*Recognized as an*
*American National Standard (ANSI)*

**IEEE Std 1284.1™-1997**
**(Stabilized 2009)**

# IEEE Standard for Information Technology—Transport Independent Printer/System Interface (TIP/SI)

Sponsor

**Microprocessor and Microcomputer Standards Committee**
**of the**
**IEEE Computer Society**

Stabilized 19 March 2009
Approved 29 July 1997

**IEEE Standards Board**

Reaffirmed 19 June 2003
Approved 24 April 1998

**American National Standards Institute**

**Abstract:** A protocol and methodology for software developers, computer vendors, and printer manufacturers to facilitate the orderly exchange of information between printers and host computers are defined in this standard. A minimum set of functions that permit meaningful data exchange is provided. Thus a foundation is established upon which compatible applications, computers, and printers can be developed, without compromising an individual organization's desire for design innovation.

**Keywords:** computers, printer command sets, printer communications, printers

The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street, New York, NY 10017-2394, USA



ISBN 1-55937-944-8

**IEEE Standards** documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied "**AS IS**."

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation, or every ten years for stabilization. When a document is more than five years old and has not been reaffirmed, or more than ten years old and has not been stabilized, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration. A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal interpretation of the IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Recommendations to change the status of a stabilized standard should include a rationale as to why a revision or withdrawal is required. Comments and recommendations on standards, and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board

445 Hoes Lane

Piscataway, NJ 08854

USA

## Introduction

[This introduction is not part of IEEE Std 1284.1-1997, IEEE Standard for Information Technology—Transport Independent Printer/System Interface (TIP/SI.)]

Local area networks are increasingly becoming the most popular means of interconnecting devices within a corporation. With costs per connection coming down, this trend shows no sign of abating. As networks grow larger, more computers and printers will be interconnected. Any weaknesses in network printing will only be magnified as more devices are made to communicate.

The absence of feedback from existing printers causes many problems in today's network environment. For example, a user could be submitting a job to a remote printer. If that printer is low on toner, most printers today do not have the capability to inform users about this condition. Additionally, if the job is large, the user risks having to wait until the job is finished before finding out that the output is incorrect. The resulting waste of paper, toner, and time could be significant when calculated over a period of time on a large network.

Standardized feedback information from a printer would solve this problem. By the use of this standard, when a printer recognizes a condition that would prevent it from accurately printing a job, it can send a standardized message to a host computer that is monitoring network printing. Upon receipt of this message, the host could then send a message to the user who submitted the job, informing him or her of the error condition. The user could then redirect a job to a more appropriate printer or undertake action to correct the defect at the target printer. When projected over a period of time and a large number of network users, the resulting monetary savings could be substantial.

The example shown above is just one of many error conditions that could occur when printing either on a standalone computer or over a network. By using this standard format for exchanging information between the printer and host, software vendors, network suppliers, and printer manufacturers will now be able to greatly improve the efficiency of network printing.

At the time this standard was completed, the key contributors to the IEEE 1284.1 Working Group were as follows:

**Forrest D. Wright**, *Chair*
**Jay Martin**, *Secretary*
**Michael Timperman,** *Editor*

Chirag Bakshi
Michael Bringmann
Andy Davidson
Neal Fischer
Bob Garcia
Rod Gerhart
Tom Hasting
Tim Hathaway
Bob Herriot
Ken Hilliard
Barry Hills
Terry Jones
Theodore Kearley
Charles Kimber
Rick Landau
Harry Lewis
Rick Lomicka
Ernie Mandese
Stan McConnell
Ron Norton
Roman Orzol
Steve Price
Suzy Price
Paul Reilly
Dave Roach
Avi Schlank
Ron Smith
Gail Songer
Larry Stein
Mike Stilz
Koji Tashiro
Randy Turner
William Wagner
Craig Whittle
Keith Winter
Lloyd Young
Steve Zilles

The following persons were on the balloting committee:

Dharma P. Agrawal
Keith D. Anthony
Charles Brill
George Horansky
Thomas M. Kurihara
Donald C. Loughry
Raymond Lutz
Edward McCreight
Darrell B. McIndoe
James D. Mooney
Roman Orzol
Roy Reed
Larry C. Sollman
Larry Stein
Robert G. Stewart
Michael Timperman
Clarence M. Weaver, Jr.
Gregory Winston
Forrest D. Wright
Oren Yuen

The final conditions for approval of this standard were met on 29 July 1997. This standard was conditionally approved by the IEEE Standards Board on 26 June 1997, with the following membership:

**Donald C. Loughry**, *Chair*
**Richard J. Holleman**, *Vice Chair*
**Andrew G. Salem**, *Secretary*

| | | |
|---|---|---|
| Clyde R. Camp | Lowell Johnson | Louis-François Pau |
| Stephen L. Diamond | Robert Kennelly | Gerald H. Peterson |
| Harold E. Epstein | E. G. "Al" Kiener | John W. Pope |
| Donald C. Fleckenstein | Joseph L. Koepfinger* | Jose R. Ramos |
| Jay Forster* | Stephen R. Lambert | Ronald H. Reimer |
| Thomas F. Garrity | Lawrence V. McCall | Ingo Rüsch |
| Donald N. Heirman | L. Bruce McClung | John S. Ryan |
| Jim Isaak | Marco W. Migliaro | Chee Kiow Tan |
| Ben C. Johnson | | Howard L. Wolfman |

*Member Emeritus

Also included are the following nonvoting IEEE Standards Board liaisons:

Satish K. Aggarwal
Alan H. Cookson

Kristin M. Dittmann
*IEEE Standards Project Editor*

**Trademarks**

The trademarks in the following list are referred to on one or more pages of this standard. This list is provided as a matter of convenience and accommodation. By providing the list, IEEE takes no position on the legal status of any trademarks listed nor of any trademarks omitted from the list. While the information in the list is believed to be correct, it is based on information supplied by those who claim to own the trademarks. Consequently, IEEE disclaims liability for any inaccuracies in the list.

Adobe and Postscript are trademarks of Adobe Systems Incorporated.

Hewlett-Packard is a registered trademark of Hewlett-Packard Company.

IBM and Presentation Manager are registered trademarks of IBM Corporation.

Internetwork Packet Exchange, IPX, NCP, and NetWare Core Protocol are trademarks of Novell, Inc.

Microsoft and Windows are registered trademarks of Microsoft.

Novell and NetWare are registered trademarks of Novell, Inc. in the United States and/or other countries.

TrueType is a registered trademark of Apple Computer, Inc.

CLAUSE PAGE

CLAUSE PAGE

# IEEE Standard for Information Technology—Transport Independent Printer/System Interface (TIP/SI)

## 1. Overview

### 1.1 Scope

A standard protocol for the control of printers that is independent of the underlying data stream or page description language (PDL) used to create the printed page is defined in this standard. This protocol is usable by all classes of printers. The scope of this standard is limited to management and control of printers and does not include management or control of a printing system or subsystems.

### 1.2 Purpose

There is currently no defined, independent standard for controlling printers. Each vendor builds some control into the underlying PDL or data stream. Without an independent, openly defined protocol, applications and operating systems cannot automatically determine the type of printer being addressed. This protocol will provide a minimum implementation subset that will allow automatic identification and configuration of printers and vendor extensibility to provide for growth and product differentiation.

### 1.3 Objectives

This standard defines a protocol for communications between a host and printer. Its intent is to provide a standard methodology for software developers, computer vendors, and printer manufacturers that facilitates the orderly exchange of information between printers and host computers. A minimum set of functions that permit meaningful data exchange is defined. Thus, this standard establishes a foundation upon which compatible applications, computers, and printers can be developed, without compromising an individual organization's desire for design innovation. The following objectives accompany this standard:

a) To simplify the printer driver development process by defining a standard set of command/response transactions between the host computer and printer.
b) To accelerate the development of communicating printers by providing a robust protocol that can be implemented in phases ranging from basic to extended functionality.
c) To ease customers' printing problems (especially over networks) by accelerating the availability of communicating printers and compatible host software.

d) To assist software developers in minimizing time to market by establishing a base set of functions that ensure a minimum level of communications between the host and printer.
e) To facilitate the creation of powerful network print management software by defining transactions that work across a wide range of printers.
f) To enable the creation of standard control/communications firmware that can be included in many peripheral devices.
g) To create a standard methodology for host and printer communications that is independent of the transport mechanism used between devices.
h) To enhance the management of printers in networks by providing a mechanism for printers to readily provide their status and configuration to the host application.
i) To permit design innovation by providing flexibility within the specification for printer manufacturers to include extensions to the original set of guidelines.
j) To ensure cross-platform host-to-printer communications by creating an operating system-independent set of guidelines.

The resultant protocol is PDL-independent with the capability of a printer to support multiple PDLs, all active at the same time, if desired.

# 2. Definitions, abbreviations, and acronyms

## 2.1 Definitions

This clause provides certain terms and definitions used in this specification that may not be generally familiar or that may be used with a very specific meaning. These definitions are not intended to be absolute but rather to give the sense of the terms as used in the context of this standard.

**2.1.1 American National Standard Code for Information Interchange (ASCII):** In this standard, a text string of some arbitrary length. The text string may contain nonprintable characters. The length must be stated in another field. The encoding is typically ISO 8859-1, but is specified by the Printer Language field in a “Request Device Characteristics—Request Summary” response. *See also:* IEEE Std 100-1996.

**2.1.2 ASCIIz:** In this standard, an ASCII string concatenated with a NULL character.

**2.1.3 big endian:** A term used to imply that bytes in a word and words in a double word are transmitted most significant byte or word first in a serial stream of bytes.

**2.1.4 bit:** A single binary integer. A set bit represents to a binary “1.” A cleared bit represents a binary “0.”

**2.1.5 bit-encoded byte:** A byte with a definition for each bit.

**2.1.6 bit-encoded word:** A word with a definition for each bit.

**2.1.7 byte:** An entity composed of 8 bits, used to define a unit element of memory or transmitted data. It is capable of describing integers in the decimal range –128 to 127.

**2.1.8 command:** A message from the host directed to the printer that may or may not include print data.

**2.1.9 connection:** A parallel interface state that is outside the scope of this standard and is not defined herein. This state is indicative of the state of the physical or logical connection between a host and the printer. Only in this state can data be transferred between a host and the printer.

**2.1.10 console language:** The human language in which information is to be displayed on local or remote consoles.

**2.1.11 deprecated:** Supported by predecessors to this standard but no longer used.

**2.1.12 desktop management interface (DMI):** A facility, normally host resident, for handling and translating defined interfaces for component information, event information, stored management information format (MIF) structures and management information. In this context, a component is an integral device or product, such as a printer; and event is an asynchronous alert or trap. The management information is used by an application such as a user management

 

program, or may be converted for use by a network management facility. The desktop management interface is controlled by the Desktop Management Task Force.

**2.1.13 Desktop Management Task Force (DMTF):** An association of software developers, host computer, and peripheral device manufacturers that promulgate a platform-independent interface standard for the management of desktop computers and the peripheral devices attached to such computers. In addition to defining the interfaces between components, the host-based service layer and management applications, the association supports the development of consistent management information format (MIF) structures that define the significant manageable and status attributes of various components incorporated into desktop computers.

**2.1.14 document:** An encoded, electronically transmittable image, set of images, or image-related information, which is handled by the printer interface control unit.

**2.1.15 double word:** A field composed of two words. In a message, the most significant word is transmitted/received first. It is capable of describing integers in the decimal range –2 147 483 648 to 2 147 483 647.

**2.1.16 duplex:** For purposes of this standard, a type of printing that involves the process of creating images or impressions on both sides of the printing media.

**2.1.17 facsimile:** A process by which textual or pictorial images are communicated, typically but not exclusively, over telephone lines. The images may be coded in raster or compressed raster format (such as CCITT group 3) or in a page description language such as Adobe™ PostScript™. Facsimile typically operates down to the physical link level and includes protocols providing control and addressing mechanisms specific to the media being used. This is distinguished from the process of communicating similarly encoded images over local or wide area networks. However, both may be considered implementations of remote printing.

**2.1.18 feed direction:** On most printers, the direction that the medium is moved through the marking engine. For a printer in which the medium is not moved, the feed direction may be considered as along the Y axis. The across feed direction is the direction orthogonal to the feed direction; it is also called the crossfeed or scan direction on some printers.

**2.1.19 finishing:** An operation or group of operations performed on the printed media after it emerges from the printer output mechanism. Finishing includes operations such as stapling (stitching), punching, binding, folding, cutting, etc., which may or may not be considered part of the printing process. Note that operations of collating and sorting are normally considered printer output functions rather than finishing.

**2.1.20 host:** Whatever is driving (i.e., providing) commands or data to the printer (e.g., a workstation, a print server, or spooler).

**2.1.21 impression:** The process of marking the media. A single-sided, one-color printer requiring one pass per sheet would produce one impression per sheet. A similar printer printing duplex would produce two impressions per sheet. A two-pass printer providing a base color and a highlight color would produce two impressions per side, etc.

**2.1.22 interpreter:** A functional entity that translates one or more printer control or page description languages into a form suitable for the marking engine. Since printers sometimes emulate original implementations of these languages, interpreters are sometimes called emulations.

**2.1.23 interpreter language:** The printer machine language, or page description language, by which information to be imaged or to be used in imaging is coded.

**2.1.24 job:** That entity originated or initiated by a user which is handled by the printer interface control unit. A job need not result in the imaging of information on media.

**2.1.25 kilobyte (kB):** Equivalent to 1024 bytes.

NOTE — As used in this document, the term kilobyte (kB) means $2^{10}$ or 1024 bytes, megabyte (MB) means 1024 kilobytes, and gigabyte (GB) means 1024 megabytes.

There is a plan to deprecate this base-2 usage because of the potential for confusion with the base-10 definition. An alternative notation for base-2 is under development. Because the above mentioned base-2 usage is widespread in some areas of application, however, it may be employed in IEEE publications for a limited time.

**2.1.26 legacy application or device:** A printing application or device in existence prior to the existence of this standard. Hence, such device is unaware of the protocol defined herein.

**2.1.27 link:** The physical or logical connection between a host and a printer.

**2.1.28 logical unit (LU):** An addressable, functional group. In the case of a printer, scanner, or facsimile it is a functional group concerned with the storage, acquisition and/or processing of a textual and/or pictorial image. A printer may have one or more interpreters. The design of a particular printer determines if these interpreters are capable of concurrent operation.

**2.1.29 management information base (MIB):** A simple network management protocol (SNMP) compatible data structure that defines the functional groups and management objects of a unit or system. This standard includes a mechanism whereby the object values can be obtained from the printer by a device acting as the SNMP agent for the printer.

**2.1.30 management information format (MIF):** A desktop management interface (DMI) compatible data structure that defines the functional groups and management objects of a unit or system.

**2.1.31 marking engine:** A set of electrical and mechanical components that moves the print media and marks that media. In some implementations, a facsimile transmission function is considered to be a marking engine.

**2.1.32 message:** A logical grouping of one or more packets sent either from host to printer (a command message) or from printer to host (a response message).

**2.1.33 message length:** Although messages can be of any length up to 65 539 bytes, the packet size should be selected for effective transmissions over the physical link without requiring disassembly and reassembly. For connections through a network, the packet size of that network would generally be the most efficient.

**2.1.34 multiple-packet error rejection:** Error handling and rejection notification occurs on a message-by-message basis. This standard assumes the existence of a reliable transport layer protocol. Error detection and packet sequencing is a transport layer function and is beyond the scope of this standard.

**2.1.35 network printing alliance protocol (NPAP):** A transport independent printing protocol from which this standard has been derived.

**2.1.36 NULL:** A byte with all bits set to zero.

**2.1.37 object identifier (OID):** In general, a unique representation (name) of a manageable object defined in a management information base (MIB).

**2.1.38 octet:** An eight-bit data entity (byte).

**2.1.39 packet:** The basic message element used by this standard; a structured field, having a start byte, a two-byte length field (the first two bytes), a flag byte, a command byte, followed by the subcommand and/or data fields.

**2.1.40 page:** A logical representation of a single unit of printing media. It is a function of the document formatting rather than the printing process. There are one or more pages per impression. A “four-up” single-color printing will typically have four pages per impression.

**2.1.41 page description language (PDL):** A formal printer machine language, consisting of commands and data (or equivalently, operators and operands) used to specify and control the content and format of printed pages. A data stream encoded in a page description language is rendered into the printed page image by an interpreter.

**2.1.42 print engine:** That set of electrical and mechanical mechanisms that move the print media or paper and marks that paper.

**2.1.43 printer:** An intelligent device that includes, as a primary function, the ability to convert an electrically transmitted or stored image into a physical image formed by colorant on some medium (such as paper).

**2.1.44 printer interface control unit (PICU):** The set of electronics that interfaces external communications ports, common peripheral interfaces (such as font cards or disk drives), the logical units, and the marking engine. It is the function of the printer interface control unit to coordinate and sequence all the functions and operations of the printer.

**2.1.45 printer language:** The human language used for the American National Standard Code for Information Interchange (ASCII) strings within all command and response messages, other than those to be printed or those to be displayed on the local or remote consoles (e.g., English, French, German).

**2.1.46 printing:** That set of operations implemented by the printer that results in an image rendered as marks on the selected media.

 

**2.1.47 print media:** That consumable upon which the marking engine marks so as to form a text and/or pictorial image, typically paper.

**2.1.48 reply:** Messages from the printer to the host. *Syn.:* response.

**2.1.49 session:** A printer state that allows the logical grouping of one or more jobs into a sequential, referenceable collection.

**2.1.50 sheet:** A cut piece of print media, such as a sheet of paper.

**2.1.51 simple network management protocol (SNMP):** A protocol used for the management of network nodes and devices, used extensively on internet and other networks. The protocol provides for the communication of status and setup information between a management console and a managed device using values of objects defined in the management information base (MIB) for the managed object.

**2.1.52 space character:** A byte (hex 20) used in text strings that represents a space.

**2.1.53 start of packet byte:** A single byte (hex A5, decimal 165) that is used by both the printer device and the host to quickly determine whether or not they are synchronized.

**2.1.54 transport or transport layer:** The middle layer in the ISO seven-layer open system communications reference model, and the boundary between the communication subnet layers (physical, data link, and network) and the host process layers (session, presentation, and application).

**2.1.55 unsigned byte:** A byte that represents positive integers in the decimal range 0–255.

**2.1.56 unsigned dword:** A dword that represents positive integers in the decimal range 0–4 294 967 295.

**2.1.57 unsigned word:** A word that represents positive integers in the decimal range 0–65 535.

**2.1.58 word:** A field composed of two eight-bit bytes. In a byte serial message, the most significant byte is transmitted/received first, (big endian). It is capable of describing integers in the decimal range –32 768 to 32 767.

## 2.2 Abbreviations and acronyms

| | |
|---|---|
| ASCII | American National Standard Code for Information Interchange |
| bbyte | bit-encoded byte |
| bword | bit-encoded word |
| CPMA | common printer MIB/MIF alerts |
| DMI | desktop management interface |
| DMTF | Desktop Management Task Force |
| DSA | device status alerts |
| dword | double word |
| IETF | Internet Engineering Task Force |
| IMA | interpreter message alerts |
| JCA | job control alerts |
| kB | kilobyte |

NOTE — See NOTE under “kilobyte” in definitions.

| | |
|---|---|
| LAN | local area network |
| LU | logical unit |
| MIB | management information base |
| MIF | management information format |
| NPAP | network printing alliance protocol |
| OID | object identifier |
| PDL | page description language |
| PICU | printer interface control unit |
| RDC | request device characteristics |
| RFC | Request For Comments (as defined by the Internet Engineering Task Force) |
| TCP/IP | Transmission Control Protocol/Internet Protocol |
| TIP/SI | transport independent printer system interface |
| udword | unsigned dword |
| uword | unsigned word |

# 3. Software model

## 3.1 IEEE 1284.1 printer model

A printer can be thought of as a minimum of three logical entities: The print engine, the printer interface control unit (PICU), and one or more logical units (LUs) (one of which is an interpreter). Figure 1 shows a logical model of the IEEE 1284.1 printer.

It is important to note that while an interpreter is an LU, the converse is not always true. An LU can be another logically separate device in the printer such as a facsimile engine or document scanner.

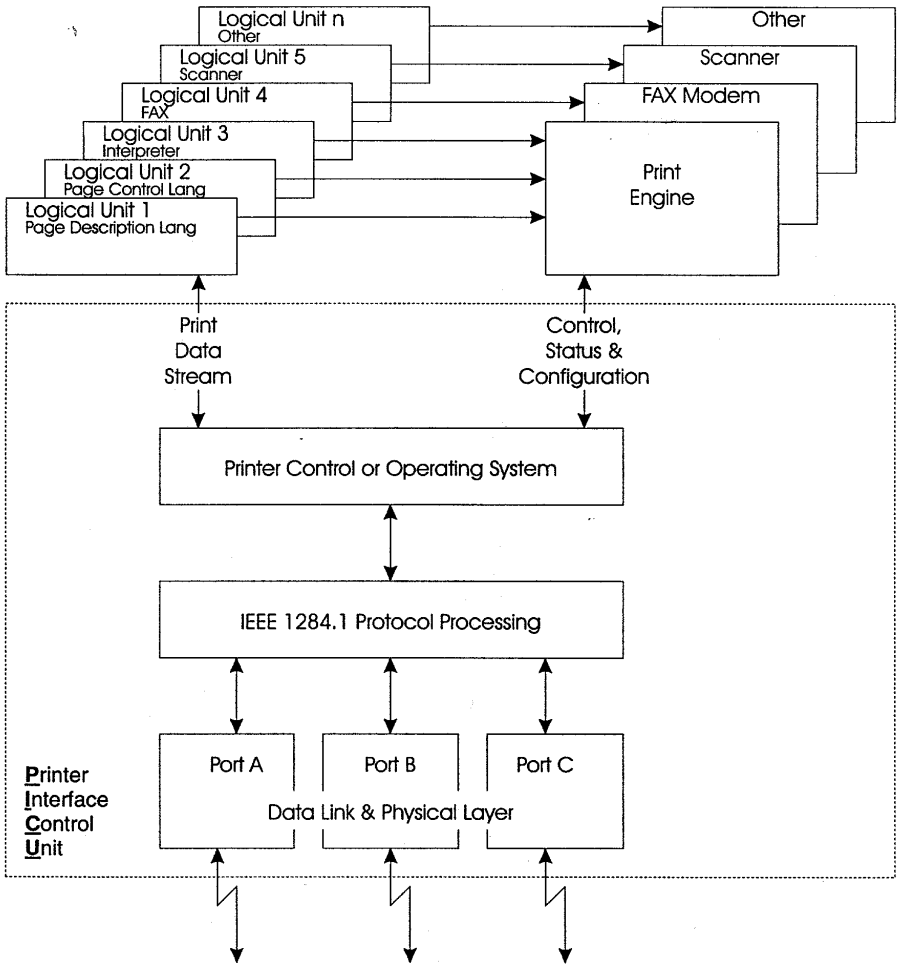


**Figure 1— Printer model**

The management of producing a printed document is complex. The task can be divided into two overlapping pieces, the management of printing and the management of the printer. Printing encompasses the entire process of producing a printed document: generation of the file to be printed, selection of a printer, choosing printing properties, routing, queuing, resource management, scheduling, and final printing including notification of the user. This standard enables communications of printer status and attributes that, when used by applications, greatly enhance both the management of printing and of the printer.

 

In this standard, commands have been defined that allow the manipulation of manageable objects that are contained in a printer and are capable of being referenced using an OID such as those print objects defined in the IETF Printer MIB, RFC1759[B6].[1] Job delivery is outside the scope of RFC1759, as it has chosen only to cover the management of the printer itself. In doing so, RFC1759's printer model differs from the one presented in Figure 1. The RFC1759 model is basically an exploded view of the print engine and control blocks used by this standard.

The function of this standard's protocol is to provide a means of returning configuration and status information in a manner that is independent of the printer's physical connection, imaging technology, or embodied LU(s). In this model, the printer is intended to be managed by some intermediate system element external to the printer. The protocol provides a method for

— Retrieving traditionally static information, such as the number and type of paper input and output trays, imaging resolution and speed, interpreter capabilities and their relationship to the print mechanism, etc.
— Returning real-time status information related to the print mechanism that is independent of the LUs, such as paper and printing supplies levels.
— A separate, logical, out-of-band communication channel for exchanging commands and responses. This out-of-band channel is key to the effective management of the printer while page processing is being done.
— Selection or activation of an LU via host control.
— Job separation.
— Reporting job statistics independent of the LU.

The protocol has been designed to be link independent. The only requirement is that the link must be capable of bidirectional data transmission. Numerous methods and media exist today to facilitate this transport of information. Included are various LAN protocols (ISO/IEC 8802-2: 1994, TCP/IP, etc.), asynchronous serial communications, and, more recently, the bidirectional parallel port defined by IEEE Std 1284-1994 [B3].

## 3.2 Printer job data flow

The printer job data flow model is used to visualize the sequence of events for printing and for the identification of the position of a particular job in the printing process. A generalized model of the data flow for a job is shown in Figure 2. Data is received by the protocol processing layer from the data link and physical layer. The model assumes that the printer is a simple buffered device and handles print data via a simple, single-threaded queue.

The first action taken by the printer is to determine if the received data is directed to an LU or is a command. This decision is facilitated by examining the flag byte in the packet header. If the data packet contains a command, it is placed in a command buffer. With the exception of the Start Session, Start Job, End Job, and End Session commands, all commands are acted on immediately. The size of the command buffer is dependent on the specific printer implementation.

[1]The numbers in brackets preceded by the letter B correspond to those of the bibliography in Annex G.

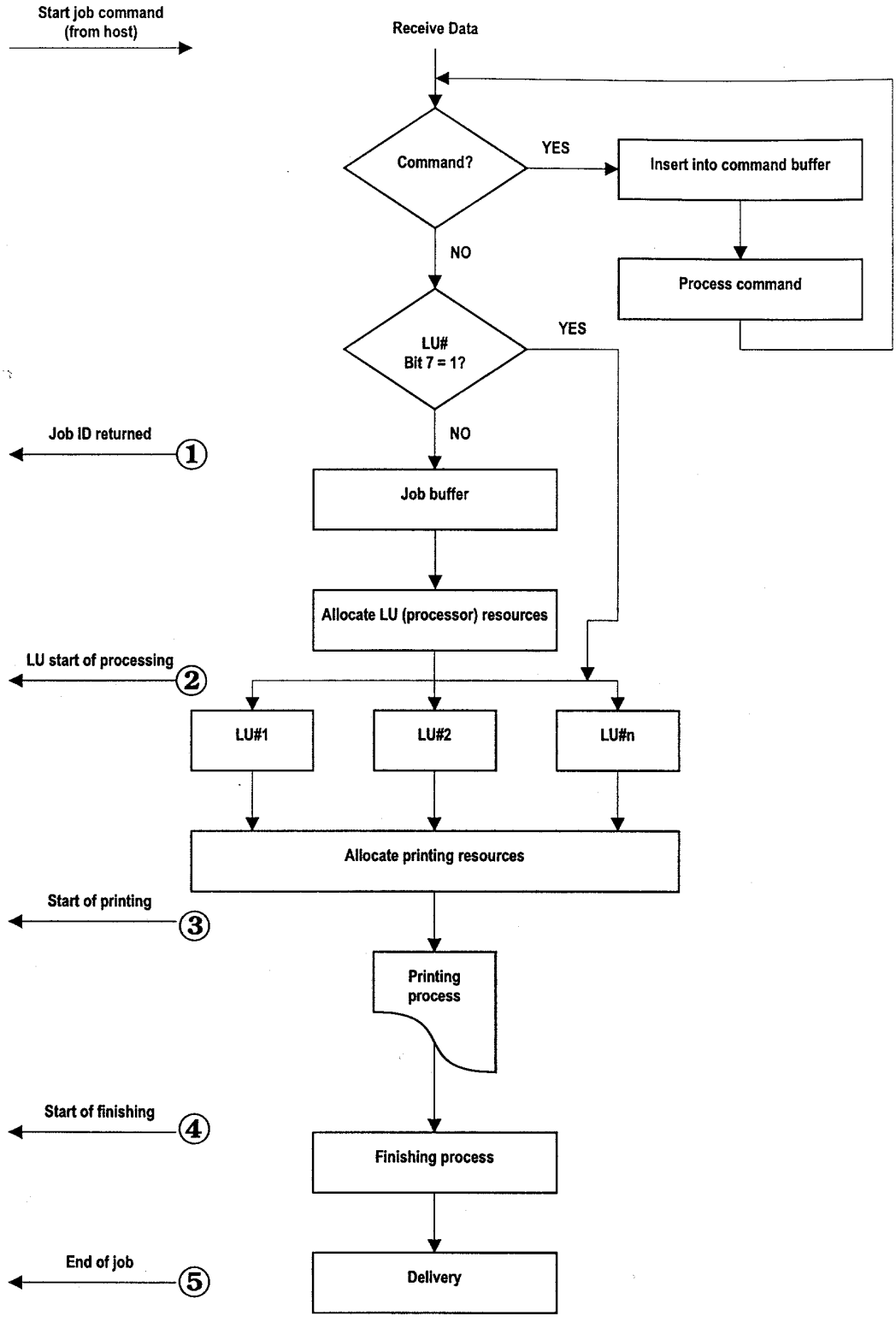


**Figure 2— Job data flow model**

When the data is LU destined, the higher order bit (bit 7) of the LU address is used to determine if the data is to be put into the job buffer or sent immediately to the active LU, bypassing the buffer.

As data is removed from the job buffer, the printer activates the appropriate LU. It is important to note that one or more LUs may be active at any point depending on printer implementation. When the LU finishes processing or reaches an appropriate point, the printer control system then allocates the print engine to the LU. For this discussion, it is assumed that a printer has only one print engine available for use, even if there are multiple LUs processing data simultaneously. Once the document has been printed, any available and requested finishing operations are performed. Upon completion of any document finishing, the document is marked is delivered and the end of job signaled to the host if requested.

The preceding explanation is a highly simplified and sequential model of the printing process. In an actual implementation most of the steps of starting an LU, processing data, printing data and finishing can be overlapped. The start of LU processing may imply that printing and or finishing resources are allocated simultaneously. It is not required that one step in the printing process be completed before the next is started.

### 3.2.1 Job data flow model

The IEEE 1284.1 protocol provides the ability for the host system to coordinate and report on the status of the various steps in the printing process so that effective printer and job management can be implemented. The model is provided as an aid to the visualization of the printing process and does not dictate any particular implementation or control system for a printer. The circled numbers on the job data flow model, Figure 2, show status points with the following brief description as to their meaning.

1) Job ID returned.
   This is the first point of synchronization where the printer returns a printer assigned job identification number to the host after a Start of Job command has been received from the host. The job ID allows the host to track a particular job.
2) LU start of processing.
   This is a synchronization point where the printer has allocated the resources necessary to run a job on a particular LU and is about to begin processing the job. The host has the option of suspending a print job at this point and interacting with the printer to determine if the appropriate resources are available to complete the particular job that is about to be processed by the LU. Checking for a particular font would be an example of an action that could be taken at this point.
3) Start of printing.
   This is a synchronization point where the printer is about to start physically printing the first page of the job. The host has the option of suspending a print job at this point to interact with the printer to determine if the appropriate resources are available to complete the printing, such as specific media installed in the printer.
4) Start of finishing.
   This is a synchronization point where the host has the option of suspending a print job to interact with the printer to provide detailed instructions on final finishing of the document.
5) End of job.
   This is the final synchronization point where the print job is complete and job statistics are returned to the host regarding the print job.

## 3.3 Printer sessions and jobs

### 3.3.1 Session and job concepts

Within the IEEE 1284.1 framework the basic logical grouping of data is the job. Jobs are delineated in the data stream by the Start Job and End Job commands. The Start Job command resets accounting statistics and may cause the printer to initialize the graphic engine. The End Job command instructs the printer to log the accounting statistics, and release any nonpersistent job-related resources.

A session allows the logical grouping of one or more jobs into a sequential, referenceable collection. Empty jobs and empty sessions are valid. Sessions are delineated in the data stream by the Start Session and End Session commands. The Start Session command is an indicator to the printer that a collection of jobs follows. The printer shall use this command to lock port rotation in a multiple link and/or multiple protocol environment. The End Session command marks the end of the collected data, causes the printer to print and eject any remaining sheets, and allows link rotation.

Because the session commands directly affect link rotation, grouping unrelated jobs in the same session could lead to an unfair allocation of printing resources. Therefore, it is recommended that a session not contain a collection of unrelated jobs. This recommendation is not an attempt to limit the number of jobs in a session. It is a recommendation that the host should refrain from concatenating unrelated jobs together in a session to circumvent link rotation.

### 3.3.2 Session Data Flow Model

Once a physical or logical connection is established between the host and printer, a start session command prompts the printer to enter an “in-session” state and return an identifying number for the session. While the printer is in this “in-session” state, the Start Job and End Job commands are used to delineate related print data. Job statistics are accumulated for this delineated data and are provided to one or more hosts. (See Figure 3.)

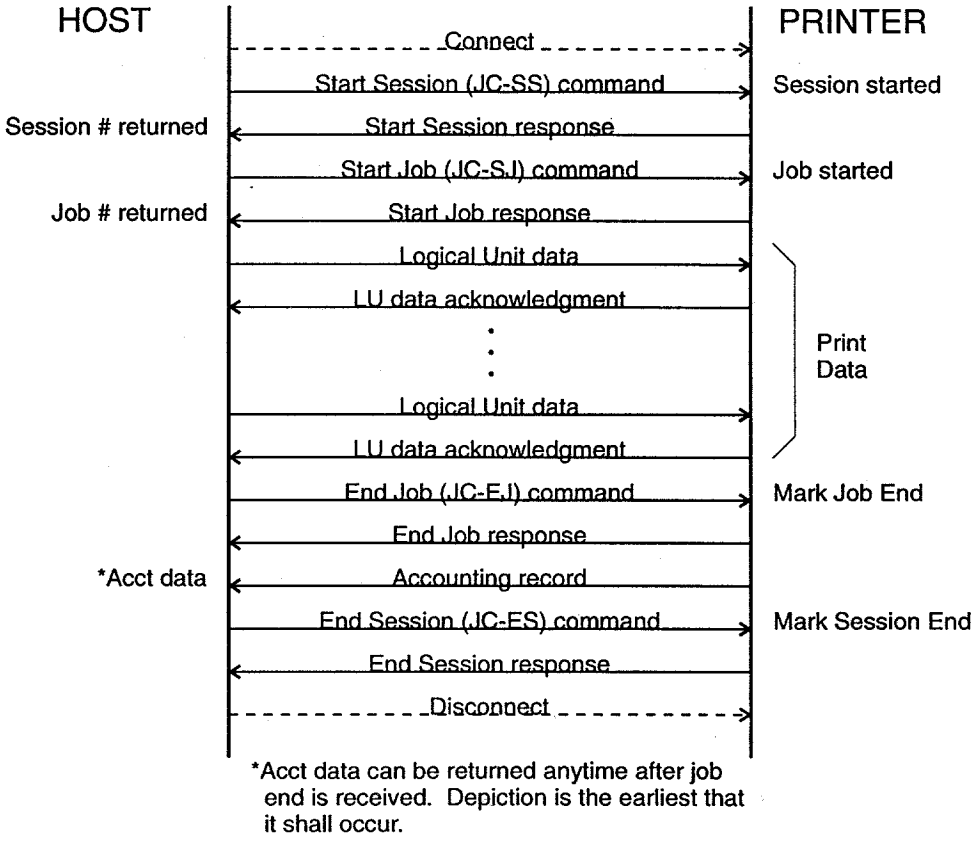


**Figure 3— Session and job chronological data flow**

# 4. The command/reply structure

## 4.1 General topics

### 4.1.1 Commands

Commands are host to printer messages instructing the printer to perform some action.

