

Microsoft Visual FoxPro ®

“Guided Tour”

© Microsoft Corporations

Traduzione di Baldarelli Gian-Carlo

webmaster@foxitaly.com

www.foxitaly.com

Visual FoxPro 6.0 – Tour guidato

Avvio

Installazione dei files della guida

1. Create una directory col nome vfp6tour nel vostro disco C
2. Scaricate da <http://www.microsoft.com/italy/visualtools/Tour/vf/demo/vfp6tour.exe>
3. Copiate il file [vfp6tour.exe](#) nella vostra directory `c:\vfp6tour`.
4. Cliccate su **vfp6tour.exe** in modo da espandere la guida nella directory creata.

Installazione del database della guida

Per semplicità, è stato fornito un database Microsoft Access 97. Con Visual FoxPro potete creare applicazioni che lavorano con qualsiasi sorgente di dati ODBC o OLE. I server di produzione sono solitamente client/server DBMS come ad esempio Microsoft SQL Server. Per eseguire delle prove può essere utilizzato Microsoft Access i cui dati possono essere raggiunti tramite ODBC.

1. Aprite il pannello di controllo da menu start.
2. Fate doppio click sull'icona **ODBC** per aprire l'utilità di configurazione ODBC.
3. Cliccate sulla scheda **System DSN**, e quindi cliccate su **Add**.
4. Fate doppio click sulla voce **Microsoft Access Driver (*.mdb)**.
5. Chiamate la sorgente di dati *Fitch_Mather*. Cliccate **Select**.
6. Nella finestra di dialogo **Select Database** localizzate `c:\vfp6tour\data\fitch_mather.mdb` e quindi cliccate su **OK**.
7. Cliccate **OK** per aggiungere il nuovo System DNS. Cliccate di nuovo **OK** per chiudere l'utilità di configurazione.

Sviluppo dell'applicazione

La programmazione ad oggetti consente un elevato grado di riutilizzo e di consistenza nelle applicazioni. Visual FoxPro 6.0 riduce lo sforzo di apprendimento con la programmazione ad oggetti e semplifica lo sviluppo fornendo un set di librerie di classi preconfezionate pronte all'uso che i programmatori possono utilizzare per inserire le funzioni di base alle loro applicazioni. I Wizard (automatismi di programmazione) utilizzano queste classi per creare un ambiente di lavoro (framework) completamente orientato agli oggetti per costruire soluzioni database. L'Application Builder viene impiegato per aggiungere maschere (form) e moduli stampa (report) all'applicazione che utilizza questo ambiente.

Creare una applicazione utilizzando Application Wizard e Application Builder

Il primo passo per creare una applicazione è quello di utilizzare Application Wizard per creare l'ossatura del progetto basandosi su un ambiente di lavoro orientato agli oggetti.

1. Nella finestra di comando (**Command** window) scrivete `cd c:\vfp6tour`.
2. Dal menu **File** scegliete **New**.
3. Nella finestra di dialogo **New**, selezionate **Project**, e quindi cliccate su **Wizard**.

4. Nella finestra di dialogo dell' **Application Wizard** scrivete il nome dell'applicazione *vfp6tour*. Il file del progetto verrà completato automaticamente. Assicuratevi che **Create Project project dDirectory sStructure** sia selezionato. Cliccate su **OK**.

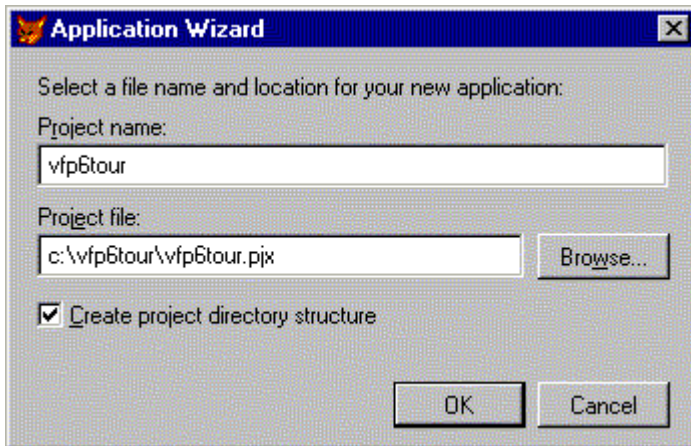


Figure 1: Uso dell' Application Wizard per creare un nuovo progetto.

Viene quindi creata l'applicazione basandosi sul nuovo scheletro di base. Potrebbe essere necessario attendere alcuni attimi. Dopo che è stata costruita l'applicazione, apparirà un Help con maggiori informazioni sul progetto e appare automaticamente l'Application Builder.

1. Chiudete la finestra di aiuto **Help**. (Questa finestra è denominata *MSDN Library Visual Studio 6.0*).

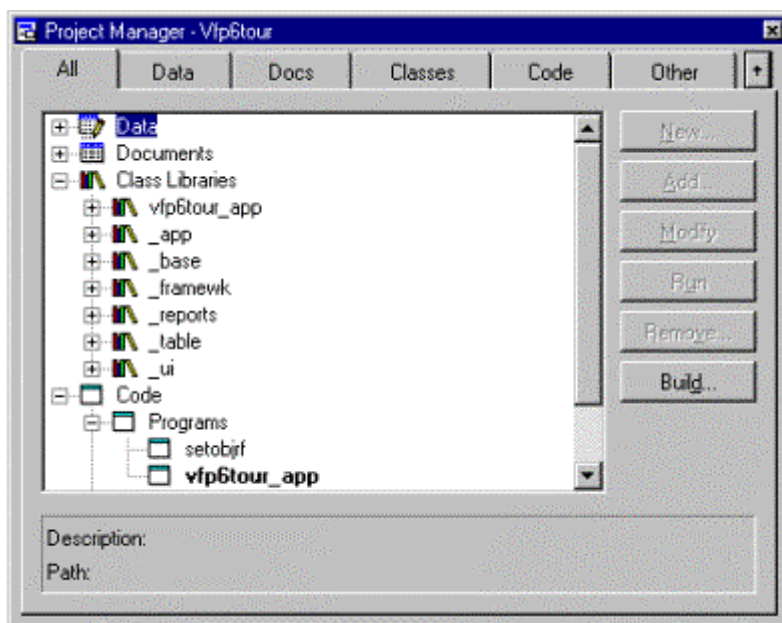


Figure 2: Il Project Manager con le classi che costituiscono la "cornice" dell'applicazione.

Uso dell' Application Builder per aggiungere Forms e Reports

Il passo successivo è quello di utilizzare l'Application Builder per aggiungere dati, maschere e moduli stampa al progetto.

1. Nella scheda **General** dell'Application Builder scrivete *Visual FoxPro 6.0 Guided Tour* come nome del progetto.

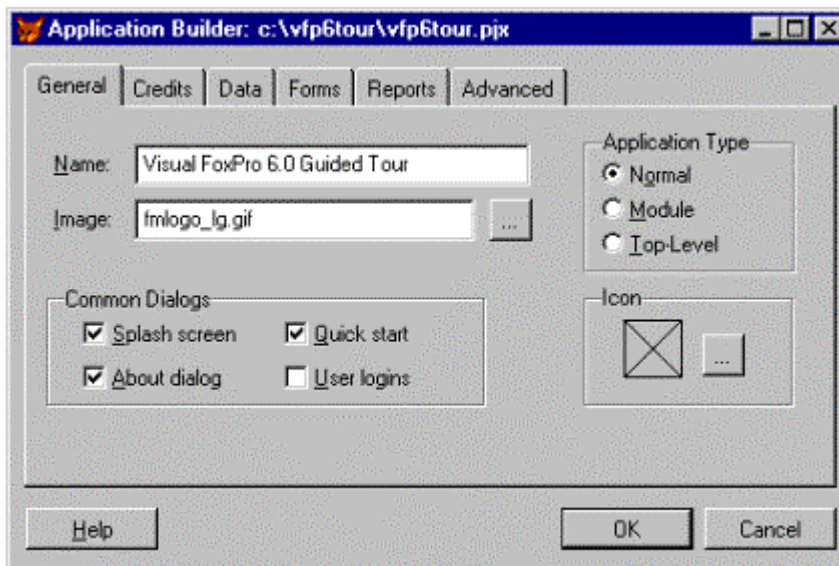


Figure 3: Uso dell'Application Builder per impostare delle opzioni specifiche.

2. Inserite *fmlogo_lg.gif* nella casella di testo **Image**. L'immagine verrà utilizzata come schermata di avvio (splash screen) quando viene avviata l'applicazione.
3. Nella scheda **Credits** scrivete il vostro nome, nome dell'azienda ed altre informazioni.
4. Nella scheda **Advanced** identificate la directory dove vengono salvati I dati selezionando "Default Data Directory" cliccando sul tasto con I puntini e nella finestra di dialogo **Select Directory** selezionate **c:\vfp6tour\data**. Cliccate su **Select**.
5. Nella scheda **Data** potete ora selezionare le tabelle e avere la creazione automatica di maschere e moduli di stampa (form & report). Cliccate **Select**. Nella finestra di dialogo **Open**, selezionate **contacts.dbf**, e cliccate su **OK** per ritornare alla scheda **Data**. (Potete anche effettuare un doppio click sul nome della tabella.) Ripetete questo passo per **departments.dbf** e **employees.dbf** nella stessa directory.

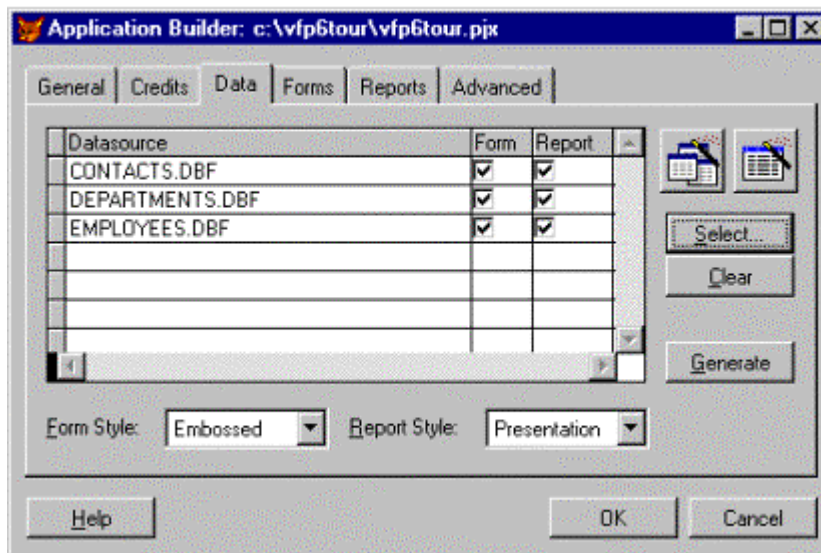


Figure 4: Create e aggiunte maschere e moduli stampa all'applicazione con l'Application Builder.

