

VISUAL FOXPRO 3.0



Proof that Microsoft is Committed to FoxPro Users

By Tamar E. Granor
Editor, *FOXPRO ADVISOR*

Don't consider them catch-up features.
The object-oriented extensions and
client/server enhancements are the pudding.

FoxPro's back. For years, Microsoft FoxPro (originally from Fox Software) was considered a technological leader in the desktop database market. Its reputation was built initially on blazing speed, but enhanced later by the design tools introduced in FoxPro 2.0. In the last year, however, other products have leapfrogged over FoxPro with object-orientation and client/server enhancements.

Visual FoxPro 3.0 for Windows jumps back into the fray. It features integrated client/server capabilities (and new data types to support them), object-oriented programming extensions, a new event model which does away entirely with the foundation READ, a grid control for one-to-many data entry, a new Form Designer similar to

the one provided by Visual Basic, and an integrated data dictionary. The Project Manager has been enhanced to let it serve as the center of all your activities.

Microsoft Visual FoxPro 3.0, Standard includes development product (with client/server functionality); Professional adds Distribution Kit, Library Construction Kit, Upsizing Wizard, Help compiler, and FoxPro help file; prices not available at press time.

Microsoft Corp.
800-426-9400, (206)882-8080
Fax (206)883-8101

In addition, the interface contains dockable toolbars (including a "standard" toolbar like those in other Microsoft applications), tool tips and other features which bring FoxPro securely into the Microsoft fold. A substantial collection of "designers" for tasks as diverse as creating input forms, laying out reports and labels, querying data, modifying data structures, and connecting to remote data, make the visual part of the Visual FoxPro name appropriate.

Wizards and Builders complete the picture by simplifying many tasks. As in FoxPro 2.6 and other Microsoft applications,

Tamar E. Granor, Ph.D., is Editor of *FOXPRO ADVISOR* Magazine and co-author of the *FOXPRO ADVISOR* "Q&A" column. Tamar is also co-author (with Ted Roche) of the forthcoming *Hacker's Guide to Visual FoxPro 3.0 for Windows* (Addison-Wesley). She has been a speaker at numerous FoxPro conferences. Tamar is an independent consultant specializing in applications using Microsoft FoxPro. CompuServe 73227,3303.

Wizards are one-shot guides to creating an object or performing a task. Builders are more powerful, as they can manipulate an object-in-progress.

Project Manager

Visual FoxPro's Project Manager is updated to make it easier to work with. It uses a tabbed interface, providing access to all objects in the project or to objects of a particular type. The Project Manager can be collapsed so it takes up minimal screen real estate. Individual tabs can then be "torn off" and placed anywhere on the screen (figure 1).

As in previous versions, the Project Manager lets you edit any object with its native tool. In Visual FoxPro, you can also run an individual object directly from the Project Manager. You have a choice as to the default behavior; double-clicking an item can either run the item or open it for modifications.

Data dictionary

Visual FoxPro has several improvements in data structures. The most far-reaching is an integrated data dictionary. The data dictionary is an additional layer over the .DBF. It contains information about the tables (and other things, such as views) that comprise the database and about their relationships. The actual data is still stored in the .DBFs. The .DBF structure has been modified to include a back-link to the data dictionary. (Unfortunately, this means that Visual FoxPro tables can't be read by earlier versions of FoxPro. The COPY TO command is enhanced to allow creation of backward compatible tables.)

Like many things in Visual FoxPro, the DataBase Container is simply a FoxPro table. It has an extension of .DBC, with associated .DCT memo and .DCX index files. You create a database, then add tables to it. You can manipulate a database either visually through the new Database Designer (figure 2) or by using commands (with new and enhanced SQL commands).

Tables that belong to a database can have many additional properties beyond the traditional Xbase structure. Among the items supported are long table names and long field names, field-level and row-level rules, and triggers. In addition, you can create truly unique indexes (nothing like Xbase's traditional INDEX ON..UNIQUE, which aren't unique) and mark one as the primary key for the table. Visual FoxPro enforces the uniqueness of such indexes. Tables are created and modified in code and with the Table Designer (figure 3).

Both the language and the Database Designer let you create persistent relations among tables. These relations are then used as the default in various situations, such as for join conditions in the Query Designer and by various Wizards and Builders.

Other data enhancements

Visual FoxPro includes null support. Any fields of a table can be defined to accept null values. Several commands and functions allow working with nulls.

