July 10, 1992

Common Printer Access Protocol (CPAP) Specification

Status of this Memo

Abstract

The Common Printer Access Protocol (CPAP) is an application layer protocol designed to layer on any network protocol suite which supports multiple virtual circuits. It specifies the encoding of messages passed between a print service and a shared printer.

Digital Equipment Corporation
Maynard, Massachusetts

Chapter 1

Introduction

This is a Protocol Design Specification. It is a description of the Common Printer Access Protocol (CPAP). It specifies the form, function and operation of a public protocol which may be used to control a network printer.

The protocol is intended to be a superset of one originally developed at Digital Equipment Corporation's Western Research Laboratory in Palo Alto by Brian K. Reid and Christopher A. Kent. Special thanks to them and to Alan Guenther, Tom Hastings, Ajay Kachrani, Tom Powers and Eric Rosen who were members of the DECprint Printer Access Protocol Architecture Committee. Credits are also due to Jay Martin and Steve Henderson whose implementations helped refine this specification, and again to Ajay Kachrani who provided Appendix D. Thanks also to Bob Tedford and Glenn Trewitt who provided editorial assistance.

Chapter 2

Summary

CPAP layers on any communication facility which supports multiple virtual circuits. Such network protocols currently include, but are not limited to, TCP/IP and DECnet. The encodings specified herein may also be used for interprocess communication to or within a print server.

Chapter 3

Environment

CPAP is used for communicating with shared network printers. It defines the communication passing between print clients and a printer. It is currently used on UNIX, OSF/1, ULTRIX and VMS systems. The use of CPAP by other clients is encouraged. Servers are currently supplied by Digital Equipment Corp., but other vendors will be providing compatible printers.

Chapter 4

Terminology

The following defines certain terms which are used within this document:

Sheet	Any media which is fed through a printer as an entity
Page	An image such as might be found in a book (Several pages may be printed on a sheet.)
PDL	A Page Description Language (such as PostScript)
Document	A sequence of pages which begins with a known default printer state
Job	A collection of documents processed consecutively
Session	Any sustained connection using a channel
Unit	One of a server's physical printers
Channel	An independently flow-controlled communications path between two processes
Supervisor	The client process which supplies jobs to the printer. A printer supervisor may also be referred to as a symbiont or printer daemon.

Chapter 5

Protocol Features

CPAP supports connections to printer supervisors (daemons), supporting hosts (management clients) and remote console clients. The protocol contains no device control capability; it is designed to be non-printer specific. This document separates features into two sets. They are the base protocol set which is designated Level I and an enhanced protocol set which is referred to as Level II.

5.1 Protocol Level I

The base protocol set is identical to that used for the original TCP/IP PrintServers (the Reid/ Kent protocol). It supports the following features:

- Establishes session and job identification between symbiont or daemon and a printer.
- Supplies job synchronization to a printer from such a print service.
- Provides for deletion of the current session on a printer.
- Supplies sheet and page accounting to the print service.
- Provides for exchange of data between the print service and a printer.
- Provides for centralized accounting and client authorization.
- Provides for centralized error logging.
- Supplies limited file service to the printer.

5.2 Protocol Level II

The enhanced protocol set includes the following additional features:

- Encodings to identify protocol and client software versions.
- Additional accounting fields.
- Provision for identification of the page description language when multiple syntaxes are supported.
- A method for a printer supervisor to solicit printer status.
- A mechanism for reporting printer options, resources and available media.
- The ability to delete a specific print job within a session. This is required to unambiguously identify which job is being deleted when jobs overlap.

- A method for categorizing data returned from the printer. It is now possible to distinguish user data from returned status from error conditions.
- Provision for translation of messages to the language of the receiver.
- Version identification for resident prologues and other resources.
- Provision for font file access.

Besides adding features which address limitations of the base protocol set, Level II provides a new method for transmitting final-form data to the printer. This method eliminates protocol headers by specifying a separate data channel so final-form data may be transmitted directly to the printer’s interpreter.

Chapter 6

Protocol Characteristics

CPAP is a multichannel protocol. The multiple channels are provided by the use of a virtual circuit service which supports connection and disconnection of circuits and the passing of messages between cooperating processes. The mechanism used for establishing these circuits is not part of this specification. It depends upon network services provided by the local system. For example, if TCP is the underlying transport, each channel is a separate TCP connection. For DECnet, each channel is a separate logical link. Identified channels are as follows:

6.1 Control Channel

The Control Channel establishes session and job identification between a print service and a printer, supplies job synchronization, provides for deletion of a print job, supplies image and page accounting to the print service and permits the printer supervisor to obtain printer status. Level I clients also supply final-form data on this channel. Use of this channel for sending data to a server's interpreter is discontinued when using Level II. Initiation of the Control Channel connection is by the printer supervisor. On network printers, multiple Control Channel connections are permitted subject to local administrative policy. The printer may spawn separate processes to service each connection.

6.2 Data Channel

The Data Channel provides transparent communication between a print service and a printer for the purpose of presenting the user's document. Any channel multiplexing is left to lower protocol layers. This channel is provided as a part of the Level II protocol. It transports data from the printer supervisor to the printer's interpreter without any headers or additional protocol. Print job identification is supplied on the control channel, and the data channel is opened by the printer supervisor for the purpose of transferring a single document to the printer. The printer must grant permission to the supervisor before the data channel may be opened, and the printer establishes the port to be used for the connection. Tokens are exchanged for this purpose. The data channel is disconnected after the transfer of each document.

6.3 Management Channels

Management channels provide services to a printer. There may be one or many management channels employed each offering a variety of services. The primary function of a management channel is for logging printer errors. Additional (optional) features of these channels are as follows:

- ACCOUNT - Centralized accounting and authorization.
- CFREAD - Configuration file service for a printer.
- FILEIO - Limited Read/ Write file service for a printer.
- FONT - Translation of font names and caching of font files.

These features are assigned to separate channels to permit such printer server functions to be off-loaded from any given printer supervisor and to permit centralized error logging and accounting. Management functions are accomplished by passing ordinary strings between the Management Service and the Printer. Enhancements to the Level I management functions provided in Level II include:

- Expansion of FILEIO to permit access to both text and binary files.
- Addition of the readfile operator for rapid access to entire files.
- Loading of resident preambles, forms and fonts.
- Random access to fonts.

