

Export và Import ra tập tin text từ Access (VB)

Hiện nay các bạn yêu thích lập trình sử dụng Access là nguồn chứa dữ liệu khá phổ biến vì đơn giản, dễ quản trị và đáp ứng được yêu cầu công việc. Hôm nay chúng tôi xin giới thiệu một đoạn code để export và import ra tập tin text từ Access (VB)

Export Text (Flat file) từ Access Ms-Access

Option Explicit

Public Sub Export_Table_2_TextFile()

On Error GoTo LocalErrorHandler

Dim dbCompany As Database

Dim rsGeneral As Recordset

Dim ExpGeneral As PubExpGeneral

Dim blnTab_Text As Boolean

Dim FullName As String

Dim FileHandle As Byte

Dim strFileToExport As String

Dim chkFileExist As String

'Give Path with File name

FullName = [E:\General](#) ' Thu mục chứa dữ liệu, bạn có thể thay đổi theo nhu cầu của mình

blnTab_Text = False

Set dbCompany = OpenDatabase(FullName)

'Ví dụ tên bảng là Company

Set rsGeneral = dbCompany.OpenRecordset(Company, dbOpenTable)

With ExpGeneral

.EmpNumber = No.

.EmpName = Name

.EmpAddress = Address

.EmpCity = City

Sử dụng TAB hoặc dấu phẩy

If blnTab_Text Then

.Delimiter1 = Chr(9)

.Delimiter2 = Chr(9)

.Delimiter3 = Chr(9)

Else

.Delimiter1 = Chr(44)

.Delimiter2 = Chr(44)

.Delimiter3 = Chr(44)

End If

.CRLF = vbCrLf

End With

FileHandle = FreeFile

'Tên tập tin

strFileToExport = C:\Exported.txt

chkFileExist = Dir(strFileToExport)

If chkFileExist <> Then

```

Kill strFileToExport
End If
Open strFileToExport For Random As FileHandle Len = Len(ExpGeneral)
Put FileHandle, , ExpGeneral
Do Until rsGeneral.EOF
With ExpGeneral
    .EmpNumber = rsGeneral(EmpNo)
    .EmpName = rsGeneral(EmpName)
    .EmpAddress = rsGeneral(EmpAddress)
    .EmpCity = rsGeneral(EmpCity)
End With
Put FileHandle, , ExpGeneral
rsGeneral.MoveNext
Loop
rsGeneral.Close
Set rsGeneral = Nothing
Close FileHandle
Exit Sub

```

```

LocalErrorHandler:
MsgBox Error Occured : & Err.Description, , Error

```

```

End Sub

```

```

'Import Text vào Ms-Access

```

```

Public Sub Import_TextFile_2_Table()
    On Error GoTo LocalErrorHandler
    Dim dbCompany As Database
    Dim rsGeneral As Recordset
    Dim FullName As String
    Dim FileHandle As Byte
    Dim ImportRecord As String
    Dim flnName As String
    Dim RowPosition As Double
    Dim EmpNumber As String
    Dim EmpName As String
    Dim EmpAddress As String
    Dim EmpCity As String
    Dim Delimiter As String

    flnName = C:\Exported.txt
    Delimiter = ,
    FileHandle = FreeFile
    Open flnName For Input As FileHandle
    Line Input #FileHandle, ImportRecord
    FullName = C:\General
    Set dbCompany = OpenDatabase(FullName)
    Set rsGeneral = dbCompany.OpenRecordset(Company, dbOpenDynaset)
    Do Until EOF(FileHandle)
    Line Input #FileHandle, ImportRecord
    RowPosition = RowPosition + 1
    EmpNumber = Trim(Mid(ImportRecord, 1, InStr(1, ImportRecord, Delimiter, 1) - 1))
    EmpName = Trim(Mid(ImportRecord, 7, 10))

```

```

EmpAddress = Trim(Mid(ImportRecord, 18, 30))
EmpCity = Trim(Mid(ImportRecord, 49))
rsGeneral.AddNew
rsGeneral(EmpNo) = EmpNumber
rsGeneral(EmpName) = EmpName
rsGeneral(EmpAddress) = EmpAddress
rsGeneral(EmpCity) = EmpCity
rsGeneral.Update
Loop
Close FileHandle
rsGeneral.Close
Set rsGeneral = Nothing
dbCompany.Close
Set dbCompany = Nothing
Exit Sub
LocalErrorHandler:
MsgBox Error Occured : & Err.Description, , Error
End Sub

```

Kỹ thuật Subclass Listbox trong Visualbasic

Bài viết này sẽ giúp bạn hiểu kỹ thuật subclassing trong VisualBasic. Bạn có thể áp dụng cho các đối tượng khác khi lập trình trong VB

Windows gửi thông điệp là một hằng số tới các form và các control của VB để báo cho chúng biết vị trí chuột ở đâu, khi nào thì cần vẽ lại, phím nào đang được nhấn và nhiều thông điệp khác. Kỹ thuật subclassing là để xử lý chặn những thông điệp này trước khi chúng đến được các form và control. Bằng cách chặn các thông điệp này và xử lý "vài thứ" trước khi chúng đến đích, chúng ta có thể có các tính năng riêng (như tự vẽ lại các control theo ý riêng).

Subclassing là một kỹ thuật tinh vi, chỉ cần một lỗi nhỏ (ví dụ như : do bạn giải phóng tài nguyên không tốt dẫn đến việc thất thoát tài nguyên của hệ thống) là có thể dẫn đến việc hệ thống của bạn bị thiếu tài nguyên làm cho hệ thống hoạt động không còn tốt nữa (chậm đi), nặng hơn là VB bị shut down, thậm chí treo máy. Tuy nhiên nói điều này là để bạn ý thức được vấn đề chứ bạn cũng không nên quá lo ngại về nó. Và thêm 1 chú ý là bạn cũng không nên bấm nút stop của VB khi chương trình đang chạy mà bạn nên đóng form 1 cách thông thường (bấm nút close) để thực hiện tốt việc giải phóng tài nguyên.

Subclassing the Main Window:

Chúng ta bắt đầu thực hiện kỹ thuật subclassing bằng cách bạn mở 1 project mới và thêm 1 module vào project (project/add module/open). Bây giờ bạn đã có Form1 và Module1 trong project.

Bạn mở Module1 ra và copy, paste đoạn code sau vào :

```

Public Const GWL_WNDPROC = (-4)
Public oldWindowProc as Long

Public Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA" ( _
ByVal hwnd As Long, _
ByVal nIndex As Long, _
ByVal dwNewLong As Long) As Long

```

Đây là một hàm API của Windows cho phép bạn thay đổi thuộc tính của 1 cửa sổ (hay control - từ bây giờ chúng ta coi như control cũng là một window), trong trường hợp của chúng ta là thay đổi hàm WinProc (hàm Winproc là hàm mà các window dùng để xử lý các thông điệp do hệ thống

(hệ điều hành Windows) gửi đến).

hwnd - tham số này có kiểu là long integer dùng để xác định 1 cửa sổ (form) hay 1 control (bạn có thể coi nó như bảng số xe dùng để xác định tính duy nhất của 1 xe vậy).

nIndex - tham số này cũng có kiểu là long integer dùng để xác định "cần thay đổi cái gì" trong hàm SetWindowLong nói trên (bạn có thể tham khảo trong bộ MSDN), trong trường hợp của chúng ta nIndex có giá trị là GWL_WNDPROC (vì chúng ta cần xử lý hàm WinProc mà).

dwNewLong - hàm này có kiểu long integer dùng để chỉ ra địa chỉ của thủ tục mới mà chúng ta cần xử lý.

Hàm WinProc mới phải có các tham số giống hệt các tham số của hàm WinProc bị thay thế. Bạn cũng phải chú ý là bạn phải gửi trả các thông điệp mà bạn không xử lý cho hàm WinProc mặc định xử lý. Bạn tiếp tục copy và dán đoạn mã sau vào Module1 :

```
Private Declare Function CallWindowProc Lib "user32" Alias "CallWindowProcA" ( _  
ByVal lpPrevWndFunc As Long, _  
ByVal hwnd As Long, _  
ByVal Msg As Long, _  
ByVal wParam As Long, _  
ByVal lParam As Long) As Long
```

```
Public Function NewWindowProc( _  
ByVal hWnd As Long, _  
ByVal uMsg As Long, _  
ByVal wParam As Long, _  
ByVal lParam As Long) As Long
```

```
Debug.Print "&H" & Hex(uMsg), wParam, lParam  
NewWindowProc = CallWindowProc(oldWindowProc, hWnd, uMsg, wParam, lParam)  
End Function
```

CallWindowProc dùng để gọi hàm WinProc mặc định ra xử lý, hàm NewWindowProc là hàm thay thế cho hàm WinProc. Hàm NewWindowProc không làm bất cứ việc gì ngoại trừ việc in ra cửa sổ Debug xem thông điệp gì được gửi đến cho cửa sổ này (cửa sổ bị subclassing). Hàm NewWindowProc sau đó gọi hàm WinProc mặc định để xử lý thông điệp 1 cách bình thường (biến oldWindowProc dùng để lưu địa chỉ hàm WinProc mặc định). Tham số mà hệ thống gửi cho hàm NewWindowProc là : hWnd - handle của cửa sổ sẽ nhận thông điệp; uMsg - thông điệp được gửi; và 2 tham số còn lại (wParam và lParam) mang thông tin của thông điệp, phụ thuộc vào thông điệp được gửi.

Bây giờ bạn có thể chạy project được, nhưng chưa có chuyện gì xảy ra cả, cửa sổ (form) của bạn chưa bị subclass. Một lần nữa xin nhắc lại là bạn không nên bấm vào nút stop để dừng chương trình và bạn cũng nên lưu project lại trước khi chạy.

Để thực hiện subclass cửa sổ (form) của bạn, bạn double vào form và copy, paste đoạn code sau vào :

```
Private Sub Form_Load()  
'Subclass the window  
oldWindowProc = SetWindowLong(Me.hWnd, GWL_WNDPROC, AddressOf NewWindowProc)  
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)  
'Unsubclass (return the original window process)
```

```
SetWindowLong Me.hWnd, GWL_WNDPROC, oldWindowProc
End Sub
```

Bây giờ thì ok, form của bạn đã bị subclass ! Bạn thử chạy project và xem điều gì xảy ra ? Cửa sổ Debug của bạn sẽ tràn ngập những thông tin về thông điệp mà hệ thống đã gửi cho form của bạn, bạn thử di chuyển chuột, thay đổi kích thước form ... mà xem. (Hàm AddressOf dùng để lấy địa chỉ của 1 hàm).

How to put a background image into a Listbox:

Bước 1 : Kéo 1 ListBox và 1 Image control vào Form1.

Bước 2 : Thêm 1 số mục (item) vào Listbox (Mục list trong ListBox control).

Bước 3 : Thêm 1 picture vào Image1 (picture này bạn sẽ dùng làm background cho ListBox).

Bước 4 : Mở Module1 ra và dán đoạn code sau vào :

```
Public gBGBrush As Long
```

```
Public Declare Function CreatePatternBrush Lib "gdi32" ( _
ByVal hBitmap As Long) As Long
```

```
Public Declare Function DeleteObject Lib "gdi32" ( _
ByVal hObject As Long) As Long
```

```
Private Declare Function SetBkMode Lib "gdi32" ( _
ByVal hdc As Long, _
ByVal nBkMode As Long) As Long
```

```
Private Const WM_CTLCOLORLISTBOX = &H134
```

Các hàm dùng cho việc vẽ nền cho ListBox, bạn có thể xem thêm trong bộ MSDN.

Bước 5 : Thay đoạn code trong Form_Load và Form_Unload lúc này bằng đoạn code mới như sau :

```
Private Sub Form_Load()
Image1.Visible = False
gBGBrush = CreatePatternBrush(Image1.Picture.Handle)
'Subclass the window
oldWindowProc = SetWindowLong(Me.hWnd, GWL_WNDPROC, AddressOf NewWindowProc)
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
'Unsubclass (return the original window process)
SetWindowLong Me.hWnd, GWL_WNDPROC, oldWindowProc
DeleteObject gBGBrush
End Sub
```

Bước 6 : Viết lại hàm NewWindowProc trong Module 1 để làm việc mà chúng ta muốn (lại copy và paste).

```
Public Function NewWindowProc( _
```

```
ByVal hWnd As Long, _  
ByVal uMsg As Long, _  
ByVal wParam As Long, _  
ByVal lParam As Long) As Long
```

```
Debug.Print "&H" & Hex(uMsg), wParam, lParam
```

```
If uMsg = WM_CTLCOLORLISTBOX And gBGBrush <> 0 Then  
'Make the words print transparently  
SetBkMode wParam, 1  
'allow the original process to set text color, etc. from the lbx properties.  
CallWindowProc oldWindowProc, hWnd, uMsg, wParam, lParam  
'Return our custom brush instead of the default one  
NewWindowProc = gBGBrush  
Else  
NewWindowProc = CallWindowProc(oldWindowProc, hWnd, uMsg, wParam, lParam)  
End If  
End Function
```

Bước 7 :Yeah ! Bạn lưu project lại và chạy thử xem.

Bây giờ ListBox của bạn đã có background phải không ? Tại sao ta làm được như vậy ? Có vài điểm cần lưu ý như sau :

Điều 1 : Chúng ta chặn thông điệp WM_CTLCOLORLISTBOX để xử lý. Thông điệp này được gửi cho parent window (cửa sổ cha mẹ, cửa sổ cấp cao hơn chứa ListBox) của ListBox trước khi hệ thống vẽ list box. Lúc này wParam mang giá trị là handle DC (devie context) dùng để vẽ list box, lParam mang giá trị là handle của list box cần vẽ. Và một điều vô cùng quan trọng là giá trị trả về của hàm WindowProc lúc này, giá trị này sẽ được hệ thống dùng để vẽ nền cho list box, do đó trong NewWindowProc chúng ta cho NewWindowProc "chỉ" đến handle của gBGBrush (NewWindowProc = gBGBrush) và trước đó chúng ta đã tạo ra gBGBrush bằng cách : gBGBrush = CreatePatternBrush(Image1.Picture.Handle). Và tất cả các công việc khác vẫn được xử lý bình thường bằng cách chúng ta gọi hàm : CallWindowProc.

Điều 2 : Chúng ta phải giải phóng tài nguyên hệ thống bằng cách, trong Form_Unload : DeleteObject gBGBrush. Nếu chúng ta không làm việc này sẽ dẫn đến hiện tượng memory leak - làm giảm tài nguyên hệ thống, gây hại cho hệ thống.

Bài này chỉ demo việc subclass 1 list box, tuy nhiên bạn có thể áp dụng kỹ thuật này để subclass mọi control mà bạn muốn, chỉ đơn giản thay đổi, các tham số cho phù hợp như : hWnd - handle của cửa sổ cần subclass, xử lý trong hàm NewWindowProc cho phù hợp với từng control, từng thông điệp.

Chạy tập tin MPEG trong VB6

Chúng ta sẽ xây dựng một Class để điều khiển các tập tin định dạng theo MPEG. Bạn có thể thao các tác vụ cơ bản và các thuộc tính của tập tin MPEG bằng Class này.

```
Private Declare Function mciGetErrorString Lib "winmm.dll" Alias "mciGetErrorStringA" (ByVal dwError As Long, ByVal lpstrBuffer As String, ByVal uLength As Long) As Long  
Private Declare Function GetShortPathName Lib "kernel32" Alias "GetShortPathNameA" (ByVal lpzLongPath As String, ByVal lpzShortPath As String, ByVal cchBuffer As Long) As Long  
Private Declare Function mciSendString Lib "winmm.dll" Alias "mciSendStringA" (ByVal lpstrCommand As String, ByVal lpstrReturnString As String, ByVal uReturnLength As Long, ByVal hwndCallback As Long) As Long
```

```

Const m_def_FileName = ""
Dim m_FileName As String

'MappingInfo=UserControl,UserControl,-1,Enabled
Public Property Get Enabled() As Boolean
Enabled = UserControl.Enabled
End Property

Public Property Let Enabled(ByVal New_Enabled As Boolean)
UserControl.Enabled() = New_Enabled
PropertyChanged "Enabled"
End Property

'MemberInfo=13,0,0,
Public Property Get FileName() As String
FileName = m_FileName
End Property

Public Property Let FileName(ByVal New_FileName As String)
m_FileName = New_FileName
PropertyChanged "FileName"
End Property

'Khởi động các thuộc tính của đối tượng
Private Sub UserControl_InitProperties()
m_FileName = m_def_FileName
End Sub

'Đọc thuộc tính đã lưu giữ
Private Sub UserControl_ReadProperties(PropBag As PropertyBag)
UserControl.Enabled = PropBag.ReadProperty("Enabled", True)
m_FileName = PropBag.ReadProperty("FileName", m_def_FileName)
End Sub

Private Sub UserControl_Terminate()
mmStop
End Sub

Private Sub UserControl_WriteProperties(PropBag As PropertyBag)
Call PropBag.WriteProperty("Enabled", UserControl.Enabled, True)
Call PropBag.WriteProperty("FileName", m_FileName, m_def_FileName)
End Sub

Public Function IsPlaying() As Boolean
Static s As String * 30
mciSendString "status MPEGPlay mode", s, Len(s), 0
IsPlaying = (Mid$(s, 1, 7) = "playing")
End Function

Public Function mmPlay()
Dim cmdToDo As String * 255
Dim dwReturn As Long
Dim ret As String * 128

Dim tmp As String * 255

```

```

Dim lenShort As Long
Dim ShortPathAndFie As String

If Dir(FileName) = "" Then
    mmOpen = "Error with input file"
    Exit Function
End If
lenShort = GetShortPathName(FileName, tmp, 255)
ShortPathAndFie = Left$(tmp, lenShort)
glo_hWnd = hWnd
cmdToDo = "open " & ShortPathAndFie & " type MPEGVideo Alias MPEGPlay Parent " &
UserControl.hWnd & " Style 1073741824"
dwReturn = mciSendString(cmdToDo, 0&, 0&, 0&)

If dwReturn <> 0 Then 'not success
    mciGetErrorString dwReturn, ret, 128
    mmOpen = ret
    MsgBox ret, vbCritical
Exit Function
End If

mmPlay = "Success"
mciSendString "play MPEGPlay", 0, 0, 0
End Function

Public Function mmPause()
    mciSendString "pause MPEGPlay", 0, 0, 0
End Function

Public Function mmStop() As String
    mciSendString "stop MPEGPlay", 0, 0, 0
    mciSendString "close MPEGPlay", 0, 0, 0
End Function

Public Function PositionInSec()
    Static s As String * 30
    mciSendString "set MPEGPlay time format milliseconds", 0, 0, 0
    mciSendString "status MPEGPlay position", s, Len(s), 0
    PositionInSec = Round(Mid$(s, 1, Len(s)) / 1000)
End Function

Public Function Position()
    Static s As String * 30
    mciSendString "set MPEGPlay time format milliseconds", 0, 0, 0
    mciSendString "status MPEGPlay position", s, Len(s), 0
    sec = Round(Mid$(s, 1, Len(s)) / 1000)
    If sec < 60 Then Position = "0:" & Format(sec, "00")
    If sec > 59 Then
        mins = Int(sec / 60)
        sec = sec - (mins * 60)
        Position = Format(mins, "00") & ":" & Format(sec, "00")
    End If
End Function

Public Function LengthInSec()
    Static s As String * 30

```



```

mciSendString "set MPEGPlay time format milliseconds", 0, 0, 0
mciSendString "status MPEGPlay length", s, Len(s), 0
LengthInSec = Round(Val(Mid$(s, 1, Len(s))) / 1000) 'Round(CInt(Mid$(s, 1, Len(s))) / 1000)
End Function

```

```

Public Function Length()
    Static s As String * 30
    mciSendString "set MPEGPlay time format milliseconds", 0, 0, 0
    mciSendString "status MPEGPlay length", s, Len(s), 0
    sec = Round(Val(Mid$(s, 1, Len(s))) / 1000) 'Round(CInt(Mid$(s, 1, Len(s))) / 1000)
    If sec < 60 Then Length = "0:" & Format(sec, "00")
    If sec > 59 Then
        mins = Int(sec / 60)
        sec = sec - (mins * 60)
        Length = Format(mins, "00") & ":" & Format(sec, "00")
    End If
End Function

```

```

Public Function About()
    frmCtlAbout.Show vbModal, Me
End Function

```

```

Public Function SeekTo(Second)
    mciSendString "set MPEGPlay time format milliseconds", 0, 0, 0
    If IsPlaying = True Then mciSendString "play MPEGPlay from " & Second, 0, 0, 0
    If IsPlaying = False Then mciSendString "seek MPEGPlay to " & Second, 0, 0, 0
End Function

```

Truyền giá trị qua trang khác với phương thức Server.Transfer (ASP.NET)

ASP.NET validation controls rất hữu dụng để kiểm tra giá trị người dùng nhập vào khi posts back trên cùng một trang. Nhưng làm thế nào để sử dụng trong các trang khác ?

