

بسم الله الرحمن الرحيم

الكامل في فيجوال بيسك دوت نت

التعامل مع ..

انواع الاقراص ، المجلدات ، الملفات

مقدمة :

ان التعامل مع اجزاء القرص الصلب والقرص المضغوط و الحصول على معلومات عنه ، وكذلك التعامل مع المجلدات و الملفات قد يبدو لأول وهلة انه من الامور الغير مهمة في البرمجة .. خاصة ان الكثير من كتب البرمجة الخاصة بلغة فيجوال بيسك دوت نت اما ان تتعرض لهذا الموضوع دون الخوض في التفاصيل (وانما مجرد معلومات و نظرة عامة) او ان لا تتطرق الى هذا الموضوع نهائيا .. وفي العادة نحن نحتاج الى التعامل مع المجلدات و الملفات او معرفة ماهي اجزاء القرص الصلب او بعض المعلومات عنها عندما نقوم بكتابة العديد من البرامج .

لذا وجدت انه من المفيد عمل بحث مصغر يحتوي على معلومات تخص هذا المجال ، ومع انني لم اتطرق لكافة الخصائص و الطرق بالتفصيل .. الا اني عملت على شرح الخصائص و الميزات الاكثر استخداما .. وقد حاولت قد الامكان ان يكون الشرح مبسطا ومفصلا لكي يكون نافع للمبرمج سواء كان بمستوى مبتدئ او متقدم على حد سواء .. كما اني قد عملت على ارفاق البرامج و الاكواد التي يتم تنفيذها من خلال هذا البحث مع البحث لكي يتسنى للطالب الاطلاع عليها و الاستفادة منها ، مع ان نصيحتي ان يقوم الطالب بكتاب الاكواد بيده لكي يتجنب الوقوع في الاخطاء ، كما ان الممارسة العملية تجعل عملية ترسخ المعلومات في الذهن افضل و اسرع من نسخ الاكواد و لصقها او الاطلاع على الاكواد من خلال فتح البرامج دون كتابتها و تطبيقها عمليا .

اخيرا ..

ارجو ان تكون معلومات هذا البحث المصغر نافعة و مفيدة للطالب ..

دعائي للجميع بالتوفيق .

المهندس
عدي ياسين
06-06-2007

التعامل مع الاقراص Drives :

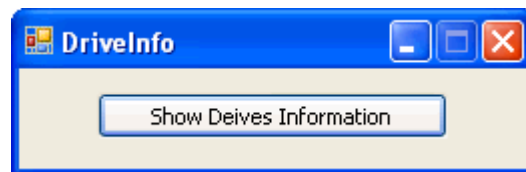
قد نحتاج في بعض برامجنا الى معرفة بعض المعلومات عن القرص الصلب (Hard Disk) .. مثل اسماء اجزاء القرص الصلب (... , E , D , C) ، او نوع نظام الملفات لأجزاء القرص الصلب (Fat , Fat32 , NTFS) ، او الحجم التخزيني للقرص الصلب او غيرها من المعلومات الاخرى ..
توفر لنا VB.NET عدة اساليب للتعامل مع القرص الصلب و التعرف على المعلومات المطلوبة التي تخص اجزاء القرص الصلب .. وسنتناول الحديث عن هذه الاساليب من خلال السطور القليلة القادمة ..

للتعامل مع القرص الصلب فاننا نستخدم الكائن (DriveInfo) .. حيث توفر لنا مجموعة من الخصائص التي تساعدنا في الحصول على المعلومات المطلوبة .
وكخطوة اولية نحتاج الى تعريف فضاء الاسماء System.IO من اجل التعامل مع الكائن DriveInfo .

لتبسيط الموضوع بصورة اكبر ، سنعمل على التطرق لخصائص الكائن DriveInfo من خلال مثال عملي ..

مثال :

نقوم في البداية بانشاء مشروع جديد يحتوي على نموذج Form1 ..
نضع الاداة Button على النموذج ونقوم باسناد النص (Show Deives Information) للخاصية Text التابعة للأداة .
يبدو تصميم النموذج كما يلي :



الان نقوم بكتابة الكود التالي لحدث النقر التابع للزر Button1 ..