### 4.1.2 Responses

Responses are printer to host messages sent as the direct result of a command from the host. Every message sent to the printer with the reply required flag (flag bit 4) set shall result in a response from the printer. In the case where the printer has no data to return to the host, a short acknowledgment consisting of the original command with the flags appropriately updated shall be returned.

### 4.1.3 Alerts

An unsolicited printer to host message is called an alert. These messages are initiated by exception conditions detected in the printer. These events can occur either asynchronously to the command stream (e.g., out of toner/paper/supplies, cover open, etc.) or synchronously to the command stream (e.g., erroneous command). In either case, an alert is issued.

The various types of alerts are as follows:

a) Device status alerts (DSA), resulting from
   1) Events that occur inside the printer asynchronous to the data stream (i.e., out of toner/paper/supplies, cover open, job completed, etc.)
   2) Message errors (i.e., “invalid logical unit,” etc.), when the host did not expect a response (flag bit 4 cleared in message from host to printer)

b) Interpreter message alerts (IMA), resulting from interpreter responses to data stream queries

c) Job control alerts (JCA), resulting from
   1) A print job completed
   2) A print job starting logical unit processing
   3) A sheet from a print job being stacked in the output device

d) Common printer MIB/MIF alerts (CPMA), resulting from a critical alert entry (as defined by RFC1759 [B6]) being added to the printer’s alert table.

### 4.1.4 General flow

In general, the host is the master and initiates all exchanges with the printer. With the exception of defined alerts, the printer shall not send an unsolicited response to the host. Solicited responses require that the reply required flag is set in the received message.

It is recommended that the host communicate with the printer with the reply required flag set in all commands sent to the printer. In this mode the host sends each message (reply required flag set) to the printer and waits for a positive acknowledgment (printer response) before sending the next message.

When a command with the reply required flag set is received, indicating that the message contains data to be delivered to a LU, the printer must send a simple response acknowledging that the data was received. Any printer response action required as a result of the LU processing data in the message is defined to be unsolicited. The decision to return this unsolicited message to the host is governed by a prior command that enabled interpreter message alerts. If these alerts are enabled, the requested response will be returned to the host as an alert (flag bit 4 cleared) from the interpreter (flag bit 6 cleared, command field = interpreter id) directly following an Interpreter Message Alert message.

### 4.1.5 Guidelines for manufacturer-unique commands

Vendors may define and implement their own unique commands, or “extensions,” to IEEE Std 1284.1-1997. The vendors who implement their own commands can still be classified as compliant to the standard, but only if they have implemented the IEEE 1284.1 commands in accordance with this standard. Support for the unique commands is the responsibility of the defining vendor.

Compliant devices shall reject properly structured commands that are not implemented in the device by setting the flag bits to indicate a data error in the response message sent back to the host. A vendor defining new manufacturer-specific commands shall comply with the architecture of commands/subcommands, packet structure, field definitions, general flow and all other protocols defined in this standard. Vendor compliance shall allow existing compliant devices to parse and properly reject the extended commands that they do not implement.

## 4.2 The structure

### 4.2.1 Basic packet structure

The basic structure for all packets (host to printer and printer to host) is of the form shown in Table 1.

**Table 1— Basic packet structure**

| Packet header | | | | Data field |
|---|---|---|---|---|
| Start of Packet Byte | Packet length | Flag | Command or LU # | Data... |

### 4.2.2 Definition of command fields

#### 4.2.2.1 Start of Packet Byte (byte 1)

This byte (hex A5) is used by the host to indicate a start of packet. When the printer is in a mode to receive a packet from the host, the first byte received must be a Start of Packet Byte. If this byte is not the first byte received, the printer will know immediately that it is out of sync with the host or is not receiving commands or data compliant with this standard.

#### 4.2.2.2 Packet length (bytes 2 and 3)

Packet length is the number of bytes in this packet not including this length field or the Start of Packet Byte (16-bit unsigned word). The default “maximum packet size” for commands is 64 bytes. This value shall be used until the host determines the printer’s exact capabilities via a Request Device Status.

#### 4.2.2.3 Flag (byte 4)

The purpose of the flag is to provide a single byte that the printer can examine to obtain control information. The flag bits are defined in Table 2.

**Table 2— Command flag bits**

| | |
|---|---|
| Bit 7 | No Operation (NOP).<br>If this bit is set, it indicates that the host requires the printer to disregard (i.e., not process) this message regardless of its contents. If the Reply Required bit is set, the printer is still required to acknowledge the receipt of this packet (see bit 4). If this bit is cleared, it indicates that the printer is to process the command. |
| Bit 6 | Destination (IEEE 1284.1 Component or LU #).<br>If set, this bit indicates that the message is for the bidirectional software component in the printer and the fifth byte in the message contains the command op code. If cleared, it indicates the message contains data for an LU with the fifth byte in the message being the logical unit number that is to receive the data. |
| Bit 5 | Continue.<br>If set, this bit indicates that the next packet is a continuation of this message. If cleared, it indicates that this message is “complete.” |
| Bit 4 | Reply Required.<br>The host sets this bit when it requires a response from the printer. If this bit is set, all commands sent from the host to the printer will be acknowledged by the IEEE 1284.1 component within the printer. If the command is data for an LU, the printer’s IEEE 1284.1 component will acknowledge the data as being received. The LU will not respond to the Reply Required flag. If the NOP bit is set, the printer’s IEEE 1284.1 component will acknowledge the data as being received and discarded (byte #5 of the printer response will have the same value as byte #5 of the host to printer command).<br>If this bit is cleared, the IEEE 1284.1 component will only respond when errors are detected. The responses will be sent as unsolicited messages to the host. |
| Bit 3 | Undefined (reserved) |
| Bit 2 | Undefined (reserved) |
| Bit 1 | Undefined (reserved) |
| Bit 0 | Undefined (reserved) |

 

#### 4.2.2.4 Command byte (byte 5)

The meaning of the command byte is dependent on the state of bit 6 in the flag byte.

If bit 6 is set, this field contains a command for the IEEE 1284.1 command processor. Commands 0–127 (hex 0 to 7F) and 240–255 (hex F0 to FF) are reserved for IEEE 1284.1 commands and future extensions. Commands 128–239 (hex 80 to EF) are available for manufacturer- or device-specific assignment. Assignment of value usage in this range is not managed by the IEEE.

If bit 6 is cleared, this field contains the logical unit number destination for this message. A value of 0 is used to select the default LU. The default LU is defined as the LU compatible with today's machine definition. For example, it may be the power on default interpreter that can be overridden by the front panel or the data stream (as in emulation switching).

#### 4.2.2.5 Data bytes (bytes 6, 7,...)

The meaning of the data bytes are dependent on the state of bit 6 in the flag byte.

If bit 6 is set, this field contains, depending on the command, additional command information and/or data for the IEEE 1284.1 command processor.

If bit 6 is cleared, this field contains data for the logical unit.

### 4.2.3 General printer to host protocol considerations

#### 4.2.3.1 Command error

If a command is received by the printer that is not understood (i.e., undefined), the printer shall set the error flag (bit 7), set bit 3, and clear bit 2 in the packet header of the response. The printer shall perform no other action on that command other than to flush it from internal buffers. If the host thinks that the printer should have accepted the command, it shall retry the command again.

#### 4.2.3.2 Rejecting a command

Both the host and the printer shall always be able to accept at least one full packet at any time. This packet does not necessarily need to be acted upon. A message sent from the host to the printer can be rejected by the printer. To indicate that a given packet has been rejected, the printer shall set the error flag (bit 7) and clear both error type flags (bits 2 and 3) in the packet header for the reply message. At this point, the host can determine why the printer could not execute the command by issuing an Request Device Status command, which is *always* honored. The host can also simply keep sending the command until it is accepted and operated upon.

#### 4.2.3.3 Data error

If a command is received by the printer that has data that is not understood (i.e., undefined), the printer shall set the error flag (bit 7), set bit 2, and clear bit 3 in the response sent back to the host. The printer shall perform no other action on that command other than to flush it from internal buffers. If the host thinks that the printer should have accepted the command, it shall retry the command again.

### 4.2.4 Definition of response fields

#### 4.2.4.1 Start of Packet Byte (byte 1)

This byte is used by the printer to indicate a start of packet. When the host is in a mode to receive a packet from the printer, the first byte received shall be a Start of Packet Byte. If this byte is not the first byte received, the host immediately know that it is out of sync with the printer.

#### 4.2.4.2 Packet length (bytes 2 and 3)

Packet length is the number of bytes in this packet not including the length field or the Start of Packet Byte (16 bit, unsigned word). The default "maximum packet size" for responses is 64 bytes. This value shall be used until the host tells the printer otherwise using the Printer Configuration Control command.

#### 4.2.4.3 Flag (byte 4)

The purpose of the flag is to provide a single byte that the host can examine to obtain control information for the message and a quick view of the printer's status. The flag bits for a printer response are defined in Table 3.

**Table 3— Response flag bits**

| | |
|---|---|
| Bit 7 | Error.<br>If set, this bit indicates that this response contains error information. This error flag is used to indicate a command or data error for the command sent to the printer. |
| Bit 6 | Source (IEEE 1284.1 Component or LU #).<br>If set, this bit indicates that the message is from the IEEE 1284.1 command processor in the printer and the command field contains the command op code to which the printer is responding. If cleared, it indicates the message contains data from an interpreter and the command field contains the logical unit number of the specific interpreter that generated the data. |
| Bit 5 | Continue.<br>If set, this bit indicates that the next packet is a continuation of this message. If cleared, it indicates that this message is "complete." |
| Bit 4 | Reply Type.<br>If this bit is set, it means that this reply was required by the host (i.e., solicited via bit 4 in the message from the host). When cleared, this bit indicates that this message was not solicited by the host but was caused by an "error" event at the printer. The error condition is asynchronous and can occur at any time. |
| Bit 3..2 | Command/data check type, encoded as follows:<br>NOTE—These bits are valid only when the Error flag (bit 7) is set.<br>Bits 7..3..2 Definition when contained in a message from the printer.<br>0xx No errors were detected in the command or data fields.<br>100 No errors were detected in the command or data fields, but the command was rejected.<br>101 Data Error<br>110 Command Error<br>111 Reserved |
| Bit 1..0 | Printer Status.<br>These two bits are set to the state of the printer at the time of the reply, independent of whether or not they were selected (i.e., "armed") by the PCC command. The bits are defined as follows:<br>Bits 1..0 Definition when contained in a message from the printer.<br>00 Printer is fine; no alert condition exists.<br>01 Printer is still OK, but a condition exists that the operator may want to know about (a warning—low paper, job end, etc.).<br>10 A condition exists that will prevent further printing, but the operator can fix it (operator intervention required—out of paper, output bin full, toner low, etc.).<br>11 A condition exists that will prevent further printing and a service/repair call is required to fix it (laser failure, fuser died, system board error, etc.). |

#### 4.2.4.4 Command byte (byte 5)

In a reply, the original op code or interpreter ID of the message that caused the reply shall be returned. If the message was unsolicited from the IEEE 1284.1 command processor (i.e., bit 4 is cleared and bit 6 is set), this field shall be set to a value in the range 240–255 (hex F0 to FF). Currently 255 (hex FF) is used to identify Device Status Alerts, 254 (hex FE) is used to identify Interpreter Message Alerts, 253 (hex FD) is used to identify Job Control Alerts, 252 (hex FC) is used to identify Common Printer MIB Alerts, 251 (hex FB) is used to identify Operator Panel Alerts, and 240 (hex F0) is reserved for manufacturer-specific Device Status Alerts.

#### 4.2.4.5 Response data bytes (bytes 6, 7,...)

The data bytes shall be interpreted as defined in Table 4.

 

**Table 4— Interpretation of response data bytes 6, 7,...**

| Flag Bits 6..4 (Byte #4) | Command (Byte #5) | Data (Bytes #6–n) |
|---|---|---|
| 00 Message from an Interpreter | LU # | Message from the interpreter identified in byte #5 |
| 01 Required acknowledgment from IEEE 1284.1 command processor that data for an interpreter was received and buffered | LU # | N/A |
| 10 Alert | 240 Manufacturer-Specific Alert<br>251 Operator Panel Alert<br>252 Common Printer MIB/MIF Alert<br>253 Job Control Alert<br>254 Interpreter Message Alert<br>255 Device Status Alert | Alert data |
| 11 Required response from bidirectional component | IEEE 1284.1 command | Command Response as defined in command details for the IEEE 1284.1 command indicated by byte #5 |

# 5. Command set

## 5.1 Overview

The defined commands and their hexadecimal equivalents are shown in Table 5.

**Table 5— Commands**

| Command | Hexadecimal equivalent | Status |
|---|---|---|
| Request Device Characteristics | 01 | current |
| Request Interpreter Characteristics | 02 | current |
| Printer Configuration Control | 03 | current |
| Request Device Status | 04 | current |
| Job Control | 05 | current |
| Request Logical Unit Characteristics | 06 | current |
| Control Packet Rejection | 07 | deprecated |
| Common Printer Variable | 08 | current |
| Remote Op Panel | 09 | current |

## 5.2 Request Device Characteristics (RDC)

### 5.2.1 Overview

The RDC command provides the host system with a method of obtaining information concerning the capabilities and characteristics of the printer. The host system would use this information to communicate to the application such data as paper size and output capabilities such as sorting. Additionally, the host could use this information for automatic scheduling and routing applications, such as that provided by print servers on a local area network.

Using the RDC command, there are various categories of characteristics information that can be obtained. Table 6 shows the various categories.

**Table 6— RDC categories**

| Category | Description |
|---|---|
| Summary | Provides for an overview of the printer and its capabilities. |
| Input | Describes the printing media, paper, input capabilities, and features of the printer. |
| Output | Describes the printing media, paper, output capabilities, and features of the printer. |
| Options | Description of the various options contained completely within the printer such as expanded memory, mass storage or communications functions, etc. |

The host queries for the desired information by sending a RDC command with a subcommand in the data field that specifies the category of information desired. The subcommand is a single unsigned byte. The subcommand values for the various categories of characteristics are shown in Table 7.

**Table 7— RDC subcommands**

| Subcommand | Characteristic description |
|---|---|
| 00 | Summary information |
| 01 | Reserved |
| 02 | Input characteristics |
| 03 | Output characteristics |
| 04 | Option characteristics |
| 05 to FF | Reserved |

### 5.2.2 Request Summary

The RDC Request Summary subcommand (RDC-RS) provides the host system with an overview of the capabilities of the printer. Using the data returned by the printer, the host knows how to enquire about the details of other features, such as input and output capabilities. Table 8 shows the format of the command sent to the printer.

**Table 8— RDC-RS command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request device characteristics | Packet header |
| 1 | 00 | Request summary | Data field |

After the printer receives the command, it will return a summary data packet back to the host as shown in Table 9.

**Table 9— RDC-RS response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request device characteristic | Packet header |

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 1 | 00 | Request summary | Data field |
| 1 | ubyte | Major revision level of standard supported | |
| 1 | ubyte | Minor revision level of standard supported | |
| 1 | ubyte | Printer specific extension revision level supported | |
| 1 | ubyte | Marking technology | |
| 1 | ubyte | Color capabilities | |
| 2 | uword | Number of monochrome/color levels supported | |
| 1 | bbyte | Duplex capabilities | |
| 2 | uword | Maximum number of entries in query job(s) completed queue | |
| 1 | ubyte | Speed units definition | |
| 1 | ubyte | Speed of printer in speed units | |
| 1 | ubyte | Length units definition | |
| 2 | uword | Horizontal logical units per length unit | |
| 2 | uword | Vertical logical units per length unit | |
| 1 | ubyte | Counter units | |
| 1 | ubyte | Reserved | |
| 4 | udword | Total installed memory (bytes) | |
| 2 | uword | Maximum receive packet size | |
| 2 | uword | Maximum number of outstanding commands | |
| 1 | 00 | Deprecated, must be zero | |
| 1 | ubyte | Number of logical units supported | |
| 1 | ubyte | Number of inputs supported | |
| 1 | ubyte | Number of outputs supported | |
| 1 | ubyte | Number of options supported | |
| 1 | ubyte | Printer language and character encoding | |
| 1 | ubyte | Length of product name (not including this byte) | |
| n | ASCII | Product name | |
| 1 | ubyte | Length of product revision (not including this byte) | |
| n | ASCII | Product revision | |
| 1 | ubyte | Length of serial number (not including this byte) | |
| n | ASCII | Serial number | |
| 2 | uword | Maximum receive command packet size | |

#### 5.2.2.1 Major/minor revision level of standard supported

The combination of these two unsigned bytes specify the revision level of the standard supported by this printer. For this standard, the major revision level is 2 and the minor revision level is 0.

#### 5.2.2.2 Printer-Specific extension revision level supported

This byte is used by the printer to indicate the presence of additional capabilities that are extensions to the standard and are unique to this device. If the value of this byte is 0, then there are no extensions to the standard. A value greater than zero indicates the presence of extensions that are specific to that manufacturer of the printer. The significance of the value of this byte, other than 0, is dependent on the manufacturer of the printer.

#### 5.2.2.3 Marking technology

This unsigned byte defines the type of marking technology used by the printer. The host system can use this field for scheduling and printer type selection. The values for this field are defined in Table 10.

**Table 10— RDC-RS Marking technology codes**

| Value | Description |
|---|---|
| 00 | Electrophotographic (LED, laser, other) |
| 01 | Impact—moving head—dot matrix (9 wire, 24 wire) |
| 02 | Impact—moving head—formed |
| 03 | Impact—band |
| 04 | Ink jet—aqueous |
| 05 | Ink jet—solid |
| 06 | Pen |
| 07 | Thermal transfer |
| 08 | Thermal sensitive |
| 09 | Thermal diffusion (sublimation) |
| 0A | Electrostatic |
| 0B | Photographic (microfiche, imagesetter, other) |
| 0C | Ion deposition |
| 0D | E-beam |
| 0E | Typesetter |
| 0F | Electroerosion |
| 10 | Ink jet—other |
| 11 | Thermal—other |
| 12 to FD | Reserved |
| FE | Other |
| FF | Unknown |

#### 5.2.2.4 Color capabilities

This unsigned byte defines the color capabilities of the printer. The host system can use this field for scheduling and printer type selection. This field has encoded meanings as defined by Table 11.

**Table 11— RDC-RS color capabilities codes**

| Value | Description |
|---|---|
| 00 | Monochrome |
| 01 | Spot color |
| 02 | 3-primary (CMY) color |
| 03 | 4-primary (CMYK) color |
| 04 to FD | Reserved |
| FE | Other |
| FF | Unknown |

#### 5.2.2.5 Monochrome/color levels supported

This unsigned word defines the number of color or monochrome levels that are supported by the printer.

For monochrome printers, this field contains the number of distinct gray scale levels supported at the printer's maximum resolution, not including "white" or "unmarked." For example, a black/white printer with no gray-scale capability should report 1 in this field.

For spot-color printers, this field contains the number of spot colors supported, not including "white" or "unmarked." For example, a black/white/red printer should report 2 in this field.

For 3-primary and 4-primary color printers, this field contains the maximum number of distinct levels, not including "white" or "unmarked," of any one primary at the printer's maximum resolution. Note that the number of distinct

 

colors achievable by combining primaries may be significantly less than the product of the number of levels in each primary.

#### 5.2.2.6 Duplex capabilities

This bit-encoded byte defines the duplex capabilities of the printer. The host system can uses this field to determine whether or not the printer supports duplex printing. This field has encoded meanings as defined by Table 12. The absence of any set bits indicates that the printer is not capable of supporting two-sided printing.

**Table 12— RDC-RS duplex capability encoding**

| Bit | Description |
|---|---|
| 0 | Short edge binding duplex capable (zero=No, one=Yes) |
| 1 | Long edge binding duplex capable (zero=No, one=Yes) |
| 2–7 | Reserved |

#### 5.2.2.7 Maximum number of entries in query job(s) completed queue

This unsigned word defines the maximum number of entries that the printer will retain in its Query Job(s) Completed Queue. (See 5.4 for further information.)

#### 5.2.2.8 Speed units definition

This unsigned byte defines the units that a printer uses when reporting various printer speed capabilities. This field is used together with the value of the byte that returns the printer speed. Table 13 shows the encoded meanings for this field.

**Table 13— RDC-RS speed unit codes**

| Value | Description |
|---|---|
| 00 | Impressions or sheets per minute |
| 01 | Inches per second |
| 02 | Feet per second |
| 03 | Yards per second |
| 04 | Millimeters per second |
| 05 | Centimeters per second |
| 06 | Decimeters per second |
| 07 | Meters per second |
| 08 | Characters per second |
| 09 | Tens of characters per second |
| 0A | Hundreds of characters per second |
| 0B | Thousands of characters per second |
| 0C | Lines per minute |
| 0D | Tens of lines per minute |
| 0E | Hundreds of lines per minute |
| 0F | Thousands of lines per minute |
| 10 to FD | Reserved |
| FE | Other |
| FF | Unknown |

#### 5.2.2.9 Speed of printer in speed units

This unsigned byte defines the maximum speed or how fast the printer can print. The printer speed is expressed in speed units as defined in Table 13. An example of the speed of a printer would be 26 impressions per minute.

#### 5.2.2.10 Length units definition

This is an unsigned byte that defines the units used by the printer when reporting values where a length is involved with the definition. Examples of this would be resolution, 300 dots per inch (dpi), or paper width of 11 in. Table 14 defines the values associated with the various units of length.

**Table 14— RDC-RS length unit codes**

| Value | Description |
|---|---|
| 00 | 0.001 in |
| 01 | 0.01 in |
| 02 | 0.1 in |
| 03 | Inches |
| 04 | 0.01 mm |
| 05 | 0.1 mm |
| 06 | Millimeters |
| 07 | Centimeters |
| 08 to FD | Reserved |
| FE | Other |
| FF | Unknown |

#### 5.2.2.11 Horizontal logical units per length unit

This unsigned word defines the logical units per length unit in the horizontal direction. Horizontal direction is defined to be orthogonal (90 degrees) to the feed or motion of the printing media. Logical units have no physical dimension.

#### 5.2.2.12 Vertical logical units per length unit

This unsigned word defines the logical units per length unit in the vertical direction. Vertical direction is defined to be in-line with the feed or motion of the printing media. Logical units have no physical dimension.

#### 5.2.2.13 Counter units

This unsigned byte defines the unit of measurement used when reporting printer statistics. Table 15 shows the encoding for this field.

**Table 15— RDC-RS counter unit codes**

| Value | Description |
|---|---|
| 00 | Pages |
| 01 | Inches |
| 02 | Feet |
| 03 | Yards |
| 04 | Millimeters |
| 05 | Centimeters |
| 06 | Decimeters |
| 07 | Meters |
| 08 | Characters |
| 09 | Tens of characters |
| 0A | Hundreds of characters |
| 0B | Thousands of characters |
| 0C | Lines |
| 0D | Tens of lines |
| 0E | Hundreds of lines |
| 0F | Thousands of lines |
| 10 to FD | Reserved |
| FE | Other |
| FF | Unknown |

#### 5.2.2.14 Total installed memory

This unsigned dword specifies the total number of bytes of random access memory (RAM) installed in the printer. This value reflects the total amount of RAM and in no way implies how much of that memory is available for various functions within the printer such as communications buffers, font storage, form storage, etc. (The amount of memory available to the user using a given interpreter is available to the host through the Request Interpreter Characteristics command.) If the printer is unable to provide this information, the hex value FFFFFFFF will be returned.

#### 5.2.2.15 Maximum receive packet size

This unsigned word defines the maximum size (in bytes) of a data packet that can be received by the printer. If the printer receives a packet that is larger than the value specified, the printer shall reject the packet. A value less than hex 40 is not supported.

#### 5.2.2.16 Maximum number of outstanding commands

This unsigned word defines the number of commands that the printer can receive and be working on at any one time. An outstanding command means any command from the host where the host expects a response from the printer and for which the printer has not yet responded. If the host sends more commands than defined by this value, the printer shall reject the command and the host will have to re-send the command. A value of zero received in this field is defined to be equivalent to a value of one.

The printer’s responses to multiple commands from the host should be in the same order in which the commands were received.

#### 5.2.2.17 Number of logical units supported

This field contains the number, or count, of logical units (LUs) supported by the printer. The host can determine the LU type (e.g., page interpreter) by using the Request Logical Unit Characteristics command.

A value of 255 (hex FF) for this field indicates that the printer does not know how many interpreters are supported.

#### 5.2.2.18 Number of inputs supported

This unsigned integer contains the number, or count, of printing media (paper) inputs that the printer currently supports. This number may change as optional input devices are added and removed from the printer. The host can inquire about the details of each input by using the Request Input Characteristic subcommand of the RDC command.

A nonzero value for this parameter indicates that the printer shall support the Request Input Characteristics subcommand of the RDC command.

A value of 255 (hex FF) for this field indicates that the printer does not know how many inputs are supported.

#### 5.2.2.19 Number of outputs supported

This unsigned byte contains the number, or count, of output destinations for the printing media (paper) that the printer currently supports. This number may change as optional output devices are added and removed from the printer. The host can enquire about the details of each output by using the Request Output Characteristic subcommand of the RDC command.

A nonzero value for this parameter indicates that the printer shall support the Request Output Characteristics subcommand of the RDC command.

A value of 255 (hex FF) for this field indicates that the printer does not know how many outputs are supported.

#### 5.2.2.20 Number of options supported

This unsigned byte contains the number, or count, of options currently installed in the printer. An option may be an optional font, memory card, mass storage device, etc. This number may change as options are added and removed from the printer. The host can inquire about the details of each option by using the Request Options Characteristics subcommand of the RDC command.

A nonzero value for this parameter indicates that the printer shall support the Request Options Characteristics subcommand of the RDC command.

A value of 255 (hex FF) for this field indicates that the printer does not know how many options are supported.

#### 5.2.2.21 Printer language and character encoding

This unsigned byte defines the language and encoding that will be used for all messages sent to the host by the printer. Table 16 defines the values associated with the various languages used.

 

**Table 16— RDC-RS printer language and character encoding codes**

| Value | Description | Character Set |
|---|---|---|
| 00 | English | ISO 8859-1 (Latin 1) |
| 01 | German | ISO 8859-1 (Latin 1) |
| 02 | French | ISO 8859-1 (Latin 1) |
| 03 | Canadian French | ISO 8859-1 (Latin 1) |
| 04 | Spanish | ISO 8859-1 (Latin 1) |
| 05 | Italian | ISO 8859-1 (Latin 1) |
| 06 | Danish | ISO 8859-1 (Latin 1) |
| 07 | Norwegian | ISO 8859-1 (Latin 1) |
| 08 | Dutch | ISO 8859-1 (Latin 1) |
| 09 | Japanese | JIS X0208;1990 (16 bit) Kanji, JIS X0201;1976 (7&8 bit) Katakana |
| 0A | Korean | KS C 5601;1987 (16 bit) |
| 0B | Traditional Chinese | BIG-5 (16 bit) Taiwan Industry Standard |
| 0C | Simplified Chinese | GB2312;1980 (16 bit) |
| 0D | Swedish | ISO 8859-1 (Latin 1) |
| 0E | Japanese | Shift-JIS |
| 0F to FD | Reserved | |
| FE | Other | |
| FF | Unknown | |

#### 5.2.2.22 Length of product name

This is an unsigned byte that defines the length of the following ASCII string that specifies the Product Name.

#### 5.2.2.23 Product name

This is an ASCII string that defines the model and manufacturer of the printer. This is a human-readable string with the intention that an operator can understand the field without translation from the host. It is recommended that the field not include commas because within some operating systems commas are used as delimiters. This will allow a self-configuring driver to use this printer descriptive string in its user dialog and store it.

The purpose of this field is to indicate printer capabilities as opposed to selection of a printer. This field shall not be used to enable drivers to selectively support certain printers. Therefore, a driver shall not assume that if the printer answers that it is a “XYZ printer,” that all “XYZ printer” features are supported. Therefore, driver software shall not infer the command set based upon this one inquiry from the host.

The suggested format is

Manufacturer’s name (as on the logo):Product Name:Model Number

For example:

ABC Printer Company:XYZ Printer:12345

#### 5.2.2.24 Length of product revision

This is an unsigned byte that specifies the length of the following ASCII string that contains the product revision.

#### 5.2.2.25 Product revision

This is an ASCII string that is defined by the manufacturer of the printer to provide the revision level of the printer and whatever other data the manufacturer deems important. This field may or may not be human readable depending on manufacturer of the printer.

#### 5.2.2.26 Length of serial number

This is an unsigned byte that specifies the length of the following ASCII string that contains the serial number.

#### 5.2.2.27 Serial number

This is an ASCII string that can be used by the printer manufacturer to return the serial number of the printer.

#### 5.2.2.28 Maximum receive command packet size

This unsigned word defines the maximum size (in bytes) of a command packet that can be received by the printer. If the printer receives a command packet that is larger than the value specified, the printer shall reject the packet. A value less than hex 40 shall not be used.

### 5.2.3 Request Input Characteristics

The RDC Request Input Characteristics subcommand (RDC-RIC) provides the host system with the details of all print media input sources of the printer. The host system can request detailed information concerning each input individually or collectively.

The host requests information for all inputs or a particular input using the byte in the data field immediately following the subcommand. The format of the RDC-RIC command is shown in Table 17.

**Table 17— RDC-RIC command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Characteristics | Packet header |
| 1 | 02 | Subcommand: Request Input Characteristic | Data field |
| 1 | ubyte | Input ID<br>00 = Return Data About all Available Inputs<br>xx = Return Data About a specific Input | |

The identification numbers for the various input devices of the printer are assigned by the printer. The host system can obtain all the identification numbers initially by requesting data for all input features.

Table 18 shows the format of the data packet returned by the printer in response to the RDC-RIC command.

**Table 18— RDC-RIC response**

| Bytes | Value (Hex) | Description | Notes |
|---|---|---|---|
| 1 | | Request device characteristic | Packet header |
| 1 | 02 | Subcommand:Request input characteristic | Data field |
| 1 | ubyte | Number of inputs to follow | |
| 1 | ubyte | Input identification number | Input #1 |
| 4 | udword | Maximum input capacity–capacity units | |
| 1 | bbyte | Input features Byte #1 | |
| 1 | bbyte | Input features Byte #2 | |
| 1 | ubyte | Input feed type | |
| 1 | ubyte | Medium characteristic | |
| 1 | ubyte | Encoded medium size | |
| 2 | uword | Minimum size of medium across feed direction in horizontal logical units | |
| 2 | uword | Minimum size of medium in feed direction in vertical logical units | |
| 2 | uword | Maximum size of medium across feed direction in horizontal logical units | |
| 2 | uword | Maximum size of medium in feed direction in vertical logical units | |
| 2 | uword | Current size of medium across feed direction in horizontal logical units | |
| 2 | uword | Current size of medium in feed direction in vertical logical units | |
| 2 | uword | Top margin in vertical logical units | |
| 2 | uword | Extent of printable area in feed direction in vertical logical units | |
| 2 | uword | Left margin in horizontal logical units | |
| 2 | uword | Extent of printable area across feed direction in horizontal logical units | |
| 1 | ubyte | Length of input description (not including this byte) | |
| n | ASCII | Input description string | |
| 1 | ubyte | Length of medium description (not including this byte) | |
| n | ASCII | Medium description string | |
| ... | ... | ... | Input n |

#### 5.2.3.1 Number of inputs to follow

This unsigned byte defines how many input descriptions that the printer is returning in this message. If the host requested the input characteristics for a specific input, then this number will be a value of one. If data for all inputs is requested, then this number will be the total number of inputs currently installed on the printer.

#### 5.2.3.2 Input identification number

This unsigned byte is the logical identification number that the printer has assigned to a particular input. The identification number for a particular input is assigned by the printer and printer manufacturer. The identification numbers for functionally equivalent input devices may be different from printer to printer, depending on the manufacturer. For example, a manual input may have an identification number of 1 for one manufacturer and an identification number of 3 for another manufacturer or different printer model from the same manufacturer.

The printer shall assign identification numbers sequentially each time the printer is initialized or powered on. This means that if the number of input options is changed, a particular input may not have the same identification number as before the change in printer configuration. However, if the printer's configuration has not changed, the assigned identification number shall be the same following each initialization.

The number of input devices that the printer returns in the RDC-RS response indicates the total number of input devices installed on the printer at that time and included in the response to an RDC-RIC.

#### 5.2.3.3 Maximum input capacity—capacity units

This unsigned double word specifies the maximum capacity of this particular input feature. Units for this value are defined by the Counter units parameter, which is returned in the RDC-RS response. Hex FFFFFFFF indicates that the printer is unable to provide this value.

#### 5.2.3.4 Input features byte #1 and #2

These bit-encoded bytes enumerate features associated with a particular input source. The bit assignments for the defined features are shown in Table 19 and Table 20.

**Table 19— RDC-RIC Input Features Byte #1 encoding**

| Bit | Description | Notes |
|---|---|---|
| 0 | Security available zero=No, one = Yes | |
| 1–7 | Reserved; Must be zero | |

**Table 20— RDC-RIC Input Features Byte #2 encoding**

| Bit | Description | Notes |
|---|---|---|
| 0–7 | Reserved; must be zero | |

#### 5.2.3.5 Input feed type

This unsigned byte specifies the general feed type for the input feature. The values that corresponds to various input feed types are shown in Table 21.

**Table 21— RDC-RIC input feed type codes**

| Value | Description | Notes |
|---|---|---|
| 00 | Reserved | |
| 01 | Sheet feed auto | |
| 02 | Sheet feed manual | |
| 03 | Continuous roll | |
| 04 | Continuous fan fold | |
| 05 to FD | Reserved | |
| FE | Other | |
| FF | Unknown | |

#### 5.2.3.6 Medium characteristic

This unsigned byte specifies the general characteristic of the medium that is currently installed in the input feature. Table 22 specifies the defined values and their meaning.

**Table 22— RDC-RIC input medium codes**

| Value | Description | Notes |
|---|---|---|
| 00 | Reserved | |
| 01 | Paper | |
| 02 | Envelope | |
| 03 | Transparency | |
| 04 to FD | Reserved | |
| FE | Other | |
| FF | Unknown | |

#### 5.2.3.7 Encoded medium size

This unsigned byte specifies the size of the medium that is installed in a particular input source. The size of the medium is an encoded value used to specify a standard ISO size. The values and the corresponding ISO medium sizes are a subset of ISO/IEC 10175 [B4] and are shown in Table . If the size of the medium is unknown, the printer will return the hex value FF.