Figure 1—The Visual FoxPro Project Manager collapsed, with one tab torn off and one expanded.

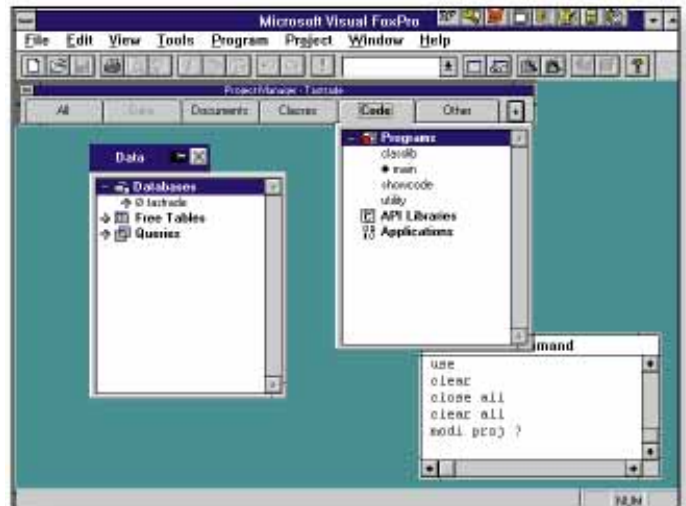


Figure 2—Visual FoxPro Database Designer.

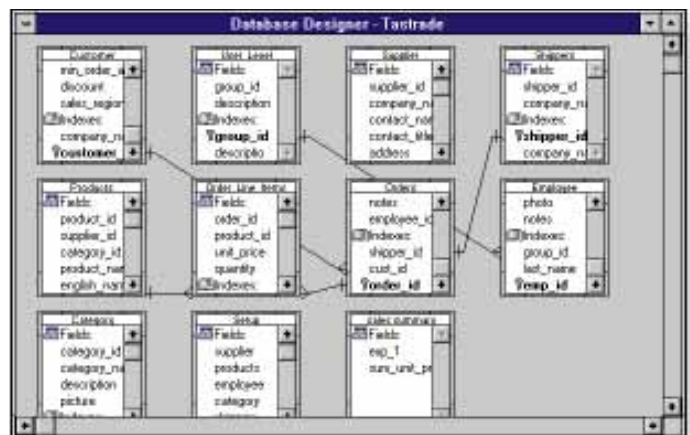


Figure 3—Visual FoxPro Table Designer.



There are three new data types. The datetime type incorporates date and time in a single value. For example, {11/1/94 8:57am} is a valid datetime value. Double applies to fields—it's converted automatically to numeric for variables. Double provides the double precision floating point values needed for scientific and statistical calculations. Each double value requires eight bytes of storage. Currency data also uses eight bytes and provides fifteen digits and four decimals places to store money amounts. Both Double and Currency are intended primarily for compatibility with back-end server data.

New functions allow conversion between Datetime and Currency values and the more familiar Xbase data types. In addition, new functions are also provided to pull individual time components out of a Datetime value. For example HOUR() returns the hour portion.

Object orientation

Visual FoxPro brings object-oriented programming to FoxPro. Both the language and the tools are enhanced for creating objects and classes. Visual FoxPro has all the features that define object-orientation: encapsulation, inheritance, polymorphism.

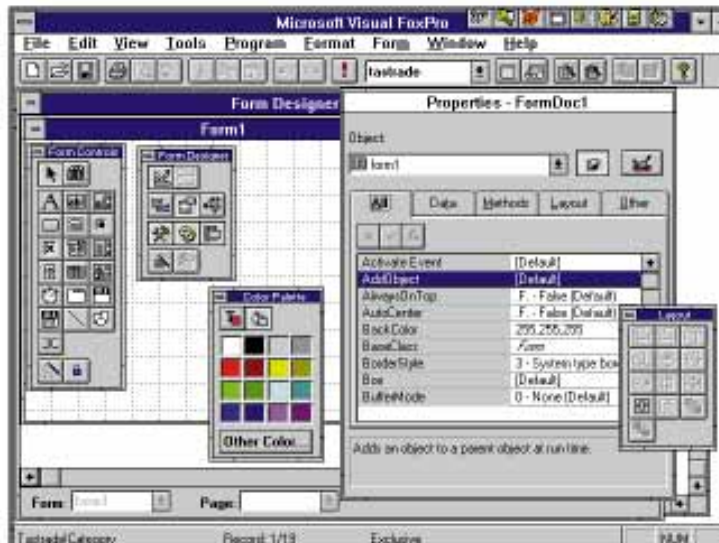
The Class Designer is used to create so-called "visual" classes. Visual classes, once created, can be used as the basis for subclasses, or instantiated to create specific forms or controls. In fact, even classes having nothing to do with forms can be created in the Class Designer. The language also allows the creation of classes and subclasses, and their instantiation.

