

أكواد متنوعة من الفيچوال بيسك دوت نت للكاتب : المهندس... خالد علام أحمد عمر

اهداء:

الى أبى و أمى

الى كل العرب

الى عائلتى

الى ابناء بلدى الحبيبه (أسوان)

و الى اصدقائى الاعزاء الذين علمونى أساسيات الفيچوال بيسك دوت نت (محمد أسامه – أحمد أشرف)

و الشكر للدكتور الذى ساعدنا فى التعمق فى البرمجه (الأستاذ الدكتور: مصطفى جمعه)

الكتاب مجانى لكل العرب

للمراسله

kh_al505@yahoo.com

kh_al505@hotmail.com



بسم الله الرحمن الرحيم

الفصل الاول التطبيقات Application

تفعيل تطبيق فى حالة عمل بواسطة الاسم----

Activate a running Application by Name --

```
Imports Microsoft.VisualBasic
-----
AppActivate("untitled - Notepad")
```

تفعيل تطبيق فى حالة عمل بواسطة رقم العملية----

Activate a Running Application by process ID--

```
Imports Microsoft.VisualBasic
-----
Dim ProcessID As Integer
ProcessID = Shell("Notepad.exe", AppWinStyle.NormalFocus)
AppActivate(ProcessID)
```

الحصول على الخرج من تطبيق كونسول---

Capture Output from a Console Application

```
Imports System
Imports System.Diagnostics
Imports Microsoft.VisualBasic
-----
Dim output As String = "non"
Dim consoleApp As New Process
Try
    With consoleApp
        .StartInfo.UseShellExecute = False
        .StartInfo.RedirectStandardOutput = True
        .StartInfo.FileName = "consoleApplication.exe"
        .Start()
        .WaitForExit()
    End With
    output = consoleApp.StandardOutput.ReadToEnd()
Catch ex As Exception
    MsgBox(ex.Message)
End Try
MsgBox(output)
```

تغيير اللون الامامى و الخلفى لتطبيق الكونسول----

Change the Foreground and background Colors in a

Console window

```
Imports Microsoft.VisualBasic
-----
```

```

Console.BackgroundColor = ConsoleColor.DarkRed
Console.ForegroundColor = ConsoleColor.Gray
Console.Clear()
----- التحقق من الاصدار الحالي بطغته زر-----
Check for Currnt Version using ClickOnce

```

```

Imports System.Deployment
-----

```

```

If My.Application.IsNetworkDeployed Then

currentAppVersion=My.Application.Deployment.CurrentVersion.ToString

lastupdateDate=My.Application.Deployment.TimeOfLastUpdateCheck

End If

----- التحقق من التحديث الحالي بطغته زر-----
Check for update to clickOnce Application

```

```

Imports System.Deployment
-----

```

```

If My.Application.IsNetworkDeployed Then
    If My.Application.Deployment.CheckForUpdate Then

        End If
    End If

```

```

----- مسح شاشة الكونسول-----
Clear the Cosole Window

```

```

Imports Microsoft.VisualBasic
-----
Console.Clear()

```

```

----- مسح ملف من منطقة تخزين معزوله-----
Delete File from Isolated Storage

```

```

Imports System.IO.IsolatedStorage
Imports Microsoft.VisualBasic
-----
Dim isolatedStorage As IsolatedStorageFile
isolatedStorage=IsolatedStorageFile.GetStore _
(IsolatedStorageScope.User Or IsolatedStorageScope.Assembly _
, Nothing, Nothing)

isolatedStorage.DeleteFile("fullpath\fileName.txt")

```

```

----- حساب المساحة المتاحة من منطقة تخزين معزوله-----
Determine the Amount of Isolated Storage Available

```

```

Imports System.IO.IsolatedStorage
Imports Microsoft.VisualBasic
-----
Dim isolatedStorage As IsolatedStorageFile =
IsolatedStorageFile.GetStore(IsolatedStorageScope.User Or
IsolatedStorageScope.Assembly, Nothing, Nothing)

```

```
Dim spaceAvailable As ULong = isolatedStorage.MaximumSize -
isolatedStorage.CurrentSize
```

