to make up for this shortcoming with more features.

You use *Current* by establishing relationships between items in your database. For example, you might want to create a view in which all items called "expense account" are presented in tabular form and summed by type of expense. Once you create the report form, the data is automatically gathered, sorted as you designate, and displayed, as simply as clicking on an icon with the left mouse button or picking the report's name from a menu. Want more detail? Double clicking on the right mouse button reveals more in-depth information about the item chosen. In your phone book, this might reveal a contact's address as well as phone number, birthday, or any other data you teach *Current* to format and display. After you learn what the key words mean within the program's context, the process of defining these views becomes almost second nature; just push the appropriate "buttons". The associations you create can be simple, such as "assign all items that reference the phrase `giant anteater' to Jane", or complex - "if today is a Tuesday and my secretary is absent, have Bob back up the network server". Once the necessary relationships between bits and pieces of your data are assigned, ad hoc data query is incredibly easy.

If you think that establishing all the relationships you need may take some doing, you're right. Although *Current* ships complete with a sample set of very useful pre-defined connections, the real benefit to be gained from using the program will require more than a little while customizing the package to fit your unique needs. This is the single largest deterrent to using many PIMs, and *Current* is not an exception.

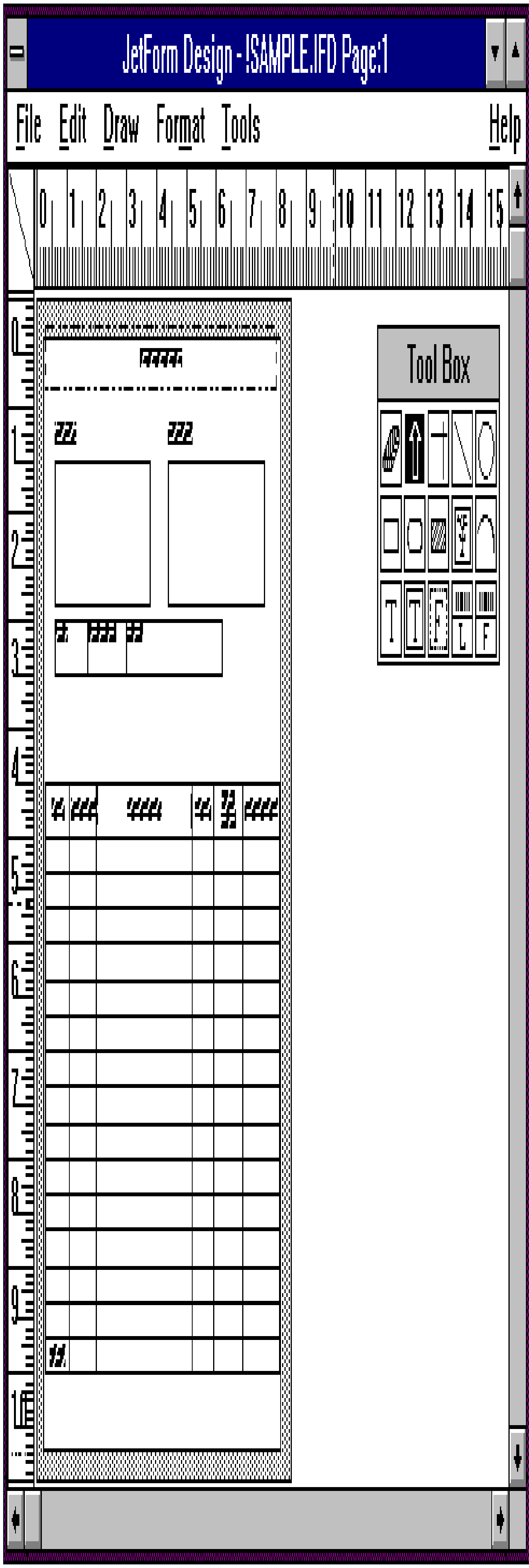
Both products have a few flaws, but while those found in *Your Way* tend to be related to documentation (nowhere is mention made of how to run a non-*Windows* program from a contact card, for example [use a .PIF file]), *Current*'s are operational. Like release 1.0, version 1.1 is completely stumped by relational date phrases containing the words "in a" (as in "in a month"); this is unacceptable in a product that is to compete seriously with *Agenda*, the master of such relationships. Further, *Current*'s calendar is based on computer era dates rather then real time, so the number year designation "'05" is deemed to come after "'98". Note one bug in *Your Way* that PRISMA has promised to fix before version 2.0 is released (perhaps as early as January): if you use nine digit zip codes and *Your Way*'s auto dialer, make sure you place phone numbers before zip codes on your contact cards, because the program interprets the first numeric string of seven characters or more as the phone number to be dialed.

This not a comparison of "brains versus brawn". The point we are trying to make is that choice of a PIM remains a personal decision. If you are looking for a supplement (or even replacement for) your flat file database program that will provide you with more meaningful, facile access to important information, you can't really do much better then *Your Way*. If your tastes closer reflect those of a project manager then a salesperson, however, *Current* is a better choice.

<u>JetForm</u>
<u>Indigo Software</u>

If flexibility is tantamount on your list of features important in a forms package, you'll have a hard time beating *JetForm*. You can use it to design forms from scratch using a wide ranging set of drawing, data entry and external communications tools, fill in forms manually or from formatted data files, and turn scanned images into forms via a tracing utility. Printed output can be generated from the main program, or by using either a *Windows* or DOS version of a utility program that prints and fills in forms that you've already created. Is there a drawback to *JetForm*? One that we see: all this potential doesn't come cheap.

Unlike FormWorx Corporation's *FormPublisher* (see review, this issue), *JetForm* requires a certain amount of artistic ability to use effectively. *JetForm* comes with only a few sample forms to help in the learning process, and although it works in a similar object oriented manner (due to *Windows*), drawing in *JetForm* is just that, while *FormPublisher* users create forms by placing predefined, complex objects on screen as selected from that product's menus. If you want to create an invoice form in *JetForm*, you must draw lines manually; there's a feature for easy duplication of objects, but you must lay the groundwork by hand.


JetForm Design - !SAMPLE.IFD Page:1
File
Edit
Draw
Format
Tools
Help
Tool Box

Which approach is better? Given total equality of all other issues - memory usage, speed, price, abilities and the like - we'd go for *FormPublisher* for the reasons found in the article on that product; it's easy to use, and it comes with an amazing collection of ready to use or modify forms. But things are **not** equal. *JetForm* runs many operations faster, uses superior printer drivers that operate outside *Windows*' strangle hold scope of control, and as easy as the other product is for novices to use, *JetForm* is ultimately a far more able design tool. If you use bar codes to track inventory or paperwork, *JetForm* even creates them automatically and places them wherever you specify with no more difficulty then positioning a straight line.