



Sử dụng ADO.Net

Biên tập bởi:

Khoa CNTT ĐHSP KT Hưng Yên

Sử dụng ADO.Net

Biên tập bởi:

Khoa CNTT ĐHSP KT Hưng Yên

Các tác giả:

Khoa CNTT ĐHSP KT Hưng Yên

Phiên bản trực tuyến:

<http://voer.edu.vn/c/6113abf3>

MỤC LỤC

1. Sử dụng ADO.Net
Tham gia đóng góp

Sử dụng ADO.Net

Mục tiêu của bài:

- Sử dụng được các lớp trong thư viện ADO.NET để thao tác CSDL
- Mô tả được chức năng và cách thức sử dụng các lớp DataSet, DataTable, DataRow để đọc dữ liệu.
- Đọc dữ liệu với DataReader và hiển thị trên các lớp kế thừa từ Listbox, DataGridView.
- Cập nhật dữ liệu với DataSet và DataAdapter

Nội dung

Thao tác CSDL với DataConnection, DataCommand, DataReader

Đề thao tác trực tiếp (Gồm Select , Insert, Delete và Update) với bảng CSDL, có thể dùng một trong 3 phương thức của đối tượng DataCommand.

- cmd.ExecuteReader() : Thực hiện câu lệnh Select và trả về một DataReader
- cmd.ExecuteScalar() : Thực hiện câu lệnh mà kết quả trả về chỉ có 1 ô (Ví dụ câu lệnh Select Count(*)...;).
- cmd.ExecuteNonQuery() : Thực hiện câu lệnh OLEDB nhưng không trả về kết quả (Delete, Update, Insert, Exec Sp_, ...)

Thêm bản ghi vào bảng CSDL

Ví dụ: Tạo form để thêm bản ghi vào trong bảng Customers.

frmInsertData

	CustomerID	CompanyName	ContactName	ContactTitle	Address
▶	ALFKI	Alfreds Futterkiste	Nguyễn Minh Quý	Sales Represent...	Obere Str. 5
	ANATR	Ana Trujillo Empa...	Ana Trujillo	Owner	Avda. de la
	ANTON	Antonio Moreno ...	Antonio Moreno	Owner	Mataderos
	AROUT	Around the Horn	Thomas Hardy	Sales Represent...	120 Hanove
	BERGS	Berglunds snabb...	Christina Berglund	Order Administrator	Berguvsväg
	BLAUS	Blauer See Delik...	Hanna Moos	Sales Represent...	Forsterstr. 5
	BLONP	Blondel père et fils	Frédérique Citeaux	Marketing Manager	24, place Kl
	BOLID	Bólido Comidas p...	Martín Sommer	Owner	C/ Araquil, 6
	BONAP	Bon app'	Laurence Lebihan	Owner	12, rue des
	BOTTI	Bottini	Elizabeth Lincoln	Accounting Man...	22, rue de

CustomerID ContactName Company

Thêm bản ghi vào CSDL, sử dụng câu lệnh **Insert**

```
Private Sub cmdAdd_Click(...) Handles cmdAdd.Click
    Dim cmd As New OleDb.OleDbCommand
    Dim strOLEDB As String

    '/// Chèn bản ghi
    strOLEDB = "Insert into Customers(CustomerID,
    ContactName, CompanyName)"
    strOLEDB &= "values ('" & txtCustomerID.Text
    &"', '" & txtContactName.Text &"', '" &
    txtCompany.Text &"')"

```

*** Lưu ý khi thêm bản ghi:

- Nếu trường nào thuộc kiểu Text (hoặc Char) thì trước khi thêm cần đặt vào trong cặp dấu nháy đơn ' '.

- Trong trường hợp bản thân dữ liệu cần thêm chứa dấu nháy đơn thì cần thay nó bằng 2 dấu nháy đơn. (Sử dụng hàm Replace để thay).

