



# Truy xuất dữ liệu sử dụng công nghệ ADO

Bởi:

Khoa CNTT ĐHSP KT Hưng Yên

## Truy xuất dữ liệu sử dụng công nghệ ADO

### Truy xuất dữ liệu sử dụng điều khiển ADO Data Control

Tạo một ứng dụng dùng điều khiển ADO Data rất đơn giản. Ta chỉ quan tâm đến việc duyệt cơ sở dữ liệu, không cần quan tâm đến lập trình. Thông qua hai thuộc tính ConnectionString và RecordSource của điều khiển Data, sau đó ràng buộc điều khiển với giao diện người sử dụng. Các bước thực hiện:

1. Bắt đầu một đề án mới của Visual Basic
2. Vào Menu Project Components, lập một tham chiếu đến “Microsoft ADO Data Control 6.0 (OLEDB)”
3. Nhấn OK, điều khiển ADO Data xuất hiện trên thanh công cụ của Visual Basic. Nhấp Double chuột lên điều khiển để tạo một instance của điều khiển trên biểu mẫu.
4. Di chuyển điều khiển đến vị trí thích hợp trên biểu mẫu.
5. Nhấn nút phải chuột lên điều khiển, Menu bật ra chọn ADODB Properties.
6. Hộp thoại Data Link Properties xuất hiện. Chọn nút tùy chọn “User Connection String” rồi nhấn Build
7. Hộp thoại Data Link Properties xuất hiện. Đây là hộp thoại dùng để kết nối với cơ sở dữ liệu, sử dụng các bước kết nối để kết nối với cơ sở dữ liệu.
8. Trong hộp thoại Property pages của điều khiển ADO Data, thông báo cho điều khiển bảng nào sẽ được dùng, chọn tab RecordSource, rồi chọn adCmdTable từ hộp kết hợp commandType
9. Chọn hộp kết hợp Table or Stored Procedure Name. hộp kết hợp hiển thị danh sách tất cả các bảng trong cơ sở dữ liệu. Chọn bảng cần kết nối rồi nhấn Ok.

Kết nối đến cơ sở dữ liệu xảy ra khi ứng dụng thi hành. Tuy nhiên, kết nối không có thông báo, bởi vì không có cách nào để hiển thị dữ liệu. Để hiển thị dữ liệu trả về từ một điều khiển dữ liệu ta phải tạo các kết nối ràng buộc với điều khiển dữ liệu. Theo các bước sau:

## Truy xuất dữ liệu sử dụng công nghệ ADO

1. Tạo các điều khiển
2. Quy định thuộc tính DataSource của các điều khiển là tên của điều khiển ADO Data
3. Chọn điều khiển, quy định thuộc tính DataField của nó là một trường trong cơ sở dữ liệu.

Người sử dụng có thể duyệt qua các mẫu tin bằng các nút của điều khiển.



*Sử dụng điều khiển ADO Data để cập nhật mẫu tin*

Khi người sử dụng sửa đổi mẫu tin hiển thị bởi điều khiển, mẫu tin đó tự động được cập nhật ngay sau khi người sử dụng di chuyển sang mẫu tin khác. hoặc bằng chương trình ta có thể thao tác với đối tượng Recordset.

Ngoài khả năng cho phép duyệt qua RecordSet, điều khiển ADO Data cho phép thi hành các hoạt động trên dữ liệu. Nhiều hoạt động không đòi hỏi lập trình. Ta có thể dùng chương trình với điều khiển dữ liệu để duyệt qua từng mẫu tin, xoá mẫu tin, và tạo mẫu tin mới.

Phần lớn chương trình phải viết khi làm việc với cơ sở dữ liệu đều tập trung trên đối tượng Recordset. Một đối tượng Recordset được sẵn sàng khi ta quy định thuộc tính ConnectionString và RecordSource của điều khiển. Để truy cập một thuộc tính hay phương thức của đối tượng Recordset của điều khiển dữ liệu trong chương trình, ta tham chiếu đến điều khiển dữ liệu, rồi tham chiếu đến đối tượng Recordset, rồi tham chiếu đến đối tượng hay phương thức của đối tượng Recordset.

