
Telesoftware for the 1980's

Sense and Readability

Mike Brown
Videotex Development Officer
Council for Educational Technology

Telesoftware, that is, the automatic transmission of computer programs and data to microcomputers, promises to be one of the most important developments in software distribution of the last ten years not only for the personal computer hobbyist but also for educational and business users.

Prestel has provided the initial impetus through the establishment of a national computerised information service and has started to dissolve the psychological barriers to computer terminals in the home. Perhaps even more importantly, the advent of Prestel has caused a great increase in the availability of cheap modems, albeit in most cases for use only with viewdata systems. Consequently, many microcomputer manufacturers are busily hammering together viewdata interfaces for their machines, all of which have the potential for telesoftware reception.

In 1977 CAP/CPP made the first experiments with Prestel telesoftware for business users based on their Micro-Cobol operating system. In 1979 the Council for Educational Technology (CET) began work on viewdata telesoftware for education and published the 'Format Recommendations for Prestel Telesoftware' in October 1980. CET has also designed a viewdata telesoftware system for the Research Machines 380Z which will be available from the start of the school year. This system has been installed in several schools throughout the UK and CET are closely monitoring their use of 25 Computer Assisted Learning (CAL) packages available on CET's Prestel database (*211#). Recently, other Information Providers (IPs) have begun to plan Prestel-based program distribution systems.

An article by Peter Blower ('Prestel & Telesoftware, where do we go from here?', PC June 1981) made a comparison of three existing telesoftware formats and suggested a fourth. The article contained a number of inaccuracies and misconceptions which I would like to rectify briefly before moving on to a wider view of the evolution of a realistic telesoftware service on Prestel.

The document 'Format Recommendations for Prestel Telesoftware' was the result of a number of discussions between CET, Prestel, Commodore, Research Machines, Cromemco, ITT, BBC, IBA and educational software agencies and represents fourteen months experimentation by CET on telesoftware transmission. It is this format which is used by CET for all programs on its database and is likely to be used for those of the BBC (Computer Literacy Programme) and Acorn Computers. It was published (single copies available free from CET) so that other interested parties could benefit from CET's research.

The document simply defines an "envelope" inside which ANY data may be sent so long as it can be represented as a sequence of 7-bit characters. This includes ALL dialects of ALL high-level languages, ANY data files and ANY machine-code in hexadecimal notation. The interpretation of received data may be performed by ANY operating system and the quality of the interface with the user is limited only by the imagination of its designer. This protocol is designed primarily to maintain the integrity of the character stream between the IP and the receiving terminal, that is, to enable a viewdata system to deliver files efficiently as a sequence of discrete data packets whose contents are not interfered with by the transmission system allowing simple but effective error detection. Prestel only sends a carriage-return-line-feed (CRLF) after lines of 38 characters or less. Trailing spaces (perhaps valid program characters) are lost. Therefore, these spaces are transmitted as so that no CRLFs interfere with the character stream within a block.

Mr Blower's suggestion that such interpretation is specifically dependent upon a BASIC operating system and that most computers would find it too complex a task is obviously ridiculous. Just as ridiculous is the belief that sending programs to Prestel in such a form is a tedious, complex and expensive business: we have a simple program at CET which does the whole job completely automatically, checksums and all.

The ResD/GEC format has the peculiar page number access to each program block using only the 'a' frame of each page. This method is rather wasteful of logical space and, since the page is the logical unit of storage in a viewdata system, it is a page, as far as possible, which should contain a program. There is no greater flexibility to amend programs stored in this way rather than on continuation frames and neither is there increased security against noise corrupting the sequence in which blocks are received.

The IPC (Practical Computing) format provides no error checking, no packing to save space and reduces the character set available to the user. Mr Blower goes on to outline yet another format every bit as complex as CET's whilst still providing no packing.

In fact, taking the author's own definition, "... the methods of implementation should allow the receiving of software to be reliable, flexible, compact, efficient, inexpensive and as far as possible independent of machine or monitor ...", CET's format, when correctly understood, is the very epitome of good software distribution.

