

أكواد متنوعة ومبتكرة للمبرمجين بالفيجوال بيسك

الجزء الثالث

تأليف وإعداد / عبدالله خضر عبدالله الهوساوي

E-Mail : prog909@yahoo.com

بسم الله الرحمن الرحيم

هذه مجموعة أكواد مهمة وفنية وتسهل الكثير من الأعمال البرمجية في الفيجوال بيسك 6 ، وهي من تصميمي ، أرجو أن ينتفع بها مبرمجينا الأعزاء ، هي هكذا تباعا :



توضيح :

هذه الأداة هي أخت الأداة ليست List ، الفرق بينهما أن أداة الكومبو Combo أداة عرضية رشيقة ولا تأخذ حيزاً كبيراً من مساحة الفورم Form ، بعكس الأداة ليست List الطولية التي يمكن أن تستحوذ على طول الفورم وعرضه ، والأداة كومبو Combo لها لسان أو سهم جانبي بالضغط عليه يفتح كل محتويات الأداة ، وهي أيضاً تعتبر (وعاء بياني) مثل أختها ليست List ، لذا كان لهذه الأداة الأخرى نصيب من الدوال الوظيفية Functions والإجراءات Subs في هذا الجزء الثالث .

((لاتسألني عن معنى كلمة كومبو Combo ☺ !! ، أنا نفسي لا أعرف بالضبط ماذا ترمي إليه ! ، يقال في القاموس أنها تعني (مجموعة) ! ، وأحاول حتى اليوم أن أفهم لماذا يحب المطورون للبرامج في شركة مايكروسوفت إطلاق أسماء غامضة ومعقدة على بعض الأدوات ! ، تستطيع القول ربما : نزوات برمجية أو حب التميز لدرجة الغرابة ☺))



1 - دالة وظيفية تعطي عدد عناصر أداة الكومبو Combo

الإسم الافتراضي لهذه الدالة يكون : **CmbCnt**

حيث يوضع إسم أداة الكومبو بين قوسين بعدها أثناء كتابتها في النص البرمجي

شرح مبسط عن عمل الدالة :

مثلاً إسم أداة الكومبو المطلوب معرفة عدد عناصرها مباشرة هي = **Combo1**

صيغة الكتابة البرمجية تكون هكذا :
Print CmbCnt (Combo1)
أو
N= CmbCnt (Combo1)
فتعطي الدالة هذه النتيجة :
5 أو 10 أو 24 أو 3 أو أي عدد آخر يكون هو عدد عناصر الكومبو .

ملاحظة مهمة : هذا الكود الآتي يوضع في الفورم في منطقة الإعلانات العامة للفورم الـ **Declarations** ، أو في ملف ميديول **Moduel** تابع لمشروع البرنامج .

نص الكود :

```
Function CmbCnt(CombName As ComboBox) As Integer  
CombName.Refresh  
G = CombName.ListCount  
CmbCnt = G  
End Function
```

2 - إجراء لنسخ ملف بيانات متسلسل إلى الكومبو Combo

حيث الإسم الافتراضي لهذا الإجراء يكون : **CopyFC**

حرف **F** إشارة إلى كلمة (ملف **File**) ، وحرف **C** إشارة إلى كلمة (كومبو **Combo**) ، أقصد من الملف إلى الكومبو **F to C**

إنتبه جيداً : يجب أن يُلحق الإسم الافتراضي في البرنامج لهذا الإجراء بالكلمة **Call** ، أي هكذا :

(إسم أداة الكومبو , مسار الملف المتسلسل) **Call CopyFC**

شرح مبسط عن عمل الإجراء :

أردت مثلاً نسخ جميع محتويات ملف متسلسل إلى كومبو في البرنامج أو أكثر من كومبو واحد وربما مرات متعددة في أماكن متغيرة في البرنامج ، وأنت تعرف فهي عملية تحتاج لتكرار كود ممل كل مرة ، ولكن إذا أصبح هذا الكود إجراءً واحداً كان ذلك أفضل وأسهل ، إليك الطريقة :

مثلاً : الملف المتسلسل المراد مساره هو :

P\$="C:\Program Data.txt" أو C:\Program Files\Data

وتريد نسخه إلى كومبو في البرنامج إسمه **Combo1** ، سيكون الإجراء هكذا :

Call CopyFC ("C:\Program Data" , Combo1)

أو

Call CopyFC (P\$, Combo1)

سترى النتيجة أن العناصر قد نُسخت من الملف المتسلسل إلى الكومبو بكتابة أمر واحد ، وهذا بدون تكرار
ممل لكتابة الكود المصدري أو نسخه كل مرة ! .