6. Potete creare automaticamente le maschere e / o I moduli di stampa per ogni tabella selezionata. Cliccate su **Generate** per richiamare il Form e/o Report wizards per ogni tabella selezionata. Nella scheda **Forms** potete vedere le maschere che sono state

aggiunte. Nella scheda **Report** potete vedere I moduli che sono stati aggiunti. Cliccate su **OK** per chiudere l'Application Builder.

7. Nella finestra del Project Manager, potete vedere le nuove maschere e I nuovi moduli aggiunti al progetto dall' Application Builder.
8. Espandete la lista delle **Class Libraries** nel Project Manager e guardate le librerie di classi nel progetto. Queste classi rappresentano l'intelaiatura dell'applicazione (Application framework).

Costruzione ed avvio dell'Applicazione

Ora che il progetto contiene dati, form, report, menu e così via sui può costruire l'applicazione.

1. Per costruire l'applicazione, scegliete **Build** dal Project Manager. Nella finestra di dialogo **Build Options** cliccate **Build Application**. Verificate che **Recompile All Files** sia selezionato e quindi selezionate **Run After Build**. Salvate l'applicazione col nome *vfp6tour.app*.

Dopo che l'applicazione è stata costruita verrà avviata automaticamente. Come prima cosa vedrete lo splash screen con le informazioni che avete inserito nelle schede General e Credits dell'Application Builder. Dopo che lo splash screen è sparito sarete in grado di aprire una delle maschere o dei moduli di stampa che avete creato.

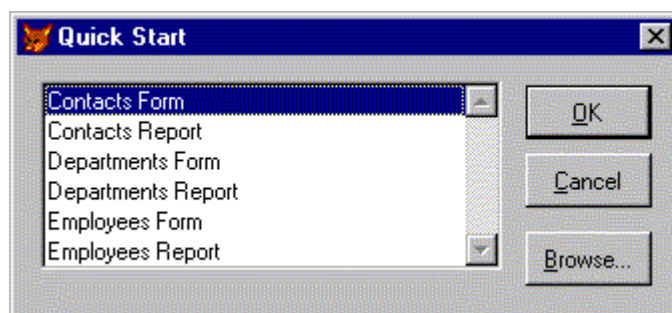


Figure 5: Utilizzate la finestra Quick Start per selezionare un form o report.

2. Per visualizzare I contatti, scegliete la maschera **Contacts Form** nella finestra **Quick Start**. Potete utilizzare i tasti di navigazione della barra degli strumenti principale per scorrere i contatti. Potete anche aprire più di un contatto alla volta. Scegliete **Open** dal menu **File** (oppure cliccate su Open nella barra degli strumenti), e cliccate di nuovo su **Contact Form**.
3. Cambiate alcuni dati nella maschera dei contatti. Chiudete la maschera cliccando sulla x della barra dei titoli della maschera. Comparirà un messaggio che vi richiede di salvare le modifiche. Cliccate su **YES**.
4. Uscite dall'applicazione scegliendo **Exit** dal menu **File**.

Usare le classi in dotazione a Visual FoxPro per aggiungere funzionalità ordinarie

Visual FoxPro 6.0 viene distribuito con delle classi di base (Visual FoxPro Foundation Classes), oltre 100 classi preconfezionate e riutilizzabili che i programmatori possono aggiungere facilmente alle loro applicazioni. Ora creerete una maschera e utilizzerete alcune di queste classi per aggiungere delle funzionalità ordinarie.

1. Nella scheda **Documents**, selezionate **Forms** nel **Project Manger**, e cliccate su **New**. Quindi cliccate **New Form**.
2. Ridimensionate la maschera in modo che riempia la finestra del Form Designer.

3. Cliccate con il destro del mouse sulla maschera e scegliete **Data Environment** dal menu scorciatoia che comparirà.
4. Selezionate **product_type**, e cliccate su **Add**. (Potete anche effettuare un doppio click su **product_type**.) Questo aggiunge la tabella all'ambiente dati delle maschere. Fate la stessa cosa per la tabella prodotti. Notate la linea fra le due tabelle. Esiste una relazione uno-a-molti fra la la tabella "prodotti" e quella "tipi prodotti".

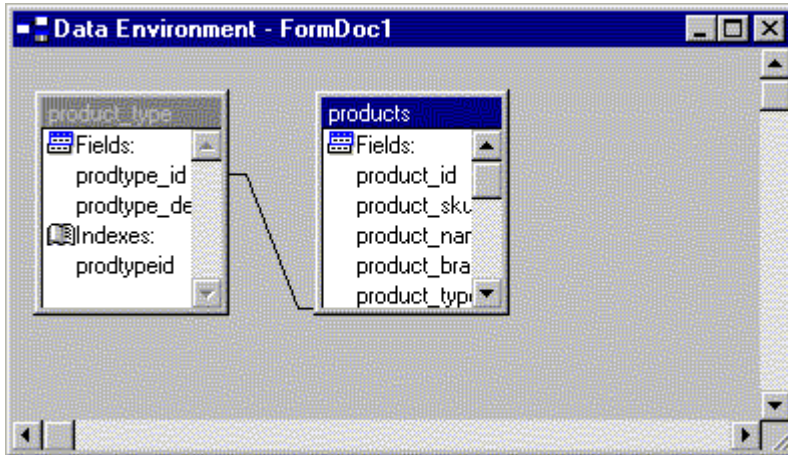


Figure 6: L'ambiente dati delle maschere "Form's data environment" mostra quali tabelle saranno utilizzate nella maschera.

5. Cliccate su **Close**.
6. Nella tabella product_type selezionate I campi **prodtype_id** e **prodtype_desc** premendo il tasto CTRL e cliccando sul campo. Mantenete premuto il tasto destro del mouse e trascinate i due campi in cima alla maschera. Dopodiché rilasciate il tasto, scegliete **Create Multiple Controls Here** dal menu scorciatoia.
7. Ora trascinate la tabella products table dall'ambiente dati verso l'area della maschera al di sotto dei due controlli appena creati. Questo creerà una griglia per mostrare più record alla volta.
8. Chiudete la finestra Data Environment.
9. Ridimensionate la griglia in modo che ci sia dello spazio nella maschera al di sotto della griglia.

A questo punto potreste desiderare aggiungere dei tasti di navigazione alla maschera. Potreste creare dei tasti vostri oppure utilizzare le classi fornite in dotazione a Visual FoxPro. Il modo più semplice per utilizzare queste classi è tramite il Component Gallery, che contiene le Visual FoxPro Foundation Classes ed altri componenti personalizzati.

1. Scegliete **Component Gallery** dal menu **Tools**. Nella parte di sinistra ci sono i cataloghi e raccoglitori (cartelle). Nella parte di sinistra potete vedere i raccoglitori all'interno dei cataloghi e le classi nei raccoglitori.
2. Espandete il raccoglitore **Foundation Classes** nel pannello di sinistra, selezionate il raccoglitore **Data Navigation**. Nel pannello di destra vedrete le classi di navigazione incluse nella dotazione di Visual FoxPro (Foundation Classes).

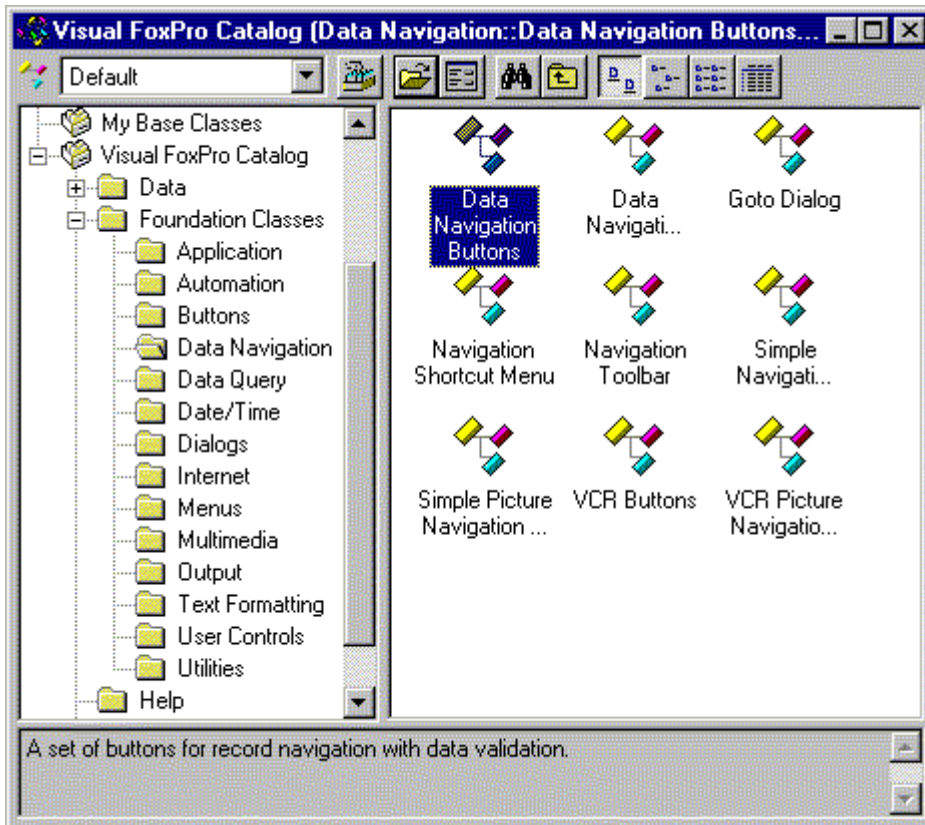


Figure 7: Utilizzate il Component Gallery per localizzare facilmente una classe e inserirla in una maschera.

3. Selezionate la classe **Data Navigation Buttons**. Trascinate la classe sulla maschera al di sotto della griglia.
4. Selezionate il raccoglitore **Buttons nel** Component Gallery, e trascinate la classe "tasto OK" nella parte alta di destra della maschera.

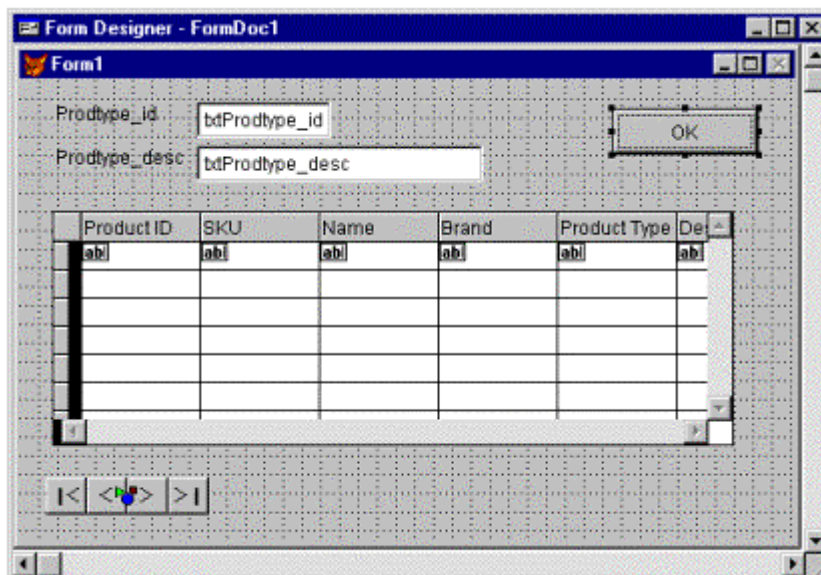


Figure 8: La maschera prodotti come appare nel Form Designer.

