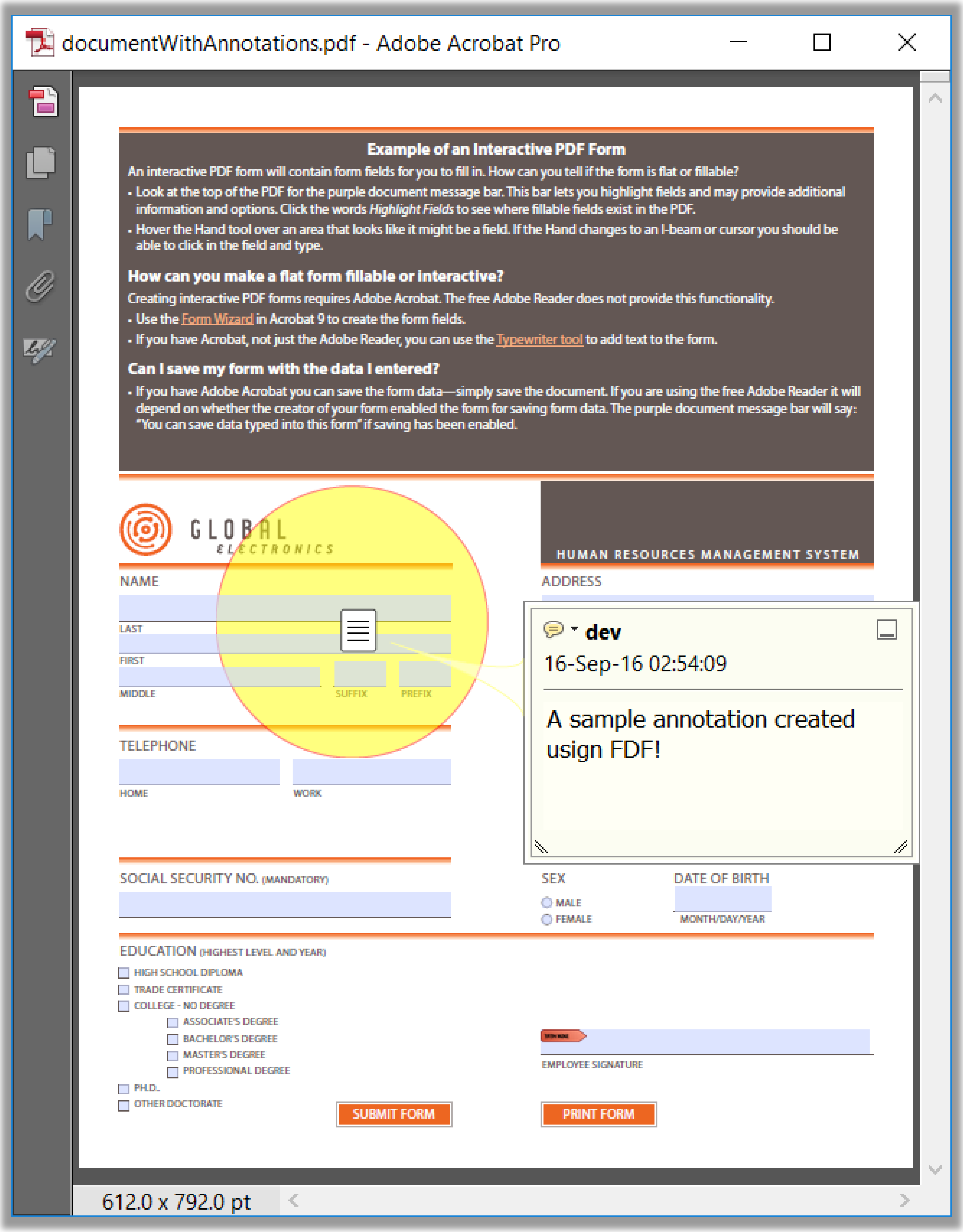
Add annotations using FDF

It’s possible to add annotations using FDF files and these annotations can contain drawings, text, images and other content allowed in regular PDF. One of the possible usage scenarios is separation of user-defined annotations and original PDF. This way you can always have the original version of the file and several sets or versions of annotations instead of saving several versions of the same PDF for every annotating session.