Ví dụ bạn có một trang, WebPostAwayA1.aspx với 2 textbox control, bạn sử dụng 2 RequiredFieldValidator control. Bạn muốn chuyển dữ liệu sang một trang thứ 2, WebPostAway2.aspx, chỉ khi các textbox đã được nhập giá trị

WebPostAwayA1.aspx:

```

<body>
  <form runat="server">
    <table border="0">
      <tbody>
        <tr>
          <td>
            <asp:Label id="Label1" runat="server">First Name:</asp:Label>
          </td>
          <td>
            <asp:TextBox id="txtFirstName" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator id="rfvFirstName" runat="server"
              ErrorMessage="First name is required."
              ControlToValidate="txtFirstName"></asp:RequiredFieldValidator>
          </td>
        </tr>
      </tbody>
    </table>
  </form>
</body>

```

```

        </td>
    </tr>
    <tr>
        <td>
            <asp:Label id="Label2" runat="server">Last Name:</asp:Label>
        </td>
        <td>
            <asp:TextBox id="txtLastName" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator id="rfvLastName" runat="server"
            ErrorMessage="Last name is required."
            ControlToValidate="txtLastName"></asp:RequiredFieldValidator>
        </td>
    </tr>
    <tr>
        <td>
            <asp:Button id="cmdPost" onclick="cmdPost_Click"
            runat="server" Text="Submit"></asp:Button>
        </td>
    </tr>
</tbody>
</table>
</form>
</body>

```

WebPostAwayA1.aspx sẽ chuyển đến WebPostAway2.aspx nếu các giá trị được nhập vào 2 textbox:

```

void cmdPost_Click(Object src, EventArgs e ) {
    if (Page.IsValid) {
        Response.Redirect("WebPostAway2.aspx");
    }
}

```

Vấn đề là dòng code trên sẽ không truyền giá trị khi redirect sang trang WebPostAway2. Chúng ta sử dụng Server.Transfer

```

void cmdPost_Click(Object src, EventArgs e ) {
    if (Page.IsValid) {
        Server.Transfer("WebPostAway2.aspx");
    }
}

```

Trong ASP.NET, Server Transfer mặc định sẽ không truyền form, query string collections từ một post back. Mặc dù vậy bạn có thể định tham số thứ 2 của phương thức Transfer thành True để các giá trị trên có thể được truyền sang một trang mới.

```

void cmdPost_Click(Object src, EventArgs e ) {
    if (Page.IsValid) {

```

```
Server.Transfer("WebPostAway2.aspx", true);  
}  
}
```

Tạo Font và Xoay Chữ

Nhiều Control trong VB cung cấp cho bạn thuộc tính Font cho phép bạn thay đổi tên Font, cỡ Font, in đậm, in nghiêng...v.v.. Hàm CreateFont cho phép bạn làm được nhiều hơn thế nữa. Có thể xoay chữ theo một góc bất kì để tạo ra những hiệu ứng thú vị.

Với 14 tham số, có vẻ như CreateFont là một hàm phức tạp nhưng thực sự nó cũng khá đơn giản.

```
Private Declare Function CreateFont Lib "gdi32" Alias "CreateFontA" (  
    ByVal font_height As Long,  
    ByVal font_width As Long,  
    ByVal escapement As Long,  
    ByVal orientation As Long,  
    ByVal weight As Long,  
    ByVal italic As Long,  
    ByVal underscore As Long,  
    ByVal strikeout As Long,  
    ByVal character_set As Long,  
    ByVal output_precision As Long,  
    ByVal clipping_precision As Long,  
    ByVal quality As Long,  
    ByVal pitch_and_family As Long,  
    ByVal face_name As String) As Long
```

Bạn có thể đặt nhiều trong số các tham số này bằng 0 để sử dụng các giá trị mặc định của Windows. Bạn chỉ cần nắm một số các tham số là đủ, các tham số khác các bạn có thể tham khảo thêm trên MSDN:

- font_height: Giá trị này là chiều cao của Font tính theo đơn vị pixels.
- font_width: Bề rộng của Font theo đơn vị Pixel. Nếu bằng 0 thì Windows sẽ dùng giá trị mặc định phù hợp với chiều cao của Font.
- escapement: Đây là góc quay của Font so với đường nằm ngang. Tính theo độ.
- orientation: Đây là hướng của chữ, nhưng Windows xem hướng của chữ là góc quay của chữ nên đối số này sẽ bị bỏ qua.
- weight: Chỉ định độ đậm của Font, là một số giữa 0 -> 900. Bạn chọn 0 để sử dụng giá trị mặc định, 400 là bình thường, và 700 là đậm.
- italic: Nếu giá trị này khác 0, chữ sẽ nghiêng.
- underscore: Nếu giá trị này khác 0, chữ sẽ được gạch dưới.
- strikeout: Nếu giá trị này khác 0, chữ sẽ được gạch giữa.
- character_set: Giá trị này chỉ định các xác lập kí tự ví dụ như Russian, Greek, and Arabic. Thông thường bạn nên đặt giá trị này bằng ANSI_CHARSET (0).
- face_name: Tên của Font, ví dụ: "Times New Roman" hoặc "Courier New"...v.v..

Hàm CreateFont trả về handle của Font vừa mới được tạo. Để có thể sử dụng Font này bạn phải dùng hàm SelectObject API để chọn Font mới cho một đối tượng như Form hoặc PictureBox...v.v... SelectObject sẽ trả về handle của Font cũ. Chương trình có thể "vẽ chữ" dùng Font đã chọn. Khi chạy xong nên dùng hàm SelectObject để trả về giá trị mặc định và dùng hàm DeleteObject để xóa Font và giải phóng bộ nhớ.

Bây giờ hãy xem một ví dụ.

```

Private Declare Function SelectObject Lib "gdi32" (ByVal hdc As Long, ByVal hObject
As Long) As Long
Private Declare Function CreateFont Lib "gdi32" Alias "CreateFontA" (ByVal H As
Long, ByVal W As Long, ByVal E As Long, ByVal O As Long, ByVal W As Long, ByVal
l As Long, ByVal u As Long, ByVal S As Long, ByVal C As Long, ByVal OP As Long,
ByVal CP As Long, ByVal Q As Long, ByVal PAF As Long, ByVal F As String) As Long
Private Declare Function DeleteObject Lib "gdi32" (ByVal hObject As Long) As Long
Private Const PROOF_QUALITY = 2

Private Sub Form_Load()
Me.AutoRedraw = True
Angle = 45 'Góc quay của chữ
escapement = CLng(Angle * 10) Mod 3600
If escapement < 0 Then escapement = escapement + 3600
If escapement = 0 Then escapement = 3600
rfont = CreateFont(20, 20, escapement, escapement, 700, 0, 0, 0, 0, 0, 0,
PROOF_QUALITY, 0, "Verdana" + Chr(0))
current = SelectObject(Me.hdc, rfont) 'Chọn font vừa tạo
Me.CurrentX = 700
Me.CurrentY = 1100
Me.Print "Text"
SelectObject Me.hdc, curent 'Trả về Font mặc định
DeleteObject (rfont) 'Giải phóng bộ nhớ
End Sub

```

Dãy số và dãy chuỗi ngẫu nhiên (PHP)

Đây là các chức năng hay gặp khi bạn thực hiện các website, cần các dãy số hay chuỗi bất kỳ để cung cấp cho người sử dụng để active account hoặc generate các password...

Số ngẫu nhiên

<?php

```

function randomdigit($digits) {
    static $startseed = 0;
    if (!$startseed) {
        $startseed = (double)microtime()*getrandmax();
        srand($startseed);
    }
    $range = 8;
    $start = 1;
    $i = 1;
    while ($i<$digits) {
        $range = $range . 9;
        $start = $start . 0;
        $i++;
    }
    return (rand()%$range+$start);
}
?>

```

Dãy chuỗi ngẫu nhiên

<?php

```

function randomPass($length = 6) {

    // all the chars we want to use
    $all = explode( " ",
        "a b c d e f g h i j k l m n o p q r s t u v w x y z "
        . "A B C D E F G H I J K L M N O P Q R S T U V W X Y Z "
        . "0 1 2 3 4 5 6 7 8 9");

    for($i=0;$i<$length;$i++) {
        srand((double)microtime()*1000000);
        $randy = rand(0, 61);
        $pass .= $all[$randy];
    }

    return $pass;
}
?>

```

PHP và Java

Một điều khá thú vị các chức năng của PHP là khả năng sử dụng các phương thức của các đối tượng Java, mở ra khả năng tích hợp các ứng dụng PHP của bạn với các ứng dụng Java. Mời các bạn tham khảo các thực hiện.

Để sử dụng bạn cần cài đặt Java Virtual Machine (JVM). Nếu bạn sử dụng JDKs từ Sun, Kaffe, IBM, hay Blackdown, bạn sẽ lợi về mặt tốc độ. Khi bạn cài đặt PHP, bạn cần thêm --with-java để thêm các thư viện dành cho java, hoặc sửa tập tin php.ini

```

[Java]
java.library.path=/path/to/library
java.class.path=/classpath/
extension_dir=/path/to/extensions
extension=libphp_java.so
bạn cần đưa đường dẫn Java chính xác.
Đoạn script sau sẽ nối và in ra màn hình bằng các thuộc tính của Java
<?
$system = new Java("java.lang.System");
echo "<P>Java version = " . $system->getProperty("java.version") . "<br>";
echo "Java vendor = " . $system->getProperty("java.vendor") . "</p>";
?>

```

Nếu bạn có hiểu biết về Java, bạn sẽ có thể xây dựng các ứng dụng tích hợp với khả năng không giới hạn.

Chuyển mục chọn trên Menu chính sang góc phải

Theo mặc định thì thanh menu chính luôn nằm ở góc bên phải. Và VB cũng không cung cấp cho chúng ta bất cứ thuộc tính gì để đưa nó sang bên phải. Nhưng với API bạn có thể dễ dàng làm được điều này.

1. Hãy tạo một Form mới, Sau đó Click vào Tools --> Menu Editor để thêm vào chương trình 3 menu chính còn các menu con thì tùy ý.
2. Dùng đoạn Code sau :
Private Declare Function GetMenu Lib "user32" (ByVal hwnd As Long) As Long

```
Private Declare Function ModifyMenu Lib "user32" Alias "ModifyMenuA" (ByVal hMenu As Long, ByVal nPosition As Long, ByVal wFlags As Long, ByVal wIDNewItem As Long, ByVal lpString As Any) As Long
```

```
Private Declare Function DrawMenuBar Lib "user32" (ByVal hwnd As Long) As Long
```

```
Private Const MF_BYPOSITION = &H400&
```

```
Private Const MF_HELP = &H4000&
```

```
Private Sub Form_Load()
```

```
res = GetMenu(Me.hwnd) ' Lấy Handle của Menu
```

```
ModifyMenu res, 2, MF_BYPOSITION Or MF_HELP, 2, "&Help"
```

```
DrawMenuBar Me.hwnd ' Vẽ lại Menu Bar
```

```
End Sub
```

Không khó lắm nhưng kết quả thật tuyệt vời phải không

Vô hiệu Desktop

Hôm nay chúng tôi tiếp tục giới thiệu với các bạn một ví dụ về cách sử dụng hàm API trong VB. Bạn có muốn làm cho Desktop bị "tê liệt" không? Bạn hoàn toàn có thể vì Desktop cũng là một cửa sổ như bao chương trình khác, điều bạn cần biết là class name của nó "Progman".

Bạn thêm vào một Module :

```
Option Explicit
```

```
DefLng A-Z
```

```
Declare Function FindWindowEx Lib "user32" _
```

```
Alias "FindWindowExA" (ByVal hWnd As Long, _
```

```
ByVal hWndChild As Long, ByVal lpzClassName As String, _
```

```
ByVal lpzWindow As String) As Long
```

```
Declare Function ShowWindow Lib "user32" _
```

```
(ByVal hWnd As Long, ByVal nCmdShow As Long) As Long
```

```
Public Const SW_HIDE = 0
```

```
Public Const SW_SHOW = 5
```

```
Public Const g_cstrShellViewWnd As String = "Progman"
```

Sau đó thêm vào Project hai button và đoạn code sau:

```
Option Explicit
```

```
DefLng A-Z
```

```
Private Function FindShellWindow() As Long
```

```
'Hàm dùng để lấy Handle của Desktop
```

```
Dim hWnd As Long
```

```
On Error Resume Next
```

```
hWnd = FindWindowEx(0&, 0&, g_cstrShellViewWnd, vbNullString)
```

```
If hWnd <> 0 Then
```

```
FindShellWindow = hWnd
```

```
End If
```

```
End Function
```

```

Private Sub HideShowWindow(ByVal hWnd As Long, Optional ByVal Hide As Boolean =
False)
    Dim lngShowCmd As Long
    On Error Resume Next
    If Hide = True Then
        lngShowCmd = SW_HIDE
    Else
        lngShowCmd = SW_SHOW
    End If
    Call ShowWindow(hWnd, lngShowCmd)
End Sub

Private Sub command2_Click()
    Dim hWnd As Long
    On Error Resume Next
    hWnd = FindShellWindow()
    If hWnd <> 0 Then
        Call HideShowWindow(hWnd)
    End If
End Sub

Private Sub command1_Click()
    Dim hWnd As Long
    On Error Resume Next
    hWnd = FindShellWindow()
    If hWnd <> 0 Then
        Call HideShowWindow(hWnd, True)
    End If
End Sub

Private Sub Form_Load()
    Command1.Caption = "Hide Desktop"
    Command2.Caption = "Show Desktop"
End Sub

```

Làm ẩn nút Start của Window

Các sức mạnh tiềm ẩn của VB nằm ở việc thao tác với các hàm WinAPI. Bạn có thể làm bất cứ việc gì nếu hiểu và nắm được cách sử dụng các hàm API. Ví dụ bạn có thể làm cho nút Start của Windows biến mất một cách dễ dàng

```

Const SWP_HIDEWINDOW = &H80
Const SWP_SHOWWINDOW = &H40
Const GW_CHILD = 5
Const GW_HWNDNEXT = 2
Private Declare Function SetWindowPos Lib "user32" (ByVal hwnd As Long, ByVal
hwndInsertAfter As Long, ByVal x As Long, ByVal y As Long, ByVal cx As Long, ByVal cy As
Long, ByVal wFlags As Long) As Long
Private Declare Function FindWindow Lib "user32" Alias "FindWindowA" (ByVal lpClassName
As String, ByVal lpWindowName As String) As Long
Private Declare Function GetWindow Lib "user32" (ByVal hwnd As Long, ByVal wCmd As
Long) As Long
Private Declare Function GetClassName Lib "user32" Alias "GetClassNameA" (ByVal hwnd As
Long, ByVal lpClassName As String, ByVal nMaxCount As Long) As Long

```

```

Dim tWnd As Long, bWnd As Long, sSave As String * 250
Private Sub Form_Load()
    tWnd = FindWindow("Shell_traywnd", vbNullString)
    bWnd = GetWindow(tWnd, GW_CHILD)
    Do
        GetClassName bWnd, sSave, 250
        If LCase(Left$(sSave, 6)) = "button" Then Exit Do
        bWnd = GetWindow(bWnd, GW_HWNDNEXT)
    Loop
    'Ấn start button
    SetWindowPos bWnd, 0, 0, 0, 0, 0, SWP_HIDEWINDOW
End Sub
Private Sub Form_Unload(Cancel As Integer)
    'hiện lại start button
    SetWindowPos bWnd, 0, 0, 0, 0, 0, SWP_SHOWWINDOW
End Sub

```

Giới hạn của biến trong VB.NET

Trong VB.6 bạn có thể khai báo giá trị kiểu như Public, Private, Friend, và (the local) Dim. Hiện nay trong VB.NET, có một cách khai báo là có giá trị trong block (block-level scope). Có nghĩa là nếu bạn khai báo một biến trong một câu lệnh dạng block (như For Next, While, Select Case, If Then...), các biến chỉ được hiểu trong phạm vi của block đó. Bạn sẽ thấy ví dụ sau đây sẽ báo lỗi khi biên dịch

```

Private Sub DoSomething()
Dim bln As Boolean = True
    If bln Then
        Dim x As Long = 10
    End If
    x = 100 'sẽ gây ra một lỗi (Nếu Option Explicit được khai báo)
End sub

```

Bạn nên lưu ý khi lập trình VB.NET

Dấu mã nguồn ASP

Chúng tôi thường nhận được các câu hỏi "Làm thế nào tôi có thể giữ bí mật được code của tôi khi host ở các Server của ISP?" Với các component rất dễ, với các script bằng ASP không đơn giản nhưng vẫn làm được.

Bí mật ở đây là một tiện ích nhỏ có tên là Windows Script Encoder của Microsoft. Script Encoder là một tool dòng lệnh đơn giản cho phép người người thiết kế mã hoá code của họ để các Web Host và Web Client không thể thay đổi source của mình.

Ví dụ bạn có một đoạn code như sau:

```

<%@ Language="VBScript" %>
<% Option Explicit %>
<html>
<head>
<title>Ví dụ Windows Script Encoding</title>
</head>
<body>
<p>

```

Bây giờ là: <% Response.Write Time() %>

vào ngày <% Response.Write Date() %>.


```
</p>
</body>
</html>
```

Sau khi mã hoá

```
<%@ Language="VBScript.Encode" %>
<#%~^EQAAAA==~}wObWP2Xw^rmbY~GwYAAA==^#~@%>
<html>
<head>
<title>Ví dụ Windows Script Encoding Sample</title>
</head>
<body>
<p>
Bây giờ là <#%~^FwAAAA==~"+daW/+c□MrY□PPks+cb,yAcAAA==^#~@%>
vào ngày<#%~^FwAAAA==~"+daW/+c□MrY□P9ID+cb,twcAAA==^#~@%>.
</p>
</body>
</html>
```

Bạn thấy VBScript chuyển VBScript.Encode và các dòng code ASP không còn khả năng để đọc.
Cú pháp câu lệnh

SCRENC [/s] [/f] [/xl] [/l defLanguage] [/e defExtension] input file output file

/s : tự động thực hiện khõngúât hiện trên màn hình
/f : Chép chõng nếu ðã có file cùng tên tồn tại
/xl : Xác ðịnh ngôn ngữ ở @language không có ở ðầu các trang ASP
/l defLanguage: ngôn ngữ mặc ðịnh(JScrip® hoặc VBScript)
/e defExtension: kiểu tập tin xuất ra
inputfile : tên tập tin xuất cần mã mã hoá
outputfile: tên tập tin sau khi ðã mã hoá
Sau ðây là 4 loại tập tin có thể ðược mã hoá bởi Script Encoder

ASP.

HTML.

Plain text

Scriptlet.

Ví dụ

screnc test.html encode.html

Một số mọ vật của ASP.NET

Hôm nay chúng ta sẽ tìm hiểu các kỹ thuật sau: Hiển thị các hộp thoại cảnh báo trên WEB (Displaying Web Warning Messages), tạo một nút Default, các bí ẩn sau các User Control.

