Microcomputers and SYSTEM 2000® DBMS

Paul Kent

SAS Institute, Inc.

SUMMARY

This paper discusses the integration of PC-based applications with larger data bases stored on mainframes. It addresses features already available in the SAS® System and SYSTEM 2000® DBMS, as well as future direction of the interface.

The functional requirements of a micro-to-host link are discussed, and features of the SAS System that make it useful as a link vehicle are placed in context. The benefits of the SAS end-to-end protocol that transfers both output and data are explored and compared with other approaches.

The SAS micro-to-host link supports a variety of communication protocols, with the SAS System, or SYSTEM 2000 DBMS executing on the (IBM, CDC or Sperry) host. The different methods of connecting to these host computers are discussed.

Potential applications of the link technology are examined, showing how one can integrate the features of the SAS System on the PC, with traditionally mainframe-based applications. One can use the PC as a host workstation, or choose to take advantage of the software features available on the PC.

MAJOR TOPICS

1. The functionality of a micro-to-host link
2. Features of the SAS System that make it useful as a link vehicle
3. Supported environments
4. Potential applications

DETAILS

The Functionality of a micro-to-host link.

There are many shades of micro-to-host links.

o Dumb start-stop terminal

A terminal emulation program provides features of some terminal hardware by emulating the same functions with software. (A package of this nature may emulate a silent 700 terminal.) Although the PC is capable of doing smarter things, this type of link software is not. For example, a program running on a PC can translate ANSI screen commands into those suitable for display on the PC, but cannot provide a scrolling facility since the terminal being emulated does not have this feature.

o Smarter dumb terminal

Here, software provides all the functions mentioned above, as well as take advantage of the nature of the PC. For example, the program may keep a session journal file that can be spooled to the disk. This provides a primitive file transfer solution, as the file could be "listed" and saved in the session journal on the PC. The program can also provide file-to-file transfer.

o PC as host terminal clone

Through hardware and software, the PC can be made to appear as if it were a 'real' terminal. Examples of this are the IRMA® card and software, or the Sperry UTS synchronous terminal emulator. This class of terminal emulation software often provides file-to-file transfer. The PC can often operate in "dual mode," functioning both as a terminal and PC, but there is no programmed connection between the two modes.

o "Transparent" connections

Unlike the other three types discussed, this class of micro-to-host link software provides the link capabilities to facilitate and/or enhance its local processing. This requires software on both CPUs to be connected with a common protocol.

The SAS micro-to-host link is an example of this -- SAS software is executing concurrently on both CPUs, while the link is established. Since the link is under the control of SAS software, it can be used to transfer output, input, and SAS data sets without the user having to overtly "change modes."

Features of the SAS System that make it useful as a link vehicle.

o SAS software provides excellent environment management.

The SAS System for microcomputers has an extremely comfortable user interface. It features pop-up windows, user-defined function keys, and a comprehensive macro capability.

o SAS software provides scrollable windows.

The window environment is well suited to 'ad hoc' sessions. It allows you to scroll backward and forward through previous output and messages, recall previous statments, and modify them for resubmission. The fact that all this scrolling is done locally on the PC means lower processing overhead on the mainframe and reduced transmission costs.

o SAS software includes a complete programming language that is the same on the PC and host.

The SAS language is a complete programming language that is compatible from mainframe to minicomputer to PC. The user can elect to process the same code on the PC, or if the size of the problem warrants, upload it to a remote host for execution. SAS software has an excellent reputation for data manipulation, transformation, and presentation abilities. The flexibility of the DATA step makes it an ideal candidate for "pipelining" information from one data source to another (from corporate data bases to PC-based spreadsheets, for example).

o The connection between SAS software and SYSTEM 2000 DBMS is the 'transparent' type.

The method for establishing contact is packaged into a 'script' by a person at your site. Logging on to the host system is easy -- the user types 'signon' and it happens, almost by magic! Not only is the method of connection transparent, but the user interface is too. The transmission of data sets is accomplished using the same tool as the remote execution of statements. And the method of reviewing results is exactly the same, regardless of where the statements were executed. Most knowledgeable computer scientists have been pleading for a consistent user interface -- the SAS solution fits this bill.

o The connection between SAS software and SYSTEM 2000 DBMS has an Error Detection / Correction scheme.

The communication between the micro and the host is via an end-to-end CRC checksum protocol. This means that a transmission that has been distorted (by line noise perhaps) will be detected, and the originating system will be asked to resend the information.

o SAS software has personalities that suit different applications.

The experienced user will be comfortable using the program editor to compose and submit SAS statements for execution. The novice may use the SAS procedure menu system to build 'ad hoc' requests until he becomes familiar with the syntax.

The 'I don't care to program' user can be accommodated, too. SAS/AF® software allows you to build canned applications using full-screen menus, fill-in-the-blank screens, and so on. The user merely has to make a selection, fill in a field, and his output can appear.