تنزيل الملفات بطريقة متزامنه-----

Download Files Synchronously with ClickOnce

```
Imports System.Deployment
```

```
-----
If My.Application.IsNetworkDeployed Then
    If Not
(My.Application.Deployment.IsFileGroupDownloaded("Media")) Then
        My.Application.Deployment.DownloadFileGroup("Media")
    End If
End If
```

ايجاد اسماء المصادر في الاسمبلى-----

Find the Names of Resources in an Assembly

```
Imports System.Reflection
Imports Microsoft.VisualBasic
-----
Dim executingAssembly
executingAssembly = Assembly.GetExecutingAssembly()
Dim resources() As String=executingAssembly.GetManifestResourceNames
```

تحميل الماوس من ملف في الاسمبلى-----

Load a Cursor File from an Assembly

```
Imports System.Reflection
Imports System.Windows.Forms
Imports Microsoft.VisualBasic
-----
Dim newCursor As New _
Cursor(My.Resources.ResourceManager.GetStream("CursorResourceName"))
Me.Cursor = newCursor
```

تحميل الاسمبلى اثناء تشغيل البرنامج-----

Load an Assembly at Run Time

```
Imports System.Reflection
Imports Microsoft.VisualBasic
-----
loadedAssembly = Assembly.LoadFile("C:\Folder\AssemblyName.dll")
```

عمل اتصال بطريقة غير متزامنه-----

Make an Asynchronous Method Call

```
Imports Microsoft.VisualBasic
Imports System.Threading
-----
Private Sub StartBackgroundTask()
    'Execute the Background Task
    BackgroundWorker1.RunWorkerAsync()
End Sub
```

```

    Private Sub BackgroundWorker1_DoWork(ByVal sender As Object,
ByVal e As System.ComponentModel.DoWorkEventArgs) Handles
BackgroundWorker1.DoWork
    ' This method will execute in the background thread created
by the BackgroundWorker componet

End Sub

    Private Sub BackgroundWorker1_RunWorkerCompleted(ByVal sender As
Object, ByVal e As System.ComponentModel.RunWorkerCompletedEventArgs)
Handles BackgroundWorker1.RunWorkerCompleted
    ' This event fires when the DoWork event completes

End Sub

```

-----القراءه و الكتابه فى نافذة الكونسول-----

Read and Write from the Console window

```

Imports Microsoft.VisualBasic
-----
Dim inputFromConsole
Dim outputToConsole
Console.WriteLine("Type in a sentence and hit Enter:")
inputFromConsole = Console.ReadLine()
Console.WriteLine(outputToConsole)

```

-----قراءة وسيطة سطر الأوامر-----

Read a command line argument

```

Imports Microsoft.VisualBasic
-----
For Each argument As String In My.Application.CommandLineArgs

Next

```

-----قراءة النص من منطقة تخزين معزوله-----

Read Text from Isolated Storage

```

Imports System.IO
Imports System.IO.IsolatedStorage
Imports Microsoft.VisualBasic
-----
Dim isolatedStore As IsolatedStorageFile = _
IsolatedStorageFile.GetStore(IsolatedStorageScope.User Or _
IsolatedStorageScope.Assembly, Nothing, Nothing)

Dim isolatedStream As New _
IsolatedStorageFileStream("myfile.txt", FileMode.Open, isolatedStore)

Using reader As New StreamReader(isolatedStream)
    InputText = reader.ReadToEnd
End Using

```

-----استرداد صورته التى هى مورد تطبيق-----

Retrieve an Image that is an Application resource

```

Imports Microsoft.VisualBasic

```

```
-----  
PictureBox1.Image = My.Resources.ImageResource
```

استرداد ملف نصي من مجوعات الاسمبلي-----

Retrieve a text File from an Assembly's Resources

```
Imports System.IO  
Imports System.Reflection  
Imports Microsoft.VisualBasic  
-----  
Dim executingAssembly As Assembly = Assembly.GetExecutingAssembly  
    Dim appStream As Stream = _  
executingAssembly.GetManifestResourceStream("AssemblyName.FileName _  
.txt")  
    Dim textStream As New StreamReader(appStream)  
    Dim text As String = textStream.ReadToEnd
```

