

لاحظت أن هناك صعوبة نوعا ما في التعامل مع أوفيس وتكاملها مع دوت نت ولقد عانيت من هذه المشكلة ولهذا وبعد البحث في المنتدى قمت بجمع ما استطعت جمعه عن هذا الموضوع والذي يعطي أساساً جيداً في التعامل معها .

وللأمانة قمت بنسخ المقالات كما هي كما كتبت اسم كاتب المقالة في نهايتها وأتمنى من أصحاب هذه المقالات السماح لي بهذا العمل ولكل منهم حق العتاب لذلك وضعت بريدي الالكتروني في نهاية هذا الكتاب ليتسنى لأي شخص مراجعتي بأي شيء يريده ، والهدف من عملي لهذا الكتيب البسيط (والذي هو في حقيقة مجموعة من المقالات أكثر منها كتابا) الهدف الوحيد منه هو التسهيل على كل باحث في هذا الموضوع ان يجد ضالته بأقل جهد ممكن مع التركيز على الحقوق الفكرية لأصحابها والذين نتقدم لهم بكل الشكر والتقدير على جهودهم الطيبة في إثراء معلوماتنا

وحفاظاً على الحقوقية الفكرية فقد قمت بتلوين كتابتي بلون مميز (هو لون هذا الخط) ليتسنى للقارئ تمييز كلامي أنا من كلام كتاب المقالات وذلك للأمانة العلمية والفكرية أيضاً

أتمنى أن يلقي عملي قبولاً لديكم ولا تحرموني من ثمنه ألا وهو الدعاء فقط لا غير

هيثم عرفات

**** نبدأ مع اكسل**

الموضوع الأول :

التكامل مع برامج مايكروسوفت Microsoft Excel التكامل مع برنامج-1

السلام عليكم ورحمة الله وبركاته وبعد

طبعا كتبت موضوعا يتكلم عن التكامل مع الاكسل وكان موضوعا بسيطا جدا بعض الاعضاء منه لعدم الشرح بصورة مرنة وقلة الصورة وعدم وجود وتذمر قررت ان اعيدته لكم بصورة جديدة تطبيقات لذا طبعا بدء من الاصدار 2005 وما فوق اصبح هناك برنامج داخلي تبع الفيچوال وهو موضوع جديد Visual Studio Tools for Office أي VSTO ستوديو وهو أتكلم عنه لاحقا ربما

نيجي لموضوعنا
Excel ؟؟؟؟ ولعل التساؤل يطرح نفسه :مالهدف من التكامل مع
أستطيع أن أخص أبرز الأسباب التي تجعلنا نتكامل معه في عدة نقاط
إنشاء تقارير باحترافية عالية-1
سهولة الطباعة باستخدام الإكسل-2
قوية ومدعومة (Forms) إمكانية إنشاء نماذج-3
وغيرها من الأسباب التي ستتضح لاحقا من خلال هذا الموضوع بإذن الله

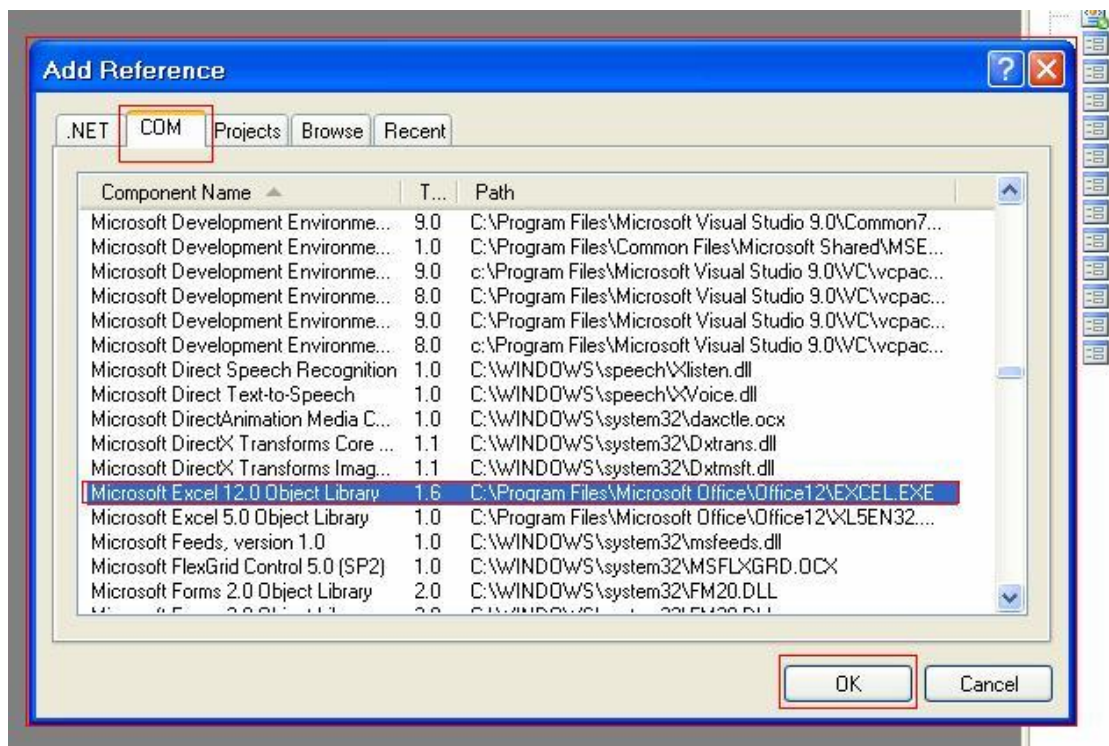
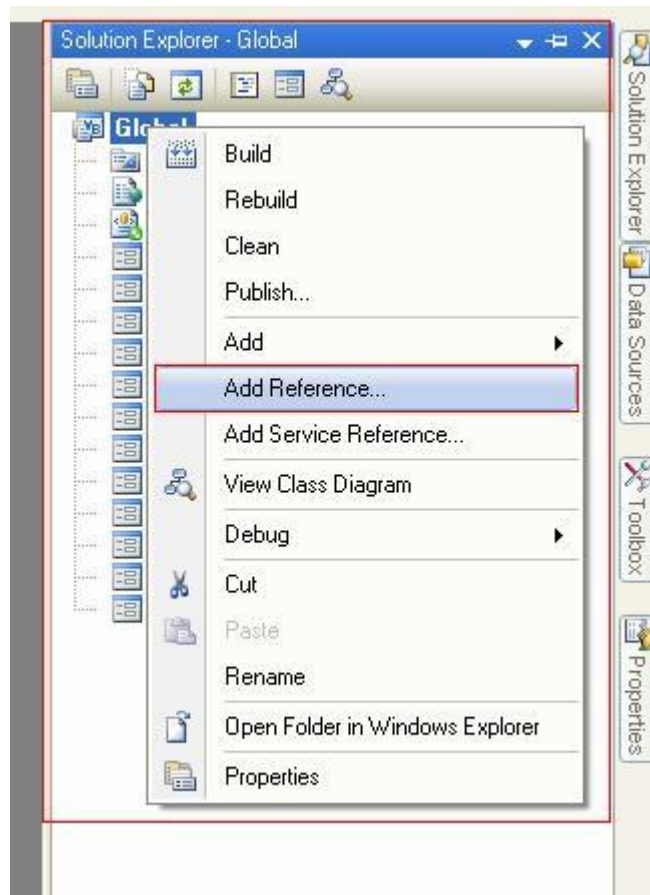
سنقوم في هذا البرنامج بإنشاء فورم لإدخال البيانات الشخصية ثم طباعتها
وهذه صورة لواجهة البرنامج ونسخة من البرنامج موجودة في المرفقات

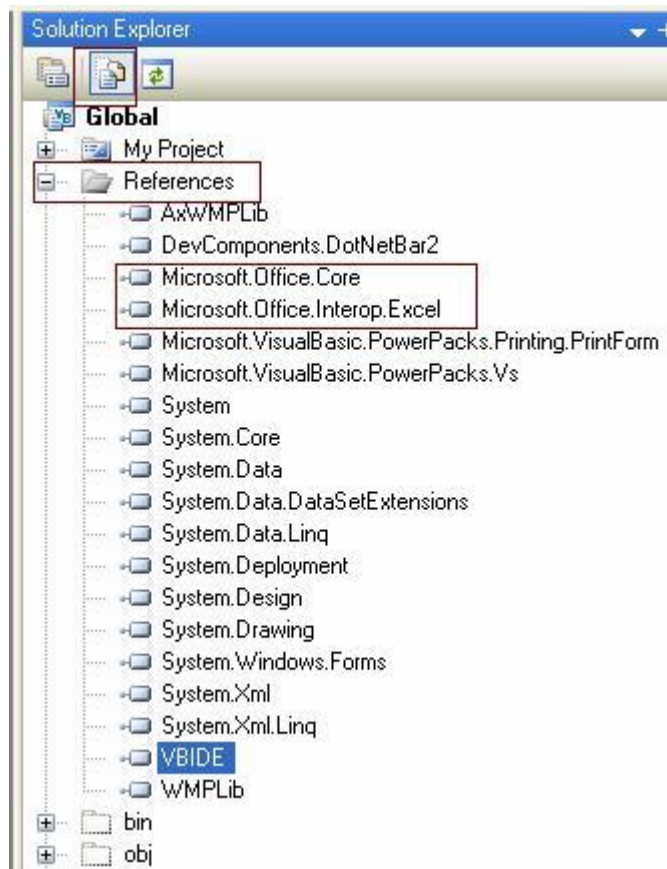
بداية علينا استيراد 3 ملفات هي على التوالي
اقتباس:

Interop.Excel.dll
Interop.Microsoft.Office.Core.dll
Interop.VBIDE.dll

وهي مرفقة في هذا الموضوع

طريقة الاستيراد ****





الهدف من استيرادنا لهذه الملفات الثلاث أنها هي الوصلة التي تربط بين برنامجنا وبين برنامج اكسيل حتى تتيح لنا التصميم داخل الاكسيل

وبعد ادراجهم نقوم باستيراده في اعلى محرر الاكواد للفورم الذي نريد ارتباطه بالاكسيل كالتالي

اقتباس:

```
Imports Excel
Imports Excel.XLLineStyle
Imports Excel.XLBordersIndex
Imports Excel.Constants
Imports Excel.XLBorderWeight
Imports Excel.XLSortOrder
Imports Excel.XIYesNoGuess
Imports Excel.XISortDataOption
```

إلى الآن أكملنا الخطوات الأساسية والتي غالباً ستفعل للمرة واحدة فقط نأتي الآن إلى الخطوة الأهم وهي التكامل مع الإكسل التعامل مع الأكسل يكون في 3 خطوات **فتح البرنامج وإنشاء ورقة جديدة أو فتح صفحة موجودة**

**(نقل البيانات من البرنامج إلى الإكسل (ثم إلى البرنامج إذا لزم الأمر)
(حفظ المستند (إختياري)
طباعة المستند اختياري**

فتح البرنامج و فتح صفحة جديدة أو موجودة-

ويتم ذلك من خلال انشاء 3 متغيرات
الأول يخص برنامج الإكسل ككل
الثاني يخص ملف الإكسل
Sheet الثالث يخص ورقة البيانات أو

رمز:

```
Dim ReportApp As New Excel.Application  
Dim ReportWB As Workbook  
Dim ReportWS As Worksheet
```

ومن ثم إسناد قيمة كل من الثاني للثالث والثالث للثاني

رمز:

```
ReportWB  
=ReportApp.Workbooks.Add(XIWBATemplate.xlWBATWorksheet)
```

لانشاء ملف اكسل جديد'

رمز:

```
ReportWS = ReportWB.Sheets.Add(, ReportWB.Worksheets(1))
```

**لفتح صفحة جديدة في ملف الإكسل'
الآن الملف أصبح موجود وإن لم تصدق فجرب إضافة هذا السطر بعد الاسطر
السابقة**

رمز:

```
ReportApp.Visible = True
```

**أكسل جديد بورقة واحدة (ولكن نصيحتي ألا فستلاحظ أنه قد تم فتح ملف
هذا الوقت على الاقل وستعرف لماذا تستخدم هذا السطر في برامجك في
لاحقا)**

**سؤال: كيف نفتح ملف محفوظ من الأساس
الإجابة بتغيير السطرين إلى**

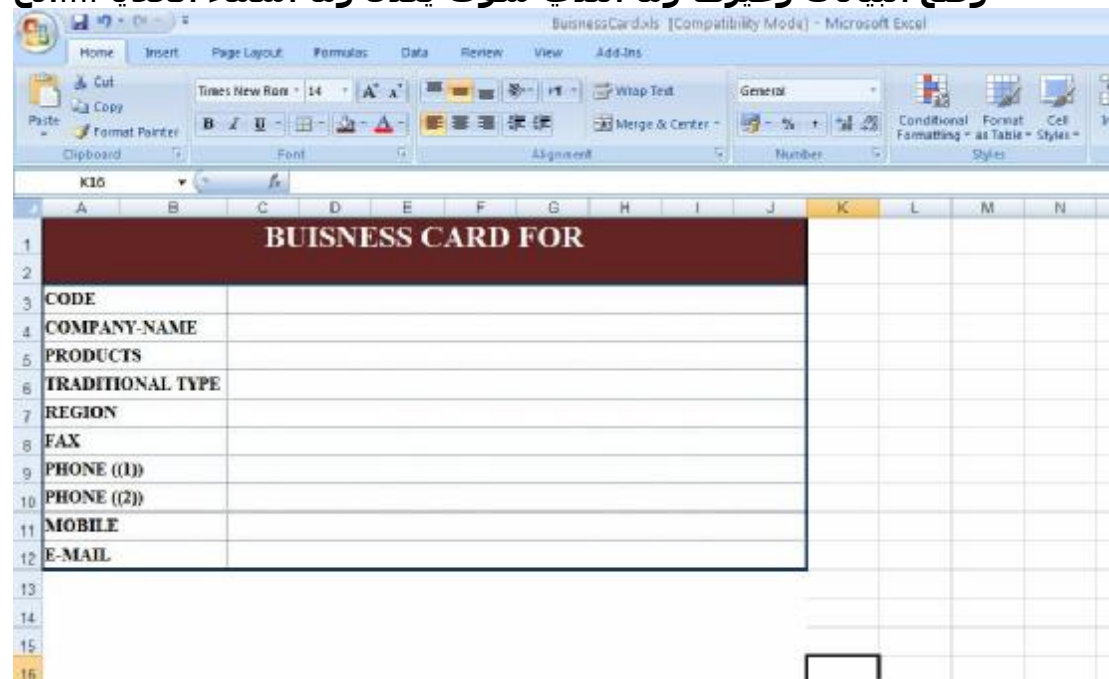
رمز:

```
ReportWB = ReportApp.Workbooks.Open("Put The Source of the  
File Here")  
ReportWS = ReportWB.Worksheets(" حدد الصفحة لو كان فيه أكثر من  
صفحة في نفس الملف")
```

