Create PDF invoice in Windows Forms application

Written by Apitron Documentation Team

Introduction

Very common scenario where our .NET PDF library comes into play is invoices or reports generation. With fixed and flow layout API offered by Apitron PDF Kit for .NET, such tasks become quickly solvable.

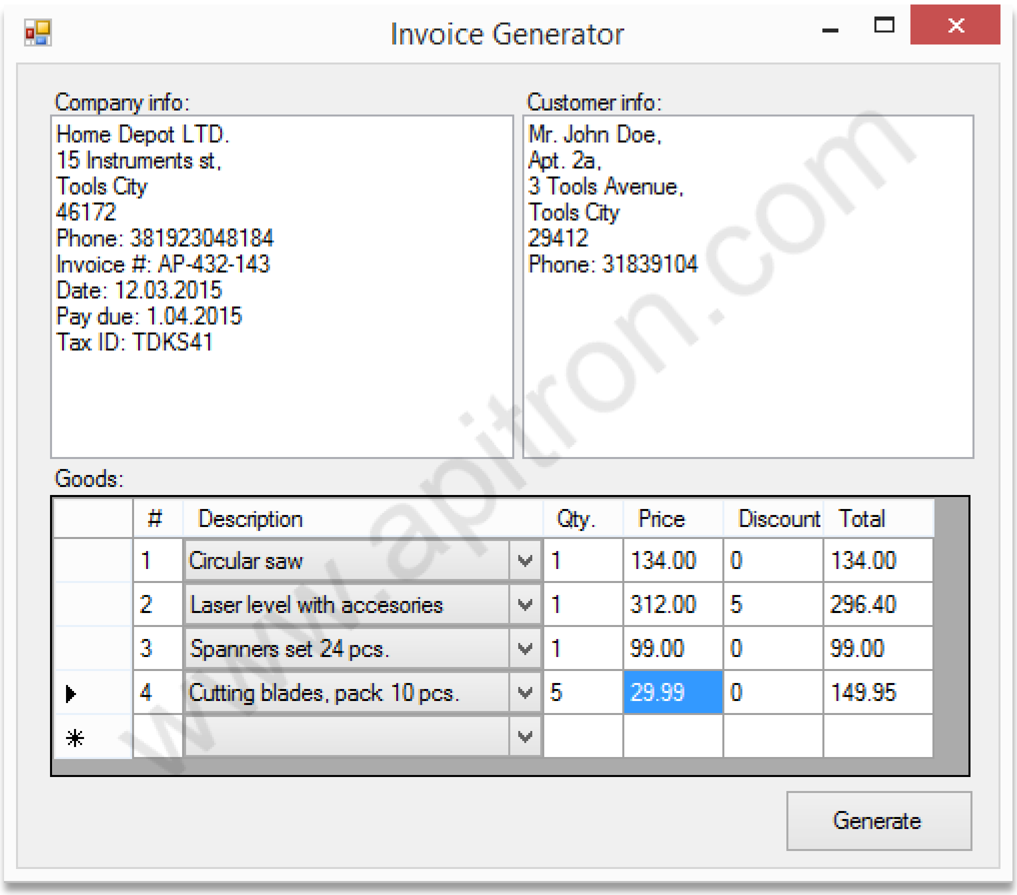
This post demonstrates how to create basic Windows Forms PDF invoice creation application, similar to the one many companies use on their daily basis. It generates PDF invoice based on entered company and customer info using Apitron PDF Kit .NET component.

The complete sample can be found in *Samples\Windows Forms* folder inside the download package available on our website.

Solution overview

Our application uses Windows Forms and is written in C#, it has just a single window with all necessary controls in place. It represents an invoicing tool for imaginary company called "Home Depot" which is selling various tools for home use.

Company and customer information should be entered in multiline textboxes and products list should be filled using the data grid below. When the user finishes with the invoice, he or she clicks the *Generate* button and gets branded PDF invoice created. Image below demonstrates the running app main window with entered information.



