



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

Creare un DSN via codice o con una Wizard

http://www.vbsimple.net/activity/act_13.htm

Richiesta di: [Gianni Rondinini](#) - 23 Marzo 2000

Difficoltà:  3 / 5

Come si crea un DSN mediante codice Visual Basic?

In un'altra richiesta è stato visto come [configurare manualmente un DSN](#) mediante il pannello di controllo. A volte può essere molto utile creare un [DSN](#) all'interno di un programma, evitando all'amministratore della rete di effettuare quest'operazione.

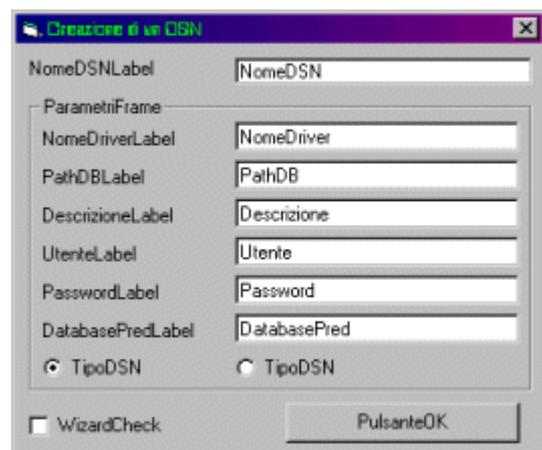
In questa richiesta saranno mostrate due soluzioni, sebbene ne esista anche una terza, leggermente più complessa, che consiste nel manipolare i valori del [registro](#) di Windows.

Per effettuare l'operazione utilizzeremo due funzioni [API](#) della [libreria](#) ODBC32 ovvero la libreria del Pannello di Controllo [ODBC](#). Le due funzioni in questione sono **SQLConfigDataSource** (per creare il DSN via codice) e **SQLCreateDataSource** (per richiamare la finestra Wizard guidata).

Cominciamo a definire il nostro form che permetterà di creare il DSN voluto.

Inseriremo una serie di coppie di *Label* **A** e *TextBox* **abl** come mostrato nella figura. Ad ogni *Label* corrisponde a fianco una *TextBox*.

Abbiamo inserito anche un *Frame* **xv** per raggruppare visivamente i parametri del DSN. Al suo interno, oltre le *Label* e le *TextBox* abbiamo anche un [array](#) (vettore) di due *OptionButton* **o** di nome **TipoDSN**, con indici 0 e 1. Essi serviranno per specificare il tipo di DSN da creare.



In fondo al form abbiamo una *CheckBox* **W** di nome **WizardCheck**. Per ultimo inseriamo un *CommandButton* **OK** di nome **PulsanteOK**.

Prima di vedere il codice cambiamo le [proprietà](#)  **Caption** e **Text** corrispondenti nella maniera visibile nella figura sotto.

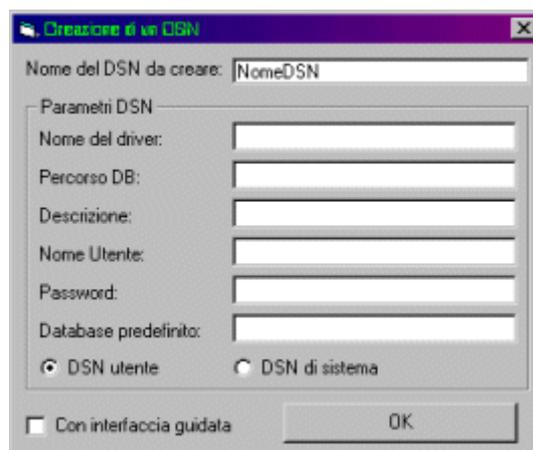


Figura 2

La CheckBox in basso determina la modalità di creazione del DSN. Quando essa è attivata (checked) il DSN sarà creato mediante finestra Wizard guidata, provveduta dal Pannello di Controllo.

Pertanto, l'attivazione della CheckBox andrà a bloccare tutti i controlli riguardanti i parametri del DSN.

Se invece la CheckBox è disattivata il DSN sarà creato mediante codice, leggendo i valori dalle TextBox.

1. Option Explicit
2. Private Declare Function SQLConfigDataSource Lib "ODBC32.DLL" (ByVal hwndParent As Long, ByVal fRequest As Integer, ByVal lpszDriver As String, ByVal lpszAttributes As String) As Long
3. Private Declare Function SQLCreateDataSource Lib "ODBC32.DLL" (ByVal hwndParent As Long, ByVal lpszDSN As String) As Long
- 4.
5. Private Const ODBC_ADD_DSN = 1
6. Private Const ODBC_ADD_SYS_DSN = 4
- 7.

