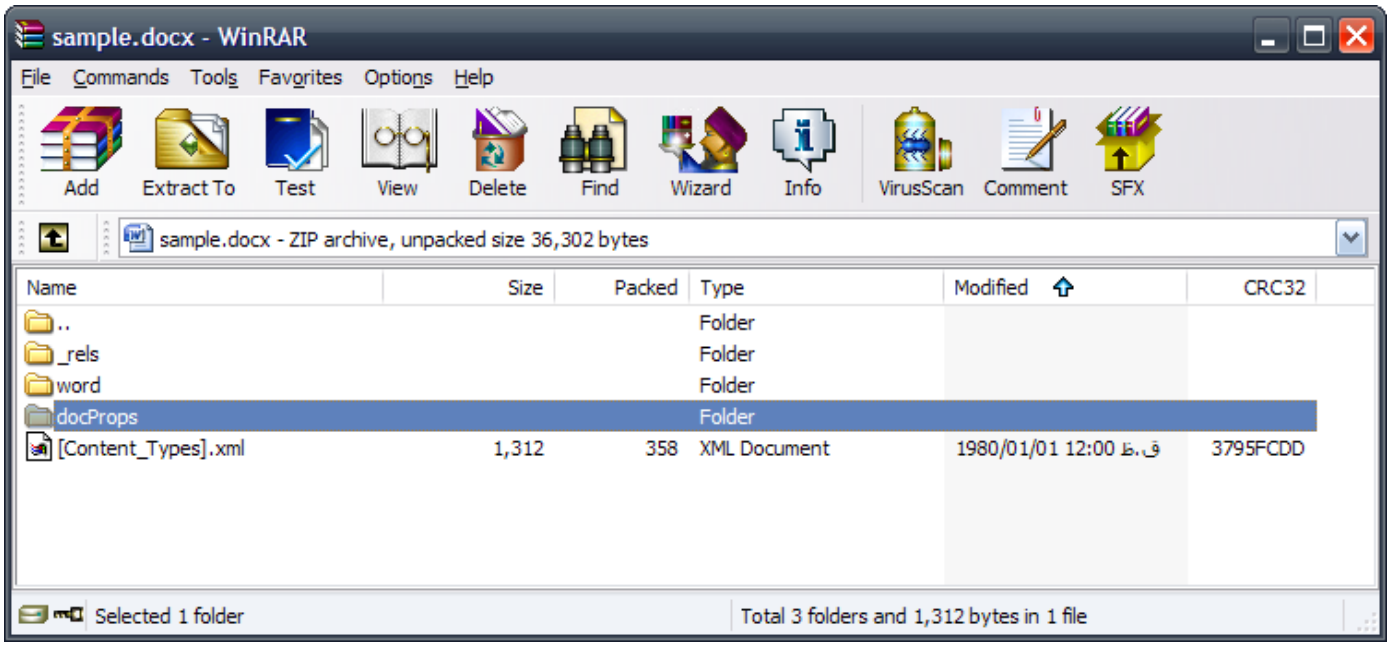
عنوان: **تولید فایل Word بدون نصب MS Word بر روی سرور**
نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۰۹/۱۳ ۱۷:۲۷:۳۴
آدرس: www.dotnettips.info
برچسب‌ها: OpenXML

