




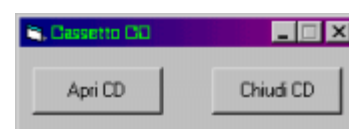
Apertura e chiusura del cassetto del lettore CD

http://www.vbsimple.net/howto/ht_039.htm

Difficoltà:  2 / 5

In alcuni programmi multimediali può essere piacevole aprire e chiudere il cassetto del lettore CD per invitare l'utente ad inserire un disco nell'lettore. Si tratta di un'operazione di cui è meglio non abusare, ma in quei rari casi in cui fosse necessaria un'operazione del genere, le funzioni [MCI](#) provvedono una via molto semplice.



Il progetto si compone di un solo form con due semplici CommandButton  di nome **cmdApriCD** e **cmdChiudiCD** e di un breve e semplice codice:



```

1. Option Explicit
2.
3. Private Type MCI_OPEN_PARMS
4.     dwCallback As Long
5.     wDeviceID As Long
6.     lpstrDeviceType As String
7.     lpstrElementName As String
8.     lpstrAlias As String
9. End Type
10.
11. Private Const MCI_OPEN = &H803&
12. Private Const MCI_OPEN_TYPE = &H2000&
13. Private Const MCI_SET = &H80D&
14. Private Const MCI_SET_DOOR_OPEN = &H100&
15. Private Const MCI_SET_DOOR_CLOSED As Long = &H200&
16. Private Const MCI_CLOSE = &H804&
17.
18. Private Declare Function mciSendCommand Lib "winmm.dll" Alias
    "mciSendCommandA" (ByVal wDeviceID As Long, ByVal uMessage As Long, ByVal dwParam1
    As Long, ByVal dwParam2 As Any) As Long
19.
20. Private openParams As MCI_OPEN_PARMS
21.

```

Innanzitutto è necessario definire i tipi, le costanti  e le funzioni [API](#): alla riga 3 viene definito un nuovo tipo di dati  denominato **MCI_OPEN_PARMS** che verrà utilizzato per contenere i parametri sull'unità CD da comandare.

Alle righe 11-16 sono definite varie costanti API che saranno utilizzate più avanti dalla funzione *mciSendCommand*, dichiarata alla riga 18.

Dichiariamo anche una variabile di nome openParams del tipo **MCI_OPEN_PARAMS** definito in precedenza.

```

22. Private Sub Form_Load()
23.     openParams.wDeviceID = 0
24.     openParams.lpstrDeviceType = "cdaudio"
25.     Call mciSendCommand(0, MCI_OPEN, MCI_OPEN_TYPE, openParams)
26. End Sub
27.

```

```
28. Private Sub Form_Unload(Cancel As Integer)
29.     Call mciSendCommand(openParams.wDeviceID, MCI_CLOSE, 0, ByVal 0&)
30. End Sub
31.
```

Prima di eseguire un comando su un'unità MCI è fondamentale aprire un [handle](#) a tale unità utilizzando la solita funzione *mciSendCommand*; al termine del suo utilizzo sarà opportuno chiudere lo stesso.

Alla riga 23 viene specificato l'ID della periferica da utilizzare il cui tipo è definito alla riga successiva (*cdaudio*). Forniti i parametri necessari sarà possibile richiedere l'apertura dell'handle alla periferica tramite la funzione API *mciSendCommand* specificando come comando ([messaggio](#)) il valore della costante **MCI_OPEN**. I parametri aggiuntivi da fornire riguardano il tipo di argomento passato, nel nostro caso una struttura di tipo **MCI_OPEN_PARMS**, identificata dalla costante API **MCI_OPEN_TYPE** ed un [puntatore](#) (infatti l'argomento è [dichiarato con ByRef](#)) alla variabile **openParams**.

Non appena effettuata l'apertura del dispositivo MCI il [membro](#) *wDeviceID* riceverà il vero ID della periferica, precedentemente inizializzato a 0.

In maniera analoga, alla chiusura del form, sarà necessario chiudere l'handle aperto utilizzando sempre la funzione *mciSendCommand* passando come parametri l'ID del dispositivo aperto e come comando (messaggio) il valore della costante **MCI_CLOSE**. A tutti gli altri argomenti può essere passato un valore di zero.

```
32. Private Sub cmdApriCD_Click()
33.     Call mciSendCommand(openParams.wDeviceID, MCI_SET, MCI_SET_DOOR_OPEN, ByVal 0&)
34. End Sub
35.
36. Private Sub cmdChiudiCD_Click()
37.     Call mciSendCommand(openParams.wDeviceID, MCI_SET, MCI_SET_DOOR_CLOSED, ByVal
    0&)
38. End Sub
```

L'apertura e la chiusura del cassetto del lettore CD si differenziano soltanto per un parametro inviato alla funzione *mciSendCommand*. Infatti in entrambi i casi sarà necessario fornire l'ID della periferica ed il messaggio corrispondente alla costante **MCI_SET**. La differenza sta nel terzo argomento della funzione: in caso di apertura dovrà essere passata la costante **MCI_SET_DOOR_OPEN**, mentre, in caso di chiusura la costante da passare sarà **MCI_SET_DOOR_CLOSED**.

Ci penserà la funzione MCI a richiedere l'apertura o la chiusura del cassetto del CD.

A volte è incredibile!

Le funzioni MCI sono in grado di sollevare il programmatore dalla scrittura decine o centinaia di righe di codice, anche per compiere i comandi più complessi.

Un altro esempio di utilizzo delle funzioni MCI si trova nella [Richiesta per riprodurre un file MIDI all'intero di un file di risorse](#).

Un problema legato a questa soluzione, però, sta nel fatto che l'apertura del cassetto avviene in maniera [sincrona](#) ovvero il programma interrompe la sua esecuzione durante l'operazione di apertura e di chiusura del cassetto del lettore CD.

[Fibia FBI](#)

27 Gennaio 2002



[Torna all'indice degli HowTo](#)
