NETWORK WORKING GROUP
REQUEST FOR COMMENT NO. 448
NIC #13299
UPDATES: RFC 354, 385, 454

R.T. Braden
UCLA/CCN
February 27, 1973

PRINT FILES IN FTP

INTRODUCTION

Many of those who contributed to the definition of the FTP and RJE protocols have expressed varying degrees of uncertainty or unhappiness with the "print file" type in FTP. This RFC is intended to review the problem of print files in preparation for the forthcoming FTP meeting. Originally drafted on the basis of RFC 385, this RFC has been updated to reflect the terminology of the latest FTP document 454. (Changing the terminology doesn't solve the problem!)

It turns out that the Network RJE protocol as presently defined (see NIC 12112) seems to force a particular interpretation of print files in FTP and in fact, this interpretation is probably different from the one assumed by most FTP implementors.

VERTICAL FORMAT CONTROL

What is a print file? Presumably it is a file which is intended to be sent (eventually) to a printer process to create a hard copy. Many operating systems (particularly those which are batch-processing oriented) allow the programmer to include control codes within a file to be printed, to control the vertical format of the printed page--for example, single/double line spacing, overprinting, and page ejection. A "print file" is one which includes such vertical format control ("VFC") information. There are three common ways for printer processes to determine VFC:

CASE N: Non-Print File

The file contains no VFC information. The printer process applies a standard format (e.g., single space, standard vertical margins) if the file is printed.

CASE T: Print File with ASCII Format Effectors

The file is "Telnet-like", containing the ASCII controls CR, LF, and FF to provide VFC.

CASE A: Print File with ASA (Fortran) VFC

The first character of each record is a VFC code; see RFC 454 for the codes.

Assuming there are to be print files in FTP, these _three_ cases need to be considered. These three cases are explicitly included within the RJE protocol as "transmission" modes; we have borrowed the RJE labels N,T, and A from NIC #12112. The current FTP (RFC 454) seems to provide only _two_ cases: _unformatted_ and _print_file_. It is unclear from RFC 454 how these two FTP formats are related to the three VFC cases. For example, it is unclear whether the FTP format is meant to be a property of the file as transferred over the Network or of the file as stored by the server.

As I understand it, the Tenex system supports only case T. The distinction between Case N and Case T is not always clear, however. If a Tenex file which contains only the CR LF combination of format effectors is printed, it may be considered Case N where CR LF delimits a logical record, and where the standard format is to begin printing each record on a new line. The RJE protocol uses this ambiguity to advantage; see below.

The IBM operating systems, on the other hand, support Cases N and A. The "output writer" process which drives the printer must know whether or not a file to be printed contains ASA VFC, so the system distinguishes internally between "print files" (Case A) and non-print files (Case N). The "print file" attribute is normally attached to a print file when it is created. For example, the language processors typically create print files for their "printer" output streams.

Hence, when CCN's server FTP executes a STOR command, it must decide whether or not the new file is to be marked with the internal print file attribute. As noted earlier, FTP does not explicitly distinguish the three possible cases. We must either add some additional assumptions or server-dependent information outside FTP, or we must add a new format to FTP.

IMPLICATIONS OF RJE

To send an output ("printer") file to a user host, the RJE server will cause his FTP user process to send the file with the following attributes*:

*Note: Making the obvious conversion from RFC 385 to RFC 454 terminology.

VFC Case	FORMat	STRUcture
N	Unformatted	Record structure
	-	-
T	Unformatted	File structure
	-	-
A	Print File	Record structure
	-	-

Thus, the authors of RJE intended to use the _structure_ attribute to resolve Cases N and T. This is perhaps a reasonable choice, but we should understand the consequences and make them explicit within the FTP document.

Assume for the moment that we want to maintain perfect consistency between FTP and RJE. An FTP server which uses ASA VFC internally should convert _every_ (Unformatted, Unstructured) file it receives to an internal print file! That is, the file must be mapped into a set of physical lines (which are really logical records internally), and an ASA VFC character must be appended to the beginning of each line before it is stored. Note that this implies that the default file structure in FTP should be changed to _record_structure_. (This reinforces the point made by Wayne Hathaway in RFC 414 that if a Tenex user transmits a source file to an IBM host and expects to manipulate it in some useful way, he'd better send it with _record_ structure.)

ANOTHER CHOICE

If the loss of (unformatted, unstructured) as a simple default case is too offensive, we can simply change FTP to include three formats corresponding to Cases N, A, and T. RJE would be changed correspondingly.

ACKNOWLEDGMENTS

Discussions with Steve Wolfe, Jon Postel, and Eric Harslem were very helpful in clarifying the print file problem in FTP.

RTB/gjm

[This RFC was put into machine readable form for entry]
[into the online RFC archives by Alex McKenzie with]
[support from GTE, formerly BBN Corp. 9/99]