Đọc dữ liệu với DataReader

- DataReader có thể đọc dữ liệu từ CSDL để sau đó ta có thể nạp vào các điều khiển khác.

- Ví dụ 1

- Nạp dữ liệu thuộc một trường của bảng vào Listbox thông qua DataReader
- `clsMyListBox` là lớp được thừa kế từ lớp Listbox và bổ sung thêm phương thức **LoadDataField**.

```
Imports System.Data.OleDb

Public Class clsMyListBox
    Inherits ListBox

    Private Cn As New OleDbConnection
    Private Cmd As New OleDbCommand
    Private dat As OleDbDataReader

    '//Hàm kết nối đến csdl. Input là tên CSDL cần kết nối.
    '//Hàm trả về true nếu thành công. Trái lại là false
    Private Function ConnectToDB(ByVal DBName As String) As Boolean
        If Cn.State = ConnectionState.Open Then Return True

        Cn.ConnectionString =
"Provider=Microsoft.Jet.OLEDB.4.0; Data source = " & DBName

        Cn.Open()
        return Cn.State = 1
    End Function

    '/// Hàm nạp tất cả các bản ghi vào Listbox.
    '/// Trường nạp vào có tên là fieldName
    '/// Lấy từ bảng có tên là InTable.
    Public Sub LoadDataField(ByVal fieldName As String,
        ByVal InTable As String, ByVal FromDatabaseName As String)

        If Me.ConnectToDB(FromDatabaseName) = False Then Return
```

```

Try
    Cmd.Connection = Cn
    Cmd.CommandText = InTable
    Cmd.CommandType = CommandType.TableDirect
    dat = Cmd.ExecuteReader()
    Me.Items.Clear()

    Do While dat.Read = True
        Me.Items.Add(dat.Item(FieldName))
    Loop

    Catch ex As Exception
        MsgBox("Có lỗi : " & ex.Message)

    Finally
        dat.Close()
        Cmd.Dispose()
    End Try
End Try

End Sub

'/// Định nghĩa là phương thức Dispose (Giải phóng)
Protected Overrides Sub Dispose(ByVal disposing As Boolean)
    MyBase.Dispose(disposing)
    If Cn.State = ConnectionState.Open Then Cn.Close()
    Cmd.Dispose()
End Sub
End Class

```

*** Tương tự cho Combobox

- Ví dụ 2: Xây dựng lớp clsDataGridView, thừa kế từ lớp DataGridView nhưng lớp mới có thêm phương thức :

Public Sub NapDuLieu(Tên_Bảng As String, TênTậpCSDL As String)

Đọc dữ liệu vào DataGrid

```
Imports System.Data.OleDb

Public Class clsDataGridView
    Inherits DataGridView

    Private datAdapter As OleDbDataAdapter
    Private datSet As New DataSet

    '//Hàm nạp dữ liệu trong bảng <TenBang> vào DataGrid
    '//Nếu thành công hàm sẽ trả về true, trái lại trả false
    Public Function NapBang(ByVal TenBang As String, ByVal
TenTepCSDL As String) As Boolean
        Try
            Dim strConn As String =
                "Provider=Microsoft.jet.oledb.4.0;" & _
                "Data source=" & TenTepCSDL

            datAdapter = New OleDbDataAdapter _
                ("Select * from " & TenBang, strConn)

            datSet.Clear()
            datAdapter.Fill(datSet, TenBang)

            '// Nạp dữ liệu ở trong DataSet vào clsDataGrid
            Me.DataSource = datSet
            Me.DataMember = TenBang

        Catch ex As Exception
            MsgBox("Có lỗi : " & ex.Message)
            Return False
        Finally
            datAdapter.Dispose()
            datSet.Dispose()
        End Try

        Return True
    End Function
End Class
```

Sửa đổi bản ghi trong bảng CSDL

Xóa bản ghi trong bảng CSDL

Thao tác dữ liệu theo mô hình phi kết nối

Lớp DataSet

- Dataset là một lớp **dùng để thao tác với dữ liệu theo mô hình dữ liệu phi kết nối**. Nó được coi như là một kho chứa các bảng (table). Người dùng có thể thay đổi dữ liệu trong các bảng này và khi thực sự muốn cập nhật vào Cơ sở dữ liệu thì DataSet sẽ thực hiện cập nhật thông qua lớp DataAdapter.