یکی از مواردی که ممکن است در محیط کاری با آن برخورد داشت، تقاضای تولید فایل word یک گزارش با فرمتی مشخص از یک برنامه ASP.Net است. برای مثال یک قالب درست کرده‌اند که header و footer و کلا یک فرمت رسمی دارد. الان برنامه شما باید این فایل word رسمی را با گزارشی که تولید می‌کند پر کند. حالا اینجاست که گرفتاری برنامه نویس شروع می‌شود! روی سرور باید word نصب باشد تا توسط اشیاء COM آن بتوان یک چنین کارهایی را آن‌هم با ASP.Net که به صورت پیش فرض کمترین سطح دسترسی را روی سیستم دارد انجام داد. یا اینکه باید به سراغ کامپوننت‌های تجاری رفت و حالا اینجا با این وضع تحریم و غیره چگونه بتوان آنها را خریداری کرد یا شاید احتمالا در سایت‌های وارز بتوان نسخه تکه پاره شده آنها را یافت. مشکلی هم که این نوع کامپوننت‌ها دارند این است که ممکن است سال دیگر اصلا ساپورت نشوند. محصولات مایکروسافت هم که مرتبا در حال به روز رسانی هستند. در این حالت برنامه متکی به این نوع کامپوننت‌های تجاری سورس بسته در همان نگارش قبلی خود مجبور است باقی بماند.
خوشبختانه با ارائه آفیس 2007 و فرمت OpenXML فایلهای آن، این مشکل تقریبا مرتفع شده است. مایکروسافت نیز برای سهولت تولید این نوع اسناد، OpenXML SDK را ارائه داده است که از آدرس زیر قابل دریافت است:
Open XML Format SDK 1.0

البته پیش نمایش نگارش دو SDK آن نیز موجود است که در مطلب جاری به آن پرداخته نخواهد شد.

فایل‌های office 2007 از یک فایلzip تشکیل شده از چند فایل xml داخل آن، ایجاد شده‌اند. برای مثال یک فایل docx را با winrar یا امثال آن باز کنید (تصویر زیر):