حفظ النص في منطقة تخزين معزوله-----

Save Text in isolated storage

```
Imports System.IO  
Imports System.IO.IsolatedStorage  
Imports Microsoft.VisualBasic  
-----  
Dim isolatedStore As IsolatedStorageFile = _  
IsolatedStorageFile.GetStore(IsolatedStorageScope.User Or _  
IsolatedStorageScope.Assembly, Nothing, Nothing)  
    Dim isoStream As New _  
IsolatedStorageFileStream("TestStore.txt", FileMode.Append, _  
FileAccess.Write, isolatedStore)  
  
    Using writer As New StreamWriter(isoStream)  
        writer.WriteLine("The Data")  
    End Using
```

أرسال المفاتيح الى تطبيق-----

Send Keystrokes to an Application

```
Imports System.Windows.Forms  
Imports Microsoft.VisualBasic  
-----  
SendKeys.SendWait("{ENTER}")
```

إيقاف تطبيق-----

Stop an Application

```
Imports System.Diagnostics  
Imports Microsoft.VisualBasic  
-----  
    For Each proc As Process In _  
Process.GetProcessesByName("Untitled - Notepad")  
        proc.CloseMainWindow()  
    Next
```

-----تحديث التطبيق بشكل غير متزامن باستخدام انقر مرة واحدة-----

Update Application Asynchronously using Click Once

```
Imports System.Deployment
Imports Microsoft.VisualBasic
-----
If My.Application.IsNetworkDeployed Then
    My.Application.Deployment.UpdateAsync()
End If
```

-----تحديث التطبيق بشكل متزامن باستخدام انقر مرة واحدة-----

Update Application Synchronously using Click Once

```
Imports Microsoft.VisualBasic
-----
If My.Application.IsNetworkDeployed Then
    My.Application.Deployment.Update()
End If
```

-----كتابة رسالة إلى سجل التطبيق-----

Write a message to the application log

```
Imports System
Imports Microsoft.VisualBasic
-----
My.Application.Log.WriteEntry("Action complete.")
```

-----الكتابة الى ملف نصي-----

Write to a text file

```
Imports System
Imports Microsoft.VisualBasic
-----
My.Computer.FileSystem.WriteAllText("TheFile.txt", _
    "TextContents", False)
```

الفصل الثاني

Collection and Arrays

-----تحويل قائمة مجموعة إلى مجموعة المكتوبة بشدة-----

Convert an array list to a strongly typed Array

```
Imports System.Collections
Imports Microsoft.VisualBasic
-----
Dim newArray() As String = _
CType(existingArrayList.ToArray(GetType(String)), String())
```

-----كتابة قائمة مع العناصر من نوع واحد-----

Write a list with items of a single type

```
Imports Microsoft.VisualBasic
-----
' Backing storage -- a generic list
Dim names As New List(Of String)()
' Add an item to the Collection
names.Add("John")
```

-----إنشاء مجموعه مرتبطه مع نوع معين-----

Create a keyed collection with one item type

```
Imports System.Collections
Imports System.Collections.Generic
Imports Microsoft.VisualBasic
-----
' Backing storage -- a generic dictionary
Dim stateCaps As New Dictionary(Of String, String)

' Add items to the dictionary
stateCaps.Add("WA", "Olympia")
stateCaps.Add("NM", "Santa Fe")
```

-----إنشاء فرز القاموس-----

Create a sorted dictionary

```
Imports Microsoft.VisualBasic
-----
' Backing storage -- a generic dictionary
Dim sortedStudents As New SortedDictionary(Of Integer, String)

' Add items to the dictionary
sortedStudents.Add(1, "Mary Chase")
sortedStudents.Add(2, "Barnaby Williams")
```

-----تحديد نوع المقارن لاستخدامها في إجراءات الفرز-----

Define a type comparer to be used in sorting routines


```
Imports System.Collections.Generic
-----

Public Class IntegerComparer
    Inherits Comparer(Of Integer)

    Public Overrides Function Compare(ByVal param1 As Integer,
ByVal param2 As Integer) As Integer
        If param1 < param2 Then
            ' result = <0 if param1 < param2
            Return -1
        ElseIf param1 > param2 Then
            ' result = >0 if param1 > param2
            Return 1
        Else
            ' result = 0 if param1 = param2
            Return 0
        End If
    End Function
End Class
```

-----فهرس القاموس باستخدام مفتاح-----

Index into a strongly typed dictionary using a key

```
Imports System.Collections.Generic
-----

Dim capitol As String = stateCaps.Item("NM")

Dim oneDimArray() As Integer = {1, 2, 3}
```