- Các bảng trong DataSet có thể do DataAdapter Fill vào hoặc cũng có thể là các bảng được tạo thành từ lớp DataTable.

- Các bảng này được quản lý bởi tập hợp Tables của lớp DataSet.

- Ví dụ 1 : Thêm một bảng vào Dataset và đặt tên cho bảng đó là KhachHang

```
Dim da As OleDbDataAdapter
```

```
Dim ds As New DataSet
```

```
da = New OleDbDataAdapter("Select * From...", "Provider...")
```

```
da.Fill(ds, "KhachHang")
```

- Ví dụ 2: Thêm một bảng (đối tượng DataTable) vào DataSet.

```

'/// Thêm một bảng (bảng này có 2 trường là MaKH và TenKH)
'/// Vào dataset. Bảng này do ta tự tạo thông qua DataTable
Public Sub DataSetFromTable()
    Dim ds As New DataSet
    Dim dt As New DataTable("KhachHang")
    dt.Columns.Add("MaKH", Type.GetType("System.String"))
    dt.Columns.Add("TenKH", Type.GetType("System.String"))

    ds.Tables.Add(dt)
End Sub

```

? Sau khi đã thêm vào DataSet theo một trong 2 cách ở trên thì sau đó ta có thể lấy các bảng này ra và hiển thị trên các DataGridView như vẫn làm trước đây :

```

dgrKhachHang.DataSource = ds
dgrKhachHang.DataMember = "KhachHang"

```

Lớp DataTable

- Dùng để lưu một bảng dữ liệu, tương đương với một bảng nằm trong DataSet. Mỗi bảng này có thể chứa nhiều dòng (DataRow), mỗi dòng bao gồm một hoặc nhiều cột (DataColumn).
- DataTable thường được dùng để thao tác cùng với DataSet trong việc đọc cũng như cập nhật vào CSDL.
- Có thể tạo một DataTable như sau:

Dim dt As New DataTable("Tên_Bảng")

- Sau khi tạo xong, cần tiếp tục chỉ ra là bảng này có mấy cột (trường) và kiểu của mỗi trường như sau:

dt.Columns.Add("MaKH" , Type.GetType("System.String"))

dt.Columns.Add("TenKH" , Type.GetType("System.String"))

- DataTable này cũng có thể được thêm vào một DataSet có sẵn, ví dụ thêm vào ds:

ds.Tables.Add (dt)

- Một đối tượng DataTable cũng có thể dùng để tham chiếu đến một Table nằm trong dataSet, ví dụ : cho dataTable tên là dt Tham chiếu đến một bảng của dataset ds như sau:

Dim dt As DataTable

Dt = ds.Tables("KhachHang")

??? Tại sao không khai báo là Dim dt As New DataTable !

- Thêm các hàng (bản ghi) vào DataTable:

Có thể thêm vào bảng bằng cách thêm lần lượt các DataRow thông qua tập hợp Rows của đối tượng DataTable, ví dụ:

```
'''// Tạo một bảng (DataTable) có 2 trường,  
'''// sau đó thêm vào 2 hàng trống.  
Public Sub TaoDataTable()  
    Dim dt As New DataTable("KhachHang")  
  
    dt.Columns.Add("MaKH", Type.GetType("System.String"))  
    dt.Columns.Add("TenKH", Type.GetType("System.String"))  
  
    dt.Rows.Add()      '''// thêm một hàng (bản ghi)  
    dt.Rows.Add()      '''// thêm hàng nữa  
End Sub
```

Kết quả ta được một "bảng" :

Muốn truy xuất đến từng hàng (bản ghi) của DataTable, có thể thông qua tập hợp Rows của DataTable. Ví dụ: dt.Rows(0) để truy xuất đến hàng đầu tiên. Còn muốn truy xuất đến một ô (có hàng i, cột j) thì viết : dt.Rows(i)(j), hoặc dt.Rows(i)("Tên_Cột").