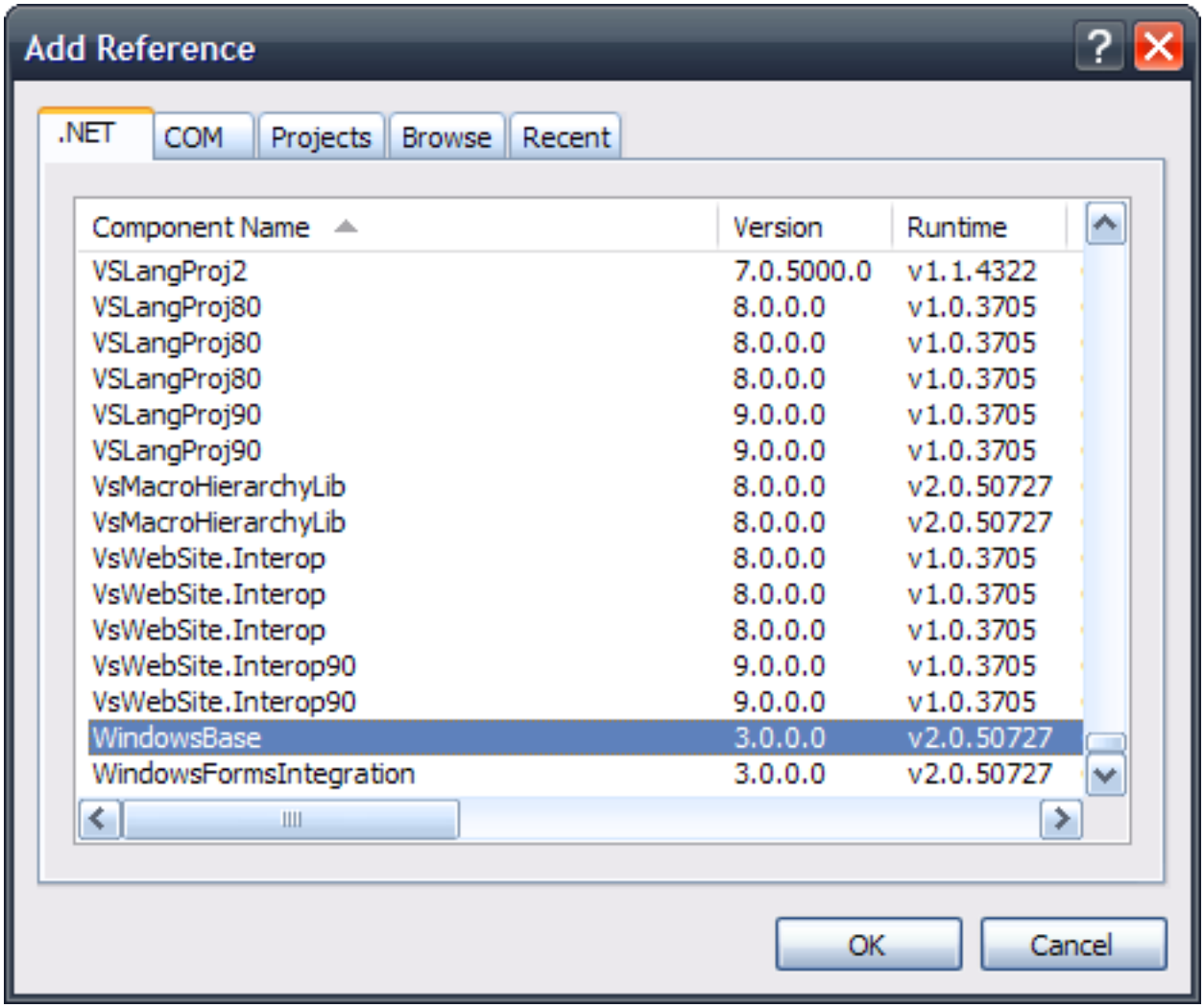


برای کار با اینگونه اسناد باید با اصطلاحات زیر آشنا شد:
Package : فایل zip شما (همان فایل برای مثال docx) اینجا یک بسته نام دارد.

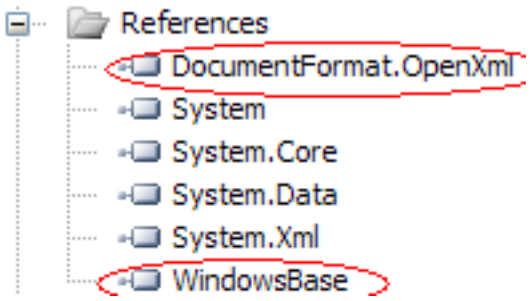
Parts : اجزای این بسته که همان فایل‌های آن هستند، parts نامیده شده اند.
Relations : اگر به فایل‌های موجود در یک بسته دقت کنید، فایلهایی با پسوند rels را خواهید دید که بیانگر نحوه ارتباط Parts با یکدیگر هستند.
Relations Ids: هر ارتباط با یک ID منحصربفرد تعریف می‌گردد.

اگر علاقمند باشید که پوستری را در این رابطه مشاهده نمائید می‌توان به آدرس زیر مراجعه نمود.
Open XML Developer Map

نحوه استفاده از OpenXML SDK در دات نت:
ابتدا باید ارجاعی را به فایل DocumentFormat.OpenXml.dll که پس از نصب در مسیر OpenXMLSDK\1.0.1825\lib قرار گرفته است به پروژه افزود. سپس نیاز است تا ارجاعی به کتابخانه WindowsBase نیز به برنامه افزوده شود (تصویر زیر). افزودن ارجاعی به این کتابخانه جهت کامپایل برنامه ضروری است (شکل زیر).



تا اینجا ارجاعات برنامه به صورت زیر خواهند بود:



یک مثال ساده:

قصد داریم یک فایل docx ساده را با استفاده از OpenXML SDK ایجاد کنیم. در مثال زیر فرمت متغیر docXml را می‌توان با ایجاد یک فایل docx ساده در word و سپس باز کردن بسته فشرده شده آن و مشاهده محتوای فایل word\document.xml بدست آورد.

```
using System.IO;
using System.Text;
using DocumentFormat.OpenXml;
using DocumentFormat.OpenXml.Packaging;

namespace OpenXMLTestApp
{
 class CWord
 {

     public static void CreateDocument(string documentFileName, string text)
     {
         using (WordprocessingDocument wordDoc =
             WordprocessingDocument.Create(documentFileName, WordprocessingDocumentType.Document))
         {
             MainDocumentPart mainPart = wordDoc.AddMainDocumentPart();

             string docXml =
                         @"<?xml version=""1.0"" encoding=""UTF-8"" standalone=""yes""?>
                         <w:document
xmlns:w=""http://schemas.openxmlformats.org/wordprocessingml/2006/main"">
                         <w:body><w:p><w:r><w:t>#REPLACE#</w:t></w:r></w:p></w:body>
                         </w:document>";

             docXml = docXml.Replace("#REPLACE#", text);

             using (Stream stream = mainPart.GetStream())
             {
                 byte[] buf = (new UTF8Encoding()).GetBytes(docXml);
                 stream.Write(buf, 0, buf.Length);
             }
         }
     }
 }
}
```

و نحوه استفاده از آن می‌تواند به صورت زیر باشد:

```
CWord.CreateDocument("test.docx", "سلام دنیا");
```

این کتابخانه کار ایجاد فایل‌های xml ، تولید روابط بین آنها و همچنین بسته بندی و zip کردن نهایی را به صورت خودکار انجام می‌دهد.

