Introduction

As the cost of personal computers (PC's) has fallen and their computing power risen, it has become both practical and economically feasible to apply this technology to the traditionally pencil and paper-oriented task of recording clinical research data. Moreover, this application of the technology has become even more viable with the proliferation of very capable and affordable personal computer-based data filing and database management programs.

The object of the ensuing discussion, then, is to explore the issues involved in constructing a PC-based data entry application and integrating this application into an overall data management strategy built around SAS software. These issues are addressed in two groups. First, the relevant decision-making criteria are examined: What determines when a personal computer-based approach should be used as opposed to SAS/FSP? What factors should be considered in selecting the personal computer equipment and software? Second, schemes are discussed for (1) organizing the data entry, (2) moving the entered data from the PC's to a mini or mainframe computer, and (3) accessing and managing the data using SAS software.

Deciding When to Use PC's and When to Use SAS/FSP

In examining the relative merits of PC's versus SAS/FSP, one encounters three types of situations: those where SAS/FSP is the clear preference, those where the PC approach is the clear preference, and those where either might be acceptable.

SAS/FSP is clearly to be favored in those circumstances where all the data collected must be current, complete, and instantly available. For example, an application in which data collected were to be incorporated into a computerized patient medical chart system would fall into this category. Also, any data entry operation that requires current data to be checked against data from earlier in a time series would meet these criteria.

The PC-based strategy is clearly preferable under two sets of circumstances: (1) when the combined cost of the PC hardware, software, and telephone charges for data transmission would be less than the telephone charges for conducting the data entry entirely online and (2) when the data entry environment does not provide easy access to either power or communications connections. So, for example, under the first set of circumstances, if the clinic site were in France and the SAS host machine were in New Jersey, it would almost certainly be much cheaper to set up the French clinic with PC's rather than to arrange for online FSP sessions. Similarly, under the second set of circumstances, the personal computer approach would be attractive when the data must be entered at a remote location where the phone lines must handle a high volume of voice calls. Also, a battery powered laptop PC might be an attractive solution when collecting data in the field, say, for public health surveys, where there may be no readily accessible electrical outlets.

The remaining situations, the ambiguous ones, are both the most common and the most difficult to deal with from a decision-making standpoint. The difficulty arises from the need to consider relative costs, many of which involve intangibles such as time and system performance.

For SAS/FSP the costs concern both the mode of the data terminal communications connection and the impact of FSP sessions upon the performance of the host system. If the FSP-mandated full-screen data terminal requires a "hard-wired" direct coaxial cable connection to the host system, then the expense of the cable or the distance-induced degradation of the communications signal may prove too great. Similarly, if the data entry application requires multiple simultaneous FSP sessions, the cost, measured in the number of teleprocessing sessions and/or machine ports required, may be too great. System response-time is another potential consideration since a large number of concurrent FSP sessions may cause response-time degradation, especially on minicomputers. Also, on some systems response-time may already be so slow or erratic as to make online data entry impractical.

For PC's the cost is to be measured in the purchase price of the basic hardware and software, which, while the trend is downward, remains several times greater than the price of the average data terminal required for online entry.
Hence, deriving full value from PC investment can sometimes demand finding other uses for the machines during slack data entry periods, such as word processing, accessing online data bases or tracking clinic finances.

Selecting the PC Hardware

Once the decision has been made to adopt the PC-based strategy, there remains the somewhat daunting task of determining exactly what products, both hardware and software, to purchase. While there are arguments both for selecting the hardware first and for selecting the software first, in actuality the two selection processes are quite interdependent. Somewhat arbitrarily, then, the logic for the hardware selection process is examined here first.

Every machine has its champions, and, in fact, just about any personal computer can be adapted to perform satisfactorily for data entry. However, since the concern here is with economical, efficient data entry, it is clear that some machines are more appropriate than others. Among the traits that a data entry-oriented personal computer should have are the following:

- A letter and number orientation. Data entry is a keyboard intensive process, requiring the keying of letters and numbers. Graphics are not easily applied to the process.

