


[Home Page](#) 
[Informazioni](#) 
[Aiuto](#) 

Accedere ad un database protetto da password

http://www.vbsimple.net/activity/act_17.htm

Richiesta di: [Francesca Mugnani](#) - 9 Luglio 2001

Difficoltà:  3 / 5

Ho creato un database Access che ho protetto con una password, vorrei creare un form per poter accedere al database attraverso una password. Come posso fare?

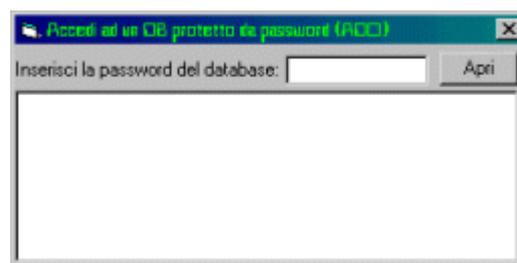
Mi serve proteggere il database e far in modo che l'utente possa aprire il form in VB solo con una password.

Il problema è abbastanza semplice e sarà sviluppato per entrambe le [interfacce](#) dati [ADO](#) e [DAO](#). Per una spiegazione più approfondita sulle interfacce dati vedi la richiesta per [Effettuare una connessione ad un database attraverso ADO](#).

Il primo esempio utilizzerà ADO mentre il secondo farà uso di DAO.

Per semplificare il problema utilizzeremo un solo form per svolgere la connessione al database protetto da password e mostrarne i dati.

Esso si compone di una *Label* descrittiva, una casella di testo di nome **txtPassword** con la proprietà  *PasswordChar* impostata a "*", un *CommandButton*  di nome **btnApri** ed una *ListBox*  di nome **lstLibri**.



Il funzionamento è molto semplice: il nostro database è protetto da una password ed è possibile leggerne i dati soltanto immettendo la password corretta nella casella di testo e premendo il pulsante **Apri**. Prima di vedere il codice è necessario aggiungere un riferimento a *Microsoft ActiveX Data Objects 2.x Library* tramite la voce Riferimenti  del [menu Progetto](#). Il codice è alquanto semplice:

```

1. Option Explicit
2.
3. Private Sub btnApri_Click()
4.     Dim Conn As ADODB.Connection
5.     Dim RS As ADODB.Recordset
6.     Set Conn = New ADODB.Connection
7.     Set RS = New ADODB.Recordset
8.     On Error GoTo Err_Handler

```

Alla riga 4 viene dichiarata la variabile **Conn** di tipo *Connection*, che sarà utilizzata per stabilire una connessione con il database. Alla riga 5 viene dichiarata la variabile **RS** di tipo *Recordset* che servirà per estrarre i dati da una tabella del database.

Di seguito vengono [istanziate](#) le due variabili.

Alla riga 8 viene aggiunta una istruzione di gestione degli errori. In caso di un errore sarà

mostrata una finestra di avviso con il messaggio di errore. Questa istruzione è necessaria perché in caso di password errata sarà generato un errore da intercettare.

```

9.     Conn.Provider = "Microsoft.Jet.OLEDB.4.0"
10.    Conn.Properties("Jet OLEDB:Database Password").Value = txtPassword.Text
11.    Conn.Open "Data Source=libri.mdb"
12.    RS.Open "SELECT * FROM LIBRI", Conn, adOpenStatic, adLockPessimistic

```

Alla riga 9 viene scelto il tipo di [provider](#) da utilizzare per aprire la connessione al database; l'utilizzo del provider **OLEDB** migliora la velocità di accesso al database. Il provider scelto è quello Jet per accedere ai database di tipo Access 97 e 2000.

Alla riga 10 impostiamo la password per il database da aprire mediante la proprietà **Jet OLEDB:Database Password**. Assegniamo ad essa il valore della casella di testo **txtPassword**.

