

PROTOD.TXT

PROTOCOL NOTES: CIS A PROTOCOL

by Bob Richardson, Carl Raff, and Scott Lowe

note: this material was not prepared for or by CIS and they have no responsibility for the accuracy or completeness of the contents. We gratefully acknowledge their cooperation and the super job they are doing, and simply wish to help document the file transfer protocol.

INTRODUCTION

The information presented here is intended to provide an outline of the written specifications for the CIS A Protocol. It is designed to be used by programmers who may need to debug or write programs to interface with Compuserve Information Services (CIS) terminal programs.

The PBM 1000 supports a communication subsystem based on the CIS protocol. One reason is that the CIS protocol is a step in the direction of software standardization and higher compatibility among different computer systems. The CIS protocol supports flexible file transfer and is practical for in-house use on a direct-connect medium speed network. PBM tests show that the protocol runs effectively at 19.2 KB and can transfer data at a rate of up to 50 KB.

In transferring files from one system to another, one CP/M based system (with a copy of any of the CIS executives that supports the A protocol) acts as a terminal. The other system, referred to as the "host," treats the "terminal" system both as a console and as the source or target of the file transfer protocol.

The following is a brief description of the CIS A Protocol used by CIS. Also included is a list of Executives compatible with it, and two new files which allow anyone with a CP/M remote system to support this protocol between CP/M systems.

Note: CP/M is a registered trademark of Digital Research.

The Executives

1.MNEXEC [Old Executive]

An older version of the Executive, MNEXEC supports ASCII transfer only. This executive is primarily useful for those who wish to buy programs from Softex, as its ESC I response is the only one that the Softex system currently recognizes as a file-xfer executive (see ESC I response). This program does not really support the A protocol, but rather a subset.

PROTOD.TXT

2.CSEXEC [The Compuserve Executive]

CSEXEC was the first "vanilla" version with binary support. It is a good executive to start with for those wishing to heavily modify the source because it suffers less from "rampant equate-itis" than BUFEXEC (i.e, it accomplishes tasks in a more general fashion without presenting too many options to the user). The current version seems to be modified for the Apple computer.

3.BUFEXEC [Buffered CSEXEC]

BUFEXEC adds memory buffered hard copy to CSEXEC, as well as many customization equates. The Osborne version is called "BUFEXO.ASM," and is identical to BUFEXEC except for modifications for the Osborne I.

4.IOBEXEC [IO/Byte Executive]

This executive is identical to BUFEXEC, except that it uses IOBYTE dispatch to access the serial port. If your system is interrupt driven, you must use IOBEXEC or modify one of the other executives. In addition, IOBEXEC allows use of cursor positioning.

PROTOCOLS

Characteristics of the physical protocol: asynchronous; 8 bits to the byte; 1 stop bit; no parity.

The protocol mode is turned "on" by an SI instruction (shift in) from host and terminal functions in protocol mode until an SO (shift out) instruction is received.

The ESC "I" Response: Each terminal responds to an ESC "I" with an ID string (see diagrams below).

The ESC "A" Sequence: This initiates a file transfer.

The ESC "L" Sequence: This starts a load sequence.

Notes:

The "." (period) is used to ACK a message (acknowledge and accept).

The "/" (slash) is used to NAK a message (negative acknowledge; request retransmission).

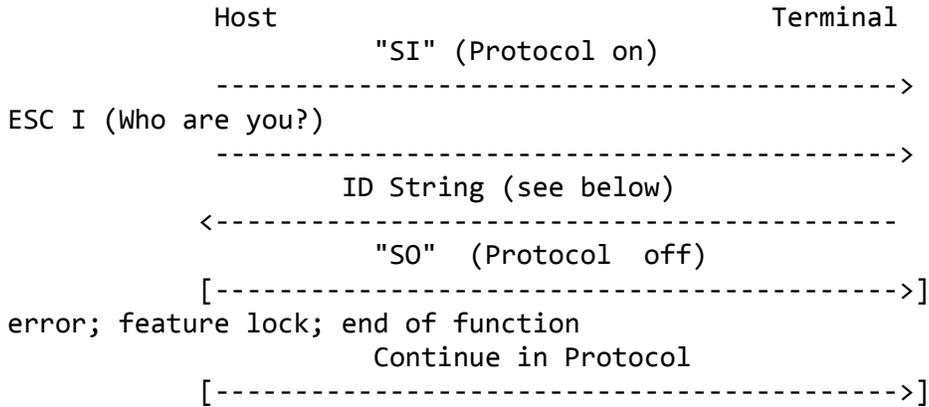
CTRL U is used to abort transmission.