**نأتي الآن إلى الخطوة الأهم وهي أساس عمل البرنامج وهو نقل البيانات
إلى Excel**

**أولا عليك معرفة أن الإكسل ماهو إلا خلايا يعني أن البيانات تكتب في خلايا
فلذلك عند تصميم البرنامج يجب أن تكون فاتح للإكسل وتضع تخيل للبيانات
في الإكسل حتى تعرف الخلايا وهي**

بمعنى اخرى قم بتصميم ماتريده في الاكسل اولا حتى تعرف اين اماكن وضع البيانات وغيرها وما الذي سوف يعدل وما اسماء الخلايا.... الخ



سنستخدم هذا الشكل في ملف الإكسل كمثال ولك الحرية في تصميم الشكل الذي يعجبك
طبعا الوظيفة الأساسية هنا هي إدخال البيانات ولكن سنضيف عليها وظيفة أخرى وهي التحكم في تنسيق الورقة حتى تخرج بالصورة المناسبة حتى نبدأ يجب أن نتبع إحدى طريقتين

كتابة كود يضع البيانات اولا ثم كتابة كود يسوي التنسيقات ثانيا والعكس -2 الصحيح

كرأي الشخصي الأفضل الطريقة الأولى ولكن مادمننا في غرض التعليم
فسنستخدم الطريقة الثانية
وسنبدا بإدخال البيانات إلى الأكسل ثم تنسيق الورقة
إدخال البيانات ****

من الافضل تصميم شكل أولي بالإكسل بحيث تحدد مكان كل معلومة في وبعد ذلك لانتاج إلى إلا كود واحد لكل شيء نريد كتابته وهو خلية محددة
رمز:

```
ReportWS.Range("Address of the Cell").Value="Value"
```

هو عنوان الخلية المراد وضع الكلام فيها "Address of the Cell" حيث "Value"

نأخذ على ذلك مثالين الأول لقيمة نكتبها بأنفسنا والثانية لقيمة يكتبها المستخدم في مربع النص

رمز:

```
ReportWS.Range("A1").Value= "Personal Information"  
ReportWS.Range("B1").Value= txtName.Text
```

هل رايتو مدى سهولة إدخال البيانات
سؤال يطرح نفسه كيف يمكن كتابة معادلة

الاجابة على هذا السؤال تعتمد على مكان الأرقام التي تريد تكوين معادلة لها فلو كانت في برنامجك تكتبها كمعادلة عادية

رمز:

```
ReportWS.Range("B1").Value = val(txtQTY.Text) * val(txtCartons)
```

أما لو كانت في ورقة الإكسل فتكتبها كأنت تكتب معادلة في إكسل

رمز:

```
ReportWS.Range("B1").Value = "=B2+B3"
```

تلميح ذكي:

أحيانا نظطر لكتابة بيانات كثيرة في الاكسل فتكون الطريقة مرهقة والحل
With > > > End With يكمن باستخدام

رمز:

```
With ReportWS
.Range("A1").Value="Personal Information"
.Range("A2").Value="Name"
.Range("B2").Value=txtName.Text
.Range("A3").Value="Age"
.Range("B3").Value=txtAge.Text
.Range("C3").Value="Gender"
.Range("D3").Value=txtGender.Text
End With
```

Range حيث إنك بمجرد كتابة النقطة (.) ستظهر قائمة منسدلة تختار منها وتكمل بعد ذلك

تنسيق البيانات ****

بصراحة هذا هو الجزء المتعب ولكن بقليل من الذكاء تجعله سهلا لو كان عندك معرفة بالبرمجة

تنسيق البيانات حرية لكل مبرمج يعني ما حد يقدر يقيدك بتنسيق معين لان منا ذوقه فلذلك قم بتنسيق ملف الاكسل براحتك وعندما ترى انه صار لكل اعراف التغييرات التي عملتها وجهز نفسك لكتابتها بالكود مناسباً طبعاً الطريقة بسيطة جدا لانها مثلها مثل كتابة البيانات لنعد إلى ملف الإكسل نلاحظ هنا التغييرات التالية

تغيير لون الخلفية إلى اللون الأحمر الغامق من لنفس المنطقة وكذلك جعل 2 لون الكتابة أبيض

4- Times New Romans)) أما باقي الخلايا فقط قمنا بتغيير الخط نوعاً

و(عرضاً 14 (Bold) وحجماً

في الكود الخاص بتنسيق البيانات فمثلاً Value. طبعاً كل اللعب على الكلمة A1 لتغيير تنسيق

رمز:

```
ReportWS.Range("A1").Font.Name = "Times New Romans"
ReportWS.Range("A1").Font.Size = 20
ReportWS.Range("A1").Font.Bold = True
ReportWS.Range("A1").Font.ColorIndex = 2
ReportWS.Range("A1").Interior.ColorIndex = 53
```

الباقى نتركه لكم

طيب بالنسبة للدمج كيف نسويه؟؟؟؟؟؟

الدمج بسيط جدا في المكان اللي تحدد فيه الخلية حدد الخلايا الداخلة في الدمج

Font وبدل الخاصية "A1" إذا نكتب D1 إلى A1 مثلا احنا هنا عندنا من

وخلص Merge تختار

Merge("A1").Merge

طيب نأتي الآن لأهم جزء وهو الذي بسببه استوردنا هذا الكم الهائل

وهو وضع الحدود على الخلايا

بصراحة هذا اللي يخلي الاكسل بايخ في موضوع الطباعة لأن كوده طويل

طبعاً حللوه مع نفسكم وهذا هو من غير فلسفة ولا نقاش

عيبه انه كل خط لازم تكتب فيه 3 أسطر حاولو تفهموا الكود وانا موضح لكم

بعض منها

.

رمز:

```
Range("A1:D3").Borders(xlDiagonalDown).LineStyle = xlNone
Range("A1:D3").Borders(xlDiagonalUp).LineStyle = xlNone
With .Range("A1:D3").Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .ColorIndex = xlAutomatic
    .TintAndShade = 0
    .Weight = xlThick
End With
With .Range("A1:D3").Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .ColorIndex = xlAutomatic
    .TintAndShade = 0
    .Weight = xlThick
End With
With .Range("A1:D3").Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .ColorIndex = xlAutomatic
    .TintAndShade = 0
    .Weight = xlThick
End With
With .Range("A1:D3").Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .ColorIndex = xlAutomatic
    .TintAndShade = 0
    .Weight = xlThick
End With
With .Range("A1:D3").Borders(xlInsideVertical)
    .LineStyle = xlContinuous
    .ColorIndex = xlAutomatic
    .TintAndShade = 0
    .Weight = xlHairline
End With
With .Range("A1:D3").Borders(xlInsideHorizontal)
    .LineStyle = xlContinuous
    .ColorIndex = xlAutomatic
    .TintAndShade = 0
    .Weight = xlHairline
End With
```

نكتفي بهذا القدر لهذا الموضوع ونكمل المرة القادمة
 تلميح: لو أردت عمل أي كود لم أقم بشرحه من أنواع التنسيق حاول اللعب
 ب Value. وإن لم تعرف كيف تكتبه فسأعطيك خطة جهنمية ادخل للأكسل
 وسوي إنشاء ماكرو جديد وسوي التنسيق اللي تبيه ووقف الماكرو وادخل
 على تحرير الماكرو وسوي نسخ ولصق ولما تلاقي أي كلمة بدت ب x في
 الكود محطوط تحت خط أزرق فاعرف انه الفيچوال ماعرفه فعشان كذا تروح
 لل Object Browser بالضغط على F2 وتبحث عنها ولما تلاقيها تاخذ مكانها
 وتسوي لها Import في أعلى الملف
 1-دمج (Merge) الخلايا من A1 إلى D1 تغيير الخط إلى (Times New
 Romans وجعل الخط (20) وعريضا-A1 1) ReportsWS.Range("A1 كتابة
 الكود الخاص بكل بيان على حدة مع كود تنسيق الخلية اللي هي فيها (مثلا نكتب كود
 يضع الاسم في b2 ونخلي الخط) Arial

كاتب الموضوع : [Athman](#)

الموضوع الثاني :

التكامل مع برامج الأوفيس

السلام عليكم ورحمة الله وبركاته وبعد
 راح نتكلم اليوم عن تكامل برامجنا اللي نسويها مع برامج الأوفيس المقدمة من عممتنا
 مايكروسوفت
 بصراحة ورد الكثير من التساؤلات عن كيفية التعامل مع برامج الأوفيس لأنها صارت
 ضرورة من ضروريات الحياة

(The Most Power) ولكن أنا هنا سوف أتكلم فقط عن التكامل مع برنامج الإكسل كونها
 الكامل للمبرمج أو الأقوى من بين إصدارات مايكروسوفت التي تتيح التحكم (Full
 بحيث يكون ذو فائدة كبيرة

Excel ؟؟؟؟ ولعل التساؤل يطرح نفسه: ما الهدف من التكامل مع
 أستطيع أن أخص أبرز الأسباب اللي تجعلنا نتكامل معه في عدة نقاط
 إنشاء تقارير باحترافية عالية-1
 سهولة الطباعة باستخدام الإكسل-2
 قوية ومدعومة (Forms) إمكانية إنشاء نماذج-3
 وغيرها من الأسباب التي ستتضح لاحقا من خلال هذا الموضوع بإذن الله
 Excel مثال عملي للتكامل مع
 وهي سنقوم في هذا البرنامج بإنشاء فورم لإدخال البيانات الشخصية ثم طباعتها
 هاتف ذكر/أنثى) وأيضا بكتابة مفكرة) Gender والجنس Ageالعمر Name الاسم
 وطباعتها
 بداية علينا استيراد 3 ملفات هي على التوالي
 Interop.Excel.dll
 Interop.Microsoft.Office.Core.dll
 Interop.VBIDE.dll
 وهي مرفقة في هذا الموضوع
 ومن ثم استيراده في محرر الأكواد في أعلى الصفحة
 إلى الآن أكملنا الخطوات الأساسية والتي غالبا ستفعل لمرة واحدة فقط
 نأتي الآن إلى الخطوة الأهم وهي التكامل مع الإكسل

التعامل مع الأكسل يكون في 3 خطوات

1-فتح البرنامج وإنشاء ورقة جديدة-

2-نقل البيانات من البرنامج إلى الأكسل (ثم إلى البرنامج إذا لزم الأمر-

3-حفظ المستند (إختياري)

=====

1-فتح البرنامج وإنشاء ورقة جديدة-

ويم ذلك من خلال انشاء 3 متغيرات

الأول يخص برنامج الأكسل ككل

الثاني يخص ملف الأكسل

Sheet الثالث يخص ورقة البيانات أو

رمز:

```
1-Dim ReportApp As New Excel.Application
```

```
2-Dim ReportWB As Workbook
```

```
3-Dim ReportWS As Worksheet
```

ومن ثم إسناد قيمة كل من الثاني والثالث والثالث للثاني

رمز:

```
ReportWB = ReportApp.Workbooks.Add(XIWBATemplate.xlWBATWorksheet)
```

لإنشاء ملف اكسل جديد'

رمز:

```
ReportWS = ReportWB.Sheets.Add(, ReportWB.Worksheets(1))
```

لفتح صفحة جديدة في ملف الأكسل'

الآن الملف أصبح موجود وإن لم تصدق فجرب إضافة هذا السطر بعد الاسطر السابقة

رمز:

```
ReportApp.Visible = True
```

ولكن نصيحتي ألا تستخدم (فستلاحظ أنه قد تم فتح ملف أكسل جديد بورقة واحدة

(هذا السطر في برامجك وستعرف لماذا لاحقا

سؤال:كيف نفتح ملف محفوظ من الأساس

الإجابة بتغيير السطرين إلى

رمز:

```
ReportWB = ReportApp.Workbooks.Open("Put The Source of the File Here")
```

```
ReportWS = ReportWB.Worksheets("حدد الصفحة لو كان فيه أكثر من صفحة في نفس الملف")
```

2-نقل البيانات من البرنامج إلى الأكسل (ثم إلى البرنامج إذا لزم الأمر-

Excel تأتي الآن إلى الخطوة الأهم وهي أساس عمل البرنامج وهو نقله الى الـ

فلذلك عند أولاً عليك معرفة أن الأكسل ماهو إلا خلايا يعني أن البيانات تكتب في خلايا

في الأكسل تصميم البرنامج يجب أن تكون فاتح للأكسل وتضع تخيل للبيانات وهي

حتى تعرف الخلايا

سنستخدم هذا الشكل في ملف الأكسل كمثال ولك الحرية في تصميم الشكل الذي

يعجبك

طبعا الوظيفة الأساسية هنا هي إدخال البيانات ولكن سنضيف عليها وظيفة أخرى

وهي التحكم في تنسيق الورقة حتى تخرج بالصورة المناسبة

حتى نبدأ يجب أن نتبع إحدى طريقتين

كتابة الكود الخاص بكل بيان على حدة مع كود تنسيق الخلية اللي هي فيها (مثلا -1

) Arial ونخلي الخط b2 نكتب كود يضع الاسم في

كتابة كود يضع البيانات اولا ثم كتابة كود يسوي التنسيق ثانيا والعكس الصحيح-2

كرأي الشخصي الأفضل الطريقة الأولى ولكن مادمننا في غرض التعليم فسنستخدم

الطريقة الثانية

وسنبدا بإدخال البيانات إلى الأكسل ثم تنسيق الورقة

إدخال البيانات*****

من الافضل تصميم شكل أولي بالإكسل بحيث تحدد مكان كل معلومة في خلية وبعد ذلك لانتاج إلى إلا كود واحد لكل شيء نريد كتابته وهو محددة

رمز:

```
ReportWS.Range("Address of the Cell").Value="Value"
```

"Value" هو عنوان الخلية المراد وضع الكلام فيها "Address of the Cell" حيث نأخذ على ذلك مثالين الأول لقيمة نكتبها بأنفسنا والثانية لقيمة يكتبها المستخدم في مربع النص