**Table 23— RDC-RIC input size codes**

| Value (hex) | Description | Millimeters | | | Approx. inches | | | Notes |
|---|---|---|---|---|---|---|---|---|
| 00 | Reserved | | | | | | | |
| 01 | A | | x | | 8 1/2 | x | 11 | US sizes |
| 02 | B | | x | | 11 | x | 17 | |
| 03 | C | | x | | 17 | x | 22 | |
| 04 | D | | x | | 22 | x | 34 | |
| 05 | E | | x | | 34 | x | 44 | |
| 06 | F | | x | | 44 | x | 68 | |
| 07 | Legal | | x | | 8 1/2 | x | 14 | |
| 08 | 1/2 Letter | | x | | 8 1/2 | x | 5 1/2 | |
| 09 | Executive | | x | | 7 1/4 | x | 101/2 | |
| 10 | 4A0 | 1682 | x | 2378 | 66 1/4 | x | 93 3/8 | ISO A series |
| 11 | 2A0 | 1189 | x | 1682 | 46 3/4 | x | 66 1/4 | |
| 12 | A0 | 841 | x | 1189 | 33 1/8 | x | 46 3/4 | |
| 13 | A1 | 594 | x | 841 | 23 3/8 | x | 33 1/8 | |
| 14 | A2 | 420 | x | 594 | 16 1/2 | x | 23 3/8 | |
| 15 | A3 | 297 | x | 420 | 11 3/4 | x | 16 1/2 | |
| 16 | A4 | 210 | x | 297 | 8 1/4 | x | 11 3/4 | |
| 17 | A5 | 148 | x | 210 | 5 7/8 | x | 8 1/4 | |
| 18 | A6 | 105 | x | 148 | 4 1/8 | x | 5 7/8 | |
| 19 | A7 | 74 | x | 105 | 2 7/8 | x | 4 1/8 | |
| 1A | A8 | 52 | x | 74 | 2 | x | 2 7/8 | |
| 1B | A9 | 37 | x | 52 | 1 1/2 | x | 2 | |
| 1C | A10 | 26 | x | 37 | 1 | 1x | 1 1/2 | |
| 1D to 1F | Reserved | | | | | | | |
| 20 | RA0 | 860 | x | 1220 | 33 7/8 | x | 48 1/8 | |
| 21 | RA1 | 610 | x | 860 | 24 1/8 | x | 33 7/8 | |
| 22 | RA2 | 430 | x | 610 | 17 | x | 24 1/8 | |
| 23 | SRA0 | 900 | x | 1280 | 35 1/2 | x | 50 3/8 | |
| 24 | SRA1 | 640 | x | 900 | 25 1/4 | x | 35 1/2 | |
| 25 | SRA2 | 450 | x | 640 | 17 7/8 | x | 25 1/4 | |
| 26 to 2F | Reserved | | | | | | | |

**Table 23— RDC-RIC input size codes (Continued)**

| Value (hex) | Description | Millimeters | | | Approx. inches | | | Notes |
|---|---|---|---|---|---|---|---|---|
| 30 | B0 | 1000 | x | 1414 | 39 3/8 | x | 55 5/8 | ISO B series |
| 31 | B1 | 707 | x | 1000 | 27 7/8 | x | 39 3/8 | |
| 32 | B2 | 500 | x | 707 | 19 5/8 | x | 27 7/8 | |
| 33 | B3 | 353 | x | 500 | 12 7/8 | x | 19 5/8 | |
| 34 | B4 | 250 | x | 353 | 9 7/8 | x | 12 7/8 | |
| 35 | B5 | 176 | x | 250 | 7 | x | 9 7/8 | |
| 36 | B6 | 125 | x | 176 | 5 | x | 7 | |
| 37 | B7 | 88 | x | 125 | 3 1/2 | x | 5 | |
| 38 | B8 | 62 | x | 88 | 2 1/2 | x | 3 1/2 | |
| 39 | B9 | 44 | x | 62 | 1 3/4 | x | 2 1/2 | |
| 3A | B10 | 31 | x | 44 | 1 1/4 | x | 1 3/4 | ISO B series |
| 3B to 3F | Reserved | | | | | | | |
| 40 | C0 | 917 | x | 1297 | 36 1/8 | x | 51 | ISO C series |
| 41 | C1 | 648 | x | 917 | 25 1/2 | x | 36 1/8 | |
| 42 | C2 | 458 | x | 648 | 18 | x | 25 1/2 | |
| 43 | C3 | 324 | x | 458 | 12 3/4 | x | 18 | |
| 44 | C4 | 229 | x | 324 | 9 | x | 12 3/4 | |
| 45 | C5 | 162 | x | 229 | 6 3/8 | x | 9 | |
| 46 | C6 | 114 | x | 162 | 4 1/2 | x | 6 3/8 | |
| 47 | C7 | 81 | x | 114 | 3 1/4 | x | 4 1/2 | |
| 48 | C8 | 57 | x | 81 | 2 1/4 | x | 3 1/4 | |
| 49 to 4F | Reserved | | | | | | | |
| 50 | Envelope C3 | 324 | x | 458 | | x | | ISO envelopes |
| 51 | Envelope B4 | 250 | x | 353 | | x | | |
| 52 | Envelope C4 | 229 | x | 324 | | x | | |
| 53 | Envelope B5 | 176 | x | 250 | | x | | |
| 54 | Envelope C5 | 162 | x | 229 | | x | | |
| 55 | Envelope B6/C4 | 125 | x | 324 | | x | | |
| 56 | Envelope B6 | 125 | x | 176 | | x | | |
| 57 | Envelope C6 | 114 | x | 162 | | x | | |
| 58 | Envelope DL | 110 | x | 220 | | x | | |
| 59 | Envelope C7/6 | 81 | x | 162 | | x | | |
| 5A | Envelope C7 | 81 | x | 114 | | x | | |
| 5B to 5F | Reserved | | | | | | | |

**Table 23— RDC-RIC input size codes (Continued)**

| Value (hex) | Description | Millimeters | | | Approx. inches | | | Notes |
|---|---|---|---|---|---|---|---|---|
| 60 | Post card | 100 | x | 148 | | | | Japanese |
| 61 to 6F | Reserved | | | | | | | |
| 70 | B0 | 1030 | x | 1456 | 40.55 | x | 57.32 | JIS B series |
| 71 | B1 | 728 | x | 1030 | 28.66 | x | 40.55 | |
| 72 | B2 | 515 | x | 728 | 20.28 | x | 28.66 | |
| 73 | B3 | 364 | x | 515 | 14.33 | x | 20.28 | |
| 74 | B4 | 257 | x | 364 | 10.12 | x | 14.33 | |
| 75 | B5 | 182 | x | 257 | 7.17 | x | 10.12 | JIS B series |
| 76 | B6 | 128 | x | 182 | 5.04 | x | 7.17 | |
| 77 | B7 | 91 | x | 128 | 3.58 | x | 5.04 | |
| 78 | B8 | 64 | x | 91 | 2.52 | x | 3.58 | |
| 79 | B9 | 45 | x | 64 | 1.77 | x | 2.52 | |
| 7A | B10 | 32 | x | 45 | 1.26 | x | 1.77 | |
| 7B to 7F | Reserved | | | | | | | |
| 80 | Envelope 7-3/4 | | | | 3 7/8 | x | 7 1/2 | U.S. envelopes |
| 81 | Envelope 9 | | | | 3 7/8 | x | 8 7/8 | |
| 82 | Envelope 10 | | | | 4 1/4 | x | 9 1/2 | |
| 83 to FD | Reserved | | | | | | | |
| FE | Other | | | | | | | |
| FF | Unknown | | | | | | | |

### 5.2.3.8 Minimum size of medium across feed direction

This unsigned word specifies the minimum size of the print medium measured orthogonal to (across) the feed direction of the medium for a particular input feature. Units for this value are specified by the printer in Horizontal Logical Units. This unit value is returned in the RDC-RS response. A Horizontal Logical Unit is the unit of measure used to define the Horizontal Print Resolution also returned in the RDC-RS response. (For example, a Horizontal Logical Unit could be dots or pels for a laser or ink jet printer and might be expressed as characters for a fully formed character printer such as a daisy wheel printer.) If the size is unknown, the printer returns a hex value of FFFF.

### 5.2.3.9 Minimum size of medium in feed direction

This unsigned word specifies the minimum size of the print medium measured in the feed or motion direction of the medium for a particular input feature. Units for this value are specified by the printer in Vertical Logical Units. This unit value is returned in the RDC-RS response. A Vertical Logical Unit is the unit of measure used to define the Vertical Print Resolution also returned in the RDC-RS response. (For example, a Vertical Logical Unit could be dots or pels for a laser or ink jet printer and might be expressed as lines for a fully formed character printer such as a daisy wheel printer.) If the size is unknown, the printer returns a hex value of FFFF.

#### 5.2.3.10 Maximum size of medium across feed direction

This unsigned word specifies the maximum size of the print medium measured orthogonal to (across) the feed direction of the medium for a particular input feature. Units for this value are specified by the printer in Horizontal Logical Units. If the size is unknown, the printer returns a hex value of FFFF.

#### 5.2.3.11 Maximum size of medium in feed direction

This unsigned word specifies the maximum size of the print medium measured in the feed or motion direction of the medium for a particular input feature. Units for this value are specified by the printer in Vertical Logical Units. If the size is unknown, the printer returns a hex value of FFFF.

#### 5.2.3.12 Current size of medium across feed direction

This unsigned word specifies the current size of the print medium measured orthogonal to (across) the feed direction of the medium for a particular input feature. Units for this value are specified by the printer in Horizontal Logical Units. If the size is unknown, the printer returns a hex value of FFFF.

#### 5.2.3.13 Current size of medium in feed direction

This unsigned word specifies the current size of the print medium measured in the feed or motion direction of the medium for a particular input feature. Units for this value are specified by the printer in Vertical Logical Units. If the size is unknown, the printer returns a hex value of FFFF.

#### 5.2.3.14 Top margin

This unsigned word defines the top unprintable area for this input as shown in Figure 4. The units for this value are specified by the printer in Vertical Logical Units.

A hex value of FFFF indicates that the printer does not have or cannot determine this value.

Note that this value specifies the limitation of the printing mechanism and that particular interpreters may place additional limitations upon the margins. The margins supported by the various interpreters can be obtained by using the Request Interpreter Characteristic command.

#### 5.2.3.15 Extent of printable area in the feed direction

This unsigned word defines the printable area in the feed direction for this input as shown in Figure 4. The units for this value are specified by the printer in Vertical Logical Units.

A hex value of FFFF indicates that the printer does not have or cannot determine this value.

Note that this value specifies the limitation of the printing mechanism and that particular page interpreters may place additional limitations upon the margins. The margins supported by the various interpreters can be obtained by using the Request Interpreter Characteristic command.

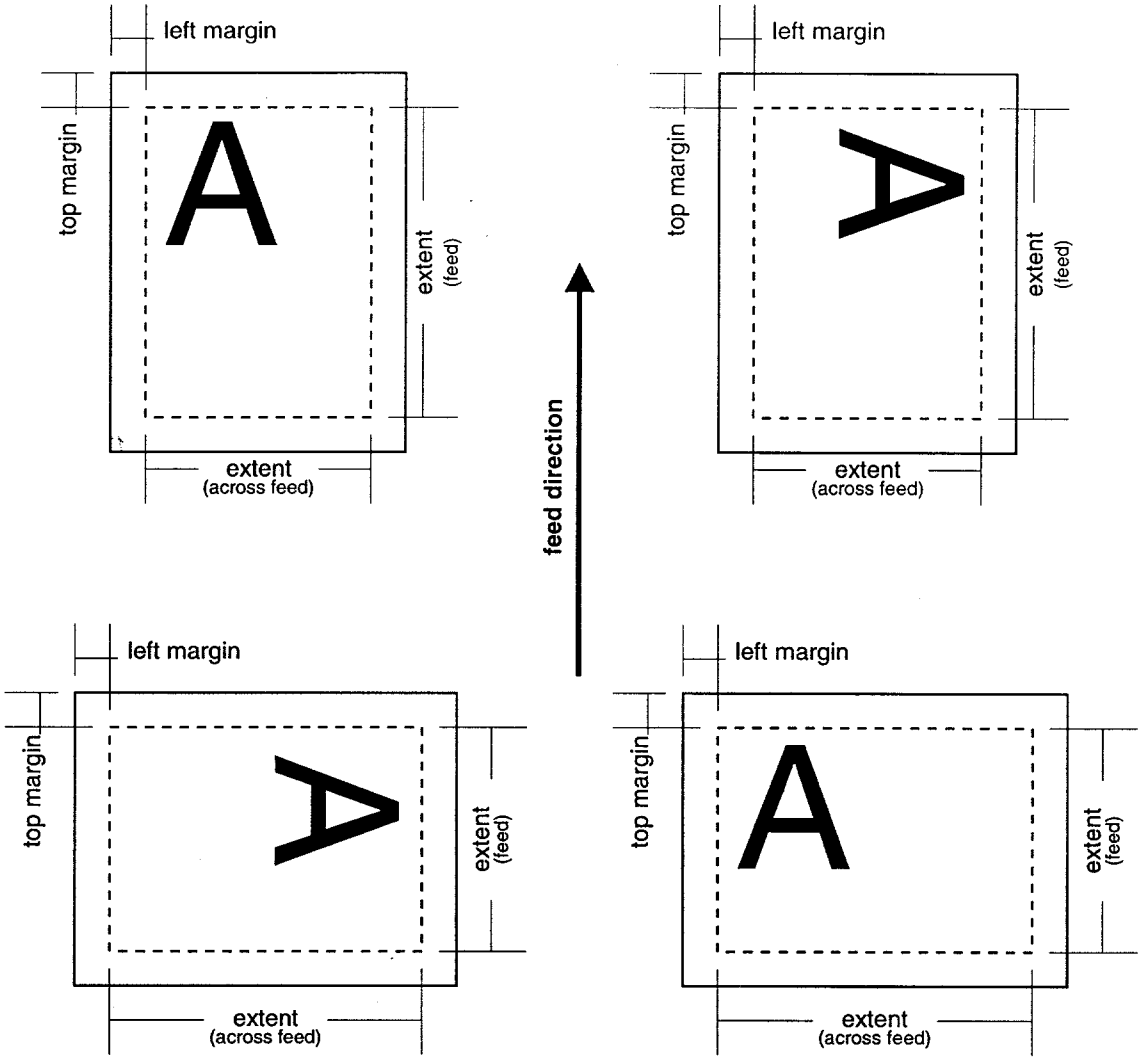


**Figure 4— Media margins and printable areas**

### 5.2.3.16 Left margin

This unsigned word defines the left unprintable area for this input as shown in Figure 4. The units for this value are specified by the printer in Horizontal Logical Units.

A hex value of FFFF indicates that the printer does not have or cannot determine this value.

Note that this value specifies the limitation of the printing mechanism and that particular page interpreters may place additional limitations upon the margins. The margins supported by the various interpreters can be obtained by using the Request Interpreter Characteristic command.

### 5.2.3.17 Extent of the printable area across feed direction

This unsigned word defines the printable area across the feed direction for this input as shown in Figure 4. The units for this value are specified by the printer in Horizontal Logical Units.

A hex value of FFFF indicates that the printer does not have or cannot determine this value.

Note that this value specifies the limitation of the printing mechanism and that particular page interpreters may place additional limitations upon the margins. The margins supported by the various interpreters can be obtained by using the Request Interpreter Characteristic command.

#### 5.2.3.18 Length of input description

This unsigned byte defines the length of the following Input Description ASCII string. If the value of this byte is zero, then there will be no Input Description String.

#### 5.2.3.19 Input description string

This is an ASCII string that can be used as a description of input feature (device) on the printer. The content of this field is printer-implementation dependent. This field is intended to provide a means of providing a human-readable description of the input feature. An example would be “Automatic Envelope Feeder” or “Standard Input Tray.”

#### 5.2.3.20 Length of medium description

This unsigned byte defines the length of the following Medium Description ASCII string. If the value of this byte is zero, then there will be no Medium Description String.

#### 5.2.3.21 Medium description string

This is an ASCII string that can be used as a description of input medium. The content of this field is printer-implementation dependent. This field is intended to provide a means of providing a human-readable description of the input feature. An example of the use of this description would be to describe the current medium that an operator has installed in the input, such as “IRS 1040—Long.”

### 5.2.4 Request Output Characteristic

The RDC Request Output Characteristic subcommand (RDC-ROC) provides the host system with the details of all print media output features of the printer. The host system can request detailed information concerning each output individually or collectively.

The host requests information for all outputs or a particular output using the byte in the data field immediately following the subcommand. The format of the RDC-ROC command is shown in Table 24.

**Table 24— RDC-ROC command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request device characteristic | Packet header |
| 1 | 03 | Subcommand: Request output characteristic | Data field |
| 1 | ubyte | Output ID<br>00 = Return data about all available outputs<br>xx = Return data about a specific output | |

The identification numbers for the various output devices of the printer are assigned by the printer. The host system can obtain all the identification numbers initially by requesting data for all output features.

Table 25 shows the format of the data packet returned by the printer in response to the RDC-ROC command.

 

**Table 25— RDC-ROC response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request device characteristic | Packet header |
| 1 | 03 | Subcommand: Request output characteristic | Data field |
| 1 | ubyte | Number of output features to follow | |
| 1 | ubyte | Output identification number | Output #1 |
| 1 | ubyte | Number of output positions | |
| 4 | udword | Maximum capacity per output position | |
| 1 | bbyte | Standard output features supported Byte 1 | |
| 1 | bbyte | Standard output features supported Byte 2 | |
| 1 | ubyte | Length of output description | |
| n | ASCII | Output description string | |
| ... | ... | ... | Output n |

#### 5.2.4.1 Number of output features to follow

This unsigned byte defines how many output descriptions that the printer is returning in this message. If the host requested the output characteristics for a specific output position, then this number will be one. If data for all output positions is requested, then this number will be the total number of outputs currently installed on the printer.

#### 5.2.4.2 Output identification number

This unsigned byte is the logical identification number that the printer has assigned to a particular output. The identification number for a particular output is assigned by the printer and printer manufacturer. The identification numbers for functionally equivalent output devices may be different from printer to printer, depending on the manufacturer. For example, an output collator may have an identification number of 1 for one manufacturer and an identification number of 4 for another manufacturer or different printer model from the same manufacturer.

The printer shall assign identification numbers sequentially each time the printer is initialized or powered on. This means that if the number of output options is changed, a particular output may not have the same identification number as before the change in printer configuration. However, if the printer's configuration has not changed, the assigned identification number shall be the same following each initialization.

The number of output devices that the printer returns in the RDC-RS response indicates the total number of output devices installed on the printer at that time and included in the response to an RDC-ROC.

#### 5.2.4.3 Number of output positions

This unsigned byte specifies the number of output positions for this particular output device. A ten-bin collator is an example of a multiple position output device and would have a hex value of A in this field.

#### 5.2.4.4 Maximum capacity per output position

This unsigned double word specifies the maximum capacity for each output position of this particular output feature. Units for this value are defined by the Counter units parameter that is returned in the RDC-RS response. The hex value FFFFFFFF indicates that the printer is unable to provide this value.

#### 5.2.4.5 Standard output features supported Bytes 1 and 2

These bit-encoded bytes define the availability of certain standard functions or features on the output device. Table 26 and Table 27 define the meaning of the individual bits in these bytes.

**Table 26— RDC-ROC output features Byte #1 encoding**

| Bit | Description | | |
|---|---|---|---|
| 0 | Face up | zero=No | one=Yes |
| 1 | Output separation capability | zero=No | one=Yes |
| 2 | Security available | zero=No | one=Yes |
| 3 | Bursting available | zero=No | one=Yes |
| 4 | Collation available | zero=No | one=Yes |
| 5 | Face down | zero=No | one=Yes |
| 6 | Level sensing available | zero=No | one=Yes |
| 7 | Reserved | | |

**Table 27— RDC-ROC output features Byte #2 encoding**

| Bit | Description | | |
|---|---|---|---|
| 0 | Stitching available | zero=No | one=Yes |
| 1 | Binding available | zero=No | one=Yes |
| 2 | Punching Available | zero=No | one=Yes |
| 3 | Reserved | | |
| 4 | Reserved | | |
| 5 | Reserved | | |
| 6 | Reserved | | |
| 7 | Additional Finishing Options Available | zero=No | one=Yes |

#### 5.2.4.6 Length of output description

This unsigned byte defines the length of the following Output Description ASCII string. If the value of this byte is zero, then there will be no Output Description String.

#### 5.2.4.7 Output description string

This is an ASCII string that can be used as a description of the output device. The content of this string is printer-implementation dependent. This field is intended to provide a human-readable description of the output device.

### 5.2.5 Request option characteristics

The RDC Request Option Characteristics subcommand (RDC-ROP) provides the host system with data about options that are installed on the printer other than input and output options. Options may be either internal or external to the printer. Examples of options are font cartridges, RAM memory expansion, hard disks, FAX modems, etc.

The format of the RDC-ROP command is shown in Table 28.

**Table 28— RDC-ROP command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request device characteristics | Packet header |
| 1 | 04 | Subcommand: Request option characteristics | Data field |

Table 29 shows the format of the data packet returned by the printer in response to the RDC-ROP command.

**Table 29— RDC-ROP response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request device characteristics | Packet header |
| 1 | 04 | Subcommand: Request option characteristics | Data field |
| 1 | ubyte | Number of option characteristics to follow | |
| 1 | ubyte | Length of Option #1 description (n) | Option #1 |
| n | ASCII string | Option #1 description string | |
| ... | ... | ... | Option n |

#### 5.2.5.1 Number of options

This unsigned byte specifies the number of options installed in this printer and included in this response to the RDC-ROP.

#### 5.2.5.2 Length of option description

This unsigned byte defines the length of the following Option Description ASCII string. If the value of this byte is zero, then there will be no Option Description String.

#### 5.2.5.3 Option description string

This is an ASCII string that can be used to describe a particular installed option. The content of this string is in the form of ASCII ordered pairs. The ordered pairs should be of the form key:value;key:value;... Any characters except colon (:), comma (,), and semicolon (;) may be included as part of the key or value strings. Any leading or trailing white space (SPACE, TAB, VTAB, CR, LF, FF) in the string should be ignored by the parsing program but still counted as part of the overall string length.

### 5.3 Request Interpreter Characteristics (RIC)

#### 5.3.1 Overview

The RIC command provides a mechanism for the host application to obtain information, such as paper trays supported, printable area, etc., for LUs defined as page control interpreters.

A given interpreter within the printer may not provide support for all the features (primarily inputs, outputs, and fonts) that the printer has provided. The fact that it is possible to have different levels of support provided by each of the different interpreters within a given printer necessitates this command.

Using the RIC command, specific interpreter characteristic information can be retrieved from the printer. Table 30 shows the various categories of information that can be retrieved.

**Table 30— RIC categories**

| Subcommand category | Description |
|---|---|
| Summary | Request a summary of the functional support provided by a specific interpreter. |
| Font Details | Request a list of printer fonts that this interpreter supports. |
| Input Characteristics | Request information regarding the input features that this interpreter supports. |
| Output Characteristics | Request information regarding the output features that this interpreter supports. |

The host queries for the desired information by sending a RIC command with a subcommand in the data field that specifies the category of information desired followed by an interpreter ID in the data field that specifies the interpreter ID for which the information is requested. The subcommand values for the various categories of information are shown in Table 31.

**Table 31— RIC subcommands**

| Value (hex) | Description |
|---|---|
| 00 | Summary Information |
| 01 | Font Details |
| 02 | Input Characteristics |
| 03 | Output Characteristics |
| 04 to FF | Reserved |

### 5.3.2 Summary Information

The RIC Summary Information subcommand (RIC-SI) is used by a host to request summary information regarding the capabilities of a given interpreter within the printer. Table 32 shows the format of the command sent from the host to the printer.

**Table 32— RIC-SI command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Interpreter Characteristics | Packet header |
| 1 | 00 | Subcommand: Summary Information | Data field |
| 1 | ubyte | Logical Unit Number (Interpreter)<br>00 Return data for the current interpreter assigned to LU #0<br>xx Specific LU # requested<br>FF Return data for all LU numbers | |

This host message is meaningful only if the reply required flag (bit 4 of the flag byte) is set. With the reply required flag set, the printer will respond to the RIC-SI command with a message like the one depicted in Table 33 provided the error flag (bit 7 of the flag byte) is not set in the response the host receives from the printer.

**Table 33— RIC-SI response**

| Bytes | Value | Description | Notes |
|---|---|---|---|
| 5 | | Request Interpreter Characteristics | Packet header |
| 1 | 00 | Subcommand: Summary Information | Data field |
| 1 | ubyte | Number of Logical Unit Numbers (Interpreters) for which data is returned in this message | |
| 1 | ubyte | #1 Logical Unit Number | Logical Unit #1 |
| 1 | bbyte | #1 Interpreter Features Summary | |
| 1 | ubyte | Reserved | |
| 4 | udword | Maximum amount of free memory for #1 interpreter (bytes) | |
| 2 | uword | Number of Fonts for #1 interpreter | |
| 1 | ubyte | Number of Input Sources for #1 interpreter | |
| 1 | ubyte | Number of Output Bins for #1 interpreter | |
| 2 | uword | Horizontal Print Resolution for #1 interpreter | |
| 2 | uword | Vertical Print Resolution for #1 interpreter | |
| 1 | ubyte (n) | Length of #1 Interpreter Name | |
| n | ASCII | #1 Interpreter Name | |
| ... | ... | ... | Logical Unit n |

If the error flag is set, a data error (bits 2 and 3 of the flag byte) indicates that an undefined logical unit number was selected. No summary information will be returned.

#### 5.3.2.1 Number of logical unit numbers

This unsigned byte indicates the number of logical units for which summary information is being returned in this printer response. If summary information for a specific interpreter was requested, this number will be 1.

#### 5.3.2.2 Logical unit number

This unsigned byte indicates the LU number that the printer will use to reference this interpreter.

#### 5.3.2.3 Interpreter features summary

The Interpreter Features Summary byte provides the host with an indication of the capabilities of this interpreter in reporting information and/or existing concurrently with other interpreters within the printer.

Bit 0, if cleared, indicates that any information returned in the remaining seven bits is useless, as the printer does not support this field and the information returned within it.

If bit 0 is set:

- Bit 1 set indicates that the printer will support interpreter state save on switching. This implies that if the printer is switched from interpreter “A” to interpreter “B”; when it is switched back to “A,” the printer will be in the same state as when it switched to “B” (i.e., all downloaded fonts will still be present).
- Bit 2 set indicates that this interpreter can run simultaneously with the other interpreters in this printer. Interpreters “A” and “B” can both be active at the same time, though most likely only one can be in control of the print engine.
- Bit 3 set indicates that this interpreter, through the IEEE 1284.1 component, can report resident font information. This indicates the font information returned via the Font Details subcommand will include resident font information.
- bit 4 set indicates that this interpreter, through the IEEE 1284.1 component, can report font card font information. This indicates the font information returned via the Font Details subcommand will include font card font information.
- Bit 5 set indicates that this interpreter, through the IEEE 1284.1 component, can report download font information. This indicates the font information returned via the Font Details subcommand will include download font information.

  Bits 6 and 7 are reserved for future use.

#### 5.3.2.4 Maximum amount of free memory for this interpreter

The value of this unsigned double word indicates the maximum amount of free memory available to this interpreter at this instant in time. The hex value FFFFFFFF indicates that the printer is not able to supply this information.

#### 5.3.2.5 Number of fonts

This unsigned word indicates the number of fonts currently available to this interpreter. The hex value FFFF indicates that the printer is not able to supply this information.

#### 5.3.2.6 Number of input sources

This unsigned byte indicates the number of input trays that this interpreter supports on this printer. The hex value FF indicates that the printer is not able to supply this information.

#### 5.3.2.7 Number of output bins

This unsigned byte indicates the number of output bins that this interpreter supports on this printer. The hex value FF indicates that the printer is not able to supply this information.

#### 5.3.2.8 Horizontal print resolution

This unsigned word defines the maximum horizontal marking resolution that this interpreter supports. Horizontal resolution is defined as the resolution in the direction orthogonal (90 degrees) to the feed or motion of the print medium. The units for this measurement are defined by the length units reported in the RDC-RS response.

#### 5.3.2.9 Vertical print resolution

This unsigned word defines the maximum vertical marking resolution that this interpreter supports. Vertical resolution is defined as the resolution in the feed or motion direction of the print medium. The units for this measurement are defined by the length units reported in the RDC-RS response.

#### 5.3.2.10 Length of interpreter name

This unsigned byte defines the length of the ASCII string Interpreter Name in bytes.

#### 5.3.2.11 Interpreter Name

This ASCII string is the name of the interpreter as known by this printer. It may include anything the printer manufacturer desires. However, the following is recommended: name:level:comment.

For example,

HAL Amateur Writer:2001:Daisy-Daisy

### 5.3.3 Font Details

The RIC Font Details subcommand (RIC-FD) is used by a host to request a list of the fonts available to a specific interpreter. Two additional bytes in the data field are used to limit the data to a specific font storage type and/or font storage type identifier. Table 34 shows the format of the command sent from the host to the printer.

**Table 34— RIC-FD command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Interpreter Characteristics | Packet header |
| 1 | 01 | Subcommand: Font Details | Data field |
| 1 | ubyte | Logical Unit Number (Interpreter) | |
| 1 | ubyte | Font Storage Type (see Table 35) | |
| 1 | ubyte | Font Storage Type Identifier | |

This host message is meaningful only if the reply required flag (bit 4 of the flag byte) is set. With the reply required flag set, the printer will respond to the Font details subcommand with a message like the one depicted in Table 36 if the error flag (bit 7 of the flag byte) is not set in the response the host receives from the printer.

#### 5.3.3.1 Logical unit number

This unsigned byte indicates the LU number that the printer will use to reference this interpreter and is returned in response to the RIF-FD command.

#### 5.3.3.2 Font storage type

This unsigned byte allows the host to limit the font information returned to a particular font storage type. Table 35 defines the acceptable values for this field and maps them to the specific font storage type.

**Table 35— RIC-FD font storage type codes**

| Value (hex) | Description |
|---|---|
| 00 | Indicates that this font is stored in permanent memory within the printer. The exact media is not important. It may be in ROM or on a nonremovable font card, but it cannot be removed or deleted. |
| 01 | Indicates that this font is stored on a removable font card currently inserted in one of the printer's font card slots. This type of font will not affected by power sequences and cannot be deleted by the software. However, a user/operator could asynchronously remove the font at any time. |
| 02 | Indicates that this font is stored on a nonvolatile hard disk within the printer. This type of font will not be affected by power sequences but can be removed under software control. |
| 03 | Indicates that this font is stored in system RAM. It is volatile to power sequences, interpreter switching (if state preservation is not supported), or software deletion. |
| 04 | Indicates that this font is stored on a nonvolatile flash memory device within the printer. This type of font will not be affected by power sequences but can be removed under software control. Flash memory devices and hard disks are similar devices. It is expected that flash memory devices will have significantly faster access times than hard disks. |
| 05 | Indicates that this font is stored on a ROM SIMM module in the printer. This type of font will not be affected by power sequences but can be removed while power is off. |
| 06 to FC | Reserved |
| FD | Other |
| FE | This hex value is only valid in the printer response. It is used to tell the host that the printer has completed the font details list prematurely. See 5.3.3.9 for additional information. |
| FF | In the host command, this hex value is used to instruct the printer to return information about all the fonts in the printer (i.e., all font storage types). In the printer response, this hex value is used by the printer to indicate that it cannot provide the font storage type. |

### 5.3.3.3 Font storage type identifier

This unsigned byte is used to enumerate multiple instances of a particular font storage type. The value 0 is used to denote the highest priority instance of a particular font storage type (e.g., removable font card #1). The hex value FE is used to denote the lowest priority instance. The hex value FF is used by the host to request font information for all instances of a font storage type.

When a font storage type = *All locations* (hex FF) has been specified in the prior byte, this byte will be set to hex FF by the host and ignored by the printer.

**Table 36— RIC-FD response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Interpreter Characteristics | Packet header |
| 1 | 01 | Subcommand: Font Details | Data field |
| 1 | ubyte | Logical Unit Number (Interpreter) | |
| 2 | uword | Number of fonts returned | |
| 1 | ubyte (a) | Font Description Type Length | Font #1 |
| a | ASCII | Font Description Type | |
| 1 | ubyte | #1 Font Storage Type | |
| 1 | ubyte | #1 Font Storage Type Identifier | |
| 2 | uword (n) | #1 Font Description Length | |
| n | ASCII | #1 Font Description | |
| ... | ... | ... | Font n |

If the error flag is set, a data error (bits 2 and 3 of the flag byte) indicates that an undefined logical unit number, an unknown font storage type, or an undefined font storage type identifier was selected. No data will be returned.

### 5.3.3.4 Number of fonts returned

This unsigned word indicates the maximum number of fonts returned in this message. This is an all-inclusive list of the fonts available to this specific interpreter currently available within this printer for the requested font storage type(s) and font storage type identifier(s).

### 5.3.3.5 Font Description Type length

This unsigned byte indicates the length in bytes of the Font Description Type ASCII string.

#### 5.3.3.6 Font description type

This ASCII string will indicate the form of the Font Description field for every font returned in this response. Two values are defined by this standard. Additional font description types may be defined as manufacturer extensions. The case-sensitive values for the two values that are supported in this field are

PCL

PostScript

#### 5.3.3.7 Font Description length

This unsigned word indicates the length in bytes of the following Font Description field.

#### 5.3.3.8 Font Description

This ASCII field is used to return the description of a font as known by this specific interpreter within this specific printer to the host. This field will consist of ASCII ordered pairs, which will be in the form of variable name:data;variable name:data;... If a variable name is not important to the description of a particular font, then it should not be returned in the description field. All variable names are case-sensitive. Table 37 and Table 38 enumerate the variable names for PCL and PostScript, respectively.

**Table 37— PCL variable names**

| Variable Name | Data | Notes |
|---|---|---|
| name | Name of the font as known by the interpreter | Optional |
| technology | *bitmap* if bitmap font<br>*Intellifont* if Intellifont outline<br>*TrueType* if TrueType® outline<br>*Type1* if Type 1 outline | Required |
| ssid | Symbol set ID supported by the font. Same value as would be accepted in the PCL Symbol Set command, ESC(id | Required symbol set or character complement |
| ccid | Character complement representation of symbol set support if the font supports more than one symbol set. Eight-Byte bit-encoded field for unbounded fonts. Any data that is not typically represented in ASCII form must be returned in ASCII representation of the binary data (i.e., a hex FF would be returned as x'4646'). | Either the symbol set or character complement must be returned |
| spacing | 0 if fixed pitch<br>1 if proportional spacing<br>Same value as would be accepted in the PCL Primary Spacing command, ESC(s#O | Required |
| pitch | Font pitch (typically only returned for fixed pitch fonts). Same value as would be accepted in the PCL Primary Pitch command, ESC(s#H | Required for fixed pitch bitmapped fonts |
| height | Font height (typically only returned for proportional spacing or typographic fonts). Same value as would be accepted in the PCL Primary Height command, ESC(s#V | Required for proportional spacing or typographic fonts |
| style | Font style or posture class. Same value as would be accepted in the PCL Primary Style command, ESC(s#S | Required |
| weight | Stroke weight or weight class. Same value as would be accepted in the PCL Primary Stroke Weight command, ESC(s#B | Required |
| typeface | Typestyle number. Same value as would be accepted in the PCL Primary Typeface command, ESC(s#T. | Required |
| resolution | Horizontal print resolution, across marking direction, in length units that a font may be limited to. (typically only returned for bitmap fonts) | Optional |
| orientation | Specific orientation that a font may be limited to (typically only returned for bitmap font). Same value as would be accepted in the PCL Page Orientation command, ESC(l#O. | Optional |
| quality | Specific quality that a font may be limited to. Same value as would be accepted in the PCL Select Quality command, ESC(s#Q. | Optional |
| family | Family name for typeface | Optional |
| position | Positional order of the font on the source or the download ID | Optional |

**Table 38— PostScript Variable Names**

| Variable Name | Data | Notes |
|---|---|---|
| name | Name of the font as known by the interpreter. Same name as used by the PostScript findfont operator. | Required |
| technology | *TrueType* if TrueType outline<br>*TypeX* where *X* is 1 if Type 1 outline<br>3 if Type 3 outline<br>5 if Type 5 outline<br>42 if Type 42 outline<br>*Unknown* if unknown | Required |

### 5.3.3.9 Font exception handling

The printer, to the best of its ability, shall accurately report the *number of fonts returned*. However, the number of fonts available may change during the building and transmitting of the response depending on data stream traffic and particular printer hardware and firmware implementations.

One exception could occur more readily in printers with large font storage capacity if, sometime between the time the printer has calculated the number of fonts for which details will be returned and the completion of sending the details list, some of the fonts are removed/deleted (by the data stream) before the printer has captured the information to be returned. In this case there would be fewer fonts returned than were actually reported at the beginning of this printer to host response. For this case the printer, after returning the last valid font detail record, shall return a detail record indicating a *Font Storage Type* = FE, *Font Storage Type Identifier* = FF, and *Font Description Length* = 0 indicating that the list has ended prematurely.