برای مطالعه بیشتر می‌توان به منابع زیر مراجعه نمود:

یک ویدیوی آموزشی رایگان از مایکروسافت
دریافت

سؤالات متداول در MSDN
http://msdn.microsoft.com/en-us/library/bb491088.aspx
البته اگر پس از نصب SDK به پوشه doc آن مراجعه نمائید، این سؤال و جواب‌ها را در فایل راهنمای chm آن نیز می‌توان پیدا کرد.

مثال دیگری در مورد ایجاد یک گزارش از بانک اطلاعاتی و گرفتن خروجی docx از آن
http://openxmldeveloper.org/articles/GenerateWordTable.aspx
البته این مثال خیلی قدیمی است و قسمت‌های کار با پکیج را با SDK ارائه شده می‌توان به صورت خودکار انجام داد. اما حداقل نحوه تولید جداول استاندارد OpenXML را می‌توان از آن ایده گرفت.

مثالی در مورد نحوه قرار دادن عکس در فایل docx تولیدی

همچنین مثال‌های بیشتری را در وبلاگ‌های مربوطه می‌توان یافت:
/http://blogs.msdn.com/brian_jones
http://blogs.msdn.com/ericwhite/default.aspx

نظرات خوانندگان

نویسنده: افشار محبی
تاریخ: ۱۳۸۷/۰۹/۲۰ ۰۹:۲۸:۰۰

این نوشته برای من خیلی مفید بود. سپاس.

