around on screen, the ruler reflects cursor position instead of the item, making precise placement difficult.

Since *Approach* uses several popular data formats, you'll enjoy an unusual degree of flexibility. However, the implementations are nonstandard, and it's easy to be lured into a false sense of security regarding your compatibility with other applications. Particularly with .DBF files, there's room to worry—not so much because of defects in *Approach*, but because any time you alter a standard you risk problems. We had no problem using *Approach*-created files that included picture fields when we accessed the files outside the program, but of course, other .DBF-compatibles could not get at the images. Of much greater concern is the fact that there is no “recall” command to bring back deleted records. Engineers at Approach inform us that this was an intentional design decision, made with an eye to making the program as easy to use as possible for novices. We question it. Not only does it create a potentially dangerous situation where data might be handled differently from outside *Approach* than within it, but *Approach* includes a “Compress” (Pack) command, illogically buried beneath “Join”. It would make more sense to at least include the ability to recall deleted records here, too.

There's at least one area

where *Approach* absolutely excels. Although the screen formatting options associated with the act are limited, creating a one-to-many relationship among fields in separate data files is child's play. The job is accomplished with just a few clicks of the mouse, and in a manner so simple as not to be worthy of a description here. Honest: anybody who does it once will never forget how. The other side of this issue is that it's a good thing the act is easy, because documentation is sparse at best. Only about twenty pages of the manual are devoted to relational links, including the space that explains what the term means.

In fact, documentation is generally weak. We understand; *Approach* is supposed to be easy, so why complicate matters? Because not everybody who uses it will be a novice, that's why. Some software gets around this quandary by providing especially strong on-line help, but *Approach* is quite weak here, as well. <F1> means nothing, and there's no context sensitivity.

Many of the small touches that color the *Approach* experience are mixed. The UNDO command seems to fail whenever you really need it, headers and footers on forms support only dates and page numbers as automatic object types, and certain types of report are supported only in *Approach*'s preview mode. Positively, *Approach* supports *Windows*' Multiple Document Interface for its proprietary .VEW files, thus imparting the ability to keep a remarkably large number of data files open simultaneously. A long list that must comprise the full line of Avery mailing labels is supported, and empty fields are automatically compensated for. There are truly useful application examples included, and the macro facility, although implemented oddly from a single dialog box (figure 3), is flexible and easy to learn.

Approach's program code requires less than two megabytes of disk space, but you can push the total considerably higher if you keep the samples and install the bundled copy of *Adobe Type Manager*. The program supports bidirectional use of .WK1, *Excel 3.0*, and text files. We'd like to see more respect for your system, as the installation procedure leaves unneeded temporary files behind, and puts two .DLL files in the SYSTEM directory that *Windows 3.1* users will already have. Finally, *Approach* requires the use of SHARE.EXE; this is a safety procedure that assumes user naivete, and many will find it annoying.

It's a neat product, and for now, *Approach* leads the *Windows* database pack. We only like it though, not love it. More's coming soon. We'll wait.

Approach
Approach Software
311 Penobscot Drive
Redwood City CA 94063
800-277-7622
$399