5. Cliccate con il destro del mouse sulla maschera e scegliete **Properties** dal menu scorciatoia per attivare la finestra delle proprietà della maschera. Assicuratevi che sia

selezionato **Form1** nella tendina di selezione nella parte alta della finestra delle proprietà.

6. Impostate la proprietà **AutoCenter** a *.T.* per centrare la maschera all'avvio.
7. Impostate la proprietà **Caption** in *Product types* cliccando la proprietà **Caption** e scrivendo *Product types* nella casella di testo in alto alla finestra delle proprietà.

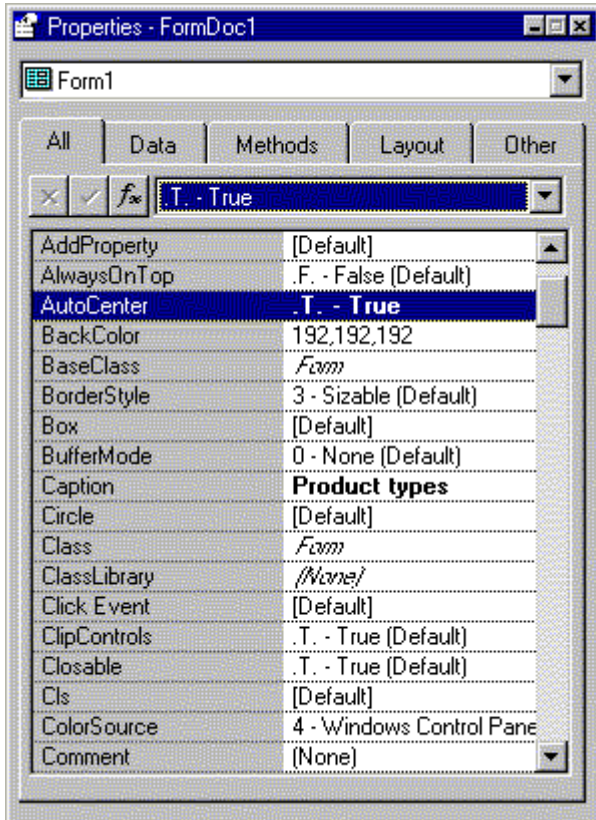


Figure 9: Utilizzate la finestra delle proprietà (Properties Window) per modificare l'aspetto della maschera e dei suoi controlli.

8. Per collaudare la maschera, cliccare su **Run** nella barra degli strumenti principale. Salvatela nella cartella *Forms* e nominatela *products*.

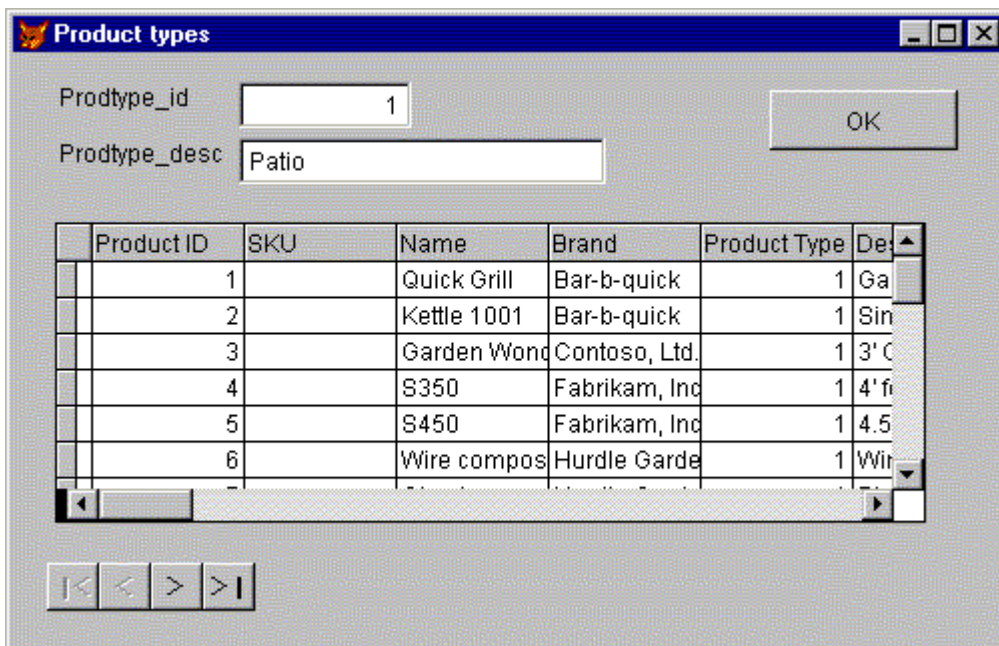


Figura 10: La Maschera prodotti come apparirà all'utente.

Ora avete una maschera pienamente funzionante per visualizzare la tipologia dei prodotti e tutti i prodotti per ogni tipologia, senza aver scritto una riga di codice. I tasti di navigazione funzionano, così come funziona il tasto **OK** per chiudere la maschera.

1. Per chiudere la maschera cliccate su **OK**.
2. Avrete notato che l'Application Builder è apparso dopo che avete salvato la maschera. Il Builder ora vi offre la possibilità di aggiungere la maschera all'applicazione "ancorandola" nella struttura dell'applicazione. Cliccate su **OK** nel Builder per eseguire quanto indicato.
3. Chiudete il Component Gallery.
4. Cliccate su **Build** nel Project Manager e ricostruite l'applicazione.

Quando l'applicazione è in esecuzione potete vedere la maschera dei prodotti nella finestra di dialogo **Quick Start**.

Nota: Se **Run After Build** non è selezionato quando ricostruite l'applicazione, potete eseguirla scrivendo `Do vfp6tour` nella Finestra di Comando.

Component Gallery

Grazie al Component Gallery si semplifica enormemente la gestione delle librerie di classi fornendo uno strumento visuale per l'organizzazione e l'utilizzo delle classi. Gli sviluppatori possono organizzare le classi in gruppi chiamate cataloghi. Questi cataloghi semplificano la classificazione, localizzazione e utilizzo delle classi durante lo sviluppo dell'applicazione.

Creare una nuova classe e aggiungerla al Component Gallery

Il Component Gallery viene utilizzato allo scopo di localizzare facilmente la classe. Ora proverete a creare una classe e a classificarla utilizzando il Gallery, quindi a localizzarla e utilizzarla.

1. Selezionate il **Products Form** nel Project Manager e cliccate su **Modify**.
2. Selezionate le due caselle di testo, le loro etichette, e la griglia creando un rettangolo di selezione con il mouse attorno agli oggetti.
3. Scegliete **Save As Class** dal menu **File**.
4. Scrivete `prodtype_products` nella casella **Name**. Questo diventa il nome della classe. Scrivete `vfp6tour` nella casella **File**. Questo sarà il nome della libreria che contiene la classe. Cliccate su **OK**

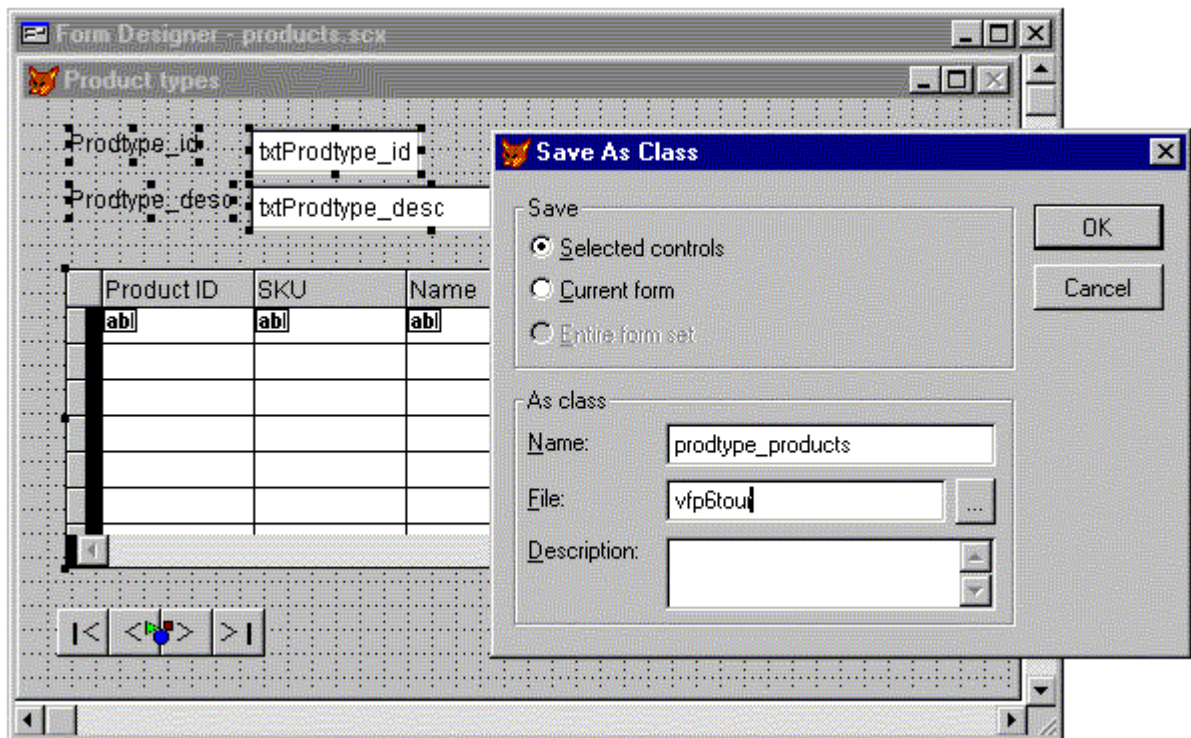


Figure 11: Memorizzate i controlli selezionati in una maschera in una classe per poterli riutilizzare successivamente.

