November, 1995

ASK ADVISOR

Q: To export data from a master list to a mailing list, I usually use choose by zip codes. I run a query and then export to a file. My problem is that I may need to specify multiple zip codes. I use a condition like:

```
Mydata zip = "33321" .or. mydata zip = "30011" .or. ;
mydata zip="23212"
```

Is there any easier way to do this?

—Craig Nelson (via CompuServe)

A: There are a couple of different ways you can do this. Choosing the right one depends on how many zip codes you'll typically be including in a mailing list and on the size of your master list.

If you normally include only a few zip codes and your total list isn't too big, you can simply create a string containing a delimited list of the zip codes and check whether each record's zip code is in the list. The list would look like:

```
"33321,30011,23212"
```

and your query would look like:

```
SELECT * FROM Master ;
  WHERE zip$"33321,30011,23212" ;
  INTO CURSOR Chosen
```

You can build the list based on user input and store it in a variable, say cZipsChosen. Then, your query would look like:

```
SELECT * FROM Master ;
  WHERE zip$cZipsChosen ;
  INTO CURSOR Chosen
```

The delimiter in the list is important since it keeps the data from running together. If you specified:

```
cZipsChosen="33221300112321"
```

without the commas, the query would also include records with zip codes of 32213, 22130, 21300 and so on. The delimiter prevents this. If you were working with data other than zip codes that might include commas in the data, you'd need to choose a different delimiter character. Good choices are characters which don't usually appear in normal data, like "~" and "|".

So, what's wrong with this approach? A couple of things. First, the $ operator can't be optimized, so with a large data set, it could be slow. Second, building the list of zip codes based on user input is tedious.

Another approach is to put the list of chosen zip codes in an array or a cursor and check against that. You can start with a list of all available zip codes and let the user choose the ones she wants by presenting a screen. (A two-column mover like the one in Alan Schwartz's August 1993 article is good for this purpose, but you can use a multi-select popup, too. In Visual FoxPro, you can use a listbox with the multi-select property.)

You can create the overall list of zip codes like this:

```
SELECT zip FROM Master ;
  GROUP BY 1 ;
  INTO ARRAY aAllZips
=ASORT(aAllZips)
```

This creates an array, aAllZips, containing each unique zip code found in the master list. Then it sorts the list into zip code order. (I'm ASORTing the resulting array rather than using ORDER BY because it tends to be faster this way.) Now you can use the array as the basis for one of the input mechanisms described above.

If your screen leaves you with an array aChosenZips of the desired zip codes, your query would look like:

```
SELECT * FROM Master ;
  WHERE ASCAN(zip,aChosenZips)<>0 ;
  INTO CURSOR Chosen
```

But ASCAN() also isn't optimizable, so another alternative is to store the chosen zip codes in a cursor (temporary table) called ChosenZips and use a query like:

```
SELECT Master.* FROM Master,ChosenZips ;
  WHERE Master.zip=ChosenZips.zip ;
  INTO CURSOR Chosen
```

This query is optimizable. Just make sure you have an index tag on the zip field of Master. (If the list of chosen zip codes is large, you'll want to index the cursor, as well.)

Setting up the actual input mechanism would take more room than we have in this column, but the suggestions above should give you somewhere to start.

–Tamar