نویسنده: مهدی پایروند
تاریخ: ۱۳۸۷/۰۹/۲۳ ۱۳:۱۵:۰۰

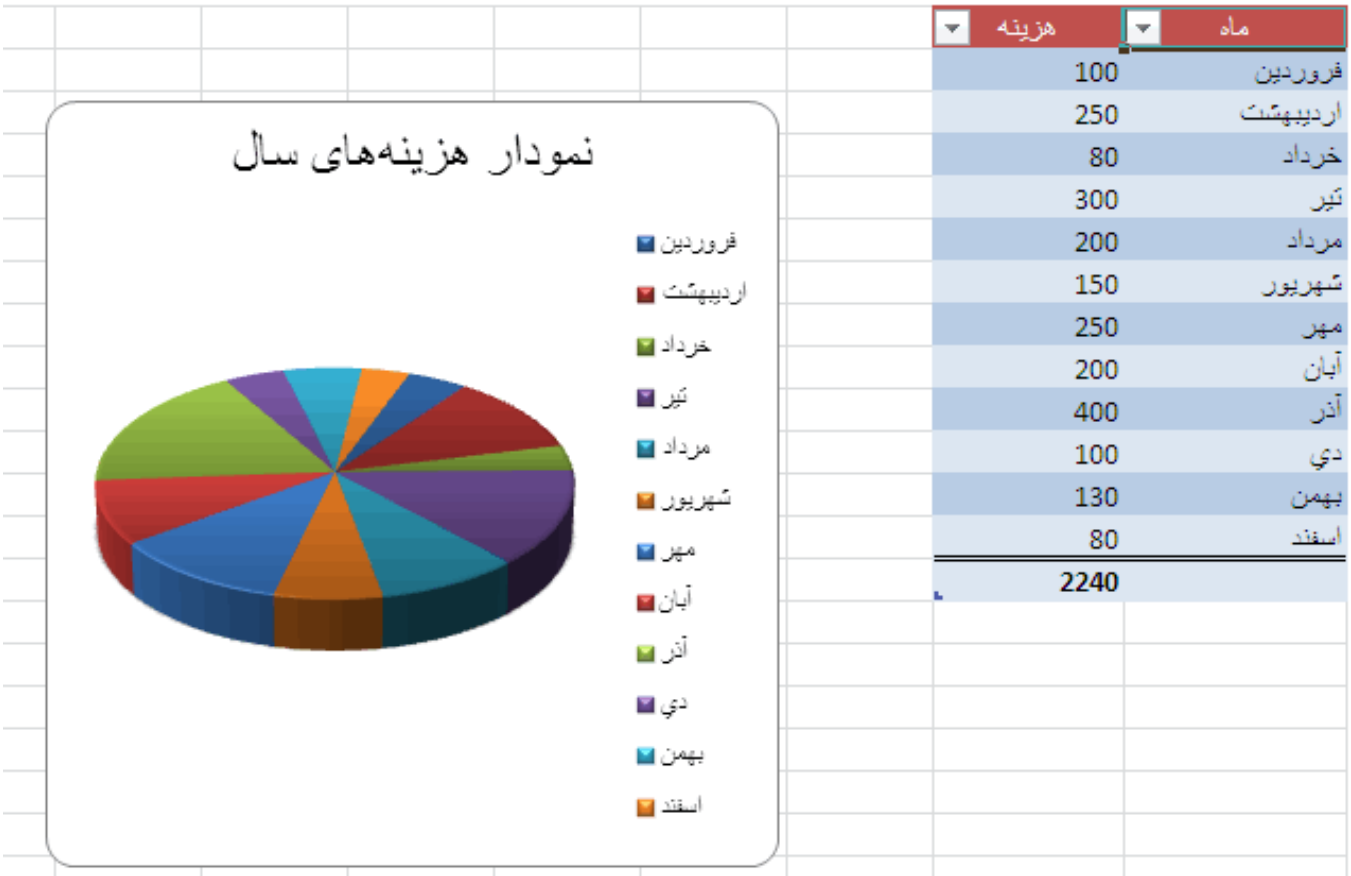
واقعا عالی بود ممنون.
من از طریق سایت مهدی یوسفی با شما آشنا شدم

عنوان: **تولید فایل‌های اکسل حرفه‌ای بدون نیاز به نصب مجموعه‌ی آفیس**
نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۱۲/۰۳ ۱۸:۱۲:۰۰
آدرس: www.dotnettips.info
گروه‌ها: OpenXML

عموما بر روی سرورهای برنامه‌های وب، نرم افزار خاصی نصب نمی‌شود. برای مثال اگر نیاز به تولید فایل اکسل بر روی سرور باشد، سرور دار بعید است که آفیس را برای شما نصب کند و همچنین مایکروسافت هم این یک مورد را اصلا توصیه و پشتیبانی نمی‌کند (ایجاد چندین وهله از برنامه آفیس (تعامل با اشیاء COM) بر روی سرور توسط یک برنامه‌ی وب چند کاربره).
اگر سایت‌ها را هم جستجو کنید پر است از مقالاتی مانند تبدیل GridView به اکسل ... که تنها هنر آن‌ها انتخاب قسمت table مانند GridView و رندر کردن آن در مرورگر با پسوندی به نام xls یا xlsx است. به عبارتی فایل نهایی تولید شده استاندارد نیست. فقط یک html table است با پسوند xls/xlsx که برنامه‌ی اکسل می‌داند به چه صورتی باید آن‌را باز کند (که گاها در این بین فارسی سازی آن مشکل ساز می‌شود). این فایل نهایی تولیدی عاری است از امکانات پیشرفته و حرفه‌ای اکسل. برای مثال اضافه کردن فرمول به آن، تبدیل اطلاعات به نمودارهای اکسل به صورت خودکار، داشتن فایلی با چندین work sheet مختلف، اعمال قالب‌های مختلف، صفحه بندی بهتر و غیره.
مایکروسافت از سال 2007 تولید فایل‌های آفیس را با معرفی استاندارد OpenXML که توسط مؤسسه ایزو هم پذیرفته شده، بسیار ساده‌تر کرده است. OpenXML SDK در دسترس است و توسط آن می‌توان فایل‌های اکسل حرفه‌ای را بدون نیاز به نصب مجموعه‌ی آفیس تولید کرد. کار کردن با OpenXML SDK هم در نگاه اول شاید ساده به نظر برسد اما آن هم ریزه کاری‌های خاص خودش را دارد که نمونه‌ای از آن‌را در مطلب " تولید فایل Word بدون نصب MS Word بر روی سرور " می‌توانید مشاهده کنید. به عبارتی این مجموعه جهت نوشتن کتابخانه‌های ویژه‌ی شما باز است ...
در این بین یکی از حرفه‌ای‌ترین کتابخانه‌هایی که امکانات تولید فایل‌های اکسل را به کمک OpenXML SDK سهولت می‌بخشد، کتابخانه‌ی سورس باز EPPlus است:
/http://epplus.codeplex.com