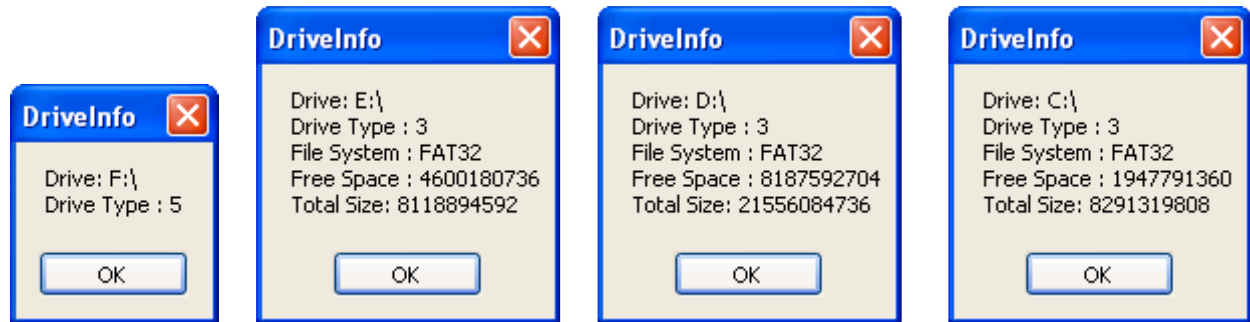
```
Imports System.IO
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim Str As String
        Dim allDrives() As DriveInfo = DriveInfo.GetDrives
        Dim Drv As DriveInfo

        For Each Drv In allDrives
            Str = "Drive: " & Drv.Name & vbCrLf
            Str &= "Drive Type : " & Drv.DriveType & vbCrLf
            If Drv.IsReady = True Then
                Str &= "File System : " & Drv.DriveFormat & vbCrLf
                Str &= "Free Space : " & Drv.TotalFreeSpace & vbCrLf
                Str &= "Total Size: " & Drv.TotalSize
            End If
            MsgBox(Str)
            Str = ""
        Next
    End Sub
End Class
```

نقوم بتنفيذ المثال بنقر مفتاح F5 من لوحة المفاتيح KeyBoard ..

ننقر على الزر Show Deives Information ..
سيتم عرض مجموعة من صناديق الرسالة Message Box على عدد اجزاء الاقراص الموجودة في الحاسب .
ويحتوي كل منها على مجموعة من المعلومات حول ذلك القرص ..



الان وبعد ان لاحظنا نجاح عمل البرنامج في عرض المعلومات المطلوبة .. ناتي الان الى مناقشة الكود .

في بداية الكود قمنا بتعريف فضاء الاسماء `Imports System.IO`

وذلك لكي نتمكن من التعامل مع الكائن `DriveInfo` .

بعد ذلك قمنا بتعريف المتغيرات المطلوبة في البرنامج وهي ..

المتغير `Str` : وهو متغير نصي نقوم بخزن المعلومات داخله لكي نقوم بعرضها في صندوق الرسالة .

المتغير `allDrives()` : وهو متغير من نوع `DriveInfo` .

المتغير `Drv` : وهو متغير من نوع `DriveInfo` .

ان الفرق بين المتغير `allDrives()` و المتغير `Drv` هو ان الاول من نوع مصفوفة وذلك لكي يتم تخزين داخله

كل الاقراص الموجودة (كما سنلاحظ من خلال متابعة الشرح) .

الان نقوم بالتجول داخل `allDrives()` للحصول على معلومات كل قرص من الاقراص ..

`For Each Drv In allDrives`

وبذلك من خلال الخطوة اعلاه نلاحظ ان المتغير `Drv` يحتوي على معلومات حول قرص واحد فقط في حين ان

المتغير `allDrives()` يحتوي على معلومات جميع الاقراص .

بعد ذلك نقوم باستخدام خصائص `Drv` من اجل الحصول على المعلومات المطلوبة ...

الخاصية `Drv.Name` : تعيد لنا اسم القرص (`C , D , ....`) .