These object-oriented extensions make it simple to create libraries of objects that can be used in many applications. Encapsulated classes will be easy for developers to share, moving closer to the idea of plug-and-play application development.

Form Designer

The new Form Designer is a significant departure from older versions of FoxPro. While table design information continues to be stored in a table (still using the .SCX/.SCT pair of extensions), forms (a.k.a., screens) no longer require generation. A new command, DO FORM, executes the form directly from the table.

Figure 4—Visual FoxPro Form Designer with all four toolbars and the Properties Sheet available.



A form or form set can have an associated Data Environment. The tables specified in the Data Environment are automatically opened, appropriately buffered (see below), when the form is instantiated. In addition, fields of tables in the Data Environment are offered as choices to which a control in the form can be bound.

When you open the Form Designer, you see a blank form (or the one you've asked to edit), and four dockable toolbars: Form Designer, Form Controls (similar to the one in FoxPro 2.x, but with additional choices), Layout (to line up and size controls), and Palette (to color objects). You can turn off any of the toolbars (even the standard toolbar). Toolbars can be

docked at the top, bottom or either side of the screen. In addition, the Properties Sheet appears. Figure 4 shows the Form Designer with all four toolbars and the Properties Sheet.

A window for editing method code is available by double-clicking on the form or any control (and several other ways as well). The Form Designer toolbar gives you quick access to form-related toolbars that are turned off, as well as to the various editing windows associated with a form (Properties Sheet, Code Window, Data Environment).

Following the object model, both forms and controls are objects. A form is a container object, meaning other objects can be added to it. Both forms and controls have properties, events, and methods, which can be set at design time or at runtime.

Controls

Available controls include familiar items, such as push buttons (now called "command buttons"), check boxes, radio buttons (now called "option buttons"), spinners and list boxes, and some new items. Textboxes and editboxes (corresponding to FoxPro 2.x's textual @.GETs and @.EDITs) are considered controls. The new combo box control can operate like FoxPro 2.x's popups or allow input of items not listed, like a standard Windows combo box.

Unlike older versions, labels, images, lines, and shapes are considered controls and have properties, events, and methods of their own. This means, for example, that you can specify an action to occur when the user clicks on a particular label.

The new control most eagerly awaited is undoubtedly the grid. This is a Browse-like object that can be embedded seamlessly in a form. A grid is a container object, composed of columns. A column contains a header and another control.

Any control can be used in a column, so you could, for example, define a numeric column to use a spinner.

Visual FoxPro also contains a timer control. This control has no physical appearance, but allows you to cause things to happen at specified intervals.

Two types of OLE objects can be added to forms: OLE containers and OLE bound objects. OLE containers are custom controls (.OCX) and objects from an OLE server such as a Word document or Excel spreadsheet. OLE bound objects are bound to General fields.

Page Frames and their contained formpages allow creation of tabbed and tab-less multi-page forms. Within a Page Frame, a single page is visible at any time. (This differs from form sets, where multiple forms are available simultaneously. In fact, each form in a form set could contain a Page Frame of multiple pages.)

The Controls toolbar also provides access to visual classes. You could, for example, define your standard tool bar (perhaps Add, Edit, Save, Delete, Cancel) as a visual class (toolbars are one of the base classes included), then insert it into new form sets when you create them.

Form templates

By default, a new form in the Form Designer is based on the Visual FoxPro base class for forms. However, you can specify a template class to be used to instantiate new forms. This means that you can design a master form once and use it as the basis for all new forms. Make changes to your master, and changes are immediately reflected in all existing forms.

The event model

Visual FoxPro does away with the clumsy foundation READ. A new command, READ EVENTS, is used to keep multiple forms active. The corresponding CLEAR EVENTS stops event processing. Additional forms can be opened at any time without concern for read-levels or any other limit, except system resources such as memory.

Multi-user support

Visual FoxPro includes a number of enhancements for multi-user applications. Row and table buffering allow you to edit fields directly, with the ability to revert to original data. Transaction support lets you commit groups of changes at once, reverting to original data if some of the data can't be updated.