Chapter 7

Conforming Implementations

Since an existing TCP/IP PrintServer protocol (Reid/Kent) defines the Level I protocol set, existing client implementations conform subject to the following restrictions:

- Job overlap is not permitted. A session must end before a new job can be issued by a client.
- Level I clients do not support translation of messages arriving from the server.
- There is no separation of a user's messages from those of operator interest.
- Errors in printer operation are not distinguished from other messages.

New servers must support the data channel and other Level II encodings.

All printers must either support error logging on the management channel or supply local logs.

Printers which support multiple sessions from varied supervisors must use the management channel CFREAD file service to obtain authorization.

Chapter 8

Record and data formats

Records are free-form strings of ASCII/ANSI characters. Eight-bit characters using the ISO 8859-1 Latin 1 encoding standard are permitted. Except for information delivered on the Data Channel, the beginning of each record is marked by a one-byte synchronization character (002, control-B), and the end of the record is determined by the Length field of the record. The record format is as follows:

```
<Record> ::= <sync byte> <Opcode> <Id> <Length> <1space>
             <Data> <ignored>
<sync byte> ::= control-B
<Opcode> ::= <decimal digit>  |  <Opcode><decimal digit>
<Id> ::= <Integer>
<Length> ::= <Integer> 0 <= Length <= 1024
<1space> ::= exactly one space character
<Data> ::= <arbitrary (context dependent)>
<ignored> ::= <arbitrary>
<integer> ::= <decimal digit>  |  <integer><decimal digit>
<decimal digit> ::= ASCII Character 0 through ASCII Character 9
<arbitrary> ::= <any character (0-255)>  |  <arbitrary> <any character>
```

The Data can contain embedded newline characters or any other character.

Except for the Data field, which is preceded by exactly one space, fields are delimited by one or more space characters (octal 40). Other characters that resemble spaces, such as tabs, are not treated as delimiters. The Data field can contain space characters; it is bounded by the Length field and not by a delimiter. The Opcode field determines the type of command or data contained in the record. The Id field contains a record identification or sequence number that may be used, if needed, to identify replies to multiple outstanding requests. The Length field designates the number of bytes in the Data field. There is exactly one byte of white space after Length and before Data. This permits leading blanks to appear within the Data. The last character of Data is determined by the value of Length. All characters that come after the last character of Data but before the next sync byte are ignored, but an implementation is free to consider this situation to be a bug warranting a non-fatal warning message.

When the Data field is used to pass more than one string argument, the multiple arguments are combined in a format called a "list of values". Such a list resembles a UNIX environment vector, or, alternatively, the "keyword parameters" often found in command languages.

A list of values is a series of entries of the form A=B, in which the entries are separated by 001 (control A) characters. Each sub-string consists of a parameter name, an "=" character and then the actual value. For example, the Data string for a "write to file" opcode might be

```
    HANDLE=1234^ACOUNT=13^AOFFSET=0^ADATA=Test message\n
```

This string defines values for the parameters HANDLE, COUNT, OFFSET, and DATA. Note that all values are strings. Unless otherwise specified, all numbers are 32-bit decimal integer encoded strings. Since a length field is used to specify the end of the string record, NUL termination is not required.

The use of a control A character may not appear within a parameter. Should a parameter appear more than once in the list, the most recent applies. Every value in the list must have the "=" character: A= is a legal item, but A is not. When a value is itself a list, tokens or values within the list are separated by commas. In some cases, parallel lists may be used to enumerate values which vary depending upon the number of similar entities such as PDLs, units, input trays, etc.

Parameters unknown to a given level or version of the protocol are simply ignored.

8.1 Control Channel Protocol

A client symbiont or daemon communicates with the printer over the control channel. The connection is established by the printer supervisor to a well known port on the printer. For TCP/IP, the port used is 170. For DECnet, this is object 50. A connect message of "PAP" is used to identify the protocol in a DECnet environment.

A simple text-oriented protocol is used. Some client-to-printer messages require replies, and some do not. Unless otherwise indicated, the opcode strings listed are supported on all levels and versions of the protocol.

Table 8–1 lists the opcode strings and their symbolic names. The use of an asterisk (*) designates Level II protocol features.

Table 8–1: Control Channel Protocol Opcodes

Opcode Symbol	Description
"0"	Null. Do nothing. No reply is expected, and all of the fields are ignored. Some implementations may use this opcode to probe the printer in order to determine that the control channel circuit is still active.

Table 8–1 (Cont.): Control Channel Protocol Opcodes

Opcode	Symbol	Description
"1"	ssn	Begin a new print session. Initialize everything. Reply required. The data array is a list of values as follows: • SESSIONID is the printer supervisor system's identifying string. • HOST is the string name of the the client. This is the name of the host making the actual link to the printer. • *CLIENTID a string identifying the version of the software running in the client; its only use is for information. • *PROTOCOL the protocol version of the client. The absence of PROTOCOL within the data array may be used to identify a Level I client. If the server has more than one physical printer attached, then the client must also specify the following: • *UNIT the unit number of the desired printer. The reply is a list of values containing information about the printer and its session number, as follows: • JOBNO the numeric string name which will be associated with the next print job. Jobs are numbered so as to permit inquiries about job status and to permit unambigous job deletion. • SERVERID a string identifying the version of the software running in the printer; its only use is for information. • NODE the network host name of the printer; this can differ from the name of the printer. • *PROTOCOL the protocol version of the server. The absence of PROTOCOL within the data array may be used to identify a Level I server. • *PRINTERTYPE the model designation for the printer • *PDLS a list of the Page Description Languages supported. Refer to Appendix C for a list of recognized interpreters. • *MEDIA a list of Media sizes which may be selected. The encodings for media size strings are specified in Appendix B. If the server is not willing to accept the connection, as for example it has not finished its initialization, it should respond with a nak opcode rather than by rejecting the connection or by waiting until it is ready. There is no "End of session" command; rather, the control channel network connection is closed to indicate the end of a session.
"2"	eoj	Wait for the completion of a print job. The client is not obligated to wait for a reply, but if it closes the connection it will not receive error messages about paper jams and the like. Error messages that originate in the printer's interpreter will be sent back to the client before a reply to an eod, but error messages that result from the paper transport mechanism can still be produced by the printer even after a job has ended. If a eoj command is issued, then when its reply is received, the client can be confident that every page that has been imaged has also been delivered to an output stacker or bin.