Hiển thị các Web Warning Message:

Ðây là các bước ðơn giản ðể tạo một cảnh báo bằng Javascript, bạn chỉ cần một một dòng lệnh và ðặt ở bất cứ ðâu trên ASP.NET

1. Ðặt các dòng script sau <body> tag:

```

2.     <script>
3.     <asp:Literal id="ItlAlert" runat="server"
4.         EnableViewState="False">
5.     </asp:Literal>
6.     </script>
7.     Kế thừa các sự kiện
8.     Protected WithEvents ItlAlert_
9.         As System.Web.UI.WebControls.Literal
10.
11. Thêm các dòng code sau Web form của bạn.
12.     Private Sub Say(ByVal Message As String)
13.         ' Format string properly
14.         Message = Message.Replace("''", "\'")
15.         Message = Message.Replace(Convert.ToChar(10), "\n")
16.         Message = Message.Replace(Convert.ToChar(13), "")
17.         ' Display as JavaScript alert
18.         ItlAlert.Text = "alert('" & Message & "')"
19.     End Sub
20. Bất cứ khi nào bạn muốn hiển thị một message, chỉ đơn giản gọi hàm say
21.     Say("Sorry, your password is invalid! " & _
22.         Microsoft.VisualBasic.vbNewLine & _
23.         "Please try again, or click the Signup button to _
24.         register now.")

```

Tạo một nút Default 'Enter'

Tưởng tượng bạn có một trang ASP.NET với một nút search. Người sử dụng đánh vào một đoạn text và nhấn Enter. Form sẽ submit và kết quả được trả về. Nói cách khác, nút search đã được set là default.

Làm cách nào? Đơn giản thêm một dòng vào sự kiện Page Load, gọi phương thức [RegisterHiddenField](#)

```
Page.RegisterHiddenField("__EVENTTARGET", "tênnút củabạn")
```

Các bí mật sau các User Control

Các User Control thường là các mảnh code dường như vô nghĩa khi bạn nhìn. Nó không quá khó, và rất hữu ích trong các website lớn và tiết kiệm công sức làm việc của bạn.

Chính xác là chúng làm gì? Nếu bạn đã từng sử dụng ASP trước đây, một user control như một include file.

Bạn có thể có một menu xanh với các icon đẹp mắt, mỗi các link đến các phần khác nhau của site. Bạn không muốn mỗi trang đều phải thiết kế menu này, bạn tạo menu như một user control và thả vào bất cứ nơi đâu bạn thấy cần thiết.

1. Trong Web application project, chọn Project, Add Web User Control từ menu. Chọn một cái tên và click Open.
2. Thiết kế user control. Và đừng quên đây không phải là một trang Web hoàn chỉnh, đó chỉ là một phần nhỏ, như một login box hoặc copyright notice. Tạo code bình thường
3. Khi kết thúc đóng và lưu lại
4. Trên trang bạn muốn thử, kéo và thả tập tin .acx từ của sổ Solution Explorer

Bạn thấy rằng có thể dễ dàng sử dụng bất cứ nơi đâu bạn muốn

Gán (binding) dữ liệu vào DropDownList (ASP.NET)

Để binding một SqlDataReader chúng ta có thể sử dụng một hàm để điền dữ liệu vào DropDownList.

Cụ thể bạn có thể tạo hàm như sau:

```
public static bool FillDropDownList(DropDownList dDI,string Select)
{
    SqlConnection m_SqlConnection = new SqlConnection(CRbvDB.RbvWebDSN);
    SqlCommand m_SqlCommand = new SqlCommand(Select,m_SqlConnection);
    try
    {
        dDI.DataValueField = "Id"; //trường này bạn có thể chỉnh sửa cho phù hợp
        dDI.DataTextField = "Name"; //trường này bạn có thể chỉnh sửa cho phù hợp
        m_SqlConnection.Open();
        SqlDataReader m_SqlDataReader = m_SqlCommand.ExecuteReader();
        dDI.DataSource = m_SqlDataReader;
        dDI.DataBind();
        m_SqlDataReader.Close(); // Close DataReader
    } // try
    catch // (Exception e) // Exception Removed
    {
        return false;
        //throw new Exception("Có lỗi khi điền DropDownLit -> " + e.ToString());
    } // catch
    finally
    {
        m_SqlCommand.Dispose();
        m_SqlConnection.Close(); // Đóng kết nối
        m_SqlConnection.Dispose();
    }
    return true;
}
```

Trong ví dụ DropDownList sẽ điền với các trường trong database.Bạn có thể truyền biến để sử dụng hàm như sau:

```
string selectCmd = "SELECT Id,LastName + ', ' + FirstName AS Name "; //(các trường dữ liệu trên đây là giá lập)
selectCmd += "FROM aRepresentative WHERE Id=SLS ORDER BY LastName ";
dDI.FillDropDownList(sslSlsId,selectCmd);
```

Tạo Control để kiểm tra Credit Card (ASP.NET + DLL)

Đầu tiên chúng ta sẽ tạo một lớp bao gồm các hàm chính của chúng ta (thuật toán để nhận diện số credit card chúng tôi đã trình bày ở phần trước)

```
1: using System;
2: using System.Web.UI;
3: using System.Web.UI.WebControls;
4:
5: namespace CustomValidators
6: {
```

```

7:  /// <summary>
8:  /// Summary description for Class1.
9:  /// </summary>
10:
11:
12: public class CreditCardValidator : BaseValidator
13: {
14:     protected override bool EvaluateIsValid()
15:     {
16:         //-- Lấy giá trị
17:         string valueToValidate = this.GetControlValidationValue(this.ControlToValidate);
18:         int indicator = 1;
19:         int firstNumToAdd = 0;
20:         int secondNumToAdd = 0;
21:         string num1;
22:
23:         string num2;
24:
25:
26:         //-- chuyển số credit sang mảng ký tự
27:         char[] ccArr = valueToValidate.ToCharArray();
28:
29:         for (int i=ccArr.Length-1;i>=0;i--)
30:         {
31:             char ccNoAdd = ccArr[i];
32:             int ccAdd = Int32.Parse(ccNoAdd.ToString());
33:             if (indicator == 1)
34:             {
35:
36:                 firstNumToAdd += ccAdd;
37:
38:                 indicator = 0;
39:             }
40:             else
41:             {
42:
43:
44:
45:                 if ((ccAdd + ccAdd) >= 10)
46:                 {
47:                     int temporary = (ccAdd + ccAdd);
48:                     num1 = temporary.ToString().Substring(0,1);
49:                     num2 = temporary.ToString().Substring(1,1);
50:                     secondNumToAdd += (Convert.ToInt32(num1) + Convert.ToInt32(num2));
51:                 }
52:                 else
53:                 {
54:
55:                     secondNumToAdd += ccAdd + ccAdd;
56:                 }
57:
58:                 indicator = 1;
59:             }
60:         }
61:     }
62:

```

```

63:
64:     bool isValid = false;
65:     if ((firstNumToAdd + secondNumToAdd) % 10 == 0)
66:     {
67:         isValid = true;
68:     }
69:     else
70:     {
71:         isValid = false;
72:     }
73:     return isValid;
74: }
75: }
76: }
77:

```

Biên dịch thành DLL:

```
csc /target:library /out:c:\inetpub\wwwroot\bin\Validator.dll *.cs /r:System.dll,System.Web.dll
```

Sử dụng DLL trong dự án của chúng ta

```

1: <%@ Register TagPrefix="custom" Namespace="CustomValidators"
Assembly="Validator" %>
2:
3: <html>
4: <body>
5: <form runat="server">
6:     <asp:TextBox ID="CCNumber" Runat="server" />
7:     <custom:CreditCardValidator
8:         ControlToValidate="CCNumber"
9:         Runat="server"
10:         ErrorMessage="Credit Card Number Invalid!" />
11:     <asp:Button Text="Validate" Runat="server" />
12: </form>
13: </body>
14: </html>

```

Thao tác các sự kiện của Web User Control

Trước khi có ASP.NET bạn sử dụng các code chung bằng tag khóa <#include>. Trong ASP.NET cho phép bạn tạo các Web user control, một khi bạn tạo được một Web user control bạn có thể kéo thả nó ở bất cứ đâu. Một câu hỏi đặt ra là khi sử dụng các Web user control, làm thế nào để nắm giữ các sự kiện phát sinh từ các control chứa trong trang ASPX .

Trong Visual Studio .NET, tạo một user control tên MyToolbar với 2 button control trên nó. Định thuộc tính button ID là btnSave và btnCancel và mỗi thuộc tính text button là Save và Cancel.

```

AutoEventWireup="false"
Codebehind="MyToolbar.ascx.vb"

```

```

    Inherits="WebApplication2.MyToolbar"
    TargetSchema=http://schemas.
    microsoft.com/intellisense/ie5%>
< asp:Button id="btnSave" Text="Save"
    runat="server">
< asp:Button id="btnCancel" Text="Save"
    runat="server">

```

Để tạo toolbar trên trang ASP bằng cách kéo control trên cửa sổ Solution Explorer vào trang. thêm label vào trang ASPX, tên label lblEvent, và tính thuộc tính Text là "". Tiếp theo bạn thêm code để xử lý sự kiện từ các control. chọn "(Overrides)" từ cửa sổ code view trong Class Name. Chọn OnBubbleEvent trong Method Name. Bạn sẽ có một sự kiện sau:

```

Protected Overrides Function _
    OnBubbleEvent(ByVal source As _
    Object, ByVal args As _
    System.EventArgs) As Boolean

```

End Function

OnBubbleEvent xử lý tất cả các sự kiện phát sinh từ control. Đối số *source* là control phát sinh sự kiện. Đối số *args* chứa bất cứ đối số chuyên vào khi sự kiện phát sinh. Khi bạn đã biết control nào phát sinh bạn có thể có những cách thích hợp để xử lý trên Web của bạn. Trong ví dụ toolbar, kiểm tra *source* là một button; sử dụng *dùng thuộc tính id* để xác định button. Thêm đoạn code vào OnBubbleEvent:

```

If source.GetType.Equals _
    (GetType(Button)) Then
    Dim oButton As Button = source
    Select Case oButton.ID
        Case "btnSave"
            lblEvent.text = "Saved record"
        Case "btnCancel"
            lblEvent.text = "Canceled changes"
        Case Else
            '.....
    End Select

```

End If

Email Attachment (ASP.NET)

Đây là một đoạn code nhỏ giúp bạn gửi mail có kèm (attach) tập tin. Hy vọng đoạn code giúp ích cho các bạn trong công việc.

```

<%@ Page Language="VB" ClientTarget="downlevel" %>
<%@ Import Namespace="System.Web.Mail" %>
<script language="VB" runat="server">

```

```

Sub btnSendMessage_OnClick(Source As Object, E As EventArgs)
    Dim myMessage As New MailMessage
    Dim myAttachment As MailAttachment
    Dim myMail As SmtMail

```

```
Dim strEmail As String
```

```
If Page.IsValid() Then  
strEmail = txtEmail.Text
```

```
myMessage.From = "webmaster@" & Request.ServerVariables("SERVER_NAME")  
myMessage.To = strEmail  
myMessage.Subject = "E-mail Sample from VASC!"  
myMessage.Body = "This message was sent from sample code "  
myAttachment = New MailAttachment(Server.MapPath("attachment.txt")) ' tập tin bạn chọn  
myMessage.Attachments.Add(myAttachment)  
myMail.SmtpServer = "localhost" ' hoặc SMTP Server nào bạn định  
myMail.Send(myMessage)  
frmEmail.Visible = False  
lblUserMessage.Text = "Your message (with attachment) has been sent to " & strEmail & "."  
End If  
End Sub  
</script>
```

```
<html>  
<head>  
<title>ASP.NET Email (with Attachment) Sample</title>  
</head>  
<body>
```

```
<asp:Label id="lblUserMessage" text="Enter your e-mail address:" runat="server" />  
<form method="post" id="frmEmail" runat="server">  
<asp:TextBox id="txtEmail" size="30" runat="server" />  
<asp:RequiredFieldValidator runat="server"  
id="validEmailRequired" ControlToValidate="txtEmail"  
errorMessage="Please enter an email address."  
display="Dynamic" />  
<asp:RegularExpressionValidator runat="server"  
id="validEmailRegExp" ControlToValidate="txtEmail"  
ValidationExpression="^[w-]+@[w-]+\.(com|net|org|edu|mil)$"  
errorMessage="Please enter a valid email address."  
Display="Dynamic" />  
  
<asp:Button id="btnSendMail" text="Send Mail!" OnClick="btnSendMail_OnClick" runat="server"  
</form>
```

```
</body>  
</html>
```

Lấy kích thước màn hình của client trong ASP.NET

Bạn muốn xác định kích thước màn hình của client để dàn trang web tự động theo cấu hình. Các dòng code sau sẽ giúp bạn lấy độ rộng và cao của client và gửi trở lại về Server
Chúng tôi sử dụng 2 tập tin

start.aspx

```
<html>  
<body onload="storeWidthAndHeightToTheServer()">  
<form id="resolution" method="post" action="default.aspx">
```

```

</form>
<script language="javascript">
function storeWidthAndHeightToTheServer()
  with (document.getElementById('resolution'))
  {
    var InitStr;
    InitStr = "default.aspx?ScreenWidth=";
    InitStr += screen.width;
    InitStr += "&ScreenHeight=";
    InitStr += screen.height;
    action = InitStr;
    submit();
  }
}
</script>
</body>
</html>

```

default.aspx

```

<html>
<head>
<script language="C#" runat="server">
void Page_Load(object s, EventArgs e)
{
  string width = HttpContext.Current.Request.QueryString["ScreenWidth"];
  string height = HttpContext.Current.Request.QueryString["ScreenHeight"];
  Session["width"] = width;
  Session["height"] = height;
  Label1.Text = "Width= " + width + " Height= " + height;
}
</script>
</head>
<body>
<form runat="server" ID="Form1">
<asp:Table Width="100%" Runat="server" ID="Table1">
<asp:TableRow>
<asp:TableCell HorizontalAlign="Center">
<asp:Label ID="Label1" Runat="server"></asp:Label>
</asp:TableCell>
</asp:TableRow>
</asp:Table>
</form>
</body>
</html>

```

Bây giờ bạn có thể có các giá trị và lưu vào session và sử dụng các giá trị ở bất cứ đâu trong web application

Áp các kiểu cho các ASP.NET Web Control

Có nhiều cách khác nhau để sử dụng các kiểu cho các Web control. Một trong các cách thường dùng nhất là sử dụng các style sheet. Bạn có thể áp dụng các của của Web control một cách tự

động và .NET Framework class library có cung cấp các lớp để thực hiện việc này.

The System.Web.UI.WebControls.Style Class

Lớp Style được định nghĩa trong System.Web.UI.WebControls namespace để thể hiện kiểu của một Web server control. Lớp này cung cấp các thuộc tính có thể được sử dụng để áp dụng cho một hoặc nhiều Web control. Sử dụng các thuộc tính trên bạn có thể đặt màu background, foreground, độ rộng border và kiểu và kích thước của Web server controlcontrols to provide a common appearance. Using these properties, you can set the . Table 1 describes the Style class properties.

Các thuộc tính của lớp Style

| | |
|-------------|--|
| BackColor | Lấy và định màu background của Web server control |
| BorderColor | Lấy và định màu border của Web server control |
| BorderStyle | Lấy và định kiểu border của the Web server control |
| BorderWidth | Lấy và định độ rộng border của Web server control |
| CssClass | Lấy và định render CSS class của Web server control trên máy client. |
| Font | Lấy và định các thuộc tính liên quan font của the Web server control |
| ForeColor | Lấy và định màu foreground của Web server control. |
| Height | Lấy và định chiều cao của Web server control. |
| Width | Lấy và định chiều rộng của Web server control. |

The System.Web.UI.WebControls.WebControl.ApplyStyle Method

Phương thức ApplyStyle của lớp WebControl dùng để áp kiểu một đối tượng Style cho một Web control. Phương thức này sử dụng đối số là một Style object. Ví dụ:

```
WebControl ctrl;  
Style s;  
ctrl.ApplyStyle(s);
```

Ví dụ cụ thể:

Bạn tạo một Web application sử dụng Visual Studio .NET và thêm 3 control vào Form - một Button, một TextBox, và một ListBox.

Bây giờ bạn tạo 2 phương thức - CreateStyle và SetControlStyle. Phương thức CreateStyle lấy các đối số của như màu background, màu foreground, độ rộng border, và các kiểu font.

```
private Style CreateStyle(Color backClr, Color foreClr, int borderWidth, string  
fntName, int fntSize, bool fntBold, bool fntItalic  
/* Bạn có thể thêm nhiều đối số */  
{  
    Style s = new Style();  
    s.BackColor = backClr;  
    s.ForeColor = foreClr;  
    s.BorderWidth = borderWidth;  
    s.Font.Name = fntName;  
    s.Font.Size = fntSize;  
    s.Font.Bold = fntBold;  
    s.Font.Italic = fntItalic;  
    return s;  
}
```

```

}

// Phương thức áp kiểu đến một Web control

private void SetControlStyle(System.Web.UI.WebControls.WebControl ctrl, Style s)
{
    ctrl.ApplyStyle(s);
}

private void Button1_Click(object sender, System.EventArgs e)
{
    Style st = CreateStyle(Color.Green, Color.Yellow, 3, "Verdana", 10, true, true);
    SetControlStyle(TextBox1, st);
    st = CreateStyle(Color.Red, Color.Black, 2, "Verdana", 12, true, true);
    SetControlStyle(SetStyleBtn, st);
    st = CreateStyle(Color.Blue, Color.Yellow, 2, "Verdana", 12, true, true);
    SetControlStyle(ListBox1, st);
}

```

Truy cập các giá trị của Server từ trong Web Service

Một trong các câu hỏi thường gặp trong các newsgroup là " Làm thế nào tôi lấy được địa chỉ IP của client browser trong một web service ?". Câu trả lời rất đơn giản. Sử dụng lớp Context trong System.Web.Services. Bạn có thể hiểu được các làm thông qua các ví dụ sau của chúng tôi.

Chúng ta sẽ xem hai ví dụ khá đơn giản

1. Nhận IP address của client browser
2. Nhận tất cả các giá trị của web server

```

<%@ WebService Language="C#" class="httpvars" %>
using System;
using System.Collections;
using System.Web.Services;

public class httpvars : WebService
{
    // Phương thức này trả về địa chỉ IP của client
    [WebMethod]
    public String ipAddress ()
    {
        return Context.Request.ServerVariables["REMOTE_ADDR"];
    }

    // Phương thức trả về tất cả các giá trị của Server
    [WebMethod]
    public String allHttpVars ()
    {
        NameValueCollection serverVars;
        String returnValue = "";
        serverVars = Context.Request.ServerVariables;
    }
}

```

```

String[] arVars = serverVars.AllKeys;

for (int x = 0; x < arVars.Length; x++)
{
    returnValue += "<b>" + arVars[x] + "</b>: ";
    returnValue += serverVars[arVars[x]] + "<br>";
}
return returnValue;
}
}
}

```

Nhiều runat=server forms trên cùng một trang (ASP.NET)

ASP.NET không hỗ trợ nhiều **runat=server forms trên cùng một trang**. Để khắc phục việc này, bạn có thể đặt mỗi form trong mỗi Panel control riêng biệt, và cho phép người dùng dễ dàng chuyển giữa các panel bằng click một radio button.