رمز:

```
ReportWS.Range("A1").Value= "Personal Information"  
ReportWS.Range("B1").Value= txtName.Text
```

هل رايتو مدى سهولة إدخال البيانات

سؤال يطرح نفسه كيف يمكن كتابة معادلة

الاجابة على هذا السؤال تعتمد على مكان الأرقام التي تريد تكوين معادلة لها فلو كانت في برنامجك تكتبها كمعادلة عادية

رمز:

```
ReportWS.Range("B1").Value= val(txtQTY.Text)*val(txtCartons)
```

أما لو كانت في ورقة الإكسل

فتكتبها كأنت تكتب معادلة في إكسل

رمز:

```
ReportWS.Range("B1").Value= "=B2+B3"
```

تلميح ذكي:

أحيانا نظطر لكتابة بيانات كثيرة في الاكسل فتكون الطريقة مرهقة والحل يكمن باستخدام With>>>End With

رمز:

```
With ReportWS  
.Range("A1").Value="Personal Information"  
.Range("A2").Value="Name"  
.Range("B2").Value=txtName.Text  
.Range("A3").Value="Age"  
.Range("B3").Value=txtAge.Text  
.Range("C3").Value="Gender"  
.Range("D3").Value=txtGender.Text  
End With
```

وتكمل Range حيث إنك بمجرد كتابة النقطة (.) ستظهر قائمة منسدلة تختار منها بعد ذلك

تنسيق البيانات*****

بصراحة هذا هو الجزء المتعب ولكن بقليل من الذكاء تجعله سهلا لو كان عندك معرفة بالبرمجة

منا تنسيق البيانات حرية لكل مبرمج يعني ماحد يقدر يقيدك بتنسيق معين لان لكل اعرف ذوقه فلذلك قم بتنسيق ملف الاكسل براحتك وعندما ترى انه صار مناسباً

التغييرات اللي عملتها وجهز نفسك لكتابتها بالكود

طبعا الطريقة بسيطة جدا لانها مثلها مثل كتابة البيانات

لنعد إلى ملف الإكسل

نلاحظ هنا التغييرات التالية

1- D1 إلى A1 الخلايا من (Merge) دمج-1
 تغيير لون الخلفية إلى اللون الأحمر الغامق من لنفس المنطقة وكذلك جعل لون 2
 الكتابة أبيض
 3- (Bold) وجعل الخط (20) وعريض (Times New Romans) تغيير الخط إلى 3
 (Bold) وحجما (Times New Romans) أما باقي الخلايا فقط قمنا بتغيير الخط نوعا-4
 (وعرضا 14)
 في الكود الخاص بتنسيق البيانات فمثلا لتغيير Value. طبعا كل اللعب على الكلمة
 A1 تنسيق
 رمز:

```
ReportWS.Range("A1").Font.Name = "Times New Romans"
ReportWS.Range("A1").Font.Size = 20
ReportWS.Range("A1").Font.Bold = True
ReportWS.Range("A1").Font.ColorIndex = 2
ReportWS.Range("A1").Interior.ColorIndex = 53
```

الباقي نتركه لكم
 طيب بالنسبة للدمج كيف نسويه؟؟؟؟؟
 الدمج بسيط جدا في المكان اللي تحدد فيه الخلية حدد الخلايا الداخلة في الدمج
 تختار Font وبدل الخاصية "A1" إذا نكتب D1 إلى A1 مثلا احنا هنا عندنا من
 Merge خلاص
 ReportsWS.Range("A1").Merge
 طيب نأتي الآن لأهم جزء وهو الذي بسببه استوردنا هذا الكم الهائل
 وهو وضع الحدود على الخلايا
 بصراحة هذا اللي يخلي الاكسل بايخ في موضوع الطباعة لأن كوده طويل طبعا حله
 مع نفسكم وهذا هو من غير فلسفة ولا نقاش
 عيبه انه كل خط لازم تكتب فيه 3 أسطر حاولو تفهموا الكود وانا موضح لكم بعض منها

رمز:

```
Range("A1:D3").Borders(xlDiagonalDown).LineStyle = xlNone
.Range("A1:D3").Borders(xlDiagonalUp).LineStyle = xlNone
With .Range("A1:D3").Borders(xlEdgeLeft)
.LineStyle = xlContinuous
.ColorIndex = xlAutomatic
.TintAndShade = 0
.Weight = xlThick
End With
With .Range("A1:D3").Borders(xlEdgeTop)
.LineStyle = xlContinuous
.ColorIndex = xlAutomatic
.TintAndShade = 0
.Weight = xlThick
End With
With .Range("A1:D3").Borders(xlEdgeBottom)
.LineStyle = xlContinuous
.ColorIndex = xlAutomatic
.TintAndShade = 0
.Weight = xlThick
End With
With .Range("A1:D3").Borders(xlEdgeRight)
.LineStyle = xlContinuous
.ColorIndex = xlAutomatic
.TintAndShade = 0
```

```

.Weight = xlThick
End With
xlInsideVertical
With .Range("A1:D3").Borders(xlInsideVertical)
.LineStyle = xlContinuous
.ColorIndex = xlAutomatic
.TintAndShade = 0
.Weight = xlHairline
End With
With .Range("A1:D3").Borders(xlInsideHorizontal)
.LineStyle = xlContinuous
.ColorIndex = xlAutomatic
.TintAndShade = 0
.Weight = xlHairline
End With

```

نكتفي بهذا القدر لهذا الموضوع ونكمل المرة القادمة
 تلميح: لو أردت عمل أي كود لم أقم بشرحه من أنواع التنسيقات حاول اللعب بـ Value.
 وإن لم تعرف كيف تكتبه فسأعطيك خطة جهنمية ادخل للأكسل وسوي انشاء ماكرو
 جديد وسوي التنسيق اللي تبيه ووقف الماكرو وادخل على تحرير الماكرو وسوي نسخ
 ولصق ولما تلاقي أي كلمة بدئت بـ xl في الكود محطوط تحت خط أزرق فأعرف انه
 الفيچوال ماعرفه فعشان كذا تروح للـ Object Browser بالضغط على F2 وتبحث عنها
 ولما تلاقيها تاخذ مكانها وتسوي لها Import في أعلى الملف

طيب راح اعطيكم تطبيق تسوونه وانتظر اجاباتكم وما راح اكمل لما تعطوني الجواب
 ابي الكود اللي يسوي فرز لعمود معين

كاتب الموضوع : [Athman](#)

الموضوع الثالث :

بسم الله الرحمن الرحيم

وصف الموضوع-

الموضوع : : أساليب حفظ والتعامل مع ملفات ومايكروسوفت أوفيس بترميز XML
 لغة التطبيق 2008 - 2005 visual BASIC :
 إعداد : محمد م [mmmdev] - arabteam [vbm] -
 وصف المحتوى : تطبيق بسيط لحفظ بيانات جدول بيانات بسيط في صورة ملف إكسل
 بتنسيق xls

مقدمة

أصبح ترميز XML قياسيا في تبادل وتداول البيانات وقامت الشركات الكبرى مثل
 مايكروسوفت بدعمها في تطبيقاتها
 كنا في السابق نستخدم مكونات com للتعامل مع برامج أوفيس المختلفة والتي كانت
 تكتب ملفات بترميز ثنائي binary
 اليوم أصبح بإمكاننا كتابة ملفات أوفيس كاملة من تطبيقاتنا باستخدام ترميز XML

- الخاص بالأوفيس والتابع لمايكروسوفت
 فى مثالنا هنا سنستخدم مكتبة جاهزة تقوم بذلك نيابة عنا ويمكننا أن نقوم نحن بذلك
 بدراستنا لترميز XML
 الخاص بمايكروسوفت أوفيس
 إسم المكتبة CarlosAg.ExcelXmlWriter.dll
 المكتبة بالمرفقات
 طريقة العمل
 • سنقوم بكتابة جدول بسيط فى data set يحتوى على معلومات أسماء الكتب
 ومؤلفيها
 • ثم سنقوم بحفظ محتويات هذا المجلد لملف إكسل على المسار c:\vbm.xls

الشكل النهائى للتطبيق

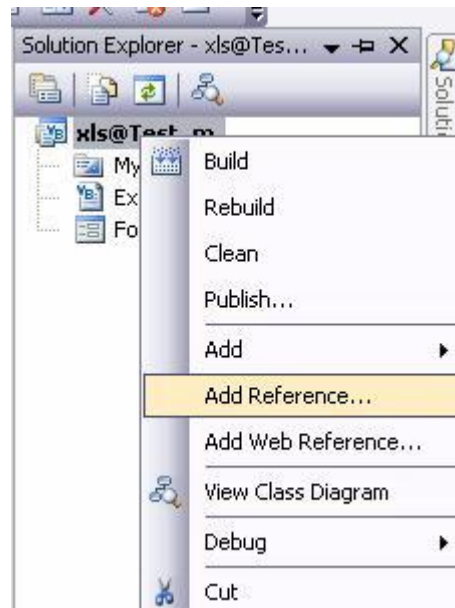


id	book	author
1	موسوعة اليهود وا...	د/عبد الوهاب المسيرى
2	خوار مع صديقي ا...	د/مصطفى محمود
3	العقريات	عباس محمود العقاد
4	قصة الحضارة	ويل ديورانت
5	الأنسان الخائر بين...	د/عبد المسخن صالح
6	موسوعة مصر ال...	د/سليم حسن
7	رسالة التوحيد	العلامة الشيخ محمد عبدة
8	حول العالم فى 8...	جول فيرن
9	اليهود إثنولوجيا	د/جمال حمدان
10	معجزة القرآن	العلامة الشيخ الشعراوي
11	أناكارينا	تولستوى
12	... لإقتصاد مصر كيف	فاروق جويدة
13	البخلاء	الجاحظ
*		

Export to Excel

خطوات العمل

1. ننشأ تطبيق ويندوز جديد.
2. نضع على التطبيق مكون dataGridView ومكون زر أمر 3. button نجعل خاصية dock لزر الأمر على القيمة Bottom
4. نجعل خاصية dock لجدول عرض البيانات على القيمة fill
5. نبدأ فى وضع الكود
 "أولا نقوم بعمل جدول البيانات الخاص بنا ومن ثم نعرضه على النموذج
 "ثانيا نقوم بعمل مرجع لمكتبة CarlosAg.ExcelXmlWriter.dll
 "عن طريق add reference"



الأكواد

- كود البرنامج الرئيسي

رمز:

'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

xml بترميز ومايكروسوفت أوفيس مع ملفات أساليب حفظ والتعامل : : الموضوع'

visual basic 2005 – 2008 : التطبيق

' : محمد م م - [m m m d e v]

xls بتنسيق فى صورة ملف إكسل لحفظ بيانات جدول بيانات بسيط تطبيق بسيط :

' www.vb4arab.com

'@@'

Imports System.Data

Public Class Form1

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
```

System.EventArgs) Handles MyBase.Load

Dim books ds As New DataSet

فئة البيانات متغير من نوع تعريف¹

```
Dim books As New DataTable("Books")
```

تعريف متغير من من جدول البيانات وسيكون الجدول الخاص بنا'

تعريف الحقول الخاصة بالجدول وهي حقول كود التعريف وإسم '

الكتاب والمؤلف

```
With books tb a.Columns.Add("id", GetType(Integer))
```

السماح بزيادة تلقائية لرقم التعريف '1

```
.AutoIncrement = True
```

```

        الزيادة تكون بمقدار 1
        .AutoIncrementSeed = 1
        عدم السماح بالقيمة الخالية
        .AllowDBNull = False
        لا يمكن التكرار
        .Unique = True
    End With

    With books_tb_a.Columns.Add("book", GetType(String))
        أقصى حجم 50
        .MaxLength = 50
    End With

    With books_tb_a.Columns.Add("author", GetType(String))
        .MaxLength = 50
    End With

'=====
        البيانات والسجلات ملا
        وربط كائن البيانات بجدول البيانات

        books_ds.Tables.Add(books_tb_a)
        .....
        books_tb_a.Rows.Add(New Object() {1, "موسوعة اليهود
        واليهودية والصهيونية", "د/عبد الوهاب المسيري
        حوار مع صديقي {2, "
        books_tb_a.Rows.Add(New Object() {2, "د/مصطفى محمود", "الملحد
        العبقريات", "عباس {3, "
        books_tb_a.Rows.Add(New Object() {3, "قصة الحضارة", "ويل {4, "
        books_tb_a.Rows.Add(New Object() {4, "الأنسان الحائر بين {5, "
        books_tb_a.Rows.Add(New Object() {5, "العلم والخرافة", "د/عبد المسحن صالح
        books_tb_a.Rows.Add(New Object() {6, "موسوعة مصر القدية {6, "
        books_tb_a.Rows.Add(New Object() {7, "رسالة التوحيد {7, "
        books_tb_a.Rows.Add(New Object() {8, "حول العالم فى 80 يوم {8, "
        books_tb_a.Rows.Add(New Object() {9, "اليهود إثنوبولوجيا {9, "
        books_tb_a.Rows.Add(New Object() {10, "معجزة القرأن {10, "
        books_tb_a.Rows.Add(New Object() {11, "أناكارينا {11, "
        books_tb_a.Rows.Add(New Object() {12, "إقتصاد مصر كيف ضاع {12, "
        books_tb_a.Rows.Add(New Object() {13, "البخلء", "الجاحظ"}))
        .....
        ربط جدول عرض البيانات على النموذج لدينا بكائن فئة البيانات
        لعرض البيانات منة

        Me.DataGridView1.DataSource = books_ds
        Me.DataGridView1.DataMember = books_tb_a.ToString

```

```

End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim fileName As String = "c:\vbm.xls"
    ' ExcelGenerator استخدام فئة
    ' ملف لتصدير البيانات وإنشاء
    ExcelGenerator.Generate(DataGridView1).Save(fileName)
    ' استدعاء التطبيق إكسل
    Process.Start(fileName)
End Sub
End Class

```

- نضع الكود الخاص بإنشاء وتنسيق ملف xls
- في مودول جديد

رمز:

```

Imports CarlosAg.ExcelXmlWriter
' استدعاء فضاء الأسماء الخاص بمكتبة التعامل مع xls
Imports System
Imports System.Drawing
Imports System.Xml
Imports System.Windows.Forms