VD: để chuyển đến mẫu tin cuối cùng của Recordset chứa trong điều khiển dữ liệu dataADO, ta dùng đoạn chương trình sau:

```
dataADO.Recordset.MoveLast
```

## Truy xuất dữ liệu sử dụng công nghệ ADO

Dữ liệu đối tượng ADO (ActiveX Data Object) là công nghệ truy cập CSDL hướng đối tượng tương tự như DAO và RDO.

ADO hiện nay được Microsoft xem là kỹ thuật để truy cập CSDL từ Web Server, sử dụng ADO để làm việc với CSDL Client / Server thì dễ hơn các kỹ thuật khác.

Phần lớn các nhà lập trình Visual Basic không tương tác trực tiếp với OLE DB. Thay vào đó, họ lập trình với ADO, mô hình đối tượng cung cấp giao diện với OLE DB.

1. Đối tượng Command : Chạy một câu truy vấn SQL hay chạy một thủ tục lưu trữ dữ liệu

Methods	Result - Type	Discription
Cancel		Huỷ bỏ việc thực thi của lệnh <i>Execute</i> hay <i>Open</i>
CreatParameter	Parameter	Tạo đối tượng Parameter mới
Execute	Recordset	- Thực thi các câu truy vấn SQL- Thực thi các hàm lưu trữ dữ liệu
Properties	Result - Type	Discription
ActiveConnection	Variant	Chỉ ra đối tượng Connection nào tác động lên đó
CommandText	String	Chứa lệnh thực thi trên CSDL
CommandTimeout	Long	Khoảng thời gian timeout khi thực thi lệnh
CommandType	CommandType	Kiểu của lệnh trong đối tượng Command
Name	String	Tên của đối tượng Command
Prepared	Boolean	Xác định có lưu hay không 1 bản dịch của lệnh trước khi thực thi
State	Long	Trạng thái của đối tượng Command
Collections	Result - Type	Discription
Parameters	Parameters	Chứa các đối tượng Parameters
Properties	Properties	Chứa các đối tượng Properties

2. Đối tượng Connection : Tạo liên kết tới kho lưu trữ dữ liệu

Methods	Result - Type	Discription
BeginTrans	Integer	Bắt đầu một transaction mới

Cancel		Hủy bỏ việc thực thi chưa hoàn tất
Close		Đóng một liên kết đang mở và các đối tượng khác đi kèm
CommitTrans		Lưu các thay đổi và kết thúc transaction hiện hành
Execute	Recordset	Chạy các truy vấn SQL, các hàm, ...
Open	Recordset	Mở một liên kết đến DataSourceCn.Open [connect], [userid], [password]
OpenSchema		Lấy các thông tin về DataBase schema
RollbackTrans		Hủy bỏ các thay đổi của transaction và kết thúc nó
Properties	Result - Type	Description
Attributes	Long	Xác định một hay nhiều kiểu đặc biệt của đối tượng connection
CommandTimeout	Long	Khoảng thời gian chờ khi thực thi lệnh
ConnectionString	String	Chứa thông tin dùng để tạo liên kết tới CSDL
ConnectionTimeout	Long	Khoảng thời gian timeout khi chờ thiết lập liên kết
CursorLocation	CursorLocationEnum String	Trả về vị trí của con trỏ hiện hành
DefaultDatabase	?	CSDL mặc định cho đối tượng Connection
IsolationLevel	?	Xác định mức độ tách biệt giữa các transaction
Mode	ModeEnum	Xác định các quyền thay đổi dữ liệu trong câu nối
Provider	String	Xác định tên nhà cung cấp cho đối tượng Connection

State	Long	Cho biết trạng thái của liên kết
Version	String	Cho biết phiên bản của ADO
Collections	Result - Type	Description
Errors	Errors	Chứa các đối tượng Error sinh ra bởi nhà cung cấp
Properties	Properties	Chứa các đối tượng Properties
Events	Result - Type	Description
BeginTransComplete		Xảy ra ngay sau khi thao tác BeginTrans hoàn tất
CommitTransComplete		Xảy ra ngay sau khi thao tác CommitTrans hoàn tất
ConnectionComplete		Bắt đầu ngay sau khi liên kết hoạt động
Disconnect		Bắt đầu ngay sau khi liên kết kết thúc
ExecuteComplete		Xảy ra sau khi lệnh được thực thi xong
InfoMessage		Xảy ra khi một thao tác ConnectionEvent thực hiện thành công
RollbackTransComplete		Bắt đầu sau khi thao tác Rollbacktrans hoàn tất
WillConnect		Xảy ra trước khi bắt đầu một liên kết
WillExecute		Xảy ra trước khi thực thi một lệnh trong liên kết