Company info:

Home Depot LTD.
15 Instruments st,
Tools City
46172
Phone: 381923048184
Invoice #: AP-432-143
Date: 12.03.2015
Pay due: 1.04.2015
Tax ID: TDKS41

Customer info:

Mr. John Doe,
Apt. 2a,
3 Tools Avenue,
Tools City
29412
Phone: 31839104

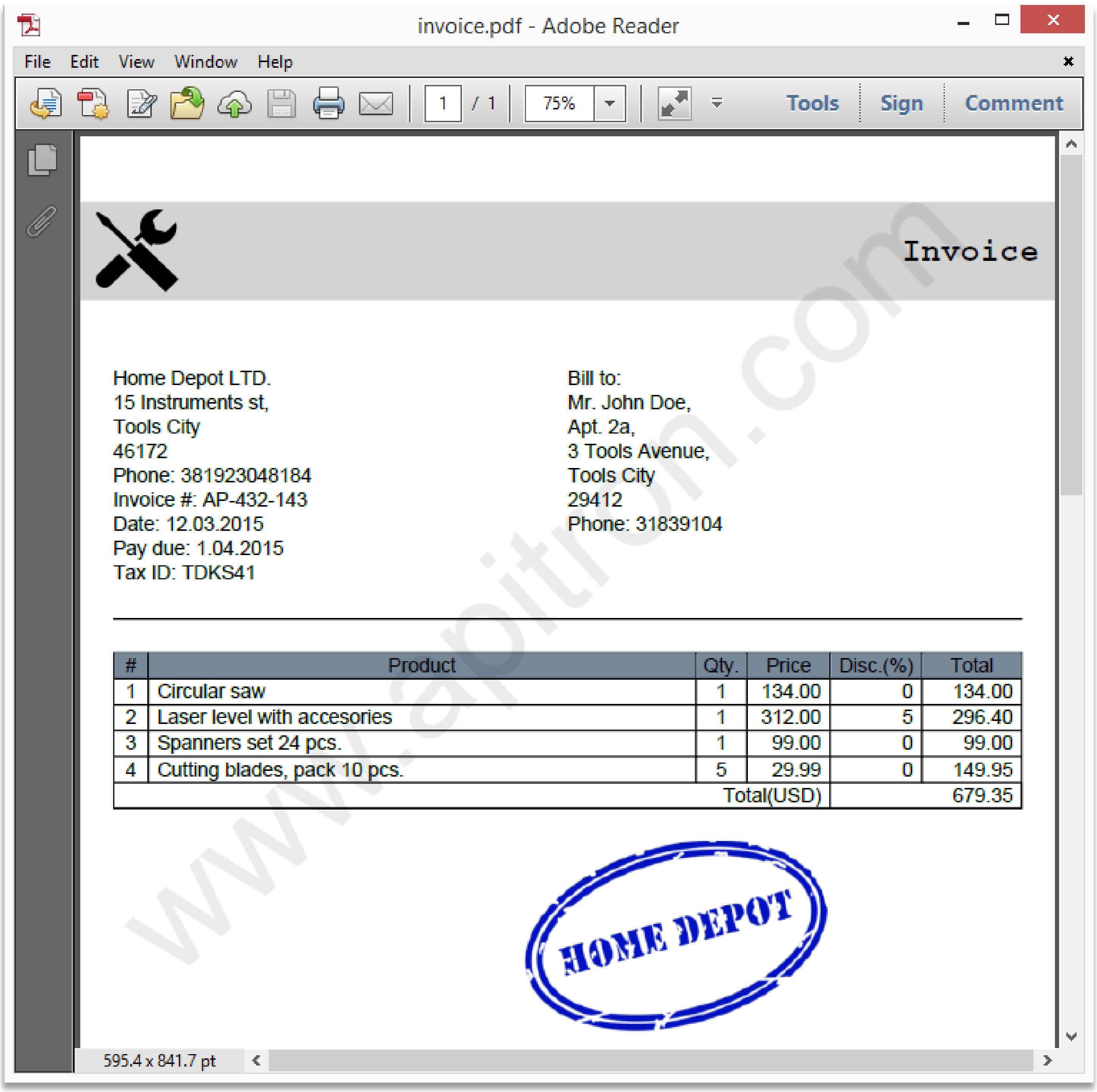
Goods:

#	Description	Qty.	Price	Discount	Total
1	Circular saw	1	134.00	0	134.00
2	Laser level with accesories	1	312.00	5	296.40
3	Spanners set 24 pcs.	1	99.00	0	99.00
4	Cutting blades, pack 10 pcs.	5	29.99	0	149.95

Pic. 1 PDF Invoice generator application, main window

As you can see from this screenshot the controls layout is very basic, but as it's not the main point in this example, we decided to keep it simple and paid the most attention to the PDF generation part of the app.