Another exception could occur if sometime between the time the printer has calculated the number of fonts for which details will be returned and the completion of sending the details list, additional fonts were added (again via the data stream). In this case there would be more fonts to be returned than were actually reported at the beginning of this printer to host response. For this case the printer, in its response to the host, shall return no more detail records than the number which was reported.

### 5.3.4 Input Characteristics

The RIC Input Characteristics subcommand (RIC-IC) is used by the host to request input characteristics for a specific interpreter in the printer. An additional byte in the data field is used by the host to limit the data returned to a specific input device. Table 39 shows the format of the command sent from the host to the printer.

**Table 39— RIC-IC command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Interpreter Characteristics | Packet header |
| 1 | 02 | Subcommand: Input Characteristics | Data field |
| 1 | ubyte | Logical Unit Number (Interpreter) | |
| 1 | ubyte | Input ID<br>00 = Return Data About All Available Inputs<br>xx = Return Data About A Specific Input ID | |

This host to printer message is meaningful only if the reply required flag (bit 4 of the flag byte) is set. With the reply required flag set, the printer shall respond to the Input Characteristics subcommand with a message like the one depicted in Table 40 provided that the error flag (bit 7 of the flag byte) is not set in this printer message.

**Table 40— RIC-IC response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Interpreter Characteristics | Packet header |
| 1 | 02 | Subcommand: Input Characteristics | Data field |
| 1 | ubyte | Logical Unit Number (Interpreter) | |
| 1 | ubyte | Number of input IDs for which Data is Returned | |
| 1 | ubyte | #1 Input ID number | Input #1 |
| 2 | uword | #1 Minimum Size of Medium Across Feed Direction | |
| 2 | uword | #1 Minimum Size of Medium in Feed Direction | |
| 2 | uword | #1 Maximum Size of Medium Across Feed Direction | |
| 2 | uword | #1 Maximum Size of Medium in Feed Direction | |
| 2 | uword | #1 Input Top Margin | |
| 2 | uword | #1 Input Extent of printable area in the feed direction | |
| 2 | uword | #1 Input Left Margin | |
| 2 | uword | #1 Input Extent of printable area across the feed direction | |
| ... | ... | ... | Input n |

If the error flag is set, a data error indicates that the data field did not contain a valid logical unit number (interpreter) or a valid input ID for the logical unit number requested. No input characteristic data shall be returned.

#### 5.3.4.1 Number of input IDs for which data is returned

This unsigned byte indicates the number of input IDs for which data is returned in this response. The rest of the data for this printer to host message will describe the tray(s) and the extent supported, such as length and nonprintable areas.

The value 0 indicates that the printer cannot provide this information for this interpreter.

#### 5.3.4.2 Input ID number

This unsigned byte is the number assigned to this tray by the printer. It is the same number returned in the RDC-RIC response.

#### 5.3.4.3 Minimum size of medium across feed direction

This unsigned word specifies the minimum size of the print medium supported by this interpreter measured orthogonal to (across) the feed direction of the medium for a particular input feature. Units for this value are specified by the printer in Horizontal Logical Units. This unit value is returned in the RDC-RS response. A Horizontal Logical Unit is the unit of measure used to define the Horizontal Print Resolution also returned in the RDC-RS response. (For example, a Horizontal Logical Unit could be dots or pels for a laser or ink jet printer and might be expressed as characters for a fully formed character printer such as a daisy wheel printer.) If the size is unknown, the printer returns a hex value of FFFF.

#### 5.3.4.4 Minimum size of medium in feed direction

This unsigned word specifies the minimum size of the print medium supported by this interpreter measured in the feed or motion direction of the medium for a particular input feature. Units for this value are specified by the printer in Vertical Logical Units. This unit value is returned in the RDC-RS response. A Vertical Logical Unit is the unit of measure used to define the Vertical Print Resolution also returned in the RDC-RS response. (For example, a Vertical Logical Unit could be dots or pels for a laser or ink jet printer and might be expressed as lines for a fully formed character printer such as a daisy wheel printer.) If the size is unknown, the printer returns a hex value of FFFF.

#### 5.3.4.5 Maximum size of medium across feed direction

This unsigned word specifies the maximum size of the print medium supported by this interpreter measured orthogonal to (across) the feed direction of the medium for a particular input feature. Units for this value are specified by the printer in Horizontal Logical Units. If the size is unknown, the printer returns a hex value of FFFF.

#### 5.3.4.6 Maximum size of medium in feed direction

This unsigned word specifies the maximum size of the print medium supported by this interpreter measured in the feed or motion direction of the medium for a particular input feature. Units for this value are specified by the printer in Vertical Logical Units. If the size is unknown, the printer returns a hex value of FFFF.

#### 5.3.4.7 Input top margin

This unsigned word defines the top unprintable area supported by this interpreter for this input. The top margin is defined as that area above the first printable line written in the marking direction. The units for this value are specified by the printer in Vertical Logical Units.

A hex value of FFFF indicates that the printer does not have or cannot determine this value.

#### 5.3.4.8 Input extent of printable area in the feed direction

This unsigned word defines the printable area in the feed direction supported by this interpreter for this input. This is defined as that area between and including the first and last printable lines in the feed direction. The units for this value are specified by the printer in Vertical Logical Units.

A hex value of FFFF indicates that the printer does not have or cannot determine this value.

#### 5.3.4.9 Input left margin

This unsigned word defines the left unprintable area supported by this interpreter for this input. The left margin is defined as that area to the left of the first printable character in a line written in the marking direction. The units for this value are specified by the printer in Horizontal Logical Units.

A hex value of FFFF indicates that the printer does not have or cannot determine this value.

#### 5.3.4.10 Input extent of the printable area across feed direction

This unsigned word defines the printable area across the feed direction supported by this interpreter for this input. This is defined as that area between and including the first and last printable characters in a line written in the marking direction. The units for this value are specified by the printer in Horizontal Logical Units.

A hex value of FFFF indicates that the printer does not have or cannot determine this value.

### 5.3.5 Output characteristics

The RIC Output Characteristics subcommand (RIC-OC) is used by the host to request output characteristics for a specific interpreter for this printer. An additional byte in the data field is used by the host to limit the data returned to a specific output device. Table 41 shows the format of the command sent from the host to the printer.

**Table 41— RIC-OC command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Interpreter Characteristics | Packet header |
| 1 | 03 | Subcommand: Output Characteristics | Data field |
| 1 | ubyte | Logical Unit Number (Interpreter) | |
| 1 | ubyte | Output ID<br>00 = Return Data for All Output IDs<br>xx = Return Data for A Specific Output ID | |

This host to printer message is meaningful only if the reply required flag (bit 4 of the flag byte) is set in the host to printer message. With the reply required flag set, the printer will respond to the Output Characteristics subcommand

with a message like the one depicted in Table 42 provided that the error flag (bit 7 of the flag byte) is not set in the printer message.

**Table 42— RIC-OC response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Interpreter Characteristics | Packet header |
| 1 | 03 | Subcommand: Output Characteristics | Data field |
| 1 | ubyte | Logical Unit Number (Interpreter) | |
| 1 | ubyte | Number of Output IDs for which Data is Returned | |
| 1 | ubyte | #1 Output ID number | Output #1 |
| 1 | ubyte | #1 Output Number of Positions (e.g., collator)<br>FF Unknown | |
| ... | ... | ... | Output n |

If the error flag is set, a data error indicates that the data field did not contain a valid logical unit number (interpreter) or a valid output ID for the logical unit number selected. No output characteristic data shall be returned.

#### 5.3.5.1 Number of output IDs for which data is returned

This unsigned byte indicates the number of output IDs for which data is returned in this response. The rest of the data in this message will describe the bin(s) and the extent supported.

A value of 0 indicates that the printer cannot provide this information for this interpreter.

#### 5.3.5.2 ID Number

The ID Number is the number assigned to this bin by the printer. It is the same number returned in the RDC-ROC response.

#### 5.3.5.3 Number of positions

This unsigned byte indicates the number of positions for this bin. For a standard output this value would be 1. However, if the printer had a collator attached to it, the value in this field would be equal to the number of positions that this interpreter supported for this collator.

The hex value FF indicates that the printer is not able to supply this information.

## 5.4 Job Control (JC)

### 5.4.1 Controlling the process of data

Job Control commands provide a mechanism for controlling the printer’s actions in processing data. The base logical grouping of data is the job. A job is a collection of data for a specific LU. A session allows the logical grouping of one or more jobs into a sequential, referenceable collection. Empty jobs and empty sessions are valid.

Jobs are delineated in the data stream to the printer by the Start Job and End Job commands. The Start Job command prompts the printer to initialize the graphic engine and reset accounting statistics. The End Job command instructs the printer to log the accounting statistics, and release any nonpersistent job-related resources.

Sessions are delineated in the data stream by the Start Session and End Session commands. The Start Session command is an indicator to the printer that a collection of jobs follows. The printer shall use this command to lock port rotation in a multiple link and/or multiple protocol environment. The End Session command marks the end of the collected data, prints and ejects any remaining sheets, and allows link rotation.

For each link (physical or logical) between a host and the printer the states for the transmission of data are identified in Table 43. IEEE 1284.1 only defines the in-session and in-job states. The definition of the other two states are outside the scope of this standard but are listed for context.

**Table 43— Job control states**

| State | Description |
|---|---|
| idle | This state is not defined by IEEE Std 1284.1-1997. It is indicative of the link state where no transmission of data in either direction is possible without first making a physical connection or, if multiple logical connections are supported on the physical link, a logical connection between a host and the printer. |
| in-connection | This state is not defined by IEEE Std 1284.1-1997. It is indicative of the state of the physical or logical connection between a host and the printer. Such connection may be temporary, persistent, or permanent. When in this state data may be transferred between a host and the printer. |
| in-session | This state can be requested or implied. It is requested by the explicit use of the Start Session command. It is implied by the reception of either a IEEE 1284.1 command, a Start Job command, or data while either in the idle or in-connection state.<br>Likewise, exiting from this state can be requested or implied. It is requested by the explicit use of the End Session command. It is implied by an indication that the connection has been closed while either in the in-session or in-job states. |
| in-job | This state can be requested or implied. It is requested by the explicit use of the Start Job command. It is implied by the reception of data at while not in the in-job state.<br>Likewise, exiting from this state can be requested or implied. It is requested by the explicit use of the End Job command. It is implied by an End Session command or an indication that the connection has been closed. |

The relationship of these states to one another are shown in Figure 5.

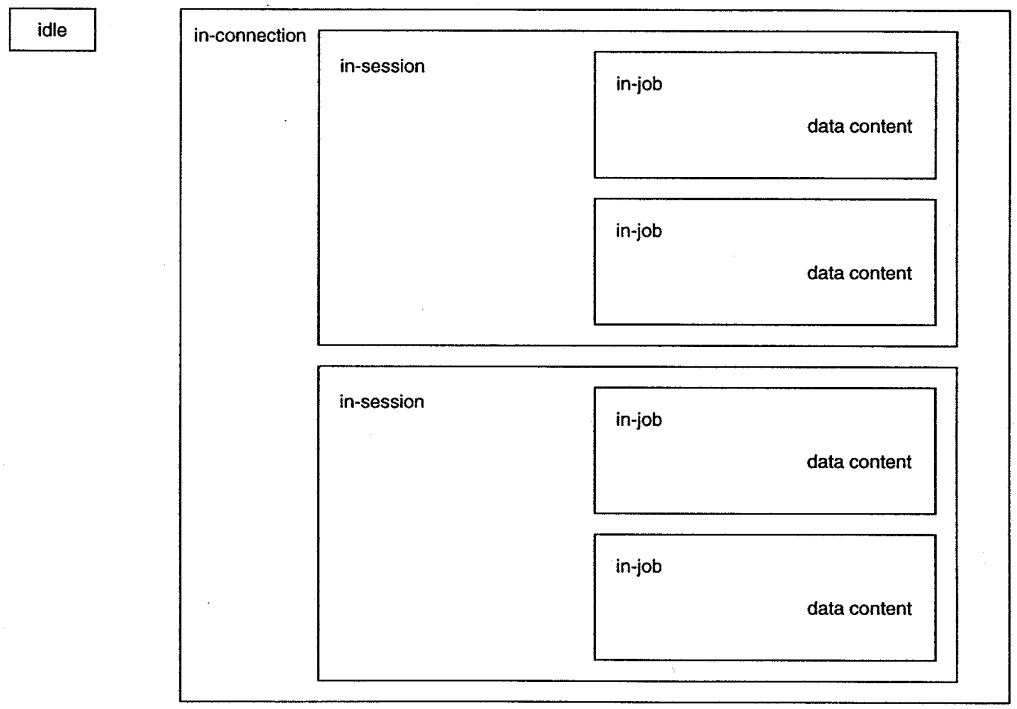


**Figure 5— Data communication state relationship**

The actions that occur as the printer enters or exists a state are defined in Table 44.

**Table 44— Actions taken on data communication state transition**

| Action | | Description |
|---|---|---|
| **soc** | start of connection | Action specific to a given transport and/or implementation |
| **sos** | start of session | Disable link or protocol rotation |
| **soj** | start of job<br>(acts as a delimiter on a per LU# basis) | Initialize LU#/graphic engine Reset accounting statistics |
| **eoj** | end of job<br>(acts as a delimiter on a per LU# basis) | Print and eject last sheet, if requested Log accounting statistics Release nonpersistent job-related resources |
| **eos** | end of session | Print and eject any remaining sheets<br>Enable link or protocol rotation |
| **eoc** | end of connection | Action specific to a given transport and/or implementation |
| **proc..** | process | Normal processing of data and/or commands |

There are eight events that shall cause the movement from one state to another. These events are listed in Table 45. The first column, current state, lists the four printer states. The top rows list the eight different events that shall precipitate a state change. The remaining cells in the table list the next state (*italic* text) and the sequence of actions (normal text) that are taken in moving from the current state to the next state.

**Table 45— Data communication state transitions**

| | **event** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **current state** | **connection** | | **session** | | **job** | | **1284.1 command** | **data** |
| | **start** | **end** | **start** | **end** | **start** | **end** | | |
| idle | *in-c*<br>soc | *idle*<br>soc<br>eoc | *in-s*<br>soc<br>sos | *idle*<br>soc<br>sos<br>eos<br>eoc | *in-j*<br>soc<br>sos<br>soj | *idle*<br>soc<br>sos<br>soj<br>eoj<br>eos<br>eoc | *idle*<br>soc<br>proc..<br>eoc | *in-j*<br>soc<br>sos<br>soj<br>proc.. |
| in-connection (in-c) | *in-c*<br>eoc<br>soc | *idle*<br>eoc | *in-s*<br>sos | *in-c*<br>sos<br>eos | *in-j*<br>sos<br>soj | *in-c*<br>sos<br>soj<br>eoj<br>eos | *in-c*<br>proc.. | *in-j*<br>sos<br>soj<br>proc.. |
| in-session (in-s) | *in-s*<br>eos<br>eoc<br>soc | *idle*<br>eos<br>eoc | *in-c*<br>eos<br>sos | *idle*<br>eos | *in-j*<br>soj | *in-s*<br>soj<br>eoj | *in-s*<br>proc.. | *in-j*<br>soj<br>proc.. |
| in-job (in-j) | *in-c*<br>eoj<br>eos<br>eoc<br>soc | *idle*<br>eoj<br>eos<br>eoc | *in-c*<br>eoj<br>eos<br>sos | *idle*<br>eoj<br>eos | *in-j*<br>eoj<br>soj | *in-s*<br>eoj | *in-j*<br>proc.. | *in-j*<br>proc.. |

### 5.4.2 Job control usage

IEEE 1284.1 compliant applications shall use Start Session, End Session, Start Job, and End Job commands explicitly. Implicit Start Session, End Session, Start Job, and End Job actions may occur due to error conditions or legacy applications.

### 5.4.3 JC subcommands

The JC command provides a mechanism for the host application to delineate jobs and sessions within the printer allowing the host to control and obtain information referencing a particular job or a group of jobs (session). The currently defined JC commands are enumerated in Table 46.

**Table 46— JC categories**

| Subcommand Category | Description |
|---|---|
| Start Job | This command is sent to a logical unit within the printer prior to sending data to that logical unit. It shall return to the host the Job ID that the host shall use for future reference to that job in the printer. |
| End Job | This command is sent by the host to a logical unit within the printer after all of the data has been sent. This marks the end of the physical entity referred to as a job. |
| Query Job(s) Completed | This command can be used by the host to get the printer to return print statistics for jobs completed by the printer. These statistics include the number of pages printed and the length of time the job required. |
| Query Job(s) Queued or Active | This command is sent by the host to obtain information about printer jobs not yet completed. |
| Start Session | This command is sent to the printer to indicate the beginning of a session (a group of related jobs) that are to follow. The printer shall return to the host a session ID that the host shall use for future reference to the session in the printer. |
| End Session | This command is sent by the host to the printer after all jobs have been sent for the session. |
| Query Session(s) Queued or Active | This command is sent by the host to obtain information about printer sessions not yet completed. |
| Change Session Priority | This command is sent by the host to change the priority of a session in the printer. |
| Delete Session | This command is sent by the host to delete a session in the printer. |
| Resume Suspended Job | This command is sent by the host to resume a suspended job. |

The host controls jobs in the printer by sending a JC command with the desired subcommand. The subcommand is the first byte in the data field. Other bytes in the data field may be required parameters for the different JC subcommands. Table 47 shows values for the currently defined JC subcommands.

**Table 47— JC subcommands**

| Value (hex) | Description |
|---|---|
| 00 | Start Job |
| 01 | End Job |
| 02 | Query Job(s) Completed |
| 03 | Query Job(s) Queued or Active |
| 04 to 07 | Reserved |
| 08 | Start Session |
| 09 | End Session |
| 0A | Query Session(s) Queued or Active |
| 0B | Change Session Priority |
| 0C | Delete Session |
| 0D to 0F | Reserved |
| 10 | Resume Suspended Job |
| 11 to FF | Reserved |

### 5.4.4 Start Job

The JC Start Job subcommand (JC-SJ) is used by a host to indicate the start of a job for a given logical unit (interpreter). Using the JC-SJ subcommand the host registers to receive Job Control Alerts for the job. The printer response to the JC-SJ subcommand shall be a data packet that contains the job identification number that the printer has assigned. The job identification number returned is used by the host to reference this job in future data exchanges. These exchanges shall include marking the end of the job and requesting status or statistics. Table 48 shows the format of the command sent from the host to the printer to signal the start of a job.

**Table 48— JC-SJ command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Job Control | Packet header |
| 1 | 00 | Subcommand: Start Job | Data field |
| 1 | ubyte | Logical Unit Number (Interpreter) | |
| 1 | bbyte | Job Alert Points<br>Bit 0 reserved<br>Bit 1 Logical Unit Start(zero=Off, one=On)<br>Bit 2-5 reserved<br>Bit 6 Sheet Complete(zero=Off, one=On)<br>Bit 7 Job Complete(zero=Off, one=On) | |
| 1 | bbyte | Processing Flags<br>Bit 0 reserved<br>Bit 1 suspended at LU start(zero=No, one=Yes)<br>Bit 2 reserved<br>Bit 3 suspend at start of printing (zero=No, one=Yes)<br>Bit 4 reserved<br>Bit 5 suspend at start of finishing (zero=No, one=Yes)<br>Bit 6 separate data channel(zero=No, one=Yes)<br>Bit 7 retain incomplete sheet(zero=Off, one=On) | |
| 1 | ubyte | Length of Host String | |
| 1–63 | ASCII String | Host String | |
| 1 | ubyte | Length of User String | |
| 1–63 | ASCII String | User String | |
| 1 | ubyte | Length of Job Name String | |
| 1–63 | ASCII String | Job Name String | |
| 1 | ubyte | Length of Additional Information String | |
| 1–63 | ASCII String | Additional Information String | |

If the host to printer message reply required flag (bit 4 of the flag byte) is set, and in the corresponding printer to host reply message the error flag (bit 7 of the flag byte) is not set, the printer response to the JC-SJ subcommand shall be a message like the one depicted in Table 49. If the host to printer message reply required flag is not set, no JC-SJ response will be issued.

**Table 49— JC-SJ response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Job Control | Packet header |
| 1 | 00 | Subcommand: Start Job | Data field |
| 1 | ubyte | Logical Unit Number (Interpreter) | |
| 2 | uword | Printer Assigned Job ID<br>NOTE—The printer shall never assign a Job ID of 0000 as this value is used in subsequent commands to request data on all jobs as opposed to a specific Job ID for any interpreter. | |
| 2 | uword | Printer Assigned Data Channel—a uword identifying the printer channel number to used by the host to transfer the data; the channel number represents a transport-specific channel number (port). If the channel number returned by the printer is zero, then it implies that the printer cannot perform data transfer via a separate channel, thereby requiring the host to use the control channel for data transfer. | |

If the error flag is set, a data error (bits 2 and 3 of the flag byte) indicates that an undefined logical unit number was selected. No Job ID shall be returned.

If the host, for any reason, does not receive the message from the printer (error checking error, etc.), the same Job ID can be returned by the printer if it is requested again prior to any data being sent to the same logical unit number. If the

printer does not return the same Job ID, and no data has been sent to the same logical unit number, then the printer shall discard the internal Job ID for which no data has been sent without placing any statistics in the job completed queue.

If a JC-SJ subcommand for a logical unit (interpreter) is sent by the host to the printer after the host has sent data to the same logical unit and prior to the host sending an JC End Job subcommand to that logical unit, the printer shall internally execute an implied JC End Job subcommand for this logical unit before processing the JC-SJ subcommand.

#### 5.4.4.1 Job alert points

##### 5.4.4.1.1 logical unit start

When set, indicates that the printer shall send a JCA-LUS alert when the LU begins processing this job.

##### 5.4.4.1.2 sheet complete

When set, indicates that the printer shall send a JCA-SC alert any time a sheet is deposited in the output bin as a result of this job.

##### 5.4.4.1.3 job complete

When set, indicates that the printer shall send a JCA-EOJ alert when the last sheet of this job is deposited in the output bin.

#### 5.4.4.2 Processing flags

##### 5.4.4.2.1 suspend at LU start

When set, the LU shall suspend processing data from the input buffer associated with this LU when the data associated with this job reached the LU. The host shall then send a JCA-SP alert to inform the host. When notified, the host has the ability to send additional commands to the LU (outside the normal data stream) by sending data to LU+128. The printer shall allow this data to bypass the normal input buffers and route it directly to the LU. Normal processing shall be continued when the host issues the JC-RSP command.

##### 5.4.4.2.2 suspend at start of printing

When set, the LU shall suspend processing data for this job when the print data associated with this job reaches the print engine. The host shall then send a JCA-SP alert to inform the host. Normal processing shall be continued when the host issues the JC-RSP command.

##### 5.4.4.2.3 suspend at start of finishing

When set, the LU shall suspend processing data for this job when the print media associated with this job reaches the finisher. The host shall then send a JCA-SP alert to inform the host. Normal processing shall be continued when the host issues the JC-RSP command.

##### 5.4.4.2.4 separate data channel

Processing flags bit 6 provides a means for the job initiator to request delivery of the print job on a different channel (or data channel) than would be used by default. When the job initiator requests a separate channel (i.e., bit 6 equal 1) on the JC-SJ command, the JC-SJ response will include a unsigned word value that indicates the channel to be used. This channel is transport dependent. In the case of TCP/IP it would represent a port number, for IPX—a socket. A zero in the response indicates the printer does not support separate data channels on the current transport. It is the responsibility of the initiator to open the channel to the printer to send the print job data.

#### 5.4.4.2.5 retain incomplete sheet

If this flag is set and the LU supports it, the printer shall not image partial pages when the end of job command is received from the host. It is assumed that the host shall complete the image in a subsequent job within the current session.

### 5.4.5 End Job

The JC End Job subcommand (JC-EJ) is used by a host to indicate the end of a job started with the JC-SJ subcommand for a given logical unit (interpreter). The data sent from the host to the printer in the JC-EJ subcommand includes the logical unit number and the Job ID returned by the printer from a prior JC-SJ subcommand response. Table 50 shows the format of the command sent from the host to the printer.

**Table 50— JC-EJ command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Job Control | Packet header |
| 1 | 01 | Subcommand: End Job | Data field |
| 1 | ubyte | Logical Unit Number (Interpreter) | |
| 2 | uword | Printer Assigned Job ID | |

It is recommended that the host set the reply required flag (bit 4 of the flag byte) in this message to the printer to receive an acknowledgment from the printer that it has received and accepted the JC-EJ subcommand. With the reply required flag set, the printer shall respond to the JC-EJ subcommand with a message like the one depicted in Table 51 if the error flag (bit 7 of the flag byte) is not set in the response the host receives from the printer.

**Table 51— JC-EJ response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Job Control | Packet header |
| 1 | 01 | Subcommand: End Job | Data field |
| 1 | ubyte | Logical Unit Number (Interpreter) | |
| 2 | uword | Printer Assigned Job ID as specified in the JC-EJ subcommand that prompted this reply. | |

The error flag shall be set, and a data error (indicated by bits 2 and 3 of the flag byte in the printer to host message) shall be posted to indicate that an undefined logical unit number/Job ID pair was selected. Both the logical unit number and a valid Job ID for that logical unit number shall be sent in the host to printer message for a data error not to be posted.

If the host, for any reason, does not receive the message from the printer (error checking error, etc.), the JC-EJ subcommand for the same logical unit number/Job ID can be sent again regardless of whether or not the printer has internally marked the end of the job.

If the host sends a JC-EJ subcommand to a logical unit number following a JC-SJ subcommand without sending any data to that logical unit between the JC-SJ and JC-EJ subcommands, the printer shall place an entry in the job completion queue with zero valued job statistics, for those features (length and/or processing time) supported, with the unsupported features fields set to a binary value of all ones (maximum value).

The printer shall ignore multiple JC-EJ subcommands and shall not return an error for a logical unit/Job ID pair if that combination is a valid job in the queue, in process, or completed and still remaining in the printer statistics FIFO queue.

### 5.4.6 Query Job(s) Completed

The JC Query Job(s) Completed subcommand (JC-QJC) is used by the host to request statistics for jobs that have completed the printing process (e.g., stacked in the output tray). The time at which a job is considered completed is defined to be when the last sheet has been stacked in the output feature to the best of the printer’s knowledge. The host can request this data at any time. It shall include data on the most recently completed jobs. The data, number of job statistics, shall be limited only by the size of the buffer in the printer that holds this information.

 

Not knowing which job completed or on which logical unit a job has completed, the printer shall maintain a FIFO queue of at least 16 entries to hold job completion statistics. (The actual size of the queue is returned in response to the RDC-RS subcommand.) This provides the host the capability of retrieving the statistics for a particular job any time prior to the completion of the sixteenth job that follows. This shall give the host ample time to obtain the information for a particular logical unit number/Job ID pair, even if the host has to request it from the printer several times due to error checking protocols, etc.

Table 52 shows the format of the command sent from the host to the printer to request statistics for completed jobs.

**Table 52— JC-QJC command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Job Control | Packet header |
| 1 | 02 | Subcommand: Query Job(s) Completed | Data field |
| 1 | ubyte | Logical Unit Number (Interpreter)<br>xx Return completed job statistics for this logical unit number.<br>FF Return completed job statistics for all logical unit numbers. | |
| 2 | uword | Requested Job ID<br>0000 Return completed job statistics for all jobs for the above specified logical unit number(s)<br>xxxx Return completed job statistics for a specific job for the above specified logical unit number. | |
| 2 | uword | Maximum Number of Job Statistics to Return—Limit the printer to host message to the this number of jobs that meet the above logical unit number and Job ID criteria. | |

If the host to printer message reply required flag (bit 4 of the flag byte) is set, and in the corresponding printer to host reply message the error flag (bit 7 of the flag byte) is not set, the printer response to the JC-QJC subcommand shall be a message like the one depicted in Table 53. If the host to printer message reply required flag is not set, no JC-QJC response will be issued.

**Table 53— JC-QJC response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Job Control | Packet header |
| 1 | 02 | Subcommand: Query Job(s) Completed | Data field |
| 2 | uword | Number of job statistics records returned. Each record consists of logical unit number, Job ID, processing time, sheets used, impressions, and counter units for each input source supported by this interpreter. Job data shall be returned in reverse order of job completion within the printer, i.e., the last job completed shall be the first one reported. | |
| 1 | ubyte (i) | Number of input sources | Job Statistic #1 |
| 1 | ubyte | Completed job #1 logical unit number | |
| 2 | uword | Completed job #1 Job ID | |
| 2 | uword | Completed job #1 processing time (in seconds) | |
| 4 | udword | Completed job #1 sheets used from input source 1 | |
| 4 | udword | Completed job #1 impressions from input source 1 | |
| 4 | udword | Completed job #1 counter units from input source 1 | |
| ... | ... | ... | |
| 4 | udword | Completed job #1 sheets used from input source i | |
| 4 | udword | Completed job #1 impressions from input source i | |
| 4 | udword | Completed job #1 counter units from input source i | |
| ... | ... | ... | Job Statistic n |

If the host requests statistics for more jobs than the printer has records to match, an error shall not occur. The printer shall return all job statistics that it has in the FIFO queue. The host shall know how many job statistics the printer is sending via the value in the printer response that indicates the number of job statistics in the printer to host message.

If the host requests statistics for fewer jobs than the printer has records to match, the printer shall return exactly the number requested by the host. The statistics returned shall be in reverse chronological order (last job completed reported first).

If the printer has not completed any jobs prior to the time it receives a host request for job statistics or does not have any job statistics matching the selection criteria of the subcommand, the data field of the printer response shall contain only one word, having a value of zero, indicating that no job statistics are available.

If the error flag is set, a data error indicates that the data field did not contain a value in the field that indicates the number of job statistics that shall be returned. No data shall be returned.

#### 5.4.6.1 Counter units

Counter units are defined in the RDC-RS subcommand response. A value returned with all bits set high indicates that the printer is incapable of reporting this information for this logical unit number.

If the counter units are defined as pages, then this refers to the number of logical pages printed. In “two-up” printing there would be two logical pages per impression. In “four-up” printing there would be four logical pages per impression.

For counter units other than pages, the fields for sheets and impressions are meaningless.

#### 5.4.6.2 Sheets

A sheet is defined as a physical piece of print media, such as a sheet of paper. A value returned with all bits set high indicates that the printer is incapable of reporting this information for this logical unit number.

#### 5.4.6.3 Impressions

An impression is defined as the number of sides printed on a sheet. Simplex print would have one impression per sheet, whereas duplex print would have two impressions per sheet. A value returned with all bits set high indicates that the printer is incapable of reporting this information for this logical unit number.

#### 5.4.6.4 Processing Time

If a printer provides support for processing time, the method for calculating time is dependent on the implementation by the printer manufacturer. It is recommended that the processing time be the actual time that the printer spends processing the job. This would not include time spent waiting for an operator to service “intervention required” conditions, etc. A value returned with all bits set high indicates that the printer is incapable of reporting this information for this logical unit number.

### 5.4.7 Query Job(s) Queued or Active

The JC Query Job(s) Queued or Active subcommand (JC-QQA) is used by the host to obtain information on jobs whose processing has not started or not completed. The data returned by the printer shall indicate whether or not the job is being held, whether or not it is waiting for processing to begin, or whether or not it is processing. Table 54 shows the format of the command sent from the host to the printer requesting information regarding a job not yet completed processing by the printer.

**Table 54— JC-QQA command**

| Bytes | Value | Description | Notes |
|---|---|---|---|
| 5 | | Job Control | Packet header |
| 1 | 03 | Subcommand: Query Job(s) Queued or Active | Data field |
| 1 | ubyte | Logical Unit Number (Interpreter)<br>xx Return queued or active job data for this logical unit number.<br>FF Return queued or active job data for all logical units. | |
| 2 | uword | Requested Job ID<br>0000 Return queued or active job data for all jobs for the above specified logical unit number(s).<br>xxxx Return queued or active job data for a specific job for the above specified logical unit number. | |

The host can specify to return information for all queued or active jobs on all logical units, for all queued or active jobs on a specific logical unit number, or for a specific job on a specific logical unit number.

If the host to printer message reply required flag (bit 4 of the flag byte) is set, and in the corresponding printer to host reply message the error flag (bit 7 of the flag byte) is not set, the printer response to the JC-QQA subcommand shall be a message like the one depicted in Table 55. If the host to printer message reply required flag is not set, no JC-QQA response will be issued.

**Table 55— JC-QQA response (continued)**

| Bytes | Value | Description | Notes |
|---|---|---|---|
| 5 | | Job Control | Packet header |
| 1 | 03 | Subcommand: Query Job(s) Queued or Active | Data field |
| 1 | ubyte (n) | Number of job detail records returned. Each record shall consists of 10 bytes: logical unit number, Job ID, status, position & size. Job detail records can be returned in any order. | |
| 1 | ubyte | Queued or Active job #1 logical unit number | Record #1 |
| 2 | uword | Queued or Active job #1 Job ID | |
| 1 | ubyte | Queued or Active job #1 status<br>00 Processing<br>01 Waiting<br>02 Deprecated<br>03 Suspended<br>04 Waiting—Incomplete Sheet Retained<br>04–7F Reserved<br>80–FF Printer Specific | |
| 1 | bbyte | Queued or Active job #1 Job Processing Position<br>Bit 0 Reserved<br>Bit 1 Logical Unit Processing<br>Bit 2 Reserved<br>Bit 3 Printing Processing<br>Bit 4 Reserved<br>Bit 5 Finishing Processing<br>Bit 6–7 Reserved | |
| 1 | bbyte | Queued or Active job #1 Job Suspension Position<br>Bit 0 Reserved<br>Bit 1 Suspended at Logical Unit Start<br>Bit 2 Reserved<br>Bit 3 Suspended at Start of Printing<br>Bit 4 Reserved<br>Bit 5 Suspended at Start of Finishing<br>Bit 6–7 Reserved | |
| 4 | Unsigned dword | Queued or Active job #1 size (in bytes)<br>FFFFFFFF Not Supported | |
| ... | ... | ... | Record n |
| 1 | ubyte | Queued or Active job #n logical unit number | |
| 2 | uword | Queued or Active job #n Job ID | |
| 1 | ubyte | Queued or Active job #n status | |
| 1 | bbyte | Queued or Active job #n Job Processing Position | |
| 1 | bbyte | Queued or Active job #n Job Suspension Position | |
| 4 | Unsigned dword | Queued or Active job #n size (in bytes)<br>FFFFFFFF Not Supported | |

The error flag shall be set, and a data error (indicated by bits 2 and 3 of the flag byte in the printer to host message) shall be posted indicating that an undefined logical unit number/Job ID pair was selected. Both the logical unit number and a valid Job ID for that logical unit number shall be sent in the host to printer message for a data error not to be posted.

### 5.4.8 Start Session

The JC Start Session subcommand (JC-SS) is used by a host to indicate the beginning of a collection of jobs. The printer shall use this command to lock port rotation in a multiple link and/or multiple protocol environment. The printer response to the JC-SS subcommand shall be a data packet that contains the session identification number that the printer has assigned. The session identification number returned is used by the host to reference this session in future data exchanges. Table 56 shows the format of the command sent from the host to the printer to signal the start of a job.

**Table 56— JC-SS command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Job Control | Packet header |
| 1 | 08 | Subcommand: Start Session | Data field |
| 1 | ubyte | Requested Priority<br>00 = Invalid (cannot arbitrarily hold session when submitting)<br>01 = Lowest Priority<br>FF = Highest Priority | |

If the host to printer message reply required flag (bit 4 of the flag byte) is set, and in the corresponding printer to host reply message the error flag (bit 7 of the flag byte) is not set, the printer response to the JC-SS subcommand shall be a message like the one depicted in Table 57. If the host to printer message reply required flag is not set, no JC-SS response will be issued.

