

# Practicing Good Form in TAOS

*By William Baker*

Randolph's early experience with TAOS: The Developer's Workbench™ was using it as a report writing tool. He never considered using TAOS tools for data entry programs, and he probably never would have if he hadn't been pushed to do so by a customer. Come to think of it, that's about the only reason Randolph ever considers doing something new, and sometimes the pushing has to be pretty strenuous.

Walt, who works in the infamously filthy warehouse I've told you about, kept asking for more and more selections on an open sales order report. Besides selecting on a range of order numbers, he wanted to select on a range of customer numbers, a range of order dates, a range of shipping dates, method of shipment, the total value of the order, and payment terms.

Randolph collected all these options with a long series of prompt for verbs, which is the TAOS command that prompts the user for one field of data entry. Inevitably, every now and then Walt typed a date or some other field wrong so that no orders were selected for printing. At first Walt blamed Randolph for writing the procedure incorrectly, but Randolph was sure that Walt was entering bad data. To prove his point, Randolph printed all the selection criteria in the report. After five or six occurrences of the report trying to find sales orders with a month greater than 31 and such, Walt had to admit he was typing in the dates incorrectly, but he still said it was Randolph's fault because the program should let him fix his mistakes. Randolph added another prompt on every screen that said something like, "Are you sure this entry is correct?", and allowed the option of re-entering the data. Walt didn't like that approach because it slowed him down too much. Then Randolph added a menu that allowed Walt to pick any selection he wanted to enter, but Walt didn't like that either.



Then Randolph came to talk to me. I suggested a date-verification routine, but Randolph pointed out that would only solve part of the problem (which was correct), and besides, he didn't want to write a date-verification routine in TAOS or any other language. Randolph agreed with Walt and me that a long series of prompts made for an awkward interface, and I suggested a data entry form that would allow input of all the selection criteria on one screen. Needless to say, Randolph did not want to write an input screen either, but after working with the TAOS Form Generator for a few weeks, Randolph had to admit it has advantages over writing input screens in a third-generation language.

The Form Generator includes an interactive form designer, and even Randolph enjoyed learning to move fields around and especially drawing lines. In fact, I think he enjoyed drawing lines too much on the form that he showed me. The part of TAOS forms that is more interesting to me is the programming to handle different actions that can occur on each field. Doing searches on fields was Randolph's greatest concern, because he needed to present a list of customer names on the customer number fields.

In a third generation language application, I would write a loop to capture and test every keystroke, looking for the search key, among others. When this loop detected a search key, it would call a search routine. In TAOS Forms this checking for certain keystrokes is done by the keyboard input routine, and all I have to do is specify the search routine to be used for that field.

The search routine that Randolph had already written for Walt to look up customers by name looked like this:

```
INIT CUSTOMER;
CHOOSE/RECORD
  /DISPLAY=CUST_NUM+ " " +NAME
  /INDEX=NAME
CUSTOMER;
```

In the form setup program, I told Randolph to enter the same routine in the field that defined the search for customers, with an extra statement at the end:

```
INIT CUSTOMER;
CHOOSE/RECORD
  /DISPLAY=CUST_NUM+ " " +NAME
  /INDEX=NAME
CUSTOMER;
%=CUSTOMER.CUST_NUM;
```

The last command is interpreted by the Form Generator to mean set the current form field value equal to the value in customer.cust\_num. The percent sign (%) is a symbol for the current field. This equation is needed because the form program has no way of knowing that the user has just executed a search routine. It's up to the programmer to move the results of the search into the form's data structure.

Randolph was suitably amazed by how easily he could insert a search routine into a data entry screen, but he had a valid concern about how long a search routine he could specify.

Form Screenshot

"I can think of two ways to have a long search routine in a form," I said. "You can use the TAOS call verb, which works a lot like call in BBx® and PRO/5™."

"What help is that for me?" Randolph asked. "I don't know how to use the call command in BBx, so what does it matter if it works the same in TAOS?"

"That's a great help. When you learn how to use it in one place, you'll know how to use it in the other. Look at this example:"

```
CALL "(BBX)CUST_SEARCH.BBX", %;
```

"This command resolves the (BBX) global string to the path where compiled TAOS programs are stored; CALLs the program `cust_search.bbx`; and passes the current form field as the only parameter. The `cust_search.bbx` has the commands to do the search, and the selected customer number is returned to the current form field."

Randolph shook his head. "Passing, returning, I never understood all that. I hope your other method makes more sense."

"The other way is to specify an include file with TAOS `#include` pre-processor command. For example:"

```
#INCLUDE CUST_SEARCH
```

"This tells the Form Generator to insert an ASCII file called `cust_search` at the right spot in the form code. This file contains the TAOS commands for the search, and it compiles exactly like TAOS code written in the form design interface."

"Well, I'll try to keep all that straight, but I guess for now I'll concentrate on keeping my searches short so I can just type them into the form. And I just realized that nothing makes me hungrier than talking about forms-unless maybe it's talking about food-so it certainly must be time for lunch."