

در حالت کلی، هر شیءایی را که بتوان تبدیل به تصویر کرد، قابلیت قرارگیری در یک فایل PDF را هم خواهد داشت. از این نمونه می‌توان به اشیاء [MSChart](#) اشاره کرد که از دات نت 4 جزئی از کتابخانه‌های اصلی دات نت شده‌اند و البته برای دات نت سه و نیم نیز به صورت جداگانه قابل دریافت است.

در ادامه مثالی را بررسی خواهیم کرد که بر اساس ردیف‌های گزارش آن، یک نمودار، به انتهای گزارش اضافه خواهد شد. بدهای کامل این مثال را در ذیل مشاهده می‌کنید:

```
using System;
using System.Collections.Generic;
using PdfReportSamples.Models;
using PdfRpt.Core.Contracts;
using PdfRpt.Core.Helper;
using PdfRpt.FluentInterface;

namespace PdfReportSamples.ChartImage
{
    public class ChartImagePdfReport
    {
        public IPdfReportData CreatePdfReport()
        {
            var chart = new MSChartHelper
            {
                AxisXTitle = "User",
                AxisYTitle = "Balance",
                ChartTitle = "Users Balance",
                AxisTitleFont = new System.Drawing.Font("Tahoma", 12f),
                LabelStyleFont = new System.Drawing.Font("Tahoma", 10f),
                ChartTitleFont = new System.Drawing.Font("Arial", 16f,
                    System.Drawing.FontStyle.Bold),
                LegendsFont = new System.Drawing.Font("Tahoma", 12f)
            };

            return new PdfReport().DocumentPreferences(doc =>
            {
                doc.RunDirection(PdfRunDirection.RightToLeft);
                doc.Orientation(PageOrientation.Portrait);
                doc.PageSize(PdfPageSize.A4);
                doc.DocumentMetadata(new DocumentMetadata { Author = "Vahid", Application = "PdfRpt",
                    Keywords = "Test", Subject = "Test Rpt", Title = "Test" });
            }).DefaultFonts(fonts =>
            {
                fonts.Path(string.Format("{0}\\fonts\\irsans.ttf", AppPath.ApplicationPath),
                    string.Format("{0}\\fonts\\verdana.ttf",
                        Environment.GetEnvironmentVariable("SystemRoot")));
            }).PagesFooter(footer =>
            {
                footer.DefaultFooter(printDate: DateTime.Now.ToString("MM/dd/yyyy"));
            }).PagesHeader(header =>
            {
                header.DefaultHeader(defaultHeader =>
                {
                    defaultHeader.ImagePath(AppPath.ApplicationPath + "\\Images\\01.png");
                    defaultHeader.Message("گزارش جدید ما");
                });
            }).MainTableTemplate(template =>
            {
                template.BasicTemplate(BasicTemplate.ClassicTemplate);
            }).MainTablePreferences(table =>
            {
                table.ColumnsWidthsType(TableColumnWidthType.Relative);
            }).MainTableDataSource(dataSource =>
            {
                var listOfRows = new List<User>();
                for (var i = 0; i < 7; i++)
```

```

        {
            listOfRows.Add(new User { Id = i, LastName = "نام خانوادگی" + i, Name = "نام" + i,
Balance = (i * 50) + 1000 });
        }
        dataSource.StronglyTypedList(listOfRows);
    })
    .MainTableEvents(events =>
    {
        events.DataSourceIsEmpty(message: "There is no data available to display.");
        events.DocumentOpened(args =>
        {
            chart.ChartInit(width: (int)args.PdfWriter.PageSize.Width - 100, height: 250);
        });
        events.RowAdded(args =>
        {
            if (args.RowType == RowType.DataTableRow)
            {
                var name = args.TableRowData.GetValueOf<User>(x => x.Name);
                if (name == null) return;

                var balance = args.TableRowData.GetValueOf<User>(x => x.Balance);
                if (balance == null) return;

                chart.AddXY(name, balance);
            }
        });
        events.DocumentClosing(args =>
        {
            chart.AddChartToPage(args.PdfDoc);
            chart.FreeResources();
        });
    })
    .MainTableSummarySettings(summary =>
    {
        summary.OverallSummarySettings("جمع");
        summary.PreviousPageSummarySettings("نقل از صفحه قبل");
    })
    .MainTableColumns(columns =>
    {
        columns.AddColumn(column =>
        {
            column.PropertyName("rowNo");
            column.IsRowNumber(true);
            column.CellsHorizontalAlignment(HorizontalAlignment.Center);
            column.IsVisible(true);
            column.Order(0);
            column.Width(1);
            column.HeaderCell("ردیف", captionRotation: 90);
        });

        columns.AddColumn(column =>
        {
            column.PropertyName<User>(x => x.Id);
            column.CellsHorizontalAlignment(HorizontalAlignment.Center);
            column.IsVisible(true);
            column.Order(1);
            column.Width(2);
            column.HeaderCell("شماره");
        });

        columns.AddColumn(column =>
        {
            column.PropertyName<User>(x => x.Name);
            column.CellsHorizontalAlignment(HorizontalAlignment.Center);
            column.IsVisible(true);
            column.Order(2);
            column.Width(2);
            column.HeaderCell("نام");
        });

        columns.AddColumn(column =>
        {
            column.PropertyName<User>(x => x.LastName);
            column.CellsHorizontalAlignment(HorizontalAlignment.Center);
            column.IsVisible(true);
            column.Order(3);
            column.Width(3);
            column.HeaderCell("نام خانوادگی");
        });

        columns.AddColumn(column =>