Data sessions let you run several copies of a form, each operating in a separate data environment. You determine

whether each session is updated to reflect changes in other data sessions.

Buffering

Row buffering means that changes to the current record are buffered until the record pointer moves from that record or you explicitly commit the changes. Table buffering provides the same service for an entire table. The availability of these buffering modes means there's no longer a reason to do input against memory variables or arrays. You can do input against buffered data with an easy way to revert if the user changes his mind. Buffering even works with newly added records—the records aren't actually saved to the table until

you move the record pointer or explicitly update.

There are two buffering modes for both row and table buffering. Pessimistic buffering locks the record as soon as you make changes to the buffered data. Optimistic buffering doesn't lock a record until you commit the buffered changes.

Row and table buffering work together with the Form Designer's Data Environment. You can set a form's BufferMode for either optimistic or pessimistic buffering and all tables in the Data Environment automatically use the specified mode.

Data Sessions let you run multiple instances of a single form. Each has its own independent set of buffers. Changes to one session can be reflected in other sessions (or not). With data sessions, you don't have to provide distinct aliases for the tables or worry about using different work areas. It's all handled behind the scenes.

Several new functions are provided to help you manage buffering. You can access the original value of a field as well as the current value on disk, to determine why an update failed (or whether an update is necessary). Functions also let you check whether a specified field has changed and find the next changed record.

Transaction processing

Sometimes it's necessary to treat a number of changes as a single transaction, something which is either entirely saved or entirely discarded. Visual FoxPro provides support for these operations as well.

As with buffering, changes in a transaction are either committed (with END TRANSACTION) or reverted (ROLLBACK). Transactions can be nested up to five levels. None of the changes are committed until you end the outermost transaction. Rolling back the outermost transaction rolls back all the nested transactions.



Figure 5—Visual FoxPro View Designer.



Figure 6—Visual FoxPro Connection Designer.



Client/server capabilities

Client/server support is built into Visual FoxPro without any additional kits, as you needed in previous versions. Two modes are provided for working with back-end data.

SQL Pass Through (or SPT) uses a model similar to that of the now defunct Connectivity Kit. A collection of functions let you connect to server data, pass commands to the server, retrieve information about the server data, and so forth. These functions provide a low-level approach to client/server computing.

The alternate approach is to use views, which can work on both local and remote data, as well as on mixed data sets. A view, which can be created either in code or using the new View Designer (figure 5), retrieves a data set into a cursor that can be updated. The cursor can be manipulated as if it were native FoxPro data. Changes are stored to the back end based on properties of the view, which you establish. Views exist only within databases and can't stand alone.

Views are particularly useful for prototyping client/server applications. You can create the view against local data and use it while designing and testing an application. The new Upsizing Wizard (included in the Visual FoxPro Professional version) then helps you create the appropriate back-end data and work against that instead.

Both views and SPT let you work with previously defined connections to back-end data sources or create new ones. The Connection Designer (figure 6) lets you define connections visually.

Wizards and Builders

Wizards walk a user through a series of questions, and use that input to perform a specified task. Visual FoxPro includes a number of Wizards for tasks, such as creating a table, performing a mail merge, importing foreign data, and many others.

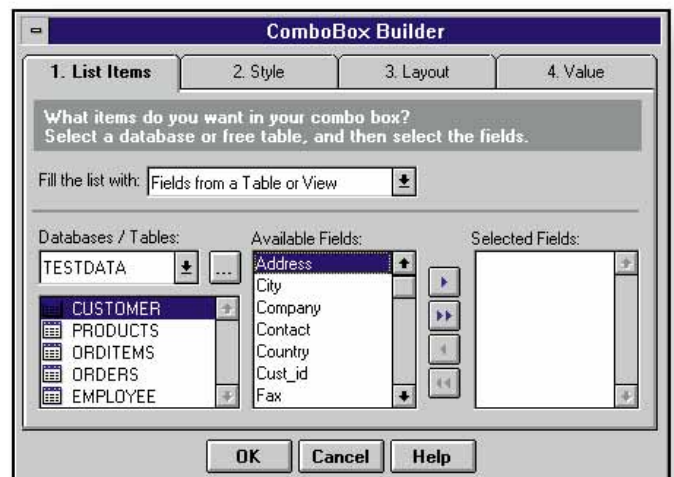
Builders, used in the Form Designer and Class Designer, are far more powerful. They can operate on new or existing objects, using current settings as a starting point. Builders allow you to modify an object without necessarily knowing everything about the object. They provide a simpler interface than the properties Sheet, since they focus on key properties that define the object.

