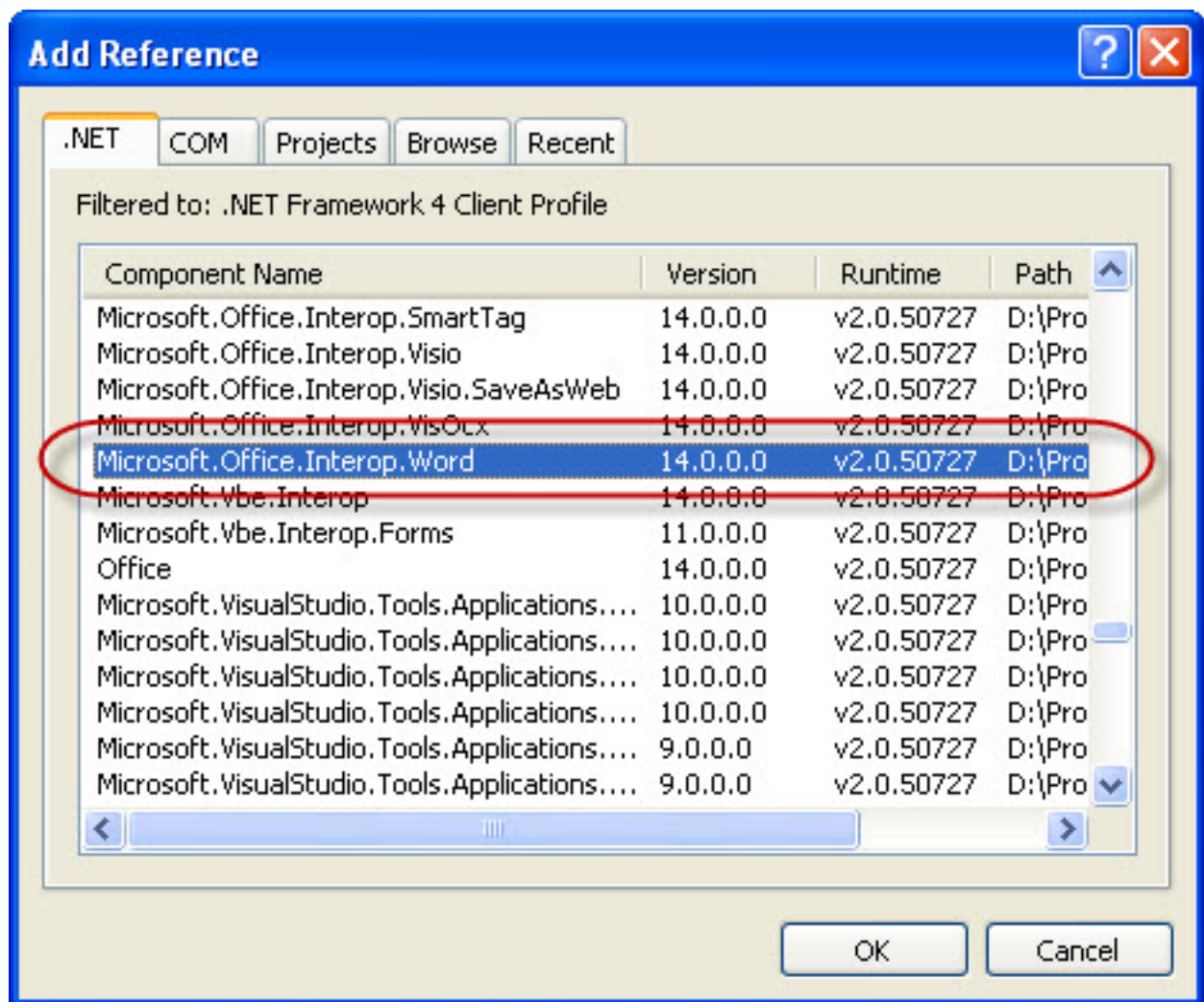


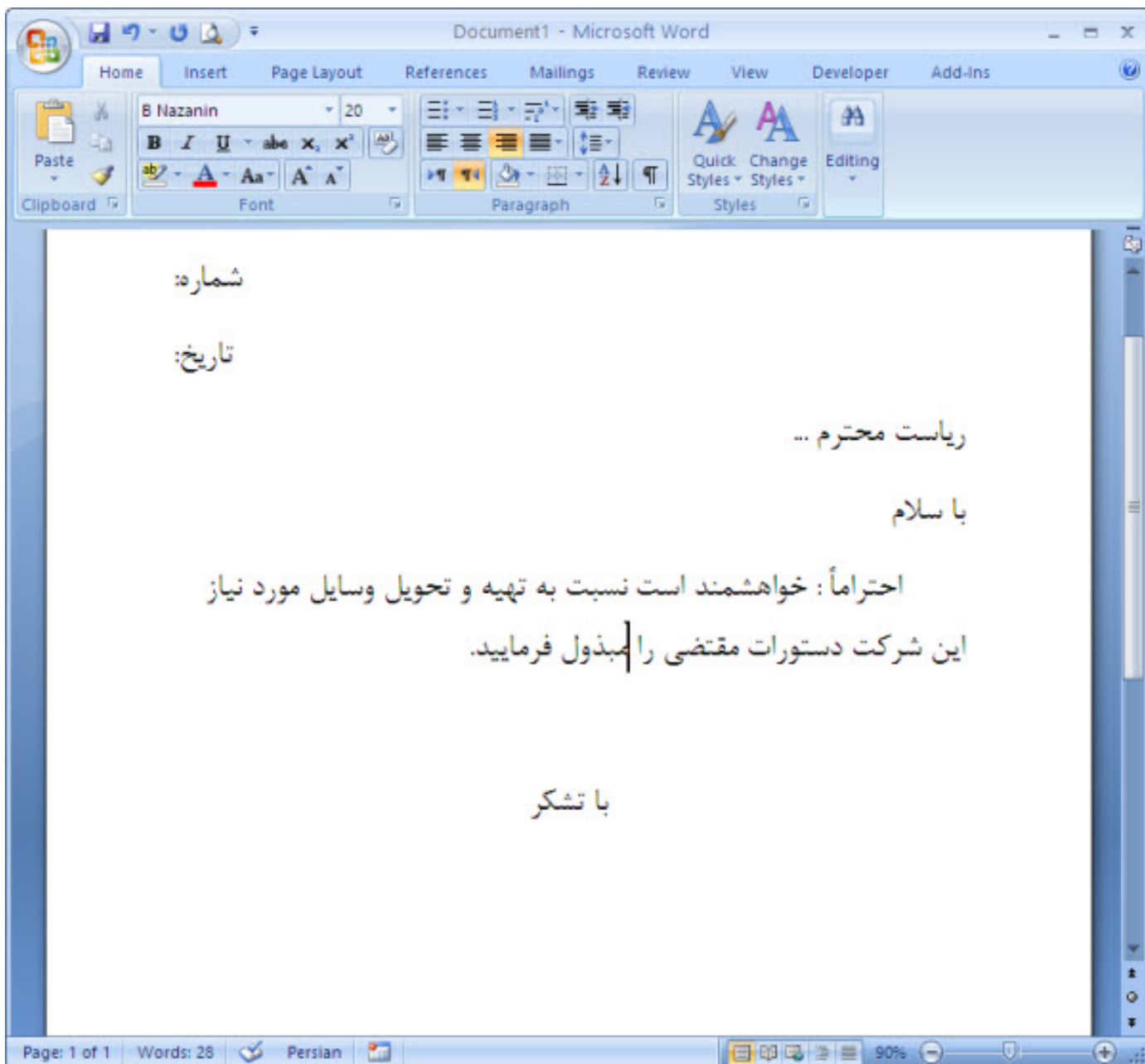
می‌خواهیم با استفاده از چاپگرهای مجازی نصب شده روی سیستم خود یک فایل Doc را به یک فایل TIF تبدیل یا به عبارت دیگر داخل فایل تصویری چاپ کنیم. یک نمونه عملی، استفاده در سیستم دبیرخانه می‌تواند باشد. کاربر متن نامه را داخل محیط Word تایپ کرده و هنگامی که نامه را ارسال می‌کند، گردش کار با این فایل تصویری انجام خواهد شد.