The code below shows how to add simple annotations using FDF:

```
private static void AddAnnotationsUsingFDF()
{
    // open pdf document that later can be used
    // as a target for applying FDF
    using (Stream inputStream = File.Open("../../data/interactiveForm.pdf", FileMode.Open),
        outputStream = File.Create("documentWithAnnotations.pdf"))
    {
        using (FixedDocument targetPDFDocument = new FixedDocument(inputStream))
        {
            // prepare our forms data document
            using (FormsDataDocument fdfDocument = new FormsDataDocument())
            {
                // set default file this document applies to
                fdfDocument.File = "interactiveForm.pdf";
                // add two annotations here, they will be added the target PDF document
                fdfDocument.Annotations.Add(new CircleAnnotation(new Boundary(100, 300, 300, 500),
                    AnnotationFlags.Default,
                    new AnnotationBorderStyle())
                {
                    Color = RgbColors.Red.Components,
                    InteriorColor = RgbColors.Yellow.Components,
                    Opacity = 0.5
                });

                fdfDocument.Annotations.Add(new TextAnnotation(190, 390)
                {
                    Contents = "A sample annotation created usign FDF!",
                    IsOpen = true
                });

                // optionally save the FDF, it can be opened later and used
                // for applying the content to PDF files
                using (FileStream fileStream = File.Create("annotations.fdf"))
                {
                    fdfDocument.Save(fileStream);
                }

                // Apply FDF and save the result. It can be also applied using Adobe PDF Reader,
                // just open it and it will be applied automatically using the file name
                // set to fdfDocument.File property.
                targetPDFDocument.ApplyFormsDataDocument(fdfDocument);

                targetPDFDocument.Save(outputStream);
            }
        }
    }
}
```

Resulting document looks as follows:



documentWithAnnotations.pdf - Adobe Acrobat Pro

Example of an Interactive PDF Form

An interactive PDF form will contain form fields for you to fill in. How can you tell if the form is flat or fillable?

- Look at the top of the PDF for the purple document message bar. This bar lets you highlight fields and may provide additional information and options. Click the words *Highlight Fields* to see where fillable fields exist in the PDF.
- Hover the Hand tool over an area that looks like it might be a field. If the Hand changes to an I-beam or cursor you should be able to click in the field and type.

How can you make a flat form fillable or interactive?

Creating interactive PDF forms requires Adobe Acrobat. The free Adobe Reader does not provide this functionality.

- Use the Form Wizard in Acrobat 9 to create the form fields.
- If you have Acrobat, not just the Adobe Reader, you can use the Typewriter tool to add text to the form.

Can I save my form with the data I entered?

- If you have Adobe Acrobat you can save the form data—simply save the document. If you are using the free Adobe Reader it will depend on whether the creator of your form enabled the form for saving form data. The purple document message bar will say: "You can save data typed into this form" if saving has been enabled.

GLOBAL ELECTRONICS

HUMAN RESOURCES MANAGEMENT SYSTEM

NAME

LAST

FIRST

MIDDLE

SUFFIX

PREFIX

ADDRESS

dev

16-Sep-16 02:54:09

A sample annotation created usign FDF!

TELEPHONE

HOME

WORK

SOCIAL SECURITY NO. (MANDATORY)

SEX

○ MALE

○ FEMALE

DATE OF BIRTH

MONTH/DAY/YEAR

EDUCATION (HIGHEST LEVEL AND YEAR)

□ HIGH SCHOOL DIPLOMA

□ TRADE CERTIFICATE

□ COLLEGE - NO DEGREE

□ ASSOCIATE'S DEGREE

□ BACHELOR'S DEGREE

□ MASTER'S DEGREE

□ PROFESSIONAL DEGREE

□ PH.D.