ملاحظة مهمة : هذا الكود الآتي يوضع في الفورم في منطقة الإعلانات العامة للفورم الـ **Declarations** ، أو في ملف
ميدول **Moduel** تابع لمشروع البرنامج .

نص الكود :

```
Sub CopyFC(FilePath As String, CmbName As ComboBox)
Dim U As String
Open FilePath For Append As 1: Close
If FileLen(FilePath) <> 0 Then
CmbName.Text = "": CmbName.Clear
Open FilePath For Input As 1
Do
Input #1, U: CmbName.AddItem U
Loop Until EOF(1)
Close
End If
End Sub
```

3 - إجراء لحفظ بيانات كومبو Combo إلى ملف متسلسل

وهذا الإجراء هو عكس الإجراء السابق ! ، حيث الإسم الافتراضي لهذا الإجراء
يكون **CopyCF** (وليس CopyFC) فقط تبديل بين موقع آخر حرفين .

وحرف **C** إشارة إلى كلمة (كومبو **Combo**) ، وحرف **F** إشارة إلى كلمة (ملف **File**) ، أقصد من الكومبو
إلى الملف **C to F**

إنتبه جيداً : يجب أن يُلحق الإسم الافتراضي في البرنامج لهذا الإجراء بالكلمة **Call** ، أي هكذا :

Call CopyCF (مسار الملف المتسلسل , إسم أداة الكومبو)

شرح مبسط عن عمل الإجراء :
أردت مثلاً عكس الإجراء السابق ، أي نسخ جميع محتويات الكومبو إلى ملف متسلسل على القرص الصلب ،
الطريقة تكون هكذا :

مثلاً : الملف المتسلسل المراد مساره وأسمه هو :
P\$="C:\Test.txt" أو C:\Program Files\GetData
وتريد نسخ جميع عناصر الكومبو **Combo1** مثلاً إلى الملف الذي سميته ، سيكون الإجراء هكذا :

Call CopyCF (P\$, Combo1)

سترى النتيجة أن العناصر قد نُسخت من الكومبو إلى الملف المتسلسل الذي سميته من عندك بكتابة أمر واحد !

ملاحظة مهمة : هذا الكود الآتي يوضع في الفورم في منطقة الإعلانات العامة للفورم الـ **Declarations** ، أو في ملف ميديول **Moduel** تابع لمشروع البرنامج .

نص الكود :

```
Sub CopyCF(FilePath As String, CmbName As ComboBox)
Dim U As String
CmbName.Refresh
If CmbName.ListCount > 0 Then
Open FilePath For Output As 1
For I = 0 To CmbName.ListCount - 1
Print #1, CmbName.List(I)
Next
Close
End If
End Sub
```

4 - إجراء لنسخ محتويات الأداة ليست List إلى الأداة Combo

حيث الإسم الافتراضي لهذا الإجراء يكون : **CopyLC**

حرف **L** إشارة إلى كلمة (ليست **List**) ، وحرف **C** إشارة إلى كلمة (كومبو **Combo**) ، أقصد من ليست إلى الكومبو **L to C**

((وهنا بداية التفاهم وعلاقات الإستيراد والتصدير بين الأختين **List** و **Combo**))

إنتبه جيداً : يجب أن يُلحق الإسم الافتراضي في البرنامج لهذا الإجراء بالكلمة **Call** ، أي هكذا :

(إسم أداة الكومبو , مسار الملف المتسلسل) **Call CopyLC**

شرح مبسط عن عمل الإجراء :

أردت مثلاً نسخ جميع محتويات ليست **List1** إلى كومبو **Combo1** في البرنامج ، سيكون الإجراء هكذا :

Call CopyLC (List1 , Combo1)

سترى النتيجة أن العناصر قد نُسخت من ليست إلى الكومبو بكتابة أمر واحد !

ملاحظة مهمة : هذا الكود الآتي يوضع في الفورم في منطقة الإعلانات العامة للفورم الـ **Declarations** ، أو في ملف **Moduel** تابع لمشروع البرنامج .

نص الكود :

```
Sub CopyLC(LstName As ListBox, CmbName As ComboBox)

Dim U As String
Dim I, L As Integer

CmbName.Clear

LstName.Refresh
L = LstName.ListCount

For I = 0 To L - 1

CmbName.AddItem LstName.List(I)

Next

End Sub
```

5 - إجراء لنسخ محتويات الأداة Combo إلى الأداة ليست List

وهذا الإجراء هو عكس الإجراء السابق ! ، حيث الإسم الافتراضي لهذا الإجراء يكون **CopyCL** (وليس **CopyLC**) فقط تبديل بين موقع آخر حرفين .