5. Ora cliccate su **Delete** per rimuovere i controlli che avete selezionato nel passo n° 2 dalla maschera. La maschera ora contiene l'ambiente dati e i tasti di navigazione, ma mancano le informazioni sui prodotti. Avete appena creato una classe allo scopo e quindi ora la inseriamo nella Component Gallery.
6. Scegliete **Component Gallery** dal menu **Tools**.
7. Selezionate il catalogo **Favorites**.
8. Cliccate con il destro del mouse sul pannello di destra e scegliete **Class** dal menu **New Item**.
9. Nella finestra di dialogo **Open** selezionate **vfp6tour.vcx**. La classe *prodtype_products* dovrebbe essere selezionata. Cliccate su **OK**. La classe viene ora aggiunta al catalogo **Favorites** del Component Gallery.
10. Trascinate la classe nella maschera.
11. Spostate la classe nella parte alta a sinistra della maschera. Se la classe copre il tasto OK, cliccate su **Send To Back** dal menu **Format**.
12. Per avviare la maschera cliccate sul tasto **Run (!)** della barra degli strumenti principale. La maschera si comporterà come quella precedente.

Noterete che, tuttavia, ha un bordo tutto attorno. Per correggere questo modificate la classe come segue.

13. Prima, chiudete la maschera perché non potete modificare una classe che è in uso in Visual FoxPro. Poiché le maschere sono attive in fase di progettazione, le classi sono in uso quando le maschere sono aperte.
14. Nel Component Gallery, fate un doppio click sull'icona della classe **prodtype_products**. Questo lancerà il Class Designer.
15. Cliccate con il destro del mouse e scegliete **Properties** nel menu scorciatoia. Assicuratevi che sia selezionato **prodtype_products** nella tendina a discesa.
16. Modificate la proprietà **BorderWidth** in **0**.
17. Modificate la proprietà **BackStyle** in **0-Transparent** in questo modo la classe si fonderà in qualsiasi maschera su cui verrà utilizzata.

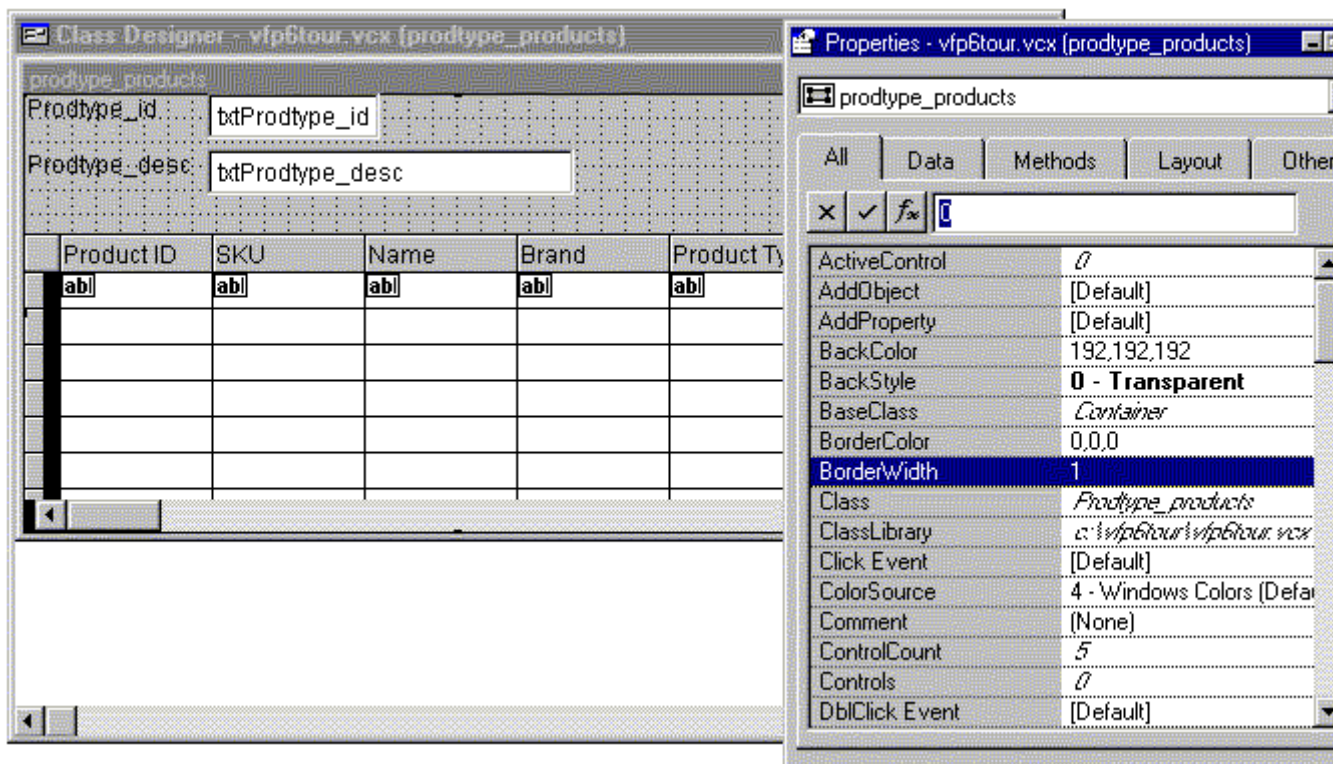


Figure 12: Uso della classe Class Designer per creare e modificare le classi.

18. Chiudete e salvate la classe.
19. Per verificare il funzionamento della maschera, selezionatela nel Project Manager e cliccate su **Run**. Notate che il bordo attorno ai controlli è sparito. Inoltre ricordatevi che la classe è riutilizzabile in altre maschere. Può inoltre essere "sottoclassata" per specifici comportamenti in altre maschere e applicazioni.
20. Infine, ricostruite l'applicazione.

Coverage Profiler

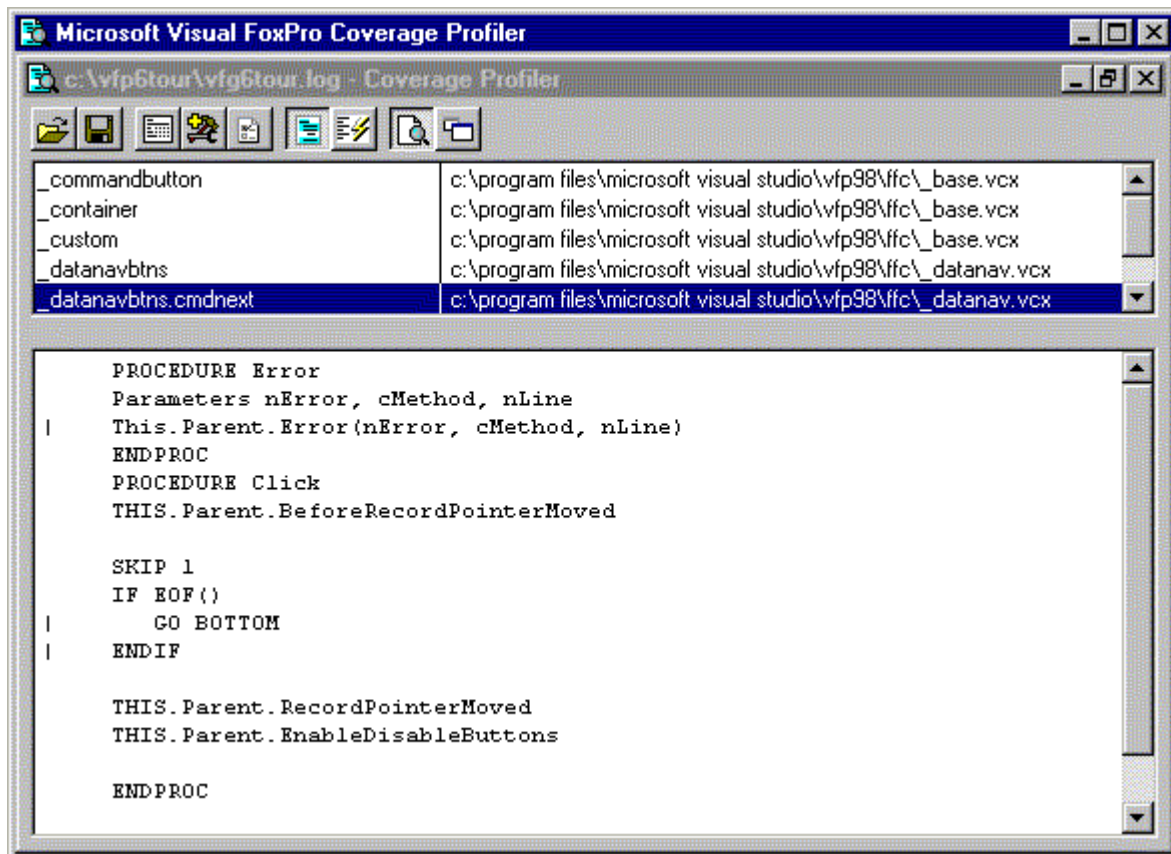
Il Coverage Profiler viene utilizzato per vedere quale linea dell'applicazione è in corso di funzionamento in un preciso istante e quanto tempo impiega per l'esecuzione. Gli sviluppatori possono verificare la validità delle loro prove dei programmi controllando quale linee di codice sono state eseguiti. Possono inoltre eseguire dei profili per evidenziare colli di bottiglia o problemi di prestazione nel codice. Gli sviluppatori possono costruire applicazioni e componenti che, non solo girano più velocemente, ma sono anche collaudati meglio.

Creare un Profile Log

1. Scegliete **Debugger** dal menu **Tools**.
2. Nella barra degli strumenti del Debugger scegliete il tasto **Toggle** coverage jogging.
3. Nella finestra di dialogo **Coverage**, selezionate **Overwrite** e inserite *vfp6tour* come nome del file log.
4. Cliccate **OK** e quindi chiudete il debugger scegliendo **Exit** dal menu **File** nella finestra del Debugger.
5. Selezionate la maschera *products* nel Project Manager e cliccate su **Run**.
6. Usate I tasti di navigazione per scorrere vari records, fatelo più volte, quindi chiudete la maschera.

Analisi del Profile Log

1. Scegliete **Coverage Profiler** dal menu **Tools**. Questo attiverà il profiler che immediatamente inizierà ad analizzare il log appena creato.
2. Ingrandite la finestra del log all'interno del Coverage Profiler.
3. Nel pannello superiore viene visualizzata la lista di tutte le procedure della maschera. Il Coverage Profiler appare in "coverage mode" di base. Di conseguenza, nella parte bassa troverete la lista di tutte le righe di codice per ogni procedura. Le righe di codice marcate con "|" non sono state eseguite, mentre quelle senza marcatura sono state eseguite.



4. Scorrete di procedura in procedura nel pannello superiore per vedere quali linee di codice è andato in esecuzione per ogni procedura.
5. Nella barra degli strumenti del Coverage Profiler's cliccate sul tasto **Profile Mode** per passare alla modalità "profile mode". In questa modalità la parte bassa del pannello mostra, per ogni procedura, quante volte ogni riga di codice è stata eseguita, quanto tempo è stato impiegato per la prima esecuzione del codice, e quanto è il tempo medio di esecuzione.