```

```

        {
            column.PropertyName<User>(x => x.Balance);
            column.HeaderCell("موجودی");
            column.ColumnItemsTemplate(template =>
            {
                template.TextBlock();
                template.DisplayFormatFormula(obj => obj == null ? string.Empty :
string.Format("{0:n0}", obj));
            });
            column.Width(2);
            column.AggregateFunction(aggregateFunction =>
            {
                aggregateFunction.NumericAggregateFunction(AggregateFunction.Sum);
                aggregateFunction.DisplayFormatFormula(obj => obj == null ? string.Empty :
string.Format("{0:n0}", obj));
            });
            column.CellsHorizontalAlignment(HorizontalAlignment.Center);
            column.IsVisible(true);
            column.Order(4);
        });
    });
    .Generate(data => data.AsPdfFile(AppPath.ApplicationPath + "\\Pdf\\RptChartSample.pdf"));
}
}
}

```

برای تهیه این گزارش و افزودن نمودار به آن، از کلاس کمکی ذیل استفاده شده است:

```

using System.Drawing;
using System.IO;
//It's part of the .NET 4.0+ now.
using System.Windows.Forms.DataVisualization.Charting;
using iTextSharp.text;
using iTextSharp.text.pdf;
using PdfRpt.Core.Helper;

namespace PdfReportSamples.ChartImage
{
    public class MSChartHelper
    {
        // MS Chart learning tutorials:
        // http://weblogs.asp.net/scottgu/archive/2010/02/07/built-in-charting-controls-vs-2010-and-
net-4-series.aspx
        Chart _chart;

        public System.Drawing.Font AxisTitleFont { set; get; }

        public string AxisXTitle { set; get; }

        public string AxisYTitle { set; get; }

        public string ChartTitle { set; get; }

        public System.Drawing.Font ChartTitleFont { set; get; }

        public System.Drawing.Font LabelStyleFont { set; get; }

        public System.Drawing.Font LegendsFont { set; get; }

        public void AddChartToPage(Document pdfDoc,
            int scalePercent = 100,
            float spacingBefore = 20,
            float spacingAfter = 10,
            float widthPercentage = 80)
        {
            using (var chartimage = new MemoryStream())
            {
                _chart.SaveImage(chartimage, ChartImageFormat.Bmp); //BMP gives the best compression
result

                var iTextSharpImage =
PdfImageHelper.GetITextSharpImageFromByteArray(chartimage.GetBuffer());
                iTextSharpImage.ScalePercent(scalePercent);
                iTextSharpImage.Alignment = Element.ALIGN_CENTER;

                var table = new PdfPTable(1)
                {
                    WidthPercentage = widthPercentage,

