

# Appunti sul C/FRONT<sup>1</sup>

*by*

**“A Strange Site”**

**<http://astrangesite.altervista.org>**

**powered by**

**dott. Alessandro Strano**

## PREMESSA

Il C/FRONT è l'interfaccia di programmazione che l'ambiente di sviluppo di Navision offre a quanti vogliano realizzare delle procedure esterne per accedere al database di Navision. Si badi che il codice “interno” previsto per i “trigger” delle tabelle non viene eseguito quando si accede al database da procedure esterne e pertanto l'accesso in scrittura (modifica, inserimento ed eliminazione di dati) deve essere fatto con estrema attenzione al fine di evitare incongruenze nei dati o addirittura la corruzione del database.

Nel seguito ci si riferisce a procedure scritte in Visual Basic<sup>2</sup>, ma quanto si dirà è valido per qualsiasi linguaggio che supporti chiamate alle funzioni delle DLL.

NB: le funzioni offerte dall'interfaccia sono state scritte in linguaggio C e seguono la convenzione \_CDECL.

## PRINCIPALI FUNZIONI

NB: tutti gli handle sono di tipo Long.

**.ConnectServer(Server as String, protocollo as String)**

Si connette al database tramite server; protocollo può essere “tcp” o “netBIOS”.

**.DisconnectServer**

Si disconnette dal database tramite server.

---

<sup>1</sup> © Navision.

<sup>2</sup> © Microsoft Corp.

**.OpenTable(tab\_handle as Long, numero\_tabella as Long) As Boolean**

Apri la tabella richiesta restituendo in handle il suo gestore.

**.CloseTable(tab\_handle) As Boolean**

Chiude la tabella.

**.TableNo (nome\_tabella As String) As Long**

Restituisce il numero della tabella di cui si conosce il nome.

**.FieldNo (tab\_handle, nome\_campo As String) As Long**

Restituisce il numero del campo di cui si conosce il nome.

**.RecCount(tab\_handle) As Long**

Restituisce il numero di record presenti nella tabella o, se è stato impostato un filtro, il numero di record che soddisfano i criteri del filtro.

**.AllocRec(tab\_handle) As Long**

Crea un record-buffer per la tabella specificata; se l'operazione ha esito positivo viene restituito l'handle del record. Questa funzione deve essere richiamata prima di operare sui record della tabella. Utilizzare FreeRec(rec\_handle) per rimuovere la memoria riservata al record.

**.GetFieldData(tab\_handle, rec\_handle, numero\_campo as Long) As Variant**

Acquisisce il valore del campo del record.

**.AssignField(tab\_handle, rec\_handle, numero\_campo as Long,  
valore as Variant) As Boolean**

Assegna un valore al campo del record puntato, ma non aggiorna il record della tabella; per aggiornarlo richiamare ModifyRec o InsertRec. La chiave primaria del record deve essere univoca.

**.DeleteRec(tab\_handle, rec\_handle) As Boolean**

Cancella il record.

**.DeleteRecs(tab\_handle) As Boolean**

Cancella tutti i record. Questa funzione è influenzata dai filtri.

**.InsertRec(tab\_handle, rec\_handle) As Boolean**

Inserisce il record nella tabella.

**.InitRec(tab\_handle, rec\_handle) As Boolean**

Inizializza i campi del record-buffer. Questa funzione deve essere utilizzata prima della funzione InsertRec.

**.ModifyRec(tab\_handle, rec\_handle) As Boolean**

Modifica il record della tabella.

**.FindRec(tab\_handle, rec\_handle, criterio As String) As Boolean**

Cerca il record della tabella che soddisfa il criterio e ne restituisce l'handle in rec\_handle. La ricerca è influenzata dai filtri impostati.