### 3. Đối tượng Error : Chứa các lỗi khi truy xuất đến CSDL

Methods	Result - Type	Description
Clear		Xoá tất cả các đối tượng Error ra khỏi tập hợp
Refresh		Cập nhật lại các thông tin về các đối tượng Error

Properties	Result - Type	Discription
Description	String	Chú thích tương ứng cho lỗi
HelpContext	Integer	Xác định ContextID trong file help chứa các lỗi
HelpFile	String	Tên của file Help
NativeError	Long	Mã lỗi đặc biệt của nhà cung cấp
Number	Long	Số ID của đối tượng Error
Source	String	Cho biết tên của đối tượng hay ứng dụng sinh ra lỗi
SQLState	String	Cho biết trạng thái của SQL cho việc xác định đối tượng Error
Collections	Result - Type	Discription
Errors		Được tạo ra khi ADO sinh ra lỗi
Count	Long	Cho biết số các đối tượng Error
Item	Error	Lập chỉ mục các đối tượng Error để tham khảo sau này

4. Đối tượng Parameter : Tham số hay đối số được dùng trong đối tượng Command khi gọi các hàm lưu trữ dữ liệu (Store Procedure)

Methods	Result - Type	Discription
AppendChunk		Đưa dữ liệu vào đối tượng Parameter dạng nhị phân
Properties	Result – Type	Discription
Attributes	Long	Xác định một hay nhiều kiểu đặc biệt cho đối tượng Parameter
Direction	ParameterDirectionEnum	Cho biết khi nào thì đối tượng Parameter là tham số nhập hay xuất, hoặc là trị trả về từ một hàm lưu trữ dữ liệu
Name	String	Tên của đối tượng Parameter
NumericScale	Byte	Cho biết độ dài của số thập phân (xuất ra màn hình)
Precision	Byte	Xác định độ chính xác của số

Size	Long	Định kích thước lớn nhất của đối tượng Parameter
Type	DataTypeEnum	Xác định kiểu dữ liệu của đối tượng Parameter
Value	Variant	Cho biết các giá trị đã gán cho đối tượng Parameter
Collections	Result - Type	Discription
Parameters		
Append		Đưa một đối tượng Parameter vào tập Parameters
Delete		Xoá một đối tượng Parameter trong tập
Refresh		Cập nhật đối tượng Parameter trong tập Parameters
Count	Long	Số đối tượng Parameter có trong tập
Item	Field	Dùng để đánh chỉ số các đối tượng trong tập Parameters
Properties	Properties	Chứa tất cả các đối tượng Property trong đối tượng Parameter

5. Đối tượng Property : Xác định các kiểu riêng biệt (tùy thuộc kiểu dữ liệu) của một đối tượng ADO

Properties	Result - Type	Discription
Attributes	Long	Xác định một hay nhiều kiểu đặc biệt cho đối tượng Property
Name	String	Tên của đối tượng Property
Type	DataTypeEnum	Xác định kiểu dữ liệu của đối tượng Property
Value	Variant	Cho biết các giá trị đã gán cho đối tượng Property

6. Đối tượng Recordset: Lưu toàn bộ tập các recordset từ một bảng hay từ kết quả thực thi câu lệnh SQL

Methods	Result - Type	Discription
---------	---------------	-------------

AddNew		Tạo một RecordSet mới
Cancel		Huỷ bỏ thao tác đang thực thi
CancelBatch		Huỷ bỏ các cập nhật bị treo
CancelUpdate		Huỷ bỏ các thay đổi đối với Recordset hiện hành
Clone		Tạo một bản sao của đối tượng Recordset
Close	Recordset	Đóng đối tượng Recordset và các đối tượng liên quan
CompareBookmarks	CompareEnum	So sánh 2 chỗ đánh dấu
Delete		Xoá Record hay một tập records hiện hành
Find		Tìm một Record
GetRows		Lấy nhiều Records đưa vào một mảng
GetString	String	Trả Recordset về dưới dạng chuỗi
Move		Di chuyển vị trí của Recordset hiện hành
MoveFirst		Di chuyển Recordset về vị trí đầu tiên trong bảng
MoveLast		Di chuyển Recordset về vị trí cuối cùng trong bảng
MoveNext		Di chuyển Recordset đến vị trí kế tiếp trong bảng
MovePrevious		Di chuyển Recordset đến vị trí trước đó trong bảng
NextRecordSet	Recordset	Xoá đối tượng Recordset hiện hành và trả về đối tượng Recordset kế tiếp
Open		Mở một Recordset
Requery		Cập nhật lại dữ liệu bằng cách chạy lại truy vấn ban đầu