**Table 57— JC-SS response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Job Control | Packet header |
| 1 | 08 | Subcommand: Start Session | Data field |
| 2 | uword | Printer Assigned Session ID<br>NOTE—The printer shall never assign a Session ID of 0000, as this value is used in subsequent commands to request data on all sessions as opposed to a specific session. | |
| 1 | ubyte | Printer Assigned Priority (may not be the same as requested) | |

### 5.4.9 End Session

The JC End Session subcommand (JC-ES) is used by a host to indicate the end of a session started with the JC-SS subcommand. The data sent from the host to the printer in the JC-ES subcommand includes the session ID returned by the printer from the prior JC-SS subcommand response. Table 58 shows the format of the command sent from the host to the printer.

**Table 58— JC-ES command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Job Control | Packet header |
| 1 | 09 | Subcommand: End Session | Data field |
| 2 | uword | Printer Assigned Session ID | |

It is recommended that the host set the reply required flag (bit 4 of the flag byte) in this message to the printer to receive an acknowledgment from the printer that it has received and accepted the JC-ES subcommand. With the reply required flag set, the printer shall respond to the JC-ES subcommand with a message like the one depicted in Table 59 provided that the error flag (bit 7 of the flag byte) is not set in the printer message.

**Table 59— JC-ES response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Job Control | Packet header |
| 1 | 09 | Subcommand: End Session | Data field |
| 2 | uword | Printer Assigned Session ID as specified in the JC-ES subcommand that prompted this reply. | |
| 1 | ubyte | Number of Jobs in this session | |
| 1 | ubyte | Logical Unit number of 1st Job | Job #1 |
| 2 | uword | Printer Assigned Job ID of 1st Job | |
| ... | ... | ... | Job n |

The error flag shall be set, and a data error (indicated by bits 2 and 3 of the flag byte in the printer to host message) shall be posted indicating that an undefined session ID was selected. A valid session ID shall be sent in the host to printer message for a data error not to be posted.

If the host, for any reason, does not receive the message from the printer (error checking error, etc.), the JC-ES subcommand for the same session ID can be sent again regardless of whether or not the printer has internally marked the end of the session.

The printer shall ignore multiple JC-ES subcommands and shall not return an error for a session ID if the session ID is a valid session in the queue, in process, or completed.

### 5.4.10 Query Session(s) Queued or Active

The JC Query Session(s) Queued or Active subcommand (JC-QSQA) is used by the host to obtain information on sessions whose processing has not started or not completed. The data returned by the printer shall indicate whether or not the session is being held, whether or not it is waiting for processing to begin, or whether or not it is processing. Table 60 shows the format of the command sent from the host to the printer requesting information regarding a job not yet completed processing by the printer.

**Table 60— JC-QSQA command**

| Bytes | Value | Description | Notes |
|---|---|---|---|
| 5 | | Job Control | Packet header |
| 1 | 0A | Subcommand: Query Session(s) Queued or Active | Data field |
| 2 | uword | Requested Session ID<br>0000 Return queued or active session data for all sessions<br>xxxx Return queued or active session data for a specific session | |

The host can specify to return information for all queued or active sessions or for a specific session.

If the host to printer message reply required flag (bit 4 of the flag byte) is set, and in the corresponding printer to host reply message the error flag (bit 7 of the flag byte) is not set, the printer response to the JC-QSQA subcommand shall be a message like the one depicted in Table 61. If the host to printer message reply required flag is not set, no JC-QSQA response will be issued.

**Table 61— JC-QSQA response**

| Bytes | Value | Description | Notes |
|---|---|---|---|
| 5 | | Job Control | Packet header |
| 1 | 0A | Subcommand: Query Session(s) Queued or Active | Data field |
| 1 | ubyte (n) | Number of session records returned | |
| 1 | ubyte | Status of 1st session<br>00 Processing<br>01 Waiting<br>02 Deprecated<br>03 Suspended (a suspended job in a session causes a suspended session)<br>04–7 FReserved<br>80–F FPrinter Specific | Session #1 |
| 1 | ubyte | Priority of 1st Session | |
| 1 | ubyte (i) | Number of jobs in 1st session | |
| 2 | uword | 1st Job ID in 1st Session | |
| ... | ... | ... | |
| 2 | uword | Last Job ID in 1st Session | |
| ... | ... | ... | Session n |

The error flag shall be set, and a data error (indicated by bits 2 and 3 of the flag byte in the printer to host message) shall be posted indicating that an undefined session was selected.

### 5.4.11 Change Session Priority Subcommand

The JC Change Session Priority subcommand (JC-CSP) is used to change the priority of a session that is waiting in the printer's queue. The data sent from the host to the printer in the JC-CSP command consists of the session number returned by the printer from a prior JC-SS response. Table 62 shows the format of the command sent from the host to the printer.

If the requested session has already become active, this command shall not affect its priority.

**Table 62— JC-CSP command**

| Bytes | Value | Description | Notes |
|---|---|---|---|
| 5 | | Job Control | Packet header |
| 1 | 0B | Subcommand: Change Session Priority | Data field |
| 2 | uword | Printer assigned session ID for 1st session<br>0000 All sessions | |
| 1 | ubyte | New priority for 1st session<br>00 hold session<br>01 lowest priority<br>...<br>FF highest priority | |

It is recommended that the host set the reply required flag (bit 4 of the flag byte) in this message to the printer to receive an acknowledgment from the printer that it has received and accepted the JC-CSP subcommand. With the reply required flag set, the printer shall respond to the JC-CSP command with a message like the one depicted in Table 63 provided that the error flag (bit 7 of the flag byte) is not set in the printer message.

**Table 63— JC-CSP response**

| Bytes | Value | Description | Notes |
|---|---|---|---|
| 5 | | Job Control | Packet header |
| 1 | 0B | Subcommand: Change Session Priority | Data field |
| 1 | ubyte | Logical Unit Number (Interpreter) | |
| 2 | uword (n) | Number of session priorities changed | |
| 2 | uword | #1 Printer Assigned Session ID changed | Session #1 |
| 1 | ubyte | #1 session's new priority | |
| ... | ... | ... | Session n |

If the error flag is set, a data error indicates no valid session ID was provided. For a session's priority to be changed, it shall either be waiting or held in the printer's queue. A request to change the priority of a session that is not either waiting or held shall result in a data error.

If a request from the host is received to change the priority of all sessions and no sessions exist in the printer's queue when the message is received, a data error shall not be posted. The resulting printer to host message's data field shall contain a single zero-valued byte indicating that no session priorities were changed.

### 5.4.12 Delete Session Subcommand

The JC Delete Session subcommand (JC-DS) is used by a host to remove a session from the printer. A session may be deleted if it is in the printer's queue (waiting or held) or if it is processing. An immediate end of session shall be created within the printer for this session when this command is sent from the host to the printer.

If a member job is processing when its session is deleted, the printer shall place the job statistics at the time of deletion in the FIFO queue for this job.

If the session was waiting or held when it was canceled, the printer shall place statistics for the member jobs in the FIFO queue with zero values for length and processing time. If the printer was incapable of reporting length and/or processing time, for consistency the value placed in the unsupported field(s) shall be all ones instead of zeros.

The data sent from the host to the printer for the JC-DS command consists of the session number returned by the printer in a prior JC-SS response. Table 64 shows the format of the command sent from the host to the printer to delete a session.

**Table 64— JC-DS command**

| Bytes | Value | Description | Notes |
|---|---|---|---|
| 5 | | Job Control | Packet header |
| 1 | 0C | Subcommand: Delete Session | Data field |
| 2 | uword | Printer Assigned Session ID to be canceled<br>0000 delete all sessions<br>xxxx session id to be deleted | |

It is recommended that the host set the reply required flag (bit 4 of the flag byte) in this message to the printer to receive an acknowledgment from the printer that it has received and accepted the JC-DS subcommand. With the reply required flag set, the printer shall respond to the JC-DS subcommand with a message like the one depicted in Table 65 provided that the error flag (bit 7 of the flag byte) is not set in the printer message.

**Table 65— JC-DS response**

| Bytes | Value | Description | Notes |
|---|---|---|---|
| 5 | | Job Control | Packet header |
| 1 | 0C | Subcommand: Delete Session | Data field |
| 2 | uword | Number of jobs canceled | |
| 2 | uword | #1 Printer Assigned Session ID deleted | Session #1 |
| ... | ... | ... | Session n |

If the error flag is set, a data error indicates that an undefined session ID was selected. For a session to be deleted, it shall be either in the printer's queue (held or waiting) or currently being processed by the printer. A request to delete a session that is not in one of these states shall cause the printer to return a data error.

If a request from the host is received to delete all sessions and no sessions are in the printer's queue when the message is received, a data error shall not be posted. The resulting printer to host message's data field shall contain a single, zero-valued byte indicating that there were no sessions deleted.

A data error shall be returned if the session has already been deleted.

### 5.4.13 Resume Suspended Job

The JC Resume Suspended Job (JC-RSJ) is used by the host to resume a job that has been suspended as a result of the host setting one or more of the processing flags in the JC-SJ command. Table 66 shows the format of the command sent from the host to the printer requesting information regarding a job not yet completed processing by the printer.

**Table 66— JC-RSJ command**

| Bytes | Value | Description | Notes |
|---|---|---|---|
| 5 | | Job Control | Packet header |
| 1 | 10 | Subcommand: Resume Suspended Job | Data field |
| 1 | ubyte | Logical Unit Number (Interpreter) | |
| 2 | uword | Suspended Job ID | |

## 5.5 Request Device Status (RDS)

### 5.5.1 Overview

The RDS command provides a mechanism for the host system to obtain status information regarding the operating condition of the printer. The host can obtain information concerning printer status in one of several categories. The printer shall always accept an RDS command without rejection due to lack of input buffer and processing space. This means that the printer shall reserve buffer space to process at least one RDS command.

The status categories for which the host can retrieve information are shown in Table 67.

**Table 67— RDS categories**

| Status category | Description |
|---|---|
| Input | Those conditions in the printer where problems with the input of printing media can cause the printer to stop. These conditions can be normally be remedied by a normal operator. An example of this would be when a particular input tray is empty. |
| Output | Those conditions in the printer where problems with the output of printing media can cause the printer to stop. These conditions can be normally be remedied by a normal operator. An example of this would be when a particular output tray is full. |
| Jam | Those conditions in the printer where normal conditions associated with the movement of printing media through the printer can cause it to stop, these conditions can be corrected by a normal operator. An example of this would be "Jam at Input Tray." |
| Supplies | Those conditions in the where normal conditions in the printer associated with supplies can cause it to stop, these conditions can be corrected by a normal operator. An example of this would be "Out of Ink." |
| Operator Intervention Required | Those conditions in the printer, other than input, output, jam or printing supplies, that can cause the printer to stop, these conditions can be corrected by a normal operator. An example of this would be "Cover Open." |
| Warnings | Those conditions that are not critical to the immediate operation of the printer: Examples are<br>— Paper Input Low<br>— Toner Low<br>— Output Bin Nearly Full |
| Device Service Required | Those conditions that are critical to printer operation that a normal operator cannot correct and requires technical service. Examples are<br>— Laser Polygon Scanner inoperative<br>— High Fuser Temperature |
| Configuration Change | Those conditions that are detected that change the current configuration of the printer. Examples are<br>— Memory Board Failure resulting in reduced memory<br>— Font Cartridge/Card has been removed |
| Printer Statistics | Those printer specific statistics that are kept by the printing mechanism. An example is the total number of pages printed by the printer. |

The host selects the category of printer status information desired by sending a RDS command with a subcommand in the data field that specifies the category of information desired.

The subcommand values for the various categories of status information to be returned by the printer are shown in Table 68.

**Table 68— RDS subcommands**

| Value (hex) | Category description | Notes |
|---|---|---|
| 00 | Summary | |
| 01 | Input Status | |
| 02 | Output Status | |
| 03 | Input Alert | |
| 04 | Output Alert | |
| 05 | Jam Alert | |
| 06 | Operator Intervention Required Alert | |
| 07 | Warnings | |
| 08 | Device Service Alert | |
| 09 | Configuration Change Alert | |
| 0A | Supplies Alert | |
| 0B | Printer Statistics | |
| 0C | Supplies Status | |
| 0C to FF | Reserved | |

### 5.5.2 Request Status Summary

The RDS Request Status Summary subcommand (RDS-RSS) provides the host system with an overview of the current status of the printer. The data returned by the printer allows the host to determine the printer's operational status and if any printer conditions exist that would require operator intervention or attention. Table 69 shows the format of the command sent to the printer to request the status summary.

**Table 69— RDS-RSS command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Status | Packet header |
| 1 | 00 | Subcommand: Request Status Summary | Data field |

After the printer receives the command, it shall return a status summary packet back to the host as shown in Table 70.

**Table 70— RDS-RSS response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Status | Packet header |
| 1 | 00 | Subcommand: Request Status Summary | Data field |
| 1 | Bit encoded | Overall Printer Status Summary | |
| 1 | Bit encoded | Device Alerts 1—Printer Status Alerts #1 | |
| 1 | Bit encoded | Device Alerts 2—Printer Status Alerts #2 | |

#### 5.5.2.1 Overall printer status summary

##### 5.5.2.1.1 Overall Printer Status Summary Codes

This is a bit-encoded field that provides an overview of the current operating condition of the printer. The bit definitions are shown in Table 71

**Table 71— RDS-RSS overall printer status summary codes**

| Bit | Description | | Notes |
|---|---|---|---|
| 0 | Power On Initialization | zero=No, one=Yes | |
| 1 | Reserved | | |
| 2 | Reserved | | |
| 3 | Reserved | | |
| 4 | Reserved | | |
| 5 | Printer Idle | zero=No, one=Yes | |
| 6 | Printer Off-line | zero=No, one=Yes | |
| 7 | Data Link Buffer Full | zero=No, one=Yes | |

##### 5.5.2.1.2 Bit 7—Data Link Buffer Full

The printer shall set the data link buffer full bit when any condition prevents the printer from continuing to accept data packets. The full indicator shall not be set if there are warning conditions that do not interfere with the immediate operation of the printer such as toner low in a electrophotograhic printer.

The printer shall always accept an RDS command regardless of the state of the full indicator. When full, the printer may reject commands other than the RDS command but this is printer-implementation dependent.

When this bit is set the host shall check the other status and alert condition indicators within this data packet to determine if the condition was caused by anything other than a link buffer full condition.

#### 5.5.2.1.3 Bit 6—Off-line

An off-line condition exists if an operator action or condition in the printer would cause the printer to halt the printing process pending action by an operator. If an off-line condition exists, then this bit shall be set. When off-line, the printer may reject commands other than the RDS command but this is printer implementation dependent.

#### 5.5.2.1.4 Bit 5—Printer Idle

The printer is inactive, awaiting instructions from a host.

#### 5.5.2.1.5 Bits 1–4—Reserved

These bits are reserved for future standardization. To ensure system upward compatibility, these bits shall be set to a zero value.

#### 5.5.2.1.6 Bit 0—Power On Initialization

Power On Initialization shall occur any time the printer performs a power on initialization or any other sequence where internal variables or states are returned to values equal to those resulting from a power-on initialization. This bit and the Power On Initialization alert bit in the device status alert shall be cleared after the Status Summary response is sent to the host.

### 5.5.2.2 Device Alerts 1 and 2—Printer Status Alerts #1 and #2

These bit-encoded fields are used to indicate the status of device alerts in the printer. Each bit corresponds to a different device alert category. If that bit has a value of zero, there is no active alert for that category. If the bit is a one, then an alert is active. If an active alert is indicated, the host can inquire about the details of that alert using the appropriate RDS subcommand. Table 72 and Table 73 define the bits and their associated alert categories.

**Table 72— RDS-RSS Device Alerts 1 codes**

| Bit | Alert category | Notes |
|---|---|---|
| 0 | Printing Supply Alert | |
| 1 | Paper Jam Alert | |
| 2 | Output Alert | |
| 3 | Input Alert | |
| 4 | Reserved | |
| 5 | Reserved | |
| 6 | Reserved | |
| 7 | Reserved | |

**Table 73— RDS-RSS Device Alerts 2 codes**

| Bit | Alert category | Notes |
|---|---|---|
| 0 | Configuration Change Alert | |
| 1 | Warnings Alert | |
| 2 | Device Service Required Alert | |
| 3 | Operator Intervention Required Alert | |
| 4 | Reserved | |
| 5 | Reserved | |
| 6 | Reserved | |
| 7 | Reserved | |

### 5.5.3 Request Input Status

The RDS Request Input Status subcommand (RDS-RIS) provides the host system with an overview of the status of printing media inputs. The host system can request status for inputs individually or collectively. Table 74 shows the format of the command sent to the printer.

**Table 74— RDS-RIS command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Status | Packet header |
| 1 | 01 | Subcommand: Request Input Status | Data field |
| 1 | ubyte | Input ID<br>00 = Return Status of All Inputs<br>xx = Return Data Concerning a specific input | |

The identification numbers for the various input devices of the printer are assigned by the printer. The host system obtains the identification numbers initially by issuing an RDC-RIC command requesting that the printer return data concerning all inputs.

Table 75 shows the format of the data packet returned to the host when the printer receives an RDS-RIS command.

**Table 75— RDS-RIS response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Status | Packet header |
| 1 | 01 | Subcommand: Request Input Status | Data field |
| 1 | ubyte | Number of Inputs IDs reporting | |
| 1 | ubyte | Input ID Number 1 | Input #1 |
| 2 | bword | Current Status of Input—Standard | |
| ... | ... | ... | Input n |

#### 5.5.3.1 Number of input IDs reporting

This unsigned byte defines how many inputs for which the printer shall be returning data. If the host requested status about a specific input, then this number shall be a value of one. If the status of all inputs was requested, then the number shall be the total number of inputs currently installed on the printer.

#### 5.5.3.2 Input identification number

This unsigned byte is the logical number that the printer has assigned to a particular input that the following data shall describe.

#### 5.5.3.3 Current status of input—Standard

##### 5.5.3.3.1 Input status codes

This is a bit-encoded word that describes the current status of a particular input. This field is reserved for standard definitions of printer status. The bits are defined in Table 76.

An alert shall be posted if paper (print media) is attempted to be fed from an empty or missing input paper source.

**Table 76— RDS-RIS input status codes**

| Bit | Description | | Notes |
|---|---|---|---|
| 0 | Current Input Level Bit 0 | | Paper picked with these 3 bits = 0 shall cause a DSA |
| 1 | Current Input Level Bit 1 | | |
| 2 | Current Input Level Bit 2 | | |
| 3 | On Request/Tray Missing | zero=No, one=Yes | Paper picked with this bit = one shall cause a DSA |
| 4 | Unavailable/Broken | zero=No, one=Yes | |
| 5 | Busy/Temporarily Unavailable | zero=No, one=Yes | |
| 6–14 | Reserved | | |
| 15 | Alert Active | zero=No, one=Yes | |

##### 5.5.3.3.2 Bit 15—Alert Active

This bit provides an indication if an alert is active for this input device. If this bit has a value of zero, then there is no alert active for this input.

##### 5.5.3.3.3 Bit 6–14—Reserved

These bits are reserved for future standardization. To ensure system upward compatibility, these bits shall be set to a value of zero.

##### 5.5.3.3.4 Bit 5—Busy/Temporarily Unavailable

This bit set indicates that the tray is allocated to some other process and is unavailable at this time. It shall return to an available state at a future time with no operator intervention.

##### 5.5.3.3.5 Bit 4—Unavailable/Broken

This bit set indicates that the tray is broken or unavailable. Intervention by an operator is required to correct this condition.

##### 5.5.3.3.6 Bit 3—On Request/Tray Missing

This bit indicates that a paper tray or other like device associated with the input is missing. A value of zero for the bit indicates that the tray is present.

##### 5.5.3.3.7 Bits 0, 1, 2—Current Input Level

These three bits together form a binary value that indicates the current level of printing media for the input device to the printer. Bit 0 is considered the least significant bit and bit 2 is considered the most significant bit. A binary value of zero for these bits indicate the input is empty and a value of seven indicates a full input. Values between zero and seven indicate various degrees of input capacity with the exact meaning printer implementation dependent.

### 5.5.4 Request Output Status

The RDS Request Output Status subcommand (RDS-ROS) provides the host system with an overview of the status of printing media output. The host system can request status of outputs individually or collectively. Table 77 show the format of the command sent to the printer.

**Table 77— RDS-ROS command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Status | Packet header |
| 1 | 02 | Subcommand: Request Output Status | Data field |
| 1 | ubyte | Output ID<br>00 = Return Status of All Outputs<br>xx = Return Data Concerning a specific output | |

The identification numbers for the various output devices of the printer are assigned by the printer. The host system initially obtains the identification numbers using the RDC-ROC command asking the printer to return information for all outputs.

After the printer receives the RDS-ROS command, it shall return a data packet back to the host similar to the one shown in Table 78.

**Table 78— RDS-ROS response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Status | Packet header |
| 1 | 02 | Subcommand: Request Output Status | Data field |
| 1 | ubyte | Number of Outputs IDs reporting | |
| 1 | ubyte | Output ID Number 1 | Output #1 |
| 2 | bword | Current Status of Output | |
| ... | ... | ... | Output n |

#### 5.5.4.1 Number of output IDs reporting

This unsigned byte defines the number of outputs for which data shall be returned. If the host requested status about a specific output, then this number shall be a value of one. If the status of all outputs was requested, then the number shall be the total number of outputs currently installed on the printer.

#### 5.5.4.2 Output identification number

This unsigned byte is the logical number that the printer has assigned to a particular output.

#### 5.5.4.3 Current status of output

##### 5.5.4.3.1 Output status codes

This is a bit-encoded word that describes the current status of a particular output. This field is reserved for standard definitions of printer status. The bits are defined in Table 79.

An alert shall be posted if the last sheet of paper (print media) that is stacked in the output tray causes the output to reach capacity or if an attempt is made to stack a sheet in a tray that is missing.

**Table 79— RDS-ROS output status codes**

| Bit | Description | | Notes |
|---|---|---|---|
| 0 | Current Output Level Bit 0 | | Output attempt with 3 bits = one shall cause a DSA |
| 1 | Current Output Level Bit 1 | | |
| 2 | Current Output Level Bit 2 | | |
| 3 | On Request/Tray Missing | zero=No, one=Yes | Output attempt with bit = one shall cause a DSA |
| 4 | Unavailable/Broken | zero=No, one=Yes | |
| 5 | Busy/Temporarily Unavailable | zero=No, one=Yes | |
| 6–14 | Reserved | | |
| 15 | Alert Active | zero=No, one=Yes | |

#### 5.5.4.3.2 Bit 15—Alert Active

This bit provides an indication if an alert is active for this output device. If this bit has a value of zero, then there is no alert active for this bit.

#### 5.5.4.3.3 Bits 6–14—Reserved

These bits are reserved for future standardization. To ensure system upward compatibility, these bits shall be set to a value of zero.

#### 5.5.4.3.4 Bit 5—Busy/Temporarily Unavailable

This bit set indicates that the tray is allocated to some other process and is unavailable at this time. It shall return to an available state at a future time with no operator intervention.

#### 5.5.4.3.5 Bit 4—Unavailable/Broken

This bit set indicates that the tray is broken or unavailable. Intervention by an operator is required to correct this condition.

#### 5.5.4.3.6 Bit 3—On Request/Tray Missing

This bit indicates that a paper tray or other like device associated with the output is missing. A value of zero for the bit indicates that the tray is present.

#### 5.5.4.3.7 Bits 0, 1, 2—Current Output Level

These three bits together form a binary value that indicates the current level on printing media for the output device to the printer. Bit 0 is considered the least significant bit and bit 2 is considered the most significant bit. A binary value of zero for these bits indicate the output is empty and a value of seven indicates a full output. Values between zero and seven indicate various degrees of output capacity with the exact meaning for a particular value being printer implementation dependent.

### 5.5.5 Request Input Alert

The RDS Request Input Alert subcommand (RDS-RIA) provides the host system the status of all inputs that have active alerts. The format of the command sent to the printer is shown in Table 80.

**Table 80— RDS-RIA command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Status | Packet header |
| 1 | 03 | Subcommand: Request Input Alert | Data field |

After the printer receives the RDS-RIA command, it shall return a data packet to the host similar to the one shown in Table 81.

**Table 81— RDS-RIA response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Status | Packet header |
| 1 | 03 | Subcommand: Request Input Alert | Data field |
| 1 | ubyte | Number of Input IDs reporting an Alert<br>00 = No Input alert active (following Data omitted) | |
| 1 | ubyte | Input ID Number | Input #1 |
| 2 | bword | Current Status of Input 1 | |
| 1 | ubyte (n) | Length of Alert Message 1 | |
| n | ASCII | Alert Message 1 | |
| ... | ... | ... | Input n |

#### 5.5.5.1 Current Status of Input

The data returned for this field is the same as that returned when the host issues an RDS-RIS command.

### 5.5.6 Request Output Alert

The RDS Request Output Alert subcommand (RDS-ROA) provides the host system the status of all outputs that have active alerts. The format of the command sent to the printer is shown in Table 82.

**Table 82— RDS-ROA command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Status | Packet header |
| 1 | 04 | Subcommand: Request Output Alerts | Data field |

After the printer receives the RDS-ROA command, it shall return a data packet to the host similar to the one shown in Table 83.

**Table 83— RDS-ROA response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Status | Packet header |
| 1 | 04 | Subcommand: Request Output Alert | Data field |
| 1 | ubyte | Number of Output IDs reporting an Alert<br>00 = No Output alert active (following Data omitted) | |
| 1 | ubyte | Output ID Number | Output #1 |
| 2 | bword | Current Status of Output 1 | |
| 1 | ubyte (n) | Length of Alert Message 1 | |
| n | ASCII | Alert Message 1 | |
| ... | ... | ... | Output n |

#### 5.5.6.1 Number of output IDs reporting an alert

This unsigned byte is the total number of printer outputs that have active alerts. If there are no outputs with active alerts, this field shall be zero. When this field has a value of zero, no additional data shall be returned in the data packet.

#### 5.5.6.2 Current Status of Output

The data returned for this field is the same as that returned when the host issues a RDS-ROS command.

### 5.5.7 Request Jam Alert

The RDS Request Jam Alert subcommand (RDS-RJA) provides the host with information about printing media jam conditions. The format of the command sent to the printer is shown in Table 84.

**Table 84— RDS-RJA command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Status | Packet header |
| 1 | 05 | Subcommand: Request Jam Alert | Data field |

The printer responds to the RDS-RJA command by sending data to the host concerning all jam conditions that can be currently identified, their location, and input/output identification number(s). Table 85 shows the format of the data returned.

**Table 85— RDS-RJA response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Status | Packet header |
| 1 | 05 | Subcommand: Request Jam Alert | Data field |
| 1 | ubyte | Number of Jam Alerts reporting<br>00 = No Jam Alert Active—the following data shall not be included | |
| 1 | ubyte | Location of Jam Alert Number 1 | Jam #1 |
| 1 | ubyte | ID of Location of Jam Alert 1<br>i.e., Input ID, Output ID<br>FF = Unknown | |
| 1 | ubyte | Position | |
| 1 | ubyte (n) | Length of Alert Message #1 in bytes (not including this byte) | |
| n | ASCII | Alert Message #1 | |
| ... | ... | ... | Jam n |

#### 5.5.7.1 Number of jam alerts reporting

This unsigned byte indicates the total number of jam locations that the printer has detected. For each jam location, three bytes and an associated ASCII string shall be returned. The three bytes specify the location, location identification number, and position. If there are no jam conditions existing in the printer at the time the command is received, the value of this field shall be set to zero and no additional data returned in the data packet.

#### 5.5.7.2 Location of jam alert

This unsigned byte specifies the general location of a jam alert within the printer. Table 86 defines the values associated with the different printer locations. If the location of the jam cannot be specified, a value of 255 shall be returned.

**Table 86— RDS-RJA location codes**

| Value (hex) | Description | Notes |
|---|---|---|
| 00 | Reserved | |
| 01 | Input | |
| 02 | Output | |
| 03 | Internal | |
| 04 | Duplex | |
| 05 | External | |
| 06 | Stacker | |
| 07 | Finisher | |
| 08 | Marker | |
| 09 | Logical Unit | |
| 0A to FD | Reserved | |
| FE | Other | |
| FF | Unknown | |

#### 5.5.7.3 ID of location of jam alert

This byte returns the identification number of the location of the jam that was specified by the previous byte. For example, if the jam occurred at an input, this byte would contain the identification number of the particular input where the jam occurred. If the printer could not determine the identification number, the printer shall return a value of 255 in this byte.

#### 5.5.7.4 Position

If an input device supports multiple positions, this byte would be used to report the position within the input device were the jam occurred. If the printer could not determine the position, the printer shall return a value of 255 in this byte.

#### 5.5.7.5 Length of alert message in bytes

This is an unsigned byte that defines the length of the following alert message. The value does not include this byte in the count. A value of zero indicates that there is no message to follow.

#### 5.5.7.6 Alert Message

This ASCII field is a message associated with a particular alert. The first part of the message is recommended to be an ASCII alphanumeric string delimited by a space character (hex 20) specifying a manufacturer's alert code for the remainder of the ASCII string. If there is no manufacturer's alert code associated with the ASCII string, the string is recommended to begin with a space character. The remainder of the ASCII string (that following the first space character) is recommended to be a human-readable message of the appropriate language detailing the alert condition.

### 5.5.8 Request Operator Intervention Required Alert

The RDS Request Operator Intervention Required Alerts subcommand (RDS-ROIRA) provides the host with information about those conditions in the printer that can cause the printer to stop printing. Operator Intervention Required conditions (OIR) are situations that can be corrected by a normal or nontechnical operator. An example of this type of message would be: "Cover Open." OIR alerts do not include those conditions that are covered in the input, output, and jam alert categories.

For each OIR alert the printer shall return a location, an ID, and an error code with an associated ASCII text string that describes the particular alert. The host can use the text string to display the message to the user and use the location, ID, and error code as an index entry into a table if additional processing is required. The printer may have more than one OIR alert active at any given time. All OIR alerts active in the printer shall be returned when the host requests OIR alerts from the printer. The host requests the OIR alerts by sending a RDS-ROIRA command as shown in Table 87.

 

**Table 87— RDS-ROIRA command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Status | Packet header |
| 1 | 06 | Subcommand: Request Operator Intervention Required Alerts | Data field |

The response by the printer to the RDS-ROIRA command is shown in Table 88.

**Table 88— RDS-ROIRA response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Status | Packet header |
| 1 | 06 | Subcommand: Request Operator Intervention Required Alerts | Data field |
| 1 | ubyte | Number of alert messages to follow | |
| 1 | ubyte | Location #1 | Alert #1 |
| 1 | ubyte | ID #1 | |
| 1 | ubyte | Alert Code #1 | |
| 1 | ubyte (n) | Length of Alert Message #1 in bytes (not including this byte) | |
| n | ASCII | Alert Message #1 | |
| ... | ... | ... | Alert n |

#### 5.5.8.1 Number of alert messages to follow

This unsigned byte is the total number of OIR alerts active in the printer and the number of message fields to follow in the data packet.

#### 5.5.8.2 Location of alert

This unsigned byte specifies the general location of the alert within the printer. Table 86 defines the values associated with the different locations with in the printer. If the location of the alert cannot be specified, a value of 255 shall be returned.

#### 5.5.8.3 ID

This unsigned byte returns the identification number of the location of the alert that was specified by the previous byte. A value of 255 in this byte indicates that an ID does not exist for this alert.

#### 5.5.8.4 Alert code

This is an unsigned byte that specifies the type of OIR alert. This byte is used by the host system to search through tables for additional error handling procedures that may be required or for printer-specific error codes. Table 89 shows the alert codes and their description.

**Table 89— RDS-ROIRA codes and messages**

| Alert code | Alert description | Notes |
|---|---|---|
| 00 | Reserved | |
| 01 | “Cover or Door Open or Ajar” | |
| 02 | “Waste Toner Box Full” | |
| 03 | “Print Ribbon Jam” | |
| 04 | “Waste Ink Reservoir Full” | |
| 05 to FD | Reserved | |
| FE | Other | |
| FF | Unknown | |

#### 5.5.8.5 Length of alert message in bytes

This is an unsigned byte that defines the length of the following alert message. The value does not include this byte in the count. A value of zero indicates that there is no message to follow.

#### 5.5.8.6 Alert Message

This ASCII field is a message associated with a particular alert. The first part of the message is recommended to be an ASCII alphanumeric string delimited by a “blank” specifying a manufacturer’s alert code for the remainder of the ASCII string. If there is no manufacturer’s alert code associated with the ASCII string, the string is recommended to begin with a “blank.” The remainder of the ASCII string (that following the first “blank”) is recommended to be a human-readable message of the appropriate language detailing the alert condition.

### 5.5.9 Request Warnings

The RDS Request Warnings subcommand (RDS-RW) provides the host with information about warning conditions that currently exist in the printer. Warnings are conditions in the printer that are not critical to immediate operation. Examples of warnings are “Paper Input Low,” “Toner Low,” or “Output Bin nearly Full.”

For each warning the printer shall return a location, an ID, and a warning code with an associated ASCII text string that describes the particular warning. The host can use the text string to display the message to the user and use the location, ID, and warning code as an index entry into a table if additional processing is required. The printer may have more than one warning active at any one time. All warnings active in the printer shall be returned when the host requests the warnings from the printer. Table 90 shows the format of the command sent to the printer.

**Table 90— RDS-RW command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Status | Packet header |
| 1 | 07 | Subcommand: Request Warnings | Data field |

The response by the printer to the RDS-RW command is shown in Table 91.

**Table 91— RDS-RW response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Status | Packet header |
| 1 | 07 | Subcommand: Request Warnings | Data field |
| 1 | ubyte | Number of warning messages to follow | |
| 1 | ubyte | Location #1 | Warning #1 |
| 1 | ubyte | ID #1 | |
| 1 | ubyte | Warning Code #1 | |
| 1 | ubyte (n) | Length of Warning Message #1 in bytes (not including this byte) | |
| n | ASCII | Warning Message #1 | |
| ... | ... | ... | Warning n |

#### 5.5.9.1 Number of warning messages to follow

This unsigned byte is the total number of warning messages active in the printer and the number message fields to follow in the data packet.

#### 5.5.9.2 Location of warning

This unsigned byte specifies the general location of the warning within the printer. Table 86 defines the values associated with the different locations within the printer. If the location of the warning cannot be specified, a value of 255 shall be returned.

#### 5.5.9.3 ID

This byte returns the identification number of the location of the warning that was specified by the previous byte that indicated the location. For example, if multiple toner supply units can generate the “Toner Supply Low” warning, then this byte can be used to distinguish which toner supply unit is low. A value of 255 in this byte indicates that an ID does not exist for this alert.

#### 5.5.9.4 Warning code

This is an unsigned byte that specifies the type of warning. Table 92 shows the warning codes and their description.

**Table 92— RDS-RW codes and descriptions**

| Alert code | Alert description | Notes |
|---|---|---|
| 00 | Reserved | |
| 01 | “Toner Supply Low” | |
| 02 | “Ink Supply Low” | |
| 03 | “Paper Input Low” | |
| 04 | “Output Bin Nearly Full” | |
| 05 | “Cover or Panel Open” | |
| 06 | “Fusing Supplies Low” | |
| 07 | “Other” | |
| 08 | “Supply Low” | |
| 09 | “Supply Nearly Full” | |
| 0A | “Supply Missing” | |
| 0B to FD | Reserved | |
| FE | Other | |
| FF | Unknown | |

#### 5.5.9.5 Length of warning message in bytes

This is an unsigned byte that defines the length of the following warning message. The value does not include this byte in the count. A value of zero indicates that no warning message is to follow.

#### 5.5.9.6 Warning Message

This ASCII field is a message associated with a particular warning type. The first part of the message is recommended to be an ASCII alphanumeric string delimited by a “blank” specifying a manufacturer’s warning code for the remainder of the ASCII string. If there is no manufacturer’s warning code associated with the ASCII string, the string is recommended to begin with a “blank.” The remainder of the ASCII string (that following the first “blank”) is recommended to be a human-readable message of the appropriate language detailing the alert condition.

