

# *Proposal draft*

**To: Printer Working Group – 1394 Printing**  
**Date: September 20, 1998**  
**Rev. 0.1a**

---

*Fumio Nagasaka*  
Epson Software Development Laboratory Inc.  
80 Hara-Shinden, Shiojiri-shi, Nagano-Ken  
JAPAN 399-07

## **Table of Contents**

PROPOSAL DRAFT .....	1
<b>1. OVERVIEW .....</b>	<b>1</b>
1.1. SCOPE .....	1
1.2. GOAL .....	1
1.3. PURPOSE .....	1
<b>2. REQUIREMENTS AND OPEN ISSUE .....</b>	<b>2</b>
2.1. REQUIREMENTS FOR SBP-2 PRINTING .....	2
2.2. OPEN ISSUE .....	2
2.2.1. <i>Compatibility with conventional LPT device</i> .....	2
2.2.2. <i>Dynamic allocation of logical channels</i> .....	2
2.2.3. <i>Bus reset and enumeration</i> .....	2
2.2.4. <i>Dynamic end points</i> .....	2
2.3. NORMATIVE REFERENCES .....	3
2.3.1. <i>Approved references</i> .....	3
2.3.2. <i>References under development</i> .....	3
2.4. COMMAND SET SPECIFIC TERMINOLOGY .....	3
2.4.1. <i>Forward data flow</i> .....	3
2.4.2. <i>Reverse data flow</i> .....	3
2.4.3. <i>Session</i> .....	3
2.4.4. <i>Indication</i> .....	3
2.4.5. <i>Request</i> .....	3
2.4.6. <i>Command owner</i> .....	4
2.4.7. <i>Initiator</i> .....	4
2.4.8. <i>Target</i> .....	4
2.5. COVERAGE OF DOCUMENTS .....	4
<b>3. TRANSACTION OVERVIEW (INFORMATIVE) .....</b>	<b>5</b>
3.1. OVERVIEW .....	5
3.1.1. <i>Design feature same as the current proposal</i> .....	5
3.1.2. <i>Design feature different from the current proposal</i> .....	5
3.2. BI-DIRECTIONAL COMMUNICATION MODEL .....	6
3.3. TARGET MODEL .....	6
3.4. INITIATOR MODEL .....	8
3.4.1. <i>Resource Session</i> .....	9
<b>4. DATA STRUCTURE .....</b>	<b>10</b>
4.1. GENERAL DATA STRUCTURE OF SBP-2 .....	10
4.1.1. <i>Command Block ORB format</i> .....	10
4.1.2. <i>Status Block</i> .....	10
<b>5. STATUS BLOCK AND TRANSACTION STATUS REPORT .....</b>	<b>12</b>
5.1. STATUS BLOCK FOR NORMAL SESSION .....	12
5.2. LEGACY MODE STATUS BLOCK .....	14
5.3. PARAMETER ENCODING .....	16
5.4. STANDARD PARAMETERS .....	17

5.4.1.	<i>MAX TASK SET SIZE</i> .....	17
5.4.2.	<i>MAX I2T DATA SIZE</i> .....	17
5.4.3.	<i>MAX T2I DATA SIZE</i> .....	17
5.4.4.	<i>MAX BUFFER SIZE</i> .....	17
5.4.5.	<i>MAX QUEUE Entity</i> .....	17
5.4.6.	<i>1284 PnP String</i> .....	17
5.5.	ERROR REPORTING PRECEDENCE.....	18
8.	COMMANDS.....	19
8.1.	COMMAND OWNER.....	19
8.2.	INITIATOR'S RESPONSIBILITY.....	19
8.3.	GENERIC IMAGING PROFILE COMMAND SET.....	19
8.4.	PRINTING CLASS SPECIFIC COMMAND SET.....	20
8.5.	COMMAND FORMAT.....	20
8.5.1.	<i>Bit field definition for op-code</i> .....	20
8.5.2.	<i>Bit field description for op-qualifier</i> .....	21
8.5.3.	<i>Where shall the initiator and the target store commands</i> .....	22
8.6.	TRANSPORT_OPEN_INDICATION.....	24
8.6.1.	<i>TRANSPORT_OPEN_INDICATION response</i> .....	24
8.7.	TRANSPORT_CLOSE_INDICATION.....	26
8.7.1.	<i>TRANSPORT_CLOSE_INDICATION response</i> .....	26
8.8.	TRANSPORT_I2T_DATA.....	28
8.8.1.	<i>TRANSPORT_I2T_DATA response</i> .....	28
8.8.2.	<i>Optional field for LEGACY MODE</i> .....	29
8.9.	NOP.....	30
8.9.1.	<i>NOP response</i> .....	30
8.10.	MONITOR.....	31
8.10.1.	<i>MONITOR response</i> .....	31
8.11.	TRANSPORT_OPEN_REQUEST.....	33
8.11.1.	<i>TRANSPORT_OPEN_REQUEST response</i> .....	33
8.12.	TRANSPORT_CLOSE_REQUEST.....	34
8.12.1.	<i>TRANSPORT_CLOSE_REQUEST response</i> .....	34
8.13.	TRANSPORT_T2I_DATA.....	35
8.13.1.	<i>TRANSPORT_T2I_DATA response</i> .....	35
8.14.	T2I_DATA_REQUEST.....	37
8.15.	OPEN_REQUEST_REPLY.....	38
8.15.1.	<i>OPEN_REQUEST_REPLY response</i> .....	38
8.16.	CLOSE_REQUEST_REPLY.....	41
8.16.1.	<i>CLOSE_REQUEST_REPLY response</i> .....	41
8.17.	GET_DEVICE_ID.....	43
8.17.1.	<i>GET_DEVICE_ID response</i> .....	43
8.18.	SET_LEGACY_ENABLE.....	45
8.18.1.	<i>SET_LEGACY_ENABLE response</i> .....	45
8.19.	SOFT_RESET.....	47
8.19.1.	<i>SOFT_RESET response</i> .....	47
8.20.	COMMANDS SUMMARY.....	49
9.	ISSUES.....	50
9.1.	LOGIN.....	50
9.2.	SERVICE DISCOVERY.....	50

9.3.	DATA PACKETS VS. DATA STREAM COMMUNICATIONS MODEL? .....	50
9.4.	FUNCTION DEVICE TYPE NUMBER IN UNIT DIRECTORY. ....	50
9.5.	MAXIMUM DATA SIZE .....	50
9.6.	UNSOLICITED STATUS REGISTER ENABLE.....	50
9.7.	TARGET LOGOUT .....	50
9.8.	TIMERS .....	51
9.9.	PLUG & PLAY SUPPORT .....	51
9.10.	NON SYMMETRIC CONNECTION.....	51
9.11.	TRANSPORT CAPABILITIES NEGOTIATION RESOURCE COMMITMENT. ....	52
9.12.	NOTIFY STATUS .....	52
9.13.	TARGET RESET .....	52
9.14.	MULTIPLE UNIT DIRECTORIES .....	52
APPENDIX A. EXAMPLE CONFIGURATION ROM OF THE 1394 PWG PRINTER .....		54
APPENDIX B. BACKWARD COMPATIBILITY.....		60

## **Table of Figures**

FIGURE 1	COVERAGE OF THIS DOCUMENT .....	4
FIGURE 2	TARGET MODEL OF CURRENT 1394 PWG PROFILE .....	6
FIGURE 3	TARGET MODEL DESCRIBED IN THIS DOCUMENT .....	7
FIGURE 4	COMMAND BLOCK ORB TYPES FOR BI-DIRECTIONAL TRANSPORT MODEL.....	8
FIGURE 5	ORB'S LIST IN AN INITIATOR DEVICE .....	8
FIGURE 6	TYPES OF COMMAND BLOCK ORB.....	9
FIGURE 7	RESOURCE SESSION .....	9
FIGURE 8	COMMAND BLOCK ORB (STANDARD FORMAT).....	10
FIGURE 9	DEFAULT STUAS BLOCK SHOWN IN SBP-2 SPECIFICATION.....	11
FIGURE 10	STATUS BLOCK FORMAT DEFINED BY THE 1394 PWG COMMAND SET .....	12
FIGURE 11	STATUS BLOCK FORMAT WHEN LEGACY MODE ENABLED .....	14
FIGURE 12	PARAMETER ID AND VALUE FORMAT.....	16
FIGURE 13	EXAMPLE OF PARAMETER ENCODING.....	16
FIGURE 14	COMMAND FORMAT.....	20
FIGURE 15	DEFAULT PLACE HOLDER OF THE INDICATION .....	23
FIGURE 16	ORB STRUCTURE CONTAINING TRANSPORT_OPEN_INDICATION.....	24
FIGURE 17	REPLY FOR TRANSPORT_OPEN_INDICATION.....	25
FIGURE 18	ORB STRUCTURE CONTAINING TRANSPORT_CLOSE_INDICATION.....	26
FIGURE 19	ORB STRUCTURE CONTAINING TRANSPORT_I2T_DATA .....	28
FIGURE 20	OPTIONAL FIELD FOR LEGACY MODE .....	29
FIGURE 21	ORB STRUCTURE CONTAINING NOP COMMAND.....	30
FIGURE 22	ORB STRUCTURE CONTAINING MONITOR COMMAND.....	31
FIGURE 23	DATA FORMAT OF TRANSPORT_OPEN_REQUEST .....	33
FIGURE 24	DATA FORMAT OF TRANSPORT_COSE_REQUEST.....	34
FIGURE 25	ORB STRUCTURE CONTAINING TRANSPORT_T2I_DATA .....	35
FIGURE 26	DATA FORMAT OF T2I_DATA_REQUEST .....	37
FIGURE 27	ORB STRUCTURE CONTAINING OPEN_REQUEST_REPLY .....	38
FIGURE 28	REPLY PARAMETER IN OPEN_REQUEST_REPLY RESPONSE.....	39
FIGURE 29	ORB STRUCTURE CONTAINING CLOSE_REQUEST_REPLY .....	41
FIGURE 30	ORB STRUCTURE CONTAINING GET_DEVICE_ID.....	43
FIGURE 31	ORB STRUCTURE CONTAINING SET_LEGACY_ENABLE.....	45
FIGURE 32	ORB STRUCTURE CONTAINING SOFT_RESET .....	47
FIGURE 33	COMMANDS SUMMARY.....	49
FIGURE 34	TARGET DEVICE WHICH HAS ONE BI-DIRECTIONAL QUEUE.....	60
FIGURE 35	TWO PHASES IMPLEMENTATION OF THE COMMAND SET .....	60

## **Table of Tables**

TABLE 1	LEGACY PORT STATUS .....	14
TABLE 2	BIT FIELD IN PORT STATUS.....	15
TABLE 3	PARAMETER ID AND DESCRIPTION .....	17
TABLE 4	GENERIC IMAGING PROFILE COMMANDS.....	20
TABLE 5	PRINTING CLASS SPECIFIC COMMANDS.....	20

TABLE 6 BIT FIELD IN THE FIRST BYTE OF A COMMAND FORMAT ..... 21  
TABLE 7 DETAIL OF EACH BIT FIELD ..... 21  
TABLE 8 BIT FIELD IN THE FIRST BYTE OF A COMMAND FORMAT ..... 21  
TABLE 9 BIT FIELD DESCRIPTION OF THE OP-QUALIFIER ..... 22  
TABLE 10 ERROR CODE ..... 40

# 1. Overview

## 1.1. Scope

The 1394 PWG is continuing effort for standardization of IEEE 1394 Printing Protocol. One of possible solution which is available under existing consumer PC operating systems is SBP-2 printing. According to this idea, a PC shall act as SBP-2 initiator and printers shall be SBP-2 target devices.

Due to limitation of SBP-2 facilities, printing session between a host device and a printer does not provide symmetrical service. Asymmetry of service is uncomfortable for peer to peer printing (or so-called "Direct-printing") applications. However other PC peripherals including storage devices and image scanners tend to support SBP-2 protocol for IEEE 1394 platform. Thus, several consumer PC operating system will provide SBP-2 initiator capability.

Currently the 1394 PWG has three meeting threads; The first thread is discussing about configuration ROM design for 1394 PWG compliant target device. Other two threads are discussing about printing protocol profile over SBP-2, involved with transport command set. This documents describes new proposals for the transport command set and transport models.

## 1.2. Goal

Standardization efforts sometimes require several months to meet common agreement. However some consumer PC operating systems will be available in the market at the end of this year or beginning of 1999. The 1394 PWG standard will not be able to synchronize beta development cycle of these operating system. Thus the SBP-2 Printing Protocol standardization through 1394 PWG shall keep backward compatibility with existing SBP-2 driver stack which will be released with these new PC operating systems. If users are using SBP-2 printer on next PC OS platform(s), users may use same SBP-2 printer model after 1394 PWG protocol had been issued.

This document propose a transport command set;

- to meets new requirements suggested through the 1394 PWG meetings,
- to keep backward compatibility with conventional printing models,
- to provide new high performance transport fully utilizing SBP-2 facilities,
- and to facilitate to make provision for new printing platform which will be available in future.

## 1.3. Purpose

Purpose of this document are listed below;

- to provide fair access from the target and the initiator,
- to provide full duplex bi-directional communication pipes,
- to provide dynamic allocation of logical channels which does not depend upon re-structure of configuration ROM,
- and to facilitate to develop common application layer programs witch covers other transport protocol driver(s).

## 2. Requirements and open issue

### 2.1. Requirements for SBP-2 printing

- Bi-directional communication between PCs and printers
- Unordered execution of SBP-2 ORBs provided in an initiator's memory space
- Multiple host devices support capability
- Multiple target devices connectivity
- Fare access both from targets and initiators
- Status service while printing

### 2.2. Open Issue

#### 2.2.1. *Compatibility with conventional LPT device*

A conventional parallel port had hardware initialization after a printing session had aborted. To use this functionality, in Windows Operating System(s), old LPT.VxD had a software "Init" signal for parallel port. To ensure fully software compatibility, a USB Printing Class function device object perform, device had

#### 2.2.2. *Dynamic allocation of logical channels*

The current proposal is assuming to use multiple LUNs to make multiple logical channels. This design seems to meet primitive requirements of printing or simple multi function peripherals. However if clients application and host devices require more logical channels, or require dynamic adding or removing logical channels, the target will be required to provide dynamic appending of new LUNs in configuration ROM. And dynamic change of the configuration ROM requires bus reset phase to inform new configuration of the node.

#### 2.2.3. *Bus rest and enumeration*

When a target device dynamically adds LUNs in configuration ROM. It requires to cause bus reset to 1394 bus system. The current design of some consumer PC operating system has several area t be developed to find new LUN and services.

#### 2.2.4. *Dynamic end points*

This proposal describes dynamic allocation of logical endpoint (called "a socket" in this document) which would be applicable for logical channels. Any device may not specify sockets. This structure would be an alternative solution to allocate multiple communication pipes without LUNs.