حيث الإسم الافتراضي لهذا الإجراء يكون : **CopyCL**

حرف **C** إشارة إلى كلمة (كومبو **Combo**) ، وحرف **L** إشارة إلى كلمة (ليست **List**) ، أفصد من الكومبو إلى الـ **C to L** ليست

((من البديهي أن يكون التعامل من كلا الطرفين بين **List** و **Combo**))

إنتبه جيداً : يجب أن يُلحق الإسم الافتراضي في البرنامج لهذا الإجراء بالكلمة **Call** ، أي هكذا :

(إسم أداة الكومبو , إسم أداة الـ **Call CopyCL**)

شرح مبسط عن عمل الإجراء :

أردت مثلاً نسخ جميع محتويات الكومبو **Combo1** إلى الـ **List1** في البرنامج ، سيكون الإجراء هكذا :

(**Call CopyCL (List1 , Combo1)**)

سترى النتيجة أن العناصر قد نُسخت من الكومبو إلى الـ **List1** بكتابة أمر واحد !

ملاحظة مهمة : هذا الكود الآتي يوضع في الفورم في منطقة الإعلانات العامة للفورم الـ **Declarations** ، أو في ملف **Moduel** تابع لمشروع البرنامج .

نص الكود :

```
Sub CopyCL(LstName As ListBox, CmbName As ComboBox)
Dim U As String
Dim I, L As Integer

LstName.Clear

CmbName.Refresh
L = CmbName.ListCount

For I = 0 To L - 1

LstName.AddItem CmbName.List(I)

Next

End Sub
```

6 - إجراء لإضافة أسماء الخطوط العربية الأساسية في الأداة كومبو Combo

حيث الإسم الافتراضي لهذا الإجراء يكون : **AddArFonts**

إنتبه جيداً : يجب أن يُلحق الإسم الافتراضي في البرنامج لهذا الإجراء بالكلمة **Call** ، أي هكذا :

Call AddArFonts (إسم أداة الكومبو)

شرح مبسط عن عمل الإجراء :

أردت مثلاً وضع أهم أسماء الخطوط العربية في الكومبو Combo1 بدون البحث هنا أو هناك لنسخ أسماء هذه الخطوط الواحدة تلو الأخرى ، جمعتها لك في هذا الإجراء البسيط ، هكذا :

Call AddArFonts (Combo1)

يعني أنا أطلب إضافتها تلقائياً في هذا الكومبو ، ستجدها في الكومبو فوراً ، سبعة أنواع .

طبعاً ليست هي جميع الخطوط العربية ، ولكنها الأكثر شيوعاً وأهمية ، ويمكنك إضافة خطوط عربية أخرى في هذه الدالة في حال حددت إسمها بالضبط ! ، تذكر " بالضبط " !

ملاحظة مهمة : هذا الكود الآتي يوضع في الفورم في منطقة الإعلانات العامة للفورم الـ **Declarations** ، أو في ملف موديول **Moduel** تابع لمشروع البرنامج .

نص الكود :

```
Sub AddArFonts(CmbName As ComboBox)
```

```
With CmbName
```

```
.Clear
```

```
.AddItem "Traditional Arabic"
```

```
.AddItem "Times New Roman (Arabic)"
```

```
.AddItem "Tahoma"
```

```
.AddItem "Arabic Transparent"
```

```
.AddItem "Arial"
```

```
.AddItem "Courier New (Arabic)"
```

```
.AddItem "Andalus"
```

```
End With
```

```
End Sub
```

7 - دالة وظيفية تعطي عدداً عشوائياً بين عددين محددين

حيث الإسم الافتراضي لهذه الدالة يكون : **RRnd** ، حرف **R** مكرر مرتين

حيث يوضع رقمين : أكبر ثم أصغر (من - إلى) بين قوسين بعدها أثناء كتابتها في النص البرمجي .

شرح مبسط عن عمل الدالة :

مثلاً أريد رقماً عشوائياً من العدد 20 إلى العدد 50

صيغة الكتابة البرمجية تكون هكذا :

Print **RRnd(20 , 50)**

أو

N= **RRnd(20 , 50)**

فتعطي الدالة هذه النتيجة :

5 أو 10 أو 24 أو 3 أو أي عدد عشوائي آخر يكون هو بين العددين المحددين .