Here you can see the resulting invoice generated from data set shown on the first picture. As an addition to the entered data, it contains the company logo in the page header and stamp image. See the image below:



Invoice

Home Depot LTD.
15 Instruments st,
Tools City
46172
Phone: 381923048184
Invoice #: AP-432-143
Date: 12.03.2015
Pay due: 1.04.2015
Tax ID: TDKS41

Bill to:
Mr. John Doe,
Apt. 2a,
3 Tools Avenue,
Tools City
29412
Phone: 31839104

#	Product	Qty.	Price	Disc.(%)	Total
1	Circular saw	1	134.00	0	134.00
2	Laser level with accesories	1	312.00	5	296.40
3	Spanners set 24 pcs.	1	99.00	0	99.00
4	Cutting blades, pack 10 pcs.	5	29.99	0	149.95
			Total(USD)		679.35

HOME DEPOT

Pic. 2 Created PDF invoice

The code

We won't dive into windows forms specifics here and focus only on PDF generation and related code. A few functions responsible for this are listed below:

```
/// <summary>
/// Generates the invoice based on entered data.
/// </summary>
/// <param name="stream">Stream to save the resulting pdf into.</param>
private void GenerateInvoice(Stream stream)
{
    // base path for images
    string imagesPath = @"..\..\images";
    // create document and register styles
    FlowDocument document = new FlowDocument();
    /* style for products table header, assigned via type + class selectors */
    document.StyleManager.RegisterStyle("gridrow.tableHeader",
        new Style() {Background = RgbColors.LightSlateGray});

    /* style matching all cells in rows with class "centerAlignedCells" set
        and all cells in rows with class "centerAlignedCell" set */
    document.StyleManager.RegisterStyle("gridrow.centerAlignedCells > *, gridrow >
*.centerAlignedCell",
        new Style() {Align = Align.Center, Margin = new Thickness(0)});

    /* style matching all elements in rows with class "leftAlignedCell" set */
    document.StyleManager.RegisterStyle("gridrow > *.leftAlignedCell",
        new Style() {Align = Align.Left, Margin = new Thickness(5, 0, 0, 0)});

    /* default style for any cell in any grid row, assigned via type + child selectors, makes it right
aligned */
    document.StyleManager.RegisterStyle("gridrow > *",
        new Style() {Align = Align.Right, Margin = new Thickness(0, 0, 5, 0)});

    // create resource manager and register image resources
    ResourceManager resourceManager = new ResourceManager();
    resourceManager.RegisterResource(new Apitron.PDF.Kit.FixedLayout.Resources.XObjects.Image("logo",
        Path.Combine(imagesPath, "storeLogo.png"), true) {Interpolate = true});
    resourceManager.RegisterResource(new Apitron.PDF.Kit.FixedLayout.Resources.XObjects.Image("stamp",
        Path.Combine(imagesPath, "stamp.png"), true) {Interpolate = true});
    // construct page header which includes store logo and the text "Invoice"
    document.PageHeader.Margin = new Thickness(0, 40, 0, 20);
    document.PageHeader.Padding = new Thickness(10, 0, 10, 0);
    document.PageHeader.Height = 120;
    document.PageHeader.Background = RgbColors.LightGray;
    document.PageHeader.LineHeight = 60;
    document.PageHeader.Add(new Image("logo") {Height = 50, Width = 50, VerticalAlign =
VerticalAlign.Middle});
    document.PageHeader.Add(new TextBlock("Invoice")
                            {
                                Display = Display.InlineBlock,
                                Align = Align.Right,
                                Font = new Font(StandardFonts.CourierBold, 20),
                                Color = RgbColors.Black
                            });
```