Environments supported.

The SAS System under PC DOS can connect with other Institute products using a variety of communications protocols.

o IBM - CMS and TSO, asynchronous, IBM PC3270 or IRMA® connections.

The link is available between Version 6 of the SAS System on the PC, and Version 5 under TSO or CMS. Version 5 includes a link to SYSTEM 2000 Release 11.5. The PC cannot execute full-screen procedures on the host.

The SASware Ballot® will be used to set priorities for supporting other modes of connection.

o Sperry - asynchronous connection

Version 6 on the PC will connect to a SYSTEM 2000 Release 4 session. The PC cannot have full-screen sessions with QueX™ , or Screen Writer™ , although this feature is under investigation.

The Institute is investigating a connection for Sperry PCs running the synchronous UTS protocol.

o CDC - Asynchronous connection

Version 6 on the PC will connect to a Version 4 SYSTEM 2000 session.

Applications

o PC as a host workstation

The powerful full-screen editor, scrollable windows and local file creation capablities of the SAS System for PC DOS make it a useful productivity tool when developing SYSTEM 2000 queries. The RECALL command makes it easy to 'try again,' and once the statements are perfected, they can be stored on the PC's disk to be used at a later date.

o Local data capture

PROC FSEDIT is a SAS/FSP® procedure that allows full-screen data capture and update. This can be used to enter and validate data on the PC, and submit it "en masse" at some suitable point. In many data capture environments this approach is convenient, as it reduces the mainframe involvement. Consider the plant quality control staff that collects QC measurements all day and sends them to a central site for analysis at the end of each shift. The local processing power of the PC could be used to capture the data and validate it on-site. At some "off-peak" time, the PC could make a connection with the host and transmit the SAS data set for further analysis.

One can develop truly distributed applications, where the validation tables are stored on the PC and refreshed each day. Complex applications can use the DATA step window feature to create custom data capture panels, and perform 'on-screen' validation, using the DISPLAY statement, and examining a SAS data set to see if the value entered is acceptable.

o SAS/AF® as a menu-building tool on the PC

SAS/AF software can be used to create "executive presentation systems." Suppose top management wants to select from a set of pre-written report formats from a menu. The system can be designed with flexibilty so that the manager can select a subset (such as department or fiscal period) using a menu. The manager can use this system without knowing any SAS statements.

o Local analysis

You can download information into SAS data sets on the PC, where further processing can be done at reduced cost.

On IBM systems, you can extract data into a mainframe SAS data set and then download it to the PC.

```
/*-- Create a temporary dataset --*/
/*-- on the IBM Mainframe       --*/
PROC S2K PW='DEMO' DBNAME=EMPLOYEE
         OUT=TEMP ACCESS=M;
   WHERE = 'WH C102 EQ MARKETING';
   SELECT C0 C100 C110;
   RUN;


/*-- And send it to the PC --*/
PROC DOWNLOAD DATA=TEMP
      OUT=PC.MKTINFO;
   RUN;
```

You can enter this code into the program editor window, and use the RSUBMIT command to remote submit the statements, or you can be copy them in from a local DOS file. A menu selection from a SAS/AF panel could generate the code, or it could exist as a member of a host PDS. (In the last case, one would submit '%include dd(memname);'.) As you can see, there is flexibility of where to store your programs.

On CDC and Sperry systems, where the SAS System is still under development, the SAS System for microcomputers connects with Release 4 of SYSTEM 2000. The DOWNLOAD command will create a SAS data set on the PC from values selected from a SYSTEM 2000 data base. The download process will create numeric or character variables and will adjust date values so that SAS processes them correctly.

```
DOWNLOAD/PC.DSNAME/
     EMPNO=EMPLOYEE NUMBER,
     LNAME=LAST NAME,
     PAYR=PAY RATE,
     EFFDT=EFFECTIVE DATE
   WH DEPARTMENT EQ MARKETING:
```

o File transfer

SAS software transfers and creates foreign files almost as well as internal ones. For example, the data set created by the previous example could be converted to a DIF file for processing by LOTUS 1-2-3® , or some other software.

```
FILENAME DIFFILE '\LOTUS\MKTINFO.DIF';
PROC DIF DATA=PC.MKTINFO DIF=DIFFILE;
   RUN;
```

TRADEMARK NOTICES

SAS, SAS/AF, SAS/FSP, SYSTEM 2000 and SASware Ballot are registered trademarks of SAS Institute, Inc., Cary, NC, USA.

QueX and Screen Writer are trademarks of SAS Institute, Inc.

IRMA is a trademark of Digital Communications Associates, Inc. LOTUS and 1-2-3 are registered trademarks of Lotus Development Corporation.

A footnote should accompany the first use of each registered trademark or trademark and should state that the referenced trademark is used to identify products or services of SAS Institute, Inc.