Public Class ExcelGenerator
    ' Methods
    ' الملف تكوين شكل
    ' برنامجنا المستخدم مع وضع البارميتر حسب
    Public Shared Function Generate(ByVal dataGridView As DataGridView) As
Workbook
        Dim workbook As New Workbook
        Dim worksheet As Worksheet = workbook.Worksheets.Add("Sheet 1")
        Dim worksheetRow As New WorksheetRow
        Dim dataGridViewColumn As DataGridViewColumn
        For Each dataGridViewColumn In dataGridView.Columns
            worksheet.Table.Columns.Add(New
WorksheetColumn(dataGridViewColumn.Width))
            worksheetRow.Cells.Add(New
WorksheetCell(dataGridViewColumn.HeaderText))
        Next
        worksheet.Table.Rows.Insert(0, worksheetRow)
        Dim worksheetDefaultStyle As WorksheetStyle =
ExcelGenerator.GetWorksheetStyle(dataGridView.DefaultCellStyle, "Default")
        workbook.Styles.Add(worksheetDefaultStyle)
        Dim rowIndex As Integer
        For rowIndex = 0 To dataGridView.RowCount - 1
            worksheetRow = worksheet.Table.Rows.Add
            Dim columnIndex As Integer
            For columnIndex = 0 To dataGridView.ColumnCount - 1

```

```

Dim cell As DataGridViewCell =
dataGridView.Item(columnIndex, rowIndex)
Dim cellStyle As WorksheetStyle =
ExcelGenerator.GetWorksheetStyle(cell.InheritedStyle, String.Concat(New Object() {"column",
columnIndex, "row", rowIndex}))
If (Not cellStyle Is Nothing) Then
    workbook.Styles.Add(cellStyle)
Else
    cellStyle = worksheetDefaultStyle
End If
Dim dataType As DataType =
ExcelGenerator.GetDataType(cell.ValueType)

worksheetRow.Cells.Add(cell.FormattedValue.ToString, dataType, cellStyle.ID)
Next columnIndex
Next rowIndex
Return workbook
End Function

Private Shared Function GetColorName(ByVal color As Color) As String
Return ("#" & color.ToArgb.ToString("X").Substring(2))
End Function

Private Shared Function GetDataType(ByVal valueType As Type) As DataType
If (Not valueType Is GetType(DateTime)) Then
    If (valueType Is GetType(String)) Then
        Return DataType.String
    End If
    If (((((valueType Is GetType(SByte)) OrElse (valueType Is
GetType(Byte))) OrElse ((valueType Is GetType(Short)) OrElse (valueType Is
GetType(UInt16)))) OrElse (((valueType Is GetType(Integer)) OrElse (valueType Is
GetType(UInt32))) OrElse ((valueType Is GetType(Long)) OrElse (valueType Is
GetType(UInt64)))) OrElse ((valueType Is GetType(Single)) OrElse (valueType Is
GetType(Double)))) OrElse (valueType Is GetType(Decimal))) Then
        Return DataType.Number
    End If
End If
Return DataType.String
End Function

Private Shared Function GetWorksheetStyle(ByVal dataGridViewCellStyle As
DataGridViewCellStyle, ByVal id As String) As WorksheetStyle
Dim worksheetStyle As WorksheetStyle = Nothing
If (Not dataGridViewCellStyle Is Nothing) Then
    worksheetStyle = New WorksheetStyle(id)
    If Not dataGridViewCellStyle.BackColor.IsEmpty Then
        worksheetStyle.Interior.Color =
ExcelGenerator.GetColorName(dataGridViewCellStyle.BackColor)
        worksheetStyle.Interior.Pattern =
StyleInteriorPattern.Solid
    End If
    If Not dataGridViewCellStyle.ForeColor.IsEmpty Then
        worksheetStyle.Font.Color =
ExcelGenerator.GetColorName(dataGridViewCellStyle.ForeColor)
    End If
    If (Not dataGridViewCellStyle.Font Is Nothing) Then

```

```

dataGridViewCellStyle.Font.Bold = worksheetStyle.Font.Bold =
dataGridViewCellStyle.Font.Name = worksheetStyle.Font.FontName =
dataGridViewCellStyle.Font.Italic = worksheetStyle.Font.Italic =
dataGridViewCellStyle.Font.Size = worksheetStyle.Font.Size =
dataGridViewCellStyle.Font.Strikeout = worksheetStyle.Font.Strikethrough =
dataGridViewCellStyle.Font.Underline = worksheetStyle.Font.Underline =
If(dataGridViewCellStyle.Font.Underline, UnderlineStyle.Single, UnderlineStyle.None)
    End If
    worksheetStyle.Borders.Add(StylePosition.Top,
LineStyleOption.Continuous, 1, "Black")
    worksheetStyle.Borders.Add(StylePosition.Right,
LineStyleOption.Continuous, 1, "Black")
    worksheetStyle.Borders.Add(StylePosition.Bottom,
LineStyleOption.Continuous, 1, "Black")
    worksheetStyle.Borders.Add(StylePosition.Left,
LineStyleOption.Continuous, 1, "Black")
    End If
    Return worksheetStyle
End Function

End Class

```

أخيرا نقوم بتشغيل التطبيق والضغط على زر التصدير
ولرؤية طريقة التنسيق المتبعة من مايكروسوفت
يمكننا عرض الملف الذي تم إنشاؤه ببرنامج المفكرة لنرى تنسيق xml

أرجو أن يكون الشرح عند حسن ظنكم

كاتب الموضوع : mmmdev

الى هنا نكتفي من الأكسل

حان وقت التكامل مع word **

ملاحظة : الموضوع الأخير يحتوي على صور تبين كيفية استيراد مكتبة الورد
الموضوع الأول :

لاحظت ان البعض يتعاملون مع برنامج Excel والذي قام بشرحة الاخ Athman ولاحظت
انه من الواجب معرفة التعامل مع Word و Outlook والتي تعتبر جزء لا يتجزأ من
Automating Microsoft Office Applications ولنبدا هنا مع برنامج الـ Word.

التكامل مع Word من خلا ل: vb.net

لاستخدام برنامج word من خلا برنامجك عليك ان تضيف الـ Reference الاتي من
القائمة Add Reference - Project ثم - التبويب Com نختار Microsoft Word 11.0
Object Library او حسب الاصدار الذي عندك ثم Ok الان اصبح الـ Reference جاهز
ولنأتي الى الكود:

لنستطيع التعامل مع هذا البرنامج من خلا ل الكود هناك مجموعة من الكائنات التي
يجب التصريح عنها ومنها
الاتي:

```
Dim WrdApp As New Word.Application
```

تشغيل او وذلك ليتسنى لنا Word هو إنشاء كائن جديد من نوع وظيفة الكود السابق
. على مستوى الفورم ويتم التصريح عن هذا المتغير , Word فتح تطبيق
مستند (يجب ان نصرح عن متغير النوع) Document او انشاء Document وفتح
ونسند لـ المتغير الذي صرحنا عنه سابقاً فمثلاً لفتح مستند جديد Document
: نستخدم الكود الاتي :

لأننا هنا نريد ان نتعامل مع Document وهو ذكرناه نصرح عن المتغير الذي : اولا
: التصريح عن المتغير الاتي فمن الطبيعي مستند

```
Dim WrdDocu As New Word.Document
```

: سنستخدم الكود الاتي ولذلك انشاء مستند جديد منا هو وهنا المطلوب

```
Me.WrdDocu = WrdApp.Documents.Add
```

يجب كتابة الكود الاتي وليتم اظهاره طيب الان تم انشاء مستند جديد ولكن لا يظهر تحت الكود السابق

```
WrdApp.Visible = True
```

ان ولكن هنا سنواجه مشكلة وهي تكدر المهام في المعالج ولو لاحظت ستلاحظ الكود الاتي وللتخلص من ذلك سنضع Process عدد المهام قد زاد اقصد مهام المعالج او Form _ FomrClosing في حدث :

```
Private Sub Form3_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.  
FormClosingEventArgs) Handles Me.FormClosing  
  
    WrdApp = Nothing  
  
End Sub
```

: ملاحظة

حتى وان حاولت اقفال المستند يدوياً ستلاحظ ان المعالج يحتوي على ال Winword.exe اقل قد افعل البرنامج مع ان البرنامج قد افعل

: ولفتح مستند موجود نستخدم الكود الاتي

```
WrdDocu = WrdApp.Documents.Open("MyWord.doc")  
WrdApp.Visible = True
```

ولناتي ,, الى الان عرفنا فقط كيف تكوين مستند جديد وفتح مستند موجود من سابق اسم تريد الى اضافة نص الى هذا المستند الجديد وحفظ المستند باسم جديد اي باي؟؟

الذي نستطيع من خلاله Range كائن مهم جداً وهو الكائن اولاً لعمل السابق هناك . اضافة نص او حذف نص الى اخره

اولاً لكتاب نص في المستند او في مستند جديد نستخدم هذا الكائن ليساعدنا في : كالاتي InsertAfter عملية اضافة نص الى المستند عن طريق الطريقة

```
WrdDocu.Range.InsertAfter("Jihad alariqi")
```

المستند وسيط واحد وهذا الوسيط هو النص المراد اضافته الى حيث ان لهذه الطريقة

: ملاحظة

ليصبح بالكود السابق تم اضافة النص مباشرة الى المستند ويمكن ان نطور البرنامج اكثر مرونة وفعالية عن طريق ادراج نص عن طريق صندوق نصوص او عن طريق استخدام صندوق نصوص . كالآتي :

```
WrdDocu.Range.InsertAfter(TextBox1.Text)
```

عن صممناه فيمكن ان نقوم بذلك في المتسند الذي واذا اردنا ان نحفظ هذا النص : طريق الكود الاتي

```
WrdApp.Documents.Close(Word.WdSaveOptions.wdSaveChanges)  
WrdApp = Nothing
```

لكن لو اردنا ان نحفظ هذا المستند باسم فكيف يمكن ذلك ؟؟
: باستخدام الكود يمكن ان نعطي هذا المستند اسم وذلك الجواب

```
Me.WrdDocu.SaveAs("MyWordApp")
```

طيب قد يسأل البعض اين تم حفظ المستند ؟؟؟

.. MyDocuments او تم حفظ المستند في المستندات

نضيف المسار المراد الى محدد يمكن ان مسار لكن ان اردنا ان نحفظ المستند في : كالآتي الاسم

```
Me.WrdDocu.SaveAs("c:\MyWordApp")
```

. جرب ان تطبق الكود وسترى
يهمنا الى الان هو ما استخدمناه في وسائط عدة ولكن الذي لها SaveAs الطريقة طبعاً
. الكود السابق

العملية لنعد الى حفظ المستند باسم ... كما اسلفت اننا نستطيع ان نقوم بهذه
اننا قمنا بحفظ ولكن ماذا لو WrdDocu التابعة للكائن SaveAs باستخدام الطريقة
المستند باسم مستند موجود من سابق فما ستكون النتيجة ؟؟؟

!!! أضمن لك ان النتيجة لن تكون مرضية

لأنك عندما تقوم بتسمية المستند او بحفظ المستند باسم لمستند موجود من سابق

وينفس الاسم سيتم حذف ما هو مكتوب في المستند القديم ويستبدل بما كتب في
!!! المستند الجديد وهذا غير مقبول

إذن ما الحل لهذه المشكلة ؟

تعطينا خيار والتي Save بالطريقة SaveAs يمكن ان نستبدل عملية الطريقة السابقة
.. الحفظ في اي مكان نريد

: وهو باستخدام الكود الاتي بسيط جداً الحل

```
WrdApp.Documents.Save()
```

: كالاتي wrdDocu التابعة للكائن Save باستخدام الطريقة او يمكن مباشرة

```
WrdDocu.Save()
```

في المكان ثم نحفظ المستند SaveDialog او يمكن ان نقوم بتطوير البرنامج باضافة
التابعة للكائن Save As الذي نريد ونستطيع تمرر الاسم الذي نريده الى الطريقة
WrdDocu كما يلي :

```
WrdDocu.SaveAs(SaveFileDialog1.FileName)
```

. هذه طرق مختلفة لعملية الحفظ

Form FormClosing حدث وضع هذا الكود في ايضاً لا ننسى

```
WrdApp.Documents.Close(Word.WdSaveOptions.wdSaveChanges)  
WrdApp.Quit()  
WrdApp = Nothing
```

من السطر الاول من الكود وظيفة اغلاق المستندات المفتوحة وان كان هناك اكثر
الوسيط وهذا (Close مستند فسيتم غلقه , وتم تمرير وسيط لهذه الطريقة (الطريقة
التابعة لل enumeration ويتم تمرير قيمة من النوع SaveChange هو عبارة عن
WDSaveOptions والتي لها ثلاث قيم هي :

wdDoNotSaveChanges,
wdPromptToSaveChanges,
wdSaveChanges.

وسيطين اخرين الى اجانب الوسيط wrdDocu التابعة للكائن Close ايضاً للطريقة
هما على الترتيب السابق
originalFormat و routeDocument

بشكل عام

هذا كان بالنسبة الى انشاء مستند والكتابة فيه , والان لناتي الى الجزء الالهم وهو
Objects That Represent Text الكائنات التي تمثل النص...

ولنا تكملة إن شا الله في الدرس الثاني .. بعنوان

Objects That Represent Text

والسلام عليكم ورحمة الله وبركاته,,

كاتب الموضوع : جهاد العريفي

الموضوع الثاني :

Office (Automating Microsoft
Office Applications) نواصل تكملة الدرس السابق بموضوع التكامل مع :

والذي يمثل حلقة الوصل لقسم النص او Range اهم هذه الكائنات هو الكائن لعل
section of text. . جزؤ من كلمة او كلمات هذا القسم قد يكون كلمة او .

؟؟ طبعاً قد يسال احدكم ما الفائدة من هذا الكلام

عندما تستخدم اي هل لاحظت انك !! Word اقول له هل سبق و تعاملت مع برنامج ال
على صفحات تستطيع ان تحدد نص معين في المستند الذي قد يحتوي Word برنامج
Word مستند من الكلمات والجمل والفقرات وهل جربت ان تبحث عن كلمة معينة في
..من قبل الخ

هذه العمليات من خلال برنامجنا عن طريق يمكننا ان نقوم بهذه العمليات واكثر من
.. ستاتي لاحقاً هذه الكائنات وغيرها من الكائنات التي

Range الكائن :

خلاله نستطيع ان نتعامل مع اصغرا لاشياء في المستند من فوائد هذا الكائن اننا من
.. المستند او حذف نص من المستند وغيرها مثل اضافة نص جديد الى

Selection الكائن :

, يستخدم هذا الكائن في الوصول الى جزء من المستند ,
هاذين الكائنن لاحقاً سيتم مناقشة .

.. بتوسع اكثر فلا تقلق

: لمعالجة وحدات معينة من نص نستخدم المجموعات الاتية

Paragraphs Collection :

. والتي تمثل فقرات النص Paragraph الفئات او الفقرات وهذه المجموعة مشتقة من
Word Collection :

والتي تمثل الكلمات في النص Word المجموعة مشتقة من الفئة هذه

Characters Collection :

بدورها تمثل ايش تمثل !!! والتي Character وهذه المجموعة بدورها مشتقة من الفئة
.. الحروف

The Documents Collection and the Document Object :

Word كما لا حظنا اسابقاً ان الكائن التالي في الترتيب الهرمي بعد الكائن من المعروف
او افعال اي Word الذي من خلاله نستطيع فتح مستند Document Object الكائن هو
مستند .

ووكل المجموعات Document collection كل المستندات المفتوحة تعود الى المجموعة
التي Add وايضاً الطريقة (رقم الصفحة المفتوحة) Count تدعم هذه المجموعة الخاصية
التي تقوم باغلاق المستند Remove من خلالها يمكن ان نضيف مستند جديد والطريقة
الحالي .

Item وثيقة مفتوحة نستخدم الطريقة وللوصول الى وثيقة مفتوحة او دخول الى
ولتحديد فهرس او دليل الوثيقة نستخدم الكودالاتي Documents Collection للمجموعة
:

```
WrdApp.Documents.Item(1)
```

: يمكن ان نستخدم اسم المستند الحالي بالكودالاتي او

```
WrdApp.Documents.Item("Chapter01.doc")
```

Document والتابعة للمجموعة Open الوثيقة الحالية نستخدم الطريقة ولفتح
: على النحو الاتي Collections

```
WrdApp.Documents.Open("Filename")
```

. هو مسار الملف Open الممر الى الطريقة الوسيط
والتابعة الى نفس المجموعة على النحو Add ولانشاء ملف جديد نستخدم الطريقة
الاتي :

<code>WrdApp.Documents.Add(template</code>	<code>newTemplate)</code>
--	---------------------------

الاول نحدد فية قالب الملف الذي سيكون قاعدة للوثيقة الجديدة , الوسيط الوسيط
True وان كان يحمل القيمة False او True من النوع المنطقي اي انه يحمل اما الثاني
. سيقوم بإنشاء قالب جديد (Word) فهذا معناه ان البرنامج

العمليات ستطبق على الوثيقة النشطة والتي تمثل بالخاصية اغلب هذه
WrdApp والذي في مثالنا هذا هو Word التابعة للكائن ActiveDocument
المحدد في المستند النشط نستخدم التعبير الاتي وللوصول الى النص :

<code>WrdApp.ActiveDocument.Section()</code>
--

Document من الكائن Active ويمكن ان نجعل اي وثيقة نشطة باستدعاء الطريقة كما
: نشط نستخدم التعبير الاتي MyDoc.doc مثلاً لجعل المستند .

<code>WrdApp.Documents("MyDoc.doc").Activate()</code>

. التابع للمستند Index يمكن ان نستخدم رقم المستند او ال او
من تحديد احدى الوثائق المفتوحة بعد Documents Collection المجموعة كما تمكنا
لجملة الاتية تنفيذ
Word الكائن فتصبح هذه الوثيقة هي الوثيقة النشطة عن طريق MyDoc.doc
: باستخدام الكود الاتي

<code>Me.Text = WrdApp.ActiveDocument.Words(1).Text</code>
--

التابعة للكائن Text الكود السابق نسند النص المكتوب في المستند الى الخاصية في
Form 1 مثلاً .

ملاحظة :

ومن ثم نقوم باستخدام الكود السابق Word صفحة ارجو منك ان لا تقوم بكتابة او نسخ
واذا دعت الحاجة يمكن ان تستبدل الكود السابق بان ... فعندها ستجد ما لا يرضيك
.. ومن ثم نقوم باسناد النص المراد الي هذه الاداة RichBox مثلاً او textbox تضيف

. الكود السابق في الاول والاخير هو توضيح المستند النشط فقط الغرض من

Printing and Saving Documents :

من داخل برنامجك موضوع ممتع فلطباعة اي مستند Word والحفظ لمستند الطباعة هذا الكائن الذي عرفناه (Wrdapp التابعة للكائن PrintOut باستدعا الطريقة نقوم : او وسائط كالاتي Parameters ولهذه الطريقة عدة (Word سابقاً وهو من النوع

Printout background, append, range, outputfilename, from, to, item, copies, pages, pageType, PrintToFile, Collate, filename, ActivePrinterMacGX, ManualDuplexPrint, PrintZoomColumn, PrintZoomRow, PrintZoomPaperWidth, PrintZoomPaperHeight

!!! وسائط كثيرة صح

لا تكون قد اصبت باليائس لرويتك لهذه البارميترات ,, !!! تحصل ارجو ان البشرى جميع هذه الوسائط والحمد لله اختيارية ,, اي ان لك مطلق الحرية في اليك تمرير هذه الوسائط الى الطريقة او عدم تمريرها .
مثلاً الوسيط الاول وهو , Print DialogBox الكائن فهذه الوسائط تقابل وسائط .. Object مع انه من النوع False ,Tru احدى القيمتين يمكن ان يحمل Background البارميتر؟؟ طيب ما الغرض من هذا نجعل قيمة هذه الغرض منه هو عندما نريد ان تكون عملية الطباعة في الخلفية True الوسيط .

الوسائط نهمل اغلب هذه الوسائط ونمرر عند استدعا طرق ولهذه الطرق عدد كبير من : كما في الحالة التالية الذي نحتاج اليه Parameter قيم الى ال

`WrdApp.ActiveDocument.PrintOut(From:=2, To:=3)`

يتم طباعة صفحات المستند النشط من الصفحة 2 الى الصفحة 3 فلاحظ هنا اننا هنا . اهملنا باقي الوسائط واستخدمنا فقط ما نريد او ما يهمنا

الحفظ او حفظ مستند وهنا كما ذكرنا سابقاً يمكن ان نستخدم الطريقة ناتى الى : كالاتي Document التابعة للكائن SaveAs : وهي ولهذه الطريقة ايضاً العديد من الوسائط

SaveAs FileName, FileFormat, LockComments, Password, AddToRecentFiles, WritePassword, ReadOnlyRecommended, EmbedTrueTypeFonts, SaveNativePictureFormat, SaveFormsData, SaveAsOCLEtter

Save As Dialog , الوسائط او تقابل الكائن وهذه الطريقة لها ايضاً نفس والتي لا يوجد لها وسائط Save تستخدم الطريقة واذا كان المستند محفوظ يمكن ان . لتمريرها الى هذه الطريقة

: وايضاً لحفظ مستند او المستند النشط باسم مختلف نستخدم الكودالاتي

```
WrdApp.ActiveDocument.SaveAs("c:\Jihad Documents")
```

Creating and Printing a DOC File : مثال عن وهذا

: كود البرنامج

```
Dim WordApp As New Word.Application()

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click

    Dim thisDoc As New Word.Document()
    thisDoc = WordApp.Documents.Add

    With thisDoc

        .Range.InsertAfter("Word طباعة مستند")
        .Paragraphs.Item(1).Range.Font.Bold = True
        .Paragraphs.Item(1).Range.Font.Size = 14
        .Range.InsertParagraphAfter()
        .Paragraphs.Item(2).Range.Font.Bold = False
        .Paragraphs.Item(2).Range.Font.Size = 12
        .Range.InsertAfter("هذا اول سطر يطبع في المستند")
        .Range.InsertParagraphAfter()
        .Range.InsertAfter("وهذا ثاني سطر سيطلع في المستند")

        Try
            .PrintOut(True, True)

        Catch exc As Exception
            MsgBox(exc.Message)
        End Try

        .Close(Word.WdSaveOptions.wdDoNotSaveChanges)

    End With

    WordApp.Quit()

End Sub
```

ان الكود واضح وقد تم شرح خطواته سابقاً وهنا لن اكرر شرح كل الاكواد اعتقد
: ولكن ساوضح بعض الاكواد الجديدة او بعض الطرق الجديدة مثل , السابقة

```
.Paragraphs.Item(1).Range.Font.Bold = True
```

للمجموعة التابعة Item نوع الخط للفقرة السابقة عن طريقة الطريقة تحديد ورقم الفقرة هنا هو Paragraphs 1 : والسطر الاتي

Paragraphs.Item(1).Range.Font.Size = 14

تحديد حجم الخط.

التابعة PrintOut السابق هو الطريقة اهم ما في المثال السابق او الغرض من المثال المستند الحالي , ومن ناحية الوسيطين والتي تستخدم لبطاعة Document للكائن background الوسيط الى الممررين الى هذه الطريقة فالاول يقابل Append . والوسيط الثاني يمرر الى الوسيط

Manipulating Text : التعامل مع النص او

هما Word الكائنين المهمين في التعامل مع النص او النصوص في كما ذكرنا سابقاً ان اي قطعة نص متجاورة او يمثل Selection و الكائن . Selection Objects وال Range ال او فقرة , كيف يعني ؟؟ متلاصقة او متقاربة اي انه يمكن ان يكون اكثر من سطر

فاننا بهذه الطريقة نتعامل مع Word اوضح عندما تريد ان تحدد نص معين في مستند او اكثر من سطر وهكذا ,, باستخدام فانت قد تحدد سطر او كلمة , Selection الكائن من قبل ؟؟؟ اي ان هذا لكائن Word الماوس فهل جربت ان تحدد نص في مستند VB.net يمثل مجموعة في .

هو ايضاً يعتبر مجموعو كما ذكرنا سابقاً ومجموعة من الرموز والكلمات او range الكائن الفقرات .

؟؟ سوال يطرح نفسه ما لفرق بين الكائنين السابقين

في Selction الفرق بين الكائنين هو انك تستطيع ان تعرف كائن واحد فقط من النوع Range المستند , بينما تستطيع ان تعرف العديد من الكائنات من الكائن

!! الحالي بدون التأثير على التحديد Range نستطيع ان نتعامل مع الكائن

انه يمكننا من Dcoument والذي يعتبر طريقة في الكائن العام Range من فوائد الكائن مرير بارميترين الى هذه انتزاع او استخراج نص معين من مستند وذلك عن طريق الوسيطين يمثلان ال وهذين البارميترين (Range الطريقة) (اقصد بالطريقة هي : لهذه الطريقة كالاتي اي البداية والنهاية للنص , الشكل العام Positions

thisDoc.Range(StartEnd)

Start هنا ان اردنا ان نستخرج نص من بداية المستند فاننا نسند الى الوسيط طبعاً : القيمة 0 اي اول حرف فمثلاً لا ستخراج اول عشرة حروف نستخدم الكود الاتي

```
thisDoc.Range(0,10)
```

ملاحظة :

سنأخذ مثال عن استخدام ان لم تفهم شي من الكود السابق فلا تخف لا نأخذ هذا الكائن وعن التعامل معه ومع بعض الطرق الخاصة فيه .

: او يمكن ان يكون الكود السابق كالآتي

: كما يلي Range النوع نعرف متغير من

```
Dim MyRange As Word.Range
```

نسند الكود السابق الى هذا المتغير هكذا ومن ثم

```
MyRange = thisDoc.Range(0,10)
```

!!! لحظة لكن

thisDoc مثالنا هذا هو والذي في Document يجب ان تعلم اننا يجب ان نضع الكائن : يجب ان يشير الى الكائن الحالي كما يلي

:

```
Dim thisDoc As Word.Document = WordApp.Documents.Item(1)
```

مع اي مستند يتعامل Document من الكلام السابق اننا يجب ان نوضح للكائن اقصد يوجد اكثر من مستند , او المستند الذي نعمل عليه او بمعنى اخر فقد . او المستند الحالي ActiveDocument : الاتي ليصبح ما رايت لو اقول لك اننا يمكن ايضاً تغيير الكود

```
Dim thisDoc As Word.Document = WordApp.ActiveDocument
```

فما رأيك ???

كاتب الموضوع : جهاد العريفي

الموضوع الثالث :

هذه اكواد تفيدك:

```
Imports Microsoft.Office.Interop
Private axWord As Word.Application
```

ليبدء وورد نستخدم الكود التالي
كود

```
axWord = New Word.Application
axWord.Visible = True
```

و لإنهاء وورد نستخدم الكود التالي
كود

```
axWord.Quit()
```

لإنشاء ملف جديدة ابدأ وورد أولاً ثم استخدم الأمر Application.Documents.Add
كود

```
Dim axDoc As Word.Document
axDoc = axWord.Documents.Add
```

لفتح ملف موجودة
كود

```
Dim axDoc As Word.Document
axDoc = axWord.Documents.Open("c:\MyDocument.docx")
```

لحفظ ملف

كود

```
axDoc.SaveAs("C:\MyDocument.docx")
```

الحصول على نص الفقرة الأولى
كود

```
Dim strText As String
strText = axDoc.Paragraphs(1).Range.Text
```

ولتغيير نص الفقرة نستخدم
كود

```
axDoc.Paragraphs(1).Range.Text = "Hello from Visual Basic 2005"
```

بإدراج نص في آخر الوثيقة وفي بدايتها ثم يبحث عن كلمة ويدخل نص مباشرة بعدها
كود

```
'Activate the document first
axDoc.Activate()
```

```
'Move to the end and add text
axWord.Selection.EndKey(Word.WdUnits.wdStory)
```

```

axWord.Selection.TypeText("This is the end")
'Move to the beginning and add text
axWord.Selection.HomeKey(Word.WdUnits.wdStory)
axWord.Selection.TypeText("This is the beginning")
axWord.Selection.Find.ClearFormatting()
'Locate Foo, then add text
'following it
axWord.Selection.Find.ClearFormatting()
axWord.Selection.Find.Text = "Foo"
axWord.Selection.Find.Execute()
axWord.Selection.MoveRight(Word.WdUnits.wdCharacter, 1)
axWord.Selection.TypeText("This is Foo")

```

الكود التالي يبين طريقة إدراج جدول بقياس 5 × 5 وإضافة تنسيق للجدول ثم ضبط نص في خلية محددة في ذلك الجدول