Table 8–1 (Cont.): Control Channel Protocol Opcodes

Opcode	Symbol	Description
		The reply is a list of values containing information about the printer resources used since the last soj command, as follows: • PAGES the number of images produced by the selected PDL. • *MEDIA a list of Media sizes currently in each input tray. The encodings for media size strings are specified in Appendix B. • *SHEETS the count of the sheets of paper (or other media) used for each input tray. • *WASTE the number of sheets wasted due to jams or other difficulties for each input tray only if any were wasted. • *TIME the amount of CPU time used by the printer's interpreter.

The values for MEDIA, SHEETS and WASTE (if any) are implemented as parallel lists.

Opcode	Symbol	Description
"3"	sod	Start of document. A document is a sub-part of a job. All of the jobs in a session come from the same client host, and they will be printed together without interruption. Each document is printed with a clean copy of the printer's interpreter. The optional data array is as follows: • *PDL the page description language to use. If the PDL is not specified, a PostScript interpreter is assumed. No reply is returned to Level I clients. For Level II, sod is used to signify the need to open a data channel. Permission for the client to do so is signified by a reply when the specified PDL is available. The reply has the following list of values: • *DOC the numeric string name assigned by the server to this document. • *PORT a string identifying which port will be monitored by the printer to accept the incoming connection. This is a numeric token which is mapped to the lower protocol levels by mutual agreement between the print service and the printer. It is supplied to eliminate any connect delays which might be encountered due to reuse of the same port when using TCP or other similar protocols which do not immediately permit reuse of the same port. For TCP, ports 1024 through 1027 correspond to tokens 1 through 4. For DECnet, port replies map to Objects 128 through 131. The number of token values may be extended to include all user ports.
"4"	eod	End of document. When used in conjuction with the Level I protocol, this command indicates the end of a data stream which has been sent as a sequence of data commands. It also is a solicitation for a reply indicating the resources which will be used in processing the print job. The reply is identical to that for the eoj command, but no client should use the data supplied in the reply for the purpose of job accounting. The server does not guarantee that the number of sheets indicated will actually be delivered. The purpose of the reply to an sod is to permit a client to print a job trailer which contains job statistics. After the reply to an eod command, the client should either continue the job by issuing another sod command or issue an eoj command in order to obtain confirmation of job completion.

Table 8–1 (Cont.): Control Channel Protocol Opcodes

Opcode	Symbol	Description
"5"	data	Ordinary data, such as text or interpreter output, that is sent to or from the printer. Reply not required. Level I clients may start sending data immediately following an sod without waiting for a reply. If the printer is not prepared to handle the job data, it will block the transmission at the network level and the client will hang on its "send" command until the printer permits transmission to resume. Ordinary data is not sent to the printer this way when using Level II. The data channel is used instead. However, the data opcode is still used to return user data to the client in both Level I and Level II implementations.
"6"	kill	Kill a print job. This opcode is generated by the client software in response to a client-side abort request. The client is encouraged to send a kill opcode following TCP urgent data or to use a DECnet expedited message for the kill request. This ensures that the server can process the abort request even when restrained from receiving additional data. Level II clients may optionally specify either or both of the following in the data array list of values: • JOBNO is the job number which was returned as the reply to the ssn opcode. • DOC is the document number string returned as a reply to the sod opcode. If neither JOBNO nor DOC are present, the printer's current job will be terminated. Boundary conditions may exist due to pipelining, so specifying a specific document is recommended. Following an abort request, the client must wait for a reply before disconnecting the control channel. The reply is the same as for an eoj command, and it may be delayed until the printer has flushed all pages in progress. The printer may send the client a kill opcode in the event of a console generated abort, but not in the event of termination in error of an ordinary print job. Should the client receive such a kill, the current document should be aborted by closing the client's data channel connection (Level II) or issuing an eod (LEVEL I).
"7"	soj	Provide user-identification information to the server. The data array is a list of values, as follows: • USERID is the string name of the user whose file is being printed. • SESSIONID is the name of the print file or print session. • HOSTNAME is the name of the host computer from which user USERID printed the file. • NOTE is a commentary that will appear on the server's console and in its log files. On some systems, the print jobs of multiple users are batched together by the line printer daemon or symbiont into a single session. When this happens, the client software sends the server an information packet whenever the user identification changes. This act causes both the client and the server to increment their job numbers. Sometimes the components of the user identification are not all changed at once; any field whose value is unchanged may be left unspecified in an soj command.

Table 8–1 (Cont.): Control Channel Protocol Opcodes

Opcode	Symbol	Description
"8"	eof	Some implementations of TCP have difficulty handling an unannounced closing of a TCP connection. The eof opcode is an advisory-only packet that, if sent, is a notification to the server that the next action on that connection will be to close it. The eof opcode may be completely ignored on any system with a properly-working TCP implementation.
"9"	flush	Some implementations of TCP can be run more efficiently by batching data records into larger blocks, to reduce the number of system calls. The client should be free to do so. If clients are batching data records, they are advised to send a flush opcode at the end of each print job, to assist the server in recognizing the end of the job. The server is free to ignore this opcode if it so chooses. In general the flush opcode is useful only if the server is actually a network relay to the true server.
"10"	*show	Level II servers permit the client to solicit status from the printer. This opcode may be used outside of a session. The data array may optionally specify the following: • UNIT the unit number of the desired printer if the server has more than one physical printer attached. If UNIT is unspecified, then the status of unit 1 will be returned. • DATA an optional parameter name for the parameter to be shown. If no parameter is specified, then the reply will be as follows: • UNITS the number of physical printers attached only if the server supports more than one. • STATE the current operational state of the printer. • CLIENTS the number of print services queued to the printer. • JOBNO the current job being interpreted. • DOC the current document being interpreted. • TIME accumulated CPU time for this document. • PRINTERTYPE the model designation for the printer. • PDLS a list of the Page Description Languages supported. Refer to Appendix C for a list of recognized interpreters. • OPTIONS a list of the optional features installed on the printer. The encodings for option strings are specified in Appendix A. • MEDIA a list of Media sizes which may be selected. The encodings for media size strings are specified in Appendix B.