الخاصية `Drv.DriveType` : تعيد لنا قيمة رقمية تمثل نوع القرص .. ومن خلال هذه القيمة الرقمية يمكننا

التمييز بين اجزاء القرص الصلب وغيرها من الاقراص ..

وتخضع هذه القيمة الرقمية للجدول التالي ..

العنصر	القيمة
Unknown	0
CDRom	1
DVDRom	2
Fixed	3
RamDisk	4
Remote	5
Removable	6

من خلال ملاحظة الجدول اعلاه يمكننا معرفة ان اجزاء القرص الصلب تاخذ القيمة 3 .

بعد هذه الخطوة ناتي الى الخطوة التالية وهي التأكد من صلاحية القرص .. وذلك من اجل الاستمرار في الحصول على المعلومات حوله .. لان القرص المضغوط مثلا لايمكننا الحصول على المعلومات حوله ان لم يكن بداخله قرص CD مثلا ..

ويتم التأكد من صلاحية جاهزية القرص باستخدام الخاصية Drv.IsReady ، فهي تعيد القيمة True ان كان القرص جاهز وتعيد القيمة False ان لم يكن القرص جاهزا ..
بعد ذلك نستمر باستخدام الخصائص للحصول على المعلومات المطلوبة ..

الخاصية Drv.DriveFormat : وهي ترجع لنا نوع نظام الملفات المستخدم (Fat , Fat32 , NTFS , ...) .
الخاصية Drv.TotalFreeSpace : تعيد لنا المساحة التخزينية الكلية الفارغة في القرص .
الخاصية Drv.TotalSize : تعيد لنا المساحة التخزينية الكلية للقرص .

بعد ذلك ، تكون المعلومات قد اكتملت وتم تخزينها في المتغير Str وبذلك يكون المتغير جاهزا لعرض محتواه في صندوق الرسالة من خلال الكود MsgBox (Str) ..
بعدها نقوم بافراغ المتغير Str باسناد القيمة "" له .. ثم نرجع الى داخل الدوارة For Each للحصول على معلومات القرص التالي ..

تجدد الإشارة الى ان DriveInfo يحتوي على عدد من الخصائص الاخرى ولكننا من خلال مثالنا هذا قمنا بالإشارة الى الخصائص الأكثر استخداما ..

لقد استخدمنا vbCrLf مع المتغير Str وذلك من اجل اضافة سطر جديد للنص ..

الى هنا انتهى مثالنا .. ننتقل الان الى القسم التالي وهو التعامل مع المجلدات Folders .

التعامل مع المجلدات Folders :

توفر لنا بيئة VB.net ادوات مهمة للتعامل مع المجلدات و استعراضها وهما :

OpenFileDialog & SaveFileDialog

ولكننا في بعض الاحيان نحتاج الى التحكم ضمينا بالمجلدات دون ان ينتبه المستخدم لذلك او ان حاجة البرنامج لا تتطلب وجود احد الاداتين اعلاه ...

بذلك فان بيئة VB.Net توفر لنا مجموعة من الخصائص التي نستطيع من خلالها التحكم بالمجلدات ... وهذا ما سنلاحظه من خلال الاسطر القليلة القادمة .

انشاء مجلد جديد :

لأنشاء مجلد جديد نستخدم الكود التالي :

```
My.Computer.FileSystem.CreateDirectory("C:\TestFolder1")
```

حيث يقوم الكود السابق بإنشاء مجلد جديد (TestFolder1) على القسم C من القرص الصلب ..
من ملاحظتنا للكود السابق يتبين لنا انه يجب علينا تحديد المسار الذي سيتم انشاء المجلد الجديد فيه وكذلك تحديد اسم المجلد الجديد .

ملاحظة :

يمكننا انشاء مجلد جديد باستخدام اسلوب اخر وهو من خلال الكود التالي :

```
Mkdir("C:\TestFolder2")
```

حيث يقوم الكود السابق بانشاء مجلد جديد (TestFolder2) في القسم C من القرص الصلب .