پس از ایجاد پروژه جدید، یک ارجاع از Microsoft.Office.Interop.Word به پروژه خود اضافه می‌کنیم



در هنگام نصب مجموعه Office می‌توانید یک چاپگر مجازی بنام Microsoft Office Document Image Writer را نصب کنید. می‌خواهیم از آن برای رسیدن به هدفمان استفاده کنیم.

حال یک فایل Word آماده می‌کنیم.



با استفاده از کدهای زیر می‌توان فایل ذخیره شده را به یک فایل تصویری با قالب TIF تبدیل کنیم

```
private void btnPrint_Click(object sender, EventArgs e)
{
    try
    {
        object varFileName = @"d:\test.doc";
        object varFalseValue = false;
        object varTrueValue = true;
        object varMissing = Type.Missing;
        Microsoft.Office.Interop.Word.Application varWord = new
        Microsoft.Office.Interop.Word.Application();

        varWord.ActivePrinter = "Microsoft Office Document Image Writer";
        // varWord.ActivePrinter = "Snagit 11";
        // varWord.ActivePrinter = "priPrinter";
        // varWord.ActivePrinter = "doPDF v7";

        Microsoft.Office.Interop.Word.Document varDoc =
        varWord.Documents.Open(ref varFileName, ref varMissing, ref varFalseValue, ref
```

```
varMissing,
                                ref varMissing, ref varMissing, ref varMissing,
                                ref varMissing, ref varMissing, ref varMissing,
                                ref varMissing, ref varMissing, ref varMissing);
    varDoc.Activate();

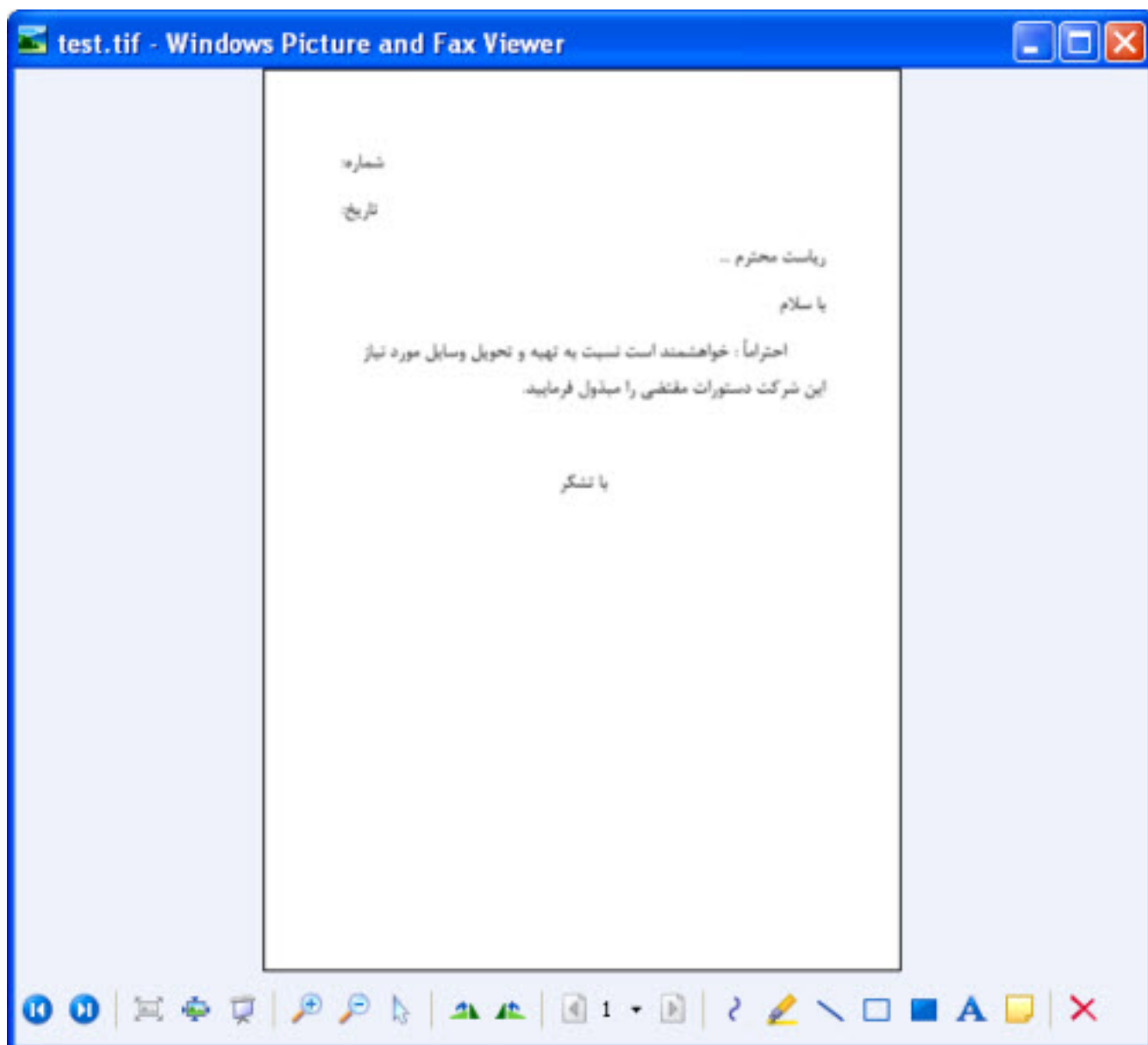
    object PrintToFile = true;
    object OutputFileName = @"d:\test.tif";
    varDoc.PrintOut(ref varMissing, ref varFalseValue, ref varMissing,
                    ref OutputFileName, ref varMissing, ref varMissing, ref varMissing,
                    ref varMissing, ref varMissing, ref varMissing, ref PrintToFile,
                    ref varMissing, ref varMissing, ref varMissing, ref varMissing,
                    ref varMissing, ref varMissing, ref varMissing);

    varDoc.Close(ref varMissing, ref varMissing, ref varMissing);
    varWord.Quit(ref varMissing, ref varMissing, ref varMissing);
}
catch (Exception varE)
{
    MessageBox.Show("Error:\n" + varE.Message, "Error message");
}
}
```