```

```

        SpacingBefore = spacingBefore,
        SpacingAfter = spacingAfter
    };
    table.AddCell(iTextSharpImage);

    pdfDoc.Add(table);
}
}

public void AddXY(object xValue, params object[] yValue)
{
    _chart.Series[0].Points.AddXY(xValue, yValue);
}

public void ChartInit(int width, int height)
{
    _chart = new Chart
    {
        Width = width,
        Height = height,
        AntiAliasing = AntiAliasingStyles.All,
        TextAntiAliasingQuality = TextAntiAliasingQuality.High,
        Palette = ChartColorPalette.BrightPastel,
        BackColor = ColorTranslator.FromHtml("#F3DFC1"),
        BackGradientStyle = GradientStyle.TopBottom
    };

    setBorder();
    setTitles();
    setChartAreas();
    setLegends();
    setSeries();
}

public void FreeResources()
{
    if (_chart != null && !_chart.IsDisposed)
        _chart.Dispose();
}

private void setBorder()
{
    _chart.BorderSkin.SkinStyle = BorderSkinStyle.Emboss;
    _chart.BorderlineWidth = 2;
    _chart.BorderlineColor = Color.FromArgb(181, 64, 1);
    _chart.BorderlineDashStyle = ChartDashStyle.Solid;
}

private void setChartAreas()
{
    _chart.ChartAreas.Add("ChartArea1");
    _chart.ChartAreas[0].AxisX.Title = AxisXTitle;
    _chart.ChartAreas[0].AxisY.Title = AxisYTitle;
    _chart.ChartAreas[0].AxisX.TitleFont = AxisTitleFont;
    _chart.ChartAreas[0].AxisY.TitleFont = AxisTitleFont;
    _chart.ChartAreas[0].AxisX.LabelStyle.Font = LabelStyleFont;
    _chart.ChartAreas[0].AxisX.LabelStyle.Angle = -90;
    _chart.ChartAreas[0].BackColor = Color.White;
    _chart.ChartAreas[0].AxisX.LineColor = Color.FromArgb(64, 64, 64);
    _chart.ChartAreas[0].AxisX.MajorGrid.LineColor = Color.FromArgb(64, 64, 64);
    _chart.ChartAreas[0].AxisY.LineColor = Color.FromArgb(64, 64, 64);
    _chart.ChartAreas[0].AxisY.MajorGrid.LineColor = Color.FromArgb(64, 64, 64);
}

private void setLegends()
{
    _chart.Legends.Add("Default");
    _chart.Legends[0].LegendStyle = LegendStyle.Row;
    _chart.Legends[0].IsTextAutoFit = false;
    _chart.Legends[0].DockedToChartArea = "ChartArea1";
    _chart.Legends[0].Docking = Docking.Bottom;
    _chart.Legends[0].IsDockedInsideChartArea = false;
    _chart.Legends[0].BackColor = Color.Transparent;
    _chart.Legends[0].Font = LegendsFont;
}

private void setSeries()
{
    _chart.Series.Add("");
    _chart.Series[0].ChartType = SeriesChartType.Column;
    _chart.Series[0].Palette = ChartColorPalette.EarthTones;
}

```

```

        _chart.Series[0].IsValueShownAsLabel = true;
        _chart.Series[0].IsVisibleInLegend = false;
    }

    private void setTitles()
    {
        _chart.Titles.Add(ChartTitle);
        _chart.Titles[0].Font = ChartTitleFont;
        _chart.Titles[0].TextStyle = TextStyle.Shadow;
        _chart.Titles[0].ShadowOffset = 3;
        _chart.Titles[0].ShadowColor = Color.FromArgb(32, 0, 0);
        _chart.Titles[0].Alignment = ContentAlignment.TopCenter;
        _chart.Titles[0].ForeColor = Color.FromArgb(26, 59, 105);
    }
}

```

توضیحات:

- استفاده از MSChart در اینجا از این جهت مناسب است که فراگیری کار کردن با آن عمومی بوده و در پروژه‌های وب و ویندوز کاربرد دارد و احتمالاً هم اکنون با نحوه کار کردن با آن آشنا هستید، زیرا [از سال 2010](#) به دات نت اضافه شده است.

- در این بین تنها متد جدید و مهم کلاس MSChartHelper، متد AddChartToPage آن است. به کمک متد chart.SaveImage می‌توان تصویر نهایی معادل یک نمودار را در حافظه ذخیره کرد. سپس با استفاده از متد PdfImageHelper.GetITextSharpImageFromByteArray، این تصویر موجود در حافظه را به معادل قابل استفاده آن در iTextSharp تبدیل کرده و به صفحه اضافه خواهیم کرد.

- در کدهای اصلی تولید گزارش، مقدار دهی کلاس کمکی MSChartHelper در قسمت رخدادهای قابل استفاده PdfReport در متد MainTableEvents آن انجام شده است:

در روال رویدادگردان DocumentOpened، بر اساس عرض واقعی صفحه، عرض نمودار را مشخص می‌کنیم:

```

events.DocumentOpened(args =>
{
    chart.ChartInit(width: (int)args.PdfWriter.PageSize.Width - 100, height: 250);
});

```

سپس در روال رویدادگردان RowAdded، فرصت خواهیم داشت به اطلاعات در حال افزوده شدن به گزارش دسترسی داشته باشیم. این اطلاعات را به متد افزودن XY نمودار ارسال خواهیم کرد:

```

events.RowAdded(args =>
{
    if (args.RowType == RowType.DataTableRow)
    {
        var name = args.TableRowData.GetValueOf<User>(x => x.Name);
        if (name == null) return;

        var balance = args.TableRowData.GetValueOf<User>(x => x.Balance);
        if (balance == null) return;

        chart.AddXY(name, balance);
    }
});