تجدد الإشارة الى ان الطريقتين اعلاه يقومان بانشاء مجلد جديد ضمن المسار المحدد لهما .. لكن ماذا لو كان المجلد موجود اصلا؟! هنا في الحالة الاولى لن يحدث أي خطأ ، اما باستخدام Mkdir سينتج لنا خطأ .

حذف مجلد :

توجد لدينا عدة خيارات توفرها لنا بيئة VB.Net عند حذف مجلد معين ...

أولاً : حذف المجلد ان كان فارغاً (لايحتوي على أي مجلدات او ملفات) :

هنا نستخدم الكود التالي :

```
My.Computer.FileSystem.DeleteDirectory("C:\TestFolder2", _
FileIO.DeleteDirectoryOption.ThrowIfDirectoryNonEmpty)
```

هنا سيتم حذف المجلد TestFolder2 ان كان فارغاً ولايحتوي على مجلدات او ملفات ..
تجدد الإشارة الى انه سيحدث عندنا خطأ في حالة وجود ملفات او مجلدات داخل المجلد المراد حذفه ، وكذلك ان كان المجلد المراد حذفه غير موجود ..

ثانياً : حذف المجلد سواء كان فارغاً او يحتوي على مجلدات و ملفات :

هنا سنستخدم الكود التالي :

```
My.Computer.FileSystem.DeleteDirectory("C:\TestFolder1", _
FileIO.DeleteDirectoryOption.DeleteAllContents)
```

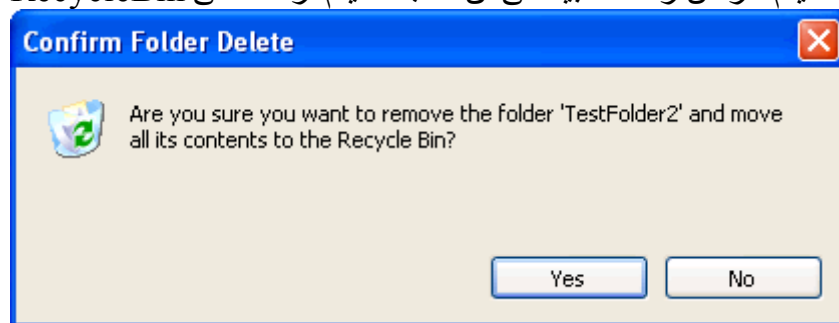
هنا تتم عملية الحذف دون الاهتمام ان كان المجلد فارغاً او فيه محتويات (مجلدات ، ملفات) .. وكذلك دون تنبيه المستخدم ان عملية الحذف ستتم ..

ثالثاً : حذف المجلد مع محتوياته وارساله الى RecycleBin :

نستخدم الكود التالي :

```
My.Computer.FileSystem.DeleteDirectory("C:\TestFolder2", _
FileIO.UIOption.AllDialogs, _
FileIO.RecycleOption.SendToRecycleBin)
```

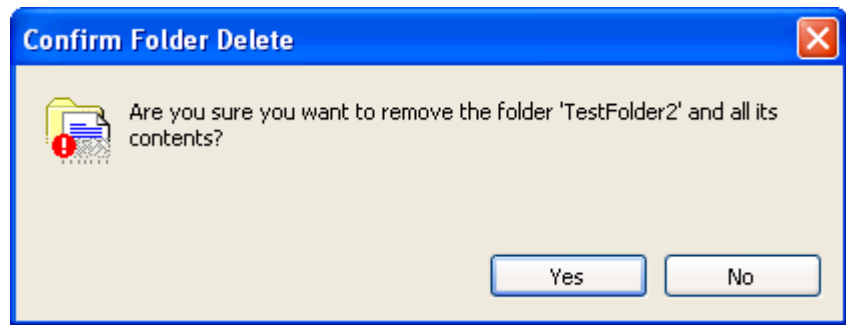
حيث عند حذف المجلد يتم عرض رسالة تنبيه الى ان المجلد سيتم ارساله الى RecycleBin ..