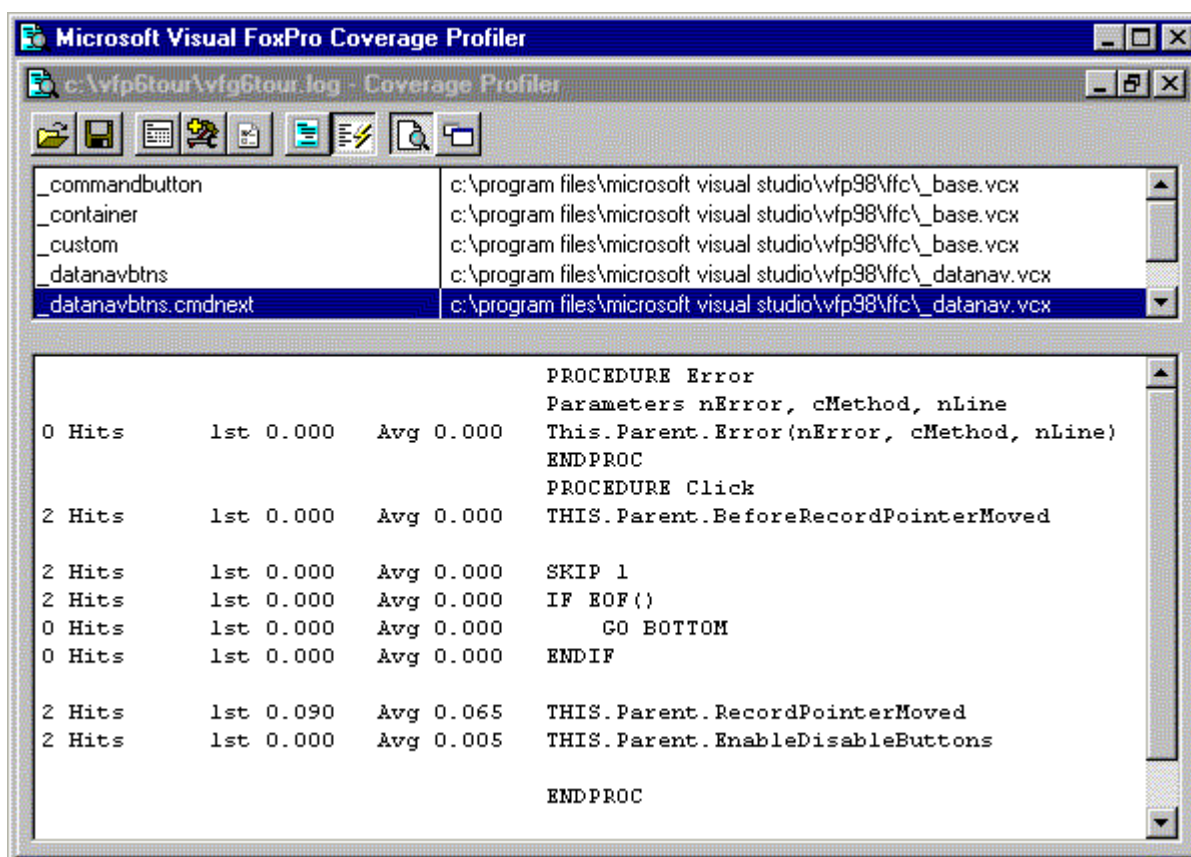


Figure 14: Uso del "profile mode" del Coverage Profiler per visualizzare quale riga di codice è stato eseguito.

6. Scorrete di procedura in procedura nel pannello superiore per visualizzare per quanto tempo diverse linee di codice sono andate in esecuzione.
7. Chiudete il Coverage Profiler. Cliccate **Cancel** nella finestra di dialogo del Coverage Profiler per disattivare il coverage logging.
8. Chiudete il progetto.

Documenti Attivi

I Documenti attivi forniscono un notevole passo dalla pura applicazione client in Visual FoxPro ad una applicazione che utilizza una interfaccia basata su HTML. Essi forniscono un modo semplice per migrare applicazioni verso il Web e forniscono funzionalità di Visual FoxPro al browser.

Creare un Form

Il primo passo nella creazione di un Applicazione Documento Attivo è quello di creare l'interfaccia utente. In questo esempio modificherete una maschera esistente per fornire una interfaccia utente per il Documento Attivo.

1. Scegliete **Open** dal menu **File**. Con **Project** selezionato nella lista **File of type**, scegliete **vfp6tour_adoc.pjx** e quindi **OK**.
2. Espandete la lista **Documents** nel Project Manager.
3. Espandete la lista **Forms** nel Project Manager.
4. Selezionate **employees_adoc** nella lista **Forms**, e cliccate su **Modify**.
5. Cliccate col destri nella maschera e scegliete **Properties** dal menù scorciatoia. Assicuratevi che il **Form1** sia selezionato nella lista a tendina in alto nella finestra delle proprietà.

6. Modificate la proprietà **DataSession** a **2 – Private Data Session**. Questo rende semplice l'uso della maschera a più utenti contemporaneamente quando lavorano con gli stessi dati.
7. Modificate la proprietà **TitleBar** a **0 - Off** Per rimuovere la barra dei titoli della maschera.
8. Modificate la proprietà **WindowState** a **2 – Maximized** per far sì che la maschera occupi tutto lo schermo.
9. Modificate la proprietà **ScrollBars** a **3 – Both** in modo che le barre di scorrimento siano disponibili quando la maschera non è tutta aperta.
10. Fate un doppio click sulla maschera per attivare la finestra Code.
11. Scegliete **Destroy** dalla lista **Procedure** e scrivete *Clear Events*, che chiuderà la maschera quando l'utente uscirà dall'applicazione in corso.

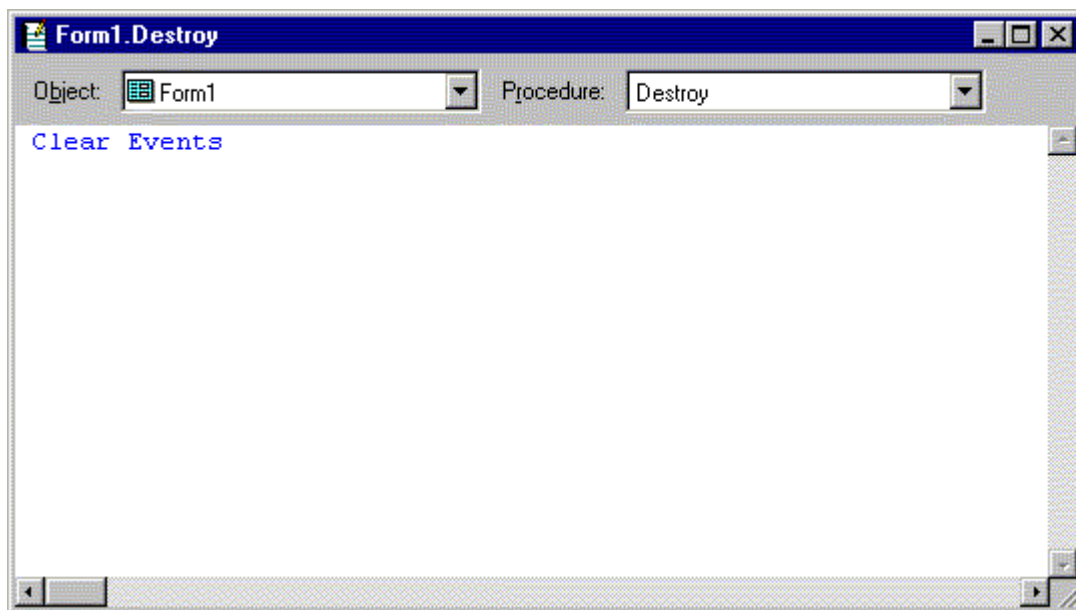


Figure 15: Usate la finestra Code per scrivere il codice che sarà eseguito quando accadono eventi.

12. Chiudete la finestra **Code**. Chiudete la maschera e salvate le modifiche.

Aggiungere una classe ActiveDoc al progetto

Una volta che è stata creata l'interfaccia potete aggiungere una classe ActiveDoc al progetto. L'oggetto ActiveDoc espone gli eventi all'ambiente gestore del Documento Attivo. Potete scrivere codice che può essere avviato quando accade l'evento. Questo è come avvierete la maschera appena creata quando l'utente naviga nel documento attivo con il browser.

1. Nel Project Manager selezionate **Class libraries** e cliccate su **New**.
2. Nella finestra di dialogo di **New Class** scrivete vfp6tour_adoc nella casella **Name**. Questo sarà il nome della nuova classe. La nuova classe si baserà sulla classe **ActiveDoc**, quindi lasciate la lista **Based On** impostato su **ActiveDoc**. Inserite vfp6tour nella casella **Store In**.

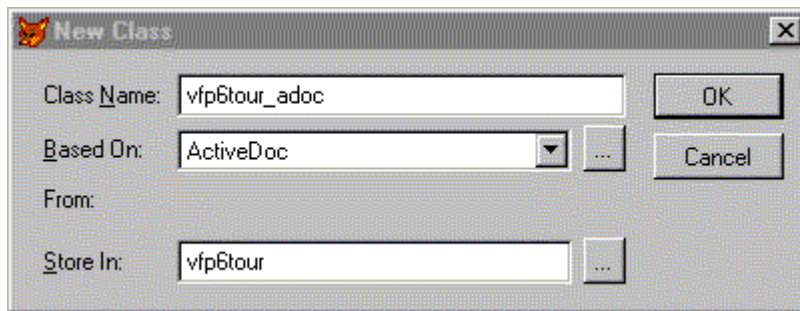


Figure 16: La finestra di dialogo *New Class* viene utilizzata quando si crea una nuova classe.

3. Cliccate su **OK** per memorizzare la classe. Questo apre il **Class Designer** in modo che possiate modificare la classe.
4. Fate doppio click sulla classe per aprire la finestra Codice (code).
5. Assicuratevi che **Procedure** sia impostato su **Run** e aggiungete il codice seguente, che avvierà la maschera appena creata.

```
DO FORM employees_adoc
```

```
READ EVENTS
```

6. Chiudete la finestra Code.
7. Cliccate sulla classe e scegliete **Properties** dal menu scorciatoia. Nella finestra **Properties** modificate la proprietà **ContainerReleaseType** a **1 – Quit Visual FoxPro**. Altrimenti, quando l'utente abbandona il Documento Attivo, Visual FoxPro resterà attivato.
8. Chiudete la classe e memorizzatela.
9. Espandete la lista **Class Libraries** nel **Project Manager**, e quindi espandete la lista **vfp6tour**. Selezionate **vfp6tour_Adoc**. Cliccate con il destro del mouse su *vfp6tour_adoc* e scegliete **Set Main** dal menu scorciatoia. Questa classe è il codice che sarà avviato per primo quando sarà avviata l'applicazione.