Note: a list of fields enclosed in [] marks

PROTOD.TXT

indicate that 1 of the list will be transmitted, depending on the conditions at the time.

ESC "I" Sequence



Action: The terminal sends its ID string to the host. The host checks the features list, etc.

ID String: A full sample of the ESC I response for the IOBEXEC protocol (assigned by CIS) is as follows:

<#> <CPM> <MYSYSTEM>, CC, HC, PA, PL, <CR>

The first character is the protocol ID, followed by the OS name and the computer name (optional). The string of four pairs of characters are a series of features to be assigned if the system supports them:

- o CC, HC, PA, and PL <CR>
- CC = full cursor control
- HC = hard copy
- PA = "A" protocol
- PL = "L" protocol

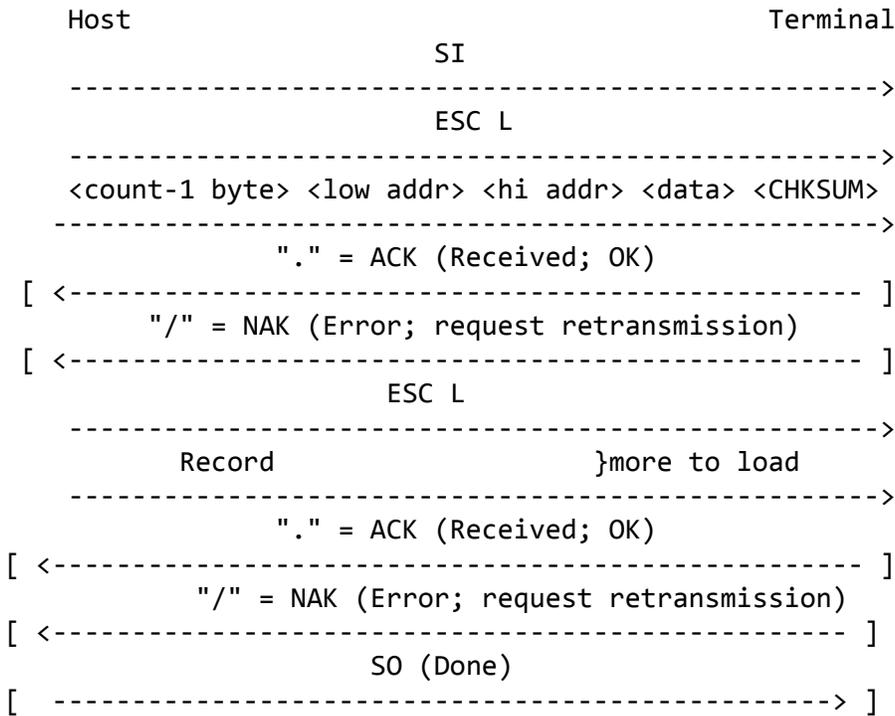
See the comments in any version of the executive for a more detailed discussion of these fields MNEEXEC is an older version of the executive, and so the ID string is slightly different in format than the one for the CIS Executive. A sample response follows:

<01>,<BF>,<SYSTEM2>,<&CPM>,<D> <CR>

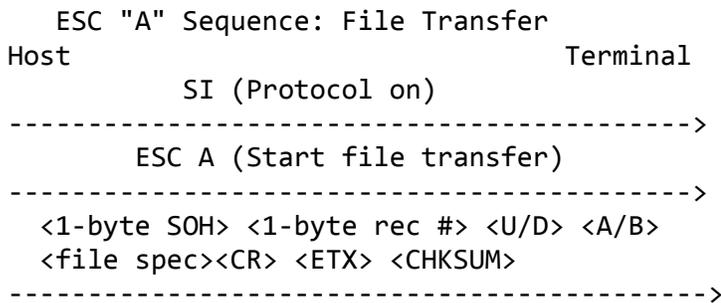
PROTOD.TXT

The first character is the page address of the lowest available page of memory in the terminal, followed by the page address of the highest available page in memory, the computer ID, OS name, and "D" for disk drives.