رابعاً : حذف المجلد مع محتوياته دون ارساله الى RecycleBin :

هنا نستخدم الكود التالي :

```
My.Computer.FileSystem.DeleteDirectory("C:\TestFolder2", _
FileIO.UIOption.AllDialogs, _
FileIO.RecycleOption.DeletePermanently)
```



خامسا : حذف المجلد باستخدام Rmdir :

وذلك باستخدام الكود :

```
Rmdir("C:\TestFolder1")
```

ملاحظة : جميع الكود السابق في التعامل مع المجلدات من ناحية الانشاء و الحذف تجده في المثال المرفق بالاسم (NewFolders_And_DeleteFolders) .
حيث ستجد نموذج يحتوي على مجموعة من الازرار ، كل منها يقوم بمهمة احد الشفرات السابقة الذكر .

معرفة ان كان المجلد موجود او لا :

كما لاحظنا من خلال الكود السابق الذكر انه عندما نحذف مجلد او نقوم بانشاء مجلد جديد باستخدام بعض الاساليب فانه يحدث عندنا خطأ ان كان المجلد غير موجود (في حالة حذف المجلد) او المجلد موجود اصلا (في حالة انشاء مجلد جديد) ..
لذلك فاننا نحتاج الى التأكد من وجود المجلد او عدم وجوده قبل الشروع بعملية الحذف او الانشاء .. كما اننا قد نحتاج الى التأكد من وجود المجلد قبل ان نبدا بنقل الملفات او المجلدات اليه وغيرها من الاحتياجات الاخرى .

للتأكد من وجود او عدم وجود مجلد معين في مسار معين نستخدم الكود التالي :

```
If My.Computer.FileSystem.DirectoryExists("C:\TestFolder1") = True Then
    MsgBox("The Directory Exist")
Else
    MsgBox("The Directory Not Exist")
End If
```

حيث عند وجود المجلد سيتم عرض صندوق رسالة يخبرنا بوجود المجلد والا فان صندوق الرسالة سيخبرنا بعدم وجود المجلد المطلوب ..

اعادة تسمية مجلد :

من خلال الكود التالي سنقوم باعادة تسمية مجلد موجود اصلا واعطائه اسم جديد ..

```
My.Computer.FileSystem.RenameDirectory("C:\TestFolder1", "New Name")
```

حيث عند تنفيذ الكود سيتم اسناد الاسم New Name الى المجلد بدل من الاسم TestFolder1 ..

نسخ محتويات مجلد الى مجلد اخر :

هنا سنقوم بنسخ محتويات المجلد TestFolder1 الى المجلد TestFolder2 ..

```
My.Computer.FileSystem.CopyDirectory("C:\TestFolder1", "C:\TestFolder2")
```

نقل محتويات مجلد الى مجلد اخر :

في الحالة السابقة قمنا بعمل نسخة من محتويات مجلد معين ولصقها داخل مجلد اخر ، اما هنا فسنقوم بنقل المحتويات من المجلد المصدر الى المجلد الهدف ..

`My.Computer.FileSystem.MoveDirectory("C:\TestFolder1", "C:\TestFolder2", True)`
 تجدر الإشارة الى اننا بعد ان نقوم بعملية نقل محتويات المجلد المصدر الى المجلد الهدف ، سيتم حذف المجلد المصدر ..

ملاحظة: من خلال مشاهدة الكود اعلاه ، وفي نهايته نلاحظ وجود `True` ، وهي تفيد انه اذا احتوي المجلد الهدف على ملفات مشابهة لملفات المجلد المصدر (من ناحية النوع و التسمية) ، فسيتم الكتابة فوقها واستبدالها .. اما لو كتابنا بدلها `False` عندها لن تحدث الكتابة الفوقية (استبدال الملفات التي من نفس النوع و الاسم) .

عرض المجلدات الموجودة داخل مجلد معين :

سنفترض في البداية ان لدينا المجلد `TestFolder1` ويحتوي داخله على مجلدين وهما `Test1 & Test2` وكذلك المجلد `Test1` يحتوي بداخله على المجلدين `New Folder1` و `New Folder2` .