□ OTHER DOCTORATE

EMPLOYEE SIGNATURE

SUBMIT FORM

PRINT FORM

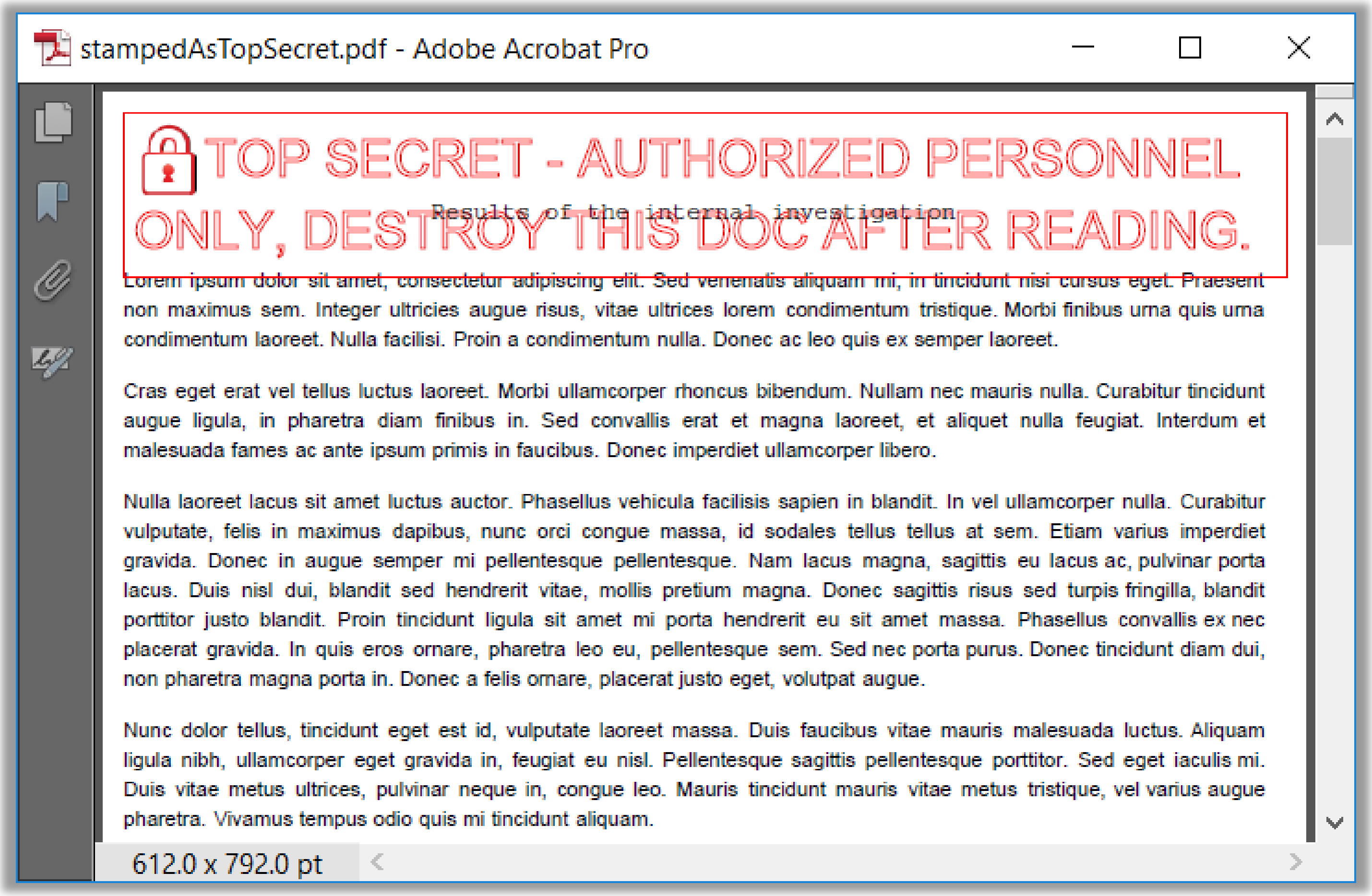
612.0 x 792.0 pt

Pic. 5 Annotations added using FDF

A more complex example shows how to add watermark with image to the PDF document.

```
private static void AddWatermarkUsingFDF()
{
    // open pdf document that later can be used
    // as a target for applying FDF
    using (Stream inputStream = File.Open("../../data/topSecretDocument.pdf", FileMode.Open),
        outputStream = File.Create("stampedAsTopSecret.pdf"))
    {
        using (FixedDocument targetPDFDocument = new FixedDocument(inputStream))
        {
            // prepare our forms data document
            using (FormsDataDocument fdfDocument = new FormsDataDocument())
            {
                // register image resource for future use
                fdfDocument.ResourceManager.RegisterResource(
                   new Image("lock","../../data/lock.png",true));

                // set default file this document applies to
                fdfDocument.File = "topSecretDocument.pdf";

                double xOffsetFromLeft = 10;
                double yOffsetFromTop = 10;
                double contentHeight = 85;
                double contentWidth = Boundaries.Letter.Width - 20;

                // add watermark annotation here
                WatermarkAnnotation watermarkAnnotation =
                    new WatermarkAnnotation(
                        new Boundary(xOffsetFromLeft,
                            Boundaries.Letter.Height-yOffsetFromTop-contentHeight,
                            xOffsetFromLeft+contentWidth, Boundaries.Letter.Height-yOffsetFromTop ),
                            AnnotationFlags.Default,
                             new AnnotationBorderStyle());

                // generate custom content
                FixedContent annotationContent = new FixedContent(Guid.NewGuid().ToString("N"),
                     new Boundary(0,0,watermarkAnnotation.Boundary.Width,
                     watermarkAnnotation.Boundary.Height));

                // create fixed size section
                Section section = new Section
                {
                    Border = new Border(1),
                    BorderColor = RgbColors.Red,
                    Width = contentWidth,
                    Height = contentHeight,
                    Padding = new Thickness(5),
                    LineHeight = 36,
                };

                // create text content
                TextBlock text = new TextBlock("TOP SECRET - AUTHORIZED PERSONNEL ONLY,
                     DESTROY THIS DOC AFTER READING.")
                {
                    Color = RgbColors.Red,
                    Font = new Apitron.PDF.Kit.Styles.Text.Font("Arial",26),
                    TextRenderingMode = TextRenderingMode.Stroke,
                    VerticalAlign = VerticalAlign.Middle
                };

                // add an image to the annotation's content,