Bạn có thể tham khảo ví dụ sau:
2FormExample.aspx

```

<%@ Page language="c#" Codebehind="2FormExample.cs" AutoEventWireup="false"
Inherits="_3leaf_app.C2FormExample" %>
<html><head>
<meta name=vs_targetSchema content="HTML 4.0">
<meta name="GENERATOR" Content="Microsoft Visual Studio 7.0">
<meta name="CODE_LANGUAGE" Content="C#"></head>
<body>

<form method="post" runat="server" ID=Form1>
    <p>Lookup by
    <asp:RadioButton id=RadioButton1 runat="server" Text="First Name"
AutoPostBack="True" groupname=g1 checked=True></asp:RadioButton>
    <asp:RadioButton id=RadioButton2 runat="server" Text="Last Name" AutoPostBack="True"
groupname=g1></asp:RadioButton></p>
    <p></p>
    <p>
    <asp:Panel id=Panel1 runat="server" visible=True>
        First Name :
        <asp:TextBox id=TextBox1 runat="server"></asp:TextBox>
        <asp:RequiredFieldValidator id=RequiredFieldValidator1 runat="server" ErrorMessage=""
ControlToValidate="TextBox1"></asp:RequiredFieldValidator>
        <asp:Button id=Button1 runat="server" Text="Submit"></asp:Button>
    </asp:Panel>
    <asp:Panel id=Panel2 runat="server" visible=False>
        Last Name :
        <asp:TextBox id=TextBox2 runat="server"></asp:TextBox>
        <asp:RequiredFieldValidator id=RequiredFieldValidator2 runat="server" ErrorMessage=""
ControlToValidate="TextBox2"></asp:RequiredFieldValidator>
        <asp:Button id=Button2 runat="server" Text="Submit"></asp:Button>
    </asp:Panel>
    <p></p>
    <p>
        <asp:label id=Label1 runat="server"></asp:label>
    </p>
</form>

```

```
</body></html>
```

2FormExample.cs

```
namespace _3leaf_app
{
    using System;
    using System.Collections;
    using System.ComponentModel;
    using System.Data;
    using System.Drawing;
    using System.Web;
    using System.Web.SessionState;
    using System.Web.UI;
    using System.Web.UI.WebControls;
    using System.Web.UI.HtmlControls;

    /// <summary>
    /// Summary Description for C2FormExample.
    /// </summary>
    public class C2FormExample : System.Web.UI.Page
    {
        protected System.Web.UI.WebControls.Button Button2;
        protected System.Web.UI.WebControls.RequiredFieldValidator RequiredFieldValidator2;
        protected System.Web.UI.WebControls.TextBox TextBox2;
        protected System.Web.UI.WebControls.Button Button1;
        protected System.Web.UI.WebControls.RequiredFieldValidator RequiredFieldValidator1;
        protected System.Web.UI.WebControls.TextBox TextBox1;
        protected System.Web.UI.WebControls.Label Label1;
        protected System.Web.UI.WebControls.Panel Panel2;
        protected System.Web.UI.WebControls.Panel Panel1;
        protected System.Web.UI.WebControls.RadioButton RadioButton2;
        protected System.Web.UI.WebControls.RadioButton RadioButton1;

        public C2FormExample()
        {
            Page.Init += new System.EventHandler(Page_Init);
        }

        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
            }
        }

        protected void Page_Init(object sender, EventArgs e)
        {
            //
            // CODEGEN: This call is required by the ASP+ Windows Form Designer.
            //
            InitializeComponent();
        }
    }
}
```

```

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
        RadioButton1.CheckedChanged += new System.EventHandler
        (this.RadioButton1_CheckedChanged);
        Button1.Click += new System.EventHandler (this.Button1_Click);
        RadioButton2.CheckedChanged += new System.EventHandler
        (this.RadioButton2_CheckedChanged);
        Button2.Click += new System.EventHandler (this.Button2_Click);
        this.Load += new System.EventHandler (this.Page_Load);
    }

    public void Button2_Click (object sender, System.EventArgs e)
    {
        Label1.Text = "You want to search on last name";
    }

    public void Button1_Click (object sender, System.EventArgs e)
    {
        Label1.Text = "You want to search on first name";
    }

    public void RadioButton2_CheckedChanged (object sender, System.EventArgs e)
    {
        Panel1.Visible = false;
        Panel2.Visible = true;
    }

    public void RadioButton1_CheckedChanged (object sender, System.EventArgs e)
    {
        Panel1.Visible = true;
        Panel2.Visible = false;
    }
}
}
}

```

Chuyển đổi giá trị từ số sang chữ (.NET)

Đây là một ví dụ đơn giản dùng để chuyển đổi số sang chữ tương ứng (tiếng Anh). Rất hữu dụng trong kế toán và các hoá đơn. Ví dụ bao gồm cả trang ASPX sử dụng service này

Numerals.asmx

```
<%@ WebService Language="VB" Class="NumberToWord" %>
```

```
Imports System
Imports System.Web.Services
```

```
Public Class NumberToWord : Inherits WebService
```

```
Public Function <WebMethod(> Int(num As double) As double
return(num-(num mod 1))
end function
```

```
Public Function <WebMethod(> BritishNumerals(numstr As double) As String
Dim tempstr as string
Dim newstr as string
numstr = Cdbl(numstr)
If numstr > 10 ^ 24 Then
return "Too big"
Exit Function
End If
```

```
If numstr >= 10 ^ 7 Then
newstr = BritishNumerals(Int(numstr / (10^7)))
numstr = ((numstr / 10 ^ 7) - Int(numstr / 10 ^ 7)) * 10 ^ 7
If numstr = 0 Then
tempstr = tempstr & newstr & "Crore "
Else
tempstr = tempstr & newstr & "Crore, "
End If
End If
```

```
If numstr >= 10 ^ 5 Then
newstr = BritishNumerals(Int(numstr / 10 ^ 5))
numstr = ((numstr / 10 ^ 5) - Int(numstr / 10 ^ 5)) * 10 ^ 5
If numstr = 0 Then
tempstr = tempstr & newstr & "Lakh "
Else
tempstr = tempstr & newstr & "Lakh, "
End If
End If
```

```
If numstr >= 10 ^ 3 Then
newstr = BritishNumerals(Int(numstr / 10 ^ 3))
numstr = ((numstr / 10 ^ 3) - Int(numstr / 10 ^ 3)) * 10 ^ 3
If numstr = 0 Then
tempstr = tempstr & newstr & "Thousand "
Else
tempstr = tempstr & newstr & "Thousand, "
End If
End If
```

```
If numstr >= 10 ^ 2 Then
newstr = BritishNumerals(Int(numstr / 10 ^ 2))
numstr = ((numstr / 10 ^ 2) - Int(numstr / 10 ^ 2)) * 10 ^ 2
If numstr = 0 Then
tempstr = tempstr & newstr & "Hundred "
Else
tempstr = tempstr & newstr & "Hundred And "
End If
End If
```

```
If numstr >= 20 Then
Select Case Int(numstr / 10)
Case 2
```

```
tempstr = tempstr & "Twenty "  
Case 3  
tempstr = tempstr & "Thirty "  
Case 4  
tempstr = tempstr & "Forty "  
Case 5  
tempstr = tempstr & "Fifty "  
Case 6  
tempstr = tempstr & "Sixty "  
Case 7  
tempstr = tempstr & "Seventy "  
Case 8  
tempstr = tempstr & "Eighty "  
Case 9  
tempstr = tempstr & "Ninety "  
End Select  
numstr = ((numstr / 10) - Int(numstr / 10)) * 10  
End If
```

```
numstr=Int(numstr+0.5)
```

```
If numstr > 0 Then  
Select Case NUMSTR MOD 100  
Case 1  
tempstr = tempstr & "One "  
Case 2  
tempstr = tempstr & "Two "  
Case 3  
tempstr = tempstr & "Three "  
Case 4  
tempstr = tempstr & "Four "  
Case 5  
tempstr = tempstr & "Five "  
Case 6  
tempstr = tempstr & "Six "  
Case 7  
tempstr = tempstr & "Seven "  
Case 8  
tempstr = tempstr & "Eight "  
Case 9  
tempstr = tempstr & "Nine "  
Case 10  
tempstr = tempstr & "Ten "  
Case 11  
tempstr = tempstr & "Eleven "  
Case 12  
tempstr = tempstr & "Twelve "  
Case 13  
tempstr = tempstr & "Thirteen "  
Case 14  
tempstr = tempstr & "Fourteen "  
Case 15  
tempstr = tempstr & "Fifteen "  
Case 16  
tempstr = tempstr & "Sixteen "
```

```

Case 17
tempstr = tempstr & "Seventeen "
Case 18
tempstr = tempstr & "Eighteen "
Case 19
tempstr = tempstr & "Nineteen "
End Select
numstr = ((numstr / 10) - Int(numstr / 10)) * 10
End If
return tempstr

```

```

End Function

```

```

Public Function <WebMethod(> AmericanNumerals(numstr As double) As String
Dim tempstr as string
Dim newstr as string
numstr = Cdbl(numstr)
If numstr > 10 ^ 24 Then
return "Too big"
Exit Function
End If

```

```

If numstr >= 10 ^ 9 Then
newstr = AmericanNumerals(Int(numstr / (10^9)))
numstr = ((numstr / 10 ^ 9) - Int(numstr / 10 ^ 9)) * 10 ^ 9
If numstr = 0 Then
tempstr = tempstr & newstr & "Billion "
Else
tempstr = tempstr & newstr & "Bullion, "
End If
End If

```

```

If numstr >= 10 ^ 6 Then
newstr = AmericanNumerals(Int(numstr / 10 ^ 6))
numstr = ((numstr / 10 ^ 6) - Int(numstr / 10 ^ 6)) * 10 ^ 6
If numstr = 0 Then
tempstr = tempstr & newstr & "Million "
Else
tempstr = tempstr & newstr & "Million, "
End If
End If

```

```

If numstr >= 10 ^ 3 Then
newstr = AmericanNumerals(Int(numstr / 10 ^ 3))
numstr = ((numstr / 10 ^ 3) - Int(numstr / 10 ^ 3)) * 10 ^ 3
If numstr = 0 Then
tempstr = tempstr & newstr & "Thousand "
Else
tempstr = tempstr & newstr & "Thousand, "
End If
End If

```

```

If numstr >= 10 ^ 2 Then
newstr = AmericanNumerals(Int(numstr / 10 ^ 2))
numstr = ((numstr / 10 ^ 2) - Int(numstr / 10 ^ 2)) * 10 ^ 2

```



```
If numstr = 0 Then
tempstr = tempstr & newstr & "Hundred "
Else
tempstr = tempstr & newstr & "Hundred And "
End If
End If
```

```
If numstr >= 20 Then
Select Case Int(numstr / 10)
Case 2
tempstr = tempstr & "Twenty "
Case 3
tempstr = tempstr & "Thirty "
Case 4
tempstr = tempstr & "Forty "
Case 5
tempstr = tempstr & "Fifty "
Case 6
tempstr = tempstr & "Sixty "
Case 7
tempstr = tempstr & "Seventy "
Case 8
tempstr = tempstr & "Eighty "
Case 9
tempstr = tempstr & "Ninety "
End Select
numstr = ((numstr / 10) - Int(numstr / 10)) * 10
End If
```

```
numstr=Int(numstr+0.5)
```

```
If numstr > 0 Then
Select Case NUMSTR MOD 100
Case 1
tempstr = tempstr & "One "
Case 2
tempstr = tempstr & "Two "
Case 3
tempstr = tempstr & "Three "
Case 4
tempstr = tempstr & "Four "
Case 5
tempstr = tempstr & "Five "
Case 6
tempstr = tempstr & "Six "
Case 7
tempstr = tempstr & "Seven "
Case 8
tempstr = tempstr & "Eight "
Case 9
tempstr = tempstr & "Nine "
Case 10
tempstr = tempstr & "Ten "
Case 11
tempstr = tempstr & "Eleven "
Case 12
```

```

tempstr = tempstr & "Twelve "
Case 13
tempstr = tempstr & "Thirteen "
Case 14
tempstr = tempstr & "Fourteen "
Case 15
tempstr = tempstr & "Fifteen "
Case 16
tempstr = tempstr & "Sixteen "
Case 17
tempstr = tempstr & "Seventeen "
Case 18
tempstr = tempstr & "Eighteen "
Case 19
tempstr = tempstr & "Nineteen "
End Select
numstr = ((numstr / 10) - Int(numstr / 10)) * 10
End If
return tempstr
End Function
End Class

```

NumberToWord.vb

```

' <autogenerated>
' This class was generated by a tool.
' Changes to this file may cause incorrect behavior and will be lost if
' the code is regenerated.
' </autogenerated>
'-----

```

```

Imports System.Xml.Serialization
Imports System.Web.Services.Protocols
Imports System.Web.Services

```

```

Public Class NumberToWord
Inherits System.Web.Services.Protocols.SoapClientProtocol

```

```

Public Sub New()
MyBase.New
Me.Url = "numerals.asmx"
End Sub

```

```

Public Function <System.Web.Services.Protocols.SoapMethodAttribute("http://tempuri.org/Int")>
Int(ByVal num As Double) As Double
Dim results() As Object = Me.Invoke("Int", New Object() {num})
Return CType(results(0),Double)
End Function
Public Function BeginInt(ByVal num As Double, ByVal callback As System.AsyncCallback, ByVal
asyncState As Object) As System.IAsyncResult
Return Me.BeginInvoke("Int", New Object() {num}, callback, asyncState)
End Function
Public Function EndInt(ByVal asyncResult As System.IAsyncResult) As Double
Dim results() As Object = Me.EndInvoke(asyncResult)
Return CType(results(0),Double)

```

```

End Function
Public Function
<System.Web.Services.Protocols.SoapMethodAttribute("http://tempuri.org/BritishNumerals")>
BritishNumerals(ByVal numstr As Double) As String
Dim results() As Object = Me.Invoke("BritishNumerals", New Object() {numstr})
Return CType(results(0),String)
End Function
Public Function BeginBritishNumerals(ByVal numstr As Double, ByVal callback As
System.AsyncCallback, ByVal asyncState As Object) As System.IAsyncResult
Return Me.BeginInvoke("BritishNumerals", New Object() {numstr}, callback, asyncState)
End Function
Public Function EndBritishNumerals(ByVal asyncResult As System.IAsyncResult) As String
Dim results() As Object = Me.EndInvoke(asyncResult)
Return CType(results(0),String)
End Function
Public Function
<System.Web.Services.Protocols.SoapMethodAttribute("http://tempuri.org/AmericanNumerals")>
AmericanNumerals(ByVal numstr As Double) As String
Dim results() As Object = Me.Invoke("AmericanNumerals", New Object() {numstr})
Return CType(results(0),String)
End Function
Public Function BeginAmericanNumerals(ByVal numstr As Double, ByVal callback As
System.AsyncCallback, ByVal asyncState As Object) As System.IAsyncResult
Return Me.BeginInvoke("AmericanNumerals", New Object() {numstr}, callback, asyncState)
End Function
Public Function EndAmericanNumerals(ByVal asyncResult As System.IAsyncResult) As String
Dim results() As Object = Me.EndInvoke(asyncResult)
Return CType(results(0),String)
End Function

End Class

```

NumberToWord.aspx

```

<html>
<script language="VB" runat="server">
Dim Op1 As Double = 0
Public Sub Submit_Click(Sender As Object, E As EventArgs)
Try
Op1 = Double.Parse(Operand1.Text)
Catch Exp As Exception
End Try
Dim Service As NumberToWord = New NumberToWord()
Select (CType(sender,Control).ID)
Case "English" :
Result.Text = "<b>Result</b> = " & Service.BritishNumerals(Op1).ToString()
Case "US" :
Result.Text = "<b>Result</b> = " & Service.AmericanNumerals(Op1).ToString()
End Select
End Sub
</script>
<body style="font: 10pt verdana">
<h4>Using a Simple Number Service </h4>
<form runat="server">

```

```

<div style="padding:15,15,15,15;background-color:beige;width:300;border-color:black;border-
width:1;border-style:solid">
Enter any number: <br><asp:TextBox id="Operand1" Text="15" runat="server"/><br>
<input type="submit" id="English" value="Show English Numeral Text"
OnServerClick="Submit_Click" runat="server">
<input type="submit" id="US" value="Show Americal Numeral Text"
OnServerClick="Submit_Click" runat="server">
<p>
<asp:Label id="Result" runat="server"/>
</div>
</form>
</body>
</h>

```

Nhận thông tin của tập tin từ ASP.NET

Trong ASP Chuẩn chúng ta vẫn có thể thực hiện việc này thông qua đối tượng Scripting.FileSystemObjec. Trong .NET chúng ta thay thế bằng System.IO namespace.

fileinfo.aspx (C#)

```

<%@ Page Language="c#" %>
<script runat="server">

```

```

protected System.IO.FileInfo objFI;
protected String filename;

```

```

protected void Page_Load(){
    if(!IsPostBack){
        // Lấy tên đường dẫn của tập tin
        filename = Request.ServerVariables["PATH_TRANSLATED"];

        // Tạo FileInfo object mới cho tập tin này
        objFI = new System.IO.FileInfo(filename);

        // Các trường thông tin cần thiết
        fullname.Text = objFI.FullName;
        name.Text = objFI.Name;
        size.Text = objFI.Length.ToString();
        created.Text = objFI.CreationTime.ToString();
        accessed.Text = objFI.LastAccessTime.ToString();
        modified.Text = objFI.LastWriteTime.ToString();
        directory.Text = objFI.DirectoryName;
        extension.Text = objFI.Extension;
    }
}
</script>
<HTML>
<HEAD>
</head>
</HEAD>
<BODY>
<form runat="server">
<b>File Information</b>
<table class="articlelist">
<tr>
    <td class="header">Full Name</td>
    <td>
        <asp:Label id="fullname" runat="server"/>
    </td>
</tr>

```

```
</td>
</tr>
<tr>
  <td class="header">Name</td>
  <td>
    <asp:Label id="name" runat="server"/>
  </td>
</tr>
<tr>
  <td class="header">Extension</td>
  <td>
    <asp:Label id="extension" runat="server"/>
  </td>
</tr>
<tr>
  <td class="header">Size</td>
  <td>
    <asp:Label id="size" runat="server"/>
  </td>
</tr>
<tr>
  <td class="header">Created</td>
  <td>
    <asp:Label id="created" runat="server"/>
  </td>
</tr>
<tr>
  <td class="header">Modified</td>
  <td>
    <asp:Label id="modified" runat="server"/>
  </td>
</tr>
<tr>
  <td class="header">Accessed</td>
  <td>
    <asp:Label id="accessed" runat="server"/>
  </td>
</tr>
<tr>
  <td class="header">Parent Folder</td>
  <td>
    <asp:Label id="directory" runat="server"/>
  </td>
</tr>
</table>
</form>
</BODY>
</HTML>
```

tml>

Đọc và ghi vào Cookies trong ASP.Net

Cookie là một khái niệm khá quen thuộc trong lập trình Web, tuy nhiên khi .NET ra đời rất nhiều người không hiểu cách sử dụng chúng có gì thay đổi và các thức làm việc có còn như xưa? Có rất nhiều cách định danh trong ASP.NET như web.config và global.asax, nhưng còn về cookie ?