The existence of a computer-readable index of all programs on Prestel, suggested by ResD/GEC is very important. This would allow an intelligent terminal to locate automatically and pull down a program previously specified by the user. This index should be designed with great care and precision. It should hold details not only of filename, language, target machine and start page but also of size, cost, application area, etc. so that the user may use his/her intelligent terminal to search on a number of parameters for the software he/she wants. It would be appropriate that such an index (which may easily be superimposed upon CET's format) is maintained by Prestel alongside an alternative conventional hierarchical index through which potential customers may browse.

But let me move on now to what I consider to be the most disturbing of allegations of the telesoftware fraternity - to READABILITY. Readability is the basis of Mr Blower's own format suggestions and the main platform of his criticism of the CET and ResD/GEC formats. There is a school of thought which maintains that all information on Prestel, whether or not is it intended for computer consumption, should be capable of being read by humans and, as a consequence, any computer programs should appear as listings, neatly laid out, indented for clarity and even colourful! If in this form they accidentally happen to lend themselves to automatic retrieval by an intelligent terminal, then all the better.

This is not, I believe, the most sensible approach to telesoftware. There is no feasible compromise between readability and "reliable, flexible, compact, efficient and inexpensive" software distribution. There may well be a market on Prestel for readable program listings which can be scribbled down from the screens of conventional viewdata sets and typed into home computers. But this must not be confused with, or allowed to interfere with, the potentially much more powerful automatic distribution of software which is a directly computer-to-computer communication.

Many IPs may decide, quite rightly, that the small number of intelligent terminals around at the moment will not justify their providing an automatic software distribution system and that colourful listings are a much more lucrative venture. Others, like CET, are concerned with the development of intelligent terminals and with powerful facilities which can be superimposed upon national viewdata networks and therefore see streamlined automatic distribution as the more valuable avenue to explore.

There are now several parties around the country engaged independently in the field of telesoftware. The lesson to be learnt from the exchanges of the past few months is that a dialogue on a national level is essential, involving users, IPs and computer manufacturers as well as Prestel representatives, so that the development of telesoftware is discussed and directed rather than being allowed to take a number of diverse paths. The question of readability, having no bearing on telesoftware, would hopefully not feature in such discussions.

FORMATTING A FILE FOR TRANSMISSION VIA Prestel

Source
file

```
PROCEDURE POSTRAV(P);
WHILE P() NIL DO
  POSTRAV(LLINK(P));
  VISIT(P);
  P:=RLINK(P)
ENDWHILE
END;
```

Take any ASCII File (in this case a small POP-2 routine but may be any file of any size).

converted
source
file

```
PROCEDURE ¼POSTRAV(P); ¼L
WHILE ¼P() NIL ¼DO ¼L
  POSTRAV(LLINK(P)); ¼L
  VISIT(P); ¼L
  P:=RLINK(P) ¼L
ENDWHILE ¼L
END; ¼L
```

Insert ¼L at the end of each line.
Replace each space with ¼.
Insert ¼F at the end of the File.
(If any ¼ or ¼ appear in the source file replace them with ¼E and ¼¼ respectively.)

blocked
file

```
¼A
PROCEDURE ¼POSTRAV(P); ¼L
WHILE ¼P() NIL ¼DO ¼L
  POSTRAV(LLINK(P)); ¼L
  VISIT(P); ¼L
  P:=RLINK(P) ¼L
ENDWHILE ¼L
END; ¼L
```

Split the converted source file into blocks of not more than 859 characters. (This file consists of only one block.)

Insert ¼A at the start of each block and ¼Z at the end followed by a 3-character checksum.

```
¼F
¼Z 105
```

Remove all CR/LFs from each block. Each block occupies one Prestel frame

appearance
on
Prestel

```
¼APROCEDURE ¼POSTRAV(P); ¼LWHILE ¼P() NIL ¼DO
¼LPOSTRAV(LLINK(P)); ¼LVISIT(P); ¼LP:=RLIN
K(P) ¼LENDWHILE ¼LEND; ¼L¼F¼Z105
```