Cooperative Processing --
Making the Most Out Of the
PC - HP 3000 Connection

Bruce N. Smith
Hewlett-Packard Company
8000 Foothills Blvd.
Roseville, CA 95678

In the beginning...

In the beginning, the PC and the HP 3000 were separated by an impenetrable wall. When the personal computer was first used in the commercial environment, connectivity with departmental computers was non-existent. Any interaction between the two was typified by the accountant who copied needed numbers from a printed report into an electronic spreadsheet, then massaged them a bit, created some elementary graphics, and, if necessary, created transactions that were then entered manually into the HP 3000. File transfer between the PC and the departmental computer was virtually unheard of.

And then there were terminal emulators...

Finally, the wall started to crumble with the introduction of terminal emulators. The PC user was now able to use his PC for real work; as a terminal hooked to the host computer. Some users were knowledgeable enough about the databases that they used to select data with Query, write a report to an MPE file (without carriage control), copy the file to the PC using their favorite emulator software (AdvanceLink of course), and import the file into their favorite spreadsheet or database package. Again, after working with the data, they still had to formulate transactions for manual input on the HP 3000.

Today...

With the advent of "cooperative processing", the wall has now been torn down. Hewlett-Packard's Cooperative Services product (HPCS) gives the third generation language application designer the ability to create PC programs that:

- Directly read, write, and update HP 3000 TurboIMAGE data bases.
- Directly read, write, and update MPE files.
- Execute MPE commands
- Exchange data with host procedures stored in segmented libraries (SLs).
- Use local PC files, printers, and plotters.
- Access the host HP 3000 via OfficeShare LAN or basic serial connections.
- Use shared discs on HP 3000 and PC servers.

- Present a consistent PC user interface using MS Windows and NewWave or other company standard screen handlers.

- Allow user mobility by running from portable computers in the field or at home.

- Can be written in COBOL, Pascal, or C languages and run on the HP Vectra PC, the HP Touchscreen, the Portable Plus, or on the IBM PC/XT, PC/AT or PS/2 personal computers attached to almost any HP 3000.

What is Cooperative Services?

Cooperative Services is the only PC and HP 3000 software required to create PC applications written in COBOL, C, or Pascal that exchange data with the HP 3000 host.

The major components of the product are:

- The HP 3000 server process.

- The "Requestor" -- a terminate-and-stay-resident (TSR) PC process.

- A set of library modules that are linked with the application program.

- Conversion modules to handle the differences between character sets and numeric representations on the PC and the host.

Why would I want to write a cooperative processing application rather than a HP 3000 terminal-based application?

Cooperative processing applications are your best choice:

- If you have already written your PC application and you would like to have your program retrieve some data from the HP 3000 or write data to a host database, file or, possibly, send data to another user via HP Desk Manager.

- If your intended user base is already using PCs for word processing, graphics or with local databases and they would prefer not to use a terminal emulator program.

- If your users are geographically separated from the

HP 3000 and you want to reduce the costs of data communications by minimizing the connection time to the host.

- If you want to automate data transfers from field personnel who use portable computers to collect and validate data on the PC prior to transfer.

- To provide your PC community with a consistent user interface with tp data processing tasks. The most obvious case would be those who are used to MS Windows and, in the near future, HP's NewWave product.

Will Cooperative Services increase the performance of our HP 3000s?

In a well planned transaction processing application, where the data is stored both on the host and on the PC and access to the HP 3000 is kept at a minimum, cooperative processing will not degrade your HP 3000 performance. In some cases, performance may actually improve by removing the screen handling and data validation from the host machine.

Moving an existing application from the HP 3000 to the PC without changing where the bulk of the processing takes place, can actually increase the load on the host.

Several actions can be taken to reduce the load on the host:

- Do as much processing on the PC as possible. This may require keeping certain fairly static data bases on the PC for validation instead of going to the host to check each item on a data entry screen. These validation files may be refreshed on a daily, weekly or monthly basis, depending on the needs of your application. The files would contain data such as account and department numbers, part numbers, or customer information.

- Use Remote Procedure Calls (RPC) to do the heavier host processing. For instance, if a transaction requires reading through long chains and returning only a subset of all that was read, most of the work should be done in a procedure stored in the group or account SL. This way, the PC will only have to make one call to the host for information, instead of performing a DBFIND and many DBGETS. The requested data can then be returned in a single block.

- An asychronous process can be spawned on the HP 3000 with an RPC to handle transactions as time allows.

Many PCs could write complete transactions to a message file, and the host process would execute them on a first come first basis. Each PC would continue immediately, as if the transaction had been posted. This would also give the PC user the impression that he had the entire machine to himself.

Is HPCS hard to use?

Writing PC applications to access host databases is not a lot different than doing it on the HP 3000. Here is an example of opening a host database from PC COBOL:

```
MOVE "  HCSAMP.PPCDICT.HPOFFICE;" TO BASE.
MOVE "MGR;" TO DB-PASSWORD.
MOVE 5 TO DB-MODE.

CALL "DBOPEN" USING BASE DB-PASSWORD DB-MODE DB-STATUS
                    HCSTATUS.

IF HCSTATUS NOT = 0
MOVE "DBOPEN" TO HC-CALL
PERFORM HC-ERR THRU HC-ERR-EXIT.

IF COND-CODE NOT = 0
```

The only difference between this and host COBOL is the addition of the HCSTATUS variable. HCSTATUS was included to provide the application the status of the data communications between the two machines.