Resync		Làm tươi lại dữ liệu trong đối tượng Recordset hiện hành
Save		Lưu Recordset vào File
Seek		Tìm chỉ mục của Recordset
Suports	Boolean	Xác định xem đối tượng Recordset có hỗ trợ chức năng gì đặc biệt không
Update		Cập nhật các thay đổi vào đĩa
UpdateBatch		Cập nhật các khối thay đổi vào đĩa
Properties	Result - Type	Discription
AbsolutePage	PositionEnum	Chỉ rõ trang nào mà record đang ở đó
AbsolutePosition	PositionEnum	Chỉ rõ vị trí của đối tượng Recordset mà chứa record hiện hành
ActiveCommand	Object	Xác định đối tượng Command nào mà Recordset đang phụ thuộc
ActiveConnection	Variant	Xác định đối tượng Connection nào mà Recordset đang phụ thuộc
BOF	Boolean	Cho biết con trỏ có ở vị trí trước bản ghi đầu tiên không
BookMark	Variant	Trả về vị trí record đã được đánh dấu
CacheSize	Long	Xác định số record được lưu trong Cache
CursorLocation	CursorLocationEnum	Vị trí của con trỏ
CursorType	CursorTypeEnum	Cho biết kiểu của con trỏ Recordset được dùng
DataMember	String	Tên của thành phần dữ liệu lấy về từ DataSource

DataSource	Object	Chỉ định đối tượng chứa dữ liệu
EditMode	EditModeEnum	Cho biết trạng thái soạn thảo của record hiện hành
EOF	Boolean	Cho biết con trỏ record có đứng ở sau record cuối cùng hay không
Filter	Variant	Xác định bộ lọc dữ liệu trong Recordset
Index	String	Tên của chỉ mục hiện hành
LockType	LockTypeEnum	Kiểu khoá áp đặt lên record hiện hành
MarshalOptions	MarshalOptionsEnum	Xác định các records sẽ được đưa về Server
MaxRecords	Long	Số records tối đa trả về trong đối tượng Recordset khi dùng truy vấn
PageCount	Long	Cho biết số trang dữ liệu chứa trong đối tượng Recordset
PageSize	Long	Xác định số records trong một trang
RecordCount	Long	Cho biết số record hiện hành trong đối tượng
Sort	String	Xác định một hay nhiều Field trong Recordset sẽ được sắp xếp
Source	String	Xác định nguồn dữ liệu trong đối tượng Recordset
State	Long	Cho biết trạng thái của Recordset hiện hành
Status	Integer	Cho biết trạng thái của Recordset hiện hành
StayInSync	Boolean	Xác định trong hệ phân cấp: record cha có thay đổi hay không khi các record con thay đổi

Collections	Result - Type	Discription
Fields	Fields	Chứa tất cả các đối tượng Field
Properties	Properties	Chứa tất cả các đối tượng Properties
Events	Result - Type	Discription
EndOfRecordset		Xảy ra khi cố di chuyển đến dòng cuối của Recordset
FetchComplete		Xảy ra khi tất cả các record đều được đưa vào Recordset
FetchProgress		Cho biết bao nhiêu hàng đang được lấy về từ lúc này
FieldChangeComplete		Xảy ra khi giá trị của một hay nhiều đối tượng field bị thay đổi
MoveComplete		Xảy ra khi vị trí hiện hành của Recordset thay đổi xong
RecordChangeComplete		Xảy ra khi 1 hay nhiều record bị thay đổi
RecordsetChangeComplete		Xảy ra khi Recordset bị thay đổi
WillChangeField		Xảy ra trước khi thay đổi trị của một hay nhiều đối tượng Fields
WillChangeRecord		Xảy ra trước khi thay đổi hàng trong Recordset
WillChangeRecordset		Xảy ra trước khi thay đổi RecordSet
WillMove		Xảy ra trước khi di chuyển vị trí Recordsaet