Visual FoxPro includes builders for most controls, as well as for entire forms (figure 7 shows the ComboBox Builder). Expect lots of builder add-ons once Visual FoxPro hits the street.

And there's more

Visual FoxPro introduces many more changes. As I described, OLE controls can be placed in forms with in-place

Figure 7—Visual FoxPro ComboBox Builder simplifies creation of combo boxes.



editing. In addition, Visual FoxPro supports OLE Automation.

Visual FoxPro provides an unbelievable 32,767 work areas. In addition, each data session gets its own set of work areas, so the number is practically unlimited. The View window (figure 8) is updated to reflect these changes and shows open tables from a single data session at any time.

FoxPro's SQL command set has been enhanced, both for designing databases and working with them. CREATE TABLE addresses the new database capabilities. The new ALTER TABLE command lets you modify a table programmatically.

On the other side, the SQL DELETE and UPDATE commands have been added, so you can modify existing data using SQL commands. One benefit is that views can be used in client/ server applications, and for updatable cursors of FoxPro data.

Two new settings let you specify the first day of the week and the first week of the year. These settings affect DOW() and CDOW(), as well as the new WEEK() function, which returns a week number within the year.

Local scoping means true black-box routines. Visual FoxPro allows both variables and parameters to be declared local. Variables declared local can be seen only in the routine in which they're declared, unlike private variables which are visible in all routines called from the first.

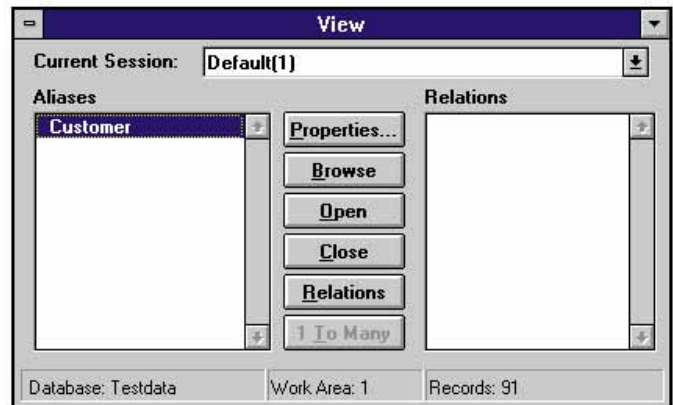
The various objects that comprise menus, including popups, now have FONT and STYLE clauses, so you can determine the look of a menu.

As in other Microsoft applications, the right-mouse button is finally useful. Clicking it in most of the visual tools brings up a menu relevant to the task at hand.

What's missing

Visual FoxPro's weaknesses are pretty much all in the area of what didn't get updated. The Report Writer (now called Report Designer) and Menu Builder (Menu Designer), though both improved, are the same as in FoxPro 2.x. The most important Report Designer change is the new TO FILE ASCII clause. Similarly, FoxPro's editor hasn't changed; it's still the same functional but basic text editor.

Figure 8—Visual FoxPro View window.



Another area that remains weak is support for programming teams. The Project Manager still has no tools for dealing with multiple programmers working simultaneously (Cascade Interactive Designs is expected to release a Visual FoxPro version of their CapCon version control utility).

Finally, it remains to be seen whether Visual FoxPro maintains the cross-platform tools and support pioneered by FoxPro 2.x. Until Mac and DOS versions are available, it's not clear how many of the new goodies will be portable.

Summary

Visual FoxPro 3.0 brings FoxPro right into the forefront of desktop databases. It has almost everything users have been asking for and much more. The new tools make it possible to create event-driven applications that look and feel like other Windows applications. The additions in multi-user functionality make those applications far more robust. Client/server capabilities in the base product makes Visual FoxPro a great choice as a front end. Other enhancements throughout the product make it clear that Microsoft considers FoxPro an important part of the database picture. ■

Article reprinted with permission from **DATA BASED ADVISOR**, Vol.13, No.3 © 1995 **Advisor Publications Inc.**, 4010 Morena Blvd., San Diego, CA 92117. For more company and subscription information, call 800-336-6060, (619)483-6400, fax (619)483-9851, CompuServe 70007,1614 or Internet 70007.1614@compuserve.com.