Since data and not just addresses are sent to the host during IMAGE and MPE file access, the first word of each buffer must contain the size of the buffer in bytes. For example, in COBOL this would be the way to define a structure for a DBGET:

```
WORKING-STORAGE SECTION.
01  ORDER-MASTER.
    05  LNGTH            PIC S9(4)
                         USAGE COMP-0.
    05  ORDER-NUMBER     PIC X(10).
    05  CUST-NUMBER      PIC X(08).
    05  ORDER-DATE       PIC X(06).
    . . .
PROCEDURE DIVISION.
    . . .
    MOVE 24 TO LNGTH OF ORDER-MASTER.
    MOVE 2  TO DB-MODE.
    MOVE "*;" TO DB-LIST.
    MOVE "ORDER-MASTER;" TO DSET.
    CALL "DBGET" USING BASE DSET DB-MODE DB-STATUS
                       DB-LIST ORDER-DETAIL DUMMY HCSTATUS.

    IF HCSTATUS NOT = 0
    . . .
```

It is also necessary to add code to make and terminate the connection with the host HP 3000. This entails providing the logon string and the server name for OfficeShare, or COM1 or COM2 for a basic serial connection. In COBOL using OfficeShare to connect to a HP 3000 named "SERVER" might look like this:

```
     MOVE "SERVER" TO CONNECTID.
     MOVE "MGR.SAMPLE" TO LOGON.

     CALL "CONNECT" USING CONNECTID LOGON HCSTATUS.
     IF HCSTATUS NOT = 0
     . . .

     MOVE 0 TO TRACING.

     CALL "STARTSERVER" USING TRACING HCSTATUS.
     IF HCSTATUS NOT = 0
     . . .
```

Disconnecting is just as easy:

```
     MOVE 0 TO DUMMY.
     CALL "STOPSERVER" USING DUMMY HCSTATUS.

     MOVE 1 TO LOGOFF.
     CALL "DISCONNECT" USING LOGOFF HCSTATUS.
```

Data Format Conversion

The remaining difference between HP 3000 and PC code is converting data between PC and HP 3000 formats.

Numeric Format

Numbers were never created equal. For example, when the personal computer was created, all integers were stored in 8-bit bytes, the low byte on the left then the high byte on the right. Even with today's 16 and 32-bit architectures, integers are still stored in the same manner. The HP 3000, on the other hand, writes data in 16-bit words -- high byte on the left and the low byte on the right, just the opposite of the PC.

Because of these types of inconsistencies the CONVERT function has been provided to convert to and from the following PC and host numeric formats:

Pascal and C Conversion Pairs

Host Format	**PC Format**
TurboIMAGE I, J	8086 1 Word Integer
TurboIMAGE I2, J2	8086 2 Word Integer
TurboIMAGE R2	8086 2 Word Floating Point
TurboIMAGE R4	8086 4 Word Floating Point
TurboIMAGE P	8086 2 Word Integer
TurboIMAGE K1	Unsigned Integer

Microsoft COBOL Conversion Pairs

TurboIMAGE P	Packed Decimal
TurboIMAGE Z	Zoned Decimal
TurboIMAGE K1	COMP-4

Micro Focus VS COBOL Conversion Pairs

TurboIMAGE P	Packed Decimal
TurboIMAGE Z	Zoned Decimal
TurboIMAGE K1	COMP (2 word integer)

Numeric conversion is necessary only when numbers are brought down to the PC from the HP 3000 or are being sent to the HP 3000 in a buffer. HPCS has no knowledge of the format of data in the program's data areas. HPCS does automatically convert some HP 3000 data, such as certain words of the TurboIMAGE status array and error numbers returned by FCHECK.

As an example, all of the TurboIMAGE intrinsics return information in the STATUS array. When the intrinsic call is successful, Cooperative Services automatically converts selected words from host to PC format. Typically, record addresses are not converted, because their value is of little use to the PC application.

If TurboIMAGE call is unsuccessful, only the condition code is converted to PC format, while the other nine words are left in host format for use by the DBERROR and DBEXPLAIN intrinsics.

Character Formats

Data on a host may be represented in either ISO-7 or Roman8 formats. Data on a PC may be represented in Roman8, IBM8 or ANSI formats.

It is difficult to make absolute statements about patterns of character code set usage. However, the Vectra family and IBM PCs are more likely to use the IBM8 character set. The HP Touchscreen typically to uses the Roman8 character set. Microsoft Windows uses the ANSI character set. Individual software packages exhibit variability in code set use.

If you expect your application to have international distribution, you may want to consider providing the following types of character set conversions:

Asian Character Set Conversion Pair

Host Character Set	PC Character Set
Host Chinese or Taiwanese	PC Chinese or Taiwanese

ISO-7 Character Sets Conversion Pairs

ISO-7 Host Format	PC Format
US Swedish/Finnish Danish/Norwegian French German UK Spanish Italian	Roman8 IBM8 ANSI
PC or Host Format	**Host Only Format**
Roman8 ANSI	IBM8
IBM8	ANSI

Conclusion

Cooperative processing is not for all possible applications. But if you are looking for a way to add value to your new or existing PC applications, to present a consistent PC user interface to your users and customers, or give mobility to a force of sales or service personnel, now is the time to consider creating cooperative processing systems.