-----تهيئة مصفوفة ذات بعد واحد-----

initialize a One - Dimensional Array

```
Imports Microsoft.VisualBasic
-----
' Iterate through a collection
For Each name As String In names

Next
```

-----التعامل من خلال مجموعه-----

Iterate through a collection

```
Imports Microsoft.VisualBasic
-----
' Iterate through a dictionary
For Each capitol As String In stateCaps.Values

Next
```

-----التعامل من خلال القاموس بواسطة المفاتيح-----

Iterate through a Sorted dictionary by a key

```
Imports Microsoft.VisualBasic
-----
' Loop through the items based on key
```

```
For Each rank As Integer In sortedStudents.Keys
Dim student As String = sortedStudents.Item(rank)

Next
```

-----تحديد عنصر في مصفوفة

Locate an element in array

```
Dim index As Integer
index = Array.IndexOf(myArray, "value")
```

-----فرز مصفوفة

Sort an array

```
Imports System.Collections.Generic
-----
Dim animals() As String = {"lion", "turtle", "ostrich"}
Array.Sort(animals)
```

الفصل الثالث

Connectivity

-----تحويل عنوان نسبي الى عنوان مطلق-----

Convert a Relative URL to an absolute URL

```
Imports System
Imports Microsoft.VisualBasic
-----
Dim baseUri As New Uri("http://www.contoso.com/")
Dim relativeUri As New Uri("images/index.htm?id=null")

' Compose absolute Uri using the base and the relative Uri.
Dim absoluteUri As New Uri(baseUri, relativeUri)

Dim absolute As String = absoluteUri.AbsolutePath()
```

-----تحديد ما إذا كانت الشبكة المتاحة-----

Determine if the network is available

```
Imports System
Imports Microsoft.VisualBasic
-----
Dim isAvailable As Boolean
isAvailable = My.Computer.Network.IsAvailable
```

-----تحميل ملف باستخدام بروتوكول انتقال النص المتشعب-----

Download a File using Http

```
My.Computer.Network.DownloadFile("http://www.URLtoDownloadFrom.net",
"C:\filename.html")
```

-----سرد المنفذ التسلسلي-----

Enumerate Serial Ports

```
For Each portName As String In My.Computer.Ports.SerialPortNames
Next
```

-----بينغ كمبيوتر آخر-----

Ping another computer

```
Imports Microsoft.VisualBasic
-----
Dim siteResponds As Boolean = False
siteResponds = My.Computer.Network.Ping("somehost.my.domain")
```

-----قراءة البيانات من المنفذ التسلسلي-----

Read Data from a Serial Port

```
Imports System.Text
Imports System.IO.Ports
-----
Dim buffer As New StringBuilder()
Using comPort As SerialPort =
My.Computer.Ports.OpenSerialPort("COM1")
    Do
        Dim line As String = comPort.ReadLine()
        If line Is Nothing Then
            Exit Do
        Else
            buffer.AppendLine(line)
        End If
    Loop
End Using
```

-----إنشاء رسالة بريد إلكتروني

Create an E-mail message

```
Imports System.Net.Mail
-----

Dim message As New MailMessage("sender@address", _
"from@address", "Subject", "Message Text")

Dim emailClient As New SmtpClient("Email Server Name")

emailClient.Send(message)
```

-----رفع ملف باستخدام شبكات الكمبيوتر الخاص بي

Upload a file using My.computer.Network

```
Imports System
Imports Microsoft.VisualBasic
-----
My.Computer.Network.UploadFile("C:\File.txt", _
"http://www.someserver.com/upload.aspx")
```

-----استخدام منفذ تسلسلي لطلب رقم الهاتف

Use a Serial port to Dial a Phone Number

```
Imports System.IO.Ports
Imports Microsoft.VisualBasic
-----
Using comPort As SerialPort =
My.Computer.Ports.OpenSerialPort("COM1", 2400)
    comPort.DtrEnable = True
    comPort.Write("ATDT 206-555-1000" & vbCrLf)

    ' All data transfer code goes here.

End Using
```

