Column: Editorial

Figures:
File for Subscriber Downloads: 02COOL.ZIP

Cool (Blue) Tool

Whil Hentzen

Last month I promised a series on moving VFP data back and forth to your Palm Pilot. Since penning that article, Doug Hennig introduced me to another tool that will basically make all the stuff I told you about in January completely irrelevant, except for one or two method calls in this new tool aren't quite working right yet. So while I work on those few pieces, I thought I'd let you know we're resurrecting another column that's often been requested – the Cool Tool. Normally it'll be an extended article available with our free subscriber downloads, but jus so that you pay attention, I'll spend this month's editorial covering the first couple.

By the way, this is the first column I'm writing in the New Year. Unlike many of you who made resolutions to lose weight, stop smoking, and go to church more, I resolved to head each column with a convoluted song title. At the end of the year there will be a quiz (and, if more than three people respond, a prize.)

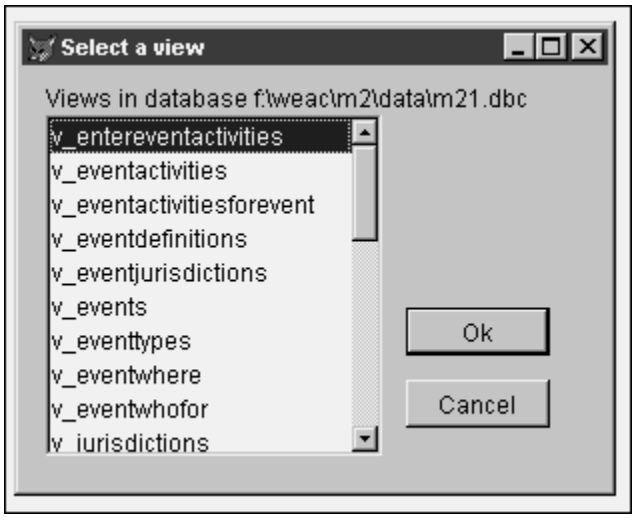
Erik Moore is a profligate tool builder on The Universal Thread, and this month I'm going to cover one of his more popular creations.

As anyone who has used the View Designer for more than trivial views knows, the View Designer is a great demo tool, and it's great to initially create simple views with, but it's pretty much useless in the real world. For instance, it's pretty hard to create complex views. "Complex" being any view that has more than, oh, say, two tables. It's even harder to work with it for any extended period of time without GPFing. And if you ever have the occasion to change an underlying table, well, all hell can break loose in the View Designer.

That's why many people choose to create basic views in the View Designer, run GENDBC on the resulting DBC to produce a program file, and then make subsequent changes to views in the DBC by tweaking the code in the program file.

But this is ***Visual*** FoxPro, so messing with code doesn't seem as "nice" as using a visual interface. Erik wrote eView to compensate for the deficiencies in the View Designer for maintaining existing views, as well as providing a couple of extra features.