توجد لدينا هنا حالتان :

أولاً: عرض المجلدات الموجودة داخل المجلد الرئيسي المطلوب فقط :

وهنا سيتم عرض المجلدين `Test1` و `Test2` دون عرض المجلدان الموجودان داخل المجلد `Test1` .

```
For Each Direc As String In
My.Computer.FileSystem.GetDirectories("C:\TestFolder1", _
FileIO.SearchOption.SearchTopLevelOnly, "**")
    MsgBox(Direc)
Next
```

ثانياً: عرض جميع المجلدات الفرعية الموجودة داخل المجلد الرئيسي المطلوب :

وهنا سيتم عرض جميع المجلدات الموجودة داخل المجلد الرئيسي .

```
For Each Direc As String In
My.Computer.FileSystem.GetDirectories("C:\TestFolder1", _
FileIO.SearchOption.SearchAllSubDirectories, "**")
    MsgBox(Direc)
Next
```

التعامل مع الملفات Files :

أولاً : لمعرفة ان كان الملف موجود أو لا :

في الكثير من الحالات نحتاج الى معرفة ان كان الملف موجود في مجلد معين او لا ، مثلاً عندما نريد فتحه و خزن معلومات فيه ، او نريد ان نقوم بانشاء ملف عند عدم تواجد الملف المطلوب او غيرها من الحالات . لتحقيق هذا الغرض نستخدم الكود التالي :

```
Imports System.IO
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim path As String = "C:\TestFolder2\mytest.txt"
        Dim val As Boolean
        val = File.Exists(path)
        If val = True Then
            MsgBox("The File Found")
        Else
            MsgBox("The File not Found")
        End If
    End Sub
End Class
```

في الكود السابق يتم البحث عن الملف mytest.txt في المجلد TestFolder2 الموجود في الجزء C من القرص الصلب ..
 هنا استخدمنا الكائن File ومن خلال الخاصية Exists التابعة له تمكنا من التحقق ان كان الملف موجود في المسار المعين او لا .. حيث يتم التأكد من خلال المتغير Val و الذي يتم خزن فيه قيمة منطقية True & False تبين لنا حالة وجود الملف او لا ...
 فان كان الملف موجود في المسار المذكور يتم ارجاع القيمة True وخزنها في المتغير val وبذلك نعرض صندوق رسالة يخبرنا ان الملف المطلوب موجود في المسار ، اما ان كانت القيمة المنطقية هي False فان صندوق الرسالة يخبرنا ان الملف غير موجود .

ثانيا : تغيير اسم ملف معين :

لغرض تغيير اسم ملف معين الى اسم جديد نستخدم الكود التالي :

```
My.Computer.FileSystem.RenameFile("C:\TestFolder2\mytest.txt", "Test1.txt")
```

حيث يقوم بتغيير اسم الملف من mytest.txt الى Test1.txt ..

كذلك توجد طريقة اخرى لتغيير اسم ملف معين وذلك باستخدام Rename كما في الكود التالي :

```
Rename("C:\TestFolder2\Test1.txt", "C:\TestFolder2\Test2.txt")
```

ثالثا : معرفة عدد الملفات الموجودة في مجلد معين :

لمعرفة عدد الملفات الموجودة في مجلد معين نستخدم الكود التالي :

```
Dim counter As System.Collections.ObjectModel.ReadOnlyCollection(Of String)
counter = My.Computer.FileSystem.GetFiles("C:\TestFolder2")
MsgBox(counter.Count.ToString)
```

باستخدام الخاصية Count التابعة لـ Counter يقوم صندوق الرسالة بعرض عدد الملفات الموجودة في المجلد C:\TestFolder2 .

رابعا : عرض الملفات الموجودة في مجلد معين :

تعلمنا في الخطوة السابقة كيف نحصل على عدد الملفات الموجودة في مجلد معين ، لكننا في بعض الاحيان نحتاج الى التعرف على هذه الملفات وعرض اسماءها ...