الفصل الرابع Database

إضافة صف جديد إلى جداول البيانات-----

Add a new row to a typed Data Tables

```
' لابد من الاستعانة بقاعدة البيانات الشهيرة '
'Northwind
Dim newRow As NorthwindDataSet.CustomersRow =
    Me.NorthwindDataSet.Customers.NewRow()
newRow.CustomerID = "A124"
newRow.CompanyName = "Acme"
Me.NorthwindDataSet.Customers.Rows.Add(newRow)
```

استدعاء أسلوب تنفيذ الاستعلام من أمر اس كيو ال-----

Call the Execute non query method of a sql command

```
Imports System.Data
Imports System.Data.SqlClient
-----

Dim conn As SqlConnection
Dim cmd As SqlCommand
Dim rowCount As Integer
Dim previousConnectionState As ConnectionState
previousConnectionState = conn.State
Try
    If conn.State = ConnectionState.Closed Then
        conn.Open()
    End If
    rowCount = cmd.ExecuteNonQuery()

Catch ex As Exception
    MsgBox(ex.Message)
Finally
    If previousConnectionState = ConnectionState.Closed Then
        conn.Close()
    End If
End Try
```

استدعاء أسلوب القراءة من أمر اس كيو ال-----

Call the Execute Reader method of a sql command

```
Imports System.Data
Imports System.Data.SqlClient
-----

Dim conn As SqlConnection
Dim cmd As SqlCommand
Dim reader As SqlDataReader
Dim previousConnectionState As ConnectionState
Try
    If conn.State = ConnectionState.Closed Then
        conn.Open()
```

```

End If
reader = cmd.ExecuteReader()
Using reader
    While reader.Read
        ' Process SprocResults datareader here.
        Console.WriteLine(reader.GetValue(0))
    End While
End Using
Finally
    If previousConnectionState = ConnectionState.Closed Then
        conn.Close()
    End If
End Try

```

استدعاء أسلوب القياسى من أمر اس كيو ال-----

Call the Execute Scaler method of a sql command

```

Imports System.Data
Imports System.Data.SqlClient
Imports WindowsApplication1.NORTHWNDDDataSet
-----
Dim conn As SqlConnection
    Dim cmd As SqlCommand
    Dim reader As SqlDataReader
    Dim result As Object
    Dim previousConnectionState As ConnectionState = conn.State
Try
    If conn.State = ConnectionState.Closed Then
        conn.Open()
    End If
    result = cmd.ExecuteScalar()
Finally
    If previousConnectionState = ConnectionState.Closed Then
        conn.Close()
    End If
End Try
Dim row As CustomersRow
    row = CType(CType(Me.CustomersBindingSource.Current, _
DataRowView).Row, CustomersRow)

```

تحويل ربط مصدر في الوقت الراهن إلى صف معين في بيانات الجدول-----

Converts BindingSource.Current to a specific row in a data Table

```

Imports System.Data
Imports System.Data.SqlClient
-----
Dim ds As New DataSet
    Dim customersOrdersRelation As New
DataRelation("CustomersOrders", _
    ds.Customers.Columns("CustomerID"), _
    ds.Orders.Columns("CustomerID"))

ds.Relations.Add(customersOrdersRelation)

```

-----إنشاء علاقة بيانات

Create a Data Relation

```
Imports System.Data
Imports System.Data.SqlClient
-----
Dim ds As New DataSet
Dim OrderTotalColumn As New DataColumn()
OrderTotalColumn.ColumnName = "OrderTotal"
OrderTotalColumn.DataType = GetType(Double)
OrderTotalColumn.Expression = "UnitPrice * Quantity"
ds.Order_Details.Columns.Add(OrderTotalColumn)
```

-----إنشاء عمود التعبير

Create an Expression Column

```
Imports System.Data
Imports System.Data.SqlClient
-----
Dim conn As New SqlConnection
Dim cmd As New SqlCommand
cmd.CommandType = CommandType.Text
cmd.CommandText = "SELECT CustomerID, CompanyName FROM
Customers WHERE CompanyName LIKE @companyName"
cmd.Connection = conn
' Create a SqlParameter for each parameter in the stored
procedure.
Dim companyNameParam As New SqlParameter("@companyName",
"a%")
cmd.Parameters.Add(companyNameParam)
```