Table 8–1 (Cont.): Control Channel Protocol Opcodes

Opcode	Symbol	Description
"11"	*showpdl	Level II servers permit the client to solicit the characteristics of a given PDL. This opcode may be used outside of a session. The data array may optionally specify the following: • UNIT the unit number of the desired printer if the server has more than one physical printer attached. If UNIT is unspecified and more than one printer is attached, then the status of unit 1 will be returned. • PDL an optional PDL name in the format of the response to the show opcode. If no PDL is specified, data for all PDLs is returned. The reply is a list of values of the format: • name=id,version where name is the PDL name, id qualifies the implementation and version is an identifying string for inforamation only. An example is: PS=L1,Adobe PostScript V48.3-37. The example indicates that the PostScript interpreter is an Adobe level 1 implementation. The encodings for other recognized PDLs are specified in Appendix C.
"12"	*showres	Level II servers permit the client to solicit which optional resources are loaded into the printer for a given PDL. This opcode may be used outside of a session. The data array may optionally specify the following: • UNIT the unit number of the desired printer if the server has more than one physical printer attached. If UNIT is unspecified, then the status of unit 1 will be returned. • PDL an optional PDL name in the format of the response to the show opcode. If no PDL is specified, PS is assumed. The reply data array is as follows: • RESOURCES a list of named resources • RESOURCETYPES a list of the type of named resource (prologue, form or font) associated with RESOURCES • VERSIONS a list of the version strings associated with RESOURCES. Resources need not necessarily be identified as to RESOURCETYPE or VERSION. NULL strings are used if no identification is provided. The server responds with nak if no optional resources are availible.
"101"	repl	Reply. This record is a reply to the record whose ID it shares. The contents of the data array depend on the command being replied to. If the text of the reply requires more than one record, then all but the last record must have opcode 102, prepl.
"102"	prepl	Beginning of a multipart reply. If the amount of text for a reply (for example, a "show queue" command) is too big to fit in one record, then all records but the last must have opcode 102. When a 101 reply is received, the reply is taken to be complete.
"103"	nak	This is a "negative reply". The recipient of a record has rejected the request made by the sender. The sender is expecting a repl opcode, but if it receives a nak, it will withdraw its request. The data field is a text string explaining the reason for the rejection of the request.

Table 8–1 (Cont.): Control Channel Protocol Opcodes

Opcode	Symbol	Description
"104"	*status	This opcode has been deprecated. New clients and servers need not implement it. In the past, this data record was used for sending private messages to device control components of VMS symbionts. It had been used for decoding of messages from printers which implement the PostScript returnstatus operator only. This operator provides a way for certain PostScript printers to return numeric tokens to the client rather than strings. Most printers parse string information returned via the data opcode instead. The data format was as follows: • CODE a message code which uniquely identifies the message. See the msg opcode for the interpretation of this string. • ARGUMENTS a list of supplemental token strings. Unlike the msg opcode, no text is provided. This is because the message had no predefined meaning in the server's context.
"105"	*msg	This is a data record private to the supervisor components of the symbiont/daemon. It provides a way of distinguishing between messages which are private to the print job from those of administrative interest. Such information includes out-of-paper, printer is jammed, etc. The data array is as follows: • CODE a message code which uniquely identifies the message. This field is supplied to permit messages to be translated to the language of the receiver. It is the value assigned to a specific token string by agreement between the client and the server. Specific values are printer specific, and they are not contained within this specification. This code need not be used if the language of the sender is acceptable to the receiver. However, when converted to a 32-bit integer, the low-order three bits of this code may be used to determine the severity of the condition being communicated. The encoding is as follows: 0 - Normal completion of some process 1 - Information for logging only 2 - Warning of an unusual condition 3 - Error which affects the current document 4 - Fatal error which prevents use of the printer Receipt of a message with a severity of 3 or 4 may affect the client state. There is no point in continuing the current document if a message with a severity of 3 is received during that document. A message with a severity of 4 affects the entire session. • ARGUMENTS a list of supplemental token strings. • TEXT the actual fully formatted message in the language of the sender. When communicating with a client which is using Level I encodings, the text is sent using the data opcode and the msg opcode is not used. Such clients are not able to perform language translations, they cannot separate the user's data from adminitrative messages, and they cannot determine the severity of any error associated with the message.

8.2 Data Channel Protocol

A data channel connection is established to the printer by the print service for each document within a print job. The printer supervisor establishes the document number and port to use by performing a handshake with the printer on the control channel. The sod command is used for this purpose. Should a new command be issued on the control channel before the client has established the data channel link, then the server will abandon the data channel link. The printer supervisor may then either reissue an sod command to obtain a different port or use the data opcode to send a limited amount of data to the selected interpreter on the control channel. After all data has been sent by the client, the open data channel is closed. Since the end of the document is signaled by disconnection of the link, the printer is unable to distinguish between normal completion and disconnects caused by communication errors. It is the responsibility of the printer supervisor to log any errors and reprint any document which does not complete because of communication difficulties.

8.3 Management Channel Protocol

The printer is passive. It does not initiate connections on management channels. However, it may not be able to print until it receives a connection from a master management client so it can read its configuration and startup data. Should a printer supervisor attempt to begin a control channel session before the printer has been contacted on a management channel, that session request is rejected until the configuration and startup have completed. It is the responsibility of the print service to make sure that periodic retries are performed until the printer is ready to print.

A management client process attempts to maintain a connection to a printer at all times. It probes at relatively short intervals (every 30 seconds or so) when it loses contact. Once a management client has established a connection with the printer, the printer becomes the active agent and the management client becomes a passive server to the printer. The printer asks the management client for certain services and receive replies; it also sends the management client accounting and logging information. Requests from the printer to read and write files are directed only to one management client.

The accounting information that is sent to the management client advertising ACCOUNT capability is separate from the accounting information transmitted to individual print services on the control channel. The purpose of this capability is to provide centralized accounting for printers that may accept print jobs from several print services. Multiple management clients may be configured to avoid loss of accounting information.

The exchanges between a print server and management clients share the same syntax and protocol as is used for the control channel. Opcode strings shared with the control channel include the following:

- data
- reply and prepl
- show
- showpdl
- showres

There are a few additional opcodes that are recognized only on management channels. Level II features designated by the use of an asterisk (*).

Table 8–2 lists the configuration and accounting opcode strings and their symbolic names.