****Lưu ý:** Có thể đặt khóa chính cho DataTable thông qua thuộc tính PrimaryKey.

Lớp DataRow (Dòng dữ liệu)

- Mỗi đối tượng thuộc lớp DataRow sẽ ứng (tương đương) với một hàng trong DataTable. DataRow thường được dùng để thêm vào một bản ghi (hàng) trong DataTable hoặc để tham chiếu và thao tác với một hàng trong DataTable.

- Khai báo biến DataRow

`Dim Dr as DataRow`

`Dr = dt.Rows.Add()` *‘/// Thêm một hàng và cho Dr trở tới.*

*** Lưu ý : Không thể khai báo :

`Dim Dr as New Datarow`

‘/// Tại sao ???

→ Muốn tạo một hàng mới, có thể viết:

`Dim Dr as DataRow`

`Dr = ds.Tables("Tên_Table").NewRow()`

- Tham chiếu đến một hàng trong DataTable: `Dr = dt.Rows(Chỉ số của hàng)`

- Ví dụ: Tham chiếu đến hàng thứ nhất trong DataTable có tên là `dt`

`Dr = dt.Rows(0)`

- Đọc giá trị trong các cột (trường) thuộc hàng vừa được trỏ:

`Dr(i)` → đọc giá trị của trường thứ `i` của hàng `Dr`

Hoặc `Dr("Tên_Cột")`

- Ví dụ: Đọc trường “MaKH” của hàng mà `Dr` vừa tham chiếu

`Dr("MaKH")`

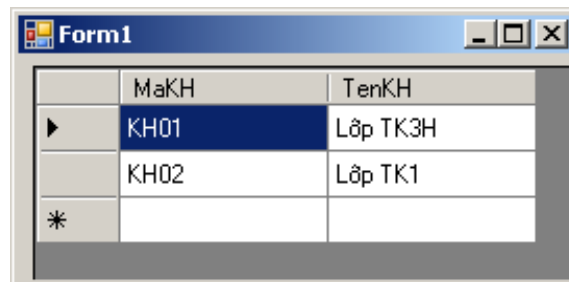
- Ghi giá trị vào các cột vào hàng vừa được `Dr` trỏ tới:

`Dr(i) = <New_Value>` → Ghi giá trị `<New_Value>` vào cột có chỉ số là `i`

`Dr("Tên_Cột") = <New_Value>`

Ví dụ tổng hợp:

Tạo một chương trình để hiển thị trên màn hình một bảng có dạng như sau:



	MaKH	TenKH
▶	KH01	Lớp TK3H
	KH02	Lớp TK1
*		

Bước 1: Kéo một DataGridView vào form

Bước 2: Tạo một DataTable và điền 2 bản ghi vào trong đó

+ Tạo DataTable có 2 trường là MaKH và TenKH cùng có kiểu String

+ Thêm vào 2 bản ghi (KH01, Lớp TK3H) và (KH02, TK1) bằng phương thức Add của tập hợp Rows của đối tượng DataTable, kết hợp với DataRow.

Bước 3: Tạo một DataSet và thêm DataTable ở trên vào.

Bước 4: Gắn kết DataSet với DataGridView để hiển thị.