```

كود
Dim axTable As Word.Table
axTable = axDoc.Tables.Add(axWord.Selection.Range, 5, 5)
axTable.Style = "Table Grid 8"
axTable.Cell(3, 3).Range.Text = "Hello World"

```

الطباعة
طباعة ملف نستخدم الطريقة PrintOut

كود

```

axDoc.PrintOut()

```

التأكد من ان الوثيقة مفتوحة
بما أن المستخدم يمكنه إغلاق الوثيقة أو الورد في أي وقت رغم أن برنامجك مازال مرتبطا به يمكنك استخدام الطريقة التالية للتأكد من ان الوثيقة مازالت مفتوحة

كود

```

Dim strName As String
Dim blnIsAvailable As Boolean
Try
strName = axDoc.Name
blnIsAvailable = True
Catch ex As Exception
blnIsAvailable = False
End Try
MsgBox("Document is available: " & blnIsAvailable)
يمكنك استخدام الطريقة التالية لإدراج بيانات من جدول في قاعدة بيانات في جدول في وثيقة وورد
كود
Function AddDataTable(ByVal tbl As DataTable) As Boolean
Dim nRowCount, nColCount, nRow, nCol As Integer
Dim axTable As Word.Table

```

```

nRowCount = tbl.Rows.Count
nColCount = tbl.Columns.Count
axWord.Selection.EndKey(wdStory)
axTable = axDoc.Tables.Add(axWord.Selection.Range, nRowCount + 1,
nColCount)
axTable.Style = "Table Grid 8"
For nCol = 1 To nColCount
axTable.Cell(1, nCol).Range.Text = tbl.Columns.Item(nCol - 1).Caption
Next
For nCol = 1 To nColCount
For nRow = 1 To nRowCount
axTable.Cell(nRow + 1, nCol).Range.Text = tbl.Rows(nRow - 1).Item(nCol -
1)
Next
Next
Return True
End Function

```

```

' To use this function
tbl = Me.CustomersDataSet.Tables(0)
WordDoc.AddDataTable(tbl)

```

كاتب الموضوع : !!! BorIS !!!

الموضوع الرابع :

كيفية استخدام التصحيح التلقائي للورد ضمن برنامجك

يمثل هذا المتغير برنامج الورد

```
public WordApp As New Word.Application()  
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button1.Click
```

يمثل هذا المتغير النص المكتوب

```
Dim DRange As Word.Range  
Me.Text = "Starting Word ..."
```

إضافة مستند جديد

```
WordApp.Documents.Add()  
Me.Text = "Checking words..."
```

تعرف النص المكتوب الحالي و الموجود في المستند الذي اصفناه

```
DRange = WordApp.ActiveDocument.Range
```

ادخال النص المكتوب في مربع النص الى المستند

```
DRange.InsertAfter(TextBox1.Text)
```

يمثل هذا المتغير الاخطاء الاملائية

```
Dim SpellCollection As Word.ProofreadingErrors
```

تعريف الاخطاء الاملائية المكتوبة في النص

```
SpellCollection = DRange.SpellingErrors
```

إذا كانت هناك اخطاء املائية في النص

```
If SpellCollection.Count > 0 Then  
Dim iword As Integer  
Dim newWord As String
```

حلقة تكرارية تضيف جميع الاخطاء الاملائية الى اللسته

```
For iword = 1 To SpellCollection.Count  
newWord = SpellCollection.Item(iword).Text  
ListBox1.Items.Add(newWord)  
Next  
End If  
Me.Text = "Word spelling Demo"  
End Sub
```

كاتب الموضوع : [egyption coder](#)

1. Development Tools Used

- Microsoft Visual Studio 2005
- Microsoft Word 2003
- Programming Language: VB

2. Word Automation using VB

Word Automation through VB is all about programmatically generating the Word Document using VB code. Working on Word is considered to be straightforward, but doing the same programmatically gets a little intricate. Word automation almost completely involves working with objects and reference types. Almost all of the tasks which we perform on word 2003 can be done programmatically using VB or C#. Tasks like Inserting Table of Contents, Linking documents, Mail Merge, Inserting Documents, Embedding documents, inserting pictures, watermark... etc can all be done programmatically.

3. Setting Up Work Environment

Starting off, the first step is to include the Word dll's to the Solution. This can be done by right clicking the Reference Folder in the Solution explorer of the project and select Add Reference.

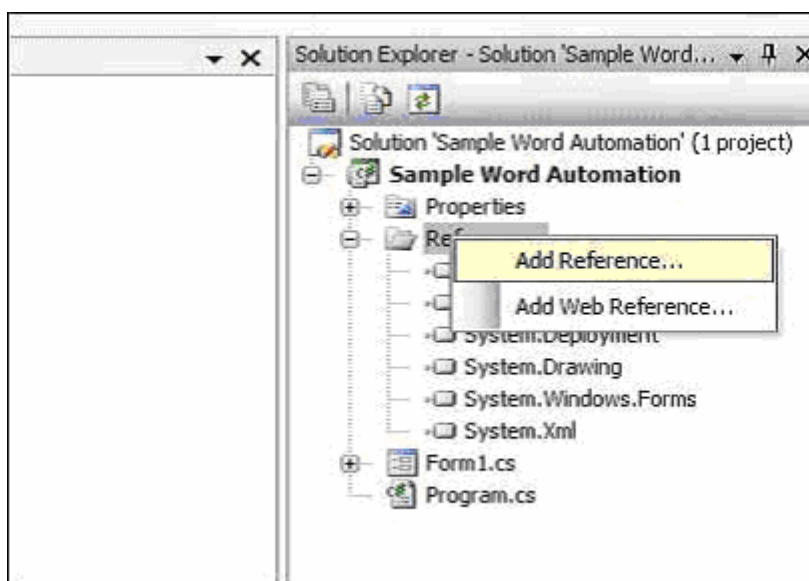


Figure 1.

Browse Through the available COM objects and Select Microsoft Office 11.0 Object Library & Microsoft Word 11.0 Object Library. This DLL has all the methods which we do to perform the automation.

Note: This dll would be present only if Microsoft Office is installed on the Machine.

Also include "

`Imports Microsoft.Office`

" in the Namespaces used.

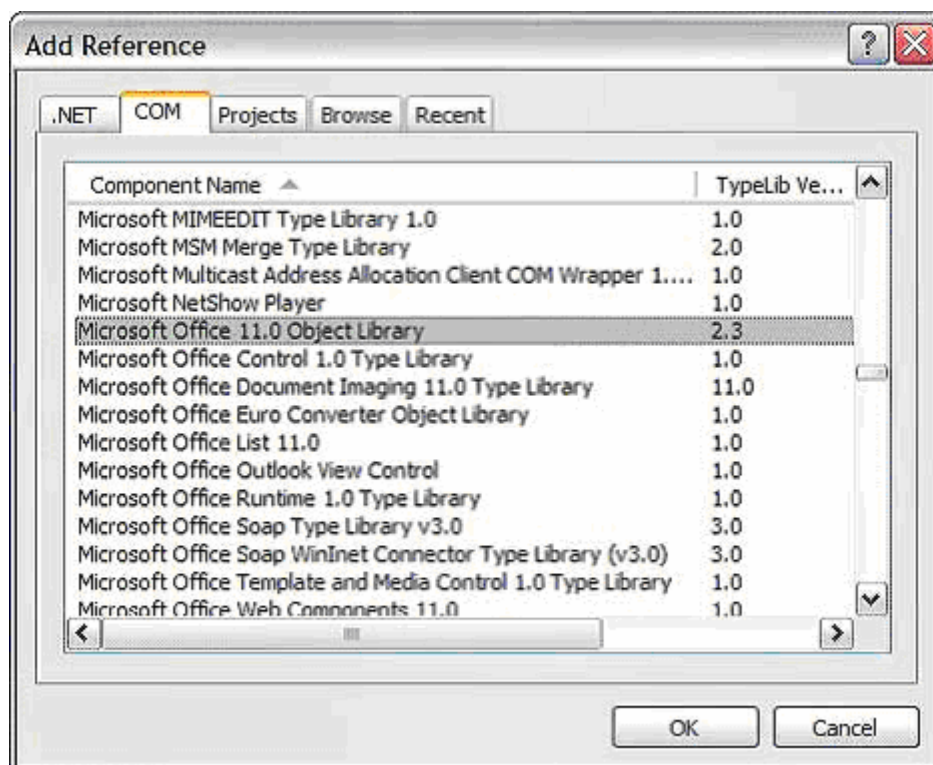


Figure 2.

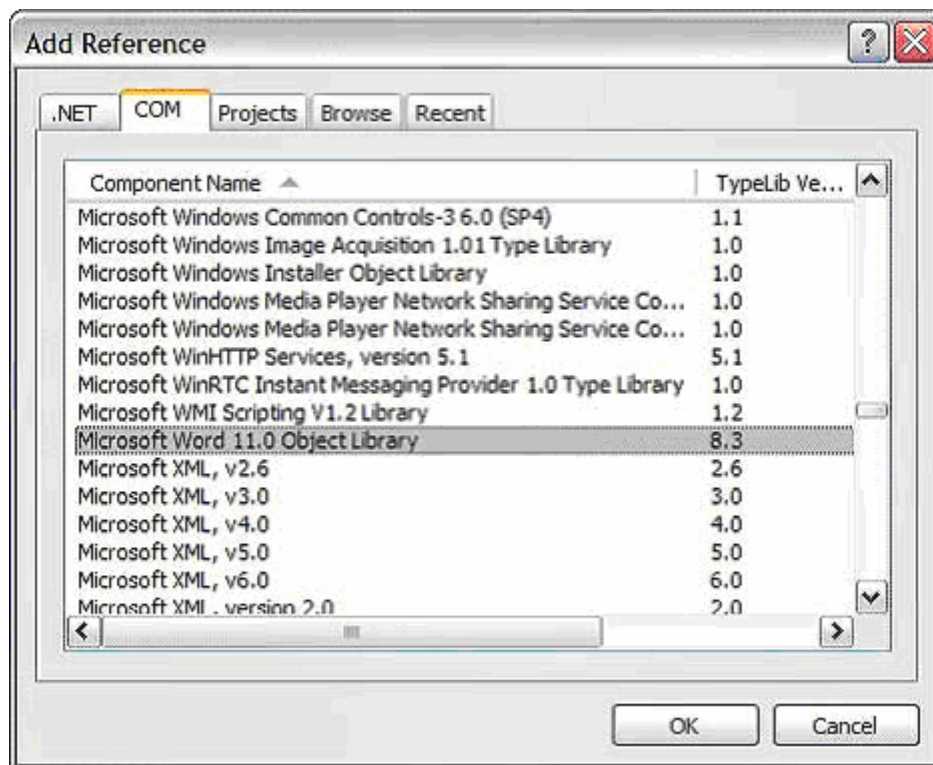


Figure 3.

4. Objects Used in Automation

All the methods used Word automation is derived either from Word.Application or Word.Document class.

Let's consider that we want to create a document using the Word Application, we might end up doing the following steps,

1. Open Word Application. (Opening Word Application creates a new document by default, but in Automation, we need to manually add a document)
2. Add a New document.
3. Edit the document.
4. Save it.

The same steps need to be done programmatically. The Word.Application and Word.Document are used to Open Word and add a new Document to it.

4.1 Word.Application

This represents in Word Application without any new document loaded in it. This is like the base class which is needed to create a new document. Creating a new instance of Word.Application can be visualized as below.

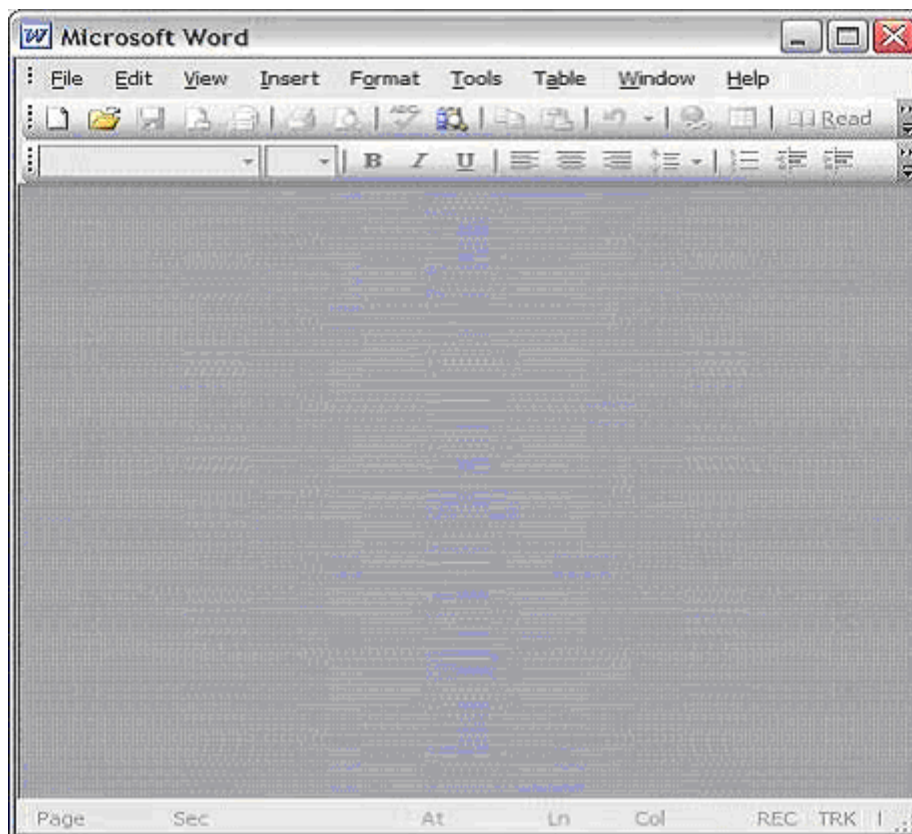


Figure 4.

4.2 Word.Document

If we need to add a new document file, first we have to create an instance of the Word.Document object and then add it to the Word.Application.