## 2.3. Normative References

### 2.3.1. Approved references

ISO/IEC 13213:1994, Control and Status Register (CSR) Architecture for Microcomputer Buses  
IEEE Std. 1394-1995, Standard for a High Performance Serial Bus

### 2.3.2. References under development

Serial Bus Protocol 2 (SBP-2) Working Draft, X3T10 Project 1155D Rev.4, May 19, 1998  
<<ftp://ftp.symbios.com/pub/standards/io/t10/drafts/sbp2/>>

**Microsoft Inc.**, Microsoft Plug and Play Design Specification for IEEE 1394 Ver. 1.0b, October 17, 1997

**Microsoft Inc.**, Microsoft Plug and Play Parallel Port Devices Ver. 1.0b March 15, 1996

**Gregory A. Shue**, Hewlett-Packard, "PWG 1394 Transport Command Set Proposal" Revision 0d, August 7, 1998

**Alan Berkema**, Hewlett-Packard, "1394 PRINTER WORKING GROUP IEEE 1394 HIGH SPEED BUS IMAGING DEVICE COMMUNICATIONS SPECIFICATION PRELIMINARY DRAFT PROPOSAL", Revision 0.40, June 5, 1998

**Greg LeClair**, Epson, "1394 PRINTER WORKING GROUP CONFIGURATION ROM for IMAGING DEVICE PROFILE DRAFT PROPOSAL", Revision 0.1, July 2, 1998

## 2.4. Command set specific terminology

### 2.4.1. Forward data flow

In PC printing application, naturally a PC behaves as a data producer and a printer acts as a data consumer. In this document, any data sent from a data producer to a data consumer is called as "forward data". And data flow consisted of "forward data" is called as "forward data flow".

### 2.4.2. Reverse data flow

In PC printing application, naturally a PC behaves as a data producer and a printer acts as a data consumer. A data flow consisted of data sent from a printer to a PC is defined as "reverse data flow".

### 2.4.3. Session

A linked listed ORBs execution subsequently caused after *DOOR\_BELL* ping or update of *ORB\_POINTER*.

### 2.4.4. Indication

A command an initiator issues to a target. An initiator shall store indications in a normal command block ORB standardized in SBP-2 specification.

### 2.4.5. Request

A command a target issues to an initiator. An initiator shall provides input buffer to receive requests from the target.

2.4.6. *Command owner*

A peer which sends an indication or a request in each transaction phase.

2.4.7. *Initiator*

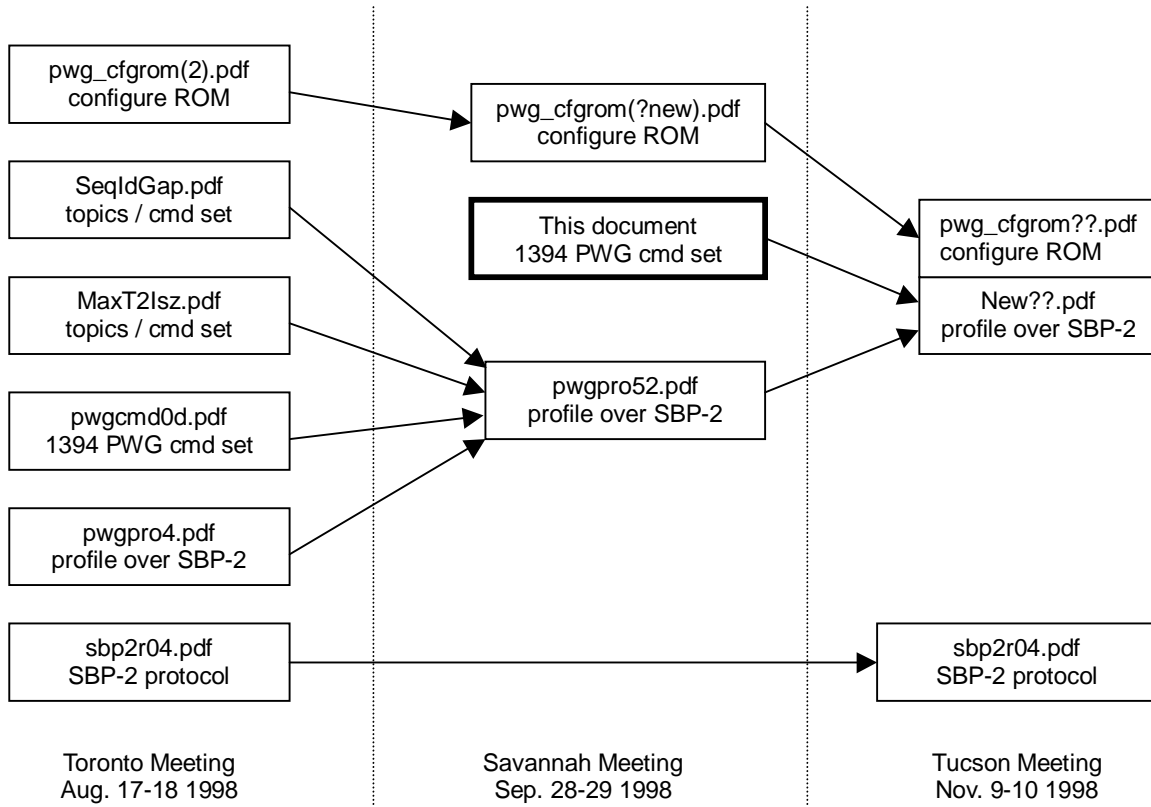
See SBP-2 specification.

2.4.8. *Target*

See SBP-2 specification.

2.5. **Coverage of documents**

This proposal only covers the 1394 PWG transport command set. Involved with 1394 printing architecture, several documents are available in the PWG ftp site. The figure shown below outlines coverage of documents.



**Figure 1 Coverage of this document**

This documents suggests additional requirements and minor improvements for current 1394 PWG command set. This document was edited based on existing documents including "pwgcmd0d.pdf" and "pwgpro4.pdf". The design of IEEE 1212 configure ROM for the target device is beyond the scope. This documents does not describe anything involved with application layer design.

This document assumes readers are familiar with SBP-2 specification, and had already reviewed current proposals available in PWG ftp site.

### 3. Transaction Overview (informative)

#### 3.1. Overview

The 1394 PWG Transport model shall provide communication facilities outlined below;

- A target shall provide two independent bi-directional queues to fetch ORBs provided by an initiator.
- The first queue is a command queue which keeps ORBs fitted with *cmd\_bit* field to be one.
- The other queue is a data queue which consumes data ORBs which has value zero in *cmd\_bit* field.
- A fetch agent of the target device shall consume ORBs with unordered ORB processing model.
- ORBs stored in each queue shall be executed in order.

The 1394 PWG Transport model shall provides two types of command set;

a) Generic imaging profile command set

This command set includes general purpose commands to service bi-directional data transfer, transport open/close and transaction resource control.

b) Printing Class specific command set

This optional command set shall provide services to ease higher layer protocol drivers to access printing device medium. In this case, the command set assumes the printer is a target device.

*3.1.1. Design feature same as the current proposal*

- 1) use two queues
- 2) fetch agent fetches ORBs with unordered execution process
- 3) both peer may open/close channel
- 4) as the initiator's responsibility, it shall keep providing at least one spare input ORB to receive any requests from the target

*3.1.2. Design feature different from the current proposal*

- 1) each queue may contain input ORBs and output ORBs
- 2) single LUN may provide logical channels
- 3) logical channels are allocated dynamically on demand
- 4) the initiator may place parameter in data buffer of an ORB or CDB structure of the ORB.
- 5) commands are abstracted from ORB data structure.

*Note*

The current 1394 PWG command set provides a logical channel facility associated with a LUN. However one consumer PC operating system does not support dynamic LUN which requires dynamic rebuild of the configuration ROM. One possible solution to service dynamic add or remove of a logical channel is multiple logical channel support with one LUN.

### 3.2. Bi-directional Communication Model

An initiator shall place at least one control ORB which is one special purpose input ORB to solicit the target to report buffer requirement for next reverse directional data session. Also the initiator shall place negotiation ORB which is on special purpose output ORB to inform buffer space, when the initiator only can provide smaller space than requested space, as input buffer.

### 3.3. Target Model

A target device may consume ORBs with unordered execution process. Figure 1 shows current target device design of the 1394 PWG profile over SBP-2. As conceptual model, the target device has two queues to fetch ORBs. The command block agent contains one command fetch agent, two command pre-fetch queues, called **WRITE queue** and **READ queue**, and two execution agents, called **WRITE execution agent** and **READ execution agent** connected to the **WRITE queue** and the **READ queue** respectively.

The command fetch agent fetches the normal command block ORB's in order. When the command fetch agent fetches the normal command block ORB, the command fetch agent examines the specified bit-field so-called "Queue ID" in the command\_block field of the command block ORB. The fetch agent dispatches the command block ORB to either the **WRITE queue** or **READ queue** according to the parameter. All WRITE commands are dispatched to the **WRITE queue**, and all READ commands are dispatched to the **READ queue**. The **WRITE execution agent** and **READ execution agent**, execute the commands queued in the **WRITE queue** and **READ queue** respectively.

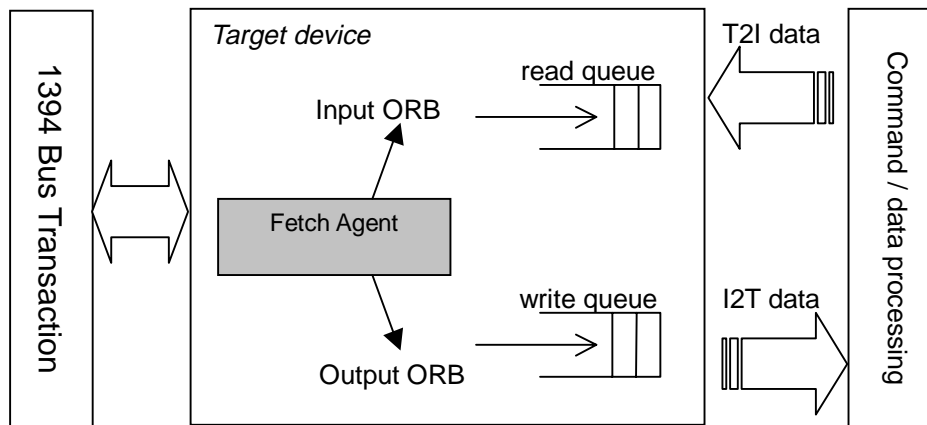
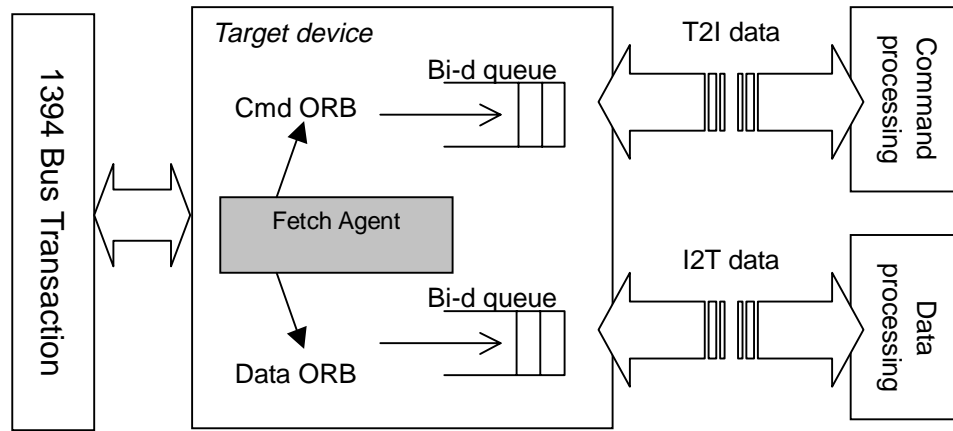


Figure 2 Target Model of current 1394 PWG profile

Each execution agent executes the dispatched command in the connected queue in order. Both execution agents are independent of each other. Each execution agent executes the data transfer associated with the command according to the parameters specified in the command. The target stores a status block in the initiator's memory according to the value of the notify bit of the command block ORB after executing the command as specified by SBP-2. Each execution agent shall store the status\_block in order of execution for that particular agent.



**Figure 3 Target Model described in this document**

Figure 2 shows conceptual target device model. The target shall provide two **bi-directional** pre-fetch queues, one for transport command processing and one for data processing.

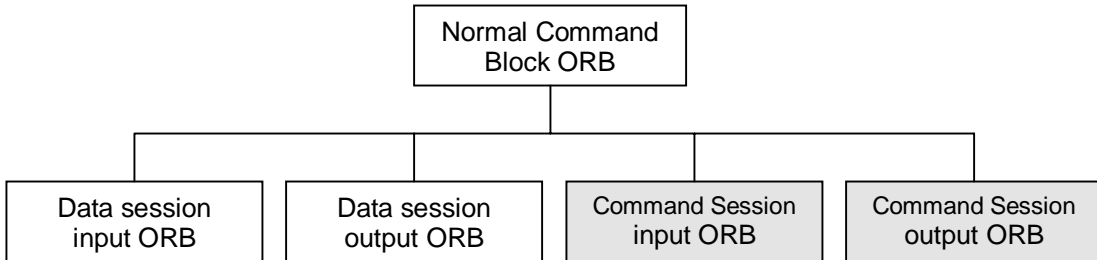
The transport model of this document defines command set for transaction. The command processor of this model establishes connection between a target and an initiator, and also controls resource to be allocated for transaction including dynamic logical end points so-called “socket” in this document. The data processor of the document handles data transfer between the target and the initiator. It consumes data processing commands and enables data path of each peer.

Behavior of each execution agent is same as current proposal. Thus each execution agent executes the dispatched command in the connected queue in order.

Both execution agents are independent of each other. Each execution agent executes the data transfer associated with the command according to the parameters specified in the command. The target stores a status block in the initiator’s memory according to the value of the notify bit of the command block ORB after executing the command as specified by SBP-2. Each execution agent shall store the status\_block in order of execution for that particular agent.

### 3.4. Initiator Model

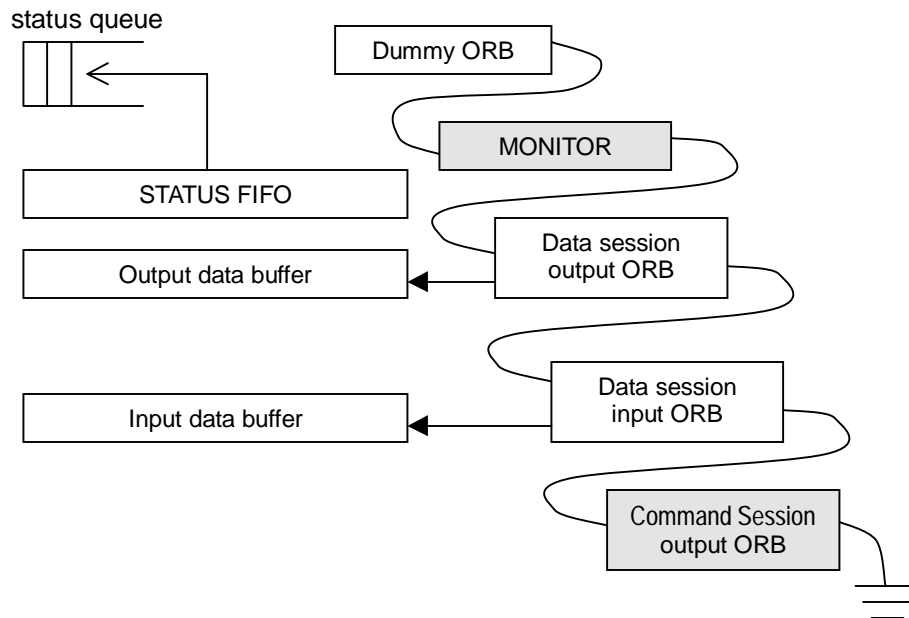
Initiators shall provide four types of Command block ORBs shown below;



**Figure 4 Command block ORB types for bi-directional Transport Model**

Each type of Command Block ORB is distinguished by the fetch agent of a target device with queue id and direction bit field in ORB data structure. If queue id is set to **one**, the command block ORB shall be considered as a command session ORB, and it shall be queued into pre-fetch queue provided for command session.