شما می‌توانید متناسب با نیاز خود از سایر چاپگرهای مجازی نصب شده استفاده نمایید.

```
varWord.ActivePrinter = "Snagit 11";
varWord.ActivePrinter = "priPrinter";
varWord.ActivePrinter = "doPDF v7";
```

نتیجه کار در قالب یک فایل TIF



[PRINTDOC.ZIP](#)

نظرات خوانندگان

نویسنده: سیروان عقیفی
تاریخ: ۱۰:۲۷ ۱۳۹۲/۱۱/۱۹

ممنون،

فقط یک نکته چاپگر Microsoft Office Document Image Writer از آفیس 2010 به بعد حذف شده، آیا برای این مورد راه حلی سراغ دارید؟
البته در آفیس 2007 به صورت پیش فرض نصب نیست و باید [این صورت](#) نصبش کنیم.

نویسنده: وحید نصیری
تاریخ: ۱۱:۴۹ ۱۳۹۲/۱۱/۱۹

- از [fax driver](#) استاندارد ویندوز می شود استفاده کرد.
+ آفیس های جدید امکان ذخیره سازی متن را [به صورت PDF](#) ، پشتیبانی می کنند.

نویسنده: شاهین
تاریخ: ۲۰:۱۹ ۱۳۹۲/۱۱/۳۰

مرسی خیلی خوب و مفید بود با آفیس 2013 اگه بخواهیم سند پی دی اف تولید کنیم چطور میشه از این برنامه استفاده کرد ؟
روشنش خیلی گنده حالا نمیدونم مشکل از منه یا سیستم ، روش دیگه ای هم هست ؟

نویسنده: رضایات
تاریخ: ۱۹:۵۸ ۱۳۹۲/۱۲/۰۱

شما بسته به نیاز خود می توانید هر چاپگری با هر قابلیتی را نصب و استفاده کنید. در این روش برای چاپ مطالب از چاپگر مجازی استفاده می شود. در خط 13-15 کد فوق از سه چاپگر مجازی دیگر استفاده شده که [doPDF](#) خروجی pdf تولید می کند. شاید سایت [colorpilot](#) و یا [gemboxsoftware](#) مفید واقع بشه.