```
'OBJECT OF MISSING "NULL VALUE"
Dim oMissing As Object = System.Reflection.Missing.Value()
'OBJECTS OF FALSE AND TRUE
Dim oTrue As Object = True
Dim oFalse As Object = False

'CREATING OBJECTS OF WORD AND DOCUMENT
Dim oWord As New Word.Application()
Dim oWordDoc As New Word.Document()

'MAKING THE APPLICATION VISIBLE
oWord.Visible = True

'ADDING A NEW DOCUMENT TO THE APPLICATION
oWordDoc = oWord.Documents.Add(oMissing, oMissing, oMissing,
oMissing)
```

This triggers the following operation in the Word Application

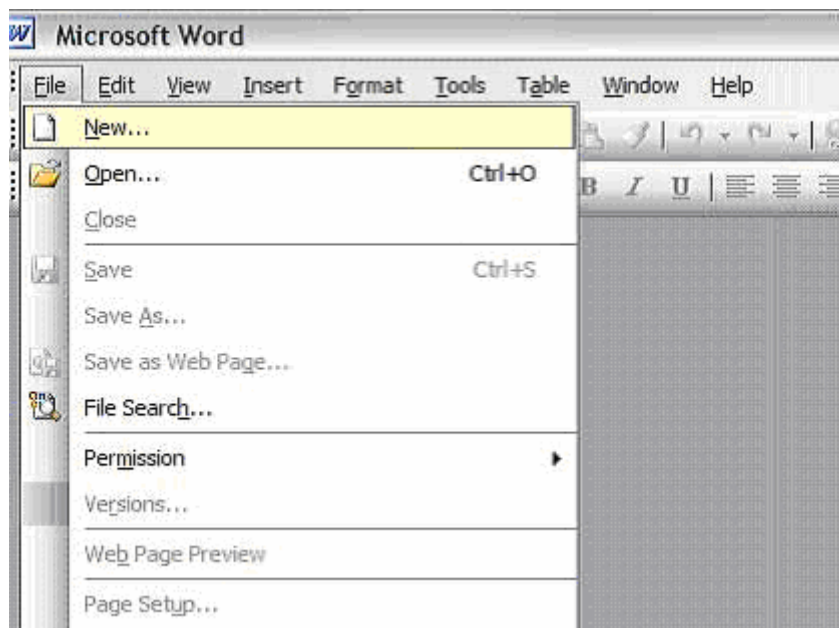


Figure 5.

Approaches to Perform Automation

1. We can either have a base template (.dot) file and open the base template file and work on it.
2. We can otherwise build a word document from scratch.

4.3 Standard Input Parameters

Most of the methods have input parameters which are of reference type, and the values are mostly true, false or missing (null). In automation it makes sense as to why most of the input parameters are of reference types; it might be because of the fact that most of the methods a multitude of input parameters (many have more than 10 input parameters) and their value is going to be either true, false or missing in most of the cases. So instead of supplying the same input parameter ten times, we can make all the input parameters point to the location same single variable in them memory.

4.3.1 Range Object

While we work on Word Application, if we want to type some text in the 11th line, then we manually take the cursor and click it on the required line and then start typing. In order to do the same task, we use the Range variable in VB. The range variable of the Word.Document object represents the location of the cursor on the current document.

There are many possible ways to point to a specific location on a document. I had extensively used the Bookmarks locators as I work on Automation using a base template. In this approach, we insert Bookmarks on the base template and we programmatically locate those Bookmarks, set the range on them and insert text or documents at that specific location. There are also many other possible ways to set the range.

'SETTING THE RANGE ON THE BOOKMARK

```
Dim oBookmarkName As Object =  
"My_Inserted_Bookmark_On_Template"  
Dim wrdRange As Word.Range =  
oWordDoc.Bookmarks.get_Item(oBookmarkName).Range.[Select]()
```

4.3.2 Selection Object

While working on word, we select a range of text by clicking and dragging the mouse pointer across contents in the document to select it. The contents can be text, formatted text, tables or any other item in the document. We programmatically represent the same by using the Selection Object derived from Word.Selection. In the previous range example, we locate a bookmark and set the range on that specific bookmark and we select it. Now the selection object represents that specific location. It's like placing the cursor on that specific bookmark location on the document. The selection across text can be done by selecting a range of text in between two ranges. Then the selected range can be copied, deleted or formatted.

4.3.3 Selecting Between Bookmarks

'BOOK MARK FOR START OF SELECTION

```
Dim oBookmarkStart As Object = "BookMark__Start"  
Dim oRngoBookMarkStart As Object =  
oWordDoc.Bookmarks.get_Item(oBookmarkDesignInfoStart).Range  
.Start
```

'BOOK MARK FOR END OF SELECTION

```
Dim oBookmarkEnd As Object = "BookMark__End"  
Dim oRngoBookMarkEnd As Object =  
oWordDoc.Bookmarks.get_Item(oBookmarkDesignInfoEnd).Range.  
Start
```

'SETTING THE RANGE ON THE BOOKMARK BETWEEN TWO BOOKMARKS

```
Dim rngBKMarkSelection As Word.Range =
```

```
oWordDoc.Range(oRngoBookMarkStart, oRngoBookMarkEnd)
```

'SELECTING THE TEXT

```
rngBKMarkSelection.[Select]()
```

```
rngBKMarkSelection.Delete(oMissing, oMissing)
```

5. Automation using a Base Template

The base template file method is preferable as it gives us much more flexibility in performing the automation and it comes very handy for performing Mail Merge.

In the base template method, when we call the Documents.Add method of the Application object, we give the path of the .dot file.

'THE LOCATION OF THE TEMPLATE FILE ON THE MACHINE

```
Dim oTemplatePath As Object = "C:\Program  
Files\MyTemplate.dot"
```

'ADDING A NEW DOCUMENT FROM A TEMPLATE

```
oWordDoc = oWord.Documents.Add(oTemplatePath, oMissing,  
oMissing, oMissing)
```

Now .dot file is opened and when we save the generated document, we save it as a new file.

6. Mail Merge

Mail merge is a useful tool in scenarios where we want to randomly generate alike documents where just a few fields change. For instance in a pay slip which has a base template and just the employee name, number and pay details needs to change for each employee. Now we can have a base template which is a word file saved as Document Template file.

In the .dot file, insert a Mail Merge Field manually by placing the cursor in the required position and Insert -> Field, and in Field Names, select "MergeField", now the Mail merged field would be represented by <<FieldName>>. The template can be like

Contact Information

For further information and discussions, please contact:

Name: <<CIFLName>>

Address: <<CIAddress>>

Phone: <<CIPhW>> (Work)

<<CIPhM>> (Cell)
Fax: <<CIFax>>
Email: <<CIMail>>

Now for programmatically replacing the Mail Merge fields using the code, the document by default has many fields in it. But the user entered fields comes with a prefix and suffix which can be used as an identifier to replace the fields.

```
'OBJECT OF MISSING "NULL VALUE"
Dim oMissing As Object = System.Reflection.Missing.Value()

'OBJECTS OF FALSE AND TRUE
Dim oTrue As Object = True
Dim oFalse As Object = False

'CREATING OBJECTS OF WORD AND DOCUMENT
Dim oWord As New Word.Application()
Dim oWordDoc As New Word.Document()

'SETTING THE VISIBILITY TO TRUE
oWord.Visible = True

'THE LOCATION OF THE TEMPLATE FILE ON THE MACHINE
Dim oTemplatePath As Object = "C:\Program
Files\MyTemplate.dot"

'ADDING A NEW DOCUMENT FROM A TEMPLATE
oWordDoc = oWord.Documents.Add(oTemplatePath, oMissing,
oMissing, oMissing)

For Each myMergeField As Word.Field In oWordDoc.Fields
    iTotalFields += 1
    Dim rngFieldCode As Word.Range = myMergeField.Code
    Dim fieldText As String = rngFieldCode.Text

' ONLY GETTING THE MAILMERGE FIELDS
    If fieldText.StartsWith(" MERGEFIELD") Then
' THE TEXT COMES IN THE FORMAT OF
' MERGEFIELD MyFieldName \ \ * MERGEFORMAT
' THIS HAS TO BE EDITED TO GET ONLY THE FIELDNAME
    "MyFieldName"
    Dim endMerge As Int32 = fieldText.IndexOf("\")
    Dim fieldNameLength As Int32 = fieldText.Length - endMerge
    Dim fieldName As String = fieldText.Substring(11, endMerge - 11)

' GIVES THE FIELDNAMES AS THE USER HAD ENTERED IN .dot FILE
    fieldName = fieldName.Trim()
```

```

' **** FIELD REPLACEMENT IMPLEMENTATION GOES HERE **** //
' THE PROGRAMMER CAN HAVE HIS OWN IMPLEMENTATIONS HERE
  If fieldName = "MyField" Then
    myMergeField.[Select]()
    oWord.Selection.TypeText("This Text Replaces the Field in
the Template")
  End If
End If
Next

```

There is one other method for replacing the [Merge Fields which is mentioned in msdn](#), which uses a rather memory hungry approach. In that method a separate document is opened and it is inserted with a table which has first row as the Mail Merge Field Name and the second row as the replacement value, then the value from the table is matched with that of the original document and replacement occurs and the second document is purged.

7. Embedding a Document

Embedding a document is done through the application by Insert-> Object-> Create from file-> Select the File-> Display as Icon. This embeds the file in the selected location as an icon and the user can double click on the icon to open the file. The same can be done through automation.

The range supposed to set at the required place and the same has to be selected (range can be set by any of the means mentioned above). Now with the selection, the file can be embedded.

```

'ICON LABEL CAN BE THE NAME OF THE FILE,
'ITS THE NAME DISPLAYED BESIDES THE EMBEDDED DOCUMENT
Dim oIconLabel As Object = "File Name"

'INCASE WE NEED THE EMBEDDED DOCUMENT TO BE DISPLAYED AS
A SPECIFIC ICON,
'WE NEED TO SPECIFY THE LOCATION OF THE ICON FILE
'ELSE SET IT TO oMissing VALUE
Dim oIconFileName As Object = "C:\Document and
Settings\IconFile.ico"

'THE BOOKMARK WHERE THE FILE NEEDS TO BE EMBEDDED
Dim oBookMark As Object = "My_Custom_BookMark"
'THE LOCATION OF THE FILE
Dim oFileDesignInfo As Object = "C:\Document and
Settings\somefile.doc"

'OTHER VARIABLES
Dim oClassType As Object = "Word.Document.8"
Dim oTrue As Object = True
Dim oFalse As Object = False
Dim oMissing As Object = System.Reflection.Missing.Value

'METHOD TO EMBED THE DOCUMENT
oWordDoc.Bookmarks.get_Item(oBookMark).Range.InlineShapes.A
ddOLEObject(oClassType, oFileDesignInfo, oFalse, oTrue,
oIconFileName, oMissing, oIconLabel, oMissing)

```

8. Inserting a Document File

Contents of a Word documents can also be inserted into the current document from the application by doing the following.

Insert -> File -> Select the File. This extracts the contents from the selected file and inserts it into the current document.

In automation, we need to follow a similar approach by placing the range at the required point and selecting it and then inserting the file.

```

'THE LOCATION OF THE FILE
Dim oFilePath As String = "C:\Document and Settings\somefile.doc"
oWordDoc.Bookmarks.get_Item(oBookMark).Range.InsertFile(oFile
Path, oMissing, oFalse, oFalse, oFalse)

```

9. Including Water Marks/Pictures in the Document Background

Including watermarks is one other important feature for any official documents as the watermark may have the company's logo, draft logo or any other picture/text. This is useful when we want a picture or some text to be present throughout the document in the background.

We insert a watermark in the application by performing the following tasks.

Format -> Background -> Printed Watermarks

The same can also be done programmatically; moreover as we manually define the values like the angle of tilt and actual location of the watermark, we have more flexibility in defining the exact location of the watermark.

9.1 Embedding Pictures in Document Header

```
'EMBEDDING LOGOS IN THE DOCUMENT
'SETTING FOCUES ON THE PAGE HEADER TO EMBED THE
WATERMARK
oWord.ActiveWindow.ActivePane.View.SeekView =
Word.WdSeekView.wdSeekCurrentPageHeader

'THE LOGO IS ASSIGNED TO A SHAPE OBJECT SO THAT WE CAN USE
ALL THE
'SHAPE FORMATTING OPTIONS PRESENT FOR THE SHAPE OBJECT
Dim logoCustom As Word.Shape = Nothing

'THE PATH OF THE LOGO FILE TO BE EMBEDDED IN THE HEADER
Dim logoPath As String = "C:\Document and Settings\MyLogo.jpg"
logoCustom =
oWord.Selection.HeaderFooter.Shapes.AddPicture(logoPath, oFalse,
oTrue, oMissing, oMissing, oMissing, oMissing, oMissing)

logoCustom.[Select](oMissing)
logoCustom.Name = "CustomLogo"
logoCustom.Left = CSng(Word.WdShapePosition.wdShapeLeft)

'SETTING FOCUES BACK TO DOCUMENT
oWord.ActiveWindow.ActivePane.View.SeekView =
Word.WdSeekView.wdSeekMainDocument
```

9.2 Inserting Text in the Centre of the Document as Water Mark

```
'THE LOGO IS ASSIGNED TO A SHAPE OBJECT SO THAT WE CAN USE
ALL THE
'SHAPE FORMATTING OPTIONS PRESENT FOR THE SHAPE OBJECT
Dim logoWatermark As Word.Shape = Nothing

'INCLUDING THE TEXT WATER MARK TO THE DOCUMENT
logoWatermark =
oWord.Selection.HeaderFooter.Shapes.AddTextEffect(Microsoft.Office
e.Core.MsoPresetTextEffect.msoTextEffect1, "Enter The Text Here",
"Arial", CSng(60), Microsoft.Office.Core.MsoTriState.msoTrue,
Microsoft.Office.Core.MsoTriState.msoFalse, 0, 0, oMissing)
logoWatermark.[Select](oMissing)
logoWatermark.Fill.Visible =
Microsoft.Office.Core.MsoTriState.msoTrue
logoWatermark.Line.Visible =
Microsoft.Office.Core.MsoTriState.msoFalse
logoWatermark.Fill.Solid()
logoWatermark.Fill.ForeColor.RGB =
DirectCast(Word.WdColor.wdColorGray30, Int32)
logoWatermark.RelativeHorizontalPosition =
Word.WdRelativeHorizontalPosition.wdRelativeHorizontalPositionM
argin
logoWatermark.RelativeVerticalPosition =
Word.WdRelativeVerticalPosition.wdRelativeVerticalPositionMargin
logoWatermark.Left =
CSng(Word.WdShapePosition.wdShapeCenter)
logoWatermark.Top =
CSng(Word.WdShapePosition.wdShapeCenter)
logoWatermark.Height = oWord.InchesToPoints(2.4F)
logoWatermark.Width = oWord.InchesToPoints(6F)