### 5.5.10 Device Service Required Alerts

The RDS Request Device Service Required Alerts subcommand (RDS-RDSRA) provides the host with information about those conditions in the printer that can cause the printer to stop printing. Device Service Required (DSR) conditions are situations that cannot be corrected by a normal or nontechnical operator. An example of this type of message would be: “Laser Polygon Scanner Inoperative.” DSR Alerts do not include those conditions that are covered in the input, output, and jam alert categories.

For each DSR alert the printer shall return a location, an ID, and an error code with an associated ASCII text string that describes the particular alert. The host can use the text string to display the message to the user and use the location, ID, and error code as an index entry into a table if additional processing is required. The printer may have more than one DSR alert active at any time. All DSR alerts active in the printer shall be returned when the host requests the DSR alerts from the printer. Table 93 shows the format of the command sent to the printer.

**Table 93— RDS-RDSRA command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Status | Packet header |
| 1 | 08 | Subcommand: Request Device Service Required Alerts | Data field |

The response by the printer to the RDS-RDSRA command is shown in Table 94.

**Table 94— RDS-RDSRA response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Status | Packet header |
| 1 | 08 | Subcommand: Request Device Service Required Alerts | Data field |
| 1 | ubyte | Number of alert messages to follow | |
| 1 | ubyte | Location #1 | Alert #1 |
| 1 | ubyte | ID #1 | |
| 1 | ubyte | Alert Code #1 | |
| 1 | ubyte (n) | Length of Alert Message #1 in bytes (not including this byte) | |
| n | ASCII | Alert Message #1 | |
| ... | ... | ... | Alert n |

#### 5.5.10.1 Number of alert messages to follow

This unsigned byte is the total number of DSR alert messages active in the printer and the number of message fields to follow in the data packet.

#### 5.5.10.2 Location of alert

This unsigned byte specifies the general location of the alert within the printer. Table 86 defines the values associated with the different locations with in the printer. If the location of the alert cannot be specified a value of 255 shall be returned.

#### 5.5.10.3 ID

This byte returns the identification number of the location of the alert that was specified by the previous byte. A value of 255 in this byte indicates that an ID does not exist for this alert.

#### 5.5.10.4 Alert code

This is an unsigned byte that specifies the code associated with the DSR alert. Table 95 shows the error codes.

**Table 95— RDS-RDSRA codes and messages**

| Alert code | Alert description | Notes |
|---|---|---|
| 00 | Reserved | |
| 01 | "Hard Error"<br>A hard error is one that cannot be recovered from. | |
| 02 | "Soft Error"<br>A soft error is an error that was detected and may or may not persist. | |
| 03 to FE | Reserved | |
| FF | Unknown | |

#### 5.5.10.5 Length of alert message in bytes

This is an unsigned byte that defines the length of the following alert message. The value does not include this byte in the count. A value of zero indicates that no message follows.

#### 5.5.10.6 Alert Message

This ASCII field is a message associated with a particular alert. The first part of the message is recommended to be an ASCII alphanumeric string delimited by a "blank" specifying a manufacturer's alert code for the remainder of the ASCII string. If there is no manufacturer's alert code associated with the ASCII string, the string is recommended to begin with a "blank." The remainder of the ASCII string (that following the first "blank") is recommended to be a human-readable message of the appropriate language detailing the alert condition.

### 5.5.11 Request Configuration Change Alert

The RDS Request Configuration Change Alert subcommand (RDS-RCCA) provides the host with information about conditions that were detected that changed the current configuration of the printer. A configuration change alert may or may not be critical to the immediate operation of the printer. Examples of alerts are "Font Cartridge Removed" or "RAM memory board failure." If the host receives a Configuration Change Alert, the host shall make the appropriate inquiries of the printer to access the extent of the change and to verify the current printer state.

For each alert the printer shall return a location, an ID, and an error code with an associated ASCII text string that describes the particular alert. The host can use the text string to display the message to the user and use the location, ID, and error code as an index entry into a table if additional processing is required. The printer may have more than one alert active at any time. All active alerts in the printer shall be returned when the host sends the RDS-RCCA command to the printer. Table 96 shows the format of the command sent to the printer.

**Table 96— RDS-RCCA command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Status | Packet header |
| 1 | 09 | Subcommand: Request Configuration Change Alerts | Data field |

The printer response to the RDS-RCCA command is shown in Table 97.

**Table 97— RDS-RCCA response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Status | Packet header |
| 1 | 09 | Subcommand: Request Configuration Change Alerts | Data field |
| 1 | ubyte | Number of Alert messages to follow | |
| 1 | ubyte | Location #1 | Alert #1 |
| 1 | ubyte | ID #1 | |
| 1 | ubyte | Alert Code #1 | |
| 1 | ubyte (n) | Length of Alert Message #1 in bytes (not including this byte) | |
| n | ASCII | Alert Message #1 | |
| ... | ... | ... | Alert n |

#### 5.5.11.1 Number of alert messages to follow

This unsigned byte is the total number of alert messages active in the printer and the number message fields to follow in the data packet.

#### 5.5.11.2 Location of alert

This unsigned byte specifies the general location of the alert within the printer. Table 86 defines the values associated with the different locations with in the printer. If the location of the alert cannot be specified, a value of 255 shall be returned.

#### 5.5.11.3 ID

This unsigned byte returns the identification number of the location of the alert that was specified by the previous byte. A value of 255 in this byte indicates that an ID does not exist for this alert.

#### 5.5.11.4 Alert code

This is an unsigned byte that specifies the code associated with the alert message. Table 98 shows the alert codes and their descriptions.

**Table 98— RDS-RCCA codes and descriptions**

| Alert code | Alert description | Notes |
|---|---|---|
| 00 | Reserved | |
| 01 | “Interpreter Related Resource Deleted” | |
| 02 | “Interpreter Related Resource Added” | |
| 03 | “Reserved” | |
| 04 | “Medium Size Changed” | |
| 05 | “Cartridge Removed” | Not related to a data stream |
| 06 | “Cartridge Installed” | |
| 07 | “Installed Memory Reduced” | |
| 08 | “Installed Memory Increased” | |
| 09 | “Option Removed” | |
| 0A | “Option Installed” | |
| 0B to FD | Reserved | |
| FE | “Configuration Change”—Printer unable to classify change, but there is a way that the host can query the printer and determine the specific change by interpreting the printer’s responses. | |
| FF | Unknown | |

#### 5.5.11.5 Length of alert message in bytes

This is an unsigned byte that defines the length of the following alert message. The value does not include this byte in the count. A value of zero indicates that no message is to follow.

#### 5.5.11.6 Alert Message

This ASCII field is a message associated with a particular alert. The first part of the message is recommended to be an ASCII alphanumeric string delimited by a “blank” specifying a manufacturer’s alert code for the remainder of the ASCII string. If there is no manufacturer’s alert code associated with the ASCII string, the string is recommended to begin with a “blank.” The remainder of the ASCII string (that following the first “blank”) is recommended to be a human-readable message of the appropriate language detailing the alert condition.

### 5.5.12 Supplies Alerts

The RDS Request Supplies Alerts subcommand (RDS-RSA) provides the host with information about those conditions associated with supplies in the printer that cause the printer to stop printing. Supplies alert conditions are situations that can be corrected by a normal or nontechnical operator. An example of this type of message would be “Out of Ink” or “Out of Toner.”

For each alert the printer shall return a location, an ID, and an error code with an associated ASCII text string that describes the particular alert. The host can use the text string to display the message to the user and use the location, ID, and error code as an index entry into a table if additional processing is required. The printer may have more than one supplies alert active at any time. All active supplies alerts in the printer shall be returned when the host sends the RDS-RSA command to the printer. Table 99 shows the format of the command sent to the printer.

**Table 99— RDS-RSA command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Status | Packet header |
| 1 | 0A | Subcommand: Request Supplies Alerts | Data field |

The printer response to the RDS-RSA command is shown in Table 100.

**Table 100— RDS-RSA response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Status | Packet header |
| 1 | 0A | Subcommand: Request Supplies Alerts | Data field |
| 1 | ubyte | Number of alert messages to follow | |
| 1 | ubyte | Location #1 | Alert #1 |
| 1 | ubyte | ID #1 | |
| 1 | ubyte | Alert Code #1 | |
| 1 | ubyte (n) | Length of Alert Message #1 in bytes (not including this byte) | |
| n | ASCII | Alert Message #1 | |
| ... | ... | ... | Alert n |

#### 5.5.12.1 Number of alert messages to follow

This unsigned byte is the total number of active supplies alert messages in the printer and the number of message fields to follow in the data packet.

#### 5.5.12.2 Location of alert

This unsigned byte specifies the general location of the alert within the printer. Table 86 defines the values associated with the different locations with in the printer. If the location of the alert cannot be specified, a value of 255 shall be returned.

#### 5.5.12.3 ID

This byte returns the identification number of the location of the alert that was specified by the previous byte. A value of 255 in this byte indicates that an ID does not exist for this alert.

#### 5.5.12.4 Alert code

This is an unsigned byte that specifies the code associated with the supplies alert message. Table 101 shows the alert codes and descriptions.

**Table 101— RDS-RSA codes and descriptions**

| Alert code | Alert description | Notes |
|---|---|---|
| 00 | Reserved | |
| 01 | “Out of Toner” | |
| 02 | “Out of Ink” | |
| 03 | “Ribbon Out” | |
| 04 | “Supply Missing” | |
| 05 | “Supply Empty” | |
| 06 | “Supply Full” | |
| 07 to FD | Reserved | |
| FE | Other | |
| FF | Unknown | |

#### 5.5.12.5 Length of alert message in bytes

This is an unsigned byte that defines the length of the following alert message. The value does not include this byte in the count. A value of zero indicates that no alert message is to follow.

#### 5.5.12.6 Alert Message

This ASCII field is a message associated with a particular alert. The first part of the message is recommended to be an ASCII alphanumeric string delimited by a “blank” specifying a manufacturer’s alert code for the remainder of the ASCII string. If there is no manufacturer’s alert code associated with the ASCII string, the string is recommended to begin with a “blank.” The remainder of the ASCII string (that following the first “blank”) is recommended to be a human-readable message of the appropriate language detailing the alert condition.

### 5.5.13 Request Printer Statistics

The RDS Request Printer Statistics subcommand (RDS-RPS) provides the host system with statistics that are kept by the printer. These statistics are related to the overall printer and not necessarily related to any one particular printing job or period of time. Table 102 shows the format of the RDS-RPS command sent by the host to the printer.

**Table 102— RDS-RPS command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Status | Packet header |
| 1 | 0B | Subcommand: Request Printer Statistics | Data field |

Upon receiving the RDS-RPS command, the printer shall issue a response similar to the one shown in Table 103.

**Table 103— RDS-RPS response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Status | Packet header |
| 1 | 0B | Subcommand: Request Printer Statistics | Data field |
| 1 | ubyte | Number of Statistics Reporting | |
| 1 | ubyte | Statistic Type | Statistic #1 |
| 4 | Unsigned dword | Statistic Value | |
| ... | ... | ... | Statistic n |

#### 5.5.13.1 Number of statistics reporting

This unsigned byte defines the number of statistics that this printer response contains. If the value is zero, then the printer does not have the capability to report any statistics and there shall not be any additional data to follow.

#### 5.5.13.2 Statistic type

This unsigned byte defines the type of statistic the printer is reporting. Table 104 defines the codes and its associated printer statistic.

**Table 104— RDS-RPS codes**

| Value | Description | Notes |
|---|---|---|
| 00 | Reserved | |
| 01 | Total counter units printed since beginning of printer life | |
| 02 | Total counter units printed since printer power on | |
| 03 | Total counter units on current printing supplies | |
| 04 | Total counter units on host controlled counter | |
| 05 to FE | Reserved | |
| FF | Unknown | |

##### 5.5.13.2.1 Total counter units printed since beginning of printer life

This statistic is the total number of counter units (e.g., pages, inches, etc.) printed since the initial manufacture of the printer. This counter is defined by the manufacturer and cannot be reset by the host.

##### 5.5.13.2.2 Total counter units printed since power on

This statistic is the total number of counter units (e.g., pages, inches, etc.) printed since the last time the printer has been turned on. This counter is only reset when the printer executes its power-on initialization sequence.

##### 5.5.13.2.3 Total counter units on current printing supplies

This counter is the total number of counter units (e.g., pages, inches, etc.) printed on the current printing supplies. For example, this is the number of pages printed since the toner cartridge or ribbon was changed in a laser printer or impact printer. The method of resetting the counter is dependent on the manufacturer of the printer.

##### 5.5.13.2.4 Total counter units on host-controlled counter

This statistic is the total number of counter units (e.g., pages, inches, etc.) printed since the last time the counter has been reset. This counter is reset in one of two ways. First, the counter is reset every time the printer executes its power-on initialization sequence. Second, the host can reset the counter by sending a “Reset Host Controlled Counter” subcommand of the Printer Configuration Control command.

#### 5.5.13.3 Statistic value

This unsigned double word is the value associated with the previous statistic type field. The units associated with this value are defined by the printer and reported in the counter units field of the RDC-RS command.

### 5.5.14 Request Supplies Status

The RDS Request Supplies Status subcommand (RDS-RSuS) provides the host with an overview of the status of supplies other than input medium of the printer. Table 105 shows the format of the RDS-RSuS command sent to the printer.

**Table 105— RDS-RSuS command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Status | Packet header |
| 1 | 0C | Subcommand: Request Supplies Status | Data field |
| 1 | ubyte | Supply ID<br>00 = Return Status of All Supplies | |

The printer's response to the RDS-RSuS command is shown in Table 106.

**Table 106— RDS-RSuS response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Request Device Status | Packet header |
| 1 | 0C | Subcommand: Request Supplies Status | Data field |
| 1 | ubyte | Number of Supply IDs reporting | |
| 1 | ubyte | Location #1 | Supply ID #1 |
| 1 | ubyte | ID #1 | |
| 2 | bword | Current Status of Supplies #1 | |
| ... | ... | ... | Supply ID n |

#### 5.5.14.1 Number of supply IDs reporting

This unsigned byte defines how many supplies for which the printer shall be returning data.

#### 5.5.14.2 Location of supply

This unsigned byte specifies the general location of the supply within the printer. Table 86 defines the values associated with the different locations with in the printer. If the location of the supply cannot be specified, a value of 255 shall be returned.

#### 5.5.14.3 ID

This byte returns the identification number of the location of the supply that was specified by the previous byte. A value of 255 in this byte indicates that an ID does not exist for this supply.

#### 5.5.14.4 Current status of supplies

This is a bit-encoded byte that describes the current status of the expendable printing supplies (ribbon, ink, toner, etc.). This field is reserved for standard definitions of printer status. The bit definitions for this byte are shown in Table 107.

An alert shall be posted if an attempt is made to use an empty supply.

**Table 107— RDS-RSuS codes**

| Bit | Description | Notes |
|---|---|---|
| 0 | Supplies Level Bit 0 | |
| 1 | Supplies Level Bit 1 | |
| 2 | Supplies Level Bit 2 | |
| 3–14 | Reserved | |
| 15 | Alert Active zero=No, one=Yes | |

#### 5.5.14.4.1 Bit 15—Alert Active

This bit provides an indication if an alert is active for this printing supply. If this bit has a value of zero, then there is no alert active.

#### 5.5.14.4.2 Bits 3-14—Reserved

These bits are reserved for future standardization. To ensure system upward compatibility, these bits shall be set to a value of zero.

#### 5.5.14.4.3 Bits 2, 1, 0—Supplies Level

These three bits together form a binary value that indicates the current level of the standard printing supplies (e.g., black ink, toner, ribbon, etc.). Bit 0 is the least significant bit and bit 2 is the most significant bit. A binary value of zero for these bits indicates the supply is empty and a value of seven indicates that it is full. Values between zero and seven indicate various degrees of supply on hand with the exact meaning printer-implementation dependent.

## 5.6 Printer Configuration Control (PCC)

### 5.6.1 Overview

The PCC command can be used by the host to set various parameters to configure operations within a printer that supports the IEEE Std 1284.1-1997 protocol. The PCC subcommands are shown in Table 108.

**Table 108— PCC subcommands**

| Value | Description | Notes |
|---|---|---|
| 00 | Read Current Printer Configuration | |
| 01 | Reset Printer | |
| 02 | *deprecated* | |
| 03 | Select Device Status Alerts | |
| 04 | Data Loop Back | |
| 05 | Select Host Packet Size | |
| 06 | Reset Host Controlled Counter | |
| 07 | Select Interpreter Messages | |
| 08 | Read Interpreter Messages Selected | |
| 09 | Select Job Alerts | |
| 0A | Read Job Alerts Selected | |
| 0B | Set Printer ID | |
| 0C | Enable Common Printer MIB Alert | |
| 0D | Disable Common Printer MIB Alert | |
| 0E | Read Printer ID | |
| 0F to FF | Reserved | |

### 5.6.2 Read Current Printer Configuration

The PCC Read Current Printer Configuration command (PCC-RCPC) allows the host to read the active printer settings selected through prior use of the PCC command. Table 109 shows the format of the PCC-RCPC command.

**Table 109— PCC-RCPC command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Printer Configuration Control | Packet header |
| 1 | 00 | Subcommand: Read Current Printer Configuration | Data field |

The printer shall respond to the PCC-RCPC command with a response similar to the one shown in Table 110.

**Table 110— PCC-RCPC response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Printer Configuration Control | Packet header |
| 1 | 00 | Subcommand: Read Current Printer Configuration | Data field |
| 2 | uword | Maximum Size in Bytes of the Data Packet to be Sent to the Host | |
| 1 | bbyte | Overall Printer Status Standard Enabled<br>Bit 0—Power On Initialization, always a one<br>Bits 1–5—Reserved, always a zero<br>Bit 6—Printer Off-line Alerts<br>Bit 7—Printer Busy Alerts | Value of zero=Do *not* Send DSA Message for the category of Printer Status |
| 1 | bbyte | Device Alerts Byte 1 Enabled<br>Bit 0—Printing Supply Alerts<br>Bit 1—Paper Jam Alerts<br>Bit 2—Output Alerts<br>Bit 3—Input Alerts<br>Bits 4-7—Reserved, always a zero | Value of one=Send DSA Message for the category of Printer Status |
| 1 | bbyte | Device Alerts Byte 2 Enabled<br>Bit 0—Configuration Change Alerts<br>Bit 1—Warning Alerts<br>Bit 2—Device Service Required Alerts<br>Bit 3—Operator Intervention Alerts<br>Bits 4–7—Reserved, always a zero | |
| 1 | bbyte | Deprecated | |

### 5.6.3 Reset Printer

The PCC Reset Printer command (PCC-RP) allows the host to reset the printer. Using the PCC-RP command, the printer can be placed in various reset states by the host. Table 111 shows the format of the PCC-RP command sent to the printer.

**Table 111— PCC-RP command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Printer Configuration Control | Packet header |
| 1 | 01 | Subcommand: Reset Printer | Data field |
| 1 | ubyte | Type of Reset<br>01 Reset printer to power on condition. This returns the printer to the 1284.1 initialized state, downloaded resources shall be deleted (or whatever normally happens to them when a Power-On-Reset occurs).<br>02. Reset 1284.1 Layer only and leave rest of printer, interpreter, in current configuration. This reset does not affect resources or the operating environment associated with the interpreters that are active.<br>03. Reset printer to power on condition and exit 1284.1. This reset does not preserve any temporary resources as in #1.<br>04. Reset Printer and exit 1284.1. 1284.1 does not exist until the printer is reset. Temporary resources are preserved.<br>05–FF Reserved | |
| | | **Reset 1284.1 State Comm Link State Interpreter**<br>01 1284.1 Default Default<br>02 1284.1 Default Unchanged<br>03 non-1284.1 non-1284.1 default Default<br>04 non-1284.1 non-1284.1 default Unchanged | |

### 5.6.4 Select Device Status Alerts

The PCC Select Device Status Alerts command (PCC-SDSA) allows the host to arm the conditions within the printer for which the printer shall transmit a Device Status Alert (DSA) message to the host. Table 112 shows the format of the PCC-SDSA command.

**Table 112— PCC-SDSA command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Printer Configuration Control | Packet header |
| 1 | 03 | Subcommand: Select Device Status Alerts | Data field |
| 1 | bbyte | Overall Printer Status Standard<br>Bit 0—Power On Initialization, ignored<br>Bits 1–4—Reserved, shall be zero<br>Bit 5—Printer Idle Alerts zero=No, one=Yes<br>Bit 6—Printer Off-line Alerts zero=No, one=Yes<br>Bit 7—Data Link Buffer Full Alerts zero=No, one=Yes | |
| 1 | bbyte | Device Alerts Byte 1<br>Bit 0—Printing Supply Alerts zero=No, one=Yes<br>Bit 1—Paper Jam Alerts zero=No, one=Yes<br>Bit 2—Output Alerts zero=No, one=Yes<br>Bit 3—Input Alerts zero=No, one=Yes<br>Bits 4–7—Reserved, shall be zero | |
| 1 | bbyte | Device Alerts Byte 2<br>Bit 0—Configuration Change Alert zero=No, one=Yes<br>Bit 1—Warnings zero=No, one=Yes<br>Bit 2—Device Service Required Alerts zero=No, one=Yes<br>Bit 3—Operator Intervention Alerts zero=No, one=Yes<br>Bits 4–7—Reserved, shall be zero | |

If a condition that would cause a DSA is active at the time that its DSA is armed using the PCC-SDSA command, then a DSA shall be sent.

The reset condition is that all conditions that cause DSA messages are disabled until explicitly enabled by the host.

### 5.6.5 Data Loop-Back

The PCC Data Loop-Back command (PCC-DLB) provides a means for the host to test the communication link to the printer. Using the PCC-DLB command the host sends data to the printer that upon reception by the printer shall immediately be sent back to the host. The minimum packet size that is supported is 64 bytes. The maximum packet size cannot exceed the Maximum Receive Command Packet Size as specified in the RDC-RS response. The Continue bit in the Flag byte of the Packet header shall be ignored on this command. Table 113 shows the format of the PCC-DLB command.

**Table 113— PCC-DLB command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Printer Configuration Control | Packet header |
| 1 | 04 | Subcommand: Data Loop Back | Data field |
| n | | Data to be sent to the printer | |

Table 114 shows the format of the printer’s response to the PCC-DLB command.

**Table 114— PCC-DLB response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Printer Configuration Control | Packet header |
| 1 | 04 | Subcommand: Data Loop Back | Data field |
| n | | Data to be sent back to the host | |

### 5.6.6 Select Host Packet Size

The PCC Select Host Packet Size command (PCC-SHPS) is sent by the host to the printer to set the maximum size data packet that the printer can use when sending data to the host. The packet size specifies the length of the entire data packet including the start of packet byte, length bytes, command, subcommand, and data bytes. If the amount of data the printer has to send to the host exceeds the maximum size set by the host, then it shall send the data in multiple packets. Table 115 shows the format of the PCC-SHPS command.

The printer shall always initialize the maximum host packet size to 64 bytes, the default packet size, during power-on initialization. The PCC-RP command shall not change this value.

**Table 115— PCC-SHPS command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Printer Configuration Control | Packet header |
| 1 | 05 | Subcommand: Select Host Packet Size | Data field |
| 2 | uword | Maximum Size in Bytes of the Data Packet to be Sent to the Host | |

### 5.6.7 Reset Host Controlled Counter

The PCC Reset Host Controlled Counter command (PCC-RHCC) is used by the host to reset the host controlled counter which tallies counter units. This counter shall only be reset by the PCC-RHCC command or by power on initialization. Table 116 shows the format of the PCC-RHCC command.

The value of the host controlled counter is read using the RDS-RPS command.

**Table 116— PCC-RHCC command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Printer Configuration Control | Packet header |
| 1 | 06 | Subcommand: Reset Host Controlled Counter | Data field |

### 5.6.8 Select Interpreter Messages

The PCC Select Interpreter Messages command (PCC-SIM) allows the host to disable/enable the printer to return to the host messages from an interpreter or logical unit. These are responses that the interpreter generates on its own independent of the IEEE 1284.1 command layer. Table 117 shows the format of the PCC-SIM command.

**Table 117— PCC-SIM command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Printer Configuration Control | Packet header |
| 1 | 07 | Subcommand: Select Interpreter Messages | Data field |
| 1 | ubyte | Logical Unit Number (Interpreter)<br>00 Default Interpreter<br>xx Enable or Disable Interpreter Messages for this interpreter | |
| 1 | bbyte | Enable Interpreter Status Alerts #1<br>Bit 0—Response to Interpreter Queries (zero=No, one=Yes, zero=default)<br>Bits 1–7—Reserved | |
| 1 | bbyte | Enable Interpreter Status Alerts #2<br>Bits 0–7—Reserved | |

### 5.6.9 Read Interpreter Messages Selected

The PCC Read Interpreter Messages Selected command (PCC-RIMS) allows the host to view the printer settings as to whether or not unsolicited interpreter messages of a particular category shall be returned to the host or discarded within the printer. These are responses that the interpreter generates on its own independent of the IEEE 1284.1 command layer. Table 118 shows the format of the PCC-RIMS command.

**Table 118— PCC-RIMS command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Printer Configuration Control | Packet header |
| 1 | 08 | Subcommand: Read Interpreter Messages Selected | Data field |
| 1 | ubyte | Logical Unit Number (Interpreter)<br>00 Default Interpreter<br>xx Specific Interpreter | |

Table 119 shows the format of the printer's response to the PCC-RIMS command.

**Table 119— PCC-RIMS response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Printer Configuration Control | Packet header |
| 1 | 08 | Subcommand: Read Interpreter Messages Selected | Data field |
| 1 | ubyte | Logical Unit Number (Interpreter)<br>00 Default Interpreter<br>xx Specific Interpreter | |
| 1 | bbyte | Enable Interpreter Status Alerts #1<br>Bit 0—Response to Interpreter Queries (zero=No, one=Yes)<br>Bits 1–7—Reserved | |
| 1 | bbyte | Enable Interpreter Status Alerts #2<br>Bits 0–7—Reserved | |

### 5.6.10 Select Job Alerts

The PCC Select Job Alerts command (PCC-SJA) allows the host to disable/enable the printer to return to the host job alerts. Table 120 shows the format of the PCC-SJA command.

**Table 120— PCC-SJA command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Printer Configuration Control | Packet header |
| 1 | 09 | Subcommand: Select Job Alerts | Data field |
| 1 | bbyte | Job Alert Points<br>Bit 0 reserved<br>Bit 1 Logical Unit Start (zero=Off, one=On)<br>Bit 2 reserved<br>Bit 3 reserved<br>Bit 4 Page Alert zero=Off, one=On)<br>Bit 5 reserved<br>Bit 6 End of Job (zero=Off, one=On)<br>Bit 7 Job Accounting (zero=Off, one=On) | |

### 5.6.11 Read Job Alerts Selected

The PCC Read Job Alerts Selected command (PCC-RJAS) allows the host to view the printer settings as to whether or not job alerts shall be returned to the host. Table 121 shows the format of the PCC-RJAS command.

**Table 121— PCC-RJAS command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Printer Configuration Control | Packet header |
| 1 | 0A | Subcommand: Read Job Alerts Selected | Data field |

Table 122 shows the format of the printer's response to the PCC-RJAS command.

**Table 122— PCC-RJAS response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Printer Configuration Control | Packet header |
| 1 | 0A | Subcommand: Read Job Alerts Selected | Data field |
| 1 | bbyte | Job Alert Points<br>Bit 0 reserved<br>Bit 1 Logical Unit Start (zero=Off, one=On)<br>Bit 2 reserved<br>Bit 3 reserved<br>Bit 4 Page Alert (zero=Off, one=On)<br>Bit 5 reserved<br>Bit 6 Extended Job Complete (zero=Off, one=On)<br>Bit 7 Job Accounting (zero=Off, one=On) | |

### 5.6.12 Set Printer ID

The PCC Set Printer ID command (PCC-SPI) is used by the host to set the ASCII string that is to be returned in the JCA-EOJ accounting message alert from the printer. The Printer ID's power on initialization state is its serial number. The command's format is shown in Table 123.

**Table 123— PCC-SPI command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Printer Configuration Control | Packet header |
| 1 | 0B | Subcommand: Set Printer ID | Data field |
| 1 | ubyte | Length of Printer ID | |
| 1–63 | ASCII | Printer ID | |

### 5.6.13 Read Printer ID

The PCC Read Printer ID command (PCC-RPI) is used by the host to read the ASCII string that the printer shall return in the JCA-EOJ accounting message. Table 124 describes the format of the command sent from the host to the printer. Table 125 describes the response to the command from the printer.

**Table 124— PCC-RPI command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Printer Configuration Control | Packet header |
| 1 | 0E | Subcommand: Read Printer ID | Data field |

**Table 125— PCC-RPI response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Printer Configuration Control | Packet header |
| 1 | 0E | Subcommand: Read Printer ID | Data field |
| 1 | ubyte | Length of Printer ID | |
| 1–63 | ASCII | Printer ID | |

### 5.6.14 Enable Common Printer MIB Alert

The PCC Enable Common Printer MIB Alert command (PCC-EMA), Table 126, is used by the host to enable the sending of the asynchronous message indicating that a critical alert entry has been added to the alert table as defined by RFC1759 [B6].

**Table 126— PCC-EMA command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Printer Configuration Control | Packet header |
| 1 | 0C | Enable Common Printer MIB Alert | Data field |

### 5.6.15 Disable Common Printer MIB Alert

The PCC Disable Common Printer MIB Alert command (PCC-DMA), Table 127, is used by the host to disable the sending of the asynchronous message indicating that a critical alert entry has been added to the alert table.

**Table 127— PCC-DMA command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Printer Configuration Control | Packet header |
| 1 | 0D | Disable Common Printer MIB Alert | Data field |

## 5.7 Request Logical Unit Characteristics (RLUC)

### 5.7.1 Overview

The RLUC command provides a means for a host application to obtain information about the different logical units in the printer. A given logical unit within the printer may provide support for a PDL interpreter (only type currently defined), scanner, etc. The fact that it is possible to have different characteristics for logical units necessitates this command for future growth.

The RLUC command is contained in the packet header and the subcommand, which selects the specific logical unit characteristic information to be retrieved from the printer, is contained in the data field. The specific RLUC subcommands available to the host are identified in Table 128.

**Table 128— RLUC categories**

| Subcommand category | Description |
|---|---|
| Summary Information | This command is sent to the printer by the host to request a summary of the functional support provided by a specific logical unit. |

The values for the subcommands for host requests of logical unit characteristic information are shown in Table 129.

**Table 129— RLUC subcommands**

| Value (hex) | Description | Notes |
|---|---|---|
| 00 | Summary Information | Required |
| 01 to FF | Reserved | |

### 5.7.2 Summary Information

The RLUC Summary Information command (RLUC-SI) is used by a host to request the capabilities of a given logical unit within the printer. Table 130 shows the format of the RLUC-SI command.

**Table 130— RLUC-SI command**

| Bytes | Value | Description | Notes |
|---|---|---|---|
| 5 | | Request Logical Unit Characteristics | Packet header |
| 1 | 00 | Subcommand: Summary Information | Data field |
| 1 | ubyte | Logical Unit Number (Interpreter)<br>00 Return data for the current interpreter assigned to LU #0<br>xx Specific LU # requested<br>FF Return data for all LU #'s | |

This host message is meaningful only if the reply required flag (bit 4 of the flag byte) is set. With the reply required flag set, the printer shall respond to the RLUC-SI command with a message like the one depicted in Table 131 provided that the error flag (bit 7 of the flag byte) is not set in the response the host receives from the printer.

**Table 131— RLUC-SI response**

| Bytes | Value | Description | Notes |
|---|---|---|---|
| 5 | | Request Logical Unit Characteristics | Packet header |
| 1 | 00 | Subcommand: Summary Information | Data field |
| 1 | ubyte | Number of Logical Unit Numbers (Interpreter) for which data is returned in this message | |
| 1 | ubyte | #1 Logical Unit Number | Logical Unit #1 |
| 2 | uword | #1 Logical Unit Identifier<br>0000 Page Description or Page Control Interpreter<br>0001 Scanner<br>0002 Modem<br>0003 FAX<br>0001–7FFF Reserved<br>8000–FFFF Manufacturer Specific | |
| ... | ... | ... | Logical Unit n |

If the error flag is set, a data error (bits 2 and 3 of the flag byte) indicates that an undefined logical unit number was selected. No summary information shall be returned.

#### 5.7.2.1 Number of Logical Unit Numbers

This unsigned byte indicates the number of logical unit numbers for which summary information is returned within this message. If summary information was requested for a specific interpreter, this number shall be 1.

#### 5.7.2.2 Logical Unit Number

This unsigned byte indicates the number that the printer shall use to reference this logical unit.

#### 5.7.2.3 Logical Unit Identifier

This unsigned word defines the Logical Unit type. This field is a value encoded field that defines the various types of Logical Units.

## 5.8 Printer Variable Commands (PVC)

### 5.8.1 Overview

The PVC commands are provided to allow the host to manipulate internal printer variables over any link regardless of the underlying transport protocol. These commands allow access to objects that would be accessible by SNMP if an agent was present.

Table 132 lists the various subcommands available to the host for manipulating these variables. Implementation of this command is not required for compliance to this standard.

**Table 132— PVC subcommands**

| Subcommand | Description |
|---|---|
| Get Printer Variable | Get a variable from the printer |
| Set Printer Variable | Set a variable in the printer |
| Get Next Printer Variable | Get the next variable from the printer |

### 5.8.2 Get Printer Variable

The PVC Get Printer Variable subcommand (PVC-GPV), described in Table 133, shall be used by a host to get a variable from the printer.

**Table 133— PVC-GPV command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | PVC Command | Packet header |
| 1 | 00 | Get Printer Variable | Data field |
| 2 | uword | Length of the variable name | |
| n | ASCII | Variable name | |

If the printer successfully executes the requested command, a response shall be returned to the host similar to the one shown in Table 134.

**Table 134— PVC-GPV successful response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | PVC Command | Packet header |
| 1 | 00 | Get Printer Variable | Data field |
| 1 | 00 | Request Successfully Completed | |
| 2 | uword | Length of the variable name | |
| n | ASCII | Variable name | |
| 2 | uword | Length of the variable value | |
| 1 | ubyte | Variable type<br>00 = Integer<br>01 = String | |
| n | | Variable value | |

#### 5.8.2.1 Variable Name

The variable name is an ASCII string encoding an object ID as defined in any RFC that the printer may choose to implement. For example: RFC1213 [B5] defines the MIB-II variable sysDescr. The variable name for this object, as defined in RFC1213 , is 1.3.6.1.2.1.1.1.0.

If the printer fails to execute the requested command a response shall be returned to the host similar to the one shown in Table 135.

**Table 135— PVC-GPV failed response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | PVC Command | Packet header |
| 1 | 00 | Get Printer Variable | Data field |
| 1 | ubyte | Request Failed<br>01 = Too big<br>02 = No such name<br>05 = General Error | |

### 5.8.3 Set Printer Variable

The PVC Set Printer Variable subcommand (PVC-SPV), described in Table 136, shall be used by a host to set a variable in the printer.

**Table 136— PVC-SPV command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | PVC Command | Packet header |
| 1 | 01 | Set Printer Variable | Data field |
| 2 | uword | Length of the variable name | |
| n | ASCII | Variable name | |
| 2 | uword | Length of the variable value | |
| n | | Variable value | |

If the printer successfully executes the requested command, a response shall be returned to the host similar to the one shown in Table 137.

**Table 137— PVC-SPV successful response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | PVC Command | Packet header |
| 1 | 01 | Set Printer Variable | Data field |
| 1 | 00 | Request Successfully Completed | |
| 2 | uword | Length of the variable name | |
| n | ASCII | Variable name | |
| 2 | uword | Length of the variable value | |
| 1 | ubyte | Type of variable<br>00 = Integer<br>01 = String | |
| n | | Variable value | |

If the printer fails to execute the requested command, a response shall be returned to the host similar to the one shown in Table 138.