یکی از متداول‌ترین کارهایی که با اسناد می‌توان انجام داد، تهیه خروجی pdf از word و پر کردن یک فایل word با مقادیر ورودی است که سعی داریم یک نمونه از آن را اینجا بررسی کنیم. کد عمومی برای جایگزین کردن:

```
public void MsInteropReplace(Microsoft.Office.Interop.Word.Application doc, object findText, object
replaceWithText)
{
    object matchCase = false;
    object matchWholeWord = true;
    object matchWildCards = false;
    object matchSoundsLike = false;
    object matchAllWordForms = false;
    object forward = true;
    object format = false;
    object matchKashida = false;
    object matchDiacritics = false;
    object matchAlefHamza = false;
    object matchControl = false;
    object read_only = false;
    object visible = true;
    object replace = 2;
    object wrap = 1;
    //execute find and replace
    doc.Selection.Find.Execute(ref findText, ref matchCase, ref matchWholeWord,
        ref matchWildCards, ref matchSoundsLike, ref matchAllWordForms, ref forward, ref wrap,
        ref format, ref replaceWithText, ref replace,
        ref matchKashida, ref matchDiacritics, ref matchAlefHamza, ref matchControl);
}
```

و یا این مورد:

```
private static void MsInteropReplace2()
{
    var doc = new
Microsoft.Office.Interop.Word.Application().Documents.Open(@"D:\temp\te1.docx");
    doc.Content.Find.Execute("@levelOrder", false, true, false, false, false, true, 1, false,
"12345", 2,
    false, false, false, false);
    object missing = System.Reflection.Missing.Value;
    doc.SaveAs(@"D:\temp\out.docx", ref missing, ref missing, ref missing, ref missing
        , ref missing, ref missing, ref missing, ref missing, ref missing, ref missing
        , ref missing, ref missing, ref missing, ref missing, ref missing);
}
```