As described previous section, this proposal allows a target device to fetch command ORBs with unordered execution process. Thus an initiator may place data session ORBs and command session ORBs mixed in a list of ORBs.



**Figure 5 ORBs list in an initiator device**

Figure above shows an example ORB list in an initiator's memory space. The initiator shall provide input/output data buffer and Status\_FIFO. The status of each transaction phase shall be queued and examined as soon as possible.

This document propose that both peer (a target and an initiator) of this transport model shall have fairness to issue open/close request for a **communication pipe**. To facilitate pseudo symmetrical access, the initiator shall provide spare command session input ORB in the list. A figure shown above provides an “MONITOR” ORB. It means this command block ORB has “MONITOR” command, and the direction bit of this ORB shall be **one** to indicate reverse direction the target device write data to buffer of the initiator.

The table summarize each type of ORBs supported in this proposal.

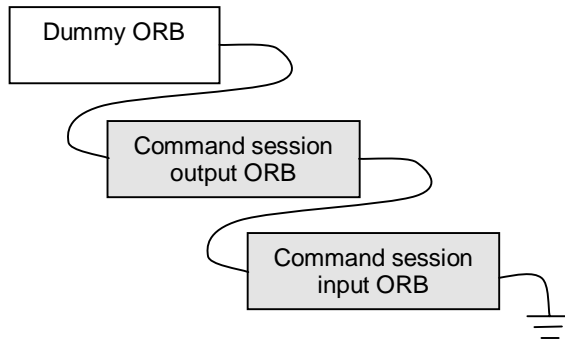
bit field	Data session Input ORB	Data session Output ORB	Command session Input ORB	Command session Output ORB
Queue ID bit	0	0	1	1
Direction bit	1	0	1	0

**Figure 6 Types of Command Block ORB**

Normally Data session ORBs transfer printing data defined by application layer. These data may be including textual data defined by PDL or other Job control language. Or if the target device is an image data source like image scanner, then Data session ORBs may transfer bit-image data generated in the target device to the initiator.

**3.4.1. Resource Session**

A session bracketed with a command session output ORB and a command session input ORB, but does not contain any normal input or output ORBs is defined as a resource session. A ORBs model of resource session is illustrated below.



**Figure 7 Resource session**

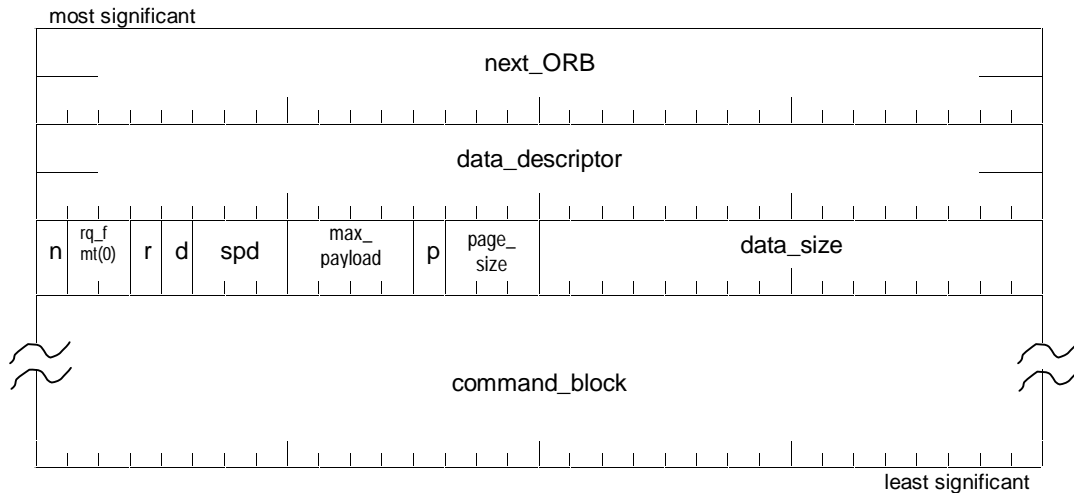
An initiator may confirm resource of the target device utilizing resource session. Or the initiator may request to send status to the target device, while other instance of an application is using LUN to print.

## 4. Data Structure

### 4.1. General data structure of SBP-2

#### 4.1.1. Command Block ORB format

The figure below illustrates the data structure of the Command Block ORB specified in SBP-2 document.



**Figure 8 Command Block ORB (standard format)**

The *next\_ORB*, *data\_descriptor*, *rq\_fmt*, *max\_payload*, *page\_size* and *lun* field and *n* (*notify*), *d* (*direction*) and *p* (*page\_table\_present*) bit are defined by SBP-2 specification.

A Command Block ORB itself is not a command. Any commands issued from this transport model shall not be assumed to be enclosed in Command Block ORBs. A target device may place a command in data buffer specified by a Command Block ORBs. An initiator may enclose a command in a Command Block ORB. In this case if this command makes the target to return reply message, the initiator may specify data buffer as a space to receive reply from the target.

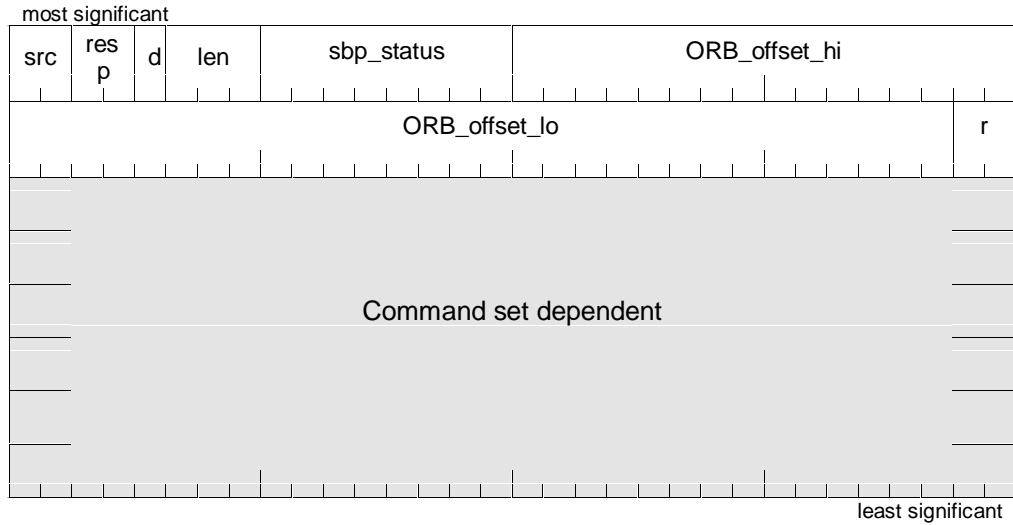
Any commands may have two portion to operate an expected effect. The first portion is an operation code (*op-code* as abbreviated description) and the second portion is a parameter (*operand* as terminology of programming model). And a command often expects to receive replay message as the result of an operation. These are abstracted programming notation of operation. Thus the transport model shall not assume that *op-code(s)* and *operand(s)* are stored in the CDB structure of a Command Block ORB nor Status Block described below.

#### 4.1.2. Status Block

The Status Block is intended to be designed to reports the result of a transaction. Thus it may include various information to complete communication between a target and an initiator. However, An initiator shall not use the Status Block as a pace holder of the reply message.



The data structure of the Status Block is shown below.



**Figure 9 Default STUAS Block shown in SBP-2 specification**

Any command set specific information shall be stored in general purpose data structure including data buffer of an initiator’s memory.

The current proposal of the 1394 PWG PWG command set is well designed from this point of view. It uses only first two quadlet of the command set dependent area of the Status Block, and has compatibility between “SCSI command and status encapsulation” described in Annex B of SBP-2 specification.

**Note**

Some of consumer PC operating system does not parse the command dependent field of the Status Block. Usually SBP-2 protocol driver program provided in these PC operating system is designed to meet general purpose requirements including storage device or other consumer products.

**Note**

In unordered execution model, an initiator easily cause critical section if an initiator removes outstanding ORBs from the list. Thus, a target device is responsible to return status block with suitable information in it’s *src* field.

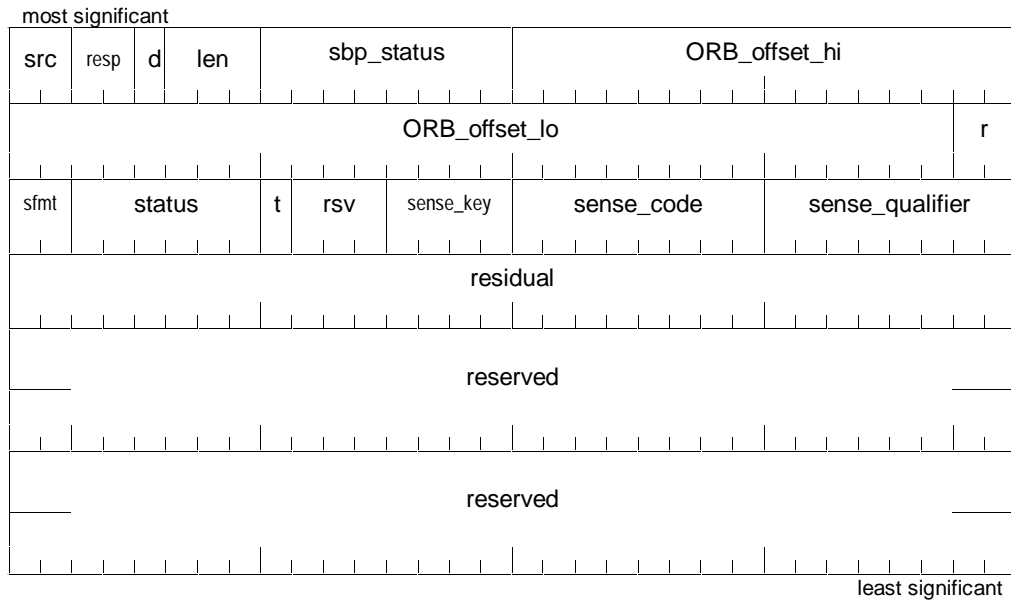
When the *src* field has value zero, the target has knowledge for a subsequent ORB to be fetched, and the ORB will never be fetched by the target. Thus the initiator may reuse or de-allocate it.

When the *src* field has value one, the target does not know the address of a subsequent ORB, until the initiator pings the door bell. And the ORB will be fetched again after the initiator pings the door bell. Because the target needs to read next\_ORB field from this ORB. Thus the initiator shall not reuse or de-allocate it, until the target returns completion status for subsequent ORBs.

## 5. Status block and Transaction status report

### 5.1. Status block for normal session

A target shall store status at an initiator *status\_FIFO* address when a request completes and either the notification bit is set, an error occurred, or the completion notification contains other than zero bits beyond the first two quadlets. The *status\_FIFO* address is obtained implicitly from the fetch agent context. Whenever the target has status to report, it shall store all or part of the status block shown below.



**Figure 10 Status Block format defined by the 1394 PWG command set**

The target shall store a minimum of eight bytes of status information and may store up to the entire 32 bytes defined above so long as the amount of data stored is an integral number of quadlets. A truncated status block shall be interpreted as if the omitted fields had been stored as zeros. The target shall use a single Serial Bus block write transaction to store the status block at the *status\_FIFO* address.

The *src*, *resp*, *len*, *sbp\_status*, *ORB\_offset\_hi*, and *ORB\_offset\_lo* fields and the *dead* bit (abbreviated as *d* in the figure above) shall be as specified by SBP-2.

The *sfmt* field shall specify the format of the status block. The table below defines permissible values for *sfmt*.

Value	Description
0	Current error; status block format defined by this standard
1,2	Reserved for future standardization

3	Status block format vender-dependent
---	--------------------------------------

The *status* field shall contain status information as defined by this standard. The status field shall only be value provided no SBP-2 errors are reported. The receipt of any status shall indicate that the associated task has ended. The following table defines permissible values for *status*.

status value	Description
0	GOOD
2	CHECK CONDITION
3F <sub>16</sub>	Unspecified error
All other values	Reserved for future standardization

Definitions for each status code are given below:

**GOOD.** This status indicates that the target has successfully completed the task.

**CHECK CONDITION.** This status indicates that the target has detected a condition that has stopped the Fetch Agent and implicitly aborted the active task set.

**UNSPECIFIED ERROR.** This status indicates that the target has detected an error condition not specified in this standard. (E.g. an internal error condition which prevents the target from continuing without a power cycle.)

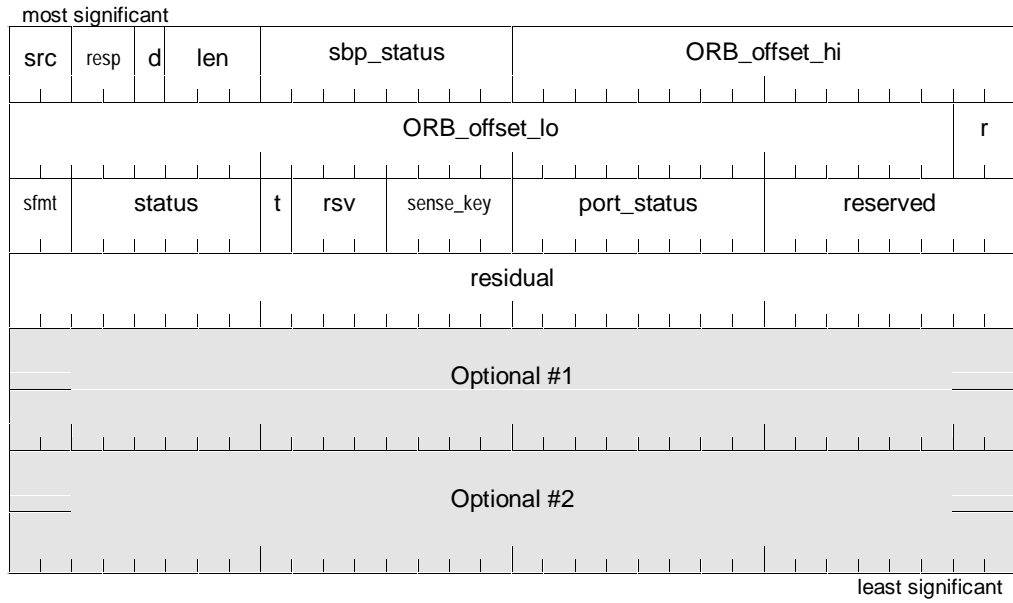
The *tag* bit (abbreviated as *t* in the figure above) shall specify the association of the data returned in the data buffer (out-of-band). This field is reserved for TRANSPORT\_I2T\_DATA, TRANSPORT\_CAPABILITIES, and TRANSPORT\_OPEN commands. When the *tag* bit is zero, the contents of the data buffer are part of the normal data stream. When the *tag* bit is one, the contents of the data buffer shall be understood to be distinct from but synchronous with the normal data stream.

The *sense\_key*, *sense\_code*, and *sense\_qualifier* fields shall contain command completion information defined in this standard.

The contents of the *residual* field are unspecified if the *sfmt* field has a value of three. For *sfmt* values of one or two, the *residual* field is reserved. If the *sfmt* field has a value of zero, this contains the residue of the requested data transfer length minus the length of actual data to be transferred, in bytes, of the command. Negative values are indicated in two's complement notation.