ملاحظة مهمة : هذا الكود الآتي يوضع في الفورم في منطقة الإعلانات العامة للفورم الـ **Declarations** ، أو في ملف ميديول **Moduel** تابع لمشروع البرنامج .

نص الكود :

```
Function RRnd(MinNumber As Variant, MaxNumber As Variant) As Integer
```

```
Randomize
```

```
A = Val(MaxNumber)
```

```
B = Val(MinNumber) + 1
```

```
C = Int((A - B) * Rnd(1)) + B
```

```
RRnd = C
```

```
End Function
```

8 - دالة إلتقاط الحروف العربية فقط من النص

حيث الإسم الافتراضي لهذه الدالة يكون : **ArPick**

حيث توضع **حزمة نصية** بين قوسين بعدها أثناء كتابتها في النص البرمجي .

شرح مبسط عن عمل الدالة :

مثلاً قد تريد في إدخال النصوص أن يكون النص عربياً خالصاً بعد إدخال النص بدون أي نوع من حروف غيرها ، وبدون أية أكواد برمجية تضيفها في التكدست Text ، الحل في مثل هذه الدالة التي تنقي الحزمة مما هو غير عربي .

صيغة الكتابة البرمجية تكون هكذا :

Print ArPick (" فيجوال بيسك 6 ")

أو

P\$= ArPick (" فيجوال بيسك 6 ")

فتعطي الدالة هذه النتيجة : **فيجوال بيسك**

ملاحظة مهمة : هذا الكود الآتي يوضع في الفورم في منطقة الإعلانات العامة للفورم الـ **Declarations** ، أو في ملف موديول **Moduel** تابع لمشروع البرنامج .

نص الكود :

```
Function ArPick(Strings As String) As String
E$ = Strings: L = Len(Trim$(E$))
For I = 1 To L
H$ = Mid$(E$, I, 1): D = Asc(H$)
If (D >= 192 And D <= 237) Or (D >= 240 And D <= 250) Or _
D = 32 Then K$ = K$ + H$
Next
ArPick = K$
End Function
```

9 - دالة جعل مسافة واحدة إجبارية بعد كل كلمة من الحزمة

حيث الإسم الافتراضي لهذه الدالة يكون : **OneSpc**

حيث توضع **حزمة نصية** بين قوسين بعدها أثناء كتابتها في النص البرمجي .

شرح مبسط عن عمل الدالة :

مثلاً قد تريد في إدخال النصوص أن تكون كلمات النص منسقة تماماً بوجود مسافة واحدة إجبارية بين جميع كلمات وعبارات النص ، ومهما كان طويلاً وبأية لغة كانت ، إليك الحل :

صيغة الكتابة البرمجية تكون هكذا مثلاً :

Print OneSpc (" فلان فلان الغلاني ")

أو

P\$= OneSpc (" فلان فلان الغلاني ")

ستلاحظ أنني تعمدت إضافة مسافات زائدة في النص ، ستعطي الدالة هذه النتيجة : **فلان فلان الفلاني**
أي صارت المسافات بين الكلمات على قدم المساواة : واحدة واحدة بدون أي تباعد غير منسق !

ملاحظة مهمة : هذا الكود الآتي يوضع في الفورم في منطقة الإعلانات العامة للفورم الـ **Declarations** ، أو في ملف
ميدول **Moduel** تابع لمشروع البرنامج .

نص الكود :

```
Function OneSpc(Word As String) As String
Dim A As String
Dim I As Integer
A = Trim$(Word)
For I = 20 To 2 Step -1
If InStr(A, Space$(I)) <> 0 Then A = Replace(A, Space$(I), " ")
Next
OneSpc = A
End Function
```

10 - دالة إختيار عنصر محدد من ترتيبه في الحزمة

حيث الإسم الافتراضي لهذه الدالة يكون : **SelStr**

شرح مبسط عن عمل الدالة :

توضع المدخلات بين قوسين بعدها أثناء كتابتها في النص البرمجي ، كالصيغة الآتية :
(رقم ترتيب الكلمة أو العبارة , نوع الفاصل , حزمة نصية) **Print SelStr**

قد تكون هذه الدالة بسيطة التركيب ، ولكن صدقني أهميتها تفوق بساطتها لأن فائدتها متعددة الأوجه
للمبرمج الفطن ، وذلك لأسباب كثيرة ، وسأشرح ذلك بعدة أمثلة ، كالآتي :

مثال أول : **P\$=SelStr ("",4,"", "لا إله إلا الله")**

المعنى : أي أريد الكلمة ذات الترتيب الرابع في اللغة العربية ابتداء من اليمين من الجملة ، والفاصل "" أو "" هو بمعنى واحد في الدالة الذي يعني مسافة واحدة ، ستكون النتيجة هي كلمة لفظ الجلالة : **الله**