Table 8–2: Management Channel Protocol Configuration and Accounting Opcodes

Opcode	Symbol	Description
"41"	mssn	Begin a new management session. Similar to the ssn opcode, except that it begins a management session rather than a print client session. The data array contains a list of values providing some required information and then naming the services that the management client is willing to provide to the printer. On networked printers, multiple clients may offer different services. Required parameters are as follows: PASSWORD the printer is free to reject the management session request if it does not like the value assigned to PASSWORD. HOST the host name requesting the management connection. The printer is free to reject the management session request if it does not like the value assigned to HOST. PRINTERHOST the network host name of the printer itself. Initially itknows only its address. *CLIENTID a string identifying the version of the software running in the management client; its only use is for information. *PROTOCOL the protocol version of the management client. The remaining parameters are all optional. To declare that it is willing to offer a certain service, the management client sets the value of the service name to 1. For example, if it is willing to provide the CFREAD service (read access to configuration files), then the string CFREAD=1 should appear in the list of values. The specific service code names are not part of the protocol, but rather are part of a private agreement between the server and its management client(s). Typical values might include: CFREAD Read access to configuration files FILEIO Read/write access to certain file directories ACCOUNT Receives and stores accounting data ERRLOG Receives and stores error and log data *FONT Read access to fonts The management client must wait for the server to reply. The reply may contain a list of values containing information about the printer as follows: *SERVERID a string identifying the version of the software running in the printer; its only use is for information. *PROTOCOL the protocol version of the printer

Table 8–2 (Cont.): Management Channel Protocol Configuration and Accounting Opcodes

Opcode	Symbol	Description
"42"	time	Request the current date and time. A management client is expected to reply to a time opcode with a reply record whose Data part is the current time. The data array of the reply contains a character string specifying the current local time, in the form `dd-mmm-yyyy hh:mm:ss` where dd is the 2-digit day of the month, 01 to 31 mmm is the 3-letter month code, JAN to DEC yyyy is the 4-digit year, 1971 or later hh is the 2-digit hour, 00 to 23 mm is the 2-digit minute, 00 to 59 ss is the 2-digit second, 00 to 59 The date and time string may have leading or trailing spaces, but the content portion must be exactly 20 characters. The month code is the first 3 letters of the English spelling of the name of the month, and can be in any combination of upper and lower case letters.

Table 8–2 (Cont.): Management Channel Protocol Configuration and Accounting Opcodes

Opcode	Symbol	Description
"43"	acct	Sent by the print server to all management clients that offered ACCOUNT service when they connected, asking the client to record accounting information. The data array passed to the management client by the printer is a list of values which should be written to the accounting file. Not every acct command will define every value; the recommended behavior of the management client is to write the entire data array to the accounting file, intact, and to let a post-processing program worry about decoding the several possible combinations of parameters. The parameters that can be defined include: • DATE the date that the print job began. • START the time that the print job began. • USER The name of the user who printed the file, as passed to the printer by the most recent soj command. • HOST the network name of the computer from which the file was printed, as passed to the printer by the most recent info command. • UNIT the unit number of the printer which performed the job if more than one printer is attached. • MEDIA a list of Media sizes currently in each input tray. The encodings for media size strings are specified in Appendix B. • PAGES the number of images produced. For clients reporting a version number, this is a list of the number of sides printed for each input tray. • SHEETS the count of the sheets of paper (or other media) used for each input tray. • WASTE the number of sheets wasted due to jams or other difficulties for each input tray. • IN the number of bytes read by the printer's interpreter. • TIME the amount of CPU time used by the printer's interpreter. The values for MEDIA, PAGES, SHEETS and WASTE (if any) are implemented as parallel lists. The management client must return a reply acknowledging receipt of the accounting data.
"44"	emsg	Event message. Nearly identical to the data opcode, except that its contents pertain to the status of the server rather than to the status of the job that the user is printing on any printer. Messages of this type will be sent to all management clients that offered ERRLOG service when they connected. The text of the logging message should identify the unit number of the physical printer if needed and more than one is attached.

Table 8–2 (Cont.): Management Channel Protocol Configuration and Accounting Opcodes

Opcode	Symbol	Description
"45"	cssn	Console session. Used to begin a "remote console" session. A management client that connects for a "remote console" session can exchange data records with the printer in a proprietary format not described herein. The console session shares the management channel with the other management functions. The data array of a cssn command is a list of values, as follows: • USER: the logged-in name of the user requesting the remote console session. • HOST: the hostname on which the user is logged in. • *CLIENTID a string identifying the version of the software running in the console client; its only use is for information. • *PROTOCOL the protocol version of the host. If the server has more than one physical printer attached, then the client may also specify the following: • *UNIT the unit number of the desired printer. The print server will reply with a reply opcode if the connection is accepted and with a nak opcode if the connection is rejected. Level II servers may respond with: • *SERVERID a string identifying the version of the software running in the printer; its only use is for information. • *PROTOCOL the protocol version of the printer After the initial reply, the server and client exchange data records with one another until the session is terminated by either end closing the connection. The format of these data records is not part of the protocol, but is rather part of the implementation-specific nature of the server code itself. Some servers may use the msg opcode in addition to the other common opcodes to send administative messages to the management client running the console session.

Table 8–3 lists opcode strings used to give the printer access to a file system. If the management client has offered FILEIO or CFREAD service, then it must be prepared to handle open, read, and close opcodes. If the management client has offered FILEIO service, then it must also be prepared to handle the write opcode, and it must have a file name space (usually a directory) available from which it can read and write files as directed by open commands.

Management clients must be able to translate the tokens $CONFIG, $DEFAULTS, $SETUP and $RESOURCES into filenames referring to printer-specific files for the printer being supported.

Table 8–3: Management Channel Protocol File Access Opcodes

Opcode	Symbol	Description
"50"	open	Sent by the print server to the management client, asking the client to open a file and send a reply back to the printer containing the file number. The data array of the open command is a list of the following values: • PATH: a string which the management client uses to locate a specific file. The string is of the form "filename" or "directory/.../filename" or a "$" prefixed token. The management client restricts the domain of the file to be within a path from an implied root; for example, the name "bin/lps_ps.exe" may be interpreted as /usr/lib/lps/bin/lps_ps.exe on a Unix system, or LPS$ROOT:[BIN]LPS_PS.EXE on a VMS system, or as \lps\bin\lps_ps.exe on an MSDOS system. • TYPE: A string describing how to open the file. If the string contains the letter "r", then the file will be opened for input; if it does not exist, then an error status will be returned. If the string contains the letter "w", then the file will be opened for output; if it does not exist it will be created, and if it does exist, then the old copy will be overwritten. If the string contains the letter "b", then a binary file will be opened instead of a text file. PATH may also be one of the tokens $CONFIG, $DEFAULTS, $RESOURCES or $SETUP. These tokens locate a printer-specific file for the purpose of establishing general printer configuration, specific PDL defaults, resources to be loaded and any special setup. If the server has more than one printer attached and/or more than one PDL is supported, then the data array may also contain the following values: • *UNIT the unit number of the desired printer. • *PDL the Page Description Language required. Refer to Appendix C for a list of recognized interpreters. If the open request is incorrectly formed, the printer will be sent back a nak reply. If the open request is correctly formed, then (even if it fails) the printer will be sent a reply record, containing a list of values as follows: • RETURN is the file handle that must be used by the printer in future "read", "write", and "close" commands to the management client. • ERROR, if present, means that the request to open a file has failed; the value of ERROR is the text of the error message. It is not permitted to open a file named $CONFIG, $DEFAULTS $RESOURCES or $SETUP for output; this is to prevent such files from being overwritten.