Listing chương trình:

```

Imports System.Data.OleDb
Imports System.Data

Public Class Form1

    Private Ds As DataSet
    Dim Dt As DataTable
    Dim dgrKhachHang As DataGridView

    '/// Tạo một DataGridView và Add vào form
    Private Sub Buoc1()
        dgrKhachHang = New DataGridView
        dgrKhachHang.Top = 0
        dgrKhachHang.Left = 0
        dgrKhachHang.Width = Me.Width
        dgrKhachHang.Height = Me.Height
        Me.Controls.Add(dgrKhachHang)
    End Sub

    '/// Tạo một dataTable và insert 2 bản ghi
    Private Sub Buoc2()
        Dt = New DataTable("KhachHang")

        '/// Tạo 2 cột (trường) của bảng
        Dt.Columns.Add("MaKH", Type.GetType("System.String"))
        Dt.Columns.Add("TenKH", Type.GetType("System.String"))

        '/// Thêm vào 2 bản ghi (2 hàng) trắng
        Dt.Rows.Add()
        Dt.Rows.Add()

        Dim dr As DataRow
        '/// Điền dữ liệu vào bản ghi trắng thứ nhất
        dr = Dt.Rows(0)
        dr("MaKH") = "KH01"
        dr("TenKH") = "Lớp TK3H"

        '/// Điền dữ liệu vào bản ghi trắng thứ hai
        dr = Dt.Rows(1)
        dr("MaKH") = "KH02"
        dr("TenKH") = "Lớp TK1"
    End Sub

    '/// Thêm bảng vừa tạo vào DataSet
    Private Sub Buoc3()

```



```

        Ds = New DataSet
        Ds.Tables.Add(Dt)
    End Sub

    '/// Gắn bảng vừa thêm vào DataSet
    '///với DataGridView để hiển thị
    Private Sub Buoc4()
        dgrKhachHang.DataSource = Ds
        dgrKhachHang.DataMember = "KhachHang"
    End Sub

    '/// Demo
    Private Sub Form1_Load(...) Handles Form1.Load
        Call Buoc1()
        Call Buoc2()
        Call Buoc3()
        Call Buoc4()
    End Sub
End Class

```

*** Lưu ý:

- Có thể thao tác với các hàng của DataGridView thông qua tập hợp DataGridView.Rows.
- Thao tác với các ô (Cell) trong mỗi hàng đó thông qua tập hợp Cells, ví dụ: DataGridView.Rows(0).Cells(0) để tham chiếu đến ô ở hàng đầu tiên và cột đầu tiên.

Cập nhật dữ liệu thực sự thông qua đối tượng DataSet

Khi thao tác với DataSet (ví dụ sửa đổi) thì đó chỉ là tạm thời, không được cập nhật ngay vào Cơ sở dữ liệu thực sự. Trong trường hợp ta muốn cập nhật trở lại CSDL thì gọi qua các phương thức Update của đối tượng DataAdapter.

Ví dụ

Cập nhật nội dung bảng dữ liệu dgr vào CSD

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    CmdUpdate.Text = "&Cập nhật"
    CmdUpdate.Dock = DockStyle.Top
    Dgr.Dock = DockStyle.Fill

    Me.Controls.Add(CmdUpdate)
    Me.Controls.Add(Dgr)

    '///Kết nối
    Cn = New OleDbConnection("Provider=Microsoft.jet.oledb.4.0; Data
source=c:\nwind.mdb")
    Cn.Open()

    Cmd.Connection = Cn
    Cmd.CommandText = "Select * from suppliers"

    Da.SelectCommand = Cmd
    Da.Fill(Ds, "NCC")

    Dgr.DataSource = Ds
    Dgr.DataMember = "NCC"

End Sub

Sub UpdateDB(ByVal sender As Object, ByVal e As EventArgs) Handles
CmdUpdate.Click
    Dim cmdbuilder As New OleDbCommandBuilder(Da)