- Adaptability. It should be an "open architecture" machine, one to which it is easy to add improved components and peripherals. Also, the machine should be one for which it is easy to write simple programs, using languages such as BASIC, to reformat or otherwise massage entered data.

- A large installed base. This is important to insure that future service and technical support will be available for the machine.

- A large memory and a fast processor. These two factors are related in that they both influence how well the machine will perform, especially with respect to response-time during data entry.

- High capacity auxiliary storage. Data entry, especially where a great deal of text material is involved, gobble up storage space rapidly. The machine must have enough capacity to store a reasonable amount of data—perhaps, a couple of days worth—before these data must be moved onto the SAS machine.

- A reasonable cost. For the personal computer-based strategy to be viable economically, the machine itself must be relatively low priced.

Of the personal computers currently on the market, the IBM PC and its compatibles are the machines that most completely possess these traits.

Having decided on the general type of personal computer to purchase, one must still configure the specific machine. The minimally configured machine should contain 256K memory and include a monochrome monitor and two floppy disk drives. For noticeably improved performance, the memory should be boosted to 640K and/or a hard disk substituted for one of the floppy drives. At least one PC from the set purchased must include a 1200 bps (bits per second) modem to be used for transferring the clinical data to the SAS mainframe or mini. Also, one inexpensive dot matrix printer should be acquired to assist in setting up applications and tracking the occasional data problem. In addition, some data entry software may require certain extra items of hardware, such as an expanded memory or a color monitor. Moreover, the nature of the data entry application may mandate other features, such as portability or the ability to run the PC off batteries.

Selecting the Data Entry Software

As mentioned in the introduction, the data entry software should be either a data filer or a database management program. The personal computer marketplace currently is awash with these types of products. The trick, then, is to make an intelligent choice from among the many. Among the features one should assess in selecting the data entry product are the following: user interface, data capacity, quality control, flexibility, data modification capabilities, and data export capabilities.

As anyone living in the Macintosh age knows, the user interface is a kind of Gestalt way of characterizing the look of a piece of software and the way it interacts with the user. Hence, a program has a good user interface if it is visually pleasing and easy to use. Visually, then, the candidate data entry software should be able to reproduce a screen facsimile of a standard paper data recording form—sometimes called a "painted screen"—into which the operator can enter the data. This feature is especially important if the data are initially recorded on paper forms, since the continuity between the paper and the screen images makes the data transcription process much less prone to error. As to ease of use, the
product should provide some mechanism, such as menus or a macro facility, that, to the greatest extent possible, frees the data entry person from the need to be familiar with the underlying functional details of the software.

The capacity of the software determines both how much and, in some instances, what types of data can be entered. Capacity is a significant consideration at the file, record, and field levels. At the file level, the number of records that can be maintained determines how much data the software can store. If this number is too small, only a subset of an application's data may be entered at one time, a sometimes critical limitation. At the record level, if the record size is too constrained, especially in the presence of a great deal of text data, there may not be enough space to store everything entered in a single screen form. This presents a primarily administrative problem since the application can always be partitioned into multiple forms. Similarly, at the field level, if the maximum number of fields permitted per record is too small—say, less than 255, a common maximum—there may not be a sufficient number to provide for all the items to be included on the form. Field capacity may also limit the kinds of data that may be entered. If the maximum size for text data fields is too small, it may, for example, be impossible to enter a complete transcript of an investigator's comments. Also, if the numeric fields are too narrow, it may not be feasible to define entry fields of the desired numerical precision.