Impostati tali dati sarà possibile aprire la connessione al database "libri.mdb" ed in seguito aprire la tabella **LIBRI**. Se la password che ha fornito l'utente è corretta il [Recordset RS](#) sarà aperto; in caso contrario sarà generato un errore che sarà gestito dalla funzione **Err_Handler**.

```

13.    lstLibri.Clear
14.    Do While Not RS.EOF
15.        lstLibri.AddItem RS.Fields(0).Value
16.        RS.MoveNext
17.    Loop
18.    RS.Close
19.    Conn.Close
20.    Set RS = Nothing
21.    Set Conn = Nothing
22.    Exit Sub

```

Aperta la tabella sarà elencato il suo contenuto ed inseriti i dati nella ListBox **lstLibri**. In seguito saranno chiuse e deallocati Recordset e Connessione. Fatto questo alla riga 22 terminerà la Sub.

```

24. Err_Handler:
25.    MsgBox Err.Description, vbCritical + vbOKOnly, Err.Source
26.    Set RS = Nothing
27.    Set Conn = Nothing
28. End Sub
29.

```

In caso di un errore sarà mostrata una finestra di messaggio con l'errore generato e saranno deallocati Recordset e Connessione.

```

30. Private Sub Form_Load()
31.    MsgBox "La password di questo database è "ciccio""
32. End Sub

```

Prima di lanciare il programma è bene indicare qual'è la password del database.

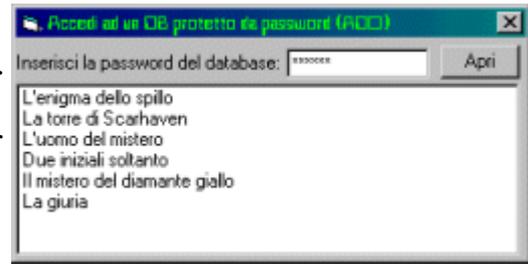
Eseguiamo il programma e proviamo ad inserire una password errata. Sarà mostrato un messaggio di errore esplicativo.

Senza la funzione di gestione degli errori il nostro programma avrebbe generato un errore e si sarebbe chiuso inesorabilmente.



Se invece la password inserita è quella corretta saranno mostrati i titoli dei libri presenti nel database.

Poiché abbiamo impostato la proprietà PasswordChar la password non sarà leggibile ma appariranno degli asterischi al posto delle singole lettere.



Viene affrontato lo stesso problema utilizzando DAO invece di ADO. È pertanto necessario specificare un riferimento  a *Microsoft DAO 3.x Object Library* dal [menu Progetto](#).

```

1. Private Sub btnApri_Click()
2.     Dim DBProtetto As DAO.Database
3.     Dim RS As DAO.Recordset
4.     On Error GoTo Err_Handler
5.     Set DBProtetto = OpenDatabase("libri.mdb", True, False, "MS Access; pwd=" &
        txtPassword.Text)
6.     Set RS = DBProtetto.OpenRecordset("SELECT * FROM LIBRI")

```

Alla riga 2 viene dichiarata una variabile di tipo *Database* di nome **DBProtetto**, mentre alla riga 3 viene dichiarata una variabile di nome **RS** di tipo Recordset.

Anche in questo caso viene installata una funzionalità di gestione degli errori (riga 4).

Alla riga 5 viene aperto il database mediante il metodo  `OpenDatabase`; viene specificato il database da aprire, i due parametri successivi (apertura esclusiva e apertura in sola lettura) sono di tipo booleano ed obbligatori se si specifica una password.

La password - specificata nella casella di testo *txtPassword* - viene indicata mediante parametro **pwd** specificato nella stringa di connessione.

Alla riga 6 viene aperto il Recordset.

```

7.     lstLibri.Clear
8.     Do While Not RS.EOF
9.         lstLibri.AddItem RS.Fields(0).Value
10.        RS.MoveNext
11.    Loop
12.    RS.Close
13.    DBProtetto.Close
14.    Set RS = Nothing
15.    Set DBProtetto = Nothing
16.    Exit Sub
17. Err_Handler:
18.    MsgBox Err.Description, vbCritical + vbOKOnly, Err.Source
19.    Set RS = Nothing
20.    Set DBProtetto = Nothing
21. End Sub

```

Il resto del codice è identico al precedente. Sarà riempita la ListBox *lstLibri* con i dati estratti dalla tabella *LIBRI*; in seguito vengono chiusi e deallocati Recordset e Database.

Alle righe 18-20 segue la funzione di gestione degli errori obbligatoria per rintracciare l'errore generato in caso di password errata.

DAO sta diventando sempre più raro lasciando il posto al suo "avversario" ADO più standard ed in grado di supportare quasi ogni tipo di database.

Ecco perché sono state specificate entrambe le soluzioni di accesso ai dati.

[Fibia FBI](#)
15 Luglio 2001



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