Criteri:

- il primo record;
- + l'ultimo record;
- < minore di (1)
- > maggiore di (1)
- = uguale a (1) Esempio "=No"
- empty lo stesso che = "" (nessuno criterio)

(1) se si utilizza uno di questi criteri, prima di invocare la funzione occorre assegnare dei valori ai campi della chiave.

**.SetFilter(tab\_handle, numero\_campo, filtro as String) As Boolean**

Imposta un filtro sul campo della tabella.

Filtri:

- < minore di
- > maggiore di
- & AND
- | OR
- = uguale a
- \* qualunque siano i caratteri che seguono
- ? qualunque sia il carattere in questa posizione

**.NextRec(tab\_handle, rec\_handle, step as Integer) As Integer**

Restituisce il numero di record che seguono - se step > 0 - o che precedono - se step < 0 - il record corrente e, salvo che non vi siano altri record, rec\_handle è modificato in accordo allo step (cioè viene restituito l'handle del record correntemente puntato). E' influenzata dall'impostazione di filtri.

**.OpenDataBase (nome\_db As String, cache\_size As Long, usa\_commit\_cache As Boolean) As Boolean**

Apri il database in modalità locale.

**.CloseDataBase (nome\_db) As Boolean**

Chiude il database che era stato aperto in modalità locale.

**.ReleaseAllObjects**

Rilascia tutte le risorse: tabelle, record, ma non le società ed il database.

## **.BWT/.EWT**

.BWT

'Begins Write Transaction

....

.EWT

'Ends Write Transaction

Il sistema si occuperà di evitare incoerenze a causa delle operazioni incluse tra la BWT e la EWT: difatti, qualora una di esse fallisca nessuna delle operazioni avrà effetto sul database.

## **.HideError = True | False**

Permette di attivare o disattivare l'intercettazione degli errori.

## **.AllowRecordNotFound, .AlowTableNotFound, .AlowRecordExists, .AllowKeyNotFound**

Indicano alle funzioni che operano sulle tabelle/record di proseguire l'esecuzione nel caso in cui si verifichi l'errore indicato (in altri termini le funzioni restituiranno il valore FALSE anziché richiamare il gestore di eccezioni).

## **ESEMPIO**

Si noti che i passi da seguire per l'apertura della società sono gli stessi che si effettuano quando si accede al database direttamente dal gestionale.

NB: @ andrà sostituito con il nome con cui nella form è stato referenziato l'oggetto c/front\_ocx\_object.

```
Dim Login      As String, lngTabSocHandle as long
Dim Password   As String, lngRecSocHandle as long
Dim Company    As String, ValoreCampo as Variant
Dim Server     As String
```

```
Login  = "ALESSANDRO"
Password = "PASSWORD"
Server = "169.254.0.1"      'indirizzo del server
Company = "STRANGE"        'nome della società da aprire
```

```
@.HideError
'attiva l'intercettazione degli errori.
```

@.LoadLicenseFile(percorso\_file\_licenza)

'legge la licenza

@.ConnectServer(Server, "tcp")

'Si connette al database tramite server; "tcp" è il protocollo utilizzato

IF @.Login(Login, Password) then

'Se il login ha esito positivo viene aperta la società

@.OpenCompany(Company)

ELSE

'non è stato possibile effettuare il login

...

END IF

'-----

' Lettura del campo n° 1

' del primo record della tabella n° 50000

'-----

IF @.OpenTable(lngTabSocHandle, 50000) Then

lngRecSocHandle = @.AllowRec(lngTabSocHandle)

@.AllowRecordNotFound

IF @.FindRec(lngTabSocHandle, lngRecSocHandle) Then

ValoreCampo = @.GetFieldData(lngTabSoc, lngRecSocHandle, 1)

ELSE

'la tabella è vuota

...

END IF

@.CloseTable(lngTabSocHandle)

ELSE

'la tabella non esiste

...

END IF

'-----

...

@.ReleaseAllObjects

@.CloseCompany(Company) 'chiude la società

@.DisconnectServer 'si disconnette