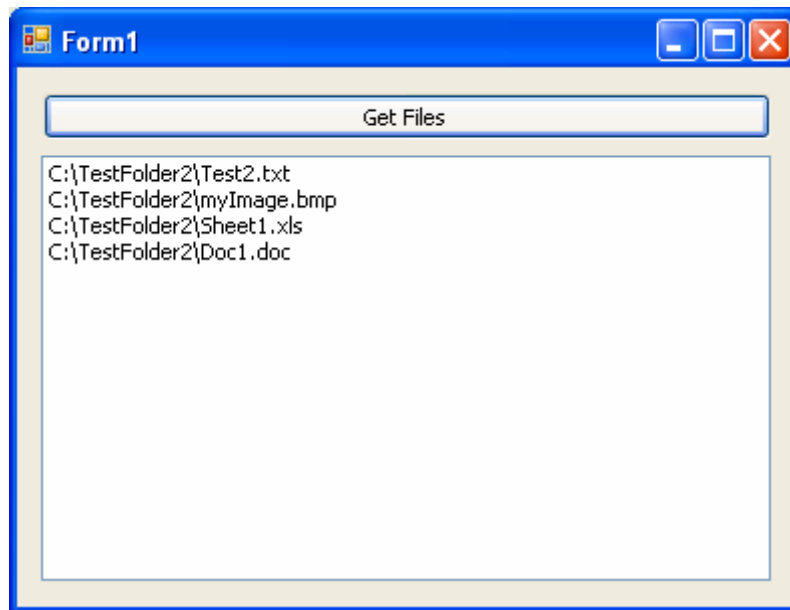
مثال :

نقوم بإنشاء مشروع جديد ونقوم بوضع الاداة Button على النموذج و الاداة ListBox ..

ثم نقوم بكتابة الكود التالي لحدث النقر للاداة Button1 :

```
Dim FileList As System.Collections.ObjectModel.ReadOnlyCollection(Of String)
FileList = My.Computer.FileSystem.GetFiles("C:\TestFolder2", _
FileIO.SearchOption.SearchTopLevelOnly, "*.*)
For Each FileFound As String In FileList
    ListBox1.Items.Add(FileFound)
Next
```

عندما نقوم بتنفيذ البرنامج وننقر على الزر سيتم عرض الملفات الموجودة في المجلد TestFolder2 داخل الاداة ListBox1 كما في الشكل التالي :



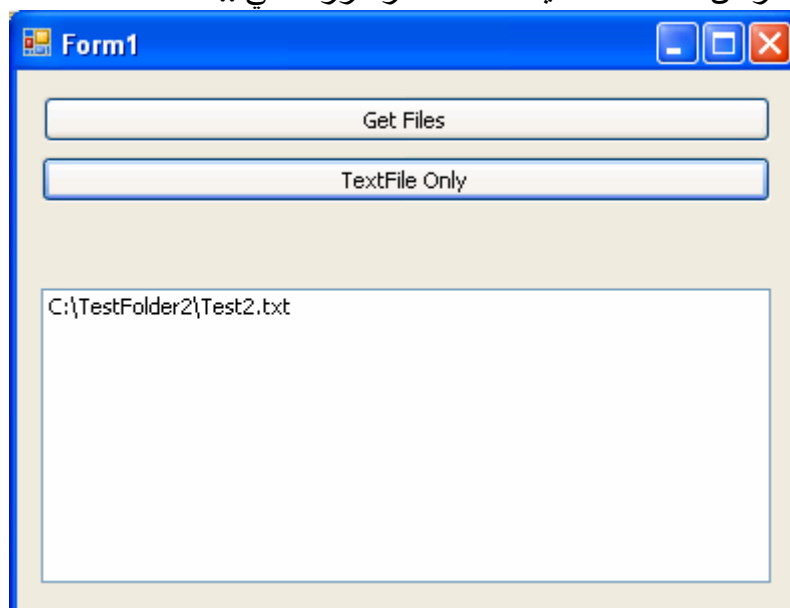
حيث يتضح لنا من الصورة اعلاه ان المجلد المذكور يحتوي على اربع ملفات ...