Alla riga 2 vediamo la dichiarazione della funzione **SQLConfigDataSource** che richiede il passaggio di quattro parametri. Il primo parametro (*hwndParent*) identifica l'[handle](#) di una finestra: esso servirà se volessimo richiamare la finestra di dialogo di configurazione del DSN; il secondo parametro (*fRequest*) è un numero indicante l'operazione da svolgere: in questo nostro esempio indicherà se aggiungere un DSN utente oppure di sistema; il terzo parametro (*lpszDriver*) indica la stringa contenente il Driver da utilizzare; l'ultimo parametro (*lpszAttributes*) è una stringa contenente la lista degli attributi da assegnare al DSN, quale il nome utente, la password ed altri.

La dichiarazione della riga 3 identifica la funzione **SQLCreateDataSource** che verrà utilizzata per richiamare la finestra di dialogo di creazione di un DSN.

La differenza tra le due finestre di dialogo che è possibile richiamare mediante le due funzioni è che la seconda permette di definire in maniera grafica il tipo di DSN da creare ed il driver da utilizzare. In sostanza, la **SQLCreateDataSource**, dopo aver richiesto tali dati, richiama la funzione **SQLConfigDataSource** con il driver corrispondente.

Alle righe 4 e 5 abbiamo definito due [costanti](#) API, di nome **ODBC_ADD_DSN** e

ODBC_ADD_SYS_DSN, che verranno utilizzate per identificare il tipo di operazione da effettuare. Esse saranno utilizzate nella chiamata alla funzione `SQLConfigDataSource`.

```

8. Private Sub WizardCheck_Click()
9.     Dim STATO As Boolean
10.    STATO = (WizardCheck.Value = vbUnchecked)
11.    ParametriFrame.Enabled = STATO
12.    NomeDriverLabel.Enabled = STATO
13.    NomeDriver.Enabled = STATO
14.    PathDBLabel.Enabled = STATO
15.    PathDB.Enabled = STATO
16.    DescrizioneLabel.Enabled = STATO
17.    Descrizione.Enabled = STATO
18.    UtenteLabel.Enabled = STATO
19.    Utente.Enabled = STATO
20.    PasswordLabel.Enabled = STATO
21.    Password.Enabled = STATO
22.    DatabasePredLabel.Enabled = STATO
23.    DatabasePred.Enabled = STATO
24.    TipoDSN(0).Enabled = STATO
25.    TipoDSN(1).Enabled = STATO
26. End Sub
27.

```

Vediamo subito la semplicissima funzione che regola l'[evento](#)  Click sulla CheckBox **WizardCheck**. Alla riga 10 otteniamo lo stato che assegneremo alle varie Label e TextBox in base alla CheckBox. Se il valore è `vbUnchecked`, ovvero non è attivata, la variabile `STATO` avrà il valore `True`, altrimenti sarà `False`. Ottenuto tale valore cambiamo lo stato delle proprietà *Enabled* dei vari controlli, in modo da attivarli o disattivarli in maniera relativa alla CheckBox.

L'ultima funzione riguarda la pressione del pulsante OK.

```

28. Private Sub PulsanteOK_Click()
29.     Dim ODBCSTRING As String
30.     Dim RISULTATO As Long
31.     If WizardCheck.Value = vbChecked Then
32.         ODBCSTRING = NomeDSN.Text
33.         RISULTATO = SQLCreateDataSource(Me.hWnd, ODBCSTRING)

```

Alla riga 31 viene verificato lo stato della CheckBox **WizardCheck**. Se essa è attivata dovrà essere richiamata la funzione mediante interfaccia guidata, altrimenti la creazione avverrà esclusivamente tramite codice.

Se la CheckBox è attivata verrà letto il nome del DSN dalla casella **NomeDSN**. Tale valore sarà passato alla funzione `SQLCreateDataSource`. L'[handle](#) richiesto dalla funzione sarà quello del nostro form, in modo che la finestra sia richiamata in maniera modale, ovvero essa bloccherà l'esecuzione del codice fintanto che l'operazione non si sarà conclusa.

Il risultato dell'operazione sarà memorizzato nella variabile **RISULTATO**.

```

33.     Else
34.         ODBCSTRING = "DSN=" & NomeDSN.Text & vbNullChar
35.         ODBCSTRING = ODBCSTRING & "DBQ=" & PathDB.Text & vbNullChar
36.         ODBCSTRING = ODBCSTRING & "DESCRIPTION=" & Descrizione.Text & vbNullChar
37.         ODBCSTRING = ODBCSTRING & "UID=" & Utente.Text & vbNullChar
38.         ODBCSTRING = ODBCSTRING & "PWD=" & Password.Text & vbNullChar
39.         ODBCSTRING = ODBCSTRING & "DATABASE=" & DatabasePred.Text & vbNullChar
40.         ODBCSTRING = ODBCSTRING & vbNullChar

```

```
41.         If TipoDSN(0).Value = True Then
42.             RISULTATO = SQLConfigDataSource(ByVal 0&, ODBC_ADD_DSN,
NomeDriver.Text, ODBCSTRING)
43.         Else
44.             RISULTATO = SQLConfigDataSource(ByVal 0&, ODBC_ADD_SYS_DSN,
NomeDriver.Text, ODBCSTRING)
45.         End If
46.     End If
47.
```

Se la **CheckBox** è disattivata il DSN verrà creato mediante codice. Sarà pertanto creata una stringa di connessione, del tutto simile a quella da utilizzare per [richiamare il DSN al fine di utilizzare il database cui punta il DSN](#).