که هر دو مورد را در stackoverflow می‌توانید پیدا کنید. به شخصه از این مورد برای replace کردن مقادیر در یک فایل template.docx استفاده می‌کردم؛ ولی بعد از مدتی فهمیدم که Header و Footer را نمیتواند پردازش کند. کد زیر در تمامی قسمت‌هایی که در یک فایل word می‌توان متغیر تعریف کرد را گشته و عمل پر کردن مقادیر را بر روی فایل نمونه، انجام می‌دهد و شامل سه متد ذیل است:

```
private static void repAll()
{
    object Missing = System.Reflection.Missing.Value;
    Application app = null;
    Microsoft.Office.Interop.Word.Document doc = null;
    try
    {
        app = new Microsoft.Office.Interop.Word.Application();
        doc = app.Documents.Open(@"D:\temp\te1.docx", Missing, Missing, Missing, Missing,
Missing, Missing, Missing, Missing, Missing);
        FindReplaceAnywhere(app, "@levelOrder", "محرم‌انه");
    }
}
```

```
        doc.SaveAs(@"D:\temp\out.docx", Missing, Missing, Missing, Missing, Missing, Missing,
Missing, Missing, Missing);
    }
    finally
    {
        try
        {
            if (doc != null) ((Microsoft.Office.Interop.Word._Document)doc).Close(true,
Missing, Missing);
        }
        finally { }
        if (app != null) ((Microsoft.Office.Interop.Word._Application)app).Quit(true, Missing,
Missing);
    }
}

private static void searchAndReplaceInStory(Microsoft.Office.Interop.Word.Range rngStory,
string strSearch, string strReplace)
{
    rngStory.Find.ClearFormatting();
    rngStory.Find.Replacement.ClearFormatting();
    rngStory.Find.Text = strSearch;
    rngStory.Find.Replacement.Text = strReplace;
    rngStory.Find.Wrap = WdFindWrap.wdFindContinue;
    object Missing = System.Reflection.Missing.Value;

    object arg1 = Missing; // Find Pattern
    object arg2 = Missing; //MatchCase
    object arg3 = Missing; //MatchWholeWord
    object arg4 = Missing; //MatchWildcards
    object arg5 = Missing; //MatchSoundsLike
    object arg6 = Missing; //MatchAllWordForms
    object arg7 = Missing; //Forward
    object arg8 = Missing; //Wrap
    object arg9 = Missing; //Format
    object arg10 = Missing; //ReplaceWith
    object arg11 = WdReplace.wdReplaceAll; //Replace
    object arg12 = Missing; //MatchKashida
    object arg13 = Missing; //MatchDiacritics
    object arg14 = Missing; //MatchAlefHamza
    object arg15 = Missing; //MatchControl

    rngStory.Find.Execute(ref arg1, ref arg2, ref arg3, ref arg4, ref arg5, ref arg6, ref arg7,
ref arg8, ref arg9, ref arg10, ref arg11, ref arg12, ref arg13, ref arg14, ref arg15);
}

// Main routine to find text and replace it,
// var app = new Microsoft.Office.Interop.Word.Application();
public static void FindReplaceAnywhere(Microsoft.Office.Interop.Word.Application app, string
findText, string replaceText)
{
    // http://forums.asp.net/p/1501791/3739871.aspx
    var doc = app.ActiveDocument;

    // Fix the skipped blank Header/Footer problem
    // http://msdn.microsoft.com/en-us/library/aa211923(office.11).aspx
    Microsoft.Office.Interop.Word.WdStoryType lngJunk =
doc.Sections[1].Headers[WdHeaderFooterIndex.wdHeaderFooterPrimary].Range.StoryType;

    // Iterate through all story types in the current document
    foreach (Microsoft.Office.Interop.Word.Range rngStory in doc.StoryRanges)
    {
        // Iterate through all linked stories
        var internalRangeStory = rngStory;

        do
        {
            searchAndReplaceInStory(internalRangeStory, findText, replaceText);

            try
            {
                // 6 , 7 , 8 , 9 , 10 , 11 -- http://msdn.microsoft.com/en-
us/library/aa211923(office.11).aspx
                switch (internalRangeStory.StoryType)
                {
                    case Microsoft.Office.Interop.Word.WdStoryType.wdEvenPagesHeaderStory: // 6
                    case Microsoft.Office.Interop.Word.WdStoryType.wdPrimaryHeaderStory: // 7
                    case Microsoft.Office.Interop.Word.WdStoryType.wdEvenPagesFooterStory: // 8
                    case Microsoft.Office.Interop.Word.WdStoryType.wdPrimaryFooterStory: // 9
                    case Microsoft.Office.Interop.Word.WdStoryType.wdFirstPageHeaderStory: //
```

```
10         case Microsoft.Office.Interop.Word.WdStoryType.wdFirstPageFooterStory: //
11
            if (internalRangeStory.ShapeRange.Count > 0)
            {
                foreach (Microsoft.Office.Interop.Word.Shape oShp in
internalRangeStory.ShapeRange)
                {
                    if (oShp.TextFrame.HasText != 0)
                    {
                        searchAndReplaceInStory(oShp.TextFrame.TextRange, findText,
replaceText);
                    }
                }
            }
            break;
        default:
            break;
    }
}
catch
{
    // On Error Resume Next
}

// ON ERROR GOTO 0 -- http://www.harding.edu/fmccown/vbnet_csharp_comparison.html

// Get next linked story (if any)
internalRangeStory = internalRangeStory.NextStoryRange;
} while (internalRangeStory != null); //
http://www.harding.edu/fmccown/vbnet_csharp_comparison.html
}
}
```

برای تهیه pdf نیز می‌توانید به کد زیر مراجعه کنید:

```
public static void getFileDocxInPdf()
{
    // Create a new Microsoft Word application object
    Microsoft.Office.Interop.Word.Application word = new
Microsoft.Office.Interop.Word.Application();

    // C# doesn't have optional arguments so we'll need a dummy value
    object oMissing = System.Reflection.Missing.Value;

    // Get list of Word files in specified directory
    DirectoryInfo dirInfo = new DirectoryInfo(@"D:\temp");
    FileInfo[] wordFiles = dirInfo.GetFiles("*.docx");

    word.Visible = false;
    word.ScreenUpdating = false;

    foreach (FileInfo wordFile in wordFiles)
    {
        // Cast as Object for word Open method
        Object filename = (Object)wordFile.FullName;

        // Use the dummy value as a placeholder for optional arguments
        Microsoft.Office.Interop.Word.Document doc = word.Documents.Open(ref filename, ref
oMissing,
        ref oMissing, ref oMissing, ref oMissing, ref oMissing,
        ref oMissing, ref oMissing, ref oMissing, ref oMissing, ref oMissing,
        ref oMissing, ref oMissing, ref oMissing, ref oMissing);
        doc.Activate();

        object outputFileName = wordFile.FullName.Replace(".docx", ".pdf");
        object fileFormat = WdSaveFormat.wdFormatPDF;

        // Save document into PDF Format
        doc.SaveAs(ref outputFileName,
            ref fileFormat, ref oMissing, ref oMissing,
            ref oMissing, ref oMissing, ref oMissing, ref oMissing,
            ref oMissing, ref oMissing, ref oMissing, ref oMissing,
            ref oMissing, ref oMissing, ref oMissing, ref oMissing);
    }
}
```



```
// Close the Word document, but leave the Word application open.
// doc has to be cast to type _Document so that it will find the
// correct Close method.
object saveChanges = WdSaveOptions.wdDoNotSaveChanges;
((_Document)doc).Close(ref saveChanges, ref oMissing, ref oMissing);
doc = null;
}

// word has to be cast to type _Application so that it will find
// the correct Quit method.
((_Application)word).Quit(ref oMissing, ref oMissing, ref oMissing);
word = null;
}
```