Đây là một cookie với một giá trị:

```
Response.Cookies("MakCookie").Value = "Welcome to VASC"
```

Đây là một cookie với một cặp tên:

```
Response.Cookies("Makovision")("Username") = "VASC"  
Response.Cookies("Makovision")("Option") = "Restricted"
```

Thật đơn giản chúng ta đã viết cookie thành công. Bây giờ chúng ta sẽ đọc giá trị trong ASP.NET

```
Response.Cookies("Makovision")("Username") = "VASC"  
Response.Cookies("Makovision")("Option") = "Restricted"  
Dim strValue as String  
for each strValue in Response.Cookies("Makovision").values  
MakCookie.Text += "<b>" & strValue & "</b>: " &  
Request.Cookies("Makovision")(strValue) & "<BR>"  
next
```

trong Body

```
<form runat="server">  
  <asp:label id="MakCookie" runat="server" />  
</form>
```

Đọc Pop3 E-mail bằng ASP.NET

Với sức mạnh của .NET bạn có thể làm được nhiều việc trước kia với ASP chuẩn bạn rất khó thực hiện và đôi lúc không thể thực hiện được. i-Today hôm nay sẽ hướng dẫn các bạn cách đọc Pop3 Email bằng ASP.NET

pop3.aspx

```
<%@page%>
<HTML><HEAD><title>Pop3 mail check</title></HEAD>
<body bgcolor=white>

<%

if isPostBack then
readMail(host.text,user.text,pass.text)
else
%>
<form id=calc method=post runat="server">
<P>
Host <asp:TextBox id=host runat="server"></asp:TextBox>
<P>
User <asp:TextBox id=user runat="server"></asp:TextBox>
<P>
Pass <asp:TextBox TextMode=Password id=pass runat="server"></asp:TextBox>
<P>
<asp:Button id=Button1 runat="server" Text="Login"></asp:Button>
</FORM>

<%

end if

%>

</body></HTML>
```

```

<script language="vb" runat="server">
dim tcpC as New system.net.sockets.TcpClient()
Function SendCommand(byRef netstream as System.Net.Sockets.NetworkStream,byVal
sToSend as String)

dim bData() as Byte = Encoding.ASCII.GetBytes(sToSend.ToCharArray)
netstream.Write(bData,0,bData.Length())
Return GetResponse(netstream)
End Function
Function GetResponse(byRef netstream as System.Net.Sockets.NetworkStream)

dim bytes(tcpC.ReceiveBufferSize) As Byte
dim ret as integer = netStream.Read(bytes, 0, bytes.length)
dim returndata As String = Encoding.ASCII.GetString(bytes)
return returndata
End Function

Function ReadMail(host as string, user as string, pass as string)
dim netstream as System.Net.Sockets.NetworkStream
dim thisResponse as string
try
tcpC.Connect(host,110)
catch ex as exception
response.write("Error connecting to host: " & ex.message & " - Please check your details and try
again")
response.end
end try
netstream = tcpC.GetStream()
thisResponse=GetResponse(netstream)
thisResponse=SendCommand(netstream,"user " & user & vbCrLf)
thisResponse=SendCommand(netstream,"pass " & pass & vbCrLf)
if not left(thisResponse,4)="-ERR" then
response.write("<font face=courier>Logged in OK <BR>")
else

```



```

response.write("Error logging in, check your user details and try again<BR>")
response.write("<P>" & thisresponse)
response.end
end if
thisResponse=SendCommand(netstream,"stat" & vbCrLf)

dim tmpArray() as string
tmpArray = split(thisResponse," ")

dim thisMess as integer
dim numMess as string = tmpArray(1)
response.write("<p><hr>")
thisResponse = ""
if cint(numMess) > 0 then
response.write("Messages: " & numMess & "<br>")
for thisMess = 1 to cint(numMess)
thisResponse += replace(SendCommand(netstream,"top " & thisMess & " 10" &
vbCrLf),vbCrLf,"<br>")
next
else
response.write("Messages: None" & "<br>")
end if
thisResponse += replace(SendCommand(netstream,"stat" & vbCrLf),vbCrLf,"<br>")

tmpArray = split(thisResponse,"+OK")
response.write(thisresponse)

dim msg as integer
for msg = 1 to tmpArray.length-1

response.write("<h3>#" & msg & "</h1>" & tmpArray(msg) & "<p>")

next
thisResponse=SendCommand(netstream,"QUIT" & vbCrLf)

```

```
tcpC.close
```

```
End Function
```

```
</script>
```

Kỹ thuật làm mờ hình (blur) với GDI + System.Drawing (.NET)

Đây là một kỹ thuật được thực hiện hết sức dễ dàng trên Web, bằng sức mạnh của .NET. Chúng ta sử dụng một thuật toán hết sức đơn giản, giá trị của mỗi pixel bằng giá trị trung bình của các pixel ở trên, trái, phải của nó.

blur.aspx

```
<%@ Page Language="vb" %>
<%@ import namespace="system.drawing" %>
<%@ import namespace="system.drawing.imaging" %>
<%@ import namespace="system.drawing.drawing2d" %>
<%
dim b as New system.drawing.bitmap(server.mappath("example1.jpg"))

dim x,y,cnt
dim addR, addB, addG as integer
dim incAmount = Request.QueryString("increase")

addR = 0
addG = 0
addB = 0

for y = 3 to b.height-3
for x = 3 to b.width-3
addR = b.GetPixel(x,y-1).r
addR += b.GetPixel(x,y+1).r
addR += b.GetPixel(x,y).r
addR += b.GetPixel(x+1,y).r
addR += b.GetPixel(x-1,y).r
addR = addR / 5
addG = b.GetPixel(x,y-1).g
addG += b.GetPixel(x,y+1).g
```

```

addG += b.GetPixel(x,y).g
addG += b.GetPixel(x+1,y).g
addG += b.GetPixel(x-1,y).g
addG = addG / 5
addB = b.GetPixel(x,y-1).b
addB += b.GetPixel(x,y+1).b
addB += b.GetPixel(x,y).b
addB += b.GetPixel(x+1,y).b
addB += b.GetPixel(x-1,y).b
addB = addB / 5
b.SetPixel(x,y,color.fromARGB(addR,addG,addB))
addR = 0
addG = 0
addB = 0
next
next
response.contentType="image/jpeg"
b.save(response.outputstream, imageformat.jpeg)
b.dispose()
%>

```

Mẹo liên quan đến việc chuyển dữ liệu quan hệ sang XML (.NET)

The DataSet Class

Với sự giới thiệu của ADO.NET và lớp DataSet, là công cụ tuyệt vời để xem dữ liệu cả ở dạng quan hệ lẫn XML. Có nhiều phương thức trong lớp DataSet như GetXml() hoặc WriteXml(). Mặc dù GetXml() chỉ đơn giản trả lại một chuỗi thể hiện cho dữ liệu XML, bạn có thể sử dụng WriteXml() để ghi XML đến các đối tượng khác nhau:

- Streams
- TextWriters
- XmlWriters
- Files

Nếu dữ liệu quan hệ cần được chuyển về cấu trúc DOM, trong thực tế việc chuyển dữ liệu quan hệ về cấu trúc DOM đơn giản như truyền DataSet trong XmlDataDocument's constructor:

```
XmlDataDocument doc = new XmlDataDocument(myDataSet);
```

SQL Server 2000

SQL Server 2000 cung cấp nhiều kỹ thuật để chuyển dữ liệu quan hệ sang XML. Khi bạn sử dụng chung với .NET platform, bạn có thể sử dụng nhiều tính năng mạnh của cả hai. Đầu tiên bạn có thể kết nối đến dữ liệu của SQL 2000 thông qua HTTP hơn ADO.NET. Thông qua sử dụng URL, bạn có thể nạp dữ liệu trực tiếp vào XmlDocument hoặc XmlTextReader. Việc này hoàn toàn có khả năng bằng các hàm đặc biệt được tích hợp vào SQL Server 2000. Ví dụ, câu SQL này sẽ chuyển dữ liệu sang XML một cách tự động

```
SELECT * FROM Customers FOR XML AUTO
```

Nếu bạn cần xác định thành phần của XML một cách riêng biệt, thêm khoá ELEMENTS:

```
SELECT * FROM Customers FOR XML AUTO, ELEMENTS
```

Nếu bạn có nhu cầu sử dụng ADO.NET Connection Class để kết nối đến CSDL như thông qua HTTP, bạn có thể sử dụng phương thức ExecuteXmlReader() của lớp Command. Việc này cho phép dữ liệu XML được nạp trực tiếp vào lớp XmlTextReader.

SQLXML Managed Classes

Các lớp XML mới của .NET dùng cho SQL Server 2000 có thể sử dụng khi bạn cài đặt SQL Server 2000 Web Release. Web Release bao gồm Microsoft.Data.SqlXml namespace (bao gồm các lớp SqlXmlCommand, SqlXmlParameter, và SqlXmlAdapter) cho phép bạn sử dụng để kết nối đến CSDL và xem XML được xuất. Ví dụ:

```
SqlXmlCommand cmd = new SqlXmlCommand(connString);  
cmd.CommandText = "Customer";  
cmd.CommandType = SqlXmlCommandType.XPath;  
cmd.RootTag = "Customers";  
cmd.SchemaPath = "customersSchema.xml";  
DataSet ds = new DataSet();  
SqlXmlAdapter adapter = new SqlXmlAdapter(cmd);  
adapter.Fill(ds);
```

Gọi một .NET Component từ một COM Component

Trong hôm nay i-Today sẽ tạo một dự án demo để làm rõ các gọi .NET Component từ COM client.

Bước 1:

Tạo một Strong Name cho .NET Component (Strong name là một tên không trùng được tạo bởi hashing một khoá 128 bit và tên của Assembly (chúng ta đặt tên là COMInterOp)). Để tạo Strong name bạn dùng tool **sn**.

Để tạo một tập tin khoá tên là COMInterOp.snk, bạn thực hiện dòng lệnh:

```
sn -k COMInterOp.snk
```

Bước 2:

Bây giờ chúng ta sẽ tạo một .NET Assembly chỉ bao gồm một class CEmp với các thuộc tính (FirstName, LastName, DOB) và phát sinh các sự kiện.

Sau đây là dòng lệnh để tạo một assembly sử dụng strong name

```
vbc /out:COMInterOp.dll /t:library /keyfile:COMInterOp.snk CEmp.vb
```

VB.NET CEmp class code

```
Imports System
Imports Microsoft.VisualBasic
Imports System.Runtime.InteropServices

<InterfaceTypeAttribute(ComInterfaceType.InterfaceIsIDispatch)> _
Public Interface evtSenior
    Sub Senior()
End Interface

<ComSourceInterfacesAttribute("evtSenior")> _
Public Class CEmp
    Private mstrFirstName As String
    Private mstrLastName As String
    Private mdtDOB As Date
    Public Event Senior()
    Public Property FirstName() As String
        Get
            FirstName = mstrFirstName
        End Get
        Set(ByVal Value As String)
            mstrFirstName = Value
        End Set
    End Property
    Public Property LastName() As String
        Get
            LastName = mstrLastName
        End Get
        Set(ByVal Value As String)
            mstrLastName = Value
        End Set
    End Property
    Public Property DOB() As Date
        Get
            DOB = mdtDOB
        End Get
        Set(ByVal Value As Date)
            mdtDOB = Value
            If DateDiff(DateInterval.Year, Value, Now) > 60 Then
                RaiseEvent Senior()
            End If
        End Set
    End Property
End Class
```

Bước 3:

Một khi assembly được tạo chúng ta tạo một Type library để COM Client có thể sử dụng Assembly. Chúng ta theo các option sau cho COM:

- **Type Library Exporter**

Sử dụng Type Library Exporter (Tlbexp.exe) các lớp và giao diện (interface) được chứa trong một assembly được chuyển thành dạng một COM Lib. Một khi TypeLib được tạo, COM client có thể tạo một instance của các lớp .NET và gọi các phương thức như chúng là một đối tượng COM.

- **TypeLibConverter Class**
TypeLibConverter Class của System.Runtime.InteropServices namespace cung cấp các phương thức để chuyển một assembly thành một TypeLib.
- **Assembly Registration Tool**
Assembly Registration Tool (Regasm.exe), đọc metadata trong một assembly và thêm các mục cần thiết vào registry. Assembly Registration tool có thể generate và đăng ký một loại thư viện khi bạn sử dụng /tlb: option. COM clients yêu cầu các loại thư viện đó khi cài đặt vào trong Windows registry. Nếu không sử dụng option này, Regasm.exe chỉ đăng ký như là một assembly, không như type library.
- **The .NET Services Installation Tool (Regsvcs.exe)** (xem thêm MSDN)

Trong ví dụ của chúng ta sẽ sử dụng RegAsm.exe để tạo TypeLib từ các class và Interface được định nghĩa trong COMInterOp.dll.

```
regasm ComInterOp.dll /tlb:ComInterOp.tlb
```

Bước 4:

Bây giờ .NET component (COMInterOp.dll) nên được cài vào GAC (global assembly cache) để làm việc với COM Code

```
dir>Gacutil -i COMInterOp.dll
```

Bước 5: Sử dụng

COM Component

```
'Class Emps  
Option Explicit
```

```
Private Emps As Scripting.Dictionary
```

```
Private Sub Class_Initialize()  
Set Emps = New Scripting.Dictionary  
Dim objEmp As CEmp  
Set objEmp = New CEmp  
objEmp.InitMe "John", "Doe", "01/01/1970"  
Emps.Add 0, objEmp  
Set objEmp = New CEmp  
objEmp.InitMe "Mike", "Edwards", "01/01/1941"  
Emps.Add 1, objEmp  
Set objEmp = New CEmp  
objEmp.InitMe "Debra", "Bunn", "01/01/1930"  
Emps.Add 2, objEmp  
End Sub
```

```
Public Function PrintEmps() As String  
PrintEmps = PrintBool(True) & PrintBool(False)  
End Function
```

```

Public Function PrintBool(ByVal xblnSeniors As Boolean) As String
    Dim intCount As Integer
    Dim objEmp As CEmp
    Dim strPrint As String
    For intCount = 0 To Emps.Count - 1
        Set objEmp = Emps(intCount)
        If xblnSeniors = objEmp.IsSenior Then
            strPrint = strPrint & PrintEmp(objEmp) & Chr(13)
        End If
    Next intCount
    PrintBool = strPrint
End Function

```

```

Private Function PrintEmp(ByVal xobjEmp As CEmp) As String
    Dim strPrint As String
    strPrint = xobjEmp.FirstName & Chr(9) & xobjEmp.LastName
    PrintEmp = strPrint
End Function
'End Class Emps

```

```

'Class Emp
Option Explicit

```

```

Private mblnIsSenior As Boolean
Private WithEvents mobjEmp As ComInterOp.CEmp

```

```

Public Sub InitMe(ByVal xstrFName As String, _
    ByVal xstrLName As String, ByVal xdtDOB As Date)
    Set mobjEmp = New ComInterOp.CEmp
    With mobjEmp
        .FirstName = xstrFName
        .LastName = xstrLName
        .DOB = xdtDOB
    End With
End Sub

```

```

Public Property Get FirstName() As String
    FirstName = mobjEmp.FirstName
End Property

```

```

Public Property Get LastName() As String
    LastName = mobjEmp.LastName
End Property

```

```

Public Property Get IsSenior() As Boolean
    IsSenior = mblnIsSenior
End Property

```

```

Private Sub mobjEmp_Senior()
    mblnIsSenior = True
End Sub
'End Class Emp

```

Xuất(Generate) động (Dynamically) tập tin PDF sử dụng ASP.NET

Hiện nay có rất nhiều cách để xuất động các tập tin PDF. Cách phổ biến được biết là sử dụng ASP với Acrobat Full Version (4.0 hoặc 5.0) và Acrobat FDF Toolkit. Với Microsoft.NET rất nhiều lập trình viên đang lúng túng để thực hiện công việc này. i-Today xin hướng dẫn các bạn cách làm.

Các công cụ

1. Adobe Acrobat 5.0 Full Version, không phải Acrobat Reader 5.0
2. [Acrobat FDF Toolkit Version 5](#), free downloaded
3. Microsoft .NET Framework SDK with Service Pack 1

Platform

Windows 2000 Server (Service Pack 2), Internet Information Server 5.0

Cài đặt

1. Cài đặt Adobe Acrobat 5.0 Full Version.
2. Go to <http://partners.adobe.com/asn/developer/acrosdk/forms.html> để download the Acrobat FDF Toolkit package.
3. Cài đặt theo hướng dẫn <http://partners.adobe.com/asn/developer/acrosdk/docs/fdfk/FDFtkRef.pdf>. Giải nén Acrobat FDF Toolkit package, tìm files: FdfAcX.dll và FdfTk.dll. Chép vào thư mục \WINNT\system32 , và đăng ký Regsvr32 FdfAcX.dll.
4. Tạo .NET compatible wrapper cho FdfAcX.dll dùng TlbImp.exe(Type Library Importer). Trong cửa sổ Command Window, đánh:
tlbimp FdfAcX.dll /out:FdfAcX_NET.dll
5. Đặt FdfAcX_NET.dll đã được generated CLR assembly vào thư mục bin của chương trình của bạn. Nên nhớ các tập tin được gen bởi Tlbimp.exe cần đặt trong thư mục chương trình\bin của ASP.NET.

Trong ASP, dùng VBScript

```
Set FdfAcX = Server.CreateObject(FdfApp.FdfApp)
FdfAcX.FDFSetFile http://www.yourserver.com/test.pdf
FdfAcX.FDFSetValue txtMemo, This is a test, false
FdfAcX.FDFSaveToFile C:\temp\test.fdf
FdfAcX.FDFClose
Set FdfAcX = nothing
```

ASP.NET

```
<%@ Import Namespace=FdfAcX_NET %>
FdfAppClass FdfAcX_App = new FdfAppClass();
FdfDoc FdfAcX_Doc = (FdfDoc)FdfAcX_App.FDFCreate();
FdfAcX_Doc.FDFSetFile(http://www.yourserver.com/test.pdf);
FdfAcX_Doc.FDFSetValue(txtMemo, This is a test, false);
FdfAcX_Doc.FDFSaveToFile(@c:\temp\test.fdf);
FdfAcX_Doc.FDFClose();
```


VB.NET:

```
Dim FdfAcX_App As FdfAppClass
FdfAcX_App = new FdfAppClass()
Dim FdfAcX_Doc As FdfDoc
FdfAcX_Doc = FdfAcX_App.FDFCreate
FdfAcX_Doc.FDFSetFile(http://www.yourserver.com/test.pdf)
FdfAcX_Doc.FDFSetValue(txtMemo, This is a test , false)
FdfAcX_Doc.FDFSaveToFile(c:\temp\test.fdf)
FdfAcX_Doc.FDFClose
```

Truyền file được gen đến người sử dụng

```
<script language=C# runat=server>
protected void Page_Load(Object Src, EventArgs E)
{
    if (!IsPostBack) {
        Response.ContentType=Application/vnd.fdf;
        Response.WriteFile(@c:\temp\test.fdf);
        Response.End();
    }
}
```

Tạo một Pop-up Calendar (ASP.NET, VB.NET)

Bạn sử dụng Calendar Control và Panel Control. Panel Control được sử dụng để tạo hiệu ứng float trên trang. Bằng cách đặt calendar control trong Panel Control, bạn có thể dễ dàng xác định vị trí và thời điểm hiển thị Panel Control.

Tạo User Class

Điều đầu tiên cần làm là tạo một tập tin class có chứa Panel và Calendar Server control.