-----إنشاء معلمات لامر الاختيار

Create a parameterized SELECT Command

```
Imports System.Data
Imports System.Data.SqlClient
-----
Dim conn As New SqlConnection
Dim cmd As New SqlCommand()
cmd.Connection = conn
cmd.CommandType = CommandType.StoredProcedure
cmd.CommandText = "CustOrderHist"

' Create a SqlParameter for each parameter in the stored
procedure.
Dim customerIDParam As New SqlParameter("@customerID",
"ALFKI")
cmd.Parameters.Add(customerIDParam)
```

-----إنشاء إجراء مخزن

Create a parameterized Stored Procedure

```
Imports System.Data
Imports System.Data.SqlClient
-----
```

```
Dim conn As New SqlConnection()
    conn.ConnectionString = "Data _
Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\Northwind.mdf; _
Integrated Security=True;User Instance=True"
```

عمل اتصال اس كيو ال-----

Creat a local SQL Connection to SQL Express

```
Imports System.Data
Imports System.Data.SqlClient
-----
Dim conn As New SqlConnection()
    conn.ConnectionString = "Data Source=ServerName;Initial _
Catalog=Northwind;Persist Security Info=True;User ID=<user _
name>;Password=<your password>"

Namespace NorthwindDataSetTableAdapters
    Partial Public Class CustomersTableAdapter

End Class
End Namespace
```

إنشاء ملقم يستند إلى خدمة المحاولة و الایجاد-----

Implement Data Try / Catch Blocks

```
Imports System
Imports System.Data
-----
Try
    ' Add your data task here.
Catch concurrencyEx As DBConcurrencyException
    Throw concurrencyEx
Catch constraintEx As ConstraintException
    Throw constraintEx
Catch deletedRowEx As DeletedRowInaccessibleException
    Throw deletedRowEx
Catch duplicateNameEx As DuplicateNameException
    Throw duplicateNameEx
Catch inRowChangingEx As InRowChangingEventException
    Throw inRowChangingEx
Catch invalidConstraintEx As InvalidConstraintException
    Throw invalidConstraintEx
Catch invalidExpressionEx As InvalidExpressionException
    Throw invalidExpressionEx
Catch missingPrimaryEx As MissingPrimaryKeyException
    Throw missingPrimaryEx
Catch noNullEx As NoNullAllowedException
    Throw noNullEx
Catch readOnlyEx As ReadOnlyException
    Throw readOnlyEx
Catch rowNotInTableEx As RowNotInTableException
    Throw rowNotInTableEx
Catch strongTypingEx As StrongTypingException
    Throw strongTypingEx
Catch typedDataSetEx As TypedDataSetGeneratorException
    Throw typedDataSetEx
Catch versionNotFoundEx As VersionNotFoundException
    Throw versionNotFoundEx
```



```

Catch dataEx As DataException
    Throw dataEx
Catch ex As Exception
    Throw ex
Finally
    ' Dispose of any resources you used in the Try block.
End Try

```

-----قراءة الاكسس من البيانات في مجموعة البيانات-----

Read Access Data into a Dataset

```

Imports System
Imports System.Data
Imports System.Data.OleDb
Imports Microsoft.VisualBasic
-----

Dim conn As String = "Provider=Microsoft.Jet.OLEDB.4.0;Data _
Source=AccessFile.mdb;Persist Security Info=False"
Dim cmd As String = "Select * from Topics"
Dim adapter As New OleDbDataAdapter(cmd, conn)
Dim topics As New DataSet
adapter.Fill(topics)

```

-----عودة الخرج لجدول الاستعلام محول-----

Return Output Parameter For a Table Adapter Query

```

Public Function GetNonQueryOutputParameter() As Integer
    Me.GetCustomerCount()
    Return Me._commandCollection(1).Parameters(1).Value
End Function