Table 8–3 (Cont.): Management Channel Protocol File Access Opcodes

Opcode	Symbol	Description
"51"	read	Sent by the print server to the management client, asking the client to read from a file and return the data so read back to the printer. The data array passed to the management client by the printer is a list of values: • HANDLE: a file handle, as returned from a call to open • OFFSET: the offset, measured in bytes from the beginning of the file, at which to begin the read. To read the second 512-byte block from a file, set OFFSET=512 and COUNT=512. If OFFSET is not present, the value of OFFSET will default to the current file position, and the file will be read sequentially. • COUNT: the integer count of the number of bytes to read from the file. The count must not exceed 512. • LINE: if the value LINE is defined to have a non-zero value, then the data returned back to the printer will never be more than one line. A "line" is defined to be the characters beginning with OFFSET through and including the next newline character in the file. IF a read operation stops at a newline character because of the LINE option, then that newline character (CTRL-J) will always be the last character in the returned data. If the read request is incorrectly formed, the printer will be sent back a nak reply. If the read request is correctly formed, then (even if it fails) the printer will be sent a reply record, containing a list of values as follows: • RETURN is the number of bytes actually read from the file. It will be equal to the incoming value of COUNT unless the file was too short or unless the LINE value was specified and a newline character was seen. • ERROR, if present, means that the read request has failed. The value of ERROR is the text of the error message. • DATA is the data actually read from the file The name DATA will be the last name in the list of values, and it can contain any character at all, including the 001 (control A) character normally used to delimit fields in a list of values. The length of the DATA string is determined by the RETURN value. A read that is near the end of a file can cause the amount of data returned to be smaller than anticipated. A read that is at the end of a file causes RETURN=0, and returns an empty data array.

Table 8–3 (Cont.): Management Channel Protocol File Access Opcodes

Opcode	Symbol	Description
"52"	write	Sent by the print server to the management client, asking the client to write to a file from the data passed to it. The Data array passed to the management client by the printer is a list of values: • HANDLE: the file handle, as returned from a call to open. • OFFSET: the offset, measured in bytes from the beginning of the file, at which to begin the write. • COUNT: the count of the number of bytes to be written, which must not exceed 512. • DATA: is the block of bytes to be written. It must be the last value in the list. The block of bytes to be written can contain any character at all, including the 001 (control A) character normally used to delimit fields in a list of values. (This is why the block of bytes to be written is the last value of the list.) If the write request is incorrectly formed, the printer will be sent back a nak reply. If the write request is correctly formed, then (even if it fails) the printer will be sent a reply record, containing a list of values as follows: • RETURN: the number of bytes actually written. • ERROR: if present, then the write request has failed and the value of ERROR is the text of the error message.
"53"	close	Sent by the print server to the management client, asking the client to close a previously-opened file. The data array passed to the management client by the printer is a list of values: • HANDLE: the file handle to be closed. If the close request is incorrectly formed, the printer will be sent back a nak reply. If the close request is correctly formed, then (even if it fails) the printer will be sent a reply record, containing a list of values as follows: • ERROR: if defined, then the close request has failed and the value of ERROR is the text of the error message.
"54"	*readfile	Sent by the print server to the management client, asking the client to send an entire file to the printer using the data channel protocol. The data array passed to the management client by the printer is a list of values: • HANDLE a file handle, as returned from a call to open • PORT a string identifying which port will be monitored by the printer to accept the incoming connection. This is a numeric token which is mapped to the lower protocol levels by mutual agreement between the management client and the printer in the same fashion as the sod command. For TCP, ports 1024 through 1027 correspond to tokens 1 through 4. For DECnet, PORT replies map to Objects 128 through 131.

Table 8–4 lists the opcode used to permit use of a font access facility. If the management client has offered FONT service, then it must be prepared for a findfont opcode, and it must know how to contact a Font Access Facility to cache font files for the printer. The printer random

accesses the cached font using FILEIO after it has been cached by the management client. A description of the findfont opcode follows:

Table 8–4: Management Channel Protocol Font Access Opcode

Opcode	Symbol	Description
"56"	*findfont	Sent by the print server to the management client, asking the client to locate a font and cache it. The data array of the command is: • FONTNAME: a name for the font which the management client must use to locate a file containing the font data. • TYPE: A string describing the font format desired. At the present time, the only format is PS-random. • USERID is the string name of the user whose file is being printed. • HOSTNAME is the name of the host computer from which user USERID is printing the file. The reply is as follows: • PATH: A name which may subsequently be used to open the file. • ERROR: if present, then either the request has failed or the font file cannot be read, and the value of ERROR is the text of the error message.

Chapter 9

Examples

9.1 A Level II Protocol Print Session

Table 9–1 lists the interactions between a print service and the printer for a print session containing a job with a trailer page:

Table 9–1: Example of a Level II Protocol Print Session

Client Action	Server Response	
Create Circuit		Open a control channel
	Accept Circuit	Virtual circuit established
SSN		Identify the printer supervisor
	REPLY	Acknowledge with the state of the printer
SOJ		Identify the user
SOD		Specify and reset the PDL
	REPLY	Tell supervisor which Port
Create Circuit		Open a data channel
	Accept Circuit	Virtual circuit established
Send Data		User's print data
Disconnect Circuit		(Data channel) Document ends
EOD		Request Info. for job trailer
	REPLY	Accounting data sent
SOD		Specify and reset the PDL
	REPLY	Tell supervisor next Port
Create Circuit		Open a new data channel
	Accept Circuit	Virtual circuit established
Send Data		Job trailer page
Disconnect Circuit		(Data channel) Trailer page ends

Table 9–1 (Cont.): Example of a Level II Protocol Print Session

Client Action	Server Response	
EOJ		Request final job accounting info.
	REPLY	Accounting data sent after all printing
Repeat above		For next job - or
Disconnect Circuit		(Control Channel) to end session

9.2 A Level I Protocol Print Session

Table 9–2 lists the interactions between a print service and the printer for an older style print session.