[popUpCalendar.ascx]

```
<%@ Control Language="vb" AutoEventWireup="false" Codebehind="popUpCalendar.ascx.vb"
Inherits="CalendarExample.popUpCalendar" %>
```

```
<asp:panel id="pnlCalendar" style="Z-INDEX: 101; LEFT: 0px; POSITION: absolute; TOP: 0px"
runat="server" Height="86px" Width="145px">
```

```
<asp:Calendar id="Calendar1" runat="server" Height="86" Width="145"
BackColor="White" BorderColor="Black" BorderStyle="Solid"
NextMonthText="<IMG src='monthright.gif' border='0'>"
PrevMonthText="<IMG src='monthleft.gif' border='0'>"
<TodayDayStyle BackColor="#FFFFFFC0"></TodayDayStyle>
<DayStyle Font-Size="8pt" Font-Names="Arial"></DayStyle>
<DayHeaderStyle Font-Size="10pt" Font-Underline="True" Font-Names="Arial"
BorderStyle="None" BackColor="#E0E0E0"></DayHeaderStyle>
<SelectedDayStyle Font-Size="8pt" Font-Names="Arial" Font-Bold="True"
ForeColor="White" BackColor="Navy"></SelectedDayStyle>
<TitleStyle Font-Size="10pt" Font-Names="Arial" Font-Bold="True"
```

```

    ForeColor="White" BackColor="Navy"></TitleStyle>
    <OtherMonthDayStyle ForeColor="Gray"></OtherMonthDayStyle>
</asp:Calendar>

</asp:panel>

```

Sau đó là tạo code để phục vụ việc hiển thị

[popUpCalendar.ascx.vb]

```

Public Class popUpCalendar : Inherits System.Web.UI.UserControl
Protected WithEvents Calendar1 As System.Web.UI.WebControls.Calendar
Protected WithEvents pnlCalendar As System.Web.UI.WebControls.Panel

#Region " Web Form Designer Generated Code "

    'This call is required by the Web Form Designer.
    <System.Diagnostics.DebuggerStepThrough(> Private Sub InitializeComponent()

    End Sub

    Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Init
        InitializeComponent()
    End Sub

#End Region

Public Sub displayCalendar(ByVal sCalToolText As String, _
ByVal dSelectedDate As Date, _
ByVal sDateFieldName As String, _
ByVal iTop As Integer, _
ByVal iLeft As Integer)
    *****
    'Hiển thị và dấu calendar
    *****
    If pnlCalendar.Visible = True And Calendar1.Attributes.Item("selectedfield") <> sDateFieldName
Then
        hideCalendar()
    End If
    If pnlCalendar.Visible = False Then
        pnlCalendar.Style.Item("top") = iTop
        pnlCalendar.Style.Item("left") = iLeft
        If IsDate(dSelectedDate) Then
            Calendar1.SelectedDate = dSelectedDate
            Calendar1.VisibleDate = dSelectedDate
        Else
            Calendar1.SelectedDate = #12:00:00 AM#
            Calendar1.VisibleDate = Now
        End If
        Calendar1.ToolTip = sCalToolText
        Calendar1.Attributes.Item("SelectedField") = sDateFieldName
        pnlCalendar.Visible = True
    Else

```

```

hideCalendar()
End If

End Sub

Public Sub Calendar1_SelectionChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Calendar1.SelectionChanged
    Dim txtDate As TextBox
    txtDate = Page.FindControl(Calendar1.Attributes.Item("SelectedField"))
    txtDate.Text = Calendar1.SelectedDate
    hideCalendar()
End Sub

Public Sub hideCalendar()
    pnlCalendar.Visible = False
End Sub
End Class

```

Sau đây là ví dụ về cách sử dụng User Control trên

[dispCalendar.aspx]

```

<%@ Register TagPrefix="sk" TagName="popUpCalendar" src="popUpCalendar.ascx"%>
<%@ Page Language="vb" AutoEventWireup="false" Codebehind="dispCalendar.aspx.vb"
Inherits="CalendarExample.dispCalendar" %>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<title>dispCalendar</title>
<meta name="GENERATOR" content="Microsoft Visual Studio.NET 7.0">
<meta name="CODE_LANGUAGE" content="Visual Basic 7.0">
<meta name="vs_defaultClientScript" content="JavaScript">
<meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
</HEAD>
<body MS_POSITIONING="GridLayout">
<form id="Form1" method="post" runat="server">

<sk:popUpCalendar id="myCalendar" runat="server" />

<H3>Example Use of Custom Calendar Control</H3>
<table>
<tr>
<td width="75">
Start Date:
</td>
<td>
<asp:TextBox id="txtStartDate" runat="server" Width="86px"></asp:TextBox>
</td>
<td>
<asp:ImageButton id="btnStartDate" runat="server"
ImageUrl="calendar.gif"></asp:ImageButton>
</td>

```

```

</tr>
<tr>
  <td width="75">
    End Date:
  </td>
  <td>
    <asp:TextBox id="txtEndDate" runat="server" Width="86px"></asp:TextBox>
  </td>
  <td>
    <asp:ImageButton id="btnEndDate" runat="server"
ImageUrl="calendar.gif"></asp:ImageButton>
  </td>
</tr>
</table>
</form>
</body>
</HTML>

```

[dispCalendar.aspx.vb]

```

Public Class dispCalendar
  Inherits System.Web.UI.Page
  Protected WithEvents txtStartDate As System.Web.UI.WebControls.TextBox
  Protected WithEvents btnEndDate As System.Web.UI.WebControls.ImageButton
  Protected WithEvents btnStartDate As System.Web.UI.WebControls.ImageButton
  Protected WithEvents txtEndDate As System.Web.UI.WebControls.TextBox
  Protected WithEvents myCalendar As popUpCalendar

  #Region " Web Form Designer Generated Code "

  'This call is required by the Web Form Designer.
  <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()

  End Sub

  Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs)
  Handles MyBase.Init
    'CODEGEN: This method call is required by the Web Form Designer
    'Do not modify it using the code editor.
    InitializeComponent()
  End Sub

  #End Region

  Private Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
  MyBase.Load
    If Not IsPostBack Then
      'Hide the calendar on initial page load
      myCalendar.hideCalendar()
    End If
  End Sub

  Private Sub btnStartDate_Click(ByVal sender As System.Object, ByVal e As
  System.Web.UI.ImageClickEventArgs) Handles btnStartDate.Click
    Dim dSelDate As Date

```

```

If IsDate(txtStartDate.Text) Then
    dSelDate = txtStartDate.Text
End If
myCalendar.displayCalendar("Select a start date", dSelDate, "txtStartDate", 59, 220)

End Sub

Private Sub btnEndDate_Click(ByVal sender As System.Object, ByVal e As
System.Web.UI.ImageClickEventArgs) Handles btnEndDate.Click
    Dim dSelDate As Date

    If IsDate(txtEndDate.Text) Then
        dSelDate = txtEndDate.Text
    End If
    myCalendar.displayCalendar("Select an end date", dSelDate, "txtEndDate", 86, 220)

End Sub

End Class

```

Đổi địa chỉ IP của máy Local sử dụng VB.NET và C#

Tất cả các thông tin setting thông số mạng đều được lưu trong Registry và để thay đổi không có gì dễ hơn là chúng ta thay đổi các thông tin trong Registry.

Bước 1:

Mở HKEY_LOCAL_MACHINE và mở khoá SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkCards\1.

1 là card mạng đầu tiên. Nếu bạn nhiều card trên máy chúng sẽ hiển thị bằng các con số. Trong khoá này có giá trị gọi là mà chúng ta cần lưu ở bước tiếp theo. Bây giờ bạn đóng khoá này lại.

Bước 2:

Mở lại HKEY_LOCAL_MACHINE và mở khoá SYSTEM\CurrentControlSet\Services\#SERVICE-NAME#\Parameters\Tcpip. Và chắc bạn mở khoá này với quyền Write.

Bước 3:

Bây giờ các bạn có thể thay đổi địa chỉ IP, DefaultGateway... các giá trị đều lưu dưới giá trị nhị phân vì vậy bạn phải chuyển thành nhị phân trước khi lưu vào Registry. (Dùng hàm GetBytes)
 Now you can change the IP address for the IPAddress, DefaultGateway keys etc. The value type of these keys is binary so you must make sure that you do not write a string to the registry or it will change its value type. Instead, use the GetBytes() method of the Encoding class to write the bytes.

```

Imports System
Imports System.Text
Imports Microsoft.Win32

```

```
Module ChangeIP
```

```

Sub Main()
    Dim regKey As RegistryKey
    Dim strServiceName As String

```

```
regKey =
```

```

Registry.LocalMachine.OpenSubKey(SOFTWARE\Microsoft\WindowsNT\CurrentVersion\NetworkCards\1)
    strServiceName = regKey.GetValue(ServiceName)
    regKey.Close()
    regKey = Registry.LocalMachine.OpenSubKey(SYSTEM\CurrentControlSet\Services\ &
strServiceName & \Parameters\Tcpip, True)
    regKey.SetValue(IPAddress, Encoding.ASCII.GetBytes(10.1.1.1\00))
    regKey.Close()
End Sub
End Module

```

Sau đây là những lưu ý khi thực hiện kỹ thuật này:

1. Bạn phải có quyền đọc và viết vào Registry. Nếu trong trường hợp bạn cần xác nhận quyền để thực thi thì sử dụng lớp RegistryPermission.
2. Nếu bạn không sử dụng Windows NT/2000 thì đổi 'Windows NT' thành 'Windows' trong bước 1
3. Nếu bạn đang dùng DHCP thì lưu ý rằng IPAddress sẽ là 0.0.0.0 và bạn sẽ cần thay đổi giá trị của EnableDHCP thành 0
4. Khi bạn xác định một địa chỉ IP để ghi vào Registry, nhớ thêm 2 giá trị null vào cuối của chúng
5. Cần boot lại máy trước khi có tác dụng.

Sử dụng SQL Server Images trong các trang ASP.NET (C#,ASP.NET)

Thường khi chúng ta hiển thị các hình ảnh trong trang ASP.NET chỉ sử dụng tag . Còn các hình ảnh được lưu trong các trường blod của SQL Server? Chúng tôi đã hướng dẫn các bạn cách để hiển thị các hình trong ASP. Nay chúng tôi sẽ hướng dẫn các bạn sử dụng ASP.NET.

<img src=<%# "image.aspx?id=" + DataBinder.Eval (Container.DataItem, "employeeid") %> />
Tập tin image.aspx sẽ tạo và output một graphic stream dựa vào các khoá chính được truyền bởi id. Trong .NET, bạn có thể nhận trường blod qua

```
byte [] img = (byte[]) command.ExecuteScalar();
```

Khi bạn có các bits, bạn có thể gửi chúng đến như một graphic object. Nhưng đầu tiên phải set Response.ContentType = "image/gif";

```

MemoryStream ms = new MemoryStream();
ms.Write(img, 0, img.Length);
Bitmap bmp = null;
bmp = new Bitmap(ms);

```

Đến thời điểm này, bạn hầu như đã làm xong. Bạn cần chuyển sang dạng format dạng hình ảnh mong muốn

```
bmp.Save(Response.OutputStream,ImageFormat.Gif);
```

Toàn bộ tập tin image.aspx (sử dụng database Northwind để thử nghiệm)

```

<%@ Page Language="C#" %>
<%@ Import Namespace="System" %>
<%@ Import Namespace="System.Data.SqlClient" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System.Drawing.Imaging" %>
<%@ Import Namespace="System.Drawing" %>
<html>
<script runat="server">
private void Page_Load(object sender,
System.EventArgs e)
{

```

```

SqlConnection cn;
cn = new SqlConnection("DATABASE=northwind;SERVER=localhost;UID=sa;");
String cmdText = "SELECT photo FROM Employees WHERE employeeid=" +
Request["id"].ToString();
SqlCommand cmd = new SqlCommand(cmdText, cn);
MemoryStream ms = new MemoryStream();
int offset = 78;
cn.Open();
byte [] img = (byte[]) cmd.ExecuteScalar();
ms.Write(img, offset, img.Length-offset);
cn.Close();
Bitmap bmp = null;
bmp = new Bitmap(ms);
Response.ContentType = "image/gif";
bmp.Save(Response.OutputStream, ImageFormat.Gif);
ms.Close();
}
</script>
</html>

```

Ghi vào Event Log (ASP.NET)

Chúng tôi đã giới thiệu cho các bạn cách đọc từ Event Log. Hôm nay chúng tôi xin hướng dẫn các bạn cách ghi vào Event Log. Chúng ta sẽ tạo một trang quản lý các lỗi khi trang ASP.NET sinh lỗi

```

<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.SQL" %>
<%@ Import Namespace="System.Diagnostics" %>

<script language="c#" runat="server">
void Page_Load(Object source, EventArgs e)
{
    try {
        SqlConnection objConn;
        objConn = new SqlConnection("server=localhost;uid=foo;pwd=bar;database=pubs");
        objConn.Open();
    }
    catch (Exception eError)
    {
        RecordError(eError, EventLogEntryType.Error);
    }
}

void RecordError(Exception eError, EventLogEntryType enumType)
{
    const String strSource = "ASP.NET", strLogName = "System";
    EventLog objLog = new EventLog(strLogName);
    objLog.Source = strSource;
    objLog.WriteEntry(eError.Message, enumType);
}

```

```
}  
</script>
```

Đoạn code sẽ ghi vào Event Log khi không kết nối vào SQL Server (dĩ nhiên là không được vì chúng ta đang thử để báo lỗi)

Đọc từ Event Log (ASP.NET)

.NET Framework có một số lớp dùng để đọc và viết vào event log. Tất cả được lưu trong System.Diagnostics namespace. Sau đây chúng tôi xin trình bày một đoạn code ASP.NET đơn giản để hiển thị các mục lỗi trong event log trong được lưu giữ trong System Log.

```
<%@ Import Namespace="System.Diagnostics" %>  
<%@ Import Namespace="System.Drawing" %>  
<script language="VB" runat="server">  
Sub Page_Load(source as Object, e as EventArgs)  
If Not Page.IsPostBack Then  
DisplayEventLog("System")  
End If  
End Sub  
Sub btnSubmit_OnClick(source as Object, e as EventArgs)  
DisplayEventLog(IstLog.SelectedItem.Value)  
End Sub  
Sub btnClear_OnClick(source as Object, e as EventArgs)  
Dim objEventLog as New EventLog(IstLog.SelectedItem.Value)  
objEventLog.Clear()  
End Sub  
Sub DisplayEventLog(strLogName as String)  
Dim objRow as New TableRow  
Dim objCell as New TableCell  
objCell.BackColor = Color.Bisque  
objCell.HorizontalAlign = HorizontalAlign.Center  
objCell.Text = "Type"  
objRow.Cells.Add(objCell)  
objCell = New TableCell  
objCell.BackColor = Color.Bisque  
objCell.HorizontalAlign = HorizontalAlign.Center  
objCell.Text = "Date"  
objCell = New TableCell  
objCell.BackColor = Color.Bisque  
objCell.HorizontalAlign = HorizontalAlign.Center  
objCell.Text = "Time"  
objRow.Cells.Add(objCell)  
objCell = New TableCell  
objCell.BackColor = Color.Bisque  
objCell.HorizontalAlign = HorizontalAlign.Center  
objCell.Text = "Source"  
objRow.Cells.Add(objCell)  
objCell = New TableCell  
objCell.BackColor = Color.Bisque  
objCell.HorizontalAlign = HorizontalAlign.Center
```



```

objCell.Text = "User"
objRow.Cells.Add(objCell)
objCell = New TableCell
objCell.BackColor = Color.Bisque
objCell.HorizontalAlign = HorizontalAlign.Center
objCell.Text = "Computer"
objRow.Cells.Add(objCell)
tblLog.Rows.Add(objRow)
Dim objEventLog as EventLog = New EventLog(strLogName)
Dim objEntry as EventLogEntry
For Each objEntry in objEventLog.Entries
objRow = New TableRow
objCell = New TableCell
If objEntry.EntryType = EventLogEntryType.Error Then
objCell.BackColor = Color.Red
objCell.ForeColor = Color.White
objCell.Text = "Error"
ElseIf objEntry.EntryType = EventLogEntryType.Information Then
objCell.Text = "Information"
ElseIf objEntry.EntryType = EventLogEntryType.Warning Then
objCell.BackColor = Color.Yellow
objCell.Text = "Warning"
ElseIf objEntry.EntryType = EventLogEntryType.SuccessAudit Then
objCell.Text = "Success Audit"
ElseIf objEntry.EntryType = EventLogEntryType.FailureAudit Then
objCell.ForeColor = Color.Red
objCell.Text = "Failure Audit"
End If
objCell.HorizontalAlign = HorizontalAlign.Center
objRow.Cells.Add(objCell)
objCell = New TableCell
objCell.Text = objEntry.TimeGenerated.ToShortDateString()
objRow.Cells.Add(objCell)
objCell = New TableCell
objCell.Text = objEntry.TimeGenerated.ToLongTimeString()
objRow.Cells.Add(objCell)
objCell = New TableCell
objCell.Text = objEntry.Source
objRow.Cells.Add(objCell)
objCell = New TableCell
If objEntry.UserName <> Nothing then
objCell.Text = objEntry.UserName
Else
objCell.Text = "N/A"
End If
objRow.Cells.Add(objCell)
objCell = New TableCell
objCell.Text = objEntry.MachineName
objRow.Cells.Add(objCell)
tblLog.Rows.Add(objRow)
Next
End Sub
</script>

```

```

<html>
<body>
<form runat="server">
  <h1>Event Log Viewer</h1>
  <asp:listbox runat="server" id="lstLog" Rows="1">
    <asp:listitem>Application</asp:listitem>
    <asp:listitem>Security</asp:listitem>
    <asp:listitem Selected="True">System</asp:listitem>
  </asp:listbox>
  <asp:button runat="server" id="btnSubmit" Text="Display Event Log"
  OnClick="btnSubmit_OnClick" />
  <hr>
  <asp:table runat="server" id="tblLog" CellPadding="5"
  CellSpacing="0" GridLines="Both" Font-Size="10pt"
  Font-Name="Verdana" />
  <hr>
  <asp:button runat="server" id="btnClear" Text="Clear Event Log"
  OnClick="btnClear_OnClick" />
</form>
</body>
</html>

```

Mã hoá (VB)

```

Function Encrypt(ByVal inpt As String) As String
  Dim temp As String
  Dim tempA As String
  Dim Rand As String
  100:
  Randomize
  Rand = Right(Rnd, 3)
  rad = Left(Rand, 1)
  If Left(Rand, 1) = "-" Then
    GoTo 100
  End If
  For i = 1 To Len(inpt)
    crntASC = Asc(Mid(inpt, i, 1))
    tempA = ((crntASC) Xor (Rand + i + rad)) + (i + rad)
    If Len(tempA) = 4 Then
      temp = temp & tempA
    ElseIf Len(tempA) = 3 Then
      temp = temp & "0" & tempA
    ElseIf Len(tempA) = 2 Then
      temp = temp & "00" & tempA
    ElseIf Len(tempA) = 1 Then
      temp = temp & "000" & tempA
    End If
  Next i
  temp = Rand & temp

```

```
Encrypt = temp  
End Function
```

```
Function Decrypt(ByVal inpt As String) As String  
    Rand = Left(inpt, 3)  
    For i = 4 To (Len(inpt) - 3) Step 4  
        z = z + 1  
        tempA = Mid(inpt, i, 4)  
        tempA = ((tempA - (z + Left(Rand, 1))) Xor (Rand + z + Left(Rand, 1)))  
        temp = temp & Chr(tempA)  
    Next i  
    Decrypt = temp  
End Function
```

Nạp chồng (C#)

Đây là một ví dụ SQL Server Data Access với nhiều câu SELECT được overload và nhiều cách khác nhau để thực thi stored procedures. Đây là một cách để bạn tham khảo để viết các lớp uyển chuyển hơn.

```
public AuthorData(string connection)  
{  
    this.connection = connection;  
}  
.....  
public SqlDataReader Select(string commandName)  
{  
    SqlDataReader dr =null;  
    try  
    {  
        SqlConnection cnn =new SqlConnection(this.connection);  
        cnn.Open();  
        SqlCommand cmd =new SqlCommand(commandName,cnn);  
        cmd.CommandType = CommandType.StoredProcedure;  
        cmd.Execute(out dr);  
        cmd.ActiveConnection =null;  
    }  
    catch(Exception e)  
    {  
        ErrorLog errLog =new ErrorLog();  
        errLog.LogError(e.Message, commandName);  
    }  
    return(dr);  
}  
.....  
public void Select(out SqlDataReader dr, string commandName)  
{  
    dr =null;  
    try{  
        SqlConnection cnn =new SqlConnection(this.connection);  
        cnn.Open();  
        SqlCommand cmd =new SqlCommand(commandName,cnn);  
        cmd.CommandType = CommandType.StoredProcedure;  
        cmd.Execute(out dr);  
        cmd.ActiveConnection =null;
```

```

}
catch(Exception e){
    ErrorLog errLog =new ErrorLog();
    errLog.LogError(e.Message, commandName);
}
}
}
.....
public void Insert(string commandName, params object[] args)
{
try
{
    SqlConnection cnn =new SqlConnection(this.connection);
    SqlParameter parm =new SqlParameter();
    cnn.Open();
    SqlCommand cmd =new SqlCommand(commandName,cnn);
    cmd.CommandType = CommandType.StoredProcedure;
    parm = cmd.Parameters.Add(new SqlParameter("@au_id", SqlDbType.VarChar, 11));
    parm.Direction = ParameterDirection.Input;
    cmd.Parameters["@au_id"].Value = args[0];
    parm = cmd.Parameters.Add(new SqlParameter("@au_lname", SqlDbType.VarChar, 40));
    parm.Direction = ParameterDirection.Input;
    cmd.Parameters["@au_lname"].Value = args[1];
    parm = cmd.Parameters.Add(new SqlParameter("@au_fname", SqlDbType.VarChar, 20));
    parm.Direction = ParameterDirection.Input;
    cmd.Parameters["@au_fname"].Value = args[2];
    parm = cmd.Parameters.Add(new SqlParameter("@Phone", SqlDbType.Char, 12));
    parm.Direction = ParameterDirection.Input;
    cmd.Parameters["@Phone"].Value = args[3];
    parm = cmd.Parameters.Add(new SqlParameter("@Address", SqlDbType.VarChar, 40));
    parm.Direction = ParameterDirection.Input;
    cmd.Parameters["@Address"].Value = args[4];
    parm = cmd.Parameters.Add(new SqlParameter("@city", SqlDbType.VarChar, 20));
    parm.Direction = ParameterDirection.Input;
    cmd.Parameters["@city"].Value = args[5];
    parm = cmd.Parameters.Add(new SqlParameter("@state", SqlDbType.Char, 2));
    parm.Direction = ParameterDirection.Input;
    cmd.Parameters["@state"].Value = args[6];
    parm = cmd.Parameters.Add(new SqlParameter("@zip", SqlDbType.VarChar, 5));
    parm.Direction = ParameterDirection.Input;
    cmd.Parameters["@zip"].Value = args[7];
    parm = cmd.Parameters.Add(new SqlParameter("@contract", SqlDbType.Bit ));
    parm.Direction = ParameterDirection.Input;
    cmd.Parameters["@contract"].Value = args[8];
    cmd.ExecuteNonQuery();
    cmd.ActiveConnection =null;
}
catch(Exception e)
{
    ErrorLog errLog =new ErrorLog();
    errLog.LogError(e.Message, commandName);
}
}
}
.....
.....
public void Update(string commandName, params object[] args)
{