The quality control features of any prospective data entry product, because of the critical purposes to which clinical data must be applied, are of paramount importance, both for individual values and for records. For values, the software must first be able to ensure that all required data—e.g., patient identifying information, data collection date, etc.—are entered for each record. Next, it must be able to guarantee that every entered value is of the proper type; i.e., date fields, decimal points in integer fields, etc. The minimum list of recognized types should include text, date, integer, and real. Also, a fixed point type, in preference to a real type, is quite useful for values requiring fixed decimal precision. In addition to type checking, the product should have a lookup facility that permits an entered value to be immediately validated against a predetermined list, defined in a separate screen entry form. An analogous feature of some data management products is a choice type, which, at data entry, actually displays the list from which the value must be chosen. For records, the software should ensure that the same data are not entered twice. A prescription accomplished by permitting a combination of fields to be designated unique, meaning they must contain unique values across all records.

The flexibility of the product is determined by the ease with which and degree to which an application may be modified once it has been set up. Of particular importance is the ability to modify a data field—for example, adding a decimal position or expanding a choice list—in such a way that none of the values already entered are adulterated or destroyed. Easy adaptability should also exist for the product's report generating facilities and data management utilities that inevitably come into play in preparing the data for transfer onto SAS minicomputer or mainframe.

The data entry software should possess data-modification capabilities, as any credible datafiler or database management system must. Such capabilities are required not only because data entry operators make mistakes but also because clinical data usually arrive in bits and pieces rather than in single coherent units. Consequently, the software should permit the retrieval of any previously entered record to the screen form, where the operator should be able (1) to change or delete exiting values, (2) to fill in heretofore missing values, or (3) to delete the entire record.

The data export feature, the mechanism by which the data must be converted into a format usable by SAS, is one of the key facilities of the candidate entry product. At minimum, the software should be able to convert each entered form of data into a series of fixed-field, fixed-length ASCII records. Once the SAS using column input. However, the fixed length export format can waste a large amount of space, especially if there are many blank fields in the data, a fact that may greatly increase the time required to transfer the data to the SAS mini or mainframe over the telephone lines. A better approach is offered by a delimited format (sometimes called a variable format) in which the values, represented in ASCII, are separated on the records by special characters called delimiters. Under this strategy, blank fields (missing values) are represented as two adjacent delimiters with no value in between. The result in most circumstances is a significantly compressed data file, one that can be transmitted over the phone lines much faster that its fixed format equivalent. Finally, other features being
equal, the product to choose is the one with the largest installed base. The reason, of course, as with hardware, is to insure that some level of technical support is available several months or years down the road after the data entry operation has become dependent upon the product. To obtain a reasonable best estimate for the installed base of a personal computer software product, one should examine its current sales volume, a piece information contained in the weekly Softsell Hotlist, a kind of best seller list for the industry. If the product in question appears on this list, it probably is a safe bet in terms of future technical support.

Selecting the Communications Software

The purpose of the communications software is to permit the transmission of the exported version of the data from the personal computer to the SAS mini or mainframe. The are two general types of communication software: terminal emulators and file transfer programs.

Terminal emulators are both the simplest and least expensive personal computer communications products. Most are available for less than $100--some can be obtained free or practically free as "shoeware"--and they will work with virtually any mini or mainframe. Their drawback is that the method of transfer they employ, simple character-by-character transmission as if the data were being typed in at a terminal keyboard, is extremely prone to telephone-line-noise-induced data errors. While the problem can be circumvented by means of laborious and time-consuming multiple transmissions of the same data, a more reasonable solution is to utilize a file transfer program.

File transfer programs are designed to transfer data between the personal computer and another computer with little or no error. They accomplish this feat by transforming the data into specially formatted units, often called packets or blocks, containing a redundant mathematical description of the data that can be checked on the destination end of the transmission to see if an error has occurred. If an error is detected, it oftentimes can be corrected on the destination end using this mathematical representation. If the error cannot be corrected, the sending program can request that the packet be retransmitted. Notice that, unlike terminal emulation, file transfer programs require complex coordination between the personal computer and mini or mainframe. This coordination is managed by a second program, running on the destination machine. This second program, or rather the cost of it, can make the file transfer approach financially prohibitive since programs for minis and mainframes are generally several orders of magnitude more expensive than programs for personal computers. Some of the newer file transfer products get around this problem by utilizing the protocol employed by KERMIT, a public domain data communications program found on many university minis and mainframes.