```

و در آخر، پیش از بسته شدن فایل PDF تولیدی (DocumentClosing)، نمودار نهایی را به صفحه اضافه کرده و منابع مرتبط با آن را آزاد خواهیم کرد:

```

events.DocumentClosing(args =>
{
    chart.AddChartToPage(args.PdfDoc);
    chart.FreeResources();
});

```

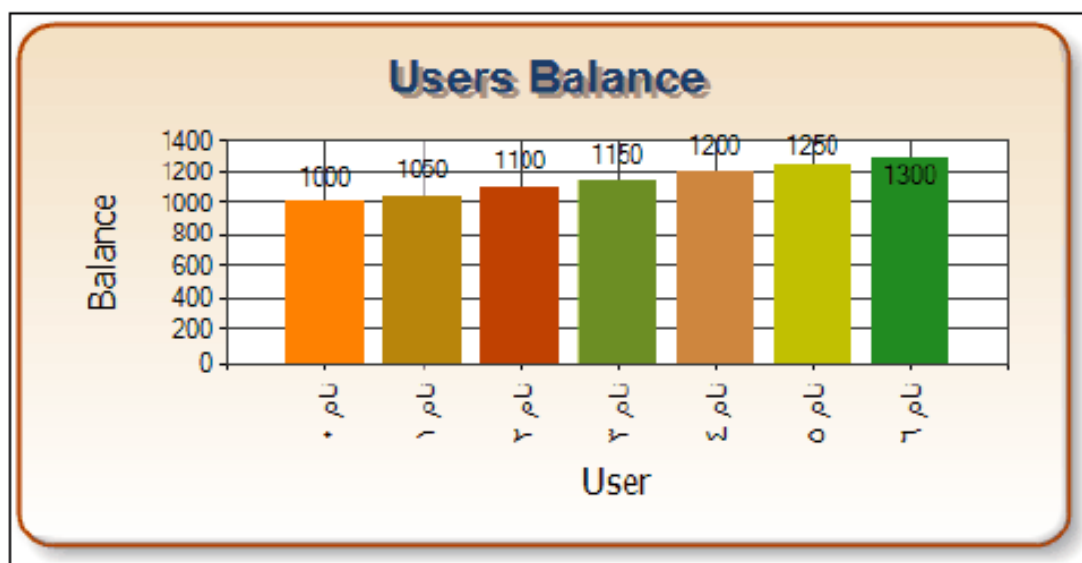
بنابراین اگر این سؤال عمومی وجود دارد که آیا می‌توان در این بین، به ابتدا و انتهای گزارش اشیایی را افزود، روش کلی آن را در

روال‌های فوق ملاحظه می‌کنید. توسط args.PdfDoc و args.PdfWriter می‌توان به زیرساخت iTextSharp دسترسی یافت.



گزارش جدید ما

ردیف	شماره	نام	نام خانوادگی	موجودی
۱	۰	نام ۰	نام خانوادگی ۰	۱,۰۰۰
۲	۱	نام ۱	نام خانوادگی ۱	۱,۰۵۰
۳	۲	نام ۲	نام خانوادگی ۲	۱,۱۰۰
۴	۳	نام ۳	نام خانوادگی ۳	۱,۱۵۰
۵	۴	نام ۴	نام خانوادگی ۴	۱,۲۰۰
۶	۵	نام ۵	نام خانوادگی ۵	۱,۲۵۰
۷	۶	نام ۶	نام خانوادگی ۶	۱,۳۰۰
جمع				۸,۰۵۰



نظرات خوانندگان

نویسنده: الهام
تاریخ: ۱۳۹۱/۰۷/۲۲ ۸:۲۵

سلام آقای نصیری

مثال مفیدی بود. من یک مشکلی داشتم با ms-chart میخوامستم از شما راهنمایی بگیرم.

آیا میشه اعداد داخل ms-chart رو به فارسی تبدیل کرد؟

ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۲۲ ۱۰:۱۸

بله. از فونت‌های فارسی تغییر یافته استفاده کنید. برای مثال فونت `irsans` ایی که در پوشه `bin` مثال‌های پروژه هست، از این نوع است. فونت‌های معمولی رو در یک [فونت ادیتور](#) باز می‌کنند و بجای اعداد انگلیسی آن، معادل اعداد فارسی موجود در همان فایل فونت را کپی و پیست می‌کنند. این روش از قدیم برای ساخت گزارشات فارسی کاربرد داشته. ضمناً اگر فونت مورد نظر بر روی سرور نصب نیست، به این صورت هم قابل بارگذاری است:

```
var privateFontCollection = new PrivateFontCollection();
privateFontCollection.AddFontFile(fontPath);
var fontFamily = privateFontCollection.Families[0];
var font = new Font(fontFamily);
```