```

```

try
{
    SqlConnection cnn =new SqlConnection(this.connection);
    SqlParameter parm =new SqlParameter();
    cnn.Open();
    SqlCommand cmd =new SqlCommand(commandName,cnn);
    cmd.CommandType = CommandType.StoredProcedure;
    parm = cmd.Parameters.Add(new SqlParameter("@au_id", SqlDbType.VarChar, 11));
    parm.Direction = ParameterDirection.Input;
    cmd.Parameters["@au_id"].Value = args[0];
    parm = cmd.Parameters.Add(new SqlParameter("@au_lname", SqlDbType.VarChar, 40));
    parm.Direction = ParameterDirection.Input;
    cmd.Parameters["@au_lname"].Value = args[1];
    parm = cmd.Parameters.Add(new SqlParameter("@au_fname", SqlDbType.VarChar, 20));
    parm.Direction = ParameterDirection.Input;
    cmd.Parameters["@au_fname"].Value = args[2];
    parm = cmd.Parameters.Add(new SqlParameter("@Phone", SqlDbType.Char, 12));
    parm.Direction = ParameterDirection.Input;
    cmd.Parameters["@Phone"].Value = args[3];
    parm = cmd.Parameters.Add(new SqlParameter("@Address", SqlDbType.VarChar, 40));
    parm.Direction = ParameterDirection.Input;
    cmd.Parameters["@Address"].Value = args[4];
    parm = cmd.Parameters.Add(new SqlParameter("@city", SqlDbType.VarChar, 20));
    parm.Direction = ParameterDirection.Input;
    cmd.Parameters["@city"].Value = args[5];
    parm = cmd.Parameters.Add(new SqlParameter("@state", SqlDbType.Char, 2));
    parm.Direction = ParameterDirection.Input;
    cmd.Parameters["@state"].Value = args[6];
    parm = cmd.Parameters.Add(new SqlParameter("@zip", SqlDbType.VarChar, 5));
    parm.Direction = ParameterDirection.Input;
    cmd.Parameters["@zip"].Value = args[7];
    parm = cmd.Parameters.Add(new SqlParameter("@contract", SqlDbType.Bit ));
    parm.Direction = ParameterDirection.Input;
    cmd.Parameters["@contract"].Value = args[8];
    cmd.ExecuteNonQuery();
    cmd.ActiveConnection =null;
}
catch(Exception e)
{
    ErrorLog errLog =
    new ErrorLog();
    errLog.LogError(e.Message, commandName);
}
}
.....

```

```

public void Delete(string commandName, string recordID)
{
    try
    {
        SqlConnection cnn =new SqlConnection(this.connection);
        SqlParameter parm =new SqlParameter();
        cnn.Open();
        SqlCommand cmd =new SqlCommand(commandName,cnn);
        cmd.CommandType = CommandType.StoredProcedure;

```

```

    parm = cmd.Parameters.Add(new SqlParameter("@au_id", SqlDbType.VarChar, 11));
    parm.Direction = ParameterDirection.Input;
    cmd.Parameters["@au_id"].Value = recordID;
    cmd.ExecuteNonQuery();
    cmd.ActiveConnection = null;
}
catch(Exception e)
{
    ErrorLog errLog =
    new ErrorLog();
    errLog.LogError(e.Message, commandName);
}
}
}
.....

```

```

public void ExecuteProc(string commandName, params object[] args)
{
    try
    {
        ADOConnection cnn =new ADOConnection(this.connection);
        cnn.Open();
        ADOCommand cmd =new ADOCommand();
        cmd.ActiveConnection = cnn;
        cmd.CommandText = commandName;
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.ResetParameters();
        int i = 0;
        foreach( ADOParameter prm in cmd.Parameters)
        {
            cmd.Parameters[i].Value = args[i];
            i++;
        }
        cmd.ExecuteNonQuery();
    }
    catch(Exception e)
    {
        ErrorLog errLog =
        new ErrorLog();
        ErrLog.LogError(e.Message, "Method: ExecuteProc, Stored Proc: " + commandName);
    }
}
string connection;
}

```

Vòng lặp trong một Form (.NET)

Đây là một đoạn code giúp bạn nhanh chóng hiển thị tất cả các giá trị của một form được post đến một trang .NET. Rất tiện dụng khi bạn xử lý thông tin được post.

```

Sub ShowFormColl()
Dim i as integer
Dim itemName as String
Dim itemValue as String

```

```

For i = 0 to Request.Form.Count - 1
itemName = Request.Form.AllKeys(i)
itemValue = Request.Form.GetValues(i)(0)
Response.Write ("<BR>" & itemName & " : " & itemValue )
Next
End Sub

```

Chạy một chương trình trên Server thông qua một trang ASP

Đây là một hàm khá hữu dụng (và nguy hiểm nếu có mục đích xấu) để bạn có thể thực thi một chương trình trên server thông qua một trang ASP. Hàm sẽ phát sinh một tiến trình trên server với các thông số nhận được.

Để chạy được chương trình yêu cầu Server phải cài đặt scripting và được phân quyền

Cú pháp:

Shell command

Ví dụ:

Để mở IIS trên server

```
<% Shell "c:\windows\system32\inetsrv\iis.msc" %>
```

Mở Notepad trên server

```
<% Shell "notepad" %>
```

Đăng ký một dll trên server

```
<% Shell "Regsvr32 C:\WINNT\System32\some.dll" %>
```

Mã nguồn::

```

<%
Private Sub Shell(byVal command)
dim wshShell, boolErr, strErrDesc
On Error Resume Next
Set wshShell = CreateObject("WScript.Shell")
wshShell.Run command
if Err Then
boolErr = True
strErrDesc = Err.Description
end if
Set wshShell = Nothing
On Error GoTo 0
if boolErr then Err.Raise 5105, "Shell Statement", strErrDesc
End Sub
%>

```

Mở tập tin Excell bằng ADO.NET

Ngày nay các kết xuất các dữ liệu thường đưa ra nhiều loại, trong đó Excell là một trong những loại rất thường được dùng. Chúng tôi xin giới thiệu cách kết xuất dữ liệu ra tập tin Excell

```
strConn = @"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\TEMP\TEST.XLS;Extended Properties=Excel 8.0;";
```

```
string sList="myspreadsheetname";
```

```
OleDbConnection oConn = new OleDbConnection();
```

```
oConn.ConnectionString = strConn;
```

```

oConn.Open();

OleDbDataAdapter oCmd = new OleDbDataAdapter("SELECT * FROM [" + sList + "$]", oConn);
DataSet oDS = new DataSet();
oCmd.Fill(oDS);

foreach(DataRow oRow in oDS.Tables[0].Rows)
{
Response.Write("Row: " + oRow["COLUMNNAME"].ToString() + "<br>");
}

if (oConn.State == ConnectionState.Open) { oConn.Close(); }

```

SCROLL WORDS (VB)

Để test thử bạn tạo một command button vào một form và sử dụng code sau dưới một timer object:

```

Public Sub ScrollWords(ByVal StringToScroll$, ByVal TargetObjectCaption As Object)
Static S%
On Error GoTo ErrorHandler:
If S < Len(StringToScroll) Then
TargetObjectCaption.Caption = TargetObjectCaption.Caption & Mid(StringToScroll, S + 1, 1)
S = S + 1
Else
TargetObjectCaption.Caption = ""
S = 0
End If
Exit Sub
ErrorHandler:
MsgBox "Error: " & Err.Description, vbCritical + vbOKOnly, "Error " & Err.Number & " in
ScrollWords!"
End Sub

```

Phân tích chuỗi (C# & VB.NET)

Chúng tôi sẽ xây dựng một chương trình Web nhỏ và triển khai bằng C# và Vb.NET

```

<%@ Page language="c#" Codebehind="StringParser.aspx.cs" AutoEventWireup="false"
Inherits="HowTo.StringParser" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<HTML>
<HEAD>
<title>StringParser</title>
<meta name="GENERATOR" Content="Microsoft Visual Studio 7.0">
<meta name="CODE_LANGUAGE" Content="C#">
<meta name="vs_defaultClientScript" content="JavaScript">
<meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
</HEAD>
<body>
<form id="StringParser" method="post" runat="server">
<P>

```



```

<STRONG>Enter a Alphanumeric String:</STRONG>
<asp:RequiredFieldValidator id="RequiredFieldValidator1" runat="server"
ErrorMessage="* Required" ControlToValidate="TextBox1"></asp:RequiredFieldValidator>
<BR>
<asp:TextBox id="TextBox1" runat="server"></asp:TextBox>
<asp:Button id="btnGo" runat="server" Text="Go!"></asp:Button>
</P>
<P>
<asp:Label id="Label1" runat="server"></asp:Label>
</P>
<P>
<asp:Label id="Label2" runat="server"></asp:Label>
</P>
</form>
</body>
</HTML>

```

Web Form objects:

```

using System;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;

```

```

namespace HowTo

```

```

{
public class StringParser : System.Web.UI.Page
{
protected System.Web.UI.WebControls.TextBox TextBox1;
protected System.Web.UI.WebControls.Button btnGo;
protected System.Web.UI.WebControls.RequiredFieldValidator RequiredFieldValidator1;
protected System.Web.UI.WebControls.Label Label2;
protected System.Web.UI.WebControls.Label Label1;

```

```

private void btnGo_Click(object sender, System.EventArgs e)
{
System.Text.StringBuilder _string = new System.Text.StringBuilder();
System.Text.StringBuilder _int = new System.Text.StringBuilder();
char[] _text;

_text = TextBox1.Text.Trim().ToCharArray(0, TextBox1.Text.Trim().Length);

```

```

for (Int32 i = 0 ; i < _text.Length; i++)
{
try
{
Int32.Parse(_text[i].ToString());
_int.Append(_text[i].ToString());
}
catch
{
_string.Append(_text[i].ToString());
}
}
}

```

```
Label1.Text = "String: " + _string.ToString();
Label1.Text += "<br>Int32: " + _int.ToString();
```

```
Int32 _newInt = Int32.Parse(_int.ToString());
```

```
Label2.Text = "The Int32 value squared is: ";
Label2.Text += (_newInt * _newInt).ToString();
}
}
```

Upload một tập tin vào database sử dụng System.Data.OleDb

Chúng tôi đã từng giới thiệu với các bạn làm thế nào để upload một tập tin vào database bằng ngôn ngữ VB, hôm nay chúng tôi xin giới thiệu với các bạn cách upload một tập tin vào database trong .NET. Sử dụng Sql .NET Data Provider rất giống insert một mảng các byte vào Database sử dụng OLEDB

SQL Code:

```
CREATE TABLE [dbo].[Images] (
[ImageID] [int] IDENTITY (1, 1) NOT NULL ,
[Image] [image] NULL ,
[ContentType] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
[ImageDescription] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
[ByteSize] [int] NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
```

Web Form Code:

```
<%@ Page Inherits="UploadSample.Main" SRC="Upload.cs"%>
<HTML>
<BODY>
<FORM ENCTYPE="multipart/form-data" RUNAT="server" ID="Form1">
<h1>
File Upload To Database Using <FONT COLOR="BLUE">System.Data.OleDb</FONT>
</h1>
<TABLE RUNAT="server" WIDTH="700" ALIGN="left" ID="Table1">
<TR>
<TD>
<B>Upload File</B>
</TD>
<TD>
<INPUT TYPE="file" ID="UP_FILE" RUNAT="server" STYLE="Width:320" ACCEPT="text/*"
NAME="UP_FILE">
</TD>
</TR>
<TR>
<TD>
<b>Description of File</b>
</TD>
<TD>
<asp:TextBox RUNAT="server" WIDTH="239" ID="txtDescription" MAINTAINSTATE="false" />
</TD>
</TR>
<TR>
<TD>
```

```

<asp:Label RUNAT="server" ID="txtMessage" FORECOLOR="red" MAINTAINSTATE="false" />
</TD>
<TD>
<asp:Button RUNAT="server" WIDTH="239" ONCLICK="Button_Submit" TEXT="Upload Image"
/>
</TD>
</TR>
</TABLE>
</FORM>
</BODY>
</HTML>

```

Đăng sau WEB Form Code

```

namespace UploadSample {

public class Main : System.Web.UI.Page {

protected System.Web.UI.HtmlControls.HtmlInputFile UP_FILE;
protected System.Web.UI.WebControls.TextBox txtDescription;
protected System.Web.UI.WebControls.Label txtMessage;
protected System.Int32 FileLength = 0;

protected void Button_Submit(System.Object sender, System.EventArgs e) {
System.Web.HttpPostedFile UpFile = UP_FILE.PostedFile;
FileLength = UpFile.ContentLength;

try {

if (FileLength == 0) {

txtMessage.Text = "<b>* You must pick a file to upload</b>";

} else {

System.Byte[] FileByteArray = new System.Byte[FileLength];
System.IO.Stream StreamObject = UpFile.InputStream;
StreamObject.Read(FileByteArray,0,FileLength);
System.Data.OleDb.OleDbConnection Con = new
System.Data.OleDb.OleDbConnection("Provider=SQLOLEDB;Data Source=localhost;" +
"Integrated Security=SSPI;Initial Catalog=northwind");
System.String SqlCommand = "INSERT INTO Images (Image, ContentType, ImageDescription,
ByteSize) VALUES (?, ?, ?, ?)";
System.Data.OleDb.OleDbCommand OleDbCmdObj = new
System.Data.OleDb.OleDbCommand(SqlCommand, Con);
OleDbCmdObj.Parameters.Add("@Image", System.Data.OleDb.OleDbType.Binary,
FileLength).Value = FileByteArray;
OleDbCmdObj.Parameters.Add("@ContentType",
System.Data.OleDb.OleDbType.VarChar,50).Value = UpFile.ContentType;
OleDbCmdObj.Parameters.Add("@ImageDescription",
System.Data.OleDb.OleDbType.VarChar,100).Value = txtDescription.Text;
OleDbCmdObj.Parameters.Add("@ByteSize",
System.Data.OleDb.OleDbType.VarChar,100).Value = UpFile.ContentLength;
Con.Open();

```

```

OleDbCmdObj.ExecuteNonQuery();
Con.Close();
txtMessage.Text = "<p><b>* Your image has been uploaded</b>";

}

} catch (System.Exception ex) {

txtMessage.Text = ex.Message.ToString();

}
}
}
}
}

```

Bởi vì giới hạn của kiểu data type Image là 2,147,483,647 và hầu hết mọi người sẽ không upload một tập tin có kích thước lớn như vậy vào trong một database nhưng bởi vì không có OleDbType.Image chúng ta phải sử dụng OleDbType.Binary với giới hạn là 8000 Byte và chúng ta set kích thước trong ví dụ này:

```

OleDbCmdObj.Parameters.Add("@Image", System.Data.OleDb.OleDbType.Binary,
FileLength).Value = FileByteArray;

```

Thêm một trường tổng vào trong một DataGrid (ASP.NET)

Trong mẹo lập trình hôm nay chúng tôi sẽ hướng dẫn các bạn cách làm thế nào để chương trình tự động tính tổng của một cột trong DataGrid, và hiển thị tổng trong footer của DataGrid.

Bạn sẽ dùng một Web Form (calcTotals.aspx) và một đoạn code sau lớp tập tin này (calcTotals.aspx.cs)

Sau đây là code calcTotals.aspx:

```

<%@ Page Inherits="myApp.calcTotals" Src="20010731T0101.aspx.cs" %>
<html>
<body bgcolor="white">
<asp:DataGrid id="MyGrid" runat="server"
AutoGenerateColumns="False"
CellPadding="4" CellSpacing="0"
BorderStyle="Solid" BorderWidth="1"
Gridlines="None" BorderColor="Black"
ItemStyle-Font-Name="Verdana"
ItemStyle-Font-Size="9pt"
HeaderStyle-Font-Name="Verdana"
HeaderStyle-Font-Size="10pt"
HeaderStyle-Font-Bold="True"
HeaderStyle-ForeColor="White"
HeaderStyle-BackColor="Blue"
FooterStyle-Font-Name="Verdana"
FooterStyle-Font-Size="10pt"
FooterStyle-Font-Bold="True"
FooterStyle-ForeColor="White"
FooterStyle-BackColor="Blue"
OnItemDataBound="MyDataGrid_ItemDataBound"
ShowFooter="True">
<Columns>
<asp:BoundColumn HeaderText="Title" DataField="title" />
<asp:BoundColumn HeaderText="Price" DataField="price"
ItemStyle-HorizontalAlign="Right"

```

```
HeaderStyle-HorizontalAlign="Center" />
</Columns>
</asp:DataGrid>
</body>
</html>
```

Trong Web Form bạn dùng dấu @ để trang sử dụng code trong phần khai báo. thuộc tính SRC chỉ code sẽ được biên dịch sử dụng bộ biên dịch JIT.
Code trong lớp sẽ xử lý 2 sự kiện Page_Load event và OnItemDataBound và một phương thức Private là CalcTotal

```
using System;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Data;
using System.Data.SqlClient;

namespace myApp
{
    public class calcTotals : Page
    {
        protected DataGrid MyGrid;
        private double runningTotal = 0;
    }

    protected void Page_Load(object sender, EventArgs e)
    {
        SqlConnection myConnection = new
        SqlConnection("server=localhost;database=pubs;uid=sa;pwd=");
        SqlCommand myCommand = new SqlCommand("SELECT title, price FROM Titles WHERE price
        > 0", myConnection);

        try
        {
            myConnection.Open();
            MyGrid.DataSource = myCommand.ExecuteReader();
            MyGrid.DataBind();
            myConnection.Close();
        }
        catch(Exception ex)
        {
            HttpContext.Current.Response.Write(ex.ToString());
        }
    }

    private void CalcTotal(string _price)
    {
        try
        {
            runningTotal += Double.Parse(_price);
        }
        catch
        {
        }
    }
}
```

```
}
```

Sự kiện MyGrid_ItemDataBound

```
public void MyDataGrid_ItemDataBound(object sender, DataGridItemEventArgs e)
{
    if (e.Item.ItemType == ListItemType.Item || e.Item.ItemType == ListItemType.AlternatingItem)
    {
        CalcTotal( e.Item.Cells[1].Text );
        e.Item.Cells[1].Text = string.Format("{0:c}", Convert.ToDouble(e.Item.Cells[1].Text));
    }
    else if(e.Item.ItemType == ListItemType.Footer )
    {
        e.Item.Cells[0].Text="Total";
        e.Item.Cells[1].Text = string.Format("{0:c}", runningTotal);
    }
}
```

Truy cập thông tin của DataGrid (.NET)

Chúng tôi có một DataGrid gọi là dgAges, một Label gọi lblName, và một Label gọi lblAge. Nó có một cột Select, một cột Bound (Name), và một cột Template (Age)

```
<asp:DataGrid
id="dgAges"
runat="server"
AutoGenerateColumns="False"
OnSelectedIndexChanged=" SelectionChanged">
<Columns>
<asp:ButtonColumn Text="Select" HeaderText="" CommandName="Select" />
<asp:BoundColumn DataField="Name" HeaderText="Name" />
<asp:TemplateColumn HeaderText="Age">
<ItemTemplate>
<%# Container.DataItem("Age") %>&nbsp;yrs. old
</ItemTemplate>
</asp:TemplateColumn>
</Columns>
</asp:DataGrid>
<P>&nbsp;</P>
<P><STRONG><U>Current Selection:</U></STRONG></P>
<P><STRONG>Name:</STRONG>&nbsp;<asp:Label id="lblName"
runat="server"></asp:Label></P>
<P><STRONG>Age:</STRONG>&nbsp;<asp:Label id="lblAge"
runat="server"></asp:Label></P>
```

Điều đầu tiên bạn nghĩ là sử dụng thuộc tính Text của cell để lấy đoạn text trong đó. Nó chỉ làm việc với cột Bound

```
Protected Sub SelectionChanged()
lblName.Text = dgAges.SelectedItem.Cells(1).Text
'Cột Template ..nó sẽ không làm việc
lblAge.Text = dgAges.SelectedItem.Cells(2).Text
```

End Sub

Bởi vì .NET coi nội dung của BoundColumn là dạng text và nội dung của TemplateColumn là một DataBoundLiteralControl. Trong .NET luôn xem nội dung của các cột Template như là một tập hợp các control server. Để set thuộc tính text của lblAge bạn phải dùng thuộc tính Text của DataBoundLiteralControl. Mỗi cell có một tập hợp các Control mà chúng ta có thể tham chiếu tới.

```
Protected Sub SelectionChanged()
```

```
'Bound Column... Đúng
```

```
lblName.Text = dgAges.SelectedItem.Cells(1).Text
```

```
'Template Column... Đúng
```

```
lblAge.Text = CType(dgAges.SelectedItem.Cells(2).Controls(0), DataBoundLiteralControl).Text
```

```
End Sub
```

Đừng thất vọng nếu bạn nghĩ làm sao có thể biết đó là DataBoundLiteralControl. Điều quan trọng là bạn hiểu cách nó làm việc. Bây giờ chúng ta đã biết .NET đưa nội dung của các cột Template vào các tập hợp collection trong mỗi cell.

