

ALTVIEW: An Alternative to View

Most of the tools we've looked at in the column have been the answer to a specific nagging problem. As a result, the outline for this column has generally taken the form of "Here's the Problem, and here's the Solution." This month will be a little different.

Oh, to be sure, this month's Cool Tool had its genesis due to someone's need that wasn't satisfied by the standard FoxPro package, or by anything else that was out there, and it may well answer that little voice in the background: "Isn't it a pain to do this every time..." But just as important as the problem it solves is the way it does it. A lot of things come across the wire to my machine, and, frankly, a fair amount of it is pretty awful. It's hard to make it work, once you've got it to work, it's not intuitive, and, it's downright ugly.

The last may seem to be a trivial complaint, but this is the mid-90's... People have higher expectations now than five years ago - and attractive, good-looking interfaces are part of the package that's considered standard. This month's Cool Tool immediately struck me as one good-looking interface, and we can all pick up a few tips.

The Problem

Altview's author, Val Matison, has long been annoyed by the number of dialogs that had to be invoked in order to get a "bird's eye view" of the current table. And the problem was compounded by the fact that this set of steps had to be repeated when changing to another table. And as we all know, if you make something difficult enough, it's not going to get done. So instead of opening up three more dialogs "just to make sure" about something, it was more likely going to get left for later... Much later <grin>.

Sure, you can use the View window to see which table are open. But if they're not all grouped together, you have to scroll through the window to see if there are any in work areas not visible in the listbox. And this is a distinct possibility due to some system's propensity for opening up certain tables in the highest numbered work areas. It sure would be nice to have a list of all open tables. Even better if you could control how they're presented - alphabetically, or by work area, for example.

Next, you're probably going to want to see the indexes. And the structure. And perhaps some sample data. That's three more dialogs right there. And suppose you want to run a command? They're all modal dialogs, right? You know what that means... Close all the dialogs, run the command, and open up the dialogs again. This can get to be a real pain, real fast.

The Solution

Val put together a single dialog that gave him a bird's eye view of all the relevant information about the open tables. To use Altview, place ALTVIEW.APP in your path and issue the command

```
DO ALTVIEW.APP
```

Note that if you don't have any tables open in the current directory, Altview will present the File Open dialog first.

[Screen shot: ALTVIEW: The Altview dialog]

As you can see, this one dialog displays all open tables. You can change the order in which the tables are displayed - by name, work area, or number of records. You can also perform a number of typical operations with just a single pushbutton - including opening and closing, browsing, and modifying structure.

Two features not immediately obvious, but incredibly useful, are provided with the Fields and All pushbuttons. The "Fields" pushbutton copies a listing of all the fields in the currently highlighted table into the windows clipboard. To retrieve this list, press CTRL+V when editing program code and the list will be pasted at the current location of the cursor. If the "Alias" checkbox in the "Structure" box is checked, the alias will be copied as well. This is a great timesaver when building SQL SELECT commands. The "All" pushbutton opens all the tables in the current directory - a very handy feature. I use Altview just for this purpose alone.

For the selected table, you can see the structure of the table, and like the tables, you can change the order in which the structure data is presented - by field name, field type, field size, or in physical table order. You can also do a "quick print" of the structure with the printer picture pushbutton.

The listbox on the bottom of the dialog displays all the indexes and index tags - hasn't the number of steps required to see the full expression for a tag always irritated you? You can also select a tag and change the order And you can see sample data for the first 20 records in the table - which is handier than you'd think at first glance.

In the center of the dialog, you can use the command editing window to issue commands from within Altview. After typing the command, press (or Tab to) the Invoke pushbutton to execute it. You can use Clear to erase the contents of the Command window and Save to copy the command to the clipboard for later use with FoxPro's Command Window or for pasting in a program.

The "Browse" button next to the command window will browse only fields checked from the field list. The Query button will perform a SQL select on all fields checked. These commands are also copied to the command window to be recalled or copied.

Another object that provides functionality not immediately obvious is the "Fields" checkbox in the structure region. If this checkbox is checked, then each time you click on a field in the fields listbox, that fieldname is echoed to the command window. Here's an example of how you can use this function.

Suppose your invoice items table has fifty fields and you only want to see transactions for customer "Rolling Stone" before the first of the year. Click on the Fields checkbox, and then click on CustName and TransDate. Those fields will be echoed to the command window, where you can edit the command to say "set filter to CustName = 'Rolling' and TransDate < {1/1/94}" and press Invoke to actually set the filter.

Next, select several fields from the Fields listbox. They'll get echoed to the command window as well, but you can ignore them. Now press Browse - voila! You're displaying only a few fields from that fifty field table, and only for a few records. Rather easy - and you can select the Save pushbutton from the Command buttons group to paste the browse command into the clipboard.

Interface Notes

This amazing variety of functionality in one dialog is only half the story, however. Let's take a look at some of the nice interface features.

First, just a general overview of the dialog - there isn't a pixel of wasted space, but it's not so crammed that you can't find what you want. And the use of fonts and data objects was well thought out. It takes about two minutes to get completely comfortable with everything that Altview has to offer. This is a goal that we should all strive for in our interfaces.

Next, notice how Val has segregated each subset of functionality with the 3D chiseled boxes. I've seen other applications that use a panoply of colors to distinguish between parts of a dialog. Not only is this a poor choice because of the possibility of users without color-sensing vision, but in practical use, the color choices made by the developer have left something to be desired.

Finally, his judicious use of picture pushbuttons helps clean up the dialog. It's immediately obvious what the printer pushbutton does, for example. And notice his use of his logo in the pushbutton below the "Done" pushbutton to indicate that this would bring up "About..." information. This is one of those little tricks that will make everyone who sees it say "Cool!" And that, after all, is what this column is all about.

Where to find it

ALTVEW.ZIP is included on this month's Companion Diskette, and can also be found in Library 9 of FoxUser. It's a 40K ZIP file that decompresses into a README text file, the ALTVIEW.APP, and the source code files. It's freeware so there's no registration fee, but do send the author, Val Matison, a note telling him thanks for his contribution.