ESC "L" Sequence



Action: The terminal loads <data> into memory starting at address <low> <hi> for <count> bytes.



PROTOD.TXT

Note: U/D=direction; A/B=ASCII or binary; file spec=standard CP/M "." ACK (Received; OK)

[<-----]
"/" NAK (Retransmit)
[<-----]

Action: The terminal sets up to read or write the file given in <file spec>. Transfer will occur either until end-of-file or, if <A> (ASCII) is being used, until the first CTRL-Z. The file will be downloaded (transferred from host to terminal) or uploaded (transferred from terminal to host) depends on the contents of the direction byte (U or D). The protocol proceeds as follows:

ESC "A" Sequence: File Transfer

Download Subsequence

Host	Download Subsequence	Terminal
<SOH> <R/N>	<Text>	<ETX> <CHKSUM>
----->		
Note: R/N = Sequential record number		
"." ACK (Accepted record)		
[<-----]		
"/" NAK (Rejected; retransmit)		
[<-----]		
<SOH> <R/N>	<Text>	<ETX> <CHKSUM>
----->		
"." ACK (Accepted record)		
<-----]		
"/" NAK (Rejected; retransmit)		
[<-----]		
<SOH> <R/N>	<EOT>	<ETX> <CHKSUM>
----->		
"." or "/"		
<-----		
S0 (Shift out; protocol off)		
[<----->		
ESC A (Next file)		
[<----->		
^		
[Note: This is currently unused by CIS.]		

ESC "A" Sequence: File Transfer

PROTOD.TXT

Upload Subsequence Host	Terminal
	"." ACK
	----->
	<SOH> <R/N> < Text > <ETX> <CHKSUM>
	-----<
	"." ACK (Record accepted)
[----->]
	"/" NAK (Record rejected; retransmit)
[----->]
	<SOH> <R/N> < Text > <ETX> <CHKSUM>
	-----<
	"." ACK (Record accepted)
[----->]
	"/" NAK (Record rejected; retransmit)
[----->]
	<SOH> <R/N> < EOT > <ETX> <CHKSUM>
	-----<
	"." ACK (Record accepted)
[----->]
	"/" NAK (Record rejected; retransmit)
[----->]
	SO (Shift out; exit protocol)
[----->]
	ESC A (Next file)
[----->]

Field Definitions

A/B Indicates file type. An "ASCII" file ends in CTRL Z; a "Binary" file ends with EOT (CTRL D).

CHKSUM Check sum on record calculated on all characters sent after (and not including) start of header (SOH) and until (and not including) ETX. CHKSUM is calculated as follows:

1. CHKSUM <--- (CHKSUM * 2) mod 256
2. CHKSUM <--- CHKSUM + character
3. CHKSUM <--- CHKSUM + carry

If the check sum is less than hex '20', it is increased by hex '40' and sent masked (preceded by a DLE character).

EOT End of transmission (CTRL D). When EOT is encountered in text field of a

PROTOD.TXT

record, it ends the current file.

ETX End of text (CTRL C). ETX ends each packet (a group or string of related characters), and is followed by CHKSUM.

R/N Indicates the record number in ASCII characters 0 through 9. It is incremented by one ASCII number after each previous record is acknowledged (ACK). Length: 1 byte.

SI Shift in (CTRL O). SI starts the protocol mode.

SO Shift out (CTRL N). SO ends the protocol mode.

SOH Start of header (CTRL A). Each record begins with SOH. Length: 1 byte.

TEXT This indicates a variable length string, which is the information content of the packet.

U/D In an ESC A header, the U/D instruction determines the direction of transfer. U = transfer to host; D = transfer from host.

Character Masking

Certain control characters are increased by hex '40' and are preceded by a DLE character when encountered in the "Text" field of a message. These characters at this time are:

Name	Value
NULL	X'00'
SOH	X'01'
STX	X'02'
ETX	X'03'
EOT	X'04'
DLE	X'10'
NAK	X'15'

Note: These are the characters masked by the executive. CIS is currently masking almost all control characters.

An exception is the EOT character in the end of file record, which is sent unmasked. In general, masking is used to prevent confusion between a character used as data (masked) and a protocol character (unmasked).