In addition to the actual mechanics of data transfer, the software should also be able to manage at least a portion of the intricate and tedious dialogue that normally must transpire during a mini or mainframe teleprocessing session. The simpler this dialogue can be made, the less skill the person effecting the transfer will be required to have. Generally speaking, even the most rudimentary terminal emulation programs permit a basic teleprocessing logon sequence to be stored and generated each time the program is run. Further, this capability can be easily added by employing any of the several inexpensive memory-resident keystroke-generator programs currently on the market. However, to fully automate this dialogue requires a script language, a feature only recently introduced for the top-of-the-line data communications products. The script language, as the phrase implies, permits a complete script of the teleprocessing dialogue to be specified in the software.

Organizing the Data Entry Application

There are infinitely many clinical research situations, each with its own peculiar data entry requirements. Hence, to illustrate the manner in which a PC-based data entry application should be organized, a representative example, in this case a FDA-type clinical drug trial, will be employed.

In a clinical drug trial, study participants randomly allocated to experimental or control groups, are assigned a therapeutic regimen built around the trial drug or a placebo, as the case may be. Then, at prespecified intervals during the course of the study, the participants are subjected to relevant batteries of clinical tests designed to measure the impact and efficacy of the prospective pharmaceutical product. From a very early stage in the research, the data from these tests must be provided to the FDA in the form of a series of very rigidly defined reports. And ultimately, of course, these results must be subjected to statistical analyses to assess the drug's true clinical performance. The data entry application, then, primarily involves entering the results gleaned from these
tests, along with relevant participant biographical information and certain FDA-prescribed clinic information.

The initial step in setting up the drug trial data entry application, as it is in any application, is to design the screen entry forms. Three entry forms are required: one for the test results, one for the participant biographical information, and one for the clinic information. The order in which the forms are constructed can be relevant, especially where data in one form must be looked up in another. A set of rules for this ordering can be derived from the pattern of form lookups. First should be constructed forms that neither require lookups themselves nor are referenced by lookups from other forms. Next, the rest of the forms must be divided into disjoint sets according to their lookup relationships. Within each of these sets the following strategy should be employed. Initially, the form at the top of the lookup hierarchy should be constructed, the one that performs no lookup itself. Next, the oldest child form should be constructed, the first that does a lookup into the top form. Next, the next oldest grandchild form should be constructed, the first that does a lookup into the oldest child form. The process should be continued in this manner until all the forms have been constructed. For the drug trial, the three forms fall into a single set, with the clinic form at the top of the hierarchy; the biographic form as the oldest child, requiring a lookup into the clinic form to validate the clinic code of each trial participant; and the test result form as the oldest grandchild, requiring a lookup to the biographic form to validate the participant identification code associated with each set of results.

Once the order of construction has been established, the creation of the individual forms may be undertaken. There are certain general guidelines that should govern this process:

- Each screen should be clearly organized and labeled, with wording and appearance corresponding as closely as possible to the paper form from which the data are to be transcribed.

- The most frequently entered fields should be located towards the top of the screen since, at data entry, the screen traversal pattern will be left to right, top to bottom.

- Identification variables—clinic code in the clinic form, participant identification code in the biographic form, and the combination of identification code and data collection date in the test result form—should be defined as required and unique, meaning that their values must always be entered and that these values may not be duplicated on any other records within the data file.

- No other variables except the identification variables should be designated required. Otherwise, the entire screen form must be filled in before a record may be saved, an unrealistic expectation in clinical research settings, where the data tend to arrive at irregular intervals.