Lưu ý không phải bao giờ Template column cũng có một DataBoundLiteralControl. Nếu bạn có một control trong temple (TextBox là EditItemTemplate).

Cách làm tốt hơn.

Chúng tôi làm theo một cách hơi khác. Đầu tiên sử dụng một label trong cột Template, vì thế chúng tôi biết cái gì trong một DataBoundLiteralControl:

```
<asp:DataGrid  
id="dgAges"  
runat="server"  
AutoGenerateColumns="False"  
OnSelectedIndexChanged=" SelectionChanged">
```

```
<Columns>
```

```
<asp:ButtonColumn Text="Select" HeaderText="" CommandName="Select" />
```

```
<asp:BoundColumn DataField="Name" HeaderText="Name" />
```

```
<asp:TemplateColumn HeaderText="Age">
```

```
<ItemTemplate>
```

```
<asp:Label RunAt="server" ID="lblThisAge" Text='<%# Container.DataItem("Age") %>'
```

```
/>&nbsp;&nbsp;&nbsp;yrs. old
```

```
</ItemTemplate>
```

```
</asp:TemplateColumn>
```

```
</Columns>
```

```
</asp:DataGrid>
```

```
<P>&nbsp;&nbsp;&nbsp;</P>
```

```
<P><STRONG><U>Current Selection:</U></STRONG></P>
```

```
<P><STRONG>Name:</STRONG>&nbsp;&nbsp;&nbsp;<asp:Label id="lblName"
```

```
runat="server"></asp:Label></P>
```

```
<P><STRONG>Age:</STRONG>&nbsp;&nbsp;&nbsp;<asp:Label id="lblAge"
```

```
runat="server"></asp:Label></P>
```

Xin lưu ý các điểm sau:

Chúng tôi biết loại control trong cột Template bởi vì chúng tôi đặt <asp:Label ... Có nghĩa là chúng tôi không cần đoán loại control có trong Template.

Chúng tôi biết rõ thông tin ID mà chúng tôi muốn nhận: ID="lblThisAge" ...

Chúng tôi có thể dùng phương thức FindControl() để nhận về Label dựa trên ID

...

```

Protected Sub SelectionChanged()
    lblName.Text = dgAges.SelectedItem.Cells(1).Text

    lblAge.Text = CType(dgAges.SelectedItem.FindControl("AgeText"), Label).Text
End Sub
...

```

Xây dựng một trang Master/Detail DataGrid trong ASP.NET (C# và VB.NET)

Bạn làm thế nào để xây dựng một trang Master|Detail ?. Ví dụ bạn có một DataGrid của các khách hàng và mỗi cột trên mỗi dòng có một DataGrid được nhúng vào với các thông tin của người khách hàng đó. Thách thức thật sự ở đây, làm thế nào để bind a DataGrid mà không được tạo cho đến khi run-time. Chúng ta không thể set thuộc tính DataSource của DataGrid hoặc sử dụng phương thức DataBind bởi vì DataGrid không tồn tại ở design-time.

Sau đây là cách chúng tôi thực hiện. Chúng ta sẽ xây dựng một trang Master|Detail sử dụng bảng Customers và Orders của CSDL NorthWind

Đầu tiên chúng ta xây dựng một form Web. Chúng ta sẽ dùng một DataGrid với BoundColumns và TemplateColumns.

Chúng bao gồm 3 cột. Cột đầu tiên là một BoundColumn với thuộc tính Visible được set là False (để ẩn nó đi). Cột này được bound đến trường CustomerID trong data source. Chúng ta sẽ dùng sau này để bind Orders DataGrid. Cột thứ hai là một HyperLinkColumn, còn bound đến CustomerID field và set đường link đến

The second column is a HyperLinkColumn, also bound to the CustomerID field, and set to link to the OrderDetailDataGrid.aspx. Cột thứ 3 và thứ 4 là TemplateColumns để thiết kế layout

Sau đây là Web Form code:

```

<%@ Page language="c#" Inherits="MasterDetail.CustomerOrderDataGrid"
EnableViewState="False" %>
<HTML>
<body style="font: x-small Verdana, Arial, sans-serif;">
<form id="CustomerOrderDataGrid" method="post" runat="server">
<p><a href="/DayOfDotNet/">Parent Directory</a></p>
<asp:DataGrid id="CustomerDataGrid" runat="server"
AutoGenerateColumns="False"
CellPadding="2" CellSpacing="0"
Font-Names="Verdana, Arial, sans-serif"
BorderColor="Black" BorderWidth="1"
GridLines="Horizontal"
OnItemDataBound="CustomerDataGrid_OnItemDataBound">
<HeaderStyle
Font-Bold="True" Font-Size="small"
Font-Name="Arial"
BackColor="Maroon" ForeColor="White" />
<ItemStyle Font-Size="x-small" />
<AlternatingItemStyle BackColor="Tan" />
<Columns>
<asp:BoundColumn
DataField="CustomerID" Visible="False" />
<asp:HyperLinkColumn
DataTextField="CustomerID"
DataNavigateUrlField="CustomerID"
DataNavigateUrlFormatString="OrderDetailDataGrid.aspx?customerid={0}"
HeaderText="ID"
ItemStyle-VerticalAlign="Top" />

```



```

<asp:TemplateColumn ItemStyle-VerticalAlign="Top" HeaderText="Customer">
<ItemTemplate>
<b><%# DataBinder.Eval(Container.DataItem, "CompanyName") %></ b><br>
<%# DataBinder.Eval(Container.DataItem, "Address" ) %><br>
<%# DataBinder.Eval(Container.DataItem, "City" ) %>,
<%# DataBinder.Eval(Container.DataItem, "Region") %>
<%# DataBinder.Eval(Container.DataItem, "PostalCode" ) %><br>
<br>
<%# DataBinder.Eval(Container.DataItem, "ContactName" ) %><br>
<%# DataBinder.Eval(Container.DataItem, "ContactTitle" ) %><br>
<%# DataBinder.Eval(Container.DataItem, "Phone" ) %>
</ItemTemplate>
</asp:TemplateColumn>
<asp:TemplateColumn ItemStyle-VerticalAlign="Top"
HeaderText="Orders">
<%-- Nhung DataGrid ở đây --%>
</asp:TemplateColumn>
</Columns>
</asp:DataGrid>
<!-- End DataGrid -->
</form>
<!-- End Web Form -->
</body>
</HTML>

```

Lets start by retrieving the data for both the customers and orders in the Page_Load() event handler.

Nhận dữ liệu từ customers and orders trong sự kiện Page_Load()

```

using System;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Configuration;

namespace MasterDetail
{
public class CustomerOrderDataGrid : System.Web.UI.Page
{
protected DataGrid CustomerDataGrid;
private DataSet ds = new DataSet();

private void Page_Load(object sender, System.EventArgs e)
{
string sqlStmnt = "SELECT * FROM Customers; SELECT * FROM Orders";
string conString = "server=localhost;database=Northwind;uid=sa;pwd=";

SqlDataAdapter sda = new SqlDataAdapter(sqlStmnt, conString);

sda.Fill(ds);
ds.Tables[0].TableName = "Customers";
ds.Tables[1].TableName = "Orders";

```

```

CustomerDataGrid.DataSource = ds.Tables["Customers"];
CustomerDataGrid.DataBind();
}
}
}

```

Trong câu SQL chúng ta chọn 2 result sets và sử dụng phương thức Fill() để tạo 2 DataTables, chúng tôi set thuộc tính TableName cho mỗi DataTables và bind CustomerDataGrid.

Lưu ý: Chúng ta khai báo DataSet (ds) ở mức lớp. Việc này sẽ cho phép chúng ta có thể kết nối đến DataSet từ sự kiện OnItemDataBound. Trong sự kiện OnItemDataBound chúng ta có thể construct động một DataGrid, và bind nó chỉ đến các record trong Orders DataTable có cùng giá trị CustomerID như CustomerID của dòng hiện thời.

Bạn hãy xem sự kiện OnItemDataBound()

```

protected void CustomerDataGrid_OnItemDataBound(object sender, DataGridItemEventArgs e)
{
    if(e.Item.ItemType == ListItemType.Item ||
        e.Item.ItemType == ListItemType.AlternatingItem)
    {
        DataGrid OrdersDataGrid = new DataGrid();

        OrdersDataGrid.BorderWidth = (Unit)1;
        OrdersDataGrid.CellPadding = 4;
        OrdersDataGrid.CellSpacing = 0;
        OrdersDataGrid.GridLines = GridLines.Horizontal;
        OrdersDataGrid.BorderColor = Color.FromName("Black");

        OrdersDataGrid.ItemStyle.Font.Name = "Verdana";
        OrdersDataGrid.ItemStyle.Font.Size = FontUnit.XSmall;

        OrdersDataGrid.AlternatingItemStyle.BackColor = Color.FromName("LightGray");

        OrdersDataGrid.ShowHeader = true;
        OrdersDataGrid.HeaderStyle.BackColor = Color.FromName("Black");
        OrdersDataGrid.HeaderStyle.ForeColor = Color.FromName("White");
        OrdersDataGrid.HeaderStyle.Font.Bold = true;
        OrdersDataGrid.HeaderStyle.Font.Size = FontUnit.XSmall;

        OrdersDataGrid.AutoGenerateColumns = false;

        BoundColumn bc = new BoundColumn();
        bc.HeaderText = "Order ID";
        bc.DataField = "OrderID";
        bc.ItemStyle.Wrap = false;
        OrdersDataGrid.Columns.Add(bc);

        bc = new BoundColumn();
        bc.HeaderText = "Order Date";
        bc.DataField = "OrderDate";
        bc.DataFormatString="{0:d}";
        bc.ItemStyle.Wrap = false;
        OrdersDataGrid.Columns.Add(bc);

        bc = new BoundColumn();
        bc.HeaderText = "Required Date";
        bc.DataField = "RequiredDate";
    }
}

```

```

bc.DataFormatString="{0:d}";
bc.ItemStyle.Wrap = false;
OrdersDataGrid.Columns.Add(bc);

bc = new BoundColumn();
bc.HeaderText = "Shipped Date";
bc.DataField = "ShippedDate";
bc.DataFormatString="{0:d}";
bc.ItemStyle.Wrap = false;
OrdersDataGrid.Columns.Add(bc);

DataView _orders = ds.Tables["Orders"].DefaultView;
_orders.RowFilter = "CustomerID=" + e.Item.Cells[0].Text + """;

OrdersDataGrid.DataSource = _orders;
OrdersDataGrid.DataBind();

e.Item.Cells[3].Controls.Add(OrdersDataGrid);
}
}

```

Tạo một VB Component để lấy thông tin Connection đến CSDL của bạn

Đầu tiên chúng ta tạo các thông số sau trong tập tin config.web

```

<appsettings>
<add key="gConn" value="server=local;uid=sa;pwd=secret;database=pubs" />
</appsettings>

```

Bây giờ chúng ta tạo tập tin dbConn.vb

```

Imports System
Imports System.Web
Imports System.Collections

Namespace WebDB
Public Class WebDBconn
Shared m_ConnectionString As String
Shared ReadOnly Property ConnectionString As String

Get
If m_ConnectionString = "" Then

Dim appsetting As Hashtable = CType(HttpContext.Current.GetConfig("appsettings"), Hashtable)
m_ConnectionString = CStr(appsetting("DBConnString"))

If m_ConnectionString = "" Then
throw new Exception("Database Connection Value not set in Config.web")
End if

End If
' Trả về giá trị kết nối
return m_connectionString

End Get
End Property

```

```
End Class
End Namespace
```

Bây giờ chúng ta tạo tập tin .dll. Tạo một tập tin batch, tên là MakeDll.bat và đặt cùng một thư mục với .dll

```
set odir=c:\temp\dbConn.dll
set assemblies=c:\winnt\complus\v2000.14.1812\System.Web.dll
vbc /t:library /out:%odir% /r:%assemblies% dbConn.vb
```

Chạy tập tin batch, sao chép dbconn.dll đến thư mục bin của web của bạn và tạo tập tin .aspx sau:

```
<%@ Page Description="ASP+ document" EnableSessionState="false"
MaintainState="false" %>
<%@ Import Namespace="WebDB" %>
<script language="VB" runat="server">
Sub Page_Load(sender As Object, e As EventArgs)
response.write(WebDBconn.ConnectionString)
End Sub
</script>
<html>
<head>
<title></title>
</head>
<body>
</body>
</html>
```

Những mẹo cần biết khi lập trình .NET

Chúng tôi xin đưa ra các phương pháp giải quyết các vấn đề mà các nhà phát triển .NET thường gặp. Hy vọng chúng sẽ giúp ích cho các bạn.

1. Làm thế nào giới hạn một chương trình chỉ chạy một lần
Trong form chính đổi thành như sau:

```
static void Main()
{
Process ThisProcess = Process.GetCurrentProcess();
Process [] AllProcesses = Process.GetProcessesByName(ThisProcess.ProcessName);
if (AllProcesses.Length > 1)
{
MessageBox.Show(ThisProcess.ProcessName + " is already running",
ThisProcess.ProcessName,
MessageBoxButtons.OK, MessageBoxIcon.Error);
}
else
{
Application.Run(new MainForm());
}
}
```

2. Di chuyển con trỏ đến dòng và cột xác định (RichTextBox)
Dùng phương thức GoToLineAndColumn

```
public void GoToLineAndColumn(int Line, int Column)
```

```

{
Cursor.Current = Cursors.WaitCursor;
int Offset = 0;
int i = 0;
foreach (String L in Lines)
{
if (i < Line - 1)
{
Offset += L.Length + 1;
}
else
{
break;
}

i++;
}
Select(Offset + Column - 1, 0);
Cursor.Current = Cursors.Arrow;
}

```

3.Xác định cột hiện thời. (RichTextBox)

```

public int GetColumn()
{
int LineNumber = GetLineFromCharIndex(SelectionStart);

int LineOffset = 0;
int i = 0;

foreach (String Line in Lines)
{
if (i < LineNumber)
{
LineOffset += Line.Length + 1;
}
else
{
break;
}

i++;
}
return SelectionStart - LineOffset + 1;
}

```

3. Chạy JScript.NET trong ứng dụng C#

Tạo một JScript.NET "package" bao gồm một phương thức toàn cục (public)

```

package JScript
{
class Eval
{
public function DoEval(expr : String) : String

```

```

{
return eval(expr);
}
}
}

```

```

try { Result = (int) Application.UserAppDataRegistry.GetValue("Resolution"); } catch(Exception)
{}

```

Và thêm một reference đến chương trình C# của bạn và sử dụng

```

JScript.Eval E = new JScript.Eval();
String Expression = ExpressionTextBox.Text;
try
{
ResultTextBox.Text = E.DoEval(Expression);
}
catch(Microsoft.JScript.JScriptException jse)

```

4. Lưu thông số cấu hình vào Registry

Đầu tiên vào AssemblyInfo.cs và bỏ tất cả các thông số từ AssemblyVersion:

```
[assembly: AssemblyVersion("1.0.0.0")]
```

Mặc dù mỗi lần bạn build ứng dụng khoá register sẽ thay đổi. Lưu giá trị bằng cách sau

```
Application.UserAppDataRegistry.SetValue("Value", Value);
```

Nạp lại các thông số :

```

try
{
Value = (int) Application.UserAppDataRegistry.GetValue("Value");
}
catch(Exception)
{
}
}

```

SQL Server: UDF IsValidNumber

Hàm trong SQL Server rất hữu dụng cho các bạn. Hàm kiểm tra một chuỗi có phải là một số không. Hàm này chấp nhận một chuỗi và kiểm tra nếu chuỗi có bao gồm các ký tự không phải 0-9 hoặc dấu thập phân (decimal). Hàm trả về 0 nếu đúng là số; 1 nếu không phải dạng số.

```
CREATE FUNCTION udfIsValidNumber
```

```

(
@thestring varchar(50),
@numdecimals int = 0
)

```

```
RETURNS int
```

```
AS
```

```
BEGIN
```

```
DECLARE @not int,
```

```
@ascii int,
```

```
@pos int,
```

```
@dec int
```

```
SET @pos = 1
```

```
SET @not = 0
```

```
SET @dec = 0
```

```

--first check to see if it is a valid number
IF @thestring IS NULL
SET @not = 1
IF len(@thestring) = 0
SET @not = 1
WHILE @pos <= len(@thestring)
BEGIN
SELECT @ascii = ascii(substring(@thestring, @pos, 1))
IF (@ascii > 57) SET @not = 1
IF (@ascii < 46) SET @not = 1
IF (@ascii = 47) SET @not = 1
IF (@ascii = 46) SET @dec = @dec + 1
SET @pos = @pos + 1
END
IF @dec > 1 SET @not = 1
IF @not > 0 RETURN @not -- invalid number
--valid number now check number of decimals

SELECT @dec = charindex('.',@thestring)
SET @pos = len(@thestring) - @dec -- find the number of characters right of decimal
IF @pos > @numdecimals SET @not = 1

RETURN @not
END

```

ADO/SQL Server nText inserts/updates

Rất nhiều lập trình viên hỏi làm thế nào để thêm (insert) dữ liệu vào trong một trường nText vào SQL Server với ADO. Phần lớn các câu SQL thường dùng string chuẩn và nó sẽ gặp vấn đề khi cập nhật các ký tự đặc biệt. Sau đây chúng tôi sẽ giúp các bạn tránh được các lỗi thường gặp đó.

```

Dim IRecs
Dim moADOCon
Dim moADCOM

```

```

Set moADOCon = Server.CreateObject("ADODB.Connection")
Set moADCOM = Server.CreateObject("ADODB.Command")

```

```

moADOCon.Open "your connection string"

```

```

With moADCOM

```

```

    .ActiveConnection = moADOCon
    .CommandText = "spPost"
    .CommandType = adCmdStoredProc
    .Parameters.Append .CreateParameter("@RETURN_VALUE",
adInteger, adParamReturnValue, 0)
    .Parameters.Append .CreateParameter("@ReplyToID", adInteger,
adParamInput, , msPostID)
    .Parameters.Append .CreateParameter("@fk_author_id",
adInteger, adParamInput, , cLng(Session("intMemberID")))
    .Parameters.Append .CreateParameter("@fk_interest_id",

```

```
adInteger, adParamInput, , msInterestID)
    .Parameters.Append .CreateParameter("@subject", adVarChar,
adParamInput, 50, msSubject)
    .Parameters.Append .CreateParameter("@bodytext",
adVarChar, adParamInput, 1073741823, msBodyText)
    .Execute IRecs, , adExecuteNoRecords
End With
```

```
moADODCon.Close
Set moADODCom = nothing
Set moADODCon = nothing
```