Table 9–2: Example of a Level I Protocol Print Session

Client Action	Server Response	
Create Circuit		Open a channel
	Accept Circuit	Virtual circuit established
SSN		Identify the printer supervisor
	REPLY	Acknowledge with the state of the printer
SOJ		Identify the user
SOD		Reset the PDL
DATA		User’s print data
...		
EOD		Request Info. for job trailer
	REPLY	Accounting data sent
SOD		Reset the PDL
DATA		User’s print data
...		
EOJ		Request final job accounting info.
	REPLY	Accounting data sent after all printing
Repeat above		For next job - or
Disconnect Circuit		End session

Use of the Level I protocol is not recommended for new implementations.

9.3 A Level II Aborted Print Session

Table 9–3 shows the interactions between a print service and the printer for an aborted job with a trailer page.

Table 9–3: Example of a Level II Protocol Aborted Print Session

Client Action	**Server Response**	
Create Circuit		Open a control channel
	Accept Circuit	Virtual circuit established
SSN		Identify the printer supervisor
	REPLY	Acknowledge with the next job number
SOJ		Identify the user
SOD		Specify and reset the PDL
	REPLY	Tell supervisor the document number
Create Circuit		Open a data channel
	Accept Circuit	Virtual circuit established
Send Data		User’s print data
Disconnect Circuit		(Data channel) Truncate document
KILL		Abort and Request Info. for job trailer
	REPLY	Accounting data sent
SOD		Specify and reset the PDL
	REPLY	Tell supervisor which Port
Create Circuit		Open a data channel
	Accept Circuit	Virtual circuit established
Send Data		Job trailer page
Disconnect Circuit		(Data channel) Trailer page ends
EOJ		Request final job accounting info.
	REPLY	Accounting data sent after all printing
Repeat above		For next job - or
Disconnect Circuit		(Control Channel) to end session

9.4 A Level I Aborted Print Session

Table 9–4 shows the interactions between a print service and the printer for an old style aborted job.

Table 9–4: Example of a Level I Protocol Aborted Print Session

Client Action	Server Response	
Create Circuit		Open a channel
	Accept Circuit	Virtual circuit established
SSN		Identify the printer supervisor
	REPLY	Acknowledge with the state of the printer
SOJ		Identify the user
SOD		Reset the PDL
DATA		User’s print data
...		
KILL		Urgent Data or Expedited Message
	REPLY	Accounting data sent
Disconnect Circuit		End session

Use of the Level I Protocol is not recommended for new implementations. Level I clients do not unambiguously identify the document being aborted, so the entire session must be killed.

Appendix A

Reported printer options

Options reported in reply to show may include any of the following:

LCIT	Large Capacity Input Tray
ENV	Envelope Tray
SCOT	Small Capacity Output Tray
LCOT	Large Capacity Output Tray
LCOS	Large Capacity Output Stacker
MB:n	Mailbox Sorter or collater with n bins
DPX	Duplexer
BIND	Binder
STPL	Stapler
PNCH	A punch or drill

Appendix B

Reported media sizes

European Paper Sizes:

A0-B10	(ISO 216)
B0-B10	(ISO 216)
C0-C10	(ISO 269)

North American Sizes:

LETTER	(8.5" by 11")
LEDGER	(11" by 17")
LEGAL	(8.5" by 14")
EXECUTIVE	(7 to 7.5" by 10")
7X9	(7" by 9")
10X14	(10" by 14")
UNIVERSAL-LARGE	(Up to 12" by 17" variable)
UNIVERSAL-SMALL	(Up to 12" by 10.125" variable)

In addition:

EMPTY	(No paper in this slot)

Appendix C

Recognized Page Description Languages

The following is a list of interpreters and their variants which may be identified by showpdl:

Interpreter	Variants	Comments
DDIF	text,graphics,images,colour	
HPGL		Hewlett-Packard
HP-PCL	pcl1,pcl2,pcl3,pcl4,pcl5	
IBM PPDS	level1,level2,level3	
IMAGE		DEC Image Print-PLUS
INTERPRESS		Xerox
ISO 6429	iso-10538-pif-level1,iso-10538-pif-level2,	
	iso-10538-pif-level3,iso-10538-fft-level1,	
	iso-10538-fft-level2a,iso-10538-fft-level2b,	
	iso-10538-fft-level2c,simple-text,	
	dec-ppl2,dec-ppl3	
LINE	ansi-cc,machine-cc,trc,sosi1,sosi2,sosi3	
MODCA	is1,pt1,dr2,dr3,fs10,fs20,bcd1,iml	
PS	L1 | L2	
ReGIS	colour	
SCS	sosi1,sosi2,sosi3	
SPDL	binary | cleartext	
tek4014		

Refer to ISO/IEC DIS 10175-1 Document printing application specification for additonal listings or updates.

Appendix D

Application Programming Interface for CPAP

D.1 Generic Printer Driver Communication Interface for Print Supervisors

Table D–1: Functional Interface for a CPAP Print Session

Client Action	Server Response	
(0)		**sts = openPrinter (cbk); - BLOCKING**
Create Circuit		Open a control channel
	Accept Circuit	Virtual circuit established
(1)		**sts = startJob (cbk, &jobID, userInfo, readCallback, rcbPtr); BLOCKING**
SSN		Identify the printer supervisor
	REPLY	Acknowledge with the state of the printer
SOJ		Identify the user
(2)		**sts = startDoc (cbk, jobID); BLOCKING**
SOD		Specify and reset the PDL
	REPLY	Tell supervisor which Port and docID
Create Circuit		Open a data channel
	Accept Circuit	Virtual circuit established
(3)		**Write Data Until Done**
Send Data		User’s print data
Disconnect Circuit		(Data channel) Document ends

Repeat [(2) and (3)] for next document

Table D–1 (Cont.): Functional Interface for a CPAP Print Session

Client Action	Server Response	
(4)	**sts = endDoc (cbk, jobID); BLOCKING**	
EOD		Request Accounting Info. for job trailer
	REPLY	Accounting data sent

Repeat [(2) and (3)] for Job Trailer page

Table D–1 (Cont.): Functional Interface for a CPAP Print Session

(5)	**sts = endJob (cbk, jobID); NON-BLOCKING for Job Overlap**	
EOJ		Request final job accounting info.
	REPLY	Accounting data sent after all job pages printed

Repeat [(1) to (5)] for next job - or

(6)	**sts = closePrinter (cbk); BLOCKS until all Jobs DONE**	
Disconnect Circuit		(Control Channel) to end session

(7)	**sts = interruptDoc (cbk, docID); BLOCKING (w/timeout)**	
KILL		Abort and Request Info. for job trailer
Disconnect Circuit		(Data channel) Truncate document
	REPLY	Accounting data sent

D.2 Formal Parameters for Generic Printer Functions

a. cbk - Connection Block
   - int discriminant; - READONLY

     indicating printer type.