## 7. Đối tượng Field : Trường dữ liệu trong một kiểu dữ liệu bảng, SQL

Methods	Result - Type	Discription
AppendChunk		
GetChunk	Variant	
Properties	Result - Type	Discription

ActualSize	Long	Cho biết độ dài thực tế của giá trị Field
Attributes	Long	Cho biết một hay nhiều thuộc tính của Field
DataFormat	Variant	Cho biết định dạng dữ liệu trong Field
DefinedSize	Long	Cho biết độ lớn của đối tượng Field
Name	String	Xác định tên của đối tượng Field
NumericScale	Byte	Xác định độ dài của các trị trong đối tượng Field
OriginalValue	Variant	Cho giá trị nguyên thủy của đối tượng Field trước khi bị thay đổi
Precision	Byte	Xác định độ chính xác của số
Type	DataTypeEnum	Xác định kiểu dữ liệu của đối tượng Field
UnderlyingValue	Variant	Cho biết giá trị hiện thời của đối tượng Field trong CSDL
Value	Variant	Cho biết các giá trị đã gán cho đối tượng Field
Collections	Result - Type	Description
Properties	Properties	Chứa các đối tượng Properties
Fields		
Append		Đưa một đối tượng Field vào tập hợp Fields
Delete		Xoá một đối tượng Field trong tập hợp Fields
Refresh		Cập nhật một đối tượng Field trong tập hợp Fields
Count		Số đối tượng Fields có trong tập hợp
Item		Dùng để đánh chỉ số các đối tượng Field trong tập hợp Fields

Cài đặt và thiết lập tham, chiếu đến ADO

+ Chỉ sử dụng đối tượng truy cập bởi ADO

- Trong đề án Visual Basic, chọn Project, Reference

- Chọn hộp đánh dấu *"Microsoft ActiveX Data Object 2.0 Library"*

- Chọn OK

Truy xuất dữ liệu sử dụng công nghệ ADO

+ Kết hợp với các thư viện đối tượng khác

Nếu như bạn tạo ứng dụng có sử dụng nhiều thư viện đối tượng truy cập dữ liệu cùng lúc như *ADO* kết hợp với *DAO*, lúc đó cần phải phân biệt rõ đối tượng *RecordSet* của *DAO* hay của *ADO*. Giải pháp rằng bạn thực hiện khai báo tường minh cho các đối tượng dùng tên lớp đầy đủ *Ví dụ*: *ADODB.RecordSet*, *DAO.RecordSet* ...., hoặc bạn thiết lập một thứ tự ưu tiên như sau:

Ví dụ để cung cấp thứ tự ưu tiên cho *DAO* mức ưu tiên cao hơn *ADO* ta làm như sau:

- Trong đề án Visual Basic, chọn Project, Reference
- Chọn hộp đánh dấu tham chiếu "*Microsoft DAO 3.51 Object Library*" và "*Microsoft ActiveX Data Object 2.0 Library*"
- Chọn OK

(Giải pháp tốt nhất là chỉ ra tên lớp đầy đủ.)

Dùng đối tượng connection của ado để kết nối nguồn dữ liệu

Chỉ ra trình cung cấp OLE DB và kết nối bằng thuộc tính *ConnectionString*

Tùy thuộc vào trình cung cấp mà ta có chuỗi kết nối khác nhau. Mở kết nối bằng phương thức *Open* của đối tượng *ADODB.Connection*

Ví dụ:

```
Public ado as ADODB.Connection
```

```
Private sub mbado()
```

```
Set ado = New ADODB.Connection
```

```
Dim Str as String
```

```
Str="Provider=SQLOLEDB.1;Password=vinhhue;          User          ID=hue;Initial  
Catalog=QLBH;Data Source=huecm-kcntt"
```

```
ado.ConnectionString = Str
```

```
ado.Open
```

Truy xuất dữ liệu sử dụng công nghệ ADO

End Sub

Mở và đóng kết nối đến nguồn dữ liệu

Dùng phương thức *Connection.Open* để mở kết nối đến nguồn dữ liệu.