The data entry strategy should closely parallel the hierarchy implicit among the screen forms. Calling once again on the clinical drug trial example, the clinic information must be entered first so that the clinic codes are available for validation when the participant's biographical data are entered. Also, if any clinics are added to the study, their data should be entered before any of the clinic's trial participants are defined or their test results keyed in. Next, the biographical data should be entered so that the cross-check for the participant identification codes will be available when the test results are entered. Again, new participants must be defined in the biographical data form before any of their test results may be entered. Finally, the test results may be entered.

Transferring the Data from the PC's to the SAS Machine

The first question that comes to mind when considering the potentially complicated process of transferring the data from the PC to the SAS mainframe or mini is simply: Why bother? The data are already housed in machines with the capability to print reports and to perform statistical analyses, so why suffer the additional headaches? The answer is that even though the small machines do possess all the basic capabilities required to fully manage the data, they simply do not have the capacity, even with the most expensive peripherals added, to adequately handle the report print load or the statistical analyses. This is especially true of the statistical analyses, which, due to the nature of their underlying mathematic algorithms, are inherently tailored to larger machines.

So with the why question resolved, attention turns to when: When should the data be transferred from the PC to the larger machine? The answer depends on the type of data involved. If the data are what could be thought of as study descriptors, such as those contained in the clinic and biographic
forms of the drug trial example, then the data should only be transferred after an update has occurred. If, on the other hand, the data are the focus of the research, such as the test results in the drug trial example, then they should be transferred either as they are needed—for generating reports, etc.—or as they fill the available PC-based storage.

With the why and when questions taken care of, one inevitably moves on to how. The manner of the transfer is essentially the same regardless of the type of data involved. The first step is to export the data, converting it from the format used by the entry software to the ASCII format accessible by SAS. Next, the exported data must be transmitted over the telephone lines from the PC to the SAS host using a modem and the communications software. Finally, once on the SAS host, the data must be integrated into a SAS data base. Also, at this point, for data other than study descriptors, their screen form files on the PC should be purged—wiped clean—to free storage so more data can be entered.

Managing the Data on the SAS Host

Because SAS software is so well-tailored to this type of application, the mainframe/mini computer portion of the systems design is by far the most straightforward. Two elements are required to manage the PC-entered clinical data using SAS: (1) a permanent SAS data base file and (2) a set of SAS programs to integrate the transferred data into this permanent data base.

All the data entered on the PC's and transferred to the mainframe or mini should be stored in a permanent SAS data base file created for that purpose. The SAS data base file should be employed rather than, say, a sequential file, not only because it permits all the data to be stored in a single location, but also because it simplifies, due to the programming facilities of SAS, report generation, statistical analysis, and utility operations, such as backing up the data to tape. Furthermore, because a SAS data base may contain within it many individual SAS files, the multiple entry screen structure of the data on the PC's may be easily reproduced. To create this data base file, one can simply issue an operating system level command. However, for efficiency's sake, it makes most sense to combine the data base creation with the creation of the SAS files corresponding to the various PC screen forms.

The programs to integrate the data into the SAS data base are logically clear: the data first must be read into a temporary SAS file using an INPUT statement; then this temporary SAS file must be used with the SAS UPDATE statement to update the correct SAS file in the permanent data base. The SAS UPDATE statement must be used so that observations in the permanent data base may be either modified or added to by means of the regular data entry process. By contrast, deletions, either of values or of observations, must be accomplished via PROC EDITOR or a SAS data step.

Conclusion

Clearly, the personal computer is a powerful tool, which, accompanied by the appropriate software, can be applied with great benefit to the task of clinical data acquisition. Moreover, when combined with mini- or mainframe-based SAS software, this technology can provide the basis for an integrated approach to the more encompassing task of clinic data management.

Robert W. Heinemeyer
President
FOREGROUND INC.
P.O. Box 6553
Cincinnati, OH 45206
(513) 559-1383

622