     Currently, the interface supports network printers only. However, the generic printer driver’s functional interface is not limited to the CPAP protocol or network printers.
   - char deviceId [ ]; - READONLY

     Node name for network Printers. Device channels for local printers.
   - int cfd; - Control Channel Descriptor - READ/WRITE
   - int dfd; - Data Channel Descriptor READ/WRITE

b. userInfo - READONLY
   - — userName
   - — clientName
   - — clientjobID
   - — miscellaneous

c. Printer jobID

   int jobID; - READ/WRITE

d. readCallback - Routine address for handling Printer messages.

e. rcbPtr

   Pointer to readCallback arguments:
   - int message_type;

     indicating: user, operator, resource, accounting, EVENT

     Types of EVENTs:
     - — INTERRUPT_ACK - Supervisor generated interrupt’s ack from Printer
     - — EOJ_ACK - from printer
     - — DOC_ABORT - from printer (for example, a PDL syntax error)
     - — JOB_ABORT - from printer’s Server Management
   - char *buffer [ ];
   - int bufCount;
   - int status;

D.3 Table Maintained by the Lower Level

D.4 State and Status Information

D.4.1 Job States

- NULL
- JOB_IN_PROGRESS
- DOCUMENT_IN_PROGRESS
- INTERRUPT_SENT
- JOB_COMPLETED
- EOJ_SENT

D.4.2 Job Completion Status

- Success
- Network Error
- Protocol Error
- Server Initiated Abort

D.4.3 Job Status

- PDL Error
- Success

D.4.4 Document Status

- PDL Error
- Success

D.5 Aborts

D.5.1 Print Supervisor Generated Aborts

For Example: lprm(1), lpc(8), print/delete, stop/queue etc.

Kills current document leaving session open for a possible trailer.

ReadCallback generates INTERRUPT_ACK event.

D.5.2 Printer Management Initiated Abort

Kills entire job. Job trailer page is NOT expected.

ReadCallback generates JOB_ABORT event.

D.5.3 Error in User Document

Resume job from the next document; Job trailer page is possible.

ReadCallback generates DOC_ABORT event.

D.6 State Tables

Table D–2: JobState vs EVENTS

		EVENTS			
JobState		**INTRRUPT_ ACK**	**EOJ_ACK**	**DOC_ ABORT**	**JOB_ABORT**
NULL	0	♦	♦	♦	♦
JOB_IN_PROGRESS	1	♦	♦	♦	5
DOCUMENT_IN_ PROGRESS	2	♦	♦	1	5
INTERRUPT_SENT	3	1	♦	1	5
EOJ_SENT	4	♦	5	♦	5
JOB_COMPLETED	5	♦	♦	♦	♦

Key

♦—ERROR in PrintServer Software
†—NO OP

Table D–3: JobState vs Calls

		CALLS				
JobState		**startJob**	**endJob**	**startDoc**	**endDoc**	**interruptDoc**
NULL	0	1	§	§	§	§
JOB_IN_PROGRESS	1	§	4	2	§	§
DOCUMENT_IN_PROGRESS	2	§	§	§	1	3
INTERRUPT_SENT	3	§	§	§	§	§
EOJ_SENT	4	§	§	§	§	§
JOB_COMPLETED	5	§	§	§	§	§

Key

§—ERROR in Application Software
†—NO OP

D.7 State Diagram

```
   +----------+
   | NULL     |
   +----------+
       |
       |
       |  startJob
       |
       v
    +---------------+       endJob            +--------+
    |JOB_IN_PROGRESS|------------------------>|EOJ_SENT|
    +------^-----^--+                         +--------+
     |     |     |                                 |
 startDoc  |     |                      EOJ_ACK,   |
     |     |     | INTERRUPT_ACK        ABORT_JOB   |
     | endDoc    |                                 V
     |     |  +---------------+             +-------------+
     |     |  |INTERRUPT_SENT |             |JOB_COMPLETED|
     |     |  +--^------------+             +-----^-------+
     | DOC_ABORT |interruptDoc                   /|\
     V     |     |                              / | \
+----------------+---+                         /  |  \
|DOCUMENT_IN_PROGRESS|                        JOB_ABORT
+--------------------+
```

Appendix E

Security Considerations

This RFC does not address security issues. At the present time, authorization and access control lists are supplied by a Master Management Client which must advertise CFREAD services to the printer. Authorization of specific hosts and users is controlled at lower protocol layers and by the exchange of unencoded passwords. Enhancements to the protocol to permit better authentication and encryption of documents is under consideration. Such extensions shall be the subject of a subsequent RFC.

Appendix F

Version information and incompatibilities

Due to the public nature of the protocol and to minimize possible problems associated with its enhancement, the following rules are established:

- Any implementation shall ignore any undefined opcodes or parameters. This assists in permitting enhancements to coexist with earlier implementations.
- If no protocol version is reported on the SSN/ REPLY handshake, then the base protocol is assumed. This document corresponds to V2.2.
- Version numbers are of the form n.d where n is the major version, and d is the minor version. A client and its server must exchange messages using the highest common major version.
- Minor version updates reflect either bug fixes or enhancements which do not materially affect the protocol. No implementation shall be expected to examine the minor version except to report it for informational purposes.

Appendix G

Author's Address

James D. Jones
Digital Equipment Corporation
4 Technology Park Dr., DSG1–1/M6
Westford, MA 01886–4196

Phone: 508–635–8612
Fax: 508–635–8673
Email: jones@regent.enet.dec.com