## 5.2. Legacy mode Status block

The legacy mode Status Block is a provision to keep backward compatibility which uses primitive transport model.



**Figure 11 Status Block format when legacy mode enabled**

The target shall store a minimum of eight bytes of status information and may store up to the entire 32 bytes defined above so long as the amount of data stored is an integral number of quadlets. A truncated status block shall be interpreted as if the omitted fields had been stored as zeros. The target shall use a single Serial Bus block write transaction to store the status block at the *status\_FIFO* address.

The *src*, *resp*, *len*, *sbp\_status*, *ORB\_offset\_hi*, and *ORB\_offset\_lo* fields and the *dead* bit (abbreviated as *d* in the figure above) shall be as specified by SBP-2.

The target may use optional field #1 and #2 to exchange buffer resource available for next transaction phase.

The port status field shall contain legacy 8 bit parallel port status illustrated below.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved		Paper empty	Select	Not Error	Reserved		

**Table 1 Legacy Port Status**

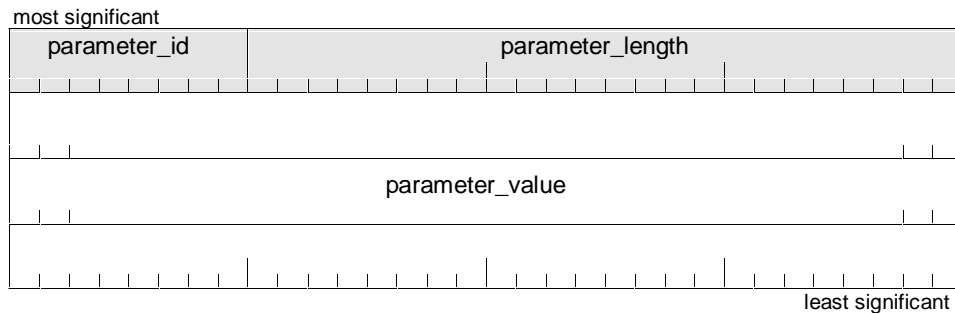
Each bit is meaning conventional flags shown below.

Bit(s)	Field	description
7 – 6	Reserved	Reserved for future standardization, device shall return these bits reset to zero.
5	Paper Empty	1 = Paper Empty, 0 = Paper Not Empty
4	Select	1 = Selected, 0 = Not Selected
3	Not Error	1 = Not Error, 0 = Error
2 – 0	Reserved	Reserved for future standardization, device shall return these bits reset to zero.

**Table 2 Bit field in Port Status**

### 5.3. Parameter Encoding

All parameters passed in the buffer associated with the ORB (whether to set or return values) shall be encoded using the format that follows. Parameter encodings may be packed together to form a list, which may be passed by reference, when supported by the command.



**Figure 12 Parameter ID and value format**

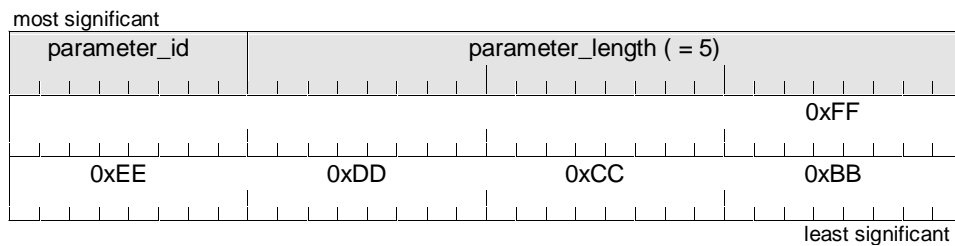
The *parameter\_id* value shall indicate what type of information is encoded in the *parameter\_value* field.

The *parameter\_length* field shall indicate the size (in bytes) of valid data in the *parameter\_value* field.

The *parameter\_value* field contains the actual data for the identified parameter. The data may be of an arbitrary byte length greater than zero. The data shall be padded with reserved bits to the next quadlet boundary. The *parameter\_length* field shall not be adjusted to include this padding.

Data shall be stored in the least significant bits of the *parameter\_value* field. For padding see section 3.2.2 SBP-2 revision 4 figure 3.

Example the 5 byte number FFEEDCCBB is encoded as follows:



**Figure 13 Example of parameter Encoding**

## 5.4. Standard Parameters

The following table describes all the standard transport parameters. All unspecified values are reserved by the 1394 PWG.

ID	Parameter	Access	Size	Description
1	MAX TASK SET SIZE	RO	14 bits	Returns the maximum number of pending commands across both command queues.
2	MAX I2T DATA SIZE	R/W	31 bits	The maximum data size supported for TRANSPORT_I2T_DATA commands
3	MAX T2I DATA SIZE	R/W	31 bits	The maximum data size generated within TRANSPORT_T2I_DATA commands.
4	MAX BUFFER SIZE	RO	31 bits	The maximum data buffer length in bytes available in a session phase.
5	MAX QUEUE Entity	RO	31 bits	The maximum number of rest of queue entities available in a session phase.
6	1284 PnP string	RO	512* bytes	Unicode and ASCII code string specifies device vendor, command set and software version.
7	ERROR CODE	RO	8 bits	Error code

(\*) to be discussed.

**Table 3 Parameter ID and description**

When a target requests open/close or other commands to an initiator, the initiator shall reply the command result in data buffer associated within a command block ORB. And when an initiator indicates open/close or other commands to a target, the target shall reply the command result in data buffer associated within a command block ORB.

### 5.4.1. MAX TASK SET SIZE

Uncovered in this document.

### 5.4.2. MAX I2T DATA SIZE

Uncovered in this document.

### 5.4.3. MAX T2I DATA SIZE

Uncovered in this document.

### 5.4.4. MAX BUFFER SIZE

See TRANSPORT\_OPEN\_REQUEST and TRANSPORT\_OPEN\_INIDICATION section.

### 5.4.5. MAX QUEUE Entity

See TRANSPORT\_OPEN\_REQUEST and TRANSPORT\_OPEN\_INIDICATION section.

### 5.4.6. 1284 PnP String

See GET\_DEVICE\_ID section in this document.

## 5.5. Error Reporting Precedence

The precedence of error reporting, and the reported values, shall be as follows:

1. SBP-2 errors

2. Unit Attention Condition. If an unit attention condition exists, the target shall report the following condition:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	6 – UNIT ATTENTION	29, 0 – RESET, OR DEVIC RESET OCCURRED

3. Data length not supported by protocol (>= 231bytes). For this error, the target shall report the following condition:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	49, 0 – INVALID PACKET SIZE

4. *command* not supported. For this error, the target shall report the following condition:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	26, 0 – INVALID COMMAND OPERATION CODE

5. If this command is issued to the wrong queue, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	21, 1 – INVALID ELEMENT ADDR

6. Command specific errors

7. Unknown or unspecified errors

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	0 – NO SENSE	0, 0 – NO ADDITIONAL SENSE INFORMATION



## 8. Commands

### 8.1. Command Owner

Each peer of the 1394 PWG Transport model may send commands. In this document, a peer which send a command in one transaction phase is called command owner. If a command owner is an initiator, the initiator shall provide input buffer to receive command execution result replied from the target, and the initiator shall set *notify* bit to one. Otherwise, if command owner is a target, then this target shall provide queue space at least large enough to receive expected reply from the initiator.

### 8.2. Initiator's responsibility

An initiator which is compliant for the 1394 PWG Transport model shall provide at least one outstanding input ORB to receive a command from the target.

### 8.3. Generic imaging profile command set

This command set includes general purpose commands to service bi-directional data transfer, transport open/close and transaction resource control.

Both the initiator and the target may open/close channels at any time. The table below lists generic image profile commands. The *queue* bit (abbreviated as q in the table below) specifies queue each command shall be stored.

Command	q	owner	description
TRSP_OPEN_INDICATION	1	Initiator	The initiator indicates to the target to open the channel to send initiator's commands or to listen target's commands
TRSP_CLOSE_INDICATION	1		The initiator request target to close an available channel
TRSP_OPEN_REQUEST	1	Target	The target requests the initiator to open the channel.
TRSP_CLOSE_REQUEST	1		The target requests the initiator to close an available channel
TRSP_I2T_DATA	0	Initiator	The target reads data from the buffer
TRSP_T2I_DATA	0		The target writes data to the buffer
T2I_DATA_REQUEST	1	Target	If no TRSP_T2I_DATA is given by the initiator, the target may invoke this request and solicit the initiator to put TRSP_T2I_DATA in the list.
NOP	1	Initiator	An initiator may place no operation code within a Command Block ORB <i>direction</i> bit is zero in the ORB list. In this case, a target shall consume ORB and shall reply Status Block immediately.
MONITOR	1		An initiator shall place a spare MONITOR command ORB with <i>direction</i> bit set to be one to receive any request from the target.
OPEN_REPLY	1	Initiator	An initiator shall reply status of open request. The status shall be stored the data buffer associated Command Block ORB.

CLOSE_REPLY	1	An initiator shall reply status of close request. The status shall be stored the data buffer associated Command Block ORB.
-------------	---	--

**Table 4 Generic Imaging Profile Commands**

#### 8.4. Printing Class specific command set

This optional command set provides services to ease higher layer protocol drivers to access a printing device medium. In this case, the command set assumes the printer is a target device. The table below lists printing class specific commands. All these commands shall be issued with *queue* bit to be one.

Command	owner	Description
GET_DEVICE_ID	initiator	An initiator device asks a printing device to reply 1284 PnP string.
SET_LEGACY_ENABLE		An initiator device asks a printing device medium to enable legacy port emulation. The printing device shall return port_status in a Status Block.
SOFT_RESET		An initiator device acts as if it assert nIntr signal line provided in conventional "Centronics-like" parallel port. if legacy mode is not enabled, this function does no effect. If legacy mode is enabled the printer shall cause soft reset.

**Table 5 Printing Class Specific Commands**

#### 8.5. Command Format

The 1394 PWG commands consists of an op-code, an op-qualifier and an operand. The command set provides three types of the command, one byte command type, one word command type and enhanced command type. The op-code and the op-qualifier field has eight bit length. The length of the operand field shall be specified by the op-qualifier field.

The figure shown below describes structure of the command.

1 byte (8 bit)	1 byte (8 bit)	0,1,2,...8 byte
op-code	op-qualifier	operand

**Figure 14 Command Format**

The command set is extensible. Any IHV may extend optional command set. The 1394 PWG is responsible to define generic imaging profile command set and primitive printing class specific command set.

##### 8.5.1. Bit field definition for op-code

The first byte of a command format shall specify command class, command owner, length attribute and command code. The table shows bit field description of the first byte.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Command class		owner bit	Length attribute	Command name label			

**Table 6 Bit field in the first byte of a command format**

The table below details meaning of the bit field.

Bit(s)	Field	Description
7 – 6	command class	10B = Printing Class Specific Command Set 11B = Image Profile Generic Command Set 01B = reserved for future standardization 00B = reserved for future standardization
5	owner bit	1 = Initiator indicates this command 0 = Target requests this command
4	length attribute	1 = command length is one byte, 0 = command length is specified in a next byte
3 – 0	command name label	000B – 111B / assigned for commands

**Table 7 Detail of each bit field**

If 4<sup>th</sup> bit of the op-code is set to one, the length of the command is one byte. Otherwise if 4<sup>th</sup> bit of the first byte of the command (op-code) is set to zero, length of the command is larger than 2 byte. Then op-qualifier shall specify the length of the command.

8.5.2. *Bit field description for op-qualifier*

Each bit field of an op-qualifier defines command characteristics, whether it has operand or not, or whether operand uses ORB field or not. The format of an op-qualifier is illustrated below.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
extension code			<i>data_descriptor</i> aware	Operand aware	operand length		

**Table 8 Bit field in the first byte of a command format**

If 4<sup>th</sup> bit of the op-code is zero, the 1394 PWG or any vendor may define extended commands. The value 0yyyy0zzB is reserved by the 1394 PWG. In here, “yyyy” means bit 3 to bit 0 of the op-code and “zz” means bit 6 to bit 5 of the op-qualifier. If a command uses *data\_descriptor* of the Command Block ORB, bit 4 shall be one. Also if a command uses extra bytes in command format, then bit 3 shall be one. In this case the length of the extra bytes is specified as value of “operand length” field plus one.

The table below summarize description of each field.

Bit(s)	Field	Description
7 – 5	extension code	This field extends “command name label” field of the op-code. Any organization or individual may define proprietary commands utilizing this attribute. 000 – 011B are reserved by the 1394 PWG 100 – 111B are provided for customize
4	<i>data_descriptor</i> aware	00B = this command has no operand 01B = this command has an operand from next byte 10B = this command has an operand and <i>data_descriptor</i> of the ORB is the operand
3	operand aware	11B = this command has two operand, one is stored in next bytes and the other is <i>data_descriptor</i> of the ORB.
2 – 0	operand length	The value of this field + 1 specifies the length of the operand

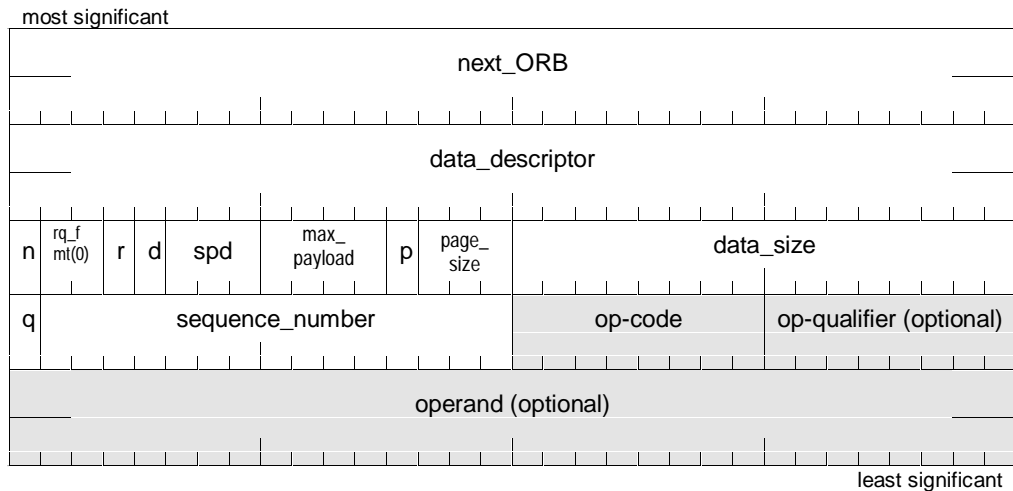
**Table 9 bit field description of the op-qualifier**

8.5.3. *Where shall the initiator and the target store commands*

The initiator shall store the command in the Command Block ORB. In this document, any command issued from the initiator and stored in the Command Block ORB is called as “indication”.

The target shall store the command in the data buffer associated with the Command Block ORB issued by the initiator with “NOP” command. (see section 5.3 “Generic imaging profile command set”). Any command delivered from the target is called as “request”.

The figure illustrated below defines default place holder of the “indication”. The *queue* bit (abbreviated as q in the figure below) specifies which queue shall the command to be stored. The *sequence\_number* field shall be required to keep in order execution for each queue.



**Figure 15 Default place holder of the indication**

The *next\_ORB*, *data\_descriptor*, *rq\_fmt*, *max\_payload*, *page\_size* and *lun* field and *n* (*notify*), *d* (*direction*) and *p* (*page\_table\_present*) bit are defined by SBP-2 specification.