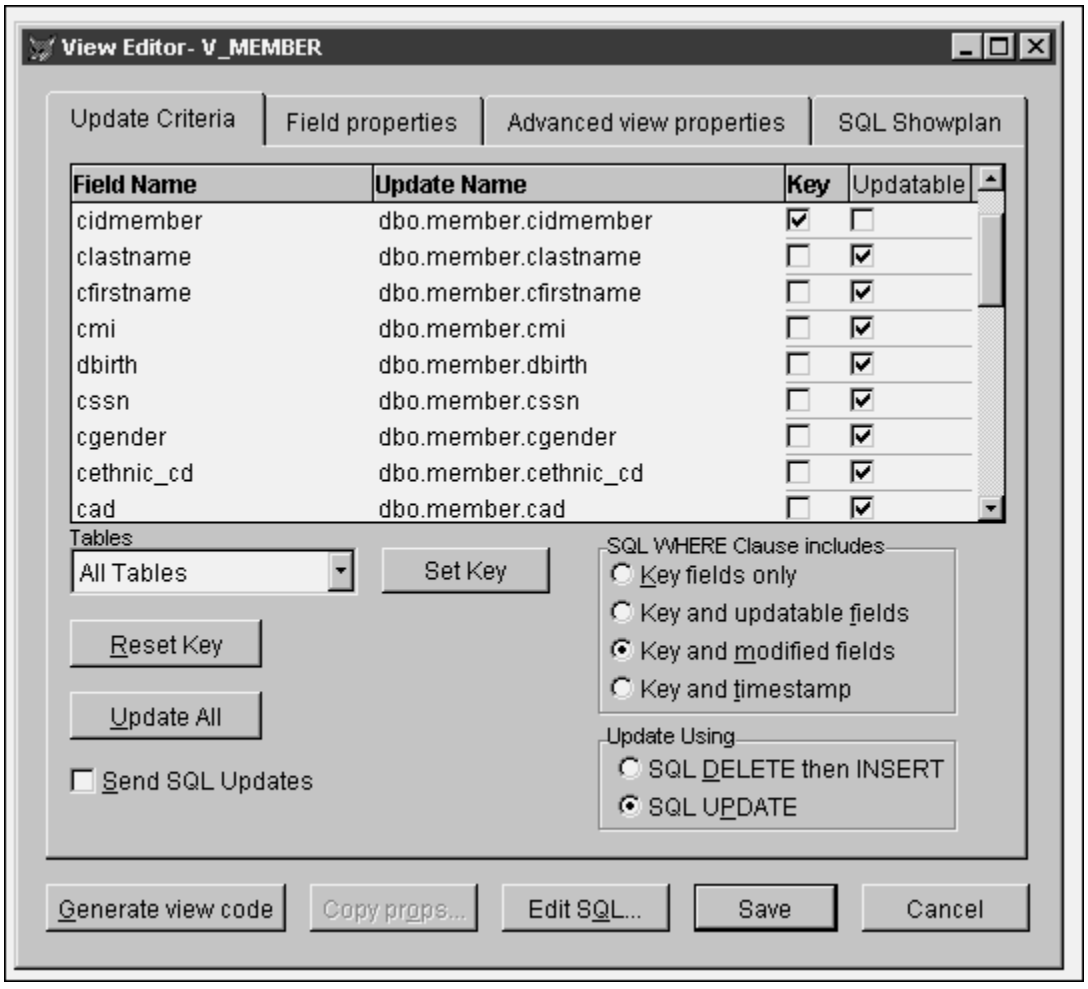
You can call EVIEW with a parameter if you want to have a specific view opened automatically, or you can simply run EVIEW.APP and select the view from the screen shown in Figure 1.



FT-01.TIF

Figure 1. Running EVIEW without specifying a view prompts you to select a view from the current database before continuing.

Once you've selected a view, the main EVIEW screen will be opened, as shown in Figure 2.



FT-02.TIF

Figure 2. The Update Criteria tab of EVIEW allows you to set all of the view's update properties in an easy to use interface.

This screen allows you to do all sorts of things – most of which will be obvious if you've used the View Designer itself. For example, the grid in the top half of the Update Criteria page displays all of the fields in the view, and allows you to edit the update name as well as set the Key and which fields are updateable. You'll immediately notice that the grid is a lot easier to work with than the tiny list boxes in the View Designer. You can also resize the screen to make the grid wider (although the page frame doesn't resize, so you're still bound to the same nine rows.) Underneath the grid you can also mark the view as updateable, configure the SQL WHERE clause, and identify how the update is performed.

The buttons at the bottom of the screen are useful additions. "Generate view code" basically performs a "GENDBC" type function on the current view (as opposed to doing so for the entire database.) The "Copy props" button is dimmed in Figure 2, but its function is to copy extended field properties from underlying tables to the view. It's dimmed in this example because the view in Figure 2 is a remote view based on a SQL Server database, and thus doesn't have underlying tables to grab extended properties from.

Edit SQL allows you to edit the SQL string that creates the view – thus, for example, allowing you to change the sort order or which fields are displayed in the view. Note that EVIEW will ask you if you want

to try to restore all previous view settings. This means you can often still keep all of the properties of the original view, such as being updateable, so that this is considerably more powerful and flexible than running CREATE SQL VIEW yourself.