وتجدر الاشارة الى اننا يمكننا تطوير البرنامج ليقوم بارجاع نوع معين من الملفات (الملفات النصية مثلا)

نقوم باضافة زر جديد للنموذج ونقوم بكتابة الكود التالي له :

```
Dim FileList As System.Collections.ObjectModel.ReadOnlyCollection(Of
String)
FileList = My.Computer.FileSystem.GetFiles("C:\TestFolder2",
FileIO.SearchOption.SearchTopLevelOnly, "*.txt")
For Each FileFound As String In FileList
    ListBox1.Items.Add(FileFound)
Next
```

فعند تنفيذ البرنامج يتم عرض الملفات النصية فقط عند نقر الزر الثاني ..



ويعود السبب الى ان الخاصية GetFiles تأخذ ثلاث متغيرات وهي :

- 1- مسار المجلد المطلوب الحصول على ملفاته .
- 2- نوع البحث عن الملفات (البحث في المجلد الرئيسي فقط ام في المجلد الرئيسي و المجلدات الفرعية).
- 3- نوع الملفات المطلوب .

فان كتابتنا في المتغير نوع الملفات المطلوب هو (*.*) عندها سيتم ارجاع جميع الملفات الموجودة في المجلد .
اما ان كتبنا (*.txt) عندها سيتم ارجاع الملفات النصية فقط وهكذا ..

خامسا : نقل الملف من مجلد الى مجلد اخر :

لكي نقوم بنقل الملف من مجلد الى مجلد اخر نستخدم الكود التالي :

```
My.Computer.FileSystem.MoveFile("C:\TestFolder2\Test1.txt", _  
"C:\TestFolder1\Test1.txt", True)
```

واما True الموجودة في اخر الكود فهي من اجل السماح بالكتابة الفوقية عند وجود الملف في المجلد الجديد .

سادسا : نسخ ملف من مجلد الى مجلد آخر :

لكي نقوم بنسخ ملف من مجلد الى مجلد اخر نستخدم الكود التالي :

```
My.Computer.FileSystem.CopyFile("C:\TestFolder1\Test1.txt", _  
"C:\TestFolder2\Test.txt", True)
```

اما من اجل نسخ الملف وتغيير اسمه في نفس الوقت فليس علينا سوى كتابة أي اسم جديد بدل الاسم القديم في الكود السابق ... وكما يلي :

```
My.Computer.FileSystem.CopyFile("C:\TestFolder1\Test1.txt", _  
"C:\TestFolder2\TestFile.txt", True)
```

ملاحظة :

جميع الكود و البرامج التي تم عملها خلال هذا البحث ، تجدها مرفقة معه ...

الى هنا انتهى بحثنا ...

والحمد لله رب العالمين ..

المهندس
عدي ياسين
engauday@hotmail.com
08-06-2007

الفهرس

رقم الصفحة	المادة
1	المقدمة.....
2	التعامل مع الاقراص.....
4	التعامل مع المجلدات.....
4	انشاء مجلد جديد.....
5	حذف مجلد.....
5	حذف المجلد ان كان فارغا.....
5	حذف المجلد سواء كان فارغا او لا.....
5	حذف المجلد وارسال محتوياته الى RecycleBin.....
5	حذف مجلد دون ارسال محتوياته الى RecycleBin.....
6	حذف مجلد باستخدام Rmdir.....
6	معرفة ان كان المجلد موجود او لا.....
6	اعادة تسمية مجلد.....
6	نسخ محتويات مجلد الى مجلد اخر.....
6	نقل محتويات مجلد الى مجلد اخر.....
7	عرض المجلدات الموجودة داخل مجلد معين.....
7	عرض المجلدات الموجودة داخل المجلد الرئيسي المطلوب فقط.....
7	عرض جميع المجلدات الفرعية الموجودة داخل المجلد الرئيسي المطلوب ..
7	التعامل مع الملفات.....
7	معرفة ان كان الملف موجود او لا.....
8	تغيير اسم ملف معين.....
8	معرفة عدد الملفات الموجودة في مجلد معين.....
10	نقل الملفات من مجلد الى مجلد اخر.....
10	نسخ الملفات من مجلد الى مجلد اخر.....

المصادر

مكتبة MSDN مع الاصدار Visual Basic.Net Express 2005