Cn.Open [connect], [userid], [password]

- Toàn bộ các tham số đều là tùy chọn, nếu ta không cung cấp các tham số đến phương thức Open, ta có thể cung cấp bởi phương thức *ConnectionString*, hiệu quả như nhau.

Dim Cn As Connection

Set Cn= New Connection

Cn.Open "DN= JetNovelty"

- Khi đã hoàn tất với đối tượng Connection của ADO, ta nên luôn luôn đóng nó bằng Close

Cn. Close

Dùng đối tượng recordset của ADO để thao tác với dữ liệu

- Đối tượng *ReccorddSet* của ADO tương tự như *ReccorddSet* của DAO và RDO:

- Cách tạo đối tượng *ReccorddSet* của ADO tương tự như RDO chỉ thay đổi là ADO không đòi hỏi đối tượng *Connection* ngầm.

*Ví dụ:*

(Các thao tác của con trỏ ReccorddSet trên ADO tương tự như ReccorddSet trên DAO)

Thêm mẫu tin mới

+ Mở *ReccorddSet*

+ Thi hành phương thức *AddNew*

+ Gán giá trị mới cho các trường trong đối tượng *ReccorddSet*

+ Lưu mẫu tin bằng thi hành phương thức *Update*

+ Đóng *ReccorddSet*

Truy xuất dữ liệu sử dụng công nghệ ADO

Sửa mẫu tin

+ Mở *ReccorddSet*

+ Khác với Dao đối tượng ADO không có phương thức Edit

+ Gán giá trị mới cho các trường trong đối tượng *ReccorddSet*

+ Lưu mẫu tin bằng thi hành phương thức *Update*

+ Đóng *ReccorddSet*

c. Dùng đối tượng Field của ADO để truy cập dữ liệu trong RecordSet

VD:

```
Private Sub Form_Load()
```

```
Set con = New ADODB.Connection
```

```
con.ConnectionString = "Provider=SQLOLEDB.1;Password=vinhhue; User  
ID=hue;Initial Catalog=QLBH;Data Source=huecm-kcntt"
```

```
con.Open
```

```
End Sub
```

-----

```
Public Sub laydulieu()
```

```
Dim rs As ADODB.Recordset
```

```
Set rs = New ADODB.Recordset
```

```
rs.Source = "Select * from loai"
```

```
rs.ActiveConnection = con
```

```
rs.Open
```

```
Do While Not rs.EOF
```

```
cbodanhmuc.AddItem rs.Fields(1)
```

Truy xuất dữ liệu sử dụng công nghệ ADO

```
'cbodanhmuc.AddItem rs!tenloai
```

```
rs.MoveNext
```

```
Loop
```

```
End Sub
```

-----

```
Private Sub Form_Unload(Cancel As Integer)
```

```
con.Close
```

```
Set rs = Nothing
```

```
End Sub
```

### Làm việc với con trỏ

ADO hỗ trợ một số kiểu con trỏ. Ngoài việc cung cấp hỗ trợ duyệt qua từng mẫu tin tại một thời điểm, các kiểu con trỏ khác nhau cho phép ta điều khiển cách quản lý của một Recordset.

Quy định vị trí của con trỏ bằng cách gán giá trị cho thuộc tính Recordset. Các kiểu con trỏ của đối tượng connection

Kiểu con trỏ	Hằng	Mô tả
Phía Client	adUseClient	Tạo con trỏ phía Client
Phía Server	adUseServer	Tạo con trỏ phía Server

Chọn con trỏ Client nghĩa là ADO và OLEDB xử lý các hoạt động của con trỏ. Con trỏ kiểu Client không có sẵn trên Server. Trong ADO ta có thể tạo một Recordset không kết nối, cho phép ta thao tác với các mẫu tin mà không có kết nối thường xuyên đến Server. Khả năng này là một chức năng của thư viện con trỏ phía Client.

Trong ADO, thuộc tính CurorLocation áp dụng cho cả đối tượng Recordset và connection. Nếu ta gán thuộc tính CurorLocation của đối tượng conection, tất cả Recordset mà ta tạo từ kết nối đó đều có cùng vị trí con trỏ như đối tượng Connection

Ngoài việc chỉ ra vị trí con trỏ, ta có 4 khả năng tạo 4 kiểu con trỏ khác nhau trong ADO. Việc chọn lựa chọn con trỏ tùy theo sự cân đối giữa chức năng và khả năng hoạt động.