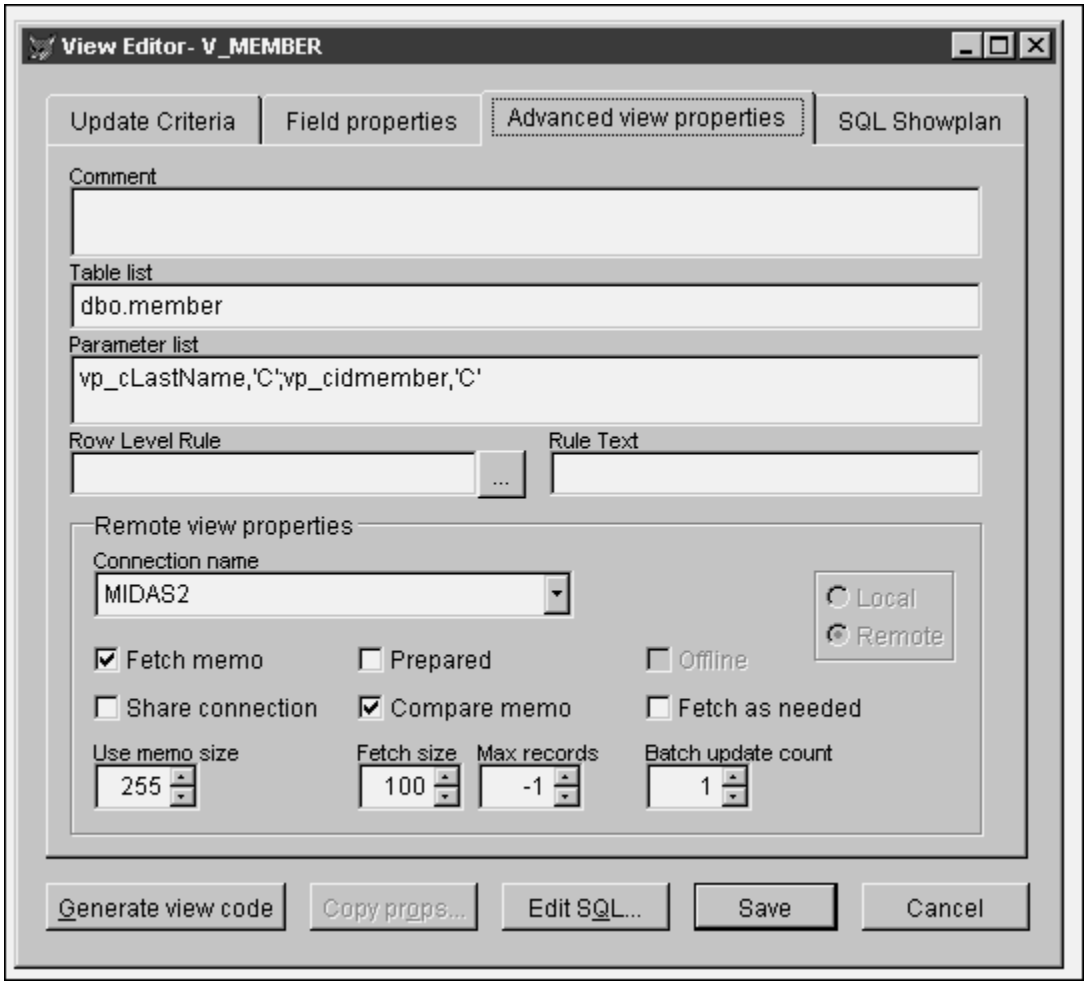
The second tab allows you to set and edit extended properties for view fields. These properties include

*\\\ Andy, these two lists can be combined into a single paragraph instead of individual bullets, if they don't fit.///

- Validation Rule Expression
- Rule Text (Message)
- Default Value
- Caption
- Format
- Input mask
- Comment
- Display class
- Display class library

The third tab, shown in Figure 3, allows you to set and edit properties for Remote views, including

- Fetch memo
- Prepared
- Offline
- Share Connection
- Compare memo
- Fetch as needed
- Memo size
- Fetch size
- Max records
- Batch update count



FT-02.TIF

Figure 3. The Advanced view properties tab allows you to set a variety of Remote View properties.

*\\\ Andy, this screen shot can go away if it doesn't fit. ///

Finally, the fourth tab will generate the SQL Showplan to show you how well optimized the view is.

Note that this isn't a complete replacement for the View Designer – but it's a replacement for using the View Designer for ongoing maintenance of existing views – which is where you spend most of your time.

As with any code that someone else writes, there are a couple of things I'd like or would do differently. First, each of the controls has a tool tip – in some cases, a huge tool tip – but I'd still like a help button. Having even a brief explanation of the ramifications of some of the actions you can perform would be useful. For instance, after playing with the Tables combo for a bit, I figured out what it was for, but a "EVIEW for Dummies" help entry would have been helpful there.

You have to pick a single view to work with first – I'd have put another control, such as combo box, at the top of the screen to allow me to switch from one view to another. And I'm still looking for that Holy Grail of a tool that allows you to make changes to an underlying table, have those changes ripple through to all of the related views, and then have those view changes ripple through to all appropriate forms and classes in the app. But these are all minor quibbles. If you do any significant work with views, EVIEW is worth checking out.

EVIEW is in the public domain, and a copy of it is included with this month's Subscriber Downloads.