                // set its float property to let the text flow.
                Image image = new Image("lock")
                {
                    Float = Float.Left
                };
                section.Add(image);
                section.Add(text);

                // set the content
                annotationContent.Content.AppendContentElement(section,
                     annotationContent.Boundary.Width,
                     annotationContent.Boundary.Height);
                watermarkAnnotation.Watermark = annotationContent;
                fdfDocument.Annotations.Add(watermarkAnnotation);

                // optionally save the FDF, it can be opened later and used
                // for applying the content to PDF files
                using (FileStream fileStream = File.Create("topSecretAnnotation.fdf"))
                {
                    fdfDocument.Save(fileStream);
                }

                // Apply FDF and save the result. It can be also applied using Adobe PDF Reader,
                // just open it and it will be applied automatically using the the file name
                // set to fdfDocument.File property.
                targetPDFDocument.ApplyFormsDataDocument(fdfDocument);

                targetPDFDocument.Save(outputStream);
            }
        }
    }
}
```

The resulting document is shown below:



TOP SECRET - AUTHORIZED PERSONNEL ONLY, DESTROY THIS DOC AFTER READING.

Results of the internal investigation

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed venenatis aliquam mi, in tincidunt nisi cursus eget. Praesent non maximus sem. Integer ultricies augue risus, vitae ultrices lorem condimentum tristique. Morbi finibus urna quis urna condimentum laoreet. Nulla facilisi. Proin a condimentum nulla. Donec ac leo quis ex semper laoreet.

Cras eget erat vel tellus luctus laoreet. Morbi ullamcorper rhoncus bibendum. Nullam nec mauris nulla. Curabitur tincidunt augue ligula, in pharetra diam finibus in. Sed convallis erat et magna laoreet, et aliquet nulla feugiat. Interdum et malesuada fames ac ante ipsum primis in faucibus. Donec imperdiet ullamcorper libero.

Nulla laoreet lacus sit amet luctus auctor. Phasellus vehicula facilisis sapien in blandit. In vel ullamcorper nulla. Curabitur vulputate, felis in maximus dapibus, nunc orci congue massa, id sodales tellus tellus at sem. Etiam varius imperdiet gravida. Donec in augue semper mi pellentesque pellentesque. Nam lacus magna, sagittis eu lacus ac, pulvinar porta lacus. Duis nisl dui, blandit sed hendrerit vitae, mollis pretium magna. Donec sagittis risus sed turpis fringilla, blandit porttitor justo blandit. Proin tincidunt ligula sit amet mi porta hendrerit eu sit amet massa. Phasellus convallis ex nec placerat gravida. In quis eros ornare, pharetra leo eu, pellentesque sem. Sed nec porta purus. Donec tincidunt diam dui, non pharetra magna porta in. Donec a felis ornare, placerat justo eget, volutpat augue.