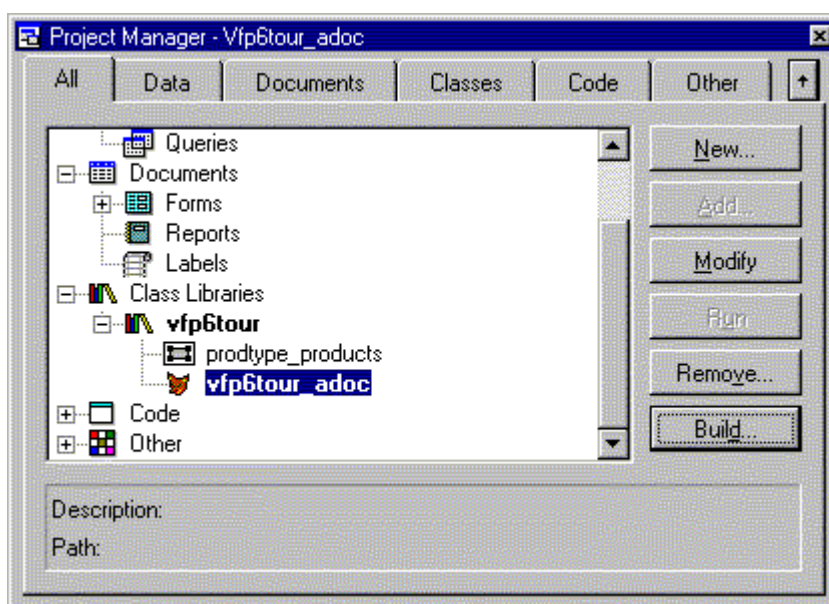


Figure 17: Il *Project Manager* come appare nel caso di applicazione *Active Document*

Costruzione e Avvio di una applicazione *Active Document*

Una applicazione Active Document viene costruita nello stesso modo di qualsiasi altra applicazioni di Visual FoxPro. Può essere all'interno di Visual FoxPro così come all'interno del browser.

1. Per costruire l'applicazione, scegliete **Build** nel Project Manager. Nella finestra di dialogo **Build Options**, selezionate **Build Application**, deselectionate **Run After Build** e quindi cliccate su **OK**. Salvate l'applicazione come *vfp6tour_adoc.app*.
2. Per lanciare il documento attivo nel browser potete scegliere **Run Active Document** dal menu **Tools**. Nella finestra di dialogo **Run Active Document** scrivete *vfp6tour_adoc.app* nella casella **Active Document** e quindi selezionate **In Browser** dal menu a tendina **Hosting**. Ciccate **Run** per lanciare la maschera.

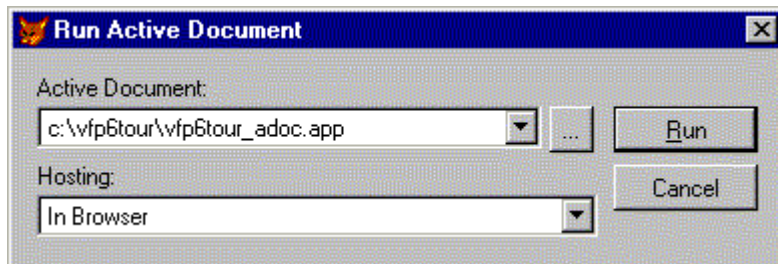


Figure 18: Utilizzate la finestra di dialogo *Run Active Document* per lanciare facilmente il documento attivo nel browser.

Questo lancia il browser e automaticamente si può navigare nel documento attivo.

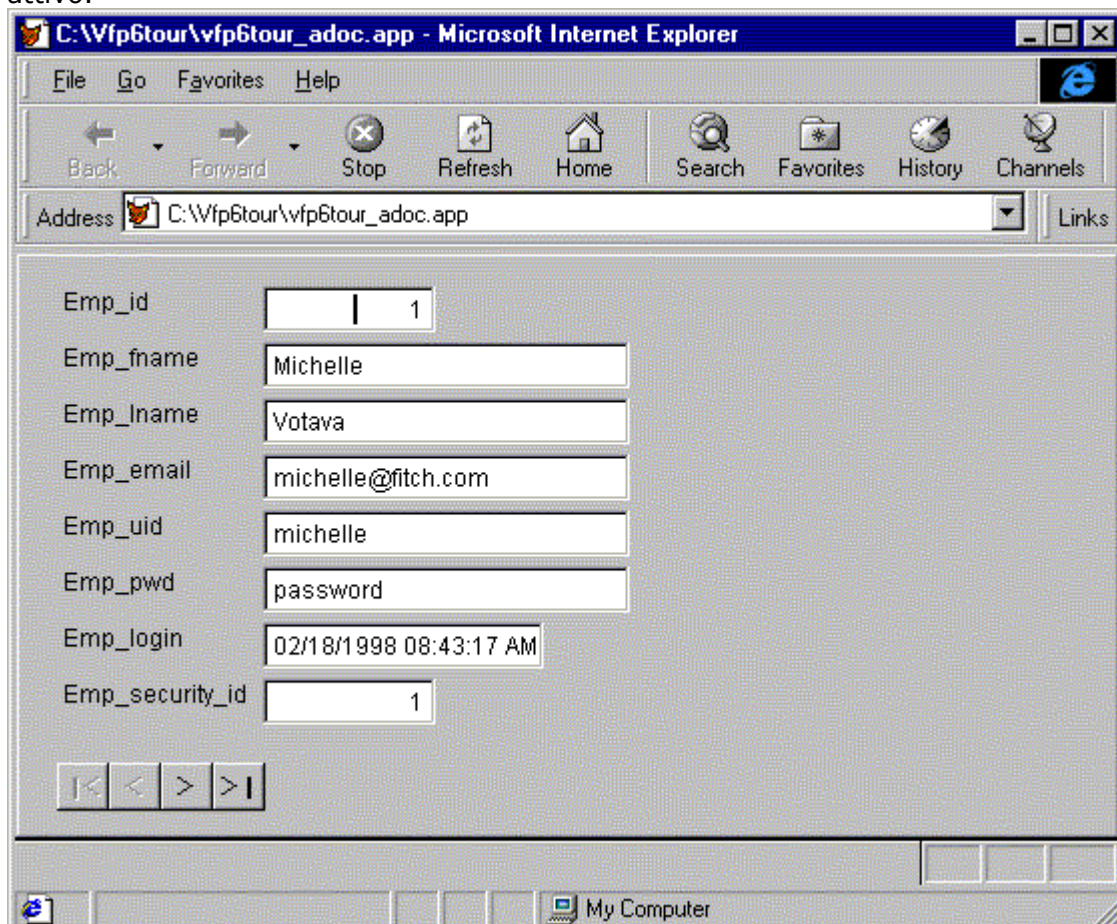


Figure 19: Il documento come appare nel browser.

3. Ora avviate la maschera in Visual FoxPro così come nel browser. Per ottenere questo scrivete *Do vfp6tour_adoc* nella finestra di comando di Visual FoxPro.
4. Scorrete sul secondo record nella maschera che è attiva in Visual FoxPro. Modificate alcune informazioni (oltre al codice ID dell'impiegato) e quindi scorrete al record successivo. Questo memorizzerà automaticamente i dati.
5. Ora passate al browser e scorrete fino al secondo record. Il nuovo record modificato sarà presente. Modificate altri dati e scorrete al record successivo in modo da memorizzare le modifiche.
6. Ritornate a Visual FoxPro e scorrete al secondo record. Le modifiche eseguite nel browser saranno visibili.
7. In Visual FoxPro, scegliete **Close** dal menu **File** per uscire dalla maschera. (Non chiudete la maschera cliccando sulla X nell'angolo a destra in alto. Perché la maschera è ingrandita e questo chiuderebbe Visual FoxPro)
8. Chiudete il browser.

Avete appena visto come trasformare una maschera di Visual FoxPro in un Documento Attivo che può essere ospitato in Internet Explorer e integrato facilmente in una applicazione web.

Componenti COM

Con il suo orientamento agli oggetti, il suo linguaggio orientato ai dati e I suoi strumenti quali le viste remote che consentono di integrare facilmente dati remoti, Visual FoxPro è uno strumento ideale per la costruzione di oggetti COM fortemente orientati ai dati. Questi componenti possono essere richiamati da una vastità di applicazioni client e possono dialogare sia con dati FoxPro e database SQL, utilizzando sia ODBC o ActiveX Data Objects (ADO).

La costruzione dei componenti consente il riutilizzo del codice esistente. Un componente COM di Visual FoxPro che lavora con i dati può utilizzare lo stesso codice di una applicazione stand-alone di Visual FoxPro che lavora con gli stessi dati. Applicazioni FoxPro non devono essere completamente riscritta per essere migrate verso applicazioni basate su componenti.

Creare e Costruire un componente COM

Nel primo passo di questo esercizio costruirete un oggetto COM che utilizza un driver ODBC per accedere ai dati di un database Microsoft Access. Il componente prende come parametro il codice ID di un prodotto e il codice ID di un negozio e restituisce il numero di unità in inventario in quel negozio.

1. Scegliete **New** dal menu **File**. Con **Project** selezionato cliccate su **New File**. Nominare il progetto *vfp6tourcom*.
2. Selezionate **Class Libraries** nel Project Manager e cliccate su **New**.
3. Nella finestra di dialogo **New Class** scrivete *inventoryclass* come nome della nuova classe.
4. Scegliete **Custom** nella lista a tendina **Based On**.
5. Scrivete *comclasses* come nome della libreria di classe. Questa è la libreria dove verrà memorizzata la classe.

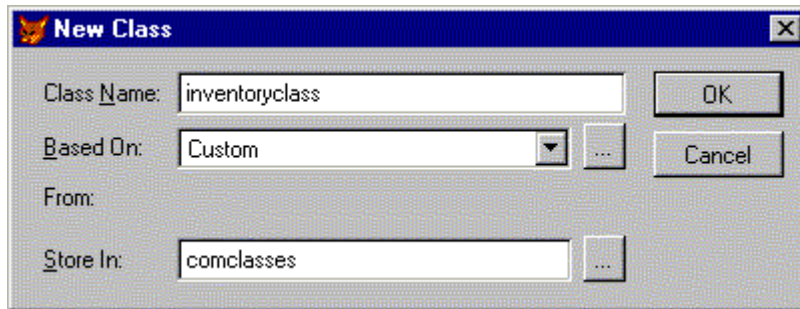


Figure 20: Il primo passo nella creazione di un componente COM è quello di creare una nuova classe.

6. Cliccate su **OK**. Questo avvierà il **Class Designer**.
7. Questa classe dovrà avere un metodo per ottenere le informazioni di inventario. Per aggiungere il metodo, scegliete **New Method** dal menu **Class**.