The *op\_qualifier* and *operand* field is optional. The initiator may use these field when the length of a command is larger than one byte.

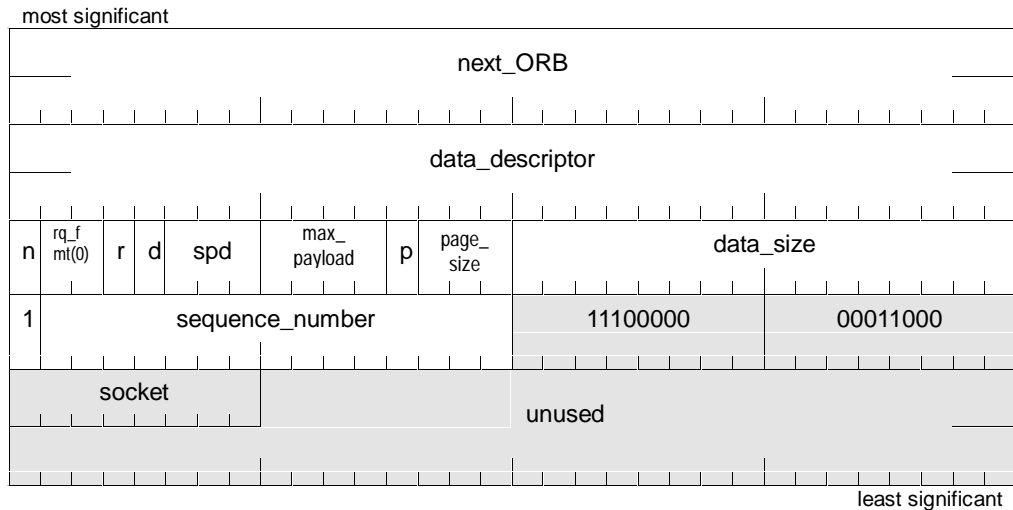
The target shall place the command in the data buffer associated with a Command Block ORB provided by the initiator with "NOP" op-code, *queue* bit set to be one and *direction* bit set to be one.

## 8.6. TRANSPORT\_OPEN\_INDICATIN

Encoded command label = 1110-0000-0001-1000 B (0xE018), each field if defined below.

Bit field	value	Description
Command class	11	Image Profile Generic command set
Owner	1	Initiator
Length attribute	0	Not a byte command
Command name label	0000	Open
Extension code	000	
<i>data_descriptor</i> aware	1	Use <i>data_descriptor</i> field of ORB
Operand aware	1	Use operand
Operand length	000	Operand length is one byte
Operand (1 byte)	any	specifies logical channel number

The direction bit (abbreviated as *d* in the figure below) and queue id shall be one. The initiator assumes the target will reply result in data buffer associated with *data\_descriptor* filed of the ORB.



**Figure 16 ORB structure containing TRANSPORT\_OPEN\_INDICATION**

The initiator shall specify one byte operand which specify the reference number of a socket. The socket is a facility consisted by an endpoint of the target. The initiator may use the socket as same as a logical channel, if necessary.

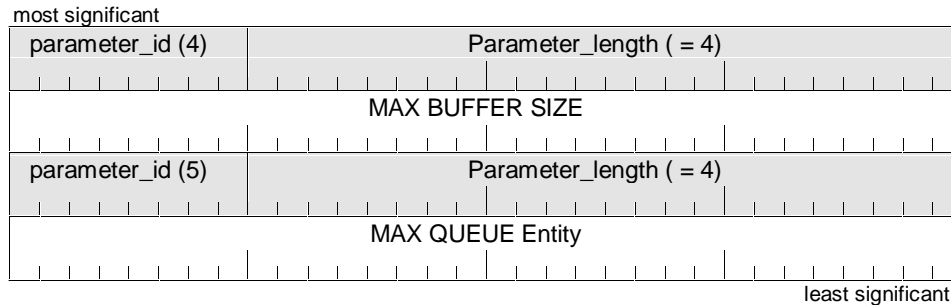
### 8.6.1. TRANSPORT\_OPEN\_INDICATION response

An initiator and a target may open or close socket dynamically. And the counter peer (a target or an initiator) shall report available resource for the requested socket.

The figure below shows a parameter encoding (see previous section) which contains two independent read only parameter MAX BUFFER SIZE and MAX QUEUE Entity.

MAX BUFFER SIZE indicates maximum data buffer space available in the target for next session. It means if the initiator provides data smaller amount than this value, all data will be consumed without stalling of target agent.

MAX QUEUE Entity specifies rest of pre-fetch queue for data session. It means if the initiator lists smaller numbers of ORBs in the ORB-list, all ORBs will be safely pre-fetched by the fetch agent of the target device medium.



**Figure 17 Reply for TRANSPORT\_OPEN\_INDICATION**

The Status\_Block specified by SBP-2 specification shall be used to report transaction status. The command response shall confirm to the format described in clause 5.1.

The *sfmt* field shall contain a value of zero (0).

If this command completes successfully, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	0 – GOOD	0 – NO SENSE	0,0 – NO ADDITIONAL SENSE TO REPORT

Else if this command contains an invalid field in the CDB, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	24, 0 – INVALID FIELD IN CDB

Else if this command is issued for the socket already opened, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	4A, 0 – COMMAND PHASE ERROR

Else if this command is issued with a NULL data descriptor, then the target shall return status of:

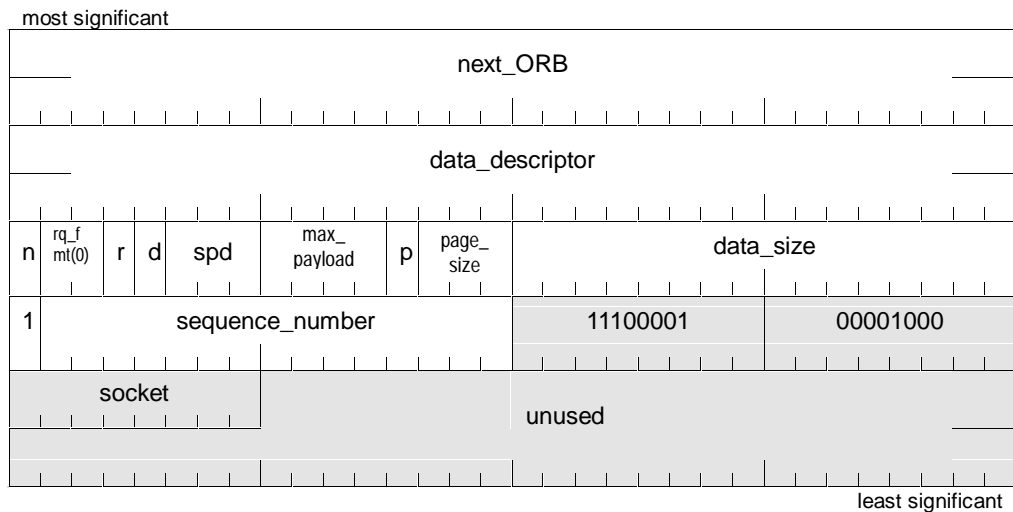
Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	24, 0 – INVALID FIELD IN CDB

## 8.7. TRANSPORT\_CLOSE\_INDICATION

Encoded command label = 1110-0001-0000-1000 B (0xE108), each field if defined below.

Bit field	value	Description
Command class	11	Image Profile Generic command set
Owner	1	Initiator
Length attribute	0	Not a byte command
Command name label	0001	Close
Extension code	000	
<i>data_descriptor</i> aware	0	Not use <i>data_descriptor</i> field of ORB
Operand aware	1	Use operand
Operand length	000	Operand length is one byte
Operand (1 byte)	any	specifies logical channel number

The direction bit (abbreviated as *d* in the figure below) and queue id shall be one. The initiator shall specify one byte operand which specify the reference number of a socket.



**Figure 18 ORB structure containing TRANSPORT\_CLOSE\_INDICATION**

The initiator does not assume the target replies result to data buffer specified by CDB.

### 8.7.1. TRANSPORT\_CLOSE\_INDICATION response

The command response shall confirm to the format described in clause 5.1.

The *sfmt* field shall contain a value of zero (0).

If this command completes successfully, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	0 – GOOD	0 – NO SENSE	0,0 – NO ADDITIONAL SENSE TO REPORT



Else if this command issued with socket reference number which is not opened, then the target shall return status of:

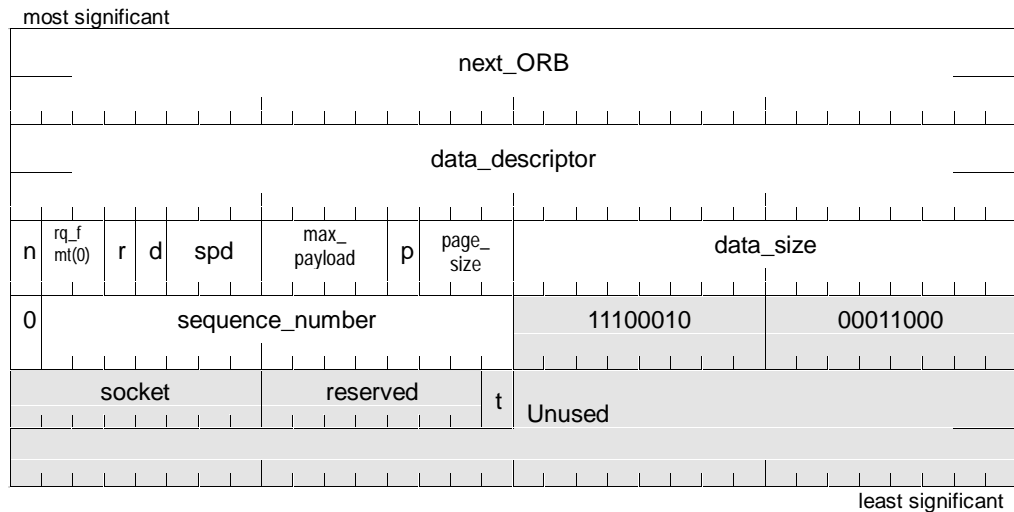
Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	24, 0 – INVALID FIELD IN CDB

## 8.8. TRANSPORT\_I2T\_DATA

Encoded command label = 1110-0010-0001-1000 B (0xE219), each field if defined below.

Bit field	value	Description
Command class	11	Image Profile Generic command set
Owner	1	Initiator
Length attribute	0	Not a byte command
Command name label	0010	I2T_data
Extension code	000	
<i>data_descriptor</i> aware	1	Use <i>data_descriptor</i> field of ORB
Operand aware	1	Use operand
Operand length	001	Operand length is two byte
Operand (2 byte)	Any	Socket reference number and tag bit

All field shall confirm to the description in the Command Block ORB Format clause. Data length of zero (0) bytes are valid and shall complete without data being transferred. The direction bit (abbreviated as *d* in the figure below) shall be **zero**. The queue id shall be **zero**. The initiator shall specify one byte operand which specify the reference number of a socket.



**Figure 19 ORB structure containing TRANSPORT\_I2T\_DATA**

The *tag* bit (abbreviated *t* in the figure above) specifies that the data in the buffer pointed to by the *data\_descriptor* field is to be treated separate from the rest of the data stream.

If the command fails, the target shall retain no data referenced by this command.

### 8.8.1. TRANSPORT\_I2T\_DATA response

The command response shall confirm to the format described in clause 5.1.

The *sfmt* field shall contain a value of zero (0).

If this command completes successfully, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	0 – GOOD	0 – NO SENSE	0,0 – NO ADDITIONAL SENSE TO REPORT

Else if this command issued with socket reference number which is not opened, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	24, 0 – INVALID FIELD IN CDB

Else if this command issued before the connection I sopen, then the target shall return status of:

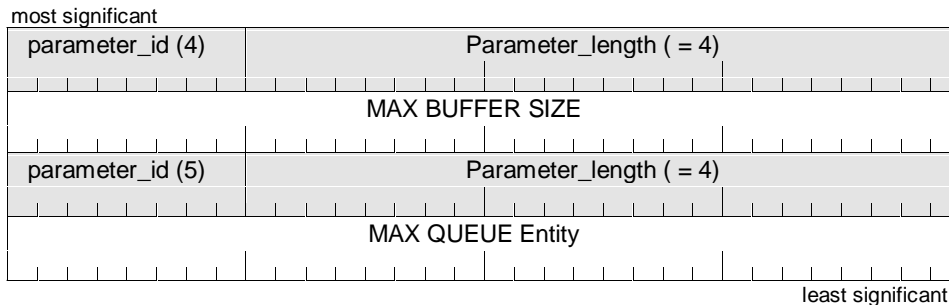
Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	4A, 0 – COMMAND PHASE ERROR

Else if this socket is closed then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	0, 5 – END OF DATA DETECTED

8.8.2. *Optional field for LEGACY MODE*

When the LEGACY MODE is enabled, the target shall report two parameters illustrated below. These parameters are enclosed in the optional field #1 and #2 of Status\_Block data structure. (See 5.2 Legacy Mode Status Block)



**Figure 20 Optional field for LEGACY MODE**

MAX BUFFER SIZE indicates maximum data buffer space available in the target for next session. It means if the initiator provides data smaller amount than this value, all data will be consumed without stalling of target agent.

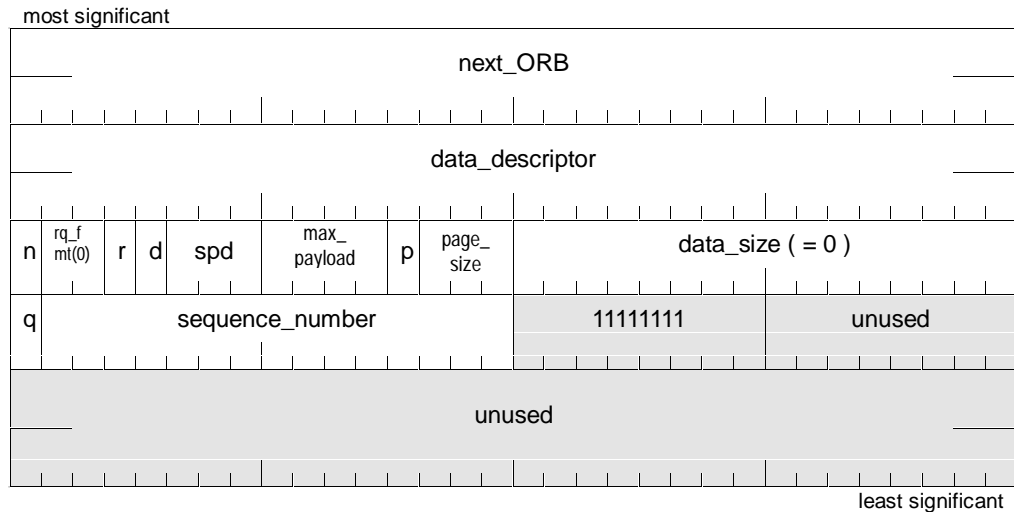
MAX QUEUE Entity specifies rest of pre-fetch queue for data session. It means if the initiator lists smaller numbers of ORBs in the ORB-list, all ORBs will be safely pre-fetched by the fetch agent of the target device medium.

## 8.9. NOP

Encoded command label = 11111111B (0xFF), each field if defined below.

Bit field	value	Description
Command class	11	Image Profile Generic command set
Owner	1	Initiator
Length attribute	1	is a byte command
Command name label	1111	NOP
Extension code	---	---
<i>data_descriptor</i> aware	---	---
Operand aware	---	---
Operand length	---	---
Operand (1 byte)	---	---

An initiator device may invoke NOP command just to receive a Status Block from the target. In any case the queue id (abbreviated as *q* in the figure below) shall be set to **one**. the direction bit (abbreviated as *d* in the figure below) shall be set to **zero**.