مثال ثان : **P\$=SelStr ("",1,"", "محمد.رسول.الله")**

المعنى : أي أريد الكلمة ذات الترتيب الأول في اللغة العربية ابتداء من اليمين من الجملة ، والفاصل يكون
"." النقطة ، ستكون النتيجة هي كلمة : **محمد**

مثال ثالث : $P\$ = \text{SelStr} (\text{Cstr}(\text{Date}), "-", 3)$

المعنى : أني أريد العنصر الثالث في الترتيب من الجملة الإنجليزية إبتداء من جهة اليسار $\text{Cstr}(\text{Date})$ وتعني هذه الجملة وضع التاريخ الجاري في صيغة حزمة **String** ، والفاصل غالباً في جملة التاريخ هذه تكون " - " علامة الشَّرْطَة أو الناقص ، وتعرف أن تنسيق التاريخ قد يكون هكذا مثلاً : السنة-الشهر-اليوم **1-2014-12** (بمعنى أنك يجب أن تعرف صيغة نمط إظهار التواريخ في نظام جهازك أولاً) ، والعدد **3** في هذا المثال يعني خانة السنة إبتداء من جهة اليسار ، ستكون النتيجة مثلاً هي كلمة : **2014** وهذا المثال يوازي الأمر $\text{Year}(\text{Date})$ للمتأمل.

مثال رابع : $P\$ = \text{SelStr} (\text{Cstr}(\text{Time}), "-", 1) : N = \text{Val}(P\$)$

المعنى : أني أريد العنصر الأول في الترتيب من الجملة الإنجليزية إبتداء من جهة اليسار $\text{Cstr}(\text{Time})$ وتعني هذه الجملة وضع الوقت الحالي في صيغة حزمة **String** ، والفاصل غالباً في جملة الوقت هذه تكون " : " علامة التنصيص ، وتعرف أن تنسيق الوقت يكون هكذا مثلاً : **12:15:33 PM** ، والعدد **1** في هذا المثال يعني خانة الساعات إبتداء من جهة اليسار ، ستكون نتيجة المتغير **N** مثلاً هي كلمة : **12** وهذا المثال يوازي الأمر $\text{Hour}(\text{Time})$.

مثال أخير وهام : $P\$ = \text{SelStr} ("الإسم^العمر^المهنة^الجنسية^عنوان السكن", "^", 1)$
 $D\$(5) \quad D\$(4) \quad D\$(3) \quad D\$(2) \quad D\$(1)$

المعنى : تستطيع في برنامج جدي للبيانات يطلب تسجيل العناصر الخمس مثلاً أعلاه أن تربط بيانات السجلات الخمس أو أكثر عن شخص أو موضوع واحد بمصفوفة حرفية $\text{Dim D}\$(5)$ في متغير حرفي واحد مع تحديد فاصل نادر وخاص من عندك " / \ | = + _ * & ^ % \$ # @ ! ~ " ، هكذا :

```
For N=1 To 5
If N=5 Then E$=""Else E$="^"
U$=U$+D$(N)+E$
Next
```

ثم الخطوة التالية :

$K\$ = \text{SelStr} (U\$, "^", \text{رقم ترتيب السجل المراد})$

أتمنى أن تكون الفكرة قد وصلت !

ملاحظة مهمة : هذا الكود الآتي يوضع في الفورم في منطقة الإعلانات العامة للفورم الـ **Declarations** ، أو في ملف ميديول **Moduel** تابع لمشروع البرنامج .

نص الكود :

```
Function SelStr(WordsSet As String, Comma As String, Number As Variant)
As String

Dim M() As String

Dim U, U2 As Integer
Dim T, N As String

N = Trim$(Comma): If N = "" Then N = " "

U = Val(Number): T = Trim$(WordsSet)

M = Split(T, Mid$(N, 1, 1)): U2 = UBound(M)

If U < 1 Then U = 1
If U - 1 > U2 Then U = U2 + 1

SelStr = Trim$(M(U - 1))

End Function
```

وفي الختام أتمنى أن يفيد هذا الموضوع من يهمله الأمر من المبرمجين بالفيجوال بيسك ،
وإلى مجموعة أكواد مبتكرة أخرى في الجزء الرابع إن شاء الله ،،

تأليف وإعداد / عبدالله خضر عبدالله الهوساوي

E-Mail : prog909@yahoo.com