```

الفصل الخامس

Crystal Report

تغيير اسم الملف واسم قاعدة البيانات-----

Change the Server Name and Database Name

```
Imports System
Imports CrystalDecisions.Shared
Imports CrystalDecisions.CrystalReports.Engine
-----

Dim report As New ReportDocument()
    Dim connection As IConnectionInfo
    Dim oldServerName As String = "OldServer"
    Dim newServerName As String = "NewServer"
    Dim oldDatabaseName As String = "OldDatabase"
    Dim newDatabaseName As String = "NewDatabase"
    Dim userID As String = "MyUserID"
    Dim password As String = "MyPassword"

    report.Load("C:\My Crystal Reports\Report Name.rpt")

    ' Change the server name and database in main reports
    For Each connection In report.DataSourceConnections
        If (String.Compare(connection.ServerName, oldServerName,
True) = 0 _
            And String.Compare(connection.DatabaseName,
oldDatabaseName, True) = 0) Then
            ' SetConnection can also be used to set new logon and
new database table
            report.DataSourceConnections(oldServerName,
oldDatabaseName).SetConnection( _
                newServerName, newDatabaseName, userID, password)
        End If
    Next

    ' Change the server name and database in subreports
    Dim subreport As ReportDocument
    For Each subreport In report.Subreports
        For Each connection In subreport.DataSourceConnections
            If (String.Compare(connection.ServerName,
oldServerName, True) = 0 _
                And String.Compare(connection.DatabaseName,
oldDatabaseName, True) = 0) Then
                ' SetConnection can also be used to set new logon
and new database table
                subreport.DataSourceConnections(oldServerName,
oldDatabaseName).SetConnection( _
                    newServerName, newDatabaseName, userID, password)
            End If
        Next
    Next
Next
```

تصدير تقرير كريستال إلى ملف كريستال-----

Export a Crystal Report to a file

```
Imports System
Imports CrystalDecisions.Shared
Imports CrystalDecisions.CrystalReports.Engine
-----
' You can change other export options like page range by
calling Export method with a ExportOptions object.
Dim report As New ReportDocument()
report.Load("C:\My Crystal Reports\Report Name.rpt")
report.ExportToDisk(ExportFormatType.PortableDocFormat,
"C:\My Crystal Reports\Report Name.rpt")
report.Close()
```

تصدير تقرير كريستال إلى ملف بي دي اف مع خيارات مخصصة لتنسيق التصدير-----

Export a Crystal Report to a pdf file with custom export format options

```
Imports System
Imports CrystalDecisions.Shared
Imports CrystalDecisions.CrystalReports.Engine
-----

Dim report As New ReportDocument()
Dim exportOptions As New ExportOptions()
Dim pdfExportFormatOptions As New PdfRtfWordFormatOptions()
Dim diskDestinationOptions As New
DiskFileDestinationOptions()

' Set the export format and format options
exportOptions.ExportFormatType = _
ExportFormatType.PortableDocFormat
pdfExportFormatOptions.FirstPageNumber = 1
pdfExportFormatOptions.LastPageNumber = 5
pdfExportFormatOptions.UsePageRange = True
exportOptions.ExportFormatOptions = pdfExportFormatOptions

' Set the disk file options.
exportOptions.ExportDestinationType =
ExportDestinationType.DiskFile
diskDestinationOptions.DiskFileName = "C:\My Crystal _
Reports\Report Name.pdf"
exportOptions.DestinationOptions = diskDestinationOptions

report.Load("C:\My Crystal Reports\Report Name.rpt")
report.Export(exportOptions)
report.Close()
```

طباعة تقرير كريستال-----

Print a Crystal Report

```
Imports System
Imports CrystalDecisions.Shared
Imports CrystalDecisions.CrystalReports.Engine
-----
```

```

' You can change more print options via PrintOptions property
of ReportDocument
Dim report As New ReportDocument()
report.Load("C:\My Crystal Reports\Report Name.rpt")
report.PrintToPrinter(1, True, 0, 0)
report.Close()

```

-----تعيين قيمة افتراضيه إلى تقرير كريستال

Set a Discrete Parameter Value to a Crystal Report

```

Imports System
Imports CrystalDecisions.Shared
Imports CrystalDecisions.CrystalReports.Engine
-----

Dim report As New ReportDocument()
report.Load("C:\My Crystal Reports\Report Name.rpt")

Dim parameterField As ParameterField
parameterField = report.ParameterFields("Parameter1")

' If the parameter is from a subreport, specify the subreport
name
' parameterField = report.ParameterFields("Parameter1",
subreportName.rpt)

' Add all parameters value here
parameterField.CurrentValues.AddValue(100.0)