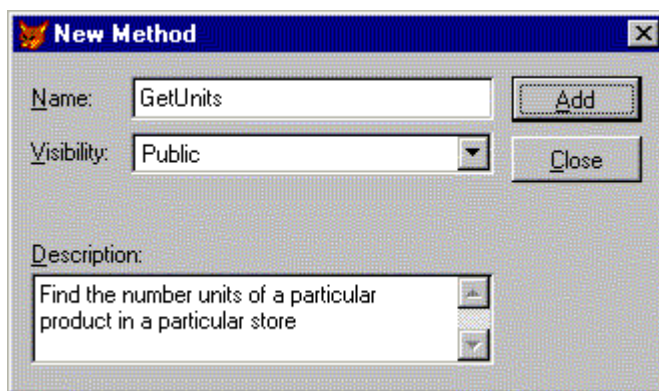


Figure 21: Il metodo della classe diventa il metodo del componente COM.

8. Scrivete *GetUnits* come nome per il metodo. Scrivete " *Find the number units of a particular product in a particular store*" (*trova il numero di unità di un prodotto in un particolare negozio*) come descrizione.
9. Scegliete **Add** per aggiungere un metodo e quindi cliccare **Close** per chiudere la finestra di dialogo.
10. Fate un doppio click sulla classe per attivare la finestra del Codice. Scegliete **GetUnits from the Procedure** dalla lista a tendina. Scrivete il seguente codice nel metodo *GetUnits*.

```
Lparameters nProductID, nStoreID
nHandle = SQLConnect("Fitch_Mather")
cSQL = "Select units From inventory " + ;
"Where product_id = " + Transform(nProductID) + "" + ;
" And store_id = " + Transform(nStoreID)
nResult = SQLExec(nHandle, cSQL, "cunits")
SQLDisconnect(nHandle)
Return cunits.units
```

Questo codice vi connette inizialmente ad una sorgente dati ODBC che punta ad un database Access. Il codice costruisce poi una dichiarazione SQL per cercare il numero di unità in un inventario per il prodotto specificato e per il negozio indicato. Quindi il codice esegue la query e disconnette il database. Il metodo restituisce il numero delle unità.

11. Chiudete la finestra Codice.
12. Scegliete **Class Info** dal menu **Class**. Selezionate **OLE Public**. Questo identificherà la classe come una classe che può essere utilizzata per costruire un componente COM

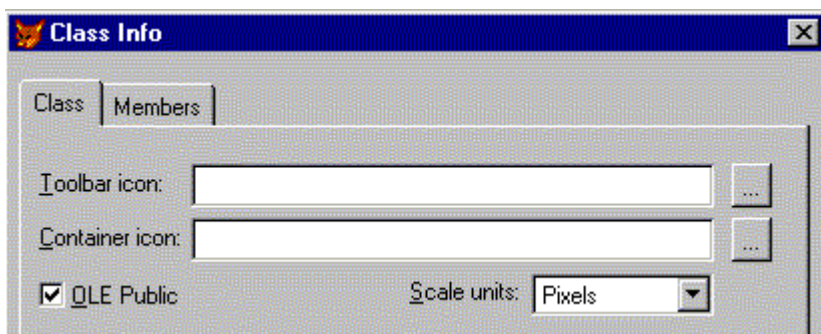


Figure 22: Definire una classe come OLE Public significa che può essere utilizzata in un componente COM.

13. Cliccate su **OK** per chiudere la finestra di dialogo **Class Info**.
14. Chiudete la classe e salvatela.
15. Nel Project Manager cliccate su **Build**.
16. Nella finestra di dialogo **Build Options**, selezionate **Build COM DLL** e quindi cliccate su **OK**. Salvate la DLL come *vfp6tourcom.dll*.

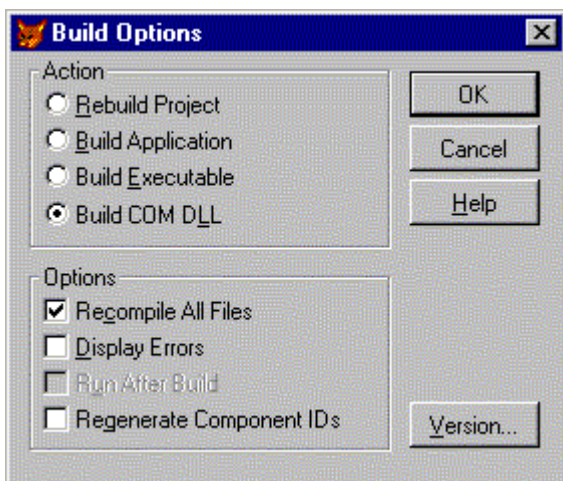


Figure 23: Utilizzate la finestra di dialogo Build per compilare la classe in un oggetto COM.

Visual FoxPro genera automaticamente il file Type Library e registra la classe COM nel sistema. Una volta che il componente COM è stato registrato, può essere richiamato da qualsiasi applicazione basata su Windows, incluso Internet Information Server.

17. Scegliete **Project Info** dal menu **Project**. Nella scheda **Servers** potete vedere elencato *inventoryclass*. Il nome completo del componente COM è quello del nome del progetto più un punto più il nome della classe server, in questo caso il nome del componente COM è *vfp6tour.inventoryclass*.

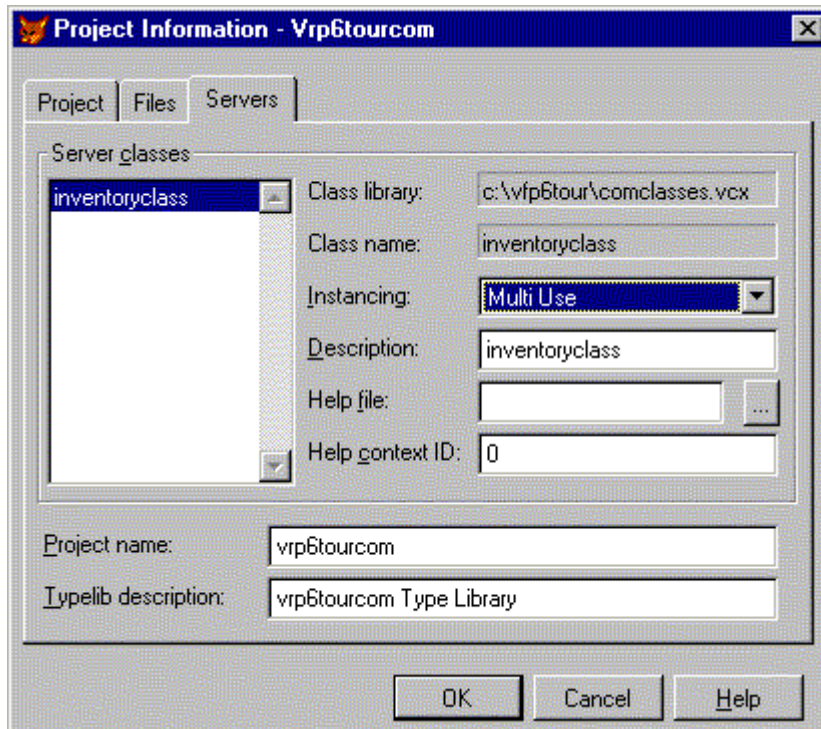


Figure 24: La finestra di dialogo Project Info mostra informazioni importanti sul componente COM.

18. Selezionate **MultiUse** nella lista a tendina **Instancing**.
19. Cliccate su **OK** per chiudere la finestra di dialogo **Project Info**. Ricompilate la DLL COM.

Collaudo del componente COM in Visual FoxPro

Nel prossimo passo effettueremo il collaudo del componente COM all'interno di Visual FoxPro.

1. Nella finestra di comando scrivete quanto segue:

```
oinventory=createobject("vfp6tourcom.inventoryclass")  
?oinventory.getunits(41,1)
```

2. Sullo schermo dovreste vedere il numero 18, che indica 18 unità del prodotto 41 nel negozio 1. Potreste avere la necessità di trascinare la finestra del Project manager da un lato per vedere il risultato sullo schermo.
3. Per rilasciare il componente COM scrivete *release oinventory* nella finestra di comando.

```
release oinventory
```

Utilizzo del componente COM in Visual Basic

Di seguito l'esempio dell'uso del componente in una maschera Visual Basic.

1. Avviate Visual Basic selezionate **Standard EXE** nella finestra di dialogo **New Project**. Cliccate su **Open**.
2. Scegliete **References** nel menu **Project**.

3. Nella finestra di dialogo **References** potete scegliere quale tipo di oggetto COM importare in Visual Basic. Cercate **vfp6tourcom Type Library** nella lista e selezionatelo. Quindi cliccate su OK. Visual Basic importerà l'oggetto COM.
4. Nella palette degli strumenti fate un doppio click sull'icona **Command Button** per aggiungere un tasto di comando alla maschera. Ora fate doppio click sull'icona **Label** per aggiungere una etichetta alla maschera. Sistematele in modo da vederli entrambi.
5. Fate doppio click sul tasto nella maschera. Quando apparirà la finestra di codice scrivete le seguenti righe di codice.

```
Dim oInventory As New vfp6tourcom.inventoryclass
label1.Caption = oInventory.GetUnits(41, 1)
```

Il codice crea la nuova variabile di tipo vfp6tour.inventoryclass e richiama il metodo, GetUnits. Fate caso al fatto che all'interno di Visual Basic, i programmatori hanno l'accesso IntelliSense a tutti gli oggetti COM di Visual FoxPro e hanno il completamento delle dichiarazioni.

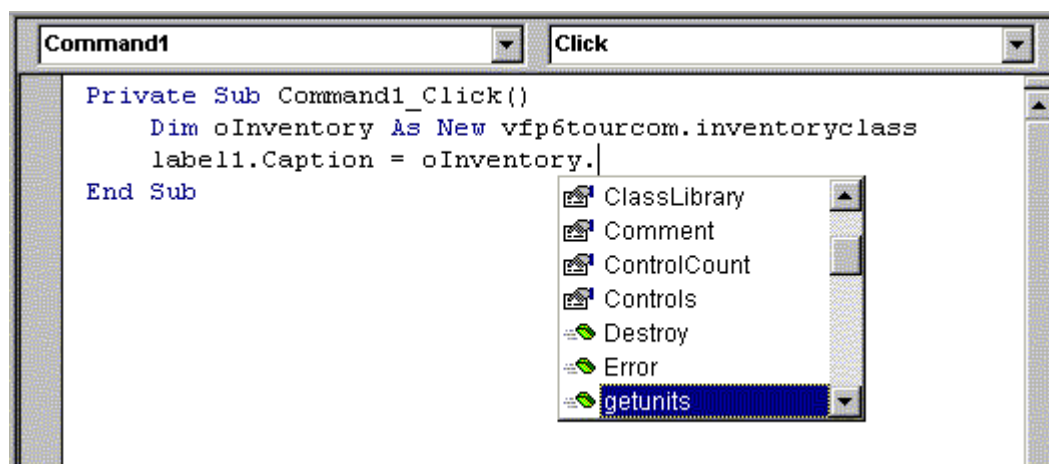


Figure 25: IntelliSense in Visual Basic is very handy when working with Visual FoxPro COM components.

