Informazioni aggiuntive sulle finestre di Windows

http://www.vbsimple.net/info/info_17.htm

Il nome *Windows* significa letteralmente ***"finestre"*** ed il pensiero ricade subito su quelle aree che contengono i controlli e formano l'interfaccia grafica di un programma. Ma uno studio più approfondito del sistema rivela che le finestre non sono soltanto quelle grandi aree contenitore.

Una banale applicazione in ambiente Windows si compone di una finestra contenitore chiamata **Application Window** contenente al suo interno una barra del titolo, i pulsanti di sistema, la barra dei menu e l'area inferiore chiamata **Client Area** o **Client Window**. Pertanto un'applicazione non si compone di una sola finestra ma di tante finestre, contenute all'interno di una grande *Application Window*. All'interno della *Client Area* sono contenute tante finestrelle che consistono in caselle di testo, pulsanti di comando, pulsanti di opzione, cornici, definite generalmente come *controlli*.

Pertanto quando si parla generalmente di finestre si indicano le Application Window o le finestre di dialogo, ma quando si fa riferimento a funzioni API, il concetto di finestra si allarga: in linea generale indica qualsiasi area, identificata da un handle unico, in grado di ricevere i messaggi del sistema e gestirli tramite una Window Procedure.

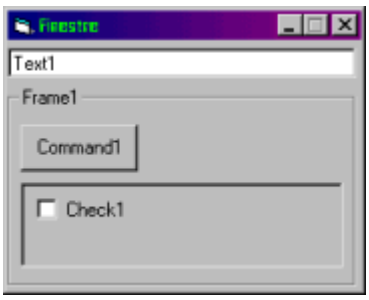
Le finestre sono organizzate gerarchicamente. All'avvio di Windows viene creata la prima finestra, il **Desktop**, la finestra Parent (genitore) di tutte le finestre; tutte le altre finestre saranno Child (figlie) della loro finestra contenitore.

Pertanto il Desktop è la prima finestra Parent e tutte le altre sono sue Child, comprese la barra delle applicazioni e la TrayBar. A loro volta, tutte le *Application Window* sono finestre Parent dei controlli in esse contenuti.

Tutte le finestre sono generalmente costituite da:

- Un handle unico detto *hWnd* (Handle Window) che identifica univocamente ciascuna finestra
- Una finestra Parent nella quale la finestra è contenuta
- Un classname che identifica la tipologia di finestra
- Un testo Caption che aggiunge una descrizione alla finestra
- Una serie di attributi e stili che ne definisce forma, posizione e caratteristiche
- Una Window Procedure che gestisce i messaggi inviati alla finestra

In linea generale i processi generano threads che a loro volta generano finestre organizzate gerarchicamente. Utilizzando *Microsoft Spy++*, fornito con Visual Basic, è possibile avere una chiara idea dell'organizzazione gerarchica e della struttura di ogni finestra. Vedi anche le Informazioni aggiuntive su Processi e Threads.



Prendiamo come esempio una semplice applicazione come quella mostrata qui a fianco: un form contenente al suo interno una **TextBox** ed un **Frame**. Il Frame conterrà al suo interno un **CommandButton** ed una **PictureBox** che a sua volta contiene un **CheckBox**.

Analizzandola con il *Microsoft Spy++* otterremmo il risultato mostrato nella Figura 2. Naturalmente gli handle delle finestre, dei processi e dei threads sono rigorosamente casuali e variano da un'esecuzione all'altra.

Nel nostro esempio l'esecuzione dell'applicazione Progetto1.EXE composto da un solo form con i predetti controlli, genera la creazione di due finestre principali: **00000EDC** di classe *ThunderRT5Form* e **00000ED8** di classe *ThunderRT5Main*. La seconda finestra sarà creata automaticamente dal compilatore ma non verrà mai mostrata. La seconda finestra rappresenta il form con i controlli posizionati.

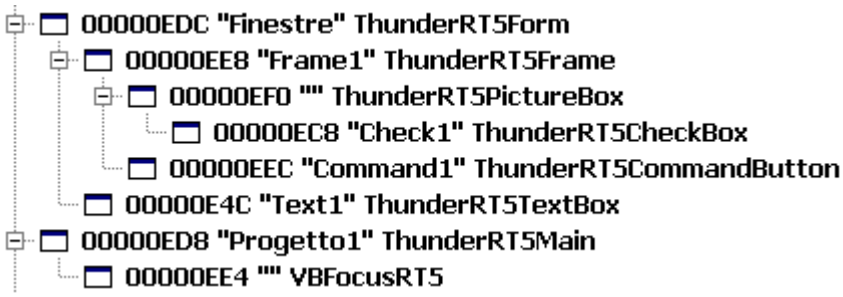


Figura 2

La finestra si presenta di classe ThunderRT5Form e contiene al suo interno altre finestre, rappresentate logicamente e graficamente in forma di gerarchia, come nella Figura 2.

Il Form (**00000EDC**) contiene al suo interno un Frame (**00000EE8**) ed una TextBox (**00000E4C**); il Frame contiene invece il CommandButton (**00000EEC**) e la PictureBox (**00000EF0**), la quale contiene a sua volta un CheckBox (**00000EC8**).

Prendiamo ad esempio l'ultima finestra, che presenta le seguenti caratteristiche:

- *hWnd* = 0x00000EC8
- *Parent* = 0x00000EF0, che identifica il controllo Contenitore PictureBox
- *ClassName* = ThunderRT5CheckBox
- *Caption* = "Check1"
- *Attributi di posizione e grandezza* = (21, 120)-(78, 133), 57x13
- *Stili* = WS_CHILD, WS_VISIBLE, etc...
- *Window Procedure* = Indirizzo 0x0F01D3F3
- *Thread ID* = Indirizzo 0xFFFA268D
- *Process ID* = Indirizzo 0xFFFA8745

Tutti questi dati sono comuni a tutte le finestre eccetto il Desktop che non possiede una finestra Parent, essendo il genitore di tutte le finestre. Naturalmente alcuni valori dei campi differiscono da finestra a finestra e da un'esecuzione all'altra, poiché è il sistema operativo ad assegnare gli handle alla finestra Child ed a quella Parent, nonché gli indirizzi del Thread e del Processo.

Da questo semplice esperimento di analisi è facile comprendere perché abbiamo chiamato chiamato Thunder la sezione dedicata ai controlli intrinseci di Visual Basic. Nello specifico le loro classi prendono il suffisso *ThunderRT5* relativamente alla versione 5 di VB.

Esistono decine di migliaia di messaggi già definiti a livello di sistema, alcuni generali, altri specifici per un tipo particolare di finestra, ad esempio i messaggi ***LB_*** riguardano esclusivamente i controlli ListBox, ovvero le finestre di classe *ThunderRT5ListBox*. Ciascuna Window Procedure è e deve essere progettata per elaborare soltanto i messaggi interessati e lasciare al sistema operativo la gestione di tutti gli altri messaggi, per eseguire ad esempio l'aggiornamento del video o lo spostamento del cursore.

Il *Microsoft Spy++* consente di eseguire tantissime verifiche del genere ed è in grado di svelare con semplicità molte delle difficoltà celate dietro questo intricato labirinto di oggetti.
Microsoft Spy++ si trova nel CD di Visual Basic 5 Professional all'interno della cartella TOOLS\SPY.

Fibia FBI
10 Ottobre 2002