Chỉ ra kiểu con trỏ bằng cách gán thuộc tính CursorType của đối tượng Recordset. Đây là các kiểu con trỏ có thể tạo trong ADO

Kiểu con trỏ	Hằng	Mô tả
Forward-Only	adOpenForwardOnly	Không dùng con trỏ - ta chỉ có thể chuyển về phía trước trong Recordset, sử dụng phương thức MovePrevious và MoveFirst sẽ sinh lỗi
Keyset(trong DAO được gọi là Dynaset)	adOpenKeyset	Ta không thể thấy các mẫu tin mới do người sử dụng khác thêm và, nhưng khi họ sửa đổi, hay xóa mẫu tin sẽ ảnh hưởng đến Recordset ta đang làm việc; đây là kiểu con trỏ hiệu quả nhất, đặc biệt là khi Recordset khá lớn
Dynamic	adOpenDynamic	Ta có thể thấy toàn bộ thay đổi trên dữ liệu do người sử dụng khác thực hiện trong khi ta đang mở Recordset; đây là kiểu con trỏ ít hiệu quả nhất, nhưng mạnh mẽ
Static (trong DAO gọi là snap-short)	adOpenStatic	Bản sao của toàn bộ dữ liệu của một Recordset; kiểu này đặc biệt hữu dụng khi ta đang tìm kiếm dữ liệu hay ta đang thi hành báo cáo; kiểu con trỏ này rất hữu dụng cho những Recordset nhỏ.

Dĩ nhiên lý do để chọn con trỏ Forward-Only thay vì các kiểu con trỏ khác là khả năng hoạt động của nó, nếu ta chỉ hiển thị dữ liệu chứa trong cơ sở dữ liệu - con trỏ kiểu Forward-only sẽ làm khả năng hoạt động của ứng dụng trở nên hiệu quả hơn.

*Chú ý:* Nếu trình cung cấp dữ liệu không thể tạo ra kiểu con trỏ mà ta yêu cầu, nó sẽ tạo ra kiểu con trỏ mà nó có thể. Nó sẽ không báo lỗi trừ khi ta cố thi hành một tác vụ nào đó vốn bị cấm đối với kiểu con trỏ này.

≅ Xác định con trỏ và các tính năng khác được hỗ trợ bởi một trình cung cấp

OLE DB và ADO được thiết kế để cho phép truy cập đến nguồn dữ liệu, ứng dụng cần xác định các tính năng do một trình cung cấp dữ liệu hỗ trợ, có thể là trong khi một hệ cơ sở dữ liệu quan hệ cho phép tạo một con trỏ Forward-only ở phía Server, hệ cơ sở dữ liệu trên máy cá nhân hoặc cơ sở dữ liệu dựa trên tập tin có thể không có tính năng này. Phương thức Supports của đối tượng ADO Recordset xác định kiểu con trỏ do trình cung cấp dữ liệu cung cấp dữ liệu hỗ trợ.

Bảng sau đây liệt kê các giá trị truyền vào phương thức Supports để xác định tính năng được hỗ trợ bởi đối tượng recordset.

Hằng	Mô tả
adAddNew	Thêm mẫu tin mới vào Recordset
AdapproxPosition	Thuộc tính AbsolutePage và AbsolutePosition có sẵn; chúng được dùng để kết hợp thuộc tính PageSize và PageCount của đối tượng Recordset để cho phép xác định trang chứa mẫu tin hiện hành.
adBookmark	Có thể quy định đánh dấu trang sách trong Recordset
adDelete	Mẫu tin được xóa trong recordset
adHoldRecords	Mẫu tin được trả về từ cơ sở dữ liệu mà không cần ghi các sửa đổi hiện có vào Server
adMovePrevious	Có thể cuộn tới lui trong Recordset
adResync	Phương thức Resync có sẵn
adUpdate	Recordset cập nhật được
adUpdateBatch	Recordset có thể cập nhật được hàng loạt với phương thức UpdateBatch. Ta có thể nạp các sửa đổi trên nhiều mẫu tin. Trong một hoạt động duy nhất, cải tiến hiệu quả Client / Server