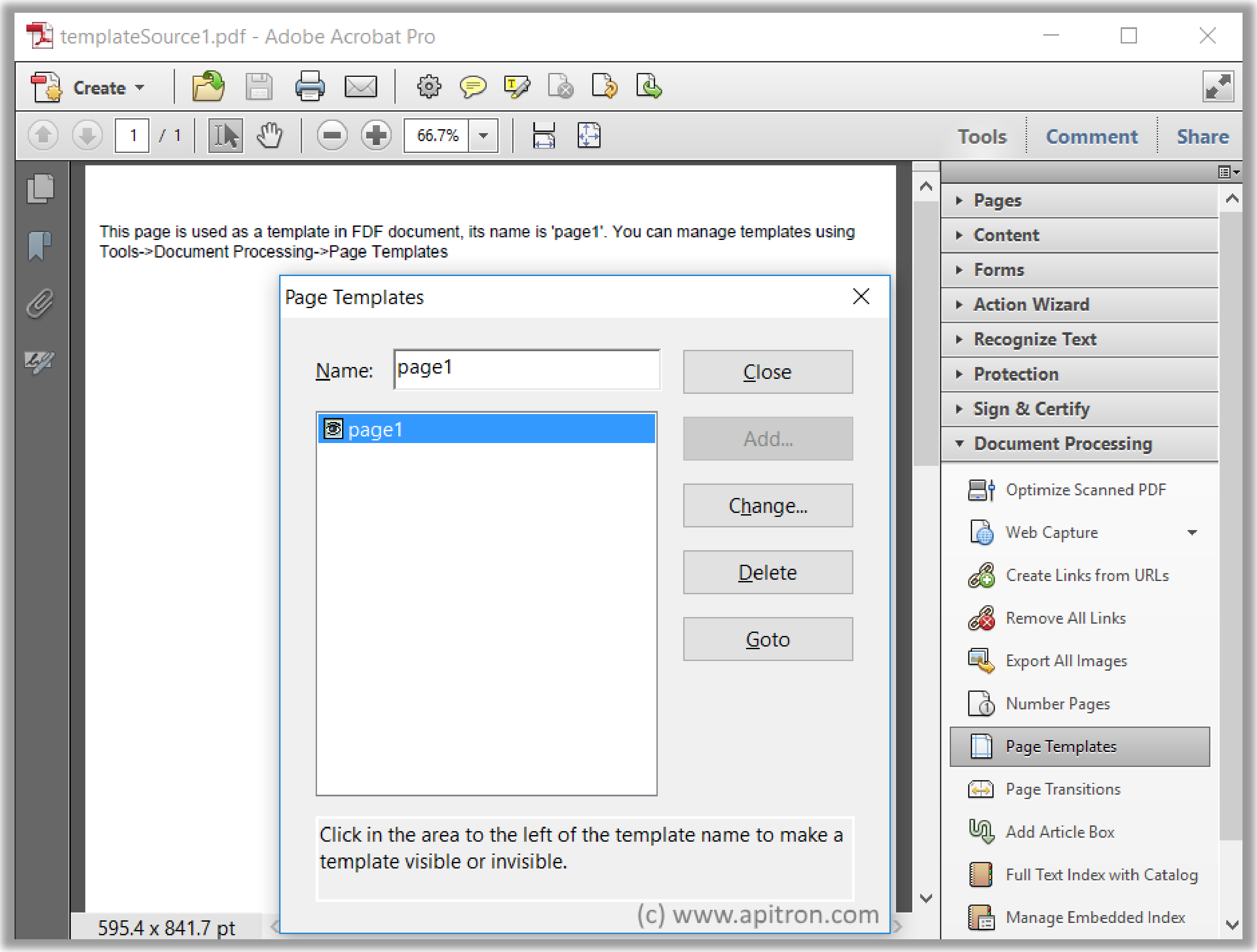
Nunc dolor tellus, tincidunt eget est id, vulputate laoreet massa. Duis faucibus vitae mauris malesuada luctus. Aliquam ligula nibh, ullamcorper eget gravida in, feugiat eu nisl. Pellentesque sagittis pellentesque porttitor. Sed eget iaculis mi. Duis vitae metus ultrices, pulvinar neque in, congue leo. Mauris tincidunt mauris vitae metus tristique, vel varius augue pharetra. Vivamus tempus odio quis mi tincidunt aliquam.