    Da.Update(Ds, "NCC")
End Sub
```



```

' Assumes connection is a valid OleDbConnection.
Dim adapter As OleDbDataAdapter = New OleDbDataAdapter(
    "SELECT CategoryID, CategoryName FROM Categories", connection)

adapter.UpdateCommand = New OleDbCommand(
    "UPDATE Categories SET CategoryName = @CategoryName " & _
    "WHERE CategoryID = @CategoryID", connection)

adapter.UpdateCommand.Parameters.Add(
    "@CategoryName", OleDbDbType.NVarChar, 15, "CategoryName")

Dim parameter As OleDbParameter =
adapter.UpdateCommand.Parameters.Add(
    "@CategoryID", OleDbDbType.Int)
parameter.SourceColumn = "CategoryID"
parameter.SourceVersion = DataRowVersion.Original

Dim dataSet As DataSet = New DataSet
adapter.Fill(dataSet, "Categories")

Dim row As DataRow = dataSet.Tables("Categories").Rows(0)
row("CategoryName") = "New Category"

adapter.Update(dataSet, "Categories")

```

Tham gia đóng góp

Tài liệu: Sử dụng ADO.Net

Biên tập bởi: Khoa CNTT ĐHSP KT Hưng Yên

URL: <http://voer.edu.vn/c/6113abf3>

Giấy phép: <http://creativecommons.org/licenses/by/3.0/>

Module: Sử dụng ADO.Net

Các tác giả: Khoa CNTT ĐHSP KT Hưng Yên

URL: <http://www.voer.edu.vn/m/cee6878b>

Giấy phép: <http://creativecommons.org/licenses/by/3.0/>

Chương trình Thư viện Học liệu Mở Việt Nam

Chương trình Thư viện Học liệu Mở Việt Nam (Vietnam Open Educational Resources – VOER) được hỗ trợ bởi Quỹ Việt Nam. Mục tiêu của chương trình là xây dựng kho Tài nguyên giáo dục Mở miễn phí của người Việt và cho người Việt, có nội dung phong phú. Các nội dung đều tuân thủ Giấy phép Creative Commons Attribution (CC-by) 4.0 do đó các nội dung đều có thể được sử dụng, tái sử dụng và truy nhập miễn phí trước hết trong môi trường giảng dạy, học tập và nghiên cứu sau đó cho toàn xã hội.

Với sự hỗ trợ của Quỹ Việt Nam, Thư viện Học liệu Mở Việt Nam (VOER) đã trở thành một cổng thông tin chính cho các sinh viên và giảng viên trong và ngoài Việt Nam. Mỗi ngày có hàng chục nghìn lượt truy cập VOER (www.voer.edu.vn) để nghiên cứu, học tập và tải tài liệu giảng dạy về. Với hàng chục nghìn module kiến thức từ hàng nghìn tác giả khác nhau đóng góp, Thư Viện Học liệu Mở Việt Nam là một kho tàng tài liệu khổng lồ, nội dung phong phú phục vụ cho tất cả các nhu cầu học tập, nghiên cứu của độc giả.

Nguồn tài liệu mở phong phú có trên VOER có được là do sự chia sẻ tự nguyện của các tác giả trong và ngoài nước. Quá trình chia sẻ tài liệu trên VOER trở lên dễ dàng như đếm 1, 2, 3 nhờ vào sức mạnh của nền tảng Hanoi Spring.

Hanoi Spring là một nền tảng công nghệ tiên tiến được thiết kế cho phép công chúng dễ dàng chia sẻ tài liệu giảng dạy, học tập cũng như chủ động phát triển chương trình giảng dạy dựa trên khái niệm về học liệu mở (OCW) và tài nguyên giáo dục mở (OER). Khái niệm chia sẻ tri thức có tính cách mạng đã được khởi xướng và phát triển tiên phong bởi Đại học MIT và Đại học Rice Hoa Kỳ trong vòng một thập kỷ qua. Kể từ đó, phong trào Tài nguyên Giáo dục Mở đã phát triển nhanh chóng, được UNESCO hỗ trợ và được chấp nhận như một chương trình chính thức ở nhiều nước trên thế giới.