**Figure 21 ORB structure containing NOP command**

This command actually cause a same effect as TRANSPORT\_I2T\_DATA with data length **zero**. But in this case queue id is one, thus the command is fetched by the pre-fetch queue for command processing. And the target device will consume it immediately, because this command requires nothing to do.

If the legacy mode enabled, the initiator device can get *Port\_Status* from the target utilizing this command.

### 8.9.1. NOP response

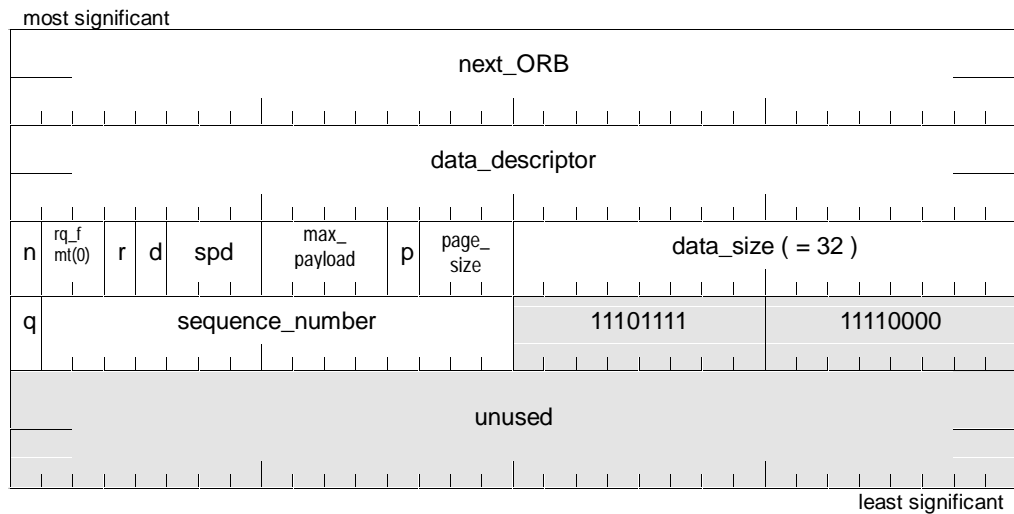
See clause for "TRANSPORT\_I2T\_DATA".

### 8.10. Monitor

Encoded command label = 1110-1111-1111-0000B (0xEFF0), each field if defined below.

Bit field	value	Description
Command class	11	Image Profile Generic command set
Owner	1	Initiator
Length attribute	0	Not a byte command
Command name label	1111	MONITOR
Extension code	111	
<i>data_descriptor</i> aware	1	Use <i>data_descriptor</i> field of ORB
Operand aware	0	Not use operand
Operand length	000	Operand length is useless
Operand (1 byte)	no	

The queue id (abbreviated as *q* in the figure below) shall be set to **one**. The direction bit (abbreviated as *d* in the figure below) shall be set to **one**.



**Figure 22 ORB structure containing MONITOR command**

The initiator shall provide at least one spare MONITOR command in the ORB list. It shall specify 32 bytes *data\_size* and shall provide data buffer which has 32 bytes length.

The target may complete MONITOR command immediately without any reply message when it does not have any request to the initiator. Or the target device may reply message witch contains a request to the initiator.

When a MONITOR command completed, the initiator shall place new MONITOR command in the ORB list as soon as possible.

#### 8.10.1. MONITOR response

The command response shall confirm to the format described in clause 5.1.

The *sfmt* field shall contain a value of zero (0).

If this command completes successfully, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	0 – GOOD	0 – NO SENSE	0,0 – NO ADDITIONAL SENSE TO REPORT

Else if this command contains an invalid field in the CDB, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	24, 0 – INVALID FIELD IN CDB

Else if this command issued before the connection I sopen, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	4A, 0 – COMMAND PHASE ERROR

Else if this socket is closed then the target shall return status of:

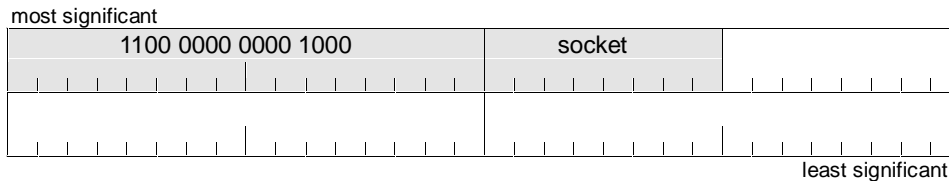
Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	0, 5 – END OF DATA DETECTED

### 8.11. TRANSPORT\_OPEN\_REQUEST

Encoded command label = 1100-0000-0000-1000 B (0xC008) , each field if defined below.

Bit field	value	Description
Command class	11	Image Profile Generic command set
Owner	0	Target
Length attribute	0	Not a byte command
Command name label	0000	Open
Extension code	000	
<i>data_descriptor</i> aware	0	Not Use <i>data_descriptor</i> field of ORB
Operand aware	1	Use operand
Operand length	000	Operand length is one byte
Operand (1 byte)	any	Specifies logical channel number

The target device may send TRANSPORT\_OPEN\_REQUEST at any time. This request shall be stored in the buffer space specified by a MONITOR command ORB. This command has an one byte operand which specify a reference number for a socket.



**Figure 23 Data format of TRANSPORT\_OPEN\_REQUEST**

#### 8.11.1. TRANSPORT\_OPEN\_REQUEST response

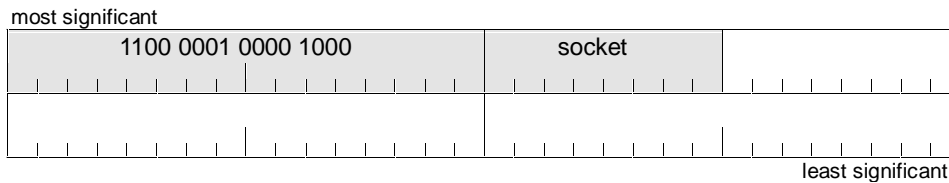
The initiator shall reply response utilizing a Command Block ORB described in clause 8.15 OPEN\_REQUEST\_REPLY.

## 8.12. TRANSPORT\_CLOSE\_REQUEST

Encoded command label = 1100-0001-0000-1000 B (0xC108), each field if defined below.

Bit field	value	Description
Command class	11	Image Profile Generic command set
Owner	0	Target
Length attribute	0	Not a byte command
Command name label	0001	Close
Extension code	000	
<i>data_descriptor</i> aware	0	Not use <i>data_descriptor</i> field of ORB
Operand aware	1	Use operand
Operand length	000	Operand length is one byte
Operand (1 byte)	any	Specifies logical channel number

The target device may send TRANSPORT\_CLOSE REQUEST at any time. This request shall be stored in the buffer space specified by a MONITOR command ORB. This command has an one byte operand which specify a reference number for a socket.



**Figure 24 Data format of TRANSPORT\_COSE\_REQUEST**

### 8.12.1. TRANSPORT\_CLOSE\_REQUEST response

The initiator shall reply response utilizing a Command Block ORB described in clause 8.16 CLOSE\_REQUEST\_REPLY.

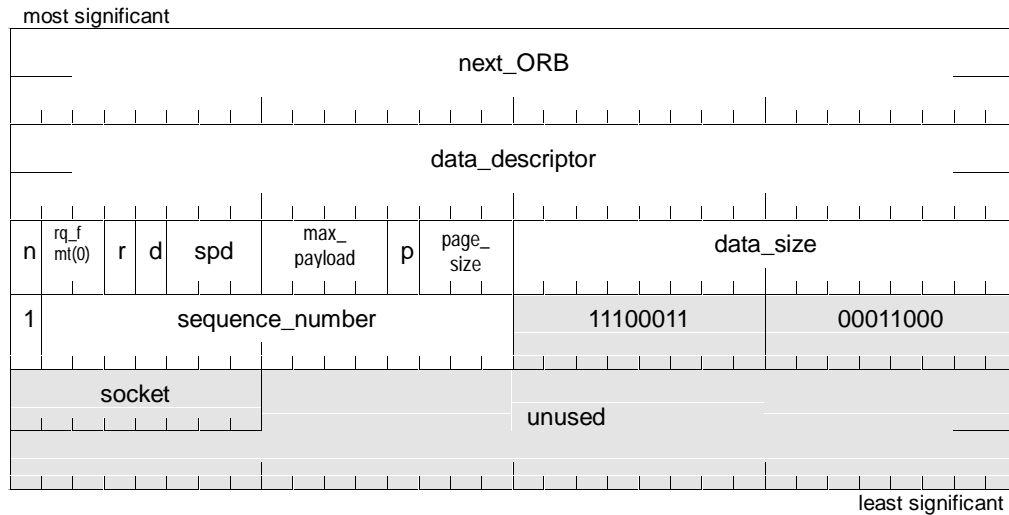


### 8.13. TRANSPORT\_T2I\_DATA

Encoded command label = 1110-0011-0001-0000 B (0xE318), each field if defined below.

Bit field	value	Description
Command class	11	Image Profile Generic command set
Owner	1	initiator
Length attribute	0	Not a byte command
Command name label	0011	T2I_data
Extension code	000	
<i>data_descriptor</i> aware	1	Use <i>data_descriptor</i> field of ORB
Operand aware	1	use operand
Operand length	000	Operand length is one byte
Operand (1 byte)	no	Not use operand

All field shall confirm to the description in the Command Block ORB Format clause. The direction bit (abbreviated as *d* in the figure below) shall be **one**. The queue id shall be **zero**. The initiator shall specify one byte operand which specify the reference number of a socket.



**Figure 25 ORB structure containing TRANSPORT\_T2I\_DATA**

#### 8.13.1. TRANSPORT\_T2I\_DATA response

The command response shall confirm to the format described in clause 5.1 Status Block for normal session.

The *tag* bit shall specify whether or not this data is distinct from the rest of the data stream. If the data block length is zero, this field is reserved.

For message-based communication, the *residual* field shall indicate either the amount of residual buffer space provided by the command, or the additional amount of data within this message which could not be sent (using twos-complement notation).

For stream-based communication, the *residual* field shall only indicate the amount of unused space in the buffer provided. The target shall not indicate the presence of additional data. Additional data shall be retained for a successive command. A target should complete the command as soon as data is available.

The *sfmt* field shall contain a value of zero (0).

If this command completes successfully, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	0 – GOOD	0 – NO SENSE	0,0 – NO ADDITIONAL SENSE TO REPORT

Else if this command issued with socket reference number which is not opened, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	24, 0 – INVALID FIELD IN CDB

Else if this command issued before the connection is open, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	4A, 0 – COMMAND PHASE ERROR

Else if this socket is closed then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	0, 5 – END OF DATA DETECTED

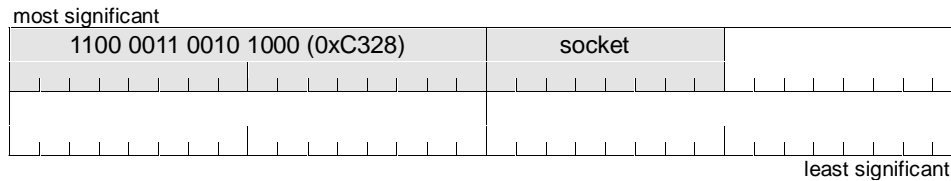
### 8.14. T2I\_DATA\_REQUEST

Encoded command label = 1100-0011-0010-1000 B (0xC328), each field if defined below.

Bit field	value	Description
Command class	11	Image Profile Generic command set
Owner	0	target
Length attribute	0	Not a byte command
Command name label	0011	T2I_data_Request
Extension code	001	
<i>data_descriptor</i> aware	0	Not use <i>data_descriptor</i> field of ORB
Operand aware	1	use operand
Operand length	000	Operand length is one byte
Operand (1 byte)	any	Specified Socket number

Normally this command is useless. However, if no TRSP\_T2I\_DATA is given by the initiator, the target may invoke this request and solicit the initiator to put TRSP\_T2I\_DATA in the list.

Data structure of the command is illustrated below.



**Figure 26 Data format of T2I\_DATA\_REQUEST**

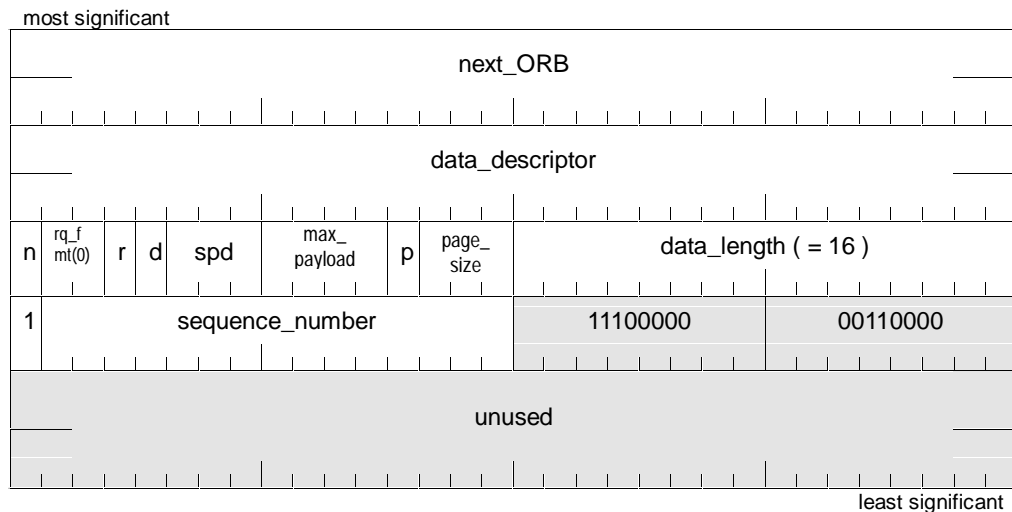
This command shall be stored in the data buffer associated with a *data\_descriptor* field of an input ORB issued by the initiator with NOP op-code.

## 8.15. OPEN\_REQUEST\_REPLY

Encoded command label = 1110-0000-0011-0000 B (0xE030), each field if defined below.

Bit field	value	Description
Command class	11	Image Profile Generic command set
Owner	1	Initiator
Length attribute	0	Not a byte command
Command name label	0000	Open_request_reply
Extension code	001	
<i>data_descriptor</i> aware	1	use <i>data_descriptor</i> field of ORB
Operand aware	0	Not use operand
Operand length	000	Operand length is useless
Operand (1 byte)	no	no

All field shall confirm to the description in the Command Block ORB Format clause. The direction bit (abbreviated as *d* in the figure below) shall be **zero**. The queue id shall be **one**.



**Figure 27 ORB structure containing OPEN\_REQUEST\_REPLY**

The initiator device shall report status for TRANSPORT\_OPEN\_REQUEST command from the target. All encoded parameters shall be written in the data buffer specified by the *data\_descriptor*, and the length of these parameters are defined by the *data\_length* field of CDB. For this command *data\_length* is 16 bytes. Contents of each parameter are described in the next section.

### 8.15.1. OPEN\_REQUEST\_REPLY response

The command response shall confirm to the format described in clause 5.1.

The *sfmt* field shall contain a value of zero (0).