نظرات خوانندگان

نویسنده: امیر نوروزیان
تاریخ: ۸:۳۸ ۱۳۹۳/۱۰/۲۱

سلام؛ من از همین روش شما استفاده کردم چند وقت پیش به وسیله bookmark:

```
private Document oDoc;
public void createdoc1()
{
    var realpath="~/template";
    var filePath = Path.Combine(HttpContext.Current.Server.MapPath("~/template"),
Lcourseid.Text + ".doc");
    var oWordApplication = new Application();
    DirectoryInfo dir = new DirectoryInfo(Server.MapPath(realpath));
    foreach (FileInfo files in dir.GetFiles())
    {
        files.Delete();
    }
    // To invoke MyMethod with the default argument value, pass
    // Missing.Value for the optional parameter.
    object missing = System.Reflection.Missing.Value;

    //object fileName = ConfigurationManager.AppSettings["DocxPath"];@"C:\DocXExample.docx";
    string fileName = @"D:\template1.dot";
    //string fileName1 = @"D:\sss.doc";
    object newTemplate = false;

    object docType = 0;
    object isVisible = true;

    //System.Reflection.Missing.Value is used here for telling that method to use default
parameter values when method execution
    oDoc = oWordApplication.Documents.Open(fileName, newTemplate, docType, isVisible, ref
missing, ref missing, ref missing, ref missing, ref missing, ref missing,
ref missing, ref missing, ref missing, ref missing, ref missing, ref missing);
    // usable in earlier versions of Microsoft Word v2003 v11
    // if(Convert.ToInt16(oWordApplication.Version) >=11)
    {
        //Sets or returns a Boolean that represents whether a document is being viewed in reading
layout view.
        oDoc.ActiveWindow.View.ReadingLayout = false;
    }

    //The active window is the window that currently has focus.If there are no windows open, an
exception is thrown.
    //microsoft.office.tools.word.
    oDoc.Activate();

    if (oDoc.Bookmarks.Exists("Title"))
    {
        oDoc.Bookmarks["Title"].Range.Text = "Test Field Entry from webform";
        oDoc.Bookmarks["Address"].Range.Text = "Address Field Entry from webform";
    }

    oDoc.SaveAs(filePath, ref missing);
    oWordApplication.Documents.Close(ref missing, ref missing, ref missing);
    //oWordApplication.Quit(ref SaveChanges, ref missing, ref missing, ref missing);
    ProcessRequest(filePath, Lcourseid.Text);
}
```

ولی این روش مشکلاتی هم داره. اول اینکه باید روی سرور تنظیمات خاصی رو انجام بدی. البته از تنظیمات منظور تنظیمات دسترسی کاربران هست. ولی استفاده از داک ایکس بیشتر استقبال میشه چون دردسرش کمتره.

نویسنده: امیر عزیزخانی
تاریخ: ۲۰:۶ ۱۳۹۳/۱۰/۲۱

البته Docx ابزار خوبی ولی توی یک مثالی که خواستم ازش استفاده کنم به EXCEPTION خورد و علتش تا اونجایی که من متوجه شدم گیر دادن به محتویات اضافه XML داخل فایل بود