مثالی در مورد نحوه‌ی استفاده از آن:
می‌خواهیم یک DataTable را به یک فایل اکسل واقعی (نه یک html table با پسوند xlsx) تبدیل کنیم با این شرایط که یکی از قالب‌های جدید آفیس به آن اعمال شود؛ جمع کل یکی از ستون‌ها توسط اکسل محاسبه گردیده و همچنین عرض دقیق ستون‌ها نیز در برنامه تنظیم گردد. نموداری نیز به صورت خودکار این اطلاعات را نمایش دهد:

مثالی از نحوه‌ی استفاده از ایی پی پلاس



ماه	هزینه
فروردین	100
اردیبهشت	250
خرداد	80
تیر	300
مرداد	200
شهریور	150
مهر	250
آبان	200
آذر	400
دي	100
بهمن	130
اسفند	80
	2240

```
using System.Data;
using System.IO;
using OfficeOpenXml;
using OfficeOpenXml.Drawing.Chart;
using OfficeOpenXml.Style;
using OfficeOpenXml.Table;

namespace EPPlusTest
{
  class Program
  {
      static void Main(string[] args)
      {
          var newFile = new FileInfo("Test.xlsx");
          if (newFile.Exists)
          {
              newFile.Delete();
          }

          //ایجاد یک سری اطلاعات دلخواه
          var table = createDt();

          using (var package = new ExcelPackage(newFile))
          {
              // اضافه کردن یک ورک شیت جدید
              ExcelWorksheet worksheet = package.Workbook.Worksheets.Add("مخارج");

              //اضافه کردن یک جدول جدید از دیتاتیبل دریافتی
              worksheet.Cells["A1"].LoadFromDataTable(table, true, TableStyles.Dark9);

              //نمایش جمع ستون هزینه‌های ماه‌ها
              var tbl = worksheet.Tables[0];
              //زیر آخرین ردیف یک سطر اضافه می‌کند
              tbl.ShowTotal = true;
              //فرمول نحوه‌ی محاسبه جمع ستون انتساب داده می‌شود
```

```
                tbl.Columns[1].TotalsRowFunction = RowFunctions.Sum;

                //تعیین عرض ستون‌های جدول
                worksheet.Column(1).Width = 14;
                worksheet.Column(2).Width = 12;

                //تنظیم متن هدر
                worksheet.HeaderFooter.oddHeader.CenteredText = "مثالی از نحوه‌ی استفاده از ایی پی پلاس";

                //می‌خواهیم سرستون‌ها در وسط ستون قرار گیرند
                worksheet.Cells["A1"].Style.HorizontalAlignment = ExcelHorizontalAlignment.Center;
                worksheet.Cells["B1"].Style.HorizontalAlignment = ExcelHorizontalAlignment.Center;

                //افزودن یک نمودار جدید به شیت جاری
                var chart = worksheet.Drawings.AddChart("chart1", eChartType.Pie3D);
                chart.Title.Text = "نمودار هزینه‌های سال";
                chart.SetPosition(Row: 2, RowOffsetPixels: 5, Column: 3, ColumnOffsetPixels: 5);
                chart.SetSize(PixelWidth: 320, PixelHeight: 360);
                chart.Series.Add("B2:B13", "A2:A13");
                chart.Style = eChartStyle.Style26;

                //تنظیم یک سری خواص فایل نهایی
                package.Workbook.Properties.Title = "مثالی از ایی پی پلاس";
                package.Workbook.Properties.Author = "وحید";
                package.Workbook.Properties.Subject = "ایجاد فایل اکسل بدون نرم افزار اکسل";

                //تنظیم نحوه‌ی نمایش فایل زمانیکه در نرم افزار اکسل گشوده می‌شود
                worksheet.View.PageLayoutView = true;
                worksheet.View.RightToLeft = true;

                // ذخیر سازی کلیه موارد اعمالی در فایل
                package.Save();
            }
        }

        private static DataTable createDt()
        {
            var table = new DataTable("مخارج");
            table.Columns.Add("ماه", typeof(string));
            table.Columns.Add("هزینه", typeof(decimal));

            table.Rows.Add("100 ,"فروردین");
            table.Rows.Add("250 ,"اردیبهشت");
            table.Rows.Add("80 ,"خرداد");
            table.Rows.Add("300 ,"تیر");
            table.Rows.Add("200 ,"مردآد");
            table.Rows.Add("150 ,"شهریور");
            table.Rows.Add("250 ,"مهر");
            table.Rows.Add("200 ,"آبان");
            table.Rows.Add("400 ,"آذر");
            table.Rows.Add("100 ,"دی");
            table.Rows.Add("130 ,"بهمن");
            table.Rows.Add("80 ,"اسفند");
            return table;
        }
    }
}
```