If this command completes successfully, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	0 – GOOD	0 – NO SENSE	0,0 – NO ADDITIONAL SENSE TO REPORT

Else if this command contains an invalid field in the CDB, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	24, 0 – INVALID FIELD IN CDB

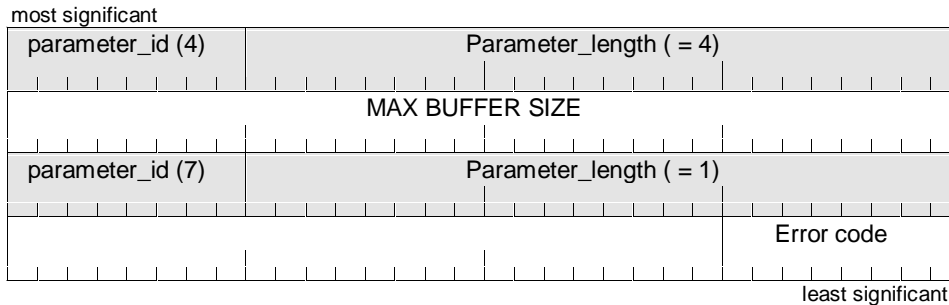
Else if this command contains an invalid parameter encoding, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	1A, 0 – PARAMETER LIST LENGTH ERROR

Else if this command contains an unsupported parameter id, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	26, 1 – PARAMETER NOT SUPPORTED

The initiator shall write encoded parameters in data buffer associated with *data\_descriptor* field of ORB. The figure shown below shows two parameters.



**Figure 28 Reply parameter in OPEN\_REQUEST\_REPLY response**

MAX BUFFER SIZE indicates maximum data buffer space available in the initiator for next session. The initiator device continuously will be able to issue `TRANSPORT_T2I_DATA` until the target had written up to MAX BUFFER SIZE bytes.

ERROR CODE shows status of the initiator.

Error code	Description
0	No error / successfully competed

1	Resource is out of service
2	Unsupported command
3	Invalid parameter

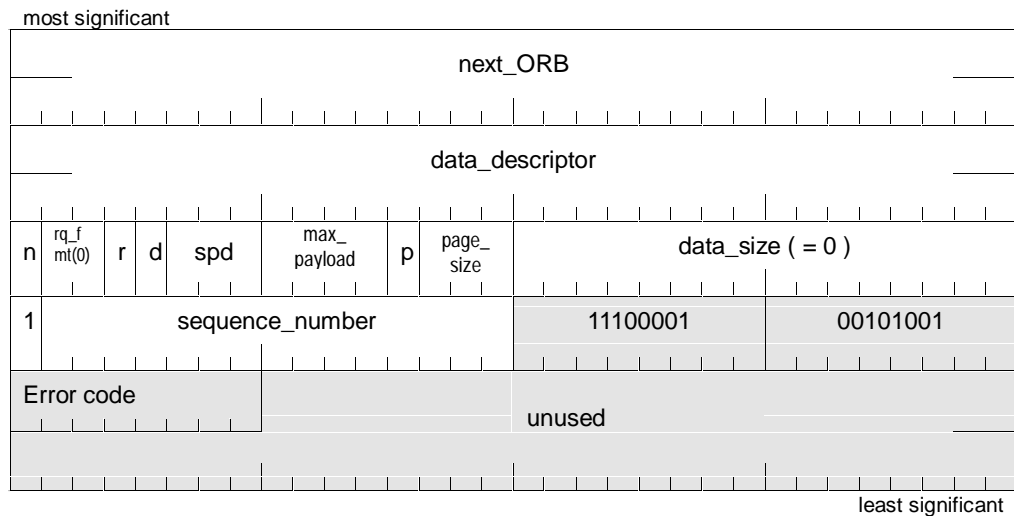
**Table 10 Error code**

## 8.16. CLOSE\_REQUEST\_REPLY

Encoded command label = 1110-0001-0010-1001 B (0xE129), each field if defined below.

Bit field	value	Description
Command class	11	Image Profile Generic command set
Owner	1	Initiator
Length attribute	0	Not a byte command
Command name label	0001	Close_request_reply
Extension code	001	
<i>data_descriptor</i> aware	0	Not use <i>data_descriptor</i> field of ORB
Operand aware	1	use operand
Operand length	001	Operand length is 2 bytes
Operand (2 byte)	specified	See description below

All field shall confirm to the description in the Command Block ORB Format clause. The direction bit (abbreviated as *d* in the figure below) shall be **zero**. The queue id shall be **one**.



**Figure 29 ORB structure containing CLOSE\_REQUEST\_REPLY**

The initiator device shall report status for TRANSPORT\_CLOSE\_REQUEST command issued from the target. No encoded parameter is written in the data buffer specified by the *data\_descriptor*. Thus the *data\_length* field of CDB shall have a value **zero**.

Error code has a value listed in Table 10 in previous clause.

### 8.16.1. CLOSE\_REQUEST\_REPLY response

The command response shall confirm to the format described in clause 5.1.

The *sfmt* field shall contain a value of zero (0).

If this command completes successfully, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	0 – GOOD	0 – NO SENSE	0,0 – NO ADDITIONAL SENSE TO REPORT

Else if this command contains an invalid field in the CDB, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	24, 0 – INVALID FIELD IN CDB

Else if this command contains an invalid parameter encoding, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	1A, 0 – PARAMETER LIST LENGTH ERROR

Else if this command contains an unsupported parameter id, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	26, 1 – PARAMETER NOT SUPPORTED



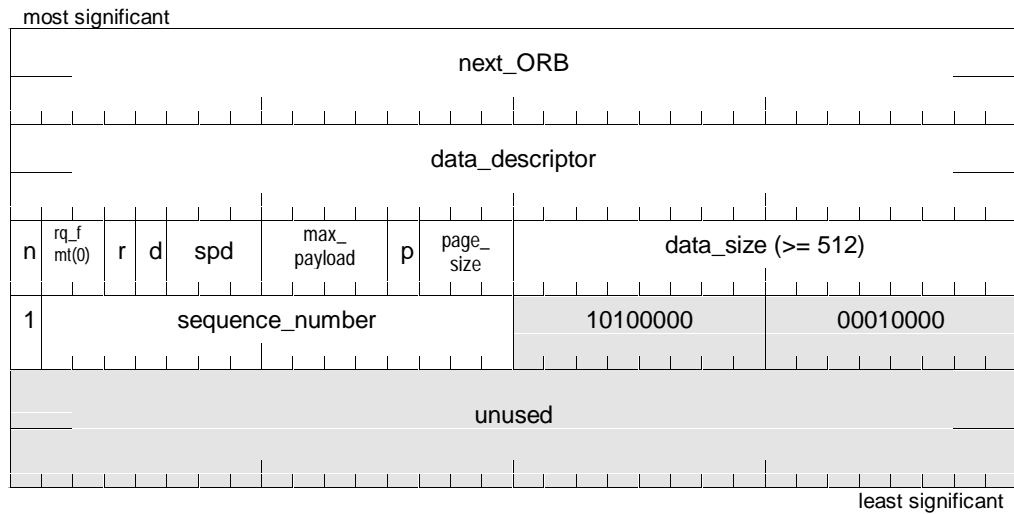
### 8.17. GET\_DEVICE\_ID

Encoded command label = 1010-0000-0001-0000B (0xA010), each field if defined below.

Bit field	value	Description
Command class	10	Printing Class Specific command set
Owner	1	Initiator
Length attribute	0	Not a byte command
Command name label	0000	Get_Device_ID
Extension code	000	
<i>data_descriptor</i> aware	1	use <i>data_descriptor</i> field of ORB
Operand aware	0	Not use operand
Operand length	000	Operand length is useless
Operand (1 byte)	no	Not use operand

This command is printer class specific.

All field shall confirm to the description in the Command Block ORB Format clause. The direction bit (abbreviated as *d* in the figure below) shall be **one**. The queue id shall be **zero**.



**Figure 30 ORB structure containing GET\_DEVICE\_ID**

The initiator assumes the target returns 1284 PnP strings which contains ASCII code and UNICODE. Thus the initiator shall provide enough data buffer to receive these strings.

The higher level printing class driver on the initiator shall not read the configuration ROM of the target device directly. Direct 1394 Bus access shall be done by the 1394 Bus System driver.

#### 8.17.1. GET\_DEVICE\_ID response

The command response shall confirm to the format described in clause 5.1.

The *sfmt* field shall contain a value of zero (0).

If this command completes successfully, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	0 – GOOD	0 – NO SENSE	0,0 – NO ADDITIONAL SENSE TO REPORT

Else if this command contains an invalid field in the CDB, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	24, 0 – INVALID FIELD IN CDB

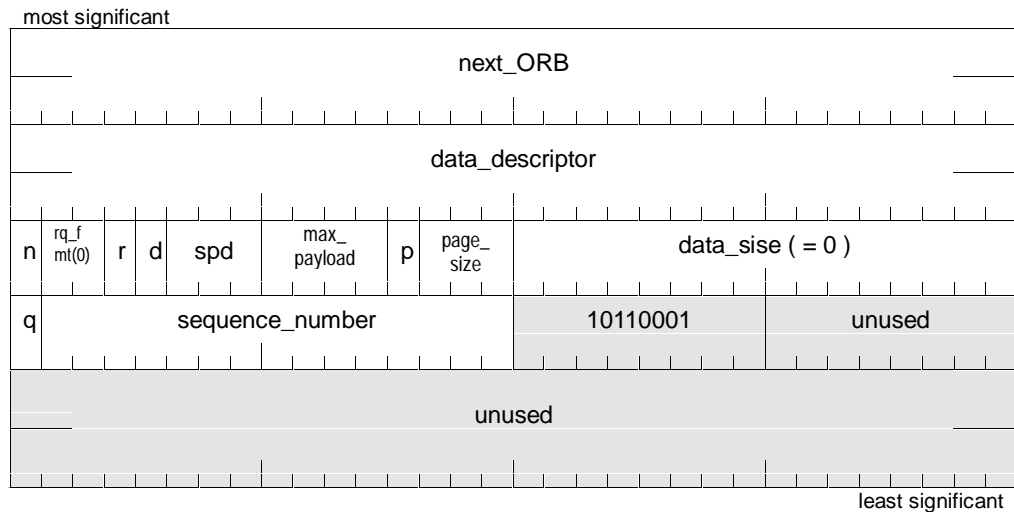
### 8.18.SET\_LEGACY\_ENABLE

Encoded command label = 10110001B (0xB1), each field if defined below.

Bit field	value	Description
Command class	10	Printing Class specific command set
Owner	1	Initiator
Length attribute	1	is a byte command
Command name label	0001	SET_LEGACY_ENABLE
Extension code	---	---
<i>data_descriptor</i> aware	---	---
Operand aware	---	---
Operand length	---	---
Operand (1 byte)	---	---

This command is printer class specific.

All field shall confirm to the description in the Command Block ORB Format clause. The direction bit (abbreviated as *d* in the figure below) shall be **one**. The queue id shall be **one**.



**Figure 31 ORB structure containing SET\_LEGACY\_ENABLE**

The initiator device does not assume the target device reply the message for this command. The target device may not support this command. If the target device support this command, it shall cause internal mode change and after this command completed, the initiator shall reply LEGACY MODE Status Bock discussed in clause 5.2.

One the LEGACY MODE is enabled, this mode shall not be cleared until the initiator device issues LOGOUT command and this command completed successfully.

#### 8.18.1. SET\_LEGACY\_ENABLE response

The command response shall confirm to the format described in clause 5.1.

The *sfmt* field shall contain a value of zero (0).

If this command completes successfully, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	0 – GOOD	0 – NO SENSE	0,0 – NO ADDITIONAL SENSE TO REPORT

Else if this command contains an invalid field in the CDB, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	24, 0 – INVALID FIELD IN CDB

Else if this command is not supported, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	26, 0 – INVALID COMMAND OPERATION CODE

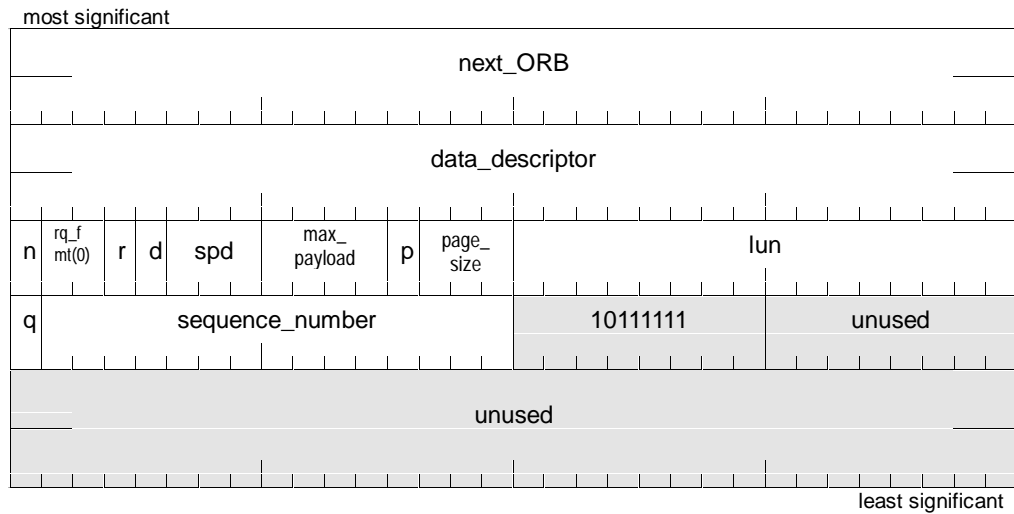
### 8.19. SOFT\_RESET

Encoded command label = 10111111B (0xBF), each field if defined below.

Bit field	Value	Description
Command class	10	Printing Class specific command set
Owner	1	Initiator
Length attribute	1	is a byte command
Command name label	1111	Reset
Extension code	---	---
<i>data_descriptor</i> aware	---	---
Operand aware	---	---
Operand length	---	---
Operand (1 byte)	---	---

This command is printing class specific.

All field shall confirm to the description in the Command Block ORB Format clause. The direction bit (abbreviated as *d* in the figure below) shall be **zero**. The queue id shall be **one**.



**Figure 32 ORB structure containing SOFT\_RESET**

The initiator does not assume the target replying messages. The target device may not support this command. If the target support this command, the target shall cause internal software reset to emulate “nInit” signal historically supported in the parallel port printer.

#### 8.19.1. SOFT\_RESET response

The command response shall confirm to the format described in clause 5.1.

The *sfmt* field shall contain a value of zero (0).

