



Effettuare una connessione ad un database attraverso ADO

http://www.vbsimple.net/activity/act_04.htm

Richiesta di: [Rosario Manto](#) - 7 Novembre 2000

Difficoltà: ►►► 3 / 5

Per utilizzare un database all'interno di un progetto Visual Basic è necessario innanzitutto definire il tipo di interfaccia da utilizzare. Per ogni utilizzo c'è un'interfaccia che è *migliore* di un'altra. Le interfacce rivolte ai dati più conosciute sono [DAO](#) ([Data Access Object](#)) e [ADO](#) ([ActiveX Data Objects](#)).

La prima è un'interfaccia più vecchia, ma più robusta e completa.

La seconda invece è più moderna, un po' più macchinosa, ma permette di utilizzare qualunque fonte dati.

L'interfaccia DAO è molto ben documentata nella guida in linea di Visual Basic 5, mentre ADO non è neanche accennato perché uscito dopo la pubblicazione di Visual Basic 5.

Vedremo quindi come effettuare una connessione a database attraverso ADO. Il database può essere un file sul disco rigido di cui sappiamo il tipo e la posizione oppure un'origine dati [DSN](#) di cui sappiamo soltanto il nome.

In ogni caso all'atto della connessione con ADO dobbiamo conoscere il [Provider](#) da utilizzare, ovvero la soluzione di accesso al database.

Nel nostro esempio faremo una semplice connessione ad un DSN ODBC. Sfrutteremo il DSN DatabaseCDROM che abbiamo creato in un'[altra richiesta](#).

Tutte le connessioni ADO richiedono l'impostazione di una stringa di connessione, un insieme di parametri separati da un punto e virgola. La sintassi tipica di una stringa di connessione è:

```
"Provider=MSDASQL; DRIVER=nomeDriver; DSN=nomeDSN; UID=utente;
PWD=password"
```

Dove Provider=MSDASQL è il provider utilizzato per accedere ai DSN. Il campo Provider non è obbligatorio. In sua assenza verrà usato il valore di default, ovvero **MSDASQL**. Un altro provider molto comune è **Microsoft.Jet.OLEDB.4.0**, utilizzato per accedere a database di Access 2000.


DRIVER è il parametro che specifica il nome del driver da utilizzare. Per Access 97 il suo nome è **MS ACCES**.

DSN naturalmente è il parametro che indica il DSN da aprire. In caso di un DSN su file il parametro si chiamerà FILEDSN. In taluni casi è necessario sostituire la parola DSN con DBQ (per le connessioni DSN-less) oppure con Data Source.

UID indica un nome utente da utilizzare per connettersi al server.

PWD è la parola d'ordine corrispondente all'utente specificato in UID.

Se un database non richiede un nome utente sarà necessario specificare l'utente ADMIN oppure saltare il parametro.

Effettuiamo in pratica la connessione al nostro DSN. Abbiamo creato un form ed inserita una ListBox  di nome ListaProgrammi in esso. All'interno della routine Form_Load scriviamo il codice seguente:

```
1. Dim CONNESSIONE As New ADODB.Connection
2. Dim TABELLA As New ADODB.Recordset
3. CONNESSIONE.Open "Provider=MSDASQL; Driver=MS Access; DSN=DatabaseCDROM"
4. TABELLA.Open "PlayPC", CONNESSIONE
5. While Not TABELLA.EOF
6.     ListaProgrammi.AddItem TABELLA.Fields(0).Value
7.     TABELLA.MoveNext
8. Wend
9. TABELLA.Close
10. CONNESSIONE.Close
11. Set TABELLA = Nothing
12. Set CONNESSIONE = Nothing
```

Le prime due righe dichiarano due variabili: una per la connessione ed una per la tabella ([Recordset](#)) da aprire.

Subito sotto, alla riga 3, abbiamo l'istruzione di apertura di connessione con un DSN.

La riga successiva dice di aprire la tabella **PlayPC** dalla connessione.

Segue un ciclo che si ripete fintanto che la tabella non è alla fine. EOF indica End Of File, ovvero fine del file.

All'interno del ciclo c'è scritto di aggiungere (*AddItem*) all'interno di ListaProgrammi il valore del primo campo (0) della tabella indicata da TABELLA.

Compiuta quest'operazione si sposta al record successivo. Questo ciclo continuerà a ripetersi per ogni singolo record della tabella.

Alla fine del ciclo, dobbiamo chiudere la tabella e la connessione, che, altrimenti rimarrebbero aperte, sebbene la vita delle variabili CONNESSIONE e TABELLA si appresta a terminare. Ci rimarrebbero quindi delle risorse bloccate inaccessibili, con un conseguente peggioramento del sistema.

Nei processi brevi è bene utilizzare tecniche di questo genere (Apertura - Uso - Chiusura), mentre nei grossi programmi la tecnica da adottare è quella dell'apertura connessione all'avvio del programma e la chiusura della connessione coincide con la chiusura del programma. Questo permette di risparmiare un po' di tempo negli accessi al database.

Una volta chiuse tabella e connessione, se siamo sicuri che non ci serviremo più delle variabili dichiarate, **siamo obbligati** a deallocare la memoria, ovvero a rendere disponibile la memoria occupata dalle variabili. In mancanza di questo la memoria rimarrà bloccata nonostante la chiusura del database e della connessione.

È buona regola, in presenza di dichiarazioni dinamiche di oggetti, ricordarsi di deallocare la memoria mediante un'istruzione Set Variabile = Nothing.

Vediamo il risultato della nostra operazione? Eccolo.



Figura 1

L'apertura di una connessione con ADO potrà sembrare complessa, ma basta conoscere due o tre parametri per avere a disposizione connessioni ai più disparati database. Infatti, mediante la specifica di Provider e Driver, ADO può aprire ogni tipo di base dati.

[Fibia FBI](#)

8 Novembre 2000



[Torna all'introduzione delle Richieste dei lettori](#)