'SETTING FOCUES BACK TO DOCUMENT
oWord.ActiveWindow.ActivePane.View.SeekView =
Word.WdSeekView.wdSeekMainDocument
```

9.3 Inserting Text in the Centre of Page, and rotating it by 90 Degrees

```
'INSERTING TEXT IN THE CENTRE RIGHT, TILTED AT 90 DEGREES
Dim midRightText As Word.Shape
midRightText =
oWord.Selection.HeaderFooter.Shapes.AddTextEffect(Microsoft.Office
e.Core.MsoPresetTextEffect.msoTextEffect1, "Text Goes Here",
"Arial", CSng(10), Microsoft.Office.Core.MsoTriState.msoTrue,
Microsoft.Office.Core.MsoTriState.msoFalse, _
0, 0, oMissing)

'FORMATTING THE SECURITY CLASSIFICATION TEXT
midRightText.[Select](oMissing)
midRightText.Name = "PowerPlusWaterMarkObject2"
midRightText.Fill.Visible =
Microsoft.Office.Core.MsoTriState.msoTrue
midRightText.Line.Visible =
Microsoft.Office.Core.MsoTriState.msoFalse
midRightText.Fill.Solid()
midRightText.Fill.ForeColor.RGB =
CInt(Word.WdColor.wdColorGray375)

'MAKING THE TEXT VERTICAL & ALIGNING
midRightText.Rotation = CSng(90)
midRightText.RelativeHorizontalPosition =
Word.WdRelativeHorizontalPosition.wdRelativeHorizontalPositionM
argin
midRightText.RelativeVerticalPosition =
Word.WdRelativeVerticalPosition.wdRelativeVerticalPositionMargin
midRightText.Top = CSng(Word.WdShapePosition.wdShapeCenter)
midRightText.Left = CSng(480)
```

10. Including Page Numbers in Page Footer

Including auto-generated page numbers in the Footer is yet another useful feature which can be simulated in the code.

```
'SETTING THE FOCUES ON THE PAGE FOOTER
oWord.ActiveWindow.ActivePane.View.SeekView =
Word.WdSeekView.wdSeekCurrentPageFooter

'ENTERING A PARAGRAPH BREAK "ENTER"
oWord.Selection.TypeParagraph()

Dim docNumber As String = "1"
Dim revisionNumber As String = "0"

'INSERTING THE PAGE NUMBERS CENTRALLY ALIGNED IN THE
PAGE FOOTER
oWord.Selection.Paragraphs.Alignment =
Word.WdParagraphAlignment.wdAlignParagraphLeft
oWord.ActiveWindow.Selection.Font.Name = "Arial"
oWord.ActiveWindow.Selection.Font.Size = 8
oWord.ActiveWindow.Selection.TypeText("Document #: " +
docNumber + " - Revision #: " + revisionNumber)

'INSERTING TAB CHARACTERS
oWord.ActiveWindow.Selection.TypeText("'" & Chr(9) & "'")
oWord.ActiveWindow.Selection.TypeText("'" & Chr(9) & "'")

oWord.ActiveWindow.Selection.TypeText("Page ")
Dim CurrentPage As Object = Word.WdFieldType.wdFieldPage
oWord.ActiveWindow.Selection.Fields.Add(oWord.Selection.Range,
CurrentPage, oMissing, oMissing)
oWord.ActiveWindow.Selection.TypeText(" of ")
Dim TotalPages As Object = Word.WdFieldType.wdFieldNumPages
oWord.ActiveWindow.Selection.Fields.Add(oWord.Selection.Range,
TotalPages, oMissing, oMissing)

'SETTING FOCUES BACK TO DOCUMENT
oWord.ActiveWindow.ActivePane.View.SeekView =
Word.WdSeekView.wdSeekMainDocument
```

11. Basic Text Formatting Options

11.1 Paragraph Break

This is equivalent to hitting the enter button in the document.

```
'ENTERING A PARAGRAPH BREAK "ENTER"  
oWord.Selection.TypeParagraph()
```

11.2 Text Formatting Option

All the text formatting options available in the Word Application can also be replicated through automation.

```
'OTHER COMMONLY USED FORMATTING OPTIONS  
oWord.Selection.Font.Bold = 1  
oWord.Selection.Font.Color = Word.WdColor.wdColorAqua  
oWord.Selection.Font.Italic = 1  
oWord.Selection.Font.Underline =  
Word.WdUnderline.wdUnderlineDashHeavy
```

11.3 Clear Formatting

When the Formatting is applied to a selection, then the same formatting gets carried on to the next lines, in order to clear the formatting, the next line needs to be selected and ClearFormatting() method needs to be called.

```
'CLEARING THE FORMATTING  
oWord.Selection.ClearFormatting()
```

12. Table of Contents

Table of Contents is very handy when it comes to official documents or some technical papers which span across many pages. Table of contents can be inserted and updated on the fly as the document gets built.

For the Table of Contents to get auto generated without any hassles, it is vital that the Headings, Sub-Headings and the Body text have their respective attributes set. When we work on the application, the values get set by themselves, we only need to edit if required. But while programming its mandatory that we set the values in the code in order to prevent any anomalies when the Table of Contents gets updated.

Below is an example of a document which was programmatically generated.

Table of Contents	
Sample Header 2	1
Sample Header 3	1

Sample Header 2

Body Message for "Sample Header 2"

Sample Header 3

Body Message for "Sample Header 3"

Figure 6.

It is apparent that the Header 2 and Header 3 and Body are formatted differently and even in the Table of Contents the Header 2 is slightly offset from the Header 1.

Open the above document and Outlining Tool bar, View -> Toolbars -> Outlining. And on moving the cursor on the Sample Header 2, we can see that the Format is Heading 2 and Outlining level is Level 2.

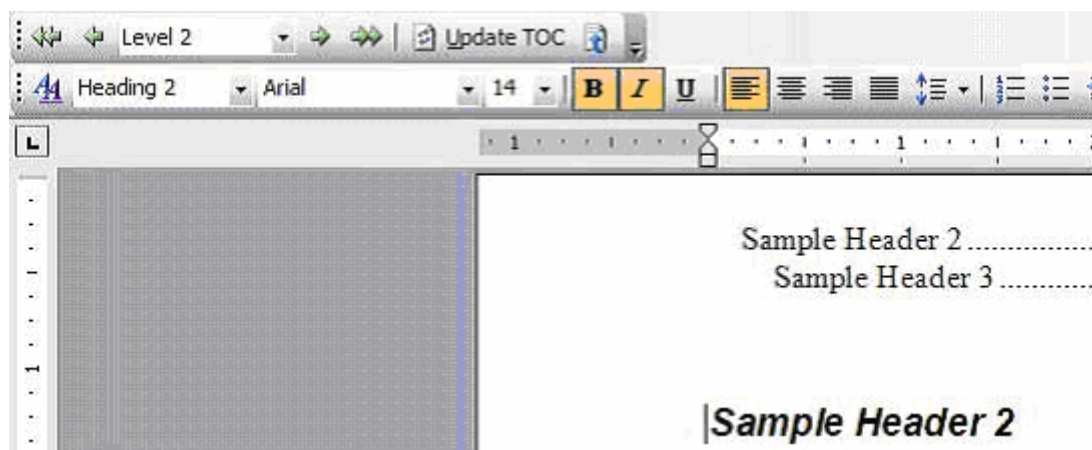


Figure 7.

And for Body, the Format is Normal + Arial, 10 pt and Outlining Level is Body text.

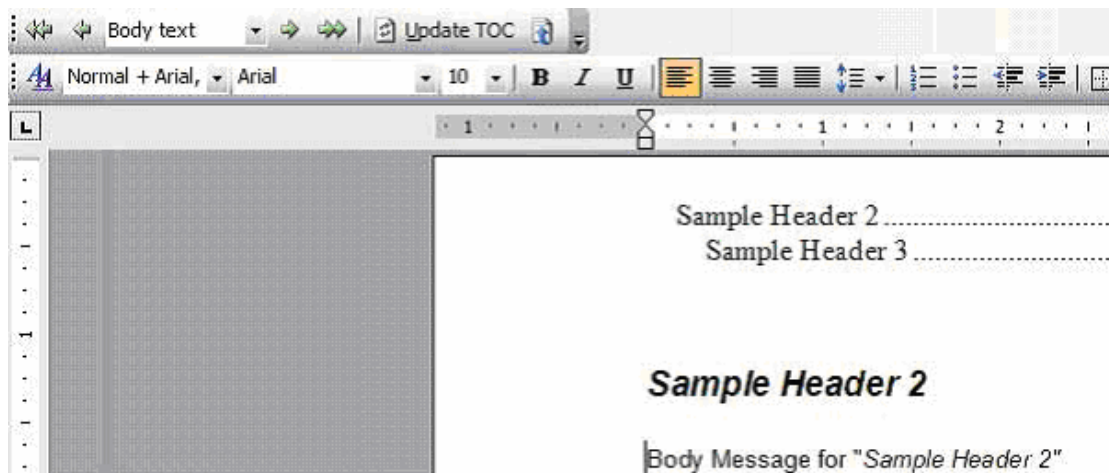


Figure 8.

The same values needs to be set programmatically for the Table of Contents to get generated.

12.1 Section Format

For setting the Format of the Selection, select the entire text (select between bookmarks like mentioned before in Selection section) and set the value

```
'SETTING THE FORMAT TYPE
'SELECT THE CONTENST TO BE FORMATTED AND SET THE VALUE

Dim styleHeading2 As Object = "Heading 2"
Dim styleHeading3 As Object = "Heading 3"

oWord.Selection.Range.set_Style(styleHeading2)
oWord.Selection.Range.set_Style(styleHeading3)
```

12.2 Outline Level

For setting the outline level, select the contents and set it to one of the values mentioned below

```
'SETTING THE OUTLINE LEVEL
'SELECT THE CONTENTS WHOSE OUTLINE LEVEL NEEDS TO BE
CHANGED AND
'SET THE VALUE
```

```
oWord.Selection.Paragraphs.OutlineLevel =
Word.WdOutlineLevel.wdOutlineLevel2
oWord.Selection.Paragraphs.OutlineLevel =
Word.WdOutlineLevel.wdOutlineLevel3
oWord.Selection.Paragraphs.OutlineLevel =
Word.WdOutlineLevel.wdOutlineLevelBodyText
```

12.3 Inserting Table of Contents

Once the Outline Levels & Section Style are set, the Table of Contents can be inserted programmatically and the page numbers gets populated automatically based on the Outline Levels & Section Style set by the user. ([Also refer this MSDN Link](#))

```
' NAME OF THE BOOKMARK IN THE DOCUMENT (.dot Template)
WHERE TABLE OF
' CONTENTS NEEDS TO BE ADDED
Dim oBookmarkTOC As Object = "Bookmark_TOC"

' SETTING THE RANGE AT THE BOOKMARK
Dim rngTOC As Word.Range =
oWordDoc.Bookmarks.get_Item(oBookmarkTOC).Range

' SELECTING THE SET RANGE
rngTOC.[Select]()

' INCLUDING THE TABLE OF CONTENTS
Dim oUpperHeadingLevel As Object = "1"
Dim oLowerHeadingLevel As Object = "3"
Dim oTOCTableID As Object = "TableOfContents"
oWordDoc.TablesOfContents.Add(rngTOC, oTrue,
oUpperHeadingLevel, oLowerHeadingLevel, oMissing, oTOCTableID,
oTrue, oTrue, oMissing, oTrue, oTrue, oTrue)
```

12.4 Updating Table of Contents

Usually the Table of Contents is inserted in the beginning of the document generation and once all the contents are populated, the locations of the Headings and Sub Headings tend to change. If the

Table of Contents is not updated, then its contents points to different pages. To overcome this hassle, the Table of Contents needs to be updated at the end of the Automation.

```
'UPDATING THE TABLE OF CONTENTS
```

```
oWordDoc.TablesOfContents(1).Update()
```

```
'UPDATING THE TABLE OF CONTENTS
```

```
oWordDoc.TablesOfContents(1).UpdatePageNumbers()
```

13. Saving/Closing & Re-Opening the File

13.1 Saving the File

```
'THE LOCATION WHERE THE FILE NEEDS TO BE SAVED
```

```
Dim oSaveAsFile As Object = DirectCast("C:\SampleDoc.doc",  
Object)
```

```
oWordDoc.SaveAs(oSaveAsFile, oMissing, oMissing, oMissing,  
oMissing, oMissing, oMissing, oMissing, oMissing, oMissing,  
oMissing, oMissing, oMissing, oMissing, oMissing, oMissing)
```

13.2 Closing the File

```
'CLOSING THE FILE
```

```
oWordDoc.Close(oFalse, oMissing, oMissing)
```

```
'QUITTING THE APPLICATION
```

```
oWord.Quit(oMissing, oMissing, oMissing)
```

13.3 Re-Opening the File

The Open () method which we use in Word2003 dll might throw an exception if the client have another version of word installed in their machine. If the client has Word 2002, then he has to open a word file only by Open2002 () method. Open () method which comes for Word 2003 might through an exception in Word 2002 environment. And for Word 2000, there is a method called Open2000 () and Open2002 () for Office 2002 and so on. So it is wise to put the Open () in a try-catch block as mentioned below.



Figure 9.

14. Tips for Word Automation to Create New Document (Non-Base Template Approach)

([Refer to this MSDN link](#))

When we proceed to create a New Document without using the Base Template, the most useful entity is the inbuilt Bookmark endofdoc. It would be a build-from-scratch approach where the programmer starts of the automation by inserting his first section of contents, then setting the range to point to the endofdoc Bookmark and selecting it and inserting his contents and again selecting the endofdoc which would be pointing to the end of the document which would now be after the two sections.

NOTE: THIS ARTICLE IS CONVERTED FROM C# TO VB.NET USING A CONVERSION TOOL. ORIGINAL ARTICLE CAN BE FOUND ON C# Corner (<http://www.c-sharpcorner.com/>).

تم بحمد الله
أتمنى منكم الدعاء
جامع المواضيع : هيثم عرفات
Haytham.arafat@yahoo.com
Haytham.arafat@hotmail.com