نظرات خوانندگان

نویسنده: سعید بابائی
تاریخ: ۱۳۹۲/۰۹/۱۶ ۱۲:۵۰

سلام آقای نصیری ، من نیاز دارم از بانک اطلاعاتی گزارش اکسل بگیرم . میخوام از EPPlus استفاده کنم اما به دلایلی روی سرور x64 نمیخوام از dllهای x86 استفاده کنم . برای همین چون EP 32 بیتی هست نمیشه اینکار رو بکنم . من سورس EP رو 64 بیلد کردم و خودش اوکی هست اما dll مربوط به WindowsBase هم 32 هست. چیکار کنیم ؟ در نظر داشته باشید در نهایت 64 بیتی بودن فوق العاده مهمه . ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۱۶ ۱۳:۴

- فایل‌های استاندارد آن روی Any CPU کامپایل شده‌اند. ادامه در اینجا
+ وابستگی WindowsBase هم یک فایل استاندارد مجموعه دات نت است و نه خارج از آن و در WPF از آن استفاده می‌شود.
بنابراین مشکلی ندارد (چون حتی اگر یک dll بومی ویندوز هم باشد، با توجه به اینکه عملا دو نگارش X86 و X64 دات نت وجود دارند، بسته به سیستم مورد استفاده، یکی از آن‌ها به صورت خودکار در ابتدای کار نصب دات نت فریم ورک، نصب خواهد شد).

نویسنده: مریم بانو
تاریخ: ۱۳۹۲/۱۱/۲۰ ۱۵:۱۳

سلام
من دارم از epplus استفاده می‌کنم. به range هایی که تعریف می‌کنم فونت "B Nazanin" میدم اما اعمال نمیشه، هر چند با انتخاب یک سلول فونت رو درست نشون میده. جالب اینکه وقتی یک فایل اکسل با فونت نازنین و درست باز دارم اگرهمزمان فایل اکسل گزارش رو باز کنم، فونت رو اعمال می‌کنه. ممنون میشم راهنماییم بفرمایید که چطور مشکل رو حل کنم. با سپاس.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۲۰ ۱۸:۳

در مثال مطلب فوق، تنظیم زیر، فونت ستون ماه را تغییر می‌دهد:

```
worksheet.Column(1).Style.Font.Name = "B Nazanin";
worksheet.Column(1).Style.Font.Size = 15;
```

ماه	هزینه
فروردین	100
اردیبهشت	250
خرداد	80
تیر	300
مرداد	200
شهریور	150
مهر	250
آبان	200
آذر	400
دی	100
بهمن	130
اسفند	80
	2240