```

-----تعيين مجال افتراضي إلى تقرير كريستال

Set a Range Parameter Value to a Crystal Report

```

Imports System
Imports CrystalDecisions.Shared
Imports CrystalDecisions.CrystalReports.Engine
-----

Dim report As New ReportDocument()
report.Load("C:\My Crystal Reports\Report Name.rpt")

Dim parameterField As ParameterField
parameterField = report.ParameterFields("Parameter1")

' If the parameter is from a subreport, specify the subreport
name
' parameterField = report.ParameterFields("Parameter1",
subreportName.rpt)

' Add all parameters value here
parameterField.CurrentValues.AddRange(0, 100,
RangeBoundType.BoundInclusive, RangeBoundType.BoundInclusive)

```

تعيين لقاعدة بيانات تسجيل الدخول إلى تقرير كريستال والتقارير الفرعية لها-----

Set Database Logon to a Crystal Report and its subreports

```
Imports System
Imports CrystalDecisions.Shared
Imports CrystalDecisions.CrystalReports.Engine
-----

Dim report As New ReportDocument()
Dim connection As IConnectionInfo
Dim serverName1 As String = "Server1"
Dim userID As String = "MyUserID"
Dim password As String = "MyPassword"

report.Load("C:\My Crystal Reports\Report Name.rpt")

' Set Database Logon to main report
For Each connection In report.DataSourceConnections
    Select Case connection.ServerName
        Case serverName1
            connection.SetLogon(userID, password)
    End Select
Next

' Set Database Logon to subreport
Dim subreport As ReportDocument
For Each subreport In report.Subreports
    For Each connection In subreport.DataSourceConnections
        Select Case connection.ServerName
            Case serverName1
                connection.SetLogon(userID, password)
        End Select
    Next
Next
Next
```

الفصل السادس

Data Types

-----تحويل صفيف من البايت إلى سلسلة-----

Convert an Array of bytes into a String

```
Imports System.Text
Imports Microsoft.VisualBasic
-----

Private Function ConvertByteArrayToString(ByVal byteArray As _
Byte()) As String
    Dim enc As Encoding = Encoding.UTF8
    Dim text As String = enc.GetString(byteArray)
    Return text
End Function
```

-----تحويل رقم إلى سلسلة ستعشري-----

Convert a Number into a Hexadecimal String

```
Imports System
Imports Microsoft.VisualBasic
-----

Dim hexString As String = Hex(48)
MsgBox(hexString)
```

-----تحويل رقم إلى سلسلة أوكتال-----

Convert a Number into a Octal String

```
Imports System
Imports Microsoft.VisualBasic
-----

Dim octal As String = Oct(48)
MsgBox(octal)
```

-----تحويل سلسلة إلى مصفوفة بايت-----

Convert a String to a Byte Array

```
Imports System.Text
-----

Dim bytes As Byte() = _
System.Text.Encoding.Unicode.GetBytes("StringToConvert")
```

-----تحديد فترة بين تاريخين-----

Determine the span between Two Dates

```
Imports System
Imports Microsoft.VisualBasic
-----

Dim oldDate As Date = #1/1/2010#
```

```

Dim newDate As Date = Now
Dim differenceInDays As Long = DateDiff(DateInterval.Day, _
oldDate, newDate)
Dim spanFromDays As TimeSpan = New _
TimeSpan(CInt(differenceInDays), 0, 0, 0)

```

----- تحديد فترة بين وقتين

Determine the span between Two times

```

Imports System
Imports Microsoft.VisualBasic
-----

```

```

Dim oldTime As Date = Today
Dim newTime As Date = Now
'you can also determine the difference in times in other
units.
Dim differenceInSeconds As Long =
DateDiff(DateInterval.Second, oldTime, newTime)
Dim spanFromSeconds As TimeSpan = New TimeSpan(0, 0, _
CInt(differenceInSeconds))

```

----- توليد سلسلة حرفية متعددة

Generate Multiple String Literals

```

Imports Microsoft.VisualBasic
-----

```

```

Dim longString As String
longString = _
    "This is the first line of my string." &
ControlChars.CrLf & _
    "This is the second line of my string." &
ControlChars.CrLf & _
    "This is the third line of my string."

```

----- إزالة أجزاء من سلسلة حرفية

Remove Parts of a String

```

Imports Microsoft.VisualBasic
-----

```

```

Dim withParts As String = "Books and Chapters and Pages"
Dim withoutParts As String = Replace(withParts, "and ", "")

```