**Table 138— PVC-SPV failed response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | PVC Command | Packet header |
| 1 | 01 | Set Printer Variable | Data field |
| 1 | ubyte | Request Failed<br>01 = Too big<br>02 = No such name<br>03 = Bad value<br>04 = Read only<br>05 = General Error | |

### 5.8.4 Get Next Printer Variable

The PVC Get Next Printer Variable subcommand (PVC-GNV), described in Table 139, shall be used by a host to get the next variable following the seed variable name in the command. Using this command, the host can retrieve all the printer variables supported within a printer without knowing which specific variables a given device supports.

**Table 139— PVC-GNV command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | PVC Command | Packet header |
| 1 | 02 | Get Next Printer Variable | Data field |
| 2 | uword | Length of the seed variable name | |
| n | ASCII | Seed variable name | |

If the printer successfully executes the requested command, a response shall be returned to the host similar to the one shown in Table 140.

**Table 140— PVC-GNV successful response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | PVC Command | Packet header |
| 1 | 02 | Get Next Printer Variable | Data field |
| 1 | 00 | Request Successfully Completed | |
| 2 | uword | Length of the next variable name | |
| n | ASCII | Next variable name | |
| 2 | uword | Length of the next variable value | |
| 1 | ubyte | Next variable type<br>00 = Integer<br>01 = String | |
| n | | Next variable value | |

#### 5.8.4.1 Seed variable name

The seed variable name is an arbitrary variable name that the printer shall use to determine the next variable to return.

If the printer fails to execute the requested command, a response shall be returned to the host similar to the one shown in Table 141.

**Table 141— PVC-GNV failed response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | PVC Command | Packet header |
| 1 | 02 | Get Next Printer Variable | Data field |
| 1 | ubyte | Request Failed<br>01 = Too big<br>02 = No such name<br>05 = General Error | |

## 5.9 Remote Operator Panel (ROP)

### 5.9.1 Overview

The ROP command not only provides the host system with a method of controlling the printer's operator panel but also the capability of obtaining information concerning the capabilities and physical characteristics of the operator panel that would allow the host to accurately depict the operator panel in a graphical user interface.

The ROP command provides several subcommands to access the different functions. Table 142 lists the various subcommands.

**Table 142— Remote Operator Panel subcommands**

| Subcommand | Description |
|---|---|
| Request Summary | Provides an overview of the operator panel and its capabilities |
| Object Characteristics | Describes the detailed characteristics of the objects on the operator panel |
| Read Object State/Contents | Read state or contents of any object on the operator panel |
| Select Alerts | Ability to be notified any time the state of an object changes |
| Operator Panel Alert | Alerts sent to host when an operator panel object changes |
| Read Alerts Selected | Ability to read what operator panel alert conditions have been selected |
| Acquire Panel Control | Request control of operator panel |
| Relinquish Panel Control | Terminate control of operator panel |
| Object Control | Ability to control any object on the operator panel |
| Read Operator Panel Variables | Read the variables that are set by the operator panel |
| Set Operator Panel Variables | Set the variables that are set by the operator panel |
| Set Operator Panel Password | Set a password that controls whether operator panel variables can be changed |
| Get Operator Panel Color Palette | Read the color palette that is currently in use |

### 5.9.2 Request Summary

The ROP Request Summary subcommand (ROP-RS), described in Table 143, provides the host system an overview of the capabilities of the operator panel.

**Table 143— ROP-RS command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Remote Operator Panel | Packet header |
| 1 | 00 | Request Summary | Data field |

After the printer receives the command, it shall return a summary data packet back to the host as depicted in Table 144.

**Table 144— ROP-RS response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Remote Operator Panel | Packet header |
| 1 | 00 | Request Summary | Data field |
| 1 | ubyte | Length Units Definition | |
| 4 | udword | Maximum Horizontal Coordinate Point | |
| 4 | udword | Maximum Vertical Coordinate Point | |
| 2 | uword | Number of Objects | |
| 2 | uword | Object Type | Object #1 |
| 2 | uword | Object ID | |
| 1 | ubyte | Object Shape ID | |
| 1 | ubyte | Number of bytes for Object Coordinates (always multiple of 4) | |
| n | udword | Object Coordinates | |
| 1 | ubyte | Length of Object Text (not including this byte) | |
| n | ASCII | Object Text String | |
| ... | ... | ... | Object n |

#### 5.9.2.1 Length Units Definition

This is an unsigned byte that defines the units, described in Table 145, used by the printer when reporting values where a length is involved within the definition.

**Table 145— ROP length units**

| Value | Description |
|---|---|
| 01 | 0.001 in |
| 02 | 0.01 in |
| 03 | 0.1 in |
| 04 | Inches |
| 05 | 0.01 mm |
| 06 | 0.1 mm |
| 07 | Millimeters |
| 08 | Centimeters |

#### 5.9.2.2 Maximum horizontal coordinate point

This unsigned double word is the maximum coordinate point in the horizontal direction of the operator panel.

#### 5.9.2.3 Maximum vertical coordinate point

This unsigned double word is the maximum coordinate point in the vertical direction of the operator panel.

#### 5.9.2.4 Number of objects

This unsigned word defines the number of operator panel objects returned in this message. Each operator panel object consists of an Object Type, an Object ID, an Object Shape, the number of coordinates needed for the Object Shape, the coordinates themselves, an Object Text String Length, and the Object Text String.

#### 5.9.2.5 Object type

This field is an unsigned word that defines the type of object being described. Table 146 contains the definitions of this word.

**Table 146— Object type**

| Value | Description |
|---|---|
| 01 | Character Display |
| 02 | Button |
| 03 | LED |
| 04 | Text Label |
| 05to DF | Reserved |
| E0 to FE | Reserved for Manufacturer Unique Object Types |
| FF | Unknown |

#### 5.9.2.6 Object ID

This field is an unsigned word that assigns an identification number to the operator panel object being described.

#### 5.9.2.7 Object shapes and coordinates

The object shape and coordinates are described in Table 147. The logical origin (0,0) is defined to be at the upper-left corner of the operator panel.

**Table 147— ROP object shapes**

| Shape ID | Name | Number of coordinates | Number of coordinate bytes | Coordinate 1 | Coordinate 2 | Coordinate 3 | Coordinate 4 |
|---|---|---|---|---|---|---|---|
| 01 | Circle | 3 | 0x'0C' | X coordinate of center of circle | Y coordinate of center of circle | Radius of circle | |
| 02 | Rectangle | 4 | 0x'10' | X coordinate of upper-left corner | Y coordinate of upper-left corner | X coordinate of lower-right corner | Y coordinate of lower-right corner |
| 03 | Line | 4 | 0x'10' | X coordinate of first endpoint | Y coordinate of first endpoint | X coordinate of second endpoint | Y coordinate of second endpoint |

#### 5.9.2.8 Length of object text

This unsigned byte defines the length in bytes of the following ROP Object Text String.

#### 5.9.2.9 Object Text String

This field is an ASCII text string that may optionally (ROP Object Text String Length not equal to zero) be returned for any object to provide an additional description of that object.

### 5.9.3 Request Object Characteristics

The ROP Request Object Characteristics subcommand (ROP-ROC), described in Table 148, provides the host system with detailed information for each object that makes up the operator panel.

**Table 148— ROP-ROC command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Remote Operator Panel | Packet header |
| 1 | 01 | Request Object Characteristics | Data field |
| 2 | uword | xxxx Specific Object Type<br>FF All Object Types | |
| 2 | uword | xxxx Specific Object ID<br>FF All Object IDs | |

After the printer receives the command, it shall return a data packet back to the host as shown in Table 149.

**Table 149— ROP-ROC response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Remote Operator Panel | Packet header |
| 1 | 01 | Request Object Characteristics | Data field |
| 2 | uword | Number of Object Characteristics Returned | |
| 2 | uword | Object Type | Object Characteristic #1 |
| 2 | uword | Object ID | |
| 2 | uword | Length of Object Characteristics | |
| n | variable | Object Characteristics (as defined for each Object Type) | |
| ... | ... | ... | Object Characteristic n |

#### 5.9.3.1 Object type and object characteristics

The object characteristics for each object type are described in Table 150.

**Table 150— Object types and object characteristics**

| Object type | Object type description | Bytes for object characteristics | 1st object characteristic | 2nd object characteristic | 3rd object characteristic | 4th object characteristic |
|---|---|---|---|---|---|---|
| 01 | Character Display | 6 | Foreground Color, Palette Index (1 byte) | Background Color, Palette Index (1 Byte) | Number of lines (2 bytes) | Number of characters per line (2 bytes) |
| 03 | LED | 1 | LED Color, Palette Index (1 byte) | | | |

The color palette can be retrieved from the printer using the ROP-GCP command.

### 5.9.4 Read Object

The ROP Read Object subcommand (ROP-RO), described in Table 151, provides the host system with the current state or contents of each object that makes up the operator panel.

**Table 151— ROP-RO command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Remote Operator Panel | Packet header |
| 1 | 02 | Read Object | Data field |
| 2 | uword | xxxx Specific Object Type<br>FFFF All Object Types | |
| 2 | uword | xxxx Specific Object ID<br>FFFF All Object Ids | |

After the printer receives the command, it shall return a data packet with information related to the requested object(s) to the host as shown in Table 152.

**Table 152— ROP-RO response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Remote Operator Panel | Packet header |
| 1 | 02 | Read Object | Data field |
| 2 | uword | Number of Object Contents Returned | |
| 2 | uword | Object Type | Object #1 |
| 2 | uword | Object ID | |
| 2 | uword | Length of Object State or Contents | |
| n | variable | Object State or Contents | |
| ... | ... | ... | Object n |

#### 5.9.4.1 Object state or contents

The object state or contents that are returned for each object type are described in Table 153.

**Table 153— Object state or contents**

| Object type | Object type description | Bytes for object contents | Object contents |
|---|---|---|---|
| 01 | Character Display | n | ASCII text string whose length is number of lines times number of characters per line |
| 02 | Button | 1 | 01—if button is depressed<br>02—if button is not depressed |
| 03 | LED | 1 | 01—if LED is on<br>02—if LED is off<br>03—if LED is blinking |
| 04 | Text Label | n | ASCII text string containing text on label |

### 5.9.5 Select Alerts

The ROP Select Alerts subcommand (ROP-SA), described in Table 154, provides the host system with the ability to be notified when the state or contents of an operator panel object changes.

**Table 154— ROP-SA command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Remote Operator Panel | Packet header |
| 1 | 03 | Select Alerts | Data field |
| 2 | uword | xxxx Specific Object Type<br>FFFF All Object Types | |
| 2 | uword | xxxx Specific Object ID<br>FFFF All Object Ids | |

### 5.9.6 Read Alerts Selected

The ROP Read Alerts Selected subcommand (ROP-RAS), described in Table 155, allows the host system to read the alerts that have been enabled for the operator panel objects.

**Table 155— ROP-RAS command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Remote Operator Panel | Packet header |
| 1 | 04 | Read Alerts Selected | Data field |

After the printer receives the command, it shall return a data packet back to the host as shown in Table 156.

**Table 156— ROP-RAS response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Remote Operator Panel | Packet header |
| 1 | 04 | Read Alerts Selected | Data field |
| 2 | uword | Number of Object Contents Returned | |
| 2 | uword | Object Type | Object #1 |
| 2 | uword | Object ID | |
| ... | ... | ... | Object n |

### 5.9.7 Acquire Panel Control

The ROP Acquire Panel Control subcommand (ROP-APC), described in Table 157, provides the host system with the means of requesting control of the operator panel.

**Table 157— ROP-APC command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Remote Operator Panel | Packet header |
| 1 | 05 | Acquire Panel Control | Data field |
| 1 | ubyte | Operator Panel Password length | |
| n | ASCIIZ | Operator Panel Password | |

#### 5.9.7.1 Manipulating operator panel objects

For a host to manipulate objects on the operator panel, it must first acquire control using the ROP-APC command. A host control ID will be returned in the printer's response that the host shall use in subsequent ROP-CPO commands manipulating operator panel objects. When operator panel processing is complete, the host shall relinquish control using the ROP-RPC command.

After the printer receives the command, it shall return a data packet back to the host, as shown in Table 158.

**Table 158— ROP-APC response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Remote Operator Panel | Packet header |
| 1 | 05 | Acquire Panel Control | Data field |
| 1 | ubyte | Host Control ID | |

#### 5.9.7.2 Operator panel password length

This unsigned byte identifies the length of the password string to follow. A zero value indicates no password.

#### 5.9.7.3 Operator panel password

This string is the password that, if set, shall be supplied in order for the printer to accept this command.

#### 5.9.7.4 Host Control ID

This unsigned byte is a number assigned by the requested device. This number shall accompany a request by a host to control the operator panel. If the returned ID is 255, then another host currently has control of the operator panel.

### 5.9.8 Relinquish Panel Control

The ROP Relinquish Panel Control subcommand (ROP-RPC), described in Table 159, provides the host system with the means of relinquishing control of the operator panel when it has completed.

**Table 159— ROP-RPC command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Remote Operator Panel | Packet header |
| 1 | 06 | Relinquish Operator Panel Control. | Data field |
| 1 | ubyte | Host Control ID<br>FFForce Relinquish (regardless of who owns control) | |

After the printer receives the command and a reply is requested, it shall return a data packet back to the host, described in Table 160.

**Table 160— ROP-RPC response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Remote Operator Panel | Packet header |
| 1 | 06 | Relinquish Operator Panel Control | Data field |
| 1 | ubyte | Host Control ID | |

### 5.9.9 Control Panel Object

The ROP Control Panel Object subcommand (ROP-CPO), described in Table 161, provides the host system with the ability of controlling any object that makes up an operator panel. In order to successfully control an object, a Host Control ID shall be assigned.

**Table 161— ROP-CPO command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Remote Operator Panel | Packet header |
| 1 | 07 | Control Panel Object | Data field |
| 1 | ubyte | Host Control ID | |
| 2 | uword | Number of Objects to be controlled | |
| 2 | uword | Object Type | Object #1 |
| 2 | uword | Object ID | |
| 2 | uword | Length of Object State or Contents Manipulations | |
| n | ubytes | Object State or Contents Manipulations | |
| ... | ... | ... | Object n |

After the printer receives the command, it shall return a data packet back to the host as shown in Table 162.

**Table 162— ROP-CPO response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Remote Operator Panel | Packet header |
| 1 | 07 | Control Panel Object | Data field |
| 1 | ubyte | Host Control ID | |

#### 5.9.9.1 Object state or contents manipulations

The methods for manipulating an object state or contents for each Object Type are described in Table 163.

**Table 163— Object state or contents manipulations**

| Object type | Object type description | Bytes for object contents | Object contents |
|---|---|---|---|
| 01 | Character Display | n | ASCII text string whose length is number of lines times number of characters per line |
| 02 | Button | 1 | 01—if button is to be depressed<br>02—if button is to be released |
| 02 | Button | 6 | 03:NN:DDDD:RRRR if button is to be depressed and released one or more times<br>NN number of times pressed and released<br>DDDD length of time depressed (ms)<br>RRRR length of time released (ms)—ignored if NN=1 |
| 03 | LED | 4 | First 2 bytes contain ON time in milliseconds (value of zero denotes that LED is off). Second 2 bytes contain OFF time in milliseconds (value of zero denotes that LED is on). |

### 5.9.10 Read Panel Variables

The ROP Read Panel Variables subcommand (ROP-RPV), described in Table 164, provides the host system with the ability of read any variable that can be set through the operator panel.

**Table 164— ROP-RPV command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Remote Operator Panel | Packet header |
| 1 | 08 | Read Panel Variables | Data field |
| 1 | ubyte | Variable Type | |
| 4 | dword | Variable ID<br>FFFFFFFF—return all variables | |

After the printer receives the ROP-RPV command, it shall return a response dependent on the variable type as shown in Table 165, Table 166, and Table 167.

**Table 165— ROP-RPV response variable type = 1**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Remote Operator Panel | Packet header |
| 1 | 08 | Read Panel Variables | Data field |
| 1 | 01 | Variable Type | |
| 2 | uword | Number of Variables returned | |
| 4 | udword | Variable ID | Data associated with variable #1 |
| 1 | ubyte | Length of Variable Description in bytes (not including this byte) | |
| n | ASCII | Variable Description | |
| 1 | ubyte | Type of Variable<br>00 signed integer<br>01 string<br>02 unsigned integer | |
| 1 | ubyte | Length of Variable Value in bytes (not including this byte) | |
| n | ... | Variable Value | |
| ... | ... | ... | Variable n |

**Table 166— ROP-RPV response variable type = 2**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Remote Operator Panel | Packet header |
| 1 | 08 | Read Panel Variables | Data field |
| 1 | 02 | Variable Type | |
| 2 | uword | Number of Variables returned | |
| 4 | udword | Variable ID | Data associated with variable #1 |
| 1 | ubyte | Type of Variable<br>00 signed integer<br>01 string<br>02 unsigned integer | |
| 1 | ubyte | Length of Variable Value in bytes (not including this byte) | |
| n | ... | Variable Value | |
| ... | ... | ... | Variable n |

**Table 167— RPV variable data type = 03**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Remote Operator Panel | Packet header |
| 1 | 08 | Read Panel Variables | Data field |
| 1 | ubyte | Variable Type | |
| 2 | uword | Number of Variables returned | |
| 4 | udword | Variable ID | Data associated with variable #1 |
| 1 | ubyte | Length of Variable Value in bytes (not including this byte) | |
| n | ... | Variable Value | |
| ... | ... | ... | Variable n |

### 5.9.11 Set Panel Variables

The ROP Set Panel Variables subcommand (ROP-SPV), described in Table 168, provides the host system with the ability to set any variable that can be set through the operator panel. If the host wants to ensure exclusive access to a variable it is recommended that it change the password using the ROP-SPP command, make the change(s) using the ROP-SPP command, and finally change the password back to the original value.

**Table 168— ROP-SPV command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Remote Operator Panel | Packet header |
| 1 | 09 | Set Panel Variables | Data field |
| 1 | ubyte | Operator Panel Password length | |
| n | ASCIIZ | Operator Panel Password | |
| 2 | uword | Number of Variables to be Set | |
| 4 | udword | Variable ID | Variable #1 |
| 1 | ubyte | Length of Variable Value in bytes (not including this byte) | |
| n | ... | Variable Value | |
| ... | ... | ... | Variable n |

If no password is set (operator panel password length equal zero), the printer shall process this command without regard to the content of the password field.

After the printer receives the ROP-SPV command, it shall return a response as shown in Table 169.

**Table 169— ROP-SPV response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Remote Operator Panel | Packet header |
| 1 | 09 | Set Panel Variables | Data field |

### 5.9.12 Set Panel Password

The ROP Set Panel Password subcommand (ROP-SPP), described in Table 170, provides the host system with the ability to set/reset the operator panel password.

**Table 170— ROP-SPP command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Remote Operator Panel | Packet header |
| 1 | 0A | Set Panel Password | Data field |
| 1 | ubyte | Current Operator Panel Password length | |
| n | ASCIIZ | Current Operator Panel Password | |
| 1 | ubyte | New Operator Panel Password length | |
| n | ASCIIZ | New Operator Panel Password | |

In its factory-default state, the printer shall not have a password set (operator panel password length equal to zero).

After the printer receives the ROP-SPP command, it shall return a response as shown in Table 171.

**Table 171— ROP-SPP response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Remote Operator Panel | Packet header |
| 1 | 0A | Set Panel Password | Data field |

A zero length indicates no password. A positive acknowledgment indicates successful completion of the operation. A negative acknowledgment (a response with flag bits indicating a data error) indicates that the specified current operator panel password was incorrect.

### 5.9.13 Get Color Palette

The ROP Get Color Palette subcommand (ROP-GCP), described in Table 172, provides the host system with the ability to read the contents of the current operator panel color palette.

**Table 172— ROP-GCP command**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Remote Operator Panel | Packet header |
| 1 | 0B | Get Color Palette | Data field |

After the printer receives the ROP-GCP command, it shall return a response as shown in Table 173.

**Table 173— ROP-GCP response**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Remote Operator Panel | Packet header |
| 1 | 0B | Get Color Palette | Data field |
| 1 | ubyte (I) | Number of palette entries | |
| 1 | ubyte | 1st color red intensity | |
| 1 | ubyte | 1st color green intensity | |
| 1 | ubyte | 1st color blue intensity | |
| ... | ... | ... | |
| 1 | ubyte | $i^{th}$ color red intensity | |
| 1 | ubyte | $i^{th}$ color green intensity | |
| 1 | ubyte | $i^{th}$ color blue intensity | |

A negative acknowledgment (a response with flag bits indicating a command error) indicates that the printer does not support this command.

# 6. Alerts

## 6.1 Device Status Alert (DSA)

### 6.1.1 Overview

A DSA is an asynchronous unsolicited message that can be sent from the printer to the host at any time to convey a change in printer status. The DSA is contained in the data field of a standard IEEE 1284.1 message. The printer can alert the host to status changes in several categories.

The status categories are shown in Table 174.

**Table 174— DSA categories**

| Alert category | Description |
|---|---|
| Input | Those conditions in the printer where problems with the input of printing media can cause the printer to stop. These conditions can normally be remedied by a normal operator. An example of this would be when a particular input tray is empty. |
| Output | Those conditions in the printer where problems with the output of printing media can cause the printer to stop. These conditions can be normally be remedied by a normal operator. An example of this would be when a particular output tray is full. |
| Jam | Those conditions in the printer where normal conditions associated with the movement of printing media through the printer can cause it to stop. These conditions can be corrected by a normal operator. An example of this would be "Jam at Input Tray." |
| Printing Supplies | Those conditions in the printer where normal conditions associated with printing supplies can cause it to stop. These conditions can be corrected by a normal operator. An example of this would be " Out of Ink." |
| Operator Intervention Required | Those conditions in the printer, other than input, output, jam, or printing supplies, that can cause the printer to stop. These conditions can be corrected by a normal operator. An example of this would be "Cover Open." |
| Warnings | Those conditions that are not critical to the immediate operation of the printer: Examples are<br>1—Paper Input Low<br>2—Toner Low<br>3—Output Bin Nearly Full |
| Device Service Required | Those conditions that are critical to printer operation that a normal operator cannot correct and requires technical service. Examples are<br>1—Laser Polygon Scanner inoperative<br>2—High Fuser Temperature |
| Configuration Change | Those conditions that are detected that change the current configuration of the printer. Examples are<br>1—Memory Board Failure resulting in reduced memory<br>2—Font Cartridge/Card has been removed |

### 6.1.2 Alert message

The format of the DSA message sent to the host is shown in Table 175.

**Table 175— DSA message**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Device Status Alert | Packet header |
| 1 | Bit-Encoded byte | Overall Printer Status Summary—Standard<br>Bit 0—Power On Initialization<br>Bits 1–5—Reserved, must be zero<br>Bit 6—Printer off-linezero=No, one=Yes<br>Bit 7—Printer Busy—Unable to Accept Datazero=No, one=Yes | Data field |
| 1 | Bit-Encoded byte | Device Alerts #1<br>Bit 0—Printing Supply<br>Bit 1—Paper Jam<br>Bit 2—Output<br>Bit 3—Input<br>Bits 4–7—Reserved, must be zero | |
| 1 | Bit-Encoded byte | Device Alerts #2<br>Bit 0—Configuration Change<br>Bit 1—Warnings<br>Bit 2—Device Service Required<br>Bit 3—Operator Intervention Required<br>Bits 4–7—Reserved, must be zero | |
| n | | Alert Detail | |

While RDS commands may be used at any time to extract the current printer status, the asynchronous DSA is used to alert the host to an immediate change.

A condition that causes a DSA is “edge triggered”; that is, a DSA is sent the first time the condition occurs. If the condition continues to persist, it is reported in status but does not cause additional DSA messages to be sent. For any alert condition that is not cleared by a RDS command, a DSA shall also be generated when the condition is cleared. The DSA that is generated when a condition is cleared shall have the bit set to zero for the category in which the alert condition existed unless other alert conditions are active in that category.

If a condition that would cause a DSA is active at the time that its DSA is armed using the PCC-SDSA command, then a DSA shall be sent.

The reset condition is that all conditions that cause DSA messages are disabled until explicitly enabled by the host except for the Power On Initialization Alert. The DSA with the Power On Initialization bit set shall automatically be generated when the printer has completed a power on initialization sequence or any other initialization similar to power on. The Power On Initialization bit in the DSA shall be cleared when a RDS-RSS response is returned to the host.

#### 6.1.3 Detail

##### 6.1.3.1 Overall printer status summary

This field in the DSA message has the same meaning as defined in the RDS-RSS command.

##### 6.1.3.2 Device Alerts #1

This field in the DSA message has the same meaning as defined in the RDS-RSS command.

##### 6.1.3.3 Device Alerts #2

This field in the DSA message has the same meaning as defined in the RDS-RSS command.

##### 6.1.3.4 Alert detail

If this DSA message was caused by a change in the overall printer status summary, there shall be no alert detail. Otherwise, the format and content of the alert detail field in this DSA message is dependent on the error or warning that caused the alert. The alert detail shall consist of the data field of the RDS response that would be returned for the device alert that caused this alert message. For example, if a configuration change is detected, the printer shall construct an alert message with bit-0 of device alert #2 set and alert detail consisting of the data field of the RDS-RCCA response.

### 6.2 Interpreter Message Alerts (IMA)

Interpreter Messages (IM) are asynchronous unsolicited messages that can be sent from the printer to the host at any time to convey a message from an interpreter within the printer. The specific IM is contained in the data field of an unsolicited IEEE 1284.1 message from an interpreter. All messages from an interpreter shall be preceded by an Interpreter Message Alert packet that describes the type of interpreter message that follows in a subsequent message.

The different types of interpreter messages that the host shall allow the printer to send to it can be selected using the PCC-SUIM command.

The categories of interpreter message alerts are shown in Table 176.

 

**Table 176— IMA categories**

| Message category | Description |
|---|---|
| Response to an Interpreter Query | Included are those conditions where an interpreter would respond to a data stream command. For example, a hex-'14' in a PostScript data stream would elicit a response of this type from a PostScript interpreter. |

The format of the IMA message sent to the host is shown in Table 177.

**Table 177— IMA message**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Interpreter Message Alert | Packet header |
| 1 | ubyte | Logical Unit Number<br>00 Currently Assigned Default Interpreter<br>XX Interpreter ID<br>FF Unknown | Data field |
| 2 | uword | Printer-Assigned Job ID<br>XXXX Job ID<br>FFFF Unknown | |
| 2 | uword | Page Number of printer-assigned Job ID<br>XXXX Page Number<br>FFFF Unknown | |
| 1 | ubyte | Interpreter Flag<br>00 Informational only, job not affected<br>01 Warning, job may not appear completely as specified by data stream<br>02 Error, job shall not print<br>03–FE Reserved<br>FF Unknown | |
| 1 | Bit-Encoded byte | Interpreter Status Alerts #1<br>Bit 0—Response to an Interpreter Query<br>Bits 1–7—Reserved, must be zero | |
| 1 | Bit-Encoded byte | Interpreter Status Alerts #2<br>Bits 0–7—Reserved, must be zero | |

### 6.2.1 Logical unit number

This unsigned byte indicates the LU # (interpreter) from which the IM was received.

### 6.2.2 Printer-Assigned Job ID

This unsigned word indicates the printer-assigned Job ID that the interpreter was processing at the time the IM was generated.

### 6.2.3 Page number of printer-assigned job ID

This unsigned word indicates the page number of the reference printer job ID that the interpreter was processing at the time the IM was generated.

### 6.2.4 Interpreter flag

This unsigned byte indicates the status of the interpreter following the IMA that has been sent.

### 6.2.5 Interpreter Status Alerts #n

This bit-encoded byte indicates the type of IM that is being passed back to the host.

If a condition that causes an IMA occurs within the interpreter and the printer has not been conditioned to return this particular type of interpreter message to the host, the printer shall discard the IM.

The reset condition is that all conditions that cause IM messages are disabled until explicitly enabled by the host using the PCC-SIM command.

## 6.3 Job Control Alerts (JCA)

A JCA is an asynchronous unsolicited message that can be sent from the printer to the host at any time to convey job control status within the printer. The JCA is contained in the data field of an unsolicited IEEE 1284.1 message. The points in the print job flow where JCAs shall happen are specified in the JC-SJ command or in the PCC-SJA command.

The host can easily identify an asynchronous JCA sent by the printer by examining the command byte of the IEEE 1284.1 packet for the hex value of FD.

### 6.3.1 JCA Logical Unit Start (LUS)

The format of the JCA-LUS alert sent to the host is shown in Table 178.

**Table 178— JCA-LUS alert**

| Bytes | Value | Description | Notes |
|---|---|---|---|
| 5 | | Job Control Status Alert | Packet header |
| 2 | 0200 | Logical Unit Start (LUS) | Data field |
| 2 | uword (i) | Number of Logical Unit Start Alerts | |
| 1 | ubyte | #1 Logical Unit Number (Interpreter) | Logical Unit Start #1 |
| 2 | uword | #1 Printer Assigned Job ID | |
| 1 | ubyte | #1 Current Status<br>00 = Job Running<br>01 = Waiting for Data<br>02–FF = Reserved | |
| ... | ...· | ... | Logical Unit Start n |

### 6.3.2 JCA Sheet Complete (SC)

The JCA-SC alert can be used by the host to track the individual pages of a print job. This alert is generated as the page or sheet is placed in the output tray. The format of the JCA-SC alert sent to the host is shown in Table 179.

**Table 179— JCA-SC alert**

| Bytes | Value | Description | Notes |
|---|---|---|---|
| 5 | | Job Control Status Alert | Packet header |
| 2 | 4000 | Sheet Complete Alert | Data field |
| 2 | uword (n) | Number of Sheet Complete Alerts | |
| 1 | (i) | Logical Unit Number | Sheet Alert #1 |
| 2 | uword | Job ID | |
| 4 | udword | Sheet Number | |
| ... | ... | ... | Sheet Alert n |

### 6.3.3 JCA Job Complete (JC)

The format of the JCA-JC alert sent to the host is shown in Table 180.

**Table 180— JCA-JC alert**

| Bytes | Value | Description | Notes |
|---|---|---|---|
| 5 | | Job Control Status Alert | Packet header |
| 2 | 8000 | EOJ Alert | Data field |
| 2 | uword (n) | Number of End of Job Alerts | |
| 1 | (i) | Number of input sources | EOJ Alert #1 |
| 1 | ubyte | Completed job #1 logical unit number | |
| 2 | uword | Completed job #1 Job ID | |
| 2 | uword | Completed job #1 processing time (in seconds)<br>FFFF Not Supported | |
| 4 | udword | Completed job #1 sheets used from input source i<br>FFFFFFFF Not Supported | |
| 4 | udword | Completed job #1 impressions from input source i<br>FFFFFFFF Not Supported | |
| 4 | udword | Completed job #1 counter units from input source i<br>FFFFFFFF Not Supported | |
| ... | ... | ... | |
| 4 | udword | Completed job #1 sheets used from input source i<br>FFFFFFFF Not Supported | |
| 4 | udword | Completed job #1 impressions from input source i<br>FFFFFFFF Not Supported | |
| 4 | udword | Completed job #1 counter units from input source i<br>FFFFFFFF Not Supported | |
| ... | ... | ... | EOJ Alert n |

### 6.3.4 JCA Suspend Processing (SP) at Logical Unit Start (LUS)

The format of the JCA-SP(LUS) alert sent to the host is shown in Table 181.

**Table 181— JCA-SP(LUS) aler**

| Bytes | Value | Description | Notes |
|---|---|---|---|
| 5 | | Job Control Status Alert | Packet header |
| 2 | 0002 | Suspend processing at logical unit start | Data field |
| 1 | ubyte | Logical Unit Number | |
| 2 | uword | Printer Assigned Job ID | |

### 6.3.5 JCA Suspend Processing at Start of Printing (SOP)

The format of the JCA-SP(SOP) alert sent to the host is shown in Table 182.

**Table 182— JCA-SP(SOP) alert**

| Bytes | Value | Description | Notes |
|---|---|---|---|
| 5 | | Job Control Status Alert | Packet header |
| 2 | 0008 | Suspend processing at start of printing | Data field |
| 1 | ubyte | Logical Unit Number | |
| 2 | uword | Printer Assigned Job ID | |

### 6.3.6 JCA Suspend Processing at Start of Finishing (SOF)

The format of the JCA-SP(SOF) alert sent to the host is shown in Table 183.

**Table 183— JCA-SP(SOF) alert**

| Bytes | Value | Description | Notes |
|---|---|---|---|
| 5 | | Job Control Status Alert | Packet header |
| 2 | 0020 | Suspend processing at start of finishing | Data field |
| 1 | ubyte | Logical Unit Number | |
| 2 | uword | Printer Assigned Job ID | |

### 6.3.7 JCA EOJ Accounting Message

The format of the JCA-EOJ alert sent to the host is shown in Table 184.

**Table 184— JCA-EOJ Accounting message (continued)**

| Bytes | Value | Description | Notes |
|---|---|---|---|
| 5 | | Job Control Status Alert | Packet header |
| 1 | 8000 | EOJ Accounting Message | Data field |
| 2 | uword (n) | Number of End of Job Accounting Messages | |
| 1 | ubyte | Completed job #1 logical unit number | Begin Message #1 |
| 2 | uword | Completed job #1 Job ID | |
| 2 | uword | Completed job #1 processing time (in seconds)<br>FFFF Not Supported | |
| 4 | udword | Elapsed time job #1 (in seconds)<br>FFFFFFFF Unknown | |
| 4 | udword | Number of Bytes in Job Data Stream<br>FFFFFFFF Unknown | |
| 4 | | Date Job Started<br>**Byte** **Description** **Range (low, high)**<br>1–2 Year (0,65535)<br>3 Month (1,12)<br>4 Day (1,31) | |
| 7 | | Time Job Started<br>**Byte** **Description** **Range (low, high)**<br>1 Hour (0,23)<br>2 Minute (0,59)<br>3 Second (0,60) use 60 for leap second<br>4 Tenths-seconds (0,9)<br>5 Direction from UTC ['+','-']<br>6 Hours from UTC (0,12)<br>7 Minutes from UTC (0,59) | |
| 1 | (i) | Number of input sources | Begin input data |
| 1 | ubyte | Input ID (input source 1) | |
| 1 | ubyte | Medium Characteristic | |
| 1 | ubyte | Encoded Medium Size | |
| 1 | ubyte | Length of Medium Description (not including this byte) | |
| n | ASCII | Medium Description String | |
| 4 | udword | Completed job #1 sheets used from this Input ID<br>FFFFFFFF Not Supported | |
| 4 | udword | Completed job #1 impressions from this Input ID<br>FFFFFFFF Not Supported | |
| 4 | udword | Completed job #1 counter units from this Input ID<br>FFFFFFFF Not Supported | |
| ... | ... | ... | end input data |
| 1 | (i) | Number of output sources | begin output data |
| 1 | ubyte | Output ID (output source 1) | |
| 4 | udword | Completed job #1 sheets deposited in this output ID<br>FFFFFFFF Not Supported | |
| ... | ... | ... | End output data |
| 2 | uword | Job Completion code | |
| 1 | ubyte | Length of Host String | |
| 1–63 | ASCII | Host String | |
| 1 | ubyte | Length of User String | |
| 1–63 | ASCII | User String | |
| 1 | ubyte | Length of Job Name | |
| 1–63 | ASCII | Job Name | |
| 1 | ubyte | Length of Additional Information String | |
| 1–63 | ASCII | Additional Information String | |
| 1 | ubyte | Length of Printer Identification | |
| n | ASCII | Printer Identification | End Message #1 |
| ... | ... | ... | Begin Message n |

#### 6.3.7.1 Job Completion code

This is an indication of whether the job completed successfully or not. The list of job completion codes is shown in Table 185.