…the code continues on next page

```
    // page content section with padding
    Section pageSection = new Section() {Padding = new Thickness(20)};

    // add company info section
    pageSection.AddItems(
        CreateInfoSubsections(new string[] {txtCompany.Text, "Bill to:\r\n" + txtCustomerInfo.Text}));

    // add horizontal line for visual separation
    pageSection.Add(new Hr() {Padding = new Thickness(0, 20, 0, 20)});

    // add products grid
    pageSection.Add(CreateProductsGrid());

    // add new line after grid
    pageSection.Add(new Br {Height = 20});

    // insert empty padding section and stamp image
    pageSection.Add(new Section() {Width = 250, Display = Display.InlineBlock});
    pageSection.Add(new Image("stamp"));

    // add page section into document
    document.Add(pageSection);
    // save document to stream
    document.Write(stream, resourceManager, new PageBoundary(Boundaries.A4));
}
```

Creates information sections located above the products table:

```
/// <summary>
/// Creates several info sections side by side based on given list of strings.
/// </summary>
/// <returns>
/// List of created sections with textual information.
/// </returns>
private IList<Section> CreateInfoSubsections(string[] info)
{
    List<Section> createdSections = new List<Section>();
    double width = 100.0/info.Length;

    for (int i = 0; i < info.Length; i++)
    {
        Section section = new Section() {Width = Length.FromPercentage(width),
Display = Display.InlineBlock};

        using (StringReader reader = new StringReader(info[i]))
        {
            string currentLine = null;

            while ((currentLine = reader.ReadLine()) != null)
            {
                section.Add(new TextBlock(currentLine));
                section.Add(new Br());
            }
        }
        createdSections.Add(section);
    }

    return createdSections;
}
```

Creates products grid:

```
/// <summary>
/// Creates products grid.
/// </summary>
private Grid CreateProductsGrid()
{

    // create grid content element and its define columnts
    Grid productsGrid = new Grid(20, Length.Auto, 30, 50, 55, 60);
    // add header row
    productsGrid.Add(new GridRow(new TextBlock("#"), new TextBlock("Product"), new TextBlock("Qty."),
        new TextBlock("Price"), new TextBlock("Disc.(%)"), new TextBlock("Total"))
                        {
                            Class = "tableHeader centerAlignedCells"
                        });

    Decimal invoiceTotal = 0;

    // enumerate the list of products and create grid rows
    foreach (ProductEntry product in products)
    {
        TextBlock pos = new TextBlock(product.Pos.ToString()) {Class = "centerAlignedCell"};
        TextBlock description = new TextBlock(product.Description) {Class = "leftAlignedCell"};
        TextBlock qty = new TextBlock(product.Qty.ToString()) {Class = "centerAlignedCell"};
        TextBlock price = new TextBlock(product.Price.ToString(CultureInfo.InvariantCulture));
        TextBlock discount = new TextBlock(product.Discount.ToString(CultureInfo.InvariantCulture));
        TextBlock total = new TextBlock(product.Total.ToString(CultureInfo.InvariantCulture));

        productsGrid.Add(new GridRow(pos, description, qty, price, discount, total));
        invoiceTotal += product.Total;
    }

    // append "total" row
    productsGrid.Add(new GridRow(new TextBlock("Total(USD)") {ColSpan = 4},
        new TextBlock(invoiceTotal.ToString(CultureInfo.InvariantCulture)) {ColSpan = 2}));
    return productsGrid;
}
```

Definitions for all used styles and their descriptions can be found at the beginning of the `GenerateInvoice` function which is responsible for combing all parts of the invoice together.

Conclusion

Using small piece of code we managed to create a fully functional Windows Forms PDF invoice generator. Apitron PDF Kit offers unlimited capabilities for developers interested in PDF processing and a good way to start is to read the book, describing it step by step with code samples and references to original specification. This free book is available for download from our website.

We created plenty of C# samples for various platforms and included them in our download package. These samples demonstrate every aspect of PDF processing and, sometimes, can be adapted to your applications with minimal modifications. Windows desktop, Windows Phone, Windows Store, Xamarin iOS and Android, .NET/Mono – our component covers them all. We also offer free evaluation and royalty-free licensing. Contact us if you have any questions and we'll do our best to help you.