If this command completes successfully, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	0 – GOOD	0 – NO SENSE	0,0 – NO ADDITIONAL SENSE TO REPORT

Else if this command contains an invalid field in the CDB, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	24, 0 – INVALID FIELD IN CDB

Elise is this command is not supported, then the target shall return status of:

Resp	Sbp_status	Status	Sense key	ASC, ASQ
0 – REQ COMPLETE	0 – NO ADDL INFO	2 – CHECK COND	5 – ILLEG REQ	26, 0 – INVALID COMMAND OPERATION CODE

## 8.20. Commands Summary

Command	Command Class	Owner	Length Attribute	Command Name Label	Extension Code	Data_descriptor Aware	Operand Aware	Operand length	HEX code
TRANSPORT_OPEN_INDICATION	11	1	0	0000	000	1	1	000	0xE018
TRANSPORT_CLOSE_INIDICATIN	11	1	0	0001	000	0	1	000	0xE108
TRANSPORT_I2T_DATA	11	1	0	0010	000	1	1	001	0xE219
NOP	11	1	1	1111					0xFF
MONITOR	11	1	0	1111	111	1	0	000	0xEFF0
TRANSPORT_OPEN_REQUEST	11	0	0	0000	000	0	1	000	0xC008
TRANSPORT_CLOSE_REQUEST	11	0	0	0001	000	0	1	000	0xC108
TRANSPORT_T2I_DATA	11	1	0	0011	000	1	1	000	0xE318
T2I_DATA_REQUEST	11	0	0	0011	001	0	1	000	0xC328
OPEN_REQUEST_REPLY	11	1	0	0000	001	1	0	000	0xE030
CLOSE_TRANSPORT_REPLY	11	1	0	0001	001	0	1	001	0xE129
GET_DEVICE_ID	10	1	0	0000	000	1	0	000	0xA010
SET_LEGACY_ENABLE	10	1	1	0001					0xB1
SOFT_RESET	10	1	1	1111					0xBF

**Figure 33 Commands summary**

Generic Imaging Profile commands are mandatory requirement. If a target device is a printer, it may support Printing Class specific commands optionally. Or any IHV may externally define additional command set to service a scanner or a fax facility as a target device. In this case, command set shall have 01B in highest bit field of op-code byte to specify class specific command set. Or if any printer has optional functionality, it may provide extended command set. In this case, this command set shall have 10B in highest bit field of op-code byte to show commands are Printing Class specific.

## **9. Issues**

### **9.1. Login**

**Issue:** Microsoft has implemented their SBP-2 driver to do a login on power up and not logout until the PC is shut down. Though this is provided for within the SBP-2 specification, it requires devices, which may be shared among multiple PCs to support multiple logins to a single service, even if the different PCs cannot simultaneously use the service.

Can imaging devices (which want to take advantage of the shared nature of 1394) tolerate the resource requirements to operate on a multiple Microsoft O/S host configuration?

**Status:** Closed, 8/17/98 - Toronto. Microsoft Implementations will not exhibit this behavior for image class devices.

### **9.2. Service Discovery**

**Issue:** 8/17/98 - Toronto. Transport client command set information?

**Status:** Open.

### **9.3. Data packets vs. Data stream communications model?**

**Issue:** 8/17/98 - Toronto. Brian to write up.

**Status:** Open.

### **9.4. Function Device Type Number in Unit Directory.**

**Issue:** 8/17/98 - Toronto. Transport client class of service? (Printing vs. Scanning vs. ???)

**Status:** Open.

### **9.5. Maximum data size**

**Issue:** Current proposal is 1 Mega byte. Discussion?

**Status:** Closed, 8/17/98 - Toronto. See command set in this document.

### **9.6. Unsolicited Status Register Enable**

**Issue:** Need policy for Initiator to always do this ASAP.

**Status:** Closed, 8/17/98 - Toronto. Have not identified a need for Unsolicited status.

### **9.7. Target Logout**

**Issue:** How does a Target Logout?

**Status:** In Progress, 8/17/98 - Toronto. Decided to send completion status to ORB in Target\_2\_Initiator direction indicating that the Target requests a Logout. If Initiator does not



to perform Logout in XX time.  
Target will reset context for that Initiator effecting a Logout.

## 9.8. Timers

**Issue:** Need explanation of behavior at initialization and reset.

1) Reconnect Timer - This one is taken care of by SBP-2 as part of the Login process.

2) Management ORB Timer - When the Initiator writes a Management ORB address to the Management Agent Register, how long should it wait for a response to be written to the management ORBs status fifo?

How is the timeout value determined and communicated?

3) Abort Task Set Timer - An Abort Task Set may issued by an Initiator in response to Target Inactivity.

How is the timeout value determined and communicated?

**Status:** Open.

## 9.9. Plug & Play Support

**Issue:** Should the discovery section contain information about Plug-N-Play support?

**Status:** Open.

**Proposal:** The target shall have provision to return 1284 PnP string when the initiator requested to send this string. The initiator assumes the target returns 1284 PnP strings which contains ASCII code and UNICODE. Thus the initiator shall provide enough data buffer to receive these strings.

The higher level printing class driver on the initiator shall not read the configuration ROM of the target device directly. Direct 1394 Bus access shall be done by the 1394 Bus System driver.

## 9.10. Non Symmetric Connection.

**Issue:** 8/17/98 - Toronto. A single transport\_open command requires a two transport\_close commands, one for each direction.

**Status:** Open?

**Proposal:** The 1394 PWG transport model shall provide symmetric connection utilizing facility described in this document. The target may issue TRANSPORT\_OPEN\_REQUEST, at anytime. Also the target may issue TRANSPORT\_CLOSE\_REQUEST when required session phase has completed and if close command is desired.

The target device may start data transferring utilizing TRANSPORT\_T2I\_DATA. If the initiator does not provide this facility, then the target may request the initiator to make transaction resource. In this case, the target shall issue T2I\_DATA\_REQUEST.

### 9.11. Transport Capabilities Negotiation Resource Commitment.

**Issue:** 8/17/98 - Toronto The transport capabilities command requires target to commit resources that may not be immediately followed by a transport\_open command. How long should the Target reserve these resources?

**Status:** Open.

### 9.12. Notify Status

**Issue:** In section 9.3 SBP-2 gives us the following:

NOTE - For targets that support the ordered model of task execution, the return of completion status for an ORB implicitly indicates that all preceding ORB's in the linked list have completed successfully, are no longer part of the task set and that the initiator may reuse or de-allocate their system memory.

Since we are using an un-ordered model does the implied status apply to all ORBS like in the ordered model or only to all the preceding ORBs in the same (Read or Write) direction?

**Status:** Open. Proposal: Only to those in the same direction.

### 9.13. Target Reset

**Issue:** One of the areas still requiring clarification is the mapping of transport and target transport clients into SBP-2 Units and Logical Units. This is an area where the SCSI roots of SBP-2 have really impacted our efforts.

Looking at the draft SCSI-3 ARCHITECTURE MODEL -2 (SAM-2) rev 0.3, I (who ??) find that "Each SCSI-3 protocol standard shall specify the response to a target reset event including the conditions under which a target hard reset shall be executed.

Looking at SBP-2, rev 4, we find TARGET RESET affects all logged-in initiators (to any logical unit within the Unit).

Some questions which need to be answered are:

- a) How is TARGET RESET used by drivers in commercial operating systems? (Both non-embedded and embedded?)
- b) Can we define how broadly (or narrowly) the TARGET RESET event affects a device using our transport?  
If so, how should it be defined?  
If not, does this mean that we must use one Unit directory for each possible instance of a service?

**Status:** Open.

### 9.14. Multiple Unit Directories

**Issue:** Multiple Unit Directories that are the same. How can we identify them to be unique?

Example was after a bus reset.

1) What I can't recall is why we have these multiple identical udirs in the first place.

2) Also if we do have them, it seems like they would need to be unique independent of bus reset.

**Status:** Open.

## Appendix A. Example Configuration ROM of the 1394 PWG printer

The 1394 PWG is still developing a specification for the 1394 PWG printer. *Command\_set\_spec\_id* field, *command\_set* field and *command\_set\_version* field shall contain values define by the 1394 PWG. The figure shown below is giving an example for the printer configuration ROM.

Block	offset	Description			
First Quadlet	400	Info_kength 04h	CRC_length 5eh	CRC_value cd53h	
Bus_info Block	404	'1394' in ASCII 31h 33h 39h 34h			
	408	0 1 0 1 reserved 0	cyc_clk_acc 32h	max_rec 70h	reserved 00h
	40C	node_vender_id 000048h			chip_id_hi FFh
	410	chip_id_lo 00000001h			
Root Directory	414	directory_length 0004h	directory_crc 34A7h		
	418	module_vender_id key 03h	Module_vender_id 000048h		
	41C	Node_capabilities Key 0Ch	Node_capabilities 008380h		
	420	Node_unique_id key 8Dh	Node_unique_id_leaf offset 000002h		
	424	Unit_directory Key D1h	Unit_directory No1 000004h		
Node Unique ID Leaf	428	Length 0002h	CRC 2DDCh		
	42C	node_vender_id 000048h			chip_id_hi FFh
	430	chip_id_lo 00000001h			

**Continued**

Block	offset	Description			
Unit Directory No1	434	Unit_directort_length 0009h	Unit_directory_crc AA02h		
	438	Unit_spec_id key 12h	Unit_spec_id 00609Eh		
	43C	Unit_sw_version key 13h	Unit_sw_version 010483h		
	440	Command_set_spec_id key 38h	Command_set_spec_id xxxxxxh		
	444	Command_set key 39h	Command_set yyyyyyh		
	448	Command_set_version key 3Bh	Command_set_version zzzzzzh		
	44C	CSR_offset key 54h	CSR_offset 004000h		
	450	Logical_unit_characteristics 3Ah	Logical_unit_characteristics 400A08h		
	454	Logical_unit_number key 14h	reserved	device_type 02h	LUN 0000h
	458	Unit_dependent_dir_offset key D4h	Unit_dependent_directory_offset 000001h		
Unit Dependent Directory	45C	Unit_dep_dir_length 0004h	Unit_dep_dir_crc 9BAEh		
	460	Vender_offset key 81h	Vender key (unicode) offset 000004h		
	464	Vender_offset key 81h	Vender key (ascii) offset 000009h		
	468	Model_offset key 82h	Model key (unicode) offset 00000Dh		
	46C	Model_offset key 82h	Model key (ascii) offset 000030h		

*continued*

Block	offset	Description			
Vender Leaf (unicode)	470	Vender Leaf (Unicode) length 0005h		Vender Leaf CRC 8CCEh	
	474	Vender spec ID 80000000h			
	478	Vender language ID 00000409h			
	47C	'E' 4500h	'P' 5000h		
	480	'S' 5300h	'O' 4F00h		
	484	'N' 4E00h	00h	00h	
Vender Leaf (ascii)	488	Vender Leaf (ascii) length 0004h		Vender Leaf CRC 52DAh	
	48C	Vender spec ID 00000000h			
	490	Vender language ID 00000000h			
	494	'E' 45h	'P' 50h	'S' 53h	'O' 4Fh
	498	'N' 4Eh	00	00	00
Model Leaf (unicode)	49C	Mode leaf (unicode) length 0023h		Model leaf (unicode) CRC 5621h	
	4A0	Vender spec ID 80000000h			
	4A4	Vender language ID 00000409h			
	4A8	M	F		
	4AC	G	:		
	4B0	E	P		
	4B4	S	O		

*continued*

Block	offset	Description	
Model Leaf (unicode)	4b8	N	;
	4bc	C	M
	4c0	D	:
	4c4	E	S
	4c8	C	P
	4cc	L	2
	4d0	E	,
	4d4	P	R
	4d8	P	X
	4dc	L	,
	4e0	B	D
	4e4	C	;
	4e8	M	D
	4ec	L	:
	4f0	S	t
	4f4	y	l
	4f8	u	s
	4fc	2000h	C

*continued*

Block	offset	Description	
Model Leaf (unicode)	500	O	L
	504	O	R
	508	2000h	8
	50c	0	0
	510	;	C
	514	L	S
	518	:	P
	51c	R	I
	520	N	T
	524	E	R
	528	;	0000h



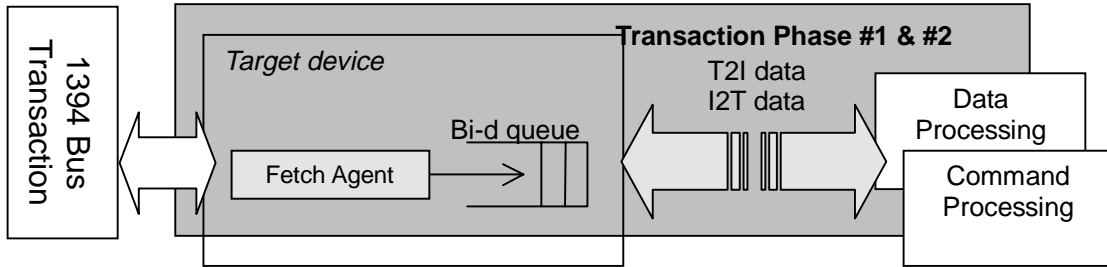
*continued*

Block	offset	Description			
Model Leaf (ascii)	52c	Model leaf (ascii) length	0013h	Model leaf (ascii) crc	C813h
	530	Model_spec_id 00000000h			
	534	Model_language_id 00000000h			
	538	M	F	G	:
	53c	E	P	S	O
	540	N	;	C	M
	544	D	:	E	S
	548	C	P	L	2
	54c	E	,	P	R
	550	P	X	L	,
	554	B	D	C	;
	558	M	D	L	:
	55c	S	t	y	I
	560	u	s	‘‘20h	C
	564	O	L	O	R
	568	‘‘20h	8	0	0
	56c	;	C	L	S
	570	:	P	R	I
	574	N	T	E	R
	578	;	00h	00h	00h

Textual descriptor:  
MFG:EPSON;CMD:ESCPL2E,PRPXL,BDC;MDL:Stylus COLOR 800;CLS:PRINTER;

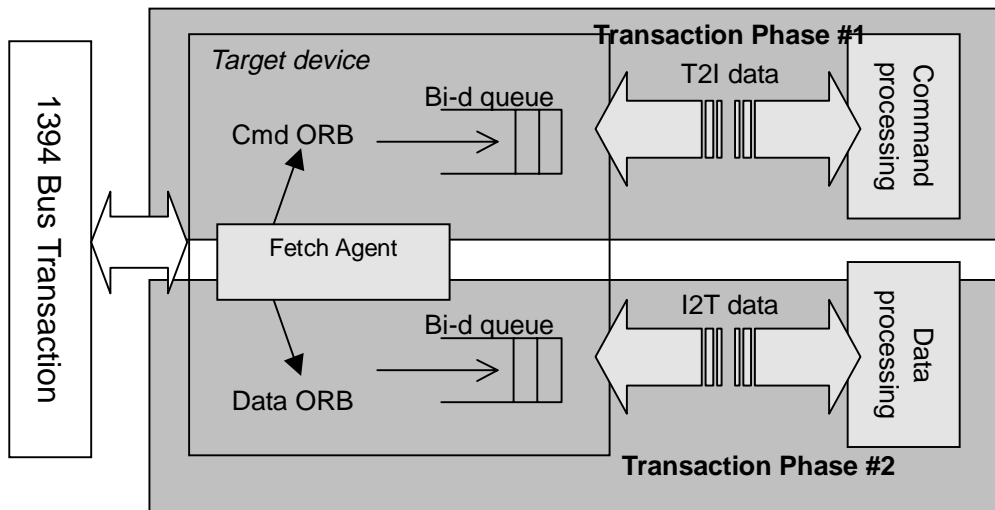
## Appendix B. Backward compatibility

Some consumer PC operating system under development is providing a SBP-2 protocol driver. It assumes that the target consumes ORBs in order and has only one queue. Thus there shall be conventional target devices which only has one bi-directional queue and consumes ORBs in order. One simple model in this type design is illustrated below.



**Figure 34 Target device which has one bi-directional queue**

In this case, a target device may implement time phases to fake the initiator as if the target has two independent bi-directional queues, one for data session and the other for command session.



**Figure 35 Two phases implementation of the command set**

However the initiator is required to issue ORBs for command session and ORBs for data session separately. To detect whether the target device is old implementation or not, the initiator may issue GET\_DEVICE\_ID command in resource session. (see 3.4.1 Resource Session).

< end of file >