**Table 185— Job completion codes**

| Code | Description |
|---|---|
| 0000 | Job completed successfully |
| 0001 | Job terminated abnormally due to a time-out condition |
| 0002 to FFFD | Reserved |
| FFFE | Job terminated abnormally but reason unknown |
| FFFF | Job canceled |

## 6.4 Common Printer MIB Alert (CPMA)

The CPMA is an asynchronous unsolicited message that can be sent from the printer to the host at any time to indicate that a critical (something that would stop printing) alert entry has been added to the RFC defined alert table implemented in the printer. The CPMA is contained in the data field of an unsolicited IEEE 1284.1 message.

The host can easily identify an asynchronous CPMA sent by the printer by examining the command byte of the IEEE 1284.1 packet for the hex value of FC.

### 6.4.1 CPMA Alert

The format of the CPMA Alert sent to the host is shown in Table 186.

**Table 186— CPMA Alert message**

| Bytes | Value | Description | Notes |
|---|---|---|---|
| 5 | | Common Printer MIB/MIF Alert | Packet header |
| 1 | ubyte | prtAlertIndex Object ID length | Data field |
| n | ASCIIZ | prtAlertIndex Object ID | |
| 4 | dword | prtAlertIndex | |
| 1 | ubyte | prtAlertSeverityLevel Object ID length | |
| n | ASCIIZ | prtAlertSeverityLevel Object ID | |
| 4 | dword | prtAlertSeverityLevel | |
| 1 | ubyte | prtAlertGroup Object ID length | |
| n | ASCIIZ | prtAlertGroup Object ID | |
| 4 | dword | prtAlertGroup | |
| 1 | ubyte | prtAlertGroupIndex Object ID length | |
| n | ASCIIZ | prtAlertGroupIndex Object ID | |
| 4 | dword | prtAlertGroupIndex | |
| 1 | ubyte | prtAlertLocation Object ID length | |
| n | ASCIIZ | prtAlertLocation Object ID | |
| 4 | dword | prtAlertLocation | |
| 1 | ubyte | prtAlertCode Object ID length | |
| n | ASCIIZ | prtAlertCode Object ID | |
| 4 | dword | prtAlertCode | |

These variables are defined in RFC1759 [B6] .

## 6.5 Operator Panel Alert (OPA)

### 6.5.1 Operator Panel Alert Message

When an operator panel object changes and alerts for that object have been enabled, the following alert (Table 187) is sent to the host.

**Table 187— Operator Panel Alert**

| Bytes | Value (hex) | Description | Notes |
|---|---|---|---|
| 5 | | Operator Panel Alert | Packet header |
| 2 | uword | Number of Object Alerts | |
| 2 | uword | Object Type with Alert active<br>xxxx specific object type<br>FFFF unknown object type | Object #1 |
| 2 | uword | Object ID with Alert active<br>xxxx specific object ID<br>FFFF unknown object ID | |
| n | ubyte | Object State or Contents | |
| ... | ... | ... | Object n |

If both the Object Type and the Object ID are unknown, the host shall re-discover (refresh) the operator panel.

### 6.5.2 Object state or contents

The object state or contents that are returned for each object type are described in Table 188.

**Table 188— ROP Object State or Contents**

| Object type | Object type description | Bytes for object contents | Object contents | | |
|---|---|---|---|---|---|
| 01 | Character Display | n | ASCII text string whose length is number of lines times number of characters per line | | |
| 02 | Button | 1 | 01—if button is depressed<br>02—if button is not depressed | | |
| 03 | LED | 2 | 01—if LED is on<br>02—if LED is off | LED Color, Palette Index | |
| 03 | LED | 3 | 03—LED blinking | “On” LED Color, Palette Index | “Off” LED Color, Palette Index |
| 04 | Text Label | n | ASCII text string containing text on label | | |

# Annex A Conformance

# (Normative)

All devices conforming to this standard shall adhere to the command/reply structure as defined in Clause 4..

Printer devices shall generate all appropriate alerts.

Printer devices shall support and respond appropriately to all commands listed in this standard except the following groups of commands, which may be omitted but if supported shall be done so as a group:

a) Printer variable and related commands and alerts
   1) Get Printer Variable
   2) Set Printer Variable
   3) Get Next Printer Variable
   4) Enable Common Printer MIB Alert
   5) Disable Common Printer MIB Alert
   6) Common Printer MIB Alert
b) Printer operator panel commands and alert
   1) All Remote Operator Panel functions (see 5.9)
   2) Operator Panel Alert

   Because of its architecture, it is not possible to list which commands shall be used by a host; therefore, it is sufficient to state that in addition to conforming to Clause 4., host devices conforming to this standard shall utilize appropriate commands, parse and interpret responses from printers, and handle alerts received from printers.

# Annex B Requirements for a Standard Information Mover Protocol with Link Encapsulation (SIMPLE)

# (Informative)

## B.1 Background

IEEE Std 1284-1994 [B3] defines and describes an updated PC parallel port by adding multiple modes of operation that provide for higher speed, bidirectional communication between the PC and the printer. The standard does not, however, define what one could call a “transport” for this point-to-point link. As such, at least one layer above the physical parallel port and below the application must be defined. This (or these) layer(s) may take on the functions and characteristics of more than one of the OSI layer model specifically data link, transport, and session. Because this is a single point-to-point connection, the functions of the network layer are not needed. The presentation and application layers are assumed to be provided by the operating system. Because this requirement does not logically match the OSI model, the terminology *data link*, *transport*, and *session* will not be used; rather, we will call this simply an *information mover protocol* (IMP).

IEEE Std 1284.1-1997 is a printer status, configuration, and control protocol designed to operate over both networks and in point-to-point environments. In network environments, all the appropriate network layers are typically present, thus there is no need to develop an information mover protocol. The problem lies in the IEEE Stds 1284-1994 parallel port; nothing is defined above the physical layer.

To address the above mentioned issues, the requirements for an information mover protocol must be defined before a solution can be determined. This annex attempts to tersely state the requirements for such a protocol. The items identified as “functional requirements” are absolutes, i.e., they must be present, while the “implementation goals” are softer requirements that would be preferred to be present in the best possible implementation.

## B.2 Functional requirements

### B.2.1 In-Order delivery of data

In-Order delivery means that packets presented at the producer’s end of the channel are delivered to the consumer in exactly the same order.

### B.2.2 Flow control

Flow control is a fundamental requirement of almost all communications methods be they point-to-point, local area network, or wide area network. Because the host may not be fully aware of the buffering capability of the peripheral, there should be a method for flow control to allow the host to adjust the rate of transmission.

Flow control (sometimes referred to as pacing) provides a means by which two major functions are accomplished:

a) The size of data packets is negotiated to be acceptable to both sender and receiver, and
b) An excess of data is not delivered to the receiver.
   If more data is sent than the recipient has room to store until processed, then effective flow control has not been provided. Flow control is a required component of the definition of the information mover protocol. Flow control must be provided separately for each logical channel supported by the protocol.
   In order to ensure initial packet size compatibility, the protocol will specify the minimum buffer size to be supported (hence a maximum length of the protocol’s initial packets). This minimum will be large enough to specifically allow negotiation of a larger packet size as well as to provide the basic functions of the protocol when the protocol is started.

 

### B.2.3 Guaranteed delivery/error detection/recovery

The term “guaranteed delivery” is used to denote the fact that higher-level functions (e.g., the IEEE 1284.1 protocol) expect the underlying information mover protocol to “guarantee” that a packet arrives at its destination, error-free. This is handled by the information mover layer performing automatic error detection and retransmission on behalf of the functions above it as required. This may include retransmission of everything following the error or retransmitting only the segment in error and a method of reassembling the data in intended order. Reporting of errors to upstream layers will occur if automatic retry/correction is not provided or achieved.

Guaranteed delivery also implies that the transmitter may not discard the transmitted data until the receiver acknowledges it. Data that is not acknowledged by the receiver will be retransmitted through a defined error recovery means.

A simple method shall be available in the protocol to determine if the link is still alive. This method will not depend on transmission of data but more likely be similar to a “ping” or “echo” command.

The protocol must specify how to report and recover from protocol errors and/or the loss of connection. This must handle both error notification originated by one of the participants and/or termination due to external means.

A quality of service selection method allows the users of the information mover protocol to balance the needs of speed vs. reliability. In most implementations, faster communications is achieved while reliability is degraded. The converse is also true. The IMP assumes the presence of inherent reliability and shall provide a means to negotiate to a higher level of reliability. Inherent depends upon the underlying reliability of the IEEE 1284 or other link. A higher level of reliability provides additional means of detecting and potentially correcting errors. The IMP must define which level of quality of service is the default. The upstream applications should be able to select a quality of service level as needed. The information mover protocol should then negotiate with the downstream layers and devices to achieve the desired quality of service.

### B.2.4 Multiple logical channels

The information mover protocol must include support for multiple logical channels, i.e., support for multiplexing. The IMP’s implementation of multiplexing must provide

a) A physical channel that cannot be allowed to become “stuck” in one direction.
b) A means by which an application on the “sending” end must be able to communicate “privately” with an application on the “receiving” end. This may be accomplished through some number or other identification of the source and destination channels or entities.
c) A means by which generally separate channels can support sending data to a device (i.e., the sending of the print data stream) as well as control and status.
d) A means to request the resting or clearing of a stuck logical channel.
In addition, the IMP must ensure that should one logical channel become blocked, other logical channels are unaffected.

### B.2.5 IEEE 1284 mode independent

The protocol should operate over any mode of the IEEE 1284 parallel port.

### B.2.6 PDL, application, and operating system independent

The protocol should not depend on a specific operating system, nor should it assume anything about nor be affected by the content of the data being carried. For example, the protocol should have no page description language (PDL) nor device control protocol (e.g., IEEE 1284.1) dependencies.

## B.3 Implementation goals

Because the following are goals, the design of the protocol should not compromise the functional requirements in order to meet these goals.

### B.3.1 Link independence

The protocol should support other links in addition to IEEE 1284 (serial, etc.).

### B.3.2 Extensibility

The protocol should allow for extensibility in such a way as to not “break” previously well-written implementations. In this case, a previously well-written application properly deals with unknown fields and function codes. This places a requirement on the protocol to include field lengths and other necessary information, consistently positioned, to allow “skipping over” unknown commands.

### B.3.3 Low system overhead on both sides of link

The protocol should not place an undue burden on either end of the point-to-point link. In fact, the burden, as much as possible, should be generally equal.

With simplicity of design and efficiency of implementation, an effective design will ensure high usage of the available channel bandwidth with minimal processor overhead on either end.

### B.3.4 Backward compatibility

The protocol should be backwards compatible. Backwards compatibility includes legacy peripherals, computers, and “old versions” of the protocol.

# Annex C Internetwork Packet (IP) transport mapping example

# (Informative)

## C.1 Introduction

IEEE Std 1284.1-1997 deals with bidirectional communications between a printer and a host. However, the transport mechanism between the printer and host is not defined. This annex describes one legacy application's use of the IP protocol to encapsulate IEEE 1284.1 messages.

An IEEE Std 1284.1-1997 implementation in a networking environment presents several problems not found in locally attached host-printer configurations, such as

— One printer being accessed simultaneously by many hosts,
— One host simultaneously accessing many printers,
— A printer server connecting more than one printer to the network with a single IP address,
— To which host(s) should a printer send Device Status Alerts (DSA)?,
— How to sequence packets to ensure no data is lost, and
— Errors.

This example addresses these and other problematic areas when implementing the IEEE Std 1284.1-1997 protocol in an IP network environment.

## C.2 Commands

### C.2.1 Overview

The legacy implementation is capable of responding to commands from multiple hosts. UDP port (9300) was chosen to transport commands, as shown in Figure C.1.

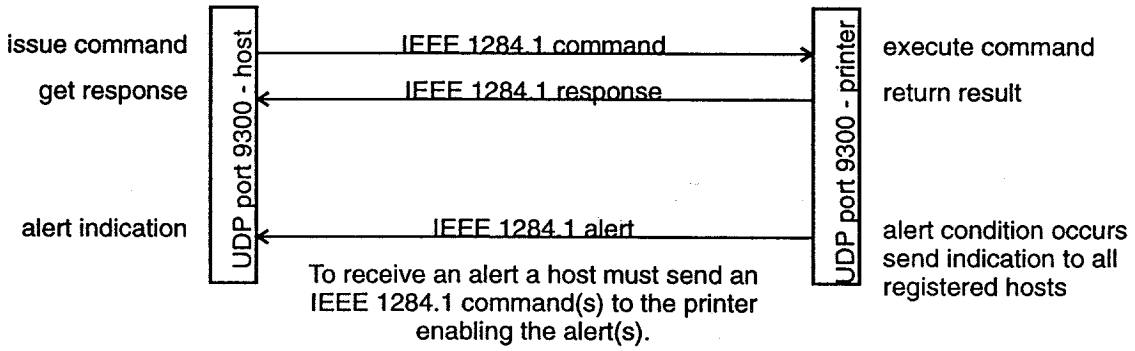


**Figure C.1 ——UDP port 9300 command, response, and alert transports**

Table C.1 shows the encapsulation that shall be used to transport commands.

**Table C.1 ——UDP port 9300 command packet encapsulation**

| Bytes | Value (hex) | Description |
|---|---|---|
| 2 | uword (n+5) | length. Number of remaining bytes in this message. |
| 1 | ubyte | deviceIndex. A single ASCII character (1–9) representative of one of multiple printers assigned to the IP address. |
| 4 | udword | idNumber. A unique number assigned by the host. |
| n | | IEEE 1284.1 message |

### C.2.2 deviceIndex

This unsigned byte is used to accommodate multiple printers at a single IP address. This allows the host to specify which printer at this IP address is to receive the command. When only one printer is located at the IP address, deviceIndex is set to a hexadecimal value of 31.

### C.2.3 idNumber

This unsigned double word is assigned at the discretion of the host. A unique number will allow the host to associate responses to prior commands as the printer shall use this value supplied by the host in the response message encapsulation.

## C.3 Responses

### C.3.1 Overview

UDP port (9300) shall be used to transport responses back to the host as shown in Figure C.1. Responses shall be encapsulated as shown in Table C.2.

**Table C.2 ——UDP port 9300 response or alert packet encapsulation**

| Bytes | Value (hex) | Description |
|---|---|---|
| 2 | uword (n+9) | length. Number of remaining bytes in this message |
| 2 | uword | ackNumber. Set to zero for a response. Set to nonzero for an alert. |
| 2 | uword | sequenceNumber. Number assigned by printer to allow for host reassembly of segmented packets. |
| 1 | ubyte | deviceIndex. A single ASCII character (1–9) representative of one of multiple printers assigned to the IP address. |
| 4 | udword | idNumber. A unique number assigned by the host. Returned by printer in any response associated with the command. |
| n | | IEEE 1284.1 alert or response message |

### C.3.2 ackNumber

This unsigned word shall be set to zero for all responses. Though UDP is an unreliable transport, nonreceipt of a response by a host is not a great concern. The remedy for a nonanswered command is for the host to reissue the command.

### C.3.3 sequenceNumber

If the printer or network interface needs to segment a response into multiple packets, the sequenceNumber, an unsigned word, shall be incremented for each packet starting with one.

## C.4 Alerts

### C.4.1 Overview

UDP port (9300) shall be used to transport alerts back to the host as shown in Figure C.1. Alerts shall be encapsulated as described in Table C.2.

### C.4.2 ackNumber

For all alerts, ackNumber shall be set to a nonzero value. Because alerts are only sent once and (due to their asynchronous nature) their existence is unknown by the host, a reliable mechanism must be provided to ensure that an alert is received by all monitors. The legacy implementation chose to use an alert acknowledgment for this mechanism. This field, when set to a nonzero value, is an indication to the host that this message is an alert and that an acknowledgment is required.

### C.4.3 idNumber

The value for this field in an alert is meaningless.

## C.5 Alert acknowledgments

### C.5.1 Overview

For all alerts, ackNumber shall be set to a nonzero value. Because alerts are only sent once and (due to their asynchronous nature) their existence is unknown by the host, a reliable mechanism must be provided to ensure that an alert is received by all monitors. The reference implementation chose to use an alert acknowledgment for this mechanism. This field, when set to a nonzero value, is an indication to the host that this message is an alert and that an acknowledgment is required.

UDP port (9301) shall be used by the host to acknowledge receipt of alerts from a printer as show in Figure C.2.

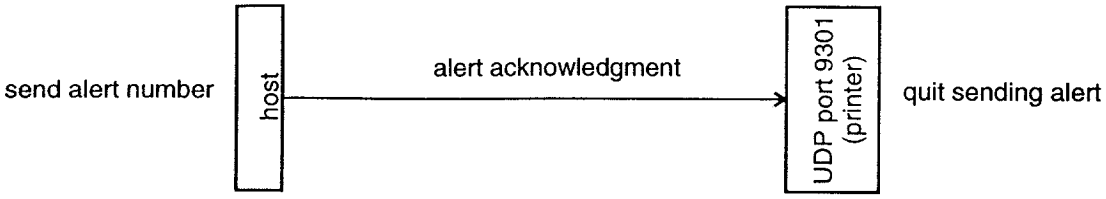


**Figure C.2 ——Alert acknowledgment transport**

The packet shown in Table C.3 shall be used to acknowledge an alert.

**Table C.3 ——UDP port 9301 alert acknowledgment**

| Bytes | Value (hex) | Description |
|---|---|---|
| 2 | uword | ackNumber. Copied from the alert that is being acknowledged. |

### C.5.2 ackNumber

After the host receives an alert, it will send a packet containing the ackNumber for the alert (in network byte order) to UDP port 9301 (the uword ackNumber is the only information transmitted). This lets the printer know that the alert has been received. If the alert acknowledgment is not received, the printer shall retransmit the alert six times

(implementation dependent). If the host does not respond, the printer will assume that the host is no longer present and will cease sending any future alerts to it. The host shall acknowledge every alert.

## C.6 Alert registration

Monitors (hosts that wish to receive alerts) must register with the printer to begin receiving alerts. The number of hosts that a printer allows to register may be limited (implementation dependent). Monitors are encouraged to register frequently to ensure that they are not removed from the printer registry as a result of any implementation dependent aging algorithm that a printer might choose to use to handle monitors that disappear without unregistering for alerts.

The actual registration process is facilitated using IEEE 1284.1 commands to enable alerts. Sending one of these commands to the printer (using the transport depicted in Figure C.1 and the encapsulation described in Table C.1) shall cause the printer to add an entry referencing the sending host to its alert registry. For each entry in the alert registry, the printer shall maintain a list of the various types of alerts for which the host has registered. When a printer alert condition occurs, the printer shall use the registry to determine which specific host(s) shall receive alert messages.

## C.7 Printing

To facilitate printing, data must be transported from a host to the printer. The legacy implementation chose to use TCP port number 9400 for print data transport, as depicted in Figure C.3.

The printer shall accept IEEE 1284.1 data and/or commands received via this transport. The printer shall use this transport to return any response to a command received on it.

The first packet of IEEE 1284.1 data (packet immediately following the data block) shall be the Start Session packet. Any other packets follow. The last packet shall be the End Session packet.

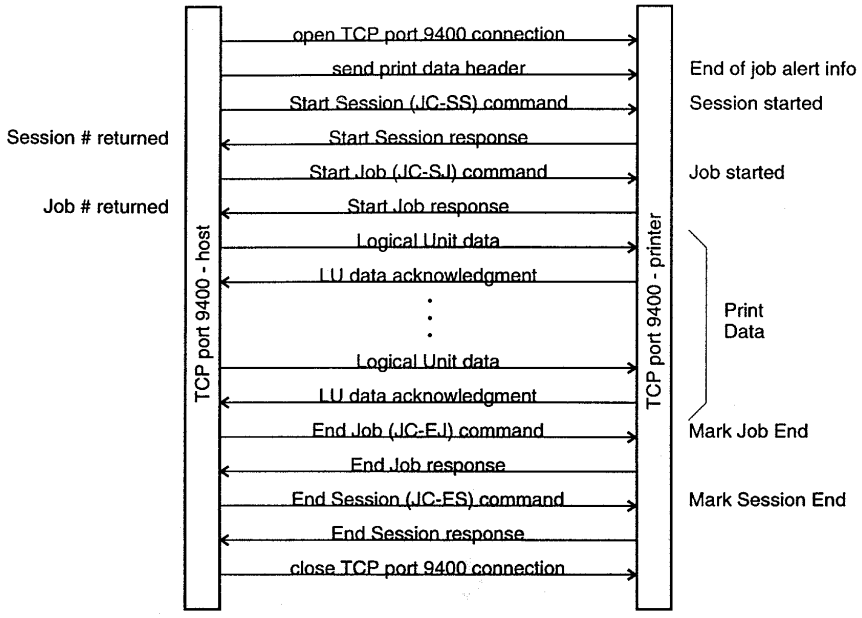


**Figure C.3 ——TCP port 9400 transport**

## C.8 Data block

The presence of a printer server (a device where multiple printers share a common IP address) adds requirements that must be dealt with outside of IEEE Std 1284.1-1997. To address these requirements, the reference implementation precedes any data sent on the port with the data block defined in Table C.4. The data block serves the following multiple purposes:

— It selects a specific printer at a IP address for a printer server.
— It provides data content information for printer server use, eliminating the need for printer server to parse data.
— It allows host to specify IP address and port number of client wishing to receive end of job information for this job.

**Table C.4 ——TCP print data header**

| Bytes | Value (hex) | Description |
|---|---|---|
| 2 | uword (n+5) | length. Number of remaining bytes in this header. |
| 2 | uword | numberOfFields. The total number of fields contained in the print data header. This value shall not be less than 7. |
| 1 | ubyte | deviceIndex. A single ASCII character (1–9) representative of one of multiple printers assigned to the IP address. |
| 1 | 00 | NULL |
| 1 | bbyte | flag. Shall be 11 hexadecimal for IEEE 1284.1 data content. |
| 1 | 00 | NULL |
| s | ASCII | hostNameOfSender |
| 1 | 00 | NULL |
| | ASCII | jobName |
| 1 | 00 | NULL |
| j | ASCII | userWhoSubmittedJob |
| 1 | 00 | NULL |
| | ASCII | queueName |
| 1 | 00 | NULL |
| | ASCII | jobAlertHostString |
| 1 | 00 | NULL |

## C.9 Definitions

### C.9.1 numberOfFields

This unsigned word is the total number of fields contained in the print data header. Each field will be followed by NULL. The value of numberOfFields shall not be less than 7. numOfFields is provided to allow for future extension of the print data header block.

### C.9.2 deviceIndex

This unsigned byte is used to accommodate multiple printers at a single IP address. This allows the host to specify which printer at this IP address is to receive the data. When only one printer is located at the IP address, deviceIndex is set to a hexadecimal value of 31.

### C.9.3 flag

The bit-encoded byte allows the host to describe the data content that follows. This allows the printer server to, if needed, filter the data content. Flag is encoded as follows:

- Bits 7, 6, and 5 shall be zero.
- Bit 4 shall be a one if data content is properly formatted IEEE 1284.1 data and/or commands, else shall be zero.
- Bit 3 shall be a one if printer server (or network interface) should parse data content and add a carriage return following each line feed to the data transported to the printer, else shall be zero.
- Bit 2 shall be a one if data content is PostScript, else shall be zero.
- Bit 1 shall be a one if data content is PCL, else shall be zero.
- Bit 0 shall be a one.

### C.9.4 hostNameOfSender

If data content contains properly formatted IEEE 1284.1 data and/or commands, then this field shall contain a copy of the host string field contained in the JC-SJ command. Otherwise it shall contain a space.

### C.9.5 jobName

If data content contains properly formatted IEEE 1284.1 data and/or commands, then this field shall contain a copy of the job string field contained in the JC-SJ command. Otherwise it shall contain a space.

### C.9.6 userWhoSubmittedJob

If data content contains properly formatted IEEE 1284.1 data and/or commands, then this field shall contain a copy of the user string field contained in the JC-SJ command. Otherwise it shall contain a space.

### C.9.7 queueName

If data content contains properly format IEEE 1284.1 data and/or commands, then this field shall contain a copy of the additional information string field contained in the JC-SJ command. Otherwise it shall contain a space.

### C.9.8 jobAlertHostString

If the data content contains properly formatted IEEE 1284.1 data and/or commands, then this field shall contain the address of the client that will receive job alert information related to this specific job (see Figure C.4). The address shall be formatted as a hexadecimal number of the form AAAAAAAAPPPP where AAAAAAAA is the IP address of the host (AA.AA.AA.AA) and PPPP is the UDP port number where the client monitor will receive the alert information. If job alerts are not desired or the data content does not contain properly formatted IEEE 1284.1 data and/or commands, then this field shall contain the hexadecimal value FFFFFFFFFFFF.

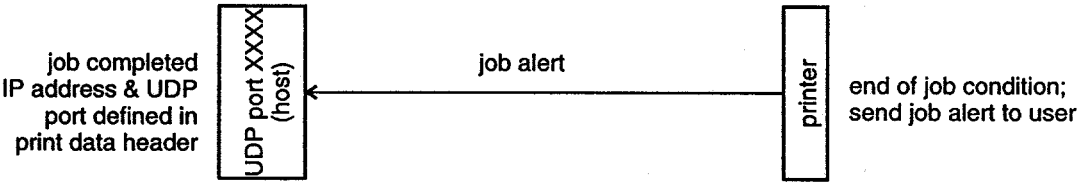


**Figure C.4 ——Host-Specified UDP port for job alerts**

# Annex D Novell® NetWare® transport mapping example

# (Informative)

## D.1 Introduction

IEEE Std 1284.1-1997 deals with bidirectional communications between a printer and a host. However, the transport mechanism between the printer and host is not defined. This annex describes an example mapping of the transport of IEEE 1284.1 packets encapsulated within Internetwork Packet Exchange™ (IPX™) datagrams.

An IEEE Std 1284.1-1997 implementation in a networking environment presents several problems not found in locally attached host-printer configurations, including the following:

— How to deal with one-to-many and many-to-many relationships where several users require access to multiple printers,
— Which user(s) receive(s) unsolicited Device Status Alerts (DSAs), and
— How to sequence packets to ensure no data is lost.

This example addresses these and other problematic areas when implementing IEEE Std 1284.1-1997 in a network environment.

## D.2 The IPX datagram and IEEE Std 1284.1-1997

The IPX datagram is widely used since it is one of Novell's main transport mechanisms. IEEE Std 1284.1-1997 shall utilize the NetWare Core Protocol™ (NCP™) packet type with specific socket numbers assigned by Novell. The IPX packet header structure is shown in Table D.1.

**Table D.1 ——IPX packet header**

| Byte | Name | Description |
|---|---|---|
| 0–1 | Checksum | Must be 0xFFFF |
| 2–3 | Packet Length | Total length of packet in bytes |
| 4 | Transport Control | Set to 0x00 |
| 5 | Packet Type | Set to 0x11-NCP packet type |
| 6–9 | Destination Network | Unique network number of the destination network |
| 10–15 | Destination Node | Unique (on local network) node address of destination |
| 16–17 | Destination Socket | Must be one of the defined IEEE 1284.1 sockets |
| 18–21 | Source Network | Unique network number of the source network |
| 22–27 | Source Node | Unique (on local network) node address of source |
| 28–29 | Source Socket | User definable |

The Destination Socket allows applications to identify IEEE 1284.1 messages without inspecting the data area and shall be an element in the list of assigned sockets. (See Table D.2.)

**Table D.2 ——IEEE 1284.1 sockets for NCP on IPX**

| Socket | Description |
|---|---|
| 0x862C | IEEE 1284.1 Printer Control, Port 0 |
| 0x862D | IEEE 1284.1 Printer Control, Port 1 |
| 0x862E | IEEE 1284.1 Printer Control, Port 2 |
| 0x862F | IEEE 1284.1 Printer Control, Port 3 |
| 0x8630 | IEEE 1284.1 Printer Control, Port 4 |
| 0x8631 | IEEE 1284.1 Printer Control, Port 5 |
| 0x8632 | IEEE 1284.1 Printer Control, Port 6 |
| 0x8633 | IEEE 1284.1 Printer Control, Port 7 |

Following the IPX packet header is the IEEE 1284.1 over IPX data structure as defined in Table D.3 or Table D.4, depending on whether the packet is a request or a reply.

**Table D.3 ——IEEE 1284.1 request packet**

| Byte | Name | Description |
|---|---|---|
| 0–1 | Checksum | Must be 0xFFFF |
| 2–3 | Packet Length | Total length of packet in bytes |
| 4 | Transport Control | Set to 0x00 |
| 5 | Packet Type | Set to 0x11-NCP packet type |
| 6–9 | Destination Network | Unique network number of the destination network |
| 10–15 | Destination Node | Unique (on local network) node address of destination |
| 16–17 | Destination Socket | Shall be one of the defined IEEE 1284.1 sockets |
| 18–21 | Source Network | Unique network number of the source network |
| 22–27 | Source Node | Unique (on local network) node address of source |
| 28–29 | Source Socket | User definable |
| 30–n | IEEE 1284.1 Data | Up to 512 bytes of IEEE 1284.1 data |

Replies and Device Status Alerts (DSAs) may span several packets and require a sequencing algorithm to ensure no data is lost. IEEE 1284.1 replies shall begin the sequencing with a value of zero (0) and shall be incremented with each continuation packet. Unsolicited DSA replies shall begin at zero (0) and increment with every DSA packet. That is, the initial DSA reply packet shall be zero. The second DSA reply packet shall be one and the $n^{th}$ DSA reply packet shall have a sequence number of (n–1). If a packet is lost, the application must ignore the entire reply, or sequence of replies, and reissue the command. Note the sequence number may wrap around from 0xFFFF ⇒ 0x0000. (See Table D.4.)

**Table D.4 ——IEEE 1284.1 reply packet**

| Byte | Name | Description |
|---|---|---|
| 0–1 | Checksum | Must be 0xFFFF |
| 2–3 | Packet Length | Total length of packet in bytes |
| 4 | Transport Control | Set to 0x00 |
| 5 | Packet Type | Set to 0x11-NCP packet type |
| 6–9 | Destination Network | Unique network number of the destination network |
| 10–15 | Destination Node | Unique (on local network) node address of destination |
| 16–17 | Destination Socket | Must be one of the defined IEEE 1284.1 sockets |
| 18–21 | Source Network | Unique network number of the source network |
| 22–27 | Source Node | Unique (on local network) node address of source |
| 28–29 | Source Socket | User definable |
| 30–31 | Sequence Number | An unsigned short integer in high-low order |
| 32–n | IEEE 1284.1 Data | Up to 256 bytes of IEEE 1284.1 data |

## D.3 Determining IEEE Std 1284.1-1997 compatibility

An application can determine support for the IEEE Std 1284.1-1997 protocol by sending any IEEE 1284.1 command to the printer in question. If a correct IEEE 1284.1 reply is received, the printer supports IEEE Std 1284.1-1997. Otherwise, the application shall not receive a reply.

## D.4 Device Status Alerts

Device Status Alerts (DSAs) are unsolicited IEEE 1284.1 replies from the printer to the host. Any application wishing to receive DSAs must register with the IEEE 1284.1 device handler. The recommended registration procedure is to send a PCC-SDSA command. The PCC-SDSA command allows an application to specify the conditions triggering a DSA reply. The IEEE 1284.1 device handler shall send only requested DSAs to the address

 

specified in the PCC-SDSA command's source NETWORK.NODE.SOCKET address. There shall be a limited number of entries in the registry table so applications should only register once.

## D.5 Polling

The application and IEEE 1284.1 device handler must both be able to determine if the other has failed. If one is powered off, there is no way of informing the other so resources may be released. Therefore, it is recommended the application periodically (once a minute recommended) send a PCC-SDSA command. If the IEEE 1284.1 device handler is still operational, the reply shall be a PCC-SDSA response. Otherwise, no reply shall be received and the application may free its resources.

It is also recommended the IEEE 1284.1 device handler age entries in the registry. If the IEEE 1284.1 device handler has not received any IEEE 1284.1 request packets within a limited amount of time (five minutes recommended), it may assume the host station is "dead" and free associated resources.

# Annex E De-multiplexing alerts

# (Informative)

## E.1 Enabling alerts

A device will not send asynchronous messages (alerts) back to the host on a port without first being told to do so. The only exception to this is the power-on alert. IEEE 1284.1 commands are provided that allow an application communicating with the printer on a port to enable various alerts for that port. It is the printer's responsibility to maintain an alert mask for this port and return only those alerts that have been requested.

## E.2 Multiple port environment

Many printers contain multiple data ports. In most instances concurrent operation over all ports is supported. In these instances, it is the printer's responsibility to maintain separate alert masks for each port and return alerts to each port based on the alert mask for the specific port.

A device will send any asynchronous message (alert) on all ports that are enabled for that type of alert. There is no filtering of alerts on a port basis. This allows monitoring agents to receive information about the device's operation even if that information is specific to a given port.

## E.3 Multiple node environment

In some cases, especially those related to network printers, the printer can be concurrently attached to multiple host nodes using the same port. In these cases it is suggested that a proxy agent be used funnel information to the printer. The key responsibilities of the proxy agent funneling traffic to the printer through a given port are to ensure that printer responses are returned to the appropriate application or device and to de-multiplex alerts coming from the printer for applications or devices that have registered for them. In this case the agent must maintain a table of alert masks. An entry would be made in the table for each application or device communicating through the agent to the printer through the port. The agent would calculate the alerts required to satisfy the requests made by the applications or devices and would enable the printer to return these alerts on the given port.

# Annex F Host ownership of an IEEE 1284.1 printer

# (Informative)

A printer supporting a bidirectional printing data stream requires some degree of ownership from a host application or the host print subsystem. The amount of ownership varies depending on the printing requirements of the application and the level of conversation maintained between the host and printer. Some of the applications that can be envisioned are as follows.

## F.1 Smart “printer server”

The printer server would be started as an application that would in turn begin a “connection-oriented” (as opposed to a “connectionless”) communications session with the printer for the duration of the communications (e.g., until the printer server is shut down). The printer server would have access to font, status, error, and other information. This information would allow software vendors to develop applications that include improved job management, font management, forms management, accounting information, etc., to both network administrators and end users on a network.

## F.2 Auto-Configurable device driver

In the Microsoft® Windows® and IBM® Presentation Manager® environments, the device driver (on first print or initialization) could enter into a conversation with the IEEE 1284.1 compliant printer to obtain an up-to-date snapshot of the printer configuration (as reported by the printer) to update the configuration tables previously created. The device driver could then instruct the printer to enter the compatible mode and proceed in the methods currently used. This information would guarantee appropriate setup information is installed in the windows or presentation manager device driver tables. It would also greatly enhance end-user ease of use in attaching and using a new printer.

## F.3 Operating system embedded support

For long-range objectives, operating system developers can embed support within the base operating system to provide information dynamically to the end-user applications. Types of information include fonts available in the printer, paper sources, paper outputs, printer features, paper sizes and types, etc.

# Annex G Bibliography

# (Informative)

## G.1

[B1] Adobe™ Systems PostScript™ Language Reference Manual, Second Edition.

[B2] Hewlett-Packard® Printer Job Language Technical Reference Manual (5961-0998).

[B3] IEEE Std 1284-1994, IEEE Standard Signaling Method for a Bidirectional Parallel Peripheral Interface for Personal Computers (ANSI).

[B4] ISO/IEC 10175, Information technology—Text and office systems—Document Printing Application (DPA).

[B5] RFC1213, *Management Information Base for network management of TCP/IP-based internets: MIB-II*. K. McCloghrie, M.T. Rose. Mar-01-1991 (retrievable as ftp://ftp.isi.edu/in-notes/frc1213.txt).

[B6] RFC1759, *Printer MIB*. R. Smith, F. Wright, T. Hastings, S. Zilles, and J. Gyllenskog. March 1995 (retrievable as ftp://ftp.isi.edu/in-notes/frc1759.txt).