La stringa di connessione si compone di una serie di coppie di valori. Ogni coppia contiene il proprio nome ed il valore. Una coppia è separata dalla successiva mediante un carattere terminatore (il carattere il cui codice ASCII è 0) corrispondente alla costante **vbNullChar**.

La stringa conterrà il nome del DSN, l'origine dati cui punta, la descrizione, il nome utente, la password e il database predefinito in case di database multipli (righe 34-39).

La stringa deve concludersi con un carattere terminatore aggiuntivo (riga 40).

Creata la stringa sarà possibile richiamare la funzione di creazione del DSN. In questo esempio abbiamo inserito due **OptionButton** per determinare se il DSN deve essere di sistema oppure utente. Pertanto, alla riga 41, viene controllato se il primo *OptionButton* è attivato.

In base a tale controllo viene richiamata la funzione API **SQLConfigDataSource** e le verrà passata la costante corrispondente al tipo di DSN da creare. Oltre a tale valore sarà passato anche il nome del driver (letto dalla casella di testo **NomeDriver**) e la stringa di connessione **ODBCSTRING** (righe 42 e 44). Il valore restituito dalla chiamata alla funzione API sarà memorizzato nella variabile **RISULTATO**.

```
49.         If RISULTATO <> 0 Then
50.             MsgBox "Creazione DSN riuscita!", vbInformation + vbOKOnly, "Crea DSN"
51.         Else
52.             MsgBox "Creazione DSN fallita!", vbCritical + vbOKOnly, "Crea DSN"
53.         End If
54.     End Sub
```

È giunto il momento di controllare se la chiamata alla funzione di creazione del DSN ha avuto effetto. Sarà controllato il valore della variabile **RISULTATO**.

Se il risultato è diverso da 0 la creazione è riuscita e sarà mostrato un messaggio informativo (riga 50). Nel caso contrario sarà mostrato un messaggio di errore (riga 52).

Possiamo passare alla prova del programma. Eseguiamo il progetto, riempiamo i campi con i dati in maniera corretta.

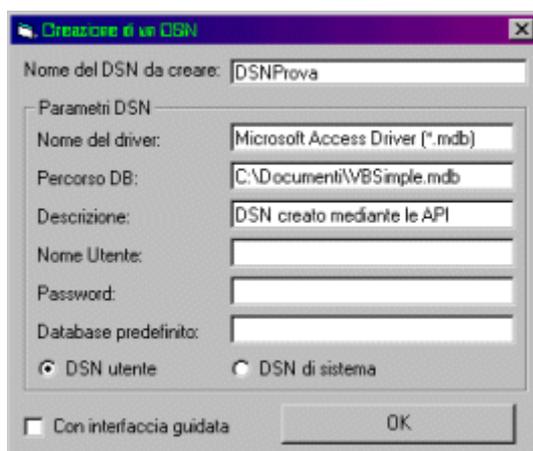


Figura 3

È fondamentale impostare il nome del DSN e il nome del driver.

In caso di mancato riempimento degli altri campi il DSN sarà comunque creato, ma il suo funzionamento dipende dalla corretta specificazione degli altri dati.

L'elenco dei driver utilizzabili è reperibile nel gestore dei DSN  nel Pannello di Controllo alla voce **Drivers**.

L'attivazione della CheckBox WizardCheck comporta la disattivazione delle varie Label e TextBox insieme al Frame  che le contiene.

La pressione del pulsante OK in questa situazione non creerà il DSN direttamente ma porrà una serie di domande e richieste all'utente.

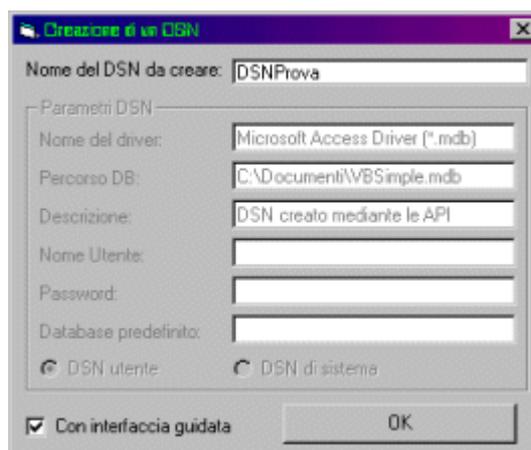


Figura 5

In ogni caso, mediante utilizzo di interfaccia guidata o no, al termine del tentativo di creazione sarà mostrata una finestra di messaggio.



Figura 6

Fibia FBI
25 Marzo 2001



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