Pic. 6 Watermark added using FDF

Insert pages to PDF documents using FDF and other PDFs as templates

FDF can be used as an intermediate link between the target PDF file and other PDFs acting as template sources for adding pages. Used this way the FDF will contain links to pages in other PDF documents making you able to assemble PDFs from many sources for particular purpose. You can even combine several pages from different files to one page if needed.

For it to work you should prepare source PDFs by marking desired pages as named templates. You can also define whether the template page can be visible or not for regular users. The image below shows how to set the page as template using Adobe Acrobat.



templateSource1.pdf - Adobe Acrobat Pro

This page is used as a template in FDF document, its name is 'page1'. You can manage templates using Tools->Document Processing->Page Templates

Page Templates

Name: page1

page1

Click in the area to the left of the template name to make a template visible or invisible.

595.4 x 841.7 pt

(c) www.apitron.com

Pic. 7 Add page template using Adobe Acrobat

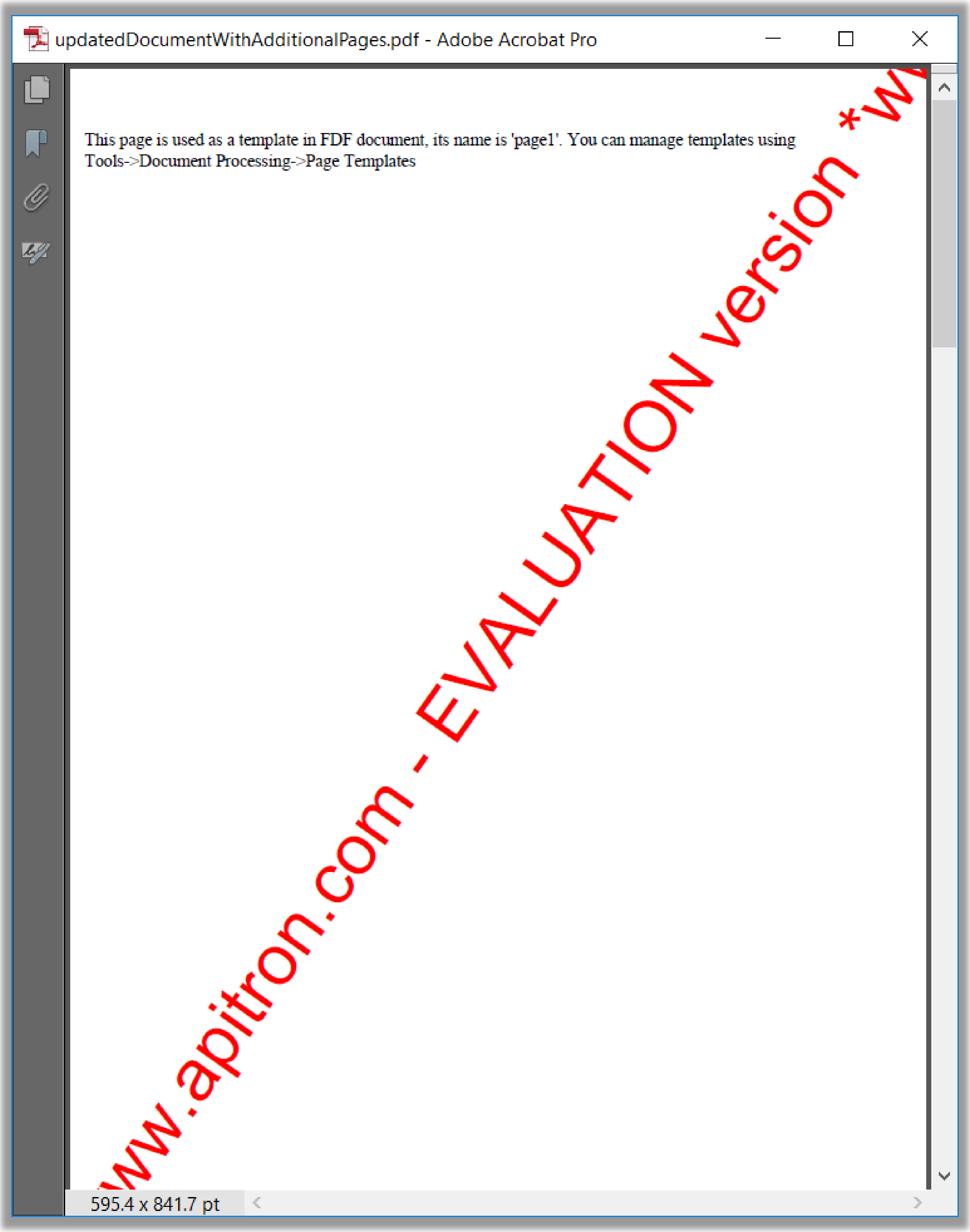
Alternatively you can do it in code, using Apitron PDF Kit:

```
// shows how to create PDF document and define page template
private static void CreatePDFDocumentAndSetTemplateName()
{
    // create PDF doc
    using (FixedDocument doc = new FixedDocument())
    {
        // create new page and add some content
        Page page = new Page();
        page.Content.Translate(10, 700);
        page.Content.AppendContentElement(
            new TextBlock(@"This page is used as a template in FDF document, its name is 'page1'.
              You can manage templates using Tools->Document Processing->Page Templates")
            { Color = RgbColors.Red }, 580, 100);

        doc.Pages.Add(page);
        // define the template by providing its name and page it refers to
        doc.Names.Pages.Add("page1",page);

        using (Stream stream = File.Create("templateSource.pdf"))
        {
            doc.Save(stream);
        }
    }
}
```