6. Quando l'applicazione è avviata, la pressione del tasto imposterà l'etichetta con il valore fornito dal metodo GetUnits per il prodotto 41 e negozio 1.

E' possibile costruire facilmente oggetti COM in Visual Basic e utilizzarli in Visual FoxPro.

Uso del componente COM in una pagina Web

In questo passo useremo il componente all'interno di Internet Information Server per consentire il passaggio dei dati dall'oggetto COM a qualsiasi Browser compatibile HTML 3.2.

1. Copiate *inventory_lookup.htm* e *background.gif* nella cartella root del vostro web server.
2. Copiate *inventory.asp* nella cartella script del web server.

La pagina di ricerca dell'inventario richiede all'utente di identificare il prodotto ed il negozio tramite l'inserimento del loro codice ID. L'utente quindi clicca sul tasto Submit ed invia l'informazione al server. Il Web server avvia quindi la pagina *inventory.asp*, che richiama il metodo *GeUnits* dell'oggetto COM di Visual FoxPro passandogli le informazioni inserite dall'utente. Il componente restituisce le informazioni di inventario alla pagina ASP, la quale costruisce la pagina web che mostra le informazioni dell'inventario. Questa pagina viene poi inviata all'utente.

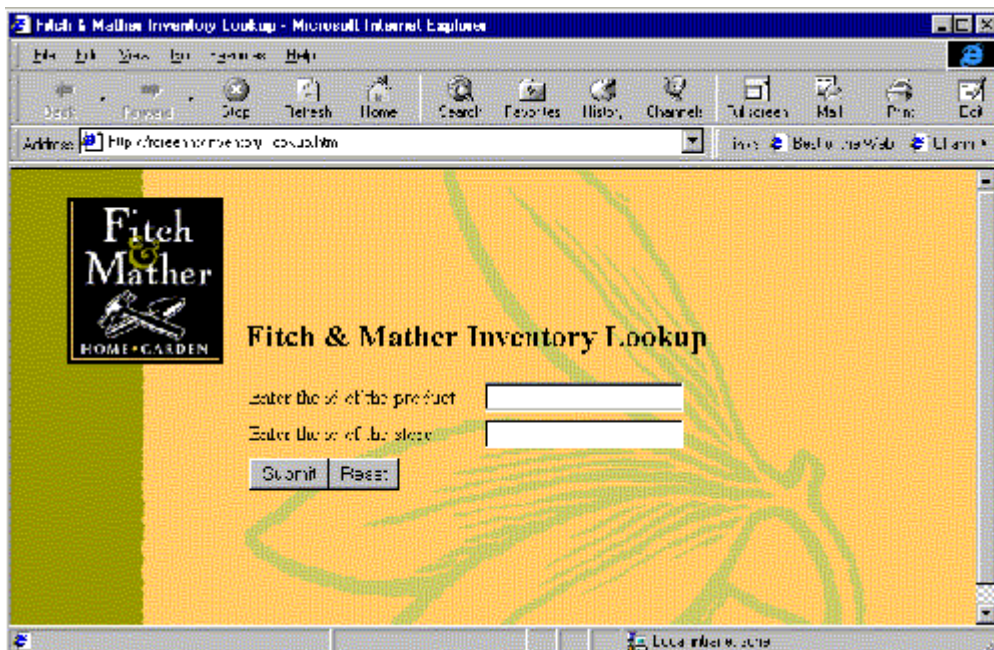


Figure 26: La pagina di ricerca dell'inventario così come appare all'utente.

Notate che l'architettura utilizzata qui, Active Server Page che chiama un componente COM, fornisce una quantità notevole di soluzioni. Tutta la logica dell'applicazione gira sul server, in modo che qualsiasi browser compatibile HTML3.2 può eseguirla.

3. Avviate il browser e navigate verso `<servername>/inventory_lookup.htm` dove `<servername>` è il nome del vostro web server. Inserite `41` come **product ID** e `1` per **store ID**, e cliccate su **Submit**.
4. Le Active Server Page avviano il metodo `GetUnits` del componente fornendo le informazioni inserite come parametri. Quindi costruiscono la pagina che mostra il risultato costituito dal numero di unità presenti in quel negozio.

Supporto del Microsoft Transaction Server

Microsoft Transaction Server è un sistema basato su componenti per sviluppare, distribuire, e gestire applicazioni Internet e Intranet con alte prestazioni e scalabilità. Microsoft Transaction Server fornisce scalabilità per le applicazioni mantenendo una serie di istanze attive del componente. Quando l'utente richiede i servizi di una server middle-tier (intermedio nell'architettura a 3 livelli - dati - applicazioni - utente) Transaction Server soddisferà la richiesta con una delle istanze della serie.

1. Avviate Microsoft Management Console dal menu Start.
2. Nel pannello di sinistra, espandete la lista per **Microsoft Transaction Server**, **Computers**, e **My Computer**. Evidenziate **Packages Installed**.
3. Cliccate col destro del mouse su **Packages Installed** e scegliete **New**. Quindi scegliete **Package**.
4. Nella finestra di dialogo **Package Wizard** scegliete **Create an empty package**.
5. Nominare il nuovo pacchetto `vfp6tour`.
6. Cliccate su **Next** e quindi cliccate su **Finish** per aggiungere il pacchetto.
7. Espandete la lista `vfp6tour` e selezionate **Components**.
8. Cliccate con il destro su **Components** e quindi su **New**. Cliccate su **Component**.
9. Nella finestra di dialogo **Component Wizard** scegliete **Install New Components**.
10. Nella finestra di dialogo **Install Components** scegliete **Add Files**.
11. Selezionate il file `vfp6tourcom.dll` e `vfp6tourcom.tlb` nella cartella `c:\vfp6tour`. Cliccate su **Open** per chiudere la finestra di dialogo **Select files to install**.

12. Ora potete vedere I due files che avete selezionato elencato nella parte alta della finestra di dialogo **Install Components**. La classe inventory è elencata nella parte bassa. Cliccate su **Finish** per chiudere la finestra di dialogo **Install Components**.
13. Comparirà l'ìcna del nuovo componente.

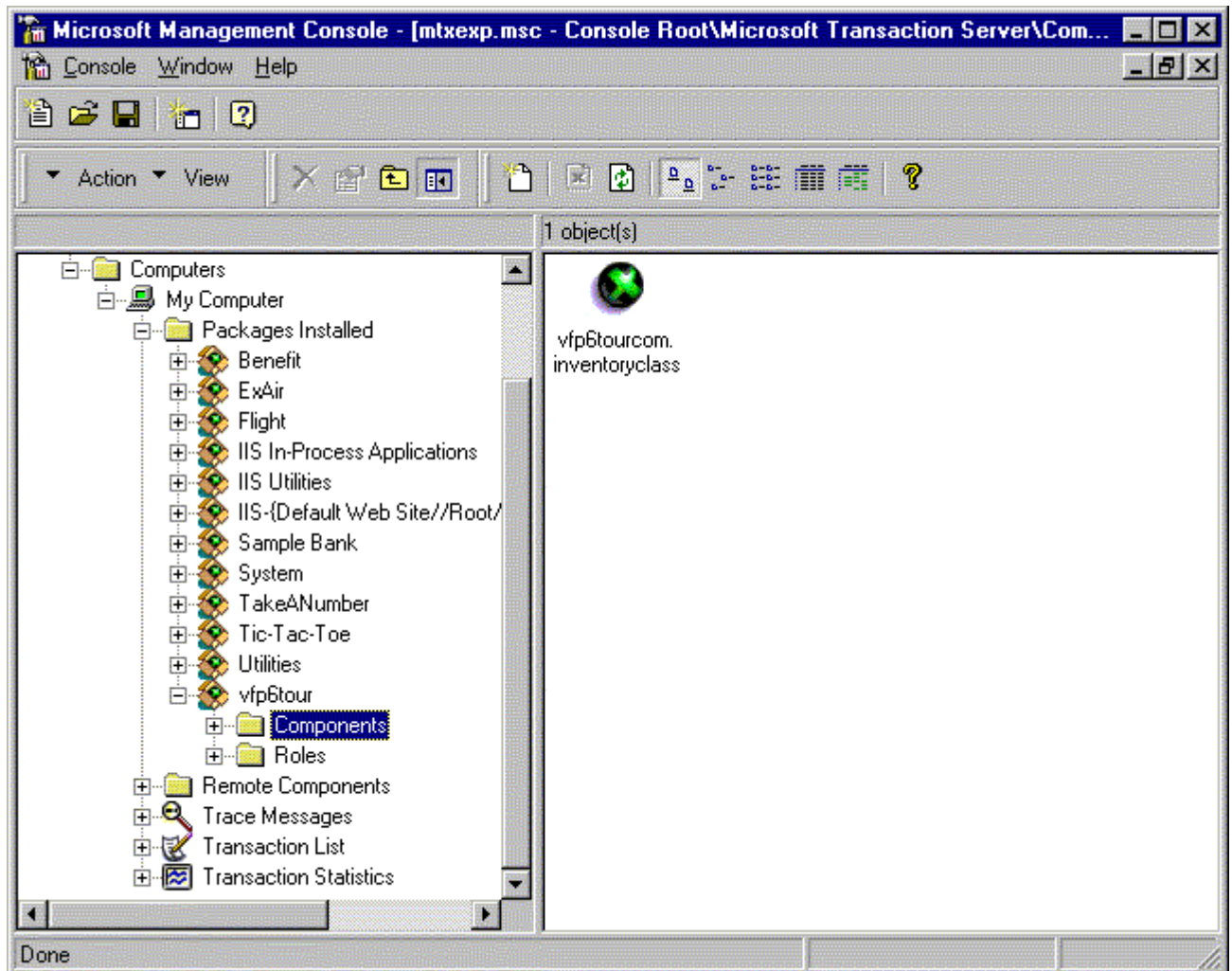


Figure 27: Usate Microsoft Transaction Server per gestire componenti Visual FoxPro COM components e ottenere una maggiore scalabilità.

14. Avviate il browser e navigate verso la pagina inventory_lookup.htm creata nel precedente esercizio.
15. Sistemate il browser e Microsoft Management Console in modo da vedere sia la maschera html e l'ìcna del componente inventario.
16. Nel browser inserite il codice ID del prodotto e del negozio e cliccate su **Submit**. Quando l'Active Server Page chiama il componente noterete rotolare l'ìcna del componete nella console Microsoft Management.