Here we just create the PDF file and add a named template to its visible templates collection (FixedDocument.Names.Pages). You can also create invisible template by adding the name and page to FixedDocument.Names.Templates collection. Having several source PDF files with defined templates we can create FDF document and use it to add all referenced pages to the target PDF document. See the code below:

```
private static void AddPagesUsingFDF()
{
    // create an empty PDF document (or you can an existing one)
    using (FixedDocument target = new FixedDocument())
    {
        // create and fill new FDF document
        using (FormsDataDocument fdfDoc = new FormsDataDocument())
        {
            // create template using first template source
            FdfNamedPageReference pageReference1 = new FdfNamedPageReference("page1",
                new FileSpecification("../../data/templateSource1.pdf"));
            FdfTemplate template1 = new FdfTemplate(pageReference1);

            // create template using second template source
            FdfNamedPageReference pageReference2 = new FdfNamedPageReference("page2",
                new FileSpecification("../../data/templateSource2.pdf"));
            FdfTemplate template2 = new FdfTemplate(pageReference2);

            // add pages based on templates created above
            fdfDoc.Pages.Add(new FdfPage(new[] {template1}));
            fdfDoc.Pages.Add(new FdfPage(new[] {template2}));
            // add combined page, its content is composed using several templates
            fdfDoc.Pages.Add(new FdfPage(new[] {template1, template2}));

            target.ApplyFormsDataDocument(fdfDoc);
        }

        using (Stream outputStream = File.Create("updatedDocumentWithAdditionalPages.pdf"))
        {
            target.Save(outputStream);
        }
    }
}
```

The resulting file contains 3 pages taken from: first PDF, second PDF and combined page composed used previous two. See the results below:

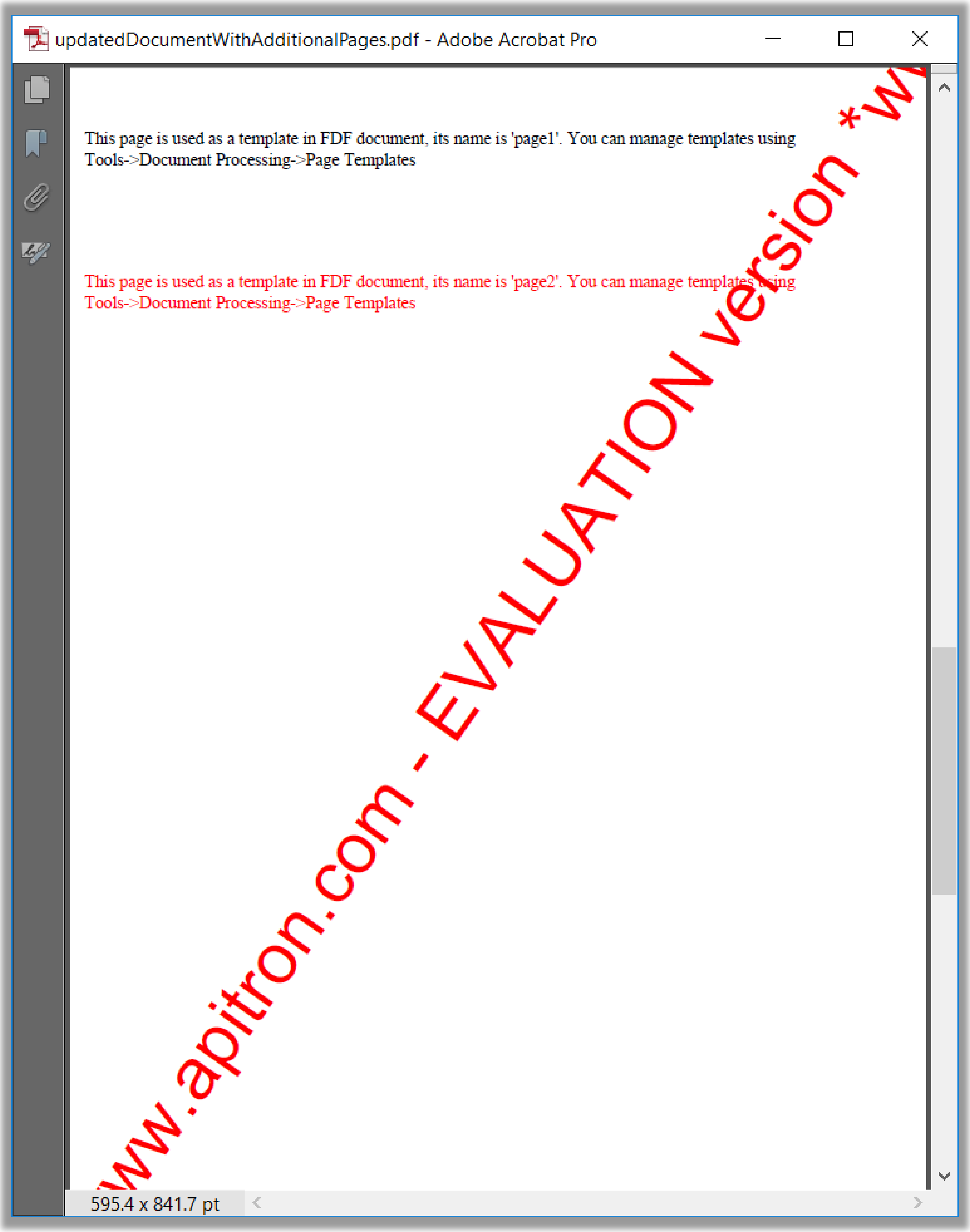


updatedDocumentWithAdditionalPages.pdf - Adobe Acrobat Pro

This page is used as a template in FDF document, its name is 'page1'. You can manage templates using Tools->Document Processing->Page Templates

www.apitron.com - EVALUATION version *w
595.4 x 841.7 pt

Pic. 8 PDF file created by combining several files in FDF document, first page



updatedDocumentWithAdditionalPages.pdf - Adobe Acrobat Pro

This page is used as a template in FDF document, its name is 'page1'. You can manage templates using Tools->Document Processing->Page Templates

This page is used as a template in FDF document, its name is 'page2'. You can manage templates using Tools->Document Processing->Page Templates

Pic. 9 Combined PDF file, page 3 created by merging two templates into one page

Summary

In a highly loaded environment with lots of PDF forms to manage, one van clearly benefit from using FDF as the forms' data storage. When it comes to assembling files from different sources, FDF sometimes may be simpler and also more convenient than other methods. It easily automates this task and can be used by end users as input for PDF processing applications. Apitron PDF Kit provides you with an ability to create and manage complex FDF and PDF documents using its easy to use and well-though API. It's a cross-platform library and can be used to create apps targeting iOS and Android (via Xamarin) as well as Windows-based and MONO platforms. Contact us and we'll be happy to answer your questions regarding our products and provide assistance if needed.